

Oracle Database Interview Questions And Answers Guide.



Global Guideline.

<https://globalguideline.com/>



Oracle Database Job Interview Preparation Guide.

Question # 1

What Is Oracle?

Answer:-

Oracle is a company. Oracle is also a database server, which manages data in a very structured way. It allows users to store and retrieve related data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

[Read More Answers.](#)

Question # 2

What Is an Oracle Database?

Answer:-

An Oracle database is a collection of data treated as a big unit in the database server.

[Read More Answers.](#)

Question # 3

What Is an Oracle Instance?

Answer:-

Every running Oracle database is associated with an Oracle instance. When a database is started on a database server (regardless of the type of computer), Oracle allocates a memory area called the System Global Area (SGA) and starts one or more Oracle processes. This combination of the SGA and the Oracle processes is called an Oracle instance. The memory and processes of an instance manage the associated database's data efficiently and serve the one or multiple users of the database.

[Read More Answers.](#)

Question # 4

What Is a Parameter File in Oracle?

Answer:-

A parameter file is a file that contains a list of initialization parameters and a value for each parameter. You specify initialization parameters in a parameter file that reflect your particular installation. Oracle supports the following two types of parameter files:

- * Server Parameter Files - Binary version. Persistent.
- * Initialization Parameter Files - Text version. Not persistent.

[Read More Answers.](#)

Question # 5

What Is a Server Parameter File in Oracle?

Answer:-

A server parameter file is a binary file that acts as a repository for initialization parameters. The server parameter file can reside on the machine where the Oracle database server executes. Initialization parameters stored in a server parameter file are persistent, in that any changes made to the parameters while an instance is running can persist across instance shutdown and startup.

[Read More Answers.](#)

Question # 6

What Is a Initialization Parameter File in Oracle?

Answer:-

An initialization parameter file is a text file that contains a list of initialization parameters. The file should be written in the client's default character set. Sample initialization parameter files are provided on the Oracle distribution medium for each operating system. A sample file is sufficient for initial use, but you will probably want to modify the file to tune the database for best performance. Any changes will take effect after you completely shut down and restart the instance.



[Read More Answers.](#)

Question # 7

What is System Global Area (SGA) in Oracle?

Answer:-

The System Global Area (SGA) is a memory area that contains data shared between all database users such as buffer cache and a shared pool of SQL statements. The SGA is allocated in memory when an Oracle database instance is started, and any change in the value will take effect at the next startup.

[Read More Answers.](#)

Question # 8

What is Program Global Area (PGA) in Oracle?

Answer:-

A Program Global Area (PGA) is a memory buffer that is allocated for each individual database session and it contains session specific information such as SQL statement data or buffers used for sorting. The value specifies the total memory allocated by all sessions, and changes will take effect as new sessions are started.

[Read More Answers.](#)

Question # 9

What Is a User Account in Oracle?

Answer:-

A user account is identified by a user name and defines the user's attributes, including the following:

- * Password for database authentication
- * Privileges and roles
- * Default tablespace for database objects
- * Default temporary tablespace for query processing work space

[Read More Answers.](#)

Question # 10

What Is the Relation of a User Account and a Schema in Oracle?

Answer:-

User accounts and schemas have a one-to-one relation. When you create a user, you are also implicitly creating a schema for that user. A schema is a logical container for the database objects (such as tables, views, triggers, and so on) that the user creates. The schema name is the same as the user name, and can be used to unambiguously refer to objects owned by the user.

[Read More Answers.](#)

Question # 11

What Is a User Role in Oracle?

Answer:-

A user role is a group of privileges. Privileges are assigned to users through user roles. You create new roles, grant privileges to the roles, and then grant roles to users.

[Read More Answers.](#)

Question # 12

What is a Database Schema in Oracle?

Answer:-

A schema is a collection of logical structures of data, or schema objects. A schema is owned by a database user and has the same name as that user. Each user owns a single schema. Schema objects can be created and manipulated with SQL and include: tables, views, and other types of data objects.

[Read More Answers.](#)

Question # 13

What Is a Database Table in Oracle?

Answer:-

A database table is a basic unit of data logical storage in an Oracle database. Data is stored in rows and columns. You define a table with a table name, such as employees, and a set of columns. You give each column a column name, such as employee_id, last_name, and job_id; a datatype, such as VARCHAR2, DATE, or NUMBER; and a width. The width can be predetermined by the datatype, as in DATE. If columns are of the NUMBER datatype, define precision and scale instead of width. A row is a collection of column information corresponding to a single record.

[Read More Answers.](#)

Question # 14

What Is a Table Index in Oracle?

Answer:-

Index is an optional structure associated with a table that allow SQL statements to execute more quickly against a table. Just as the index in this manual helps you locate information faster than if there were no index, an Oracle Database index provides a faster access path to table data. You can use indexes without rewriting any queries. Your results are the same, but you see them more quickly.



[Read More Answers.](#)

Question # 15

What Is an Oracle Tablespace?

Answer:-

An Oracle tablespace is a big unit of logical storage in an Oracle database. It is managed and used by the Oracle server to store structures data objects, like tables and indexes.

[Read More Answers.](#)

Question # 16

What Is an Oracle Data File?

Answer:-

An Oracle data file is a big unit of physical storage in the OS file system. One or many Oracle data files are organized together to provide physical storage to a single Oracle tablespace.

[Read More Answers.](#)

Question # 17

What Is a Static Data Dictionary in Oracle?

Answer:-

Data dictionary tables are not directly accessible, but you can access information in them through data dictionary views. To list the data dictionary views available to you, query the view `DICTIONARY`. Many data dictionary tables have three corresponding views:

- * An `ALL_` view displays all the information accessible to the current user, including information from the current user's schema as well as information from objects in other schemas, if the current user has access to those objects by way of grants of privileges or roles.

- * A `DBA_` view displays all relevant information in the entire database. `DBA_` views are intended only for administrators. They can be accessed only by users with the `SELECT ANY TABLE` privilege. This privilege is assigned to the `DBA` role when the system is initially installed.

- * A `USER_` view displays all the information from the schema of the current user. No special privileges are required to query these views.

[Read More Answers.](#)

Question # 18

What Is a Dynamic Performance View in Oracle?

Answer:-

Oracle contains a set of underlying views that are maintained by the database server and accessible to the database administrator user `SYS`. These views are called dynamic performance views because they are continuously updated while a database is open and in use, and their contents relate primarily to performance. Although these views appear to be regular database tables, they are not. These views provide data on internal disk structures and memory structures. You can select from these views, but you can never update or alter them.

[Read More Answers.](#)

Question # 19

What Is a Recycle Bin in Oracle?

Answer:-

Recycle bin is a logical storage to hold the tables that have been dropped from the database, in case it was dropped in error. Tables in recycle bin can be recovered back into database by the Flashback Drop action. Oracle database recycle save the same purpose as the recycle bin on your Windows desktop.

Recycle bin can be turned on or off in the `recyclebin=on/off` in your parameter file.

[Read More Answers.](#)

Question # 20

What Is SQL*Plus?

Answer:-

SQL*Plus is an interactive and batch query tool that is installed with every Oracle Database Server or Client installation. It has a command-line user interface, a Windows Graphical User Interface (GUI) and the `iSQL*Plus` web-based user interface.

[Read More Answers.](#)

Question # 21

What Is Transport Network Substrate (TNS) in Oracle?

Answer:-

TNS, Transport Network Substrate, is a foundation technology, built into the Oracle Net foundation layer that works with any standard network transport protocol.

[Read More Answers.](#)

Question # 22

What Is Open Database Communication (ODBC) in Oracle?

Answer:-

ODBC, Open Database Communication, a standard API (application program interface) developed by Microsoft for Windows applications to communicate with database management systems.

Oracle offers ODBC drivers to allow Windows applications to connect Oracle server through ODBC.



[Read More Answers.](#)

Question # 23

What is Oracle Database 10g Express Edition?

Answer:-

Based on Oracle Web site: Oracle Database 10g Express Edition (Oracle Database XE) is an entry-level, small-footprint database based on the Oracle Database 10g Release 2 code base that's free to develop, deploy, and distribute; fast to download; and simple to administer. Oracle Database XE is a great starter database for:

- * Developers working on PHP, Java, .NET, and Open Source applications
- * DBAs who need a free, starter database for training and deployment
- * Independent Software Vendors (ISVs) and hardware vendors who want a starter database to distribute free of charge
- * Educational institutions and students who need a free database for their curriculum

[Read More Answers.](#)

Question # 24

What Are the Limitations Oracle Database 10g XE?

Answer:-

Oracle Database XE is free for runtime usage with the following limitations:

- * Supports up to 4GB of user data (in addition to Oracle system data)
- * Single instance only of Oracle Database XE on any server
- * May be installed on a multiple CPU server, but only executes on one processor in any server
- * May be installed on a server with any amount of memory, but will only use up to 1GB RAM of available memory

[Read More Answers.](#)

Question # 25

What Operating Systems Are Supported by Oracle Database 10g XE?

Answer:-

Oracle Database 10g Express Edition is available for two types of operating Systems:

- * Linux x86 - Debian, Mandriva, Novell, Red Hat and Ubuntu
- * Microsoft Windows

[Read More Answers.](#)

Question # 26

How To Download Oracle Database 10g XE?

Answer:-

If you want to download a copy of Oracle Database 10g Express Edition, visit <http://www.oracle.com/technology/software/products/database/xe/>.

If you are using Windows systems, there are downloads available for you:

* Oracle Database 10g Express Edition (Western European) - Single-byte LATIN1 database for Western European language storage, with the Database Homepage user interface in English only.

* Oracle Database 10g Express Edition (Universal) - Multi-byte Unicode database for all language deployment, with the Database Homepage user interface available in the following languages: Brazilian Portuguese, Chinese (Simplified and Traditional), English, French, German, Italian, Japanese, Korean and Spanish.

* Oracle Database 10g Express Client

You need to download the universal edition, OracleXEUniv.exe, (216,933,372 bytes) and client package, OracleXEClient.exe (30,943,220 bytes).

[Read More Answers.](#)

Question # 27

How To Install Oracle Database 10g XE?

Answer:-

To install 10g universal edition, double click, OracleXEUniv.exe, the install wizard starts. It will guide you to finish the installation process. You should take notes about:

- * The SYSTEM password you select: globalguideline.
- * Database server port: 1521.
- * Database HTTP port: 8080.
- * MS Transaction Server port: 2030.
- * The directory where 10g XE is installed: oraclexe
- * Hard disk space taken: 1655MB.

[Read More Answers.](#)

Question # 28

How To Check Your Oracle Database 10g XE Installation?

Answer:-

If you want to check your fresh installation of 10g Express Edition without using any special client programs, you can use a Web browser with this address, <http://localhost:8080/apex/>.

You will see the login page. Enter SYSTEM as the user name, and the password (globalguideline), you selected during the installation to log into the server.

Visit different areas on your 10g XE server home page to make sure your server is running OK.

You can also get to your 10g XE server home page by going through the start menu. Select All Programs, then Oracle Database 10g Express Edition, and then Go To Database Home Page

[Read More Answers.](#)



Question # 29

How To Shutdown Your 10g XE Server?

Answer:-

If you want to shutdown your 10g Express Edition server, go to the Services manager in the control panel. You will a service called OracleServiceXE, which represents your 10g Express Edition server.

Select OracleServiceXE, and use the right mouse click to stop this service. This will shutdown your 10g Express Edition server.

You can also shutdown your 10g XE server through the start menu. Select All Programs, then Oracle Database 10g Express Edition, and then Stop Database.

[Read More Answers.](#)

Question # 30

How To Start Your 10g XE Server?

Answer:-

Go to the Start menu, select All Programs, Oracle Database 10g Express Edition, and Start Database.

[Read More Answers.](#)

Question # 31

How Much Memory Your 10g XE Server Is Using?

Answer:-

Your 10g XE Server is using about 180MB of memory even there is no users on the server. The server memory usage is displayed on your server home page, if you log in as SYSTEM.

[Read More Answers.](#)

Question # 32

How To Start Your 10g XE Server from Command Line?

Answer:-

You can start your 10g XE server from command line by:

- * Open a command line window.
- * Change directory to oraclexeapporacleproduct10.2.0serverBIN.
- * Run StartDB.bat.

The batch file StartDB.bat contains:

```
net start OracleXETNSListener
net start OracleServiceXE
@oradim -startup -sid XE -startype inst > nul 2>&1
```

[Read More Answers.](#)

Question # 33

How To Shutdown Your 10g XE Server from Command Line?

Answer:-

You can shutdown your 10g XE server from command line by:

- * Open a command line window.
- * Change directory to oraclexeapporacleproduct10.2.0serverBIN.
- * Run StopDB.bat.

The batch file StopDB.bat contains:

```
net stop OracleServiceXE
```

[Read More Answers.](#)

Question # 34

How To Unlock the Sample User Account in Oracle?

Answer:-

Your 10g XE server comes with a sample database user account called HR. But this account is locked. You must unlock it before you can use it:

- * Log into the server home page as SYSTEM.
- * Click the Administration icon, and then click Database Users.
- * Click the HR schema icon to display the user information for HR.
- * Enter a new password (hr) for HR, and change the status to Unlocked.
- * Click Alter User to save the changes.

Now user account HR is ready to use.

[Read More Answers.](#)

Question # 35

How To Change System Global Area (SGA) in Oracle?

Answer:-

Your 10g XE server has a default setting for System Global Area (SGA) of 140MB. The SGA size can be changed to a new value depending on how many concurrent sessions connecting to your server. If you are running this server just for yourself to improve your DBA skill, you should change the SGA size to 32MB by:

- * Log into the server home page as SYSTEM.
- * Go to Administration, then Memory.
- * Click Configure SGA.
- * Enter the new memory size: 32



- * Click Apply Changes to save the changes.
- * Re-start your server.

[Read More Answers.](#)

Question # 36

How To Change Program Global Area (PGA) in Oracle?

Answer:-

Your 10g XE server has a default setting for Program Global Area (PGA) of 40MB. The PGA size can be changed to a new value depending on how much data a single session should be allocated. If you think your session will be short with a small amount of data, you should change the PGA size to 16MB by:

- * Log into the server home page as SYSTEM.
- * Go to Administration, then Memory.
- * Click Configure PGA.
- * Enter the new memory size: 16
- * Click Apply Changes to save the changes.
- * Re-start your server.

[Read More Answers.](#)

Question # 37

What Happens If You Set the SGA Too Low in Oracle?

Answer:-

Let's you made a mistake and changed to SGA to 16MB from the SYSTEM admin home page. When you run the batch file StartDB.bat, it will return a message saying server started. However, if you try to connect to your server home page: <http://localhost:8080/apex/>, you will get no response. Why? Your server is running, but the default instance XE was not started.

If you go the Control Panel and Services, you will see service OracleServiceXE is listed not in the running status.

[Read More Answers.](#)

Question # 38

What To Do If the StartBD.bat Failed to Start the XE Instance?

Answer:-

If StartBD.bat failed to start the XE instance, you need to try to start the instance with other approaches to get detail error messages on why the instance can not be started.

One good approach to start the default instance is to use SQL*Plus. Here is how to use SQL*Plus to start the default instance in a command window:

```
>cd (OracleXE home directory)
>. instartdb
>. insqlplus
```

```
Enter user-name: SYSTEM
Enter password: globalguideline
```

ERROR:

ORA-01034: ORACLE not available

ORA-27101: shared memory realm does not exist

The first "cd" is to move the current directory the 10g XE home directory. The second command ". instartdb" is to make sure the TNS listener is running. The third command ". insqlplus" launches SQL*Plus. The error message "ORA-27101" tells you that there is a memory problem with the default instance.

So you can not use the normal login process to the server without a good instance. See other tips on how to log into a server without any instance.

[Read More Answers.](#)

Question # 39

How To Login to the Server without an Instance?

Answer:-

If your default instance is in trouble, and you can not use the normal login process to reach the server, you can use a special login to log into the server without any instance. Here is how to use SQL*Plus to log in as a system BDA:

```
>cd (OracleXE home directory)
```

```
>. instartdb
```

```
>. insqlplus
```

```
Enter user-name: SYSTEM/globalguideline AS SYSDBA
```

```
Connected to an idle instance
```

```
SQL> show instance
```

```
instance "local"
```

The trick is to put user name, password and login options in a single string as the user name. "AS SYSDBA" tells the server to not start any instance, and connect the session the idle instance.

Log in as SYSDBA is very useful for performing DBA tasks.

[Read More Answers.](#)

Question # 40

How To Use "startup" Command to Start Default Instance?

Answer:-

If you logged in to the server as a SYSDBA, you start the default instance with the "startup" command. Here is how to start the default instance in SQL*Plus in SYSDBA mode:

```
>. insqlplus
```

```
Enter user-name: SYSTEM/globalguideline AS SYSDBA
```

```
Connected to an idle instance
```

```
SQL> show instance
```



instance "local"

```
SQL> startup
```

ORA-00821: Specified value of sga_target 16M is too small, needs to be at least 20M

Now the server is telling you more details about the memory problem on your default instance: your SGA setting of 16MB is too small. It must be increased to at least 20MB.

[Read More Answers.](#)

Question # 41

Where Are the Settings Stored for Each Instance in Oracle?

Answer:-

Settings for each instance are stored in a file called Server Parameter File (SPFile). Oracle supports two types of parameter files, Text type, and Binary type. parameter files should be located in \$ORACLE_HOME database directory. A parameter file should be named like "init\$SID.ora", where \$SID is the instance name.

[Read More Answers.](#)

Question # 42

What To Do If the Binary SPFile Is Wrong for the Default Instance?

Answer:-

Let's say the SPFile for the default instance is a binary file, and some settings are wrong in the SPFile, like SGA setting is below 20MB, how do you change a setting in the binary file? This seems to be a hard task, because the binary SPFile is not allowed to be edited manually. It needs to be updated by the server with instance started. But you can not start the instance because the SPFile has a wrong setting.

One way to solve the problem is to stop using the binary SPFile, and use a text version of the a parameter file to start the instance. Here is an example of how to use the backup copy (text version) of the parameter file for the default instance to start the instance:

```
>. insqlplus
```

```
Enter user-name: SYSTEM/globalguideline AS SYSDBA
```

```
Connected to an idle instance
```

[Read More Answers.](#)

Question # 43

How To Check the Server Version in Oracle?

Answer:-

Oracle server version information is stored in a table called: PRODUCT_COMPONENT_VERSION. You can use a simple SELECT statement to view the version information like this:

```
>. insqlplus
```

```
Enter user-name: SYSTEM/globalguideline AS SYSDBA
```

```
Connected to an idle instance
```

```
SQL> COL PRODUCT FORMAT A35
```

```
SQL> COL VERSION FORMAT A15
```

```
SQL> COL STATUS FORMAT A15
```

```
SQL> SELECT * FROM PRODUCT_COMPONENT_VERSION;
```

```
<pre>
```

PRODUCT	VERSION	STATUS
NLSRTL	10.2.0.1.0	Production
Oracle Database 10g Express Edition	10.2.0.1.0	Product
PL/SQL	10.2.0.1.0	Production
TNS for 32-bit Windows:	10.2.0.1.0	Production

[Read More Answers.](#)

Question # 44

Explain What Is SQL*Plus?

Answer:-

SQL*Plus is an interactive and batch query tool that is installed with every Oracle Database Server or Client installation. It has a command-line user interface, a Windows Graphical User Interface (GUI) and the iSQL*Plus web-based user interface.

SQL*Plus has its own commands and environment, and it provides access to the Oracle Database. It enables you to enter and execute SQL, PL/SQL, SQL*Plus and operating system commands to perform the following:

- * Format, perform calculations on, store, and print from query results

- * Examine table and object definitions

- * Develop and run batch scripts

- * Perform database administration

You can use SQL*Plus to generate reports interactively, to generate reports as batch processes, and to output the results to text file, to screen, or to HTML file for browsing on the Internet. You can generate reports dynamically using the HTML output facility of SQL*Plus, or using the dynamic reporting capability of iSQL*Plus to run a script from a web page.

[Read More Answers.](#)

Question # 45

How To Start the Command-Line SQL*Plus?

Answer:-

If you Oracle server or client installed on your windows system, you can start the command-line SQL*Plus in two ways:

1. Click Start > All Programs > Oracle ... > Start SQL Command Line. The SQL*Plus command window will show up with a message like this:

```
SQL*Plus: Release 10.2.0.1.0 - Production on Tue ...
```




Copyright (c) 1982, 2005, Oracle. All rights reserved.

```
SQL>
2. Click Start > Run..., enter "cmd" and click OK. A Windows command window will show up. You can then use Windows commands to start the command-line
SQL*Plus as shown in the tutorial exercise below:
>cd c:\oracle\app\oracleproduct10.2.0\server
>. insqlplus /nolog
SQL*Plus: Release 10.2.0.1.0 - Production on Tue ...
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

[Read More Answers.](#)

Question # 46

How To Get Help at the SQL Prompt?

Answer:-

Once SQL*Plus is started, you will get a SQL prompt like this: SQL>. This where you can enter commands for SQL*Plus to run. To get help information at the SQL prompt, you can use the HELP command as shown in the following tutorial example:

```
SQL> HELP INDEX
Enter Help [topic] for help.


```
<pre> @ COPY PAUSE SHUTDOWN
@@ DEFINE PRINT SPOOL
/ DEL PROMPT SQLPLUS
ACCEPT DESCRIBE QUIT START
APPEND DISCONNECT RECOVER STARTUP
ARCHIVE LOG EDIT REMARK STORE
ATTRIBUTE EXECUTE REPFOOTER TIMING
BREAK EXIT REPHEADER TTITLE
...
COMPUTE LIST SET XQUERY
CONNECT PASSWORD SHOW</pre>
SQL> HELP CONNECT
CONNECT

```


```

[Read More Answers.](#)

Question # 47

What Information Is Needed to Connect SQL*Plus an Oracle Server?

Answer:-

If you want to connect your SQL*Plus session to an Oracle server, you need to know the following information about this server:

- * The network hostname, or IP address, of the Oracle server.
- * The network port number where the Oracle server is listening for incoming connections.
- * The name of the target database instance managed by the Oracle server.
- * The name of your user account predefined on in the target database instance.
- * The password of your user account predefined on in the target database instance.

[Read More Answers.](#)

Question # 48

What Is a Connect Identifier?

Answer:-

A "connect identifier" is an identification string of a single set of connection information to a specific target database instance on a specific Oracle server. Connect identifiers are defined and stored in a file called tnsnames.ora located in \$ORACLE_HOME/network/admin/ directory. Here is one example of a "connect identifier" definition:

```
ggl_XE =
(DESCRIPTION =
  (ADDRESS =
    (PROTOCOL = TCP)
    (HOST = www.globalguideline.com)
    (PORT = 1521)
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)
```

The above "connect identifier" defines "TNS_XE" with the following connection information:

- * The network hostname: www.globalguideline.com.
- * The network port number: 1521.
- * The name of the target database instance: XE.

[Read More Answers.](#)

Question # 49

How To Connect a SQL*Plus Session to an Oracle Server?

Answer:-

In order to connect a SQL*Plus session to an Oracle server, you need to:

1. Obtain the connection information from the Oracle server DBA.



2. Define a new "connect identifier" called "ggl_XE" in your tnsnames.org file with the given connection information.
3. Run the CONNECT command in SQL*Plus as shown in the tutorial exercise below:

```
>cd c:\oracle\app\oracle\product10.2.0\server
>. insqlplus /nolog
SQL*Plus: Release 10.2.0.1.0 - Production on Tue ...
Copyright (c) 1982, 2005, Oracle. All rights reserved.
SQL> CONNECT ggl/retneclgg@ggl_XE;
Connected.
SQL> SELECT SYSDATE FROM DUAL;
SYSDATE
-----
05-MAR-06
```

[Read More Answers.](#)

Question # 50

What Happens If You Use a Wrong Connect Identifier?

Answer:-

Of course, you will get an error, if you use a wrong connect identifier. Here is an example of how SQL*Plus react to a wrong connect identifier:

```
SQL> CONNECT ggl/retneclgg@WRONG;
ERROR:
ORA-12154: TNS:could not resolve the connect identifier
specified
```

Warning: You are no longer connected to ORACLE.

What you need to do in this case:

- * Check the CONNECT command to make sure that the connect identifier is entered correctly.
- * Check the tnsnames.ora file to make sure that the connect identifier is defined correctly.
- * Check the tnsnames.ora file to make sure that there is no multiple definitions of the same connect identifier.
- * Check your files system to see if you have multiple copies of tnsnames.ora in different Oracle home directories, because you installed multiple versions of Oracle. If you do have multiple copies, make sure your SQL*Plus session is picking up the correct copy of tnsnames.ora.

[Read More Answers.](#)

Question # 51

What To Do If DBA Lost the SYSTEM Password?

Answer:-

If the DBA lost the password of the SYSTEM user account, he/she can go to the Oracle server machine, and run SQL*Plus on server locally with the operating system authentication method to gain access to the database. The tutorial exercise below shows you how:

(Terminal server to the Oracle server machine)

(Start SQL*Plus)

```
SQL>CONNECT / AS SYSDBA
```

Connected.

```
SQL> ALTER USER SYSTEM IDENTIFIED BY ssap_lgg;
```

User altered.

Notice that the (/) in the CONNECT command tells SQL*Plus to use the current user on local operating system as the connection authentication method.

[Read More Answers.](#)

Question # 52

What Types of Commands Can Be Executed in SQL*Plus?

Answer:-

There are 4 types of commands you can run at the SQL*Plus command line prompt:

1. SQL commands - Standard SQL statements to be executed on target database on the Oracle server. For example: "SELECT * FROM ggl_faq;" is a SQL command.
2. PL/SQL commands - PL/SQL statements to be executed by the Oracle server. For example: "EXECUTE DBMS_OUTPUT.PUT_LINE('Welcome to www.globalguideline.com');" runs a PL/SQL command.
- SQL*Plus commands - Commands to be executed by the local SQL*Plus program itself. For example: "SET NULL 'NULL'" is a SQL*Plus command.
- OS commands - Commands to be executed by the local operating system. For example: "HOST dir" runs an operating system command on the local machine.

[Read More Answers.](#)

Question # 53

How To Run SQL Commands in SQL*Plus?

Answer:-

If you want to run a SQL command in SQL*Plus, you need to enter the SQL command in one or more lines and terminated with (;). The tutorial exercise below shows a good example:

```
SQL> SELECT 'Welcome!' FROM DUAL;
'WELCOME
-----
Welcome!
SQL> SELECT 'Welcome to globalguideline.com tutorials!'
2 FROM DUAL
3 ;
'WELCOMETOglobalguideline.COMTUTORIALS!'
```

Welcome to globalguideline.com tutorials!

[Read More Answers.](#)



Question # 54

How To Run PL/SQL Statements in SQL*Plus?

Answer:-

If you want to run a single PL/SQL statement in SQL*Plus, you need to use the EXECUTE command as shown in the following tutorial example:

```
SQL> SET SERVEROUTPUT ON
SQL> EXECUTE DBMS_OUTPUT.PUT_LINE('Welcome to globalguideline!')
Welcome to globalguideline!
PL/SQL procedure successfully completed.
```

[Read More Answers.](#)

Question # 55

How To Change SQL*Plus System Settings?

Answer:-

SQL*Plus environment is controlled a big list of SQL*Plus system settings. You can change them by using the SET command as shown in the following list:

- * SET AUTOCOMMIT OFF - Turns off the auto-commit feature.
- * SET FEEDBACK OFF - Stops displaying the "27 rows selected." message at the end of the query output.
- * SET HEADING OFF - Stops displaying the header line of the query output.
- * SET LINESIZE 256 - Sets the number of characters per line when displaying the query output.
- * SET NEWPAGE 2 - Sets 2 blank lines to be displayed on each page of the query output.
- * SET NEWPAGE NONE - Sets for no blank lines to be displayed on each page of the query output.
- * SET NULL 'null' - Asks SQL*Plus to display 'null' for columns that have null values in the query output.
- * SET PAGESIZE 60 - Sets the number of lines per page when displaying the query output.
- * SET TIMING ON - Asks SQL*Plus to display the command execution timing data.
- * SET WRAP OFF - Turns off the wrapping feature when displaying query output.

[Read More Answers.](#)

Question # 56

How To Look at the Current SQL*Plus System Settings?

Answer:-

If you want to see the current values of SQL*Plus system settings, you can use the SHOW command as shown in the following tutorial exercise:

```
SQL> SHOW AUTOCOMMIT
autocommit OFF
SQL> SHOW HEADING
heading ON
SQL> SHOW LINESIZE
linesize 80
SQL> SHOW PAGESIZE
pagesize 14
SQL> SHOW FEEDBACK
FEEDBACK ON for 6 or more rows
SQL> SHOW TIMING
timing OFF
SQL> SHOW NULL
null ""
SQL> SHOW ALL
appinfo is OFF and set to "SQL*Plus"
arraysize 15
autocommit OFF
autoprint OFF
autorecovery OFF
autotrace OFF
blockterminator "." (hex 2e)
cmdsep OFF
colsep " "
compatibility version NATIVE
concat "." (hex 2e)
copycommit 0
COPYTYPECHECK is ON
define "&" (hex 26)
describe DEPTH 1 LINENUM OFF INDENT ON
echo OFF
```

[Read More Answers.](#)

Question # 57

What Are SQL*Plus Environment Variables?

Answer:-

Behaviors of SQL*Plus are also controlled a some environment variables predefined on the local operating system. Here are some commonly used SQL*Plus environment variables:

- * ORACLE_HOME - The home directory where your Oracle client application is installed.
- * PATH - A list of directories where SQL*Plus will search for executable or DLL files. PATH should include \$ORACLE_HOME in.
- * SQLPLUS - The directory where localization messages are stored. SQLPLUS should be set to \$ORACLE_HOME/sqlplusmsg
- * TNS_ADMIN - The directory where the connect identifier file, tnsnames.ora is located. TNS_ADMIN should be set to \$ORACLE_HOME/network/admin.

[Read More Answers.](#)



Question # 58

How To Generate Query Output in HTML Format?

Answer:-

If you want your query output to be generated in HTML format, you can use the "SET MARKUP HTML ON" to turn on the HTML feature. The following tutorial exercise gives you a good example:

```
SQL> connect HR/retneclgg
SQL> SET MARKUP HTML ON
SQL> SELECT FIRST_NAME, LAST_NAME, HIRE_DATE
<br>
  2 FROM EMPLOYEES WHERE FIRST_NAME LIKE 'Joh%';
<br>
<p>
<table border='1' width='90%' align='center' summary='Script output'>
<tr>
<th scope="col">
FIRST_NAME
</th>
<th scope="col">
LAST_NAME
</th>
<th scope="col">
HIRE_DATE
</th>
</tr>
<tr>
<td>
John
</td>
<td>
Seo
</td>
<td>
12-FEB-98
</td>
</tr>
<tr>
<td>
John
</td>
<td>
Russell
</td>
<td>
01-OCT-96
</td>
</tr>
</table>
<p>
```

[Read More Answers.](#)

Question # 59

What Is Output Spooling in SQL*Plus?

Answer:-

The output spooling a nice feature of the command-line SQL*Plus tool. If the spooling feature is turned on, SQL*Plus will send a carbon copy of the everything on your screen to a specified local file.

Output spooling is used mostly for quick dump of data to local files. Here are the commands to turn on and off output spooling in SQL*Plus:

- * SPOOL fileName - Turning on output spooling with the specified file.
- * SPOOL OFF - Turning off output spooling and close the spool file.

[Read More Answers.](#)

Question # 60

How To Save Query Output to a Local File?

Answer:-

Normally, when you run a SELECT statement in SQL*Plus, the output will be displayed on your screen. If you want the output to be saved to local file, you can use the "SPOOL fileName" command to specify a local file and start the spooling feature. When you are done with your SELECT statement, you need to close the spool file with the "SPOOL OFF" command. The following tutorial exercise gives you a good example:

```
SQL> connect HR/retneclgg
SQL> SET HEADING OFF
SQL> SET FEEDBACK OFF
SQL> SET LINESIZE 1000
SQL> SPOOL empemployees.lst
SQL> SELECT * FROM EMPLOYEES;
.....
SQL> SPOOL OFF
```

You should get all records in employees.lst with fixed length fields.

[Read More Answers.](#)



Question # 61

What Is Input Buffer in SQL*Plus?

Answer:-

Input buffer is a nice feature of the command-line SQL*Plus tool. It allows you to revise a multiple-line command and re-run it with a couple of simple commands. By default, input buffer is always turned on in SQL*Plus. The last SQL statement is always stored in the buffer. All you need is to remember to following commonly used commands:

- * LIST - Displays the SQL statement (the last executed SQL statement) in the buffer.
- * RUN - Runs the SQL statement in the buffer again. ";" is a quick command equivalent to RUN.
- * CLEAR BUFFER - Removes the SQL statement in the buffer.
- * INPUT line - Adds a new line into the buffer.
- * APPEND text - Appends more text to the last line in the buffer.
- * DEL - Deletes one line from the buffer.
- * CHANGE /old/new - Replaces 'old' text with 'new' text in the buffer.

[Read More Answers.](#)

Question # 62

How To Revise and Re-Run the Last SQL Command?

Answer:-

If executed a long SQL statement, found a mistake in the statement, and you don't want enter that long statement again, you can use the input buffer commands to the correct last statement and re-run it. The following tutorial exercise gives you a good example:

```
SQL> connect HR/retneclgg
SQL> SELECT FIRST_NAME, LAST_NAME, HIRE_DATE
  2 FROM EMPLOYEE WHERE FIRST_NAME LIKE 'Joh%';
FROM EMPLOYEE WHERE FIRST_NAME LIKE 'Joh%'
*
```

ERROR at line 2:

ORA-00942: table or view does not exist

```
SQL> LIST
```

```
  1 SELECT FIRST_NAME, LAST_NAME, HIRE_DATE
  2* FROM EMPLOYEES WHERE FIRST_NAME LIKE 'Joh%'
```

```
SQL> CHANGE /EMPLOYEE/EMPLOYEES/
  2* FROM EMPLOYEES WHERE FIRST_NAME LIKE 'Joh%'
```

```
SQL> RUN
```

(Query output)

```
SQL> INPUT ORDER BY FIRE_DATE
```

```
SQL> LIST
```

```
  1 SELECT FIRST_NAME, LAST_NAME, HIRE_DATE
  2 FROM EMPLOYEE WHERE FIRST_NAME LIKE 'Joh%'
  3* ORDER BY HIRE_DATE
```

```
SQL> RUN
```

(Query output)

```
SQL> CLEAR BUFFER
```

buffer cleared

```
SQL> LIST
```

SP2-0223: No lines in SQL buffer.

[Read More Answers.](#)

Question # 63

How Run SQL*Plus Commands That Are Stored in a Local File?

Answer:-

If you have a group of commands that you need to run them repeatedly every day, you can save those commands in a file (called SQL script file), and using the "@fileName" command to run them in SQL*Plus. If you want to try this, create a file called empinput.sql with:

```
SELECT 'Welcome to' FROM DUAL;
SELECT 'globalguideline.com!' FROM DUAL;
```

Then run the "@" command in SQL*Plus as:

```
SQL> connect HR/retneclgg
```

```
SQL> @ empinput.sql
```

```
'WELCOMETO
```

```
-----
```

```
Welcome to
'globalguideline.COM
```

```
-----
```

```
globalguideline.com!
```

[Read More Answers.](#)

Question # 64

How To Use SQL*Plus Built-in Timers?

Answer:-

If you don't have a stopwatch/timer and want to measure elapsed periods of time, you can SQL*Plus Built-in Timers with the following commands:

- * TIMING - Displays number of timers.
- * TIMING START [name] - Starts a new timer with or without a name.
- * TIMING SHOW [name] - Shows the current time of the named or not-named timer.
- * TIMING STOP [name] - Stops the named or not-named timer.

The following tutorial exercise shows you a good example of using SQL*Plus built-in timers:



```
SQL> TIMING START timer_1
(some seconds later)
SQL> TIMING START timer_2
(some seconds later)
SQL> TIMING START timer_3
(some seconds later)
SQL> TIMING SHOW timer_1
timing for: timer_2
Elapsed: 00:00:19.43
(some seconds later)
SQL> TIMING STOP timer_2
timing for: timer_2
Elapsed: 00:00:36.32
SQL> TIMING
2 timing elements in use
```

[Read More Answers.](#)

Question # 65

What Is Oracle Server Autotrace in Oracle?

Answer:-

Autotrace is Oracle server feature that generates two statement execution reports very useful for performance tuning:

- * Statement execution path - Shows you the execution loop logic of a DML statement.
- * Statement execution statistics - Shows you various execution statistics of a DML statement.

To turn on the autotrace feature, the Oracle server DBA need to:

- * Create a special table called PLAN_TABLE.
- * Create a special security role called PLUSTRACE.
- * Grant PLUSTRACE role your user account.

[Read More Answers.](#)

Question # 66

How To Set Up Autotrace for a User Account?

Answer:-

If an Oracle user wants to use the autotrace feature, you can use the tutorial as an example to create the required table PLAN_TABLE, the required security role PLUSTRACE, and grant the role to that user:

```
SQL> CONNECT HR/retneclgg
SQL> @oraclexeapporacleproduct10.2.0server
RDBMSADMINUTLXPLAN.SQL
Table (HR.PLAN_TABLE) created.
SQL> CONNECT / AS SYSDBA
SQL> @C:\oraclexeapporacleproduct10.2.0server
SQLPLUSADMINPLUSTRCE.SQL
SQL> drop role plustrace;
Role (PLUSTRACE) dropped.
```

```
SQL> create role plustrace;
Role (PLUSTRACE) created.
```

```
SQL> grant plustrace to dba with admin option;
Grant succeeded.
```

```
SQL> GRANT PLUSTRACE TO HR;
Grant succeeded.
```

Remember that PLAN_TABLE table must be created under the user schema HR.

[Read More Answers.](#)

Question # 67

How To Get Execution Path Reports on Query Statements?

Answer:-

If your user account has autotrace configured by the DBA, you can use the "SET AUTOTRACE ON EXPLAIN" command to turn on execution path reports on query statements. The tutorial exercise bellow shows you a good example:

```
SQL> CONNECT HR/retneclgg
SQL> SET AUTOTRACE ON EXPLAIN
SQL> SELECT E.LAST_NAME, E.SALARY, J.JOB_TITLE
 2 FROM EMPLOYEES E, JOBS J
 3 WHERE E.JOB_ID=J.JOB_ID AND E.SALARY>12000;


```
<pre>LAST_NAME SALARY JOB_TITLE
```


```

```
-----
King                24000 President
Kochhar              17000 Administration Vice President
De Haan              17000 Administration Vice President
Russell              14000 Sales Manager
Partners             13500 Sales Manager
Hartstein            13000 Marketing Manager
6 rows selected.
```

[Read More Answers.](#)



Question # 68

How To Get Execution Statistics Reports on Query Statements?

Answer:-

If your user account has autotrace configured by the DBA, you can use the "SET AUTOTRACE ON STATISTICS" command to turn on execution statistics reports on query statements. The tutorial exercise below shows you a good example:

```
SQL> CONNECT HR/retnec1gg
SQL> SET AUTOTRACE ON STATISTICS
SQL> SELECT E.LAST_NAME, E.SALARY, J.JOB_TITLE
  2 FROM EMPLOYEES E, JOBS J
  3 WHERE E.JOB_ID=J.JOB_ID AND E.SALARY>12000;


```
LAST_NAME SALARY JOB_TITLE

```


```

```
King           24000 President
Kochhar        17000 Administration Vice President
De Haan        17000 Administration Vice President
Russell        14000 Sales Manager
Partners       13500 Sales Manager
Hartstein      13000 Marketing Manager
```

6 rows selected.

[Read More Answers.](#)

Question # 69

What Is SQL in Oracle?

Answer:-

SQL, SEQUEL (Structured English Query Language), is a language for RDBMS (Relational Database Management Systems). SQL was developed by IBM Corporation.

[Read More Answers.](#)

Question # 70

How Many Categories of Data Types in Oracle?

Answer:-

Oracles supports the following categories of data types:

- * Oracle Built-in Datatypes.
- * ANSI, DB2, and SQL/DS Datatypes.
- * User-Defined Types.
- * Oracle-Supplied Types.

[Read More Answers.](#)

Question # 71

What Are the Oracle Built-in Data Types?

Answer:-

There are 20 Oracle built-in data types, divided into 6 groups:

- * Character Datatypes - CHAR, NCHAR, NVARCHAR2, VARCHAR2
- * Number Datatypes - NUMBER, BINARY_FLOAT, BINARY_DOUBLE
- * Long and Row Datatypes - LONG, LONG RAW, RAW
- * Datetime Datatypes - DATE, TIMESTAMP, INTERVAL YEAR TO MONTH, INTERVAL DAY TO SECOND
- * Large Object Datatypes - BLOB, CLOB, NCLOB, BFILE
- * Row ID Datatypes - ROWID, UROWID

[Read More Answers.](#)

Question # 72

What Are the Differences between CHAR and NCHAR in Oracle?

Answer:-

Both CHAR and NCHAR are fixed length character data types. But they have the following differences:

- * CHAR's size is specified in bytes by default.
- * NCHAR's size is specified in characters by default. A character could be 1 byte to 4 bytes long depending on the character set used.
- * NCHAR stores characters in Unicode.

[Read More Answers.](#)

Question # 73

What Are the Differences between CHAR and VARCHAR2 in Oracle?

Answer:-

The main differences between CHAR and VARCHAR2 are:

- * CHAR stores values in fixed lengths. Values are padded with space characters to match the specified length.
- * VARCHAR2 stores values in variable lengths. Values are not padded with any characters.

[Read More Answers.](#)

Question # 74



What Are the Differences between NUMBER and BINARY_FLOAT in Oracle?

Answer:-

The main differences between NUMBER and BINARY_FLOAT in Oracle are:

- * NUMBER stores values as fixed-point numbers using 1 to 22 bytes.
- * BINARY_FLOAT stores values as single precision floating-point numbers.

[Read More Answers.](#)

Question # 75

What Are the Differences between DATE and TIMESTAMP in Oracle?

Answer:-

The main differences between DATE and TIMESTAMP in Oracle are:

- * DATE stores values as century, year, month, date, hour, minute, and second.
- * TIMESTAMP stores values as year, month, day, hour, minute, second, and fractional seconds.

[Read More Answers.](#)

Question # 76

What Are the Differences between INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND?

Answer:-

The main differences between INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND are:

- * INTERVAL YEAR TO MONTH stores values as time intervals at the month level.
- * INTERVAL DAY TO SECOND stores values as time intervals at the fractional seconds level.

[Read More Answers.](#)

Question # 77

What Are the Differences between BLOB and CLOB in Oracle?

Answer:-

The main differences between BLOB and CLOB in Oracle are:

- * BLOB stores values as LOB (Large Object) in bitstreams.
- * CLOB stores values as LOB (Large Object) in character streams.

[Read More Answers.](#)

Question # 78

What Are the ANSI Data Types Supported in Oracle?

Answer:-

The following ANSI data types are supported in Oracle:

- * CHARACTER(n) / CHAR(n)
- * CHARACTER VARYING(n) / CHAR VARYING(n)
- * NATIONAL CHARACTER(n) / NATIONAL CHAR(n) / NCHAR(n)
- * NATIONAL CHARACTER VARYING(n) / NATIONAL CHAR VARYING(n) / NCHAR VARYING(n)
- * NUMERIC(p,s)
- * DECIMAL(p,s)
- * INTEGER / INT
- * SMALLINT
- * FLOAT
- * DOUBLE PRECISION
- * REAL

[Read More Answers.](#)

Question # 79

How To Write Text Literals in Oracle?

Answer:-

There are several ways to write text literals as shown in the following samples:

```
SELECT 'globalguideline.com' FROM DUAL -- The most common format  
globalguideline.com
```

```
SELECT 'It's Sunday!' FROM DUAL -- Single quote escaped  
It's Sunday!
```

```
SELECT N'Allo, C'est moi.' FROM DUAL -- National chars  
Allo, C'est moi.
```

```
SELECT Q'It's Sunday!/' FROM DUAL -- Your own delimiter  
It's Sunday!
```

[Read More Answers.](#)

Question # 80

How To Write Numeric Literals in Oracle?

Answer:-

Numeric literals can be coded as shown in the following samples:

```
SELECT 255 FROM DUAL -- An integer  
255
```




```
SELECT -6.34 FROM DUAL -- A regular number
-6.34
SELECT 2.14F FROM DUAL -- A single-precision floating point
2.14
SELECT -0.5D FROM DUAL -- A double-precision floating point
-0.5
```

[Read More Answers.](#)

Question # 81

How To Write Date and Time Literals in Oracle?

Answer:-

Date and time literals can coded as shown in the following samples:
SELECT DATE '2002-10-03' FROM DUAL -- ANSI date format
03-OCT-07
SELECT TIMESTAMP '0227-01-31 09:26:50.124' FROM DUAL
31-JAN-07 09.26.50.124000000 AM
-- This is ANSI format

[Read More Answers.](#)

Question # 82

How To Write Date and Time Interval Literals in Oracle?

Answer:-

Date and time interval literals can coded as shown in the following samples:
SELECT DATE '2002-10-03' + INTERVAL '123-2' YEAR(3) TO MONTH
FROM DUAL
-- 123 years and 2 months is added to 2002-10-03
03-DEC-25
SELECT DATE '2002-10-03' + INTERVAL '123' YEAR(3) FROM DUAL
-- 123 years is added to 2002-10-03
03-OCT-25
SELECT DATE '2002-10-03' + INTERVAL '299' MONTH(3) FROM DUAL
-- 299 months years is added to 2002-10-03
03-SEP-27
SELECT TIMESTAMP '1997-01-31 09:26:50.124'
+ INTERVAL '4 5:12:10.222' DAY TO SECOND(3) FROM DUAL
04-FEB-97 02.39.00.346000000 PM
SELECT TIMESTAMP '1997-01-31 09:26:50.124'
+ INTERVAL '4 5:12' DAY TO MINUTE FROM DUAL
04-FEB-97 02.38.50.124000000 PM
SELECT TIMESTAMP '1997-01-31 09:26:50.124'
+ INTERVAL '400 5' DAY(3) TO HOUR FROM DUAL
07-MAR-98 02.26.50.124000000 PM
SELECT TIMESTAMP '1997-01-31 09:26:50.124'
+ INTERVAL '400' DAY(3) FROM DUAL
07-MAR-98 09.26.50.124000000 AM
SELECT TIMESTAMP '1997-01-31 09:26:50.124'
+ INTERVAL '11:12:10.2222222' HOUR TO SECOND(7) FROM DUAL
31-JAN-97 08.39.00.346222200 PM

[Read More Answers.](#)

Question # 83

How To Convert Numbers to Characters in Oracle?

Answer:-

You can convert numeric values to characters by using the TO_CHAR() function as shown in the following examples:
SELECT TO_CHAR(4123.4570) FROM DUAL
123.457
SELECT TO_CHAR(4123.457, '\$9,999,999.99') FROM DUAL
\$4,123.46
SELECT TO_CHAR(-4123.457, '9999999.99EEEE') FROM DUAL
-4.12E+03

[Read More Answers.](#)

Question # 84

How To Convert Characters to Numbers in Oracle?

Answer:-

You can convert characters to numbers by using the TO_NUMBER() function as shown in the following examples:
SELECT TO_NUMBER('4123.4570') FROM DUAL
4123.457
SELECT TO_NUMBER(' \$4,123.46', '\$9,999,999.99') FROM DUAL
4123.46
SELECT TO_NUMBER(' -4.12E+03') FROM DUAL
-4120

[Read More Answers.](#)



Question # 85

How To Convert Dates to Characters in Oracle?

Answer:-

You can convert dates to characters using the TO_CHAR() function as shown in the following examples:

```
SELECT TO_CHAR(SYSDATE, 'DD-MON-YYYY') FROM DUAL;
-- SYSDATE returns the current date
07-MAY-2006
SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD') FROM DUAL;
2006/05/07
SELECT TO_CHAR(SYSDATE, 'MONTH DD, YYYY') FROM DUAL;
MAY 07, 2006
SELECT TO_CHAR(SYSDATE, 'fmMONTH DD, YYYY') FROM DUAL;
May 7, 2006
SELECT TO_CHAR(SYSDATE, 'fmDAY, MONTH DD, YYYY') FROM DUAL;
SUNDAY, MAY 7, 2006
```

[Read More Answers.](#)

Question # 86

How To Convert Characters to Dates in Oracle?

Answer:-

You can convert dates to characters using the TO_DATE() function as shown in the following examples:

```
SELECT TO_DATE('07-MAY-2006', 'DD-MON-YYYY') FROM DUAL;
07-MAY-06
SELECT TO_DATE('2006/05/07', 'YYYY/MM/DD') FROM DUAL;
07-MAY-06
SELECT TO_DATE('MAY 07, 2006', 'MONTH DD, YYYY')
FROM DUAL;
07-MAY-06
SELECT TO_DATE('May 7, 2006', 'fmMONTH DD, YYYY') FROM DUAL;
07-MAY-06
SELECT TO_DATE('SUNDAY, MAY 7, 2006',
'fmDAY, MONTH DD, YYYY') FROM DUAL;
07-MAY-06
```

[Read More Answers.](#)

Question # 87

How To Convert Times to Characters in Oracle?

Answer:-

You can convert dates to characters using the TO_CHAR() function as shown in the following examples:

```
SELECT TO_CHAR(SYSDATE, 'HH:MI:SS') FROM DUAL;
04:49:49
SELECT TO_CHAR(SYSDATE, 'HH24:MI:SS.FF') FROM DUAL;
-- Error: SYSDATE has no fractional seconds

SELECT TO_CHAR(SYSTIMESTAMP, 'HH24:MI:SS.FF9') FROM DUAL;
16:52:57.847000000
SELECT TO_CHAR(SYSDATE, 'SSSS') FROM DUAL;
-- Seconds past midnight
69520
```

[Read More Answers.](#)

Question # 88

How To Convert Characters to Times in Oracle?

Answer:-

You can convert dates to characters using the TO_CHAR() function as shown in the following examples:

```
SELECT TO_CHAR(TO_DATE('04:49:49', 'HH:MI:SS'),
'DD-MON-YYYY HH24:MI:SS') FROM DUAL;
-- Default date is the first day of the current month
01-MAY-2006 04:49:49
SELECT TO_CHAR(TO_TIMESTAMP('16:52:57.847000000',
'HH24:MI:SS.FF9'), 'DD-MON-YYYY HH24:MI:SS.FF9')
FROM DUAL;
01-MAY-2006 16:52:57.847000000
SELECT TO_CHAR(TO_DATE('69520', 'SSSS'),
'DD-MON-YYYY HH24:MI:SS') FROM DUAL;
01-MAY-2006 19:18:40
```

[Read More Answers.](#)

Question # 89

What Is NULL value in Oracle?

Answer:-

NULL is a special value representing "no value" in all data types. NULL can be used on in operations like other values. But most operations has special rules when



NULL is involved. The tutorial exercise below shows you some examples:

```
SET NULL 'NULL'; -- Make sure NULL is displayed
SELECT NULL FROM DUAL;
```

N

-

N

U

L

L

```
SELECT NULL + NULL FROM DUAL;
```

NULL+NULL

NULL

```
SELECT NULL + 7 FROM DUAL;
```

NULL+7

NULL

```
SELECT NULL * 7 FROM DUAL;
```

NULL*7

NULL

```
SELECT NULL || 'A' FROM DUAL;
```

N

-

A

```
SELECT NULL + SYSDATE FROM DUAL;
```

NULL+SYSD

NULL

[Read More Answers.](#)

Question # 90

How To Use NULL as Conditions in Oracle?

Answer:-

If you want to compare values against NULL as conditions, you should use the "IS NULL" or "IS NOT NULL" operator. Do not use "=" or "<>" against NULL. The sample script below shows you some good examples:

```
SELECT 'A' IS NULL FROM DUAL;
```

-- Error: Boolean is not data type.

-- Boolean can only be used as conditions

```
SELECT CASE WHEN 'A' IS NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

FALSE

```
SELECT CASE WHEN '' IS NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

TRUE

```
SELECT CASE WHEN 0 IS NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

FALSE

```
SELECT CASE WHEN NULL IS NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

TRUE

```
SELECT CASE WHEN 'A' = NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

-- Do not use "="

FALSE

```
SELECT CASE WHEN 'A' <> NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

-- Do not use "<>"

FALSE

```
SELECT CASE WHEN NULL = NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

-- Do not use "="

FALSE

FALSE

```
SELECT CASE WHEN NULL = NULL THEN 'TRUE' ELSE 'FALSE' END
FROM DUAL;
```

-- Do not use "="

FALSE

FALSE

FALSE

[Read More Answers.](#)

Question # 91

How To Concatenate Two Text Values in Oracle?

Answer:-

There are two ways to concatenate two text values together:

- * CONCAT() function.

- * '||' operation.

Here is some examples on how to use them:

```
SELECT 'gg|' || 'Center' || '.com' FROM DUAL;
```

globalguideline.com

```
SELECT CONCAT('globalguideline', '.com') FROM DUAL;
```

globalguideline.com



[Read More Answers.](#)

Question # 92

How To Increment Dates by 1 in Oracle?

Answer:-

If you have a date, and you want to increment it by 1. You can do this by adding the date with a date interval. You can also do this by adding the number 1 directly on the date. The tutorial example below shows you how to adding numbers to dates, and take date differences:

```
SELECT TO_DATE('30-APR-06') + 1 FROM DUAL;
-- Adding 1 day to a date
01-MAY-06
SELECT TO_DATE('01-MAY-06') - TO_DATE('30-APR-06')
FROM DUAL;
-- Taking date differences
1
SELECT SYSTIMESTAMP + 1 FROM DUAL;
-- The number you add is always in days.
08-MAY-06
SELECT TO_CHAR(SYSTIMESTAMP+1,'DD-MON-YYYY HH24:MI:SS.FF3')
FROM DUAL;
-- Error: Adding 1 to a timestamp makes it a date.
```

[Read More Answers.](#)

Question # 93

How To Calculate Date and Time Differences in Oracle?

Answer:-

If you want to know how many years, months, days and seconds are there between two dates or times, you can use the date and time interval expressions: YEAR ... TO MONTH and DAY ... TO SECOND. The tutorial exercise below gives you some good examples:

```
SELECT
(TO_DATE('01-MAY-2006 16:52:57','DD-MON-YYYY HH24:MI:SS')
-
TO_DATE('31-JAN-1897 09:26:50','DD-MON-YYYY HH24:MI:SS'))
YEAR(4) TO MONTH FROM DUAL;
-- 109 years and 3 months
109-3
SELECT
(TO_DATE('01-MAY-2006 16:52:57','DD-MON-YYYY HH24:MI:SS')
-
TO_DATE('31-JAN-1897 09:26:50','DD-MON-YYYY HH24:MI:SS'))
DAY(9) TO SECOND FROM DUAL;
-- 39901 days and some seconds
39901 7:26:7.0
SELECT
(TO_TIMESTAMP('01-MAY-2006 16:52:57.847',
'DD-MON-YYYY HH24:MI:SS.FF3') -
TO_TIMESTAMP('31-JAN-1897 09:26:50.124',
'DD-MON-YYYY HH24:MI:SS.FF3'))
YEAR(4) TO MONTH FROM DUAL;
-- 109 years and 3 months
109-3
SELECT
(TO_TIMESTAMP('01-MAY-2006 16:52:57.847',
'DD-MON-YYYY HH24:MI:SS.FF3') -
TO_TIMESTAMP('31-JAN-1897 09:26:50.124',
'DD-MON-YYYY HH24:MI:SS.FF3'))
DAY(9) TO SECOND
FROM DUAL;
-- 39
```

[Read More Answers.](#)

Question # 94

How To Use IN Conditions in Oracle?

Answer:-

An IN condition is single value again a list of values. It returns TRUE, if the specified value is in the list. Otherwise, it returns FALSE.

```
Some examples are given in the script below:
SELECT CASE WHEN 3 IN (1,2,3,5) THEN
'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
SELECT CASE WHEN 3 NOT IN (1,2,3,5) THEN
'TRUE' ELSE 'FALSE' END FROM DUAL;
FALSE
SELECT CASE WHEN 'Y' IN ('F','Y','T') THEN
'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
```

[Read More Answers.](#)



Question # 95

How To Use LIKE Conditions in Oracle?

Answer:-

LIKE condition is also called pattern patch. There 3 main rules on using LIKE condition:

- * '_' is used in the pattern to match any one character.
- * '%' is used in the pattern to match any zero or more characters.
- * ESCAPE clause is used to provide the escape character in the pattern.

The following script provides you some good pattern matching examples:

```
SELECT CASE WHEN 'globalguideline.com' LIKE '%Center%'
  THEN 'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
SELECT CASE WHEN 'globalguideline.com' LIKE '%CENTER%'
  THEN 'TRUE' ELSE 'FALSE' END FROM DUAL;
-- Case sensitive by default
FALSE
SELECT CASE WHEN 'globalguideline.com' LIKE '%Center_com'
  THEN 'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
SELECT CASE WHEN '100% correct' LIKE '100%' ESCAPE "
  THEN 'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
```

[Read More Answers.](#)

Question # 96

How To Use Regular Expression in Pattern Match Conditions in Oracle?

Answer:-

If you have a pattern that is too complex for LIKE to handle, you can use the regular expression pattern patch function: REGEXP_LIKE().

The following script provides you some good examples:

```
SELECT CASE WHEN REGEXP_LIKE ('globalguideline.com', '.*gg1.*',
  'i') THEN 'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
SELECT CASE WHEN REGEXP_LIKE ('globalguideline.com', '.*com$',
  'i') THEN 'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
SELECT CASE WHEN REGEXP_LIKE ('globalguideline.com', '^F.*', 'i')
  THEN 'TRUE' ELSE 'FALSE' END FROM DUAL;
TRUE
```

[Read More Answers.](#)

Question # 97

What Are DDL Statements in Oracle?

Answer:-

DDL (Data Definition Language) statements are statements to create and manage data objects in the database. There are 3 primary DDL statements:

- * CREATE - Creating a new database object.
- * ALTER - Altering the definition of an existing data object.
- * DROP - Dropping an existing data object.

[Read More Answers.](#)

Question # 98

How To Create a New Table in Oracle?

Answer:-

If you want to create a new table in your own schema, you can log into the server with your account, and use the CREATE TABLE statement. The following script shows you how to create a table:

```
>. insqlplus /nolog
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE tip (id NUMBER(5) PRIMARY KEY,
  2  subject VARCHAR(80) NOT NULL,
  3  description VARCHAR(256) NOT NULL,
  4  create_date DATE DEFAULT (sysdate));
Table created.
```

This script creates a testing table called "tip" with 4 columns in the schema associated with the log in account "HR".

[Read More Answers.](#)

Question # 99

How To Create a New Table by Selecting Rows from Another Table?

Answer:-

Let's say you have a table with many data rows, now you want to create a backup copy of this table of all rows or a subset of them, you can use the CREATE TABLE...AS SELECT statement to do this. Here is an example script:

```
>. insqlplus /nolog
SQL> connect HR/globalguideline
Connected.
```



```
SQL> CREATE TABLE emp_dept_10
  2 AS SELECT * FROM employees WHERE department_id=10;
Table created.
SQL> SELECT first_name, last_name, salary
  2 FROM emp_dept_10;


```

FIRST_NAME LAST_NAME SALARY

```


```

```
Jennifer      Whalen      4400
```

As you can see, this SQL scripts created a table called "emp_dept_10" using the same column definitions as the "employees" table and copied data rows of one department.

This is really a quick and easy way to create a table.

[Read More Answers.](#)

Question # 100

How To Add a New Column to an Existing Table in Oracle?

Answer:-

If you have an existing table with existing data rows, and want to add a new column to that table, you can use the ALTER TABLE ... ADD statement to do this. Here is an example script:

```
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE emp_dept_110
  2 AS SELECT * FROM employees WHERE department_id=110;
Table created.
SQL> ALTER TABLE emp_dept_110 ADD (vacation NUMBER);
Table altered.
SQL> SELECT first_name, last_name, vacation
  2 FROM emp_dept_110;


```

FIRST_NAME LAST_NAME VACATION

```


```

```
Shelley      Higgins
William      Gietz
```

This SQL script added a new column called "vacation" to the "emp_dept_110" table. NULL values were added to this column on all existing data rows.

[Read More Answers.](#)

Question # 101

How To Delete a Column in an Existing Table in Oracle?

Answer:-

If you have an existing column in a table and you need that column any more, you can delete it with ALTER TABLE ... DROP COLUMN statement. Here is an example SQL script:

```
SQL> CREATE TABLE emp_dept_90
  2 AS SELECT * FROM employees WHERE department_id=90;
Table created.
SQL> SELECT last_name FROM emp_dept_90;
LAST_NAME
-----
```

```
King
Kochhar
De Haan
```

```
SQL> ALTER TABLE emp_dept_90 DROP COLUMN last_name;
Table altered.
```

```
SQL> SELECT last_name FROM emp_dept_90;
```

ERROR at line 1:

ORA-00904: "LAST_NAME": invalid identifier

As you can see the column "last_name" is gone.

[Read More Answers.](#)

Question # 102

How To Drop an Existing Table in Oracle?

Answer:-

If you want to delete an existing table and its data rows, you can use the DROP TABLE statement as shown in this script:

```
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE emp_dept_10
  2 AS SELECT * FROM employees WHERE department_id=10;
Table created.
SQL> DROP TABLE emp_dept_10;
Table dropped.
```

Be careful, when you use the DROP TABLE statement. All data rows are gone too.

[Read More Answers.](#)

Question # 103

How To Create a Table Index in Oracle?

Answer:-



If you have a table with a lots of rows, and you know that one of the columns will be used often a search criteria, you can add an index for that column to improve the search performance. To add an index, you can use the CREATE INDEX statement as shown in the following script:

```
CREATE TABLE tip (id NUMBER(5) PRIMARY KEY,  
subject VARCHAR(80) NOT NULL,  
description VARCHAR(256) NOT NULL,  
create_date DATE DEFAULT (sysdate));  
Table created.
```

```
CREATE INDEX tip_subject ON tip(subject);  
Index created.
```

[Read More Answers.](#)

Question # 104

How To Rename an Index in Oracle?

Answer:-

Let's say you have an existing index, and you don't like its name anymore for some reason, you can rename it with the ALTER INDEX ... RENAME TO statement. Here is an example script on how to rename an index:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,  
first_name VARCHAR(80) NOT NULL,  
last_name VARCHAR(80) NOT NULL,  
birth_date DATE NOT NULL,  
social_number VARCHAR(80) UNIQUE NOT NULL);  
Table created.
```

```
SELECT index_name, table_name, uniqueness  
FROM USER_INDEXES WHERE table_name = 'STUDENT';  
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES
```

```
-----  
SYS_C004153      STUDENT          UNIQUE  
SYS_C004154      STUDENT          UNIQUE</pre>  
ALTER INDEX SYS_C004153 RENAME TO student_pk;  
Statement processed.
```

```
SELECT index_name, table_name, uniqueness  
FROM USER_INDEXES WHERE table_name = 'STUDENT';  
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES
```

```
-----  
STUDENT_PK      STUDENT          UNIQUE  
SYS_C004154      STUDENT          UNIQUE</pre>
```

[Read More Answers.](#)

Question # 105

How To Drop an Index in Oracle?

Answer:-

If you don't need an existing index any more, you should delete it with the DROP INDEX statement. Here is an example SQL script:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,  
first_name VARCHAR(80) NOT NULL,  
last_name VARCHAR(80) NOT NULL,  
birth_date DATE NOT NULL,  
social_number VARCHAR(80) UNIQUE NOT NULL);  
Table created.
```

```
CREATE INDEX student_birth_date ON student(birth_date);  
Index created.
```

```
SELECT index_name, table_name, uniqueness  
FROM USER_INDEXES WHERE table_name = 'STUDENT';  
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES
```

```
-----  
SYS_C004129      STUDENT          UNIQUE  
SYS_C004130      STUDENT          UNIQUE  
STUDENT_BIRTH_DATE STUDENT          NONUNIQUE </pre>  
DROP INDEX STUDENT_BIRTH_DATE;  
Index dropped.
```

[Read More Answers.](#)

Question # 106

How To Create a New View in Oracle?

Answer:-

You can create a new view based on one or more existing tables by using the CREATE VIEW statement as shown in the following script:

```
CREATE VIEW employee_department AS  
SELECT e.employee_id, e.first_name, e.last_name,  
e.email, e.manager_id, d.department_name  
FROM employees e, departments d  
WHERE e.department_id = d.department_id;  
View created.
```

```
SELECT first_name, last_name, department_name  
FROM employee_department WHERE manager_id = 101;  
<pre>FIRST_NAME      LAST_NAME      DEPARTMENT_NAME
```



```
-----
Nancy      Greenberg   Finance
Jennifer   Whalen     Administration
Susan      Mavris     Human Resources
Hermann    Baer       Public Relations
Shelley    Higgins    Accounting</pre>
```

[Read More Answers.](#)

Question # 107

How To Drop an Existing View in Oracle?

Answer:-

If you have an existing view, and you don't want it anymore, you can delete it by using the DROP VIEW statement as shown in the following script:
DROP VIEW employee_department;
View dropped.

[Read More Answers.](#)

Question # 108

What Are DML Statements in Oracle?

Answer:-

DML (Data Manipulation Language) statements are statements to change data values in database tables. There are 3 primary DML statements:

- * INSERT - Inserting new rows into database tables.
- * UPDATE - Updating existing rows in database tables.
- * DELETE - Deleting existing rows from database tables.

[Read More Answers.](#)

Question # 109

How To Create a Testing Table in Oracle?

Answer:-

If you want to practice DML statements, you should create a testing table as shown in the script below:

```
CREATE TABLE ggl_links (id NUMBER(4) PRIMARY KEY,
url VARCHAR2(80) NOT NULL,
notes VARCHAR2(1024),
counts NUMBER,
created DATE DEFAULT (sysdate));
```

Table created.

You should keep this table for to practice other tutorial exercises presented in this collection.

[Read More Answers.](#)

Question # 110

How To Set Up SQL*Plus Output Format in Oracle?

Answer:-

If you want to practice SQL statements with SQL*Plus, you need to set up your SQL*Plus output formatting parameter properly. The following SQL*Plus commands show you some examples:

```
COLUMN id FORMAT 9999;
COLUMN url FORMAT A24;
COLUMN notes FORMAT A12;
COLUMN counts FORMAT 999999;
SET NULL 'NULL';
```

[Read More Answers.](#)

Question # 111

How To Insert a New Row into a Table in Oracle?

Answer:-

To insert a new row into a table, you should use the INSERT INTO statement with values specified for all columns as shown in the following example:

```
INSERT INTO ggl_links VALUES (101,
'http://www.globalguideline.com',
NULL,
0,
'30-Jul-2008');
1 row created.
SELECT * FROM ggl_links;


```
<pre> ID URL NOTES COUNTS CREATED

101 http://www.globalguideline.com NULL 0 30-Jul-08</pre>
```


```

[Read More Answers.](#)

Question # 112

How To Specify Default Values in INSERT Statement using Oracle?

Answer:-



If a column is defined with a default value in a table, you can use the key word DEFAULT in the INSERT statement to take the default value for that column. The following tutorial exercise gives a good example:

```
INSERT INTO ggl_links VALUES (102,
'http://www.globalguideline.com',
NULL,
0,
DEFAULT);
1 row created.
SELECT * FROM ggl_links;


```

ID URL NOTES COUNTS CREATED

101 http://www.globalguideline.com NULL 0 30-Jul-08
102 http://www.globalguideline.com NULL 0 07-MAY-06
```


```

[Read More Answers.](#)

Question # 113

How To Omit Columns with Default Values in INSERT Statement in Oracle?

Answer:-

If you don't want to specify values for columns that have default values, or you want to specify values to columns in an order different than how they are defined, you can provide a column list in the INSERT statement. If a column is omitted in the column, Oracle applies 3 rules:

- * If default value is defined for the column, that default value will be used.
- * If no default value is defined for the column and NULL is allowed, NULL will be used.
- * If no default value is defined for the column and NULL is not allowed, error will be returned.

The following tutorial exercise gives a good example:

```
INSERT INTO ggl_links (url, id)
VALUES ('http://www.globalguideline.com', 103);
1 row created.
SELECT * FROM ggl_links;


```

ID URL NOTES COUNTS CREATED

101 http://www.globalguideline.com NULL 0 30-Jul-08
102 http://www.globalguideline.com NULL 0 07-MAY-06
103 http://www.globalguideline.com NULL NULL 07-MAY-06
```


```

[Read More Answers.](#)

Question # 114

How To Insert Multiple Rows with One INSERT Statement in Oracle?

Answer:-

If you want to insert multiple rows with a single INSERT statement, you can use a subquery instead of the VALUES clause. Rows returned from the subquery will be inserted the target table.

The following tutorial exercise gives a good example:

```
INSERT INTO ggl_links
SELECT department_id, department_name||'.com', NULL, NULL,
SYSDATE FROM departments WHERE department_id >= 250;
3 row created.
SELECT * FROM ggl_links;


```

ID URL NOTES COUNTS CREATED

101 http://www.globalguideline.com NULL 0 30-Apr-06
102 http://www.globalguideline.com NULL 0 07-MAY-06
103 http://www.globalguideline.com NULL NULL 07-MAY-06
250 Retail Sales.com NULL NULL 07-MAY-06
260 Recruiting.com NULL NULL 07-MAY-06
270 Payroll.com NULL NULL 07-MAY-06
```


```

[Read More Answers.](#)

Question # 115

How To Update Values in a Table in Oracle?

Answer:-

If you want to update some values in one row or multiple rows in a table, you can use the UPDATE statement. The script below shows a good example:

```
UPDATE ggl_links SET counts = 999, notes = 'Good site.'
WHERE id = 101;
1 row updated.
SELECT * FROM ggl_links WHERE id = 101;


```

ID URL NOTES COUNTS CREATED

101 http://www.globalguideline.com Good site. 999 07-MAY-06
```


```

[Read More Answers.](#)

Question # 116

How To Update Values on Multiple Rows in Oracle?

Answer:-

If the WHERE clause in an UPDATE matches multiple rows, the SET clause will be applied to all matched rows. This rule allows you to update values on multiple



rows in a single UPDATE statement. Here is a good example:

```
UPDATE ggl_links SET counts = 9, notes = 'Wrong URL'
  WHERE id >= 250;
3 rows updated.
SELECT * FROM ggl_links WHERE id >= 250;


```
<pre> ID URL NOTES COUNTS CREATED

250 Retail Sales.com Wrong URL 9 07-MAY-06
260 Recruiting.com Wrong URL 9 07-MAY-06
270 Payroll.com Wrong URL 9 07-MAY-06</pre>
```


```

This statement updated 3 rows with the same new values on all 3 rows.

[Read More Answers.](#)

Question # 117

How To Use Existing Values in UPDATE Statements using Oracle?

Answer:-

If a row matches the WHERE clause in a UPDATE statement, existing values in this row can be used in expressions to provide new values in the SET clause. Existing values are represented by columns in the expressions. The tutorial exercise below shows a good example:

```
UPDATE ggl_links SET id = 1000 + id, counts = id*2
  WHERE id >= 250;
3 rows updated.
SELECT * FROM ggl_links WHERE id >= 250;


```
<pre> ID URL NOTES COUNTS CREATED

1250 Retail Sales.com Wrong URL 500 07-MAY-06
1260 Recruiting.com Wrong URL 520 07-MAY-06
1270 Payroll.com Wrong URL 540 07-MAY-06</pre>
```


```

This statement increased values in the id column by 1000.

[Read More Answers.](#)

Question # 118

How To Use Values from Other Tables in UPDATE Statements using Oracle?

Answer:-

If you want to update values in one with values from another table, you can use a subquery in the SET clause. The subquery should return only one row for each row in the update table that matches the WHERE clause. The tutorial exercise below shows a good example:

```
UPDATE ggl_links SET (notes, created) =
  (SELECT last_name, hire_date FROM employees
   WHERE employee_id = id)
  WHERE id < 110;
3 rows updated.
SELECT * FROM ggl_links WHERE id < 110;


```
<pre> ID URL NOTES COUNTS CREATED

101 http://www.globalguideline.com Kochhar 999 21-SEP-89
102 http://www.globalguideline.com De Haan 0 13-JAN-93
103 http://www.globalguideline.com Hunold NULL 03-JAN-90</pre>
```


```

This statement updated 3 rows with values from the employees table.

[Read More Answers.](#)

Question # 119

What Happens If the UPDATE Subquery Returns Multiple Rows?

Answer:-

If a subquery is used in a UPDATE statement, it must return exactly one row for each row in the update table that matches the WHERE clause. If it returns multiple rows, Oracle server will give you an error message. To test this out, you can try the following tutorial exercise:

```
UPDATE ggl_links SET (notes, created) =
  (SELECT last_name, hire_date FROM employees
   WHERE employee_id < id)
  WHERE id < 110;
ERROR at line 1:
ORA-01427: single-row subquery returns more than one row
The problem is the criteria in the subquery: "employee_id < id"
```

[Read More Answers.](#)

Question # 120

How To Delete an Existing Row from a Table in Oracle?

Answer:-

If you want to delete an existing row from a table, you can use the DELETE statement with a WHERE clause to identify that row. Here is good sample of DELETE statements:

```
INSERT INTO ggl_links (url, id)
  VALUES ('http://www.myspace.com', 301);
1 row created.
SELECT * FROM ggl_links WHERE id = 301;


```
<pre> ID URL NOTES COUNTS CREATED
```


```



```
-----
301 http://www.myspace.com NULL NULL 07-MAY-06 </pre>
DELETE FROM ggl_links WHERE id = 301;
1 row deleted.
SELECT * FROM ggl_links WHERE id = 301;
no rows selected
```

[Read More Answers.](#)

Question # 121

How To Delete Multiple Rows from a Table in Oracle?

Answer:-

You can delete multiple rows from a table in the same way as deleting a single row, except that the WHERE clause will match multiple rows. The tutorial exercise below deletes 3 rows from the ggl_links table:

```
SELECT * FROM ggl_links WHERE id >= 250;
<pre> ID URL          NOTES    COUNTS CREATED
-----
1250 Retail Sales.com Wrong URL 500 07-MAY-06
1260 Recruiting.com Wrong URL 520 07-MAY-06
1270 Payroll.com Wrong URL 540 07-MAY-06</pre>
DELETE FROM ggl_links WHERE id >= 250;
3 row deleted.
SELECT * FROM ggl_links WHERE id >= 250;
no rows selected
```

[Read More Answers.](#)

Question # 122

How To Delete All Rows a Table in Oracle?

Answer:-

If you want to delete all rows from a table, you have two options:

- * Use the DELETE statement with no WHERE clause.
- * Use the TRUNCATE TABLE statement.

The TRUNCATE statement is more efficient the DELETE statement. The tutorial exercise shows you a good example of TRUNCATE statement:

```
SELECT COUNT(*) FROM ggl_links;
COUNT(*)
-----
3
TRUNCATE TABLE ggl_links;
Table truncated.
SELECT COUNT(*) FROM ggl_links;
COUNT(*)
-----
0
```

[Read More Answers.](#)

Question # 123

What Is a SELECT Query Statement in Oracle?

Answer:-

The SELECT statement is also called the query statement. It is the most frequently used SQL statement in any database application. A SELECT statement allows you to retrieve data from one or more tables, or views, with different selection criteria, grouping criteria and sorting orders.

[Read More Answers.](#)

Question # 124

How To Select All Columns of All Rows from a Table in Oracle?

Answer:-

The simplest query statement is the one that selects all columns of all rows from a table: "SELECT * FROM table_name;". The (*) in the SELECT clause tells the query to return all columns. The tutorial exercise below gives you a good example:

```
SQL> SELECT * FROM departments;
<pre>DEPARTMENT_ID DEPARTMENT_NAME    MANAGER_ID LOCATION_ID
-----
10 Administration      200      1700
20 Marketing           201      1800
30 Purchasing          114      1700
40 Human Resources     203      2400
50 Shipping            121      1500
60 IT                  103      1400
70 Public Relations    204      2700
80 Sales               145      2500
90 Executive           100      1700
.....</pre>
```

[Read More Answers.](#)

Question # 125



How To Select Some Columns from a Table in Oracle?

Answer:-

If you want explicitly tell the query to some columns, you can specify the column names in SELECT clause. The following select statement returns only two columns from the table "departments":

```
SQL> SELECT location_id, department_name FROM DEPARTMENTS;  
<pre>LOCATION_ID DEPARTMENT_NAME
```

```
-----  
1700 Administration  
1800 Marketing  
1700 Purchasing  
2400 Human Resources  
1500 Shipping  
1400 IT  
2700 Public Relations  
2500 Sales  
1700 Executive  
.....</pre>
```

[Read More Answers.](#)

Question # 126

How To Select Some Rows from a Table in Oracle?

Answer:-

If you don't want select all rows from a table, you can specify a WHERE clause to tell the query to return only the rows that meets the condition defined in the WHERE clause. The following select statement only returns rows that has department name starts with the letter "C":

```
SQL> SELECT * FROM departments  
2 WHERE department_name LIKE 'C%';  
<pre>DEPARTMENT_ID DEPARTMENT_NAME    MANAGER_ID LOCATION_ID
```

```
-----  
130 Corporate Tax          1700  
140 Control And Credit     1700  
180 Construction          1700  
190 Contracting           1700  
.....</pre>
```

[Read More Answers.](#)

Question # 127

How To Sort the Query Output in Oracle?

Answer:-

If you want the returning rows to be sorted, you can specify a sorting expression in the ORDER BY clause. The following select statement returns rows sorted by the values in the "manager_id" column:

```
SQL> SELECT * FROM departments ORDER BY manager_id;  
<pre>DEPARTMENT_ID DEPARTMENT_NAME    MANAGER_ID LOCATION_ID
```

```
-----  
90 Executive          100    1700  
60 IT                 103    1400  
100 Finance           108    1700  
30 Purchasing         114    1700  
50 Shipping           121    1500  
80 Sales              145    2500  
10 Administration    200    1700  
20 Marketing          201    1800  
.....</pre>
```

[Read More Answers.](#)

Question # 128

Can the Query Output Be Sorted by Multiple Columns in Oracle?

Answer:-

You can specifying multiple columns in the ORDER BY clause as shown in the following example statement, which returns employees' salaries sorted by department and salary value:

```
SQL> SELECT department_id, first_name, last_name, salary  
FROM employees ORDER BY department_id, salary;  
<pre>DEPARTMENT_ID FIRST_NAME    LAST_NAME    SALARY
```

```
-----  
10 Jennifer    Whalen      4400  
20 Pat         Fay         6000  
20 Michael     Hartstein   13000  
30 Karen       Colmenares  2500  
30 Guy         Himuro      2600  
30 Sigal       Tobias       2800  
30 Shelli     Baida       2900  
30 Alexander   Khoo        3100  
30 Den         Raphaely    11000  
40 Susan       Mavris      6500  
50 TJ          Olson       2100  
.....</pre>
```



[Read More Answers.](#)

Question # 129

How To Sort Output in Descending Order in Oracle?

Answer:-

If you want to sort a column in descending order, you can specify the DESC keyword in the ORDER BY clause. The following SELECT statement first sorts the department in descending order, then sorts the salary in ascending order:

```
SQL> SELECT department_id, first_name, last_name, salary
FROM employees ORDER BY department_id DESC, salary;


```
DEPARTMENT_ID FIRST_NAME LAST_NAME SALARY

110 Kimberly Grant 7000
110 William Gietz 8300
110 Shelley Higgins 12000
100 Luis Popp 6900
100 Ismael Sciarra 7700
100 Jose Manuel Urman 7800
100 John Chen 8200
100 Daniel Faviet 9000
.....</pre>
```


```

[Read More Answers.](#)

Question # 130

How To Use SELECT Statement to Count the Number of Rows in Oracle?

Answer:-

If you want to count the number of rows, you can use the COUNT(*) function in the SELECT clause. The following select statement returns the number of rows in the "department" table:

```
SQL> SELECT COUNT(*) FROM departments;
COUNT(*)
-----
27
```

So there are 27 rows in the "departments" table.

[Read More Answers.](#)

Question # 131

Can SELECT Statements Be Used on Views in Oracle?

Answer:-

Select (query) statements can be used on views in the same way as tables. The following tutorial exercise helps you creating a view and running a query statement on the view:

```
SQL> CREATE VIEW managed_dept AS
SELECT * FROM departments WHERE manager_id IS NOT NULL;
View created.
SQL> SELECT * FROM managed_dept WHERE location_id = 1700;


```
DEPARTMENT_ID DEPARTMENT_NAME MANAGER_ID LOCATION_ID

10 Administration 200 1700
30 Purchasing 114 1700
90 Executive 100 1700
100 Finance 108 1700
110 Accounting 205 1700</pre>
```


```

[Read More Answers.](#)

Question # 132

How To Filter Out Duplications in the Returning Rows using Oracle?

Answer:-

If there are duplications in the returning rows, and you want to remove the duplications, you can use the keyword DISTINCT or UNIQUE in the SELECT clause. The tutorial exercise below shows you that DISTINCT works on selected columns only:

```
SQL> CREATE TABLE ggl_team AS
SELECT first_name, last_name FROM employees
WHERE first_name = 'John';
Table created.
SQL> INSERT INTO ggl_team VALUES ('John', 'Chen');
SQL> INSERT INTO ggl_team VALUES ('James', 'Chen');
SQL> INSERT INTO ggl_team VALUES ('Peter', 'Chen');
SQL> INSERT INTO ggl_team VALUES ('John', 'Chen');
SQL> SELECT * FROM ggl_team;


```
FIRST_NAME LAST_NAME

John Chen
John Russell
John Seo
John Chen
James Chen
```


```



```
Peter      Chen
John       Chen</pre>
SQL> SELECT DISTINCT * FROM ggl_team;
<pre>FIRST_NAME      LAST_NAME
```

```
Peter      Chen
John       Chen
James      Chen
John       Seo
John       Russell</pre>
```

[Read More Answers.](#)

Question # 133

What Are Group Functions in Oracle?

Answer:-

Group functions are functions applied to a group of rows. Examples of group functions are:

- * COUNT(*) - Returns the number of rows in the group.
- * MIN(exp) - Returns the minimum value of the expression evaluated on each row of the group.
- * MAX(exp) - Returns the maximum value of the expression evaluated on each row of the group.
- * AVG(exp) - Returns the average value of the expression evaluated on each row of the group.

[Read More Answers.](#)

Question # 134

How To Use Group Functions in the SELECT Clause using Oracle?

Answer:-

If group functions are used in the SELECT clause, they will be used on the rows that meet the query selection criteria, the output of group functions will be returned as output of the query. The following select statement returns 4 values calculate by 4 group functions on all rows of the "departments" table:

```
SQL> SELECT COUNT(*), MIN(department_id),
 2 MAX(department_id) FROM departments;
<pre> COUNT(*) MIN(DEPARTMENT_ID) MAX(DEPARTMENT_ID)
-----
 27          10          270</pre>
```

[Read More Answers.](#)

Question # 135

Can Group Functions Be Mixed with Non-group Selection Fields?

Answer:-

If a group function is used in the SELECT clause, all other selection fields must be group level fields. Non-group fields can not be mixed with group fields in the SELECT clause. The script below gives you an example of invalid SELECT statements with group and non-group selection fields:

```
SQL> SELECT COUNT(*), department_id FROM departments;
ORA-00937: not a single-group group function
In this example, COUNT(*) is a group field and department_id is a non-group field.
```

[Read More Answers.](#)

Question # 136

How To Divide Query Output into Groups in Oracle?

Answer:-

You can divide query output into multiple groups with the GROUP BY clause. It allows you specify a column as the grouping criteria, so that rows with the same value in the column will be considered as a single group. When the GROUP BY clause is specified, the select statement can only be used to return group level information. The following script gives you a good GROUP BY example:

```
SQL> SELECT department_id, MIN(salary), MAX(salary),
 2 AVG(salary) FROM employees GROUP BY department_id;
<pre>DEPARTMENT_ID MIN(SALARY) MAX(SALARY) AVG(SALARY)
-----
 100          6900          12000          8600
 30          2500          11000          4150
           7000          7000          7000
 90          17000          24000          19333.3333
 20          6000          13000          9500
 70          10000          10000          10000
 110          8300          12000          10150
 50          2100          8200          3475.55556
.....</pre>
```

[Read More Answers.](#)

Question # 137

How To Apply Filtering Criteria at Group Level in Oracle?

Answer:-

If you want to return only specific groups from the query, you can apply filtering criteria at the group level by using the HAVING clause inside the GROUP BY clause. The following script gives you a good HAVING example:

```
SQL> SELECT department_id, MIN(salary), MAX(salary),
```



```

2 AVG(salary) FROM employees GROUP BY department_id
3 HAVING AVG(salary) < 5000;
<pre>DEPARTMENT_ID MIN(SALARY) MAX(SALARY) AVG(SALARY)
-----
30      2500      11000      4150
50      2100      8200 3475.55556
10      4400      4400      4400</pre>

```

[Read More Answers.](#)

Question # 138

How To Count Duplicated Values in a Column in Oracle?

Answer:-

If you have a column with duplicated values, and you want to know what are those duplicated values are and how many duplicates are there for each of those values, you can use the GROUP BY ... HAVING clause as shown in the following example. It returns how many duplicated first names in the employees table:

```

SQL> SELECT first_name, COUNT(*) FROM employees
      GROUP BY first_name HAVING COUNT(*) > 1;
<pre>FIRST_NAME      COUNT(*)
-----

```

```

Peter           3
Michael         2
Steven          2
John            3
Julia           2
William         2
Karen           2
Kevin           2
.....</pre>

```

[Read More Answers.](#)

Question # 139

Can Multiple Columns Be Used in GROUP BY in Oracle?

Answer:-

You can use multiple columns in the GROUP BY clause as shown in the following example. It returns how many employees are having the same salary in each department:

```

SQL> SELECT department_id, salary, count(*)
      2 FROM employees GROUP BY department_id,
      3 salary HAVING count(*) > 1;
<pre>DEPARTMENT_ID  SALARY  COUNT(*)
-----

```

```

90      17000      2
50      3200      4
50      2200      2
50      3600      2
80      10500     2
80      9000      2
50      2700      2
.....</pre>

```

[Read More Answers.](#)

Question # 140

Can Group Functions Be Used in the ORDER BY Clause in Oracle?

Answer:-

If the query output is aggregated as groups, you can sort the groups by using group functions in the ORDER BY clause. The following statement returns how many employees are having the same salary in each department. The group output is sorted by the count in each group in descending order:

```

SQL> SELECT department_id, salary, count(*)
      2 FROM employees GROUP BY department_id,
      3 salary HAVING count(*) > 1
      ORDER BY COUNT(*) DESC;
<pre>DEPARTMENT_ID  SALARY  COUNT(*)
-----

```

```

50      2500      5
50      3200      4
50      2800      3
80      10000     3
80      9500      3
50      3100      3
50      2600      3
.....</pre>

```

[Read More Answers.](#)

Question # 141

How To Join Two Tables in a Single Query using Oracle?

Answer:-



Two tables can be joined together in a query in 4 ways in Oracle:

- * Inner Join: Returns only rows from both tables that satisfy the join condition.
- * Left Outer Join: Returns rows from both tables that satisfy the join condition, and the rest of rows from the first (left) table.
- * Right Outer Join: Returns rows from both tables that satisfy the join condition, and the rest of rows from the second (right) table.
- * Full Outer Join: Returns rows from both tables that satisfy the join condition, the rest of rows from the first (left) table, and the rest of rows from the second (right) table.

[Read More Answers.](#)

Question # 142

How To Write a Query with an Inner Join in Oracle?

Answer:-

If you want to query from two tables with an inner join, you can use the INNER JOIN ... ON clause in the FROM clause. The following query returns output with an inner join from two tables: employees and departments. The join condition is that the department ID in the employees table equals to the department ID in the departments table:

```
SQL> SELECT employees.first_name, employees.last_name,
2 departments.department_name
3 FROM employees INNER JOIN departments
4 ON employees.department_id=departments.department_id;


```


| FIRST_NAME | LAST_NAME | DEPARTMENT_NAME |
|------------|-----------|-----------------|
| Steven | King | Executive |
| Neena | Kochhar | Executive |
| Lex | De Haan | Executive |
| Alexander | Hunold | IT |
| Bruce | Ernst | IT |
| David | Austin | IT |
| Valli | Pataballa | IT |


```


```

.....</pre>
Note that when multiple tables are used in a query, column names need to be prefixed with table names in case the same column name is used in both tables

[Read More Answers.](#)

Question # 143

How To Define and Use Table Alias Names in Oracle?

Answer:-

When column names need to be prefixed with table names, you can define table alias name and use them to prefix column names as shown in the following select statement:

```
SQL> SELECT e.first_name, e.last_name, d.department_name
FROM employees e INNER JOIN departments d
ON e.department_id=d.department_id;


```


| FIRST_NAME | LAST_NAME | DEPARTMENT_NAME |
|------------|-----------|-----------------|
| Steven | King | Executive |
| Neena | Kochhar | Executive |
| Lex | De Haan | Executive |
| Alexander | Hunold | IT |
| Bruce | Ernst | IT |
| David | Austin | IT |
| Valli | Pataballa | IT |


```


```

[Read More Answers.](#)

Question # 144

How To Write a Query with a Left Outer Join in Oracle?

Answer:-

If you want to query from two tables with a left outer join, you can use the LEFT OUTER JOIN ... ON clause in the FROM clause. The following query returns output with a left outer join from two tables: departments and employees. The join condition is that the manager ID in the departments table equals to the employee ID in the employees table:

```
SQL> set NULL 'NULL'
SQL> SELECT d.department_name, e.first_name, e.last_name
2 FROM departments d LEFT OUTER JOIN employees e
3 ON d.manager_id = e.employee_id;


```


| DEPARTMENT_NAME | FIRST_NAME | LAST_NAME |
|----------------------|------------|-----------|
| Administration | Jennifer | Whalen |
| Marketing | Michael | Hartstein |
| Purchasing | Den | Raphaely |
| Human Resources | Susan | Mavris |
| Shipping | Adam | Fripp |
| IT | Alexander | Hunold |
| | | |
| Treasury | NULL | NULL |
| Corporate Tax | NULL | NULL |
| Control And Credit | NULL | NULL |
| Shareholder Services | NULL | NULL |
| Benefits | NULL | NULL |
| Manufacturing | NULL | NULL |
| Construction | NULL | NULL |


```


```




[Read More Answers.](#)

Question # 145

How To Write a Query with a Right Outer Join in Oracle?

Answer:-

If you want to query from two tables with a right outer join, you can use the RIGHT OUTER JOIN ... ON clause in the FROM clause. The following query returns output with a right outer join from two tables: departments and employees. The join condition is that the manager ID in the departments table equals to the employee ID in the employees table:

```
SQL> set NULL 'NULL'
SQL> SELECT d.department_name, e.first_name, e.last_name
 2 FROM departments d RIGHT OUTER JOIN employees e
 3 ON d.manager_id = e.employee_id;


```


| DEPARTMENT_NAME | FIRST_NAME | LAST_NAME |
|-----------------|------------|------------|
| Administration | Jennifer | Whalen |
| Marketing | Michael | Hartstein |
| Purchasing | Den | Raphaely |
| Human Resources | Susan | Mavris |
| Shipping | Adam | Fripp |
| IT | Alexander | Hunold |
| | | |
| NULL | Clara | Vishney |
| NULL | Jason | Mallin |
| NULL | Hazel | Philtanker |
| NULL | Nanette | Cambrault |
| NULL | Alana | Walsh |
| NULL | Karen | Partners |
| NULL | Bruce | |


```


```

[Read More Answers.](#)

Question # 146

How To Write a Query with a Full Outer Join in Oracle?

Answer:-

If you want to query from two tables with a full outer join, you can use the FULL OUTER JOIN ... ON clause in the FROM clause. The following query returns output with a full outer join from two tables: departments and employees. The join condition is that the manager ID in the departments table equals to the employee ID in the employees table:

```
SQL> set NULL 'NULL'
SQL> SELECT d.department_name, e.first_name, e.last_name
 2 FROM departments d FULL OUTER JOIN employees e
 3 ON d.manager_id = e.employee_id;

```

[Read More Answers.](#)

Question # 147

How To Write an Inner Join with the WHERE Clause in Oracle?

Answer:-

If you don't want to use the INNER JOIN ... ON clause to write an inner join, you can put the join condition in the WHERE clause as shown in the following query example:

```
SQL> SELECT d.department_name, e.first_name, e.last_name
 2 FROM departments d, employees e
 3 WHERE d.manager_id = e.employee_id;


```


| DEPARTMENT_NAME | FIRST_NAME | LAST_NAME |
|-----------------|------------|-----------|
| Administration | Jennifer | Whalen |
| Marketing | Michael | Hartstein |
| Purchasing | Den | Raphaely |
| Human Resources | Susan | Mavris |
| Shipping | Adam | Fripp |
| IT | Alexander | Hunold |
| | | |


```


```

[Read More Answers.](#)

Question # 148

How To Write a Left Outer Join with the WHERE Clause in Oracle?

Answer:-

If you don't want to use the LEFT OUTER JOIN ... ON clause to write a left outer join, you can use a special criteria in the WHERE clause as "left_table.column = right_table.column(+)". The select statement below is an example of a left outer join written with the WHERE clause:

```
SQL> set NULL 'NULL'
SQL> SELECT d.department_name, e.first_name, e.last_name
 2 FROM departments d, employees e
 3 WHERE d.manager_id = e.employee_id(+);


```


| DEPARTMENT_NAME | FIRST_NAME | LAST_NAME |
|-----------------|------------|-----------|
| Administration | Jennifer | Whalen |


```


```



```
Marketing      Michael      Hartstein
Purchasing     Den         Raphaely
Human Resources Susan        Mavris
Shipping       Adam        Fripp
IT             Alexander   Hunold
.....
Treasury       NULL        NULL
Corporate Tax  NULL        NULL
Control And Credit NULL      NULL
Shareholder Services NULL      NULL
Benefits       NULL        NULL
Manufacturing  NULL        NULL
.....</pre>
```

[Read More Answers.](#)

Question # 149

How To Name Query Output Columns in Oracle?

Answer:-

Each column in the query output has a default name. If you don't like the default name, you can specify a new name for any column in the query output by using the AS clause. The following statement shows you a good example:

```
SQL> SELECT department_id AS ID, MIN(salary) AS Low,
2 MAX(salary) AS High, AVG(salary) AS Average
3 FROM employees GROUP BY department_id
4 HAVING AVG(salary) < 5000;
<pre> ID LOW HIGH AVERAGE
```

```
-----
30 2500 11000 4150
50 2100 8200 3475.55556
10 4400 4400 4400</pre>
```

[Read More Answers.](#)

Question # 150

What Is a Subquery in Oracle?

Answer:-

A subquery is a SELECT statement used as part of the selection criteria of the main SELECT statement. The subquery specified in the WHERE clause will be evaluated repeated on each row of the selection base table. The output of the subquery will be used in the final evaluation of the criteria. Usually, subqueries are used in the following Boolean operations:

- * "expression IN (subquery)"
- * "expression NOT IN (subquery)"
- * "EXISTS (subquery)"
- * "NOT EXISTS (subquery)"

[Read More Answers.](#)

Question # 151

How To Use Subqueries with the IN Operator using Oracle?

Answer:-

A subquery can be used with the IN operator as "expression IN (subquery)". The subquery should return a single column with one or more rows to form a list of values to be used by the IN operation. The following tutorial exercise shows you how to use a subquery with the IN operator:

```
SQL> SELECT first_name, last_name FROM employees
2 WHERE department_id IN (
3 SELECT department_id FROM departments
4 WHERE location_id = 1700
5 );
<pre>FIRST_NAME LAST_NAME
```

```
-----
Steven King
Neena Kochhar
Lex De Haan
Nancy Greenberg
Daniel Faviet
John Chen
Ismael Sciarra
.....</pre>
```

[Read More Answers.](#)

Question # 152

How To Use Subqueries with the EXISTS Operator in Oracle?

Answer:-

A subquery can be used with the EXISTS operator as "EXISTS (subquery)", which returns true if the subquery returns one or more rows. The following statement is a good example of "EXISTS (subquery)". It returns rows from employees table that there are rows existing in the departments table linked to the employees table with location_id = 1700.

```
SQL> SELECT first_name, last_name FROM employees e
2 WHERE EXISTS (
```



```

3 SELECT * FROM departments d
4 WHERE e.department_id = d.department_id
5 AND d.location_id = 1700
6 );

```

```


```
<pre>FIRST_NAME LAST_NAME

Steven King
Neena Kochhar
Lex De Haan
Nancy Greenberg
Daniel Favier
John Chen
Ismael Sciarra
.....</pre>

```


```

[Read More Answers.](#)

Question # 153

How To Use Subqueries in the FROM clause in Oracle?

Answer:-

If you have a query returning many rows of data, and you want to perform another query on those rows, you can put the first query as a subquery in the FROM clause of the second query. The following statement shows you how to use a subquery as base table for the main query:

```

SQL> SELECT * FROM (
2  SELECT first_name, last_name, department_name
3  FROM employees e, departments d
4  WHERE e.department_id = d.department_id
5  ) WHERE department_name LIKE 'S%' ORDER BY last_name;


```
<pre>FIRST_NAME LAST_NAME DEPARTMENT_NAME

```


```

```

Ellen      Abel      Sales
Sundar     Ande      Sales
Mozhe      Atkinson  Shipping
Amit       Banda     Sales
Elizabeth  Bates     Sales
Sarah      Bell      Shipping</pre>

```

[Read More Answers.](#)

Question # 154

How To Count Groups Returned with the GROUP BY Clause in Oracle?

Answer:-

If you use the COUNT(*) function on groups returned with the GROUP BY clause, it will count the number of rows within each group, not the number of groups. If you want to count the number of groups, you can put the GROUP BY query into a subquery and apply the COUNT(*) function on the main query as shown in the following tutorial exercise:

```

SQL> SELECT first_name, COUNT(*) FROM employees
  GROUP BY first_name HAVING COUNT(*) > 1;


```
<pre>FIRST_NAME COUNT(*)

```


```

```

Peter          3
Michael        2
Steven         2
John           3
Julia          2
William        2
Karen          2
Kevin          2
.....

```

```

SQL> SELECT COUNT(*) FROM (
  SELECT first_name, COUNT(*) FROM employees
  GROUP BY first_name HAVING COUNT(*) > 1
);
COUNT(*)
-----
13</pre>

```

[Read More Answers.](#)

Question # 155

How To Return Top 5 Rows in Oracle?

Answer:-

If you want the query to return only the first 5 rows, you can use the pseudo column called ROWNUM in the WHERE clause. ROWNUM contains the row number of each returning row from the query. The following statement returns the first 5 rows from the employees table:

```

SQL> SELECT employee_id, first_name, last_name
  FROM employees WHERE ROWNUM <= 5;


```
<pre>EMPLOYEE_ID FIRST_NAME LAST_NAME

```


```

```

100 Steven      King
101 Neena       Kochhar
102 Lex         De Haan

```



103 Alexander Hunold
104 Bruce Ernst</pre>

[Read More Answers.](#)

Question # 156

What Is an Oracle Transaction?

Answer:-

A transaction is a logical unit of work requested by a user to be applied to the database objects. Oracle server introduces the transaction concept to allow users to group one or more SQL statements into a single transaction, so that the effects of all the SQL statements in a transaction can be either all committed (applied to the database) or all rolled back (undone from the database).

[Read More Answers.](#)

Question # 157

How To Start a New Transaction in Oracle?

Answer:-

There is no SQL statement to explicitly start a new transaction. Oracle server implicitly starts a new transaction with the following two conditions:

- * The first executable statement of a new user session will automatically start a new transaction.
- * The first executable statement after a previous transaction has been ended will automatically start a new transaction.

[Read More Answers.](#)

Question # 158

How To End the Current Transaction in Oracle?

Answer:-

There are several ways the current transaction can be ended:

- * Running the COMMIT statement will explicitly end the current transaction.
- * Running the ROLLBACK statement will explicitly end the current transaction.
- * Running any DDL statement will implicitly end the current transaction.
- * Disconnecting a user session will implicitly end the current transaction.
- * Killing a user session will implicitly end the current transaction.

[Read More Answers.](#)

Question # 159

How To Create an Oracle Testing Table?

Answer:-

If you want to practice DML statements, you should create a testing table as shown in the script below:

```
>cd (OracleXE home directory)
>. insqlplus /nolog
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE ggl_links (id NUMBER(4) PRIMARY KEY,
url VARCHAR2(16) NOT NULL,
notes VARCHAR2(16),
counts NUMBER(4),
created DATE DEFAULT (sysdate));
Table created.
You should keep this table for to practice other tutorial exercises presented in this collection.
```

[Read More Answers.](#)

Question # 160

How To Commit the Current Transaction in Oracle?

Answer:-

If you have used some DML statements updated some data objects, and you want to have the updates to be permanently recorded in the database, you can use the COMMIT statement. It will make all the database changes made in the current transaction become permanent and end the current transaction. The following tutorial exercise shows you how to use COMMIT statements:

```
SQL> connect HR/globalguideline
SQL> INSERT INTO ggl_links (url, id)
  2 VALUES ('globalguideline.com', 101);
SQL> INSERT INTO ggl_links (url, id)
  2 VALUES ('globalguideline.com/html', 110);
SQL> SELECT * FROM ggl_links;


```

ID URL NOTES COUNTS CREATED

101 globalguideline.com 07-MAY-06
110 globalguideline.com/html 07-MAY-06</pre>
SQL> COMMIT;
Commit complete.
```


```

[Read More Answers.](#)

Question # 161

How To Rollback the Current Transaction in Oracle?



Answer:-

If you have used some DML statements updated some data objects, you find a problem with those updates, and you don't want those updates to be permanently recorded in the database, you can use the ROLLBACK statement. It will remove all the database changes made in the current transaction and end the current transaction. The following tutorial exercise shows you how to use ROLLBACK statements:

```
SQL> connect HR/globalguideline
SQL> INSERT INTO ggl_links (url, id)
  2 VALUES ('google.com', 102);
SQL> INSERT INTO ggl_links (url, id)
  3 VALUES ('myspace.com', 103);
SQL> SELECT * FROM ggl_links;
 ID URL          NOTES          COUNTS CREATED
-----
 101 globalguideline.com          07-MAY-06
 110 globalguideline.com/html    07-MAY-06
 102 google.com                07-MAY-06
 103 myspace.com                07-MAY-06</pre>
```

[Read More Answers.](#)

Question # 162

What Happens to the Current Transaction If a DDL Statement Is Executed?

Answer:-

If a DDL statement is executed, the current transaction will be committed and ended. All the database changes made in the current transaction will become permanent. This is called an implicit commit by a DDL statement. The following tutorial exercise shows you that the CREATE TABLE statement forced the current transaction to be committed and ended. The subsequent ROLLBACK statement has no effects on the closed transaction.

```
SQL> connect HR/globalguideline
SQL> INSERT INTO ggl_links (url, id)
  2 VALUES ('oracle.com', 112);

SQL> INSERT INTO ggl_links (url, id)
  2 VALUES ('sql.com', 113);
SQL> CREATE TABLE ggl_temp AS (SELECT * FROM ggl_links);
Table created.
SQL> ROLLBACK;
Rollback complete.
SQL> SELECT * FROM ggl_links;
<pre> ID URL          NOTES          COUNTS CREATED
-----
 101 globalguideline.com          07-MAY-06
 110 globalguideline.com/html    07-MAY-06
 112 oracle.com                07-MAY-06
 113 sql.com                    07-MAY-06</pre>
```

[Read More Answers.](#)

Question # 163

What Happens to the Current Transaction If the Session Is Ended?

Answer:-

If a session is ended, the current transaction in that session will be committed and ended. All the database changes made in the current transaction will become permanent. This is called an implicit commit when session is ended. The following tutorial exercise shows you that the "disconnect" command forces the current transaction to be committed and ended. When the session is reconnected, you can see the changes made by the UPDATE statements.

```
SQL> connect HR/globalguideline
SQL> UPDATE ggl_links SET url = 'globalguideline.COM'
  2 WHERE id = 101;

SQL> UPDATE ggl_links SET url = 'globalguideline.COM/html'
  2 WHERE id = 110;
SQL> disconnect
SQL> connect HR/globalguideline
SQL> SELECT * FROM ggl_links;
<pre> ID URL          NOTES          COUNTS CREATED
-----
 101 globalguideline.COM          07-MAY-06
 110 globalguideline.COM/html    07-MAY-06
 112 oracle.com                07-MAY-06
 113 sql.com                    07-MAY-06</pre>
```

[Read More Answers.](#)

Question # 164

What Happens to the Current Transaction If the Session Is Killed?

Answer:-

If a session is killed by the DBA, the current transaction in that session will be rolled back and ended. All the database changes made in the current transaction will be removed. This is called an implicit rollback when session is killed. The following tutorial exercise shows you that the DBA KILL SESSION command forces the current transaction to be rolled back with all the changes uncommitted.

```
SQL> connect HR/globalguideline
SQL> SELECT * FROM ggl_links;
<pre> ID URL          NOTES          COUNTS CREATED
```



```

-----
101 globalguideline.COM          07-MAY-06
110 globalguideline.COM          07-MAY-06
112 oracle.com                   07-MAY-06
113 sql.com                       07-MAY-06</pre>
SQL> DELETE FROM ggl_links where id = 112;
1 row deleted.
SQL> DELETE FROM ggl_links where id = 113;
1 row deleted.
SQL> SELECT * FROM ggl_links;
<pre>  ID URL          NOTES          COUNTS CREATED
-----
101 globalguideline.COM          07-MAY-06
110 globalguideline.COM          07-MAY-06</pre>

```

[Read More Answers.](#)

Question # 165

How Does Oracle Handle Read Consistency?

Answer:-

Oracle supports two options for you on how to maintain read consistency:

- * READ WRITE (the default option), also called statement-level read consistency.
- * READ ONLY, also called transaction-level read consistency.

[Read More Answers.](#)

Question # 166

What Is a READ WRITE Transaction in Oracle?

Answer:-

A READ WRITE transaction is a transaction in which the read consistency is set at the statement level. In a READ WRITE transaction, a logical snapshot of the database is created at the beginning of the execution of each statement and released at the end of the execution. This guaranties that all reads within a single statement get consistent data from the database.

For example, if you have a query statement that takes 10 minutes to be executed, a snapshot of the database will be created for this statement for 10 minutes. If a subquery is used in this statement, it will get the consistent data no matter when it gets executed within this 10 minutes. In another word, data changes made during this 10 minutes by other users will not impact the execution of this query statement.

By default, all transactions are started as READ WRITE transactions.

[Read More Answers.](#)

Question # 167

What Is a READ ONLY Transaction in Oracle?

Answer:-

A READ ONLY transaction is a transaction in which the read consistency is set at the transaction level. In a READ ONLY transaction, a logical snapshot of the database is created at the beginning of the transaction and released at the end of the transaction. This guaranties that all reads in all statements within this transaction get consistent data from the database.

For example, if you have a transaction with many statements that takes 10 hours to be executed, a snapshot of the database will be created for this transaction for 10 hours. If a query statement is executed at the beginning of the transaction and at the end of the transaction, it will return the same result guarantied. In another word, data changes made during this 10 hours by other users will not impact the execution of statements within this transaction.

[Read More Answers.](#)

Question # 168

How To Set a Transaction To Be READ ONLY in Oracle?

Answer:-

If you want a transaction to be set as READ ONLY, you need to the transaction with the SET TRANSACTION READ ONLY statement. Note that a DML statement will start the transaction automatically. So you have to issue the SET TRANSACTION statement before any DML statements. The tutorial exercise below shows you a good example of READ ONLY transaction:

```

SQL> connect HR/globalguideline
SQL> SET TRANSACTION READ ONLY;
Transaction set.
SQL> SELECT * FROM ggl_links;
<pre>  ID URL          NOTES          COUNTS CREATED
-----
101 globalguideline.COM          07-MAY-06
110 globalguideline.COM          07-MAY-06
112 oracle.com                   07-MAY-06
113 sql.com                       07-MAY-06</pre>

```

[Read More Answers.](#)

Question # 169

What Are the Restrictions in a Oracle READ ONLY Transaction?

Answer:-

There are lots of restrictions in a READ ONLY transaction:

- * You can not switch to READ WRITE mode.
- * You can not run any INSERT, UPDATE, DELETE statements.



* You can run SELECT query statements.

The tutorial exercise below shows you some of the restrictions:

```
SQL> connect HR/globalguideline
```

```
SQL> SET TRANSACTION READ ONLY;
```

Transaction set.

```
SQL> SET TRANSACTION READ WRITE;
```

```
ORA-01453: SET TRANSACTION must be first statement of transaction
```

```
SQL> INSERT INTO ggl_links (url, id)
```

```
  2 VALUES ('sql.com', 113);
```

```
ORA-01456: may not perform insert/delete/update operation inside a READ ONLY transaction
```

```
SQL> DELETE FROM ggl_links where id = 110;
```

```
ORA-01456: may not perform insert/delete/update operation inside a READ ONLY transaction
```

```
SQL> SELECT * FROM ggl_links;
```

```
<pre>  ID URL          NOTES          COUNTS CREATED
-----
  101 globalguideline.COM          07-MAY-06
  110 globalguideline.COM          07-MAY-06</pre>
```

[Read More Answers.](#)

Question # 170

What Are the General Rules on Data Consistency?

Answer:-

* All SQL statements always work with a snapshot of the database to provide data consistency.

* For READ WRITE transactions, the snapshot is taken when each statement starts.

* For READ ONLY transactions, the snapshot is taken when the transaction starts.

* The snapshot never include uncommitted changes from other transactions.

* The snapshot always include uncommitted changes from its own transaction.

[Read More Answers.](#)

Question # 171

What Are Transaction Isolation Levels Supported by Oracle?

Answer:-

Oracle supports two transaction isolation levels:

* READ COMMITTED (the default option). If the transaction contains DML that requires row locks held by another transaction, then the DML statement waits until the row locks are released.

* SERIALIZABLE. If a serializable transaction contains data manipulation language (DML) that attempts to update any resource that may have been updated in a transaction uncommitted at the start of the serializable transaction, then the DML statement fails.

[Read More Answers.](#)

Question # 172

What Is a Data Lock in Oracle?

Answer:-

A data lock is logical flag the Oracle server is placed on data objects to give an exclusive right to a transaction. Statements in other transactions needs to respect data locks based on certain rules. Rules on data locks are:

* SELECT query statements do not create any data locks.

* INSERT, UPDATE, and DELETE statements create data locks on the affected rows.

* Data locks are released when the owner transaction ends.

[Read More Answers.](#)

Question # 173

How Data Locks Are Respected in Oracle?

Answer:-

Here are the rules on how data locks are respected:

* All statements ignore data locks owned its own transaction.

* SELECT query statements ignores data locks owned by any transactions.

* INSERT, UPDATE, and DELETE statements in a READ COMMITTED transaction will wait for data locks on their targeted rows by other transactions to be released.

* INSERT, UPDATE, and DELETE statements in a SERIALIZABLE transaction will fail if their targeted rows has data locks owned by other transactions.

[Read More Answers.](#)

Question # 174

How To Experiment a Data Lock in Oracle?

Answer:-

If you want to have some experience with data locks, you can create two windows runing two SQL*Plus sessions. In session 1, you can run a UPDATE statements to create a data lock. Before committing session 2, switch to session 2, and run a UPDATE statements on the same row as session 1. The UPDATE statement will be put into wait status because of the data lock. Follow the tutorial exercise below to experience yourself:

(session 1)

```
SQL> connect HR/globalguideline
```



```
SQL> SET TRANSACTION
ISOLATION LEVEL
READ COMMITTED;
Transaction set.
SQL> SELECT * FROM ggl_links;
ID URL          NOTES
-----
101 globalguideline.COM
110 globalguideline.COM
SQL> UPDATE ggl_links
SET url='globalguideline.com'
WHERE id=101;
1 row updated.
(lock created on row id=101)
```

[Read More Answers.](#)

Question # 175

How To View Existing Locks on the Database?

Answer:-

As can be seen from the previous tutorial exercise, performance of the second session is greatly affected by the data lock created on the database. To maintain a good performance level for all sessions, you need to monitor the number of data locks on the database, and how long do they last. Oracle maintains current existing data locks in a Dynamic Performance View called V\$LOCK with columns like:

- * SID - Session ID to identify the session that owns this lock.
- * TYPE - The type of the lock, like TM (DML enqueue) and TX (Transaction enqueue).
- * LMODE - The lock mode in which the session holds the lock.
- * REQUEST - The lock mode in which the session requests the lock.
- * CTIME - The time since current lock mode was granted
- * BLOCK - A value of either 0 or 1, depending on whether or not the lock in question is the blocker.

The following tutorial exercise shows you how to view existing locks on the database:

```
(session 1)
SQL> connect HR/globalguideline
SQL> UPDATE ggl_links
SET url='globalguideline.com'
WHERE id=110;
1 row updated.
```

[Read More Answers.](#)

Question # 176

What Is a Dead Lock in Oracle?

Answer:-

A dead lock is a phenomenon that happens between two transactions with each of them holding a lock that blocks the other transaction as shown in the following diagram:

```
<pre>(transaction 1)      (transaction 2)
update row X to create lock 1
                        update row Y to create lock 2
                        update row X
                        (blocked by lock 1)
update row Y</pre>
(blocked by lock 2)
(dead lock created)
```

[Read More Answers.](#)

Question # 177

How Oracle Handles Dead Locks?

Answer:-

Oracle server automatically detects dead locks. When a dead lock is detected, Oracle server will select a victim transaction, and fail its statement that is blocked in the dead lock to break the dead lock. The tutorial exercise below shows you an example of statements failed by Oracle server because of dead locks:

```
(session 1)
SQL> connect HR/globalguideline
SQL> UPDATE ggl_links
SET notes='Session 1'
WHERE id=101;
1 row updated.
<pre>
(session 2)
SQL> connect HR/globalguideline
SQL> UPDATE ggl_links
SET notes='Session 2'
WHERE id=110;
1 row updated.

SQL> UPDATE ggl_links
SET notes='Session 2'
WHERE id=101;
(blocked by lock id=101)
SQL> UPDATE ggl_links
SET notes='Session 1'
WHERE id=110;
```




(blocked by lock on row id=110)
ORA-00060: deadlock
detected w

[Read More Answers.](#)

Question # 178

What Is an Oracle User Account?

Answer:-

A user account is identified by a user name and defines the user's attributes, including the following:

- * Password for database authentication
- * Privileges and roles
- * Default tablespace for database objects
- * Default temporary tablespace for query processing work space

[Read More Answers.](#)

Question # 179

What Is the Relation of a User Account and a Schema?

Answer:-

User accounts and schemas have a one-to-one relation. When you create a user, you are also implicitly creating a schema for that user. A schema is a logical container for the database objects (such as tables, views, triggers, and so on) that the user creates. The schema name is the same as the user name, and can be used to unambiguously refer to objects owned by the user.

[Read More Answers.](#)

Question # 180

What Is an Oracle User Role?

Answer:-

A user role is a group of privileges. Privileges are assigned to users through user roles. You create new roles, grant privileges to the roles, and then grant roles to users.

[Read More Answers.](#)

Question # 181

What Are the System Predefined User Roles?

Answer:-

Oracle 10g XE comes with 3 predefined roles:

- * CONNECT - Enables a user to connect to the database. Grant this role to any user or application that needs database access.
- * RESOURCE - Enables a user to create certain types of schema objects in his own schema. Grant this role only to developers and to other users that must create schema objects. This role grants a subset of the create object system privileges.
- * DBA - Enables a user to perform most administrative functions, including creating users and granting privileges; creating and granting roles; creating and dropping schema objects in other users' schemas; and more. It grants all system privileges, but does not include the privileges to start up or shut down the database. It is by default granted to user SYSTEM.

[Read More Answers.](#)

Question # 182

What Are Internal User Account in Oracle?

Answer:-

An internal user account is a system predefined user account. Oracle 10g XE comes with a number of internal accounts:

- * SYSTEM - This is the user account that you log in with to perform all administrative functions other than starting up and shutting down the database. SYSTEM is automatically created when you install the server. Its password is the one you specified during the installation process.
- * SYS - This is another user account automatically created when you install the server. Its password is the one you specified during the installation process. All base tables and views for the database data dictionary are stored in the SYS schema. So avoid log in as user SYS as much as possible to reduce the risk of damaging those important data objects. User SYSTEM is preferred for all administrative tasks except starting up and shutting down.
- * Other internal user accounts - Other special user accounts are predefined for special purposes. For example, CTXSYS is a special user account used by the Oracle Text product.

[Read More Answers.](#)

Question # 183

How To Connect to the Server with User Account: SYS?

Answer:-

SYS is a very special user account. It has been associated with the highest privilege call SYSDBA. Normally, you should not connect to the server with SYS. But if you want to use it, you need to use a special connect command:

```
>cd (OracleXE home directory)
>. insqlplus /nolog
SQL> connect SYS/globalguideline AS SYSDBA
Connected.
SQL> quit
```

Note that the "/nolog" option is used to start SQL*Plus without login immediately. A special form of the "connect" command is used to include the user name, password, and the privilege in the same line.

You can not log in with SYS without SYSDBA privilege.



[Read More Answers.](#)

Question # 184

How To Use Windows User to Connect to the Server?

Answer:-

During the installation process, 10g XE will create a special Windows user group called ORA_DBA, and put your Windows user into this group. Any Windows users in this group can be connected to Oracle server with SYSDBA privilege without any Oracle server user account. This process is called connecting the server as SYSDBA with OS Authentication. Here is how to do this with a special form of the "connect" command:

(Log in with the same user you used to install 10g XE)

```
>cd (OracleXE home directory)
```

```
>. instatldb
```

```
>. insqlplus /nolog
```

```
SQL> connect / AS SYSDBA
```

```
Connected.
```

```
SQL> quit
```

So if "connect" is used without user name and password, the current Windows user will be trusted if he/she is in the ORA_DBA user group on the Windows system.

[Read More Answers.](#)

Question # 185

How To List All User Accounts in Oracle?

Answer:-

User accounts can be accessed through a system view called ALL_USERS. A simple SELECT statement can be used to get a list of all user accounts. Try the following script:

```
>. insqlplus /nolog
```

```
SQL> connect SYSTEM/globalguideline
```

```
Connected.
```

```
SQL> SELECT * FROM ALL_USERS;
```

```
<pre>USERNAME                USER_ID CREATED
-----
FLOWS_020100              35 07-FEB-06
FLOWS_FILES                34 07-FEB-06
HR                          33 07-FEB-06
MDSYS                      32 07-FEB-06
ANONYMOUS                  28 07-FEB-06
XDB                         27 07-FEB-06
CTXSYS                     25 07-FEB-06
DBSNMP                     23 07-FEB-06
TSMSYS                     20 07-FEB-06
DIP                         18 07-FEB-06
OUTLN                      11 07-FEB-06
SYSTEM                     5 07-FEB-06
SYS                          0 07-FEB-06</pre>
```

[Read More Answers.](#)

Question # 186

How To Create a New User Account in Oracle?

Answer:-

If you want to create a new user account, you can log in as SYSTEM and use the CREATE USER command as shown in the following example:

```
>. insqlplus /nolog
```

```
SQL> connect SYSTEM/globalguideline
```

```
Connected.
```

```
SQL> CREATE USER DEV IDENTIFIED BY developer ACCOUNT UNLOCK;
```

```
User created.
```

Note that CREATE is a SQL statement, so you need to terminate it with ";". This command creates a user called "DEV", with a password of "developer". You can test this account by log in with DEV from SQL*Plus.

[Read More Answers.](#)

Question # 187

How To Change User Password in Oracle?

Answer:-

If you want to change a user's password, you can log in as SYSTEM and use the ALTER USER command as shown in the following example:

```
>. insqlplus /nolog
```

```
SQL> connect SYSTEM/globalguideline
```

```
Connected.
```

```
SQL> ALTER USER DEV IDENTIFIED BY beginner;
```

```
User altered.
```

Note that ALTER is SQL statement, so you need to terminate it with ";". This command resets DEV's password to "beginner".

[Read More Answers.](#)

Question # 188

How To Delete a User Account in Oracle?



Answer:-

If you want to delete a user account and its associated schema, you can log in as SYSTEM and use the DROP USER command as shown in the following example:

```
>. insqlplus /nolog
SQL> connect SYSTEM/globalguideline
Connected.
SQL> DROP USER DEV CASCADE;
User dropped.
SQL> CREATE USER DEV IDENTIFIED BY developer ACCOUNT UNLOCK;
User created.
Note that CASCADE tells the server drop the associated schema.
```

[Read More Answers.](#)

Question # 189

What Privilege Is Needed for a User to Connect to Oracle Server?

Answer:-

Oracle deny connection to users who has no CREATE SESSION privilege. Try the following tutorial exercise, you will find out how Oracle denies connection:

```
>. insqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> CREATE USER DEV IDENTIFIED BY developer ACCOUNT UNLOCK;
User created.
SQL> disconnect
SQL> CONNECT DEV/developer
ORA-01045: user DEV lacks CREATE SESSION privilege;
  logon denied
Oracle error message is pretty clear.
```

[Read More Answers.](#)

Question # 190

How To Grant CREATE SESSION Privilege to a User in Oracle?

Answer:-

If you want give a user the CREATE SESSION privilege, you can use the GRANT command. The following tutorial exercise shows you how to grant DEV the privilege to connect to the server:

```
>. insqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> GRANT CREATE SESSION TO dev;
Grant succeeded.
SQL> disconnect
SQL> CONNECT DEV/developer
Connected.
```

[Read More Answers.](#)

Question # 191

How To Revoke CREATE SESSION Privilege from a User in Oracle?

Answer:-

If you take away the CREATE SESSION privilege from a user, you can use the REVOKE command as shown in the following example script:

```
>. insqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> REVOKE CREATE SESSION FROM dev;
Revoke succeeded.
SQL> GRANT CREATE SESSION TO dev;
Grant succeeded.
This script restored the CREATE SESSION privilege to user "dev", so you can continue other example scripts below.
```

[Read More Answers.](#)

Question # 192

How To Lock and Unlock a User Account in Oracle?

Answer:-

If you want to lock a user account for a short period of time, and unlock it later, you can use the ALTER USER ... ACCOUNT command. The following sample script shows how to use this command:

```
>. insqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> ALTER USER dev ACCOUNT LOCK;
User altered.
SQL> disconnect
SQL> CONNECT DEV/developer
ORA-28000: the account is locked
SQL> disconnect
SQL> connect SYSTEM/globalguideline
SQL> ALTER USER dev ACCOUNT UNLOCK;
User altered.
SQL> disconnect
SQL> CONNECT DEV/developer
Connected.
```



[Read More Answers.](#)

Question # 193

What Privilege Is Needed for a User to Create Tables in Oracle?

Answer:-

To be able to create tables in a user's own schema, the user needs to have the CREATE TABLE privilege, or the CREATE ANY TABLE privilege, which is more powerful, and allows the user to create tables in other user's schema. The following tutorial exercise gives you a good example on CREATE TABLE privilege:

```
>. insqlplus /nolog
SQL> CONNECT DEV/developer
SQL> CREATE TABLE ggl (id NUMBER);
ORA-01031: insufficient privileges
SQL> disconnect
SQL> connect SYSTEM/globalguideline
SQL> GRANT CREATE TABLE TO dev;
Grant succeeded.
SQL> disconnect
SQL> CONNECT DEV/developer
SQL> CREATE TABLE ggl (id NUMBER);
ORA-01950: no privileges on tablespace 'SYSTEM'
The above error message tells that user "dev" is not allowed to use the tablespace "SYSTEM". See the next question for answers.
```

[Read More Answers.](#)

Question # 194

How To Assign a Tablespace to a Users in Oracle?

Answer:-

When you create a new user, Oracle will assign the SYSTEM tablespace to the user by default. If you want to change this, you can assign a different table space to a user using the ALTER USER command. The following tutorial exercise changes user dev's default tablespace, and assigns 4MB of space to dev:

```
>. insqlplus /nolog
SQL> CONNECT DEV/developer
SQL> ALTER USER dev DEFAULT TABLESPACE USERS;
User altered.
SQL> ALTER USER dev QUOTA 4M ON USERS;
User altered.
SQL> disconnect
SQL> CONNECT DEV/developer
SQL> CREATE TABLE ggl (id NUMBER);
Table created.
SQL> DROP TABLE ggl;
Table dropped.
SQL> CREATE TABLE ggl (id NUMBER);
Table created.
```

As you can see, "dev" can create and drop tables now. You can also let "dev" to create tables in any tablespace without any restriction by granting him the UNLIMITED TABLESPACE system privilege.

[Read More Answers.](#)

Question # 195

What Privilege Is Needed for a User to Create Views in Oracle?

Answer:-

To be able to create views in a user's own schema, the user needs to have the CREATE VIEW privilege, or the CREATE ANY VIEW privilege, which is more powerful, and allows the user to create views in other user's schema. The following tutorial exercise gives you a good example on CREATE VIEW privilege:

```
>. insqlplus /nolog
SQL> CONNECT DEV/developer
SQL> CREATE VIEW ggl_view AS SELECT * FROM ggl;
ORA-01031: insufficient privileges
SQL> disconnect
SQL> connect SYSTEM/globalguideline
SQL> GRANT CREATE VIEW TO dev;
Grant succeeded.
SQL> disconnect
SQL> CONNECT DEV/developer
SQL> CREATE VIEW ggl_view AS SELECT * FROM ggl;
View created.
SQL> DROP VIEW ggl_view;
View dropped.
SQL> CREATE VIEW ggl_view AS SELECT * FROM ggl;
View created.
```

As you can see, "dev" can create and drop views now.

[Read More Answers.](#)

Question # 196

What Privilege Is Needed for a User to Create Indexes in Oracle?

Answer:-

For a user to create indexes, he/she needs the same privilege as the creating tables. Just make sure he/she has the CREATE TABLE privilege. The following tutorial



exercise gives you a good example on creating view privilege:

```
>. insqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> GRANT CREATE TABLE TO dev;
Grant succeeded.
SQL> disconnect
SQL> CONNECT DEV/developer
SQL> CREATE INDEX ggl_index ON ggl(id);
Index created.
SQL> DROP INDEX ggl_index;
Index dropped.
SQL> CREATE INDEX ggl_index ON ggl(id);
Index created.
```

[Read More Answers.](#)

Question # 197

What Privilege Is Needed for a User to Query Tables in Another Schema?

Answer:-

For a user to run queries (SELECT statements) on tables of someone else's schema, he/she needs the SELECT ANY TABLE privilege. The following tutorial exercise gives you a good example of granting "dev" to query tables in "hr" schema:

```
>. insqlplus /nolog
SQL> CONNECT DEV/developer
SQL> SELECT COUNT(*) FROM hr.employees;
ORA-01031: insufficient privileges
SQL> disconnect
SQL> connect SYSTEM/globalguideline
SQL> GRANT SELECT ANY TABLE TO dev;
Grant succeeded.
SQL> disconnect
SQL> CONNECT DEV/developer
SQL> SELECT COUNT(*) FROM hr.employees;
COUNT(*)
```

107

As you can see, "dev" can query tables in any schema now.

You also need to remember that table name must be prefixed with the schema name (same as owner user name).

[Read More Answers.](#)

Question # 198

What Privilege Is Needed for a User to Insert Rows to Tables in Another Schema?

Answer:-

For a user to insert rows into tables of someone else's schema, he/she needs the INSERT ANY TABLE privilege. The following tutorial exercise gives you a good example of granting "dev" to insert rows in "hr" schema:

```
>. insqlplus /nolog
SQL> CONNECT DEV/developer
SQL> INSERT INTO hr.jobs
VALUES ('DV.ggl', 'Dev ggl Consultant', 7700, 8800);
ORA-01031: insufficient privileges
SQL> disconnect
SQL> connect SYSTEM/globalguideline
SQL> GRANT INSERT ANY TABLE TO dev;
Grant succeeded.
SQL> disconnect
SQL> CONNECT DEV/developer
SQL> INSERT INTO hr.jobs
VALUES ('DV.ggl', 'Dev ggl Consultant', 7700, 8800);
1 row created.
```

As you can see, "dev" can insert rows in any schema now. But you should be careful when giving this privilege to a regular developer.

[Read More Answers.](#)

Question # 199

What Privilege Is Needed for a User to Delete Rows from Tables in Another Schema?

Answer:-

For a user to delete rows from tables of someone else's schema, he/she needs the DELETE ANY TABLE privilege. The following tutorial exercise gives you a good example of granting "dev" to delete rows in "hr" schema:

```
>. insqlplus /nolog
SQL> CONNECT DEV/developer
SQL> DELETE FROM hr.jobs WHERE job_id = 'DV.ggl';
ORA-01031: insufficient privileges
SQL> disconnect
SQL> connect SYSTEM/globalguideline
SQL> GRANT DELETE ANY TABLE TO dev;
Grant succeeded.
SQL> disconnect
SQL> CONNECT DEV/developer
SQL> DELETE FROM hr.jobs WHERE job_id = 'DV.ggl';
```



1 row deleted.

As you can see, "dev" can delete rows in any schema now. But you should be careful when giving this privilege to a regular developer.

[Read More Answers.](#)

Question # 200

How To Find Out What Privileges a User Currently Has in Oracle?

Answer:-

Privileges granted to users are listed in two system views: DBA_SYS_PRIVS, and USER_SYS_PRIVS. You can find out what privileges a user currently has by running a query on those views as shown in the tutorial exercise below:

```
>. insqlplus /nolog
SQL> CONNECT DEV/developer
SQL> SELECT username, privilege FROM USER_SYS_PRIVS;


```

<pre>USERNAME PRIVILEGE

DEV SELECT ANY TABLE
DEV INSERT ANY TABLE
DEV CREATE SESSION
DEV CREATE VIEW
DEV DELETE ANY TABLE
DEV CREATE ANY TABLE</pre>
SQL> disconnect
SQL> connect SYSTEM/globalguideline
```


```

[Read More Answers.](#)

Question # 201

What Is an Oracle Database Table?

Answer:-

A database table is a basic unit of data logical storage in an Oracle database. Data is stored in rows and columns. You define a table with a table name, such as employees, and a set of columns. You give each column a column name, such as employee_id, last_name, and job_id; a datatype, such as VARCHAR2, DATE, or NUMBER; and a width. The width can be predetermined by the datatype, as in DATE. If columns are of the NUMBER datatype, define precision and scale instead of width. A row is a collection of column information corresponding to a single record.

[Read More Answers.](#)

Question # 202

How Many Types of Tables Supported by Oracle?

Answer:-

Oracle supports 4 types of tables based on how data is organized in storage:

- * Ordinary (heap-organized) table - This is the basic, general purpose type of table. Its data is stored as an unordered collection (heap)
- * Clustered table - A clustered table is a table that is part of a cluster. A cluster is a group of tables that share the same data blocks because they share common columns and are often used together.
- * Index-organized table - Unlike an ordinary (heap-organized) table, data for an index-organized table is stored in a B-tree index structure in a primary key sorted manner. Besides storing the primary key column values of an index-organized table row, each index entry in the B-tree stores the nonkey column values as well.
- * Partitioned table - Partitioned tables allow your data to be broken down into smaller, more manageable pieces called partitions, or even subpartitions. Each partition can be managed individually, and can operate independently of the other partitions, thus providing a structure that can be better tuned for availability and performance.

[Read More Answers.](#)

Question # 203

How To Create a New Table in Your Schema?

Answer:-

If you want to create a new table in your own schema, you can log into the server with your account, and use the CREATE TABLE statement. The following script shows you how to create a table:

```
>. insqlplus /nolog
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE tip (id NUMBER(5) PRIMARY KEY,
  2  subject VARCHAR(80) NOT NULL,
  3  description VARCHAR(256) NOT NULL,
  4  create_date DATE DEFAULT (sysdate));
Table created.
```

This script creates a testing table called "tip" with 4 columns in the schema associated with the log in account "HR".

[Read More Answers.](#)

Question # 204

How To Rename an Existing Table?

Answer:-

If you don't like the name of an existing table, you change it by using the CREATE TABLE ... RENAME TO statement. Here is a sample script:

```
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE emp_dept_10
  2 AS SELECT * FROM employees WHERE department_id=10;
```



Table created.
SQL> ALTER TABLE emp_dept_10 RENAME TO emp_dept_dba;
Table altered.
SQL> SELECT first_name, last_name, salary FROM emp_dept_dba;
<pre>FIRST_NAME LAST_NAME SALARY

Jennifer Whalen 4400</pre>

[Read More Answers.](#)

Question # 205

How To Drop an Existing Table?

Answer:-

If you want to delete an existing table and its data rows, you can use the DROP TABLE statement as shown in this script:

```
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE emp_dept_10
  2 AS SELECT * FROM employees WHERE department_id=10;
Table created.
SQL> DROP TABLE emp_dept_10;
Table dropped.
Be careful, when you use the DROP TABLE statement. All data rows are gone too.
```

[Read More Answers.](#)

Question # 206

How To Add a New Column to an Existing Table?

Answer:-

If you have an existing table with existing data rows, and want to add a new column to that table, you can use the ALTER TABLE ... ADD statement to do this. Here is an example script:

```
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE emp_dept_110
  2 AS SELECT * FROM employees WHERE department_id=110;
Table created.
SQL> ALTER TABLE emp_dept_110 ADD (vacation NUMBER);
Table altered.
SQL> SELECT first_name, last_name, vacation
  2 FROM emp_dept_110;
<pre>FIRST_NAME      LAST_NAME      VACATION  
-----  
Shelley      Higgins  
William      Gietz</pre>
```

This SQL script added a new column called "vacation" to the "emp_dept_110" table. NULL values were added to this column on all existing data rows.

[Read More Answers.](#)

Question # 207

How To Add a New Column to an Existing Table with a Default Value?

Answer:-

If you want to add a new column to an existing table, and insert a default value in this column on all existing data rows, you can use the ALTER TABLE ... ADD statement with the DEFAULT clause. Here is an example script:

```
SQL> CREATE TABLE emp_dept_90
  2 AS SELECT * FROM employees WHERE department_id=90;
Table created.
SQL> ALTER TABLE emp_dept_90
  2 ADD (vacation NUMBER DEFAULT 10);
Table altered.
SQL> SELECT first_name, last_name, vacation
  2 FROM emp_dept_90;
<pre>FIRST_NAME      LAST_NAME      VACATION  
-----  
Steven      King      10  
Neena      Kochhar      10  
Lex      De Haan      10</pre>
```

As you can see, the "DEFAULT 10" clause did inserted 10 to all existing data rows.

[Read More Answers.](#)

Question # 208

How To Rename a Column in an Existing Table?

Answer:-

Let's say you have an existing with an existing column, but you don't like the name of that column, can you rename that column name? The answer is yes. You can use the ALTER TABLE ... RENAME COLUMN statement to do this. See the following SQL script:

```
SQL> CREATE TABLE emp_dept_90
  2 AS SELECT * FROM employees WHERE department_id=90;
Table created.
```



```
SQL> SELECT first_name, last_name FROM emp_dept_90;

```

[Read More Answers.](#)

Question # 209

How To Delete a Column in an Existing Table?

Answer:-

If you have an existing column in a table and you need that column any more, you can delete it with ALTER TABLE ... DROP COLUMN statement. Here is an example SQL script:

```
SQL> CREATE TABLE emp_dept_90
  2 AS SELECT * FROM employees WHERE department_id=90;
Table created.
SQL> SELECT last_name FROM emp_dept_90;
LAST_NAME
-----
King
Kochhar
De Haan
SQL> ALTER TABLE emp_dept_90 DROP COLUMN last_name;
Table altered.
SQL> SELECT last_name FROM emp_dept_90;
ERROR at line 1:
ORA-00904: "LAST_NAME": invalid identifier
As you can see the column "last_name" is gone.
```

[Read More Answers.](#)

Question # 210

How To View All Columns in an Existing Table?

Answer:-

If you have an existing table and want to know how many columns are in the table and how they are defined, you can use the system view USER_TAB_COLUMNS as shown in the following tutorial exercise:

```
SQL> COL data_type FORMAT A12;
SQL> SELECT column_name, data_type, data_length
  FROM user_tab_columns WHERE table_name = 'EMPLOYEES';

```

[Read More Answers.](#)

Question # 211

How To Recover a Dropped Table in Oracle?

Answer:-

If you accidentally dropped a table, can you recover it back? The answer is yes, if you have the recycle bin feature turned on. You can use the FLASHBACK TABLE ... TO BEFORE DROP statement to recover a dropped table from the recycle bin as shown in the following SQL script:

```
SQL> CREATE TABLE emp_dept_90
  2 AS SELECT * FROM employees WHERE department_id=90;
Table created.
SQL> SELECT COUNT(*) FROM emp_dept_90;
COUNT(*)
-----
3
SQL> DROP TABLE emp_dept_90;
```




Table dropped.

```
SQL> FLASHBACK TABLE emp_dept_90 TO BEFORE DROP  
  2 RENAME TO emp_dept_bck;
```

Flashback complete.

```
SQL> SELECT COUNT(*) FROM emp_dept_bck;  
COUNT(*)
```

```
-----  
  3  
The FLASHBACK statement in this script recovered the dropped table "emp_dept_90" to new name "emp_dept_bck". All the data rows are recovered nicely.
```

[Read More Answers.](#)

Question # 212

What Is an Oracle Recycle Bin?

Answer:-

Recycle bin is a logical storage to hold the tables that have been dropped from the database, in case it was dropped in error. Tables in recycle bin can be recovered back into database by the Flashback Drop action. Oracle database recycle save the same purpose as the recycle bin on your Windows desktop. Recycle bin can be turned on or off in the recyclebin=on/off in your parameter file.

[Read More Answers.](#)

Question # 213

How To Turn On or Off Recycle Bin for the Instance?

Answer:-

You can turn on or off the recycle bin feature for an instance in the instance parameter file with "recyclebin=on/off". You can also turn on or off the recycle bin feature on the running instance with a SQL*Plus command, if you log in as SYSTEM. See the following example:

```
SQL> connect SYSTEM/globalguideline
```

Connected.

```
SQL> SHOW PARAMETERS RECYCLEBIN  
<pre>NAME                TYPE        VALUE
```

```
-----  
recyclebin            string      on</pre>
```

```
SQL> ALTER SYSTEM SET RECYCLEBIN = OFF;  
System altered.
```

```
SQL> SHOW PARAMETERS RECYCLEBIN  
<pre>NAME                TYPE        VALUE
```

```
-----  
recyclebin            string      OFF</pre>
```

Warning: Turning off the recycle bin feature will give your users hard times on recovering dropped tables.

[Read More Answers.](#)

Question # 214

How To View the Dropped Tables in Your Recycle Bin?

Answer:-

You can look what's in your recycle bin through the predefined view called RECYCLEBIN. You can use the SELECT statement to list the dropped tables as shown in the following script:

```
SQL> connect HR/globalguideline
```

Connected.

```
SQL> CREATE TABLE emp_dept_90  
  2 AS SELECT * FROM employees WHERE department_id=90;
```

Table created.

```
SQL> SELECT COUNT(*) FROM emp_dept_90;  
COUNT(*)
```

```
-----  
  3  
SQL> DROP TABLE emp_dept_90;  
Table dropped.
```

```
SQL> COL original_name FORMAT A14  
SQL> SELECT object_name, original_name, droptime
```

```
  2 FROM recyclebin;  
<pre>OBJECT_NAME                ORIGINAL_NAME DROPTIME
```

```
-----  
BIN$uaSS/heeQuys53HgXRhEEQ==$ EMP_DEPT_10 06-04-01:18:57:  
BIN$gSt95r7ATKGUPuALIH4dw==$ EMP_DEPT_10 06-04-01:19:59:  
BIN$bLukbcgSQ6mK1P2QVRf+fQ==$ EMP_DEPT_90 06-04-01:20:47:</pre>
```

As you can see the EMP_DEPT_10 was dropped twice. If the same table was dropped multiple times, you need to restore by using the object name in the recycle bin with FLASHBACK statement.

Note that RECYCLEBIN is just an alias of USER_RECYCLEBIN.

[Read More Answers.](#)

Question # 215

How To Empty Your Oracle Recycle Bin?

Answer:-

If your recycle bin is full, or you just want to clean your recycle bin to get rid of all the dropped tables, you can empty it by using the PURGE statement in two formats:



- * PURGE RECYCLEBIN - Removes all dropped tables from your recycle bin.
- * PURGE TABLE table_name - Removes the specified table from your recycle bin.

Here is an example script on how to use the PURGE statement:

```
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE emp_dept_10
  2 AS SELECT * FROM employees WHERE department_id=10;
Table created.
SQL> DROP TABLE emp_dept_10;
Table dropped.
SQL> CREATE TABLE emp_dept_90
  2 AS SELECT * FROM employees WHERE department_id=90;
Table created.
SQL> DROP TABLE emp_dept_90;
Table dropped.
SQL> SELECT COUNT(*) FROM recyclebin;
COUNT(*)
-----
5
SQL> PURGE TABLE emp_dept_90;
Table purged.
SQL> SELECT COUNT(*) FROM recyclebin;
COUNT(*)
-----
4
SQL> PURGE RECYCLEBIN;
Recyclebin purged.
SQL> SELECT COUNT(*) FROM recyclebin;
COUNT(*)
-----
0
```

[Read More Answers.](#)

Question # 216

How To Turn On or Off Recycle Bin for the Session?

Answer:-

If you want to control the recycle bin feature in your own session, you can use the ALTER SESSION statement to turn on or off. Here is an example SQL script:

```
SQL> connect HR/globalguideline
Connected.
SQL> SELECT COUNT(*) FROM recyclebin;
COUNT(*)
-----
0
SQL> ALTER SESSION SET recyclebin = off;
Session altered.
SQL> CREATE TABLE emp_dept_90
  2 AS SELECT * FROM employees WHERE department_id=90;
Table created.
SQL> DROP TABLE emp_dept_90;
Table dropped.
SQL> SELECT COUNT(*) FROM recyclebin;
COUNT(*)
-----
0
```

Warning: Turning off the recycle bin feature in your session will give yourself hard times on recovering dropped tables.

[Read More Answers.](#)

Question # 217

How To List All Tables in Your Schema?

Answer:-

If you log in with your Oracle account, and you want to get a list of all tables in your schema, you can get it through the USER_TABLES view with a SELECT statement, as shown in the following SQL script:

```
SQL> connect HR/globalguideline
Connected.
SQL> SELECT table_name, status, num_rows FROM USER_TABLES;
<pre>TABLE_NAME          STATUS    NUM_ROWS
-----
REGIONS              VALID     4
LOCATIONS             VALID    23
DEPARTMENTS          VALID    27
JOBS                  VALID    19
EMPLOYEES             VALID   107
JOB_HISTORY           VALID    10
COUNTRIES             VALID   25</pre>
7 rows selected.
```

[Read More Answers.](#)



Question # 218

What Is a Table Index?

Answer:-

Index is an optional structure associated with a table that allow SQL statements to execute more quickly against a table. Just as the index in this manual helps you locate information faster than if there were no index, an Oracle Database index provides a faster access path to table data. You can use indexes without rewriting any queries. Your results are the same, but you see them more quickly.

[Read More Answers.](#)

Question # 219

How To Run SQL Statements through the Web Interface?

Answer:-

If you don't like the command line interface offered by SQL*Plus, you can use the Web interface to run SQL statements. Here is how:

- * Open your Web browser to <http://localhost:8080/apex/>
- * Log in to the server with the predefined sample user account: HR/globalguideline
- * Click the SQL icon
- * Click the SQL Commands icon
- * Enter any SQL statement like: "SELECT COUNT(*) FROM USER_TABLES;", in the text area and click Run button
- * Your Oracle will execute the statement, and display the result in the result area.

[Read More Answers.](#)

Question # 220

How To Create a Table Index?

Answer:-

If you have a table with a lots of rows, and you know that one of the columns will be used often a search criteria, you can add an index for that column to in improve the search performance. To add an index, you can use the CREATE INDEX statement as shown in the following script:

```
CREATE TABLE tip (id NUMBER(5) PRIMARY KEY,  
subject VARCHAR(80) NOT NULL,  
description VARCHAR(256) NOT NULL,  
create_date DATE DEFAULT (sysdate));  
Table created.
```

```
CREATE INDEX tip_subject ON tip(subject);  
Index created.
```

[Read More Answers.](#)

Question # 221

How To List All Indexes in Your Schema?

Answer:-

If you log in with your Oracle account, and you want to get a list of all indexes in your schema, you can get it through the USER_INDEXES view with a SELECT statement, as shown in the following SQL script:

```
SELECT index_name, table_name, uniqueness  
FROM USER_INDEXES WHERE table_name = 'EMPLOYEES';  
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES  
-----  
EMP_EMAIL_UK    EMPLOYEES        UNIQUE  
EMP_EMP_ID_PK   EMPLOYEES        UNIQUE  
EMP_DEPARTMENT_IX  EMPLOYEES        NONUNIQUE  
EMP_JOB_IX       EMPLOYEES        NONUNIQUE  
EMP_MANAGER_IX   EMPLOYEES        NONUNIQUE  
EMP_NAME_IX      EMPLOYEES        NONUNIQUE</pre>
```

As you can see, the pre-defined table EMPLOYEES has 6 indexes defined in the default sample database.

[Read More Answers.](#)

Question # 222

What Is an Index Associated with a Constraint?

Answer:-

An index associated with a constraint because this constraint is required to have an index. There are two types of constraints are required to have indexes: UNIQUE and PRIMARY KEY. When you defines a UNIQUE or PRIMARY KEY constraint in a table, Oracle will automatically create an index for that constraint. The following script shows you an example:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,  
first_name VARCHAR(80) NOT NULL,  
last_name VARCHAR(80) NOT NULL,  
birth_date DATE NOT NULL,  
social_number VARCHAR(80) UNIQUE NOT NULL);  
Table created.
```

```
SELECT index_name, table_name, uniqueness  
FROM USER_INDEXES WHERE table_name = 'STUDENT';  
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES  
-----  
SYS_C004123     STUDENT          UNIQUE  
SYS_C004124     STUDENT          UNIQUE</pre>
```



The result confirms that Oracle automatically created two indexes for you.

[Read More Answers.](#)

Question # 223

How To Rename an Index?

Answer:-

Let's say you have an existing index, and you don't like its name anymore for some reason, you can rename it with the ALTER INDEX ... RENAME TO statement.

Here is an example script on how to rename an index:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
first_name VARCHAR(80) NOT NULL,
last_name VARCHAR(80) NOT NULL,
birth_date DATE NOT NULL,
social_number VARCHAR(80) UNIQUE NOT NULL);
Table created.
```

```
SELECT index_name, table_name, uniqueness
FROM USER_INDEXES WHERE table_name = 'STUDENT';

```

```
SYS_C004153      STUDENT          UNIQUE
SYS_C004154      STUDENT          UNIQUE</pre>
ALTER INDEX SYS_C004153 RENAME TO student_pk;
```

Statement processed.

```
SELECT index_name, table_name, uniqueness
FROM USER_INDEXES WHERE table_name = 'STUDENT';

```

```
STUDENT_PK      STUDENT          UNIQUE
SYS_C004154      STUDENT          UNIQUE</pre>
```

[Read More Answers.](#)

Question # 224

How To Drop an Index?

Answer:-

If you don't need an existing index any more, you should delete it with the DROP INDEX statement. Here is an example SQL script:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
first_name VARCHAR(80) NOT NULL,
last_name VARCHAR(80) NOT NULL,
birth_date DATE NOT NULL,
social_number VARCHAR(80) UNIQUE NOT NULL);
Table created.
```

```
CREATE INDEX student_birth_date ON student(birth_date);
```

Index created.

```
SELECT index_name, table_name, uniqueness
FROM USER_INDEXES WHERE table_name = 'STUDENT';

```

```
SYS_C004129      STUDENT          UNIQUE
SYS_C004130      STUDENT          UNIQUE
STUDENT_BIRTH_DATE STUDENT          NONUNIQUE</pre>
```

```
DROP INDEX STUDENT_BIRTH_DATE;
```

Index dropped.

[Read More Answers.](#)

Question # 225

Can You Drop an Index Associated with a Unique or Primary Key Constraint?

Answer:-

You can not delete the index associated with a unique or primary key constraint. If you try, you will get an error like this: ORA-02429: cannot drop index used for enforcement of unique/primary key.

[Read More Answers.](#)

Question # 226

What Happens to Indexes If You Drop a Table?

Answer:-

If you drop a table, what happens to its indexes? The answer is that if a table is dropped, all its indexes will be dropped too. Try the following script to see yourself:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
first_name VARCHAR(80) NOT NULL,
last_name VARCHAR(80) NOT NULL,
birth_date DATE NOT NULL,
social_number VARCHAR(80) UNIQUE NOT NULL);
Table created.
```

```
CREATE INDEX student_birth_date ON student(birth_date);
```

Index created.



```
SELECT index_name, table_name, uniqueness
  FROM USER_INDEXES WHERE table_name = 'STUDENT';
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES
-----
SYS_C004141      STUDENT          UNIQUE
SYS_C004142      STUDENT          UNIQUE
STUDENT_BIRTH_DATE  STUDENT          NONUNIQUE</pre>
DROP TABLE STUDENT;
Table dropped.
SELECT index_name, table_name, uniqueness
  FROM USER_INDEXES WHERE table_name = 'STUDENT';
no data found
No records in USER_INDEXES view found for your name STUDENT, after you dropped STUDENT table.
```

[Read More Answers.](#)

Question # 227

How To Recover a Dropped Index?

Answer:-

If you have the recycle bin feature turned on, dropped indexes are stored in the recycle bin. But it seems to be command to restore a dropped index out of the recycle bin. FLASHBACK INDEX is not a valid statement. See the following script:

```
ALTER SESSION SET recyclebin = on;
Statement processed.
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
  first_name VARCHAR(80) NOT NULL,
  last_name VARCHAR(80) NOT NULL,
  birth_date DATE NOT NULL,
  social_number VARCHAR(80) UNIQUE NOT NULL);
Table created.
CREATE INDEX student_birth_date ON student(birth_date);
Index created.
DROP INDEX STUDENT_BIRTH_DATE;
Index dropped.
SELECT object_name, original_name, type, droptime
  FROM recyclebin;
<pre>OBJECT_NAME      ORIGINAL_NAME      TYPE      DROPTIME
-----
BIN$1LlSjTxERKq+C7A==$0 STUDENT_BIRTH_DATE INDEX 2006-04-01</pre>
FLASHBACK INDEX student_birth_date TO BEFORE DROP;
ORA-00905: missing keyword
```

[Read More Answers.](#)

Question # 228

What Happens to the Indexes If a Table Is Recovered?

Answer:-

If you dropped a table, and recovered it back from the recycle bin, what happens to its indexes? Are all indexes recovered back automatically? The answer is that all indexes will be recovered, if you recover a dropped table from the recycle bin. However, the indexes' names will not be the original names. Indexes will be recovered with the system assigned names when they were dropped into the cycle bin. The following SQL script shows you this behavior:

```
ALTER SESSION SET recyclebin = on;
Statement processed.
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
  first_name VARCHAR(80) NOT NULL,
  last_name VARCHAR(80) NOT NULL,
  birth_date DATE NOT NULL,
  social_number VARCHAR(80) UNIQUE NOT NULL);
Table created.
CREATE INDEX student_birth_date ON student(birth_date);
Index created.
SELECT index_name, table_name, uniqueness
  FROM USER_INDEXES WHERE table_name = 'STUDENT';
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES
-----
SYS_C004141      STUDENT          UNIQUE
SYS_C004142      STUDENT          UNIQUE
STUDENT_BIRTH_DATE  STUDENT          NONUNIQUE</pre>
DROP TABLE student;
Table dropped.
```

[Read More Answers.](#)

Question # 229

How To Rebuild an Index in Oracle?

Answer:-

If you want to rebuild an index, you can use the "ALTER INDEX ... REBUILD" statement as shown in the following SQL script:

```
SELECT index_name, table_name, uniqueness
  FROM USER_INDEXES WHERE table_name = 'EMPLOYEES';
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES
-----
```



```
EMP_EMAIL_UK      EMPLOYEES      UNIQUE
EMP_EMP_ID_PK     EMPLOYEES      UNIQUE
EMP_DEPARTMENT_IX EMPLOYEES      NONUNIQUE
EMP_JOB_IX        EMPLOYEES      NONUNIQUE
EMP_MANAGER_IX   EMPLOYEES      NONUNIQUE
EMP_NAME_IX       EMPLOYEES      NONUNIQUE</pre>
ALTER INDEX EMP_NAME_IX REBUILD;
Statement processed.
```

[Read More Answers.](#)

Question # 230

How To See the Table Columns Used in an Index?

Answer:-

You can a list of indexes in your schema from the USER_INDEXES view, but it will not give you the columns used in each index in the USER_INDEXES view. If you want to see the columns used in an index, you can use the USER_IND_COLUMNS view. Here is an example script for you:

```
SELECT index_name, table_name, column_name
FROM USER_IND_COLUMNS WHERE table_name = 'EMPLOYEES';
<pre>INDEX_NAME      TABLE_NAME      COLUMN_NAME
```

```
-----
EMP_EMAIL_UK     EMPLOYEES      EMAIL
EMP_EMP_ID_PK    EMPLOYEES      EMPLOYEE_ID
EMP_DEPARTMENT_IX EMPLOYEES      DEPARTMENT_ID
EMP_JOB_IX       EMPLOYEES      JOB_ID
EMP_MANAGER_IX   EMPLOYEES      MANAGER_ID
EMP_NAME_IX      EMPLOYEES      LAST_NAME
EMP_NAME_IX      EMPLOYEES      FIRST_NAME</pre>
```

[Read More Answers.](#)

Question # 231

How To Create a Single Index for Multiple Columns?

Answer:-

If you know a group of multiple columns will be always used together as search criteria, you should create a single index for that group of columns with the "ON table_name(col1, col2, ...)" clause. Here is an example of one index for two columns:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
first_name VARCHAR(80) NOT NULL,
last_name VARCHAR(80) NOT NULL,
birth_date DATE NOT NULL,
social_number VARCHAR(80) UNIQUE NOT NULL);
Table created.
```

```
CREATE INDEX student_names ON student(first_name,last_name);
Index created.
```

```
SELECT index_name, table_name, uniqueness
FROM USER_INDEXES WHERE table_name = 'STUDENT';
<pre>INDEX_NAME      TABLE_NAME      UNIQUENES
```

```
-----
SYS_C004123     STUDENT          UNIQUE
SYS_C004124     STUDENT          UNIQUE
STUDENT_NAMES   STUDENT          NONUNIQUE</pre>
```

[Read More Answers.](#)

Question # 232

How a Tablespace Is Related to Data Files?

Answer:-

Each tablespace in an Oracle database consists of one or more files called datafiles, which are physical structures that conform to the operating system in which Oracle is running.

[Read More Answers.](#)

Question # 233

How a Database Is Related to Tablespaces?

Answer:-

A database's data is collectively stored in the datafiles that constitute each tablespace of the database. For example, the simplest Oracle database would have one tablespace and one datafile. Another database can have three tablespaces, each consisting of two datafiles (for a total of six datafiles).

[Read More Answers.](#)

Question # 234

How To View the Tablespaces in the Current Database?

Answer:-

If you want to get a list of all tablespaces used in the current database instance, you can use the DBA_TABLESPACES view as shown in the following SQL script example:

```
SQL> connect SYSTEM/globalguideline
```



Connected.

```
SQL> SELECT TABLESPACE_NAME, STATUS, CONTENTS
       2 FROM USER_TABLESPACES;

```

```
<pre>TABLESPACE_NAME      STATUS      CONTENTS
-----
SYSTEM                  ONLINE     PERMANENT
UNDO                    ONLINE     UNDO
SYSAUX                  ONLINE     PERMANENT
TEMP                    ONLINE     TEMPORARY
USERS                   ONLINE     PERMANENT</pre>
```

[Read More Answers.](#)

Question # 235

What Are the Predefined Tablespaces in a Database?

Answer:-

When you create a new database, Oracle server will create 4 required tablespaces for the new database:

- * SYSTEM Tablespace - Every Oracle database contains a tablespace named SYSTEM, which Oracle creates automatically when the database is created. The SYSTEM tablespace is always online when the database is open.
- * SYSAUX Tablespace - The SYSAUX tablespace was installed as an auxiliary tablespace to the SYSTEM tablespace when you created your database. Some database components that formerly created and used separate tablespaces now occupy the SYSAUX tablespace.
- * UNDO Tablespace - UNDO tablespaces are special tablespaces used solely for storing undo information. You cannot create any other segment types (for example, tables or indexes) in undo tablespaces. Each database contains zero or more undo tablespaces. In automatic undo management mode, each Oracle instance is assigned one (and only one) undo tablespace. Undo data is managed within an undo tablespace using undo segments that are automatically created and maintained by Oracle.
- * TEMP Tablespace - When the SYSTEM tablespace is locally managed, you must define at least one default temporary tablespace when creating a database. A locally managed SYSTEM tablespace cannot be used for default temporary storage.

[Read More Answers.](#)

Question # 236

How To View the Data Files in the Current Database?

Answer:-

If you want to get a list of all tablespaces used in the current database instance, you can use the DBA_TABLESPACES view as shown in the following SQL script example:

```
SQL> connect SYSTEM/globalguideline
Connected.
SQL> col tablespace_name format a16;
SQL> col file_name format a36;
SQL> SELECT TABLESPACE_NAME, FILE_NAME, BYTES
       2 FROM DBA_DATA_FILES;


TABLESPACE_NAME	FILE_NAME	BYTES
USERS	ORACLEXEORADATAXEUSERS.DBF	104857600
SYSAUX	ORACLEXEORADATAXESYSAUX.DBF	461373440
UNDO	ORACLEXEORADATAXEUNDO.DBF	94371840
SYSTEM	ORACLEXEORADATAXESYSTEM.DBF	356515840


```

[Read More Answers.](#)

Question # 237

How To Create a new Oracle Data File?

Answer:-

There is no dedicated statement to create a data file. Data files are created as part of statements that manages other data structures, like tablespace and database.

[Read More Answers.](#)

Question # 238

How To Create a New Tablespace in Oracle?

Answer:-

If you want a new dataspaces, you can use the CREATE TABLESPACE ... DATAFILE statement as shown in the following script:

```
SQL> CREATE TABLESPACE my_space
       2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> SELECT TABLESPACE_NAME, STATUS, CONTENTS
       2 FROM USER_TABLESPACES;


TABLESPACE_NAME	STATUS	CONTENTS
SYSTEM	ONLINE	PERMANENT
UNDO	ONLINE	UNDO
SYSAUX	ONLINE	PERMANENT
TEMP	ONLINE	TEMPORARY
USERS	ONLINE	PERMANENT
MY_SPACE	ONLINE	PERMANENT



```
SQL> SELECT TABLESPACE_NAME, FILE_NAME, BYTES
 2 FROM DBA_DATA_FILES;

| TABLESPACE_NAME | FILE_NAME | BYTES |
|-----------------|-----------------------------|-----------|
| USERS | ORACLEXEORADATAXEUSERS.DBF | 104857600 |
| SYSAUX | ORACLEXEORADATAXESYSAUX.DBF | 461373440 |
| UNDO | ORACLEXEORADATAXEUNDO.DBF | 94371840 |
| SYSTEM | ORACLEXEORADATAXESYSTEM.DBF | 356515840 |
| MY_SPACE | /temp/my_space.dbf | 10485760 |


```


```



```
USERS      ORACLEXEORADATAXEUSERS.DBF 104857600
SYSAUX    ORACLEXEORADATAXESYSAUX.DBF 461373440
UNDO      ORACLEXEORADATAXEUNDO.DBF 94371840
SYSTEM    ORACLEXEORADATAXESYSTEM.DBF 356515840
MY_SPACE  TEMPMY_SPACE.DBF          10485760</pre>
```

So one statement created two structures: a tablespace and a data file. If you check your file system with Windows file explorer, you will see the data file is located

[Read More Answers.](#)

Question # 239

How To Rename a Tablespace in Oracle?

Answer:-

You can easily rename a tablespace by using the ALTER TABLESPACE ... RENAME TO statement as shown in the example below:

```
SQL> CREATE TABLESPACE my_space
  2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> ALTER TABLESPACE my_space RENAME TO your_space;
Tablespace created.
SQL> SELECT TABLESPACE_NAME, STATUS, CONTENTS
  2 FROM USER_TABLESPACES;
<pre>NAME STATUS CONTENTS
-----
SYSTEM ONLINE PERMANENT
UNDO ONLINE UNDO
SYSAUX ONLINE PERMANENT
TEMP ONLINE TEMPORARY
USERS ONLINE PERMANENT
YOUR_SPACE ONLINE PERMANENT</pre>
```

[Read More Answers.](#)

Question # 240

How To Drop a Tablespace?

Answer:-

If you have an existing tablespace and you don't want it anymore. You can delete a tablespace by using the DROP TABLESPACE statement as shown in the example below:

```
SQL> CREATE TABLESPACE my_space
  2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> DROP TABLESPACE my_space;
Tablespace dropped.
```

[Read More Answers.](#)

Question # 241

What Happens to the Data Files If a Tablespace Is Dropped?

Answer:-

If a tablespace is dropped, what happens to its data files? By default, data files will remain in OS file system, even if the tablespace they are mapped is dropped. Of course, you delete the data files using OS commands, if they are no longer needed.

Another way of deleting data files is to use the INCLUDING clause in the DROP TABLESPACE statement. Here is a SQL sample script:

```
SQL> CREATE TABLESPACE my_space
  2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> DROP TABLESPACE my_space INCLUDING CONTENTS
  2 AND DATAFILES;
Tablespace dropped.
```

With the INCLUDING CONTENTS AND DATAFILES clause, all contents and mapped data files are also deleted.

[Read More Answers.](#)

Question # 242

How To Create a Table in a Specific Tablespace?

Answer:-

After you have created a new tablespace, you can give it to your users for them to create tables in the new tablespace. To create a table in a specific tablespace, you need to use the TABLESPACE clause in the CREATE TABLE statement. Here is a sample script:

```
SQL> connect SYSTEM/globalguideline
Connected.
SQL> CREATE TABLESPACE my_space
  2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLE my_team TABLESPACE my_space
  2 AS SELECT * FROM employees;
Table created.
SQL> SELECT table_name, tablespace_name, num_rows
  2 FROM USER_TABLES
```




```
3 WHERE tablespace_name in ('USERS', 'MY_SPACE');
<pre>TABLE_NAME          TABLESPACE_NAME  NUM_ROWS
-----
MY_TEAM              MY_SPACE          -
EMPLOYEES            USERS              107
...</pre>
```

[Read More Answers.](#)

Question # 243

How To See Free Space of Each Tablespace?

Answer:-

One of the important DBA tasks is to watch the storage usage of all the tablespaces to make sure there are enough free space in each tablespace for database applications to function properly. Free space information can be monitored through the USER_FREE_SPACE view. Each record in USER_FREE_SPACE represents an extent, a contiguous area of space, of free space in a data file of a tablespace.

Here is SQL script example on how to see free space of a tablespace:

```
SQL> connect HR/globalguideline
Connected.
SQL> SELECT TABLESPACE_NAME, FILE_ID, BYTES
2 FROM USER_FREE_SPACE
3 WHERE TABLESPACE_NAME IN ('USERS', 'MY_SPACE');
```

[Read More Answers.](#)

Question # 244

How To Bring a Tablespace Offline?

Answer:-

If you want to stop users using a tablespace, you can bring it offline using the ALTER TABLESPACE ... OFFLINE statement as shown in the following script:

```
SQL> connect HR/globalguideline
Connected.
SQL> CREATE TABLESPACE my_space
2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> ALTER TABLESPACE my_space OFFLINE NORMAL;
Tablespace altered.
After bringing a tablespace offline, you can backup or rename the data file safely.
```

[Read More Answers.](#)

Question # 245

How To Bring a Tablespace Online?

Answer:-

If you have brought a tablespace offline, now you want to make it available to users again, you can use the ALTER TABLESPACE ... ONLINE statement as shown in the following script:

```
SQL> connect HR/globalguideline
SQL> CREATE TABLESPACE my_space
2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> ALTER TABLESPACE my_space OFFLINE NORMAL;
Tablespace altered.
...
SQL> ALTER TABLESPACE my_space ONLINE;
Tablespace altered.
```

[Read More Answers.](#)

Question # 246

How To Add Another Datafile to a Tablespace?

Answer:-

If you created a tablespace with a data file a month ago, now 80% of the data file is used, you should add another data file to the tablespace. This can be done by using the ALTER TABLESPACE ... ADD DATAFILE statement. See the following sample script:

```
SQL> connect HR/globalguideline
SQL> CREATE TABLESPACE my_space
2 DATAFILE '/temp/my_space.dbf' SIZE 10M;
Tablespace created.
SQL> ALTER TABLESPACE my_space
2 DATAFILE '/temp/my_space_2.dbf' SIZE 5M;
Tablespace altered.
SQL> SELECT TABLESPACE_NAME, FILE_NAME, BYTES
2 FROM DBA_DATA_FILES;
<pre>TABLESPACE_NAME FILE_NAME          BYTES
-----
USERS          C:ORACLEXEORADATAXEUSERS.DBF 104857600
SYSAUX        C:ORACLEXEORADATAXESYSAUX.DBF 461373440
UNDO          C:ORACLEXEORADATAXEUNDO.DBF 94371840
SYSTEM        C:ORACLEXEORADATAXESYSTEM.DBF 356515840
MY_SPACE      C:TEMPMY_SPACE.DBF          10485760
```



```
MY_SPACE C:TEMPMY_SPACE_2.DBF 5242880</pre>
```

[Read More Answers.](#)

Question # 247

What Happens If You Lost a Data File?

Answer:-

After you shutting down an Oracle database, you accidentally deleted a data file from the operating system. If you try to start the database again you will get error when Oracle tries to open the database after mounting the database. The following tutorial examples shows you what will happen if the data file c: emppy_space.dbf is deleted. Oracle can still start the database instance and mount the database. But it will fail on opening the database as shown below in a SQL*Plus session:

```
>sqlplus /nolog
SQL> connect SYSTEM/globalguideline AS SYSDBA
SQL> STARTUP
ORACLE instance started.
<pre>Total System Global Area 100663296 bytes
Fixed Size 1285956 bytes
Variable Size 58720444 bytes
Database Buffers 37748736 bytes
Redo Buffers 2908160 bytes</pre>
Database mounted.
ORA-01157: cannot identify/lock data file 5 - see DBWR
trace file
ORA-01110: data file 5: 'C:TEMPMY_SPACE.DBF'
SQL> SHUTDOWN;
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
```

[Read More Answers.](#)

Question # 248

How Remove Data Files before opening a Database?

Answer:-

Let's say you have a corrupted data file or lost a data file. Oracle can mount the database. But it will not open the database. What you can do is to set the bad data file as offline before opening the database. The tutorial exercise shows you how to set two data files offline and open the database without them:

```
>sqlplus /nolog
SQL> connect SYSTEM/globalguideline AS SYSDBA
SQL> STARTUP MOUNT;
ORACLE instance started.
<pre>Total System Global Area 100663296 bytes
Fixed Size 1285956 bytes
Variable Size 58720444 bytes
Database Buffers 37748736 bytes
Redo Buffers 2908160 bytes
Database mounted.</pre>
SQL> ALTER DATABASE DATAFILE ' emppy_space.dbf'
2 OFFLINE DROP;
Database altered.
SQL> ALTER DATABASE DATAFILE ' emppy_space_2.dbf'
2 OFFLINE DROP;
Database altered.
```

[Read More Answers.](#)

Question # 249

How To Create an Oracle Database?

Answer:-

There are two ways to create a new database:

- * Use the Database Configuration Assistant (DBCA) to create a database interactively.
- * Use the CREATE DATABASE statement to create a database manually.

[Read More Answers.](#)

Question # 250

How To Create an Oracle Database Manually?

Answer:-

Based on Oracle's Administrator Guide, there are 11 steps to create a database with the CREATE DATABASE statement:

- * Step 1: Decide on Your Instance Identifier (SID)
- * Step 2: Establish the Database Administrator Authentication Method
- * Step 3: Create the Initialization Parameter File
- * Step 4: Connect to the Instance
- * Step 5: Create a Server Parameter File (Recommended)
- * Step 6: Start the Instance
- * Step 7: Issue the CREATE DATABASE Statement
- * Step 8: Create Additional Tablespaces
- * Step 9: Run Scripts to Build Data Dictionary Views
- * Step 10: Run Scripts to Install Additional Options (Optional)



* Step 11: Back Up the Database.

Other items in this FAQ collection will follow those steps to help you creating a new database manually from beginning to end.

[Read More Answers.](#)

Question # 251

How To Select an Oracle System ID (SID)?

Answer:-

This is Step 1. If you are planning to create a new database, you need to select an Oracle System ID (SID). This ID will be used to identify the new Oracle database and its Oracle instance. SID must be unique if you want to run multiple databases on a single server.

Let's set SID for the new database to be: ggl.

[Read More Answers.](#)

Question # 252

How To Establish Administrator Authentication to the Server?

Answer:-

This is Step 2. There are two ways to establish administrator authentication to a new database.

* Use a password file.

* Use operating system (OS) authentication.

Using OS authentication is easier on Windows system. If you used your own Windows user account to install Oracle server, it will put your Windows user account into a special Window's user group called SYSDBA. This Window's user group will be fully trusted by Oracle server with SYSDBA privilege.

To continue with other steps, make sure you logged into the Windows system with a user account in the SYSDBA group.

[Read More Answers.](#)

Question # 253

How To Create an Initialization Parameter File?

Answer:-

This is Step 3. To run an Oracle database as an Oracle instance, you need to create an initialization parameter file, which contains a set of initialization parameters. The easiest way to create an initialization parameter file to copy from the sample file provided by Oracle. You can do this in a command window as shown below:

```
>cd $ORACLE_HOME
>copy .configscripts\init.ora .databaseinitggl_ini.ora
>edit .databaseinitggl_ini.ora
(replace XE by ggl)
```

In this example, only the SID is changed from XE to ggl. All other parameters are maintained as is.

[Read More Answers.](#)

Question # 254

How To Connect the Oracle Server as SYSDBA?

Answer:-

This is Step 4. The best way to connect to the Oracle server as an administrator is to use SQL*Plus. You need to run SQL*Plus with /nolog option and use the CONNECT with blank user name, blank password and AS SYSDBA option. Here is a sample session:

```
>cd $ORACLE_HOME
>. insqlplus /nolog
SQL> CONNECT / AS SYSDBA
Connected.
```

[Read More Answers.](#)

Question # 255

How To Create a Server Parameter File?

Answer:-

This is Step 5. The initialization parameter file is good to get an Oracle database instance started. But it is not ideal run an instance as production. You need to convert the initialization parameter file into a Server Parameter File (SPFile) using the CREATE SPFILE statement. The script below shows you how to do this:

```
SQL> CREATE SPFILE=$ORACLE_HOME/dbs/SPFILEggl.ora
  2 FROM PFILE=$ORACLE_HOME/database/initggl_ini.ora;
File created.
```

Note that \$ORACLE_HOME should be replaced by the real path name where your Oracle server is installed.

The SPFile should be located in the expected directory and named as SPFILE(\$SID).ora.

[Read More Answers.](#)

Question # 256

How To Start an Oracle Instance?

Answer:-

This is Step 6. Now you are ready to start the new Oracle Instance without any database. This instance will be used to create a database. Starting an instance without database can be done by using STARTUP NOMOUNT statement as shown below:

```
>. insqlplus /nolog
SQL> CONNECT / AS SYSDBA
Connected.
SQL> SHUTDOWN
ORACLE instance shut down.
SQL> STARTUP NOMOUNT
```



ORA-00821: Specified value of sga_target 16M is too small, needs to be at least 20M
The SHUTDOWN command is need to bring the default instance XE down.
The STARTUP NOMOUNT command failed because it tried to start the default instance XE, and there is a bad parameter in the XE instance SPFile.
See the next FAQ question to find another way to start the new instance ggl.

[Read More Answers.](#)

Question # 257

How To Start a Specific Oracle Instance?

Answer:-

A simple way to start a specific Oracle instance is to start the instance with the PFILE option as shown in the following example:

```
> . insqlplus /nolog
SQL> CONNECT / AS SYSDBA
Connected.
SQL> STARTUP NOMOUNT
  PFILE=$ORACLE_HOME/database/initggl_ini.ora
ORA-02778: Name given for the log directory is invalid
The PFILE option allows you to specify the initialization parameter file of a specific Oracle instance. But the initialization parameter file created in Step 3 has some problem with the log directory.
```

[Read More Answers.](#)

Question # 258

How To Start Instance with a Minimal Initialization Parameter File?

Answer:-

The sample initialization parameter file provided by Oracle seems to be not working. But we can try to start the new instance with a minimal initialization parameter file (PFile). First you can create another PFile, \$ORACLE_HOME/database/initggl_ini_min.ora, as shown below:

```
db_name=ggl
control_files=('oraclexeoradatagglcontrol.dbf')
undo_management=AUTO
Then start the ggl instance again:
SQL> CONNECT / AS SYSDBA
Connected.
SQL> STARTUP NOMOUNT
  PFILE=$ORACLE_HOME/database/initggl_ini_min.ora
ORACLE instance started.


```
<pre>Total System Global Area 113246208 bytes
Fixed Size 1286028 bytes
Variable Size 58720372 bytes
Database Buffers 50331648 bytes
Redo Buffers 2908160 bytes</pre>
```


```

[Read More Answers.](#)

Question # 259

How To Run CREATE DATABASE Statement?

Answer:-

This is Step 7. Oracle Administrator Guide provided a sample CREATE DATABASE statement. But it is a long statement. You can modify and save it in a file, \$ORACLE_HOME/configscripts/create_database_ggl.sql, and run the file within SQL*Plus. Here is a copy of the modified CREATE DATABASE statement:

```
CREATE DATABASE ggl
USER SYS IDENTIFIED BY globalguideline
USER SYSTEM IDENTIFIED BY globalguideline
LOGFILE GROUP 1 ('/oraclexe/oradata/ggl/redo01.log') SIZE 10M,
  GROUP 2 ('/oraclexe/oradata/ggl/redo02.log') SIZE 10M,
  GROUP 3 ('/oraclexe/oradata/ggl/redo03.log') SIZE 10M
MAXLOGFILES 5
MAXLOGMEMBERS 5
MAXLOGHISTORY 1
MAXDATAFILES 100
MAXINSTANCES 1
CHARACTER SET US7ASCII
NATIONAL CHARACTER SET AL16UTF16
DATAFILE '/oraclexe/oradata/ggl/system01.dbf' SIZE 32M REUSE
EXTENT MANAGEMENT LOCAL
SYSAUX DATAFILE '/oraclexe/oradata/ggl/sysaux01.dbf' SIZE 32M REUSE
DEFAULT TABLESPACE tbs_1
DEFAULT TEMPORARY TABLESPACE tempts1
  TEMPFILE '/oraclexe/oradata/ggl/temp01.dbf' SIZE 20M REUSE
UNDO TABLESPACE undotbs
  DATAFILE '/oraclexe/oradata/ggl/undotbs01.dbf'
  SIZE 20M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```

[Read More Answers.](#)

Question # 260

How To Do Clean Up If CREATE DATABASE Failed?

Answer:-



To better organize data files, you should create a dedicated directory for each Oracle database. This can be done by using Windows file explorer to create the oraclexeoradataggl directory. Try the CREATE DATABASE statement again, when you have the directory ready. If your CREATE DATABASE statement failed half way again, you may have to clean up the partial result of the CREATE DATABASE statement. Here is a list of suggestions for you:

- * Run SHUTDOWN command to stop the partial started database instance.
- * Remove all files in ggl directory: oraclexeoradataggl
- * Run STARTUP NOMOUNT PFILE command to start the empty instance again to be ready for CREATE DATABASE statement.

[Read More Answers.](#)

Question # 261

How To Run CREATE DATABASE Statement Again?

Answer:-

After cleaning up the results of a previously failed CREATE DATABASE statement, you can run the CREATE DATABASE statement again as shown below:

```
SQL> @$ORACLE_HOME\config\scripts\create_database_ggl.sql;
CREATE DATABASE ggl
```

*

ERROR at line 1:

ORA-01092: ORACLE instance terminated. Disconnection forced

Something went wrong again. This time it might related the limitation that Oracle 10g XE only supports one database instance. With the default instance called "XE" already defined, creating another database instance might be not allowed.

[Read More Answers.](#)

Question # 262

How To Create Additional Tablespaces for an New Database?

Answer:-

This is Step 8. Creating additional tablespaces can be done by using the CREATE TABLESPACE statement as shown in the following sample script:

```
SQL> CREATE TABLESPACE users
  2 DATAFILE '/oraclexe/oradata/ggl/users01.dbf' SIZE 10M;
SQL> CREATE TABLESPACE indx
  2 DATAFILE '/oraclexe/oradata/ggl/indx01.dbf' SIZE 10M;
```

[Read More Answers.](#)

Question # 263

How To Build Data Dictionary View an New Database?

Answer:-

This is Step 9. The Oracle Administrator Guide suggests to run two SQL scripts provided by Oracle as shown bellow:

```
SQL> @/u01/oracle/rdbms/admin/catalog.sql
SQL> @/u01/oracle/rdbms/admin/catproc.sql
```

[Read More Answers.](#)

Question # 264

What Is PL/SQL?

Answer:-

PL/SQL is a modern, block-structured programming language. It provides several features that make developing powerful database applications very convenient. For example, PL/SQL provides procedural constructs, such as loops and conditional statements, that are not available in standard SQL.

PL/SQL code runs on the server, so using PL/SQL lets you centralize significant parts of your database applications for increased maintainability and security. It also enables you to achieve a significant reduction of network overhead in client/server applications.

[Read More Answers.](#)

Question # 265

What Are the Types PL/SQL Code Blocks?

Answer:-

There are 3 types of PL/SQL code blocks:

- * Anonymous Block - A block of codes with no name. It may contain a declaration part, an execution part, and exception handlers.
- * Stored Program Unit - A block of codes with a name. It is similar to an anonymous block. But it can take parameters and return values.
- * Trigger - A block of code that can be defined to fire based an specific event.

[Read More Answers.](#)

Question # 266

How To Define an Anonymous Block?

Answer:-

An anonymous block must have an execution part, which is a group of other PL/SQL statements enclosed in the BEGIN ... END statement. Here is a script on how to define a simple anonymous block with SQL*Plus:

```
SQL> set serveroutput on;
SQL> begin
  2 dbms_output.put_line('Hello world!');
  3 end;
  4 /
Hello world!
```



PL/SQL procedure successfully completed.
"set serveroutput on;" allows dbms_output.put_line() to work.
"/" runs the anonymous block, which print the "Hello world!" message.

[Read More Answers.](#)

Question # 267

How Many Anonymous Blocks Can Be Defined?

Answer:-

An anonymous block is stored in the user's current session without any name. So you can only define one anonymous block at any time. If you define another anonymous block, the new block will replace the previously defined block, as shown in the following script:

```
SQL> set serveroutput on;
SQL> begin
  2  dbms_output.put_line('Hello world!');
  3  end;
  4  /
Hello world!
PL/SQL procedure successfully completed.
SQL> begin
  2  dbms_output.put_line('This is a PL/SQL FAQ.');
```

```
  3  end;
  4  /
This is a PL/SQL FAQ.
PL/SQL procedure successfully completed.
```

[Read More Answers.](#)

Question # 268

How To Run the Anonymous Block Again?

Answer:-

If you have an anonymous block defined in your session, you can run it any time by using the "/" command as shown in the following script:

```
SQL> set serveroutput on;
SQL> begin
  2  dbms_output.put_line('This is a PL/SQL FAQ.');
```

```
  3  end;
  4  /
This is a PL/SQL FAQ.
PL/SQL procedure successfully completed.
```

```
SQL> /
This is a PL/SQL FAQ.
PL/SQL procedure successfully completed.
```

What Is Stored Program Unit?

A stored program unit is a named block of codes which:

- * Has a name.
- * Can take parameters, and can return values.
- * Is stored in the data dictionary.
- * Can be called by many users.

[Read More Answers.](#)

Question # 269

How To Create a Stored Program Unit?

Answer:-

If you want to create a stored program unit, you can use the CREATE PROCEDURE or FUNTION statement. The example script below creates a stored program unit:

```
SQL> set serveroutput on;
SQL> CREATE PROCEDURE Hello AS
  2  BEGIN
  3  DBMS_OUTPUT.PUT_LINE('Hello world!');
```

```
  4  END;
  5  /
Procedure created.
```

[Read More Answers.](#)

Question # 270

How To Execute a Stored Program Unit?

Answer:-

If you want to execute a stored program unit, you can use the EXECUTE statement. The example script below shows how to executes a stored program unit:

```
SQL> set serveroutput on;
SQL> CREATE PROCEDURE Hello AS
  2  BEGIN
  3  DBMS_OUTPUT.PUT_LINE('Hello world!');
```

```
  4  END;
  5  /
Procedure created.
```

```
SQL> EXECUTE Hello;
```



Hello world!

[Read More Answers.](#)

Question # 271

How Many Data Types Are Supported?

Answer:-

PL/SQL supports two groups of data types:

- * SQL Data Types - All data types used for table columns.
- * PL/SQL Special Data Types - Like BOOLEAN or PLS_INTEGER.

The script below shows some data type examples:

```
SQL> set serveroutput on;
SQL> DECLARE
2  title VARCHAR(8);
3  salary NUMBER;
4  seeking_job BOOLEAN;
5  BEGIN
6  title := 'DBA';
7  salary := 50000;
8  seeking_job := TRUE;
9  DBMS_OUTPUT.PUT_LINE('Job Title: ' || title);
10 DBMS_OUTPUT.PUT_LINE('Expected salary: '
11 || TO_CHAR(salary));
12 END;
13 /
```

Job Title: DBA

Expected salary: 50000

[Read More Answers.](#)

Question # 272

What Are the Execution Control Statements?

Answer:-

PL/SQL supports three groups of execution control statements:

- * IF Statements - Conditionally executes a block of statements.
- * CASE Statements - Selectively executes a block of statements.
- * LOOP Statements - Repeatedly executes a block of statements.
- * GOTO Statements - Unconditional changes the execution flow to a specified statement.

The script below shows execution control statements:

```
SQL> set serveroutput on;
SQL> DECLARE
2  total NUMBER;
3  BEGIN
4  total := 0;
5  LOOP
6  total := total+1;
7  IF total >= 10 THEN
8  GOTO print;
9  END IF;
10 END LOOP;
11 <
12 DBMS_OUTPUT.PUT_LINE('Total counts: '
13 || TO_CHAR(total));
14 END;
15 /
```

Total counts: 10

This sample script sh

[Read More Answers.](#)

Question # 273

How To Use SQL Statements in PL/SQL?

Answer:-

SQL DML (Data Manipulation Language) statements can be included in PL/SQL code blocks directly without any changes. See the script below for examples:

```
SQL> CREATE TABLE tip (id NUMBER(5) PRIMARY KEY,
2  subject VARCHAR(80) NOT NULL,
3  description VARCHAR(256) NOT NULL);
```

Table created.

```
SQL> BEGIN
2  INSERT INTO tip VALUES(1, 'PL/SQL',
3  'Good for beginners. ');
4  UPDATE tip SET description = 'Good for beginners.';
5  END;
6 /
```

PL/SQL procedure successfully completed.

```
SQL> COL subject FORMAT A12;
SQL> COL description FORMAT A24;
SQL> SELECT * FROM tip;
<pre> ID SUBJECT DESCRIPTION
```



1 PL/SQL Good for beginners.</pre>

SQL> DROP TABLE tip;

Table dropped.

This script example actually has 3 parts:

* Part 1 - A single DDL (Data Definition Language) statement, CREATE TABLE, outside the PL/SQL code block.

* Part 2 - A PL/SQL code block defined and executed to insert and updated a record.

* Part 3 - A couple of SQL statements outside the PL/SQL code block to view the record and drop the table.

[Read More Answers.](#)

Question # 274

How To Process Query Result in PL/SQL?

Answer:-

You can run queries (SELECT statements) in a PL/SQL code blocks, and process the results a loop as shown in the following script example:

SQL> set serveroutput on;

SQL> BEGIN

2 FOR row IN

3 (SELECT * FROM employees WHERE manager_id = 101)

4 LOOP

5 DBMS_OUTPUT.PUT_LINE('Name = ' || row.last_name);

6 END LOOP;

7 END;

8 /

Name = Greenberg

Name = Whalen

Name = Mavris

Name = Baer

Name = Higgins

The loop statement in this script automatically sets a temporary variable "row" each row in the result set returned from the SELECT statement.

[Read More Answers.](#)

Question # 275

How To Create an Array in PL/SQL?

Answer:-

If you want create an array data structure, you can use the collection type VARRAY. VARRAY stands for Variable Array. Here is a sample script on how to use VARRAY:

SQL> set serveroutput on;

SQL> DECLARE

2 TYPE list IS VARRAY(3) OF VARCHAR(16);

3 domain_list list;

4 BEGIN

5 domain_list := list('google','globalguideline','myspace');

6 DBMS_OUTPUT.PUT_LINE('Second domain: '

7 || domain_list(2));

8 END;

9 /

Second domain: globalguideline

As you can see, to define an array variable, you need to create your own array type, and use that type to define array variables.

[Read More Answers.](#)

Question # 276

How To Manage Transaction Isolation Level?

Answer:-

Transaction isolation level can be managed in a procedure by using the SET TRANSACTION and COMMIT statements. Here is a sample script on how to manage transaction isolation level:

<pre>SQL> CREATE OR REPLACE PROCEDURE HR.TOTAL_SALARY AS

2 total_salary NUMBER(12,2);

3 BEGIN

4 SET TRANSACTION READ ONLY;

5 SELECT SUM (salary) INTO total_salary FROM employees;

6 DBMS_OUTPUT.PUT_LINE('Total salary 1: '

7 || total_salary);

8 -- Other sessions may change salaries of some records

9 SELECT SUM (salary) INTO total_salary FROM employees;

10 DBMS_OUTPUT.PUT_LINE('Total salary 2: '

11 || total_salary);

12 COMMIT;

13 END;

14 /</pre>

SQL> EXECUTE TOTAL_SALARY;

Total salary 1: 691400

Total salary 2: 691400

"READ ONLY" transaction level takes a read only snapshot of the database. This allows other sessions to update the database without any locks. All queries in the session will produces identical results. So both SELECT statements in this script will return the same value guaranteed.

[Read More Answers.](#)



Question # 277

How To Pass Parameters to Procedures?

Answer:-

Store procedures or functions can take parameters. You need to define parameters while defining the procedure, and providing values to parameters while calling the procedure. The script below shows you how to do this:

```
SQL> CREATE OR REPLACE PROCEDURE DBA_TASK (day VARCHAR2) AS
2 BEGIN
3 IF day = 'MONDAY' THEN
4   DBMS_OUTPUT.PUT_LINE('Checking log files.');
```

```
5 ELSIF day = 'FRIDAY' THEN
6   DBMS_OUTPUT.PUT_LINE('Rebuild indexes.');
```

```
7 ELSE
8   DBMS_OUTPUT.PUT_LINE('Reading some papers.');
```

```
9 END IF;
10 END;
11 /
SQL> EXECUTE DBA_TASK('MONDAY');
```

Checking log files.

```
SQL> EXECUTE DBA_TASK('SUNDAY');
```

Reading some papers.

As you can see, procedures with parameters can make procedures more flexible.

[Read More Answers.](#)

Question # 278

How To Define a Procedure inside Another Procedure?

Answer:-

Define a procedure inside another procedure is supported by PL/SQL. The following tutorial script shows you an example:

```
SQL> CREATE OR REPLACE PROCEDURE HR.DBA_WEEK AS
2 PROCEDURE DBA_TASK (day VARCHAR2) AS
3 BEGIN
4 IF day = 'MONDAY' THEN
5   DBMS_OUTPUT.PUT_LINE('Checking log files.');
```

```
6 ELSIF day = 'FRIDAY' THEN
7   DBMS_OUTPUT.PUT_LINE('Rebuild indexes.');
```

```
8 ELSE
9   DBMS_OUTPUT.PUT_LINE('Reading some papers.');
```

```
10 END IF;
11 END;
12 BEGIN
```

```
13 DBA_TASK('MONDAY');
```

```
14 DBA_TASK('TUESDAY');
```

```
15 END;
16 /
```

```
SQL> EXECUTE DBA_WEEK;
```

Checking log files.

Reading some papers.

Remember that procedures used inside a procedure must be defined in the declaration block.

[Read More Answers.](#)

Question # 279

What Do You Think about PL/SQL?

Answer:-

After following through the tutorials in the FAQ collection, you probably agree that PL/SQL is indeed a general purpose database programming language. PL/SQL is a natural extension of SQL. It is very useful for DBA to automate specific administration tasks or for developers to develop simple business backend applications.

[Read More Answers.](#)

Question # 280

What Is Oracle SQL Developer?

Answer:-

Oracle SQL Developer is a new, free graphical tool that enhances productivity and simplifies database development tasks. With SQL Developer, you can browse database objects, run SQL statements and SQL scripts, and edit and debug PL/SQL statements. You can also run any number of provided reports, as well as create and save your own.

Oracle SQL Developer is formerly called Project Raptor.

[Read More Answers.](#)

Question # 281

What Operating Systems Are Supported by Oracle SQL Developer?

Answer:-

Oracle SQL Developer is available for three types of operating Systems:

* Windows

* Linux



* Mac OSX

[Read More Answers.](#)

Question # 282

How To Download Oracle SQL Developer?

Answer:-

If you want to download a copy of Oracle SQL Developer, visit <http://www.oracle.com/technology/software/products/sql/>.

If you are using Windows systems, click the "Oracle SQL Developer for Windows" link. This allows you to download the Windows version in ZIP format with file name of sqldeveloper-v1.0.0.zip in 55,295,721 bytes.

[Read More Answers.](#)

Question # 283

How To Install Oracle SQL Developer?

Answer:-

Once you have sqldeveloper-v1.0.0.zip downloaded, extract the zip file into the root directory: .

When the extraction is done, the installation is completed. Your copy of Oracle SQL Developer is installed in directory: sqldeveloper, and ready to run.

[Read More Answers.](#)

Question # 284

How To Start Oracle SQL Developer?

Answer:-

To start Oracle SQL Developer, go to sqldeveloper and click sqldeveloper.exe. The Oracle SQL Developer window shows up. Your copy of Oracle SQL Developer is running now. There will be nothing under the Connections icon, because you haven't defined any connections yet.

You should visit different menu items to explore the user interface. One nice place to visit is the About item under the Help menu. It provides some very useful information about your copy of Oracle SQL Developer.

[Read More Answers.](#)

Question # 285

Is Oracle SQL Developer written in Java?

Answer:-

Oracle SQL Developer is written in Java. It requires JDK 1.5, which is already included in your download file.

[Read More Answers.](#)

Question # 286

How To Connect to a Local Oracle 10g XE Server?

Answer:-

If you have your Oracle 10g XE server running on your local machine, you can connect your Oracle SQL Developer to the server with the following steps:

- * Start Oracle SQL Developer
- * Right-click on Connections
- * Select New Database Connection
- * Enter Connection Name as: Local_XE
- * Enter Username as: HR
- * Enter Password as: globalguideline
- * Enter Hostname as: 127.0.0.1
- * Enter SID as: XE
- * Click Test
- * Click Connect, if you see Status changed to Success

You will see the SQL Statement area opened ready to take any SQL statements.

[Read More Answers.](#)

Question # 287

How To Connect to a Remote Server?

Answer:-

If you have an Oracle server running remotely on a network, and you know all the access information needed, you can follow the following steps to connect your Oracle SQL Developer:

- * Start Oracle SQL Developer
- * Right-click on Connections
- * Select New Database Connection
- * Enter Connection Name as: Remote_Server
- * Enter Username as: _username_
- * Enter Password as: _password_
- * Enter Hostname as: _hostname_
- * Enter SID as: _sidname_
- * Click Test
- * Click Connect, if you see Status changed to Success

You will see the SQL Statement area opened ready to take any SQL statements.

[Read More Answers.](#)



Question # 288

How To Run SQL Statements with Oracle SQL Developer?

Answer:-

Once a connection is established with an Oracle server, you can enter any SQL statements in the SQL Statement area. Try yourself with the following steps:

- * Go to the SQL Statement area
 - * Enter `SELECT username, default_tablespace FROM USER_USERS;`
 - * Press F9 or click the Execute Statement icon
- You will see the statement results in the Results area.

[Read More Answers.](#)

Question # 289

How To Export Your Connection Information to a File?

Answer:-

SQL Developer allows you to export your connection information into an XML file. Here is how to do this:

- * Right-click on Connections
- * Select Export Connection...
- * Enter File Name as: `empconnections.xml`
- * Click OK

Open `empconnections.xml`, you should see an XML message like this:

```
<pre><?xml version = '1.0'?>
<!DOCTYPE connections>
<connections>
  <connection>
    <JDBC_PORT>1521</JDBC_PORT>
    <HOSTNAME>127.0.0.1</HOSTNAME>
    <ConnectionType>JDBC</ConnectionType>
    <DeployPassword>>false</DeployPassword>
    <user>HR</user>
    <ConnectionName>Local_XE</ConnectionName>
    <SID>XE</SID>
    <JdbcDriver>oracle.jdbc.driver.OracleDriver
    </JdbcDriver>
    <PWD>
      <![CDATA[0519D91AA309BF15EA9E54BF55F863710496...]]>
    </PWD>
    <ORACLE_JDBC_TYPE>thin</ORACLE_JDBC_TYPE>
  </connection>
</connections></pre>
```

[Read More Answers.](#)

Question # 290

How To Run SQL*Plus Commands in SQL Developer?

Answer:-

Most of the time, you only run SQL statements in SQL Worksheet, the SQL statement area. But you can also run some SQL*Plus commands in SQL Worksheet. The example below shows you how to run the DESCRIBE command in SQL Developer:

- * Go to SQL Worksheet - SQL statement area
- * Enter `DESCRIBE USER_USERS`
- * Press F9 to run the command

You will see the following output:

```
<pre>Name                Null  Type
-----
USERNAME              NOT NULL VARCHAR2(30)
USER_ID                NOT NULL NUMBER
ACCOUNT_STATUS        NOT NULL VARCHAR2(32)
LOCK_DATE              DATE
EXPIRY_DATE            DATE
DEFAULT_TABLESPACE    NOT NULL VARCHAR2(30)
TEMPORARY_TABLESPACE  NOT NULL VARCHAR2(30)
CREATED                NOT NULL
DATE
INITIAL_RSRC_CONSUMER_GROUP  VARCHAR2(30)
EXTERNAL_NAME          VARCHAR2(4000)</pre>
```

The DESCRIBE command returns you the detailed information about the specified table.

[Read More Answers.](#)

Question # 291

How To Work with Data Objects Interactively?

Answer:-

You can work with data objects through SQL statements in statement area. If you want to work with data objects interactively, you should use the object browser. The following tutorial steps help you to browse data objects:

- * Click the Connections tab.
- * Open the Connections icon.
- * Open a connection, like Local_XE. You see a list of data object types.
- * Open the Connections icon.



- * Open Tables. You see list of existing tables.
 - * Open a table, like EMPLOYEES. You see a list of columns.
 - * Double-click a table, like EMPLOYEES. You see a list of tabs opened for this table.
- Once a table opened, SQL Developer offers you many views on this table, like Columns, Data, Indexes, Constraints, Statistics, Triggers, etc.

[Read More Answers.](#)

Question # 292

How To Get a CREATE Statement for an Existing Table?

Answer:-

You have an existing table and want to get a CREATE statement for that table, you can use the SQL tab in the object view. The following tutorial steps show you how to use SQL Developer to generate a CREATE statement on an existing table:

- * Double-click the table name JOBS in the object tree area.
- * Click the SQL tab in the object view area.

In a moment, you will get the following CREATE statements:

```
REM HR JOBS
CREATE TABLE "HR"."JOBS"
 ("ID" VARCHAR2(10 BYTE),
 "TITLE" VARCHAR2(35 BYTE) CONSTRAINT "TITLE_NN"
 NOT NULL ENABLE,
 "MIN_SALARY" NUMBER(6,0),
 "MAX_SALARY" NUMBER(6,0),
 CONSTRAINT "ID_PK" PRIMARY KEY ("ID") ENABLE
 );
```

```
REM HR ID_PK
CREATE UNIQUE INDEX "HR"."ID_PK" ON "HR"."JOBS" ("ID")
;
```

[Read More Answers.](#)

Question # 293

How To Create a Table Interactively?

Answer:-

If you don't want to use SQL statements to create a new table, you use SQL Developer to create one interactively. Follow the steps below to create a new table called: TIP

- * Right-click on the Tables in the object tree area.
- * Select Create TABLE. Create Table window shows up.
- * Enter Name as: TIP
- * Enter Column Name as: ID, for column 1.
- * Select Type as: INTEGER, for column 1.
- * Click Primary as: Checked, for column 1.
- * Click Add Column to add column 2.
- * Enter Column Name as: SUBJECT, for column 2.
- * Select Type as: VARCHAR2, for column 2.
- * Select Size as: 80, for column 2.
- * Click OK.

[Read More Answers.](#)

Question # 294

How To Enter a New Row into a Table Interactively?

Answer:-

If you don't like to use the INSERT statement to enter a new row into a table, you can use the object view to enter it interactively. Follow the steps below to enter new row into table TIP:

- * Double-click on the table name TIP.
- * Click the Data tab in the object view.
- * Click the Insert Row icon, the + sign.
- * Enter ID as: 101.
- * Enter SUBJECT as: Backup #1.
- * Click the Commit Changes icon.

You know the new row is inserted, because the log area shows you:
INSERT INTO "HR"."TIPS" (ID, SUBJECT) VALUES ('101', 'Backup #1')
Commit Successful

[Read More Answers.](#)

Question # 295

What Is the Reports View in Oracle SQL Developer?

Answer:-

The Reports view lets you browse database information that organized by the server as special views. Information can be browsed include:

- * Database parameters
- * Storage information
- * Session information
- * Cursors
- * Data objects
- * User accounts



* Security information

[Read More Answers.](#)

Question # 296

How To Get a List of All Tables in the Database?

Answer:-

If you don't like to use a SELECT statement to get a list of all tables in the current database, you can use the Reports view to do this as shown in the following tutorial example:

- * Click menu View.
 - * Selects Reports from the menu.
 - * Open Reports.
 - * Open Data Dictionary Reports.
 - * Open Table.
 - * Double-click User Tables. You see the Enter Bind Values window.
 - * Click Apply.
- You should get a list all current tables in the database.

[Read More Answers.](#)

Question # 297

How To Get a List of All User Accounts in the Database?

Answer:-

If you don't like to use a SELECT statement to get a list of all user accounts in the current database, you can use the Reports view to do this as shown in the following tutorial example. You need to connect to the server as SYSTEM to do this:

- * Click menu View.
- * Selects Reports from the menu.
- * Open Reports.
- * Open Data Dictionary Reports.
- * Open Database Administration.
- * Open Users.
- * Double-click All Users.

You should get a list all current user accounts in the database.

[Read More Answers.](#)

Question # 298

How To Get a List of All Background Sessions in the Database?

Answer:-

If you don't like to use a SELECT statement to get a list of all background sessions in the current database, you can use the Reports view to do this as shown in the following tutorial example. You need to connect to the server as SYSTEM to do this:

- * Click menu View.
- * Selects Reports from the menu.
- * Open Reports.
- * Open Data Dictionary Reports.
- * Open Database Administration.
- * Open Sessions.
- * Double-click Background Sessions.

You should get a list all current background sessions in the database.

[Read More Answers.](#)

Question # 299

How To Create Your Own Reports in SQL Developer?

Answer:-

Oracle SQL Developer also lets you create your own reports. See the following steps on how to do this:

- * Click menu View.
- * Selects Reports from the menu.
- * Open Reports.
- * Right-click on User Defined Reports.
- * Select Add Report.
- * Enter Name as: My Team.
- * Enter Description as: List of employees in my team.
- * Enter SQL as: SELECT * FROM HR.EMPLOYEES WHERE MANAGER_ID=100
- * Click Apply.

Your report "My Team" is ready.

[Read More Answers.](#)

Question # 300

How Many File Formats Are Supported to Export Data?

Answer:-

Oracle SQL Developer can allow to export table data into files in the following formats:

- * TXT - Tab delimited fields file format.
- * CSV - Comma Separated Values (CSV) file format.
- * LOADER - File format used by SQL*Loader.



- * XML - XML file format.
- * INSERT - INSERT statements format.

[Read More Answers.](#)

Question # 301

How To Export Data to a CSV File?

Answer:-

If you want to export data from a table to a file in CSV format, you can use the following steps:

- * Right-click the table name, EMPLOYEES, in the object tree view.
- * Select Export.
- * Select CSV. The Export Data window shows up.
- * Click Format tab.
- * Select Format as: CSV.
- * Enter File as: empMyTeam.csv.
- * Click Columns tab.
- * Check columns: FIRST_NAME, LAST_NAME, MANAGER_ID
- * Click Where tab.
- * Enter Where clause as: MANAGER_ID=100.
- * Click Apply.

Your CSV file, empMyTeam.csv, is ready. Open it, you will see:

```
"FIRST_NAME","LAST_NAME","MANAGER_ID"  
"Gerald","Cambrault","100"  
"Lex","De Haan","100"  
"Alberto","Errazuriz","100"  
"Adam","Fripp","100"  
"Michael","Hartstein","100"  
"Payam","Kaufling","100"  
"Neena","Kochhar","100"
```

...

CSV files can be open by Microsoft Excel.

[Read More Answers.](#)

Question # 302

What Is Oracle SQL Developer?

Answer:-

Oracle SQL Developer is a new, free graphical tool that enhances productivity and simplifies database development tasks. With SQL Developer, you can browse database objects, run SQL statements and SQL scripts, and edit and debug PL/SQL statements. You can also run any number of provided reports, as well as create and save your own.

Oracle SQL Developer is formerly called Project Raptor.

[Read More Answers.](#)

Question # 303

What Operating Systems Are Supported by Oracle SQL Developer?

Answer:-

Oracle SQL Developer is available for three types of operating Systems:

- * Windows
- * Linux
- * Mac OSX

[Read More Answers.](#)

Question # 304

How To Download Oracle SQL Developer?

Answer:-

If you want to download a copy of Oracle SQL Developer, visit <http://www.oracle.com/technology/software/products/sql/>.

If you are using Windows systems, click the "Oracle SQL Developer for Windows" link. This allows you to download the Windows version in ZIP format with file name of sqldeveloper-v1.0.0.zip in 55,295,721 bytes.

[Read More Answers.](#)

Question # 305

How To Install Oracle SQL Developer?

Answer:-

Once you have sqldeveloper-v1.0.0.zip downloaded, extract the zip file into the root directory: .

When the extraction is done, the installation is completed. Your copy of Oracle SQL Developer is installed in directory: sqldeveloper, and ready to run.

[Read More Answers.](#)

Question # 306

What Is PL/SQL Language Case Sensitive?

Answer:-

PL/SQL language is not case sensitive:



- * Reserved words are not case sensitive. For example: CASE and Case are identical.
- * Variable names and other names are not case sensitive. For example: TOTAL_SALARY and total_salary are identical. But values in string literals are case sensitive. For example: 'DBA' and 'dba' are different.

[Read More Answers.](#)

Question # 307

How To Place Comments in PL/SQL?

Answer:-

There are two ways to place comments into PL/SQL codes:

- * SQL Statement Style: Starts you comment any where in the line but prefixed with '--'. The comment ends at the end of the line.
- * C Language Style: Starts you comment any where in the line with '/*' and ends it with '*/' in the same line or some lines below.

Here is some example of PL/SQL comments:

```
BEGIN
  -- This is a comment
  /* To do:
     Need to write some codes here
  */
END;
```

[Read More Answers.](#)

Question # 308

What Are the Types PL/SQL Code Blocks?

Answer:-

There are 3 types of PL/SQL code blocks:

- * Anonymous Block - A block of codes with no name. It may contain a declaration part, an execution part, and exception handlers.
- * Stored Program Unit - A block of codes with a name. It is similar to an anonymous block. But it can take parameters and return values.
- * Trigger - A block of code that can be defined to fire based on a specific event.

[Read More Answers.](#)

Question # 309

What Is an Anonymous Block?

Answer:-

An anonymous block is a PL/SQL code block with no name. It consists of three parts:

- * Declaration Part - Defining local variables and local procedures. Declaration part is optional.
- * Execution Part - Defining execution logic with executable statements. Execution part is required.
- * Exception Part - Defining error handling logics. Exception part is optional.

Here how a complete anonymous block should look like:

```
DECLARE
  -- Declaration statements
BEGIN
  -- Executable statements
EXCEPTION
  -- Error handling statements
END;
```

[Read More Answers.](#)

Question # 310

What Is a Named Program Unit?

Answer:-

A named program unit is a PL/SQL code block with a name. It consists of three parts:

- * Declaration Part - Defining the program unit name, calling parameters, local variables and local procedures. Declaration part is required.
- * Execution Part - Defining execution logic with executable statements. Execution part is required.
- * Exception Part - Defining error handling logics. Exception part is optional.

There are two types of named program units:

- * Procedure - Has no return values.
- * Function - Has a return value.

[Read More Answers.](#)

Question # 311

What Is a Procedure?

Answer:-

A procedure is a named program unit. It consists of three parts:

- * Declaration Part - Defining the procedure name, calling parameters, local variables and local procedures. Declaration part is required.
- * Execution Part - Defining execution logic with executable statements. Execution part is required.
- * Exception Part - Defining error handling logics. Exception part is optional.

Here how a complete procedure should look like:

```
PROCEDURE name (parameter_1, parameter_2) AS
  -- Declaration statements
BEGIN
  -- Executable statements
EXCEPTION
```



```
-- Error handling statements  
END;
```

[Read More Answers.](#)

Question # 312

What Is a Function?

Answer:-

A function is a named program unit. It consists of three parts:

* Declaration Part - Defining the function name, calling parameters, return value type, local variables and local procedures. Declaration part is required.

* Execution Part - Defining execution logic with executable statements. Execution part is required.

* Exception Part - Defining error handling logics. Exception part is optional.

Here how a complete procedure should look like:

```
FUNCTION name (parameter_1, parameter_2) RETURN type AS
```

```
-- Declaration statements  
BEGIN  
-- Executable statements  
  RETURN value;  
EXCEPTION  
-- Error handling statements  
END;
```

[Read More Answers.](#)

Question # 313

How To Declare a Local Variable?

Answer:-

A local variable can be defined in the declaration part with a declaration statement, which is a variable name followed a data type identifier. Below are some examples of declaration statements:

```
PROCEDURE proc_var_1 AS
```

```
  domain VARCHAR2(80);
```

```
  price REAL;
```

```
  is_for_sale CHAR;
```

```
BEGIN
```

```
-- Executable statements
```

```
END;
```

[Read More Answers.](#)

Question # 314

How To Initialize Variables with Default Values?

Answer:-

There are two ways to assign default values to variables at the time of declaration:

* Using key word DEFAULT - Appending "DEFAULT value" to the end of declaration statements.

* Using assignment operator - Appending ":= value" to the end of declaration statements.

The script below show you some examples of declaration statements with default values:

```
PROCEDURE proc_var_1 AS
```

```
  domain VARCHAR2(80) := 'globalguideline.com';
```

```
  price REAL DEFAULT 999999.99;
```

```
  is_for_sale CHAR := 'N';
```

```
BEGIN
```

```
-- Executable statements
```

```
END;
```

[Read More Answers.](#)

Question # 315

How To Assign Values to Variables?

Answer:-

You can use assignment statements to assign values to variables. An assignment statement contains an assignment operator ":", which takes the value specified on the right to the variable on left. The script below show you some examples of assignment statements:

```
PROCEDURE proc_var_2 AS
```

```
  is_done BOOLEAN;
```

```
  counter NUMBER := 0;
```

```
  message VARCHAR2(80);
```

```
BEGIN
```

```
  is_done := FALSE;
```

```
  counter := counter + 1;
```

```
  message := 'Hello world!';
```

```
END;
```

[Read More Answers.](#)

Question # 316

What Are the Arithmetic Operations?

Answer:-



There are 4 basic arithmetic operations on numeric values as shown in the following sample script:

```
PROCEDURE proc_arithmetic AS
  addition NUMBER;
  subtraction NUMBER;
  multiplication NUMBER;
  division NUMBER;
BEGIN
  addition := 7 + 8;
  subtraction := addition - 7;
  multiplication := subtraction * 5;
  division := multiplication / 8;
  -- division should contain 5 now
END;
```

[Read More Answers.](#)

Question # 317

What Are the Numeric Comparison Operations?

Answer:-

PL/SQL supports 6 basic numeric comparison operations as shown in the following sample script:

```
PROCEDURE proc_comparison AS
  res BOOLEAN;
BEGIN
  res := 1 = 2;
  res := 1 < 2;
  res := 1 > 2;
  res := 1 <= 2;
  res := 1 >= 2;
  res := 1 <> 2;
  -- more statements
END;
```

[Read More Answers.](#)

Question # 318

What Are the Logical Operations?

Answer:-

PL/SQL supports 3 logical operations as shown in the following sample script:

```
PROCEDURE proc_comparison AS
  x BOOLEAN := TRUE;
  y BOOLEAN := FALSE;
  res BOOLEAN;
BEGIN
  res = x AND y;
  res = x OR y;
  res = NOT x;
  -- more statements
END;
```

[Read More Answers.](#)

Question # 319

How Many Categories of Data Types?

Answer:-

PL/SQL data types are grouped into 4 categories:

- * Scalar Data Types: A scalar data type holds a single value.
- * Composite Data Types: A composite data type has internal components, such as the elements of an array.
- * LOB Data Types: A LOB data type holds a lob locator that specifies the location of a large object.
- * Reference Data Types: A reference data type holds a pointer that points to another data object.

[Read More Answers.](#)

Question # 320

How Many Scalar Data Types Are Supported in PL/SQL?

Answer:-

PL/SQL supports many scalar data types divided into 4 groups:

- * Numeric Types: BINARY_DOUBLE, BINARY_FLOAT, BINARY_INTEGER, DEC, DECIMAL, DOUBLE PRECISION, FLOAT, INT, INTEGER, NATURAL, NATURALN, NUMBER, NUMERIC, PLS_INTEGER, POSITIVE, POSITIVEN, REAL, SIGNTYPE, SMALLINT.
- * Character Types: CHAR, CHARACTER, LONG, LONG RAW, NCHAR, NVARCHAR2, RAW, ROWID, STRING, UROWID, VARCHAR, VARCHAR2.
- * Boolean Types: BOOLEAN.
- * Date Types: DATE, TIMESTAMP, TIMESTAMP WITH TIMEZONE, TIMESTAMP WITH LOCAL TIMEZONE, INTERVAL YEAR TO MONTH, INTERVAL DAY TO SECOND.

[Read More Answers.](#)

Question # 321

How To Convert Character Types to Numeric Types?



Answer:-

You can convert character types to numeric types in two ways:

- * Explicitly by using TO_NUMBER() function.
- * Implicitly by putting character data in a numeric operation.

The sample script below shows you how to convert character types to numeric types:

```
PROCEDURE proc_convert_1 AS
start_time CHAR(5);
finish_time CHAR(5);
elapsed_time NUMBER(5);
BEGIN
start_time := '12052';
finish_time := '15314';
elapsed_time := TO_NUMBER(finish_time)
- TO_NUMBER(start_time);
elapsed_time := finish_time - start_time; -- same as above
END;
```

[Read More Answers.](#)

Question # 322

What Are the Execution Control Statements in Oracle?

Answer:-

PL/SQL supports three groups of execution control statements:

- * IF Statements - Conditionally executes a block of statements.
- * CASE Statements - Selectively executes a block of statements.
- * LOOP Statements - Repeatedly executes a block of statements.
- * GOTO Statements - Unconditional changes the execution flow to a specified statement.

The script below shows some execution control statements:

```
DECLARE
total NUMBER;
BEGIN
total := 0;
LOOP
total := total+1;
IF total >= 10 THEN
GOTO print;
END IF;
END LOOP;
<
DBMS_OUTPUT.PUT_LINE('Total counts: ' || TO_CHAR(total));
END;
```

This script should print this:

Total counts: 10

[Read More Answers.](#)

Question # 323

How To Use "IF" Statements on Multiple Conditions?

Answer:-

If you have multiple blocks of codes to be executed based on different conditions, you can use the "IF ... ELSIF" statement. Here is a sample script on IF statements:

```
DECLARE
day VARCHAR2;
BEGIN
day := 'SUNDAY';
IF day = 'THURSDAY' THEN
DBMS_OUTPUT.PUT_LINE('Checking log files. ');
ELSIF day = 'TUESDAY' THEN
DBMS_OUTPUT.PUT_LINE('Helping developers. ');
ELSIF day = 'FRIDAY' THEN
DBMS_OUTPUT.PUT_LINE('Rebuild indexes. ');
ELSE
DBMS_OUTPUT.PUT_LINE('Reading some papers. ');
END IF;
END;
```

This script should print this:

Reading some papers.

[Read More Answers.](#)

Question # 324

How To Use "WHILE" Statements in Oracle?

Answer:-

If you have a block of codes to be executed repeatedly based a condition, you can use the "WHILE ... LOOP" statement. Here is a sample script on WHILE statements:

```
DECLARE
total NUMBER;
BEGIN
total := 0;
```



```
WHILE total < 10 LOOP
  total := total+1;
END LOOP;
DBMS_OUTPUT.PUT_LINE('Total counts: ' || TO_CHAR(total));
END;
```

This script should print this:
Total counts: 10

[Read More Answers.](#)

Question # 325

How To Use "FOR" Statements in Oracle?

Answer:-

If you have a block of codes to be executed repeatedly over a range of values, you can use the "FOR ... LOOP" statement. Here is a sample script on FOR statements:

```
DECLARE
  total NUMBER := 0;
BEGIN
  FOR i IN 1..10 LOOP
    total := total + i;
  END LOOP;
  DBMS_OUTPUT.PUT_LINE('Total: ' || TO_CHAR(total));
END;
```

Note that temporary variable "i" used in the FOR loop needs no declaration. This script should print this:
Total: 55

[Read More Answers.](#)

Question # 326

What Is NULL in PL/SQL?

Answer:-

NULL is a reserved key word and it stands for two things in PL/SQL:

* NULL is an executable statement, and means doing nothing.

* NULL is a data value, and means no value.

The following sample script shows you examples of using NULL keyword:

```
DECLARE
  next_task CHAR(80);
BEGIN
  next_task := NULL; -- NULL value
  IF next_task IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('I am busy.');
```

ELSE
DBMS_OUTPUT.PUT_LINE('I am free.');

```
END IF;

IF next_task IS NULL THEN
  NULL; -- NULL statement
ELSE
  DBMS_OUTPUT.PUT_LINE('... working on ' || next_task);
END IF;
END;
```

This script should print this:
I am free.

[Read More Answers.](#)

Question # 327

How To Test NULL Values?

Answer:-

There are two special comparison operators you can use on NULL values:

* "variable IS NULL" - Returns TRUE if the variable value is NULL.

* "variable IS NOT NULL" - Return TRUE if the variable value is not NULL.

The following sample script shows you examples of comparing NULL values:

```
DECLARE
  next_task CHAR(80);
BEGIN
  next_task := NULL;
  IF next_task IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('I am busy.');
```

ELSE
DBMS_OUTPUT.PUT_LINE('I am free.');

```
END IF;

IF next_task IS NULL THEN
  NULL;
ELSE
  DBMS_OUTPUT.PUT_LINE('... working on ' || next_task);
END IF;
END;
```

Note that "variable = NULL" is not a valid operation. This script should print this:



I am free.

[Read More Answers.](#)

Question # 328

What Is a Procedure in Oracle?

Answer:-

A procedure is a named program unit. It consists of three parts:

- * Declaration Part - Defining the procedure name, calling parameters, local variables and local procedures. Declaration part is required.
- * Execution Part - Defining execution logic with executable statements. Execution part is required.
- * Exception Part - Defining error handling logics. Exception part is optional.

Here how a complete procedure should look like:

```
PROCEDURE name (parameter_1, parameter_2) AS
  -- Declaration statements
BEGIN
  -- Executable statements
EXCEPTION
  -- Error handling statements
END;
```

[Read More Answers.](#)

Question # 329

What Is an Oracle Function?

Answer:-

A function is a named program unit. It consists of three parts:

- * Declaration Part - Defining the function name, calling parameters, return value type, local variables and local procedures. Declaration part is required.
- * Execution Part - Defining execution logic with executable statements. Execution part is required.
- * Exception Part - Defining error handling logic. Exception part is optional.

Here how a complete procedure should look like:

```
FUNCTION name (parameter_1, parameter_2) RETURN type AS
  -- Declaration statements
BEGIN
  -- Executable statements
  RETURN value;
EXCEPTION
  -- Error handling statements
END;
```

[Read More Answers.](#)

Question # 330

How To Define an Anonymous Procedure without Variables?

Answer:-

Anonymous procedure is a procedure without any name. If you don't have any variables to declare, you can define an anonymous procedure by using the BEGIN keyword directly in SQL*Plus as shown in the following tutorial script:

```
SQL> set serveroutput on;
SQL> begin
  2  dbms_output.put_line('Welcome to globalguideline!');
  3  end;
  4  /
```

Welcome to globalguideline!

PL/SQL procedure successfully completed.

"set serveroutput on;" allows dbms_output.put_line() to work.

"/" runs the anonymous block, which print the "Welcome to globalguideline!" message.

[Read More Answers.](#)

Question # 331

How To Define an Anonymous Procedure with Variables?

Answer:-

Anonymous procedure is a procedure without any name. If you have some variables to declare, you can define an anonymous procedure by using the DECLARE keyword in SQL*Plus as shown in the following tutorial script:

```
SQL> set serveroutput on;
SQL> declare
  2  site char(80) := 'globalguideline';
  3  begin
  4  dbms_output.put_line('Welcome to ' || site);
  5  end;
  6  /
```

Welcome to globalguideline

PL/SQL procedure successfully completed.

SQL> /

Welcome to globalguideline

PL/SQL procedure successfully completed.

"/" runs the anonymous block, which print the "Welcome to globalguideline" message.

[Read More Answers.](#)



Question # 332

How To Create a Stored Procedure in Oracle?

Answer:-

A stored procedure is a procedure with a specified name and stored into the current database. If you want to create a stored procedure, you can use the CREATE PROCEDURE statement. The example script below creates a stored procedure:

```
SQL> CREATE PROCEDURE Greeting AS
2 BEGIN
3   DBMS_OUTPUT.PUT_LINE('Welcome to globalguideline!');
4 END;
5 /
```

Procedure created.

[Read More Answers.](#)

Question # 333

How To Execute a Stored Procedure in Oracle?

Answer:-

If you want to execute a stored procedure, you can use the EXECUTE statement. The example script below shows how to executes a stored procedure:

```
SQL> set serveroutput on;
SQL> CREATE PROCEDURE Greeting AS
2 BEGIN
3   DBMS_OUTPUT.PUT_LINE('Welcome to globalguideline!');
4 END;
5 /
```

Procedure created.

```
SQL> EXECUTE Greeting;
Welcome to globalguideline!
```

[Read More Answers.](#)

Question # 334

How To Drop a Stored Procedure in Oracle?

Answer:-

If there is an existing stored procedure and you don't want it any more, you can remove it from the database by using the DROP PROCEDURE statement as shown in the following script example:

```
SQL> CREATE PROCEDURE Greeting AS
2 BEGIN
3   DBMS_OUTPUT.PUT_LINE('Welcome to globalguideline!');
4 END;
5 /
```

Procedure created.

```
SQL> DROP PROCEDURE Greeting;
Procedure dropped.
```

[Read More Answers.](#)

Question # 335

How To Pass Parameters to Procedures in Oracle?

Answer:-

Store procedures or functions can take parameters. You need to define parameters while defining the procedure, and providing values to parameters while calling the procedure. The script below shows you how to do this:

```
SQL> CREATE OR REPLACE PROCEDURE DBA_TASK (day VARCHAR2) AS
2 BEGIN
3   IF day = 'MONDAY' THEN
4     DBMS_OUTPUT.PUT_LINE('Checking log files. ');
5   ELSIF day = 'FRIDAY' THEN
6     DBMS_OUTPUT.PUT_LINE('Rebuild indexes. ');
7   ELSE
8     DBMS_OUTPUT.PUT_LINE('Reading some papers. ');
9   END IF;
10 END;
11 /
```

```
SQL> EXECUTE DBA_TASK('MONDAY');
```

Checking log files.

```
SQL> EXECUTE DBA_TASK('SUNDAY');
```

Reading some papers.

As you can see, procedures with parameters can make procedures more flexible.

[Read More Answers.](#)

Question # 336

How To Create a Stored Function in Oracle?

Answer:-

A stored function is a function with a specified name and stored into the current database. If you want to create a stored function, you can use the CREATE FUNCTION statement. The example script below creates a stored procedure:

```
SQL> CREATE OR REPLACE FUNCTION GET_SITE
```



```
2 RETURN VARCHAR2 AS
3 BEGIN
4 RETURN 'globalguidelinee.com';
5 END;
6 /
```

Function created.

[Read More Answers.](#)

Question # 337

How To Call a Stored Function in Oracle?

Answer:-

A stored function can be called as part of expression in any PL/SQL statement. One simplest way to call a stored function is to a dummy SELECT statement as shown in the following tutorial script using SQL*Plus:

```
SQL> CREATE OR REPLACE FUNCTION GET_SITE
2 RETURN VARCHAR2 AS
3 BEGIN
4 RETURN 'globalguidelinee.com';
5 END;
6 /
```

Function created.

```
SQL> SELECT get_site() FROM dual;
GET_SITE()
```

globalguidelinee.com

DUAL is not a real table or view. It is commonly used to with SELECT statement to evaluate expressions.

[Read More Answers.](#)

Question # 338

How To Drop a Stored Function?

Answer:-

If there is an existing stored function and you don't want it any more, you can remove it from the database by using the DROP FUNCTION statement as shown in the following script example:

```
SQL> CREATE OR REPLACE FUNCTION GET_SITE
2 RETURN VARCHAR2 AS
3 BEGIN
4 RETURN 'globalguidelinee.com';
5 END;
6 /
```

Function created.

```
SQL> DROP FUNCTION GET_SITE;
Function dropped.
```

[Read More Answers.](#)

Question # 339

How To Call a Stored Function with Parameters?

Answer:-

You can define a function that takes parameters, provide values to those parameters when calling the function. Here is a good example of a function with a parameter:

```
SQL> CREATE OR REPLACE FUNCTION GET_DOUBLE(X NUMBER)
2 RETURN NUMBER AS
3 BEGIN
4 RETURN X * 2;
5 END;
6 /
```

Function created.

```
SQL> SELECT GET_DOUBLE(36) FROM DUAL;
GET_DOUBLE(36)
```

72

[Read More Answers.](#)

Question # 340

How To Define an Oracle Sub Procedure?

Answer:-

A sub procedure is a named procedure defined and used inside another procedure or function. You need to define a sub procedure in the declaration part of the enclosing procedure or function. Sub procedure definition starts with the PROCEDURE key word. Here is a sample script showing you how to define and use a sub procedure:

```
SQL> CREATE OR REPLACE PROCEDURE HR.DBA_WEEK AS
2 PROCEDURE DBA_TASK (day VARCHAR2) AS
3 BEGIN
4 IF day = 'MONDAY' THEN
5 DBMS_OUTPUT.PUT_LINE('Checking log files. ');
6 ELSIF day = 'FRIDAY' THEN
7 DBMS_OUTPUT.PUT_LINE('Rebuild indexes.');
```



```
8 ELSE
9   DBMS_OUTPUT.PUT_LINE('Reading some papers. ');
10 END IF;
11 END;
12 BEGIN
13   DBA_TASK('MONDAY');
14   DBA_TASK('TUESDAY');
15 END;
16 /
```

SQL> EXECUTE DBA_WEEK;
Checking log files.
Reading some papers.

[Read More Answers.](#)

Question # 341

How To Call a Sub Procedure?

Answer:-

To call a sub procedure, just use the sub procedure name as a statement. Here is another example of calling a sub procedure:

```
SQL> CREATE OR REPLACE PROCEDURE WELCOME AS
2  PROCEDURE WELCOME_PRINT(S CHAR) AS
3  BEGIN
4    DBMS_OUTPUT.PUT_LINE('Welcome to ' || S);
5  END;
6 BEGIN
7   WELCOME_PRINT('globalguideline');
8 END;
9 /
```

SQL> EXECUTE WELCOME;
Welcome to globalguideline

[Read More Answers.](#)

Question # 342

How To Define a Sub Function?

Answer:-

A sub function is a function defined and used inside another procedure or function. You need to define a sub function in the declaration part of the enclosing procedure or function. Sub function definition starts with the FUNCTION key word. Here is a sample script showing you how to define and use a sub function:

```
SQL> CREATE OR REPLACE PROCEDURE SUM_TEST AS
2  FUNCTION MY_SUM(X NUMBER, Y NUMBER)
3  RETURN NUMBER AS
4  BEGIN
5    RETURN X + Y;
6  END;
7 BEGIN
8   DBMS_OUTPUT.PUT_LINE('3 + 5 = ' ||
9     TO_CHAR(MY_SUM(3,5)));
10  DBMS_OUTPUT.PUT_LINE('5 + 3 = ' ||
11    TO_CHAR(MY_SUM(5,3)));
12 END;
13 /
```

SQL> EXECUTE SUM_TEST;
3 + 5 = 8
5 + 3 = 8

[Read More Answers.](#)

Question # 343

Can Sub Procedure/Function Be Called Recursively?

Answer:-

PL/SQL allows sub procedures or functions to be called recursively. The tutorial example below shows you how to calculate factorial values with a recursive sub function:

```
SQL> CREATE OR REPLACE PROCEDURE FACTORIAL_TEST AS
2  FUNCTION FACTORIAL(N NUMBER)
3  RETURN NUMBER AS
4  BEGIN
5    IF N <= 1 THEN
6      RETURN 1;
7    ELSE
8      RETURN N*FACTORIAL(N-1);
9    END IF;
10  END;
11 BEGIN
12  DBMS_OUTPUT.PUT_LINE('3! = ' ||
13    TO_CHAR(FACTORIAL(3)));
14  DBMS_OUTPUT.PUT_LINE('10! = ' ||
15    TO_CHAR(FACTORIAL(10)));
```



```
16 DBMS_OUTPUT.PUT_LINE('64! = ' ||
17 TO_CHAR(FACTORIAL(64)));
18 END;
19 /
```

```
SQL> EXECUTE FACTORIAL_TEST;
```

```
3! = 6
```

```
10! = 3628800
```

```
64! = 126886932185884164103433389335161480802000000000000...
```

There must be something wrong with the FACTORIAL() definition that causes those many extra '0's in the '64!' result.

[Read More Answers.](#)

Question # 344

What Happens If Recursive Calls Get Out of Control?

Answer:-

What happens if your code has bug on recursive procedure calls, which causes an infinite number nested procedure calls? The answer is so good. Oracle server seems to offer no protection calling stack limit. The script below shows you a badly coded recursive procedure. If you run it on an Oracle 10g XE server on Windows, your server will out of control and keep using virtual memory to satisfy the growing calling stack. You have to reboot your server to control back.

```
SQL> CREATE OR REPLACE PROCEDURE STACK_TEST AS
```

```
2 --Warning: do not run this procedure on your server
```

```
3 PROCEDURE STACK AS
```

```
4 BEGIN
```

```
5 STACK;
```

```
6 END;
```

```
7 BEGIN
```

```
8 STACK;
```

```
9 END;
```

```
10 /
```

```
SQL> EXECUTE STACK_TEST;
```

(your server keep running with 100% CPU and memory usage)

[Read More Answers.](#)

Question # 345

What Is the Order of Defining Local Variables and Sub Procedures/Functions?

Answer:-

In the declaration part, you must define all local variables before defining any sub procedures or sub functions. See the following sample script:

```
SQL> CREATE OR REPLACE PROCEDURE WELCOME AS
```

```
2 SITE CHAR(80) := 'globalguideline';
```

```
3 PROCEDURE WELCOME_PRINT(S CHAR) AS
```

```
4 BEGIN
```

```
5 DBMS_OUTPUT.PUT_LINE('Welcome to ' || S);
```

```
6 END;
```

```
7 BEGIN
```

```
8 WELCOME_PRINT(SITE);
```

```
9 END;
```

```
10 /
```

```
SQL> EXECUTE WELCOME;
```

```
Welcome to globalguideline
```

Notice that variable SITE should be declared before procedure WELCOME_PRINT

[Read More Answers.](#)

Question # 346

What Is the Difference between Formal Parameters and Actual Parameters?

Answer:-

Formal parameter and actual parameter are two different terms related parameters used in the procedures and functions:

* A formal parameter is a term used to refer to a parameter defined in the procedure or function declaration statement.

* An actual parameter is a term used to refer to a parameter provided by the calling statement to a procedure or a function.

[Read More Answers.](#)

Question # 347

What Are the Parameter Modes Supported by PL/SQL?

Answer:-

PL/SQL supports 3 parameter modes on procedure/function parameters:

* IN: This is the default mode. IN parameters allow the calling code to pass values into the procedure or function.

* OUT: OUT parameters allow the procedure or function to pass values back to the calling code.

* IN OUT: IN OUT parameters allow the calling code to pass values into and receive values from procedure or function.

[Read More Answers.](#)

Question # 348

How To Use "IN" Parameter Properly?

Answer:-



Here are the rules about IN parameters:

- * A formal IN parameter acts like constant. It can not be assigned with new values.
- * An actual IN parameter can take a value or a variable.
- * An actual IN parameter is passed by reference to the specified value or the value of the specified variable.
- * An actual IN parameter will not receive any value from the formal parameter.

Here is good example of a procedure with an IN parameter:

```
SQL> CREATE OR REPLACE PROCEDURE WELCOME AS
  2  SITE CHAR(80) := 'globalguideline.com';
  3  PROCEDURE WELCOME_PRINT(S IN CHAR) AS
  4  BEGIN
  5    DBMS_OUTPUT.PUT_LINE('Welcome to ' || S);
  6    -- S := 'Google.com'; -- Not allowed
  7  END;
  8 BEGIN
  9  WELCOME_PRINT('MySpace.com');
 10  WELCOME_PRINT(SITE);
 11 END;
 12 /
SQL> EXECUTE WELCOME;
Welcome to MySpace.com
Welcome to globalguideline.com
```

[Read More Answers.](#)

Question # 349

How To Use "OUT" Parameter Properly?

Answer:-

Here are the rules about OUT parameters:

- * A formal OUT parameter acts like an un-initialized variable. It must be assigned with new values before the end of the procedure or function.
- * An actual OUT parameter must be a variable.
- * An actual OUT parameter will not pass any value to the formal parameter.
- * An actual OUT parameter will receive a copy of the value from the formal parameter at the end of the procedure or function.

Here is good example of a procedure with an OUT parameter:

```
SQL> CREATE OR REPLACE PROCEDURE WELCOME AS
  2  SITE CHAR(40) := 'globalguideline.com';
  3  MESSAGE CHAR(80);
  4  PROCEDURE WELCOME_PRINT(S IN CHAR, M OUT CHAR) AS
  5  BEGIN
  6    M := 'Welcome to ' || S;
  7  END;
  8 BEGIN
  9  WELCOME_PRINT('MySpace.com', MESSAGE);
 10  DBMS_OUTPUT.PUT_LINE(MESSAGE);
 11  WELCOME_PRINT(SITE, MESSAGE);
 12  DBMS_OUTPUT.PUT_LINE(MESSAGE);
 13 END;
 14 /
SQL> EXECUTE WELCOME;
Welcome to MySpace.com
Welcome to globalguideline.com
```

[Read More Answers.](#)

Question # 350

How To Use "IN OUT" Parameter Properly?

Answer:-

Here are the rules about IN OUT parameters:

- * A formal IN OUT parameter acts like an initialized variable.
- * An actual IN OUT parameter must be a variable.
- * An actual IN OUT parameter passes a copy of its value to the formal parameter when entering the procedure or function.
- * An actual IN OUT parameter will receive a copy of the value from the formal parameter at the end of the procedure or function.

Here is good example of a procedure with IN OUT parameters:

```
SQL> CREATE OR REPLACE PROCEDURE SWAP_TEST AS
  2  A NUMBER := 3;
  3  B NUMBER := 8;
  4  PROCEDURE MY_SWAP(X IN OUT NUMBER, Y IN OUT NUMBER) AS
  5  T NUMBER;
  6  BEGIN
  7    T := X;
  8    X := Y;
  9    Y := T;
 10  END MY_SWAP;
 11 BEGIN
 12  MY_SWAP(A,B);
 13  DBMS_OUTPUT.PUT_LINE('A = ' || TO_CHAR(A));
 14  DBMS_OUTPUT.PUT_LINE('B = ' || TO_CHAR(B));
 15 END;
 16 /
SQL> EXECUTE SWAP_TEST;
A = 8
```



B = 3

[Read More Answers.](#)

Question # 351

How To Define Default Values for Formal Parameters?

Answer:-

If you have an IN parameter, you can make it as an optional parameter for the calling statement by defining the formal parameter with the DEFAULT clause. This gives you the freedom of not providing the actual parameter when calling this procedure or function. See the following tutorial script shows you an example procedure with an optional parameter:

```
SQL> CREATE OR REPLACE PROCEDURE WELCOME AS
  2  PROCEDURE GREETING(S IN CHAR DEFAULT 'globalguideline') AS
  3  BEGIN
  4    DBMS_OUTPUT.PUT_LINE('Welcome to ' || S);
  5  END;
  6 BEGIN
  7  GREETING('MySpace.com');
  8  GREETING;
  9 END;
 10 /
```

Procedure created.

```
SQL> EXECUTE WELCOME;
Welcome to MySpace.com
Welcome to globalguideline
```

[Read More Answers.](#)

Question # 352

What Are Named Parameters?

Answer:-

Named parameters are actual parameters specified not by position but by providing formal parameter names when calling the procedure or function. The main advantage of named parameters is that the caller don't have to remember the position of each parameter. But the caller have to remember the formal parameter names.

The script below illustrates how to use named parameters:

```
SQL> CREATE OR REPLACE PROCEDURE SWAP_TEST AS
  2  A NUMBER := 3;
  3  B NUMBER := 8;
  4  PROCEDURE MY_SWAP(X IN OUT NUMBER, Y IN OUT NUMBER) AS
  5    T NUMBER;
  6  BEGIN
  7    T := X;
  8    X := Y;
  9    Y := T;
 10  END MY_SWAP;
 11 BEGIN
 12  MY_SWAP(Y=>B, X=>A); -- same as (X=>A, Y=B), OR (A,B)
 13  DBMS_OUTPUT.PUT_LINE('A = ' || TO_CHAR(A));
 14  DBMS_OUTPUT.PUT_LINE('B = ' || TO_CHAR(B));
 15 END;
 16 /
```

```
SQL> EXECUTE SWAP_TEST;
```

A = 8

B = 3

[Read More Answers.](#)

Question # 353

What Is the Scope of a Local Variable?

Answer:-

The scope of a variable can be described with these rules:

* A variable is valid within the procedure or function where it is defined.

* A variable is also valid inside a sub procedure or function defined.

* If a variable name is collided with another variable in a sub procedure or function, this variable becomes not visible in that sub procedure or function.

Here is a sample script to show you those rules:

The script below illustrates how to use named parameters:

```
SQL> CREATE OR REPLACE PROCEDURE PARENT AS
  2  X CHAR(10) := 'gg!';
  3  Y NUMBER := 999999.00;
  4  PROCEDURE CHILD AS
  5    Y CHAR(10) := 'CENTER';
  6    Z NUMBER := -1;
  7  BEGIN
  8    DBMS_OUTPUT.PUT_LINE('X = ' || X); -- X from PARENT
  9    DBMS_OUTPUT.PUT_LINE('Y = ' || Y); -- Y from CHILD
 10    DBMS_OUTPUT.PUT_LINE('Z = ' || TO_CHAR(Z));
 11  END;
 12 BEGIN
 13  DBMS_OUTPUT.PUT_LINE('X = ' || X); -- X from PARENT
 14  DBMS_OUTPUT.PUT_LINE('Y = ' || TO_CHAR(Y));
 15  -- DBMS_OUTPUT.PUT_LINE('Z = ' || TO_CHAR(Z));
```



```
16 CHILD;
17 END;
18 /
SQL> EXECUTE PARENT;
X = ggl
Y = 999999
X = ggl
Y = CENTER
Z = -1
```

[Read More Answers.](#)

Question # 354

Can DML Statements Be Used in PL/SQL?

Answer:-

Yes, you can run almost any DML statements in PL/SQL directly. To manipulate Oracle database data you can include INSERT, UPDATE, and DELETE statements, directly in PL/SQL programs, without any special notation, as shown in the following sample code:

(Connect to XE with SQL*Plus)

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
  first_name VARCHAR(80) NOT NULL,
  last_name VARCHAR(80) NOT NULL);
```

Table created.

```
SELECT COUNT(*) FROM student;
COUNT(*)
```

```
-----
0
CREATE OR REPLACE PROCEDURE HELLO AS
BEGIN
  INSERT INTO student VALUES(29, 'Bob', 'Henry');
  INSERT INTO student VALUES(30, 'Joe', 'Bush');
  UPDATE student SET first_name = 'ggl' WHERE id = 30;
  DELETE FROM student WHERE id = 29;
END;
```

```
/
SELECT * FROM student;
<pre>ID    FIRST_NAME LAST_NAME
```

```
-----
30    ggl      Bush</pre>
```

[Read More Answers.](#)

Question # 355

Can DDL Statements Be Used in PL/SQL?

Answer:-

No, you can not run any DDL statements in PL/SQL directly. If you try to use the DROP TABLE statement inside PL/SQL, you will get a compilation error as shown below:

(Connect to XE with SQL*Plus)

```
BEGIN
  DROP TABLE student; -- compilation error
END;
```

```
/
```

[Read More Answers.](#)

Question # 356

Can Variables Be Used in SQL Statements?

Answer:-

Yes, you can use variables in SQL statements as part of any expressions. The tutorial script provides you some good examples:

(Connect to XE with SQL*Plus)

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
  first_name VARCHAR(80) NOT NULL,
  last_name VARCHAR(80) NOT NULL);
```

Table created.

```
DECLARE
  var_id NUMBER;
  var_name CHAR(10);
BEGIN
  var_id := 29;
  var_name := 'Bob';
  INSERT INTO student VALUES(var_id, var_name, 'Henry');
  var_name := 'Joe';
  INSERT INTO student VALUES(var_id+1, var_name, 'Bush');
  var_name := 'ggl';
  UPDATE student SET first_name = var_name
  WHERE id = var_id+1;
  DELETE FROM student WHERE id = var_id;
END;
```

```
/
```

```
SELECT * FROM student;
```



```
<pre>ID    FIRST_NAME LAST_NAME
```

```
-----  
30    ggl      Bush </pre>
```

[Read More Answers.](#)

Question # 357

What Happens If Variable Names Collide with Table/Column Names?

Answer:-

When a variable name collides with a column name, PL/SQL will use it as the variable if it is used where variable is allowed; It will be used as the column, if it is used where variable is not allowed but column is allowed. Here is a good example of variable names collide with column names:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,  
  first_name VARCHAR(80) NOT NULL,  
  last_name VARCHAR(80) NOT NULL);  
Table created.  
DECLARE  
  id NUMBER;  
  first_name CHAR(10);  
BEGIN  
  id := 29;  
  first_name := 'Bob';  
  INSERT INTO student VALUES(id, first_name, 'Henry');  
  first_name := 'Joe';  
  INSERT INTO student VALUES(id+1, first_name, 'Bush');  
  first_name := 'ggl';  
  UPDATE student SET first_name = first_name WHERE id = 29;  
  -- 1 row updated  
  -- Both 'first_name's are treated as column names  
  UPDATE student SET first_name = first_name  
  WHERE id = id+1;  
  -- 0 rows updated  
  -- Both "id"s are treated as variable names  
  
  DELETE FROM student WHERE id = id;  
  -- 2 rows deleted  
END;  
/
```

[Read More Answers.](#)

Question # 358

How To Resolve Name Conflicts between Variables and Columns?

Answer:-

The best way to resolve name conflicts is to avoid using column names for variables.

[Read More Answers.](#)

Question # 359

How To Assign Query Results to Variables?

Answer:-

If you want to assign results from SELECT statements to variables, you can use the INTO clause, which an extension of SELECT statements for PL/SQL. The sample code below shows some good example on INTO clause:

```
DECLARE  
  total NUMBER;  
  now DATE;  
  fname VARCHAR2(10);  
  lname VARCHAR2(10);  
BEGIN  
  SELECT COUNT(*) INTO total FROM employees;  
  DBMS_OUTPUT.PUT_LINE('Count = ' || TO_CHAR(total));  
  SELECT SYSDATE INTO now FROM DUAL;  
  DBMS_OUTPUT.PUT_LINE('Now = ' || TO_CHAR(now, 'SSSS'));  
  SELECT first_name, last_name INTO fname, lname  
  FROM employees  
  WHERE employee_id = 100;  
  DBMS_OUTPUT.PUT_LINE('Name = ' || fname || ' ' || lname);  
END;  
/
```

Count = 107
Now = 82375
Name = Steven King

[Read More Answers.](#)

Question # 360

Can You Assign Multiple Query Result Rows To a Variable?

Answer:-



You can use "SELECT ... INTO variable" to assign query results to variables. But what happens if the SELECT statements return multiple rows? The answer is that you will get a run time error. The following tutorial exercise shows this error condition:

```
DECLARE
  fname VARCHAR2(10);
  lname VARCHAR2(10);
BEGIN
  SELECT first_name, last_name INTO fname, lname
    FROM employees WHERE employee_id = 100;
  DBMS_OUTPUT.PUT_LINE('Name = ' || fname || ' ' || lname);
  SELECT first_name, last_name INTO fname, lname
    FROM employees WHERE employee_id > 100;
  DBMS_OUTPUT.PUT_LINE('Name = ' || fname || ' ' || lname);
END;
/
ORA-01422: exact fetch returns more than requested number
of rows
ORA-06512: at line 8
Name = Steven King
```

[Read More Answers.](#)

Question # 361

How To Run SQL Functions in PL/SQL?

Answer:-

Of course, you can run SQL functions in SQL statements. But many SQL functions can also be executed in regular PL/SQL statements, as shown in the following sample script:

```
DECLARE
  now DATE;
  id NUMBER;
  str VARCHAR2(40);
BEGIN
  now := SYSDATE;
  DBMS_OUTPUT.PUT_LINE('Time #1 = ' ||
    TO_CHAR(now, 'HH24:MI:SS'));
  SELECT SYSDATE INTO now FROM DUAL;
  DBMS_OUTPUT.PUT_LINE('Time #2 = ' ||
    TO_CHAR(now, 'HH24:MI:SS'));
  id := UID;
  DBMS_OUTPUT.PUT_LINE('User id #2 = ' || TO_CHAR(id));
  SELECT UID INTO id FROM DUAL;
  DBMS_OUTPUT.PUT_LINE('User id #2 = ' || TO_CHAR(id));

  str := CHR(70)||CHR(89)||CHR(73);
  DBMS_OUTPUT.PUT_LINE('String #1 = ' || str);
  SELECT CHR(70)||CHR(89)||CHR(73) INTO str FROM DUAL;
  DBMS_OUTPUT.PUT_LINE('String #2 = ' || str);
END;
/
Time #1 = 21:41:16
Time #2 = 21:41:16
User id #2 = 33
User id #2 = 33
String #1 = ggl
String #2 = ggl
```

[Read More Answers.](#)

Question # 362

How To Retrieve the Count of Updated Rows?

Answer:-

After running an UPDATE statement, the database server returns a count of updated rows. You can retrieve this count from a special predefined variable called SQL%ROWCOUNT, as shown in the following tutorial:

```
CREATE TABLE emp_temp AS SELECT * FROM employees;
BEGIN
  UPDATE emp_temp SET salary = salary * 1.05
    WHERE salary < 5000;
  DBMS_OUTPUT.PUT_LINE('# of rows updated: ' ||
    SQL%ROWCOUNT);
END;
/
# of rows updated: 49
```

[Read More Answers.](#)

Question # 363

What Is the Implicit Cursor in Oracle?

Answer:-

The implicit cursor is the cursor automatically defined by PL/SQL for you. Whenever a SQL statement is executed, this cursor will be assigned to represent the execution of this statement. This implicit cursor is called SQL. It has many attributes representing some good information about the execution like:



- * SQL%FOUND - True, if the SQL statement has changed any rows.
- * SQL%NOTFOUND - True, if the SQL statement has not changed any rows.
- * SQL%ROWCOUNT - The number of rows affected by the SQL statement.

See the example below on how to use the implicit cursor:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,  
  first_name VARCHAR(80) NOT NULL,  
  last_name VARCHAR(80) NOT NULL);
```

Table created.

[Read More Answers.](#)

Question # 364

How To Assign Data of the Deleted Row to Variables?

Answer:-

If a DELETE statement is deleting a single row, you can assign column values of the deleted row to variables by using the RETURNING clause, which an extension of DELETE statements for PL/SQL. The tutorial script shows you how to do this:

```
CREATE TABLE emp_temp AS SELECT * FROM employees;  
DECLARE  
  fname VARCHAR2(10);  
  lname VARCHAR2(10);  
BEGIN  
  DELETE FROM emp_temp WHERE employee_id = 100;  
  RETURNING first_name, last_name INTO fname, lname;  
  DBMS_OUTPUT.PUT_LINE('Name deleted = ' || fname || '  
  || lname);  
  -- This will not work because multiple rows deleted  
  -- DELETE FROM emp_temp WHERE employee_id > 100;  
  -- RETURNING first_name, last_name INTO fname, lname;  
END;
```

Name deleted = Steven King

Similar to SELECT ... INTO, RETURNING ... INTO will not work if multiple rows are deleted.

[Read More Answers.](#)

Question # 365

What Is a RECORD in PL/SQL?

Answer:-

RECORD is a composite data type in PL/SQL. It can have many fields representing data elements with different data types. Variables of RECORD type can be designed to hold data from database table rows. To use RECORD data type, you need to define a specific RECORD type with each field defined with a name and a data type. Then you can use your specific RECORD type to define your RECORD type variables.

[Read More Answers.](#)

Question # 366

How To Define a Specific RECORD Type?

Answer:-

If you want to define a specific RECORD type, you need to use the TYPE ... IS RECORD statement in the declaration part of any procedure or function. The following example script defines a RECORD type called STUDENT:

```
CREATE OR REPLACE PROCEDURE HELLO AS  
  TYPE student IS RECORD (  
    id NUMBER(5),  
    first_name VARCHAR(80),  
    last_name VARCHAR(80)  
  );  
BEGIN  
  NULL;  
END;
```

[Read More Answers.](#)

Question # 367

How To Define a Variable of a Specific RECORD Type?

Answer:-

Once you have your specific RECORD type defined, you can define new variables with this specific RECORD type like any other data type. In the sample script below, several variables are defined with a regular data type and a specific RECORD type:

```
CREATE OR REPLACE PROCEDURE HELLO AS  
  TYPE student IS RECORD (  
    id NUMBER(5),  
    first_name VARCHAR(80),  
    last_name VARCHAR(80)  
  );  
  best_student student;  
  another_student student;  
  class_name VARCHAR2(80);  
BEGIN
```



```
NULL;  
END;  
/
```

[Read More Answers.](#)

Question # 368

How To Assign Values to Data Fields in RECORD Variables?

Answer:-

If a variable is a RECORD variable, you can assign values to its data fields by using fields names prefixed with variable name as "variable.field_name". Here is a sample script assigning values to data fields of RECORD variables:

```
CREATE OR REPLACE PROCEDURE HELLO AS
```

```
TYPE student IS RECORD (  
  id NUMBER(5),  
  first_name VARCHAR(80),  
  last_name VARCHAR(80)  
);  
best_student student;  
another_student student;  
class_name VARCHAR2(80);  
BEGIN  
class_name := 'gg1 Center 2006';  
best_student.first_name := 'The';  
best_student.last_name := 'Best';  
END;  
/
```

[Read More Answers.](#)

Question # 369

How To Retrieve Values from Data Fields in RECORD Variables?

Answer:-

If a variable is a RECORD variable with data fields assigned values, you can retrieve those values out of its data fields by using fields names prefixed with variable name as "variable.field_name". Here is a sample script showing you how to retrieve field values from RECORD variables:

```
CREATE OR REPLACE PROCEDURE HELLO AS
```

```
TYPE student IS RECORD (  
  id NUMBER(5),  
  first_name VARCHAR(80),  
  last_name VARCHAR(80)  
);  
best_student student;  
another_student student;  
class_name VARCHAR2(80);  
BEGIN  
class_name := 'gg1 Center 2006';  
best_student.first_name := 'The';  
best_student.last_name := 'Best';  
DBMS_OUTPUT.PUT_LINE('Best student = ' ||  
  best_student.first_name || '  
  || best_student.last_name);  
END;  
/
```

Best student = The Best

[Read More Answers.](#)

Question # 370

How To Define a Data Field as NOT NULL?

Answer:-

When defining a specific RECORD type, you can define a data field as NOT NULL to make sure variables with this RECORD type to always have values in this field. A field defined as NOT NULL must have a default value. Here is a tutorial script showing you how to define a data field as NOT NULL:

```
CREATE OR REPLACE PROCEDURE HELLO AS
```

```
TYPE student IS RECORD (  
  id NUMBER(5) NOT NULL DEFAULT 0,  
  first_name VARCHAR(80),  
  last_name VARCHAR(80)  
);  
best_student student;  
another_student student;  
class_name VARCHAR2(80);  
BEGIN  
class_name := 'gg1 Center 2006';  
best_student.first_name := 'The';  
best_student.last_name := 'Best';  
DBMS_OUTPUT.PUT_LINE('Best student ID = '  
  || best_student.id);  
DBMS_OUTPUT.PUT_LINE('Best student = ' ||  
  best_student.first_name || '  
  || best_student.last_name);
```



```
END;
/  
Best student ID = 0  
Best student = The Best
```

[Read More Answers.](#)

Question # 371

How To Define a RECORD Variable to Store a Table Row?

Answer:-

If you have a table, and want to define a RECORD variable to store all the data elements of a row from that table, you can use table_name%ROWTYPE to define the RECORD variable as shown in the following sample script:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,  
  first_name VARCHAR(80) NOT NULL,  
  last_name VARCHAR(80) NOT NULL);
```

Table created.

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
```

```
  best_student student%ROWTYPE;  
  another_student student%ROWTYPE;  
  class_name VARCHAR2(80);
```

BEGIN

```
  class_name := 'ggl Center 2006';  
  best_student.first_name := 'The';  
  best_student.last_name := 'Best';  
  DBMS_OUTPUT.PUT_LINE('Best student ID = '  
    || best_student.id);  
  DBMS_OUTPUT.PUT_LINE('Best student = '  
    || best_student.first_name || '  
    || best_student.last_name);
```

END;

/

```
Best student ID =  
Best student = The Best
```

[Read More Answers.](#)

Question # 372

How To Assign a Table Row to a RECORD Variable?

Answer:-

If you have a table, and want to assign a data row of that table to a RECORD variable, you need to define this RECORD variable to match the table column structure, then use the SELECT ... INTO statement to assign a data row that RECORD variable. The script below shows you how to do this:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS  
  manager employees%ROWTYPE;
```

BEGIN

```
  SELECT * INTO manager FROM employees  
    WHERE employee_id = 100;  
  DBMS_OUTPUT.PUT_LINE('My manager = ' ||  
    || manager.first_name || ' ' || manager.last_name);
```

END;

/

```
My manager = Steven King
```

[Read More Answers.](#)

Question # 373

How To Insert a Record into a Table?

Answer:-

If you have a RECORD variable with data fields matching a table structure, you can insert a row to this table with this RECORD variable using the INSERT statement as shown in the example below:

```
CREATE TABLE emp_temp AS SELECT * FROM employees;
```

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS  
  manager employees%ROWTYPE;
```

BEGIN

```
  SELECT * INTO manager FROM employees  
    WHERE employee_id = 100;  
  manager.employee_id := 299;  
  INSERT INTO emp_temp VALUES manager;  
  DBMS_OUTPUT.PUT_LINE('# rows inserted = '  
    || SQL%ROWCOUNT);
```

END;

/

```
# rows inserted = 1
```

[Read More Answers.](#)

Question # 374

How To Update a Table Row with a Record?

Answer:-



If you have a RECORD variable with data fields matching a table structure, you can update a row in this table with this RECORD variable using the UPDATE ... SET ROW statement as shown in the sample script below:

```
CREATE TABLE emp_temp AS SELECT * FROM employees;
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
  manager employees%ROWTYPE;
BEGIN
  SELECT * INTO manager FROM employees
    WHERE employee_id = 100;
  manager.employee_id := 299;
  INSERT INTO emp_temp VALUES manager;
  manager.first_name := 'ggl';
  manager.last_name := 'Center';
  UPDATE emp_temp SET ROW = manager WHERE employee_id = 299;
  DBMS_OUTPUT.PUT_LINE('# rows updated = ' || SQL%ROWCOUNT);
END;
/
# rows updated = 1
```

[Read More Answers.](#)

Question # 375

How To Define a Variable to Match a Table Column Data Type?

Answer:-

If you have a table, and want to define some variables to have exactly the same data types as some columns in that table, you can use table_name.column_name%TYPE as data types to define those variables. The tutorial sample below shows you how to do this:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
  my_email employees.email%TYPE;
  my_salary employees.salary%TYPE;
BEGIN
  SELECT email, salary INTO my_email, my_salary
    FROM employees WHERE employee_id = 101;
  DBMS_OUTPUT.PUT_LINE('My email = ' || my_email);
  DBMS_OUTPUT.PUT_LINE('My salary = ' || my_salary);
END;
/
My email = NKOCHHAR
My salary = 17000
```

[Read More Answers.](#)

Question # 376

What Is a Cursor in Oracle?

Answer:-

A cursor looks like a variable, but it is not a variable. A cursor looks like a procedure, but it is not a procedure. A cursor is a cursor. It is a logical representation of a resource connects to a set of data rows related to a DML statement. A cursor consists of:

- * A DML statement with or without parameters.
- * A status to indicate whether it is connected or not.
- * A pointer to indicate the current row in the resource set.
- * A number of attributes about the cursor, like FOUND, NOTFOUND and ROWCOUNT.

[Read More Answers.](#)

Question # 377

How Many Types of Cursors Supported in PL/SQL?

Answer:-

PL/SQL supports two types of cursors:

- * The implicit cursor - A single default cursor that automatically connects to the last DML statement executed.
- * Explicit cursors - User defined cursors with specific DML statements and execution statuses.

[Read More Answers.](#)

Question # 378

What Is the Oracle Implicit Cursor?

Answer:-

There is only one implicitly cursor in a session. The implicit cursor is the cursor automatically defined by PL/SQL for you. Whenever a SQL statement is executed, this cursor will be assigned to represent the execution of this statement. This implicit cursor is called SQL. It has many attributes representing some good information about the execution like:

- * SQL%FOUND - True, if the SQL statement has changed any rows.
- * SQL%NOTFOUND - True, if the SQL statement has not changed any rows.
- * SQL%ROWCOUNT - The number of rows affected by the SQL statement.
- * %ISOPEN - True, if there is a SQL statement being associated to the cursor.

[Read More Answers.](#)

Question # 379

How To Use Attributes of the Implicit Cursor in Oracle?



Answer:-

Right after executing a DML statement, you retrieve any attribute of the implicit cursor by using SQL%attribute_name, as shown in the following tutorial exercise:

```
CREATE TABLE student (id NUMBER(5) PRIMARY KEY,
  first_name VARCHAR(80) NOT NULL,
  last_name VARCHAR(80) NOT NULL);
Table created.
DECLARE
  id NUMBER;
  first_name CHAR(10);
BEGIN
  id := 29;
  first_name := 'Bob';
  INSERT INTO student VALUES(id, first_name, 'Henry');
  first_name := 'Joe';
  INSERT INTO student VALUES(id+1, first_name, 'Bush');
  DBMS_OUTPUT.PUT_LINE('# of rows inserted: '
    || SQL%ROWCOUNT);

  first_name := 'ggl';
  UPDATE student SET first_name = first_name WHERE id = 29;
  IF SQL%FOUND THEN
    DBMS_OUTPUT.PUT_LINE('# of rows updated: '
      || SQL%ROWCOUNT);
  END IF;
  UPDATE student SET first_name = first_name
    WHERE id = id+1;
  IF SQL%NOTFOUND THEN
    DBMS_OUTPUT.PUT_LINE('No records updated.');
```

END IF;

```
DELETE FROM student WHERE id = id;
DBMS_OUTPUT.PUT_LINE('# of rows deleted: '
  || SQL%ROWCOUNT);
END;
/
# of rows inserted: 1
# of rows updated: 1
No records updated.
# of r
```

[Read More Answers.](#)

Question # 380

How To Loop through Data Rows in the Implicit Cursor?

Answer:-

You use the FOR ... IN ... LOOP statement to loop through data rows in the implicit cursor as the following syntax:

```
FOR row IN dml_statement LOOP
  (statement block with row.field)
END LOOP;
```

Here "row" is a local RECORD type variable with fields automatically defined to match the fields in the data rows resulted from the DML statement. Here is a good tutorial exercise on loop through data rows with the implicit cursor:

```
BEGIN
  FOR row IN (SELECT * FROM employees
    WHERE manager_id = 101) LOOP
    DBMS_OUTPUT.PUT_LINE('Name = ' || row.last_name);
  END LOOP;
END;
/
Name = Greenberg
Name = Whalen
Name = Mavris
Name = Baer
Name = Higgins
```

[Read More Answers.](#)

Question # 381

How To Define an Explicit Cursor in Oracle?

Answer:-

An explicit cursor must be defined in the declaration part of a procedure or function with the CURSOR ... IS statement as shown in the following sample script:

```
DECLARE
  CURSOR c_list IS SELECT * FROM countries;
  CURSOR t_list IS SELECT * FROM employees
    WHERE employee_id = 100;
BEGIN
  NULL;
END;
/
```

[Read More Answers.](#)



Question # 382

How To Open and Close an Explicit Cursor in Oracle?

Answer:-

An existing cursor can be opened or closed by the OPEN or CLOSE statement as shown in the following sample script:

```
DECLARE
  CURSOR c_list IS SELECT * FROM countries;
  CURSOR t_list IS SELECT * FROM employees
    WHERE employee_id = 100;
BEGIN
  OPEN c_list;
  OPEN t_list;
  CLOSE c_list;
  CLOSE t_list;
END;
/
```

[Read More Answers.](#)

Question # 383

How To Retrieve Data from an Explicit Cursor?

Answer:-

If you have a cursor opened ready to use, you can use the FETCH ... INTO statement to retrieve data from the cursor into variables. FETCH statement will:

* Retrieve all the fields from the row pointed by the current cursor pointer and assign them to variables listed in the INTO clause.

* Move the cursor pointer to the next row.

* Update cursor attributes like FOUND and NOTFOUND.

Here is a sample script showing you how to use FETCH statement:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
  CURSOR t_list IS SELECT first_name, last_name
    FROM employees;
  f_name VARCHAR2(10);
  l_name VARCHAR2(10);
BEGIN
  OPEN t_list;
  FETCH t_list INTO f_name, l_name;
  DBMS_OUTPUT.PUT_LINE('Name = ' || f_name || ' '
    || l_name);
  FETCH t_list INTO f_name, l_name;
  DBMS_OUTPUT.PUT_LINE('Name = ' || f_name || ' '
    || l_name);
  -- FETCH t_list INTO l_name; -- must have two variables
  CLOSE t_list;
END;
/
Name = Ellen Abel
Name = Sundar Ande
```

[Read More Answers.](#)

Question # 384

How To Retrieve Data from an Cursor to a RECORD?

Answer:-

If you have a cursor opened ready to use, you can also use the FETCH statement to retrieve data from the cursor into a RECORD variable as shown in the tutorial exercise below:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
  CURSOR t_list IS SELECT first_name, last_name
    FROM employees;
  TYPE name_rec IS RECORD (
    f_name VARCHAR2(10),
    l_name VARCHAR2(10)
  );
  n name_rec;
BEGIN
  OPEN t_list;
  FETCH t_list INTO n;
  DBMS_OUTPUT.PUT_LINE('Name = ' || n.f_name || ' '
    || n.l_name);
  FETCH t_list INTO n;
  DBMS_OUTPUT.PUT_LINE('Name = ' || n.f_name || ' '
    || n.l_name);
  CLOSE t_list;
END;
/
Name = Ellen Abel
Name = Sundar Ande
```

[Read More Answers.](#)

Question # 385



How To Use FETCH Statement in a Loop?

Answer:-

If you have a cursor opened ready to use, you can also use the FETCH statement in a loop to retrieve data from the cursor more efficiently. But you need to remember to use an EXIT statement break the loop when the cursor pointer reaches the end. The script below gives you a good example:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
  CURSOR emp_cur IS SELECT * FROM employees
    WHERE manager_id = 101;
  emp_rec employees%ROWTYPE;
BEGIN
  OPEN emp_cur;
  LOOP
    FETCH emp_cur INTO emp_rec;
    EXIT WHEN emp_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Name = ' ||
      emp_rec.first_name || ' ' || emp_rec.last_name);
  END LOOP;
  CLOSE emp_cur;
END;
```

```
/
Name = Nancy Greenberg
Name = Jennifer Whalen
Name = Susan Mavris
Name = Hermann Baer
Name = Shelley Higgins
```

[Read More Answers.](#)

Question # 386

How To Use an Explicit Cursor without OPEN Statements?

Answer:-

If you want to open a cursor and loop through its data rows in quick way, you can use the FOR ... IN ... LOOP statement in the same way as the implicit cursor. The following tutorial exercise gives you a good example:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
  CURSOR emp_cur IS SELECT * FROM employees
    WHERE manager_id = 101;
BEGIN
  FOR row IN emp_cur LOOP
    DBMS_OUTPUT.PUT_LINE('Name = ' ||
      row.first_name || ' ' || row.last_name);
  END LOOP;
END;
```

```
/
Name = Nancy Greenberg
Name = Jennifer Whalen
Name = Susan Mavris
Name = Hermann Baer
Name = Shelley Higgins
```

[Read More Answers.](#)

Question # 387

Can Multiple Cursors Being Opened at the Same Time?

Answer:-

Yes, multiple cursors can be opened at the same time. See the following example:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
  CURSOR emp_cur IS SELECT * FROM employees;
  emp_rec employees%ROWTYPE;
  CURSOR dpt_cur IS SELECT * FROM departments;
  dpt_rec departments%ROWTYPE;
BEGIN
  OPEN emp_cur;
  OPEN dpt_cur;
  FETCH emp_cur INTO emp_rec;
  FETCH dpt_cur INTO dpt_rec;
  DBMS_OUTPUT.PUT_LINE('Department name = ' ||
    dpt_rec.department_name);
  DBMS_OUTPUT.PUT_LINE('Employee name = ' ||
    emp_rec.first_name || ' ' || emp_rec.last_name);
  CLOSE emp_cur;
  CLOSE dpt_cur;
END;
```

```
/
Department name = Administration
Employee name = Steven King
```

[Read More Answers.](#)

Question # 388

How To Pass a Parameter to a Cursor in Oracle?



Answer:-

When you define a cursor, you can set a formal parameter in the cursor. The formal parameter will be replaced by an actual parameter in the OPEN cursor statement.

Here is a good example of a cursor with two parameters:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
CURSOR emp_cur(low NUMBER, high NUMBER)
IS SELECT * FROM employees WHERE salary >= low
AND salary <= high;
BEGIN
FOR row IN emp_cur(12000,15000) LOOP
DBMS_OUTPUT.PUT_LINE(row.first_name || ' '
|| row.last_name
|| ': ' || row.salary);
END LOOP;
END;
```

```
/
Nancy Greenberg: 12000
John Russell: 14000
Karen Partners: 13500
Alberto Errazuriz: 12000
Michael Hartstein: 13000
Shelley Higgins: 12000
```

[Read More Answers.](#)

Question # 389

What Is an Oracle Cursor Variable?

Answer:-

A cursor variable is a variable of a specific REF CURSOR data type, which is a pointer to a data structure resource connects to query statement result, similar to the CURSOR data type.. The advantage of using cursor variables is that cursor variables can be used as variables to pass between procedures and functions.

[Read More Answers.](#)

Question # 390

How To Define an Oracle Cursor Variable?

Answer:-

To define cursor variable, you must decide which REF CURSOR data type to use. There are 3 ways to select a REF CURSOR data type:

- * Define your own specific REF CURSOR types using the TYPE ... RETURN statement.
- * Define your own generic REF CURSOR type using the TYPE ... statement.
- * Use the system defined REF CURSOR type: SYS_REFCURSOR.

The following tutorial exercise defines 3 cursor variables in 3 different ways:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
TYPE emp_ref IS REF CURSOR RETURN employees%ROWTYPE;
TYPE any_ref IS REF CURSOR;
emp_cur emp_ref;
any_cur any_ref;
sys_cur SYS_REFCURSOR;
BEGIN
NULL;
END;
```

[Read More Answers.](#)

Question # 391

How To Open a Cursor Variable?

Answer:-

A cursor variable must be opened with a specific query statement before you can fetch data fields from its data rows. To open a cursor variable, you can use the OPEN ... FOR statement as shown in the following tutorial exercise:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
TYPE emp_ref IS REF CURSOR RETURN employees%ROWTYPE;
TYPE any_ref IS REF CURSOR;
emp_cur emp_ref;
any_cur any_ref;
sys_cur SYS_REFCURSOR;
BEGIN
OPEN emp_cur FOR SELECT * FROM employees;
OPEN any_cur FOR SELECT * FROM employees;
OPEN sys_cur FOR SELECT * FROM employees;
CLOSE sys_cur;
CLOSE any_cur;
CLOSE emp_cur;
END;
```

[Read More Answers.](#)

Question # 392

How To Loop through a Cursor Variable?



Answer:-

Once a cursor variable is opened with a query statement, it will have the same attributes as a normal cursor and it can be used in the same way a normal cursor too. The following sample script shows you how to loop through a cursor variable:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
TYPE emp_ref IS REF CURSOR RETURN employees%ROWTYPE;
emp_cur emp_ref;
emp_rec employees%ROWTYPE;
BEGIN
OPEN emp_cur FOR SELECT * FROM employees
WHERE manager_id = 101;
LOOP
FETCH emp_cur INTO emp_rec;
EXIT WHEN emp_cur%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Name = ' ||
emp_rec.first_name || ' ' || emp_rec.last_name);
END LOOP;
CLOSE emp_cur;
END;
/
Name = Nancy Greenberg
Name = Jennifer Whalen
Name = Susan Mavris
Name = Hermann Baer
Name = Shelley Higgins
```

[Read More Answers.](#)

Question # 393

How To Pass a Cursor Variable to a Procedure?

Answer:-

A cursor variable can be passed into a procedure like a normal variable. The sample script below gives you a good example:

```
CREATE OR REPLACE PROCEDURE ggl_CENTER AS
sys_cur SYS_REFCURSOR;
PROCEDURE emp_print(cur SYS_REFCURSOR) AS
emp_rec employees%ROWTYPE;
BEGIN
LOOP
FETCH cur INTO emp_rec;
EXIT WHEN cur%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Name = ' ||
emp_rec.first_name || ' ' || emp_rec.last_name);
END LOOP;
END;
BEGIN
OPEN sys_cur FOR SELECT * FROM employees
WHERE manager_id = 101;
emp_print(sys_cur);
CLOSE sys_cur;
END;
/
Name = Nancy Greenberg
Name = Jennifer Whalen
Name = Susan Mavris
Name = Hermann Baer
Name = Shelley Higgins
```

[Read More Answers.](#)

Question # 394

Why Cursor Variables Are Easier to Use than Cursors?

Answer:-

Cursor variables are easier to use than cursors because:

- * Cursor variables are easier to define. No need to give a specific query statement.
- * Cursor variables are easier to open. You can specify the query dynamically at the time of open.
- * Cursor variables can be passed into procedures or functions.

[Read More Answers.](#)

Question # 395

What Is the Simplest Tool to Run Commands on Oracle Servers?

Answer:-

The simplest tool to connect to an Oracle server and run commands to manage data is SQL*Plus. It is an Oracle database client tool that works as a command-line user interface to the database server. SQL*Plus allows you:

- * Format, perform calculations on, store, and print from query results.
- * Examine table and object definitions.
- * Develop and run batch scripts.
- * Perform database administration.

[Read More Answers.](#)



Question # 396

What Is the Quickest Way to Export a Table to a Flat File?

Answer:-

The quickest way to export a table to a flat file is probably to use the SQL*Plus SPOOL command. It allows you to record SELECT query result to a text file on the operating system. The following tutorial exercise shows you how control the output format, start the spooler, and dump all record from a table to a flat text file:

```
>mkdir oraclexe est
>sqlplus /nolog
SQL> connect HR/globalguideline
SQL> SET HEADING OFF;
SQL> SET FEEDBACK OFF;
SQL> SET LINESIZE 1000;
SQL> SPOOL oraclexe estemployees.txt;
SQL> SELECT * FROM EMPLOYEES;
.....
SQL> SPOOL OFF;
You should get all records in employees.txt with fixed length fields.
```

[Read More Answers.](#)

Question # 397

How To Export Data with a Field Delimiter?

Answer:-

The previous exercise allows you to export data with fixed field lengths. If you want export data with variable field lengths and field delimiters, you can concatenate your fields with an expression in the SELECT clause as shown in the tutorial exercise below:

```
SQL> SET HEADING OFF;
SQL> SET FEEDBACK OFF;
SQL> SET LINESIZE 1000;
SQL> SPOOL oraclexe estggl_links.txt;
SQL> SELECT id ||','|| url ||','|| notes ||','|| counts
||','|| created FROM ggl_links;
.....
SQL> SPOOL OFF;
You should see all records in ggl_links.txt with ',' delimited fields as shown here:
101,globalguideline.com,Session 1,,17-MAY-06
110,globalguideline.com,Session 1,,17-MAY-06
```

[Read More Answers.](#)

Question # 398

What Is SQL*Loader?

Answer:-

SQL*Loader is a database tool that allows to load data from external files into database tables. SQL*Loader is available as part of the free Oracle 10g Expression Edition. It has some interesting features as:

- * Can load data from multiple data files into multiple tables in one load session.
- * Can specify character set of the data.
- * Can generate sophisticated error reports.
- * Can load data directly to Oracle datafiles, bypassing normal record inserting process.

[Read More Answers.](#)

Question # 399

What Is a SQL*Loader Control File?

Answer:-

A SQL*Loader control file a text that defines how data files should be loaded into the database. It allows you to specify:

- * Where is the input data file.
- * The format of the input data file.
- * The target table where the data should be loaded.
- * How input data fields should be mapped to target table columns.
- * Select criteria to select input records for loading.
- * Where to output errors.

[Read More Answers.](#)

Question # 400

How To Load Data with SQL*Loader?

Answer:-

Let's say you have a table defined as:

```
CREATE TABLE student (jd NUMBER(5) PRIMARY KEY,
first_name VARCHAR(80) NOT NULL,
last_name VARCHAR(80) NOT NULL,
birth_date DATE NOT NULL,
social_number VARCHAR(80) UNIQUE NOT NULL);
```

There is an input data file stored at oraclexe extstudent.txt with the following records:

```
1,Steven,King,17-JUN-77,515.123.4567
2,Neena,Kochhar,21-SEP-79,515.123.4568
3,Lex,De Haan,13-JAN-83,515.123.4569
```



4,Alexander,Hunold,03-JAN-80,590.423.4567
5,Bruce,Ernst,21-MAY-81,590.423.4568
6,David,Austin,25-JUN-87,590.423.4569
7,Valli,Pataballa,05-FEB-88,590.423.4560
8,Diana,Lorentz,07-FEB-89,590.423.5567
9,Nancy,Greenberg,17-AUG-84,515.124.4569
10,Daniel,Faviet,16-AUG-84,515.124.4169

[Read More Answers.](#)

Question # 401

What Is an External Table?

Answer:-

An external table is a table defined in the database with data stored outside the database. Data of an external table is stored in files on the operating systems. Accessing data of external tables are done through data access drivers. Currently, Oracle supports two data access drivers: ORACLE_LOADER and ORACLE_DATAPUMP.

External tables can be used to load data from external files into database, or unload data from database to external files.

[Read More Answers.](#)

Question # 402

How To Load Data through External Tables?

Answer:-

If you have data stored in external files, you can load it to database through an external table by the steps below:

- * Create an external table with columns matching data fields in the external file.
- * Create a regular table with the same columns.
- * Run an INSERT INTO ... SELECT statement to load data from the external file to the regular table through the external table.

[Read More Answers.](#)

Question # 403

What Are the Restrictions on External Table Columns?

Answer:-

When creating external table columns, you need to watch out some restrictions:

- * "PRIMARY KEY" is not allowed.
- * "NOT NULL" is not allowed.
- * "DEFAULT value" is not allowed.

[Read More Answers.](#)

Question # 404

What Is a Directory Object?

Answer:-

A directory object is a logical alias for a physical directory path name on the operating system. Directory objects can be created, dropped, and granted access permissions to different users. The following tutorial exercise shows you some good examples:

```
>sqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> CREATE DIRECTORY test_dir AS '/oracle/test';
Directory created.
SQL> GRANT READ ON DIRECTORY test_dir TO hr;
Grant succeeded.
SQL> GRANT WRITE ON DIRECTORY test_dir TO hr;
Grant succeeded.
SQL> CREATE DIRECTORY temp_dir AS '/oracle/temp';
Directory created.
SQL> DROP DIRECTORY temp_dir;
Directory dropped.
```

[Read More Answers.](#)

Question # 405

How To Define an External Table with a Text File?

Answer:-

You can use the CREATE TABLE statement to create external tables. But you need to use ORGANIZATION EXTERNAL clause to specify the external file location and the data access driver. The tutorial exercise below shows you how to define an external table as a text file:

```
>sqlplus /nolog
SQL> connect HR/globalguideline
SQL> CREATE TABLE ext_ggl_links (
id NUMBER(4),
url VARCHAR2(16),
notes VARCHAR2(16),
counts NUMBER(4),
created DATE
) ORGANIZATION EXTERNAL (
TYPE ORACLE_LOADER
DEFAULT DIRECTORY test_dir
```




```
LOCATION ('ext_ggl_links.txt')
);
Table created.
SQL> SELECT table_name, tablespace_name, num_rows
FROM USER_TABLES;


```
TABLE_NAME TABLESPACE_NAME NUM_ROWS

REGIONS USERS 4
LOCATIONS USERS 23
DEPARTMENTS USERS 27
JOBS USERS 19
EMPLOYEES USERS 107
JOB_HISTORY USERS 10
ggl_LINKS USERS 2
EXT_ggl_LINKS
COUNTRIES 25</pre>
```


```

[Read More Answers.](#)

Question # 406

How To Run Queries on External Tables?

Answer:-

If you have an external table defined as a text file with the ORACLE_LOADER driver, you can add data to the text file, and query the text file through the external table. By default, data fields in the text file should be terminated by ','. The tutorial exercise below shows you how add data to the external table defined in the previous exercise:

```
>edit /oracle/test/ext_ggl_links.txt
1101,www.globalguideline,Link #1,88,07-MAY-06
1110,www.globalguideline,Link #2,88,07-MAY-06
>sqlplus /nolog
SQL> connect HR/globalguideline
SQL> SELECT * FROM ext_ggl_links;


```
<pre> ID URL NOTES COUNTS CREATED

1101 www.globalguideline Link #1 88 07-MAY-06
1110 www.globalguideline Link #2 88 07-MAY-06</pre>
```


```

[Read More Answers.](#)

Question # 407

How To Load Data from External Tables to Regular Tables?

Answer:-

Once you have your data entered in a text file, and an external table defined to this text file, you can easily load data from this text file to a regular table. The following tutorial exercise shows you how to load data from the text file and the external table defined in the previous exercises to a regular table:

```
SQL> CREATE TABLE ggl_links (
id NUMBER(4) PRIMARY KEY,
url VARCHAR2(16) NOT NULL,
notes VARCHAR2(16),
counts NUMBER(4),
created DATE DEFAULT (sysdate)
);
SQL> INSERT INTO ggl_links SELECT * FROM ext_ggl_links;
2 rows created.
SQL> SELECT * FROM ggl_links;


```
<pre> ID URL NOTES COUNTS CREATED

101 globalguideline.com Session 1 07-MAY-06
110 centerggl.com Session 1 07-MAY-06
1101 www.globalguideline Link #1 88 07-MAY-06
1110 www.globalguideline Link #2 88 07-MAY-06</pre>
```


```

[Read More Answers.](#)

Question # 408

What Is the Data Pump Export Utility?

Answer:-

Oracle Data Pump Export utility is a standalone programs that allows you to export data objects from Oracle database to operating system files called dump file set, which can be imported back to Oracle database only by Oracle Data Pump Import utility.

The dump file set can be imported on the same system or it can be moved to another system and loaded there.

The dump file set is made up of one or more disk files that contain table data, database object metadata, and control information. The files are written in a proprietary, binary format. During an import operation, the Data Pump Import utility uses these files to locate each database object in the dump file set. Because the dump files are written by the server, rather than by the client, the data base administrator (DBA) must create directory objects.

[Read More Answers.](#)

Question # 409

What Is the Data Pump Import Utility?

Answer:-



Oracle Data Pump Import utility is a standalone programs that allows you to import data objects from an Oracle dump file set into Oracle database. Oracle dump file set is written in a proprietary binary format by the Data Pump Export utility.

Import can also be used to load a target database directly from a source database with no intervening dump files. This allows export and import operations to run concurrently, minimizing total elapsed time. This is known as a network import. Data Pump Import enables you to specify whether a job should move a subset of the data and metadata from the dump file set or the source database (in the case of a network import), as determined by the import mode. This is done using data filters and metadata filters, which are implemented through Import commands.

[Read More Answers.](#)

Question # 410

How To Invoke the Data Pump Export Utility?

Answer:-

The Data Pump Export utility is distributed as executable file called "expdp.exe". To invoke this utility, you should go to the "bin" directory of your Oracle server installation and run the "expdp" command. Here is tutorial exercise on how to invoke the export utility:

```
>cd oracle\expapp\oracleproduct10.2.0\serverBIN
>expdp help=y
```

Export: Release 10.2.0.1.0 -

The Data Pump export utility provides a mechanism for transferring data objects between Oracle databases. The utility is invoked with the following command:

Example: expdp scott/tiger DIRECTORY=dmpdir
DUMPFIL=scott.dmp

You can control how Export runs by entering the 'expdp' command followed by various parameters:

Format: expdp KEYWORD=value or
KEYWORD=(value1,value2,...,valueN)

Example: expdp scott/tiger DUMPFIL=scott.dmp
DIRECTORY=dmpdir
SCHEMAS=scott or TABLES=(T1:P1,T1:P2)

[Read More Answers.](#)

Question # 411

How To Invoke the Data Pump Import Utility?

Answer:-

The Data Pump Import utility is distributed as executable file called "impdp.exe". To invoke this utility, you should go to the "bin" directory of your Oracle server installation and run the "impdp" command. Here is tutorial exercise on how to invoke the import utility:

```
>cd oracle\expapp\oracleproduct10.2.0\serverBIN
>impdp help=y
```

Import: Release 10.2.0.1.0 -

The Data Pump Import utility provides a mechanism for transferring data objects between Oracle databases. The utility is invoked with the following command:

Example: impdp scott/tiger DIRECTORY=dmpdir
DUMPFIL=scott.dmp

You can control how Import runs by entering the 'impdp' command followed by various parameters.

Format: impdp KEYWORD=value or
KEYWORD=(value1,value2,...,valueN)

Example: impdp scott/tiger DIRECTORY=dmpdir
DUMPFIL=scott.dmp

[Read More Answers.](#)

Question # 412

What Are Data Pump Export and Import Modes?

Answer:-

Data pump export and import modes are used to determine the type and portions of database to be exported and imported. Oracle 10g supports 5 export and import modes:

* Full: Exports and imports a full database. Use the FULL parameter to specify this mode.

* Schema: Enables you to export and import all objects that belong to a schema. Use the SCHEMAS parameter to specify this mode. This is the default mode.

* Table: Enables you to export and import specific tables and partitions. Use the TABLES parameter to specify this mode.

* Tablespace: Enables a privileged user to move a set of tablespaces from one Oracle database to another. Use the TABLESPACES parameter to specify this mode.

* Tablespace: Enables a privileged user to move metadata from the tables within a specified set of tablespaces from one Oracle database to another. Use the TRANSPORT_TABLESPACES parameter to specify this mode.

[Read More Answers.](#)

Question # 413

How To Estimate Disk Space Needed for an Export Job?

Answer:-

If you just want to know how much disk space for the dump without actually exporting any data, you can use the ESTIMATE_ONLY=y parameter on the expdp command. The following tutorial exercise shows you how a system user wants to see the disk space estimates on a full database export:

```
>cd oracle\expapp\oracleproduct10.2.0\serverBIN
```

```
>expdp SYSTEM/globalguideline FULL=y ESTIMATE_ONLY=y
Starting "SYSTEM"."SYS_EXPORT_FULL_01": SYSTEM/**** FULL=y
ESTIMATE_ONLY=y
```



```
<pre>Estimate in progress using BLOCKS method...
Processing object DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA
. estimated "FLOWS_020100"."WWV_FLOW_PAGE_PLUGS" 42 MB
. estimated "FLOWS_020100"."WWV_FLOW_STEP_ITEMS" 27 MB
. estimated "FLOWS_020100"."WWV_FLOW_STEP_PROCESSING" 16 MB
.....
. estimated "SYSTEM"."SQLPLUS_PRODUCT_PROFILE" 0 KB
. estimated "TSMYSYS"."SR$$" 0 KB
Total estimation using BLOCKS method: 169.8 MB
Job "SYSTEM"."SYS_EXPORT_FULL_01" successfully completed</pre>
Now you know that you need 170 MB disk space to export the entire data base.
Oracle also records the screen output in a log file called export.log at oraclexeapporacleadminXEpdump.
```

[Read More Answers.](#)

Question # 414

How To Do a Full Database Export?

Answer:-

If you are ready to do a full database export, you can use the FULL=y parameter on the expdp command, as shown in the following tutorial exercise:

```
>expdp SYSTEM/globalguideline FULL=y ESTIMATE_ONLY=y
Starting "SYSTEM"."SYS_EXPORT_FULL_01": SYSTEM/**** FULL=y
Estimate in progress using BLOCKS method...
Processing object DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 169.8 MB
Processing object type DATABASE_EXPORT/TABLESPACE
Processing object type DATABASE_EXPORT/SYS_USER/USER
Processing object type DATABASE_EXPORT/SCHEMA/USER
.....<pre>
. . exported FLOWS_020100.WWV_FLOW_PAGE_PLUGS 32.51MB
. . exported FLOWS_020100.WWV_FLOW_STEP_ITEMS 21.68MB
. . exported FLOWS_020100.WWV_FLOW_STEP_PROCESSING 11.17MB</pre>
.....
Master table "SYSTEM"."SYS_EXPORT_FULL_01" unloaded
*****
Dump file set for SYSTEM.SYS_EXPORT_FULL_01 is:
C:\ORACLE\XEAPPORACLEADMINXEDPDUMPEXPDAT.DMP
Job "SYSTEM"."SYS_EXPORT_FULL_01" successfully completed
```

[Read More Answers.](#)

Question # 415

Where Is the Export Dump File Located?

Answer:-

If you are not specifying the dump directory and file name, the dump file will be stored in the default dump directory with the default file name. The tutorial exercise below tells you find what is your default dump directory and locate the dump file.

```
>sqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> COL owner FORMAT A8;
SQL> COL directory_name FORMAT A16;
SQL> COL directory_path FORMAT A40;
SQL> SELECT * FROM dba_directories;
<pre>OWNER DIRECTORY_NAME DIRECTORY_PATH
-----
SYS DATA_PUMP_DIR oraclexeapporacleadminXEpdump
SYS TEST_DIR /oraclexe/test
SYS ORACLECLDIR oraclexeapporacleadminXEpdump
server inclr</pre>
```

Obviously, the default dump directory is directory object defined to oraclexeapporacleadminXEpdump. If you go to that directory, you will find the full database dump file is called "expdat.dmp".

[Read More Answers.](#)

Question # 416

How To Export Your Own Schema?

Answer:-

If you have a non-system user account and you want to export all data objects in the schema associated with your account, you can use the "expdp" command with the SCHEMAS parameter. Running "expdp" command with a non-system user account requires a directory object granted to this user account. The following tutorial exercise shows you how to define a directory object and export a schema:

```
>mkdir oraclexe_hr_dump
>cd oraclexeapporacleadminXEpdump
>sqlplus /nolog
SQL> connect SYSTEM/globalguideline
SQL> CREATE DIRECTORY hr_dump AS 'oraclexe_hr_dump';
Directory created.
SQL> GRANT READ ON DIRECTORY hr_dump TO hr;
Grant succeeded.
SQL> GRANT WRITE ON DIRECTORY hr_dump TO hr;
Grant succeeded.
SQL> quit
```



```
>expdp hr/globalguideline SCHEMAS=hr
  DIRECTORY=hr_dump DUMPFILE=schema.dmp LOGFILE=schema.log
```

[Read More Answers.](#)

Question # 417

How To Export Several Tables Together?

Answer:-

If you don't want to export the entire schema and only want to export several tables only, you can use the "expdp" command with the "TABLES" parameter as shown in the following tutorial exercise:

```
>cd oraclexeapporacleproduct10.2.0serverBIN
>expdp hr/globalguideline TABLES=employees,departments
  DIRECTORY=hr_dump DUMPFILE=tables.dmp LOGFILE=tables.log
Starting "HR"."SYS_EXPORT_TABLE_01": hr/*****
  TABLES=employees,departments DIRECTORY=hr_dump
  DUMPFILE=tables.dmp
  LOGFILE=tables.log
```

Estimate in progress using BLOCKS method...

Processing object type TABLE_EXPORT/TABLE/TABLE_DATA

Total estimation using BLOCKS method: 128 KB

Processing object type TABLE_EXPORT/TABLE/TABLE

Processing object type TABLE_EXPORT/TABLE/INDEX/INDEX

Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CON...

Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTI...

Processing object type TABLE_EXPORT/TABLE/COMMENT

Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/REF...

Processing object type TABLE_EXPORT/TABLE/TRIGGER

Processing object type TABLE_EXPORT/TABLE/STATISTICS/TAB...

. . exported "HR"."DEPARTMENTS" 6.632 KB 27 rows

. . exported "HR"."EMPLOYEES" 15.76 KB 107 rows

Master table "HR".

[Read More Answers.](#)

Question # 418

What Happens If the Imported Table Already Exists?

Answer:-

If the import process tries to import a table that already exists, the Data Pump Import utility will return an error and skip this table. The following exercise shows you a good example:

```
>cd oraclexeapporacleproduct10.2.0serverBIN
>impdp hr/globalguideline TABLES=employees
  DIRECTORY=hr_dump DUMPFILE=tables.dmp LOGFILE=tables.log
Master table "HR"."SYS_IMPORT_TABLE_01" loaded/unloaded
Starting "HR"."SYS_IMPORT_TABLE_01": hr/** TABLES=employees
  DIRECTORY=hr_dump DUMPFILE=tables.dmp LOGFILE=tables.log
Processing object type TABLE_EXPORT/TABLE/TABLE
ORA-39151: Table "HR"."EMPLOYEES" exists. All dependent
  metadata and data will be skipped due to table_exists_action
  of skip
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
Processing object type TABLE_EXPORT/TABLE/INDEX/INDEX
.....
```

[Read More Answers.](#)

Question # 419

How To Import One Table Back from a Dump File?

Answer:-

If you only want to import one table back to the database, you can use a dump file that was created by full export, schema export or a table export. The following tutorial exercise shows you how to import the "ggl_links" table from a dump file created by a schema export:

```
>cd oraclexeapporacleproduct10.2.0serverBIN
>sqlplus /nolog
SQL> connect HR/globalguideline
SQL> DROP TABLE ggl_links;
Table dropped.
SQL> exit;
>impdp hr/globalguideline TABLES=ggl_links DIRECTORY=hr_dump
  DUMPFILE=schema.dmp LOGFILE=tables.log
Master table "HR"."SYS_IMPORT_TABLE_01" loaded/unloaded
Starting "HR"."SYS_IMPORT_TABLE_01": hr/** TABLES=ggl_links
  DIRECTORY=hr_dump DUMPFILE=schema.dmp LOGFILE=tables.log
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "HR"."GGL_LINKS" 6.375 KB 4 rows
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CON...
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTI...
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TAB...
Job "HR"."SYS_IMPORT_TABLE_01" successfully completed.
```



[Read More Answers.](#)

Question # 420

What Are the Original Export and Import Utilities?

Answer:-

Oracle original Export and Import utilities are standalone programs that provide you a simple way for you to transfer data objects between Oracle databases, even if they reside on platforms with different hardware and software configurations.

When you run Export against an Oracle database, objects (such as tables) are extracted, followed by their related objects (such as indexes, comments, and grants), if any. The extracted data is written to an export dump file. The Import utility reads the object definitions and table data from the dump file.

An export file is an Oracle binary-format dump file that is typically located on disk or tape. The dump files can be transferred using FTP or physically transported (in the case of tape) to a different site. The files can then be used with the Import utility to transfer data between databases that are on systems not connected through a network. The files can also be used as backups in addition to normal backup procedures.

Export and Import utilities are now being replaced by Data Pump Export and Import utilities in Oracle 10g. But you can still use them.

[Read More Answers.](#)

Question # 421

How To Invoke the Original Export Import Utilities?

Answer:-

If you really want to run the original export import utilities, you can still go to "bin" directory of the Oracle server path and run the "exp" or "imp" command. The tutorial exercise below tells you how to run the export and import utilities in help modes:

```
>cd oracle\exp\oracleproduct10.2.0\server\BIN
```

```
>exp help=y
```

You can let Export prompt you for parameters by entering the EXP command followed by your username/password:

Example: EXP SCOTT/TIGER

Or, you can control how Export runs by entering the EXP command followed by various arguments. To specify parameters, you use:

Format: EXP KEYWORD=value or

KEYWORD=(value1,value2,...,valueN)

Example: EXP SCOTT/TIGER GRANTS=Y TABLES=(EMP,DEPT,MGR)

or TABLES=(T1:P1,T1:P2), if T1 is partitioned table

```
.....
```

```
>imp help=y
```

```
.....
```

[Read More Answers.](#)

Question # 422

What Is Oracle Open Database Communication (ODBC)?

Answer:-

ODBC, Open Database Communication, a standard API (application program interface) developed by Microsoft for Windows applications to communicate with database management systems.

Oracle offers ODBC drivers to allow Windows applications to connect Oracle server through ODBC.

[Read More Answers.](#)

Question # 423

How To Install Oracle ODBC Drivers?

Answer:-

Oracle offers different ODBC drivers for different versions of Oracle servers. When you install an Oracle server or a client tools on your Windows system, the appropriate ODBC driver will be installed for you automatically.

If you want to install a specific ODBC driver, you need to go to <http://www.oracle.com/technology/software/tech/windows/odbc/index.html>. Download the right ODBC driver and follow the instructions to install it.

[Read More Answers.](#)

Question # 424

How To Find Out What Oracle ODBC Drivers Are Installed?

Answer:-

To find out what Oracle ODBC drivers are installed on your Windows system, you can use the ODBC manager to look at them:

* Go to Control Panel.

* Go to Administrative Tools.

* Run Data Sources (ODBC).

* Go to System DSN tab.

* Click the Add button.

You should a list of all ODBC drivers installed on your system. Oracle ODBC drivers should be on the list. For example, your installed ODBC driver list could look like the one the picture below with two Oracle ODBC drivers, "Oracle in OraClient10g_home1" and "Oracle in XE":

Oracle ODBC Drivers

[Read More Answers.](#)

Question # 425



How Can Windows Applications Connect to Oracle Servers?

Answer:-

A Windows application can connect to an Oracle server directly, if it knows how to use the Oracle TNS technology.

A Windows application can connect to an Oracle server indirectly through Windows ODBC manager, be caused offers ODBC drivers to support the ODBC API. The diagram below shows how MS Access can connect to an Oracle server through the ODBC driver:

Oracle ODBC Connection

[Read More Answers.](#)

Question # 426

How To Create Tables for ODBC Connection Testing?

Answer:-

If you want to follow the tutorial exercises in the sections below, you need to create a user account and a table for ODBC connection testing as shown here:

```
SQL> CONNECT system/retneclgg
Connected.
SQL> CREATE USER ggl IDENTIFIED BY retneclgg ACCOUNT UNLOCK;
User created.
SQL> GRANT CREATE SESSION TO ggl;
Grant succeeded.
SQL> GRANT CREATE TABLE TO ggl;
Grant succeeded.
SQL> ALTER USER ggl DEFAULT TABLESPACE USERS;
User altered.
SQL> ALTER USER dev QUOTA 4M ON USERS;
User altered.
SQL> connect ggl/retneclgg;
Connected.
SQL> CREATE TABLE dev_faq (id NUMBER);
SQL> INSERT INTO dev_faq VALUES (3);
SQL> INSERT INTO dev_faq VALUES (5);
SQL> INSERT INTO dev_faq VALUES (7);
```

[Read More Answers.](#)

Question # 427

How To Check the Oracle TNS Settings?

Answer:-

If you have installed an Oracle server or an Oracle client tool on your local system, the TNS is automatically installed with a simple configuration file, tnsnames.ora, to define Oracle connect identifiers.

For example, if you have Oracle XE server installed, you will have the tnsnames.ora located at oraclexeapporacleproduct10.2.0serverNETWORKADMIN. It should contain a connect identifier called XE defined as:

```
XE =
(DESCRIPTION =
  (ADDRESS =
    (PROTOCOL = TCP)(HOST = localhost)
    (PORT = 1521)
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)
```

[Read More Answers.](#)

Question # 428

How To Define a Data Source Name (DSN) in ODBC Manager?

Answer:-

DSN (Data Source Name) is an ODBC connection identifier for Windows applications. Here is how you can define a DSN on your Windows system:

- * Go to Control Panel.
- * Go to Administrative Tools.
- * Run Data Sources (ODBC).
- * Go to System DSN tab.
- * Click the Add button.
- * Select the "Oracle in XE" driver.
- * Enter Data Source Name: ggl_DSN.
- * Enter Description: globalguideline DSN Oracle Setting.
- * Enter TNS Service Name: XE.
- * Click the Test Connection button.
- * Enter User Name: ggl.
- * Enter Password: retneclgg.
- * Click the OK button.

You should see a "Connection successful" as shown in the following picture:

DSN Setting for Oracle ODBC Driver

[Read More Answers.](#)

Question # 429



How To Connect MS Access to Oracle Servers?

Answer:-

Once you got a DSN defined in the ODBC manager that connects to an Oracle server, you can connect a normal MS Access document to the Oracle server, and link an Access table to Oracle table. The tutorial below gives you a good example:

- * Start MS Access with a new database file.
- * Go to File menu.
- * Select Get External Data.
- * Select Import.... The Import dialog box shows up.
- * Select Files of type: ODBC Database(). The Select Data Source dialog box shows up.
- * Click the Machine Data Source tab. You should see the DSN name "ggl_DSN" you defined earlier.
- * Select "ggl_DSN".
- * Enter User Name: ggl.
- * Enter Password: retneclgg.

You should see the Oracle ODBC Driver Connect dialog box as shown in the picture below:

Import Oracle tables to MS Access

Click the OK button to continue. You should see a list of tables available for you to import from the Oracle server as shown in the picture below:

Importing Oracle tables via ODBC

[Read More Answers.](#)

Question # 430

How To Connect ASP Pages to Oracle Servers?

Answer:-

If you are running Windows IIS Web server and serving ASP Web pages, you can get data from Oracle servers into your ASP pages through ODBC drivers. To do this, you need to install the correct Oracle ODBC driver and define a DSN on the IIS Web server.

Then you can use ADODB objects to connect to the Oracle server over the ODBC driver in your ASP pages. The tutorial example below gives you a good example:

```
<pre><%
Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open "DSN=ggl_DSN;UID=ggl;PWD=retneclgg"
Set oRS = oConn.Execute("SELECT * FROM dev_faq")
Response.write("<p>Data from Oracle server via ODBC:")
Response.write("")
Do While NOT oRS.EOF
    Response.Write(oRS("ID") & vbcrLf)
    oRS.MoveNext
Loop
Response.write("")
oRS.close
oConn.close
%></pre>
```

[Read More Answers.](#)

Oracle Most Popular Interview Topics.

- 1 : [Oracle PL-SQL Frequently Asked Interview Questions and Answers Guide.](#)
- 2 : [Oracle DBA Frequently Asked Interview Questions and Answers Guide.](#)
- 3 : [Oracle GL Frequently Asked Interview Questions and Answers Guide.](#)
- 4 : [Oracle Scenarios Frequently Asked Interview Questions and Answers Guide.](#)
- 5 : [Oracle ETL Frequently Asked Interview Questions and Answers Guide.](#)
- 6 : [Oracle D2K Frequently Asked Interview Questions and Answers Guide.](#)
- 7 : [Oracle AR Frequently Asked Interview Questions and Answers Guide.](#)
- 8 : [Database Architecture Frequently Asked Interview Questions and Answers Guide.](#)
- 9 : [Oracle Developer Frequently Asked Interview Questions and Answers Guide.](#)
- 10 : [Memory Management Frequently Asked Interview Questions and Answers Guide.](#)

About Global Guideline.

Global Guideline is a platform to develop your own skills with thousands of job interview questions and web tutorials for fresher's and experienced candidates. These interview questions and web tutorials will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts. Global Guideline invite you to unlock your potentials with thousands of [Interview Questions with Answers](#) and much more. Learn the most common technologies at Global Guideline. We will help you to explore the resources of the World Wide Web and develop your own skills from the basics to the advanced. Here you will learn anything quite easily and you will really enjoy while learning. Global Guideline will help you to become a professional and Expert, well prepared for the future.

* This PDF was generated from <https://GlobalGuideline.com> at **November 29th, 2023**

* If any answer or question is incorrect or inappropriate or you have correct answer or you found any problem in this document then don't hesitate feel free and [e-mail us](#) we will fix it.

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers

Follow us on Twitter for latest Jobs and interview preparation guides
<https://twitter.com/InterviewGuide>

Best Of Luck.

Global Guideline Team
<https://GlobalGuideline.com>
Info@globalguideline.com