

1. What is the difference between the “WHERE” clause and the “HAVING” clause?

Where clause is used to filters the data from grouped records. It is used to provide the conditions.

Having clause is used to provide conditions and filter the data from grouped data based on conditions.

2. What is the difference between “Primary Key” and “foreign Key”?

The relation between primary key and foreign key like as mother and child relation .

Primary key allows only unique values and doesn't allow duplicate values. It acts like a mother

Foreign key is taken references on primary key table. It acts like a child

3. What is the difference between primary key and unique constraints?

Primary key is used to allow only unique values it not allowed duplicate records and also null values.

Unique key is similar to primary key but it allows only one null value.

4. What is database normalization?      Normalization :- Normalization is the process to eliminate data redundancy and enhance data integrity in the table. Normalization also helps to organize the data in the database. It is a multi-step process that sets the data into tabular form and removes the duplicated data from the relational tables.

The types of normalization:-

First normal form (1 NF)

Second normal form (2 NF)

Third normal form (3 NF)

Boyce code normal form or fourth normal form (BCNF or 4 NF)

Fifth normal form (5 NF)

Sixth normal form (6 NF)

## 5. What are the differences between DDL, DML and DCL in SQL?

DDL means data definition language this are auto commit commands .This are used to database objects. This are session independent .The DDL commands are create, alter, drop, rename, flashback and purge.

DML means data manipulating language. It is used to handle the data in database object. These are non-auto commit commands. These are session dependent. The DML commands are insert, update, delete.

DCL means data control language. The DCL is used to provide the access to the user .The DCL commands are

GRANTS: - it is used to give permission to the user.

REVOKE: - it is used to cancel the permission to the user.

## 6. What is a view in SQL? How to create one?

These are advanced of synonyms. The virtual table to hide the base table and its work like a mirror image. It's doesn't have own structure .all views are stored in all views. There are several types of views in SQL.

They are 1.simple views, 2.complex views, 3.force views, 4.horizontal views, 5.vertical views, 6.functional views, 7. Materialized views, 8.partition views and 9.Inline views.

Create view on table

Syntax: - create view view name as select \* from table name;

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the following SQL statements:

```

create view V_1 as select * from emp;
select * from v_1;

```

Annotations with red arrows point to specific parts of the code:

- An arrow points to the identifier `V_1` with the label "view name".
- An arrow points to the identifier `emp` with the label "table name".
- An arrow points to the word `select` in the second statement with the label "Execute".
- An arrow points to the tab labeled "Query Result 2" with the label "Result".

In the bottom-right pane, the "Query Result 2" tab displays the output of the query `select * from v_1;`. The results are presented in a grid table:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
5	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
6	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
7	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
8	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
9	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
10	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30

## 7. What are the uses of view?

### The uses of views

To join tables and simplify multiple tables into one table.

It like mirror images of base table.

It is used to data security purposes.

## 8. What is a transaction

Transactions:- A transactions has only two results : success or failure. A transaction work on DML commands and not work on DDL commands. The transaction includes commit, rollback, save point and truncate.

## 9. What are indexes?

The indexes are simple defined as to receive data from data base quickly.

To increase performance while retrieve the data from the data base.

It will make use of user id's. All indexes are stored in all\_indexes.

There are several types of indexes are in SQL. They are 1.simple index, 2.complex index, 3.unique index, 4.functional indexes, and 5.bitmap indexes.

## 10.What is a Table in a database?

All data stored in table structure. In RDBMS all data stored in table structure.

It is easy to analyse data and filter the data. The data is logically organised by rows and columns.

## 11.What is database lock and what are the different types of lock?

Database locks: - Database lock basically signifies the transaction about the current status of the data item i.e. whether that data is being used by other transactions or not at that point of time.

There are two types of lock are in database they are Shared lock and Exclusive lock.

## 12.What are the different types of normalization?

The types of normalization:-

First normal form (1 NF)

Second normal form (2 NF)

Third normal form (3 NF)

Boyce code normal form or fourth normal form (BCNF or 4 NF)

Fifth normal form (5 NF)

Sixth normal form (6 NF)

13.What is a materialized view?

The materialized\_view different compare to all views. All views are don't have their own structure but the materialized view have own structure.

It doesn't allow the DML operations on views. It is used to store historical data.

Syntax: - create materialized view view name as select \* from table name;

14.Explain the difference between DELETE, TRUNCATE commands?

DELETE	TRUNCATE
It is used to delete the data in table.	It is used to delete the data from the database. It work like a delete + truncate
In case we delete the data from table before we commit we can roll back again.	It is permanent deleted.
Syntax:-delete from table name where condition;	Syntax:-truncate table table name;

## 15.What is Union, minus and Intersect commands?

Union:-union is used to removes the same values in both tables.

It doesn't display the duplicate values.

Syntax: - select \* from table 1 union select \* from table2;

Minus:- It displays the first query records, which are not found in the second query records.

Syntax:-select \* from table1 minus select \* from table2;

Intersect: - It displays the common values from two tables.

Syntax: - select \* from table1 intersect select \* from table 2;

## 16.What are constraints? Explain different types of constraints

Constraints: - The constraints are rules which are used to allow the valid data.

There are several types of constraints, they are

1. Primary key
2. Composite primary key
3. Unique
4. Not null
5. Default
6. Foreign key / Reference key.

1. Primary key: - The primary key doesn't allow the duplicate records and null values.

Syntax: - create table tablename (slno number (7) primary key);

2. Composite primary key: - The composite primary key is used to create primary key on multiple columns.

Syntax: - create table tablename (slno number (5), Ename varchar2 (10) primary key (slno, Ename) ;

3. Unique: - unique allows only unique values it doesn't allow duplicate records but in unique only one null values will be allowed.

Syntax: - create table tablename (slno number (10) unique);

4. Not null: - It allows only not null values and it doesn't allows null values.

Syntax: - create table tablename (slno number (5) not null);

5. Check:-It is used to check the condition.

Syntax: - create table tablename (slno number (5), check (slno>0) ;

6. Default: - It is used to insert default values.

Syntax: - create table tablename (slno number (5), grade char (2) default 'A');

7. Foreign Key: - It is used to maintain a reference from one table to another table.

Syntax: - create table table1 (slno number (5) primary key) Create table table2 (dno number (5), dname varchar2 (10), sno number (5) references table1 (sno);

17.What is a join and explain different types of joins with an example?

Joins: - The joins are used to join the tables. The joins are divided into some types. They are 1.simple join, 2.self-join, 3.outer join

1). Simple Join:-

a) Equi Join it is used to join two tables based on equal condition.

Syntax: - select \* from EMP, dept where emp.deptno=dept.deptno;

b) Non Equi Join it is used to join two tables based on not equal condition

Syntax: - select \* from EMP, dept where emp.deptno! = dept.deptno;

2) Self join: - It is used to join the table itself.

Syntax: select \* from EMP e1, EMP e2 where e1.deptno=e2.deptno;

3) Outer Join

a). Left Outer Join:- It is used to display the full details of the left table and matched records of the right table.

Syntax: select \* from EMP e, dept d where emp.deptno = dept.deptno (+); b)

b). Right Outer Join: - It is used to display the full details of the right table and matched records of the left table.

Syntax: select \* from EMP e, dept d where emp.deptno (+) =dept.deptn;

c).Full Outer Join: - If you join left and right outer joins with union operators such joins are called full outer join.

Syntax: select \* from EMP e, dept d where emp.deptno (+) = dept.deptno

Union

Select \* from EMP e, dept d where emp.deptno  
(+) = dept.deptno.

#### 18.What is alias of a table?

Alias of the table: - In SQL the alias are used to give temporary name for a table, table in the column when the query is lengthy we place a name for that query in short word or single letters. It reduces time also.

#### 19.Write all ways of create table and Insert data into a table options.

##### a. Create regular syntax

Create table student\_1 (idno number (5), name varchar2 (10),  
marks number (5)),

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL command:

```
create table student_1 (idno number (5),name varchar2(10),marks number(5));
```

A red arrow points from the word "EXECUTE" at the end of the command to the "Execute" button in the toolbar below the code editor. The bottom-left pane shows the "Script Output" tab with the following error message:

```
Error starting at line : 465 in command -  
create table task (slno number (8),name  varchar2(16),phno number (16))  
Error report -  
ORA-00955: name is already used by an existing object  
00955. 00000 - "name is already used by an existing object"  
*Cause:  
*Action:
```

Below the error message, the text "Table STUDENT\_1 created." is displayed, preceded by a red arrow pointing to it. To the right of this text, the word "RESULT" is written in red capital letters.

b. Create table using select method

Create table student\_2 as select \* from student\_1;

The screenshot shows the Oracle SQL Developer interface. In the top query editor window, the following SQL code is written:

```
create table student_1 (idno number (5),name varchar2(10),marks number(5));
select * from student_1;

create table student_2 as select * from student_1; ← EXECUTE
```

A red arrow points to the word "EXECUTE" in the command line. Below the code, the "Script Output" tab is active, showing the results of the execution:

Task completed in 0.371 seconds

Error report -  
ORA-00955: name is already used by an existing object  
00955. 00000 - "name is already used by an existing object"  
\*Cause:  
\*Action:

Table STUDENT\_1 created.

Table STUDENT\_2 created. ← RESULT

c. Insert regular syntax

Insert into student\_1 values ('111','kushal','65');

Insert into student\_1 values ('112','rakesh','75');

Insert into student\_1 values ('113','ramesh','95');

The screenshot shows the Oracle SQL Developer interface. In the top window, a query is being run:

```
select * from student_2;  
  
insert into student_1 values ('111','kushal','65');  
insert into student_1 values ('112','rakesh','75');  
insert into student_1 values ('113','ramesh','95');
```

A red arrow points to the word "EXECUTE" in the toolbar below the query editor.

In the bottom window, the results of the execution are displayed:

Table STUDENT\_2 created.

1 row inserted.

1 row inserted.

1 row inserted.

A red arrow points to the word "RESULT" in the toolbar below the results window.

The status bar at the bottom shows the file path: C:\DUMMY.sql.

d. Insert table using select method

Insert into student\_2 (idno, name, marks)

Select idno, name, marks

from student\_1;

The screenshot shows a SQL development environment with a query editor and a results viewer.

In the query editor, the following PL/SQL block is written:

```
insert into student_2 (idno,name,marks)
select idno, name, marks
from student_1;
```

A red arrow points from the word "EXECUTE" at the end of the block to the "RESULT" section below.

In the results viewer, the output is:

```
Script Output x | Query Result x | Query Result 1 x
✖️ ↗️ 📁 | Task completed in 0.075 seconds
select id, name, marks
from student_1
Error at Command Line : 492 Column : 8
Error report -
SQL Error: ORA-00904: "ID": invalid identifier
00904. 00000 -  "%s: invalid identifier"
*Cause:
*Action:

3 rows inserted.
```

A red arrow points from the text "3 rows inserted." to the "RESULT" section below.

20. Display all the information of the EMP table?

Select \* from emp;

The screenshot shows the SQL Developer interface. In the top-left pane, there is a code editor containing the SQL command: `select * from emp;`. A red arrow points from the word "EXECUTE" to the right side of the editor. In the bottom-right pane, the results of the query are displayed in a grid. The grid has 14 rows and 10 columns, labeled as follows:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30	
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10	
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20	
5	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	
6	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20	
7	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20	
8	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30	
9	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30	
10	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30	
11	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30	
12	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20	
13	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30	
14	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10	

21. write different types of steps to import data into a table?

1. SQL developer method.

SQL developer method

Steps to import data from source data

Click on schema → click on table name → right click → import → browse file → select source file → check the data and next → then select or matched the columns check data → finish.

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, multiple SQL files are listed: Welcome Page, SYSTEM.sql, Dbms Output, prac.sql, emp.sql, KING, IND, Team1.sql, and Relational\_1 (Untitled\_1). The main workspace shows a query builder with the following SQL code:

```
insert into mediatype values(2,'Protected AAC audio file');
insert into mediatype values(3,'Protected MPEG-4 video file');
insert into mediatype values(4,'Purchased AAC audio file');
insert into mediatype values(5,'AAC audio file');

select * from artist;
```

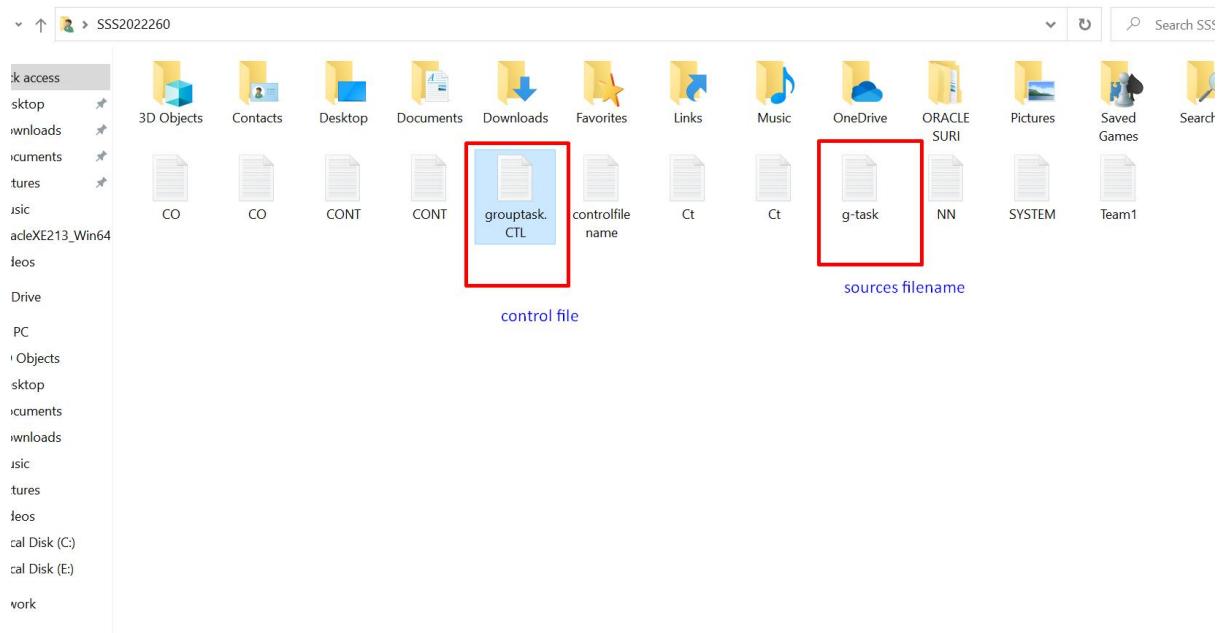
An 'Import Data' dialog box is open, displaying the message: "Import Data. Task successful and import committed." with an "OK" button.

## 2. SQL star loader method.

### SQL star loader method

Steps followed by to insert data in sql star loader method

Create a files in admin path 1 is control file 2 is sources file



### Control file

The screenshot shows a Windows desktop environment. In the foreground, a Notepad window is open with the title "grouptask.CTL - Notepad". The content of the Notepad is as follows:

```
File Edit Format View Help
OPTIONS (SKIP=1)
LOAD DATA
INFILE 'g-task.TXT'
DISCARDFILE 'DICARD.TXT'
TRUNCATE INTO TABLE track1
FIELDS TERMINATED BY ','
(
TRACKID,
NAME,
ALBUMID,
MEDIATYPEID,
GENREID,
COMPOSER,
MILLISECOUND,
BYTES,
UNITPRICE)
```

Below the Notepad window, a file explorer window is visible, showing a folder structure. The folder path is "C:\Users\CO\grouptask.CTL". Inside the folder, there are several files and folders: "Links", "Music", "OneDrive", "ORACLE SURI", "Ct", "grouptask.CTL" (which is highlighted), "g-task", and "NN".

## Sources file

This file contain data.

SSS2022260

```

g-task - Notepad
File Edit Format View Help
TRACKID,NAME,ALBUMID,MEDIATYPEID,GENREID,COMPOSER,MILLISECOND,BYTES,UNITPRIC
323,"Dig-Dig, Lambe-Lambe (Ao Vivo)",29,1,9,Cassiano Costa/Cintia Maviane/J. F
324,Pererê,29,1,9,Augusto Conceição/Chiclete Com Banana,198661,6643207,0.99
325,TriboTchan,29,1,9,Cal Adan/Paulo Levi,194194,6507950,0.99
326,"Tapa Aqui, Descobre Ali",29,1,9,Paulo Levi/W. Rangel,188630,6327391,0.99
327,Daniela,29,1,9,Jorge Cardoso/Pierre Onasis,230791,7748006,0.99
328,Bate Lata,29,1,9,Fábio Nolasco/Gal Sales/Ivan Brasil,206733,7034985,0.99
329,Garotas do Brasil,29,1,9,"Garay, Ricardo Engels/Luca Predabom/Ludwig, Car
330,Levada do Amor (Ailoviu),29,1,9,Luiz Wanderley/Paulo Levi,190093,6457752,
331,Lavadeira,29,1,9,"Do Vale, Valverde/Gal Oliveira/Luciano Pinto",214256,72
332,Reboladeira,29,1,9,Cal Adan/Ferrugem/Julinho Carioca/Triona Ni Dhomhnaill
333,É que Nessa Encarnação Eu Nasci Manga,29,1,9,Lucina/Luli,196519,6568081,0
334,Reggae Tchan,29,1,9,"Cal Adan/Del Rey, Tension/Edu Casanova",206654,69313
335,My Love,29,1,9,Jauperi/Zeu Góes,203493,6772813,0.99
336,Latinha de Cerveja,29,1,9,Adriano Bernandes/Edmar Neves,166687,5532564,0.
337,You Shook Me,30,1,1,J B Lenoir/Willie Dixon,315951,10249958,0.99
338,I Can't Quit You Baby,30,1,1,Willie Dixon,263836,8581414,0.99
339,Communication Breakdown,30,1,1,Jimmy Page/John Bonham/John Paul Jones,192
340,Dazed and Confused,30,1,1,Jimmy Page,401920,13035765,0.99
341,The Girl I Love She Got Long Black Wavy Hair,30,1,1,Jimmy Page/John Bonham
342,What is and Should Never Be,30,1,1,Jimmy Page/Robert Plant,260675,8497116
343,Communication Breakdown(2),30,1,1,Jimmy Page/John Bonham/John Paul Jones,
344,Travelling Riverside Blues,30,1,1,Jimmy Page/Robert Johnson/Robert Plant,
345,Whole Lotta Love,30,1,1,Jimmy Page/John Bonham/John Paul Jones/Robert Plant
346,Somethin' Else,30,1,1,Bob Cochran/Sharon Sheeley,127869,4165650,0.99

```

Ln 1, Col 1      100%      Windows (CRLF)      UTF-8

ited 243 KB

```

23-09-2022 19:47 <DIR> Favorites
11-10-2022 16:39 249,103 g-task.txt -> SOURCES FILE
11-10-2022 17:16 342 grouptask.CTL.log
11-10-2022 16:43 226 grouptask.txt -> CONTROL FILE
23-09-2022 19:47 <DIR> Links
23-09-2022 19:47 <DIR> Music
30-09-2022 17:46 163 NN.txt
23-09-2022 19:50 <DIR> OneDrive
28-09-2022 13:43 <DIR> ORACLE SURI
27-09-2022 15:26 <DIR> Pictures
23-09-2022 19:47 <DIR> Saved Games
23-09-2022 19:48 <DIR> Searches
04-10-2022 16:18 10,771 SYSTEM.sql
11-10-2022 14:47 1,619 Team1.sql
03-10-2022 09:45 <DIR> Videos
      13 File(s)    267,750 bytes
      16 Dir(s)   56,186,871,808 bytes free

```

C:\Users\SSS2022260>SQLLDR C##TEAM1/TEAM1 CONTROL=GROUPTASK.TXT

```

SQL*Loader: Release 21.0.0.0.0 - Production on Tue Oct 11 17:20:57 2022
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle and/or its affiliates. All rights reserved.

Path used: Conventional
Commit point reached - logical record count 250
Commit point reached - logical record count 500
Table TRACK1:
  219 Rows successfully loaded. -> ROW LOADED
Check the log file:
  GROUPTASK.log

```

These are type to import data into database.

22.write a query to difference between the materialized view and force view?

MATERIALIZED VIEW	FORCE VIEW
<p>The materialized view different compare to other views because it have own structure.</p> <p>The DML commands are not allowed.</p> <p>In case we do any modifies the base table doesn't affected.</p>	<p>It doesn't have own structure. It is baseless table.</p> <p>The DML commands are allowed.</p> <p>In case we do any modifies the base table also affected.</p>

23.What is sub query how many types explain it?

**Sub query:-** Query within the query is called as a sub query.

The types of sub query,

- 1.Simple Sub Query
- 2.Co related Sub Query

1. Simple Sub Query :-

In simple sub query first inner query is executed independently, based on inner query value outer query is executed.

Outer query is depend on inner query but inner query doesn't depend on outer query. Syntax : select \* from emp where empno= ( select \* from emp ); Ex-1 : Display the employees who are working in research department?

Syntax:- Select empno,ename from emp where deptno=(select deptno from dept where dname='RESEARCH');

## 2. Co related Sub Query

In this query first outer query get executes based on outer query value inner query get executed and return a value and very finally based on the inner query value outer query value will be displayed.

Syntax : select \* from emp e where 1=(select count(\*) from emp where e.sal<=sal);

## 24. Write query to find random records like 1,4,5 and 7 in emp?

A. select \* from (select emp.\* ,rownum rn from emp) where rn in (1,4,5,7);

The screenshot shows the Oracle SQL Developer interface. In the top-left query editor, the following SQL code is written:

```
select * from (select emp.* ,rownum rn from emp)
where rn in (1,4,5,7);
```

A red arrow points from the word "EXECUTE" in the toolbar to the right side of the code editor. Below the editor, the "Script Output" tab is active, showing the results of the query. The results are presented in a table with columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO, and RN. The data is as follows:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	RN
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	1
2	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20	4
3	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	5
4	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20	7

A red arrow points from the word "RESULT" in the bottom right corner to the right edge of the result table.

## 25. Write a query to get last record in EMP?

A. Select \* from  
(Select emp.\* ,rownum rn from EMP)  
Where rn= (select count (empno) from EMP);

The screenshot shows a SQL developer interface. In the top-left pane, there is a code editor containing the following SQL query:

```
Select * from
(SELECT emp.* ,  rownum rn from EMP)
Where rn= (select count (empno) from EMP);
```

A red arrow points from the word "EXECUTE" to the right side of the code editor, indicating where the query should be run.

In the bottom-right pane, the results of the query are displayed in a grid format:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	RN
1	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10	14

A red arrow points from the word "RESULT" to the left side of the results grid, indicating where the results are located.

26. Display the employee details who are getting maximum salary?

A. select \* from emp where sal in(select max (sal) from emp);

The screenshot shows a SQL developer interface. In the top-left pane, there is a code editor containing the following SQL query:

```
select * from emp where sal in
(select max (sal) from emp);
```

A red arrow points from the word "EXECUTE" to the right edge of the code editor.

In the bottom-right pane, the results of the query are displayed in a grid format:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10

A red arrow points from the word "RESULT" to the left edge of the result grid.

27.write about Sequence and explain it?

**Sequence:-**

It is used to create sequence on columns in a table.

While insert the data into tables we use the sequence.

All sequences are stored in all\_sequences.

Syntax : create sequence sequence\_name;

Create sequence sequence\_name increment by 1 start with 1; Currval : it is used to insert current value.

Nextval : it is used to insert next value

```
Create table t1 ( sno number(5), cno number(5));
Insert into t1 values ( seq1.currval, seq1.nextval );
```

28. List the emps in the asc order of their Salaries?

A. select \* from emp order by sal asc;

The screenshot shows the Oracle SQL Developer interface. In the top query editor window, the SQL command `select * from emp order by sal asc;` is entered. A red arrow points from the word "QUERY" to this command. Below the editor, the application bar displays tabs for "Script Output", "Query Result", "Query Result 1", and "Query Result 2". The "Query Result" tab is active, showing the results of the executed query. Another red arrow points from the word "RESULT" to the results table. The results table contains 14 rows of employee data, with columns labeled: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
2	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
3	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
4	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
5	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
6	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10
7	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
8	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
9	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
10	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
11	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
12	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
13	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
14	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10

29. List the details of the emps in asc order of the Dptnos and desc of Jobs?

A. select \* from emp order by deptno asc, job desc;

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, a query is written:

```
select * from emp order by deptno asc, job desc;
```

A red arrow points to this line with the label "QUERY".

In the bottom-right pane, the results of the query are displayed in a grid:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
3	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
5	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
6	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
7	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
8	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
9	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
10	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
11	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
12	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
13	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
14	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30

A red arrow points to the results grid with the label "RESULT".

30. Display all the unique job groups in the descending order?

A. Select distinct job from emp order by job desc;

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, a query is written:

```
Select distinct job from emp order by job desc;
```

A red arrow points to this line with the label "QUERY".

In the bottom-right pane, the results of the query are displayed in a grid:

JOB
SALESMAN
PRESIDENT
MANAGER
CLERK
ANALYST

A red arrow points to the results grid with the label "RESULT".

31. Display all the details of all 'Mgrs'

A. Select \* from emp where job = 'MANAGER';

The screenshot shows the Oracle SQL Developer interface. The top pane displays the SQL query:

```
Select * from emp where job = 'MANAGER';
```

A red arrow points from the word "QUERY" to the right side of the query text. The bottom pane shows the results of the query execution:

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x

SQL | All Rows Fetched: 3 in 0.011 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
2	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
3	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20

A red arrow points from the word "RESULT" to the right side of the results table.

32. List the emps who joined before 1981.

A. select \* from emp where hiredate <('1-jan-1981');

The screenshot shows the Oracle SQL Developer interface. The top pane displays the SQL query:

```
select * from emp where hiredate <('1-jan-1981');
```

A red arrow points from the word "QUERY" to the right side of the query text. The bottom pane shows the results of the query execution:

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x

SQL | All Rows Fetched: 1 in 0.015 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20

A red arrow points from the word "RESULT" to the right side of the results table.

33. List the Empno, Ename, Sal, Daily sal of all emps in the asc order of Annsal.

A. select Empno, Ename, Sal,trunc(sal/30) dsal,sal\*12 asal from emp  
order by asal asc;

The screenshot shows the Oracle SQL Developer interface. The top part displays a SQL query:

```
select Empno, Ename, Sal,trunc(sal/30) dsal,sal*12 asal   from emp  order by asal asc;
```

A red arrow points from the word "QUERY" to the right side of the query text. The bottom part shows the execution results in a grid:

	EMPNO	ENAME	SAL	DSAL	ASAL
1	7369	SMITH	800	26	9600
2	7900	JAMES	950	31	11400
3	7876	ADAMS	1100	36	13200
4	7654	MARTIN	1250	41	15000
5	7521	WARD	1250	41	15000
6	7934	MILLER	1300	43	15600
7	7844	TURNER	1500	50	18000
8	7499	ALLEN	1600	53	19200
9	7782	CLARK	2450	81	29400
10	7698	BLAKE	2850	95	34200
11	7566	JONES	2975	99	35700
12	7902	FORD	3000	100	36000
13	7788	SCOTT	3000	100	36000
14	7839	KING	5000	166	60000

A red arrow points from the word "RESULT" to the left side of the results grid.

34. Display the Empno, Ename, job, Hiredate, Exp of all Mgrs?

A. select Emp.\* ,to\_char(sysdate,'yyyy')-to\_char(hiredate,'yyyy') from emp where job = 'MANAGER';

The screenshot shows the Oracle SQL Developer interface. The top part displays a SQL query:

```
select Emp.* ,to_char(sysdate,'yyyy')-to_char(hiredate,'yyyy') from emp where job = 'MANAGER';
```

A red arrow points from the word "QUERY" to the right side of the query text. The bottom part shows the execution results in a grid:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	TO_CHAR(SYSDATE,'YYYY')-TO_CHAR(HIREDATE,'YYYY')
1	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30	41
2	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10	41
3	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20	41

A red arrow points from the word "RESULT" to the left side of the results grid.

35. List the emps along with their Exp and Daily Sal is more than Rs.100?

A. select Emp.\* ,to\_char (sysdate,'yyyy')-to\_char(hiredate,'yyyy') EXP ,TRUNC(SAL/30) DSAL from emp WHERE (SAL/30) >100;

The screenshot shows the Oracle SQL Developer interface. The top part contains a code editor with the following SQL query:

```
select Emp.* ,to_char (sysdate,'yyyy')-to_char(hiredate,'yyyy') EXP ,TRUNC(SAL/30) DSAL from emp WHERE (SAL/30) >100;
```

To the right of the code editor, a red arrow points to the word "QUERY". Below the code editor is a toolbar with tabs: Script Output, Query Result, Query Result 1, and Query Result 2. The "Query Result" tab is selected. Underneath the toolbar, it says "All Rows Fetched: 1 in 0.014 seconds". The main area displays a table with the following data:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EXP	DSAL
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	41	166

To the right of the table, a red arrow points to the word "RESULT".

36. List the emps who are either 'CLERK' or 'ANALYST' in the Desc order?

A. select \* from emp where job = 'CLERK' or job = 'ANALYST' order by job desc;

The screenshot shows the Oracle SQL Developer interface. The top part contains a code editor with the following SQL query:

```
select * from emp where job = 'CLERK' or job = 'ANALYST' order by job desc;
```

To the right of the code editor, a red arrow points to the word "QUERY". Below the code editor is a toolbar with tabs: Script Output, Query Result, Query Result 1, and Query Result 2. The "Query Result" tab is selected. Underneath the toolbar, it says "All Rows Fetched: 6 in 0.013 seconds". The main area displays a table with the following data:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
2	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
3	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10
4	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
5	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
6	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20

To the right of the table, a red arrow points to the word "RESULT".

37. List the emps who joined on 1-MAY-81,3-DEC-81,17-DEC-81,19-JAN-80 in ascorder of seniority.

A. select \* from emp where hiredate in ('01-may-81','31-dec-81','17-dec-81','19-jan-80')order by hiredate asc;

```
SELECT * from emp where hiredate in ('01-may-81','03-dec-81','17-dec-81','19-jan-80') order by HIREDATE asc;
```

Script Output x | Query Result x  
SQL | All Rows Fetched: 3 in 0.005 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
2	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
3	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20

38. List the emp who are working for the Deptno 10 or20?

A. select \* from emp where deptno in ('10','20');

```
select * from emp where deptno in ('10','20');
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x  
SQL | All Rows Fetched: 8 in 0.016 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
3	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
4	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
5	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
6	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
7	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
8	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10

39. List the emps who are joined in the month of Aug 1980?

A. select \* from emp WHERE to\_CHAR(hiredate, 'MON-YYYY') ='AUG-1980';

The screenshot shows the SQL developer interface. The query window contains the SQL statement: `select * from emp WHERE to_CHAR(hiredate, 'MON-YYYY') ='AUG-1980';`. A red arrow points to this statement with the label "QUERY". Below the query window is the results tab, which displays the employee details for those hired in August 1980. A red arrow points to the results with the label "RESULT". The results table has columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO. The data shown is:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO

40. List the Enames those are starting with 'S' and with five characters?

A.SELECT \* FROM EMP WHERE ENAME LIKE 'S%' AND LENGTH(ENAME)=5 ;

The screenshot shows the SQL developer interface. The query window contains the SQL statement: `SELECT * FROM EMP WHERE ENAME LIKE 'S%' AND LENGTH(ENAME)=5 ;`. A red arrow points to this statement with the label "QUERY". Below the query window is the results tab, which displays the employee names starting with 'S' and having exactly 5 characters. A red arrow points to the results with the label "RESULT". The results table has columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO. The data shown is:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7788 SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
2	7369 SMITH	CLERK	7902	17-12-80	800	(null)	20

41. List the emps those are having four chars and third character must be 'r'?

A. SELECT \* FROM EMP WHERE ENAME LIKE '\_\_R%' AND LENGTH(ENAME)=4;

The screenshot shows the Oracle SQL Developer interface. The top part displays a query in the SQL editor:

```
SELECT * FROM EMP WHERE ENAME LIKE '_R%' AND LENGTH (ENAME)=4;
```

A red arrow points from the text "QUERY" to the right edge of the SQL editor. The bottom part shows the results of the query in the Query Result tab:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7902	FORD	ANALYST	7566 03-12-81	3000	(null)	20
2	7521	WARD	SALESMAN	7698 22-02-81	1250	500	30

A red arrow points from the text "RESULT" to the right edge of the results grid.

42. List the emps who joined in January?

A. SELECT \* FROM EMP WHERE TO\_CHAR(HIREDATE , 'MON') = 'JAN';

The screenshot shows the Oracle SQL Developer interface. The top part displays a query in the SQL editor:

```
SELECT * FROM EMP WHERE TO_CHAR (HIREDATE , 'MON') = 'JAN';
```

A red arrow points from the text "QUERY" to the right edge of the SQL editor. The bottom part shows the results of the query in the Query Result tab:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7934	MILLER	CLERK	7782 23-01-82	1300	(null)	10

A red arrow points from the text "RESULT" to the right edge of the results grid.

43. List the emps who joined in the month of which second character is 'a'?

A. SELECT \* FROM EMP WHERE TO\_CHAR(HIREDATE,'MONTH') LIKE '\_A%';

The screenshot shows the Oracle SQL Developer interface. The top part displays a query in the SQL editor:

```
SELECT * FROM EMP WHERE TO_CHAR(HIREDATE, 'MONTH') LIKE '_A%';
```

A red arrow points from the text "QUERY" to the right edge of the SQL editor. The bottom part shows the results of the query in the Query Result tab:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698	BLAKE	MANAGER	7839 01-05-81	2850	(null)	30
2	7876	ADAMS	CLERK	7788 23-05-87	1100	(null)	20
3	7934	MILLER	CLERK	7782 23-01-82	1300	(null)	10

A red arrow points from the text "RESULT" to the right edge of the results grid.

44. List the emps whose Sal is four-digit number ending with Zero?

A. SELECT \* FROM EMP WHERE SAL LIKE '%0' AND LENGTH (SAL)=4;

The screenshot shows the Oracle SQL Developer interface. The query editor at the top contains the SQL statement: `SELECT * FROM EMP WHERE SAL LIKE '%0' AND LENGTH (SAL)=4;`. A red arrow points from the word "QUERY" to the right side of the query editor. Below the editor is a toolbar with tabs: Script Output, Query Result, Query Result 1, and Query Result 2. The "Query Result" tab is selected. A status message below the toolbar says "All Rows Fetched: 11 in 0.007 seconds". The main area displays a grid of 11 rows of employee data from the EMP table. The columns are labeled: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The data includes rows for KING, BLAKE, CLARK, SCOTT, FORD, ALLEN, WARD, MARTIN, TURNER, ADAMS, and MILLER.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
5	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
6	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
7	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
8	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
9	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
10	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
11	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10

45. List the emps whose names having a character set 'll' together?

A. `SELECT * FROM EMP WHERE ENAME LIKE '%LL%';`

The screenshot shows the Oracle SQL Developer interface. The query editor at the top contains the SQL statement: `SELECT * FROM EMP WHERE ENAME LIKE '%LL%';`. A red arrow points from the word "QUERY" to the right side of the query editor. Below the editor is a toolbar with tabs: Script Output, Query Result, Query Result 1, and Query Result 2. The "Query Result" tab is selected. A status message below the toolbar says "All Rows Fetched: 2 in 0.015 seconds". The main area displays a grid of 2 rows of employee data from the EMP table. The columns are labeled: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The data includes rows for ALLEN and MILLER.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
2	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10

46. List the emps who does not belong to Deptno 20?

A. `SELECT * FROM EMP WHERE DEPTNO<>20;`

The screenshot shows the Oracle SQL Developer interface. At the top, a red arrow points from the text "QUERY" to the SQL statement in the editor:

```
SELECT * FROM EMP WHERE DEPTNO<>20;
```

Below the editor, the results are displayed in a table. A red arrow points from the text "RESULT" to the table header:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
5	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
6	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
7	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
8	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
9	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10

47. List all the emps except 'PRESIDENT' & 'MGR" in asc order of Salaries?

A. SELECT \* FROM EMP WHERE JOB NOT IN ('PRESIDENT', 'MANAGER') ORDER BY SAL ASC;

The screenshot shows the Oracle SQL Developer interface. At the top, a red arrow points from the text "QUERY" to the SQL statement in the editor:

```
SELECT * FROM EMP WHERE JOB NOT IN ('PRESIDENT', 'MANAGER') ORDER BY SAL ASC;
```

Below the editor, the results are displayed in a table. A red arrow points from the text "RESULT" to the table header:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
2	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
3	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
4	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
5	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
6	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10
7	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
8	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
9	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
10	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20

48. List the emps who joined in any year but not belongs to the month of March?

A. SELECT \* FROM EMP WHERE TO\_CHAR(HIREDATE,'MONTH')<>'MARCH';

The screenshot shows the Oracle SQL Developer interface. A red arrow points from the text "QUERY" to the SQL command in the top-left pane:

```
SELECT * FROM EMP WHERE TO_CHAR(HIREDATE, 'MONTH') <> 'MARCH';
```

The top-right pane is labeled "RESULT". A red arrow points from the word "RESULT" to the data grid below. The data grid displays 14 rows of employee information from the EMP table, filtered to exclude March hires.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
5	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
6	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
7	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
8	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
9	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
10	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
11	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
12	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
13	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
14	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10

49. List the emps of Deptno 30 or 10 joined in the year 1981?

A. SELECT \* FROM EMP WHERE TO\_CHAR(HIREDATE,'YYYY') = 1981 AND DEPTNO IN (30,10);

The screenshot shows the Oracle SQL Developer interface. A red arrow points from the text "QUERY" to the SQL command in the top-left pane:

```
SELECT * FROM EMP WHERE TO_CHAR(HIREDATE, 'YYYY') = 1981 AND DEPTNO IN (30,10);
```

The top-right pane is labeled "RESULT". A red arrow points from the word "RESULT" to the data grid below. The data grid displays 8 rows of employee information from the EMP table, filtered to show employees hired in 1981 who work in either department 30 or 10.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
5	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
6	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
7	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
8	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30

50. List all the Clerks of Deptno 20?

A.SELECT \* FROM EMP WHERE JOB = 'CLERK' AND DEPTNO=20;

The screenshot shows the Oracle SQL Developer interface. The top part displays the SQL query:

```
SELECT * FROM EMP WHERE JOB = 'CLERK' AND DEPTNO=20;
```

An arrow points from the word "QUERY" to the right side of the query text. The bottom part shows the results of the query execution:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902 17-12-80	800	(null)	20
2	7876	ADAMS	CLERK	7788 23-05-87	1100	(null)	20

An arrow points from the word "RESULT" to the right side of the table.

51. Display the details of SMITH?

A.SELECT \* FROM EMP WHERE ENAME = 'SMITH';

The screenshot shows the Oracle SQL Developer interface. The top part displays the SQL query:

```
SELECT * FROM EMP WHERE ENAME = 'SMITH';
```

An arrow points from the word "QUERY" to the right side of the query text. The bottom part shows the results of the query execution:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902 17-12-80	800	(null)	20

An arrow points from the word "RESULT" to the right side of the table.

52. List the total information of EMP table along with DNAME and Loc of all theemps Working Under 'ACCOUNTING' & 'RESEARCH' in the asc Deptno?

A. SELECT E.\* ,D.\* FROM EMP E,DEPT D WHERE D.DNAME IN ('ACCOUNTING','RESEARCH') ORDER BY E.DEPTNO ASC;

The screenshot shows the SQL Server Management Studio interface. A red arrow points from the text area at the top to the word 'QUERY'. Another red arrow points from the data grid below to the word 'RESULT'.

```
SELECT E.* , D.* FROM EMP E,DEPT D WHERE D.DNAME IN ('ACCOUNTING','RESEARCH') ORDER BY E.DEPTNO ASC;
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO_1	DNAME	LOC
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	20	RESEARCH	DALLAS
2	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10	10	ACCOUNTING	NEW YORK
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10	10	ACCOUNTING	NEW YORK
4	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	10	ACCOUNTING	NEW YORK
5	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10	20	RESEARCH	DALLAS
6	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10	20	RESEARCH	DALLAS
7	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20	20	RESEARCH	DALLAS
8	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20	10	ACCOUNTING	NEW YORK
9	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	10	ACCOUNTING	NEW YORK
10	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	20	RESEARCH	DALLAS
11	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20	20	RESEARCH	DALLAS
12	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20	20	RESEARCH	DALLAS
13	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20	10	ACCOUNTING	NEW YORK
14	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20	10	ACCOUNTING	NEW YORK

53. List the Empno, Ename, Sal, Dname of all the 'MGRS' and 'ANALYST' working in New York, Dallas with an exp more than 7 years without receiving theComm asc order of Loc?

A. SELECT \* FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPTNO AND E.JOB IN('MANAGER','ANALYST') AND D.LOC IN ('NEW YORK','DALLAS') AND TO\_CHAR(SYSDATE,'YYYY')-TO\_CHAR (HIREDATE,'YYYY')>7 ;

The screenshot shows the SQL Server Management Studio interface. A red arrow points from the text area at the top to the word 'QUERY'. Another red arrow points from the data grid below to the word 'RESULT'.

```
SELECT * FROM EMP E,DEPT D WHERE E.DEPTNO=D.DEPTNO AND E.JOB IN('MANAGER','ANALYST') AND D.LOC IN ('NEW YORK','DALLAS') AND TO_CHAR(SYSDATE,'YYYY')-TO_CHAR (HIREDATE,'YYYY')>7 ;
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO_1	DNAME	LOC
1	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10	10	ACCOUNTING	NEW YORK
2	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20	20	RESEARCH	DALLAS
3	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	20	RESEARCH	DALLAS
4	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20	20	RESEARCH	DALLAS

54. Display the Empno, Ename, Sal, Dname, Loc, Deptno, Job of all emps working at CICAGO or working for ACCOUNTING dept with Ann Sal>28000, but the Sal should not be=3000 or 2800 who doesn't belong to the Mgr and whose no is having a digit '7' or '8' in 3<sup>rd</sup> position in the asc order of Deptno

and desc orderof job.

A.SELECT

\*

FROM

(

SELECT

e.empno,  
e.ename,  
e.job,  
e.mgr,  
e.hiredate,  
e.sal,  
e.comm,  
E.DEPTNO,  
d.dname,  
d.loc,  
e.sal \* 12 asal

FROM

emp e,  
dept d

WHERE

e.deptno = d.deptno

)

WHERE

loc = 'CHICAGO'

OR dname = 'ACCOUNTING'

AND ( sal \* 12 ) > 2800

AND sal NOT IN ( 2800, 3000 )

AND EMPNO NOT IN(SELECT MGR FROM EMP)

AND EMPNO IN (SELECT EMPNO FROM EMP WHERE EMPNO LIKE '\_\_7%' OR

EMPNO LIKE '\_8%')

ORDER BY JOB DESC;

The screenshot shows the Oracle SQL Developer interface. The top part is the 'Worksheet' tab, labeled 'Query Builder'. It contains the following SQL code:

```

SELECT * FROM (SELECT e.empno, e.ename, e.job, e.mgr, e.hiredate, e.sal, e.comm, E.DEPTNO, D.dname, d.loc, e.sal * 12 ASAL FROM emp e,dept d
  WHERE
    e.deptno = d.deptno
)
WHERE
  loc = 'CHICAGO'
  OR dname = 'ACCOUNTING'
  AND ( sal * 12 ) > 2800
  AND sal NOT IN ( 2800, 3000 )
  AND EMPNO NOT IN (SELECT MGR FROM EMP)
  AND EMPNO IN (SELECT EMPNO FROM EMP WHERE EMPNO LIKE '_7%' OR
EMPNO LIKE '_8%')
ORDER BY JOB DESC;

```

A red arrow points from the word 'QUERY' to the code area.

The bottom part is the 'Script Output' tab, labeled 'SQL | All Rows Fetched: 6 in 0.03 seconds'. It displays the results of the query:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DNAME	LOC	ASAL
1	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30	SALES	CHICAGO	15000
2	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30	SALES	CHICAGO	18000
3	7499	ALLLEN	SALESMAN	7698	20-02-81	1600	300	30	SALES	CHICAGO	19200
4	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30	SALES	CHICAGO	15000
5	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30	SALES	CHICAGO	34200
6	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30	SALES	CHICAGO	11400

A red arrow points from the word 'RESULT' to the results table.

55. Display all Grade 4,5 Analyst and Mgr?

A. Select \* from emp e, salgrade s where e.sal between s.losal and s.hisal and s.grade in (4,5) and e.empno in (select e.empno from emp e where e.job in ('MANAGER','ANALYST'));

The screenshot shows the Oracle SQL Developer interface. The top part is the 'Worksheet' tab, labeled 'Query Builder'. It contains the following SQL code:

```

Select * from emp e, salgrade s where e.sal between s.losal and s.hisal and
s.grade in (4,5) and e.empno in (select e.empno from emp e where e.job in
('MANAGER','ANALYST') );

```

A red arrow points from the word 'QUERY' to the code area.

The bottom part is the 'Script Output' tab, labeled 'SQL | All Rows Fetched: 5 in 0.001 seconds'. It displays the results of the query:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	GRADE	LOSLA	HISAL
1	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30	4	2001	3000
2	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10	4	2001	3000
3	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20	4	2001	3000
4	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	4	2001	3000
5	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20	4	2001	3000

A red arrow points from the word 'RESULT' to the results table.

56. List the Empno, Ename, Sal, Dname, Grade, Exp, and Ann Sal of emps working for Dept10 or20?

**A. SELECT**

```

e.empno,
e.ename,
e.sal,
d.dname,
s.grade,
to_char(sysdate, 'YYYY') - to_char(hiredate, 'YYYY') exp,
( e.sal * 12 ) ann_sal,E.DEPTNO

FROM
emp      e,
dept      d,
salgrade s

WHERE
e.deptno IN ( 10, 20 )
AND e.deptno = d.deptno
AND e.sal BETWEEN s.losal AND s.hisal;

```

**QUERY**

The screenshot shows the Oracle SQL Developer interface. On the left, the SQL tab contains the SQL query. On the right, the Results tab displays the output of the query. Red arrows point from the labels 'QUERY' and 'RESULT' to their respective tabs.

**Query:**

```

e.empno,
e.ename,
e.sal,
d.dname,
s.grade,
to_char(sysdate, 'YYYY') - to_char(hiredate, 'YYYY') exp,
( e.sal * 12 ) ann_sal,E.DEPTNO

FROM
emp      e,
dept      d,
salgrade s

WHERE
e.deptno IN ( 10, 20 )
AND e.deptno = d.deptno
AND e.sal BETWEEN s.losal AND s.hisal;

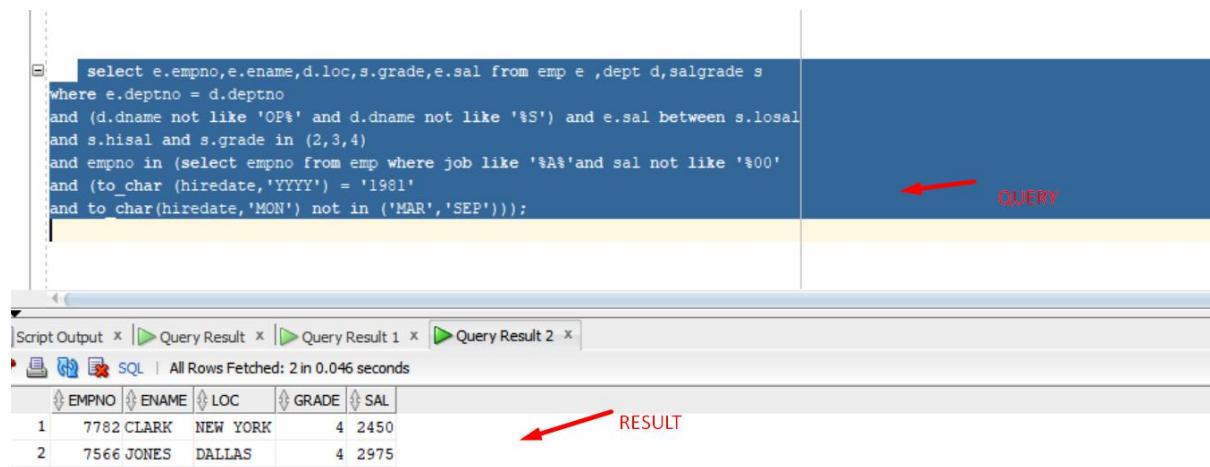
```

**Result:**

	EMPNO	ENAME	SA	DNAME	GRADE	EXP	ANN_SAL	DEPTNO
1	7934 MILLER	1300 ACCOUNTING	2	40	15600	10		
2	7782 CLARK	2450 ACCOUNTING	4	41	29400	10		
3	7839 KING	5000 ACCOUNTING	5	41	60000	10		
4	7369 SMITH	800 RESEARCH	1	42	9600	20		
5	7876 ADAMS	1100 RESEARCH	1	35	13200	20		
6	7566 JONES	2975 RESEARCH	4	41	35700	20		
7	7902 FORD	3000 RESEARCH	4	41	36000	20		
8	7788 SCOTT	3000 RESEARCH	4	35	36000	20		

57. List all the information of emp with Loc and the Grade of all the emps belong to the Grade range from 2 to 4 working at the Dept those are not starting with char set 'OP' and not ending with 'S' with the designation having a char 'a' anywhere joined in the year 1981 but not in the month of Mar or Sep and Sal not end with '00' in the asc order of Grades?

A. select e.empno,e.ename,d.loc,s.grade,e.sal from emp e ,dept d,salgrade s  
 where e.deptno = d.deptno  
 and (d.dname not like 'OP%' and d.dname not like '%S') and e.sal between s.losal  
 and s.hisal and s.grade in (2,3,4)  
 and empno in (select empno from emp where job like '%A%'and sal not like '%00'  
 and (to\_char (hiredate,'YYYY') = '1981'  
 and to\_char(hiredate,'MON') not in ('MAR','SEP')));



The screenshot shows the Oracle SQL Developer interface. The top half displays the SQL query in the 'QUERY' tab, which is highlighted with a red arrow. The bottom half shows the 'Script Output' tab, which contains the query results. A red arrow points to the results table.

```

select e.empno,e.ename,d.loc,s.grade,e.sal from emp e ,dept d,salgrade s
where e.deptno = d.deptno
and (d.dname not like 'OP%' and d.dname not like '%S') and e.sal between s.losal
and s.hisal and s.grade in (2,3,4)
and empno in (select empno from emp where job like '%A%'and sal not like '%00'
and (to_char (hiredate,'YYYY') = '1981'
and to_char(hiredate,'MON') not in ('MAR','SEP')));
  
```

EMPNO	ENAME	LOC	GRADE	SAL
1	7782 CLARK	NEW YORK	4	2450
2	7566 JONES	DALLAS	4	2975

58. List the emps whose Jobs are same as ALLEN?

A. SELECT \* FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME ='ALLEN');

```
SELECT * FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME ='ALLEN');
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7499 ALLEN	SALESMAN	7698 20-02-81	1600	300	30	
2	7521 WARD	SALESMAN	7698 22-02-81	1250	500	30	
3	7654 MARTIN	SALESMAN	7698 28-09-81	1250	1400	30	
4	7844 TURNER	SALESMAN	7698 08-09-81	1500	0	30	

59. List the emps who are senior to King?

A.SELECT \* FROM EMP WHERE HIREDATE < ( SELECT HIREDATE FROM EMP WHERE ENAME = 'KING');

```
SELECT * FROM EMP WHERE HIREDATE < ( SELECT HIREDATE FROM EMP WHERE ENAME = 'KING')
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698 BLAKE	MANAGER	7839 01-05-81	2850	(null)	30	
2	7782 CLARK	MANAGER	7839 09-06-81	2450	(null)	10	
3	7566 JONES	MANAGER	7839 02-04-81	2975	(null)	20	
4	7369 SMITH	CLERK	7902 17-12-80	800	(null)	20	
5	7499 ALLEN	SALESMAN	7698 20-02-81	1600	300	30	
6	7521 WARD	SALESMAN	7698 22-02-81	1250	500	30	
7	7654 MARTIN	SALESMAN	7698 28-09-81	1250	1400	30	
8	7844 TURNER	SALESMAN	7698 08-09-81	1500	0	30	

60. List the Emps who are senior to their own MGRS?

A. `SELECT * FROM EMP E,EMP E2 WHERE E.EMPNO=E2.MGR AND E.HIREDATE > E2.HIREDATE;`

The screenshot shows a SQL query being run in Oracle SQL Developer. The query is:

```
SELECT * FROM EMP E,EMP E2 WHERE E.EMPNO=E2.MGR AND E.HIREDATE > E2.HIREDATE;
```

Below the query, the results are displayed in a grid. Red arrows point from the text labels 'QUERY' and 'RESULT' to the respective parts of the interface.

**Script Output** | **Query Result** | **Query Result 1** | **Query Result 2**

All Rows Fetched: 6 in 0.026 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EMPNO_1	ENAME_1	JOB_1	MGR_1	HIREDATE_1	SAL_1	COMM_1	DEPTNO_1
1	7839 KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	7698 BLAKE	MANAGER	7839 01-05-81	2850	(null)	30			
2	7839 KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	7782 CLARK	MANAGER	7839 09-06-81	2450	(null)	10			
3	7839 KING	PRESIDENT	(null)	17-11-81	5000	(null)	10	7566 JONES	MANAGER	7839 02-04-81	2975	(null)	20			
4	7902 FORD	ANALYST	7566 03-12-81	3000	(null)	20	7369 SMITH	CLERK	7902 17-12-80	800	(null)	20				
5	7698 BLAKE	MANAGER	7839 01-05-81	2850	(null)	30	7499 ALLEN	SALESMAN	7698 20-02-81	1600	300	30				
6	7698 BLAKE	MANAGER	7839 01-05-81	2850	(null)	30	7521 WARD	SALESMAN	7698 22-02-81	1250	500	30				

61. List the Emps of Deptno 20 whose Jobs are same as Deptno10.

A. `SELECT*FROM EMP WHERE DEPTNO=20 AND JOB IN (SELECT JOB FROM EMP WHERE DEPTNO=10);`

```

40
41  SELECT*FROM EMP WHERE DEPTNO=20 AND
42  JOB IN (SELECT JOB FROM EMP WHERE DEPTNO=10);
43

```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.002 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
2	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
3	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20

62. List the Emps whose Sal is same as FORD or SMITH in desc order of Sal.A)

- A. SELECT\*FROM EMP WHERE SAL IN (SELECT SAL FROM EMP WHERE ENAME IN('SMITH','FORD'))ORDER BY SAL DESC;

```

L46
L47  SELECT*FROM EMP WHERE SAL IN
L48  (SELECT SAL FROM EMP WHERE ENAME IN('SMITH', 'FORD'))ORDER BY SAL DESC;
L49

```

Script Output x Query... x

SQL | All Rows Fetched: 3 in 0.015 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
2	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
3	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20

63. List the emps Whose Jobs are same as MILLER or Sal is more than ALLEN.

- A. **SELECT\*FROM EMP WHERE JOB IN (SELECT JOB FROM EMP WHERE ENAME='MILLER') OR SAL>(SELECT SAL FROM EMP WHERE ENAME='ALLEN');**

The screenshot shows the SQL Developer interface. The SQL tab contains the following code:

```

.58 | SELECT*FROM EMP WHERE JOB IN (SELECT JOB FROM EMP WHERE ENAME='MILLER')
.59 | OR SAL>(SELECT SAL FROM EMP WHERE ENAME='ALLEN');

```

The Query Result tab shows the output of the query:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
5	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
6	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
7	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
8	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
9	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
10	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10

64. List the Emps whose Sal is > the total remuneration of the SALESMAN.

- A. **SELECT\*FROM EMP WHERE SAL > (SELECT SUM(NVL2(COMM,SAL+COMM,SAL))FROM EMP WHERE JOB='SALESMAN');**

The screenshot shows the SQL Developer interface. The SQL tab contains the following code:

```

SELECT*FROM EMP WHERE
SAL > (SELECT SUM(NVL2(COMM, SAL+COMM, SAL)) FROM EMP WHERE JOB='SALESMAN');

```

The Query Result tab shows the output of the query:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO

65. List the emps who are senior to BLAKE working at CHICAGO & BOSTON.

- A. SELECT\*FROM EMP E,DEPT D WHERE D.LOC IN('CHICAGO','BOSTON')  
AND E.DEPTNO=D.DEPTNO AND E.HIREDATE <(SELECT E.HIREDATE  
FROM EMP E WHERE E.ENAME='BLAKE');

```
SELECT*FROM EMP E,DEPT D WHERE D.LOC IN('CHICAGO', 'BOSTON')
AND E.DEPTNO=D.DEPTNO
AND E.HIREDATE <(SELECT E.HIREDATE FROM EMP E WHERE E.ENAME='BLAKE');
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.002 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO_1	DNAME	LOC
1	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30	30	SALES	CHICAGO
2	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30	30	SALES	CHICAGO

66. List the Emps of Grade 3,4 belongs to the dept ACCOUNTING and RESEARCH?

- A. SELECT \* FROM EMP E WHERE E.DEPTNO IN (SELECT D.DEPTNO FROM DEPT D WHERE D.DNAME IN ('ACCOUNTING','RESEARCH'))  
AND E.EMPNO IN (SELECT E.EMPNO FROM EMP E ,SALGRADE S WHERE E.SAL  
BETWEEN S.LOSAL AND S.HISAL AND S.GRADE IN (3,4) ) ;

```
SELECT * FROM EMP E WHERE E.DEPTNO IN (SELECT D.DEPTNO FROM
DEPT D WHERE D.DNAME IN ('ACCOUNTING','RESEARCH'))
AND E.EMPNO IN (SELECT E.EMPNO FROM EMP E ,SALGRADE S WHERE E.SAL
BETWEEN S.LOSAL AND S.HISAL AND S.GRADE IN (3,4)) ;
```

Script Output x Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 4 in 0.074 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
2	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
3	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
4	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20

67. whose Sal is more than ALLEN and exp more than SMITH in the asc order of EXP.

- A. SELECT\*FROM EMP WHERE SAL>(SELECT SAL FROM EMP WHERE ENAME='ALLEN') AND HIREDATE>(SELECT HIREDATE FROM EMP WHERE ENAME='SMITH') ORDER BY HIREDATE;

```

203
204  SELECT*FROM EMP WHERE SAL>(SELECT SAL FROM EMP WHERE ENAME='ALLEN')
205  AND HIREDATE>(SELECT HIREDATE FROM EMP WHERE ENAME='SMITH')
206  ORDER BY HIREDATE;

```

Script Output x | Query Result x

SQL | All Rows Fetched: 6 in 0.01 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
5	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
6	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20

68. List the emps whose jobs same as SMITH or ALLEN.

- A. SELECT\*FROM EMP WHERE JOB IN (SELECT JOB FROM EMP WHERE ENAME IN ('SMITH','ALLEN'));

```

J9
L0  SELECT*FROM EMP WHERE
L1  JOB IN (SELECT JOB FROM EMP WHERE ENAME IN ('SMITH', 'ALLEN'));

```

Script Output x Query Result x

SQL | All Rows Fetched: 8 in 0.013 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
2	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
3	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30
4	7934	MILLER	CLERK	7782	23-01-82	1300	(null)	10
5	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
6	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
7	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
8	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30

69. Write a Query to display the details of emps whose Sal is same as of

- A. SELECT\*FROM EMP WHERE SAL IN (SELECT SAL FROM EMP E WHERE EMP.EMSSPNO<>E.EMPNO);

```

3
4  SELECT*FROM EMP WHERE SAL IN
5  (SELECT SAL FROM EMP E WHERE EMP.EMPNO<>E.EMPNO);

```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.014 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
2	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
3	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
4	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30

70. Any jobs of deptno 10 those that are not found in deptno 20.

- A. `SELECT E.JOB FROM emp E WHERE E.DEPTNO = 10 AND E.JOB NOT IN (SELECT JOB FROM emp WHERE DEPTNO =20);`

```
26 | 
27 |   SELECT E.JOB FROM emp E
28 |   WHERE E.DEPTNO = 10
29 |   AND E.JOB NOT IN (SELECT JOB FROM
30 |   emp WHERE DEPTNO =20);
```

Script Output x Query Result x  
SQL | All Rows Fetched: 1 in 0.005 seconds

JOB
1 PRESIDENT

71. List of emps of emp1 who are not found in emp2.

72. Find the highest sal of EMP table.

- A. `SELECT MAX(SAL) FROM EMP;`

```
15 | 
16 |   SELECT MAX(SAL)  FROM EMP;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 1 in 0.005 seconds

MAX(SAL)
1 5000

73. Find details of highest paid employee

- A. `SELECT*FROM EMP WHERE SAL=(SELECT MAX(SAL)FROM EMP);`

```
7
3 | SELECT*FROM EMP WHERE SAL=(SELECT MAX(SAL)FROM EMP);
4 |
5 | Script Output x Query Result x
6 | SQL | All Rows Fetched: 1 in 0.007 seconds
7 |   EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
8 | 1 7839 KING PRESIDENT (null) 17-11-81 5000 (null) 10
```

74. Find the highest paid employee of sales department.

- A. `SELECT*FROM EMP WHERE SAL IN(SELECT MAX(SAL)FROM EMP WHERE DEPTNO IN (SELECT D.DEPTNO FROM DEPT D WHERE D.DNAME='SALES'));`

```
:45
:46 | SELECT*FROM EMP WHERE SAL IN(SELECT MAX(SAL)FROM EMP
:47 | WHERE DEPTNO IN (SELECT D.DEPTNO FROM DEPT D WHERE D.DNAME='SALES'));
```

```
Script Output x Query Result x
SQL | All Rows Fetched: 1 in 0.009 seconds
  EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
1 7698 BLAKE MANAGER 7839 01-05-81 2850 (null) 30
```

75. List the most recently hired emp of grade3 belongs to location CHICAGO.

- A. `SELECT * FROM EMP E WHERE E.DEPTNO IN ( SELECT D.DEPTNO FROM DEPT D WHERE D.LOC = 'CHICAGO') AND E.HIREDATE IN (SELECT MAX(HIREDATE) FROM EMP WHERE EMPNO IN (SELECT EMPNO FROM EMP E,SALGRADE S WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL AND S.GRADE = 3)) ;`

```

50
51 SELECT * FROM EMP E WHERE E.DEPTNO IN ( SELECT D.DEPTNO FROM DEPT D WHERE
52 D.LOC = 'CHICAGO') AND
53 E.HIREDATE IN (SELECT MAX(HIREDATE) FROM EMP WHERE EMPNO IN (SELECT EMPNO
54 FROM EMP E,SALGRADE S
55 WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL AND S.GRADE = 3)) ;

```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.002 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7844	TURNER	SALESMAN	7698 08-09-81	1500	0	30

76. List the employees who are senior to most recently hired employee working under king

A. SELECT \* FROM EMP WHERE HIREDATE < (SELECT MAX(HIREDATE)  
FROM EMP WHERE MGR IN (SELECT EMPNO FROM EMP WHERE ENAME  
= 'KING')) ;

```

56
57 SELECT * FROM EMP WHERE HIREDATE < (SELECT MAX(HIREDATE) FROM EMP WHERE MGR
58 IN (SELECT EMPNO FROM EMP WHERE ENAME = 'KING')) ;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.012 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698	BLAKE	MANAGER	7839 01-05-81	2850	(null)	30
2	7566	JONES	MANAGER	7839 02-04-81	2975	(null)	20
3	7369	SMITH	CLERK	7902 17-12-80	800	(null)	20
4	7499	ALLEN	SALESMAN	7698 20-02-81	1600	300	30
5	7521	WARD	SALESMAN	7698 22-02-81	1250	500	30

77. List the details of the employee belong to New York with grade 3 to 5 except 'PRESIDENT' whose sal> the highest paid employee of Chicago in a groupwhere there is manager and salesman not working under king

- A. SELECT \* FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DEPT.LOC ='NEW YORK') AND EMPNO IN (SELECT EMPNO FROM EMP E,SALGRADE S WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL AND S.GRADE IN (3,4,5) ) AND JOB != 'PRESIDENT' AND SAL >(SELECT MAX(SAL) FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DEPT.LOC = 'CHICAGO') AND JOB IN ('MANAGER','SALESMAN')) AND MGR NOT IN (SELECT EMPNO FROM EMP WHERE ENAME = 'KING'));

```

62
63 |   SELECT * FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DEPT.LOC
64 |   ='NEW YORK') AND EMPNO IN (SELECT EMPNO FROM EMP E,SALGRADE S WHERE E.SAL BETWEEN S.LOSAL
65 |   AND S.HISAL AND S.GRADE IN (3,4,5) ) AND JOB != 'PRESIDENT' AND SAL >(SELECT MAX(SAL) FROM EMP
66 |   WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DEPT.LOC = 'CHICAGO') AND JOB IN
67 |   ('MANAGER','SALESMAN')) AND MGR NOT IN (SELECT EMPNO FROM EMP WHERE ENAME = 'KING'));

```

Script Output x Query Result x

P SQL | All Rows Fetched: 1 in 0.003 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7782	CLARK	MANAGER	7839 09-06-81	2450	(null)	10

78. List the details of the senior employee belong to 1981.

- A. SELECT\*FROM EMP WHERE HIREDATE IN(SELECT MIN(HIREDATE)FROM EMP WHERE TO\_CHAR(HIREDATE,'YYYY')=1981);

```

1
2 |   SELECT*FROM EMP WHERE HIREDATE
3 |   IN(SELECT MIN(HIREDATE)FROM EMP WHERE TO_CHAR(HIREDATE, 'YYYY')=1981);

```

Script Output x Query Result x

P SQL | All Rows Fetched: 1 in 0.009 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7499	ALLEN	SALESMAN	7698 20-02-81	1600	300	30

79. List the employees who joined in 1981 with the job same as the most seniorperson of the year 1981.

A. SELECT\*FROM EMP WHERE HIREDATE IN (SELECT HIREDATE FROM EMP WHERE JOB IN(SELECT JOB FROM EMP WHERE HIREDATE IN(SELECT MIN(HIREDATE)FROM EMP WHERE TO\_CHAR(HIREDATE,'YYYY')=1981)));

```

9
10 |   SELECT*FROM EMP WHERE
11 |     HIREDATE IN (SELECT HIREDATE FROM EMP WHERE
12 |       JOB IN(SELECT JOB FROM EMP WHERE
13 |         HIREDATE IN(SELECT MIN(HIREDATE)FROM EMP WHERE
14 |           TO_CHAR(HIREDATE,'YYYY')=1981)));

```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.022 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
2	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
3	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
4	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30

80. List the most senior empl working under the king and grade is more than 3

A. SELECT\*FROM EMP WHERE MGR IN(SELECT EMPNO FROM EMP WHERE ENAME='KING') AND EMPNO IN(SELECT EMPNO FROM EMP E,SALGRADE S WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL AND S.GRADE>3);

```

1 |   SELECT*FROM EMP WHERE
2 |     MGR IN(SELECT EMPNO FROM EMP WHERE ENAME='KING')
3 |     AND EMPNO IN(SELECT EMPNO FROM EMP E,SALGRADE S WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL AND S.GRADE>3);

```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.041 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
2	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
3	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20

81. Find the total sal given to the MGR?

```
select sum(sal) from emp where empno in  
(select distinct e.mgr from emp e,emp m where e.mgr=m.empno);
```

The screenshot shows the SQL tab of the SSMS interface. The query window contains the following code:

```
select sum(sal) from emp where empno in  
(select distinct e.mgr from emp e,emp m where e.mgr=m.empno);
```

The results pane shows the output:

SUM(SAL)
19275

Below the results, a message indicates "All Rows Fetched: 1 in 0.031 seconds".

82. Find the total annual sal to distribute job wise in the year 81.

```
SELECT * FROM EMP;  
SELECT JOB, SUM(12*SAL) FROM EMP WHERE TO_CHAR(HIREDATE, 'YYYY')='1981' GROUP BY JOB;
```

The screenshot shows the SQL tab of the SSMS interface. The query window contains the following code:

```
SELECT * FROM EMP;  
SELECT JOB, SUM(12*SAL) FROM EMP WHERE TO_CHAR(HIREDATE, 'YYYY')='1981' GROUP BY JOB;
```

The results pane shows the output:

JOB	SUM(12*SAL)
1 PRESIDENT	60000
2 MANAGER	99300
3 ANALYST	36000
4 SALESMAN	67200
5 CLERK	11400

Below the results, a message indicates "All Rows Fetched: 5 in 0.004 seconds".

83. Display total sal employee belonging to grade 3.

The screenshot shows a SQL query window with a red box highlighting the following code:

```
SELECT SUM(SAL)FROM EMP WHERE EMPNO IN (SELECT EMPNO FROM EMP E,SALGRADE S WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL AND S.GRADE=3);
```

Below the query window, the status bar indicates "All Rows Fetched: 1 in 0.006 seconds". The results pane shows a single row:

	SUM(SAL)
1	3100

84. Display the average salaries of all the clerks.

The screenshot shows a SQL query window with a red box highlighting the following code:

```
SELECT AVG(SAL) FROM EMP WHERE JOB='CLERK';
```

Below the query window, the status bar indicates "All Rows Fetched: 1 in 0.006 seconds". The results pane shows a single row:

	AVG(SAL)
1	1037.5

85. List the employeein dept 20 whose sal is >the average sal Of dept 10 emps.

```
select * from emp where deptno =20 and sal >(select avg (sal) from emp
where deptno = 10);
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Script Output x | Query Result 4 x

SQL | All Rows Fetched: 3 in 0.002 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
2	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
3	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20

86. Display the number of employees for each job group deptno wise.

```
select deptno ,job ,count(*) from emp group by deptno,job;
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Script Output x | Query Result 4 x

SQL | All Rows Fetched: 9 in 0.017 seconds

	DEPTNO	JOB	COUNT(*)
1	10	PRESIDENT	1
2	30	MANAGER	1
3	10	MANAGER	1
4	20	MANAGER	1
5	20	ANALYST	2
6	20	CLERK	2
7	30	SALESMAN	4
8	30	CLERK	1
9	10	CLERK	1

87. List the manager no and the number of employees working for those mgrs in the ascending Mgrno.

```
SELECT E.MGR,COUNT(*) FROM EMP E,EMP E_1 WHERE E.MGR=E_1.EMPNO GROUP BY E.MGR ORDER BY E.MGR ASC;
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Script Output x | Query Result 4 x

SQL | All Rows Fetched: 14 in 0.015 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESI...	(n...)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
5	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
6	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
7	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
8	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
9	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
10	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30

88. List the department details where at least two emps are working

```
SELECT DEPTNO,COUNT (*) FROM EMP GROUP BY DEPTNO HAVING COUNT(*)>=2;
```

Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Script Output x | Query Re

SQL | All Rows Fetched: 3 in 0.004 seconds

	DEPTNO	COUNT(*)
1	10	3
2	30	6
3	20	5

89. Display the Grade, Number of emps, and max sal of each grade.

```
select s.grade ,count(*),max(sal) from emp e,salgrade s where e.sal between s.losal and s.hisal
group by s.grade;
```

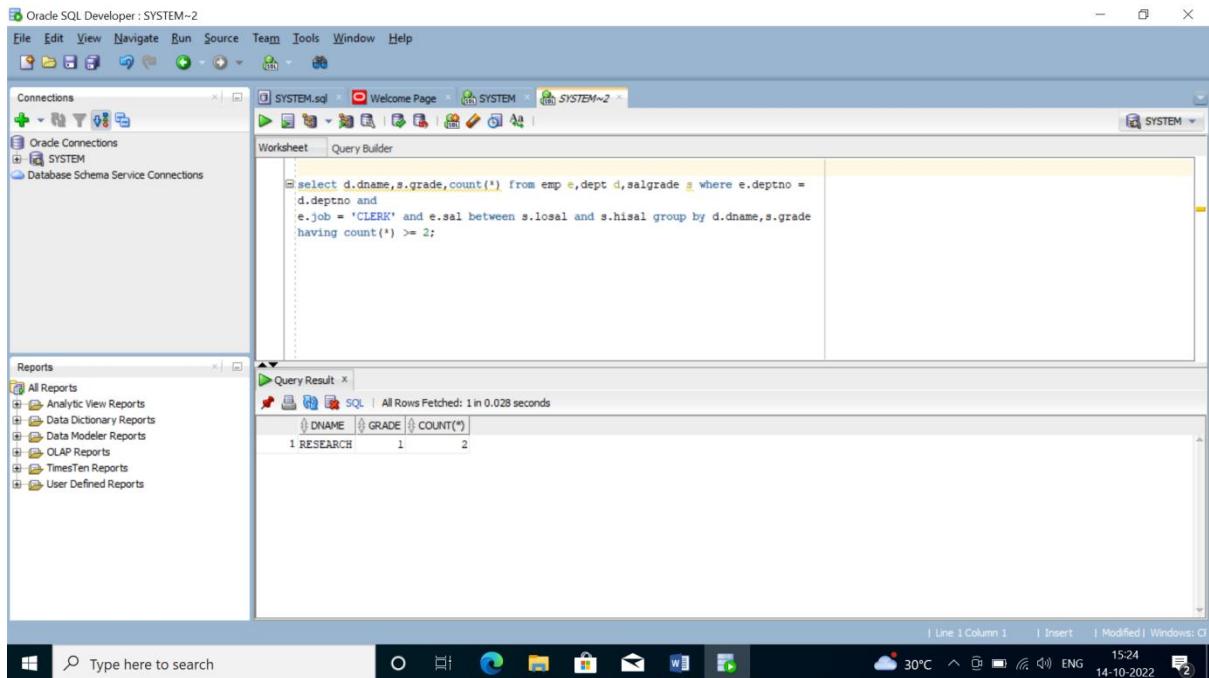
Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Script Output x | Query Re

SQL | All Rows Fetched: 5 in 0.005 seconds

GRADE	COUNT(*)	MAX(SAL)
1	1	1100
2	2	1300
3	3	1600
4	4	3000
5	5	5000

90. Display dname, grade, No. of emps where at least two emps are clerks.

A. select d.dname,s.grade,count(\*) from emp e,dept d,salgrade s where e.deptno = d.deptno and e.job = 'CLERK' and e.sal between s.losal and s.hisal  
group by d.dname,s.grade having count(\*) >= 2;



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL query:

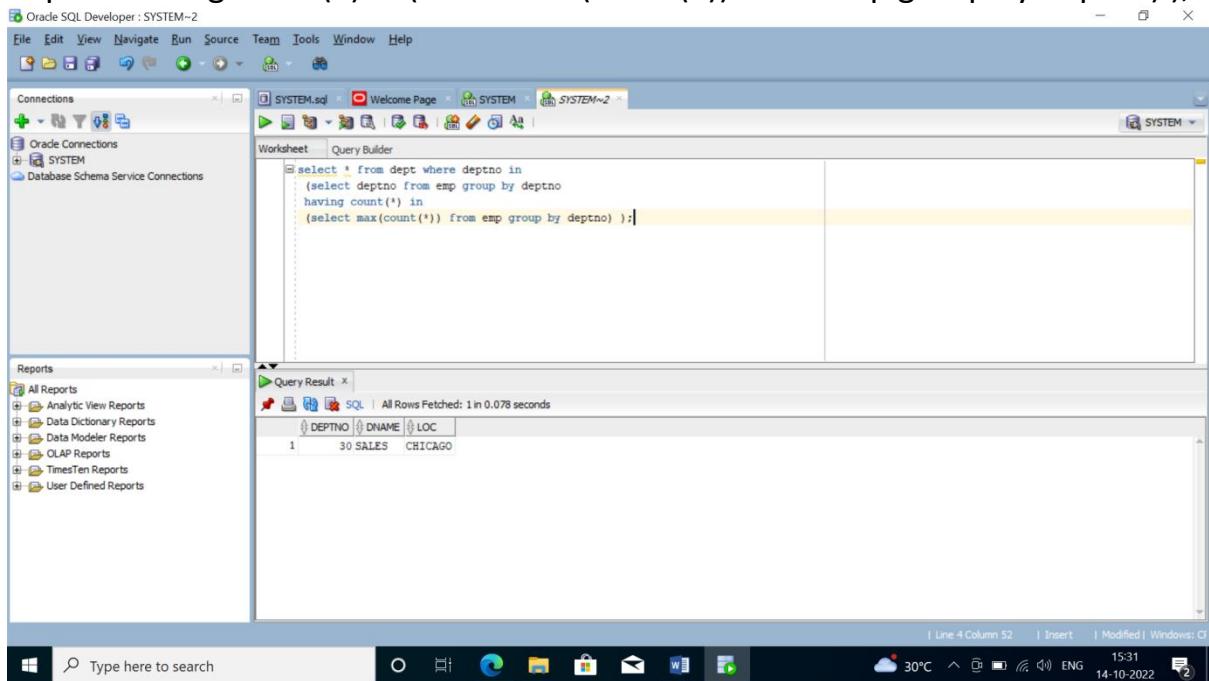
```
@select d.dname,s.grade,count(*) from emp e,dept d,salgrade s where e.deptno = d.deptno and e.job = 'CLERK' and e.sal between s.losal and s.hisal group by d.dname,s.grade having count(*) >= 2;
```

The 'Query Result' tab shows the output of the query:

DNAME	GRADE	COUNT(*)
RESEARCH	1	2

91. List the details of the department where maximum number of emps are working.

A. select \* from dept where deptno in (select deptno from emp group by deptno having count(\*) in (select max(count(\*)) from emp group by deptno) );



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL query:

```
select * from dept where deptno in
(select deptno from emp group by deptno
having count(*) in
(select max(count(*)) from emp group by deptno) );
```

The 'Query Result' tab shows the output of the query:

DEPTNO	DNAME	LOC
30	SALES	CHICAGO

92. Display the emps whose manager name is JONES.

A. select \* from emp where mgr in (select empno from emp where ename = 'JONES');

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, the following SQL code is entered:

```
select * from emp where mgr in
(select empno from emp where ename = 'JONES');
```

In the 'Query Result' tab, the output is displayed as a table:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20

93. List the employees whose salary is more than 3000 after giving 20% increment.

A. SELECT \* FROM EMP WHERE  $(1.2 * \text{SAL}) > 3000$  ;

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, the following SQL code is entered:

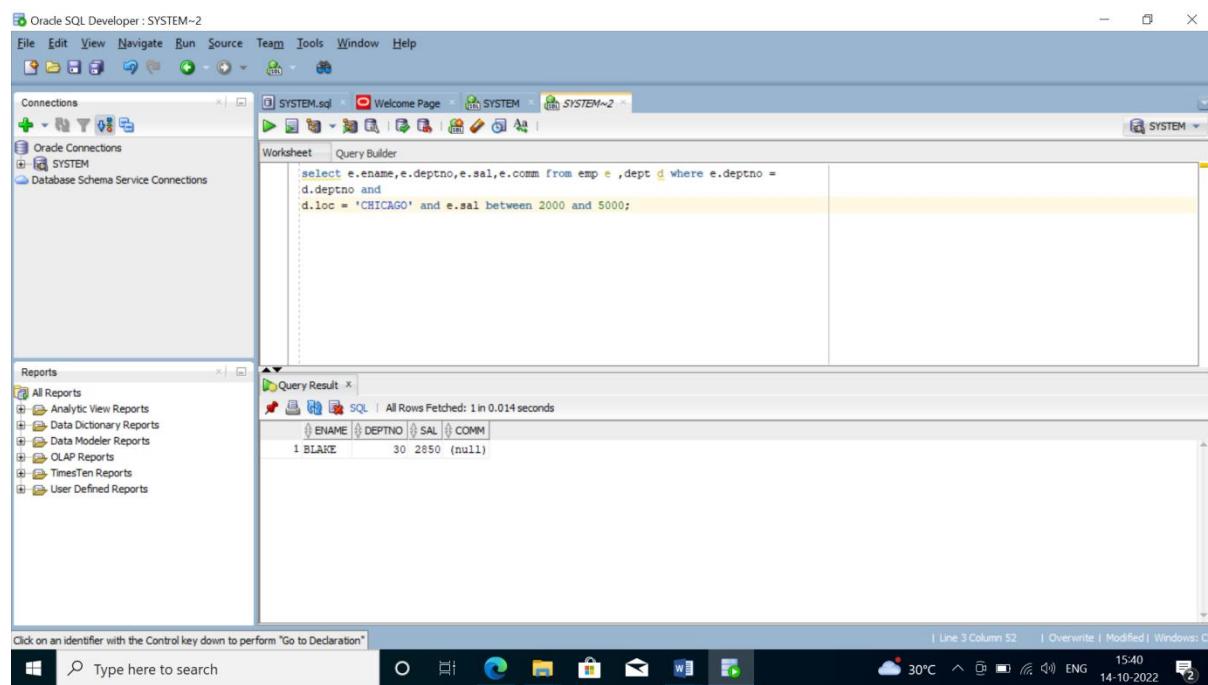
```
SELECT * FROM EMP WHERE  $(1.2 * \text{SAL}) > 3000$  ;
```

In the 'Query Result' tab, the output is displayed as a table:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20

94. List the emps name, dept, sal and comm. For those whose salary is between 2000and 5000 while loc is Chicago.

A. select e.ename,e.deptno,e.sal,e.comm from emp e ,dept d where e.deptno = d.deptno and d.loc = 'CHICAGO' and e.sal between 2000 and 5000;



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the following SQL query:

```
select e.ename,e.deptno,e.sal,e.comm from emp e ,dept d where e.deptno = d.deptno and d.loc = 'CHICAGO' and e.sal between 2000 and 5000;
```

The 'Query Result' tab displays the output of the query:

ENAME	DEPTNO	SAL	COMM
BLAKE	30	2850	(null)

95. List the emps whose sal is greater than his managers salary?

A. select \* from emp w,emp m where w.mgr = m.empno and w.sal > m.sal;

The screenshot shows the Oracle SQL Developer interface. In the Worksheet pane, the following SQL query is run:

```
select * from emp w,emp m where w.mgr = m.empno and w.sal > m.sal;
```

The Query Result pane displays the output:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	EMPNO_1	ENAME_1	JOB_1	MGR_1	HIREDATE_1	SAL_1	COMM_1	DEPTNO_1
1	7788 SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	7566 JONES	MANAGER	7839	02-04-81	2975	(null)	20	
2	7802 FORD	ANALYST	7566	03-12-81	3000	(null)	20	7566 JONES	MANAGER	7839	02-04-81	2975	(null)	20	

96. List the grade, EMP name for the deptno 10 or deptno 30 but sal grade is not 4while they joined the company before '31-dec-82'?

A. select s.grade ,e.ename from emp e,salgrade s where e.deptno in (10,20) and hiredate < ('31-DEC-82') and (e.sal between s.losal and s.hisal and s.grade not in (4));

The screenshot shows the Oracle SQL Developer interface. In the Worksheet pane, the following SQL query is run:

```
select s.grade ,e.ename from emp e,salgrade s where e.deptno in (10,20) and hiredate < ('31-DEC-82') and (e.sal between s.losal and s.hisal and s.grade not in (4));
```

The Query Result pane displays the output:

GRADE	ENAME
1	1 SMITH
2	2 MILLER
3	5 KING

97. List the name, job, dname, location for those who are working as MGRS.

ANS)select e.ename,e.job,d.dname,d.loc from emp e ,dept d where e.deptno = d.deptno and e.empno in (select mgr from emp ) ;

```
select e.ename,e.job,d.dname,d.loc from emp e ,dept d
where e.deptno = d.deptno
and e.empno in (select mgr from emp ) ;
```

ENAME	JOB	DNAME	LOC
KING	PRESIDENT	ACCOUNTING	NEW YORK
CLARK	MANAGER	ACCOUNTING	NEW YORK
JONES	MANAGER	RESEARCH	DALLAS
FORD	ANALYST	RESEARCH	DALLAS
SCOTT	ANALYST	RESEARCH	DALLAS
BLAKE	MANAGER	SALES	CHICAGO

98.List the emps whose mgr name is jones and also list their manager's name?

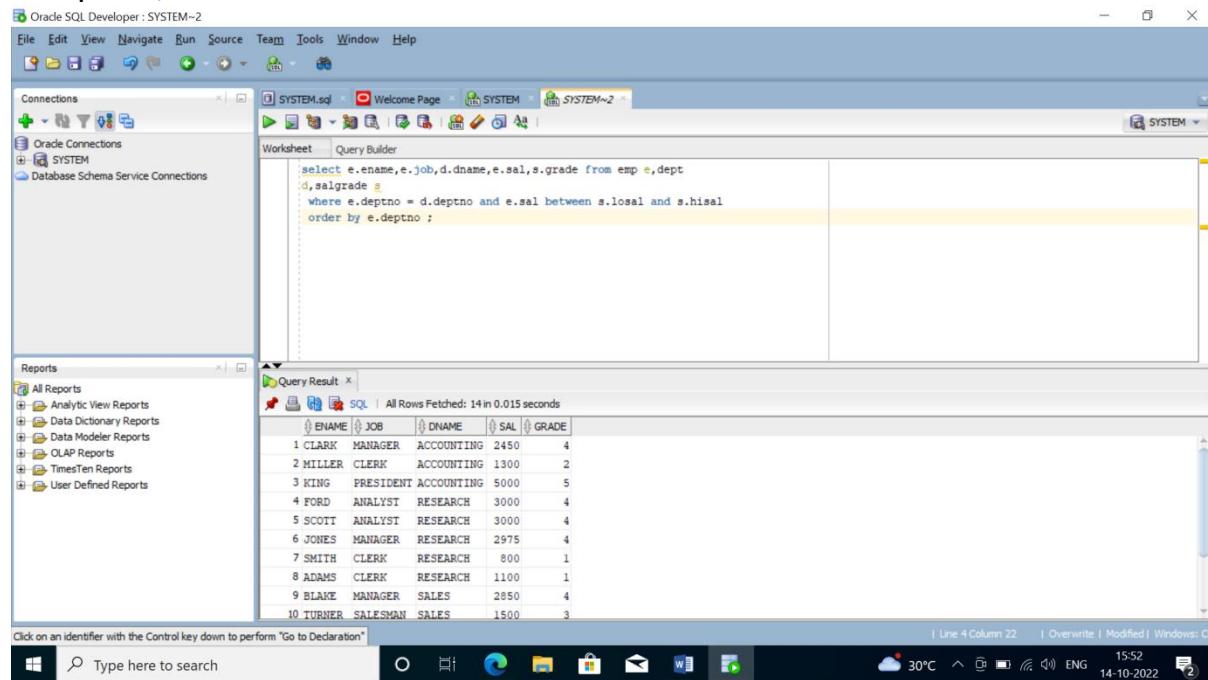
S select w.empno,w.ename,w.job,w.mgr,w.hiredate,w.sal,w.deptno,m.ename  
from emp w ,emp m where w.mgr = m.empno and m.ename = 'JONES';

```
select w.empno,w.ename,w.job,w.mgr,w.hiredate,w.sal,w.deptno,m.ename
from emp w ,emp m
where w.mgr = m.empno and m.ename = 'JONES';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	ENAME_1
7788	SCOTT	ANALYST	7566	19-04-87	3000	20	JONES
7902	FORD	ANALYST	7566	03-12-81	3000	20	JONES

99. List the name, job, dname ,sal, grade dept wise

A. select e.ename,e.job,d.dname,e.sal,s.grade from emp e,dept d,salgrade s where e.deptno = d.deptno and e.sal between s.losal and s.hisal order by e.deptno ;



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL query:

```
select e.ename,e.job,d.dname,e.sal,s.grade from emp e,dept
d,salgrade
where e.deptno = d.deptno and e.sal between s.losal and s.hisal
order by e.deptno ;
```

The 'Query Result' tab shows the output of the query, which is a table with the following data:

ENAME	JOB	DNAME	SAL	GRADE
1 CLARK	MANAGER	ACCOUNTING	2450	4
2 MILLER	CLERK	ACCOUNTING	1300	2
3 KING	PRESIDENT	ACCOUNTING	5000	5
4 FORD	ANALYST	RESEARCH	3000	4
5 SCOTT	ANALYST	RESEARCH	3000	4
6 JONES	MANAGER	RESEARCH	2975	4
7 SMITH	CLERK	RESEARCH	800	1
8 ADAMS	CLERK	RESEARCH	1100	1
9 BLAKE	MANAGER	SALES	2850	4
10 TURNER	SALESMAN	SALES	1500	3

S100. List the emp name, job, sal, grade and dname except clerks and sort on the basisof highest sal.

A. select e.ename,e.job,e.sal,s.grade,d.dname from emp e ,dept d ,salgrade s where e.deptno = d.deptno and e.sal between s.losal and s.hisal and e.job not in('CLERK') order by e.sal desc;

```

select e.ename,e.job,e.sal,s.grade,d.dname from emp e ,dept d
, salgrade s where e.deptno = d.deptno and e.sal between s.loSal
and s.hiSal and
e.job not in('CLERK')
order by e.sal desc;

```

ENAME	JOB	SAL	GRADE	DNAME
KING	PRESIDENT	5000	5	ACCOUNTING
FORD	ANALYST	3000	4	RESEARCH
SCOTT	ANALYST	3000	4	RESEARCH
JONES	MANAGER	2975	4	RESEARCH
BLAKE	MANAGER	2850	4	SALES
CLARK	MANAGER	2450	4	ACCOUNTING
ALLEN	SALESMAN	1600	3	SALES
TURNER	SALESMAN	1500	3	SALES
WARD	SALESMAN	1250	2	SALES
MARTIN	SALESMAN	1250	2	SALES

101. List the emps whose sal is equal to the average of max and minimum

```

101. LIST the emps whose sal is equal to the average of max and minimum

select * from emp where sal =(select (max(sal)+min(sal))/2 from emp);

select sum(sal) from emp where empno in
(select distinct e.mgr from emp e,emp m where e.mgr=m.empno)

```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
-------	-------	-----	-----	----------	-----	------	--------

102. List the no. of emps in each department where the no. is more than 3.

```
select * from (select e.* , count(empno) over(partition by deptno) cn
from emp e) where cn>3;
```

```
102. List the no. of emps in each department where the no. is more than 3.
select deptno, count(*) from emp
group by deptno
having count(empno)>3
order by deptno;

select * from (select e.* , count(empno) over(partition by deptno) cn from emp e) where cn>3;
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	CN
1	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20	5
2	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20	5
3	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20	5
4	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20	5
5	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20	5
6	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30	6
7	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30	6
8	7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30	6
9	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30	6
10	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30	6

103. List the names of depts. Where atleast 3 are working in that department.

```
. List the names of depts. Where atleast 3 are working in that department.
select * from dept;
```

```
select d.DEPTNO, d.DNAME, d.LOC, e.EMPNO, e.ENAME, e.JOB, e.MGR, e.HIREDATE, e.SAL, e.COMM
from dept d,emp e
where e.deptno=d.deptno and e.deptno in
(select deptno from dept where dname in
(select dname from dept d,emp e where e.deptno=d.deptno
group by dname
having count(e.empno)>=3));
```

DEPTNO	DNAME	LOC	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
1	ACCOUNTING	NEW YORK	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)
2	SALES	CHICAGO	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)
3	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)
4	RESEARCH	DALLAS	7566	JONES	MANAGER	7839	02-04-81	2975	(null)
5	RESEARCH	DALLAS	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)
6	RESEARCH	DALLAS	7902	FORD	ANALYST	7566	03-12-81	3000	(null)
7	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17-12-80	800	(null)
8	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300
9	SALES	CHICAGO	7521	WARD	SALESMAN	7698	22-02-81	1250	500
10	SALES	CHICAGO	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400

104. List the managers whose sal is more than his employess avg salary.

The screenshot shows a SQL query window with a red box highlighting the query code. The code selects employees whose manager's salary is greater than the average salary of their employees. Below the code is a red box highlighting the resulting table of 6 rows.

```
select * from (select * from emp where empno in(select mgr from emp))m where m.sal > (select avg (e.sal) from emp e,emp m where e.mgr=m.empno);  
m.EMPNO, m.ENAME, m.JOB, m.MGR, m.HIREDATE, m.SAL, m.COMM, m.DEPTNO
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | SQL | All Rows Fetched: 6 in 0.003 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
3	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
4	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
5	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
6	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10

105. List the name,salary,comm. For those employees whose net pay is greater than or equal to any other employee salary of the company

The screenshot shows the Oracle SQL Developer interface. A red box highlights the SQL query in the top editor:

```
select ename ,sal,comm from emp where sal+nvl(comm,0)>=sal;
```

The bottom window shows the results of the query:

	ENAME	SAL	COMM
1	KING	5000	(null)
2	BLAKE	2850	(null)
3	CLARK	2450	(null)
4	JONES	2975	(null)
5	SCOTT	3000	(null)
6	FORD	3000	(null)
7	SMITH	800	(null)
8	ALLEN	1600	300
9	WARD	1250	500
10	MARTIN	1250	1400

The status bar at the bottom right indicates "Line 90 Column 61 | Ins".

106. List the emp whose sal<his manager but more than any other manager.

The screenshot shows the Oracle SQL Developer interface. A red box highlights the SQL query in the top editor:

```
select e.sal,e.ename,m.ename,m.sal from emp e,emp m where e.mgr=m.empno
and e.sal<m.sal
;
List the emp whose sal<his manager but more than any other manager.
```

The bottom window shows the results of the query:

	EMNAME	EMSAL	MGRNAME	MGSAL
1	BLAKE	2850	KING	5000
2	CLARK	2450	KING	5000
3	JONES	2975	KING	5000

107. List the employee names and his average salary department wise.

The screenshot shows the Oracle SQL Developer interface. A red box highlights the SQL query in the top pane:

```
    : group by deptno);
select ename ,trunc(avg(sal) over(partition by deptno ) )avgsal,deptno from emp;
```

The bottom pane displays the results of the query, also enclosed in a red box. The results show employees from department 10 with an average salary of 2916.

	ENAME	AVGSAL	DEPTNO
1	CLARK	2916	10
2	MILLER	2916	10
3	KING	2916	10
4	FORD	2175	20
5	SCOTT	2175	20
6	JONES	2175	20
7	SMITH	2175	20

108. Find out least 5 earners of the company.

```
select * from emp;
Select * from emp e where 5> (select count(*) from emp where
e.sal >sal);
```

Script Output x | Query Result 6 x | Query Result 7 x

All Rows Fetched: 5 in 0.017 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
2	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
3	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
4	7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20
5	7900	JAMES	CLERK	7698	03-12-81	950	(null)	30

109. List the Name, Salary, Comm and Net Pay is more than any other employee.

```
Select e.ename,e.sal,e.comm,nvl2(comm,sal+comm,sal)
NETPAY
from emp e
where nvl2(comm,sal+comm,sal) > any (select sal from emp
where empno =e.empno) ;
```

Script Output x | Query Result 6 x | Query Result 7 x

All Rows Fetched: 3 in 0.003 seconds

	ENAME	SAL	COMM	NETPAY
1	ALLEN	1600	300	1900
2	WARD	1250	500	1750
3	MARTIN	1250	1400	2650

110.List the Enames who are retiring after 31-Dec-89 the max Job period is 20Y.

The screenshot shows the Oracle SQL Developer interface. In the top-left query editor, the following SQL code is written:

```
select * from emp;  
select ename from emp where add_months(hiredate,240) > '31-DEC-89';
```

The second query is highlighted with a red rectangle. In the bottom-right results window, the output is displayed as a table:

ENAME
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
ALLEN
WARD
MARTIN

A red rectangle also highlights the results table. The status bar at the bottom indicates: "All Rows Fetched: 14 in 0.004 seconds".

111.List the emps whose first 2 chars from Hiredate=last 2 characters of Salary.

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing three SQL statements. The third statement, which is highlighted with a red border, is:

```
select * from emp  
where substr(hiredate,1,2) = substr(sal,length(sal)-1,length(sal));  
  
Select substr(hiredate,1,2),substr (sal,length(sal)-1,length(sal)) ,sal from emp  
where substr(hiredate,1,2)=substr (sal,length(sal)-1,length(sal));  
B select * from emp where sal like '%substr(hiredate,1,2)';
```

In the bottom-right pane, there is a results grid with the following columns:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO

A red box highlights the entire results grid.

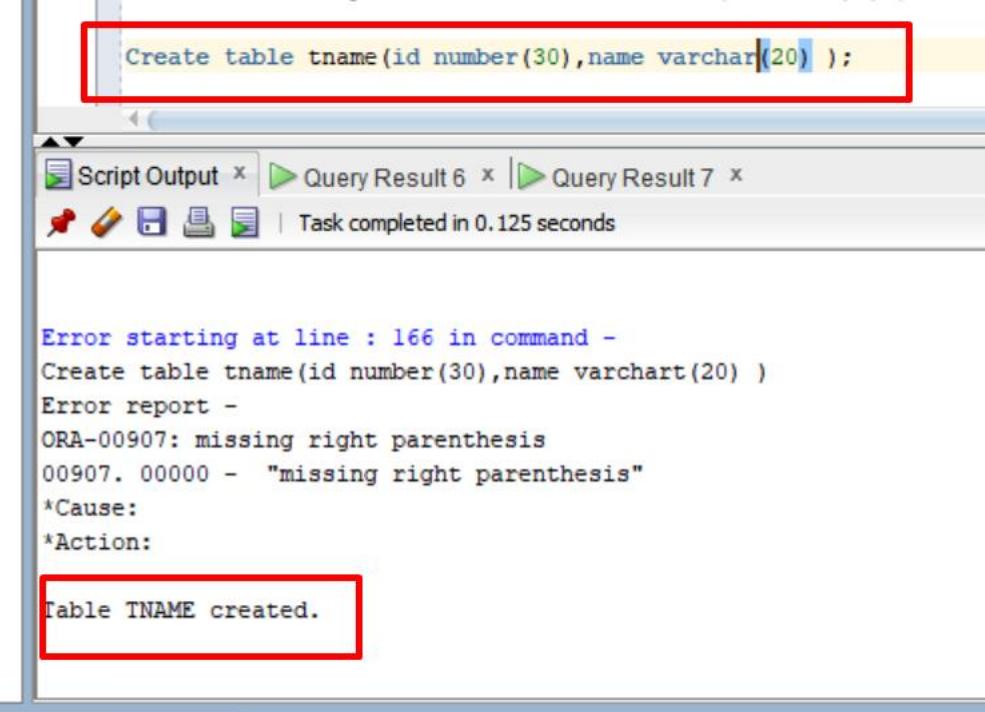
112.write a definition of how to create a table?

Create a table by using DDL command, a table contain column and row , a column contain their own data type .

A table is database object.

Syntax

```
Create table tname(id number(30),name varchar(20) );
```



The screenshot shows a SQL developer interface. A red box highlights the command:

```
Create table tname(id number(30),name varchar(20) );
```

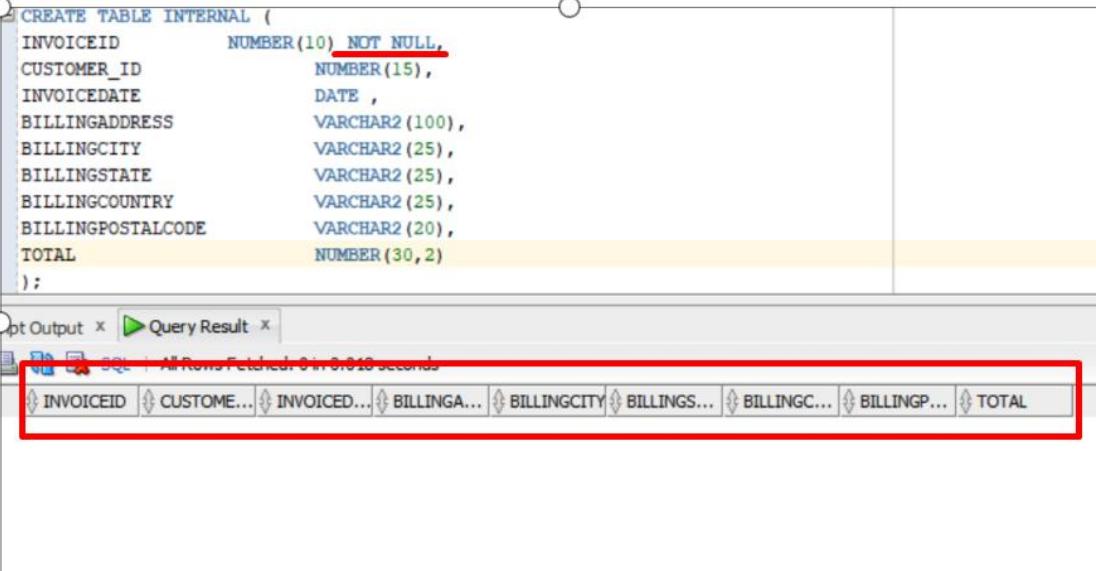
Below the command, the status bar shows "Task completed in 0.125 seconds".

An error message follows:

```
Error starting at line : 166 in command -
Create table tname(id number(30),name varchar(20) )
Error report -
ORA-00907: missing right parenthesis
00907. 00000 -  "missing right parenthesis"
*Cause:
*Action:
```

A red box highlights the message "Table TNAME created.".

113.Create the Table by using the not null condition.



The screenshot shows a SQL developer interface. A red box highlights the NOT NULL constraint in the column definition:

```
CREATE TABLE INTERNAL (
    INVOICEID      NUMBER(10) NOT NULL,
    CUSTOMER_ID    NUMBER(15),
    INVOICEDATE    DATE ,
    BILLINGADDRESS VARCHAR2(100),
    BILLINGCITY    VARCHAR2(25),
    BILLINGSTATE   VARCHAR2(25),
    BILLINGCOUNTRY VARCHAR2(25),
    BILLINGPOSTALCODE VARCHAR2(20),
    TOTAL          NUMBER(30,2)
);
```

The status bar shows "All Rows Fetched: 0 in 0.010 seconds". A red box highlights the header of the query result table:

INVOICEID	CUSTOME...	INVOICED...	BILLINGA...	BILLINGCITY	BILLINGS...	BILLINGC...	BILLINGP...	TOTAL
-----------	------------	-------------	-------------	-------------	-------------	-------------	-------------	-------

114.Create the tables by using the primary key and foreign key?

```
alter table tname add constraint pk_tname primary key(id);
create table link (id number(20),name varchar(20),tid number(20),
constraint pk_link_id primary key (id),
constraint fk_tid foreign key(tid) references tname(id));
```

Script Output x | Query Result 6 x | Query Result 7 x  
Task completed in 0.045 seconds  
00907. 00000 - "missing right parenthesis"  
\*Cause:  
\*Action:  
Table TNAME created.  
Table TNAME altered.  
Table LINK created.

115.After creating the table write different types of altered conditions?

Add -- alter table link add address varchar(20);

Rename -- alter table link rename column address to location;

Drop -- alter table link drop column location ;

Modify ---alter table link modify name varchar(50);

```
115.After creating the table write different types of altered condit
alter table link add address varchar(20);
alter table link rename column address to location;
alter table link drop column location ;
alter table link modify name varchar(50);
```

Script Output x | Query Result 6 x | Query Result 7 x  
Task completed in 0.074 seconds  
Table LINK created.  
Table LINK altered.  
Table LINK altered.  
Table LINK altered.

116.Create the Table by using the default condition.

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
insert into loca values (1,'kadapa',115881);
insert into loca (id,pincode) values(2,8762209);

select * from loca;
```

The second insert statement is highlighted with a red box. In the bottom-right pane, there are three tabs: "Script Output", "Query Result 6", and "Query Result 7". The "Query Result 7" tab is selected and shows the following table data:

ID	LNAME	PINCODE
1	kadapa	115881
2	tirupati	8762209

117. Create the Table by using the check condition.

```
create table check_table (id number(30),name varchar(30),age int check (age<=15));
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
117.Create the Table by using the check condition.
create table check_table (id number(30),name varchar(30),age int check (age<=15));
```

The entire script is highlighted with a red box. In the bottom-right pane, there are three tabs: "Script Output", "Query Result 6", and "Query Result 7". The "Query Result 7" tab is selected and shows the following error message:

Error at Command Line : 184 Column : 34  
Error report -  
SQL Error: ORA-00917: missing comma  
00917. 00000 - "missing comma"  
\*Cause:  
\*Action:

1 row inserted.

Table CHECK\_TABLE created.

118.What is an index. What different types of indexes?

Indexes: - Index is used to increase performance of table,

When we create index on a table then we can retrieve data faster.

By creating index it will sort out the row in a serial wise.

It is a database object.

Types of indexes: -simple index, complex index, unique index, functional index.

Syntax: - create index index\_name on tablename (columnname).

119.How to create an index in SQL?

The screenshot shows a SQL developer interface. A red box highlights the SQL command: `create index ix_loca on loca(id);`. Below the command, an error message is displayed: `00917. 00000 - "missing comma"`, followed by cause and action details. The output pane shows the creation of a table and an index, both highlighted with red boxes.

```
create index ix_loca on loca(id);
00917. 00000 - "missing comma"
*Cause:
*Action:

1 row inserted.

Table CHECK_TABLE created.

Index IX_LOCA created.
```

120.What difference between Bitmap Index and index?

Each bit in the bitmap corresponds to a possible rowid, and if the bit is set, it means that the row with the corresponding rowid contains the key value. A mapping function converts the bit position to an actual rowid, so that the bitmap index provides the same functionality as a regular index. If the number of different key values is small, bitmap indexes save space.

121.How to create a multi-column index in SQL

Ans:-

Multi column index (or) Complex Index :-

It is used to create a index on multiple columns of a table.

Syntax :- create index index\_name on table\_name ( col1, col2 ,col3.....);

122.How to create a bit map index?

Ans:-

Bitmap Index :-

It is used to create a bit map index on column.

Syntax :- create bitmap index index\_name on emp ( empno );

123.What is the difference between a Clustered and Non-Clustered Index?

Ans:-

Clustered:- 1. A clustered index is used to define the order or to sort the table or arrange the data by alphabetical order just like a dictionary.

2. It is faster than a non-clustered index

Non-Clustered:- 1. A non-clustered index collects the data at one place and records at another place.

2. It is slower than the clustered index.

124. write steps to export data into a table?

Ans:-

Based on the query Exporting the record query is

Go to oracle SQL Developer sheet and select \* from emp;

--->Then query results in display the emp table

---> Then right clack and Export icon then clack the export icon then

Export Wizard step 1 of 2 Window is open in format column select  
the excel 2003+(xlsx)

---> Then go to Browse and select then file patch name is saved Then  
clack next it will open next Export

summary window then clack to finish Export data is Completed

Save the data

#### EXPORTING THE DATA INTO SQL DEVELOPER INTO LOCALMECHINE

125. How to delete duplicate records in a table?

Ans :-

To remove duplicate rows from a result set, you use the DISTINCT operator in the SELECT clause as follows:

Syntax :-SELECT DISTINCT column1, column2, ... FROM table1;

Syntax :- SELECT DISTINCT salary FROM employees ORDER BY salary DESC;

---> Use DISTINCT operator in the SELECT clause to remove duplicate rows from the result set.

126. How many types to find duplicate records in a table?

Ans :-

1. Using the GROUP BY clause to group all rows by the target column(s) – i.e. the column(s) you want to check for duplicate values on.
- 2.Using the COUNT function in the HAVING clause to check if any of the groups have more than 1 entry; those would be the duplicate values.

127. How to find duplicate records count in the table?

Ans:-

The screenshot shows the Oracle SQL Developer interface. At the top, there are two tabs: 'Script Output' and 'Query Result 6'. Below them is a toolbar with icons for 'SQL', 'All Rows Fetched', and a timer showing '0.002 seconds'. The main area contains two queries:

```
select count(distinct job)-count(job) from emp;  
select count(*)-count (distinct empno ) from dummy;  
select count (distinct empno )from dummy;
```

The second query is highlighted with a red box. The results are displayed in a table:

COUNT(*)-COUNT(DISTINCTEMPNO)
14

The result value '14' is also highlighted with a red box.

128.write steps to import data into a table?

Ans:-

SQL developer method

Steps to import data from source data

Click on schema → click on table name → right click → import → browse file → select source file → check the data and next → then select or matched the columns check data → finish.

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, several tabs are open, including 'SYSTEM.sql', 'prac.sql', 'emp.sql', 'KING', 'IND', 'Team1.sql', and 'Relational\_1 (Untitled\_1)'. The main workspace is titled 'Worksheet' and contains the following SQL code:

```
insert into mediatype values(2,'Protected AAC audio file');
insert into mediatype values(3,'Protected MPEG-4 video file');
insert into mediatype values(4,'Purchased AAC audio file');
insert into mediatype values(5,'AAC audio file');

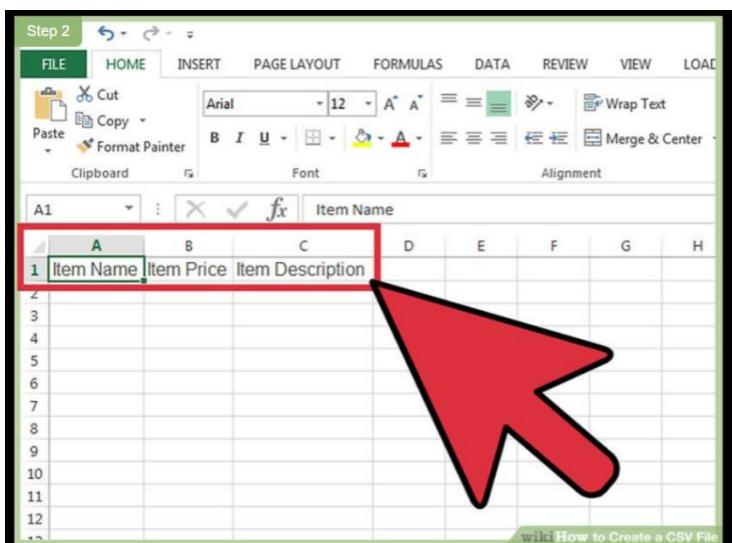
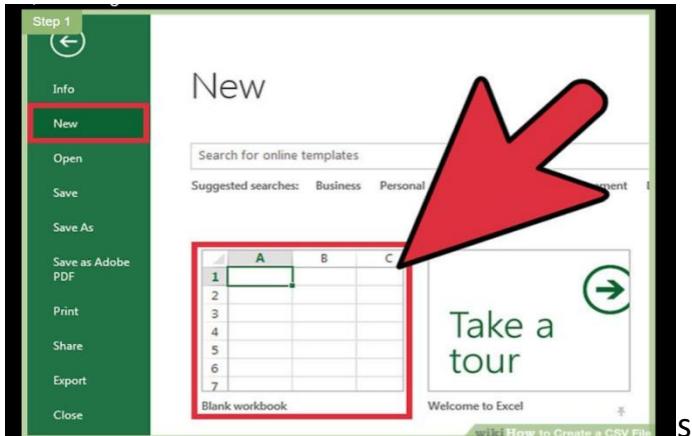
select * from artist;
```

An 'Import Data' dialog box is displayed in the center, showing the message: 'Import Data. Task successful and import committed.' with an 'OK' button. Below the dialog, a table named 'ARTISTID' is visible, showing columns 'ARTISTID' and 'NAME'. The status bar at the bottom left indicates the date and time: '-15.08.31' and '11-15.43.20'.

129.What is a CSV file how to create CSV?

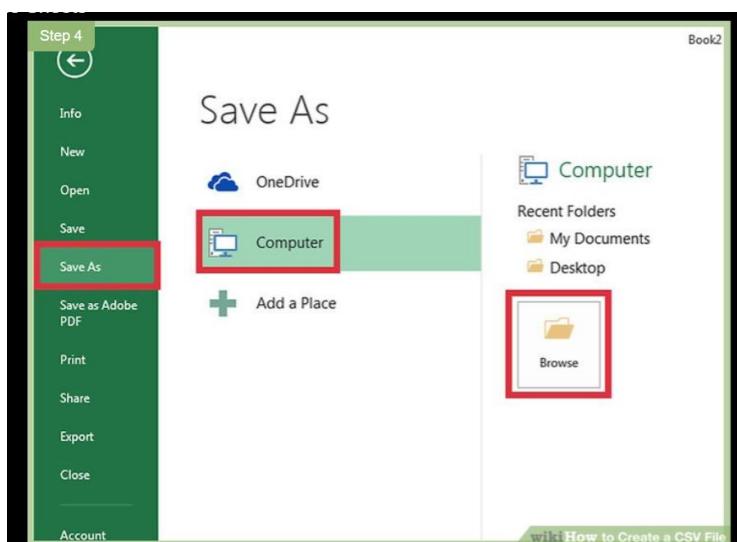
Ans:-

- 1.CSV stands for Comma Separated Values.
2. A CSV file is a plain text file that stores tables and spreadsheet information.
3. The contents are often a table of text, numbers, or dates.
- 4.CSV files can be easily imported and exported using programs that store data in tables.



Step 3

Item Name	Item Price	Item Description
Pencil	\$1	School Items
Sharpener	\$1	School Items
Notebook	\$2	School Items
Bag	\$8	School Items





Type a name for your CSV file, then select "Save." You have now created a CSV file, and commas will automatically be added to the file to separate each field.

130. Write different types of data loading methods with ex?

### LOAD DATA BY USING DIFFERENT METHODS

Type 1: - insert method

This are two types 1. direct method

2. reference method.

The screenshot shows a SQL query window in SSMS. The code is as follows:

```
select * from mediatype;
truncate table mediatype;

insert into mediatype values(1,'MPEG audio file');
insert into mediatype values(2,'Protected AAC audio file');
insert into mediatype values(3,'Protected MPEG-4 video file');
insert into mediatype(mediatypeid,name) values(4,'Purchased AAC audio file');
insert into mediatype (mediatypeid,name) values(5,'AAC audio file');
```

The first four insert statements are highlighted with a green box and labeled "reference method". The last two insert statements are highlighted with a red box and labeled "direct method".

Below the code, the "Script Output" tab shows the results of the execution:

```
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
```

The status bar at the bottom right indicates "Line 24 Column 1 | Insert | Modified".

Type2

SQL developer method

Steps to import data from source data

Click on schema -click on table- click right click- browse file -select source file- check the data and next-then select or matched the columns- check data-s finish.

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, several tabs are open, including 'SYSTEM.sql', 'prac.sql', 'emp.sql', 'KING', 'IND', 'Team1.sql', and 'Relational\_1 (Untitled\_1)'. The main area is a 'Worksheet' tab showing a query builder with the following SQL code:

```

    insert into mediatype values(2,'Protected AAC audio file');
    insert into mediatype values(3,'Protected MPEG-4 video file');
    insert into mediatype values(4,'Purchased AAC audio file');
    insert into mediatype values(5,'AAC audio file');

    select * from artist;
  
```

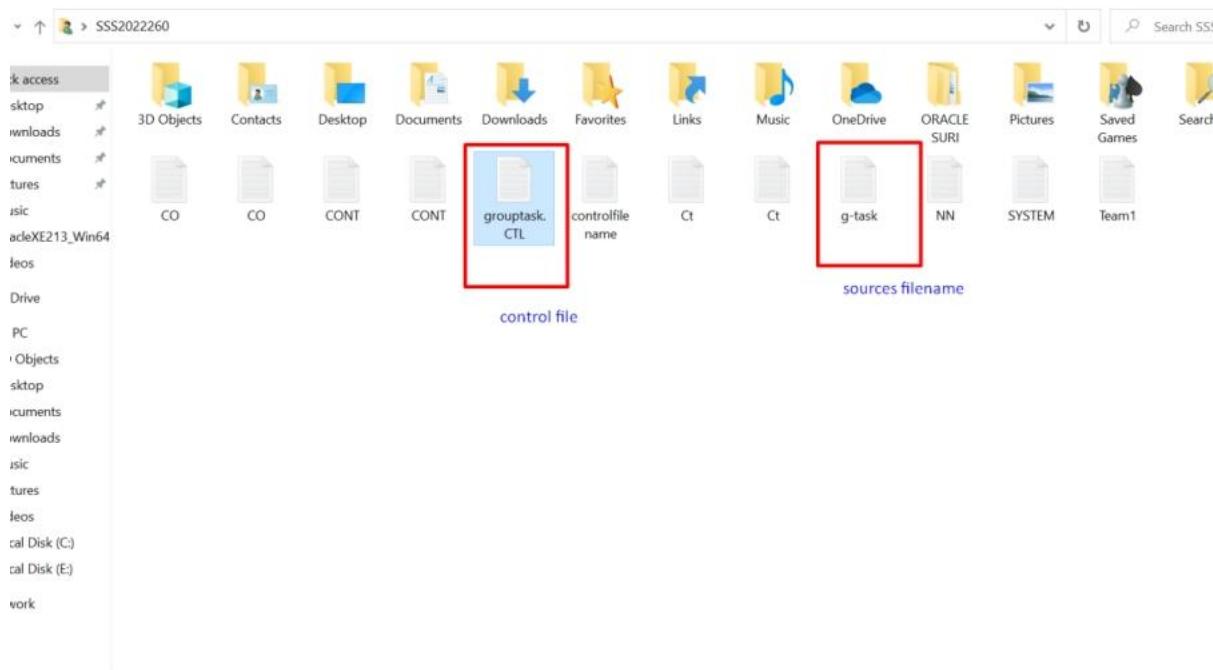
An 'Import Data' dialog box is displayed, stating 'Import Data. Task successful and import committed.' with an 'OK' button. Below the worksheet, there is a table preview for the 'Artist' table with columns 'ARTISTID' and 'NAME'.

Type3

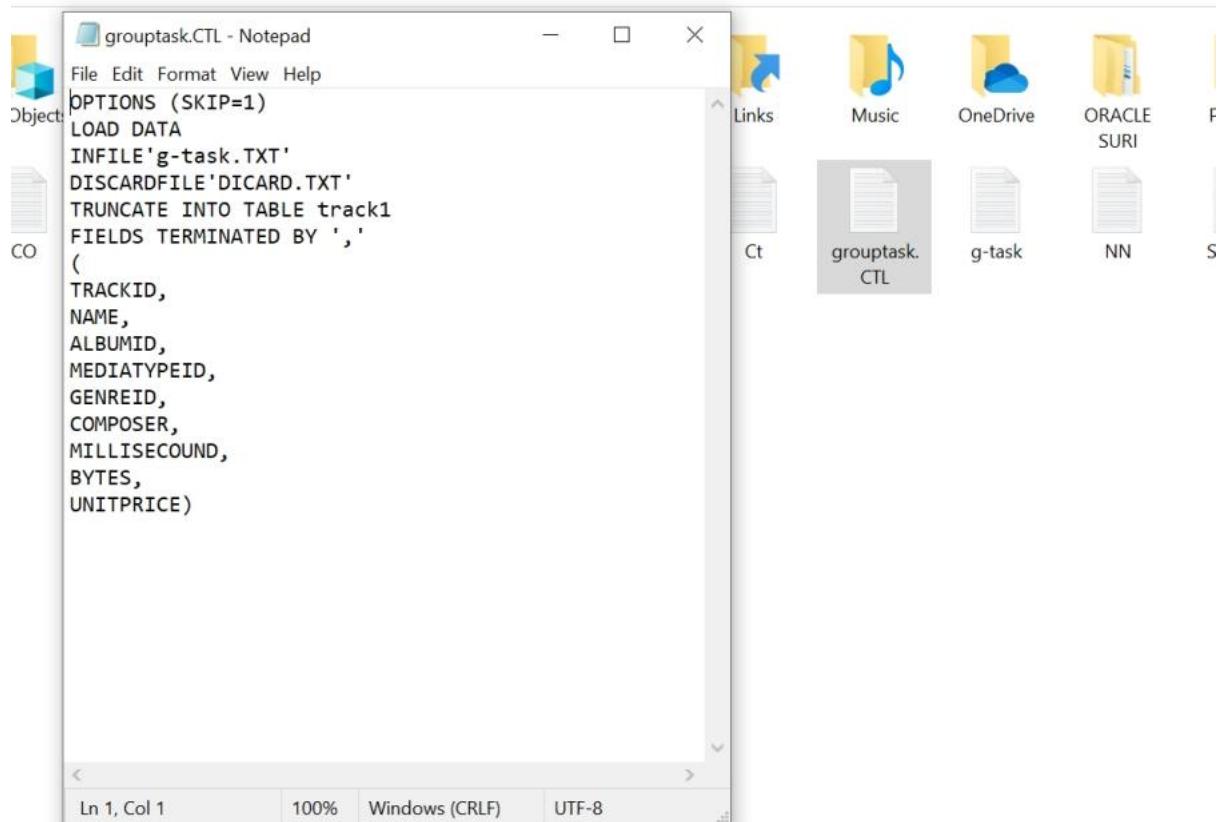
SQL star loader method

Steps followed by to insert data in sql star loader method

Create a files in admin path 1 is control file 2 is sources file



## Control file



The screenshot shows a Windows desktop environment. On the left, a Notepad window is open with the title "grouptask.CTL - Notepad". The content of the window is a SQL control file (CTL) script:

```
OPTIONS (SKIP=1)
LOAD DATA
INFILE 'g-task.TXT'
DISCARDFILE 'DICARD.TXT'
TRUNCATE INTO TABLE track1
FIELDS TERMINATED BY ','
(
TRACKID,
NAME,
ALBUMID,
MEDIATYPEID,
GENREID,
COMPOSER,
MILLISECOND,
BYTES,
UNITPRICE)
```

On the right, a file explorer window is visible, showing a folder structure with icons for Links, Music, OneDrive, ORACLE SURI, and several files including "grouptask.CTL", "g-task", and "NN".

## Sources file

This file contain data.

```
SSS2022260

g-task - Notepad
File Edit Format View Help
TRACKID,NAME,ALBUMID,EDI
3D Object: TRACKID,NAME,ALBUMID,EDI
CO TRACKID,NAME,ALBUMID,EDI
64 TRACKID,NAME,ALBUMID,EDI

323,"Dig-Dig, Lambe-Lambe (Ao Vivo)",29,1,9,Cassiano Costa/Cintia Maviane/J.F Pereré,29,1,9,Augusto Conceição/Chiclete Com Banana,198661,6643207,0.99
325,TriboTchan,29,1,9,Cal Adan/Paulo Levi,194194,6507950,0.99
326,"Tapa Aqui, Descobre Ali",29,1,9,Paulo Levi/W. Rangel,188630,6327391,0.99
327,Daniela,29,1,9,Jorge Cardoso/Pierre Onasis,230791,7748006,0.99
328,Bate Lata,29,1,9,Fábio Nolasco/Gal Sales/Ivan Brasil,206733,7034985,0.99
329,Garotas do Brasil,29,1,9,"Garay, Ricardo Engels/Luca Predabom/Ludwig, Car Levada do Amor (Ailoviu)",29,1,9,Luiz Wanderley/Paulo Levi,190093,6457752,0.99
330,Levada do Amor (Ailoviu),29,1,9,Luiz Wanderley/Paulo Levi,190093,6457752,0.99
331,Lavadeira,29,1,9,"Do Vale, Valverde/Gal Oliveira/Luciano Pinto",214256,72
332,Reboladeira,29,1,9,Cal Adan/Ferrugem/Julinho Carioca/Triona Ni Dhomhnaill
333,É que Nessa Encarnação Eu Nasci Manga,29,1,9,Lucina/Luli,196519,6568081,0
334,Reggae Tchan,29,1,9,"Cal Adan/Del Rey, Tension/Edu Casanova",206654,69313
335,My Love,29,1,9,Jauperi/Zeu Góes,203493,6772813,0.99
336,Latinha de Cerveja,29,1,9,Adriano Bernandes/Edmar Neves,166687,5532564,0.99
337,You Shook Me,30,1,1,J B Lenoir/Willie Dixon,315951,10249958,0.99
338,I Can't Quit You Baby,30,1,1,Willie Dixon,263836,8581414,0.99
339,Communication Breakdown,30,1,1,Jimmy Page/John Bonham/John Paul Jones,192
340,Dazed and Confused,30,1,1,Jimmy Page,491920,13035765,0.99
341,The Girl I Love She Got Long Black Wavy Hair,30,1,1,Jimmy Page/John Bonham/John Paul Jones,192
342,What is and Should Never Be,30,1,1,Jimmy Page/Robert Plant,260675,8497116
343,Communication Breakdown(2),30,1,1,Jimmy Page/John Bonham/John Paul Jones,192
344,Travelling Riverside Blues,30,1,1,Jimmy Page/Robert Johnson/Robert Plant,192
345,Whole Lotta Love,30,1,1,Jimmy Page/John Bonham/John Paul Jones/Robert Plant,192
346,Somethin' Else,30,1,1,Bob Cochran/Sharon Sheeley,127869,4165650,0.99
```

```
ted 243 KB

25-09-2022 19:47    <DIR>          Favorites
11-10-2022 16:39          249,103 g-task.txt   SOURCES FILE
11-10-2022 17:16          342 grouptask.CTL.log
11-10-2022 16:43          226 grouptask.txt   CONTROL FILE
23-09-2022 19:47    <DIR>          Links
23-09-2022 19:47    <DIR>          Music
30-09-2022 17:46          163 NN.txt
23-09-2022 19:50    <DIR>          OneDrive
28-09-2022 13:43    <DIR>          ORACLE SURI
27-09-2022 15:26    <DIR>          Pictures
23-09-2022 19:47    <DIR>          Saved Games
23-09-2022 19:48    <DIR>          Searches
04-10-2022 16:18          10,771 SYSTEM.sql
11-10-2022 14:47          1,619 Team1.sql
03-10-2022 09:45    <DIR>          Videos
     13 File(s)      267,750 bytes
     16 Dir(s)  56,186,871,808 bytes free

C:\Users\SSS2022260>SQLLDR C##TEAM1/TEAM1 CONTROL=GROUPTASK.TXT

SQL*Loader: Release 21.0.0.0.0 - Production on Tue Oct 11 17:20:57 2022
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle and/or its affiliates. All rights reserved.

Path used:      Conventional
Commit point reached - logical record count 250
Commit point reached - logical record count 500

Table TRACK1:
  219 Rows successfully loaded.                                ROW LOADED

Check the log file:
  GROUPTASK.log
```

These are type to import data into database.

131. .Write a query to insert the DATE records in the table?

A. INSERT INTO COACH VALUES ('110','RAHUL');

The screenshot shows a SQL development environment with a query editor and a results pane. In the query editor, a red arrow points to the query text: `INSERT INTO COACH VALUES ('110','RAHUL');`. In the results pane, another red arrow points to the output message: `1 row inserted.`. The interface includes tabs for Script Output, Query Result, and Query Result 1, along with a status bar indicating the task completed in 0.121 seconds.

132. Write a Query to insert time records in a table?

```
INSERT INTO TIMES VALUES(to_date(sysdate, 'yyyy/mm/dd hh24:mi:ss'),01);
CREATE TABLE TIMES(
IN_TIME TIMESTAMP,
SLNO NUMBER(4)
);
select*from times;
```

Output x  Query Result x

  SQL | All Rows Fetched: 1 in 0.03 seconds

IN_TIME	SLNO
22-10-14 12:00:00.000000000 AM	1

133. Write a query to insert default constrain records in a table?

```
create table sreekar(idno number,name varchar(20) default 'chandragiri');
insert into sreekar values(4,'tirupati');
insert into sreekar values(5,'hyderabad');
insert into sreekar(idno) values(9);
select*from sreekar;
```

The screenshot shows a database interface with two tabs: 'Output' and 'Query Result'. The 'Query Result' tab is active, displaying the SQL code and its execution results. The results show three rows inserted into the 'sreekar' table, with the 'NAME' column defaulting to 'chandragiri' for the third row.

IDNO	NAME
4	tirupati
5	hyderabad
9	chandragiri

134. .Write a query to update all attribute records in the table?

```
select*from sreekar;  
update sreekar set idno='10',name='malasyia';
```

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'Script Output' and 'Query Result'. The 'Query Result' tab is active, indicated by a green play button icon. Below the tabs, it says 'SQL | All Rows Fetched: 3 in 0.002 seconds'. The results are displayed in a table with two columns: 'IDNO' and 'NAME'. The data shows three rows, all with the value '10' in the IDNO column and 'malasyia' in the NAME column.

IDNO	NAME
10	malasyia
10	malasyia
10	malasyia

135 .How to find duplicate records count in the table?

```
SELECT COUNT(DISTINCT job),  
       job  
  FROM emp  
 GROUP BY job;
```

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'Script Output' and 'Query Result'. The 'Query Result' tab is active, indicated by a green play button icon. Below the tabs, it says 'SQL | All Rows Fetched: 5 in 0.005 seconds'. The results are displayed in a table with two columns: 'COUNT(JOB)' and 'JOB'. The data shows five rows, where the count of jobs is 1 for PRESIDENT, 2 for ANALYST, 3 for CLERK, 4 for SALESMAN, and 3 for MANAGER.

COUNT(JOB)	JOB
1	MANAGER
2	ANALYST
3	CLERK
4	SALESMAN
5	PRESIDENT

136. Write a query to delete records in a table?

```
truncate table emp;
select*from emp;
```

Output | Query Result | All Rows Fetched: 0 in 0.005 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
-------	-------	-----	-----	----------	-----	------	--------

137. Write a query to delete records in an attribute?

```
delete from emp where deptno=10;
select*from emp;
```

Output | Query Result | All Rows Fetched: 11 in 0.002 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-09-81	1500	0	30
7876	ADAMS	CLERK	7788	23-05-87	1100	(null)	20

138. Write difference between equi joint and non equi joint?

Equi Join and Non-Equi Joins are types of Inner Joins. Equi Join in SQL is used to retrieve data from multiple tables using an equality condition with the WHERE clause. Non-Equi in SQL is used to retrieve data from multiple tables using any other operator except the equality condition.

139. .Write about delete?

DELETE: 1.Delete is a DML command

2. It is a non-autocommit command.
3. It is used to delete a particular record in a table.
4. we can rollback the data after deletion.

140. Write about Analytics Function.

Analytical function: Analytical functions in sql explains various analytical functions with suitable examples.

1. CUME\_DIST
2. LEAD,LAG
3. RANK,DENSE\_RANK,ROW\_NUMBER
4. FIRST, LAST
5. FIRST\_VALUE, LAST\_VALUE, NTH\_VALUE
6. NTILE

141. List the details of the emps whose Salaries more than the employee BLAKE.

A. Select \* from emp where sal > (select sal from emp where ename='BLAKE');

```
select * from emp where sal > (select sal from emp where ename='BLAKE');
```

Script Output x Query Result x  
SQL | All Rows Fetched: 4 in 0.006 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7566 JONES	MANAGER	7839 02-04-81	2975	(null)		20
2	7788 SCOTT	ANALYST	7566 19-04-87	3000	(null)		20
3	7902 FORD	ANALYST	7566 03-12-81	3000	(null)		20
4	7839 KING	PRESIDENT	(null) 17-11-81	5000	(null)		10

142. Display the employees details to get the particular maximum salary employee?

A. SELECT \* FROM EMP WHERE SAL=(SELECT MAX(SAL) FROM EMP);

```
SELECT * FROM EMP WHERE SAL=(SELECT MAX(SAL) FROM EMP);
```

Script Output x Query Result x  
SQL | All Rows Fetched: 1 in 0.016 seconds

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839 KING	PRESIDENT	(null) 17-11-81	5000	(null)		10

143. write about Clusters and explain it?

A. A cluster is a schema object that contains data from one or more tables, all of which have one or more columns in common. Oracle Database stores together all the rows from all the tables that share the same cluster key.

Clusters in SQL are used to store data that is from different tables in the same physical data blocks. They are used if records from those tables are frequently queried together. By storing same data blocks, the number of database block reads needed to full fill such queries decreases which improves performance.

- Each cluster stores tables data and maintains a clustered index to sort data.
- Columns within the cluster index are called clustered keys. These determine the physical placement of rows within the cluster.
- Cluster key is usually a foreign key of one table that references the primary key of another table in cluster.

144. write about decode and explain it with ex?

A. DECODE compares the expression to each search value one by one.

If expression is equal to a search, then the corresponding result is returned by the Oracle Database. If a match is not found, then default is returned. If default is omitted, then Oracle returns null.

- It is use to replace the more than one string
- It works like the if condition but it not allow relations operators.

Ex: select job decode (job,'managre','mgr','clerk','clk',job) from dual;

145. Write a query to create a backup table with records in another database as same table name?

A. CREATE TABLE Table\_Name AS SELECT \* FROM Source\_Table\_Name;

```
create table emp as select * from C##dummy.emp;
select * from emp.
```

Script Output | Query Result | All Rows Fetched: 14 in 0.012 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7698	BLAKE	MANAGER	7839	01-05-81	2850	(null)	30
3	7782	CLARK	MANAGER	7839	09-06-81	2450	(null)	10
4	7566	JONES	MANAGER	7839	02-04-81	2975	(null)	20
5	7788	SCOTT	ANALYST	7566	19-04-87	3000	(null)	20
6	7902	FORD	ANALYST	7566	03-12-81	3000	(null)	20
7	7369	SMITH	CLERK	7902	17-12-80	800	(null)	20
8	7499	ALLEN	SALESMAN	7698	20-02-81	1600	300	30
9	7521	WARD	SALESMAN	7698	22-02-81	1250	500	30
10	7654	MARTIN	SALESMAN	7698	28-09-81	1250	1400	30

146. Write a query to create a backup table with no records in another database as same table name?

A. create table emp as select \* from C##dummy.emp where 1=4;

```
create table emp as select * from C##dummy.emp where 1=4;
```

Script Output | Query Result | All Rows Fetched: 0 in 0.007 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO

147. Write a query to create a backup table of emp with location as CHICAGO in another database as same table name?

A. create table SURI as select \* from C##DUMMY.emp where DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE LOC='CHICAGO');

```
SELECT * FROM SURI;
```

Script Output | Query Result | SQL | All Rows Fetched: 6 in 0.012 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698	BLAKE	MANAGER	7839 01-05-81	2850	(null)	30	
2	7499	ALLEN	SALESMAN	7698 20-02-81	1600	300	30	
3	7521	WARD	SALESMAN	7698 22-02-81	1250	500	30	
4	7654	MARTIN	SALESMAN	7698 28-09-81	1250	1400	30	
5	7844	TURNER	SALESMAN	7698 08-09-81	1500	0	30	
6	7900	JAMES	CLERK	7698 03-12-81	950	(null)	30	

148. Write a query to create a backup table of emp with dname as accounting in another database as same table name?

A. create table NIKE as select \* from C##DUMMY.emp where DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME='ACCOUNTING');

```
create table NIKE as select * from C##DUMMY.emp where DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME='ACCOUNTING');

SELECT * FROM NIKE;
```

Script Output | Query Result | SQL | All Rows Fetched: 3 in 0.003 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7839	KING	PRESIDENT	(null)	17-11-81	5000	(null)	10
2	7782	CLARK	MANAGER	7839 09-06-81	2450	(null)	10	
3	7934	MILLER	CLERK	7782 23-01-82	1300	(null)	10	

149. Write a query to create a backup table of emp with dname as sales in another database as same table name?

A. create table KUMAR as select \* from C##DUMMY.emp where DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME='SALES');

```
create table KUMAR as select * from C##DUMMY.emp where DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE DNAME='SALES');

SELECT * FROM KUMAR;
```

Script Output | Query Result | SQL | All Rows Fetched: 6 in 0.002 seconds

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7698	BLAKE	MANAGER	7839 01-05-81	2850	(null)	30	
2	7499	ALLEN	SALESMAN	7698 20-02-81	1600	300	30	
3	7521	WARD	SALESMAN	7698 22-02-81	1250	500	30	
4	7654	MARTIN	SALESMAN	7698 28-09-81	1250	1400	30	
5	7844	TURNER	SALESMAN	7698 08-09-81	1500	0	30	
6	7900	JAMES	CLERK	7698 03-12-81	950	(null)	30	

150. Write a query to create a backup table of emp with loc as bostan in another database as same table name?

A. create table PAVAN as select \* from C##DUMMY.emp where DEPTNO NOT IN (SELECT DEPTNO FROM DEPT WHERE LOC='BOSTON');

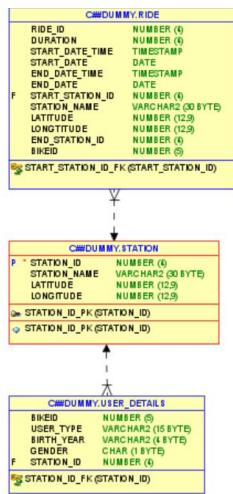
```
create table PAVAN as select * from C##DUMMY.emp where DEPTNO NOT IN (SELECT DEPTNO FROM DEPT WHERE LOC='BOSTON');

SELECT * FROM PAVAN;
```

t Output x Query Result x  
SQL | All Rows Fetched: 0 in 0.002 seconds  
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

151.What is the E-R model?

A. Entity-Relationship (ER) model is a visual representation of the table's structure and the relationships between logically related tables. In ER modeling the database structure is represented as a diagram known as ER diagram (ERD). An ER diagram gives a better understanding of the overall database structure.



152.Many-to-many relationships in er diagram?

A. Many to many relationship is type of cardinality that refers to a relation between two entities in ENTITY RELATIONSHIP DIAGRAM (Between two tables in a database).A simple example can have multiple courses and each course is for multiple students.

153. Many-to-one relationship?

A. Many-to-One relationship in DBMS is a relationship between more than one instances of an entity with one instance of another entity.

154. One-to-many relationship?

A. One-to-Many relationship in DBMS is a relationship between instances of an entity with more than one instance of another entity.

155. Notation of ER diagram with example diagram?

A. if you right click over an empty area of your ER diagram (LOGICAL MODEL) then the drop-down menu has a notation option that allows you to choose barker, Bachman or information engineering notation.

156. Mapping Constraints in ER diagram?

A. A mapping constraint is a data constraint that expresses the number of entities to which another entity can be related via a relationship set. It is most useful in describing the relationship sets that involve more than two entity sets.

157. write about case and explain with ex?

A. It is use to replace mora then one string by using relation operators

EX:SELECT CASE WHEN DEPTNO=10 AND JOB='MANAGER' THEN MGR ELSEJOB  
END J FROM EMP;

158. write about Conversion Functions and explain with ex?

The conversion function are there types

- To\_date: it is used to convert system format into user format.  
Ex:select to\_char ('21','dd') from dual;
- To\_char: it is used to convert user format into user format  
Ex: select to\_char(sysdate,'day') from dual;
- To\_number: it is used to translate the value of char or varchar data type in to number formate.  
Ex:select to\_number('20') from dual;

159. write difference between replace and translate?

A. REPLACE: it is used to replace entire string and its not possible to replace more then one string.

TRANSLATE: it is used to translate the character wise in a given string, if the character is found. And its not possible to translate entire string.

160. write about Concat and explain it?

A. CONCAT: it is used to merge the two strings. And we have to use '||' symbol while merge the two strings.

Syntax: select concat('abc','limited') from dual;