



**SUBMITTED BY:**

DEEPALI

**ID:** BTBTI20137

**ROLL NO.:** 2016736

**BRANCH:** BTECH IT (A)

**BATCH:** 2



**SUBMITTED TO:**

DR. KHANDAKAR F. RAHMAN

**SUBJECT:** DBMS LAB

**SUBJECT CODE:** CS212L

**TOTAL PAGES SUBMITTED:** 97

## ASSIGNMENT – 1

### ON : CREATING AND MANAGING TABLES – DDL

**Q1.** Create the DEPT table based on the following table instance chart. Place the syntax in a script called lab1\_1.sql, then execute the statement in the script to create the table. Confirm that the table is created.

Column Name	ID	NAME
<b>Key Type</b>		
<b>Nulls/Unique</b>		
<b>FK Table</b>		
<b>FK Column</b>		
<b>Data type</b>	<b>NUMBER</b>	<b>VARCHAR2</b>
<b>Length</b>	<b>7</b>	<b>25</b>

Name	Null?	Type
<b>ID</b>		<b>NUMBER(7)</b>
<b>NAME</b>		<b>VARRCHAR2(25)</b>

Ans1.

CREATE TABLE DEPT

( ID NUMBER(7),

    NAME VARCHAR2(25));

DESCRIBE DEPT;

#### Verification table-

The screenshot shows the Oracle SQL\*Plus interface. At the top, there's a toolbar with icons for Logout, New Session, History, Preferences, and Help. Below the toolbar, the title bar says "ORACLE SQL\*Plus". The main area is titled "Work Screen". It has a "File or URL" field with "Choose File" and "No file chosen" options, and a "Load Script" button. Below that is a "Enter statements:" text area containing the SQL code for creating and describing the DEPT table. The code is:

```
CREATE TABLE DEPT
(
    ID NUMBER(7),
    NAME VARCHAR2(25));
DESCRIBE DEPT;
```

At the bottom of the screen, there are several buttons: "Execute", "Save Script", "Clear Screen", and "Cancel". Below these buttons is a table showing the columns of the DEPT table:

Name	Null?	Type
ID		NUMBER(7)
NAME		VARCHAR2(25)

**Q2. Populate the DEPT table with data from the DEPARTMENTS table. Include only columns that you need.**

Ans2.

INSERT INTO DEPT

SELECT department\_id,department\_name

FROM departments;

SELECT \* FROM DEPT;

### **Verification table-**

The screenshot shows the Oracle SQL\*Plus interface. The command `SELECT * FROM DEPT;` has been entered into the command line. The results are displayed in a grid:

ID	NAME
10	Administration
20	Marketing
30	Shipping
40	IT
50	Sales
60	Executive
70	Accounting
80	Contracting

Below the grid, it says "8 rows selected."

**Q3. Create the EMP table based on the following table instance chart. Place the syntax in a script called lab1\_3.sql, then execute the statement in the script to create the table. Confirm that the table is created.**

Column Name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
<b>Key Type</b>				
<b>Nulls/Unique</b>				
<b>FK Table</b>				
<b>FK Column</b>				
<b>Data type</b>	<b>NUMBER</b>	<b>VARCHAR2</b>	<b>VARRCHAR2</b>	<b>NUMBER</b>
<b>Length</b>	<b>7</b>	<b>25</b>	<b>25</b>	<b>7</b>

Name	Null?	Type
ID		<b>NUMBER(7)</b>
LAST_NAME		<b>VARCHAR2(25)</b>
FIRST_NAME		<b>VARCHAR2(25)</b>

<b>DEPT_ID</b>		<b>NUMBER(7)</b>
----------------	--	------------------

Ans3.

```
CREATE TABLE EMP
(
  ID NUMBER(7),
  LAST_NAME VARCHAR2(25),
  FIRST_NAME VARCHAR2(25),
  DEPT_ID NUMBER(7));
```

DESCRIBE EMP;

### Verification table-

The screenshot shows the Oracle SQL\*Plus interface. At the top, it says "ORACLE iSQL\*Plus". Below that is a toolbar with icons for Help, Preferences, and Session History. The main area is titled "Work Screen" and contains a text input field with the command "DESCRIBE EMP;". Below the input field is a table showing the column details:

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

At the bottom of the screen are buttons for "Execute", "Save Script", "Clear Screen", and "Cancel".

**Q4. Modify the EMP table to allow for longer employee last names. Confirm your modification.**

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(50)
FIRST_NAME		VARCHAR2(50)
DEPT_ID		NUMBER(7)

Ans4.

ALTER TABLE EMP

```
MODIFY (LAST_NAME VARCHAR2(50));
```

```
DESCRIBE EMP;
```

### Verification table-

The screenshot shows the Oracle SQL\*Plus interface. In the command window, the command `DESCRIBE EMP;` is entered. Below the command, the system displays the structure of the `EMP` table:

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(50)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

**Q5. Confirm that both the DEPT and EMP tables are stored in the data dictionary. (HINT: USER\_TABLES)**

TABLE_NAME
DEPT
EMP

Ans5.

```
SELECT TABLE_NAME FROM USER_TABLES
```

```
WHERE TABLE_NAME IN ('DEPT','EMP');
```

### Verification table-

The screenshot shows the Oracle SQL\*Plus interface. In the command window, the query `SELECT TABLE_NAME FROM USER_TABLES WHERE TABLE_NAME IN ('DEPT','EMP');` is entered. Below the command, the system displays the results:

TABLE_NAME
DEPT
EMP

**Q6. Create the EMPLOYEES2 table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, AND DEPARTMENT\_ID columns. Name the columns in your new table ID, FIRST\_NAME, LAST\_NAME, SALARY, DEPT\_ID, respectively.**

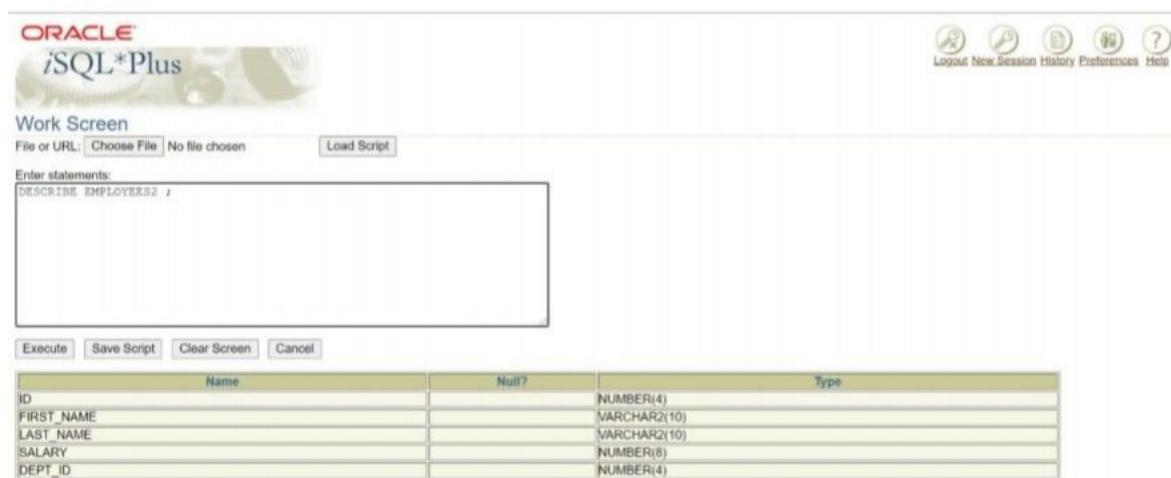
Ans6.

```
CREATE TABLE EMPLOYEES2
```

AS

```
SELECT EMPLOYEE_ID ID, FIRST_NAME,  
       LAST_NAME, SALARY,  
       DEPARTMENT_ID DEPT_ID  
FROM EMPLOYEES;  
  
DESCRIBE EMPLOYEES2;
```

#### **Verification table-**



The screenshot shows the Oracle iSQL\*Plus interface. The title bar says "ORACLE iSQL\*Plus". The main area is titled "Work Screen". It has a toolbar with icons for Logout, New Session, History, Preferences, and Help. Below the toolbar, there are buttons for File or URL, Choose File (No file chosen), Load Script, Execute, Save Script, Clear Screen, and Cancel. A text input field labeled "Enter statements:" contains the SQL command: "DESCRIBE EMPLOYEES2 ;". Below the input field is a table showing the structure of the EMPLOYEES2 table:

Name	Null?	Type
ID		NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
SALARY		NUMBER(6)
DEPT_ID		NUMBER(4)

**Q7. Drop the EMP table.**

Ans7.

```
DROP TABLE EMP;
```

```
SELECT TABLE_NAME FROM USER_TABLES WHERE TABLE_NAME='EMP';
```

#### **Verification table-**

**ORACLE**  
*iSQL\*Plus*



Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT TABLE_NAME FROM USER_TABLES WHERE TABLE_NAME = 'EMP';
```

no rows selected

## **Q8. Rename the EMPLOYEES2 table as EMP.**

**Ans8.**

**RENAME EMPLOYEES2 TO EMP;**

**SELECT TABLE\_NAME FROM USER\_TABLES WHERE TABLE\_NAME='EMPLOYEES2';**

**DESCRIBE EMP;**

### **Verification table-1**

**ORACLE**  
*iSQL\*Plus*



Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT TABLE_NAME FROM USER_TABLES WHERE TABLE_NAME='EMPLOYEES2';
```

no rows selected

### **Verification table-2**

**ORACLE**  
*iSQL\*Plus*



Work Screen

File or URL:  No file chosen

Enter statements:

```
DESCRIBE EMP;
```

Name	Null?	Type
ID		NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
SALARY		NUMBER(8)
DEPT_ID		NUMBER(4)

**Q9. Add a comment to the DEPT and EMP table definitions describing the tables. Confirm your additions in the data dictionary.**

Ans9.

COMMENT ON TABLE DEPT

IS 'Department Information';

COMMENT ON TABLE EMP

IS 'Employee Information';

SELECT \* FROM USER\_TAB\_COMMENTS

WHERE TABLE\_NAME='DEPT' OR TABLE\_NAME='EMP';

**Verification table-**

The screenshot shows the iSQL\*Plus Work Screen interface. At the top, there's a toolbar with icons for Logout, New Session, History, Preferences, and Help. Below the toolbar, the title bar says "Work Screen". Underneath the title bar, there are buttons for "File or URL:" (with "Choose File" and "No file chosen"), "Load Script", and "Execute" (which is highlighted). The main area is a text box labeled "Enter statements:" containing the SQL query: "SELECT \* FROM USER\_TAB\_COMMENTS WHERE TABLE\_NAME='DEPT' OR TABLE\_NAME='EMP';". Below this text box are buttons for "Save Script", "Clear Screen", and "Cancel". At the bottom of the screen is a table with three columns: "TABLE\_NAME", "TABLE\_TYPE", and "COMMENTS". The table contains two rows: one for the DEPT table (Type: TABLE, Comment: Department Information) and one for the EMP table (Type: TABLE, Comment: Employee Information).

TABLE_NAME	TABLE_TYPE	COMMENTS
DEPT	TABLE	Department Information
EMP	TABLE	Employee Information

**Q10. Drop the FIRST\_NAME column from the EMP table. Confirm your modification by checking the description of the table.**

Ans10.

ALTER TABLE EMP

DROP COLUMN FIRST\_NAME;

DESCRIBE EMP;

**Verification table-**

ORACLE  
iSQL\*Plus



## Work Screen

File or URL: Choose File No file chosen

Load Script

Enter statements:  
DESCRIBE EMP;

Execute Save Script Clear Screen Cancel

Name	Null?	Type
ID		NUMBER(4)
LAST_NAME		VARCHAR2(10)
SALARY		NUMBER(8)
DEPT_ID		NUMBER(4)

**Q11. In the EMP table, mark the DEPT\_ID column in the EMP table as UNUSED. Confirm your modification by checking the description of the table.**

Ans11.

ALTER TABLE EMP

SET UNUSED (DEPT\_ID);

DESCRIBE EMP;

**Verification table-**

ORACLE  
iSQL\*Plus



## Work Screen

File or URL: Choose File No file chosen

Load Script

Enter statements:  
DESCRIBE EMP;

Execute Save Script Clear Screen Cancel

Name	Null?	Type
ID		NUMBER(4)
LAST_NAME		VARCHAR2(10)
SALARY		NUMBER(8)

**Q12. Drop all the UNUSED columns from the EMP table. Confirm your modification by checking the description of the table.**

Ans12.

ALTER TABLE EMP

DROP UNUSED COLUMNS;

```
SELECT * FROM USER_UNUSED_COL_TABS;
```

```
DESCRIBE EMP;
```

### Verification table-1

The screenshot shows the iSQL\*Plus interface with the title bar "ORACLE iSQL\*Plus". Below the title bar is a toolbar with icons for Logout, New Session, History, Preferences, and Help. The main area is titled "Work Screen". It contains a text input field for "Enter statements:" with the SQL command "SELECT \* FROM USER\_UNUSED\_COL\_TABS;". Below the input field are four buttons: "Execute", "Save Script", "Clear Screen", and "Cancel". At the bottom of the screen, it says "no rows selected".

### Verification table-2

The screenshot shows the iSQL\*Plus interface with the title bar "ORACLE iSQL\*Plus". Below the title bar is a toolbar with icons for Logout, New Session, History, Preferences, and Help. The main area is titled "Work Screen". It contains a text input field for "Enter statements:" with the SQL command "DESCRIBE EMP;". Below the input field are four buttons: "Execute", "Save Script", "Clear Screen", and "Cancel". To the right of the input field, there is a table showing the structure of the EMP table:

Name	Null?	Type
ID		NUMBER(4)
LAST_NAME		VARCHAR2(10)
SALARY		NUMBER(8)

## ASSIGNMENT – 2

### ON: MANIPULATING DATA – DML

**Q1. Run the statement in the lab2\_1.sql script to build the MY\_EMPLOYEE table to be used for the lab.**

Ans1.

```
CREATE TABLE MY_EMPLOYEE
( ID    NUMBER(4) NOT NULL,
LAST_NAME  VARCHAR2(25),
FIRST_NAME VARCHAR2(25),
USERID   VARCHAR2(8),
SALARY    NUMBER(9,2));
```

**Q2. Describe the structure of the MY\_EMPLOYEE table to identify the column names.**

NAME	NULL?	TYPE
ID		NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)

Ans2.

```
DESCRIBE MY_EMPLOYEE;
```

**Verification table-**

Work Screen

File or URL:  No file chosen

Enter statements:  
DESCRIBE MY\_EMPLOYEE;

Name	Null?	Type
ID	NOT NULL	NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)

**Q3.** Add the first row of data to the MY\_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audrey	aropebur	1550

Ans3.

INSERT INTO MY\_EMPLOYEE

VALUES (1, 'Patel', 'Ralph', 'rpatel', 895);

**Q4.** Populate the MY\_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.

Ans4.

INSERT INTO MY\_EMPLOYEE

(ID, LAST\_NAME, FIRST\_NAME, USERID, SALARY)

VALUES (2, 'Dancs', 'Betty', 'bdancs', 860);

**Q5.** Confirm your addition to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860

Ans5.

```
SELECT * FROM MY_EMPLOYEE;
```

### **Verification table-**

The screenshot shows the Oracle iSQL\*Plus interface. At the top, there are icons for Logout, New Session, History, and Preferences. Below the title bar, it says "Work Screen". A toolbar has buttons for Choose File, Load Script, Execute, Save Script, Clear Screen, and Cancel. The main area contains a text input box with the SQL command "SELECT \* FROM MY\_EMPLOYEE;". Below the input box is a table with the following data:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860

**Q6. Populate the MY\_EMPLOYEE table with the third, forth, and fifth row of the sample data provided in question 3. Verify the insertions.**

Ans6.

```
INSERT INTO MY_EMPLOYEE  
(ID, LAST_NAME, FIRST_NAME, USERID, SALARY)  
VALUES (&ID, '&LAST_NAME', '&FIRST_NAME', '&USERID', &SALARY);
```

### **Pop up window 1-**

The screenshot shows the Oracle iSQL\*Plus interface with a "Substitution Variables" dialog box open. The title bar says "Substitution Variables". It contains a table for entering variable values:

Variable	Value
id	3
last_name	Biri
first_name	Ben
userid	bbiri
salary	1100

At the bottom right of the dialog box are "OK" and "Cancel" buttons.

### **Pop up window 2-**



Work Screen &gt; Substitution Variables

### Substitution Variables

Enter values for substitution variables in the script to execute:

Variable	Value
id	4
last_name	Newman
first_name	Chad
userid	cnewman
salary	750

## Pop up window 3-



Work Screen &gt; Substitution Variables

### Substitution Variables

Enter values for substitution variables in the script to execute:

Variable	Value
id	5
last_name	Ropeburn
first_name	Audrey
userid	arophebur
salary	1550

```
SELECT * FROM MY_EMPLOYEE;
```

## Verification table-



Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT * FROM MY_EMPLOYEE;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audrey	arophebur	1550

## ASSIGNMENT – 3

### ON: DATA CONTROL – DCL

**Q6.** Write an INSERT statement in a text file named loademp.sql to load rows into the MY\_EMPLOYEE table. Concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID.

Ans6.

```
INSERT INTO MY_EMPLOYEE  
(ID, LAST_NAME, FIRST_NAME, USERID, SALARY)  
VALUES (&ID, '&LAST_NAME', '&FIRST_NAME',  
       LOWER(SUBSTR('&FIRST_NAME',1,1) || SUBSTR('&LAST_NAME',1,7)),  
       &SALARY);
```

#### Verification table 1 -

The screenshot shows the Oracle iSQL\*Plus interface. At the top, there's a toolbar with icons for New Session, History, Preferences, and Help. Below the toolbar, the title bar says "iSQL\*Plus". Underneath the title bar, there's a menu bar with "File", "Edit", "View", "Tools", "Help", and "Logout". A status bar at the bottom shows "Last Session" and "Substitution Variables". The main area is titled "Substitution Variables". It contains a section for "Enter values for substitution variables in the script to execute:" followed by a table with four rows:

Variable	Value
ID	1
last_name	Patel
first_name	Ralph

At the bottom right of the dialog are "OK" and "Cancel" buttons.

#### Verification table 2 -



Work Screen > Substitution Variables

### Substitution Variables

Enter values for substitution variables in the script to execute:

Variable	Value
id	2
last_name	Davies
first_name	Billy
salary	800

**Q7. Populate the table with the next two rows of sample data by running the INSERT statement in the script that you created.**

Ans7.

```
INSERT INTO MY_EMPLOYEE  
(ID, LAST_NAME, FIRST_NAME, USERID, SALARY)  
VALUES (&ID, '&LAST_NAME', '&FIRST_NAME',  
       LOWER(SUBSTR('&FIRST_NAME',1,1) || SUBSTR('&LAST_NAME',1,7)),  
       &SALARY);
```

### Verification table 1 -



Work Screen > Substitution Variables

### Substitution Variables

Enter values for substitution variables in the script to execute:

Variable	Value
id	3
last_name	Elin
first_name	Ben
salary	1100

### Verification table 2 -



Work Screen > Substitution Variables

### Substitution Variables

Enter values for substitution variables in the script to execute:

Variable	Value
id	4
last_name	Newman
first_name	Chad
salary	750

## Q8. Confirm your additions to the table.

Ans8.

```
SELECT * FROM MY_EMPLOYEE;
```

### Verification table -



Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT * FROM MY_EMPLOYEE;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Paioli	Ralph	rpaioli	895
2	Dancs	Betty	bdancs	880
3	Bini	Ben	bbini	1100
4	Newman	Chad	cnewman	750

## Q9. Make the data additions permanent.

Ans9.

```
COMMIT;
```

Update and delete data in the MY\_EMPLOYEE table.

Q10. Change the last name of employee 3 to Drexler.

Ans10.

```
UPDATE MY_EMPLOYEE
```

```
SET LAST_NAME='Drexler'
```

```
WHERE ID=3;
```

**Q11. Change the salary to 1000 for all employees with a salary less than 900.**

Ans11.

```
UPDATE MY_EMPLOYEE
```

```
SET SALARY=1000
```

```
WHERE SALARY<900;
```

**Q12. Verify your changes to the table.**

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
2	Dancs	Betty	bdancs	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Ans12.

```
SELECT * FROM MY_EMPLOYEE;
```

**Verification table -**

The screenshot shows the Oracle iSQL\*Plus Work Screen interface. At the top, there's a toolbar with icons for Logout, New Session, History, and Preferences. Below the toolbar, the title bar says "Work Screen". Underneath the title bar, there are input fields for "File or URL" (containing "Choose File") and "Load Script", and a "Logout" button. A message "No file chosen" is displayed. Below these, a section titled "Enter statements:" contains the SQL command "SELECT \* FROM MY\_EMPLOYEE;". At the bottom of the screen, there are buttons for "Execute", "Save Script", "Clear Screen", and "Cancel". The main area displays the results of the query in a table:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
2	Dancs	Betty	bdancs	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

**Q13. Delete Betty Dancs from the MY\_EMPLOYEE table.**

Ans13.

DELETE FROM MY\_EMPLOYEE

WHERE LAST\_NAME='Dancs';

**Q14. Confirm your changes to the table.**

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Ans14.

SELECT \* FROM MY\_EMPLOYEE;

**Verification table -**

The screenshot shows the Oracle iSQL\*Plus interface. At the top, there are icons for Logout, New Session, History, and Help. Below that is a toolbar with 'Work Screen' and file-related buttons like 'Choose File' and 'Load Script'. A text area labeled 'Enter statements:' contains the SQL command 'SELECT \* FROM MY\_EMPLOYEE;'. Below this is a results grid displaying four rows of data from the table:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

**Q15. Commit all pending changes.**

Ans15.

COMMIT;

Control data transaction to the MY\_EMPLOYEE table.

**Q16. Populate the table with the last row of sample data by modifying the statements in the script that you created in step 6. Run the statements in the script.**

Ans16.

```

INSERT INTO MY_EMPLOYEE
(ID, LAST_NAME, FIRST_NAME, USERID, SALARY)
VALUES (&ID, '&LAST_NAME', '&FIRST_NAME',
LOWER(SUBSTR('&FIRST_NAME',1,1) || SUBSTR('&LAST_NAME',1,7)),
&SALARY);

```

### Verification table -

ORACLE  
SQL\*Plus

Work Screen > Substitution Variables

Substitution Variables

Enter values for substitution variables in the script to execute:

Variable	Value
id	5
last_name	Ropeburn
first_name	Audrey
salary	1550

OK Cancel

### Q17. Confirm your addition to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

Ans17.

```
SELECT * FROM MY_EMPLOYEE;
```

### Verification table -

Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT * FROM MY_EMPLOYEE;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

**Q18. Mark an intermediate point in the processing of the transaction.**

Ans18.

```
SAVEPOINT STEP17;
```

**Q19. Empty the entire table.**

Ans19.

```
DELETE FROM MY_EMPLOYEE;
```

**Q20. Confirm that the table is empty.**

Ans20.

```
SELECT * FROM MY_EMPLOYEE;
```

**Verification table -**

## Work Screen

File or URL:  No file chosen

Enter statements:

SELECT \* FROM MY\_EMPLOYEE;

no rows selected

**Q21. Discard the most recent DELETE operation without discarding the earlier INSERT operation.**

Ans21.

ROLLBACK TO STEP17;

**Q22. Confirm that the new row is still intact.**

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

Ans22.

SELECT \* FROM MY\_EMPLOYEE;

**Verification table -**

Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT * FROM MY_EMPLOYEE;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	jropebur	1550

## Q23. Make the data addition permanent.

Ans23.

COMMIT;

## ASSIGNMENT – 4

### ON: DATABASE CONSTRAINTS

Q1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my\_emp\_id\_pk.  
Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.

Ans1.

```
ALTER TABLE EMP
```

```
ADD CONSTRAINT my_emp_id_pk
```

```
PRIMARY KEY(ID);
```

Q2. Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my\_deptid\_pk.  
Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.

Ans2.

```
ALTER TABLE DEPT
```

```
ADD CONSTRAINT my_deptid_pk
```

```
PRIMARY KEY(ID);
```

Q3. Add a column DEPT\_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my\_emp\_dept\_id\_fk.

Ans3.

```
ALTER TABLE EMP
```

```
ADD (DEPT_ID NUMBER(8));
```

```
ALTER TABLE EMP
```

```
ADD CONSTRAINT my_emp_dept_id_fk
```

```
FOREIGN KEY (DEPT_ID)
```

REFERENCES EMP(ID);

**Q4.** Confirm that the constraints were added by querying the USER\_CONSTRAINTS view. Note the types and names of the constraints. Save your statement text in a file called lab4\_4.sql.

CONSTRAINT_NAME	C
MY_DEPTID_PK	P
MY_EMP_ID_PK	P
MY_EMP_DEPT_ID_FK	R

Ans4.

```
SELECT constraint_name, constraint_type  
FROM user_constraints  
WHERE table_name='EMP' OR table_name='DEPT';
```

**Verification table -**

The screenshot shows the Oracle iSQL\*Plus interface. At the top, there's a logo for 'ORACLE iSQL\*Plus'. To the right are icons for Logout, New Session, History, and Help. Below the logo, it says 'Work Screen'. There are buttons for 'Choose File' (disabled), 'No file chosen', and 'Load Script'. A text area labeled 'Enter statements:' contains the SQL query: 'SELECT constraint\_name, constraint\_type FROM user\_constraints WHERE table\_name='EMP' OR table\_name='DEPT';'. At the bottom of this area are buttons for 'Execute', 'Save Script', 'Clear Screen', and 'Cancel'. Below the input area is a table titled 'CONSTRAINT\_NAME' with three rows corresponding to the ones in the question. The table has two columns: 'CONSTRAINT\_NAME' and 'C'. The rows are: MY\_DEPTID\_PK (P), MY\_EMP\_ID\_PK (P), and MY\_EMP\_DEPT\_ID\_FK (R).

CONSTRAINT_NAME	C
MY_DEPTID_PK	P
MY_EMP_ID_PK	P
MY_EMP_DEPT_ID_FK	R

**Q5.** Display the object names and types from the USER\_OBJECTS data dictionary view for the EMP and DEPT tables. Notice that the new tables and a new index were created.

Ans5.

```
SELECT object_name, object_type  
FROM user_objects  
WHERE object_name='EMP' OR object_name='DEPT';
```

### Verification table -

The screenshot shows the Oracle iSQL\*Plus interface. At the top, there are icons for Logout, New Session, and History, along with a 'F' button. Below the title bar, it says 'Work Screen'. A file navigation bar includes 'File or URL:' with 'Choose File' and 'No file chosen', and buttons for 'Load Script', 'Execute', 'Save Script', 'Clear Screen', and 'Cancel'. The main area is titled 'Enter statements:' and contains the SQL query:

```
SELECT object_name, object_type  
FROM user_objects  
WHERE object_name='EMP' OR object_name='DEPT';
```

Below the query, the results are displayed in a table:

OBJECT_NAME	OBJECT_TYPE
DEPT	TABLE
EMP	TABLE

If you have time, complete the following exercise.

Q6. Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

Ans6.

```
ALTER TABLE EMP  
ADD COMMISSION NUMBER(2,2) CONSTRAINT emp_comm_min  
CHECK (COMMISSION>0);
```

# ASSIGNMENT – 5(1)

## ON: RESTRICTING AND SORTING DATA

Q1. Show the structure of the EMPLOYEES table. Create a query to display the last name, job code, hire date and employee number for each employee, with employee number appearing first.

Ans1.

DESCRIBE EMPLOYEES;

SELECT EMPLOYEE\_ID, LAST\_NAME, JOB\_ID, HIRE\_DATE

FROM EMPLOYEES;

### Verification table 1 -

The screenshot shows the iSQL\*Plus Work Screen interface. At the top, there's a toolbar with icons for Logout, New Session, History, and Preferences. Below the toolbar, the title bar says "Work Screen". Underneath the title bar, there are buttons for "File or URL:" (with "Choose File" and "No file chosen" options), "Load Script", and "Execute" (which is highlighted). A text input field labeled "Enter statements:" contains the command "DESCRIBE EMPLOYEES;". Below this input field is a large text area where the table structure will be displayed. At the bottom of the screen, there are buttons for "Save Script", "Clear Screen", and "Cancel".

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(4)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
EMAIL	NOT NULL	VARCHAR2(15)
PHONE_NUMBER		VARCHAR2(10)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8)
COMMISSION_PCT		NUMBER(4)
MANAGER_ID		NUMBER(10)
DEPARTMENT_ID		NUMBER(4)

### Verification table 2-

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hundold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorenz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98
144	Vargas	ST_CLERK	09-JUL-98
149	Zlotkey	SA_MAN	29-JAN-00
174	Abel	SA REP	11-MAY-96
176	Taylor	SA REP	24-MAR-98
EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
178	Grant	SA REP	24-MAY-99
200	Whalen	AD_ASST	14-SEP-87
201	Hartstein	MK_MAN	17-FEB-96
202	Fay	MK REP	17-AUG-97
205	Higgins	AC_MGR	07-JUN-94
206	Gietz	AC_ACCOUNT	07-JUN-94
999	Taylor	ST_CLERK	07-JUN-99

21 rows selected.

## Q2. Create a query to display unique job codes from the EMPLOYEES table.

Ans2.

SELECT DISTINCT JOB\_ID

FROM EMPLOYEES;

### Verification table -

File or URL:  No file chosen

Enter statements:

```
SELECT DISTINCT JOB_ID
FROM EMPLOYEES;
```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
IT_PROG
MK_MAN
MK REP
SA_MAN
SA REP
ST_CLERK
ST_MAN

12 rows selected.

## Q3. Create a query to display the column headings in EMPLOYEES table as. Emp #, Employee, Job, and Hire Date, respectively.

Ans3.

```
SELECT EMPLOYEE_ID "Emp#", LAST_NAME "Employee",
JOB_ID "Job", HIRE_DATE "Hire Date"
FROM EMPLOYEES;
```

#### Verification table -

Emp#	Employee	Job	Hire Date
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorentz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98
144	Vargas	ST_CLERK	09-JUL-98
149	Zlotkey	SA_MAN	29-JAN-00
174	Abel	SA REP	11-MAY-96
176	Taylor	SA REP	24-MAR-98
Emp#	Employee	Job	Hire Date
178	Grant	SA REP	24-MAY-99
200	Whalen	AD_ASST	14-SEP-87
201	Hartstein	MK_MAN	17-FEB-96
202	Fay	MK REP	17-AUG-97
205	Higgins	AC_MGR	07-JUN-94
206	Gietz	AC_ACCOUNT	07-JUN-94
999	Taylor	ST_CLERK	07-JUN-99

21 rows selected.

Q4. Display the last name concatenated with the job ID, separated by a comma and space, and name the column Employee and Title.

Employee and Title
King, AD_PRES
Kochhar, AD_VP
De Haan, AD_VP
Hunold, IT_PROG
Ernst, IT_PROG
Lorentz, IT_PROG
Mourgos, ST_MAN
Rajs, ST_CLERK
Davies, ST_CLERK

Ans4.

```
SELECT LAST_NAME||', '|||JOB_ID "Employee and Title"
```

FROM EMPLOYEES;

**Verification table -**

Employee and Title
King, AD_PRES
Kochhar, AD_VP
De Haan, AD_VP
Hundold, IT_PROG
Emsl, IT_PROG
Lorentz, IT_PROG
Mourgos, ST_MAN
Rajs, ST_CLERK
Davies, ST_CLERK
Matos, ST_CLERK
Vargas, ST_CLERK
Zlotkey, SA_MAN
Abel, SA REP
Taylor, SA REP
Employee and Title
Grant, SA REP
Whalen, AD_ASST
Hartstein, MK_MAN
Fay, MK REP
Higgins, AC_MGR
Gietz, AC_ACCOUNT
Taylor, ST_CLERK

21 rows selected.

Q5. Create a query to display all the data from the EMPLOYEES table. Separate each column by a comma. Name the column THE\_OUTPUT.

THE_OUTPUT
100,Steven,King,SKING,515.123.4567,AD_PRES,,17-JUN-87,24000,,90
101,Neena,Kochhar,NKOCHHAR,515.123.4568,AD_VP,100,21-SEP-89,17000,,90
102,Lex,De Haan,LDEHAAN,515.123.4568,AD_VP,100,13-JAN-93,17000,,90
103,Alexander,Hunold,AHUNOLD,590.423.4567,IT_PROG,102,03-JAN-90,9000,,60
104,Bruce,Ernst,BERNST,590.423.4568,IT_PROG,103,21-MAY-91,6000,,60
107,Diana,Lorentz,DLORENTZ,590.423.5567,IT_PROG,103,07-FEB-99,4200,,60
124,Kevin,Mourgos,KMOURGOS,650.123.5234,ST_MAN,100,16-NOV-99,5800,,50
141,Trenna,Rajs,TRAJS,650.121.8009,ST_CLERK,124,17-OCT-95,3500,,50

Ans5.

```
SELECT EMPLOYEE_ID||','||FIRST_NAME||','||LAST_NAME  
||','||EMAIL||','||PHONE_NUMBER||','||HIRE_DATE  
||','||JOB_ID||','||SALARY||','||COMMISSION_PCT  
||','||MANAGER_ID||','||DEPARTMENT_ID "THE_OUTPUT"
```

FROM EMPLOYEES;

### Verification table-

THE_OUTPUT
100,Stewen,King,Sking,121473456,17-JUN-87,AD_PRES,2400,,90
101,Neena,Kochhar,NKochhar,4587256123,21-SEP-89,AD_VP,17000,,100,90
102,Lex,De Haan,LDEhaan,14321123,13-JAN-93,AD_VP,17000,,100,90
103,Alexander,Hundold,Ahunold,9180490001,03-JAN-90,IT_PROG,9000,,102,60
104,Bruce,Emst,Bernst,9080490111,21-MAY-91,IT_PROG,4200,,103,60
107,Diana,Lorenz,Dlorentz,9070410121,07-FEB-99,IT_PROG,5800,,100,50
124,Kevin,Mourgos,Kmourgos,9979510331,16-NOV-99,ST_MAN,3500,,124,50
141,Trenna,Rajs,Traje,9978410121,17-OCT-95,ST_CLERK,3100,,124,50
142,Curtis,Davies,Ddavies,997854621,29-JAN-97,ST_CLERK,2600,,124,50
143,Randall,Matos,Rmatos,6501212874,15-MAR-98,ST_CLERK,2600,,124,50
144,Peter,Vargas,Pvargas,6501212004,09-JUL-98,ST_CLERK,2500,,124,50
149,Eleni,Zlotkey,Ezlotkey,6501212005,29-JAN-00,SA_MAN,10500,2,100,80
178,Jonathon,Taylor,Jtaylor,1644429265,24-MAR-98,SA REP,8600,2,149,80
THE_OUTPUT
178,Kimberely,Grant,Kgrant,1644429263,24-MAY-99,SA REP,7000,15,149,
200,Jennifer,Whalen,Jwhalen,5151234444,14-SEP-87,AD_ASST,4400,,101,10
201,Michael,Hartstein,Mhartste,5151235555,17-FEB-96,MK_MAN,13000,,100,20
202,Pat,Fay,Pfay,6031236666,17-AUG-97,MK REP,6000,,201,20
205,Shelley,Higgins,Shiggins,5151238080,07-JUN-94,AC_MGR,12000,,101,110
206,Williams,Gietz,Wgietz,5151238181,07-JUN-94,AC_ACCOUNT,8300,,205,110
999,,Taylor,Dtaylor,,07-JUN-99,ST_CLERK,5000,,50

21 rows selected.

## ASSIGNMENT – 5(2)

### ON: RESTRICTING AND SORTING DATA

Q1. Create a query to display the last name and salary of employees earning more than \$12,000. Place your SQL statement in a text file named lab5\_1.sql. Run your query.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hartstein	13000

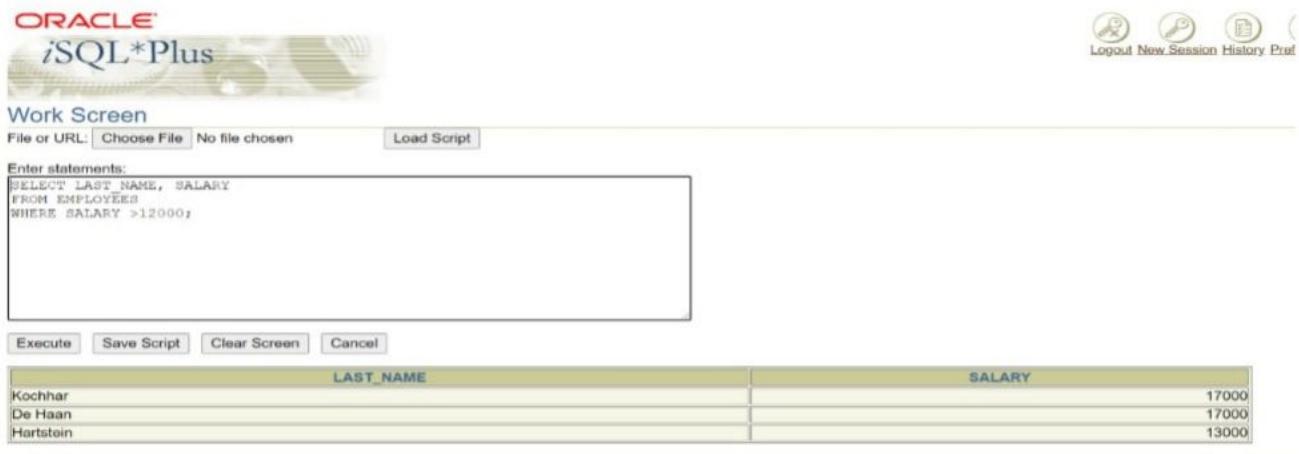
Ans1.

```
SELECT LAST_NAME, SALARY
```

```
FROM EMPLOYEES
```

```
WHERE SALARY >12000;
```

## Verification table -



The screenshot shows the iSQL\*Plus Work Screen interface. At the top, there's a toolbar with icons for Logout, New Session, History, and Print. Below the toolbar, the title 'Work Screen' is displayed, along with a file navigation bar ('File or URL: Choose File No file chosen' and 'Load Script'). A large text area labeled 'Enter statements:' contains the SQL query: 'SELECT LAST\_NAME, SALARY FROM EMPLOYEES WHERE SALARY >12000;'. Below this area are several buttons: 'Execute', 'Save Script', 'Clear Screen', and 'Cancel'. To the right, a table displays the results of the query:

LAST_NAME	SALARY
Kochhar	17000
De Haan	17000
Hartstein	13000

Q2. Create a query to display the employee last name and department number for each employee number 176.

LAST_NAME	DEPARTMENT_ID
Taylor	80

Ans2.

```
SELECT LAST_NAME, DEPARTMENT_ID
FROM EMPLOYEES
WHERE EMPLOYEE_ID=176;
```

## Verification table -



The screenshot shows the iSQL\*Plus Work Screen interface. At the top, there's a toolbar with icons for Logout, New Session, History, and Print. Below the toolbar, the title 'Work Screen' is displayed, along with a file navigation bar ('File or URL: Choose File No file chosen' and 'Load Script'). A large text area labeled 'Enter statements:' contains the SQL query: 'SELECT LAST\_NAME, DEPARTMENT\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID=176;'. Below this area are several buttons: 'Execute', 'Save Script', 'Clear Screen', and 'Cancel'. To the right, a table displays the results of the query:

LAST_NAME	DEPARTMENT_ID
Taylor	80

Q3. Modify lab5\_1.sql to display the last name and salary for all employees whose salary is not in the range of \$5,000 and \$12,000. Place your SQL statement in a text file named lab5\_3.sql.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Lorentz	4200
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Whalen	4400
Hartstein	13000

10 rows selected.

Ans3.

```
SELECT LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE SALARY NOT BETWEEN 5000 AND 12000;
```

#### Verification table -

File or URL:  No file chosen

Enter statements:

```
SELECT LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE SALARY NOT BETWEEN 5000 AND 12000;
```

Execute Save Script Clear Screen Cancel

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Ernst	4200
Mourgos	3500
Rajs	3100
Davies	2600
Matos	2600
Vargas	2500
Abel	1100
Whalen	4400
Hartstein	13000

12 rows selected.

Q4. Display the employee last name, job ID, and start date of employees hired between February 20, 1998 and May 1, 1998. Order the query in ascending order by start date.

LAST_NAME	JOB_ID	HIRE_DATE
Matos	ST_CLERK	15-MAR-98
Taylor	SA_REP	24-MAR-98

Ans4.

```
SELECT LAST_NAME,JOB_ID,HIRE_DATE
FROM EMPLOYEES
WHERE HIRE_DATE BETWEEN '20-Feb-1998' AND '01-MAY-1998'
ORDER BY HIRE_DATE ASC;
```

#### Verification table -

The screenshot shows the Oracle iSQL\*Plus interface. At the top, there are icons for Logout, New Session, History, and Preferences. Below that is the title "Work Screen". A toolbar has buttons for File or URL, Choose File (No file chosen), Load Script, Execute, Save Script, Clear Screen, and Cancel. The main area contains a text box with the following SQL code:

```
SELECT LAST_NAME, JOB_ID, HIRE_DATE
FROM EMPLOYEES
WHERE HIRE_DATE BETWEEN '20-Feb-1998' AND '01-MAY-1998'
ORDER BY HIRE_DATE ASC;
```

Below the text box are four buttons: Execute, Save Script, Clear Screen, and Cancel. The results are displayed in a grid:

LAST_NAME	JOB_ID	HIRE_DATE
Matos	ST_CLERK	15-MAR-98
Taylor	SA_REP	24-MAR-98

Q5. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.

LAST_NAME	DEPARTMENT_ID
Davies	50
Fay	20
Hartstein	20
Matos	50
Mourgos	50
Rajs	50
Vargas	50

7 rows selected.

Ans5.

```

SELECT LAST_NAME,DEPARTMENT_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (20,50)
ORDER BY LAST_NAME ASC;

```

**Verification table -**

Work Screen	
File or URL:	<input type="button" value="Choose File"/> No file chosen <input type="button" value="Load Script"/>
Enter statements:	
<pre>SELECT LAST_NAME,DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID IN (20,50) ORDER BY LAST_NAME ASC;</pre>	
<input type="button" value="Execute"/> <input type="button" value="Save Script"/> <input type="button" value="Clear Screen"/> <input type="button" value="Cancel"/>	
LAST_NAME	DEPARTMENT_ID
Davies	50
Fay	20
Hartstein	20
Lorenz	50
Matos	50
Mourgos	50
Rajs	50
Taylor	50
Vargas	50

9 rows selected.

Q6. Modify lab5\_3.sql to list the last name and salary of employees who earn between \$5,000 and \$12000, and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively. Resave lab5\_3.sql as lab5\_6.sql. Run the statement in lab5\_6.sql.

Employee	Monthly Salary
Mourgos	5800
Fay	6000

Ans6.

```

SELECT LAST_NAME "Employee", SALARY "Monthly Salary"
FROM EMPLOYEES
WHERE SALARY BETWEEN 5000 AND 12000
AND DEPARTMENT_ID IN (20,50);

```

**Verification table-**

Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT LAST_NAME "Employee", SALARY "Monthly Salary"
FROM EMPLOYEES
WHERE SALARY BETWEEN 5000 AND 12000
AND DEPARTMENT_ID IN (20,50);
```

Employee	Monthly Salary
Lorenz	5800
Fay	6000
Taylor	5000

**Q7. Display the last name and hire date of every employee who was hired in 1994.**

LAST_NAME	HIRE_DATE
Higgins	07-JUN-94
Gietz	07-JUN-94

Ans7.

```
SELECT LAST_NAME, HIRE_DATE
FROM EMPLOYEES
WHERE HIRE_DATE LIKE '%94';
```

**Verification table -**

Work Screen

File or URL:  No file chosen

Enter statements:

```
SELECT LAST_NAME, HIRE_DATE
FROM EMPLOYEES
WHERE HIRE_DATE LIKE '%94';
```

LAST_NAME	HIRE_DATE
Higgins	07-JUN-94
Gietz	07-JUN-94

**Q8. Display the last name and job title of all employees who do not have a manager.**

LAST_NAME	JOB_ID
King	AD_PRES

Ans8.

```
SELECT LAST_NAME, JOB_ID
FROM EMPLOYEES
WHERE MANAGER_ID IS NULL;
```

#### Verification table -

The screenshot shows the Oracle iSQL\*Plus interface. At the top, there's a logo for 'ORACLE iSQL\*Plus' and some navigation icons for Logout, New Session, History, and Pref. Below that is a toolbar with 'File or URL:' (containing 'Choose File' and 'No file chosen'), 'Load Script', and buttons for 'Execute', 'Save Script', 'Clear Screen', and 'Cancel'. The main area is labeled 'Enter statements:' and contains the SQL query: 'SELECT LAST\_NAME, JOB\_ID FROM EMPLOYEES WHERE MANAGER\_ID IS NULL;'. Below the query is a result grid with two rows. The first row has 'King' in the LAST\_NAME column and 'AD\_PRES' in the JOB\_ID column. The second row has 'Taylor' in the LAST\_NAME column and 'ST\_CLERK' in the JOB\_ID column. The grid has 'LAST\_NAME' and 'JOB\_ID' as column headers.

LAST_NAME	JOB_ID
King	AD_PRES
Taylor	ST_CLERK

Q9. Display the last name, salary and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

LAST_NAME	SALARY	COMMISSION_PCT
Abel	11000	.3
Zlotkey	10500	.2
Taylor	8600	.2
Grant	7000	.15

Ans9.

```
SELECT LAST_NAME, SALARY, COMMISSION_PCT
FROM EMPLOYEES
WHERE COMMISSION_PCT IS NOT NULL
```

ORDER BY SALARY DESC, COMMISSION\_PCT DESC;

### Verification table -

The screenshot shows the iSQL\*Plus Work Screen interface. At the top, there's a banner with the Oracle logo and the text "iSQL\*Plus". To the right are icons for Logout, New Session, History, and Help. Below the banner, the title "Work Screen" is displayed, along with a "File or URL:" field containing "Choose File" and a "Load Script" button. A "Enter statements:" text area contains the following SQL query:

```
SELECT LAST_NAME, SALARY, COMMISSION_PCT
FROM EMPLOYEES
WHERE COMMISSION_PCT IS NOT NULL
ORDER BY SALARY DESC, COMMISSION_PCT DESC;
```

Below the query are four buttons: Execute, Save Script, Clear Screen, and Cancel. The results are presented in a table:

LAST_NAME	SALARY	COMMISSION_PCT
Zlotkey	10500	2
Taylor	8600	2
Grant	7000	15
Abel	1100	3

**Q10.** Display the last names of all employees where the third letter of the name is an a.

#### LAST\_NAME

Grant
Whalen

Ans10.

```
SELECT LAST_NAME
FROM EMPLOYEES
WHERE LAST_NAME LIKE '__a%';
```

### Verification table -



Logout New Session History Preferences

Work Screen

File or URL: Choose File No file chosen

Load Script

Enter statements:

```
SELECT LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME LIKE '__a%';
```

Execute Save Script Clear Screen Cancel

LAST\_NAME

Grant
Whalen

**Q11. Display the last names of all employees who have ana and an e in their last name.**

LAST_NAME
De Haan
Davies
Whalen
Hartstein

Ans11.

```
SELECT LAST_NAME
```

```
FROM EMPLOYEES
```

```
WHERE LAST_NAME LIKE '%a%' AND LAST_NAME LIKE '%e%';
```

**Verification table -**



Logout New Session History Preferences

Work Screen

File or URL: Choose File No file chosen

Load Script

Enter statements:

```
SELECT LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME LIKE '%a%' AND LAST_NAME LIKE '%e%';
```

Execute Save Script Clear Screen Cancel

LAST\_NAME

De Haan
Davies
Whalen
Hartstein

Q12. Display the last name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to \$2,500, \$3,500, or \$7,000.

LAST_NAME	JOB_ID	SALARY
Davies	ST_CLERK	3100
Matos	ST_CLERK	2600
Abel	SA_REP	11000
Taylor	SA_REP	8600

Ans12.

```
SELECT LAST_NAME, JOB_ID, SALARY
FROM EMPLOYEES
WHERE JOB_ID='SA_REP' OR JOB_ID='ST_CLERK'
AND SALARY NOT IN (2500,3500,7000);
```

#### Verification table -

The screenshot shows the iSQL\*Plus interface with the following details:

- Header:** iSQL\*Plus, Logout, New Session, History, Pre
- Toolbar:** Work Screen, File or URL: Choose File, No file chosen, Load Script
- Script Area:** Enter statements:
 

```
SELECT LAST_NAME, JOB_ID, SALARY
FROM EMPLOYEES
WHERE JOB_ID='SA_REP' OR JOB_ID='ST_CLERK'
AND SALARY NOT IN (2500,3500,7000);
```
- Buttons:** Execute, Save Script, Clear Screen, Cancel
- Results Area:** Displays the output of the query:
 

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3100
Davies	ST_CLERK	2600
Matos	ST_CLERK	2600
Abel	SA_REP	1100
Taylor	SA_REP	8600
Grant	SA_REP	7000
Taylor	ST_CLERK	5000
- Status:** 7 rows selected.

Q13. Modify lab5\_6.sql to display the last name, salary, and commission for all employees whose commission amount is 200%. Resave lab5\_6.sql as lab5\_13.sql. Rerun the statement in lab5\_13.sql.

Employee	Monthly Salary	COMMISSION_PCT
Zlotkey	10500	2
Taylor	8600	2

Ans13.

```
SELECT LAST_NAME "Employee", SALARY "Monthly Salary", COMMISSION_PCT  
FROM EMPLOYEES  
WHERE COMMISSION_PCT=2;
```

**Verification table-**

ORACLE  
iSQL\*Plus

Work Screen

File or URL: Choose File: No file chosen Load Script

Enter statements:

```
SELECT LAST_NAME "Employee", SALARY "Monthly Salary", COMMISSION_PCT  
FROM EMPLOYEES  
WHERE COMMISSION_PCT=2;
```

Execute Save Script Clear Screen Cancel

Employee	Monthly Salary	COMMISSION_PCT
Zlotkey	10500	2
Taylor	8000	2

# ASSIGNMENT – 6

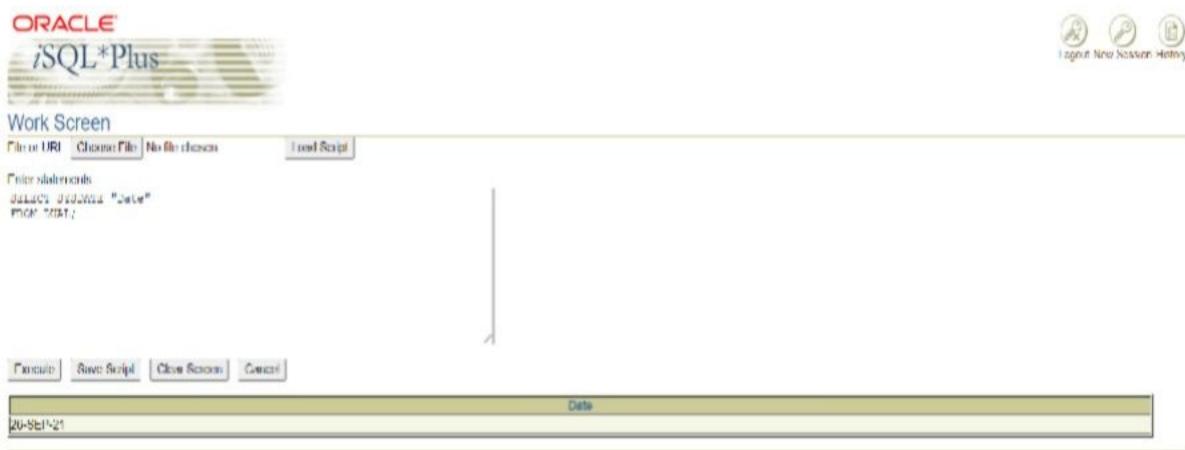
## ON: SINGLE ROW FUNCTIONS IN SQL

**Q1.** Write a query to display the current date. Label the column Date.

Ans1.

```
SELECT SYSDATE "Date"  
FROM DUAL;
```

**Verification table -**



**Q2.** For each employee, display the employee number, last name, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary. Place your SQL statement in a text file named lab6\_2.sql.

Ans2.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY,  
       ROUND(SALARY*1.15, 0) "New Salary"
```

```
FROM EMPLOYEES;
```

**Q3.** Run your query in the file lab lab6\_2.sql.

Ans3.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY,
```

```
ROUND(SALARY*1.15, 0) "New Salary"
```

```
FROM EMPLOYEES;
```

### Verification table -

EMPLOYEE_ID	LAST_NAME	SALARY	New Salary
101 King		2400	2700
101 Kochhar		1700	1950
102 De Haan		1700	1900
103 Ullman		6000	10350
104 Faisel		4500	4875
105 Lorenz		5800	6370
106 Mazouz		3500	4025
107 Rajs		3100	3515
108 Davolio		2800	3120
109 Mettis		2600	2910
110 Vespucci		2500	2875
111 Wilcox		10500	12075
112 Nash		1100	1245
113 Tacer		8600	9545
EMPLOYEE_ID	LAST_NAME	SALARY	New Salary
114 Gundotra		7000	8000
115 Vilains		4400	5000
116 Hartstein		13000	14300
117 Klem		6000	6600
118 Bergl		12000	13800
119 Ernst		8500	9525
120 Ferrer		5000	5625

21 rows selected

**Q4. Modify your query lab6\_2.sql to add a column that subtracts the old salary from the new salary. Label the column Increase. Save the contents of the file as lab6\_4.sql. Run the revised query.**

Ans4.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY,  
ROUND(SALARY*1.15, 0) "New Salary",  
ROUND(SALARY*1.15, 0)-SALARY "Increase"
```

```
FROM EMPLOYEES;
```

### Verification table -

EMPLOYEE_ID	LAST_NAME	SALARY	New Salary	Increase
100	King	2400	2760	360
101	Kochhar	17000	19550	2550
102	De Haan	17000	19550	2550
103	Hundt	9000	10350	1350
104	Eensi	4200	4830	630
105	Lorenz	5800	6670	870
124	Morgos	3500	4025	525
141	Rajs	3100	3565	465
142	Davies	2600	2990	390
143	Matos	2600	2990	390
144	Vargas	2500	2875	375
149	Zlotkey	10500	12075	1575
174	Abel	1100	1265	165
176	Taylor	8600	9890	1290
EMPLOYEE_ID	LAST_NAME	SALARY	New Salary	Increase
178	Grant	7000	8050	1050
200	Whalen	4400	5060	660
201	Hartstein	13000	14950	1950
202	Fay	6000	6900	900
205	Higgins	12000	13800	1800
206	Getz	8300	9545	1245
999	Taylor	5000	5750	750

21 rows selected.

**Q5.** Write a query that displays the employee's last names with the first letter capitalized and all other letters lowercase and the length of the name for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

Ans5.

```
SELECT INITCAP(LAST_NAME) "Name",
       LENGTH (LAST_NAME) "Length"
  FROM EMPLOYEES
 WHERE LAST_NAME LIKE 'J%'
   OR LAST_NAME LIKE 'M%'
   OR LAST_NAME LIKE 'A%'
 ORDER BY LAST_NAME;
```

Verification table -

## Work Screen

File or URL: Choose File No file chosen

Load Script

Enter statements:

```
SELECT INITCAP(LAST_NAME) "Name",
       LENGTH (LAST_NAME) "Length"
  FROM EMPLOYEES
 WHERE LAST_NAME LIKE 'Ab%'  

  OR LAST_NAME LIKE 'Ma%'  

  OR LAST_NAME LIKE 'Al%'  

 ORDER BY LAST_NAME;
```

Execute Save Script Clear Screen Cancel

Name	Length
Abel	4
Malos	5
Moungos	7

**Q6.** For each employee, display the employee's last name, and calculate the number of month between today and the date the employee was hired. Label the column **MONTHS\_WORKED**. Order your results by the number of months employed. Round the number of months up to the closest whole number.

**Note:** Your results will differ.

**Ans6.**

```
SELECT      LAST_NAME,ROUND(MONTHS_BETWEEN(SYSDATE,HIRE_DATE))
MONTHS_WORKED
FROM EMPLOYEES
ORDER BY MONTHS_BETWEEN(SYSDATE,HIRE_DATE);
```

### Verification table -

LAST_NAME	MONTHS_WORKED
Zlotkey	280
Moungos	262
Taylor	260
Grant	260
Lorenz	272
Vargas	279
Taylor	282
Malos	282
Fay	286
Davies	296
Abel	306
Harstein	307
Rajs	311
Higgins	328
LAST_NAME	MONTHS_WORKED
Gietz	326
De Haan	344
Ernst	364
Jnnicold	381
Kochhar	384
Whalen	400
King	411

21 rows selected

Q7. Write a query that produces the following for each employee:  
<employee last name> earns <salary> monthly but wants <3 times salary>. Label the column Dream Salaries.

Ans7.

```
SELECT LAST_NAME || ' earns '  
|| TO_CHAR(SALARY, 'fm$99,999.00')  
|| ' monthly but wants '  
|| TO_CHAR(SALARY * 3, 'fm$99,999.00')  
|| '.' "Dream Salaries"  
  
FROM EMPLOYEES;
```

**Verification table -**

Dream Salaries
King earns \$2,400.00 monthly but wants \$7,200.00
Kochhar earns \$17,000.00 monthly but wants \$51,000.00
De Hean earns \$17,000.00 monthly but wants \$51,000.00
Hundt earns \$9,000.00 monthly but wants \$27,000.00
Cristi earns \$4,200.00 monthly but wants \$12,600.00
Lorentz earns \$5,800.00 monthly but wants \$17,400.00
Moursig earns \$3,500.00 monthly but wants \$10,500.00
Rays earns \$3,100.00 monthly but wants \$9,300.00
Davies earns \$2,600.00 monthly but wants \$7,800.00
Matos earns \$2,600.00 monthly but wants \$7,800.00
Vargas earns \$2,500.00 monthly but wants \$7,500.00
Zlotkey earns \$10,500.00 monthly but wants \$31,500.00
Abel earns \$1,100.00 monthly but wants \$3,300.00
Taylor earns \$8,600.00 monthly but wants \$25,800.00
Dream Salaries
Grant earns \$7,000.00 monthly but wants \$21,000.00
Whalen earns \$4,400.00 monthly but wants \$13,200.00
Harrelson earns \$13,000.00 monthly but wants \$39,000.00
Fay earns \$6,000.00 monthly but wants \$18,000.00
Higgins earns \$12,000.00 monthly but wants \$36,000.00
Gintz earns \$8,300.00 monthly but wants \$24,900.00
Taylor earns \$5,000.00 monthly but wants \$15,000.00

28 rows selected.

Q8. Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with \$. Label the column SALARY.

Ans8.

```
SELECT LAST_NAME,  
LPAD(SALARY, 15, '$') SALARY  
  
FROM EMPLOYEES;
```

**Verification table -**

LAST_NAME	SALARY
King	\$\$\$\$\$\$\$\$\$\$52400
Kochhar	\$\$\$\$\$\$\$\$\$\$17000
De Haan	\$\$\$\$\$\$\$\$\$\$17000
Hundt	\$\$\$\$\$\$\$\$\$\$59000
Crist	\$\$\$\$\$\$\$\$\$\$54200
Lorentz	\$\$\$\$\$\$\$\$\$\$59000
Mourgos	\$\$\$\$\$\$\$\$\$\$55500
Rajs	\$\$\$\$\$\$\$\$\$\$53100
Davies	\$\$\$\$\$\$\$\$\$\$52600
Matos	\$\$\$\$\$\$\$\$\$\$52600
Vargas	\$\$\$\$\$\$\$\$\$\$557500
Zlotkey	\$\$\$\$\$\$\$\$\$\$10500
Abel	\$\$\$\$\$\$\$\$\$\$51100
Taylor	\$\$\$\$\$\$\$\$\$\$59000
LAST_NAME	SALARY
Grant	\$\$\$\$\$\$\$\$\$\$57000
Whalen	\$\$\$\$\$\$\$\$\$\$54100
Harstein	\$\$\$\$\$\$\$\$\$\$13000
Fay	\$\$\$\$\$\$\$\$\$\$59000
Higgins	\$\$\$\$\$\$\$\$\$\$12000
Gitz	\$\$\$\$\$\$\$\$\$\$58500
Taylor	\$\$\$\$\$\$\$\$\$\$55500

21 rows selected.

**Q9. Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to "Monday, the Thirty-First of July, 2000."**

Ans9.

SELECT LAST\_NAME, HIRE\_DATE,

```
TO_CHAR(NEXT_DAY(ADD_MONTHS(HIRE_DATE, 6),'MONDAY'),
' fmDay, "the" Ddspt "of" Month, YYYY". " ') REVIEW
```

FROM EMPLOYEES;

#### Verification table -

LAST_NAME	HIRE_DATE	REVIEW
King	17-JUN-87	Monday, the Twenty-First of December, 1987
Kochhar	21-SEP-89	Monday, the Twenty-Sixth of March, 1990
De Haan	13-JAN-93	Monday, the Nineteenth of July, 1993
Hundt	03-JAN-90	Monday, the Ninth of July, 1990
Crist	21-MAY-91	Monday, the Twenty-Fifth of November, 1991
Lorentz	07-FEB-99	Monday, the Ninth of August, 1999
Mourgos	16-NOV-99	Monday, the Twenty-Second of May, 2000
Rajs	17-OCT-95	Monday, the Twenty-Second of April, 1996
Davies	29-JAN-97	Monday, the Fourth of August, 1997
Matos	15-MAR-98	Monday, the Twenty-First of September, 1998
Vargas	09-JUL-98	Monday, the Eleventh of January, 1999
Zlotkey	29-JAN-00	Monday, the Thirty-First of July, 2000
Abel	11-MAY-98	Monday, the Eighteenth of November, 1998
Taylor	24-MAR-98	Monday, the Twenty-Eighth of September, 1998
LAST_NAME	HIRE_DATE	REVIEW
Grant	24-MAY-99	Monday, the Twenty-Ninth of November, 1999
Whalen	14-SEP-87	Monday, the Twenty-First of March, 1988
Harstein	17-FEB-90	Monday, the Nineteenth of August, 1990
Fay	17-AUG-97	Monday, the Twenty-Third of February, 1998
Higgins	07-JUN-94	Monday, the Twelfth of December, 1994
Gitz	07-JUN-94	Monday, the Twelfth of December, 1994
Taylor	07-JUN-99	Monday, the Thirteenth of December, 1999

21 rows selected.

**Q10. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week starting with**

**Monday.**

Ans10.

```
SELECT LAST_NAME, HIRE_DATE,  
       TO_CHAR(HIRE_DATE, 'DAY') DAY  
FROM EMPLOYEES  
ORDER BY TO_CHAR(HIRE_DATE - 1, 'd');
```

**Verification table -**

LAST_NAME	HIRE_DATE	DAY
Grant	24-MAY-99	MONDAY
Whalen	14-SEP-87	MONDAY
Taylor	07-JUN-99	MONDAY
Ernst	21-MAY-91	TUESDAY
Morgos	16-NOV-99	TUESDAY
Taylor	24-MAR-98	TUESDAY
Cleveland	07-JUN-94	TUESDAY
Higgins	07-JUN-94	TUESDAY
Rajs	17-OCT-95	TUESDAY
King	17-JUN-87	WEDNESDAY
De Haan	13-JAN-83	WEDNESDAY
Davies	29-JAN-87	WEDNESDAY
Hundt	03-JAN-90	WEDNESDAY
Kochhar	21-SEP-89	THURSDAY
LAST_NAME	HIRE_DATE	DAY
Vargas	09-JUL-98	THURSDAY
Zlotkey	28-JAN-00	SATURDAY
Harstain	17-FEB-98	SATURDAY
Abel	11-MAY-96	SATURDAY
Lorenz	07-FEB-99	SUNDAY
Fay	17-AUG-97	SUNDAY
Matos	15-MAR-98	SUNDAY

21 rows selected.

**Q11. Create a query that displays the employees' last names and commission amounts. If an employee does not earn commission, put "No Commission". Label the column COMM.**

Ans11.

```
SELECT LAST_NAME,  
       NVL(TO_CHAR(COMMISSION_PCT), 'No Commission') COMM  
FROM EMPLOYEES;
```

**Verification table -**

LAST_NAME	COMM
King	No Commission
Kochhar	No Commission
De Haan	No Commission
Hundtold	No Commission
Cmst	No Commission
Lorentz	No Commission
Mourgos	No Commission
Rajs	No Commission
Davies	No Commission
Metos	No Commission
Vargas	No Commission
Zlotkey	2
Abel	3
Taylor	9
LAST_NAME	COMM
Grant	15
Whalen	No Commission
Hartstein	No Commission
Fay	No Commission
Higgins	No Commission
Gietz	No Commission
Taylor	No Commission

21 rows selected.

**Q12. Create a query that displays the employees' last names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column EMPLOYEES\_AND THEIR\_SALARIES.**

Ans12.

```
SELECT RPAD(LAST_NAME, 8) || ' ' || RPAD(' ', SALARY/1000+1, '*')
```

EMPLOYEES\_AND\_THEIR\_SALARIES

FROM EMPLOYEES

ORDER BY SALARY DESC;

**Verification table -**

EMPLOYEES_AND_THEIR_SALARIES	
Kochhar *****	
De Haan *****	
Hartstein *****	
Higgins *****	
Zlotkey *****	
Hundtold *****	
Taylor *****	
Gietz *****	
Grant *****	
Fay ****	
Lorentz ****	
Taylor ****	
Whalen ***	
Cmst ***	
EMPLOYEES_AND_THEIR_SALARIES	
Mourgos **	
Rajs **	
Davies **	
Metos **	
Vargas **	
King **	
Abel *	

21 rows selected.

**Q13. Using the DECODE function, write a query that displays the grade of all**

employees based on the value of the column JOB\_ID, as per the following data:

JOB	GRADE
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
None of the above	0

Ans13.

```
SELECT JOB_ID, DECODE (JOB_ID,  
    'ST_CLERK', 'E',  
    'SA_REP', 'D',  
    'IT_PROG', 'C',  
    'ST_MAN', 'B',  
    'AD_PRES', 'A',  
    '0')GRADE
```

FROM EMPLOYEES;

#### Verification table -

JOB_ID	G
AD_PRES	A
AD_VP	B
AD_VP	B
IT_PROG	C
IT_PROG	C
IT_PROG	C
ST_MAN	B
ST_CLERK	E
ST_CLERK	E
ST_CLERK	E
ST_CLERK	F
SA_MAN	B
SA_REP	D
SA_REP	D
JOB_ID	G
SA_REP	D
AD_ASST	D
MKT_MAN	D
MKT_REP	D
AC_MGR	D
AC_ACCOUNT	D
ST_CLERK	E

21 rows selected.

Q14. Rewrite the statement in the preceding question using the CASE syntax.

Ans14.

```
SELECT JOB_ID, CASE JOB_ID  
    WHEN 'ST_CLERK' THEN 'E'  
    WHEN 'SA_REP' THEN 'D'  
    WHEN 'IT_PROG' THEN 'C'  
    WHEN 'ST_MAN' THEN 'B'  
    WHEN 'AD_PRES' THEN 'A'  
    ELSE 'O' END GRADE
```

FROM EMPLOYEES;

**Verification table -**

JOB_ID	G
AD_PRES	A
AD_VP	B
AD_VP	B
IT_PROG	C
IT_PROG	C
IT_PROG	C
ST_MAN	D
ST_CLERK	E
ST_CLERK	E
ST_CLERK	E
ST_CLERK	F
SA_MAN	F
SA_REP	D
SA_REP	D
SA_REP	D
JOB_ID	G
SA_REP	D
AD_ASST	D
MK_MAN	D
MK_REP	D
AC_MGR	D
AC_ACCOUNT	D
ST_CLERK	F

21 rows selected.

## ASSIGNMENT – 7

### ON: JOINS OR DISPLAYING DATA FROM MULTIPLE TABLES

Q1. Write a query to display the last name, department number, and department name for all employees.

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Mourgos	50	Shipping
Rajs	50	Shipping
Davies	50	Shipping
Matos	50	Shipping
Vargas	50	Shipping
Higgins	110	Accounting
Gietz	110	Accounting

19 rows selected.

Ans1.

#### Syntax of Oracle Proprietary Joins (8i and prior):

```
SELECT e.LAST_NAME, e.DEPARTMENT_ID, d.DEPARTMENT_NAME
FROM EMPLOYEES e, DEPARTMENTS d
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID;
```

#### Verification table-

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Kochhar	90	Executive
De Haan	90	Executive
Hundt	60	IT
Ernst	60	IT
Lorentz	50	Shipping
Mourgos	50	Shipping
Rajs	50	Shipping
Davies	50	Shipping
Matos	50	Shipping
Vargas	50	Shipping
Zlotkey	80	Sales
Acel	80	Sales
Taylor	80	Sales
LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Higgins	110	Accounting
Gietz	110	Accounting
Taylor	50	Shipping

20 rows selected.

## Syntax of SQL: 1999 Compliant Joins:

```
SELECT e.LAST_NAME, e.DEPARTMENT_ID, d.DEPARTMENT_NAME  
FROM EMPLOYEES e JOIN DEPARTMENTS d  
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID);
```

## Verification table-

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Kochhar	90	Executive
De Haan	90	Executive
Ullman	60	IT
Ford	60	IT
Lorentz	50	Shipping
Mourgos	50	Shipping
Telesh	50	Shipping
DeWees	50	Shipping
Malos	50	Shipping
Verges	50	Shipping
Zlotkey	80	Sales
Azevedo	80	Sales
Taylor	80	Sales
LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Higgins	110	Accounting
Getz	110	Accounting
Taylor	50	Shipping

20 rows selected.

Q2. Create a unique listing of all jobs that are in department 80. Include the location of department 80 in the output.

JOB_ID	LOCATION_ID
SA_MAN	2500
SA_REP	2500

Ans2.

## Syntax of Oracle Proprietary Joins (8i and prior):

```
SELECT DISTINCT JOB_ID, LOCATION_ID  
FROM EMPLOYEES e, DEPARTMENTS d  
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID  
AND e.DEPARTMENT_ID = 80;
```

## Verification table-

Enter statements

```
SELECT DISTINCT e.JOB_ID, d.LOCATION_ID
FROM EMPLOYEES e, DEPARTMENTS d
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
AND e.DEPARTMENT_ID = 80;
```

Execute Save Script Clear Screen Cancel

JOB_ID	LOCATION_ID
SA MAN	2500
SA REP	2500

## Syntax of SQL: 1999 Compliant Joins:

```
SELECT DISTINCT e.JOB_ID, d.LOCATION_ID
FROM EMPLOYEES e JOIN DEPARTMENTS d
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID)
WHERE e.DEPARTMENT_ID = 80;
```

## Verification table-

Enter statements

```
SELECT DISTINCT e.JOB_ID, d.LOCATION_ID
FROM EMPLOYEES e JOIN DEPARTMENTS d
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID)
WHERE e.DEPARTMENT_ID = 80;
```

Execute Save Script Clear Screen Cancel

JOB_ID	LOCATION_ID
SA MAN	2500
SA REP	2500

Q3. Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission.

LAST_NAME	DEPARTMENT_NAME	LOCATION_ID	CITY
-----------	-----------------	-------------	------

Zlotkey	Sales	2500	Oxford
Abel	Sales	2500	Oxford
Taylor	Sales	2500	Oxford

Ans3.

### Syntax of Oracle Proprietary Joins (8i and prior):

```
SELECT e.LAST_NAME, d.DEPARTMENT_NAME, d.LOCATION_ID, l.CITY
FROM EMPLOYEES e, DEPARTMENTS d, LOCATIONS l
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
AND d.LOCATION_ID = l.LOCATION_ID
AND e.COMMISSION_PCT IS NOT NULL;
```

### Verification table-

Enter statements:

```
SELECT e.LAST_NAME, d.DEPARTMENT_NAME, d.LOCATION_ID, l.CITY
FROM EMPLOYEES e, DEPARTMENTS d, LOCATIONS l
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
AND d.LOCATION_ID = l.LOCATION_ID
AND e.COMMISSION_PCT IS NOT NULL;
```

Execute Save Script Clear Screen Cancel

LAST_NAME	DEPARTMENT_NAME	LOCATION_ID	CITY
Zlotkey	Sales	2500	Oxford
Abel	Sales	2500	Oxford
Taylor	Sales	2500	Oxford

### Syntax of SQL: 1999 Compliant Joins:

```
SELECT LAST_NAME, DEPARTMENT_NAME, LOCATION_ID, CITY
FROM EMPLOYEES e
JOIN DEPARTMENTS d
ON (d.DEPARTMENT_ID = e.DEPARTMENT_ID)
JOIN LOCATIONS l
```

```
ON (d.LOCATION_ID = l.LOCATION_ID)
AND e.COMMISSION_PCT IS NOT NULL;
```

### Verification table-

Enter statements:

```
SELECT LAST_NAME, DEPARTMENT_NAME, LOCATION_ID, CITY
FROM EMPLOYEES e
JOIN DEPARTMENTS d
ON (d.DEPARTMENT_ID = e.DEPARTMENT_ID)
JOIN LOCATIONS l
ON (e.LOCATION_ID = l.LOCATION_ID)
AND e.COMMISSION_PCT IS NOT NULL;
```

LAST_NAME	DEPARTMENT_NAME	LOCATION_ID	CITY
Zulkay	Sales	2500	Oxford
Abel	Sales	2500	Oxford
Taylor	Sales	2500	Oxford

Q4. Display the employee last name and department name for all employees who have an a (lowercase) in their last names. Place your SQL statement in a text file named lab7\_4.sql.

LAST_NAME	DEPARTMENT_NAME
Whalen	Administration
Hartstein	Marketing
Fay	Marketing
Rajs	Shipping
Davies	Shipping
Matos	Shipping
Vargas	Shipping
Taylor	Sales
Kochhar	Executive
De Haan	Executive

10 rows selected.

Ans4.

### Syntax of Oracle Proprietary Joins (8i and prior):

```
SELECT LAST_NAME, DEPARTMENT_NAME
FROM EMPLOYEES e, DEPARTMENTS d
```

```
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
```

```
AND LAST_NAME LIKE '%a%';
```

### Verification table-

Enter statements.

```
SELECT LAST_NAME, DEPARTMENT_NAME  
FROM EMPLOYEES e, DEPARTMENTS d  
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID  
AND LAST_NAME LIKE '%a%';
```

Execute Save Script Clear Screen Cancel

LAST_NAME	DEPARTMENT_NAME
Kochhar	Executive
De Haan	Executive
Rajs	Shipping
Davies	Shipping
Matos	Shipping
Verges	Shipping
Taylor	Sales
Whalen	Administration
Harstein	Marketing
Fay	Marketing
Taylor	Shipping

11 rows selected.

### Syntax of SQL: 1999 Compliant Joins:

```
SELECT e.LAST_NAME, d.DEPARTMENT_NAME
```

```
FROM EMPLOYEES e JOIN DEPARTMENTS d
```

```
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID)
```

```
AND LAST_NAME LIKE '%a%';
```

### Verification table-

Enter statements.

```
SELECT e.LAST_NAME, d.DEPARTMENT_NAME  
FROM EMPLOYEES e JOIN DEPARTMENTS d  
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID)  
AND LAST_NAME LIKE '%a%';
```

Execute Save Script Clear Screen Cancel

LAST_NAME	DEPARTMENT_NAME
Kochhar	Executive
De Haan	Executive
Rajs	Shipping
Davies	Shipping
Matos	Shipping
Verges	Shipping
Taylor	Sales
Whalen	Administration
Harstein	Marketing
Fay	Marketing
Taylor	Shipping

11 rows selected.

**Q5. Write a query to display the last name, job, department number, and department name for all employees who work in Toronto.**

LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
Hartstein	MK_MAN	20	Marketing
Fay	MK_REP	20	Marketing

Ans5.

**Syntax of Oracle Proprietary Joins (8i and prior):**

```
SELECT e.LAST_NAME, e.JOB_ID, e.DEPARTMENT_ID, d.DEPARTMENT_NAME
FROM EMPLOYEES e, DEPARTMENTS d, LOCATIONS l
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
AND d.LOCATION_ID = l.LOCATION_ID
AND l.CITY = 'Toronto';
```

**Verification table-**

Enter statements:

```
SELECT e.LAST_NAME, e.JOB_ID, e.DEPARTMENT_ID, d.DEPARTMENT_NAME
FROM EMPLOYEES e, DEPARTMENTS d, LOCATIONS l
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
AND d.LOCATION_ID = l.LOCATION_ID
AND l.CITY = 'Toronto';
```

Execute Save Script Clear Screen Cancel

LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
Hartstein	MK MAN	20	Marketing
Fay	MK REP	20	Marketing

**Syntax of SQL: 1999 Compliant Joins:**

```
SELECT e.LAST_NAME, e.JOB_ID, e.DEPARTMENT_ID, d.DEPARTMENT_NAME
FROM EMPLOYEES e JOIN DEPARTMENTS d
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID)
```

JOIN LOCATIONS I

ON (d.LOCATION\_ID = l.LOCATION\_ID)

WHERE l.CITY = 'Toronto';

### Verification table-

Enter statements

```
SELECT e.LAST_NAME, e.JOB_ID, e.DEPARTMENT_ID, d.DEPARTMENT_NAME
FROM EMPLOYEES e JOIN DEPARTMENTS d
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID)
JOIN LOCATIONS l
ON (d.LOCATION_ID = l.LOCATION_ID)
WHERE l.CITY = 'Toronto'
```

Execute | Save Script | Clear Screen | Cancel

LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
Hartstein	MK_MAN	20	Marketing
Fay	MK_REP	20	Marketing

Q6. Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. Place your SQL statement in a text file named lab7\_6.sql.

Employee	Emp#	Manager	Mgr#
Kochhar	101	King	100
De Haan	102	King	100
Mourgos	124	King	100
Zlotkey	149	King	100
Abel	174	Zlotkey	149
Taylor	176	Zlotkey	149
Grant	178	Zlotkey	149
Fay	202	Hartstein	201
Gietz	206	Higgins	205

19 rows selected.

Ans6.

Syntax of Oracle Proprietary Joins (8i and prior):

```

SELECT e.LAST_NAME "Employee", e.EMPLOYEE_ID "Emp#",
       m.LAST_NAME "Manager", m.EMPLOYEE_ID "Mgr#"
FROM EMPLOYEES e, EMPLOYEES m
WHERE e.MANAGER_ID = m.EMPLOYEE_ID;

```

### Verification table-

Employee	Emp#	Manager	Mgr#
Kochhar	101	King	100
De Haan	102	King	100
Hundtold	103	De Haan	102
Ernst	104	Hundtold	103
Lorenz	107	King	100
Moungos	124	Moungos	124
Rais	141	Moungos	124
Eavens	142	Moungos	124
Matos	143	Moungos	124
Vargas	144	Moungos	124
Zlotkey	149	King	100
Abel	174	Zlotkey	149
Taylor	176	Zlotkey	149
Grant	178	Zlotkey	149
Employee	Emp#	Manager	Mgr#
Whalen	200	Kochhar	101
Hartstein	201	King	100
Fay	202	Hartstein	201
Higgins	205	Kochhar	101
Gietz	206	Higgins	205

19 rows selected.

### Syntax of SQL: 1999 Compliant Joins:

```

SELECT e.LAST_NAME "Employee", e.EMPLOYEE_ID "Emp#",
       m.LAST_NAME "Manager", m.EMPLOYEE_ID "Mgr#"
FROM EMPLOYEES e JOIN EMPLOYEES m
ON (e.MANAGER_ID = m.EMPLOYEE_ID);

```

### Verification table-

Employee	Emp#	Manager	Mgr#
Kochhar	101	King	100
De Haan	102	King	100
Hundtold	103	De Haan	102
Ernst	104	Hundtold	103
Lorenz	107	King	100
Moungos	124	Moungos	124
Rais	141	Moungos	124
Eavens	142	Moungos	124
Matos	143	Moungos	124
Vargas	144	Moungos	124
Zlotkey	149	King	100
Abel	174	Zlotkey	149
Taylor	176	Zlotkey	149
Grant	178	Zlotkey	149
Employee	Emp#	Manager	Mgr#
Whalen	200	Kochhar	101
Hartstein	201	King	100
Fay	202	Hartstein	201
Higgins	205	Kochhar	101
Gietz	206	Higgins	205

19 rows selected.

**Q7. Modify lab7\_6.sql to display all employees including King, who has no manager. Order the results by the employee number.**

**Place your SQL statement in a text file named lab7\_7.sql. Run the query in lab7\_7.sql.**

Employee	Emp#	Manager	Mgr#
King	100		
Kochhar	101	King	100
De Haan	102	King	100
Hunold	103	De Haan	102
Ernst	104	Hunold	103
Lorentz	107	Hunold	103
Mourgos	124	King	100
Higgins	205	Kochhar	101
Gietz	206	Higgins	206

**20 rows selected.**

**Ans7.**

**Syntax of Oracle Proprietary Joins (8i and prior):**

```
SELECT e.LAST_NAME "Employee", e.EMPLOYEE_ID "Emp#",  
       m.LAST_NAME "Manager", m.EMPLOYEE_ID "Mgr#"  
FROM EMPLOYEES e, EMPLOYEES m  
WHERE e.MANAGER_ID = m.EMPLOYEE_ID(+)  
ORDER BY e.EMPLOYEE_ID;
```

**Verification table-**

Employee	Emp#	Manager	Mgr#
King	100		
Kochhar	101	King	100
De Haan	102	King	100
Hundt	103	De Haan	102
Ernst	104	Hundt	103
Lorenz	107	King	100
Mourgos	124	Mourgos	124
Reus	141	Mourgos	124
Deves	142	Mourgos	124
Malon	143	Mourgos	124
Verges	144	Mourgos	124
Zlotkey	149	King	100
Abel	174	Zlotkey	149
Taylor	176	Zlotkey	149
Employee	Emp#	Manager	Mgr#
Grant	178	Zlotkey	149
Whalen	200	Kochhar	101
Hartstein	201	King	100
Jay	202	Hartstein	201
Higgins	205	Kochhar	101
Gietz	206	Higgins	205
Taylor	995		

21 rows selected.

## Syntax of SQL: 1999 Compliant Joins:

```
SELECT e.LAST_NAME "Employee", e.EMPLOYEE_ID "Emp#",
       m.LAST_NAME "Manager", m.EMPLOYEE_ID "Mgr#"
  FROM EMPLOYEES e
 LEFT OUTER JOIN EMPLOYEES m
    ON (e.MANAGER_ID = m.EMPLOYEE_ID)
 ORDER BY e.EMPLOYEE_ID;
```

## Verification table-

Employee	Emp#	Manager	Mgr#
King	100		
Kochhar	101	King	100
De Haan	102	King	100
Hundt	103	De Haan	102
Ernst	104	Hundt	103
Lorenz	107	King	100
Mourgos	124	Mourgos	124
Reus	141	Mourgos	124
Deves	142	Mourgos	124
Malon	143	Mourgos	124
Verges	144	Mourgos	124
Zlotkey	149	King	100
Abel	174	Zlotkey	149
Taylor	176	Zlotkey	149
Employee	Emp#	Manager	Mgr#
Grant	178	Zlotkey	149
Whalen	200	Kochhar	101
Hartstein	201	King	100
Jay	202	Hartstein	201
Higgins	205	Kochhar	101
Gietz	206	Higgins	205
Taylor	995		

21 rows selected.

**Q8. Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each**

column an appropriate label.

DEPARTMENT	EMPLOYEE	COLLEAGUE
20	Fay	Hartstein
20	Harstein	Fay
50	Davies	Matos
50	Davies	Mourgos
50	Davies	Rajs
50	Davies	Vargas
50	Matos	Davies
50	Matos	Mourgos
50	Matos	Rajs
50	Matos	Vargas
110	Gietz	Higgins
110	Higgins	Gietz

42 rows selected.

Ans8.

#### Syntax of Oracle Proprietary Joins (8i and prior):

```
SELECT e.DEPARTMENT_ID DEPARTMENT, e.LAST_NAME EMPLOYEE,  
      c.LAST_NAME COLLEAGUE  
  FROM EMPLOYEES e, EMPLOYEES c  
 WHERE e.DEPARTMENT_ID = c.DEPARTMENT_ID  
   AND e.EMPLOYEE_ID <> c.EMPLOYEE_ID;
```

#### Verification table-

DEPARTMENT	EMPLOYEE	COLLEAGUE
20	Fay	Hartstein
20	Korelson	Fay
50	Margus	Eduard
50	Teas	Lorenz
50	Taylor	Lorenz
50	Verges	Lorenz
50	Matos	Lorenz
50	Davies	Lorenz
50	Lorenz	Mourgos
50	Raps	Mourgos
50	Taylor	Mourgos
50	Verges	Mourgos
50	Matos	Mourgos
50	Davies	Mourgos
DEPARTMENT	EMPLOYEE	COLLEAGUE
50	Lorenz	Raps
50	Mourgos	Raps
50	Taylor	Raps
50	Verges	Raps
50	Matos	Raps
50	Davies	Raps
50	Lorenz	Taylor
50	Mourgos	Taylor
50	Raps	Taylor
50	Verges	Taylor
50	Matos	Taylor
50	Davies	Taylor
50	Lorenz	Verges
50	Mourgos	Verges
DEPARTMENT	EMPLOYEE	COLLEAGUE
50	Raps	Verges
50	Taylor	Verges
50	Matos	Verges

50	Davies	Margas
50	Lorenz	Matos
50	Mourgos	Matos
50	Raps	Matos
50	Taylor	Matos
50	Verges	Matos
50	Matos	Matos
50	Davies	Matos
50	Lorenz	Davies
50	Mourgos	Davies
50	Raps	Davies
50	Taylor	Davies
DEPARTMENT	EMPLOYEE	COLLEAGUE
50	Verges	Davies
50	Matos	Davies
60	Ernst	Funk
60	Hundt	Ernst
60	Taylor	Funk
60	Abel	Funk
60	Zlotkey	Taylor
60	Abel	Taylor
60	Zlotkey	Abel
60	Taylor	Abel
60	Kochhar	King
60	De Haan	King
60	King	Kochhar
60	De Haan	Kochhar
DEPARTMENT	EMPLOYEE	COLLEAGUE
90	King	De Haan
90	Kochhar	De Haan
110	Gietz	Higgins
110	Higgins	Gietz

60 rows selected.

## Syntax of SQL: 1999 Compliant Joins:

```

SELECT e.DEPARTMENT_ID DEPARTMENT, e.LAST_NAME EMPLOYEE,
       c.LAST_NAME COLLEAGUE
  FROM EMPLOYEES e JOIN EMPLOYEES c
 WHERE e.DEPARTMENT_ID = c.DEPARTMENT_ID
   AND e.EMPLOYEE_ID <> c.EMPLOYEE_ID;
  
```

## Verification table-

DEPARTMENT	EMPLOYEE	COLLEAGUE
SC	Fay	Harrison
SC	Hartstein	Fay
SC	Mourgos	Lorentz
SC	Raph	Lorentz
SC	Teyya	Lorentz
SC	Vargas	Lorentz
SC	Matos	Lorentz
SC	Davies	Lorentz
SC	Veronika	Morgan
SC	Horng	Mourgos
SC	Levi	Mourgos
SC	Verpas	Mourgos
SC	Matos	Mourgos
SC	Davies	Mourgos
SC	Veronika	Mourgos
SC	Horng	Mourgos
SC	Levi	Mourgos
SC	Verpas	Mourgos
SC	Matos	Mourgos
SC	Davies	Mourgos
DEPARTMENT	EMPLOYEE	COLLEAGUE
SC	Veronika	Raph
SC	Horng	Raph
SC	Levi	Itas
SC	Verpas	Itas
SC	Matos	Raph
SC	Davies	Raph
SC	Veronika	Levi
SC	Horng	Levi
SC	Levi	Taylor
SC	Verpas	Taylor
SC	Matos	Levi
SC	Davies	Levi
SC	Veronika	Morgan
SC	Horng	Morgan
DEPARTMENT	EMPLOYEE	COLLEAGUE
SC	SBS	Verges
SC	Teyya	Verges
SC	Matos	Verges

SC	Davies	Verges
SC	Veronika	Miron
SC	Horng	Matos
SC	Levi	Matos
SC	Verpas	Matos
SC	Matos	Veronika
SC	Davies	Veronika
SC	Veronika	Levi
SC	Horng	Levi
SC	Levi	Levi
SC	Verpas	Levi
SC	Matos	Levi
SC	Davies	Levi
DEPARTMENT	EMPLOYEE	COLLEAGUE
SC	Morgan	Davies
SC	Matos	Davies
SC	Levi	Hundred
SC	Verpas	Hundred
SC	Matos	Hundred
SC	Davies	Hundred
SC	Veronika	Hundred
SC	Horng	Hundred
SC	Levi	Hundred
SC	Verpas	Hundred
SC	Matos	Hundred
SC	Davies	Hundred
SC	Veronika	Hundred
SC	Horng	Hundred
SC	Levi	Hundred
SC	Verpas	Hundred
SC	Matos	Hundred
SC	Davies	Hundred
DEPARTMENT	EMPLOYEE	COLLEAGUE
SC	Jong	De Haan
SC	Kochhar	De Haan
IT	Meier	Eiggen
IT	Padil	Eiggen
IT	Pettijohn	Eiggen
IT	Wang	Kochhar
IT	Yousouf	Kochhar

60 rows selected

Q9. Show the structure of the JOB\_GRADES table. Create a query that displays the name, job, department name, salary, and grade for all employees.

Name	Null?	Type
GRADE_LEVEL		VARCHAR2(3)
LOWEST_SAL		NUMBER
HIGHEST_SAL		NUMBER

LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY	GRA
Matos	ST_CLERK	Shipping	2600	A
Vargas	ST_CLERK	Shipping	2500	A
Lorentz	IT_PROG	IT	4200	B
Mourgos	ST_MAN	Shipping	5800	B

Rajs	ST_CLERK	Shipping	3500	B
Davies	ST_CLERK	Shipping	3100	B
Whalen	AD_ASST	Administration	4400	B
De Haan	AD_VP	Executive	17000	E

19 rows selected.

Ans9.

### Syntax of Oracle Proprietary Joins (8i and prior):

DESCRIBE JOB\_GRADES;

SELECT e.LAST\_NAME, e.JOB\_ID, d.DEPARTMENT\_NAME,

e.SALARY, j.GRADE\_LEVEL

FROM EMPLOYEES e, DEPARTMENTS d, JOB\_GRADES j

WHERE e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

AND e.SALARY BETWEEN j.LOWEST\_SAL AND j.HIGHEST\_SAL;

### Verification table-

Name		Null?	Type
GRADE LEVEL			VARCHAR2(25)
LOWEST_SAL			NUMBER(6)
HIGHEST_SAL			NUMBER(6)
LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY
Davies	ST_CLERK	Shipping	2600
Meteks	ST_CLERK	Shipping	2600
Verges	ST_CLERK	Shipping	2500
Emlit	IT_PROG	IT	4200
Lorentz	IT_PROG	Shipping	5000
Mourgos	ST_MAN	Shipping	3500
Rajs	ST_CLERK	Shipping	3100
Whalen	AD_ASST	Administration	4400
Taylor	ST_CLERK	Shipping	5000
Hundtold	IT_PROG	IT	9000
Taylor	SA REP	Sales	8000
Erby	MKT REP	Marketing	6000
Gietz	AC_ACCOUNT	Accounting	8300
Zlotory	SA_MAN	Sales	10500
LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY
Abel	SA HEI	Sales	11000
Hentstein	MKT MAN	Marketing	13000
Higgins	AD_MGR	Accounting	17000
King	AD_PRS	Executive	24000
Kochhar	AD_VP	Executive	17000
De Haan	AD_VP	Executive	17000

20 rows selected.

### Syntax of SQL: 1999 Compliant Joins:

DESCRIBE JOB\_GRADES;

SELECT e.LAST\_NAME, e.JOB\_ID, d.DEPARTMENT\_NAME,

```

e.SALARY, j.GRADE_LEVEL
FROM EMPLOYEES E JOIN DEPARTMENTS d
ON (e.DEPARTMENT_ID = d.DEPARTMENT_ID)
JOIN JOB_GRADES j
ON (e.SALARY BETWEEN j.LOWEST_SAL AND j.HIGHEST_SAL);

```

### Verification table-

Name	Null?	Type
GRADE_LEVEL		[VARCHAR2(3)]
LOWEST_SAL		[NUMBER(6)]
HIGHEST_SAL		[NUMBER(6)]
LAST_NAME	JOB_ID	DEPARTMENT_NAME
Davies	ST_CLERK	Shipping
Matos	ST_CLERK	Shipping
Vargas	ST_CLERK	Shipping
Ernst	IT_PROG	IT
Lorentz	IT_PROG	Shipping
Mourgos	ST_MAN	Shipping
Reus	ST_CLERK	Shipping
De Haan	SA_ASST	Administration
Taylor	ST_CLERK	Shipping
Hundt	IT_PROG	IT
Taylor	SA REP	Sales
Fay	MK_Rep	Marketing
Getz	AC_ACCOUNT	Accounting
Zlotkey	SA_MAN	Sales
LAST_NAME	JOB_ID	DEPARTMENT_NAME
Abed	SA REP	Sales
Harstein	MK_MAN	Marketing
Higgins	AC_MGR	Accounting
King	AN_PRES	Executive
Kochhar	AD_VP	Executive
De Haan	AD_VP	Executive

20 rows selected.

Q10. Create a query to display the name and hire date of any employee hired after employee Davies.

LAST_NAME	HIRE_DATE
Lorentz	07-FEB-99
Mourgos	16-NOV-99
Matos	15-MAR-98
Vargas	09-JUL-98
Zlotkey	29-JAN-00
Taylor	24-MAR-98
Grant	24-MAY-99
Fay	17-AUG-97

8 rows selected.

Ans10.

### Syntax of Oracle Proprietary Joins (8i and prior):

```

SELECT e.LAST_NAME, e.HIRE_DATE
FROM EMPLOYEES e, EMPLOYEES d
WHERE d.LAST_NAME = 'Davies'
AND d.HIRE_DATE < e.HIRE_DATE;

```

### Verification table-

Enter statements:

```

SELECT e.LAST_NAME, e.HIRE_DATE
FROM EMPLOYEES e, EMPLOYEES d
WHERE d.LAST_NAME = 'Davies'
AND d.HIRE_DATE < e.HIRE_DATE;

```

Execute | Save Script | Clear Screen | Cancel

LAST_NAME	HIRE_DATE
Lorenz	07-FEB-99
Mourgos	16-NOV-99
Matos	15-MAR-98
Vargas	09-JUL-98
Zlotkey	29-JAN-00
Taylor	24-MAR-98
Grant	24-MAY-99
Jay	17-AUG-97
Taylor	07-JUN-99

9 rows selected

### Syntax of SQL: 1999 Compliant Joins:

```

SELECT e.LAST_NAME, e.HIRE_DATE
FROM EMPLOYEES e JOIN EMPLOYEES d
ON (d.LAST_NAME = 'Davies')
WHERE d.HIRE_DATE < e.HIRE_DATE ;

```

### Verification table-

```
Enter statements:
SELECT e.LAST_NAME, e.HIRE_DATE
FROM EMPLOYEES e JOIN EMPLOYEES m
ON (e.LAST_NAME = 'Davies')
WHERE e.HIRE_DATE < m.HIRE_DATE ;
```

LAST_NAME	HIRE_DATE
Lorentz	07-FEB-99
Mourgos	16-NOV-99
Matos	15-MAR-98
Vargas	09-JUL-98
Zlotkey	29-JAN-00
Taylor	24-MAR-98
Grant	24-MAY-99
Fay	17-AUG-97
Taylor	07-JUN-99

9 rows selected

---

**Q11.** Display the names and hire dates for all employees who were hired before their managers, along with their manager's names and hire dates. Label the column Employee, Emp Hired, Manager, and Mgr Hired, respectively.

Employee	Emp Hired	Manager	Mgr Hired
Whalen	17-SEP-87	Kochhar	21-SEP-89
Hunold	03-JAN-90	De Haan	13-JAN-93
Rajs	17-OCT-95	Mourgos	16-NOV-99
Davies	29-JAN-97	Mourgos	16-NOV-99
Matos	15-MAR-98	Mourgos	16-NOV-99
Vargas	09-JUL-98	Mourgos	16-NOV-99
Abel	11-MAY-96	Zlotkey	29-JAN-00
Taylor	24-MAR-98	Zlotkey	29-JAN-00
Grant	24-MAY-99	Zlotkey	29-JAN-00

9 rows selected.

Ans11.

### Syntax of Oracle Proprietary Joins (8i and prior):

SELECT e.LAST\_NAME "Employee", e.HIRE\_DATE "Emp Hired",

m.LAST\_NAME "Manager", m.HIRE\_DATE "Mgr Hired"

FROM EMPLOYEES e, EMPLOYEES m

WHERE e.MANAGER\_ID = m.EMPLOYEE\_ID

AND e.HIRE\_DATE < m.HIRE\_DATE;

## Verification table-

Enter statements:

```

SELECT e.LAST_NAME "Employee", e.HIRE_DATE "Emp Hired",
       m.LAST_NAME "Manager", m.HIRE_DATE "Mgr Hired"
  FROM EMPLOYEES e, EMPLOYEES m
 WHERE e.MANAGER_ID = m.EMPLOYEE_ID
   AND e.HIRE_DATE < m.HIRE_DATE;

```

Execute | Save Script | Clear Screen | Cancel

Employee	Emp Hired	Manager	Mgr Hired
Hundtold	03-JAN-90	De Haan	13-JAN-93
Rajs	17-OCT-95	Mourgos	16-NOV-99
Davies	29-JAN-97	Mourgos	16-NOV-99
Malos	15-MAR-98	Mourgos	16-NOV-99
Vargas	09-JUL-98	Mourgos	16-NOV-99
Abel	11-MAY-98	Zlotkey	29-JAN-00
Taylor	24-MAR-98	Zlotkey	29-JAN-00
Grant	24-MAY-99	Zlotkey	29-JAN-00
Whalen	14-SEP-97	Kochhar	21-SEP-99

9 rows selected.

## Syntax of SQL: 1999 Compliant Joins:

```

SELECT e.LAST_NAME "Employee", e.HIRE_DATE "Emp Hired",
       m.LAST_NAME "Manager", m.HIRE_DATE "Mgr Hired"
  FROM EMPLOYEES e JOIN EMPLOYEES m
    ON (e.MANAGER_ID = m.EMPLOYEE_ID)
   WHERE e.HIRE_DATE < m.HIRE_DATE;

```

## Verification table-

Enter statements:

```

SELECT e.LAST_NAME "Employee", e.HIRE_DATE "Emp Hired",
       m.LAST_NAME "Manager", m.HIRE_DATE "Mgr Hired"
  FROM EMPLOYEES e JOIN EMPLOYEES m
    ON (e.MANAGER_ID = m.EMPLOYEE_ID)
   WHERE e.HIRE_DATE < m.HIRE_DATE;

```

Execute | Save Script | Clear Screen | Cancel

Employee	Emp Hired	Manager	Mgr Hired
Hundtold	03-JAN-90	De Haan	13-JAN-93
Rajs	17-OCT-95	Mourgos	16-NOV-99
Davies	29-JAN-97	Mourgos	16-NOV-99
Malos	15-MAR-98	Mourgos	16-NOV-99
Vargas	09-JUL-98	Mourgos	16-NOV-99
Abel	11-MAY-98	Zlotkey	29-JAN-00
Taylor	24-MAR-98	Zlotkey	29-JAN-00
Grant	24-MAY-99	Zlotkey	29-JAN-00
Whalen	14-SEP-97	Kochhar	21-SEP-99

9 rows selected.

## ASSIGNMENT – 8

### ON: GROUP FUNCTIONS AND AGGREGATING DATA

Determine the validity of the following three statements. Circle either True or False.

**Q1. Group functions work across many rows to produce one result per group.**

Ans1.

True

**Q2. Group functions include nulls in calculations.**

Ans2.

False

We can use NVL to include null values.

**Q3. The WHERE clause restricts rows prior to inclusion in a group calculation.**

Ans3.

True

**Q4. Display the highest, lowest, sum, average salary of all employees. Label the columns Maximum, Minimum, Sum, Average, respectively. Round your results to the nearest whole number. Place your SQL statement in a text file named lab8\_4.sql.**

Maximum	Minimum	Sum	Average
24000	2500	175500	8775

Ans4.

```
SELECT ROUND(MAX(SALARY),0) "Maximum",
       ROUND(MIN(SALARY),0) "Minimum",
       ROUND(SUM(SALARY),0) "Sum",
       ROUND(AVG(SALARY),0) "Average"
```

FROM EMPLOYEES;

### Verification table –

The screenshot shows the Oracle SQL\*Plus interface. At the top, it says "ORACLE iSQL\*Plus" and "Local Net Service Hahn B". Below that is a "Work Screen" header with tabs for "File or URL", "Choose File", "No file chosen", and "Load Script". The main area contains the following SQL code:

```
SELECT ROUND(MAX(SALARY),0) "Maximum",
       ROUND(MIN(SALARY),0) "Minimum",
       ROUND(TOTAL(SALARY),0) "Sum",
       ROUND(AVG(SALARY),0) "Average"
  FROM EMPLOYEES;
```

Below the code, there are buttons for "Execute", "New Script", "Clear Screen", and "Cancel". The results are displayed in a table:

Maximum	Minimum	Sum	Average
17000	100	145600	8913

Q5. Modify the query in lab8\_4.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab8\_4.sql to lab8\_5.sql. Run the statement in a lab8\_5.sql.

JOB_ID	Maximum	Minimum	Sum	Average
AC_ACCOUNT	8300	8300	8300	8300
AC_MGR	12000	12000	12000	12000
AD_ASST	4400	4400	4400	4400
AD_PRES	24000	24000	24000	24000
AD_VP	17000	17000	34000	17000
IT_PROG	9000	4200	19200	6400
MK_MAN	13000	13000	13000	13000
MK_REP	6000	6000	6000	6000
SA_MAN	10500	10500	10500	10500
SA_REP	11000	7000	26600	8867
ST_CLERK	3500	2500	11700	2925
ST_MAN	5800	5800	5800	5800

Ans5.

```
SELECT JOB_ID, ROUND(MAX(SALARY),0) "Maximum",
       ROUND(MIN(SALARY),0) "Minimum",
       ROUND(SUM(SALARY),0) "Sum",
       ROUND(AVG(SALARY),0) "Average"
```

FROM EMPLOYEES

GROUP BY JOB\_ID;

**Verification table –**

JOB_ID	Minimum	Maximum	Sum	Average
AC_ACCOUNT	8300	8300	8300	8300
AC_MGR	12000	12000	12000	12000
AD_ASST	4400	4400	4400	4400
AD_PRES	2400	2400	2400	2400
AD_VP	17000	17000	17000	17000
IT_PROG	9000	42000	19000	8333
MK_MAN	12000	12000	12000	12000
MK_REP	8000	8000	8000	8000
SA_MAN	10500	10500	10500	10500
SA_REP	8000	10000	18000	9000
ST_CLERK	5000	25000	15000	3167
ST_MAN	10000	10000	10000	10000

12 rows selected.

**Q6. Write a query to display the number of people with the same job.**

JOB_ID	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1
AD_PRES	1
AD_VP	2
IT_PROG	3
MK_MAN	1
MK_REP	1
SA_MAN	1
SA_REP	3
ST_CLERK	4
ST_MAN	1

12 rows selected.

Ans6.

SELECT JOB\_ID, COUNT(\*)

FROM EMPLOYEES

GROUP BY JOB\_ID;

**Verification table –**

iSQL\*Plus  
Work Screen

File or URL Choose File No file chosen Load Script

Enter statements:

```
SELECT JOB_ID, COUNT(*)  
FROM EMPLOYEES  
GROUP BY JOB_ID;
```

Execute Save Script Clear Screen Cancel

JOB_ID	COUNT(*)
AC_ACCOUNT	1
AD_ASST	1
AD_PRES	1
AD_VP	2
IT_PROG	3
MK_MAN	1
MK_REP	1
SA_MAN	3
SA_REP	5
ST_CLERK	1

12 rows selected.

Q7. Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER\_ID column to determine the number of managers.

Number of Managers
8

Ans7.

```
SELECT COUNT(DISTINCT MANAGER_ID) "Number of Managers"  
FROM EMPLOYEES;
```

Verification table –

ORACLE  
iSQL\*Plus

Work Screen

File or URL Choose File No file chosen Load Script

Enter statements:

```
SELECT COUNT(DISTINCT MANAGER_ID) "Number of Managers"  
FROM EMPLOYEES;
```

Execute Save Script Clear Screen Cancel

Number of Managers
8

Q8. Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

DIFFERENCE
21500

Ans8.

```
SELECT MAX(SALARY)-MIN(SALARY) DIFFERENCE  
FROM EMPLOYEES;
```

**Verification table –**



Q9. Display the manager number and the salary of the lowest paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is less than \$6,000. Sort the output in descending order of salary.

MANAGER_ID	MIN(SALARY)
102	9000
205	8300
149	7000

Ans9.

```
SELECT MANAGER_ID, MIN(SALARY)  
FROM EMPLOYEES  
WHERE MANAGER_ID IS NOT NULL  
GROUP BY MANAGER_ID  
HAVING MIN(SALARY) >= 6000  
ORDER BY MIN(SALARY) DESC;
```

**Verification table –**

The screenshot shows the Oracle SQL\*Plus interface. At the top, it says "ORACLE" and "SQL\*Plus". Below that is a toolbar with icons for Help, Home, and Log Off. The main area is titled "Work Screen" and has tabs for "File or URL", "Choose File", "No file chosen", and "Load Script". A "Enter Statements" text area contains the following SQL query:

```
SELECT MANAGER_ID, MIN(SALARY)
FROM EMPLOYEES
WHERE MANAGER_ID IS NOT NULL
GROUP BY MANAGER_ID
HAVING MIN(SALARY) >= 4000
ORDER BY MIN(SALARY) DESC;
```

Below the query are buttons for "Execute", "Save Script", "Clear Screen", and "Cancel". The results are displayed in a grid with two columns: "MANAGER\_ID" and "MIN(SALARY)". The data shows three rows with Manager IDs 100, 200, and 201, and corresponding minimum salaries of \$5000, \$5000, and \$5000 respectively.

**Q10.** Write a query to display each department's name, location, number of employees, and the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.

Name	Location	Number of People	Salary
Accounting	1700	2	10150
Administration	1700	1	4400
Executive	1700	3	19333.33
IT	1400	3	6400
Marketing	1800	2	9500
Sales	2500	3	10033.33
Shipping	1500	5	3500

7 rows selected.

Ans10.

```
SELECT d.department_name "Name", d.location_id "Location",
```

```
COUNT(*) "Number of People",
```

```
ROUND(AVG(Salary),2) "Salary"
```

```
FROM employees e, departments d
```

```
WHERE e.department_id = d.department_id
```

```
GROUP BY d.department_name, d.location_id;
```

Verification table –



Work Screen

File or URL | Choose File | No No chosen

Load Script

```
Prvior statements
SELECT COUNT(*) "Name", d.department_id "Location",
       COUNT(*) "Number of People",
       ROUND(AVG(salary),2) "Salary"
FROM employees e, departments d
WHERE e.department_id=d.department_id
GROUP BY d.department_name, d.location_id;
```

Execute | Save Script | Clear Screen | Cancel

Name	Location	Number of People	Salary
IT	1400	20	6000
Sales	2900	3	6733.33
Shipping	1500	2	5987.50
Executive	1700	2	12133.33
Marketing	1800	20	9500
Accounting	1700	20	10100
Administration	1700	1	6400

7 rows selected.

**Q11. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.**

TOTAL	1995	1996	1997	1998
20	1	2	2	3

Ans11.

```
SELECT COUNT(*) TOTAL,
```

```
    SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1995,1,0)) "1995",
```

```
    SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1996,1,0)) "1996",
```

```
    SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1997,1,0)) "1997",
```

```
    SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1998,1,0)) "1998"
```

```
FROM EMPLOYEES;
```

### Verification table –



Work Screen

File or URL | Choose File | No No chosen

Load Script

```
Prvior statements
SELECT COUNT(*) TOTAL,
       SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1995,1,0)) "1995",
       SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1996,1,0)) "1996",
       SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1997,1,0)) "1997",
       SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1998,1,0)) "1998"
FROM EMPLOYEES;
```

Execute | Save Script | Clear Screen | Cancel

TOTAL	1995	1996	1997	1998
20	1	2	2	3

Q12. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
AC_ACCOUNT					8300
AC_MGR					12000
AD_ASST					4400
AD_PRES				24000	24000
AD_VP				34000	34000
IT_PROG					19200
MK_MAN	13000				13000
MK_REP	6000				6000
SA_MAN			10500		10500
SA_REP			19600		26600
ST_CLERK		11700			11700
ST_MAN		5800			5800

12 rows selected.

Ans12.

```

SELECT JOB_ID "Job",
       SUM(DECODE(DEPARTMENT_ID, 20, SALARY)) "Dept 20",
       SUM(DECODE(DEPARTMENT_ID, 50, SALARY)) "Dept 50",
       SUM(DECODE(DEPARTMENT_ID, 80, SALARY)) "Dept 80",
       SUM(DECODE(DEPARTMENT_ID, 90, SALARY)) "Dept 90",
       SUM(SALARY) "Total"
  
```

FROM EMPLOYEES

GROUP BY JOB\_ID;

Verification table –



## ASSIGNMENT – 9

### ON: SUBQUERIES

Q1. Write a query to display the last name and hire date of any employee in the same department as Zlotkey. Exclude Zlotkey.

LAST_NAME	HIRE_DATE
Abel	11-MAY-96
Taylor	24-MAR-98

Ans1.

```
SELECT LAST_NAME, HIRE_DATE  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID  
                        FROM EMPLOYEES  
                        WHERE LAST_NAME = 'Zlotkey')  
AND LAST_NAME <> 'Zlotkey';
```

#### Verification table -

The screenshot shows the Oracle SQL\*Plus interface. In the command window, the following SQL query is entered:

```
SELECT LAST_NAME, HIRE_DATE  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID  
                        FROM EMPLOYEES  
                        WHERE LAST_NAME = 'Zlotkey')  
AND LAST_NAME <> 'Zlotkey';
```

Below the command window, the results are displayed in a grid:

	LAST_NAME	HIRE_DATE
Abel		11-MAY-96
Taylor		24-MAR-98

Q2. Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.

EMPLOYEE_ID	LAST_NAME	SALARY
103	Hunold	9000
149	Zlotkey	10500

174	Abel	11000
205	Higgins	12000
201	Hartstein	13000
101	Kochhar	17000
102	De Haan	17000
100	King	24000

8 rows selected.

Ans2.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY
FROM EMPLOYEES
WHERE SALARY > (SELECT AVG(SALARY)
                  FROM EMPLOYEES)
```

ORDER BY SALARY ASC;

### Verification table -

EMPLOYEE_ID	LAST_NAME	SALARY
174	Abel	11000
205	Higgins	12000
201	Hartstein	13000
101	Kochhar	17000
102	De Haan	17000
100	King	24000

Q3. Write a query that displays the employee numbers and last names of all employees who work in a department with any employee whose last name contains a u. Place your SQL statement in a text file named lab9\_3.sql. Run your query.

EMPLOYEE_ID	LAST_NAME
124	Mourgos
141	Rajs
142	Davies
143	Matos
144	Vargas
103	Hunold

104	Ernst
107	Lorentz

8 rows selected.

Ans3.

```
SELECT EMPLOYEE_ID, LAST_NAME
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID
FROM EMPLOYEES
WHERE LAST_NAME LIKE '%u%');
```

### Verification table -

The screenshot shows the Oracle SQL\*Plus interface. The command entered is:

```
SELECT EMPLOYEE_ID, LAST_NAME
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID
FROM EMPLOYEES
WHERE LAST_NAME LIKE '%u%');
```

The results table has columns 'EMPLOYEE\_ID' and 'LAST\_NAME'. The data is as follows:

EMPLOYEE_ID	LAST_NAME
104	Ernst
107	Lorentz
110	Tucker
111	Colleen
112	Sergio
113	Klaus
114	Elma
115	Mark
116	James
117	Michael
118	Howard
119	David
120	Patricia
121	John
122	Adam
123	Robert
124	Laura
125	David
126	Mike
127	John
128	Patricia
129	David
130	Adam
131	Robert
132	Laura
133	David
134	Mike
135	John
136	Patricia
137	David
138	Adam
139	Robert
140	Laura
141	David
142	Mike
143	John
144	Patricia
145	David
146	Adam
147	Robert
148	Laura
149	David
150	Mike
151	John
152	Patricia
153	David
154	Adam
155	Robert
156	Laura
157	David
158	Mike
159	John
160	Patricia
161	David
162	Adam
163	Robert
164	Laura
165	David
166	Mike
167	John
168	Patricia
169	David
170	Adam
171	Robert
172	Laura
173	David
174	Mike
175	John
176	Patricia
177	David
178	Adam
179	Robert
180	Laura
181	David
182	Mike
183	John
184	Patricia
185	David
186	Adam
187	Robert
188	Laura
189	David
190	Mike
191	John
192	Patricia
193	David
194	Adam
195	Robert
196	Laura
197	David
198	Mike
199	John
200	Patricia
201	David
202	Adam
203	Robert
204	Laura
205	David
206	Mike
207	John
208	Patricia
209	David
210	Adam
211	Robert
212	Laura
213	David
214	Mike
215	John
216	Patricia
217	David
218	Adam
219	Robert
220	Laura
221	David
222	Mike
223	John
224	Patricia
225	David
226	Adam
227	Robert
228	Laura
229	David
230	Mike
231	John
232	Patricia
233	David
234	Adam
235	Robert
236	Laura
237	David
238	Mike
239	John
240	Patricia
241	David
242	Adam
243	Robert
244	Laura
245	David
246	Mike
247	John
248	Patricia
249	David
250	Adam
251	Robert
252	Laura
253	David
254	Mike
255	John
256	Patricia
257	David
258	Adam
259	Robert
260	Laura
261	David
262	Mike
263	John
264	Patricia
265	David
266	Adam
267	Robert
268	Laura
269	David
270	Mike
271	John
272	Patricia
273	David
274	Adam
275	Robert
276	Laura
277	David
278	Mike
279	John
280	Patricia
281	David
282	Adam
283	Robert
284	Laura
285	David
286	Mike
287	John
288	Patricia
289	David
290	Adam
291	Robert
292	Laura
293	David
294	Mike
295	John
296	Patricia
297	David
298	Adam
299	Robert
300	Laura
301	David
302	Mike
303	John
304	Patricia
305	David
306	Adam
307	Robert
308	Laura
309	David
310	Mike
311	John
312	Patricia
313	David
314	Adam
315	Robert
316	Laura
317	David
318	Mike
319	John
320	Patricia
321	David
322	Adam
323	Robert
324	Laura
325	David
326	Mike
327	John
328	Patricia
329	David
330	Adam
331	Robert
332	Laura
333	David
334	Mike
335	John
336	Patricia
337	David
338	Adam
339	Robert
340	Laura
341	David
342	Mike
343	John
344	Patricia
345	David
346	Adam
347	Robert
348	Laura
349	David
350	Mike
351	John
352	Patricia
353	David
354	Adam
355	Robert
356	Laura
357	David
358	Mike
359	John
360	Patricia
361	David
362	Adam
363	Robert
364	Laura
365	David
366	Mike
367	John
368	Patricia
369	David
370	Adam
371	Robert
372	Laura
373	David
374	Mike
375	John
376	Patricia
377	David
378	Adam
379	Robert
380	Laura
381	David
382	Mike
383	John
384	Patricia
385	David
386	Adam
387	Robert
388	Laura
389	David
390	Mike
391	John
392	Patricia
393	David
394	Adam
395	Robert
396	Laura
397	David
398	Mike
399	John
400	Patricia
401	David
402	Adam
403	Robert
404	Laura
405	David
406	Mike
407	John
408	Patricia
409	David
410	Adam
411	Robert
412	Laura
413	David
414	Mike
415	John
416	Patricia
417	David
418	Adam
419	Robert
420	Laura
421	David
422	Mike
423	John
424	Patricia
425	David
426	Adam
427	Robert
428	Laura
429	David
430	Mike
431	John
432	Patricia
433	David
434	Adam
435	Robert
436	Laura
437	David
438	Mike
439	John
440	Patricia
441	David
442	Adam
443	Robert
444	Laura
445	David
446	Mike
447	John
448	Patricia
449	David
450	Adam
451	Robert
452	Laura
453	David
454	Mike
455	John
456	Patricia
457	David
458	Adam
459	Robert
460	Laura
461	David
462	Mike
463	John
464	Patricia
465	David
466	Adam
467	Robert
468	Laura
469	David
470	Mike
471	John
472	Patricia
473	David
474	Adam
475	Robert
476	Laura
477	David
478	Mike
479	John
480	Patricia
481	David
482	Adam
483	Robert
484	Laura
485	David
486	Mike
487	John
488	Patricia
489	David
490	Adam
491	Robert
492	Laura
493	David
494	Mike
495	John
496	Patricia
497	David
498	Adam
499	Robert
500	Laura
501	David
502	Mike
503	John
504	Patricia
505	David
506	Adam
507	Robert
508	Laura
509	David
510	Mike
511	John
512	Patricia
513	David
514	Adam
515	Robert
516	Laura
517	David
518	Mike
519	John
520	Patricia
521	David
522	Adam
523	Robert
524	Laura
525	David
526	Mike
527	John
528	Patricia
529	David
530	Adam
531	Robert
532	Laura
533	David
534	Mike
535	John
536	Patricia
537	David
538	Adam
539	Robert
540	Laura
541	David
542	Mike
543	John
544	Patricia
545	David
546	Adam
547	Robert
548	Laura
549	David
550	Mike
551	John
552	Patricia
553	David
554	Adam
555	Robert
556	Laura
557	David
558	Mike
559	John
560	Patricia
561	David
562	Adam
563	Robert
564	Laura
565	David
566	Mike
567	John
568	Patricia
569	David
570	Adam
571	Robert
572	Laura
573	David
574	Mike
575	John
576	Patricia
577	David
578	Adam
579	Robert
580	Laura
581	David
582	Mike
583	John
584	Patricia
585	David
586	Adam
587	Robert
588	Laura
589	David
590	Mike
591	John
592	Patricia
593	David
594	Adam
595	Robert
596	Laura
597	David
598	Mike
599	John
600	Patricia
601	David
602	Adam
603	Robert
604	Laura
605	David
606	Mike
607	John
608	Patricia
609	David
610	Adam
611	Robert
612	Laura
613	David
614	Mike
615	John
616	Patricia
617	David
618	Adam
619	Robert
620	Laura
621	David
622	Mike
623	John
624	Patricia
625	David
626	Adam
627	Robert
628	Laura
629	David
630	Mike
631	John
632	Patricia
633	David
634	Adam
635	Robert
636	Laura
637	David
638	Mike
639	John
640	Patricia
641	David
642	Adam
643	Robert
644	Laura
645	David
646	Mike
647	John
648	Patricia
649	David
650	Adam
651	Robert
652	Laura
653	David
654	Mike
655	John
656	Patricia
657	David
658	Adam
659	Robert
660	Laura
661	David
662	Mike
663	John
664	Patricia
665	David
666	Adam
667	Robert
668	Laura
669	David
670	Mike
671	John
672	Patricia
673	David
674	Adam
675	Robert
676	Laura
677	David
678	Mike
679	John
680	Patricia
681	David
682	Adam
683	Robert
684	Laura
685	David
686	Mike
687	John
688	Patricia
689	David
690	Adam
691	Robert
692	Laura
693	David
694	Mike
695	John
696	Patricia
697	David
698	Adam
699	Robert
700	Laura
701	David
702	Mike
703	John
704	Patricia
705	David
706	Adam

```

FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID
                           FROM DEPARTMENTS
                           WHERE LOCATION_ID = 1700);

```

**Verification table -**

The screenshot shows the Oracle SQL\*Plus interface with the title bar "ORACLE /SQL\*Plus". Below it is a toolbar with icons for Undo, Redo, and Save. The main area is titled "Work Screen" and contains a "File or URL" field with "None chosen" and a "Load Script" button. Under "Edit statements", the query is displayed:

```

SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID
                           FROM DEPARTMENTS
                           WHERE LOCATION_ID = 1700);

```

Below the query is a table with the following data:

LAST_NAME	DEPARTMENT_ID	JOB_ID
Kochhar	100	AD_PRES
De Haan	100	AD_VP
Mourgos	100	AD_ASST
Zlotkey	100	TRAVEL_MGR
Hartstein	100	TRAVEL_ACCOUNTANT

At the bottom left, it says "8 rows selected".

**Q5. Display the last name and salary of every employee who reports to King.**

LAST_NAME	SALARY
Kochhar	17000
De Haan	17000
Mourgos	5800
Zlotkey	10500
Hartstein	13000

Ans5.

```

SELECT LAST_NAME, SALARY
FROM EMPLOYEES
WHERE MANAGER_ID = (SELECT EMPLOYEE_ID
                           FROM EMPLOYEES
                           WHERE LAST_NAME = 'King');

```

**Verification table -**

```

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jul 19 10:45:00 2013
Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Work Screen
File or URL | Choose File | No file chosen | Load Script

Execution statements
SELECT LAST_NAME, SALARY
FROM EMPLOYEES
WHERE MANAGER_ID = (SELECT EMPLOYEE_ID
                      FROM EMPLOYEES
                      WHERE LAST_NAME = 'King');

Execute | Save Script | Clear Screen | Cancel |
```

LAST_NAME	SALARY
Kochhar	11500
De Haan	11500
Lamont	8000
Covey	11500
Harrell	11500

**Q6. Display the department number, last name, and job ID for every employee in the Executive department.**

DEPARTMENT_ID	LAST_NAME	JOB_ID
90	King	AD_PRES
90	Kochhar	AD_VP
90	De Haan	AD_VP

Ans6.

```

SELECT DEPARTMENT_ID, LAST_NAME, JOB_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID
                       FROM DEPARTMENTS
                       WHERE DEPARTMENT_NAME = 'Executive');
```

### Verification table -

```

SQL*Plus: Release 11.2.0.1.0 Production on Fri Jul 19 10:45:00 2013
Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Work Screen
File or URL | Choose File | No file chosen | Load Script

Execution statements
SELECT DEPARTMENT_ID, LAST_NAME, JOB_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID
                       FROM DEPARTMENTS
                       WHERE DEPARTMENT_NAME = 'Executive');

Execute | Save Script | Clear Screen | Cancel |
```

DEPARTMENT_ID	LAST_NAME	JOB_ID
90	King	AD_PRES
90	Kochhar	AD_VP
90	De Haan	AD_VP

**Q7. Modify the query in lab9\_3.sql to display the employee numbers, last names, and salaries of all employees who earn more than the average salary and who work in a department with any employee with a u in their name. Resave lab9\_3.sql to lab9\_7.sql. Run the statement in lab9\_7.sql.**

EMPLOYEE_ID	LAST_NAME	SALARY
103	Hunold	9000

Ans7.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY
FROM EMPLOYEES
WHERE SALARY > (SELECT AVG(SALARY)
                  FROM EMPLOYEES)
AND DEPARTMENT_ID IN (SELECT DEPARTMENT_ID
                      FROM EMPLOYEES
                      WHERE LAST_NAME LIKE '%u%');
```

### Verification table -

The screenshot shows the Oracle SQL\*Plus interface. The command window displays the SQL query used to find employees earning above average salary and working in a department containing an employee with a 'u' in their last name. The results window shows a single row of data: Employee ID 103, Last Name Hunold, and Salary 9000.

EMPLOYEE_ID	LAST_NAME	SALARY
103	Hunold	9000

## ASSIGNMENT – 10

### ON: USING SET OPERATORS

Q1. List the department IDs for departments that do not contain the job ID ST\_CLERK, using set operators.

DEPARTMENT_ID
10
20
60
80
90
110
190

7 rows selected.

Ans1.

```
SELECT DEPARTMENT_ID  
FROM DEPARTMENTS  
MINUS  
SELECT DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE JOB_ID = 'ST_CLERK';
```

#### Verification table -

The screenshot shows the Oracle iSQL\*Plus Work Screen interface. The top menu bar includes 'File or URL' (with 'Choose File' and 'No file chosen'), 'Load Script', and icons for 'Logout', 'New Session', and 'Help'. Below the menu is a toolbar with 'Execute', 'Save Script', 'Clear Screen', and 'Cancel' buttons. The main area has a 'Enter statement:' text input field containing the SQL query from the previous text block. Below the input field is a preview pane titled 'DEPARTMENT\_ID' which displays the same data as the table above (10, 20, 60, 80, 90, 110, 190). At the bottom left of the preview pane, it says '7 rows selected.'

**Q2. Display the country ID and the name of the countries that have no departments located in them, using set operators.**

CO	COUNTRY_NAME
DE	Germany

Ans2.

```
SELECT COUNTRY_ID, COUNTRY_NAME
```

```
FROM COUNTRIES
```

```
MINUS
```

```
SELECT l.COUNTRY_ID, c.COUNTRY_NAME
```

```
FROM LOCATIONS l, COUNTRIES c
```

```
WHERE l.COUNTRY_ID = c.COUNTRY_ID;
```

#### Verification table -

The screenshot shows the Oracle SQL\*Plus interface. The command window displays the SQL query used to find countries with no departments. The results window shows a single row where CO (Country ID) is DE and COUNTRY\_NAME is Germany.

CO	COUNTRY_NAME
DE	Germany

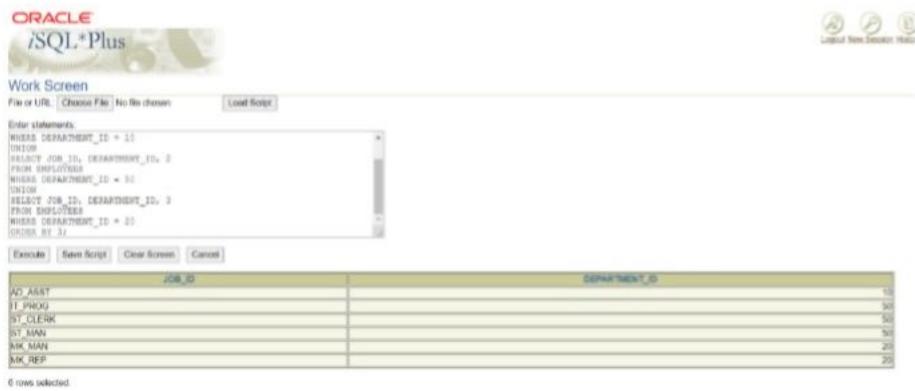
**Q3. Produce a list of jobs for departments 10, 50, and 20, in that order. Display job ID and department ID, using set operators.**

JOB_ID	DEPARTMENT_ID
AD_ASST	10
ST_CLERK	50
ST_MAN	50
MK_MAN	20
MK_REP	20

Ans3.

```
COLUMN dummy NOPRINT  
  
SELECT JOB_ID, DEPARTMENT_ID, 1 dummy  
  
FROM EMPLOYEES  
  
WHERE DEPARTMENT_ID = 10  
  
UNION  
  
SELECT JOB_ID, DEPARTMENT_ID, 2  
  
FROM EMPLOYEES  
  
WHERE DEPARTMENT_ID = 50  
  
UNION  
  
SELECT JOB_ID, DEPARTMENT_ID, 3  
  
FROM EMPLOYEES  
  
WHERE DEPARTMENT_ID = 20  
  
ORDER BY 3;
```

#### Verification table -



The screenshot shows the Oracle SQL\*Plus Work Screen interface. The top menu bar includes 'File or URL', 'Choose File' (No file chosen), and 'Load Script'. Below the menu is a toolbar with icons for Help, Logout, New Session, Help, and a magnifying glass. The main area has tabs for 'Work Screen' and 'Script'. A status bar at the bottom shows '6 rows selected.'

**Enter statements:**

```
WHERE DEPARTMENT_ID = 10
UNION
SELECT JOB_ID, DEPARTMENT_ID, 2
FROM EMPLOYEES
WHERE DEPARTMENT_ID = 50
UNION
SELECT JOB_ID, DEPARTMENT_ID, 3
FROM EMPLOYEES
WHERE DEPARTMENT_ID = 20
ORDER BY 3;
```

**Buttons:** Run, Save Script, Clear Screen, Cancel.

**Result Table:**

JOB_ID	DEPARTMENT_ID
AD_ASST	10
IT_PROG	50
ST_CLERK	50
ST_MAN	50
AK_MAN	20
AK_REP	20

Q4. List the employee IDs and job IDs of those employees who are currently in the job title that they have held once before during their tenure with the company.

EMPLOYEE_ID	JOB_ID
176	SA_REP

Ans4.

```
SELECT EMPLOYEE_ID, JOB_ID
FROM EMPLOYEES
INTERSECT
SELECT EMPLOYEE_ID, JOB_ID
FROM JOB_HISTORY;
```

#### Verification table -

The screenshot shows the Oracle SQL\*Plus interface. The command window contains the following SQL code:

```
SELECT EMPLOYEE_ID, JOB_ID
FROM EMPLOYEES
INTERSECT
SELECT EMPLOYEE_ID, JOB_ID
FROM JOB_HISTORY;
```

The results window displays the following data:

EMPLOYEE_ID	JOB_ID
170SA REP	2009AD ASST

Q5. Write a compound query that lists the following:

- List names and department ID of all the employees from the EMPLOYEES table, irrespective of the fact whether they belong to any department or not.
- Department ID and department name of all the departments from the DEPARTMENTS table, irrespective of the fact whether they have employees working in them or not.

LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
Abel	80	
Davies	50	
De Haan	90	
Ernst	60	
Fay	20	
Gietz	110	

Grant		
Hartstein	20	
Higgins	110	
Hunold	60	
King	90	
Kochhar	90	
Lorentz	50	
Matos	50	
Mourgos	50	
Rajs	50	
Taylor	80	
Vargas	50	
Whalen	10	
Zlotkey	80	
	10	Administration
	20	Marketing
	50	Shipping
	60	IT
	80	Sales
	90	Executive
	110	Accounting
	190	Contracting

28 rows selected.

Ans5.

```
SELECT LAST_NAME, DEPARTMENT_ID, TO_CHAR(NULL)
FROM EMPLOYEES
UNION
SELECT TO_CHAR(NULL), DEPARTMENT_ID, DEPARTMENT_NAME
FROM DEPARTMENTS;
```

Verification table -

LAST_NAME	DEPARTMENT_ID	HO_CHARTWELL
Adele	50	
Davies	50	
De Marin	50	
Eden	50	
Fay	20	
Gallo	110	
Keast	1	
Macmillan	20	
Magnus	100	
Spedding	50	
King	50	
Archibald	50	
Brennan	50	
Evans	50	
Hoyle	50	
Alvarez	50	
Hicks	100	
Taylor	100	
Spicer	100	
Watson	100	
Whelan	100	
Zilkley	50	
	100	HO Administration
	100	HO Marketing
	100	HO Accounting
	100	HO Sales
	100	HO Services
	100	HO Contracting

29 rows selected.

## ASSIGNMENT – 11

### ON: CREATING VIEWS

Q1. Create a view called EMPLOYEES\_VU based on the employee numbers, employee names, and department numbers from the EMPLOYEES table. Change the heading for the employee name to EMPLOYEE.

Ans1.

```
CREATE OR REPLACE VIEW EMPLOYEES_VU AS
```

```
SELECT EMPLOYEE_ID, LAST_NAME EMPLOYEE, DEPARTMENT_ID  
FROM EMPLOYEES;
```

Q2. Display the contents of the EMPLOYEES\_VU view.

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	50
206	Gietz	110

20 rows selected.

Ans2.

```
SELECT * FROM EMPLOYEES_VU;
```

#### Verification table-

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	50
145	Vergara	50
146	Sanchez	50
147	Rodriguez	50
148	Dharmalingham	50
149	Martinez	50
150	Fuller	50
151	Garrett	50
152	Nguyen	50
153	Montgomery	50
154	Farrell	50
155	Patel	50
156	Chen	50
157	Wong	50
158	Allen	50
159	充沛	50
160	Wong	50
161	Zheng	50
162	Ho	50
163	Yip	50
164	Tsai	50
165	Smith	50
166	Bradley	50
167	Price	50
168	Colmenares	50
169	Garrett	50
170	Nguyen	50
171	Montgomery	50
172	Farrell	50
173	Patel	50
174	Chen	50
175	Wong	50
176	Allen	50
177	充沛	50
178	Wong	50
179	Zheng	50
180	Ho	50
181	Yip	50
182	Tsai	50
183	Smith	50
184	Bradley	50
185	Colmenares	50
186	Garrett	50
187	Nguyen	50
188	Montgomery	50
189	Farrell	50
190	Chen	50
191	Wong	50
192	Allen	50
193	充沛	50
194	Wong	50
195	Zheng	50
196	Ho	50
197	Yip	50
198	Tsai	50
199	Smith	50
200	Bradley	50
201	Colmenares	50
202	Garrett	50
203	Nguyen	50
204	Montgomery	50
205	Farrell	50
206	Chen	50
207	Wong	50
208	Allen	50
209	充沛	50
210	Wong	50
211	Zheng	50
212	Ho	50
213	Yip	50
214	Tsai	50
215	Smith	50
216	Bradley	50
217	Colmenares	50
218	Garrett	50
219	Nguyen	50
220	Montgomery	50
221	Farrell	50
222	Chen	50
223	Wong	50
224	Allen	50
225	充沛	50
226	Wong	50
227	Zheng	50
228	Ho	50
229	Yip	50
230	Tsai	50
231	Smith	50
232	Bradley	50
233	Colmenares	50
234	Garrett	50
235	Nguyen	50
236	Montgomery	50
237	Farrell	50
238	Chen	50
239	Wong	50
240	Allen	50
241	充沛	50
242	Wong	50
243	Zheng	50
244	Ho	50
245	Yip	50
246	Tsai	50
247	Smith	50
248	Bradley	50
249	Colmenares	50
250	Garrett	50
251	Nguyen	50
252	Montgomery	50
253	Farrell	50
254	Chen	50
255	Wong	50
256	Allen	50
257	充沛	50
258	Wong	50
259	Zheng	50
260	Ho	50
261	Yip	50
262	Tsai	50
263	Smith	50
264	Bradley	50
265	Colmenares	50
266	Garrett	50
267	Nguyen	50
268	Montgomery	50
269	Farrell	50
270	Chen	50
271	Wong	50
272	Allen	50
273	充沛	50
274	Wong	50
275	Zheng	50
276	Ho	50
277	Yip	50
278	Tsai	50
279	Smith	50
280	Bradley	50
281	Colmenares	50
282	Garrett	50
283	Nguyen	50
284	Montgomery	50
285	Farrell	50
286	Chen	50
287	Wong	50
288	Allen	50
289	充沛	50
290	Wong	50
291	Zheng	50
292	Ho	50
293	Yip	50
294	Tsai	50
295	Smith	50
296	Bradley	50
297	Colmenares	50
298	Garrett	50
299	Nguyen	50
300	Montgomery	50
301	Farrell	50
302	Chen	50
303	Wong	50
304	Allen	50
305	充沛	50
306	Wong	50
307	Zheng	50
308	Ho	50
309	Yip	50
310	Tsai	50
311	Smith	50
312	Bradley	50
313	Colmenares	50
314	Garrett	50
315	Nguyen	50
316	Montgomery	50
317	Farrell	50
318	Chen	50
319	Wong	50
320	Allen	50
321	充沛	50
322	Wong	50
323	Zheng	50
324	Ho	50
325	Yip	50
326	Tsai	50
327	Smith	50
328	Bradley	50
329	Colmenares	50
330	Garrett	50
331	Nguyen	50
332	Montgomery	50
333	Farrell	50
334	Chen	50
335	Wong	50
336	Allen	50
337	充沛	50
338	Wong	50
339	Zheng	50
340	Ho	50
341	Yip	50
342	Tsai	50
343	Smith	50
344	Bradley	50
345	Colmenares	50
346	Garrett	50
347	Nguyen	50
348	Montgomery	50
349	Farrell	50
350	Chen	50
351	Wong	50
352	Allen	50
353	充沛	50
354	Wong	50
355	Zheng	50
356	Ho	50
357	Yip	50
358	Tsai	50
359	Smith	50
360	Bradley	50
361	Colmenares	50
362	Garrett	50
363	Nguyen	50
364	Montgomery	50
365	Farrell	50
366	Chen	50
367	Wong	50
368	Allen	50
369	充沛	50
370	Wong	50
371	Zheng	50
372	Ho	50
373	Yip	50
374	Tsai	50
375	Smith	50
376	Bradley	50
377	Colmenares	50
378	Garrett	50
379	Nguyen	50
380	Montgomery	50
381	Farrell	50
382	Chen	50
383	Wong	50
384	Allen	50
385	充沛	50
386	Wong	50
387	Zheng	50
388	Ho	50
389	Yip	50
390	Tsai	50
391	Smith	50
392	Bradley	50
393	Colmenares	50
394	Garrett	50
395	Nguyen	50
396	Montgomery	50
397	Farrell	50
398	Chen	50
399	Wong	50
400	Allen	50
401	充沛	50
402	Wong	50
403	Zheng	50
404	Ho	50
405	Yip	50
406	Tsai	50
407	Smith	50
408	Bradley	50
409	Colmenares	50
410	Garrett	50
411	Nguyen	50
412	Montgomery	50
413	Farrell	50
414	Chen	50
415	Wong	50
416	Allen	50
417	充沛	50
418	Wong	50
419	Zheng	50
420	Ho	50
421	Yip	50
422	Tsai	50
423	Smith	50
424	Bradley	50
425	Colmenares	50
426	Garrett	50
427	Nguyen	50
428	Montgomery	50
429	Farrell	50
430	Chen	50
431	Wong	50
432	Allen	50
433	充沛	50
434	Wong	50
435	Zheng	50
436	Ho	50
437	Yip	50
438	Tsai	50
439	Smith	50
440	Bradley	50
441	Colmenares	50
442	Garrett	50
443	Nguyen	50
444	Montgomery	50
445	Farrell	50
446	Chen	50
447	Wong	50
448	Allen	50
449	充沛	50
450	Wong	50
451	Zheng	50
452	Ho	50
453	Yip	50
454	Tsai	50
455	Smith	50
456	Bradley	50
457	Colmenares	50
458	Garrett	50
459	Nguyen	50
460	Montgomery	50
461	Farrell	50
462	Chen	50
463	Wong	50
464	Allen	50
465	充沛	50
466	Wong	50
467	Zheng	50
468	Ho	50
469	Yip	50
470	Tsai	50
471	Smith	50
472	Bradley	50
473	Colmenares	50
474	Garrett	50
475	Nguyen	50
476	Montgomery	50
477	Farrell	50
478	Chen	50
479	Wong	50
480	Allen	50
481	充沛	50
482	Wong	50
483	Zheng	50
484	Ho	50
485	Yip	50
486	Tsai	50
487	Smith	50
488	Bradley	50
489	Colmenares	50
490	Garrett	50
491	Nguyen	50
492	Montgomery	50
493	Farrell	50
494	Chen	50
495	Wong	50
496	Allen	50
497	充沛	50
498	Wong	50
499	Zheng	50
500	Ho	50
501	Yip	50
502	Tsai	50
503	Smith	50
504	Bradley	50
505	Colmenares	50
506	Garrett	50
507	Nguyen	50
508	Montgomery	50
509	Farrell	50
510	Chen	50
511	Wong	50
512	Allen	50
513	充沛	50
514	Wong	50
515	Zheng	50
516	Ho	50
517	Yip	50
518	Tsai	50
519	Smith	50
520	Bradley	50
521	Colmenares	50
522	Garrett	50
523	Nguyen	50
524	Montgomery	50
525	Farrell	50
526	Chen	50
527	Wong	50
528	Allen	50
529	充沛	50
530	Wong	50
531	Zheng</	

**Q3. Select the view name and text from the USER\_VIEWS data dictionary view.**

**Note:** Another view already exists. The EMP\_DETAILS\_VIEW was created as part of your schema.

**Note:** To see more contents of a long column, use the iSQL\*Plus command SET LONG n, where n is the value of the number of characters of the LONG column that you want to see.

VIEW_NAME	TEXT
EMPLOYEES_VU	SELECT employee_id, last_name employee, department_id FROM employees
EMP_DETAILS_VIEW	SELECT e.employee_id, e.job_id, e.manager_id , e.department_id, d.location_id, l.country_id, e.first_name, e.last_name, e.salary, e.commission_pct, d.department_name, j.job_title, l.city, l.state_province, c.country_name, r.region_name FROM employees e, departments d , jobs j, locations l, countries c, regions r WHERE e.department_id = d.department_id AND d.loaction_id = l.location_id AND l.country_id = c. country_id AND c.region_id = r.region_id AND j.job_id = e.job_id WITH READ ONLY

Ans3.

SET LONG 1000

SELECT VIEW\_NAME, TEXT

FROM USER\_VIEWS;

**Verification table-**

Work Screen

File or URL: Choose File: No file chosen

Load Script

Enter statements:  
SET LONG 1000  
SELECT VIEW\_NAME, TEXT  
FROM USER\_VIEWS;

Execute Save Script Clear Screen Cancel

VIEW_NAME	TEXT
DEPT_SUM_VU	select d.department_name, MIN(e.salary), MAX(e.salary), AVG(e.salary) from employees e, departments d where e.department_id=d.department_id group by d.department_name
EMPLOYEES_VU	SELECT employee_id, last_name, employee_name, department_id FROM EMPLOYEES
EMPVU10	Select employee_id, last_name, job_id from employees where department_id=10 with read only
SALVU50	Select employee_id, number, last_name, name, salary*12 ANN_SALARY from employees where department_id=50

**Q4. Using your EMPLOYEES\_VU view, enter a query to display all employee names and department numbers.**

EMPLOYEE	DEPARTMENT_ID
King	90
Kochhar	90
De Haan	90
Gietz	110

20 rows selected.

Ans4.

SELECT EMPLOYEE, DEPARTMENT\_ID

FROM EMPLOYEES\_VU;

**Verification table-**

EMPLOYEE	DEPARTMENT_ID
King	90
Kochhar	90
De Haan	90
Hundt	90
Ernest	90
Janet	90
Mourgos	90
Nap	90
Davies	90
Mats	90
Vergara	90
Zlotkey	90
Abed	90
Taylor	90
EMPLOYEE	DEPARTMENT_ID
Grant	90
Whalen	90
Kurtzman	90
Ernest	90
Higgins	110
Gietz	110
Taylor	110

21 rows selected.

**Q5. Create a view named DEPT50 that contains the employee numbers, employee last names, and department numbers for all employees in department 50. Label the**

view columns EMPNO, EMPLOYEE, and DEPTNO. Don not follow an employee to be reassigned to another department through the view.

Ans5.

```
CREATE OR REPLACE VIEW DEPT50 AS
```

```
SELECT EMPLOYEE_ID EMPNO, LAST_NAME EMPLOYEE, DEPARTMENT_ID DEPTNO  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID=50  
WITH CHECK OPTION CONSTRAINT EMP_DEPT_50;
```

Q6. Display the structure and contents of the DEPT50 view.

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(6)
EMPLOYEE	NOT NULL	VARCHAR2(25)
DEPTNO		NUMBER(4)

EMPNO	EMPLOYEE	DEPTNO
124	Mourgos	50
141	Rajs	50
142	Davies	50
143	Matos	50
144	Vargas	50

Ans6.

```
DESCRIBE DEPT50;
```

```
SELECT * FROM DEPT50;
```

Verification table-

ORACLE SQL\*Plus

Work Screen

File or URL: Choose File No file chosen Load Script

Enter statements:

```
DESCRIBE DEPT50;
SELECT * FROM DEPT50;
```

Name	Null	Type
DEPTNO	NOT NULL	NUMBER(2)
EMPLOYEE		VARCHAR2(14)
EXPNNO		VARCHAR2(4)

EMPNO	EMPLOYEE	DEPTNO
101	Janice	
102	George	
103	Elise	
104	Eduardo	
105	Matos	
106	Megan	
107	Taylor	

7 rows selected.

**Q7. Attempt to reassign Matos to department 80.**

Ans7.

UPDATE DEPT50

SET DEPTNO=80

WHERE EMPLOYEE = 'MATOS';

### Verification table-

ORACLE iSQL\*Plus

Work Screen

File or URL: Choose File No file chosen Load Script

Enter statements:

```
UPDATE DEPT50
SET DEPTNO=80
WHERE EMPLOYEE = 'MATOS';
```

0 rows updated.

If you have time, complete the following exercise:

**Q8. Create a view called SALARY\_VU based on the employee last names, department names, salaries, and salary grades for all employees. Use the EMPLOYEES, DEPARTMENTS, and JOB\_GRADES tables. Label the columns Employee, Department, Salary, and Grade, respectively.**

Ans8.

CREATE OR REPLACE VIEW SALARY\_VU AS

```

SELECT e.LAST_NAME "Employee", d.DEPARTMENT_NAME "Department",
       e.SALARY "Salary", j.GRADE_LEVEL "Grades"
FROM EMPLOYEES e, DEPARTMENTS d, JOB_GRADES j
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID
AND e.SALARY BETWEEN j.LOWEST_SAL AND j.HIGHEST_SAL;

```

### Verification table-

Employee	Department	Salary	Grade
James	Shipping	2000A	
Jonathon	Shipping	2000A	
Karren	Shipping	2500A	
Mark	IT	4200B	
Larry	Shipping	3000B	
Donna	Shipping	5000B	
Albert	Shipping	5000B	
Winston	Administration	4500B	
Taylor	Shipping	4000B	
Lundt	IT	6000C	
Sally	R&D	6000C	
Eddy	Marketing	6000C	
David	Accounting	10000C	
ynthia	R&D	10000D	
Employee	Department	Salary	Grade
Abe	Sales	11000D	
Hutton	Marketing	12000D	
Lynn	Accounting	12000D	
Tom	Executive	24000F	
Kothari	Executive	17000E	
Le Hall	Executive	17000E	

20 rows selected

---