

Creating and Managing Tables

EX_NO:1

DATE:

- 1.Create the DEPT table based on the DEPARTMENT following the table instance chart below. Confirm that the table is created.

Column name	ID	NAME
Key Type		
Nulls/Unique		
FK table		
FK column		
Data Type	Number	Varchar2
Length	7	25

QUERY:

Create table dept(id int,name varchar2(25));

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands tab, the following SQL code is entered:

```
1 create table department(
2   id int,
3   name varchar(25)
4 );
5
```

The results tab shows the output of the command:

Table dropped.
0.06 seconds

2.Create the EMP table based on the following instance chart. Confirm that the table is created.

Column name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
Key Type				
Nulls/Unique				
FK table				
FK column				
Data Type	Number	Varchar2	Varchar2	Number
Length	7	25	25	7

QUERY:

```
Create table emp(id number(7),Last_Name varchar2(25),First_Name varchar2(25),Dept_id number(7));
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'create table emp(id number(7),last_name varchar2(25),first_name varchar2(25),dept_id number(7));'. Below the code, the 'Results' tab is active, displaying the message 'Table created.' and a execution time of '0.04 seconds'.

```
create table emp(id number(7),last_name varchar2(25),first_name varchar2(25),dept_id number(7));
```

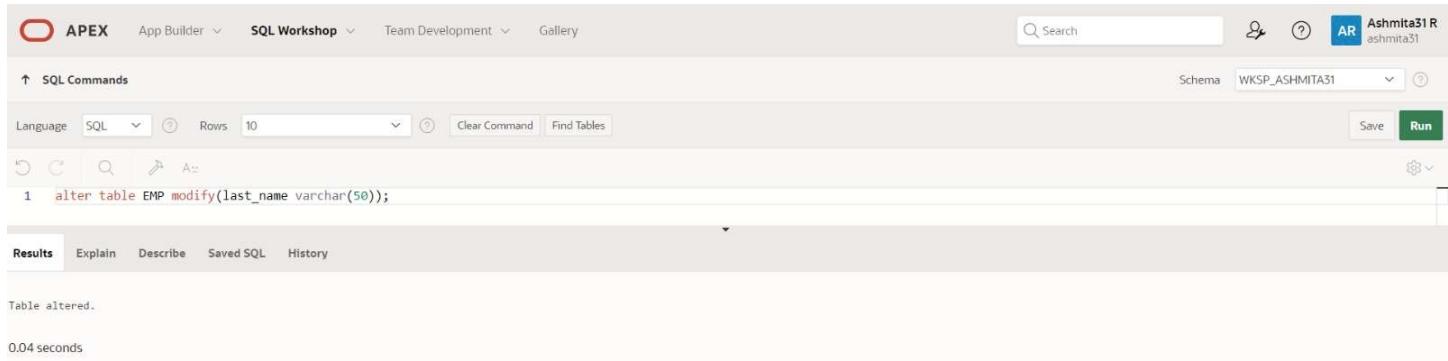
3.Modify the EMP table to allow for longer employee last names. Confirm the modification.(Hint: Increase the

size to 50)

QUERY:

```
Alter table emp modify(Last_Name varchar2(50));
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands tab, the command `alter table emp modify(last_name varchar2(50));` is entered. The Results tab displays the output: `Table altered.` and `0.04 seconds`.

4.Create the EMPLOYEES2 table based on the structure of EMPLOYEES table. Include Only the Employee_id, First_name, Last_name, Salary and Dept_id coloumns. Name the columns Id, First_name, Last_name, salary and Dept_id respectively.

QUERY:

```
Create table employees2(id number(6),first_name varchar2(20),Last_name varchar2(25),Salary number(8,2),Dept_id number(6)not null);
```

OUTPUT:

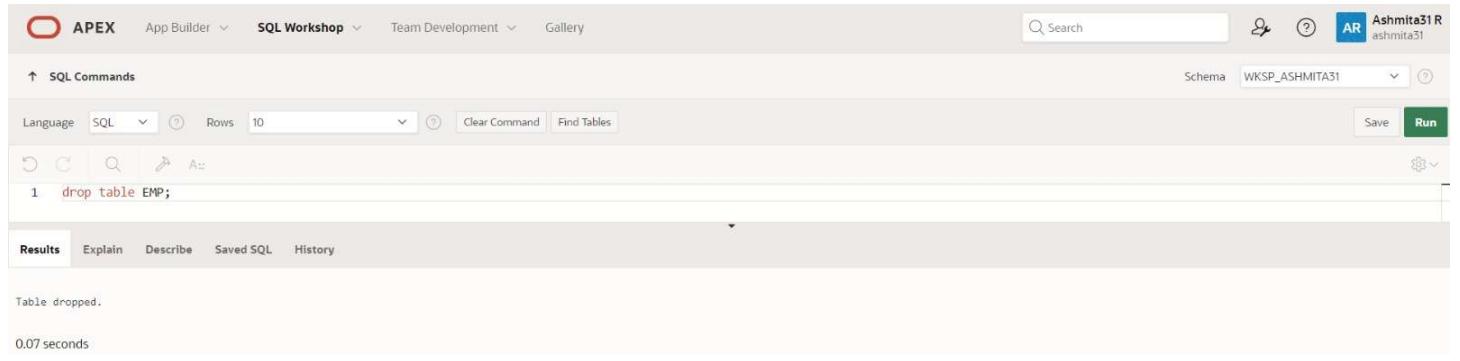


The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands tab, the command `create table employees2(id number(6)not null,first_name varchar(20),last_name varchar(25)not null,salary number(8,2),dept_id number(6)not null);` is entered. The Results tab displays the output: `Table created.` and `0.04 seconds`.

5.Drop the EMP table.

QUERY:

Drop table emp;

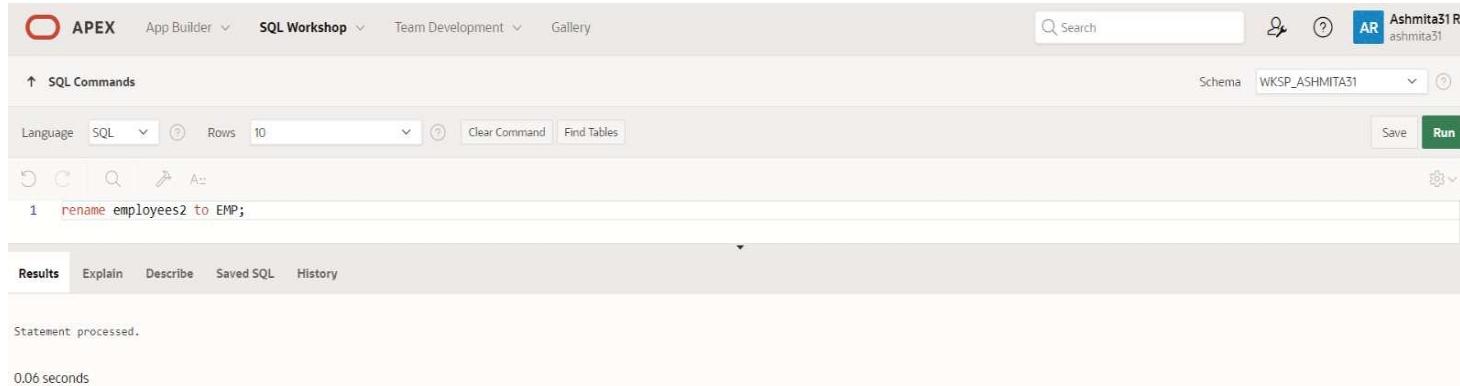
OUTPUT:

A screenshot of the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a schema dropdown set to 'WKSP_ASHMITA31'. The main workspace shows a SQL command line with the following text:
1 drop table EMP;
Below the command, the 'Results' tab is selected, showing the output:
Table dropped.
0.07 seconds

6.Rename the EMPLOYEES table as EMP.

QUERY:

Rename employees2 to emp;

OUTPUT:

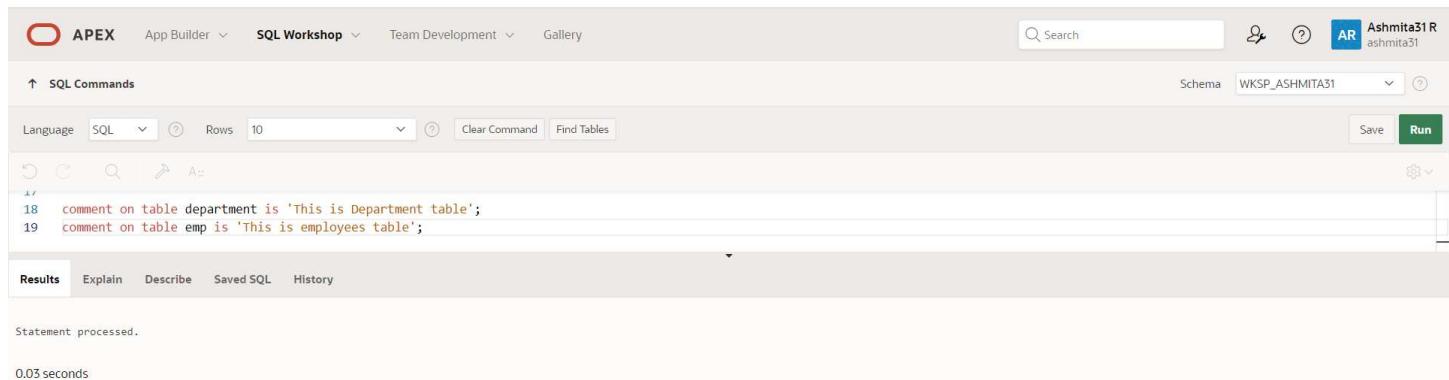
A screenshot of the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a schema dropdown set to 'WKSP_ASHMITA31'. The main workspace shows a SQL command line with the following text:
1 rename employees2 to EMP;
Below the command, the 'Results' tab is selected, showing the output:
Statement processed.
0.06 seconds

7.Add a comment on DEPT and EMP tables. Confirm the modification by describing the table.

QUERY:

comment on table dept is 'This is Department table';
comment on table emp is 'This is Employee table';

OUTPUT:



A screenshot of the Oracle SQL Workshop interface. The top navigation bar shows 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, user profile, and session information 'Ashmita31 R ashmita31'. The main area is titled 'SQL Commands'. It shows the following SQL code:

```
18  comment on table department is 'This is Department table';
19  comment on table emp is 'This is employees table';
```

The 'Results' tab is selected. The output shows:

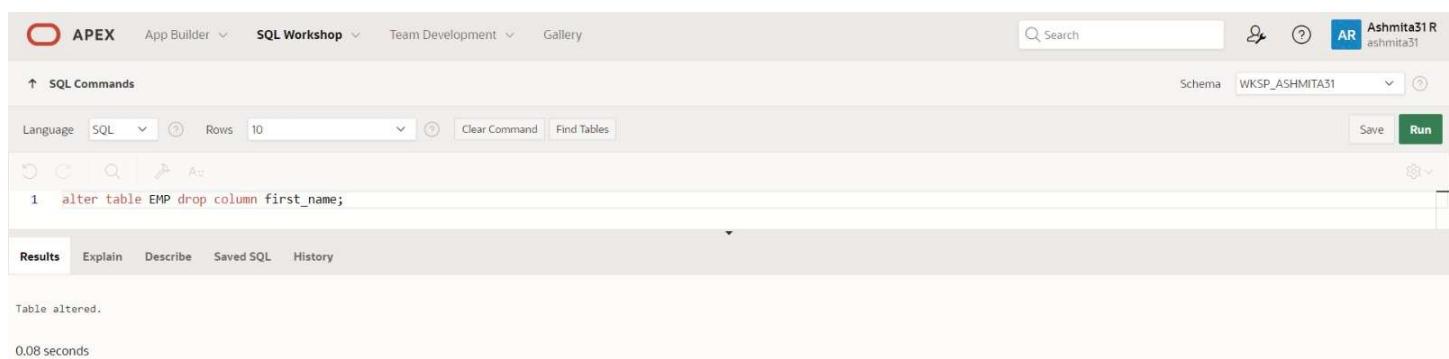
```
Statement processed.
0.03 seconds
```

8.Drop the First_name column from the EMP table and confirm it.

QUERY:

Alter table emp drop column first_name;

OUTPUT:



A screenshot of the Oracle SQL Workshop interface, similar to the previous one but with a different query. The top navigation bar and session information are the same. The main area is titled 'SQL Commands'. It shows the following SQL code:

```
1  alter table EMP drop column first_name;
```

The 'Results' tab is selected. The output shows:

```
Table altered.
0.08 seconds
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

MANIPULATING DATA

EX_NO:2

DATE:

1.Create MY_EMPLOYEE table with the following structure

NAME	NULL?	TYPE
ID	Not null	Number(4)
Last_name		Varchar(25)
First_name		Varchar(25)
Userid		Varchar(25)
Salary		Number(9,2)

QUERY:

```
create table my_employee(id number(4)not null,last_name varchar(25),first_name varchar(25),user_id  
varchar(25),salary number(9,2));
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. On the right side, there are search, schema selection (WKSP_ASHMITA31), and user profile (ASHMITA31 R ashmita31) buttons. The main workspace is titled "SQL Commands". It contains a toolbar with icons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run. Below the toolbar, a command line shows the SQL code for creating the table. The results tab at the bottom displays the message "Table created." and "0.03 seconds".

```
1 create table my_employee(id number(4)not null,last_name varchar(25),first_name varchar(25),user_id  
varchar(25),salary number(9,2));
```

Results Explain Describe Saved SQL History

Table created.
0.03 seconds

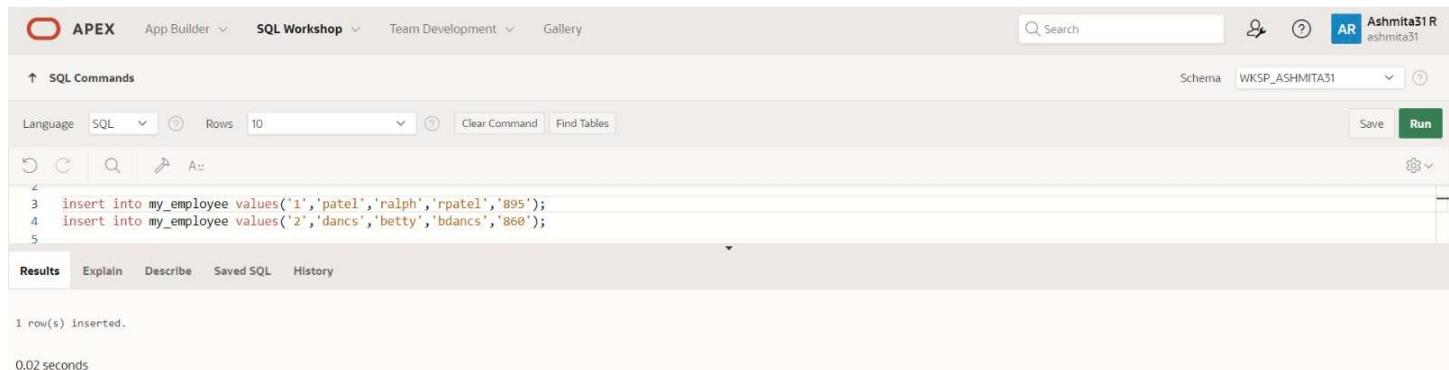
2.Add the first and second rows data to MY_EMPLOYEE table from the following sample data.

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	Ropebur	Audrey	aropebur	1550

QUERY:

```
insert into my_employee values('1','patel','ralph','rpatel','895');  
insert into my_employee values('2','dancs','betty','bdancs','860');
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'ashmita31'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), and various buttons like Clear Command, Find Tables, Save, and Run. The SQL editor contains the following code:

```
3: insert into my_employee values('1','patel','ralph','rpatel','895');  
4: insert into my_employee values('2','dancs','betty','bdancs','860');
```

The results tab shows the output of the executed commands:

```
1 row(s) inserted.  
0.02 seconds
```

3. Display the table with values.

QUERY:

```
select*from my_employee;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and the schema name WKSP_ASHMITA31. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the executed query: "6 select*from my_employee;". The Results tab displays a table with two rows of data from the my_employee table:

ID	LAST_NAME	FIRST_NAME	USER_ID	SALARY
2	dancs	betty	bdancs	860
1	patel	ralph	rpatel	895

Below the table, it says "2 rows returned in 0.02 seconds" and there's a "Download" link.

4. Populate the next two rows of data from the sample data. Concatenate the first letter of the first_name with the first seven characters of the last_name to produce Userid.

QUERY:

```
insert into my_employee values('3','biri','ben','bbiri','1100');  
insert into my_employee values('4','newmann','chad','cnewmann','750');
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and the schema name WKSP_ASHMITA31. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the executed insert statements:

```
8: insert into my_employee values('3','biri','ben','bbiri','1100');  
9: insert into my_employee values('4','newmann','chad','cnewmann','750');
```

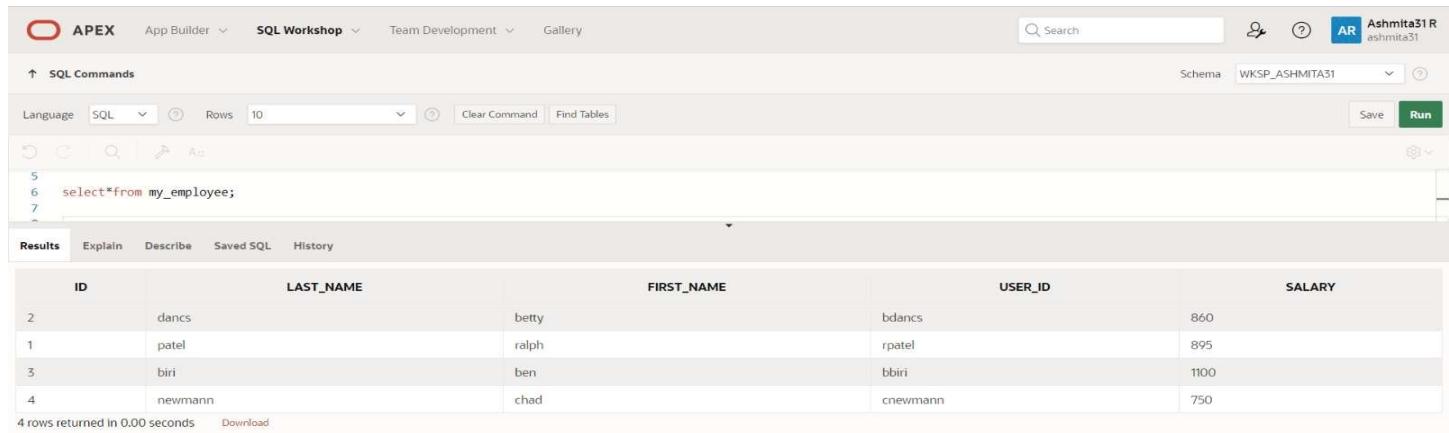
The Results tab shows the output: "1 row(s) inserted." and "0.00 seconds".

5. Make the data additions permanent.

QUERY:

```
select*from my_employee;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'Ashmita31 R ashmita31'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the executed query: 'select*from my_employee;'. The Results tab displays a table with four rows of employee data:

ID	LAST_NAME	FIRST_NAME	USER_ID	SALARY
2	dancs	betty	bdancs	860
1	patel	ralph	rpatel	895
3	biri	ben	bbiri	1100
4	newmann	chad	cnewmann	750

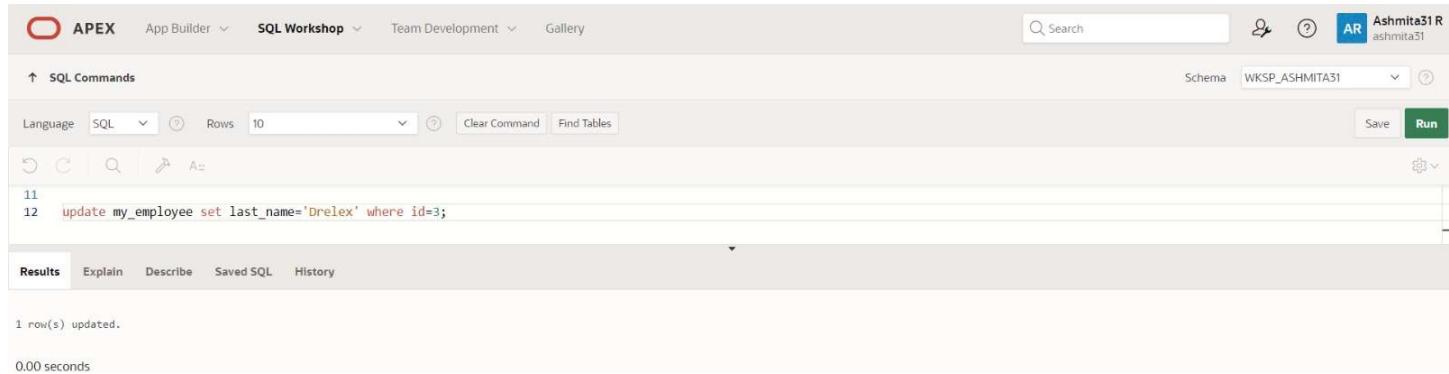
Below the table, it says '4 rows returned in 0.00 seconds' and has a 'Download' link.

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

QUERY:

```
update my_employee set last_name='Drelex' where id=3;
```

OUTPUT:



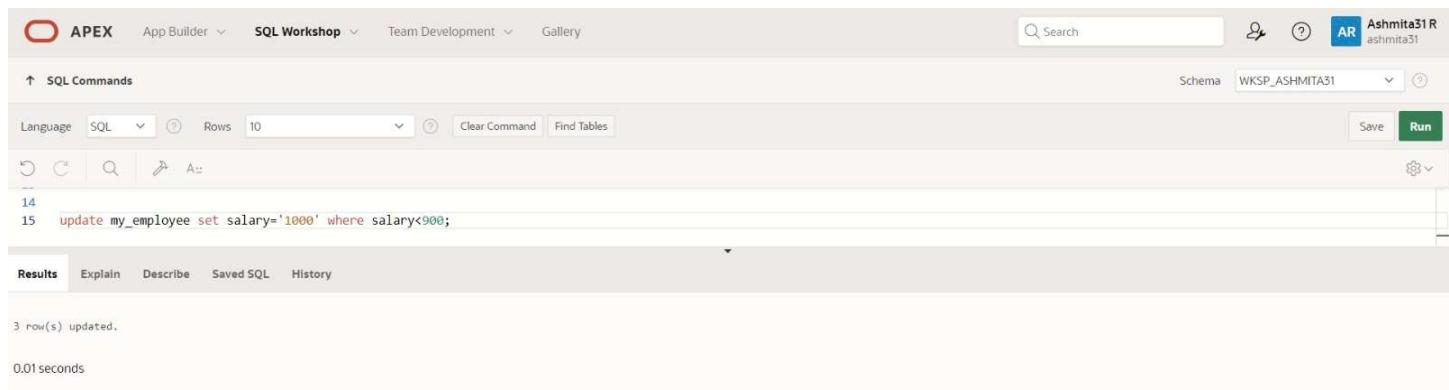
The screenshot shows the Oracle SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar and session information are the same. The SQL Commands tab shows the update query: 'update my_employee set last_name='Drelex' where id=3;'. The Results tab shows the message '1 row(s) updated.' and '0.00 seconds'.

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

QUERY:

```
update my_employee set salary='1000' where salary<900;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, help icon, and a session identifier (AR Ashmita31). The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the following command:

```
14  
15 update my_employee set salary='1000' where salary<900;
```

The Results tab displays the output:

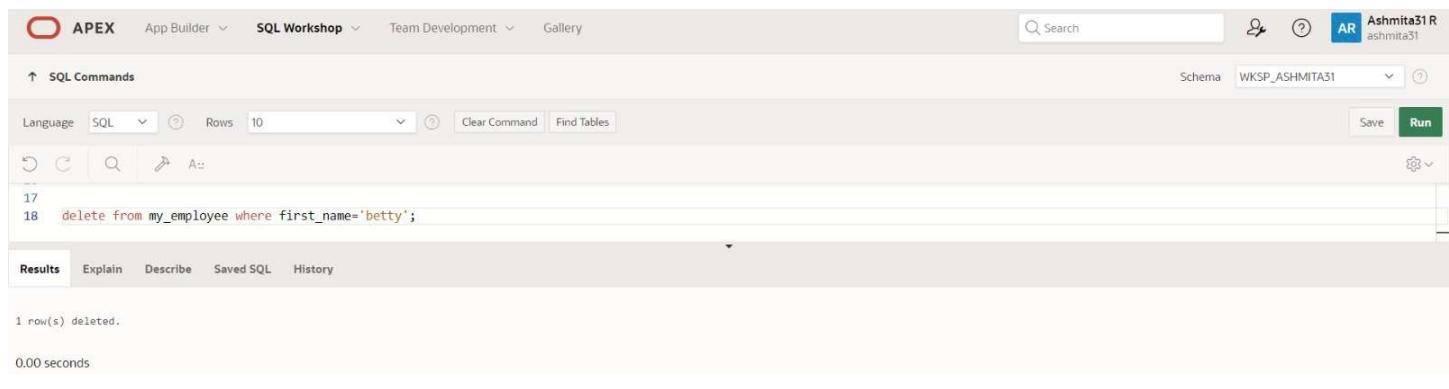
```
3 row(s) updated.  
0.01 seconds
```

8.Delete Betty from MY_EMPLOYEE table.

QUERY:

```
delete from my_employee where first_name='betty';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, help icon, and a session identifier (AR Ashmita31). The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the following command:

```
17  
18 delete from my_employee where first_name='betty';
```

The Results tab displays the output:

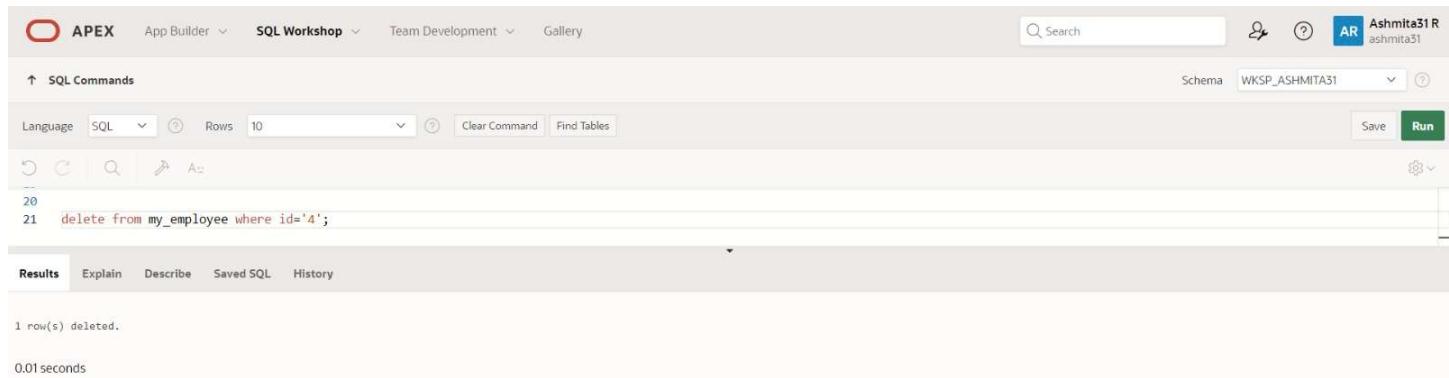
```
1 row(s) deleted.  
0.00 seconds
```

9.Empty the fourth row of the emp table.

QUERY:

```
delete from my_employee where id='4';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Ashmita31 R' and the schema 'WKSP_ASHMITA51'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The command entered is 'delete from my_employee where id='4';'. The results tab shows the output: '1 row(s) deleted.' and '0.01 seconds'. Other tabs available include Explain, Describe, Saved SQL, and History.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

INCLUDING CONSTRAINTS

EX_NO:3

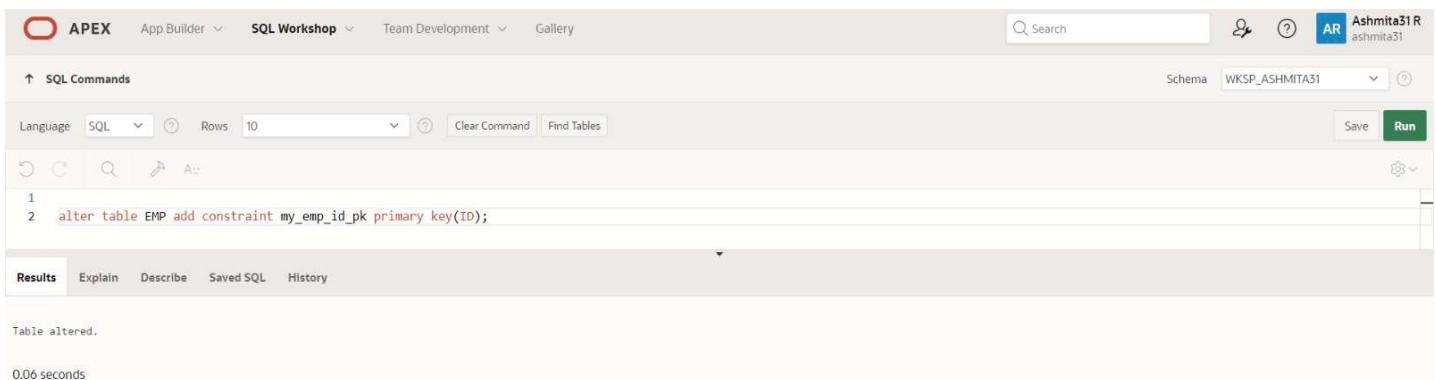
DATE:

- 1.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.

QUERY:

```
alter table EMP add constraint my_emp_id_pk primary key(ID);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information (Ashmita31 R), and a schema dropdown set to WKSP_ASHMITA31. The main workspace is titled "SQL Commands". It has tabs for Language (SQL selected), Rows (10), Clear Command, and Find Tables. Below these are standard toolbar icons for copy, paste, search, and refresh. The SQL editor contains the following code:

```
1 alter table EMP add constraint my_emp_id_pk primary key(ID);
```

The "Results" tab is active, showing the output: "Table altered." Below it, the time taken is listed as "0.06 seconds".

2.Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my_dept_id_pk.

QUERY:

```
alter table department add constraint my_dept_id_pk primary key(ID);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'AR Ashmita31 R ashmita31'. The main area is titled 'SQL Commands' with a 'Run' button. Below it, the schema is set to 'WKSP_ASHMITA31'. The code entered is:

```
3  
4  
5 alter table department add constraint my_dept_id_pk primary key(ID);
```

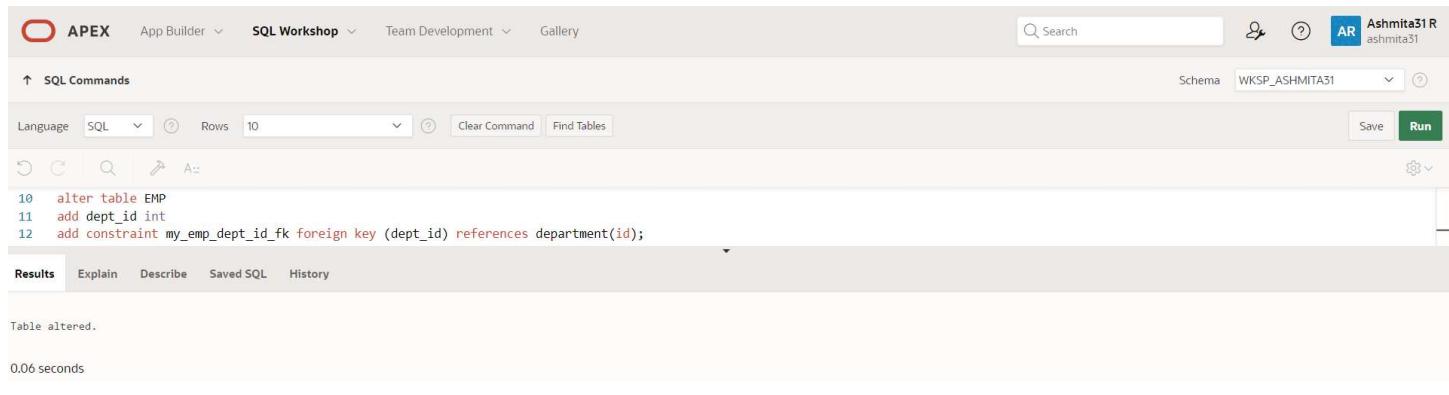
The results section shows the output: 'Table altered.' and a execution time of '0.08 seconds'.

3.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 nonexistent department. Name the constraint my_emp_dept_id_fk.

QUERY:

```
alter table emp add dept_id int  
add constraint my_emp_dept_id_fk foreign key (dept_id) references department(id);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'AR Ashmita31 R ashmita31'. The main area is titled 'SQL Commands' with a 'Run' button. Below it, the schema is set to 'WKSP_ASHMITA31'. The code entered is:

```
10 alter table EMP  
11 add dept_id int  
12 add constraint my_emp_dept_id_fk foreign key (dept_id) references department(id);
```

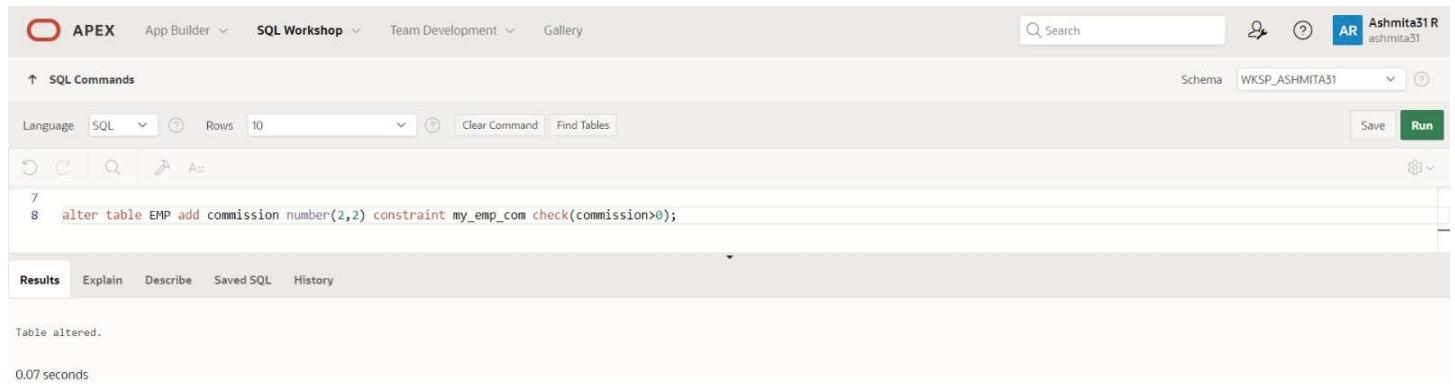
The results section shows the output: 'Table altered.' and a execution time of '0.06 seconds'.

4.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.

QUERY:

```
alter table EMP add commission number(2,2) constraint my_emp_com check(commission>0);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays the following SQL command:

```
7  
8 alter table EMP add commission number(2,2) constraint my_emp_com check(commission>0);
```

The 'Results' tab is active, showing the output:

```
Table altered.  
0.07 seconds
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

Writing Basic SQL SELECT Statements

EX_NO:4

DATE:

- 1.The following statement executes successfully.

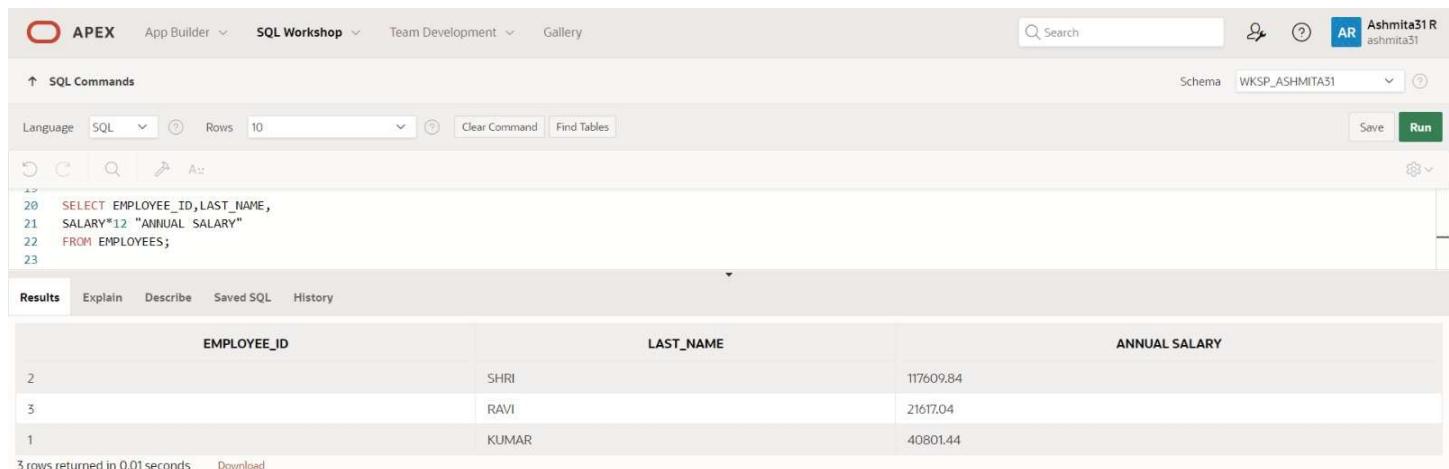
Identify the Errors

```
SELECT employee_id, last_name  
sal*12 ANNUAL SALARY  
FROM employees;
```

QUERY: False

```
SELECT EMPLOYEE_ID, LAST_NAME,  
SALARY*12 "ANNUAL SALARY"  
FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Ashmita31' (ashmita31). The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the following code:

```
20: SELECT EMPLOYEE_ID, LAST_NAME,  
21: SALARY*12 "ANNUAL SALARY"  
22: FROM EMPLOYEES;  
23:
```

The Results tab displays the output of the query:

EMPLOYEE_ID	LAST_NAME	ANNUAL SALARY
2	SHRI	117609.84
3	RAVI	21617.04
1	KUMAR	40801.44

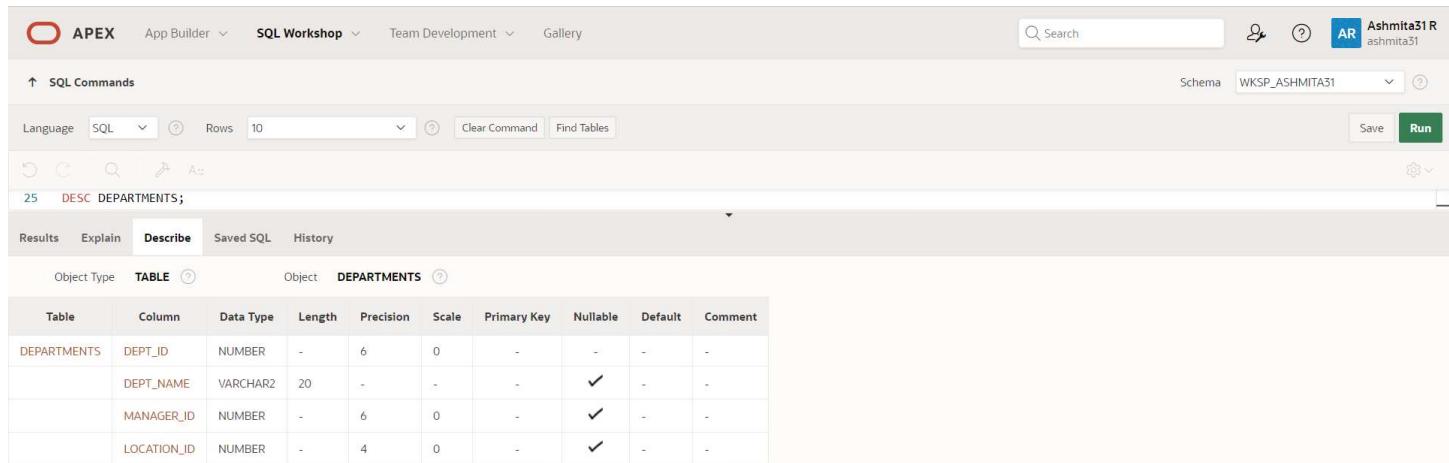
Below the table, it says '3 rows returned in 0.01 seconds' and there is a 'Download' link.

2. Show the structure of departments the table. Select all the data from it.

QUERY:

DESC DEPARTMENTS;

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Ashmita31 R ashmita31', and a schema dropdown set to 'WKSP_ASHMITA31'. The main area displays the results of the 'DESC' command run at line 25:

```
25 DESC DEPARTMENTS;
```

The results pane shows the structure of the 'DEPARTMENTS' table with four columns:

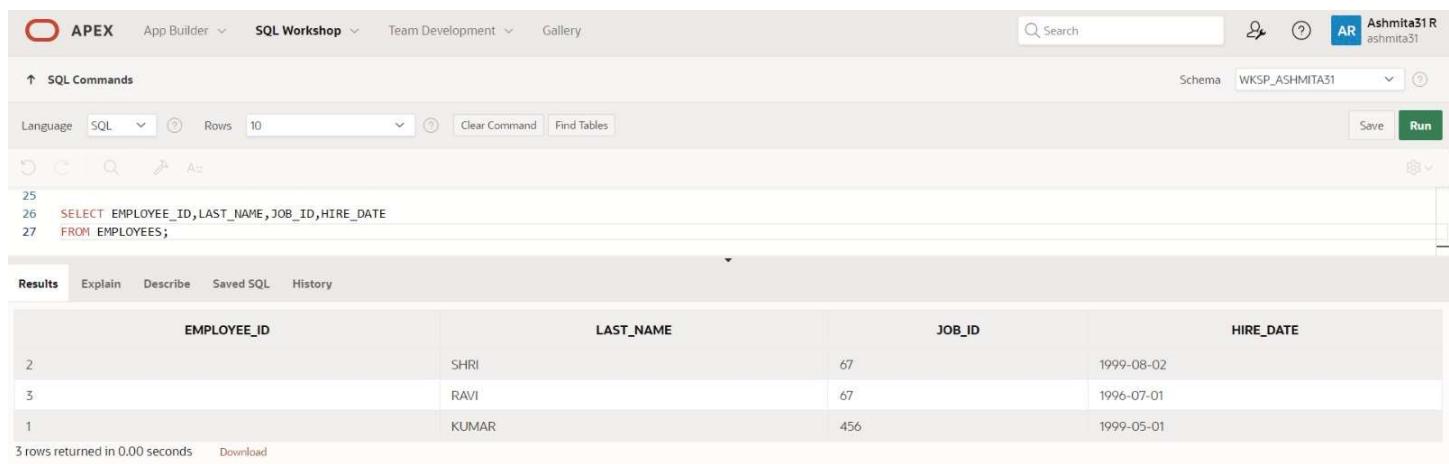
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENTS	DEPT_ID	NUMBER	-	6	0	-	-	-	-
	DEPT_NAME	VARCHAR2	20	-	-	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	LOCATION_ID	NUMBER	-	4	0	-	✓	-	-

3.Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first.

QUERY:

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, HIRE_DATE  
FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Ashmita31 R' and the schema 'WKSP_ASHMITA31'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

```
25  
26  SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, HIRE_DATE  
27  FROM EMPLOYEES;
```

The Results tab displays the output in a table:

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
2	SHRI	67	1999-08-02
3	RAVI	67	1996-07-01
1	KUMAR	456	1999-05-01

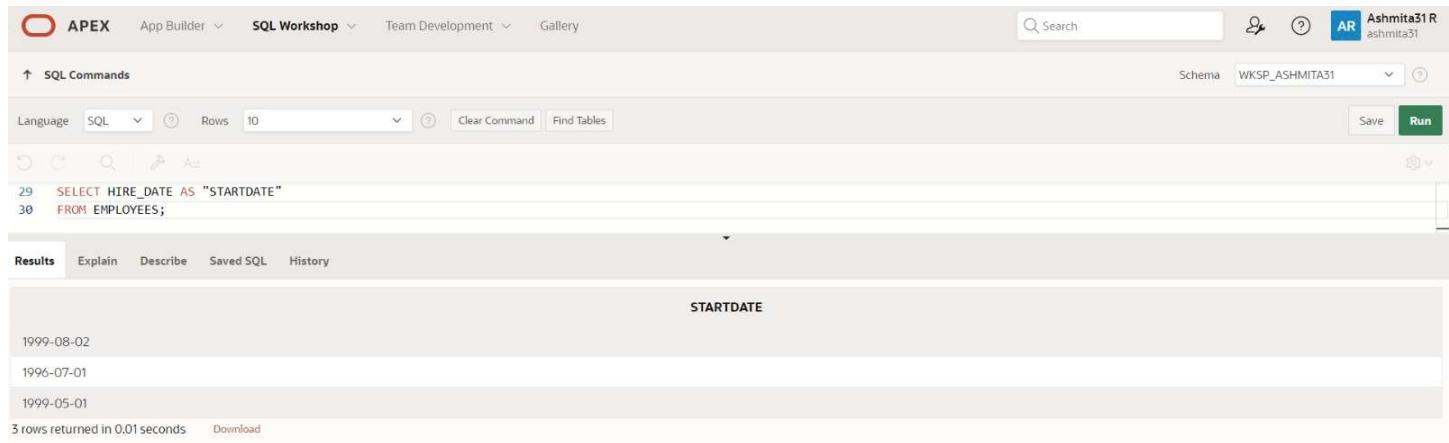
Below the table, it says '3 rows returned in 0.00 seconds' and has a 'Download' link.

4.Provide an alias STARTDATE for the hire date.

QUERY:

```
SELECT HIRE_DATE AS "STARTDATE"  
FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Ashmita31 R' and the schema 'WKSP_ASHMITA31'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

```
29  
30  SELECT HIRE_DATE AS "STARTDATE"  
   FROM EMPLOYEES;
```

The Results tab displays the output in a table:

STARTDATE
1999-08-02
1996-07-01
1999-05-01

Below the table, it says '3 rows returned in 0.01 seconds' and has a 'Download' link.

5.Create a query to display unique job codes from the employee table.

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Ashmita31R' (ashmita31). The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to 'WKSP_ASHMITA31'. Below the command input field, there are buttons for 'Save' and 'Run'. The command entered is:

```
32  SELECT DISTINCT JOB_ID  
33  FROM EMPLOYEES;
```

The results section shows the output:

JOB_ID
456
67

2 rows returned in 0.01 seconds. There is a 'Download' link at the bottom.

6.Display the last name concatenated with the job ID , separated by a comma and space, and name the column EMPLOYEE and TITLE.

QUERY:

```
SELECT LAST_NAME||','||'|JOB_ID as "EMPLOYEE and TITLE"  
from EMPLOYEES;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Ashmita31R' (ashmita31). The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to 'WKSP_ASHMITA31'. Below the command input field, there are buttons for 'Save' and 'Run'. The command entered is:

```
35  SELECT LAST_NAME||','||'|JOB_ID as "EMPLOYEE and TITLE"  
36  from EMPLOYEES;
```

The results section shows the output:

EMPLOYEE and TITLE
SHRI, 67
RAVI, 67
KUMAR, 456

3 rows returned in 0.00 seconds. There is a 'Download' link at the bottom.

7.Create a query to display all the data from the employees table. Separate each column by a comma. Name the column THE_OUTPUT.

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for Ashmita31 R. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the query code. The Results tab shows the output in a table format with a single column labeled 'THE_OUTPUT'. The output contains three rows of employee data.

THE_OUTPUT
2,ANU,SHRI,423@gmail.com,789,1999-08-02,67,9800.82,.34,.22,982
3,KAVITHA,RAVI,788@gmail.com,899,1996-07-01,67,1801.42,.56,.64,582
1,PRIYA,KUMAR,125@gmail.com,1237,1999-05-01,456,3400.12,.12,.56,452

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

RESTRICTING AND SORTING DATA

EX.NO.5

DATE:

Find the Solution for the following:

- 1.Create a query to display the last name and salary of employees earning more than 12000.

QUERY:

```
SELECT LAST_NAME,SALARY  
FROM EMPLOYEES  
WHERE SALARY>12000;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Ashmita31R', and a schema dropdown set to 'WKSP_ASHMITA31'. The main area has tabs for SQL Commands and Results. Under SQL Commands, the query is pasted:

```
20  SELECT LAST_NAME,SALARY  
21  FROM EMPLOYEES  
22  WHERE SALARY>12000;  
23
```

The Results tab is selected, displaying the output:

LAST_NAME	SALARY
JAY	15400.12
JANE	23400.12

2 rows returned in 0.03 seconds. There are also 'Download' and 'History' buttons at the bottom of the results table.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The command entered is:

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

The results table displays the following data:

LAST_NAME	DEPARTMENT_ID
EMANUEL	30

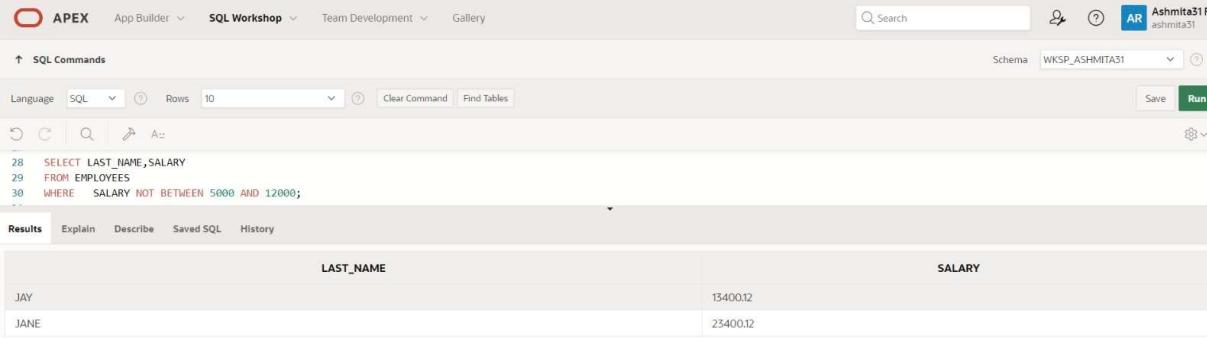
1 rows returned in 0.02 seconds

3. Create a query to display the last name and salary of employees whose salary is not in the range of 5000 and 12000. (hints: not between)

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The command entered is:

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

The results table displays the following data:

LAST_NAME	SALARY
JAY	13400.12
JANE	23400.12

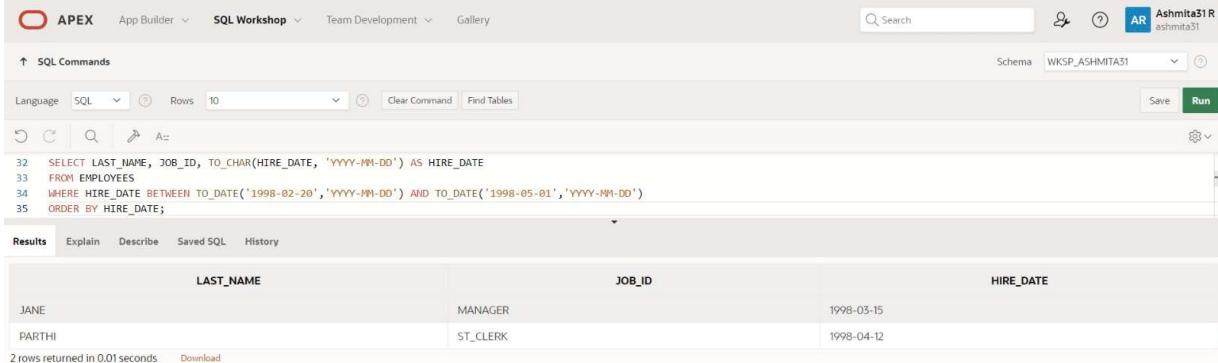
2 rows returned in 0.04 seconds

4. 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.(hints: between)

QUERY:

```
SELECT LAST_NAME, JOB_ID, TO_CHAR(HIRE_DATE, 'YYYY-MM-DD') AS HIRE_DATE  
FROM EMPLOYEES  
  
WHERE HIRE_DATE BETWEEN TO_DATE('1998-02-20','YYYY-MM-DD') AND  
TO_DATE('1998-05-01','YYYY-MM-DD')  
  
ORDER BY HIRE_DATE;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information (Ashmita31 R, ashmita31), and a schema dropdown set to WKSP_ASHMITA31. Below the toolbar, the SQL command window displays the query code. The results tab is selected, showing the output of the query:

LAST_NAME	JOB_ID	HIRE_DATE
JANE	MANAGER	1998-03-15
PARTHI	ST_CLERK	1998-04-12

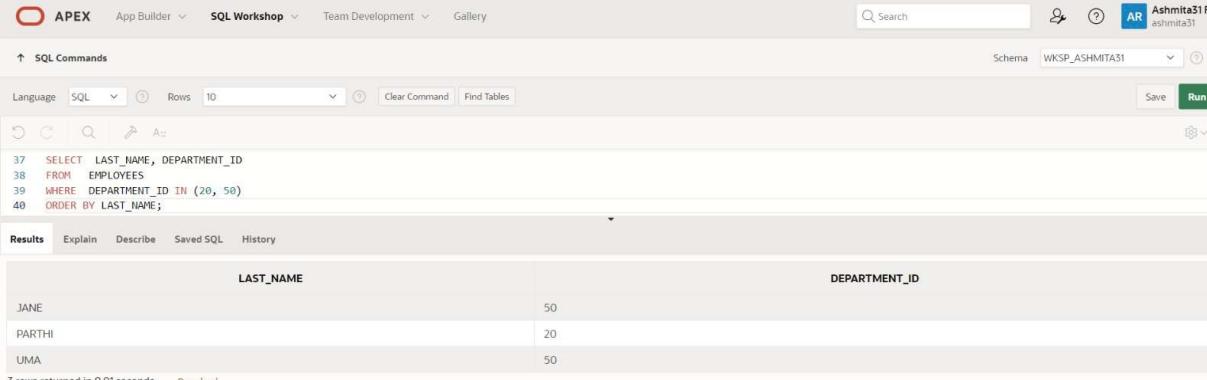
2 rows returned in 0.01 seconds [Download](#)

5. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.(hints: in, orderby)

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user's profile: Ashmita31 R and ashmita31. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the following code:

```
37  SELECT LAST_NAME, DEPARTMENT_ID  
38  FROM EMPLOYEES  
39  WHERE DEPARTMENT_ID IN (20, 50)  
40  ORDER BY LAST_NAME;
```

The Results tab shows the output of the query:

LAST_NAME	DEPARTMENT_ID
JANE	50
PARTHI	20
UMA	50

Below the table, it says "3 rows returned in 0.01 seconds" and there is a "Download" link.

6. Display the last name and salary of all employees who earn between 5000 and 12000 and are in departments 20 and 50 in alphabetical order by name. Label the columns EMPLOYEE, MONTHLY SALARY respectively.(hints: between, in)

QUERY:

```
SELECT LAST_NAME AS "EMPLOYEE", department_id AS "MONTHLY_SALARY"  
FROM EMPLOYEES  
WHERE SALARY BETWEEN 5000 AND 12000 AND DEPARTMENT_ID IN (20, 50)  
ORDER BY LAST_NAME;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The SQL command window contains the following code:

```
42: SELECT LAST_NAME AS "EMPLOYEE", department_id AS "MONTHLY_SALARY"  
43: FROM EMPLOYEES  
44: WHERE SALARY BETWEEN 5000 AND 12000 AND DEPARTMENT_ID IN (20, 50)  
45: ORDER BY LAST_NAME;
```

The results window displays the output:

EMPLOYEE	MONTHLY_SALARY
PARTHI	20
UMA	50

2 rows returned in 0.01 seconds

7. Display the last name and hire date of every employee who was hired in 1994.(hints: like)

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The SQL command window contains the following code:

```
48: SELECT LAST_NAME,HIRE_DATE  
49: FROM EMPLOYEES  
50: WHERE HIRE_DATE LIKE '%94';
```

The results window displays the output:

LAST_NAME	HIRE_DATE
EMANUEL	05/01/1994

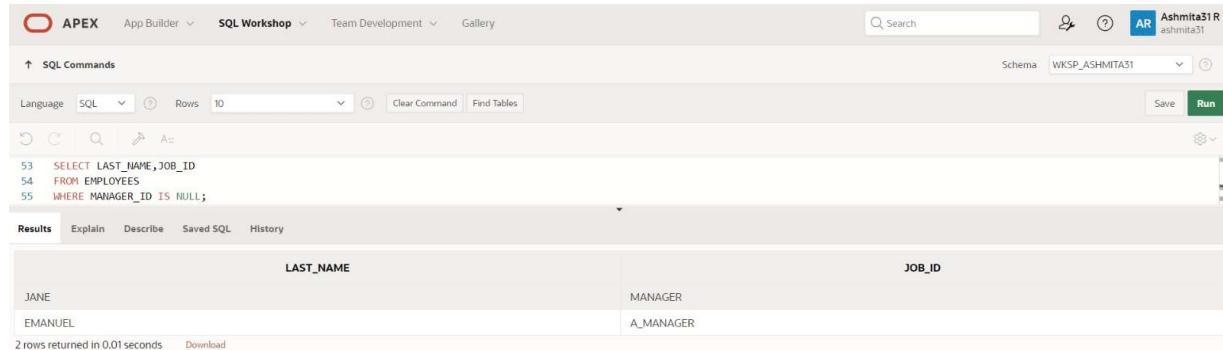
1 rows returned in 0.01 seconds

8. Display the last name and job title of all employees who do not have a manager.(hints: is null)

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab contains the following query:

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

The Results tab displays the output:

LAST_NAME	JOB_ID
JANE	MANAGER
EMANUEL	A_MANAGER

2 rows returned in 0.01 seconds

9. Display the last name, salary, and commission for all employees who earn commissions.

Sort data in descending order of salary and commissions.(hints: is not null,orderby)

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab contains the following query:

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

The Results tab displays the output:

LAST_NAME	SALARY	COMMISSION_PCT
JANE	23400.12	.28
JAY	13400.12	.1
EMANUEL	12000	.2

3 rows returned in 0.01 seconds

10. Display the last name of all employees where the third letter of the name is a.(hints:like)

QUERY:

```
SELECT LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME LIKE '__A%'
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query in the command window is:

```
63: SELECT LAST_NAME  
64: FROM EMPLOYEES  
65: WHERE LAST_NAME LIKE '__A%'
```

The results window displays the following data:

LAST_NAME
UMA
EMANUEL

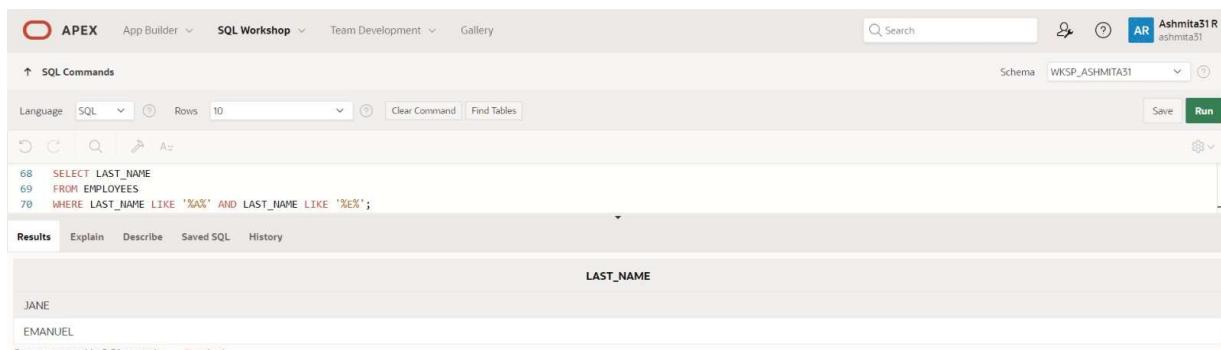
2 rows returned in 0.00 seconds

11. Display the last name of all employees who have an a and an e in their last name.(hints: like)

QUERY:

```
SELECT LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME LIKE '%A%' AND LAST_NAME LIKE '%E%';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query in the command window is:

```
68: SELECT LAST_NAME  
69: FROM EMPLOYEES  
70: WHERE LAST_NAME LIKE '%A%' AND LAST_NAME LIKE '%E%';
```

The results window displays the following data:

LAST_NAME
JANE
EMANUEL

2 rows returned in 0.01 seconds

12. Display the last name and job and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to 2500 ,3500 or 7000.(hints:in,not in)

QUERY:

```
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);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query in the command window is:

```
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);
```

The results window displays the following data:

LAST_NAME	JOB_ID	SALARY
PARTHI	ST_CLERK	10650.82

1 rows returned in 0.01 seconds

13. Display the last name, salary, and commission for all employees whose commission amount is 20%.(hints:use predicate logic)

QUERY:

```
SELECT LAST_NAME,SALARY,COMMISSION_PCT  
FROM EMPLOYEES  
WHERE COMMISSION_PCT = .20;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query in the command window is:

```
SELECT LAST_NAME,SALARY,COMMISSION_PCT  
FROM EMPLOYEES  
WHERE COMMISSION_PCT = .20;
```

The results window displays the following data:

LAST_NAME	SALARY	COMMISSION_PCT
EMANUEL	12000	.2

1 rows returned in 0.01 seconds

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

SINGLE ROW FUNCTIONS

EX-NO : 6

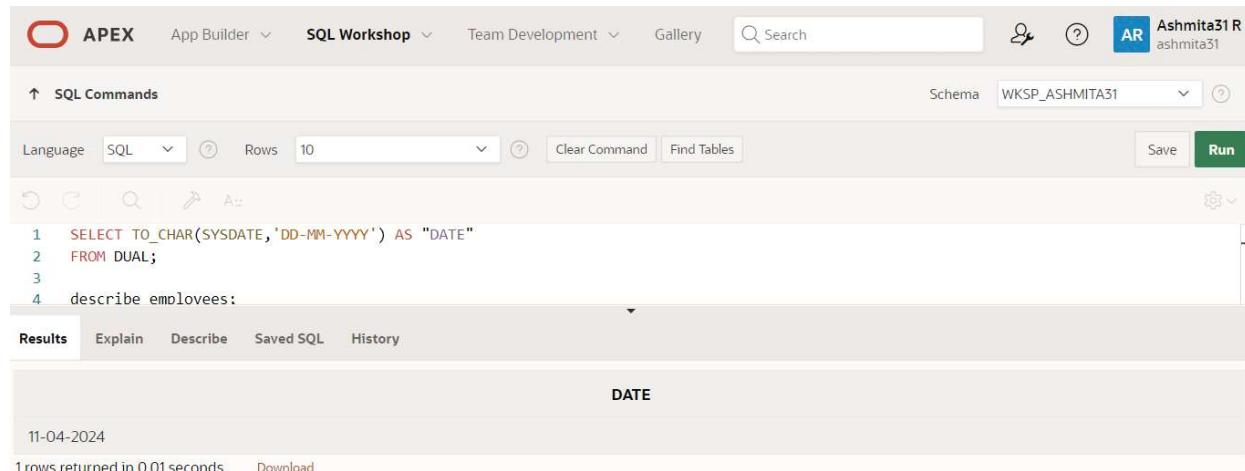
DATE:

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

QUERY:

```
SELECT TO_CHAR(SYSDATE,'YYYY-MM-DD') AS "DATE" FROM DUAL;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and a user profile for Ashmita31 R. The main workspace is titled 'SQL Commands'. The language is set to SQL, and the results are displayed over 10 rows. The query entered is:

```
1 SELECT TO_CHAR(SYSDATE,'DD-MM-YYYY') AS "DATE"
2 FROM DUAL;
3
4 describe employees;
```

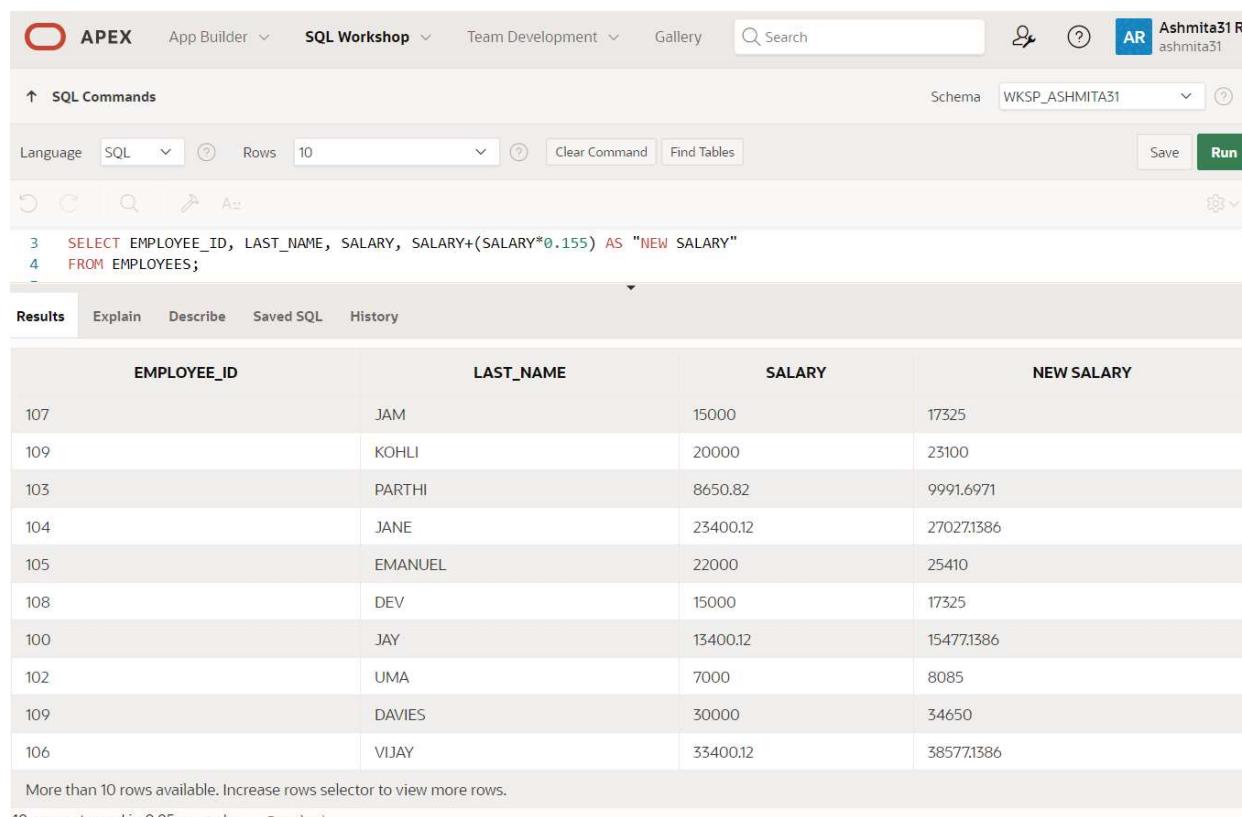
The results section shows a single row with the date '11-04-2024' under the column 'DATE'. Below the results, it says '1 rows returned in 0.01 seconds'.

2. The HR department needs a report to display the employee number, last name, salary, and increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary.

QUERY:

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY, SALARY+(SALARY*0.155) AS  
"NEW SALARY"  
FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, Gallery, and a search bar. The user is signed in as Ashmita31 R (ashmita31). The main area is titled "SQL Commands". The query entered is:

```
3  SELECT EMPLOYEE_ID, LAST_NAME, SALARY, SALARY+(SALARY*0.155) AS "NEW SALARY"  
4  FROM EMPLOYEES;
```

The results section displays the following data:

EMPLOYEE_ID	LAST_NAME	SALARY	NEW SALARY
107	JAM	15000	17325
109	KOHLI	20000	23100
103	PARTHI	8650.82	9991.6971
104	JANE	23400.12	27027.1386
105	EMANUEL	22000	25410
108	DEV	15000	17325
100	JAY	13400.12	15477.1386
102	UMA	7000	8085
109	DAVIES	30000	34650
106	VIJAY	33400.12	38577.1386

Text at the bottom of the results pane: "More than 10 rows available. Increase rows selector to view more rows."

At the bottom left: "10 rows returned in 0.05 seconds" and "Download".

3. Modify your query lab_03_02.sql to add a column that subtracts the old salary from the new salary. Label the column Increase.

QUERY:

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY, SALARY+(SALARY*15.5/100) AS "NEW SALARY", (SALARY+(SALARY*15.5/100))-SALARY AS "INCREASE"  
FROM EMPLOYEES;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, Gallery, a search bar, and user information for Ashmita31 R. The SQL Workshop tab is selected. Below the toolbar, there are buttons for Language (SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The main area displays the SQL query:

```
6  SELECT EMPLOYEE_ID, LAST_NAME, SALARY, SALARY+(SALARY*15.5/100) AS "NEW SALARY", (SALARY+(SALARY*15.5/100))-SALARY AS "INCREASE"  
7  FROM EMPLOYEES;
```

The Results tab is selected, showing the output of the query:

EMPLOYEE_ID	LAST_NAME	SALARY	NEW SALARY	INCREASE
107	JAM	15000	17325	2325
109	KOHLI	20000	23100	3100
103	PARTHI	8650.82	9991.6971	1340.8771
104	JANE	23400.12	27027.1386	3627.0186
105	EMANUEL	22000	25410	3410
108	DEV	15000	17325	2325
100	JAY	13400.12	15477.1386	2077.0186
102	UMA	7000	8085	1085
109	DAVIES	30000	34650	4650
106	VIJAY	33400.12	38577.1386	5177.0186

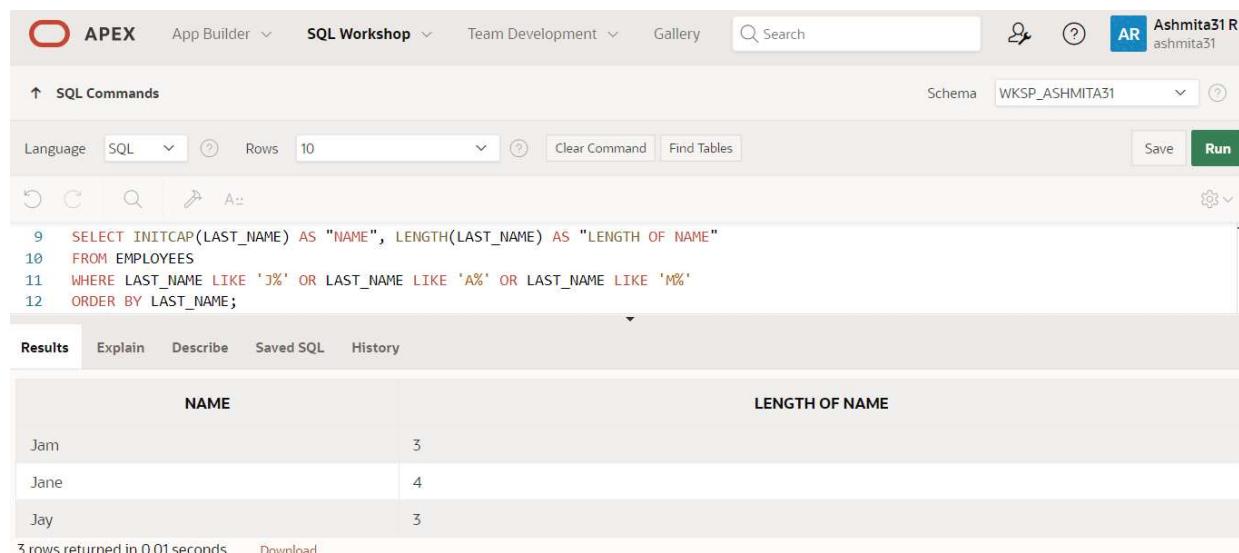
Below the table, a message states "More than 10 rows available. Increase rows selector to view more rows." At the bottom left, it says "10 rows returned in 0.02 seconds". There is also a "Download" link.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The right side shows a user profile for 'Ashmita31 R' (ashmita31). The main workspace is titled 'SQL Commands'. It has tabs for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. Below these are icons for Undo, Redo, Search, Insert, and Paste. The SQL command area contains the query from the question. The results tab is selected, showing a table with two columns: 'NAME' and 'LENGTH OF NAME'. The data returned is:

NAME	LENGTH OF NAME
Jam	3
Jane	4
Jay	3

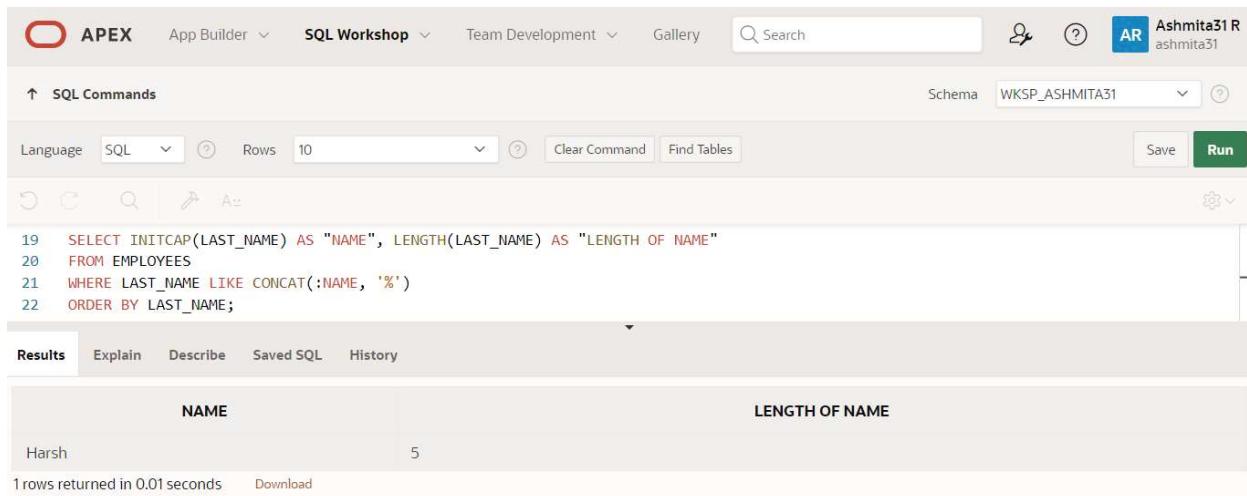
At the bottom left, it says '3 rows returned in 0.01 seconds'. There is also a 'Download' link.

5. Rewrite the query so that the user is prompted to enter a letter that starts the last name. For example, if the user enters H when prompted for a letter, then the output should show all employees whose last name starts with the letter H.

QUERY:

```
SELECT INITCAP(LAST_NAME) AS "NAME", LENGTH(LAST_NAME) AS  
"LENGTH OF NAME"  
FROM EMPLOYEES  
WHERE LAST_NAME LIKE CONCAT(:NAME, '%')  
ORDER BY LAST_NAME;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a user profile for 'Ashmita31 R' are also present. The main workspace displays the SQL command entered by the user:

```
19  SELECT INITCAP(LAST_NAME) AS "NAME", LENGTH(LAST_NAME) AS "LENGTH OF NAME"  
20  FROM EMPLOYEES  
21  WHERE LAST_NAME LIKE CONCAT(:NAME, '%')  
22  ORDER BY LAST_NAME;
```

Below the command, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing the output of the query:

NAME	LENGTH OF NAME
Harsh	5

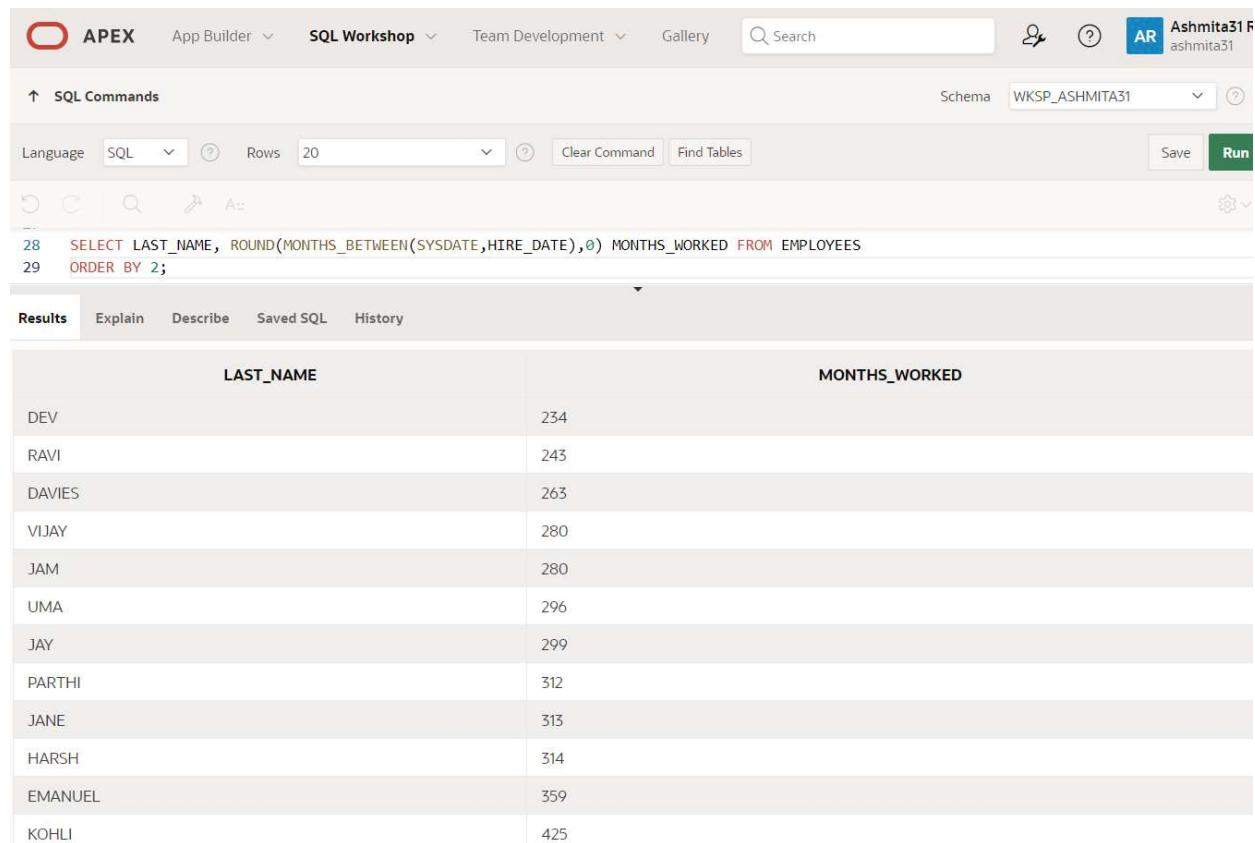
At the bottom left, it says '1 rows returned in 0.01 seconds'. There is also a 'Download' link at the bottom right.

6. The HR department wants to find the length of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which 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.

QUERY:

```
SELECT LAST_NAME, ROUND(MONTHS_BETWEEN(SYSDATE,HIRE_DATE),0)
MONTHS_WORKED FROM EMPLOYEES
ORDER BY 2;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information (Ashmita31). Below the toolbar, the schema is set to WKSP_ASHMITA31. The main area displays the SQL command and its execution results.

SQL Commands

```
28  SELECT LAST_NAME, ROUND(MONTHS_BETWEEN(SYSDATE,HIRE_DATE),0) MONTHS_WORKED FROM EMPLOYEES
29  ORDER BY 2;
```

Results

LAST_NAME	MONTHS_WORKED
DEV	234
RAVI	243
DAVIES	263
VIJAY	280
JAM	280
UMA	296
JAY	299
PARTHI	312
JANE	313
HARSH	314
EMANUEL	359
KOHLI	425

7. Create a report that produces the following for each employee: earns monthly but wants . Label the column Dream Salaries.

QUERY:

```
SELECT LAST_NAME||' EARNS $'||SALARY||' MONTHLY BUT WANTS  
$'||SALARY*3 "DREAM SALARY"  
FROM EMPLOYEES;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', 'Gallery', a search bar, and a user profile 'Ashmita31 R ashmita31'. Below the toolbar, the schema is set to 'WKSP_ASHMITA31'. The main area is titled 'SQL Commands' and contains the following SQL code:

```
31  SELECT LAST_NAME||' EARNS $'||SALARY||' MONTHLY BUT WANTS $'||SALARY*3 "DREAM SALARY"  
32  FROM EMPLOYEES;
```

The 'Results' tab is selected, displaying the output of the query:

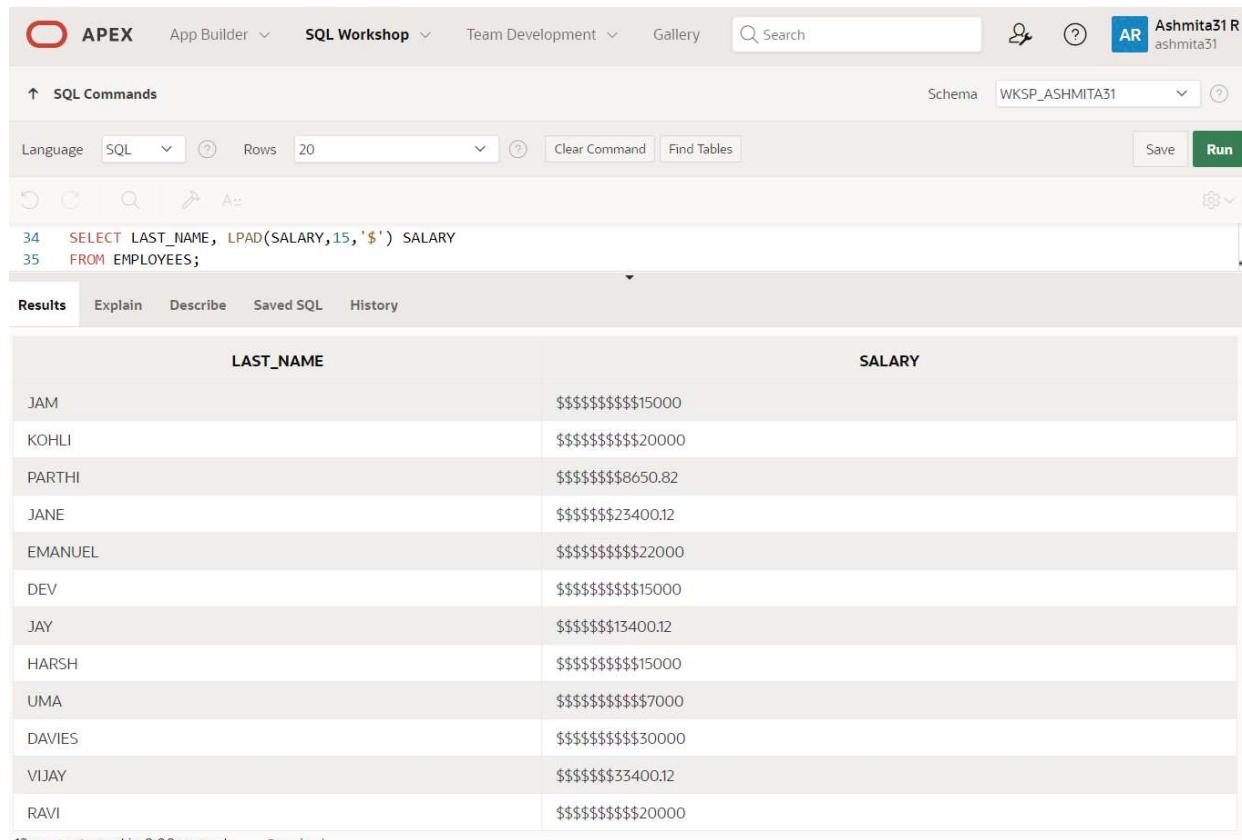
DREAM SALARY
JAM EARNS \$15000 MONTHLY BUT WANTS \$45000
KOHLI EARNS \$20000 MONTHLY BUT WANTS \$60000
PARTHI EARNS \$8650.82 MONTHLY BUT WANTS \$25952.46
JANE EARNS \$23400.12 MONTHLY BUT WANTS \$70200.36
EMANUEL EARNS \$22000 MONTHLY BUT WANTS \$66000
DEV EARNS \$15000 MONTHLY BUT WANTS \$45000
JAY EARNS \$13400.12 MONTHLY BUT WANTS \$40200.36
HARSH EARNS \$15000 MONTHLY BUT WANTS \$45000
UMA EARNS \$7000 MONTHLY BUT WANTS \$21000
DAVIES EARNS \$30000 MONTHLY BUT WANTS \$90000
VIJAY EARNS \$33400.12 MONTHLY BUT WANTS \$100200.36
RAVI EARNS \$20000 MONTHLY BUT WANTS \$60000

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information for Ashmita31. The main area is titled "SQL Commands". The query entered is:

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

The results tab is selected, displaying the output:

LAST_NAME	SALARY
JAM	\$\$\$\$\$\$\$\$\$\$15000
KOHLI	\$\$\$\$\$\$\$\$\$\$20000
PARTHI	\$\$\$\$\$\$\$\$\$8650.82
JANE	\$\$\$\$\$\$\$\$\$23400.12
EMANUEL	\$\$\$\$\$\$\$\$\$\$22000
DEV	\$\$\$\$\$\$\$\$\$\$15000
JAY	\$\$\$\$\$\$\$\$\$13400.12
HARSH	\$\$\$\$\$\$\$\$\$\$15000
UMA	\$\$\$\$\$\$\$\$\$\$7000
DAVIES	\$\$\$\$\$\$\$\$\$\$30000
VIJAY	\$\$\$\$\$\$\$\$\$33400.12
RAVI	\$\$\$\$\$\$\$\$\$\$20000

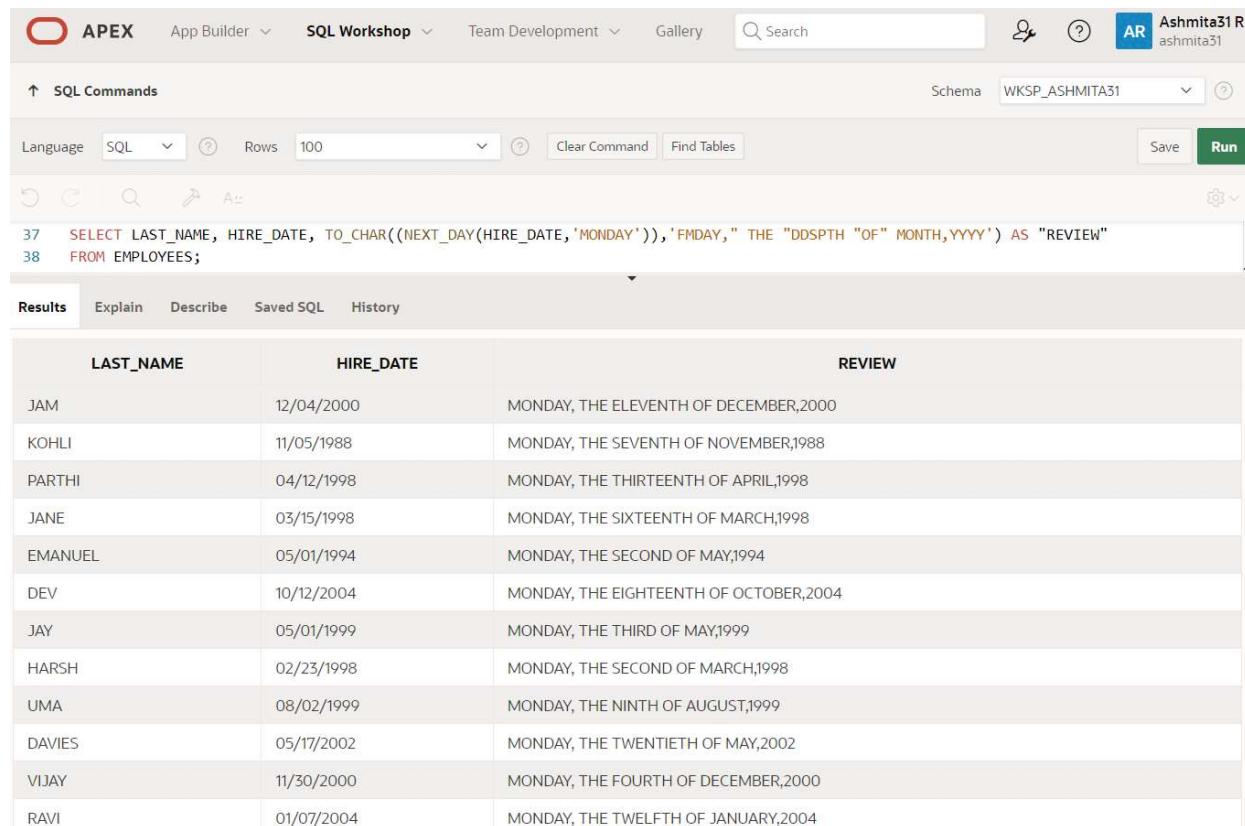
At the bottom, it says "12 rows returned in 0.00 seconds" and "Download".

9. 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."

QUERY:

```
SELECT LAST_NAME, HIRE_DATE,  
TO_CHAR((NEXT_DAY(HIRE_DATE,'MONDAY')),'FMDAY," THE "DDSPHTH "OF"  
MONTH,YYYY') AS "REVIEW" FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information for Ashmita31 R. The SQL Commands tab is active, showing the following SQL code:

```
37  SELECT LAST_NAME, HIRE_DATE, TO_CHAR((NEXT_DAY(HIRE_DATE,'MONDAY')),'FMDAY," THE "DDSPHTH "OF" MONTH,YYYY') AS "REVIEW"  
38  FROM EMPLOYEES;
```

The Results tab is selected, displaying the output of the query:

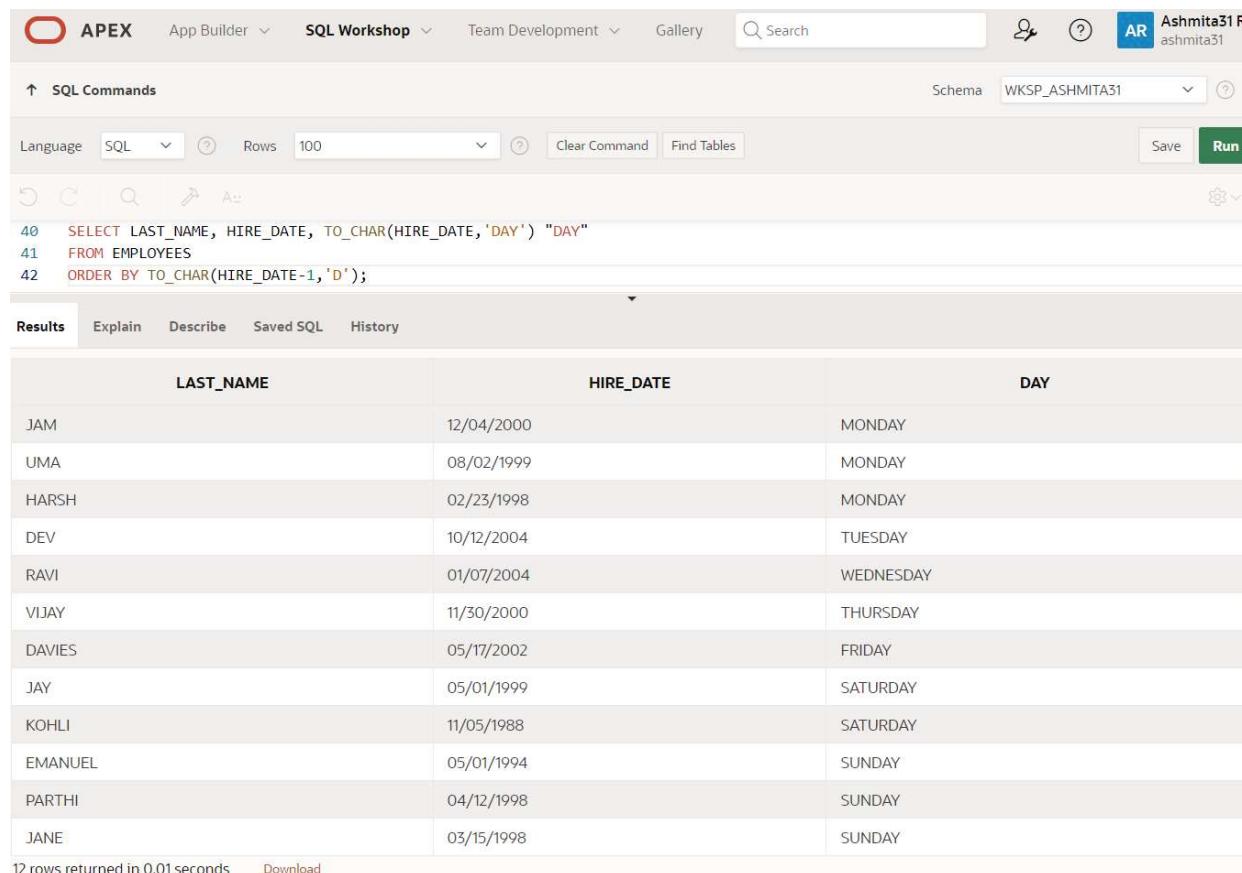
LAST_NAME	HIRE_DATE	REVIEW
JAM	12/04/2000	MONDAY, THE ELEVENTH OF DECEMBER,2000
KOHLI	11/05/1988	MONDAY, THE SEVENTH OF NOVEMBER,1988
PARTHI	04/12/1998	MONDAY, THE THIRTEENTH OF APRIL,1998
JANE	03/15/1998	MONDAY, THE SIXTEENTH OF MARCH,1998
EMANUEL	05/01/1994	MONDAY, THE SECOND OF MAY,1994
DEV	10/12/2004	MONDAY, THE EIGHTEENTH OF OCTOBER,2004
JAY	05/01/1999	MONDAY, THE THIRD OF MAY,1999
HARSH	02/23/1998	MONDAY, THE SECOND OF MARCH,1998
UMA	08/02/1999	MONDAY, THE NINTH OF AUGUST,1999
DAVIES	05/17/2002	MONDAY, THE TWENTIETH OF MAY,2002
VIJAY	11/30/2000	MONDAY, THE FOURTH OF DECEMBER,2000
RAVI	01/07/2004	MONDAY, THE TWELFTH OF JANUARY,2004

10. 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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, Gallery, a search bar, and user information for 'ashmita31'. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_ASHMITA31'. The main area displays the SQL command and its execution results.

SQL Commands:

```
40: SELECT LAST_NAME, HIRE_DATE, TO_CHAR(HIRE_DATE,'DAY') "DAY"  
41: FROM EMPLOYEES  
42: ORDER BY TO_CHAR(HIRE_DATE-1,'D');
```

Results:

LAST_NAME	HIRE_DATE	DAY
JAM	12/04/2000	MONDAY
UMA	08/02/1999	MONDAY
HARSH	02/23/1998	MONDAY
DEV	10/12/2004	TUESDAY
RAVI	01/07/2004	WEDNESDAY
VIJAY	11/30/2000	THURSDAY
DAVIES	05/17/2002	FRIDAY
JAY	05/01/1999	SATURDAY
KOHLI	11/05/1988	SATURDAY
EMANUEL	05/01/1994	SUNDAY
PARTHI	04/12/1998	SUNDAY
JANE	03/15/1998	SUNDAY

12 rows returned in 0.01 seconds [Download](#)

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT :

DISPLAYING DATA FROM MULTIPLE TABLES

EX-NO : 6

DATE:

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

QUERY:

```
SELECT E.LAST_NAME, E.DEPARTMENT_ID, D.DEPT_NAME  
FROM EMPLOYEES E, DEPARTMENT D  
WHERE E.DEPARTMENT_ID = D.DEPT_ID;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, Gallery, and a search bar. The user is signed in as Ashmita31 R (ashmita31). The SQL Commands tab is selected, showing the following SQL code:

```
46  SELECT E.LAST_NAME, E.DEPARTMENT_ID, D.DEPT_NAME  
47  FROM EMPLOYEES E, DEPARTMENTS D  
48  WHERE E.DEPARTMENT_ID = D.DEPT_ID;
```

The Results tab is active, displaying the query results in a grid format:

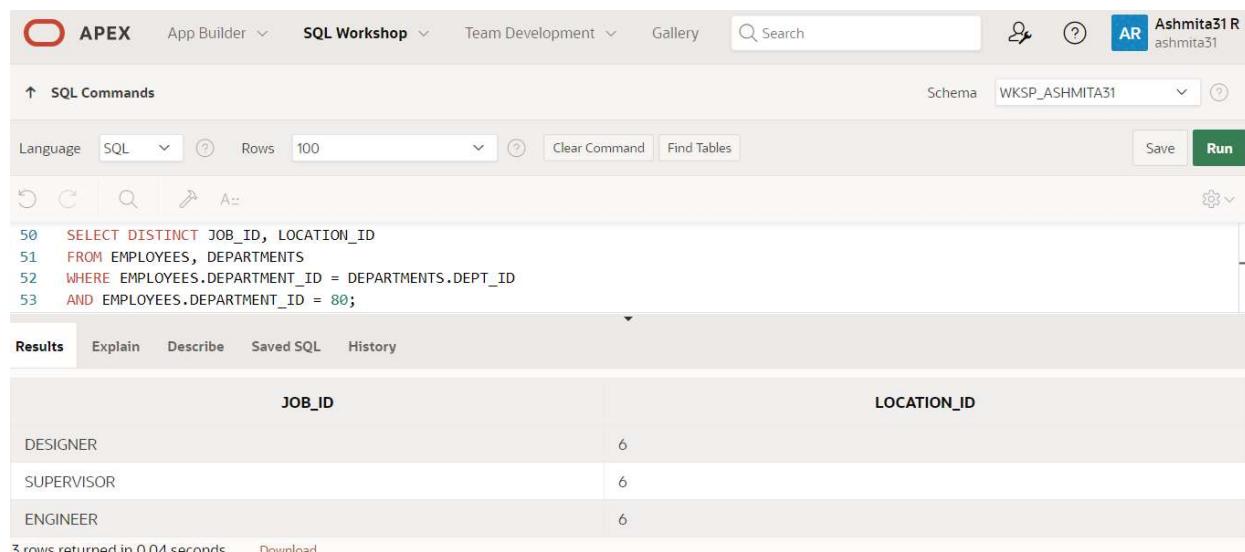
LAST_NAME	DEPARTMENT_ID	DEPT_NAME
JAM	80	MANUFACTURING
KOHLI	80	MANUFACTURING
PARTHI	20	STOCK
JANE	50	HR
EMANUEL	30	FINANCE
DEV	80	MANUFACTURING
JAY	10	MARKETING
UMA	20	STOCK
DAVIES	30	FINANCE
VIJAY	40	MANAGEMENT
RAVI	10	MARKETING

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

QUERY:

```
SELECT DISTINCT JOB_ID, LOCATION_ID
FROM EMPLOYEES, DEPARTMENT
WHERE EMPLOYEES.DEPARTMENT_ID = DEPARTMENT.DEPT_ID
AND EMPLOYEES.DEPARTMENT_ID = 80;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information (Ashmita31 R). The SQL Commands tab is active, showing the executed SQL code:

```
50  SELECT DISTINCT JOB_ID, LOCATION_ID
51  FROM EMPLOYEES, DEPARTMENTS
52  WHERE EMPLOYEES.DEPARTMENT_ID = DEPARTMENTS.DEPT_ID
53  AND EMPLOYEES.DEPARTMENT_ID = 80;
```

The Results tab is selected, displaying the query results in a table:

JOB_ID	LOCATION_ID
DESIGNER	6
SUPERVISOR	6
ENGINEER	6

Below the table, it says "3 rows returned in 0.04 seconds".

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

QUERY:

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME, D.LOCATION_ID, L.CITY  
FROM EMPLOYEES E, DEPARTMENT D, LOCATION L  
WHERE DEPARTMENT_ID = DEPT_ID  
AND D.LOCATION_ID = L.LOCATION_ID  
AND COMMISSION_PCT IS NOT NULL;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user information (Ashmita31 R, ashmita31) are also present. The main area is titled 'SQL Commands' and contains the executed SQL code. Below the code, the 'Results' tab is selected, displaying a grid of employee data. The results show seven rows of data with columns: LAST_NAME, DEPT_NAME, LOCATION_ID, and CITY.

LAST_NAME	DEPT_NAME	LOCATION_ID	CITY
JAM	MANUFACTURING	6	TORONTO
KOHLI	MANUFACTURING	6	TORONTO
JANE	HR	5	LONDON
EMANUEL	FINANCE	3	VALHALLA
DEV	MANUFACTURING	6	TORONTO
DAVIES	FINANCE	3	VALHALLA
VIJAY	MANAGEMENT	4	DC

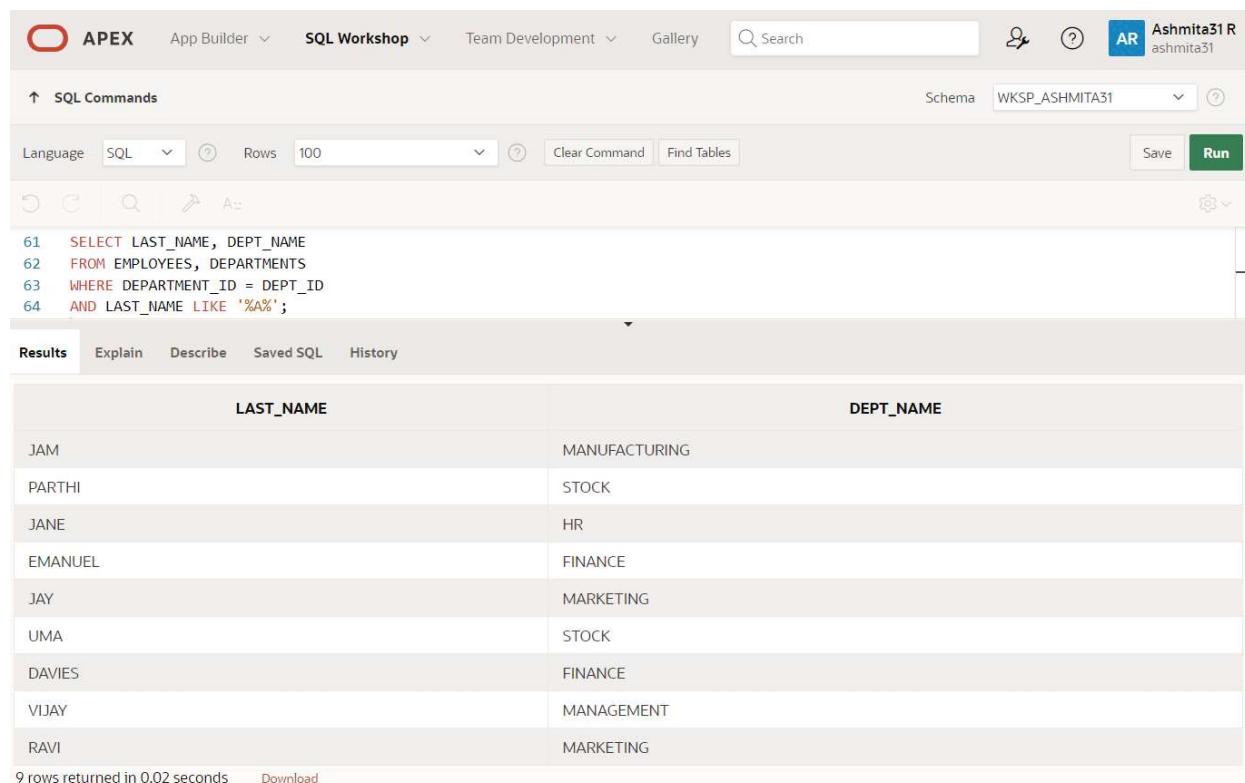
7 rows returned in 0.03 seconds [Download](#)

4. Display the employee last name and department name for all employees who have an a(lowercase) in their last names.

QUERY:

```
SELECT LAST_NAME, DEPT_NAME  
FROM EMPLOYEES, DEPARTMENT  
WHERE DEPARTMENT_ID = DEPT_ID  
AND LAST_NAME LIKE '%a%';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, Gallery, and a search bar. The user is signed in as 'Ashmita31 R' with the schema 'WKSP_ASHMITA31'. The main area is titled 'SQL Commands' and contains the following SQL code:

```
61  SELECT LAST_NAME, DEPT_NAME  
62  FROM EMPLOYEES, DEPARTMENTS  
63  WHERE DEPARTMENT_ID = DEPT_ID  
64  AND LAST_NAME LIKE '%a%';
```

The 'Results' tab is selected, displaying the output of the query:

LAST_NAME	DEPT_NAME
JAM	MANUFACTURING
PARTHI	STOCK
JANE	HR
EMANUEL	FINANCE
JAY	MARKETING
UMA	STOCK
DAVIES	FINANCE
VIJAY	MANAGEMENT
RAVI	MARKETING

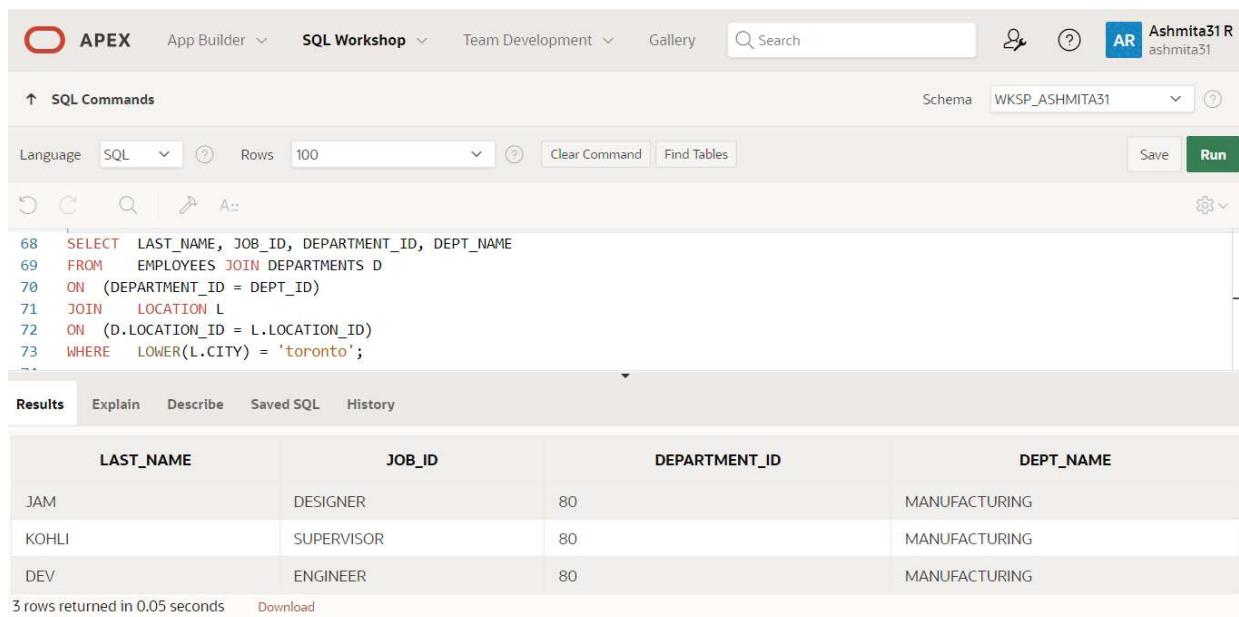
At the bottom of the results pane, it says '9 rows returned in 0.02 seconds' and has a 'Download' link.

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

QUERY:

```
SELECT LAST_NAME, JOB_ID, DEPARTMENT_ID, DEPT_NAME  
FROM EMPLOYEES JOIN DEPARTMENT D  
ON (DEPARTMENT_ID = DEPT_ID)  
JOIN LOCATION L  
ON (D.LOCATION_ID = L.LOCATION_ID)  
WHERE LOWER(L.CITY) = 'toronto';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information (Ashmita31 R). The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_ASHMITA31'. The SQL editor contains the query from the previous step. The 'Run' button is highlighted in green. Below the editor, the 'Results' tab is selected, displaying a grid of employee data. The columns are LAST_NAME, JOB_ID, DEPARTMENT_ID, and DEPT_NAME. The data shows three rows for employees JAM, KOHLI, and DEV, all working in the MANUFACTURING department (DEPARTMENT_ID 80).

LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPT_NAME
JAM	DESIGNER	80	MANUFACTURING
KOHLI	SUPERVISOR	80	MANUFACTURING
DEV	ENGINEER	80	MANUFACTURING

3 rows returned in 0.05 seconds [Download](#)

6. 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

QUERY:

```
SELECT W.LAST_NAME "EMPLOYEE", W.EMPLOYEE_ID "EMP#",  
M.LAST_NAME "MANAGER", M.EMPLOYEE_ID "MGR#"  
FROM EMPLOYEES W JOIN EMPLOYEES M  
ON (W.MANAGER_ID = M.EMPLOYEE_ID);
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar is present, along with user information for 'Ashmita31 R' and 'ashmita31'. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_ASHMITA31'. The SQL command area contains the following code:

```
75  SELECT W.LAST_NAME "EMPLOYEE", W.EMPLOYEE_ID "EMP#",  
76  M.LAST_NAME "MANAGER", M.EMPLOYEE_ID "MGR#"  
77  FROM EMPLOYEES W JOIN EMPLOYEES M  
78  ON (W.MANAGER_ID = M.EMPLOYEE_ID);
```

The results tab is active, displaying the output of the query:

EMPLOYEE	EMP#	MANAGER	MGR#
DEV	108	JAM	107
HARSH	113	JANE	104
DAVIES	109	EMANUEL	105
JAM	107	VIJAY	106
KOHLI	109	VIJAY	106
JANE	104	VIJAY	106
EMANUEL	105	VIJAY	106
RAVI	101	VIJAY	106
PARTHI	103	RAVI	101
JAY	100	RAVI	101
UMA	102	RAVI	101

7. Modify lab4_6.sql to display all employees including King, who has no manager. Order the results by the employee number.

QUERY:

```
SELECT W.LAST_NAME "EMPLOYEE", W.EMPLOYEE_ID "EMP#",  
M.LAST_NAME "MANAGER", M.EMPLOYEE_ID "MGR#" FROM EMPLOYEES W  
LEFT OUTER JOIN EMPLOYEES M ON (W.MANAGER_ID = M.EMPLOYEE_ID)  
ORDER BY W.EMPLOYEE_ID;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', 'Gallery', a search bar, and a user profile 'Ashmita31 R'. Below the toolbar, the schema is set to 'WKSP_ASHMITA31'. The main area displays the SQL command and its results.

SQL Commands:

```
80  SELECT W.LAST_NAME "EMPLOYEE", W.EMPLOYEE_ID "EMP#",  
81  M.LAST_NAME "MANAGER", M.EMPLOYEE_ID "MGR#"  
82  FROM EMPLOYEES W  
83  LEFT OUTER JOIN EMPLOYEES M  
84  ON (W.MANAGER_ID = M.EMPLOYEE_ID)  
85  ORDER BY W.EMPLOYEE_ID;
```

Results:

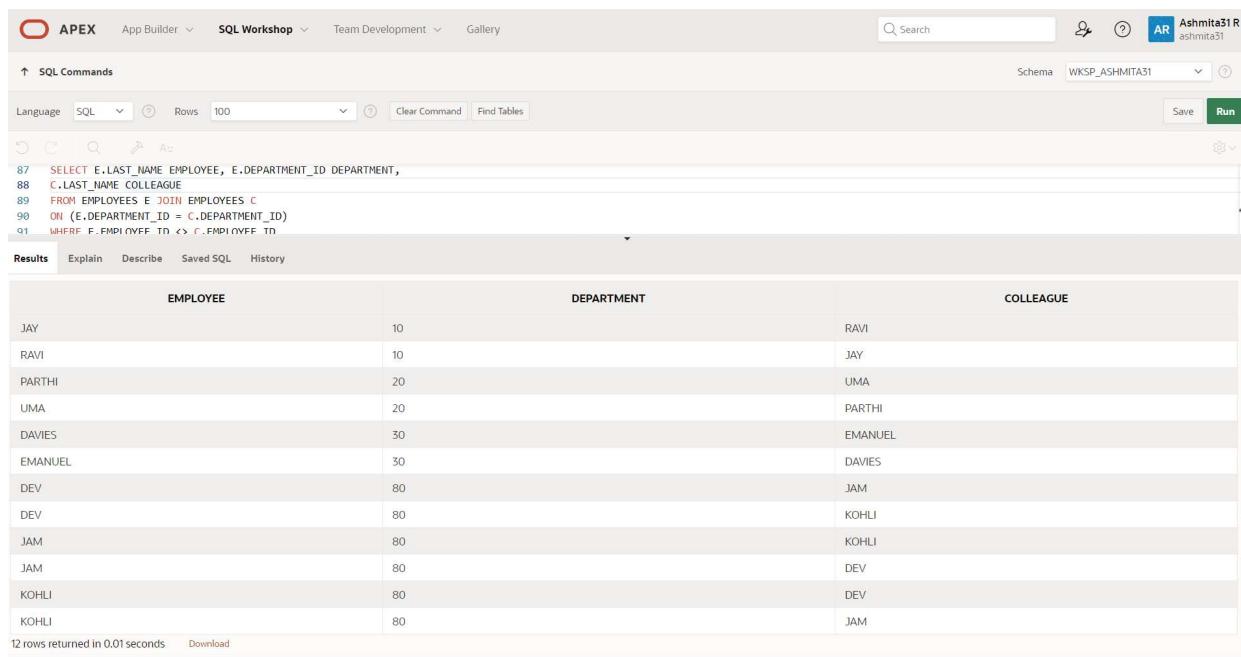
EMPLOYEE	EMP#	MANAGER	MGR#
JAY	100	RAVI	101
RAVI	101	VIJAY	106
UMA	102	RAVI	101
PARTHI	103	RAVI	101
JANE	104	VIJAY	106
EMANUEL	105	VIJAY	106
VIJAY	106	-	-
JAM	107	VIJAY	106
DEV	108	JAM	107

8. 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.

QUERY:

```
SELECT E.LAST_NAME EMPLOYEE, E.DEPARTMENT_ID DEPARTMENT,
C.LAST_NAME COLLEAGUE
FROM EMPLOYEES E JOIN EMPLOYEES C
ON (E.DEPARTMENT_ID = C.DEPARTMENT_ID)
WHERE E.EMPLOYEE_ID <> C.EMPLOYEE_ID
ORDER BY E.DEPARTMENT_ID, E.LAST_NAME;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query has been run, and the results are displayed in a grid. The columns are labeled 'EMPLOYEE', 'DEPARTMENT', and 'COLLEAGUE'. The data shows pairs of employees from the same department, excluding themselves.

EMPLOYEE	DEPARTMENT	COLLEAGUE
JAY	10	RAVI
RAVI	10	JAY
PARTHI	20	UMA
UMA	20	PARTHI
DAVIES	30	EMANUEL
EMANUEL	30	DAVIES
DEV	80	JAM
DEV	80	KOHLI
JAM	80	KOHLI
JAM	80	DEV
KOHLI	80	DEV
KOHLI	80	JAM

9. Show the structure of the JOB_GRADES table. Create a query that displays the name, job, department name, salary, and grade for all employees.

QUERY:

DESC JOB_GRADES;

```
SELECT E.LAST_NAME, E.JOB_ID, D.DEPT_NAME, E.SALARY, J.GRADE_LEVEL  
FROM EMPLOYEES E JOIN DEPARTMENT D ON (E.DEPARTMENT_ID = D.DEPT_ID)  
JOIN JOB_GRADES J ON (E.SALARY BETWEEN J.LOWEST_SAL AND  
J.HIGHEST_SAL);
```

OUTPUT:

The screenshot shows two instances of the Oracle SQL Workshop interface. The top instance displays the structure of the JOB_GRADES table, while the bottom instance shows the execution of a query.

Table Structure (Top Window):

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
JOB_GRADES	GRADE_LEVEL	VARCHAR2	2	-	-	-	✓	-	-
	LOWEST_SAL	NUMBER	22	-	-	-	✓	-	-
	HIGHEST_SAL	NUMBER	22	-	-	-	✓	-	-

Query Result (Bottom Window):

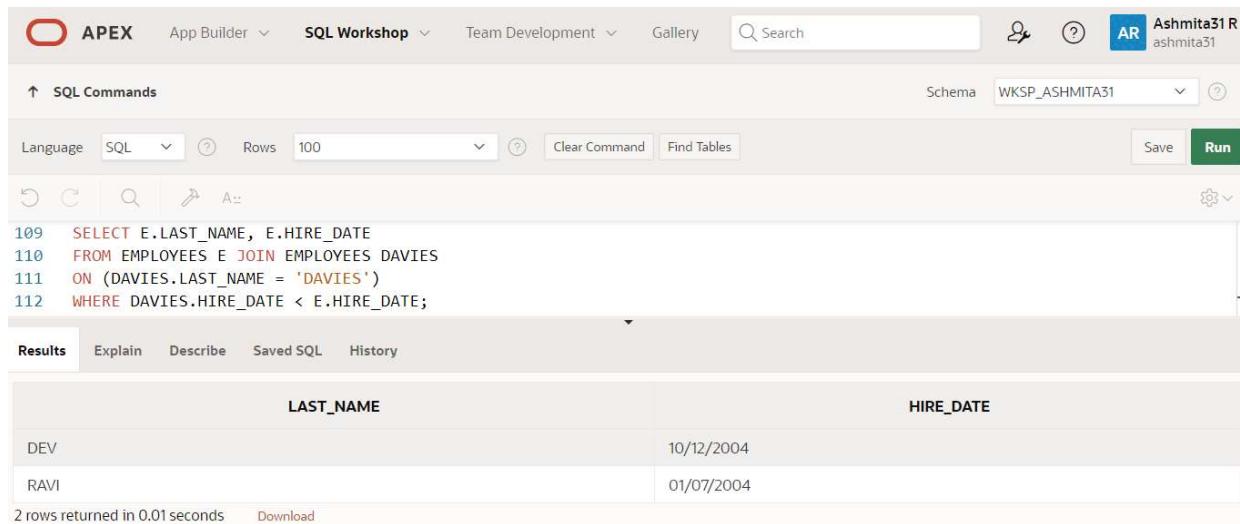
Employee Name	Job	Department	Salary	Grade Level
JAM	DESIGNER	MANUFACTURING	15000	C
DEV	ENGINEER	MANUFACTURING	15000	C
KOHLI	SUPERVISOR	MANUFACTURING	20000	D
VIJAY	MANAGER	MANAGEMENT	33400.12	E
EMANUEL	FI_MANAGER	FINANCE	22000	D
DAVIES	AUDITOR	FINANCE	30000	E
JANE	HR_MANAGER	HR	25400.12	D
UMA	ST_CLERK	STOCK	7000	A
PARTHI	ST_CLERK	STOCK	8650.82	A
JAY	SL REP	MARKETING	13400.12	B
RAVI	MK_MANAGER	MARKETING	20000	D

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

QUERY:

```
SELECT E.LAST_NAME, E.HIRE_DATE FROM EMPLOYEES E JOIN EMPLOYEES  
DAVIES ON (DAVIES.LAST_NAME = 'DAVIES') WHERE DAVIES.HIRE_DATE <  
E.HIRE_DATE;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and a user profile for Ashmita31. The main area is titled 'SQL Commands' and shows the following SQL code:

```
109  SELECT E.LAST_NAME, E.HIRE_DATE  
110  FROM EMPLOYEES E JOIN EMPLOYEES DAVIES  
111  ON (DAVIES.LAST_NAME = 'DAVIES')  
112  WHERE DAVIES.HIRE_DATE < E.HIRE_DATE;
```

The 'Results' tab is selected, displaying the output in a table:

LAST_NAME	HIRE_DATE
DEV	10/12/2004
RAVI	01/07/2004

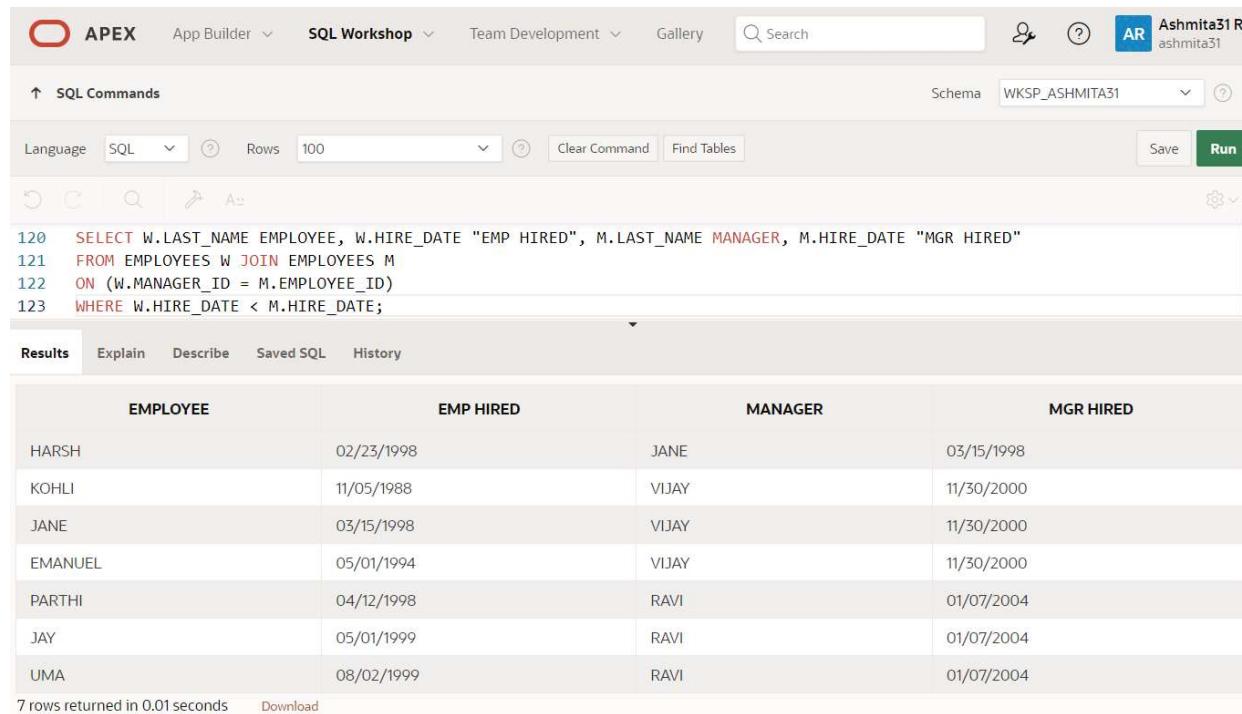
Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link.

11. 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 columns Employee, Emp Hired, Manager, and Mgr Hired, respectively.

QUERY:

```
SELECT W.LAST_NAME EMPLOYEE, W.HIRE_DATE "EMP HIRED", M.LAST_NAME  
MANAGER, M.HIRE_DATE "MGR HIRED" FROM EMPLOYEES W JOIN EMPLOYEES  
M ON (W.MANAGER_ID = M.EMPLOYEE_ID) WHERE W.HIRE_DATE <  
M.HIRE_DATE;
```

OUTPUT :



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and a user profile for Ashmita31. The SQL Commands tab is active, showing the following SQL code:

```
120  SELECT W.LAST_NAME EMPLOYEE, W.HIRE_DATE "EMP HIRED", M.LAST_NAME MANAGER, M.HIRE_DATE "MGR HIRED"  
121  FROM EMPLOYEES W JOIN EMPLOYEES M  
122  ON (W.MANAGER_ID = M.EMPLOYEE_ID)  
123  WHERE W.HIRE_DATE < M.HIRE_DATE;
```

The Results tab is selected, displaying the output in a grid format:

EMPLOYEE	EMP HIRED	MANAGER	MGR HIRED
HARSH	02/23/1998	JANE	03/15/1998
KOHLI	11/05/1988	VIJAY	11/30/2000
JANE	03/15/1998	VIJAY	11/30/2000
EMANUEL	05/01/1994	VIJAY	11/30/2000
PARTHI	04/12/1998	RAVI	01/07/2004
JAY	05/01/1999	RAVI	01/07/2004
UMA	08/02/1999	RAVI	01/07/2004

Below the table, it says "7 rows returned in 0.01 seconds" and there is a "Download" link.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT :

AGGREGATING DATA USING GROUP FUNCTIONS

EX-NO : 8

DATE:

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

True/False

TRUE

2. Group functions include nulls in calculations.

True/False

FALSE

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

True/False

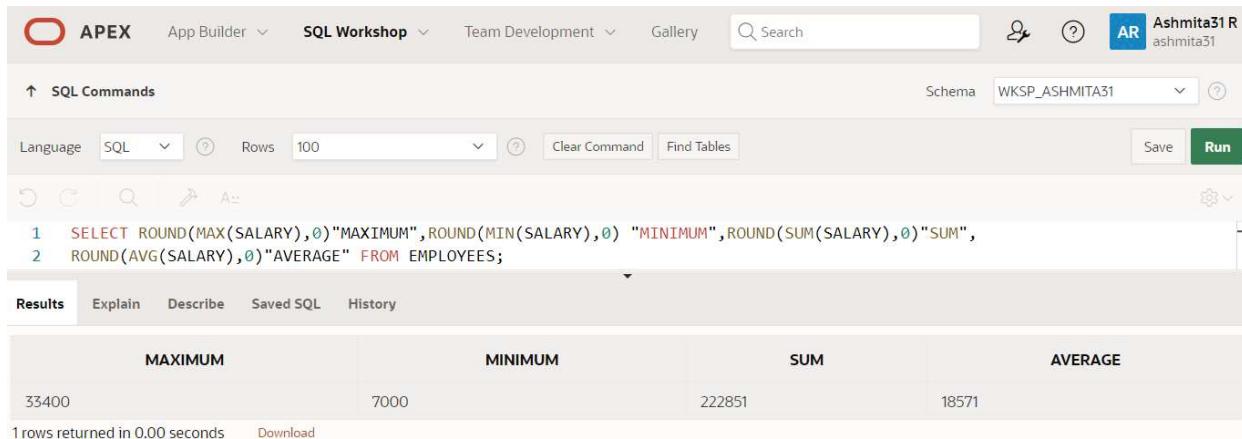
FALSE

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

QUERY:

```
SELECT ROUND(MAX(SALARY),0) AS "MAXIMUM",
ROUND(MIN(SALARY),0) AS "MINIMUM", ROUND(SUM(SALARY),0) AS "SUM",
ROUND(AVG(SALARY),0) AS "AVERAGE" FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. A search bar and user profile are also present. The main workspace displays the following content:

- SQL Commands:** The query is displayed:

```
1 SELECT ROUND(MAX(SALARY),0) "MAXIMUM", ROUND(MIN(SALARY),0) "MINIMUM", ROUND(SUM(SALARY),0) "SUM",
2 ROUND(AVG(SALARY),0) "AVERAGE" FROM EMPLOYEES;
```
- Results:** The results are presented in a table:

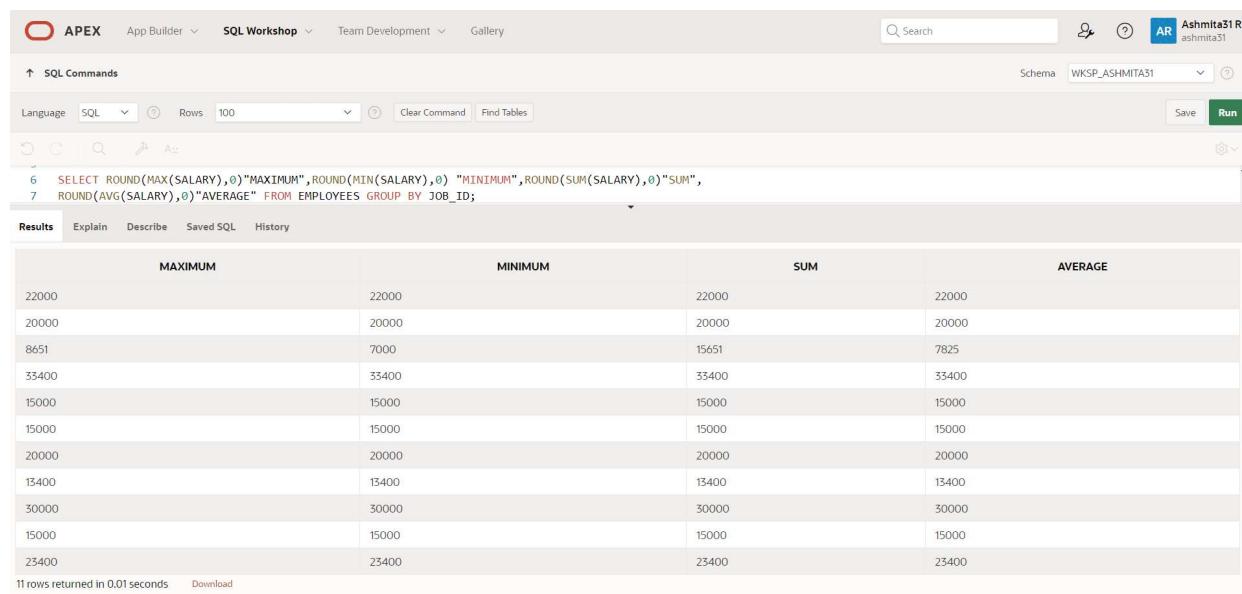
MAXIMUM	MINIMUM	SUM	AVERAGE
33400	7000	222851	18571
- Timing:** The message "1 rows returned in 0.00 seconds" is shown at the bottom left.

5. Modify the above query to display the minimum, maximum, sum, and average salary for each job type.

QUERY:

```
SELECT JOB_ID,ROUND(MAX(SALARY),0) AS "MAXIMUM",
ROUND(MIN(SALARY),0) AS "MINIMUM",
ROUND(SUM(SALARY),0) AS "SUM", ROUND(AVG(SALARY),0) AS "AVERAGE"
FROM EMPLOYEES GROUP BY JOB_ID;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL command entered is:

```
6  SELECT ROUND(MAX(SALARY),0)"MAXIMUM",ROUND(MIN(SALARY),0) "MINIMUM",ROUND(SUM(SALARY),0)"SUM",
7  ROUND(AVG(SALARY),0)"AVERAGE" FROM EMPLOYEES GROUP BY JOB_ID;
```

The results are displayed in a table:

	MAXIMUM	MINIMUM	SUM	AVERAGE
22000	22000	22000	22000	22000
20000	20000	20000	20000	20000
8651	7000	15651	7825	7825
33400	33400	33400	33400	33400
15000	15000	15000	15000	15000
15000	15000	15000	15000	15000
20000	20000	20000	20000	20000
13400	13400	13400	13400	13400
30000	30000	30000	30000	30000
15000	15000	15000	15000	15000
23400	23400	23400	23400	23400

11 rows returned in 0.01 seconds Download

6. Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. A search bar and a user profile for 'Ashmita' are also at the top. Below the tabs, there's a toolbar with icons for Undo, Redo, Find, Replace, and others. The main area is divided into two sections: 'SQL Commands' and 'Results'. In the 'SQL Commands' section, the code is displayed:

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

In the 'Results' section, the output is shown in a table:

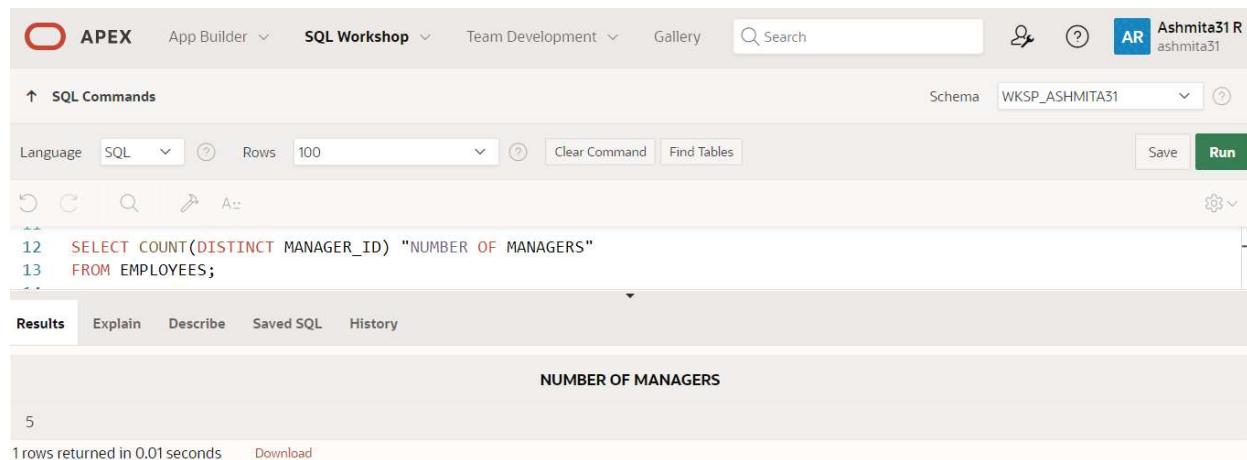
JOB_ID	COUNT(*)
FL_MANAGER	1
SUPERVISOR	1
ST_CLERK	2
MANAGER	1
ENGINEER	1
QA	1
MK_MANAGER	1
SL_REP	1
AUDITOR	1
DESIGNER	1
HR_MANAGER	1

7. 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.

QUERY:

```
SELECT COUNT(DISTINCT MANAGER_ID) "NUMBER OF MANAGERS"  
FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The user is signed in as 'Ashmita31 R' (ashmita31). The schema is set to 'WKSP_ASHMITA31'. The main area displays the SQL command entered:

```
12  SELECT COUNT(DISTINCT MANAGER_ID) "NUMBER OF MANAGERS"  
13  FROM EMPLOYEES;
```

The results tab is selected, showing the output:

NUMBER OF MANAGERS
5

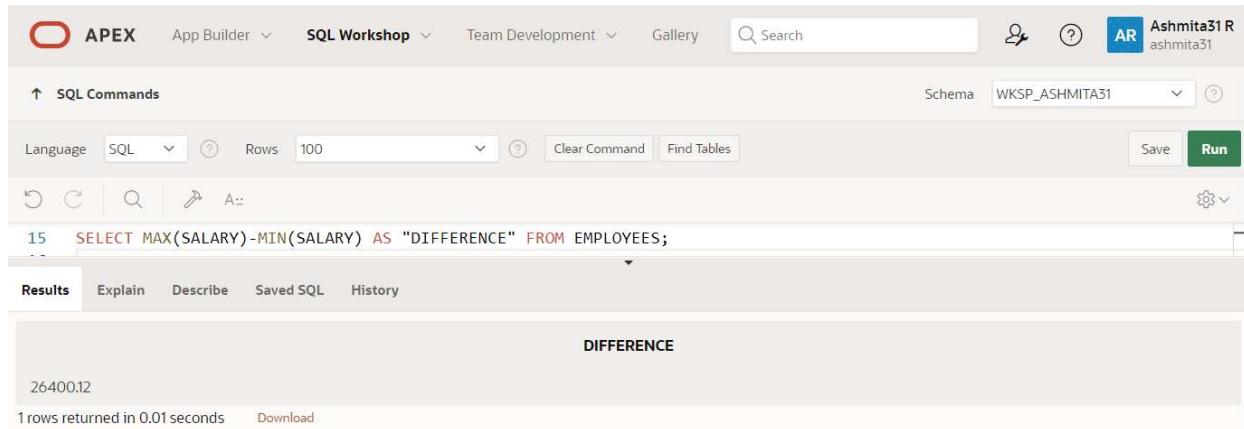
Below the results, it says '1 rows returned in 0.01 seconds' and has a 'Download' link.

8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile information ('Ashmita31 R') are also present. The main workspace displays the SQL command:

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

The results section shows a single row with the value '26400.12' under the column header 'DIFFERENCE'. Below the results, it indicates '1 rows returned in 0.01 seconds'.

9. Create a report to 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 \$6,000 or less. Sort the output in descending order of salary.

QUERY:

```
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;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery, along with a search bar and user information for 'Ashmita31'.

In the SQL Commands section, the query is displayed:

```
17  SELECT MANAGER_ID,MIN(SALARY)  
18  FROM EMPLOYEES  
19  WHERE MANAGER_ID IS NOT NULL  
20  GROUP BY MANAGER_ID  
21  HAVING MIN(SALARY) > 6000  
22  ORDER BY MIN(SALARY) DESC;
```

The Results tab is selected, showing the output of the query:

MANAGER_ID	MIN(SALARY)
105	30000
106	15000
107	15000
104	15000
101	7000

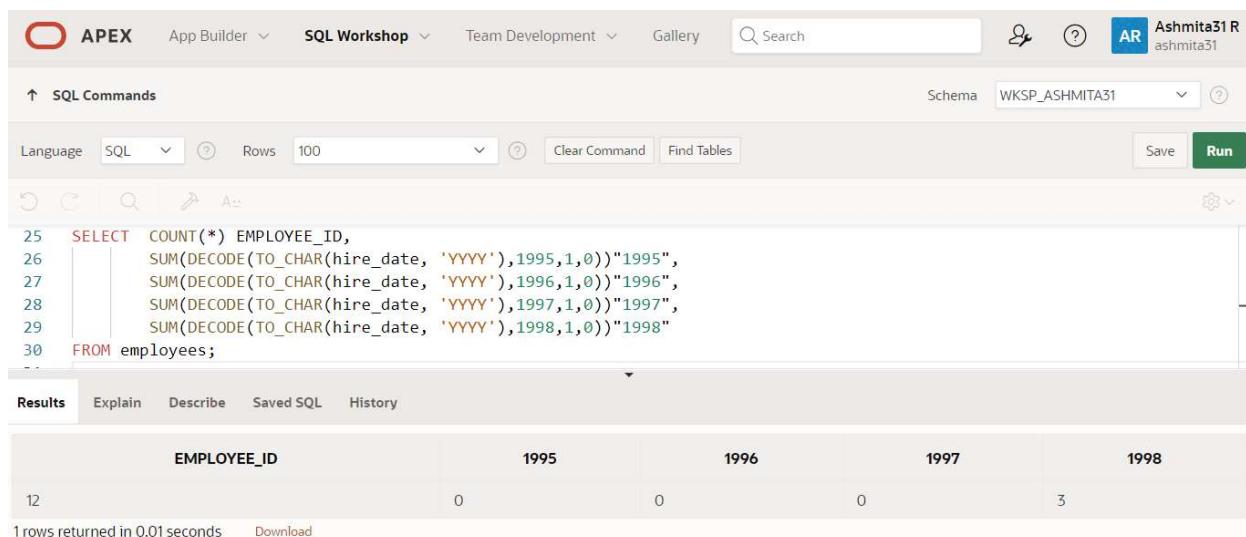
Below the table, it says '5 rows returned in 0.01 seconds' and there is a 'Download' link.

10. Create a query to 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.

QUERY:

```
SELECT COUNT(*) AS TOTAL,  
SUM(DECODE(EXTRACT(YEAR FROM HIRE_DATE), 1995, 1, 0)) AS "1995",  
SUM(DECODE(EXTRACT(YEAR FROM HIRE_DATE), 1996, 1, 0)) AS "1996",  
SUM(DECODE(EXTRACT(YEAR FROM HIRE_DATE), 1997, 1, 0)) AS "1997",  
SUM(DECODE(EXTRACT(YEAR FROM HIRE_DATE), 1998, 1, 0)) AS "1998"  
FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema is set to WKSP_ASHMITA31. The SQL command window contains the query from the previous step. The results window displays the output:

EMPLOYEE_ID	1995	1996	1997	1998
12	0	0	0	3

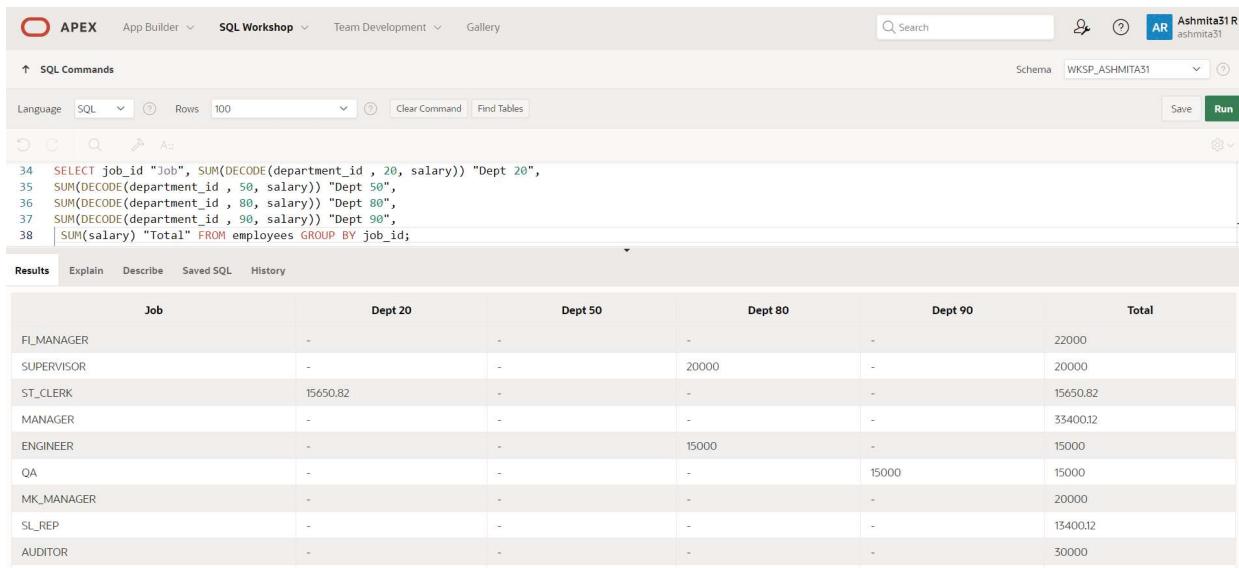
1 rows returned in 0.01 seconds

11. 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.

QUERY:

```
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;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The SQL Workshop tab has a sub-menu for SQL Commands. The schema is set to WKSP_ASHMITA31. The main area shows the SQL code and its execution results.

SQL Commands:

```
34  SELECT job_id "Job", SUM(DECODE(department_id , 20, salary)) "Dept 20",
35  SUM(DECODE(department_id , 50, salary)) "Dept 50",
36  SUM(DECODE(department_id , 80, salary)) "Dept 80",
37  SUM(DECODE(department_id , 90, salary)) "Dept 90",
38  | SUM(salary) "Total" FROM employees GROUP BY job_id;
```

Results:

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
FL_MANAGER	-	-	-	-	22000
SUPERVISOR	-	-	20000	-	20000
ST_CLERK	15650.82	-	-	-	15650.82
MANAGER	-	-	-	-	35400.12
ENGINEER	-	-	15000	-	15000
QA	-	-	-	15000	15000
MK_MANAGER	-	-	-	-	20000
SL REP	-	-	-	-	13400.12
AUDITOR	-	-	-	-	30000

12. Write a query to display each department's name, location, number of employees, and the average salary for all the employees in that department. Label the column name-Location, Number of people, and salary respectively. Round the average salary to two decimal places.

QUERY:

```
SELECT D.DEPT_NAME "NAME", D.LOCATION_ID "LOCATION ", COUNT(*)  
"NUMBER OF PEOPLE", ROUND(AVG(SALARY),2) "SALARY" FROM EMPLOYEES  
E, DEPARTMENT D WHERE E.DEPARTMENT_ID = D.DEPT_ID GROUP BY  
D.DEPT_NAME, D.LOCATION_ID;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows a user profile for 'Ashmita51 R' (ashmita51). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
40 select d.dept_name"NAME",d.location_id "LOCATIONS", count(*)"NUMBER OF PEOPLE",round(avg(e.salary),2) "SAL"  
41 from departments d, employees e  
42 where e.department_id=d.dept_id  
43 group by d.dept_name,d.location_id;
```

The Results tab displays the query output as a table:

NAME	LOCATIONS	NUMBER OF PEOPLE	SAL
STOCK	2	2	7825.41
FINANCE	3	2	26000
MARKETING	1	2	16700.06
HR	5	1	23400.12
MANUFACTURING	6	3	16666.67
MANAGEMENT	4	1	33400.12

Below the table, it says '6 rows returned in 0.04 seconds'.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT :

SUB-QUERIES

EX-NO : 9

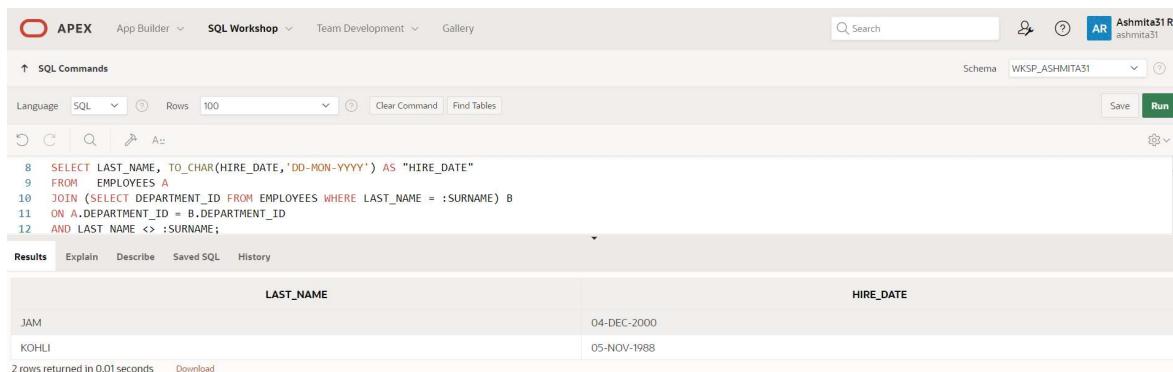
DATE:

1. The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

QUERY:

```
SELECT LAST_NAME, TO_CHAR(HIRE_DATE,'DD-MON-YYYY') AS "HIRE_DATE"
FROM EMPLOYEES A JOIN (SELECT DEPARTMENT_ID FROM EMPLOYEES
WHERE LAST_NAME = :SURNAME) B
ON A.DEPARTMENT_ID = B.DEPARTMENT_ID AND LAST_NAME <> :SURNAME;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL command window contains the following code:

```
8  SELECT LAST_NAME, TO_CHAR(HIRE_DATE,'DD-MON-YYYY') AS "HIRE_DATE"
9  FROM EMPLOYEES A
10 JOIN (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME = :SURNAME) B
11 ON A.DEPARTMENT_ID = B.DEPARTMENT_ID
12 AND LAST_NAME <> :SURNAME;
```

The results section shows two rows of data:

LAST_NAME	HIRE_DATE
JAM	04-DEC-2000
KOHLI	05-NOV-1988

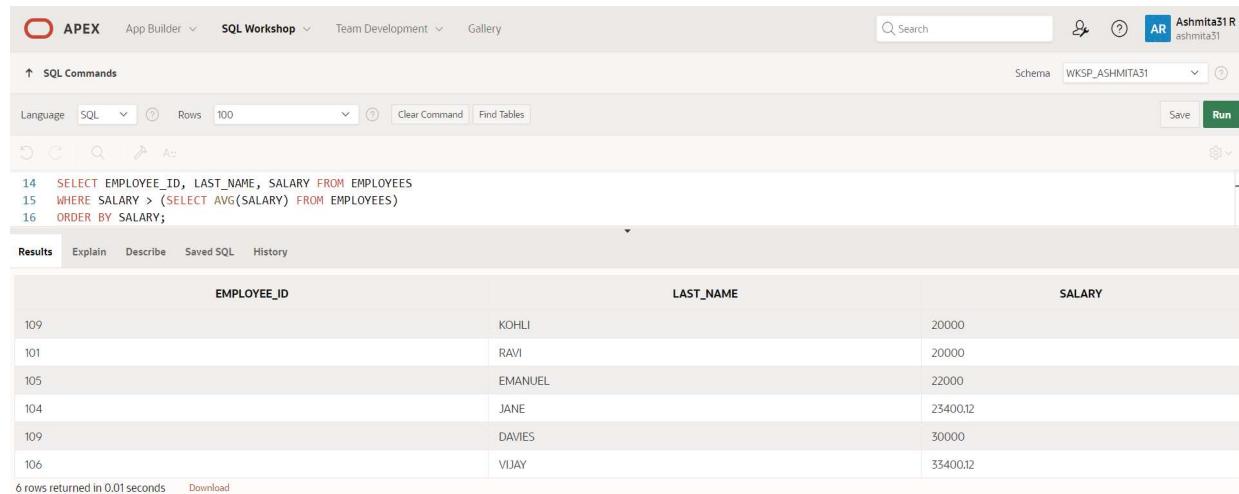
Below the table, it says "2 rows returned in 0.01 seconds".

2. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon, and a connection named 'Ashmita31 R ashmita31'. The main area is titled 'SQL Commands' with a sub-section '14-16'. The code entered is:

```
14: SELECT EMPLOYEE_ID, LAST_NAME, SALARY FROM EMPLOYEES  
15: WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES)  
16: ORDER BY SALARY;
```

Below the code, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the query output in a grid format:

EMPLOYEE_ID	LAST_NAME	SALARY
109	KOHLI	20000
101	RAVI	20000
105	EMANUEL	22000
104	JANE	23400.12
109	DAVIES	30000
106	VIJAY	33400.12

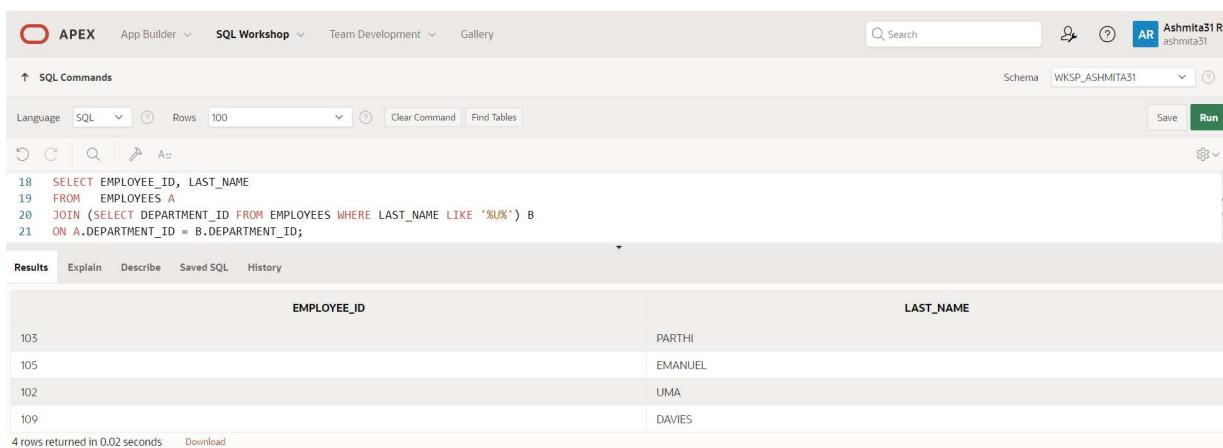
At the bottom left, it says '6 rows returned in 0.01 seconds' and there's a 'Download' link.

3. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a u.

QUERY:

```
SELECT EMPLOYEE_ID, LAST_NAME FROM EMPLOYEES A JOIN (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME LIKE '%U%') B ON A.DEPARTMENT_ID = B.DEPARTMENT_ID;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query is displayed in the SQL Commands pane:

```
18: SELECT EMPLOYEE_ID, LAST_NAME
19: FROM EMPLOYEES A
20: JOIN (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME LIKE '%U%') B
21: ON A.DEPARTMENT_ID = B.DEPARTMENT_ID;
```

The Results pane shows the output:

EMPLOYEE_ID	LAST_NAME
103	PARTHI
105	EMANUEL
102	UMA
109	DAVIES

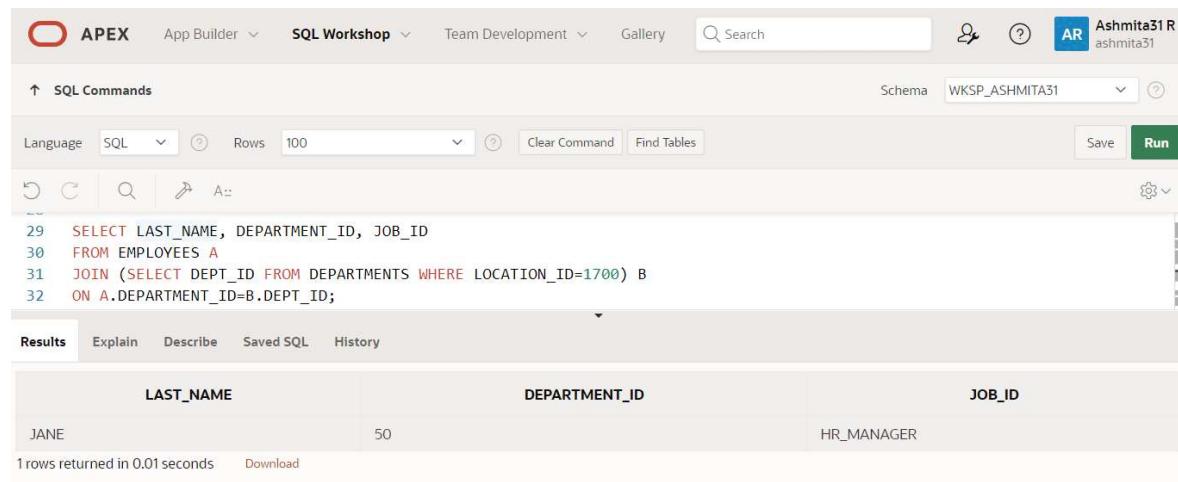
4 rows returned in 0.02 seconds Download

4. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

QUERY:

```
SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID FROM EMPLOYEES A  
JOIN (SELECT DEPT_ID FROM DEPARTMENT WHERE LOCATION_ID=1700) B  
ON A.DEPARTMENT_ID=B.DEPT_ID;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user information (Ashmita31 R) are also present. The main workspace shows the following content:

SQL Commands tab selected. The schema is set to WKSP_ASHMITA31. The SQL editor contains the following code:

```
29  SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID  
30  FROM EMPLOYEES A  
31  JOIN (SELECT DEPT_ID FROM DEPARTMENTS WHERE LOCATION_ID=1700) B  
32  ON A.DEPARTMENT_ID=B.DEPT_ID;
```

The Results tab is selected, displaying the query results in a table:

LAST_NAME	DEPARTMENT_ID	JOB_ID
JANE	50	HR_MANAGER

Below the table, it says "1 rows returned in 0.01 seconds" and there is a "Download" link.

5. Create a report for HR that displays the last name and salary of every employee who reports to King.

QUERY:

```
SELECT LAST_NAME, SALARY FROM EMPLOYEES  
WHERE MANAGER_ID IN (SELECT EMPLOYEE_ID FROM EMPLOYEES WHERE  
MANAGER_ID IS NULL);
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and a user profile for Ashmita31. The main area is titled 'SQL Commands' with tabs for Language (SQL selected), Rows (100), Clear Command, Find Tables, Save, and Run. The code entered is:

```
39: SELECT LAST_NAME, SALARY FROM EMPLOYEES  
40: WHERE MANAGER_ID IN (SELECT EMPLOYEE_ID FROM EMPLOYEES WHERE MANAGER_ID IS NULL);
```

The 'Results' tab is selected, displaying a table with two columns: LAST_NAME and SALARY. The data returned is:

LAST_NAME	SALARY
JAM	15000
KOHLI	20000
JANE	23400.12
EMANUEL	22000
RAVI	20000

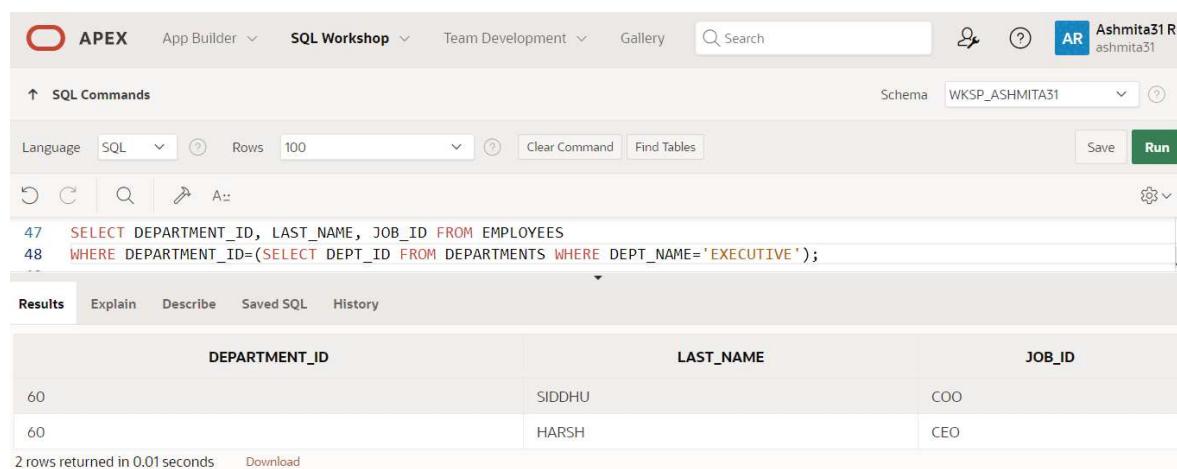
At the bottom, it says '5 rows returned in 0.02 seconds' and has a 'Download' link.

6. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

QUERY:

```
SELECT DEPARTMENT_ID, LAST_NAME, JOB_ID FROM EMPLOYEES  
WHERE DEPARTMENT_ID=(SELECT DEPT_ID FROM DEPARTMENT WHERE  
DEPT_NAME='EXECUTIVE');
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile 'Ashmita31 R' are also present. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_ASHMITA31'. The code editor contains two lines of SQL:

```
47  SELECT DEPARTMENT_ID, LAST_NAME, JOB_ID FROM EMPLOYEES  
48  WHERE DEPARTMENT_ID=(SELECT DEPT_ID FROM DEPARTMENTS WHERE DEPT_NAME='EXECUTIVE');
```

The 'Results' tab is selected, displaying the query results in a grid format:

DEPARTMENT_ID	LAST_NAME	JOB_ID
60	SIDDHU	COO
60	HARSH	CEO

Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link.

7. Modify the query 3 to display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a u.

QUERY:

```
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%');
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and a user profile for Ashmita31. The SQL Commands tab is active, showing the following SQL code:

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

Below the code, the Results tab is selected, displaying the query results in a grid:

EMPLOYEE_ID	LAST_NAME	SALARY
114	SIDDHU	40000
113	HARSH	45000
105	EMANUEL	22000
109	DAVIES	30000

At the bottom left, it says "4 rows returned in 0.01 seconds".

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT :

USING THE SET OPERATORS

EX-NO : 10

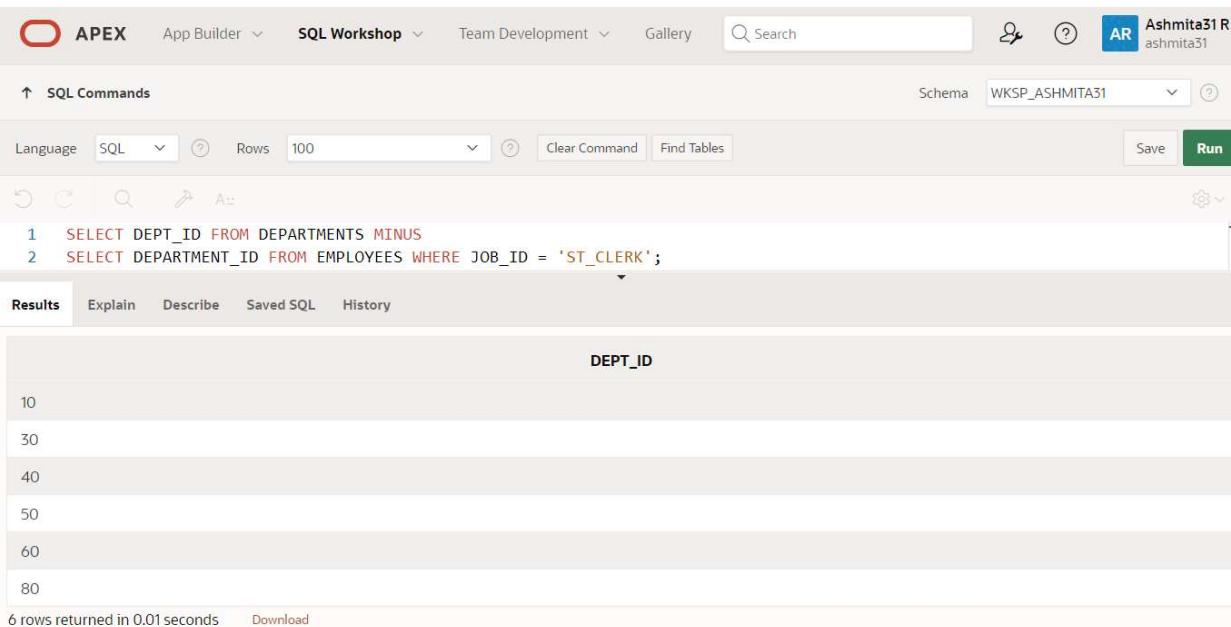
DATE:

1. The HR department needs a list of department IDs for departments that do not contain the job ID ST_CLERK. Use set operators to create this report.

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is signed in as Ashmita31 R (ashmita31). The SQL Commands tab is selected, showing the following code:

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

The Results tab is active, displaying the output:

DEPT_ID
10
30
40
50
60
80

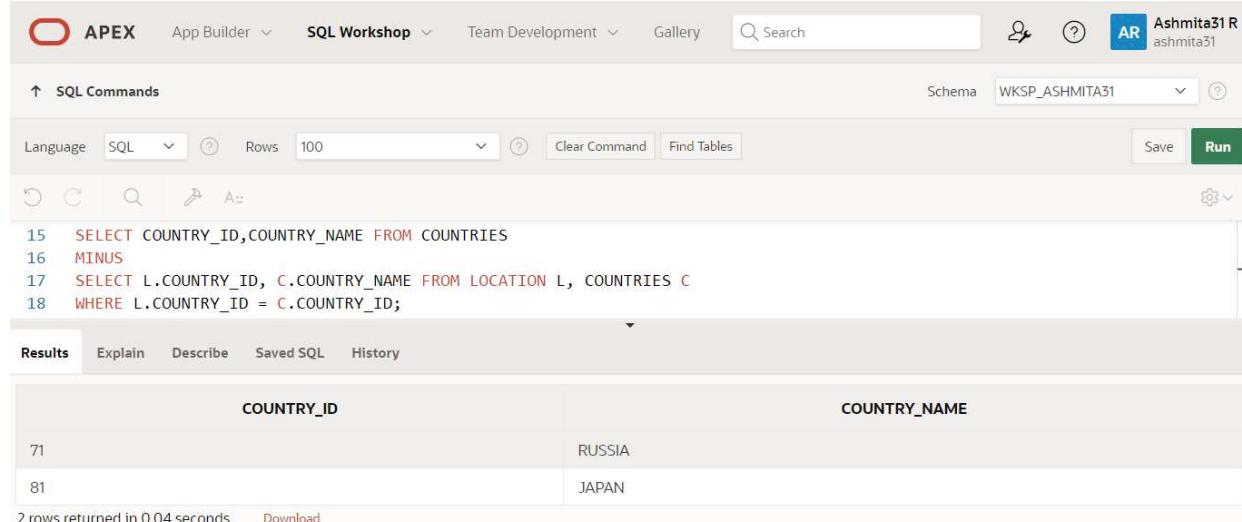
At the bottom left, it says "6 rows returned in 0.01 seconds".

2. The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use set operators to create this report.

QUERY:

```
SELECT COUNTRY_ID,COUNTRY_NAME FROM COUNTRIES  
MINUS  
SELECT L.COUNTRY_ID,C.COUNTRY_NAME FROM LOCATION L, COUNTRIES C  
WHERE L.COUNTRY_ID = C.COUNTRY_ID;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user information for 'Ashmita31 R' are also present. The main area displays the SQL command history and results.

SQL Commands:

```
15  SELECT COUNTRY_ID,COUNTRY_NAME FROM COUNTRIES  
16  MINUS  
17  SELECT L.COUNTRY_ID, C.COUNTRY_NAME FROM LOCATION L, COUNTRIES C  
18  WHERE L.COUNTRY_ID = C.COUNTRY_ID;
```

Results:

COUNTRY_ID	COUNTRY_NAME
71	RUSSIA
81	JAPAN

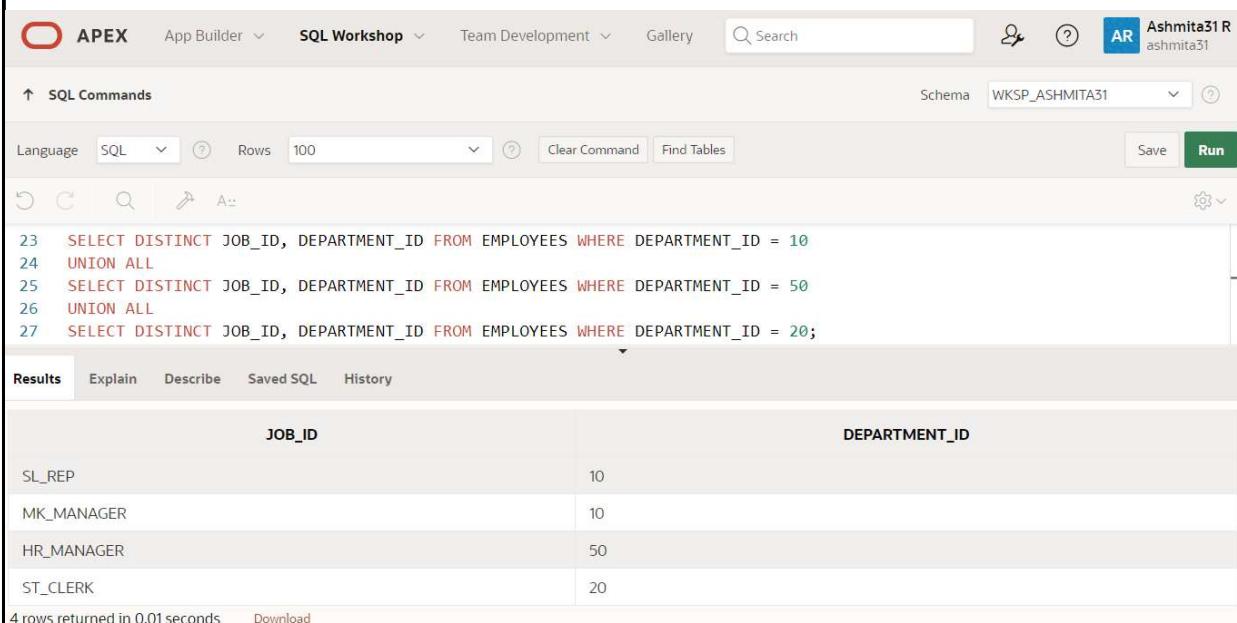
2 rows returned in 0.04 seconds [Download](#)

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

QUERY:

```
SELECT DISTINCT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES WHERE  
DEPARTMENT_ID = 10  
UNION ALL  
SELECT DISTINCT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES WHERE  
DEPARTMENT_ID = 50  
UNION ALL  
SELECT DISTINCT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES WHERE  
DEPARTMENT_ID = 20;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and a user profile for Ashmita31. The main area is titled 'SQL Commands' with a schema dropdown set to WKSP_ASHMITA31. The command history pane contains the three SELECT statements from the 'QUERY:' section. The results pane shows a table with two columns: 'JOB_ID' and 'DEPARTMENT_ID'. The data returned is:

JOB_ID	DEPARTMENT_ID
SL_REP	10
MK_MANAGER	10
HR_MANAGER	50
ST_CLERK	20

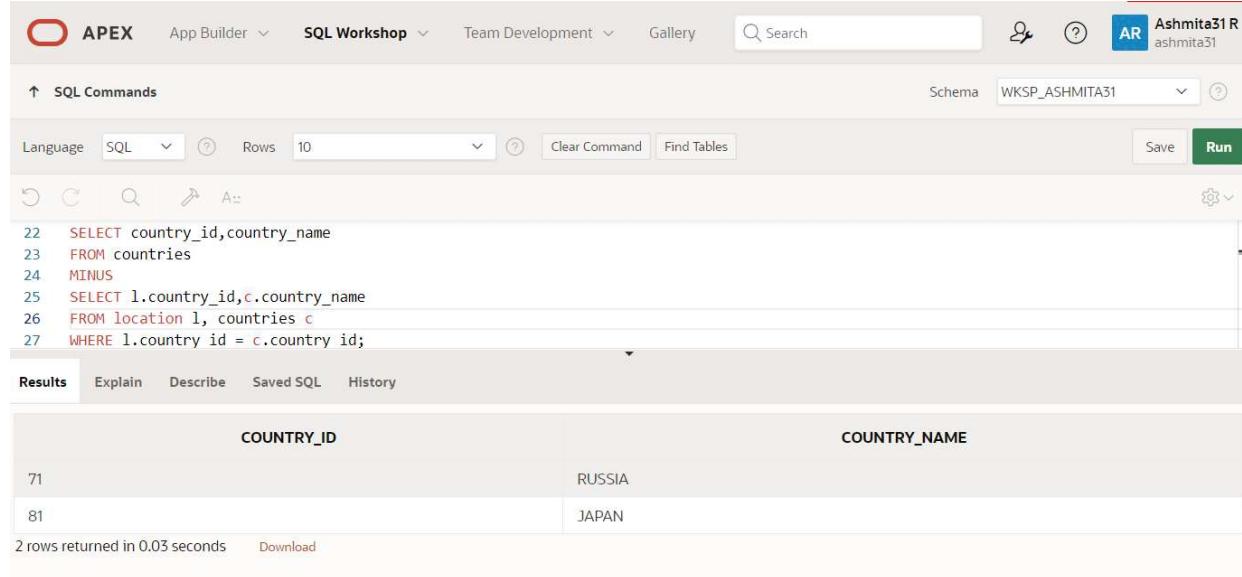
Below the table, it says '4 rows returned in 0.01 seconds' and there is a 'Download' link.

4. Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs but have now gone back to doing their original job).

QUERY:

```
SELECT country_id, country_name  
FROM countries  
MINUS  
SELECT l.country_id, c.country_name  
FROM location l, countries c  
WHERE l.country_id = c.country_id;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The right side shows the user 'Ashmita31 R' and the schema 'WKSP_ASHMITA31'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the query code. The Results tab shows the output in a table format.

COUNTRY_ID	COUNTRY_NAME
71	RUSSIA
81	JAPAN

2 rows returned in 0.03 seconds [Download](#)

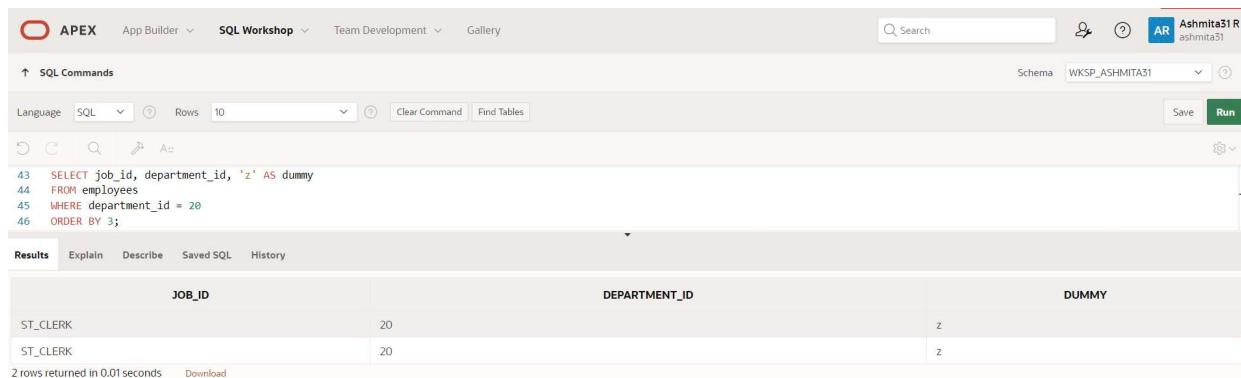
5. The HR department needs a report with the following specifications:

- Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department.
- Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them Write a compound query to accomplish this.

QUERY:

```
SELECT LAST_NAME,DEPARTMENT_ID,TO_CHAR(NULL) FROM EMPLOYEES  
UNION SELECT TO_CHAR(NULL),DEPT_ID,DEPT_NAME FROM DEPARTMENT;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL command entered is:

```
43  SELECT job_id, department_id, 'z' AS dummy  
44  FROM employees  
45  WHERE department_id = 20  
46  ORDER BY 3;
```

The results section displays the following data:

JOB_ID	DEPARTMENT_ID	DUMMY
ST_CLERK	20	z
ST_CLERK	20	z

2 rows returned in 0.01 seconds

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT :

CREATING VIEWS

EXP NO: 11

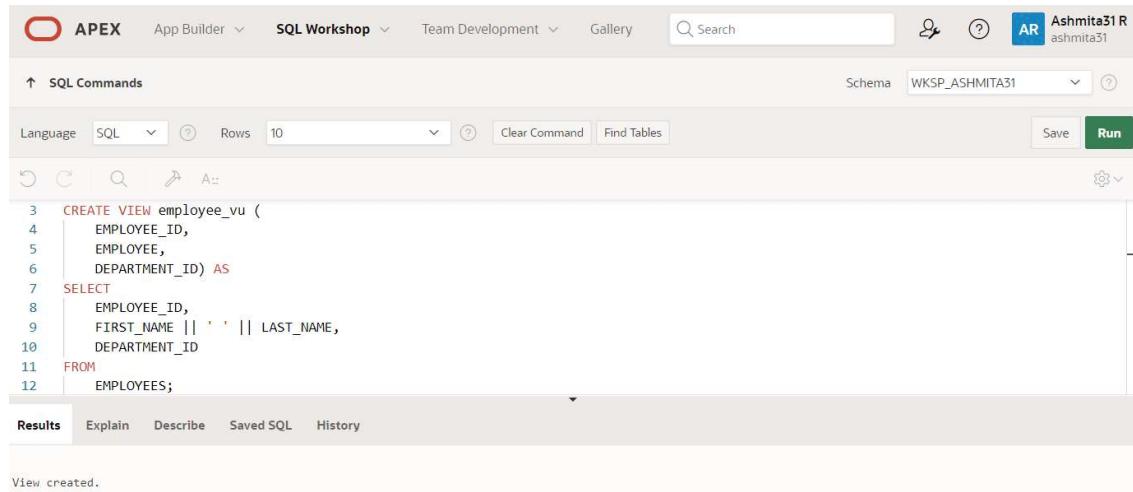
DATE:

-
1. Create a view called EMPLOYEE_VU based on the employee numbers, employee names and department numbers from the EMPLOYEES table. Change the heading for the employee name to EMPLOYEE.

QUERY:

```
CREATE VIEW employee_vu (EMPLOYEE_ID, EMPLOYEE, DEPARTMENT_ID) AS
SELECT EMPLOYEE_ID, FIRST_NAME || ' ' || LAST_NAME, DEPARTMENT_ID
FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The right side shows the user's profile: AR Ashmita31 R ashmita31. Below the toolbar, the schema is set to WKSP_ASHMITA31. The main area is titled "SQL Commands" and contains a code editor with the following SQL script:

```
3 CREATE VIEW employee_vu (
4   EMPLOYEE_ID,
5   EMPLOYEE,
6   DEPARTMENT_ID) AS
7 SELECT
8   EMPLOYEE_ID,
9   FIRST_NAME || ' ' || LAST_NAME,
10  DEPARTMENT_ID
11 FROM
12  EMPLOYEES;
```

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab shows the message "View created."

2. Display the contents of the EMPLOYEES_VU view.

QUERY:

```
SELECT * FROM EMPLOYEE_VU;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile 'Ashmita31 R' are also present. The main area displays a SQL command line with the query 'select*from employee_vu;'. The results tab is selected, showing a table with three columns: EMPLOYEE_ID, EMPLOYEE, and DEPARTMENT_ID. The data consists of ten rows of employee information. A message at the bottom indicates that more than 10 rows are available.

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
114	VJ SIDDHU	60
107	JIM JAM	80
109	AKAY KOHLI	80
103	SANJ PARTHI	20
104	MARY JANE	50
105	SANA EMANUEL	30
108	MIKA DEV	80
100	SAHANA JAY	10
113	KHAN HARSH	90
102	TARA UMA	20

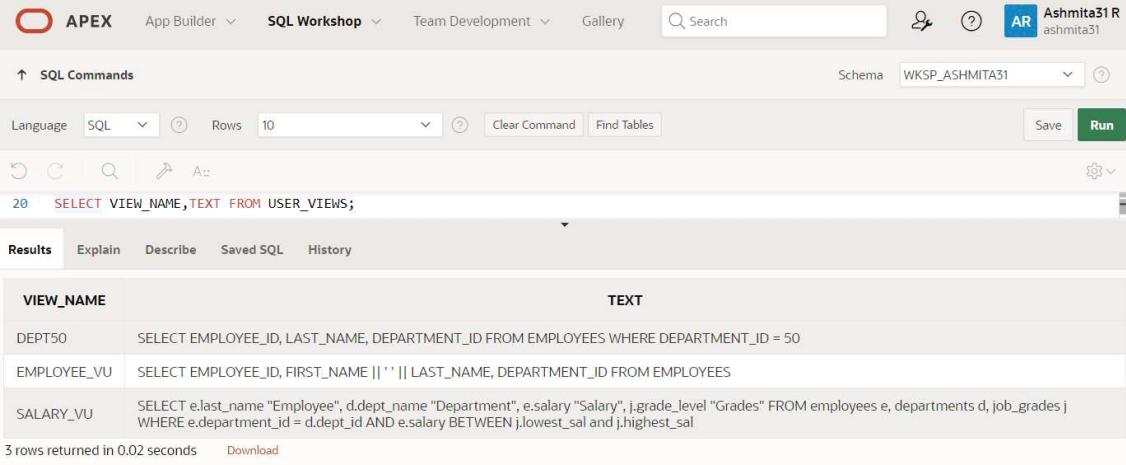
More than 10 rows available. Increase rows selector to view more rows.

3. Select the view name and text from the USER_VIEWS data dictionary views.

QUERY:

```
SELECT VIEW_NAME,TEXT FROM USER_VIEWS;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. There is a search bar and a user profile for Ashmita31R. The main workspace has a toolbar with various icons. The SQL Commands tab is active, showing the schema as WKSP_ASHMITA31. The query entered is:

```
20  SELECT VIEW_NAME,TEXT FROM USER_VIEWS;
```

The results section displays the output:

VIEW_NAME	TEXT
DEPT50	SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 50
EMPLOYEE_VU	SELECT EMPLOYEE_ID, FIRST_NAME ' ' LAST_NAME, DEPARTMENT_ID FROM EMPLOYEES
SALARY_VU	SELECT e.last_name "Employee", d.dept_name "Department", e.salary "Salary", j.grade_level "Grades" FROM employees e, departments d, job_grades j WHERE e.department_id = d.dept_id AND e.salary BETWEEN j.lowest_sal and j.highest_sal

Below the results, it says "3 rows returned in 0.02 seconds" and there is a "Download" link.

4. Using your EMPLOYEES_VU view, enter a query to display all employees names and department.

QUERY:

```
SELECT EMPLOYEE,DEPARTMENT_ID FROM employee_vu;
```

OUTPU

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user information for 'Ashmita31' are also present. The main area is titled 'SQL Commands' and contains the following query:

```
22  SELECT EMPLOYEE,DEPARTMENT_ID FROM employee_vu;
```

The results tab is selected, displaying the output of the query:

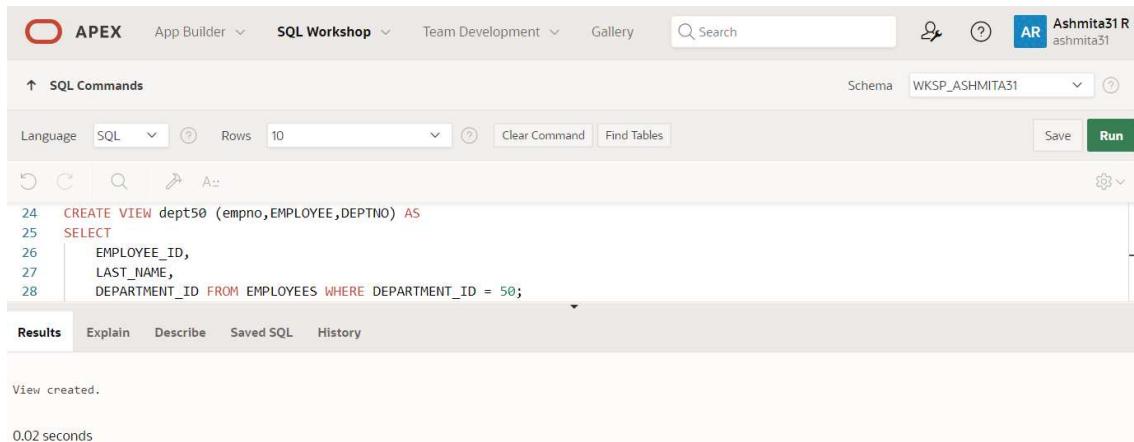
EMPLOYEE	DEPARTMENT_ID
VJ SIDDHU	60
JIM JAM	80
AKAY KOHLI	80
SANJ PARTHI	20
MARY JANE	50
SANA EMANUEL	30
MIKA DEV	80
SAHANA JAY	10
KHAN HARSH	90
TARA UMA	20

5. Create a view named DEPT50 that contains the employee number, employee last names and department numbers for all employees in department 50. Label the view columns EMPNO, EMPLOYEE and DEPTNO. Do not allow an employee to be reassigned to another department through the view.

QUERY:

```
CREATE VIEW dept50 (empno,EMPLOYEE,DEPTNO) AS  
SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID  
FROM EMPLOYEES WHERE DEPARTMENT_ID = 50;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', 'Gallery', a search bar, and a user profile 'Ashmita31 R'. The main area is titled 'SQL Commands'. The schema dropdown is set to 'WKSP_ASHMITA31'. The code editor contains the following SQL command:

```
24 CREATE VIEW dept50 (empno,EMPLOYEE,DEPTNO) AS  
25 SELECT  
26   EMPLOYEE_ID,  
27   LAST_NAME,  
28   DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 50;
```

The 'Results' tab is selected at the bottom. The output pane displays the message 'View created.' and '0.02 seconds'.

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

QUERY:

```
describe dept50;
```

```
SELECT*FROM dept50;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The command entered is 'describe dept50;'. The results pane displays the structure of the DEPT50 view, which contains three columns: EMPNO, EMPLOYEE, and DEPTNO. The DEPTNO column is defined as NUMBER(4,0) and is marked as nullable.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT50	EMPNO	NUMBER	-	6	0	-	-	-	-
	EMPLOYEE	VARCHAR2	25	-	-	-	-	-	-
	DEPTNO	NUMBER	-	4	0	-	✓	-	-

The screenshot shows the Oracle SQL Workshop interface. The command entered is 'SELECT*FROM dept50;'. The results pane displays one row from the DEPT50 view, with values 104 for EMPNO, JANE for EMPLOYEE, and 50 for DEPTNO.

EMPNO	EMPLOYEE	DEPTNO
104	JANE	50

1 rows returned in 0.05 seconds Download

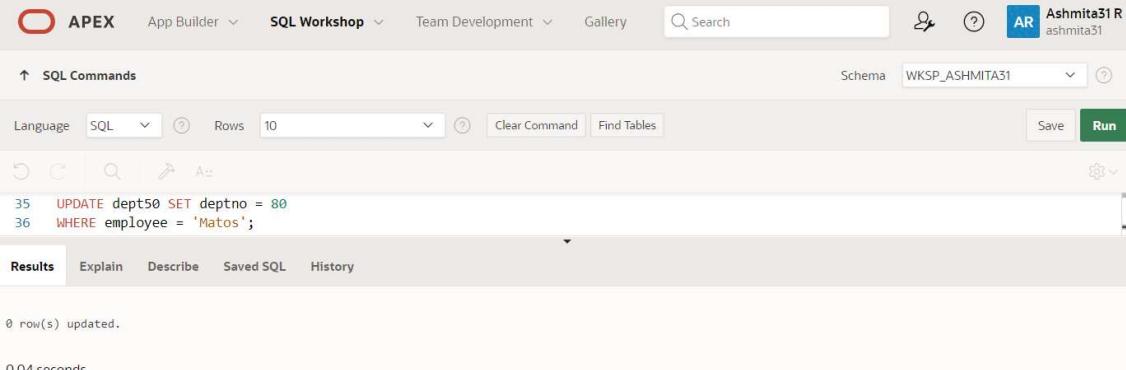
7. Attempt to reassign Matos to department 80.

QUERY:

```
UPDATE dept50 SET deptno = 80
```

```
WHERE employee = 'Matos';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. There is also a search bar and a user profile for 'Ashmita31R'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
35  UPDATE dept50 SET deptno = 80
36  WHERE employee = 'Matos';
```

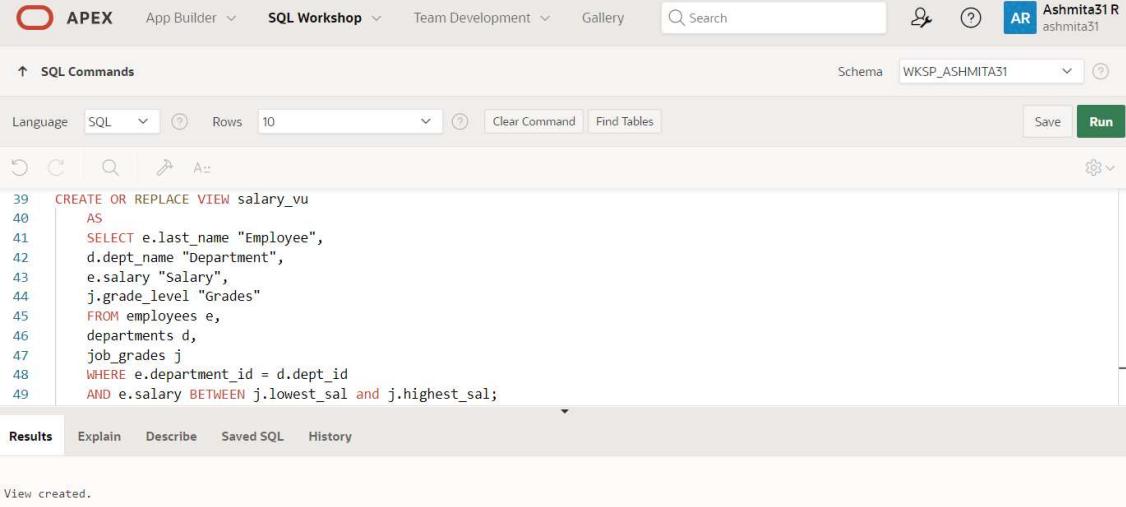
Below the code, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the output: '0 row(s) updated.' and '0.04 seconds'. The interface has a light gray background with various toolbars and status indicators.

8. 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_GRADE tables. Label the column Employee, Department, salary, and Grade respectively.

QUERY:

```
CREATE OR REPLACE VIEW salary_vu AS
SELECT e.last_name "Employee",d.dept_name "Department", e.salary "Salary",j.grade_level "Grades"
FROM employees e,departments d, job_grades j
WHERE e.department_id = d.dept_id
AND e.salary BETWEEN j.lowest_sal and j.highest_sal;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and a user profile for Ashmita31 R. The main area is titled 'SQL Commands' and shows the following SQL code:

```
39 CREATE OR REPLACE VIEW salary_vu
40 AS
41 SELECT e.last_name "Employee",
42       d.dept_name "Department",
43       e.salary "Salary",
44       j.grade_level "Grades"
45 FROM employees e,
46      departments d,
47      job_grades j
48 WHERE e.department_id = d.dept_id
49 AND e.salary BETWEEN j.lowest_sal and j.highest_sal;
```

The code is highlighted in blue and red, indicating syntax. Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, and it displays the message: "View created."

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

INTRO TO CONSTRAINTS: NOT NULL AND UNIQUE CONSTRAINTS

EX-NO : 12

DATE:

Global Fast Foods has been very successful this past year and has opened several new stores. They need to add a table to their database to store information about each of their store's locations. The owners want to make sure that all entries have an identification number, date opened, address, and city and that no other entry in the table can have the same email address. Based on this information, answer the following questions about the global_locations table. Use the table for your answers.

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
Id						
name						
date_opened						
address						
city						
zip/postal code						
phone						
email						
manager_id						
Emergency contact						

1. What is a “constraint” as it relates to data integrity?

Ans:

Database can be as reliable as the data in it, and database rules are implemented as Constraint to maintain data integrity.

2. What are the limitations of constraints that may be applied at the column level and at the table level?

Ans:

- ? Constraints referring to more than one column are defined at Table Level.
- ? NOT NULL constraint must be defined at column level as per ANSI/ISO SQL standard.

3. Why is it important to give meaningful names to constraints?

Ans:

- ? If a constraint is violated in a SQL statement execution, it is easy to identify the cause with user-named constraints.
- ? It is easy to alter names/drop constraint.

4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.

Ans:

Global Fast Foods global_locations Table						
NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE
id	pk	NUMBER	6	0		No
name		VARCHAR2	50			
date_opened		DATE				No
address		VARCHAR2	50			No
city		VARCHAR2	30			No
zip_postal_code		VARCHAR2	12			
phone		VARCHAR2	20			
email	uk	VARCHAR2	75			
manager_id		NUMBER	6	0		
emergency_contact		VARCHAR2	20			

5. Use “(nullable)” to indicate those columns that can have null values.

Ans:

Global Fast Foods global_locations Table						
NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE
id	pk	NUMBER	6	0		No
name		VARCHAR2	50			Yes
date_opened		DATE				No
address		VARCHAR2	50			No
city		VARCHAR2	30			No
zip_postal_code		VARCHAR2	12			Yes
phone		VARCHAR2	20			Yes
email	uk	VARCHAR2	75			Yes
manager_id		NUMBER	6	0		Yes
emergency_contact		VARCHAR2	20			Yes

6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level.

Ans:

```
CREATE TABLE f_global_locations
( id NUMBER(6,0) CONSTRAINT f_gln_id_pk PRIMARY KEY ,
name VARCHAR2(50),
date_opened DATE CONSTRAINT f_gln_dt_opened_nn NOT NULL ENABLE,
address VARCHAR2(50) CONSTRAINT f_gln_add_nn NOT NULL ENABLE,
city VARCHAR2(30) CONSTRAINT f_gln_city_nn NOT NULL ENABLE,
zip_postal_code VARCHAR2(12),
phone VARCHAR2(20),
email VARCHAR2(75) CONSTRAINT f_gln_email_uk UNIQUE,
manager_id NUMBER(6,0),
emergency_contact VARCHAR2(20)
);
```

7. Execute the CREATE TABLE statement in Oracle Application Express.

Ans:

Table Created.

8. Execute a DESCRIBE command to view the Table Summary information.

Ans:

DESCRIBE f_global_locations;

9. Rewrite the CREATE TABLE statement for the Global Fast Foods locations table to define the UNIQUE constraints at the table level. Do not execute this statement.

NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
id	number	4				
loc_name	varchar2	20			X	
	date					
address	varchar2	30				
city	varchar2	20				
zip_postal	varchar2	20			X	
phone	varchar2	15			X	
email	varchar2	80			X	
manager_id	number	4			X	
contact	varchar2	40			X	

Ans:

```
CREATE TABLE f_global_locations
( id NUMBER(6,0) CONSTRAINT f_gln_id_pk PRIMARY KEY ,
name VARCHAR2(50),
date_opened DATE CONSTRAINT f_gln_dt_opened_nn NOT NULL ENABLE,
```

```
address VARCHAR2(50) CONSTRAINT f_gln_add_nn NOT NULL ENABLE,
city VARCHAR2(30) CONSTRAINT f_gln_city_nn NOT NULL ENABLE,
zip_postal_code VARCHAR2(12),
phone VARCHAR2(20), email VARCHAR2(75) ,
manager_id NUMBER(6,0),
emergency_contact VARCHAR2(20),
CONSTRAINT f_gln_email_uk UNIQUE(email)
);
```

PRIMARY KEY, FOREIGN KEY, AND CHECK CONSTRAINTS

1. What is the purpose of a

- PRIMARY KEY
- FOREIGN KEY
- CHECK CONSTRAINT

Ans:

a. PRIMARY KEY

Uniquely identify each row in table.

b. FOREIGN KEY

Referential integrity constraint links back parent table's primary/unique key to child table's column.

c. CHECK CONSTRAINT

Explicitly define condition to be met by each row's fields. This condition must be returned as true or unknown.

2. Using the column information for the animals table below, name constraints where applicable at the table level, otherwise name them at the column level. Define the primary key (animal_id). The license_tag_number must be unique. The admit_date and vaccination_date columns cannot contain null values.

Ans:

animal_id NUMBER(6) - **PRIMARY KEY**
name VARCHAR2(25)
license_tag_number NUMBER(10) - **UNIQUE**
admit_date DATE -**NOT NULL**
adoption_id NUMBER(5),
vaccination_date DATE -**NOT NULL**

3. Create the animals table. Write the syntax you will use to create the table.

Ans:

```
CREATE TABLE animals
( animal_id NUMBER(6,0) CONSTRAINT anl_anl_id_pk PRIMARY KEY ,
name VARCHAR2(25),
license_tag_number NUMBER(10,0) CONSTRAINT anl_l_tag_num_uk
UNIQUE,
admit_date DATE CONSTRAINT anl_adt_dat_nn NOT NULL ENABLE,
adoption_id NUMBER(5,0),
vaccination_date DATE CONSTRAINT anl_vcc_dat_nn NOT NULL ENABLE
);
```

4. Enter one row into the table. Execute a SELECT * statement to verify your input. Refer to the graphic below for input.

ANIMAL_ID	NAME	LICENSE_TAG_NUMBER	ADMIT_DATE	ADOPTION_ID	VACCINATION_DATE
-----------	------	--------------------	------------	-------------	------------------

101	Spot	35540	10-Oct-2004	205	12-Oct-2004
-----	------	-------	-------------	-----	-------------

Ans:

```
INSERT INTO animals (animal_id, name, license_tag_number, admit_date, adoption_id, vaccination_date) VALUES( 101, 'Spot', 35540, TO_DATE('10-Oct-2004', 'DD-Mon-YYYY'), 205, TO_DATE('12-Oct-2004', 'DD-Mon-YYYY'));
```

```
SELECT * FROM animals;
```

5. Write the syntax to create a foreign key (adoption_id) in the animals table that has a corresponding primary-key reference in the adoptions table. Show both the column-level and table-level syntax. Note that because you have not actually created an adoptions table, no adoption_id primary key exists, so the foreign key cannot be added to the animals table.

Ans:

COLUMN LEVEL STATEMENT:

```
ALTER TABLE animals MODIFY ( adoption_id NUMBER(5,0) CONSTRAINT anl_adopt_id_fk REFERENCES adoptions(id) ENABLE );
```

TABLE LEVEL STATEMENT:

```
ALTER TABLE animals ADD CONSTRAINT anl_adopt_id_fk FOREIGN KEY (adoption_id) REFERENCES adoptions(id) ENABLE;
```

6. What is the effect of setting the foreign key in the ANIMAL table as:

a. ON DELETE CASCADE

```
ALTER TABLE animals ADD CONSTRAINT anl_adopt_id_fk FOREIGN KEY (adoption_id) REFERENCES adoptions(id) ON DELETE CASCADE ENABLE ;
```

b. ON DELETE SET NULL

```
ALTER TABLE animals ADD CONSTRAINT anl_adopt_id_fk FOREIGN KEY (adoption_id) REFERENCES adoptions(id) ON DELETE SET NULL ENABLE ;
```

7. What are the restrictions on defining a CHECK constraint?

Ans:

- ? I cannot specify check constraint for a view however in this case I could use WITH CHECK OPTION clause
- ? I am restricted to columns from self table and fields in self row.
- ? I cannot use subqueries and scalar subquery expressions.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT :

EXERCISE 13

Creating Views

1. What are three uses for a view from a DBA's perspective?

- ❖ **Restrict access and display selective columns**
- ❖ **Reduce complexity of queries from other internal systems. So, providing a way to view same data in a different manner.**
- ❖ **Let the app code rely on views and allow the internal implementation of tables to be modified later.**

2. Create a simple view called view_d_songs that contains the ID, title and artist from the DJs on Demand table for each "New Age" type code. In the subquery, use the alias "Song Title" for the title column.

```
CREATE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

3. SELECT * FROM view_d_songs. What was returned?

Results	Explain	Describe	Saved SQL	History									
<table border="1"><thead><tr><th>ID</th><th>Song Title</th><th>ARTIST</th></tr></thead><tbody><tr><td>47</td><td>Hurrah for Today</td><td>The Jubilant Trio</td></tr><tr><td>49</td><td>Lets Celebrate</td><td>The Celebrants</td></tr></tbody></table>					ID	Song Title	ARTIST	47	Hurrah for Today	The Jubilant Trio	49	Lets Celebrate	The Celebrants
ID	Song Title	ARTIST											
47	Hurrah for Today	The Jubilant Trio											
49	Lets Celebrate	The Celebrants											
2 rows returned in 0.00 seconds				Download									
2 rows returned in 0.00 seconds Download													

4. REPLACE view_d_songs. Add type_code to the column list. Use aliases for all columns. Or use alias after the CREATE statement as shown.

```
CREATE OR REPLACE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist, d_songs.type_code
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

5. Jason Tsang, the disk jockey for DJs on Demand, needs a list of the past events and those planned for the coming months so he can make arrangements for each event's equipment setup. As the company manager, you do not want him to have access to the price that clients paid for their events. Create a view for Jason to use that displays the name of the event, the event date, and the theme description. Use aliases for each column name.

```
CREATE OR REPLACE VIEW view_d_events_pkgs AS
SELECT evt.name "Name of Event", TO_CHAR(evt.event_date, 'dd-Month-yyyy') "Event date",
thm.description "Theme description"
FROM d_events evt INNER JOIN d_themes thm ON evt.theme_code = thm.code
WHERE evt.event_date <= ADD_MONTHS(SYSDATE,1);
```

6. It is company policy that only upper-level management be allowed access to individual employee salaries. The department managers, however, need to know the minimum, maximum, and average salaries, grouped by department. Use the Oracle database to prepare a view that displays the needed information for department managers.

```
CREATE OR REPLACE VIEW view_min_max_avg_dpt_salary ("Department Id", "Department Name",
"Max Salary", "Min Salary", "Average Salary") AS
SELECT dpt.department_id, dpt.department_name, MAX(NVL(emp.salary,0)),
MIN(NVL(emp.salary,0)), ROUND(AVG(NVL(emp.salary,0)),2)
FROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id =
emp.department_id
GROUP BY (dpt.department_id, dpt.department_name);
```

DML Operations and Views

Use the DESCRIBE statement to verify that you have tables named copy_d_songs, copy_d_events, copy_d_cds, and copy_d_clients in your schema. If you don't, write a query to create a copy of each.

1. Query the data dictionary USER_UPDATABLE_COLUMNS to make sure the columns in the base tables will allow UPDATE, INSERT, or DELETE. All table names in the data dictionary are stored in uppercase.

```
SELECT owner,table_name,column_name,updatable,insertable,deletable  
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_songs';
```

```
SELECT owner,table_name,column_name,updatable,insertable,deletable  
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_events';
```

```
SELECT owner,table_name,column_name,updatable,insertable,deletable  
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_cds';
```

2. Use the CREATE or REPLACE option to create a view of *all* the columns in the copy_d_songs table called view_copy_d_songs.

```
CREATE OR REPLACE VIEW view_copy_d_songs AS  
SELECT *  
FROM copy_d_songs;
```

```
SELECT * FROM view_copy_d_songs;
```

3. Use view_copy_d_songs to INSERT the following data into the underlying copy_d_songs table. Execute a SELECT * from copy_d_songs to verify your DML command. See the graphic.

ID	TITLE	DURATION	ARTIST	TYPE_CODE
88	Mello Jello	2	The What	4

```
INSERT INTO view_copy_d_songs(id,title,duration,artist,type_code)  
VALUES(88,'Mello Jello','2 min','The What',4);
```

4. Create a view based on the DJs on Demand COPY_D_CDS table. Name the view read_copy_d_cds. Select all columns to be included in the view. Add a WHERE clause to restrict the year to 2000. Add the WITH READ ONLY option.

```
CREATE OR REPLACE VIEW read_copy_d_cds AS  
SELECT *  
FROM copy_d_cds  
WHERE year = '2000'  
WITH READ ONLY ;
```

```
SELECT * FROM read_copy_d_cds;
```

5. Using the read_copy_d_cds view, execute a DELETE FROM read_copy_d_cds WHERE cd_number = 90;

ORA-42399: cannot perform a DML operation on a read-only view

6. Use REPLACE to modify read_copy_d_cds. Replace the READ ONLY option with WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds. Execute a SELECT * statement to verify that the view exists.

```
CREATE OR REPLACE VIEW read_copy_d_cds AS
```

```
SELECT *  
FROM copy_d_cds  
WHERE year = '2000'  
WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds;
```

7. Use the read_copy_d_cds view to delete any CD of year 2000 from the underlying copy_d_cds.

```
DELETE FROM read_copy_d_cds  
WHERE year = '2000';
```

8. Use the read_copy_d_cds view to delete cd_number 90 from the underlying copy_d_cds table.

```
DELETE FROM read_copy_d_cds  
WHERE cd_number = 90;
```

9. Use the read_copy_d_cds view to delete year 2001 records.

```
DELETE FROM read_copy_d_cds  
WHERE year = '2001';
```

10. Execute a SELECT * statement for the base table copy_d_cds. What rows were deleted?

Only the one in problem 7 above, not the one in 8 and 9

11. What are the restrictions on modifying data through a view?

DELETE,INSERT,MODIFY restricted if it contains:

Group functions
GROUP BY CLAUSE
DISTINCT
pseudocolumn ROWNUM Keyword

12. What is Moore's Law? Do you consider that it will continue to apply indefinitely? Support your opinion with research from the internet.

It roughly predicted that computing power nearly doubles every year. But Moore also said in 2005 that as per nature of exponential functions, this trend may not continue forever.

13. What is the "singularity" in terms of computing?

Singularity is the hypothesis that the invention of artificial superintelligence will abruptly trigger runaway technological growth, resulting in unfathomable changes to human civilization

Managing Views

1. Create a view from the copy_d_songs table called view_copy_d_songs that includes only the title and artist. Execute a SELECT * statement to verify that the view exists.

```
CREATE OR REPLACE VIEW view_copy_d_songs AS  
SELECT title,artist  
FROM copy_d_songs;
```

```
SELECT * FROM view_copy_d_songs;
```

2. Issue a DROP view_copy_d_songs. Execute a SELECT * statement to verify that the view has been deleted.

```
DROP VIEW view_copy_d_songs;  
SELECT * FROM view_copy_d_songs;
```

ORA-00942: table or view does not exist

3. Create a query that selects the last name and salary from the Oracle database. Rank the salaries from highest to lowest for the top three employees.

```
SELECT * FROM  
(SELECT last_name,salary FROM employees ORDER BY salary DESC)  
WHERE ROWNUM <= 3;
```

4. Construct an inline view from the Oracle database that lists the last name, salary, department ID, and maximum salary for each department. Hint: One query will need to calculate maximum salary by department ID.

```
SELECT empm.last_name,empm.salary,dptmx.department_id  
FROM  
(SELECT dpt.department_id,MAX(NVL(emp.salary,0)) max_dpt_sal  
FROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id =  
emp.department_id  
GROUP BY dpt.department_id) dptmx LEFT OUTER JOIN employees empm ON dptmx.department_id =  
empm.department_id  
WHERE NVL(empm.salary,0) = dptmx.max_dpt_sal;
```

5. Create a query that will return the staff members of Global Fast Foods ranked by salary from lowest to highest.

```
SELECT ROWNUM,last_name,salary  
FROM  
(SELECT * FROM f_staffs ORDER BY SALARY);
```

Indexes and Synonyms

1. What is an index and what is it used for?

Definition: These are schema objects which make retrieval of rows from table faster.

Purpose: An index provides direct and fast access to row in table. They provide indexed path to locate data quickly, so hereby reduce necessity of heavy disk input/output operations.

2. What is a ROWID, and how is it used?

Indexes use ROWID's (base 64 string representation of the row address containing block identifier, row location in the block and the database file identifier) which is the fastest way to access any particular row.

3. When will an index be created automatically?

Primary key/unique key use already existing unique index but if index is not present already, it is created while applying unique/primary key constraint.

4. Create a nonunique index (foreign key) for the DJs on Demand column (cd_number) in the D_TRACK_LISTINGS table. Use the Oracle Application Express SQL Workshop Data Browser to confirm that the index was created.

```
CREATE INDEX d_tlg_cd_number_fk_i  
on d_track_listings (cd_number);
```

5. Use the join statement to display the indexes and uniqueness that exist in the data dictionary for the DJs on Demand D_SONGS table.

```
SELECT ucm.index_name, ucm.column_name, ucm.column_position, uix.uniqueness  
FROM user_indexes uix INNER JOIN user_ind_columns ucm ON uix.index_name = ucm.index_name  
WHERE ucm.table_name = 'D_SONGS';
```

6. Use a SELECT statement to display the index_name, table_name, and uniqueness from the data dictionary USER_INDEXES for the DJs on Demand D_EVENTS table.

```
SELECT index_name, table_name,uniqueness FROM user_indexes where table_name = 'D_EVENTS';
```

7. Write a query to create a synonym called dj_tracks for the DJs on Demand d_track_listings table.

CREATE SYNONYM dj_tracks FOR d_track_listings;

8. Create a function-based index for the last_name column in DJs on Demand D_PARTNERS table that makes it possible not to have to capitalize the table name for searches. Write a SELECT statement that would use this index.

**CREATE INDEX d_ptr_last_name_idx
ON d_partners(LOWER(last_name));**

9. Create a synonym for the D_TRACK_LISTINGS table. Confirm that it has been created by querying the data dictionary.

CREATE SYNONYM dj_tracks2 FOR d_track_listings;

SELECT * FROM user_synonyms WHERE table_NAME = UPPER('d_track_listings');

10. Drop the synonym that you created in question

DROP SYNONYM dj_tracks2;

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

OTHER DATABASE OBJECTS

EX_NO:14

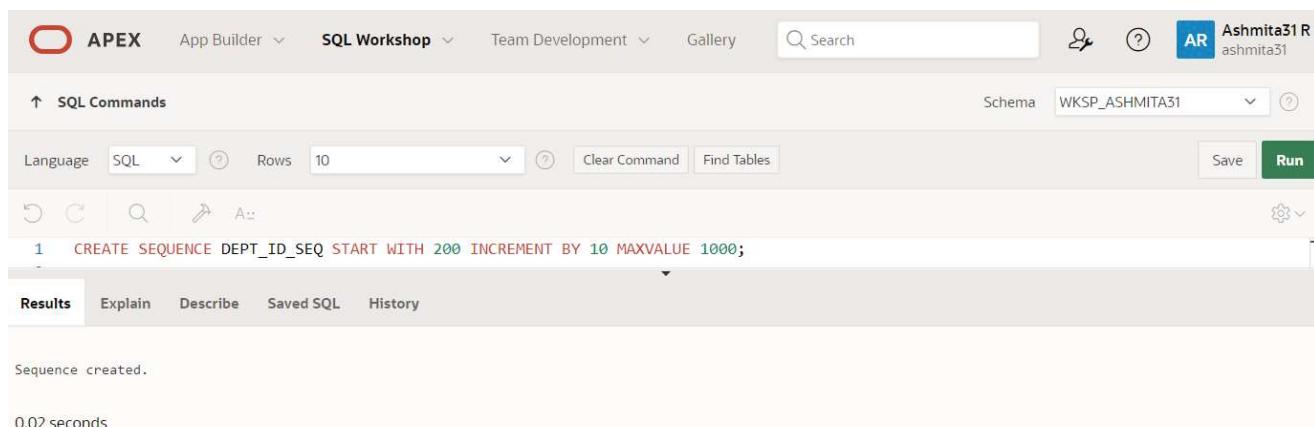
DATE:

1.)Create a sequence to be used with the primary key column of the DEPT table. The sequence should start at 200 and have a maximum value of 1000. Have your sequence increment by ten numbers. Name the sequence DEPT_ID_SEQ

QUERY:

```
CREATE SEQUENCE dept_id_seq START WITH 200 INCREMENT BY 10 MAXVALUE 1000;
```

OUTPUT:



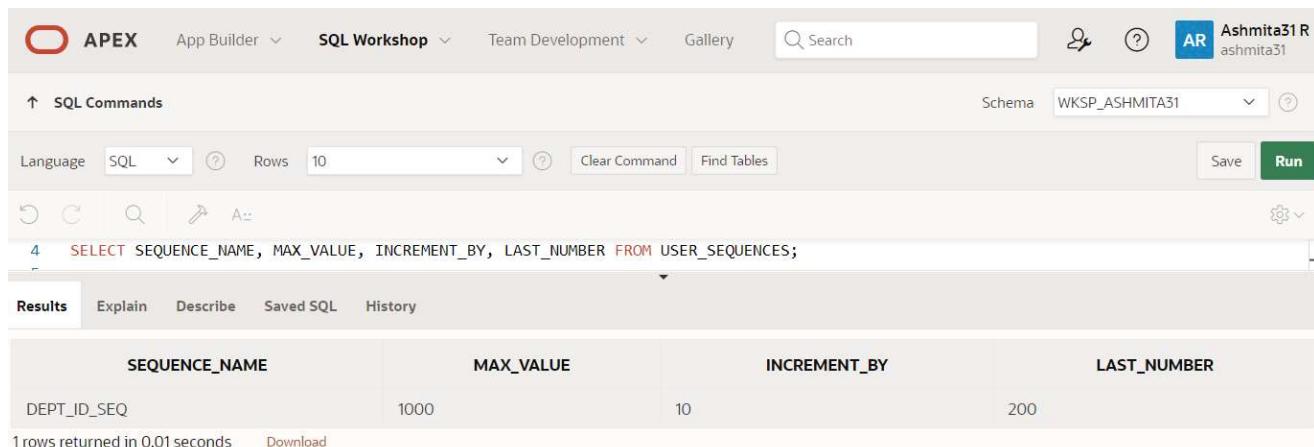
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', 'Gallery', a search bar, and user information 'Ashmita31 R ashmita31'. Below the navigation is a toolbar with icons for SQL, Clear Command, Find Tables, Save, and Run. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_ASHMITA31'. The SQL editor contains the command: '1 CREATE SEQUENCE DEPT_ID_SEQ START WITH 200 INCREMENT BY 10 MAXVALUE 1000;'. The results tab shows the output: 'Sequence created.' and a execution time of '0.02 seconds'.

2.)Write a query in a script to display the following information about your sequences: sequence name, maximum value, increment size, and last number

QUERY:

```
SELECT sequence_name, max_value, increment_by, last_number FROM user_sequences;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface with the 'SQL Workshop' tab selected. The SQL editor contains the command: '4 SELECT sequence_name, max_value, increment_by, last_number FROM user_sequences;'. The results tab displays a table with four columns: 'SEQUENCE_NAME', 'MAX_VALUE', 'INCREMENT_BY', and 'LAST_NUMBER'. A single row is shown: 'DEPT_ID_SEQ' with values '1000', '10', and '200'. The message '1 rows returned in 0.01 seconds' is at the bottom.

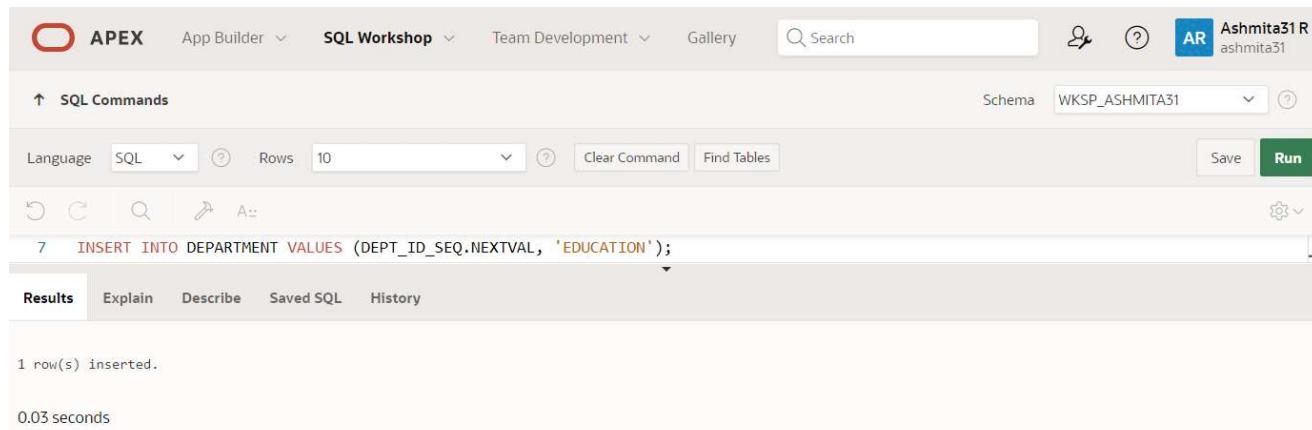
SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_ID_SEQ	1000	10	200

3.) Write a script to insert two rows into the DEPT table. Name your script lab12_3.sql. Be sure to use the sequence that you created for the ID column. Add two departments named Education and Administration. Confirm your additions. Run the commands in your script.

QUERY:

```
INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education');
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user information (Ashmita31 R) are also present. The main area is titled "SQL Commands". The schema is set to "WKSP_ASHMITA31". The SQL command entered is:

```
7  INSERT INTO DEPARTMENT VALUES (DEPT_ID_SEQ.NEXTVAL, 'EDUCATION');
```

The results section shows the output:

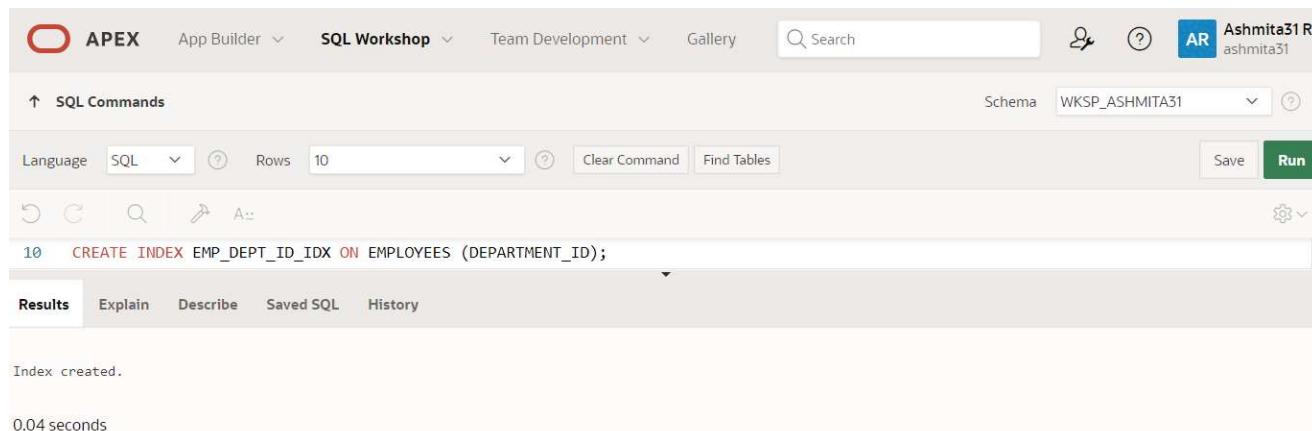
```
1 row(s) inserted.  
0.03 seconds
```

4.) Create a nonunique index on the foreign key column (DEPT_ID) in the EMP table.

QUERY:

```
CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user information (Ashmita31 R) are also present. The main area is titled "SQL Commands". The schema is set to "WKSP_ASHMITA31". The SQL command entered is:

```
10  CREATE INDEX EMP_DEPT_ID_IDX ON EMPLOYEES (DEPARTMENT_ID);
```

The results section shows the output:

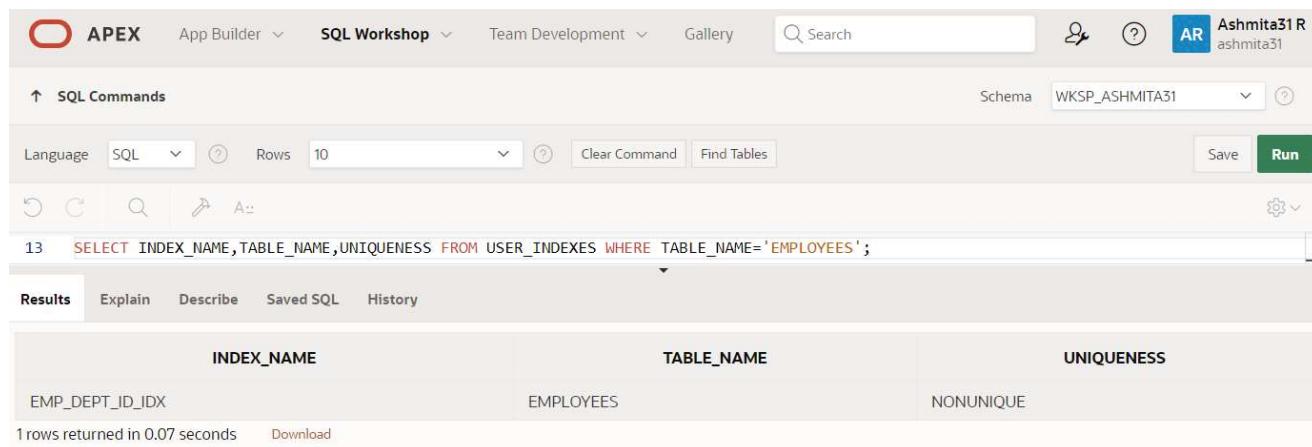
```
Index created.  
0.04 seconds
```

5.)Display the indexes and uniqueness that exist in the data dictionary for the EMP table.

QUERY:

```
SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, Gallery, a search bar, and a user profile for Ashmita31. The SQL Workshop tab is selected. The main area shows a SQL command line with the following content:

```
13  SELECT INDEX_NAME, TABLE_NAME,UNIQUENESS FROM USER_INDEXES WHERE TABLE_NAME='EMPLOYEES';
```

Below the command line, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, displaying the following table:

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

At the bottom left, it says "1 rows returned in 0.07 seconds".

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

CONTROLLING USER ACCESS

EX_NO:15

DATE:

1. What privilege should a user be given to log on to the Oracle Server? Is this a system or an object privilege?

The CREATE SESSION system privilege

2. What privilege should a user be given to create tables?

The CREATE TABLE privilege

3. If you create a table, who can pass along privileges to other users on your table?

You can, or anyone you have given those privileges to by using the WITH GRANT OPTION.

4. You are the DBA. You are creating many users who require the same system privileges. What should you use to make your job easier?

Create a role containing the system privileges and grant the role to the users

5. What command do you use to change your password?

The ALTER USER statement

6. Grant another user access to your DEPARTMENTS table. Have the user grant you query access to his or her DEPARTMENTS table.

Team 2 executes the GRANT statement. GRANT select ON departments TO <user1>;

Team 1 executes the GRANT statement. GRANT select ON departments TO <user2>;

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

8. Add a new row to your DEPARTMENTS table. Team 1 should add Education as department number 500. Team 2 should add Human Resources department number 510. Query the other team's table.

Team 1 executes this INSERT statement. INSERT INTO departments(department_id, department_name) VALUES (500, 'Education'); COMMIT;

Team 2 executes this INSERT statement. INSERT INTO departments(department_id, department_name) VALUES (510, 'Administration'); COMMIT;

9. Query the USER_TABLES data dictionary to see information about the tables that you own.

SELECT table_name FROM user_tables;

10. Revoke the SELECT privilege on your table from the other team.

Team 1 revokes the privilege.

```
REVOKE select  
ON departments  
FROM user2;
```

Team 2 revokes the privilege.

```
REVOKE select  
ON departments  
FROM user1;
```

11. Remove the row you inserted into the DEPARTMENTS table in step 8 and save the changes.

Team 1 executes this INSERT statement.

```
DELETE FROM departments  
WHERE department_id = 500;  
COMMIT;
```

Team 2 executes this INSERT statement.

```
DELETE FROM departments  
WHERE department_id = 510;  
COMMIT;
```

<u>Evaluation Procedure</u>	<u>Marks awarded</u>
<u>Practice Evaluation (5)</u>	
<u>Viva(5)</u>	
<u>Total (10)</u>	
<u>Faculty Signature</u>	

RESULT:

PL/SQL

CONTROL STRUCTURES

EX_NO:

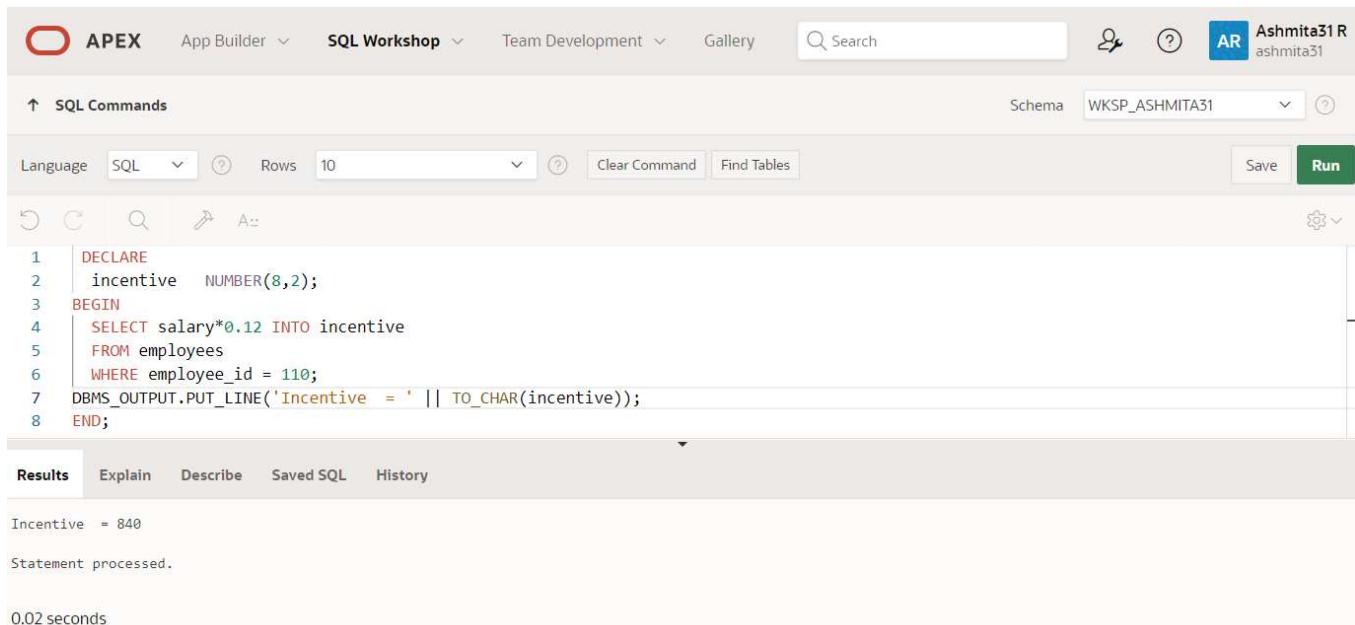
DATE:

1.) Write a PL/SQL block to calculate the incentive of an employee whose ID is 110.

QUERY:

```
DECLARE
incentive NUMBER(8,2);
BEGIN
SELECT salary*0.12 INTO incentive
FROM employees
WHERE employee_id = 110;
DBMS_OUTPUT.PUT_LINE('Incentive = ' || TO_CHAR(incentive));
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The user is signed in as Ashmita31R (ashmita31). The schema is set to WKSP_ASHMITA31. The main area is titled 'SQL Commands' with options for Language (SQL selected), Rows (10), Clear Command, Find Tables, Save, and Run. The code editor contains the provided PL/SQL block. The results tab is active, displaying the output: 'Incentive = 840' and 'Statement processed.' Below the results, it says '0.02 seconds'.

```
1 DECLARE
2     incentive  NUMBER(8,2);
3 BEGIN
4     SELECT salary*0.12 INTO incentive
5     FROM employees
6     WHERE employee_id = 110;
7     DBMS_OUTPUT.PUT_LINE('Incentive = ' || TO_CHAR(incentive));
8 END;
```

Incentive = 840
Statement processed.
0.02 seconds

2.) Write a PL/SQL block to show an invalid case-insensitive reference to a quoted and without quoted user-defined identifier

QUERY:

```
DECLARE
WELCOME varchar2(10) := 'welcome';
BEGIN
DBMS_Output.Put_Line("Welcome");
END;
/
```

```
DECLARE
WELCOME varchar2(10) := 'welcome';
BEGIN
DBMS_Output.Put_Line("Welcome");
END;
/
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The right side shows the user profile 'Ashmita31 R' and the schema 'WKSP_ASHMITA31'. The main area is titled 'SQL Commands' and contains a code editor with the following PL/SQL block:

```
15  DECLARE
16  | "WELCOME" varchar2(10) := 'welcome'; -- identifier with quotation
17  BEGIN
18  | DBMS_Output.Put_Line(Welcome); --reference to the identifier without quotation
19  END;
```

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing the output:

```
welcome
Statement processed.
```

Execution time is listed as 0.00 seconds.

3.) Write a PL/SQL block to adjust the salary of the employee whose ID 122.

QUERY:

```
DECLARE
    salary_of_emp NUMBER(8,2);
PROCEDURE approx_salary (
    emp      NUMBER,
    empsal IN OUT NUMBER,
    addless  NUMBER
) IS
BEGIN
    empsal := empsal + addless;
END;

BEGIN
    SELECT salary INTO salary_of_emp
    FROM employees
    WHERE employee_id = 122;
    DBMS_OUTPUT.PUT_LINE
        ('Before invoking procedure, salary_of_emp: ' || salary_of_emp);
    approx_salary (100, salary_of_emp, 1000);
    DBMS_OUTPUT.PUT_LINE
        ('After invoking procedure, salary_of_emp: ' || salary_of_emp);
END;
/
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

25 DECLARE
26 | v_employee_id NUMBER := 122;
27 | v_new_salary NUMBER;
28 BEGIN
29 | -- Retrieve current salary
30 | SELECT salary INTO v_new_salary
31 | FROM employees
32 | WHERE employee_id = v_employee_id;
33 | -- Adjust the salary by 10%
34 | v_new_salary := v_new_salary * 1.1;
35 | -- Update the salary
36 | UPDATE employees
37 | SET salary = v_new_salary
38 | WHERE employee_id = v_employee_id;
39 | -- Commit the transaction
40 | COMMIT;
41 | -- Display success message
42 | DBMS_OUTPUT.PUT_LINE('Salary of employee ' || v_employee_id || ' has been adjusted.');

43 EXCEPTION
44 WHEN NO_DATA_FOUND THEN
45 | DBMS_OUTPUT.PUT_LINE('Employee with ID ' || v_employee_id || ' not found.');

Results Explain Describe Saved SQL History

Employee with ID 122 not found.

1 row(s) updated.

4.) Write a PL/SQL block to create a procedure using the "IS [NOT] NULL Operator" and show AND operator returns TRUE if and only if both operands are TRUE.

QUERY:

```
CREATE OR REPLACE PROCEDURE pri_bool(
  boo_name  VARCHAR2,
  boo_val   BOOLEAN
) IS
BEGIN
  IF boo_val IS NULL THEN
    DBMS_OUTPUT.PUT_LINE( boo_name || ' = NULL');
  ELSIF boo_val = TRUE THEN
    DBMS_OUTPUT.PUT_LINE( boo_name || ' = TRUE');
  ELSE
    DBMS_OUTPUT.PUT_LINE( boo_name || ' = FALSE');
  END IF;
END;
/
```

OUTPUT:

↑ SQL Commands

Schema

WKSP_ASHMITA31

Save

Run



```
24
55
56 CREATE OR REPLACE PROCEDURE pri_bool(
57   boo_name    VARCHAR2,
58   boo_val     BOOLEAN
59 ) IS
60 BEGIN
61   IF boo_val IS NULL THEN
62     DBMS_OUTPUT.PUT_LINE( boo_name || ' = NULL');
63   ELSIF boo_val = TRUE THEN
64     DBMS_OUTPUT.PUT_LINE( boo_name || ' = TRUE');
65   ELSE
66     DBMS_OUTPUT.PUT_LINE( boo_name || ' = FALSE');
67   END IF;
68 END;
69 DECLARE
70   PROCEDURE pri_not_m (
71     m      BOOLEAN
72   ) IS
73   BEGIN
74     pri_bool ('m', m);
75     pri_bool ('NOT m', NOT m);
76   END pri_not_m;
77
78 BEGIN
79   DBMS_OUTPUT.PUT_LINE('----- FOR m TRUE -----');
80   pri_not_m (TRUE);
81   DBMS_OUTPUT.PUT_LINE('----- FOR m FALSE -----');
82   pri_not_m (FALSE);
83   DBMS_OUTPUT.PUT_LINE('----- FOR m NULL -----');
84   pri_not_m (NULL);
85 END;
```

Results Explain Describe Saved SQL History

```
----- FOR m TRUE -----
m = TRUE
NOT m = FALSE
----- FOR m FALSE -----
m = FALSE
NOT m = TRUE
----- FOR m NULL -----
m = NULL
NOT m = NULL
```

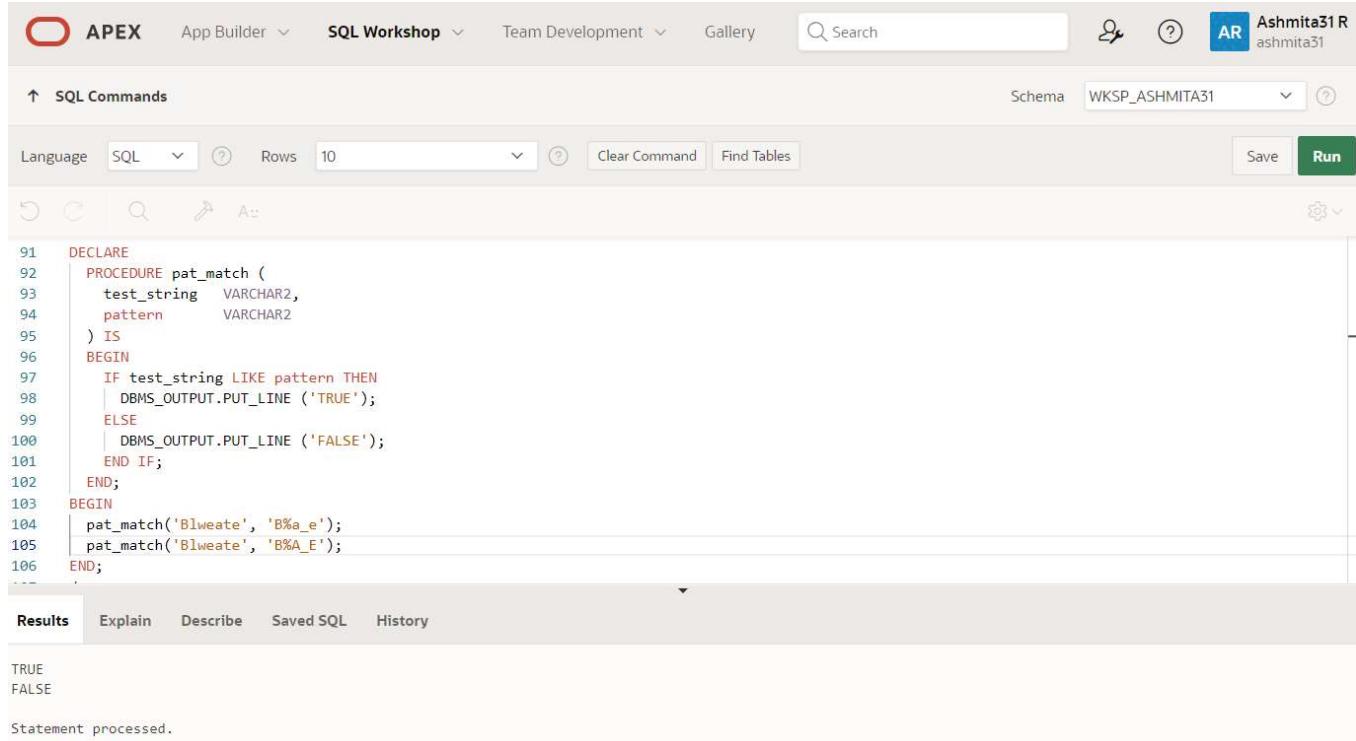
Statement processed.

5.) Write a PL/SQL block to describe the usage of LIKE operator including wildcard characters and escape character.

QUERY:

```
DECLARE
  PROCEDURE pat_match (
    test_string  VARCHAR2,
    pattern      VARCHAR2
  ) IS
BEGIN
  IF test_string LIKE pattern THEN
    DBMS_OUTPUT.PUT_LINE ('TRUE');
  ELSE
    DBMS_OUTPUT.PUT_LINE ('FALSE');
  END IF;
END;
/
BEGIN
  pat_match('Blweate', 'B%a_e');
  pat_match('Blweate', 'B%A_E');
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information for Ashmita31. The main area is titled 'SQL Commands' and shows the PL/SQL code from the previous block. The code is numbered from 91 to 106. The 'Results' tab at the bottom is selected, displaying the output: 'TRUE' and 'FALSE'. Below the results, a message says 'Statement processed.'

6.) Write a PL/SQL program to arrange the number of two variable in such a way that the small number will store in num_small variable and large number will store in num_large variable

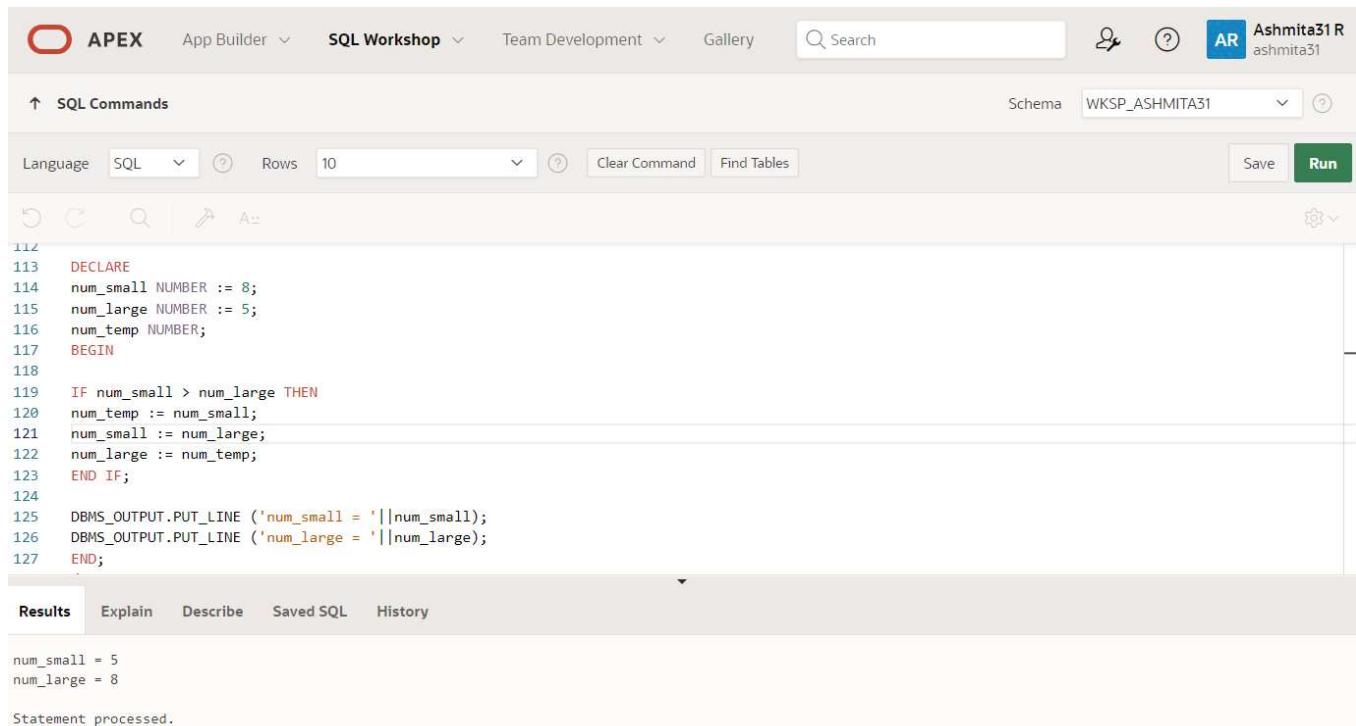
QUERY:

```
DECLARE
num_small NUMBER := 8;
num_large NUMBER := 5;
num_temp NUMBER;
BEGIN

IF num_small > num_large THEN
num_temp := num_small;
num_small := num_large;
num_large := num_temp;
END IF;

DBMS_OUTPUT.PUT_LINE ('num_small = '||num_small);
DBMS_OUTPUT.PUT_LINE ('num_large = '||num_large);
END;
/
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information (AR, Ashmita31). The main workspace is titled 'SQL Commands' and shows the executed PL/SQL code. The code declares variables num_small and num_large, initializes them to 8 and 5 respectively, swaps their values if num_small is greater than num_large, and then prints both values using DBMS_OUTPUT.PUT_LINE. The output pane at the bottom displays the results: num_small = 5 and num_large = 8, followed by a message stating 'Statement processed.'

```
112
113  DECLARE
114    num_small NUMBER := 8;
115    num_large NUMBER := 5;
116    num_temp NUMBER;
117    BEGIN
118
119    IF num_small > num_large THEN
120      num_temp := num_small;
121      num_small := num_large;
122      num_large := num_temp;
123    END IF;
124
125    DBMS_OUTPUT.PUT_LINE ('num_small = '||num_small);
126    DBMS_OUTPUT.PUT_LINE ('num_large = '||num_large);
127  END;
```

Results Explain Describe Saved SQL History

```
num_small = 5
num_large = 8
Statement processed.
```

7.) Write a PL/SQL procedure to calculate the incentive on a target achieved and display the message either the record updated or not.

QUERY:

```
DECLARE
  PROCEDURE test1 (
    sal_achieve NUMBER,
    target_qty NUMBER,
    emp_id NUMBER
  )
  IS
    incentive NUMBER := 0;
    updated VARCHAR2(3) := 'No';
BEGIN
  IF sal_achieve > (target_qty + 200) THEN
    incentive := (sal_achieve - target_qty)/4;
    UPDATE employees
    SET salary = salary + incentive
    WHERE employee_id = emp_id;
    updated := 'Yes';
  END IF;
  DBMS_OUTPUT.PUT_LINE (
    'Table updated? ' || updated || ',' ||
    'incentive = ' || incentive || '.'
  );
END test1;
BEGIN
  test1(2300, 2000, 144);
  test1(3600, 3000, 145);
END;
/
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 Save Run

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

Save Run

135 DECLARE
136 PROCEDURE test1 (
137 sal_achieve NUMBER,
138 target_qty NUMBER,
139 emp_id NUMBER
140)
141 IS
142 incentive NUMBER := 0;
143 updated VARCHAR2(3) := 'No';
144 BEGIN
145 IF sal_achieve > (target_qty + 200) THEN
146 incentive := (sal_achieve - target_qty)/4;
147
148 UPDATE employees
149 SET salary = salary + incentive
150 WHERE employee_id = emp_id;
151
152 updated := 'Yes';
153 END IF;
154
155 DBMS_OUTPUT.PUT_LINE (
156 'Table updated? ' || updated || ', ' ||
157 'incentive = ' || incentive || '.';
158);
159 END test1;
160 BEGIN
161 test1(2300, 2000, 144);
162 test1(3600, 3000, 145);
163 END;

Results Explain Describe Saved SQL History

Table updated? Yes, incentive = 75.
Table updated? Yes, incentive = 150.
1 row(s) updated.

8.) Write a PL/SQL procedure to calculate incentive achieved according to the specific sale limit

QUERY:

```
DECLARE
  PROCEDURE test1 (sal_achieve NUMBER)
  IS
    incentive NUMBER := 0;
  BEGIN
    IF sal_achieve > 44000 THEN
      incentive := 1800;
    ELSIF sal_achieve > 32000 THEN
      incentive := 800;
    ELSE
      incentive := 500;
    END IF;
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE (
      'Sale achieved : ' || sal_achieve || ', incentive : ' || incentive || '');
  END test1;
BEGIN
  test1(45000);
  test1(36000);
  test1(28000);
END;
/
```

output:-

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 R ashmita31

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

168 DECLARE
169 PROCEDURE test1 (sal_achieve NUMBER)
170 IS
171 incentive NUMBER := 0;
172 BEGIN
173 IF sal_achieve > 44000 THEN
| incentive := 1800;
174 ELSIF sal_achieve > 32000 THEN
| incentive := 800;
177 ELSE
| incentive := 500;
179 END IF;
180 DBMS_OUTPUT.NEW_LINE;
181 DBMS_OUTPUT.PUT_LINE (
| 'Sale achieved : ' || sal_achieve || ', incentive : ' || incentive || '.'
);
184 END test1;
185 BEGIN
186 test1(45000);
187 test1(36000);
188 test1(28000);
189 END;

Results Explain Describe Saved SQL History

Sale achieved : 45000, incentive : 1800.
Sale achieved : 36000, incentive : 800.
Sale achieved : 28000, incentive : 500.
Statement processed.

9.) Write a PL/SQL program to count number of employees in department 50 and check whether this department have any vacancies or not. There are 45 vacancies in this department.

QUERY:

```
SET SERVEROUTPUT ON
DECLARE
    tot_emp NUMBER;
    get_dep_id NUMBER;

BEGIN
    get_dep_id := 80;
    SELECT Count(*)
    INTO tot_emp
    FROM employees e
        join departments d
        ON e.department_id = d.department_id
    WHERE e.department_id = get_dep_id;
    dbms_output.Put_line ('The employees are in the department'||get_dep_id||' is: '
        ||To_char(tot_emp));
    IF tot_emp >= 45 THEN
        dbms_output.Put_line ('There are no vacancies in the department'||get_dep_id);
    ELSE
        dbms_output.Put_line ('There are'||to_char(45-tot_emp)||' vacancies in department'||get_dep_id );
    END IF;
END;
/

```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 ashmita31

↑ SQL Commands Schema: WKSP_ASHMITA31 Save Run

Language: SQL Rows: 10 Clear Command Find Tables

194 DECLARE
195 v_emp_count NUMBER;
196 v_vacancies NUMBER := 45;
197 BEGIN
198 -- Count the number of employees in department 50
199 SELECT COUNT(*)
200 INTO v_emp_count
201 FROM employees
202 WHERE department_id = 50;
203
204 -- Display the number of employees in department 50
205 DBMS_OUTPUT.PUT_LINE('Number of employees in department 50: ' || v_emp_count);
206
207 -- Check if there are any vacancies
208 IF v_emp_count < v_vacancies THEN
209 DBMS_OUTPUT.PUT_LINE('There are vacancies in department 50.');

210 ELSE
211 DBMS_OUTPUT.PUT_LINE('There are no vacancies in department 50.');

212 END IF;
213 END;

Results Explain Describe Saved SQL History

Number of employees in department 50: 1
There are vacancies in department 50.

Statement processed.

10.) Write a PL/SQL program to count number of employees in a specific department and check whether this department have any vacancies or not. If any vacancies, how many vacancies are in that department.

QUERY:

```
DECLARE
    tot_emp NUMBER;
    get_dep_id NUMBER;

BEGIN
    get_dep_id := 80;
    SELECT Count(*)
    INTO tot_emp
    FROM employees e
        join departments d
        ON e.department_id = d.dept_id
    WHERE e.department_id = get_dep_id;

    dbms_output.Put_line ('The employees are in the department '||get_dep_id||' is: '
        ||To_char(tot_emp));

    IF tot_emp >= 45 THEN
        dbms_output.Put_line ('There are no vacancies in the department '||get_dep_id);
    ELSE
        dbms_output.Put_line ('There are '||to_char(45-tot_emp)||' vacancies in department '||get_dep_id );
    END IF;
END;
/
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 ashmita31

↑ SQL Commands Schema: WKSP_ASHMITA31 Save Run

Language: SQL Rows: 10 Clear Command Find Tables

↻ ↺ 🔍 A:: ⚙️

```
218 DECLARE
219   v_dept_id employees.department_id%TYPE := 50; -- Change this to the desired department ID
220   v_dept_count NUMBER;
221   v_vacancies NUMBER := 45; -- Change this to the number of vacancies in the department
222 BEGIN
223   -- Count the number of employees in the specified department
224   SELECT COUNT(*)
225     INTO v_dept_count
226     FROM employees
227    WHERE department_id = v_dept_id;

228   -- Display the count
229   DBMS_OUTPUT.PUT_LINE('Number of employees in department ' || v_dept_id || ': ' || v_dept_count);
230
231   -- Check for vacancies
232   IF v_dept_count < v_vacancies THEN
233     DBMS_OUTPUT.PUT_LINE('There are vacancies in department ' || v_dept_id || '.');
234     DBMS_OUTPUT.PUT_LINE('Number of vacancies: ' || (v_vacancies - v_dept_count));
235   ELSE
236     DBMS_OUTPUT.PUT_LINE('There are no vacancies in department ' || v_dept_id || '.');
237   END IF;
238
239 END;
```

Results Explain Describe Saved SQL History

Number of employees in department 50: 1
There are vacancies in department 50.
Number of vacancies: 44

Statement processed.

11.) Write a PL/SQL program to display the employee IDs, names, job titles, hire dates, and salaries of all employees

QUERY:

```
DECLARE
v_employee_id employees.employee_id%TYPE;
v_full_name employees.first_name%TYPE;
v_job_id employees.job_id%TYPE;
v_hire_date employees.hire_date%TYPE;
v_salary employees.salary%TYPE;
CURSOR c_employees IS
  SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
  FROM employees;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Employee ID | Full Name | Job Title | Hire Date | Salary');
  DBMS_OUTPUT.PUT_LINE('-----');
  OPEN c_employees;
  FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
  WHILE c_employees%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(v_employee_id || ' ' || v_full_name || ' ' || v_job_id || ' ' ||
v_hire_date || ' ' || v_salary);
    FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
  END LOOP;
  CLOSE c_employees;
END;
/
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 ashmita31

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

244 DECLARE
245 CURSOR employee_cursor IS
246 SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
247 FROM employees;
248 BEGIN
249 -- Loop through the cursor and display employee information
250 FOR employee_rec IN employee_cursor LOOP
251 DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
252 DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
253 DBMS_OUTPUT.PUT_LINE('Job Title: ' || employee_rec.job_id);
254 DBMS_OUTPUT.PUT_LINE('Hire Date: ' || TO_CHAR(employee_rec.hire_date, 'DD-MON-YYYY'));
255 DBMS_OUTPUT.PUT_LINE('Salary: ' || employee_rec.salary);
256 DBMS_OUTPUT.PUT_LINE('-----');
257 END LOOP;
258 END;

Results Explain Describe Saved SQL History

Employee ID: 114
Employee Name: VJ SIDDHARTH
Job Title: COO
Hire Date: 12-JUL-1999
Salary: 40000

Employee ID: 107
Employee Name: JIM JAM
Job Title: DESIGNER
Hire Date: 04-DEC-2000
Salary: 15000

Employee ID: 109
Employee Name: AKAY KOHLI
Job Title: SUPERVISOR
Hire Date: 05-NOV-1988
Salary: 20000

Employee ID: 103
Employee Name: SANJ PARTHI
Job Title: ST_CLERK
Hire Date: 12-APR-1998
Salary: 8650.82

Employee ID: 104
Employee Name: MARY JANE
Job Title: HR_MANAGER
Hire Date: 15-MAR-1998
Salary: 23400.12

Employee ID: 105
Employee Name: SANA EMANUE
Job Title: FI_MANAGER
Hire Date: 01-MAY-1994
Salary: 22000

Employee ID: 108
Employee Name: MIKA DEV
Job Title: ENGINEER
Hire Date: 12-OCT-2004

Employee ID: 100
Employee Name: SAHANA JAY
Job Title: SL_REP
Hire Date: 01-MAY-1999

Employee ID: 113
Employee Name: KHAN HARSH
Job Title: QA
Hire Date: 23-FEB-1998

Employee ID: 110
Employee Name: TARA UMA
Job Title: ST_CLERK
Hire Date: 02-AUG-1999

Employee ID: 109
Employee Name: BEN DAVI
Job Title: AUDITOR
Hire Date: 17-MAY-2002

Employee ID: 106
Employee Name: JOE VIJAY
Job Title: MANAGER
Hire Date: 30-NOV-2000

Employee ID: 113
Employee Name: KHAN HAR
Job Title: CEO
Hire Date: 23-FEB-1998

Salary: 45000

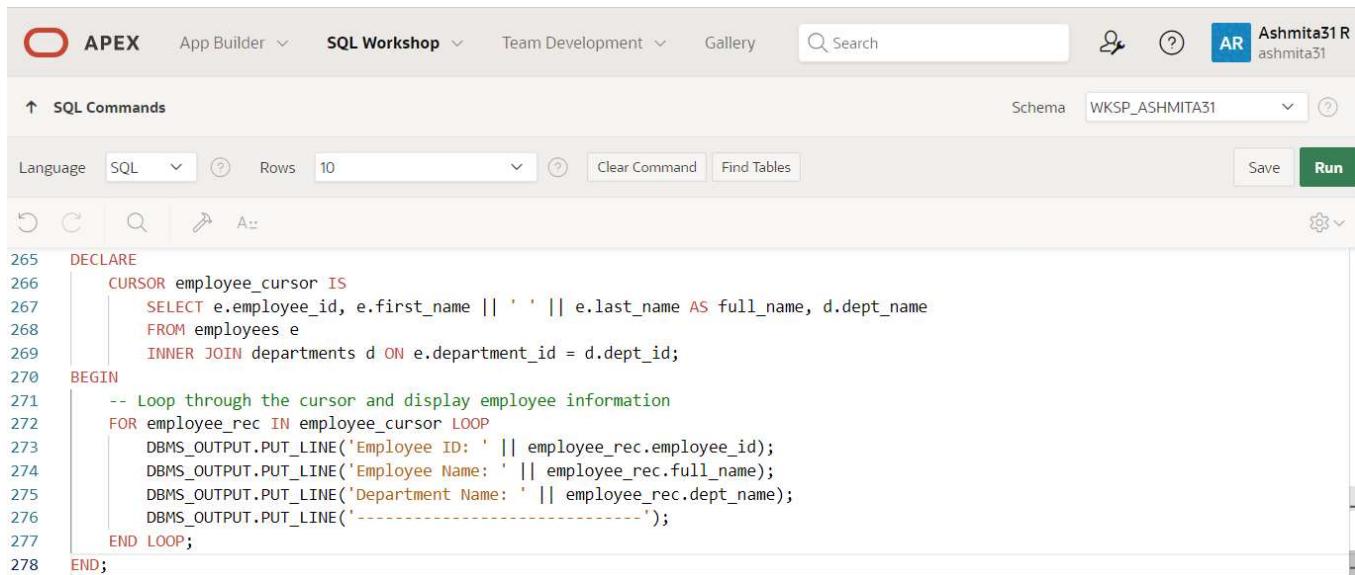
Employee ID: 101
Employee Name: ASHA RAVI
Job Title: MK_MANAGER
Hire Date: 07-JAN-2004

12.) Write a PL/SQL program to display the employee IDs, names, and department names of all employees.

QUERY:

```
DECLARE
CURSOR emp_cursor IS
SELECT e.employee_id, e.first_name, m.first_name AS manager_name
FROM employees e
LEFT JOIN employees m ON e.manager_id = m.employee_id;
emp_record emp_cursor%ROWTYPE;
BEGIN
OPEN emp_cursor;
FETCH emp_cursor INTO emp_record;
WHILE emp_cursor%FOUND LOOP
DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.employee_id);
DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_record.first_name);
DBMS_OUTPUT.PUT_LINE('Manager Name: ' || emp_record.manager_name);
DBMS_OUTPUT.PUT_LINE('-----');
FETCH emp_cursor INTO emp_record;
END LOOP;
CLOSE emp_cursor;
END;
/
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', 'Gallery', a search bar, and a user profile for 'Ashmita31 R ashmita31'. The main workspace is titled 'SQL Commands' and shows the following PL/SQL code:

```
265  DECLARE
266    CURSOR employee_cursor IS
267      SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, d.dept_name
268      FROM employees e
269      INNER JOIN departments d ON e.department_id = d.dept_id;
270  BEGIN
271    -- Loop through the cursor and display employee information
272    FOR employee_rec IN employee_cursor LOOP
273      DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
274      DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
275      DBMS_OUTPUT.PUT_LINE('Department Name: ' || employee_rec.dept_name);
276      DBMS_OUTPUT.PUT_LINE('-----');
277    END LOOP;
278  END;
```

The code is numbered from 265 to 278. The 'Run' button at the bottom right of the editor is highlighted in green.

Results Explain Describe Saved SQL History

Employee ID: 114
Employee Name: VJ SIDDHU
Department Name: EXECUTIVE

Employee ID: 107
Employee Name: JIM JAM
Department Name: MANUFACTURING

Employee ID: 109
Employee Name: AKAY KOHLI
Department Name: MANUFACTURING

Employee ID: 103
Employee Name: SANJ PARTHI
Department Name: STOCK

Employee ID: 104
Employee Name: MARY JANE
Department Name: HR

Employee ID: 105
Employee Name: SANA EMANUEL
Department Name: FINANCE

Employee ID: 108
Employee Name: MIKA DEV
Department Name: MANUFACTURING

Employee ID: 100
Employee Name: SAHANA JAY
Department Name: MARKETING

Employee ID: 110
Employee Name: TARA UMA
Department Name: STOCK

Employee ID: 109
Employee Name: BEN DAVIES
Department Name: FINANCE

Employee ID: 106
Employee Name: JOE VIJAY
Department Name: MANAGEMENT

Employee ID: 113
Employee Name: KHAN HARSH
Department Name: EXECUTIVE

Employee ID: 101
Employee Name: ASHA RAVI
Department Name: MARKETING

Statement processed.

0.03 seconds

13.) Write a PL/SQL program to display the job IDs, titles, and minimum salaries of all jobs

QUERY:

```
DECLARE
  CURSOR job_cursor IS
    SELECT e.job_id, j.lowest_sal
      FROM job_grade j,employees e;
  job_record job_cursor%ROWTYPE;
BEGIN
  OPEN job_cursor;
  FETCH job_cursor INTO job_record;
  WHILE job_cursor%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_record.job_id);
    DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_record.lowest_sal);
    DBMS_OUTPUT.PUT_LINE('-----');
    FETCH job_cursor INTO job_record;
  END LOOP;
  CLOSE job_cursor;
END;
/
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 AR ashmita31

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

284 DECLARE
285 CURSOR job_cursor IS
286 SELECT job_id, MIN(salary) AS min_salary
287 FROM employees
288 GROUP BY job_id;
289 BEGIN
290 -- Loop through the cursor and display job information
291 FOR job_rec IN job_cursor LOOP
292 DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_rec.job_id);
293 DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_rec.min_salary);
294 DBMS_OUTPUT.PUT_LINE('-----');
295 END LOOP;
296 END;
297 /
298
299

Results Explain Describe Saved SQL History

```
Job ID: CEO
Minimum Salary: 45000
-----
Job ID: FI_MANAGER
Minimum Salary: 22000
-----
Job ID: SUPERVISOR
Minimum Salary: 20000
```

14.) Write a PL/SQL program to display the employee IDs, names, and job history start dates of all employees.

QUERY:

```
DECLARE
  CURSOR employees_cur IS
    SELECT employee_id, last_name, job_id, start_date
    FROM employees NATURAL JOIN job_history;
  emp_start_date DATE;
BEGIN
  dbms_output.Put_line(Rpad('Employee ID', 15) || Rpad('Last Name', 25) || Rpad('Job Id', 35)
  || 'Start Date');
  dbms_output.Put_line('-----');
  FOR emp_sal_rec IN employees_cur LOOP
    -- find out most recent end_date in job_history
    SELECT Max(end_date) + 1
    INTO emp_start_date
    FROM job_history
    WHERE employee_id = emp_sal_rec.employee_id;
    IF emp_start_date IS NULL THEN
      emp_start_date := emp_sal_rec.start_date;
    END IF;
    dbms_output.Put_line(Rpad(emp_sal_rec.employee_id, 15)
      || Rpad(emp_sal_rec.last_name, 25)
      || Rpad(emp_sal_rec.job_id, 35)
      || To_char(emp_start_date, 'dd-mon-yyyy'));
  END LOOP;
END;
/
```

OUTPUT:

↑ SQL Commands

Schema

WKSP_ASHMITA31

Save

Run



15.) Write a PL/SQL program to display the employee IDs, names, and job history end dates of all employees.

QUERY:

```
DECLARE
v_employee_id employees.employee_id%TYPE;
v_first_name employees.last_name%TYPE;
v_end_date job_history.end_date%TYPE;
CURSOR c_employees IS
  SELECT e.employee_id, e.first_name, jh.end_date
    FROM employees e
   JOIN job_history jh ON e.employee_id = jh.employee_id;
BEGIN
  OPEN c_employees;
  FETCH c_employees INTO v_employee_id, v_first_name, v_end_date;
  WHILE c_employees%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || v_employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_first_name);
    DBMS_OUTPUT.PUT_LINE('End Date: ' || v_end_date);
    DBMS_OUTPUT.PUT_LINE('-----');
    FETCH c_employees INTO v_employee_id, v_first_name, v_end_date;
  END LOOP;
  CLOSE c_employees;
END;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop (selected), Team Development, and Gallery, along with a search bar and user authentication information (Ashmita31 R ashmita31).

The main workspace displays a PL/SQL script:

```
325  DECLARE
326      CURSOR employee_cursor IS
327          SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.end_date
328          FROM employees e
329          JOIN job_history jh ON e.employee_id = jh.employee_id;
330
331      BEGIN
332          -- Loop through the cursor and display employee information
333          FOR employee_rec IN employee_cursor LOOP
334              DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
335              DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
336
337              -- Check if the end date is NULL (meaning the employee is currently in the job)
338              IF employee_rec.end_date IS NULL THEN
339                  DBMS_OUTPUT.PUT_LINE('Job History End Date: (Still Employed)');
340
341              ELSE
342                  DBMS_OUTPUT.PUT_LINE('Job History End Date: ' || TO_CHAR(employee_rec.end_date, 'DD-MON-YYYY'));
343              END IF;
344
345          END LOOP;
346      END;
```

The script uses a cursor to select employee details and their job history. It then loops through the cursor, printing each employee's ID and name. If the employee's end date is null, it prints a message indicating they are still employed. Otherwise, it prints the end date formatted as DD-MON-YYYY.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

PROCEDURES AND FUNCTIONS

EX_NO: 17

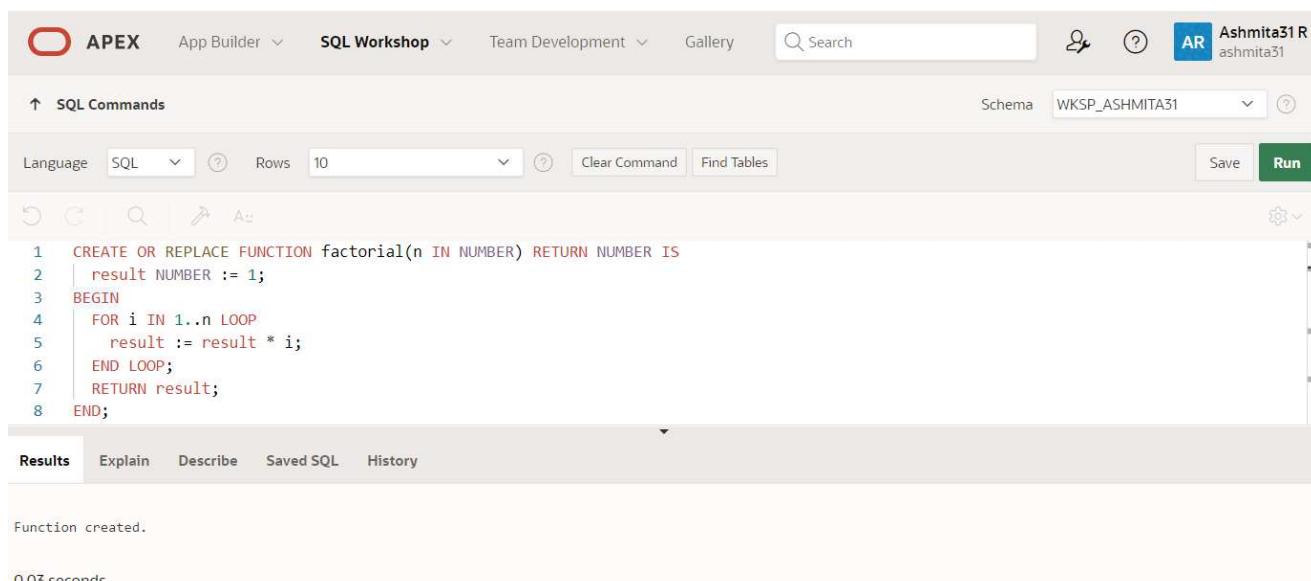
DATE:

1.) Factorial of a number using function.

QUERY:

```
DECLARE
    fac NUMBER := 1;
    n NUMBER := :1;
BEGIN
    WHILE n > 0 LOOP
        fac := n * fac;
        n := n - 1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(fac);
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The right side shows the user's profile: Ashmita31 R, ashmita31. The main workspace is titled 'SQL Commands' and shows the schema 'WKSP_ASHMITA31'. The code area contains the following PL/SQL code:

```
1 CREATE OR REPLACE FUNCTION factorial(n IN NUMBER) RETURN NUMBER IS
2     result NUMBER := 1;
3 BEGIN
4     FOR i IN 1..n LOOP
5         result := result * i;
6     END LOOP;
7     RETURN result;
8 END;
```

The 'Results' tab at the bottom shows the output: 'Function created.' and '0.03 seconds'.

2.) Write a PL/SQL program using Procedures IN,INOUT,OUT parameters to retrieve the corresponding book information in library.

QUERY:

```
CREATE OR REPLACE PROCEDURE get_book_info (
    p_book_id IN NUMBER,
    p_title IN OUT VARCHAR2,
    p_author OUT VARCHAR2,
    p_year_published OUT NUMBER
)
AS
BEGIN
    SELECT title, author, year_published INTO p_title, p_author, p_year_published
    FROM books
    WHERE book_id = p_book_id;

    p_title := p_title || ' - Retrieved';
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        p_title := NULL;
        p_author := NULL;
        p_year_published := NULL;
END;

DECLARE
    v_book_id NUMBER := 1;
    v_title VARCHAR2(100);
    v_author VARCHAR2(100);
    v_year_published NUMBER;
BEGIN
    v_title := 'Initial Title';

    get_book_info(p_book_id => v_book_id, p_title => v_title, p_author => v_author,
    p_year_published => v_year_published);

    DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
    DBMS_OUTPUT.PUT_LINE('Author: ' || v_author);
    DBMS_OUTPUT.PUT_LINE('Year Published: ' || v_year_published);
```

END;

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user authentication information (Ashmita31 R, ashmita31) are also present. The main workspace is titled "SQL Commands" and contains the following PL/SQL code:

```
56  DECLARE
57      v_book_id NUMBER := 1; -- Example book ID
58      v_title VARCHAR2(100);
59      v_author VARCHAR2(100);
60      v_year_published NUMBER;
61  BEGIN
62      -- Call the procedure
63      get_book_info(v_book_id, v_title, v_author, v_year_published);
64
65      -- Display the retrieved information
66      IF v_title IS NOT NULL THEN
67          DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
68          DBMS_OUTPUT.PUT_LINE('Author: ' || v_author);
69          DBMS_OUTPUT.PUT_LINE('Year Published: ' || v_year_published);
70      ELSE
71          DBMS_OUTPUT.PUT_LINE('Book information could not be retrieved.');
72      END IF;
73  END;
```

The results tab is selected, showing the output of the executed code:

```
Title: 1984
Author: George Orwell
Year Published: 1949

Statement processed.
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

TRIGGER

EX_NO: 18

DATE:

1.) Write a code in PL/SQL to develop a trigger that enforces referential integrity by preventing the deletion of a parent record if child records exist

QUERY:

```
CREATE OR REPLACE TRIGGER prevent_parent_deletion
BEFORE DELETE ON parent_table
FOR EACH ROW
DECLARE
    child_exists EXCEPTION;
    PRAGMA EXCEPTION_INIT(child_exists, -20001);
    v_child_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id =
    :OLD.parent_id;
    IF v_child_count > 0 THEN
        RAISE child_exists;
    END IF;
EXCEPTION
    WHEN child_exists THEN
        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child records
exist.');
END;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 Ashmita31 R ashmita31

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

↻ 🔍 A::

```
14 CREATE OR REPLACE TRIGGER prevent_parent_deletion
15 BEFORE DELETE ON parent_table
16 FOR EACH ROW
17 DECLARE
18     child_exists EXCEPTION;
19     PRAGMA EXCEPTION_INIT(child_exists, -20001);
20     v_child_count NUMBER;
21 BEGIN
22     SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id = :OLD.parent_id;
23     IF v_child_count > 0 THEN
24         RAISE child_exists;
25     END IF;
26 EXCEPTION
27     WHEN child_exists THEN
28         RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child records exist.');
29 END;
```

Results Explain Describe Saved SQL History

Trigger created.

2.) Write a code in PL/SQL to create a trigger that checks for duplicate values in a specific column and raises an exception if found

QUERY:

```
CREATE OR REPLACE TRIGGER check_duplicates
BEFORE INSERT OR UPDATE ON unique_values_table
FOR EACH ROW
DECLARE
    duplicate_found EXCEPTION;
    PRAGMA EXCEPTION_INIT(duplicate_found, -20002);
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM unique_values_table
    WHERE unique_col = :NEW.unique_col AND id != :NEW.id;
    IF v_count > 0 THEN
        RAISE duplicate_found;
    END IF;
EXCEPTION
    WHEN duplicate_found THEN
        RAISE_APPLICATION_ERROR(-20002, 'Duplicate value found in unique_col.');
END;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 AR Ashmita31 R ashmita31

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

↻ ⌂ 🔍 🖊 A::

```
40 CREATE OR REPLACE TRIGGER check_duplicates
41 BEFORE INSERT OR UPDATE ON unique_values_table
42 FOR EACH ROW
43 DECLARE
44     duplicate_found EXCEPTION;
45     PRAGMA EXCEPTION_INIT(duplicate_found, -20002);
46     v_count NUMBER;
47 BEGIN
48     SELECT COUNT(*) INTO v_count FROM unique_values_table
49     WHERE unique_col = :NEW.unique_col AND id != :NEW.id;
50     IF v_count > 0 THEN
51         RAISE duplicate_found;
52     END IF;
53 EXCEPTION
54     WHEN duplicate_found THEN
55         RAISE_APPLICATION_ERROR(-20002, 'Duplicate value found in unique_col.');
56 END;
```

Results Explain Describe Saved SQL History

Trigger created.

0.05 seconds

3.) Write a code in PL/SQL to create a trigger that restricts the insertion of new rows if the total of a column's values exceeds a certain threshold

QUERY:

```
CREATE OR REPLACE TRIGGER check_threshold
BEFORE INSERT OR UPDATE ON threshold_table
FOR EACH ROW
DECLARE
    threshold_exceeded EXCEPTION;
    PRAGMA EXCEPTION_INIT(threshold_exceeded, -20003);
    v_sum NUMBER;
    v_threshold NUMBER := 10000; -- Set your threshold here
BEGIN
    SELECT SUM(value_col) INTO v_sum FROM threshold_table;
    v_sum := v_sum + :NEW.value_col;
    IF v_sum > v_threshold THEN
        RAISE threshold_exceeded;
    END IF;
EXCEPTION
    WHEN threshold_exceeded THEN
        RAISE_APPLICATION_ERROR(-20003, 'Threshold exceeded for value_col.');
END;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search AR Ashmita31 R ashmita31

↑ SQL Commands Schema WKSP_ASHMITA31 Rows 10 Save Run

Language SQL Clear Command Find Tables

↻ ↺ 🔍 ↻ A..

```
65 CREATE OR REPLACE TRIGGER check_threshold
66 BEFORE INSERT OR UPDATE ON threshold_table
67 FOR EACH ROW
68 DECLARE
69     threshold_exceeded EXCEPTION;
70     PRAGMA EXCEPTION_INIT(threshold_exceeded, -20003);
71     v_sum NUMBER;
72     v_threshold NUMBER := 10000; -- Set your threshold here
73 BEGIN
74     SELECT SUM(value_col) INTO v_sum FROM threshold_table;
75     v_sum := v_sum + :NEW.value_col;
76     IF v_sum > v_threshold THEN
77         RAISE threshold_exceeded;
78     END IF;
79 EXCEPTION
80     WHEN threshold_exceeded THEN
81         RAISE_APPLICATION_ERROR(-20003, 'Threshold exceeded for value_col.');
82 END;
83
```

Results Explain Describe Saved SQL History

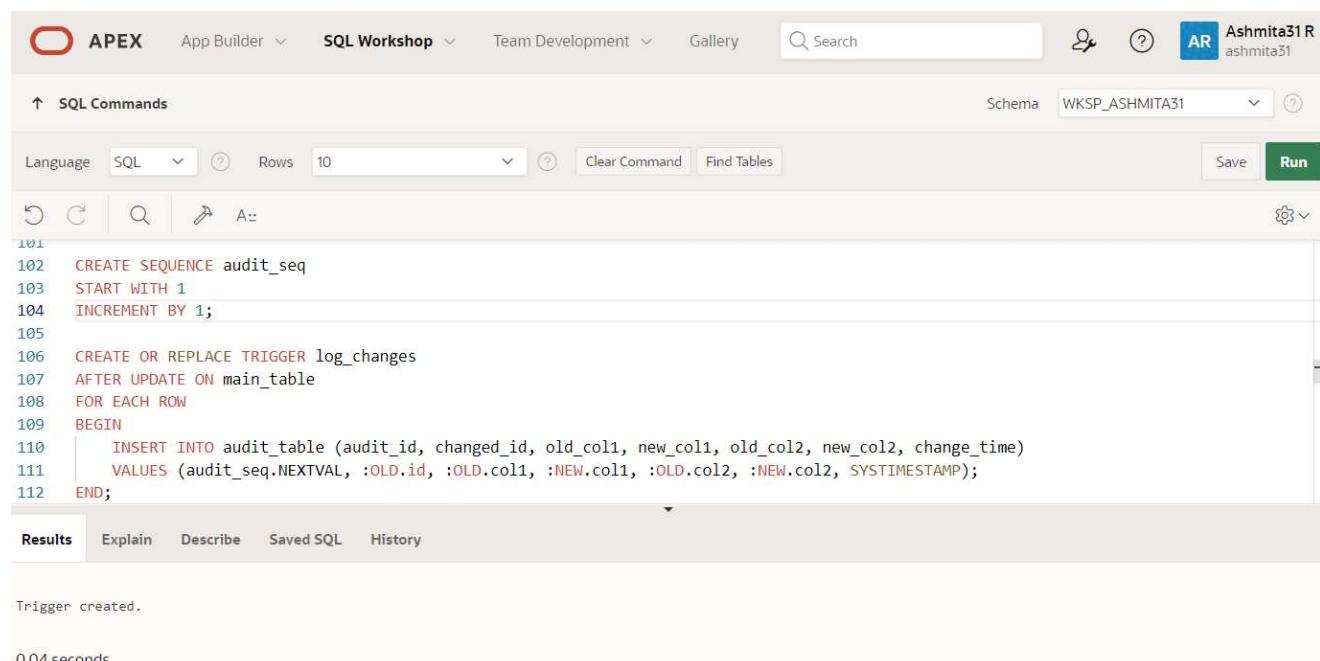
Trigger created.

4.) Write a code in PL/SQL to design a trigger that captures changes made to specific columns and logs them in an audit table.

QUERY:

```
CREATE OR REPLACE TRIGGER log_changes
AFTER UPDATE ON main_table
FOR EACH ROW
BEGIN
    INSERT INTO audit_table (audit_id, changed_id, old_col1, new_col1, old_col2, new_col2,
change_time)
    VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2, :NEW.col2,
SYSTIMESTAMP);
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile are also present. The main area is titled "SQL Commands". The schema is set to "WKSP_ASHMITA31". The code editor contains the following PL/SQL code:

```
101
102 CREATE SEQUENCE audit_seq
103 START WITH 1
104 INCREMENT BY 1;
105
106 CREATE OR REPLACE TRIGGER log_changes
107 AFTER UPDATE ON main_table
108 FOR EACH ROW
109 BEGIN
110     INSERT INTO audit_table (audit_id, changed_id, old_col1, new_col1, old_col2, new_col2, change_time)
111     VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2, :NEW.col2, SYSTIMESTAMP);
112 END;
```

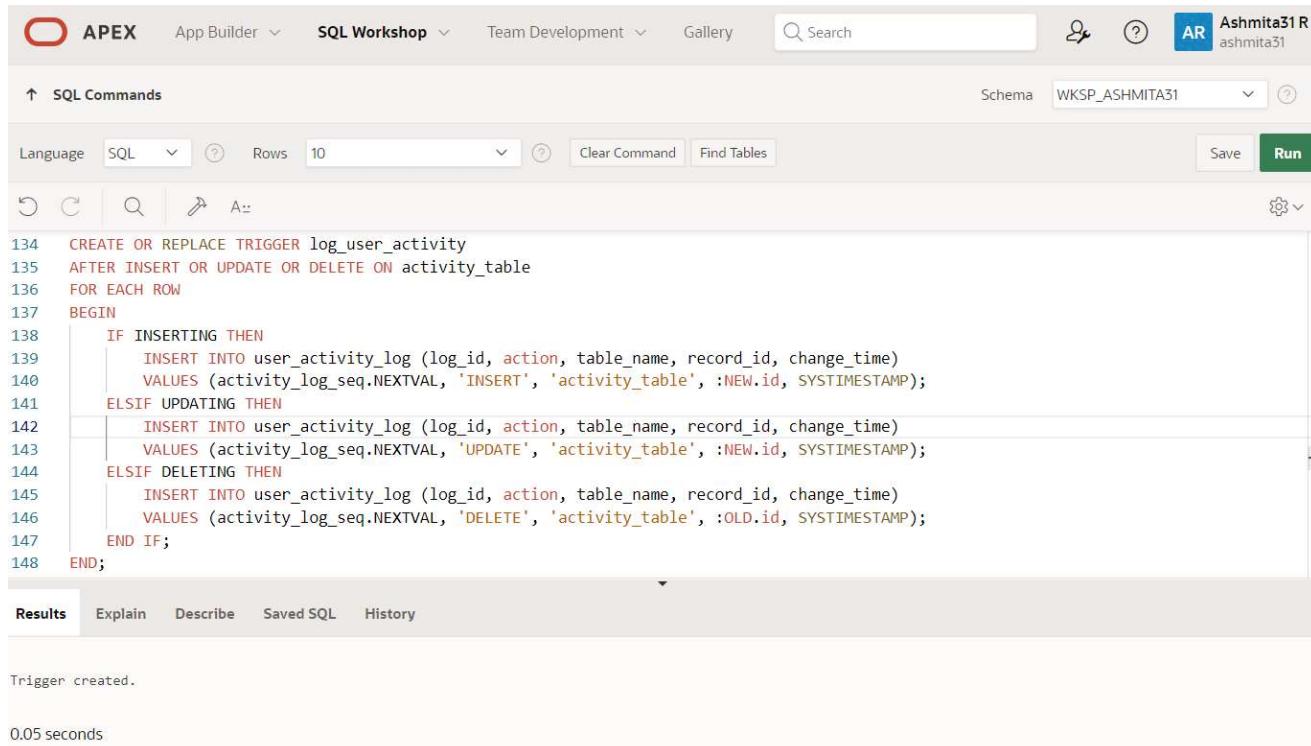
The code is highlighted with syntax coloring. Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The results pane displays the message "Trigger created." and "0.04 seconds".

5.) Write a code in PL/SQL to implement a trigger that records user activity (inserts, updates, deletes) in an audit log for a given set of tables.

QUERY:

```
CREATE OR REPLACE TRIGGER log_user_activity
AFTER INSERT OR UPDATE OR DELETE ON activity_table
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
    VALUES (activity_log_seq.NEXTVAL, 'INSERT', 'activity_table', :NEW.id, SYSTIMESTAMP);
  ELSIF UPDATING THEN
    INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
    VALUES (activity_log_seq.NEXTVAL, 'UPDATE', 'activity_table', :NEW.id,
SYSTIMESTAMP);
  ELSIF DELETING THEN
    INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
    VALUES (activity_log_seq.NEXTVAL, 'DELETE', 'activity_table', :OLD.id, SYSTIMESTAMP);
  END IF;
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', 'Gallery', and a search bar. The right side shows a user profile for 'Ashmita31' (ashmita31). The main workspace is titled 'SQL Commands'. It has tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below these are icons for Undo, Redo, Find, and Paste. The SQL code area contains numbered lines from 134 to 148, representing the trigger creation script. The code is color-coded for syntax. At the bottom, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The results pane displays the message 'Trigger created.' and '0.05 seconds'.

```
134 CREATE OR REPLACE TRIGGER log_user_activity
135 AFTER INSERT OR UPDATE OR DELETE ON activity_table
136 FOR EACH ROW
137 BEGIN
138   IF INSERTING THEN
139     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
140     VALUES (activity_log_seq.NEXTVAL, 'INSERT', 'activity_table', :NEW.id, SYSTIMESTAMP);
141   ELSIF UPDATING THEN
142     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
143     VALUES (activity_log_seq.NEXTVAL, 'UPDATE', 'activity_table', :NEW.id, SYSTIMESTAMP);
144   ELSIF DELETING THEN
145     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
146     VALUES (activity_log_seq.NEXTVAL, 'DELETE', 'activity_table', :OLD.id, SYSTIMESTAMP);
147   END IF;
148 END;
```

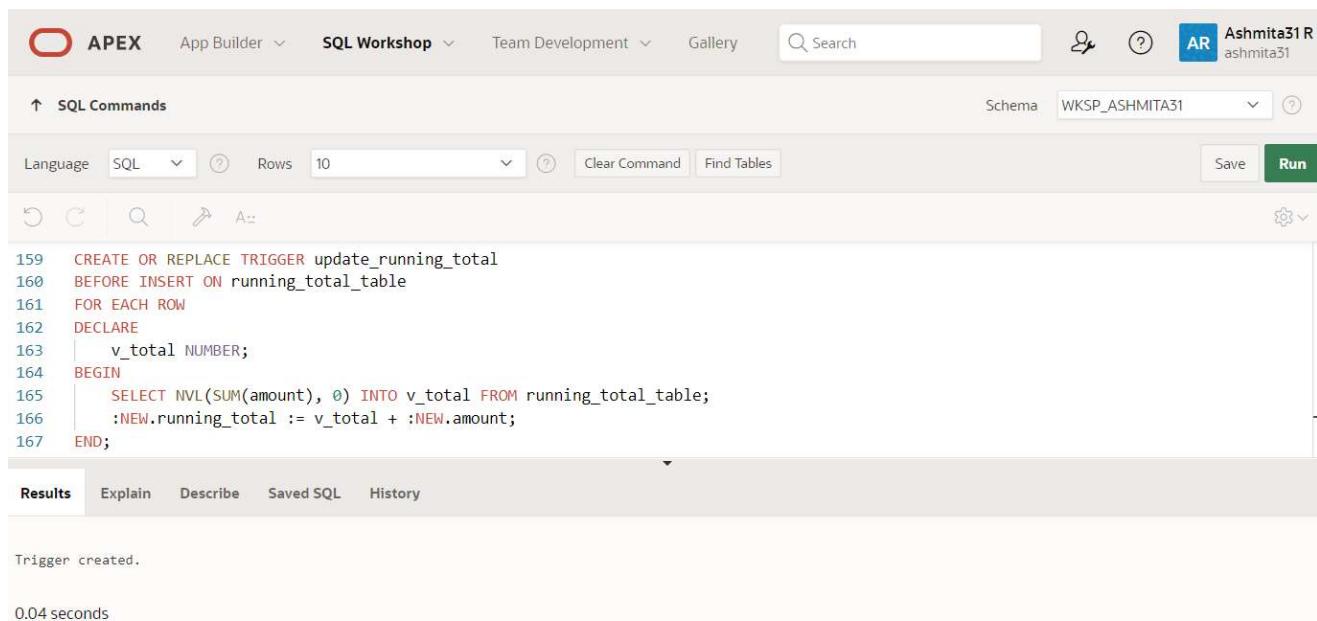
Trigger created.
0.05 seconds

6.) Write a code in PL/SQL to implement a trigger that automatically calculates and updates a running total column for a table whenever new rows are inserted

QUERY:

```
CREATE OR REPLACE TRIGGER update_running_total
BEFORE INSERT ON running_total_table
FOR EACH ROW
DECLARE
    v_total NUMBER;
BEGIN
    SELECT NVL(SUM(amount), 0) INTO v_total FROM running_total_table;
    :NEW.running_total := v_total + :NEW.amount;
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, a search bar, and user information (Ashmita31 R). The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_ASHMITA31'. Below the title are buttons for Language (SQL selected), Rows (10), Clear Command, Find Tables, Save, and Run. The code area contains the PL/SQL trigger definition. The bottom section shows the results of the command execution.

```
159 CREATE OR REPLACE TRIGGER update_running_total
160 BEFORE INSERT ON running_total_table
161 FOR EACH ROW
162 DECLARE
163     v_total NUMBER;
164 BEGIN
165     SELECT NVL(SUM(amount), 0) INTO v_total FROM running_total_table;
166     :NEW.running_total := v_total + :NEW.amount;
167 END;
```

Results tab is active. The output pane displays the message "Trigger created." and "0.04 seconds".

7.) Write a code in PL/SQL to create a trigger that validates the availability of items before allowing an order to be placed, considering stock levels and pending orders

QUERY:

```
CREATE OR REPLACE TRIGGER validate_order
BEFORE INSERT ON orders
FOR EACH ROW
DECLARE
    v_stock NUMBER;
    insufficient_stock EXCEPTION;
    PRAGMA EXCEPTION_INIT(insufficient_stock, -20004);
BEGIN
    SELECT stock_quantity INTO v_stock FROM items WHERE item_id = :NEW.item_id;
    IF v_stock < :NEW.order_quantity THEN
        RAISE insufficient_stock;
    END IF;
    UPDATE items SET stock_quantity = stock_quantity - :NEW.order_quantity WHERE item_id
    = :NEW.item_id;
EXCEPTION
    WHEN insufficient_stock THEN
        RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');
END;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_ASHMITA31 AR Ashmita31 R

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

↻ ↺ 🔍 🗑️ A:

```
185 CREATE OR REPLACE TRIGGER validate_order
186 BEFORE INSERT ON orders
187 FOR EACH ROW
188 DECLARE
189     v_stock NUMBER;
190     insufficient_stock EXCEPTION;
191     PRAGMA EXCEPTION_INIT(insufficient_stock, -20004);
192 BEGIN
193     SELECT stock_quantity INTO v_stock FROM items WHERE item_id = :NEW.item_id;
194     IF v_stock < :NEW.order_quantity THEN
195         RAISE insufficient_stock;
196     END IF;
197     UPDATE items SET stock_quantity = stock_quantity - :NEW.order_quantity WHERE item_id = :NEW.item_id;
198 EXCEPTION
199     WHEN insufficient_stock THEN
200         RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');
201 END;
```

Results Explain Describe Saved SQL History

Trigger created.

0.06 seconds

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

MONGO DB

EX_NO: 19

DATE:

1.) Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

QUERY:

```
db.restaurants.find( { $or: [ { name: /^Wil/ }, { cuisine: { $nin: ['American', 'Chinese'] } } ] , { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 } );
```

OUTPUT:

```
ashmita_31>

...   {
...     $or: [
...       { name: /^Wil/ },
...       { cuisine: { $nin: ['American', 'Chinese'] } }
...     ]
...   },
...   {
...     restaurant_id: 1,
...     name: 1,
...     borough: 1,
...     cuisine: 1
...   }
...
[
{
  _id: ObjectId('564c2d949eb21ad392f1d6de'),
  borough: 'Manhattan',
  cuisine: 'Other',
  name: '',
  restaurant_id: '50017887'
},
```

2.) Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08- 11T00:00:00Z" among many of survey dates.

QUERY:

```
db.restaurants.find( { grades: { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 } );
```

OUTPUT:

```
ashmita_31> db.restaurants.find( { grades: { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") }}
```

3.) Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z".

QUERY:

```
db.restaurants.find( {"grades.1.grade": "A", "grades.1.score": 9, "grades.1.date": ISODate("2014-08-11T00:00:00Z") }, { restaurant_id: 1, name: 1, grades: 1 } );
```

OUTPUT:

```
ashmita_31>
...
{
  "grades.1.grade": "A",
  "grades.1.score": 9,
  "grades.1.date": ISODate("2014-08-01T00:00:00Z")
},
{
  "restaurant_id": 1,
  "name": 1,
  "grades": 1
}
ashmita_31> .
```

4.) Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52

QUERY:

```
db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1})
```

OUTPUT:

```
ashmita_31> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1})
ashmita_31>
```

5.) Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

QUERY:

```
db.restaurants.find( {}, { _id: 0 }).sort( { name: 1 } );
```

OUTPUT:

```
ashmita_31> db.restaurants.find({}, { _id: 0 }).sort({ name: 1 });
[ {
  address: {
    building: '154',
    coord: [ -73.9189064, 40.8654529 ],
    street: 'Post Ave',
    zipcode: '10034'
  },
  borough: 'Manhattan',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017887'
},
{
  address: {
    building: '508',
    coord: [ -73.999813, 40.683876 ],
    street: 'Henry St',
  },
  borough: 'Brooklyn',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017910'
},
{
  address: {
    building: '15',
    coord: [ -73.9966882, 40.7139264 ],
    street: 'Division St',
    zipcode: '10002'
  },
  borough: 'Manhattan',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017912'
},
{
  address: {
    building: '4704',
    coord: [ -74.013391, 40.64943 ],
    street: '3Rd Ave',
    zipcode: '11220'
  },
  borough: 'Brooklyn',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017925'
}
]
ashmita_31>
```

6.) Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

QUERY:

```
db.restaurants.find({}, { _id: 0 }).sort({ name: -1 })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({}, { _id: 0 }).sort({ name: -1 })
[
  {
    address: {
      building: '154',
      coord: [ -73.9189064, 40.8654529 ],
      street: 'Post Ave',
      zipcode: '10034'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017887'
  },
  {
    address: {
      building: '508',
      coord: [ -73.999813, 40.683876 ],
      street: 'Henry St',
      zipcode: '11231'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017910'
  },
  {
    address: {
      building: '15',
      coord: [ -73.9966882, 40.7139264 ],
      street: 'Division St',
      zipcode: '10002'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017912'
  },
  {
    address: {
      building: '4704',
      coord: [ -74.013391, 40.64943 ],
      street: '3Rd Ave',
      zipcode: '11220'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017925'
  }
]
```

7.) Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.

QUERY:

```
db.restaurants.find({}, { _id: 0 }).sort({ cuisine: 1, borough: -1 })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({}, { _id: 0 }).sort({ cuisine: 1, borough: -1 })
[
  {
    address: {
      building: '154',
      coord: [ -73.9189064, 40.8654529 ],
      street: 'Post Ave',
      zipcode: '10034'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017887'
  },
  {
    address: {
      building: '15',
      coord: [ -73.9966882, 40.7139264 ],
      street: 'Division St',
      zipcode: '10002'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017912'
  },
  {
    address: {
      building: '508',
      coord: [ -73.999813, 40.683876 ],
      street: 'Henry St',
      zipcode: '11231'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017910'
  },
  {
    address: {
      building: '4704',
      coord: [ -74.013391, 40.64943 ],
      street: '3Rd Ave',
      zipcode: '11220'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017925'
  }
]
ashmita_31>
```

8.) Write a MongoDB query to know whether all the addresses contains the street or not.

QUERY:

```
db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })
[
  {
    _id: ObjectId('564c2d949eb21ad392f1d6de'),
    address: {
      building: '154',
      coord: [ -73.9189064, 40.8654529 ],
      street: 'Post Ave',
      zipcode: '10034'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017887'
  },
  {
    _id: ObjectId('564c2d949eb21ad392f1d6ec'),
    address: {
      building: '508',
      coord: [ -73.999813, 40.683876 ],
      street: 'Henry St',
      zipcode: '11231'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017910'
  },
  {
    _id: ObjectId('564c2d949eb21ad392f1d6ed'),
    address: {
      building: '15',
      coord: [ -73.9966882, 40.7139264 ],
      street: 'Division St',
      zipcode: '10002'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017912'
  },
  {
    _id: ObjectId('564c2d949eb21ad392f1d6f5'),
    address: {
      building: '4704',
      coord: [ -74.013391, 40.64943 ],
      street: '3Rd Ave',
      zipcode: '11220'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017925'
  }
]
ashmita_31>
```

9.) Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.

QUERY:

```
db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
[
  {
    _id: ObjectId('564c2d949eb21ad392f1d6de'),
    address: {
      building: '154',
      coord: [ -73.9189064, 40.8654529 ],
      street: 'Post Ave',
      zipcode: '10034'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017887'
  },
  {
    _id: ObjectId('564c2d949eb21ad392f1d6ec'),
    address: {
      building: '508',
      coord: [ -73.999813, 40.683876 ],
      street: 'Henry St',
      zipcode: '11231'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: []
  }
]
```

10. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

QUERY:

```
db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 });
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 });
ashmita_31> ■
```

11. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.

QUERY:

```
db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
ashmita_31> ■
```

12. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.

QUERY:

```
db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
ashmita_31> ■
```

13. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })  
ashmita_31> ■
```

14. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })  
ashmita_31> ■
```

15. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
ashmita_31> -
```

16. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
ashmita_31> -
```

17. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })  
ashmita_31>
```

18. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })  
ashmita_31>
```

19. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })  
ashmita_31> ■
```

20. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
ashmita_31>
```

21. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
ashmita_31>
```

22. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })  
ashmita_31>
```

23. Write a MongoDB query to find the restaurants that have a grade with a score of 2 or a grade with a score of 6.

QUERY:

```
db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
```

OUTPUT:

```
ashmita_31> db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })  
ashmita_31>
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

MONGO DB

EX_NO: 20

DATE:

1.) Find all movies with full information from the 'movies' collection that released in the year 1893.

QUERY:

```
db.movies.find({ year: 1893 })
```

OUTPUT:

```
ashmita_31> db.movies.find({ year: 1893 })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kayser', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  }
]
```

2.) Find all movies with full information from the 'movies' collection that have a runtime greater than 120 minutes.

QUERY:

```
db.movies.find({ runtime: { $gt: 120 } })
```

OUTPUT:

```
ashmita_31> db.movies.find({ runtime: { $gt: 120 } })
[
  {
    _id: ObjectId('573a1390f22313caabcd5967'),
    plot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',
    genres: [ 'Action', 'Adventure', 'Crime' ],
    runtime: 399,
    rated: 'NOT RATED',
    cast: [ 'Musidora', 'éduardMathè', 'Marcel Lèvesque', 'Jean Aymè' ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTC1NTY3NDIzMjIwMjIwXkFtZTgwNTIyODg5MTE@._V1_SY1000_SX677_AL_.jpg',
    title: 'Les vampires',
    fullplot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',
    languages: [ 'French' ],
    released: ISODate('1916-11-23T00:00:00.000Z'),
    directors: [ 'Louis Feuillade' ],
    writers: [ 'Louis Feuillade' ],
    awards: { wins: 0, nominations: 1, text: '1 nomination.' },
    lastupdated: '2015-09-02 00:24:27.333000000',
    year: 1915,
    imdb: { rating: 6.8, votes: 2878, id: 6206 },
    countries: [ 'France' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.8, numReviews: 2118, meter: 82 },
      dvd: ISODate('2000-05-16T00:00:00.000Z'),
      critic: { rating: 8.8, numReviews: 13, meter: 100 },
      lastUpdated: ISODate('2015-09-15T17:02:33.000Z'),
      rotten: 0,
      fresh: 13
    }
  }
]
ashmita_31>
```

3.) Find all movies with full information from the 'movies' collection that have "Short" genre.

QUERY:

```
db.movies.find({ genres: 'Short' })
```

OUTPUT:

```
] ashmita_31> db.movies.find({ genres: 'Short' })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kayser', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  }
]
ashmita_31>
```

4.) Retrieve all movies from the 'movies' collection that were directed by "William K.L. Dickson" and include complete information for each movie.

QUERY:

```
db.movies.find({ directors: 'William K.L. Dickson' })
```

OUTPUT:

```
ashmita_31> db.movies.find({ directors: 'William K.L. Dickson' })
[ {
  _id: ObjectId('573a1390f29313caabcd4135'),
  plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
  genres: [ 'Short' ],
  runtime: 1,
  cast: [ 'Charles Kayser', 'John Ott' ],
  num_mflix_comments: 1,
  title: 'Blacksmith Scene',
  fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
  countries: [ 'USA' ],
  released: ISODate('1893-05-09T00:00:00.000Z'),
  directors: [ 'William K.L. Dickson' ],
  rated: 'UNRATED',
  awards: { wins: 1, nominations: 0, text: '1 win.' },
  lastupdated: '2015-08-26 00:03:50.133000000',
  year: 1893,
  imdb: { rating: 6.2, votes: 1189, id: 5 },
  type: 'movie',
  tomatoes: {
    viewer: { rating: 3, numReviews: 184, meter: 32 },
    lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
  }
}
]
ashmita_31>
```

5.) Retrieve all movies from the 'movies' collection that were released in the USA and include complete information for each movie.

QUERY:

```
db.movies.find({ countries: 'USA' })
```

OUTPUT:

```
ashmita_31> db.movies.find({ countries: 'USA' })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kayser', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  }
]
ashmita_31>
```

6.) Retrieve all movies from the 'movies' collection that have complete information and are rated as "UNRATED".

QUERY:

```
db.movies.find({ rated: 'UNRATED' })
```

OUTPUT:

```
ashmita_31> db.movies.find({ rated: 'UNRATED' })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kayser', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  }
]
```

7.) Retrieve all movies from the 'movies' collection that have complete information and have received more than 1000 votes on IMDb.

QUERY:

```
db.movies.find({ 'imdb.votes': { $gt: 1000 } })
```

OUTPUT:

```
ashmita_31> db.movies.find({ 'imdb.votes': { $gt: 1000 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kayser', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  },
  {
    _id: ObjectId('573a1390f22313caabcd5967'),
    plot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires',
    genres: [ 'Action', 'Adventure', 'Crime' ],
    runtime: 399,
    rated: 'NOT RATED',
    cast: [ 'Musidora', 'édouardMathé', 'Marcel Lèvesque', 'Jean Aymé' ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTC1NTY3NDIzM15BanBnXkFtZTgwNTIyODg5MTE@._V1_SY1000_SX677_AL_.jpg',
    title: 'Les vampires',
    fullplot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires',
    languages: [ 'French' ],
    released: ISODate('1916-11-23T00:00:00.000Z'),
    directors: [ 'Louis Feuillade' ],
    writers: [ 'Louis Feuillade' ],
    awards: { wins: 0, nominations: 1, text: '1 nomination.' },
    lastupdated: '2015-09-02 00:24:27.333000000',
    year: 1915,
    imdb: { rating: 6.8, votes: 2878, id: 6206 },
    countries: [ 'France' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.8, numReviews: 2118, meter: 82 },
      dvd: ISODate('2000-05-16T00:00:00.000Z'),
      critic: { rating: 8.8, numReviews: 13, meter: 100 },
      lastUpdated: ISODate('2015-09-15T17:02:33.000Z'),
      rotten: 0,
      fresh: 13
    }
  }
]
ashmita_31>
```

8.) Retrieve all movies from the 'movies' collection that have complete information and have an IMDb rating higher than 7.

QUERY:

```
db.movies.find({ 'imdb.rating': { $gt: 7 } })
```

OUTPUT:

```

ashmita_31> db.movies.find({ 'imdb.rating': { $gt: 7 } })
[
  {
    _id: ObjectId('573a1391f29313caabcd7a34'),
    plot: 'A kept woman runs into her one-time fianc  and finds herself torn between love and comfort.',
    genres: [ 'Drama', 'Romance' ],
    runtime: 78,
    rated: 'TV-PG',
    cast: [
      'Edna Purviance',
      'Clarence Geldart',
      'Carl Miller',
      'Lydia Knott'
    ],
    num_mflix_comments: 3,
    poster: 'https://m.media-amazon.com/images/M/MV5BZjjiMTU2NGQtNWRkNi00ZjExLWxMTUtMmNkNTU0NzRlMTA3XkEyXkFqcGdeQXVyNjUwNzk3NDE@._V1_SV1000_SX677_AL_.jpg',
    title: 'A Woman of Paris: A Drama of Fate',
    fullplot: 'Marie St. Clair believes she has been jilted by her artist fiance Jean when he fails to meet her at the railway station. She goes off to Paris alone. A year later, mistress of wealthy Pierre Revel, she meets Jean again. Misinterpreting events she bounces back and forth between apparent security and true love.',
    countries: [ 'USA' ],
    released: ISODate('1923-11-04T00:00:00.000Z'),
    directors: [ 'Charles Chaplin' ],
    writers: [ 'Charles Chaplin' ],
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-09-02 00:22:09.303000000',
    year: 1923,
    imdb: { rating: 7.1, votes: 3179, id: 14624 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 886, meter: 78 },
      dvd: ISODate('2004-03-02T00:00:00.000Z'),
      critic: { rating: 7.4, numReviews: 11, meter: 91 },
      lastUpdated: ISODate('2015-08-23T18:34:44.000Z'),
      rotten: 1,
      production: 'Criterion Collection',
      fresh: 10
    }
  },
  {
    _id: ObjectId('573a1391f29313caabcd8945'),
    plot: 'A married farmer falls under the spell of a slatternly woman from the city, who tries to convince him to draw on his wife.',
    genres: [ 'Drama', 'Romance' ],
    runtime: 94,
    rated: 'NOT RATED',
    cast: [
      'George O'Brien',
      'Janet Gaynor',
      'Margaret Livingston',
      'Bodil Rosing'
    ],
    num_mflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BNDVkyMYW2ItNzRiMy00NMQ4LTlhMjMtNDI1ZDYyOGVmMzJjXkEyXkFqcGdeQXVyNTgzMzUSMDI@._V1_SV1000_SX677_AL_.jpg',
    title: 'Sunrise',
    fullplot: 'In this fable-morality subtitled "A Song of Two Humans", the "evil" temptress is a city woman who bewitches farmer Anses and tries to convince him to murder his neglected wife, Indre.',
    countries: [ 'USA' ],
    released: ISODate('1927-11-04T00:00:00.000Z'),
    directors: [ 'F.W. Murnau' ],
    writers: [
      'Carl Mayer (scenario)',
      'Hermann Sudermann (from an original theme by)',
      'Katherine Hilliker (titles)',
      'H.H. Caldwell (titles)'
    ],
    awards: {
      wins: 5,
      nominations: 1,
      text: 'Won 3 Oscars. Another 2 wins & 1 nomination.'
    },
    lastupdated: '2015-09-12 00:26:13.493000000',
    year: 1927,
    imdb: { rating: 8.4, votes: 24480, id: 18455 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 4.4, numReviews: 9134, meter: 92 },
      dvd: ISODate('2008-12-09T00:00:00.000Z'),
      critic: { rating: 8.9, numReviews: 48, meter: 98 },
      lastUpdated: ISODate('2015-09-10T19:15:02.000Z'),
      consensus: 'Boasting masterful cinematography to match its well-acted, wonderfully romantic storyline, Sunrise is perhaps the final -- and arguably definitive -- statement of the silent era.',
      rotten: 1,
      production: 'Fox Films',
      fresh: 47
    }
  }
]
ashmita_31>

```

9.) Retrieve all movies from the 'movies' collection that have complete information and have a viewer rating higher than 4 on Tomatoes.

QUERY:

```
db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
```

OUTPUT:

```
ashmita_31> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
[
  {
    _id: ObjectId('573a1391f29313caabcd8945'),
    plot: 'A married farmer falls under the spell of a slatternly woman from the city, who tries to convince him to drown his wife.',
    genres: [ 'Drama', 'Romance' ],
    runtime: 94,
    rated: 'NOT RATED',
    cast: [
      "George O'Brien",
      'Janet Gaynor',
      'Margaret Livingston',
      'Bodil Rosing'
    ],
    num_mflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BNDVkYmYwM2ItNzRiMy00NWQ4LTlhMjMtNDI1ZDYyOGVmMzJjXkEyXkFqcGdeQXVyNTExMzU5MDI@._V1_SY1000_SX677_AL_.jpg',
    title: 'Sunrise',
    fullplot: 'In this fable-morality subtitled "A Song of Two Humans", the "evil" temptress is a city woman who bewitches farmer Anses and tries to convince him to murder his neglected wife, Indre.',
    countries: [ 'USA' ],
    released: ISODate('1927-11-04T00:00:00.000Z'),
    directors: [ 'F.W. Murnau' ],
    writers: [
      'Carl Mayer (scenario)',
      'Hermann Sudermann (from an original theme by)',
      'Katherine Hilliker (titles)',
      'H.H. Caldwell (titles)'
    ],
    awards: {
      wins: 5,
      nominations: 1,
      text: 'Won 3 Oscars. Another 2 wins & 1 nomination.'
    },
    lastupdated: '2015-09-12 00:26:13.493000000',
    year: 1927,
    imdb: { rating: 8.4, votes: 24480, id: 18455 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 4.4, numReviews: 9134, meter: 92 },
      dvd: ISODate('2008-12-09T00:00:00.000Z'),
      critic: { rating: 8.9, numReviews: 48, meter: 98 },
      lastUpdated: ISODate('2015-09-10T19:15:02.000Z'),
      consensus: 'Boasting masterful cinematography to match its well-acted, wonderfully romantic storyline, Sunrise is perhaps the final -- and arguably definitive -- statement of the silent era.',
      rotten: 1,
      production: 'Fox Films',
      fresh: 47
    }
  }
]
ashmita_31>
```

10.) Retrieve all movies from the 'movies' collection that have received an award.

QUERY:

```
db.movies.find({ 'awards.wins': { $gt: 0 } })
```

OUTPUT:

```

ashmita_31> db.movies.find({ 'awards.wins': { $gt: 0 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kaysen', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  },
  {
    _id: ObjectId('573a1391f29313caabcd7a34'),
    plot: 'A kept woman runs into her one-time fianc  and finds herself torn between love and comfort.',
    genres: [ 'Drama', 'Romance' ],
    runtime: 78,
    rated: 'TV-PG',
    cast: [
      'Edna Purviance',
      'Clarence Geldart',
      'Carl Miller',
      'Lydia Knott'
    ],
    num_mflix_comments: 3,
    poster: 'https://m.media-amazon.com/images/M/MV5BZjJiMTU2NGtNWRkNi00ZjExLWExMTUtMmNkNTU0NzRlMTA3XkEyXkFqcGdeQXVyMjUwNzklNDc@._V1_Sy1000_SX677_AL_.jpg',
    title: 'A Woman of Paris: A Drama of Fate',
    fullplot: 'Marie St. Clair believes she has been jilted by her artist fiance Jean when he fails to meet her at the railway station. She goes off to Paris alone. A year later, mistress of wealthy Pierre Revel, she meets Jean again. Misinterpreting events she bounces back and forth between apparent security and true love.',
    countries: [ 'USA' ],
    released: ISODate('1923-11-04T00:00:00.000Z'),
    directors: [ 'Charles Chaplin' ],
    writers: [ 'Charles Chaplin' ],
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-09-02 00:22:09.303000000',
    year: 1923,
    imdb: { rating: 7.1, votes: 3179, id: 14624 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 896, meter: 78 },
      dvd: ISODate('2004-03-02T00:00:00.000Z'),
      critic: { rating: 7.4, numReviews: 11, meter: 91 },
      lastUpdated: ISODate('2015-08-23T18:34:44.000Z'),
      rotten: 1,
      production: 'Criterion Collection',
      fresh: 10
    }
  },
  {
    _id: ObjectId('573a1391f29313caabcd8945'),
    plot: 'A married farmer falls under the spell of a slatternly woman from the city, who tries to convince him to drown his wife.',
    genres: [ 'Drama', 'Romance' ],
    runtime: 94,
    rated: 'NOT RATED',
    cast: [
      'George O'Brien',
      'Janet Gaynor',
      'Margaret Livingston',
      'Bodie Rosing'
    ],
    num_mflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BNDVYmYmM2ITNzRiMyOBNWQ4LT1hmjMtnDI1ZDYyOGVmMzJjXkEyXkFqcGdeQXVyNTgzMzU5MDI@._V1_Sy1000_SX677_AL_.jpg',
    title: 'Sunrise',
    fullplot: 'In this fable-morality subtitled "A Song of Two Humans", the "evil" temptress is a city woman who bewitches farmer Anses and tries to convince him to murder his neglected wife, Indre.',
    countries: [ 'USA' ],
    released: ISODate('1927-11-04T00:00:00.000Z'),
    directors: [ 'F.W. Murnau' ],
    writers: [
      'Carl Mayer (scenario)',
      'Hermann Sudermann (from an original theme by)',
      'Katherine Hilliker (titles)',
      'H.H. Caldwell (titles)'
    ],
    awards: {
      wins: 5,
      nominations: 1,
      text: 'Won 3 Oscars. Another 2 wins & 1 nomination.'
    },
    lastupdated: '2015-09-12 00:26:13.493000000',
    year: 1927,
    imdb: { rating: 8.4, votes: 24480, id: 18455 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 4.4, numReviews: 9134, meter: 92 },
      dvd: ISODate('2008-12-09T00:00:00.000Z'),
      critic: { rating: 8.9, numReviews: 48, meter: 98 },
      lastUpdated: ISODate('2015-09-10T19:15:02.000Z'),
      consensus: 'Boasting masterful cinematography to match its well-acted, wonderfully romantic storyline, Sunrise is perhaps the final -- and arguably definitive -- statement of the silent era.',
      rotten: 1,
      production: 'Fox Films',
      fresh: 47
    }
  }
]
ashmita_31>

```

11.) Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB that have at least one nomination.

QUERY:

```
db.movies.find( { 'awards.nominations': { $gt: 0 } }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 } )
```

OUTPUT:

```
ashmita_31> db.movies.find( { 'awards.nominations': { $gt: 0 } }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 } )
[
  {
    _id: ObjectId('573a1390f22313caabcd5967'),
    genres: [ 'Action', 'Adventure', 'Crime' ],
    runtime: 399,
    cast: [ 'Musidora', 'édouardMathè', 'Marcel Lèvesque', 'Jean Aymè' ],
    title: 'Les vampires',
    languages: [ 'French' ],
    released: ISODate('1916-11-23T00:00:00.000Z'),
    directors: [ 'Louis Feuillade' ],
    writers: [ 'Louis Feuillade' ],
    awards: { wins: 0, nominations: 1, text: '1 nomination.' },
    year: 1915,
    countries: [ 'France' ]
  },
  {
    _id: ObjectId('573a1391f29313caabcd8945'),
    genres: [ 'Drama', 'Romance' ],
    runtime: 94,
    cast: [
      "George O'Brien",
      'Janet Gaynor',
      'Margaret Livingston',
      'BodilRosing'
    ],
    title: 'Sunrise',
    countries: [ 'USA' ],
    released: ISODate('1927-11-04T00:00:00.000Z'),
    directors: [ 'F.W. Murnau' ],
    writers: [
      'Carl Mayer (scenario)',
      'Hermann Sudermann (from an original theme by)',
      'Katherine Hilliker (titles)',
      'H.H. Caldwell (titles)'
    ],
    awards: {
      wins: 5,
      nominations: 1,
      text: 'Won 3 Oscars. Another 2 wins & 1 nomination.'
    },
    year: 1927
  }
]
ashmita_31>
```

12.) Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB with cast including "Charles Kayser".

QUERY:

```
db.movies.find( { cast: 'Charles Kayser' }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })
```

OUTPUT:

```
ashmita_31> db.movies.find( { cast: 'Charles Kayser' }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })
[ {
  _id: ObjectId('573a1390f29313caabcd4135'),
  genres: [ 'Short' ],
  runtime: 1,
  cast: [ 'Charles Kayser', 'John Ott' ],
  title: 'Blacksmith Scene',
  countries: [ 'USA' ],
  released: ISODate('1893-05-09T00:00:00.000Z'),
  directors: [ 'William K.L. Dickson' ],
  awards: { wins: 1, nominations: 0, text: '1 win.' },
  year: 1893
}
]
ashmita_31> ■
```

13.) Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that released on May 9, 1893.

QUERY:

```
db.movies.find( { released: ISODate("1893-05-09T00:00:00.000Z") }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )
```

OUTPUT:

```
ashmita_31> db.movies.find( { released: ISODate("1893-05-09T00:00:00.000Z") }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    title: 'Blacksmith Scene',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ]
  }
]
ashmita_31> -
```

14.) Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that have a word "scene" in the title.

QUERY:

```
db.movies.find( { title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })
```

OUTPUT:

```
ashmita_31> db.movies.find( { title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })
[{
  "_id": ObjectId('573a1390f29313caabcd4135'),
  "title": "Blacksmith Scene",
  "countries": [ "USA" ],
  "released": ISODate('1893-05-09T00:00:00.000Z'),
  "directors": [ "William K.L. Dickson" ]
}]
ashmita_31> .
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT: