

Creating and Managing Tables

Exp No:1

Date:20/02/2024

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 department( id i
```

OUTPUT:

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, 'APEX' and 'App Builder' are visible. Below the bar, there's a toolbar with icons for SQL Commands, Language (set to SQL), Rows (10), and other database operations. The main area has three tabs: 'SQL' (selected), 'Explain', and 'Describe'. The SQL tab contains the following code:

```
1 create table department( id i
2
3
```

Below the code, the message 'Table created.' is displayed.

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

3.Modify the EMP table to allow for longer employee last names. Confirm the modification.(Hint: Increase the size to 50)

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. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. A search bar and user profile 'Aishwarya Chozhan' are also present. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), and 'Clear Command'. Below this is a toolbar with icons for copy, paste, run, etc. The SQL editor contains the following code:

```
1 create table EMP(id number(7),last_name varchar2(25),first_name varchar2(25),dept_id number(7));
2
3
```

The 'Results' tab is selected, showing the output 'Table created.' and a execution time of '0.03 seconds'.

QUERY:

```
alter table EMP modify(last_name varchar(50));
```

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 profile "Aishwarya Chozhan" and the schema "WKSP_AISHWARYA17". The main area is titled "SQL Commands" with tabs for Language (set to SQL), Rows (set to 10), Clear Command, and Find Tables. Below these are icons for Undo, Redo, Search, and Paste. The SQL editor contains the following code:

```
1 alter table EMP modify(last_name varchar(50));
2
```

Below the code, the "Results" tab is selected, showing the output: "Table altered." and "0.05 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)not null,first_name varchar(20),last_name varchar(25)not null,salary number(8,2),dept_id number(6)not null);
```

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 the user profile "Aishwarya Chozhan" and the schema "WKSP_AISHWARYA17". The main area is titled "SQL Commands" with tabs for Language (set to SQL), Rows (set to 10), Clear Command, and Find Tables. Below these are icons for Undo, Redo, Search, and Paste. The SQL editor contains the following code:

```
1 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);
2
```

Below the code, the "Results" tab is selected, showing the output: "Table created." and "0.04 seconds".

5.Drop the EMP table.

QUERY:

```
drop table EMP
```

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: Aishwarya Chozhan (aishwarya17). The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to WKSP_AISHWARYA17. Below the search bar are buttons for Save and Run. The SQL editor contains the following code:

```
1 drop table EMP;
2
```

The 'Results' tab is selected, showing the output: "Table dropped." and "0.08 seconds".

6.Rename the EMPLOYEES2 table as EMP.

QUERY:

```
rename employees2 to EMP;
```

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 the user's profile: Aishwarya Chozhan (aishwarya17). The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to WKSP_AISHWARYA17. Below the search bar are buttons for Save and Run. The SQL editor contains the following code:

```
1 rename employees2 to EMP;
2
```

The 'Results' tab is selected, showing the output: "Statement processed." and "0.05 seconds".

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

QUERY:

```
comment on table dept is 'Department info';
comment on table emp is Employee info;
```

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 user profile for 'Aishwarya Chozhan' (aishwarya17). The main workspace is titled 'SQL Commands'. It features a toolbar with icons for Undo, Redo, Find, Replace, and Save. Below the toolbar, the schema is set to 'WKSP_AISHWARYAT7'. The SQL editor contains the following command:

```
1 comment on table department is 'Department info';
```

Below the editor, the results tab is selected, showing the output of the command:

```
Statement processed.
0.01 seconds
```

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

QUERY:

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

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:

```
1 alter table EMP drop column first_name;
2
```

Below the command, the output is shown:

Table altered.
0.07 seconds

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

Faculty Signature

RESULT:

MANIPULATING DATA

EX_NO : 2

DATE:23/02/2024

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. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is active. The main area is titled 'SQL Commands'. A single line of SQL code is entered in the command input field: '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));'. The 'Run' button is highlighted in green. Below the command, the results show 'Table created.' and a execution time of '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 APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, the user 'Aishwarya Chozhan' (aishwarya17) is logged in. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. The language is set to 'SQL' and rows are set to 10. The command entered is:

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

The results section shows the output of the query:

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

Execution time: 0.02 seconds.

3. Display the table with values.

QUERY:

```
select*from my_employee;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar and user information are identical to the previous screenshot. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. The language is set to 'SQL' and rows are set to 10. The command entered is:

```
1 select*from my_employee;
2
```

The results section displays the data from the 'my_employee' table:

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

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. In the SQL Commands pane, two INSERT statements are executed:

```
1 insert into my_employee values('3','biri','ben','bbiri','1100');  
2
```

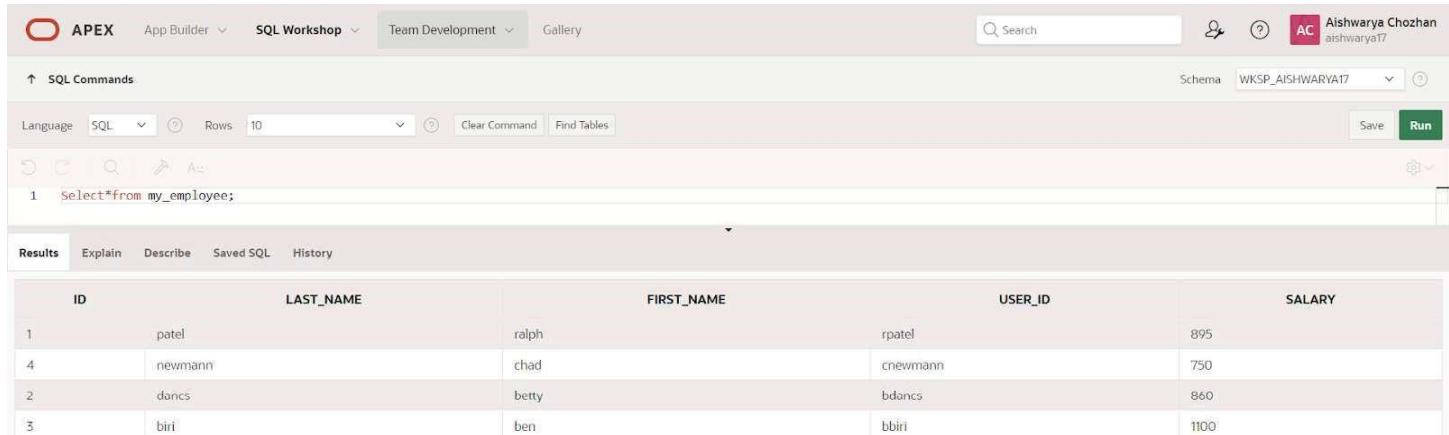
The Results pane displays the message "1 row(s) inserted." and a timestamp of "0.01 seconds".

5.Make the data additions permanent.

QUERY:

```
select*from my_employee;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands pane, a SELECT statement is run:

```
1 Select*from my_employee;
```

The Results pane displays the data from the my_employee table:

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

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. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name, Aishwarya Chozhan, and their schema, WKSP_AISHWARYA17. The main area is titled "SQL Commands". It contains a toolbar with icons for Undo, Redo, Find, Copy, Paste, and Run. Below the toolbar is a search bar and a "Schema" dropdown set to WKSP_AISHWARYA17. The SQL command entered is: `1 update my_employee set last_name='Drelex' where id=3;`. The results tab shows the output: `1 row(s) updated.` and `0.01 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, Team Development, and Gallery. The right side of the header shows the user's name, Aishwarya Chozhan, and their schema, WKSP_AISHWARYA17. The main area is titled "SQL Commands". It contains a toolbar with icons for Undo, Redo, Find, Copy, Paste, and Run. Below the toolbar is a search bar and a "Schema" dropdown set to WKSP_AISHWARYA17. The SQL command entered is: `1 update my_employee set salary='1000' where salary<900;`. The results tab shows the output: `3 row(s) updated.` and `0.01 seconds`.

8.Delete Betty dancs from MY _EMPLOYEE table.

QUERY:

delete from my_employee where first_name='betty';

OUTPUT:



A screenshot of 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 'Aishwarya Chozhan' are on the right. The main area shows a SQL command line with the following content:

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

The results tab shows the output:

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

Execution time: 0.02 seconds.

9.Empty the fourth row of the emp table.

QUERY:

delete from my_employee where id='4';

OUTPUT:



A screenshot of the Oracle SQL Workshop interface, identical to the previous one but with a different SQL command:

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

The results tab shows the output:

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

Execution time: 0.01 seconds.

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

RESULT:

INCLUDING CONSTRAINTS

EX_NO : 3

DATE:27/02/2024

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:

```
Create table emp(id int,last_name varchar(25),first_name varchar(25),dept_id int,constraint my_emp_id_pk primary key(id));
```

OUTPUT:

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

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

The output window displays the result:

```
Table altered.
```

Execution time: 0.07 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_id17_pk primary key(ID);
```

OUTPUT:



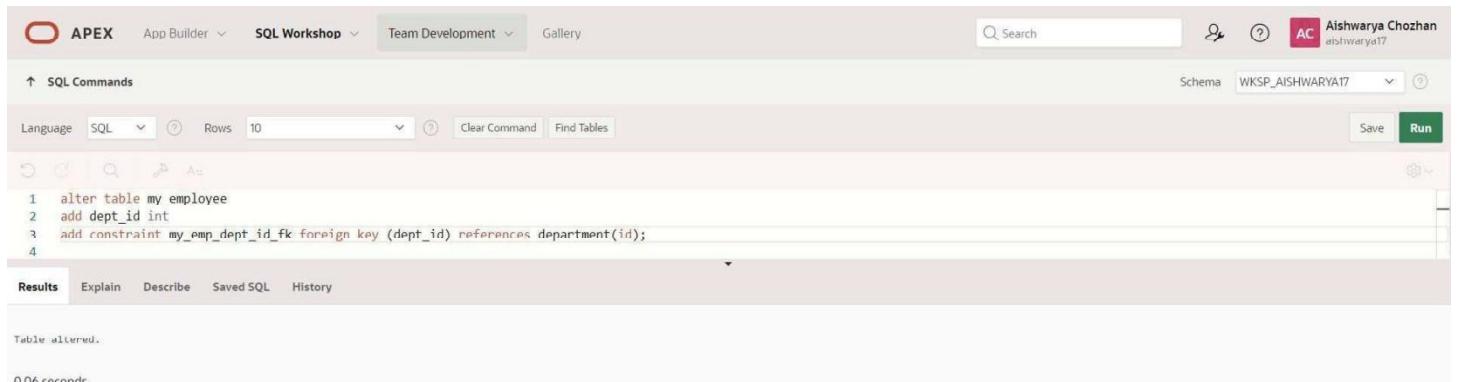
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows the user profile "Aishwarya Chozhan" and the schema "WKSP_AISHWARYA17". The main area is titled "SQL Commands" with tabs for Language (set to SQL), Rows (set to 10), Clear Command, and Find Tables. Below these are several icons for copy, paste, search, and refresh. The SQL editor contains the following command:
2 alter table department add constraint my_dept_id17_pk primary key(ID);
The results tab is selected, showing the output: "Table altered." and a timestamp of "0.07 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 my_employee
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 APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows the user profile "Aishwarya Chozhan" and the schema "WKSP_AISHWARYA17". The main area is titled "SQL Commands" with tabs for Language (set to SQL), Rows (set to 10), Clear Command, and Find Tables. Below these are several icons for copy, paste, search, and refresh. The SQL editor contains the following commands:
1 alter table my_employee
2 add dept_id int
3 add constraint my_emp_dept_id_fk foreign key (dept_id) references department(id);
4
The results tab is selected, showing the output: "Table altered." and a timestamp 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 SQL command: 'alter table EMP add commission number(2,2) constraint my_emp_com check(commission>0);'. Below the command, the output shows 'Table altered.' and a time of '0.06 seconds'. The results tab is active.

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:

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

OUTPUT :

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

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

In the Results pane, the output is displayed as a table:

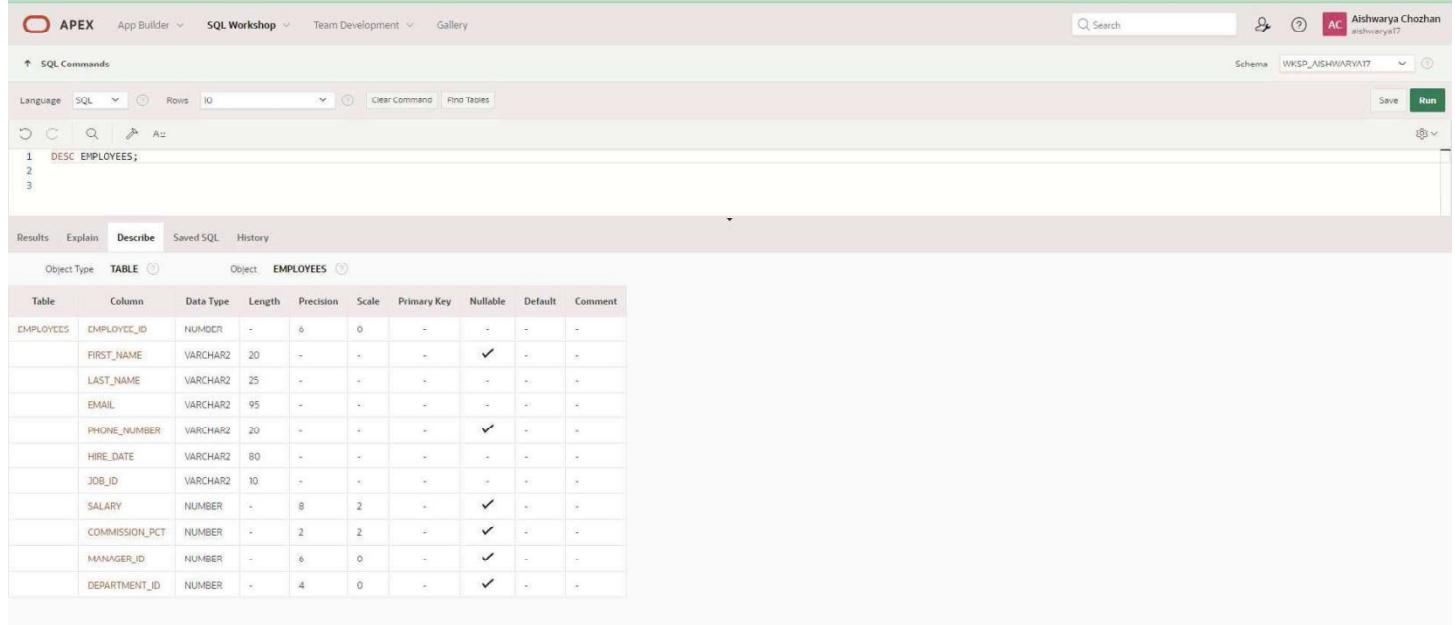
EMPLOYEE_ID	LAST_NAME	ANNUAL SALARY
2	SHRI	117609.84
12	EMANUEL	160801.44
312	JAAGADEESH	160801.44
3	SHRI	21617.04
1	KUMAR	40801.44
176	PARTHI	211809.84

6 rows returned in 0.00 seconds

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

QUERY:
DESC EMPLOYEES;

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, the 'SQL Workshop' tab is active. The main area contains the following SQL command:

```
1 DESC EMPLOYEES;
2
3
```

Below the command, the 'Describe' tab is selected, showing the structure of the 'EMPLOYEES' table. The table has 10 columns:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEES	EMPLOYEE_ID	NUMBER	-	6	0	-	-	-	-
	FIRST_NAME	VARCHAR2	20	-	-	-	✓	-	-
	LAST_NAME	VARCHAR2	25	-	-	-	-	-	-
	EMAIL	VARCHAR2	95	-	-	-	-	-	-
	PHONE_NUMBER	VARCHAR2	20	-	-	-	✓	-	-
	HIRE_DATE	VARCHAR2	80	-	-	-	-	-	-
	JOB_ID	VARCHAR2	10	-	-	-	-	-	-
	SALARY	NUMBER	-	8	2	-	✓	-	-
	COMMISSION_PCT	NUMBER	-	2	2	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-
	DEPARTMENT_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 with the 'Results' tab selected. The results of the query are displayed in a table:

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
2	SHRI	67	1999-08-02
12	EMANUEL	ST_CLERK	1994-05-01
312	JAAGADEESH	SA_REP	1998-03-15
3	SHRI	67	1998-07-01
1	KUMAR	456	1999-05-01
176	PARTHI	90	1999-01-04

At the bottom left, it says '6 rows returned in 0.01 seconds'. At the bottom right, there is 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. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the query is displayed:

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

Under 'Results', the output shows the column 'STARTDATE' with the following values:

STARTDATE
1999-08-02
1994-05-01
1998-03-15
1996-07-01
1999-05-01
1999-01-04

At the bottom left, it says '6 rows returned in 0.00 seconds'. There are 'Download' and 'Run' buttons at the bottom right.

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'. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the query is displayed:

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

Under 'Results', the output shows the column 'STARTDATE' with the following values:

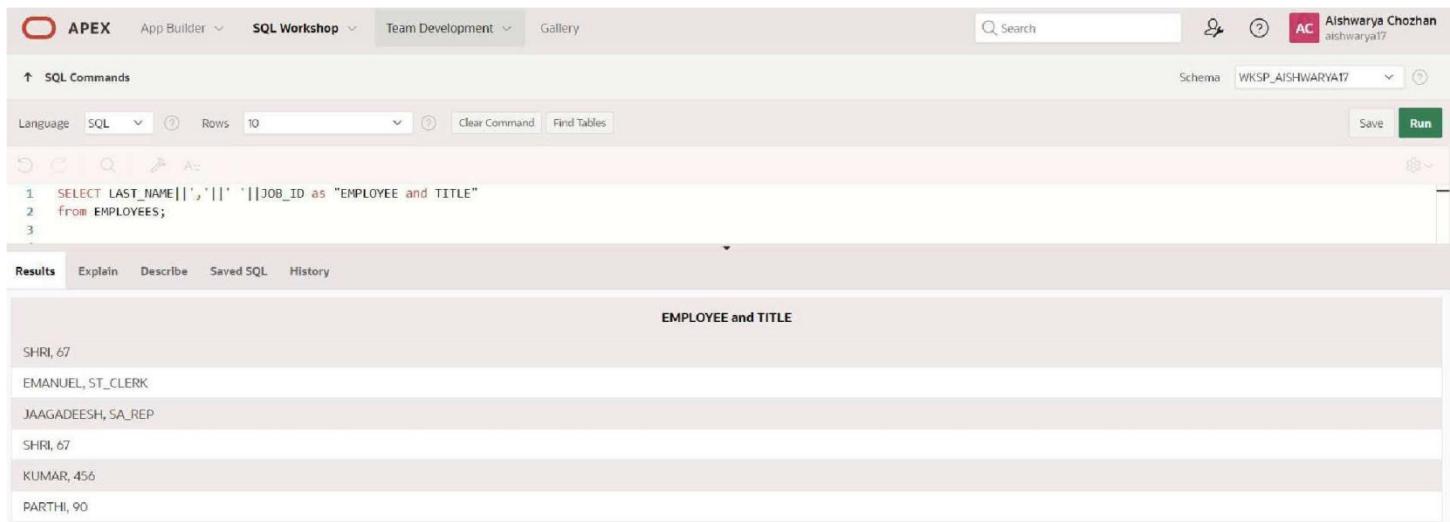
STARTDATE
1999-08-02
1994-05-01
1998-03-15
1996-07-01
1999-05-01
1999-01-04

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. On the right, there's a search bar, user information for 'Aishwarya Chozhan' (aichwarya17), and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area shows a SQL command window with the following content:

```
1 SELECT LAST_NAME||','||' '||JOB_ID as "EMPLOYEE and TITLE"  
2 from EMPLOYEES;  
3
```

Below the command window, the results tab is selected, displaying the output:

EMPLOYEE and TITLE
SHRI, 67
EMANUEL, ST_CLERK
JAAGADEESH, SA REP
SHRI, 67
KUMAR, 456
PARTHI, 90

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. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and session details 'WKSP_AISHWARYA17'. The main workspace has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
1 SELECT EMPLOYEE_ID||','||FIRST_NAME||','||LAST_NAME||','||EMAIL||','||PHONE_NUMBER||','||HIRE_DATE||','||  
2 JOB_ID||','||SALARY||','||COMMISSION_PCT||','||MANAGER_ID||','||DEPARTMENT_ID AS "THE_OUTPUT"  
3 FROM EMPLOYEES;  
4  
5
```

The Results tab displays the output of the query, with the column header 'THE_OUTPUT' and the following data rows:

THE_OUTPUT
2,ANU,SHRI,423@gmail.com,789,1999-08-02,67,9800.82,.34,.22,.982
12,SAHA,EMANUEL,123@gmail.com,1237,1994-05-01,ST_CLERK,13400.12,.9,.432
312,MARY,JAAGADEESH,123@gmail.com,1237,1998-03-15,SA_REP,13400.12,.2,.30
3,KAVITHA,SHRI,788@gmail.com,899,1996-07-01,67,1801.42,.56,.64,.582
1,PRIYA,KUMAR,123@gmail.com,1237,1999-05-01,456,3400.12,.12,.56,.432
176,SANJ,PARTHI,789@gmail.com,134,1999-01-04,90,17650.82,.23,.71,.999

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

RESULT:

RESTRICTING AND SORTING DATA

EX.NO:5

DATE:5/3/2024

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 SQL Commands tab is active, displaying the following query:

```
1 SELECT LAST_NAME, SALARY  
2 FROM EMPLOYEES  
3 WHERE SALARY > 12000;  
4
```

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

LAST_NAME	SALARY
EMANUEL	13400.12

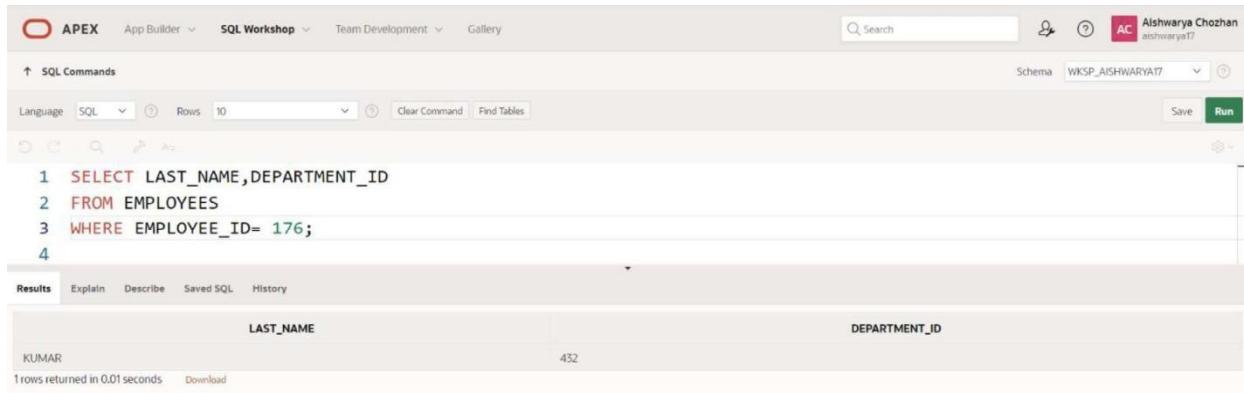
Below the table, it says "1 rows returned in 0.01 seconds".

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 top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Aishwarya Chozhan' (aishwaryach@T). The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the following query:

```
1 SELECT LAST_NAME,DEPARTMENT_ID  
2 FROM EMPLOYEES  
3 WHERE EMPLOYEE_ID= 176;  
4
```

The Results tab shows the output of the query:

LAST_NAME	DEPARTMENT_ID
KUMAR	432

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

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 SQL command entered is:

```
1 SELECT LAST_NAME,SALARY
2 FROM EMPLOYEES
3 WHERE SALARY NOT BETWEEN 5000 AND 12000;
4
```

The results table displays three rows of data:

LAST_NAME	SALARY
KUMAR	5400.12
SHRI	1801.42
EMANUEL	13400.12

3 rows returned in 0.01 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,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 user profile for 'Aishwarya Chozhan' (aishwaryat). The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the following code:

```
1 SELECT LAST_NAME,JOB_ID,HIRE_DATE  
2 FROM EMPLOYEES  
3 WHERE HIRE_DATE BETWEEN TO_DATE('1998-02-20','YYYY-MM-DD') AND TO_DATE('1998-05-01','YYYY-MM-DD')  
4 ORDER BY HIRE_DATE;  
5
```

The Results tab shows the output of the query:

LAST_NAME	JOB_ID	HIRE_DATE
JAAGADEESH	SA_REP	03/15/1998

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 query entered is:

```
1 SELECT LAST_NAME, DEPARTMENT_ID
2 FROM EMPLOYEES
3 WHERE DEPARTMENT_ID IN (20, 50)
4 ORDER BY LAST_NAME;
5
```

The results table has two columns: LAST_NAME and DEPARTMENT_ID. The data row is:

LAST_NAME	DEPARTMENT_ID
JAAGADEESH	20

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 query entered is:

```
1 SELECT LAST_NAME AS "EMPLOYEE", department_id AS "MONTHLY_SALARY"
2 FROM EMPLOYEES
3 WHERE SALARY BETWEEN 5000 AND 12000 AND DEPARTMENT_ID IN (20, 50)
4 ORDER BY LAST_NAME;
5
```

The results table has two columns: EMPLOYEE and MONTHLY_SALARY. The data row is:

EMPLOYEE	MONTHLY_SALARY
JAAGADEESH	20

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 query entered is:

```
1 SELECT LAST_NAME,HIRE_DATE  
2 FROM EMPLOYEES  
3 WHERE HIRE_DATE LIKE '%94';  
4
```

The results table has two columns: LAST_NAME and HIRE_DATE. One row is returned:

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 query entered is:

```
1 SELECT LAST_NAME,JOB_ID  
2 FROM EMPLOYEES  
3 WHERE MANAGER_ID IS NULL;  
4
```

The results table has two columns: LAST_NAME and JOB_ID. Two rows are returned:

LAST_NAME	JOB_ID
JAAGADEESH	SA_REP
EMANUEL	ST_CLERK

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 query window contains the following SQL code:

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

The results pane displays the following data:

LAST_NAME	SALARY	COMMISSION_PCT
EMANUEL	13400.12	.9
JAAGADEESH	11000.12	.2
SHRI	9800.82	.34
KUMAR	5400.12	.12
SHRI	1801.42	.56

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 window contains the following SQL code:

```
1 SELECT LAST_NAME  
2 FROM EMPLOYEES  
3 WHERE LAST_NAME LIKE '__A%';
```

The results pane displays the following data:

LAST_NAME
JAAGADEESH
EMANUEL

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 entered is:

```
1 SELECT LAST_NAME  
2 FROM EMPLOYEES  
3 WHERE LAST_NAME LIKE '%A%' AND LAST_NAME LIKE '%E%';  
4
```

The results table has a single column labeled "LAST_NAME" with two rows: "JAAGADEESH" and "EMANUEL".

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 entered is:

```
1 SELECT LAST_NAME,JOB_ID,SALARY  
2 FROM EMPLOYEES  
3 WHERE JOB_ID='SA_REP' OR JOB_ID='ST_CLERK'  
4 AND salary NOT IN (2500, 3500, 7000);  
5
```

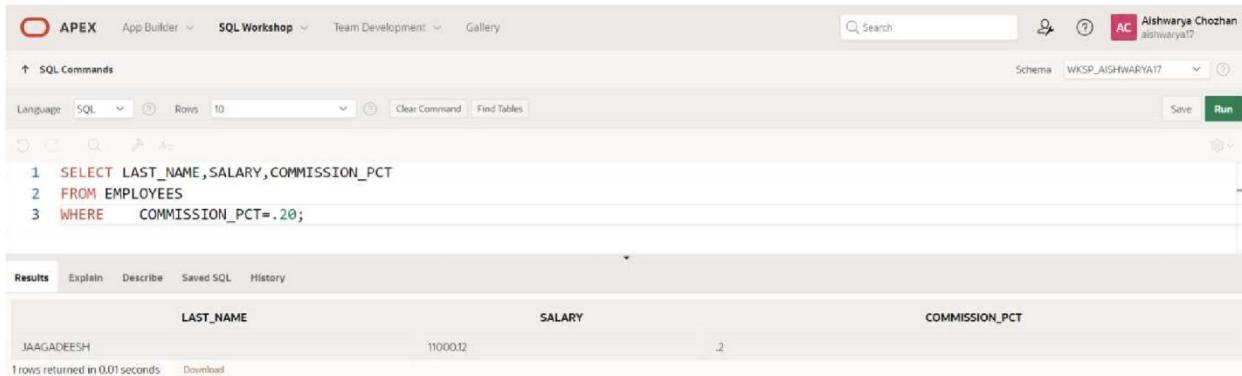
The results table has three columns: "LAST_NAME", "JOB_ID", and "SALARY". It contains two rows: one for "JAAGADEESH" with "SA_REP" and "11000.12", and another for "EMANUEL" with "ST_CLERK" and "13400.12".

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 top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user's profile: Aishwarya Chozhan (aishwarya17). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

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

The Results tab displays the output:

LAST_NAME	SALARY	COMMISSION_PCT
JAAGADEESH	11000.12	.2

1 rows returned in 0.01 seconds. There is a Download button.

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

RESULT:

SINGLE ROW FUNCTIONS

EX-NO : 6

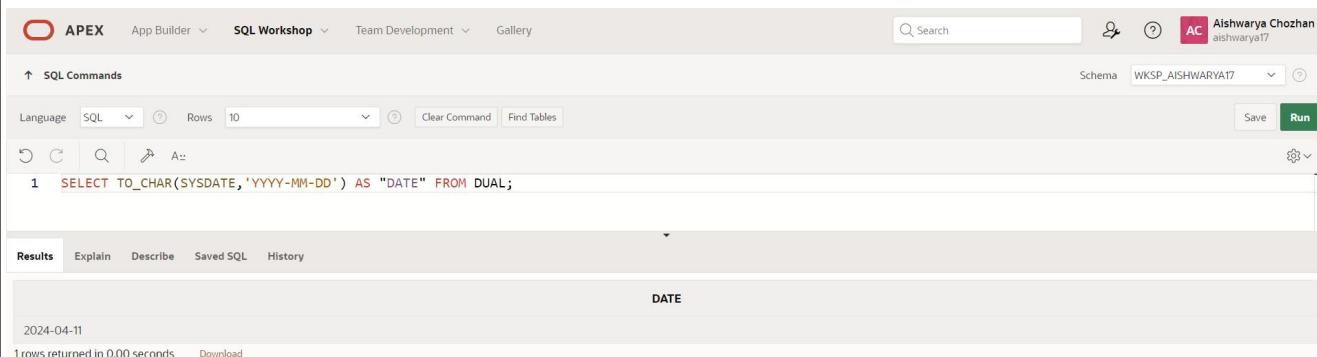
DATE:12/03/2024

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 SQL command entered is:

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

The results pane displays the output:

DATE
2024-04-11

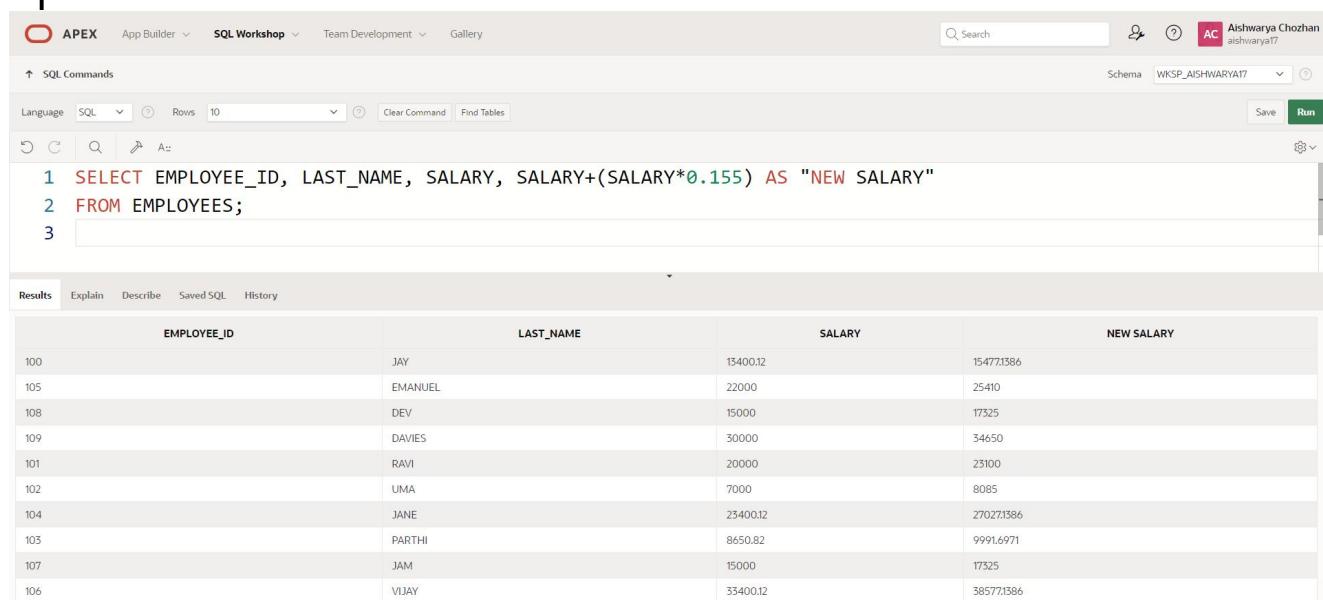
1 rows returned in 0.00 seconds

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 APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and a 'Run' button. Below the toolbar, the SQL command is displayed:

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

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

EMPLOYEE_ID	LAST_NAME	SALARY	NEW SALARY
100	JAY	15400.12	154771386
105	EMANUEL	22000	25410
108	DEV	15000	17525
109	DAVIES	30000	34650
101	RAVI	20000	23100
102	UMA	7000	8085
104	JANE	23400.12	270271386
105	PARTHI	8650.82	9991.6971
107	JAM	15000	17525
106	VIJAY	35400.12	385771386

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 (selected), Team Development, and Gallery. The right side shows user information: Aishwarya Chozhan, ashwaryat7, and a schema dropdown for WKSP_AISHWARAYA17. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the following query:

```
1 SELECT EMPLOYEE_ID, LAST_NAME, SALARY, SALARY+(SALARY*15.5/100) AS "NEW SALARY", (SALARY+(SALARY*15.5/100))-SALARY AS "INCREASE"  
2 FROM EMPLOYEES;  
3
```

The Results tab shows the output of the query:

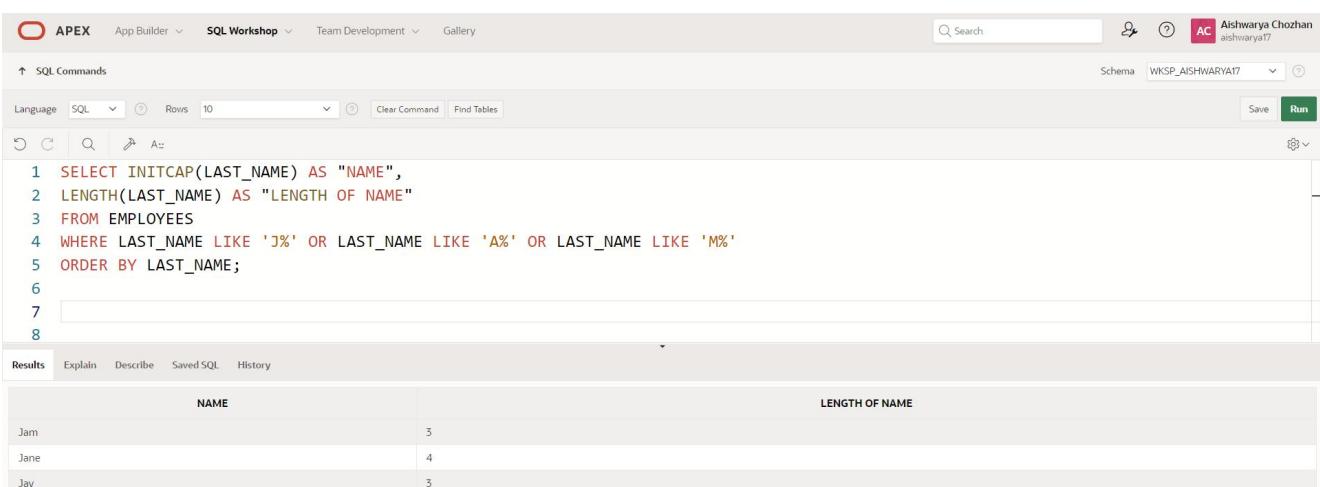
EMPLOYEE_ID	LAST_NAME	SALARY	NEW SALARY	INCREASE
100	JAY	13400.12	15477.1386	2077.0186
105	EMANUEL	22000	25410	3410
108	DEV	15000	17325	2325
109	DAVIES	30000	34650	4650
101	RAVI	20000	23100	3100
102	UMA	7000	8085	1085
104	JANE	23400.12	27027.1386	3627.0186
103	PARTHI	8650.82	9991.6971	1340.8771
107	JAM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

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 links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name, Aishwarya Chozhan, and their schema, WKSP_AISHWARYA17. The main workspace contains the SQL command entered by the user:

```
1 SELECT INITCAP(LAST_NAME) AS "NAME",
2 LENGTH(LAST_NAME) AS "LENGTH OF NAME"
3 FROM EMPLOYEES
4 WHERE LAST_NAME LIKE 'J%' OR LAST_NAME LIKE 'A%' OR LAST_NAME LIKE 'M%'
5 ORDER BY LAST_NAME;
6
7
8
```

Below the command, the 'Results' tab is selected, showing the output of the query:

NAME	LENGTH OF NAME
Jam	3
Jane	4
Jay	3

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 APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. The main workspace displays the SQL command and its execution results.

SQL Commands

```
1 SELECT INITCAP(LAST_NAME) AS "NAME", LENGTH(LAST_NAME) AS "LENGTH OF NAME"  
2 FROM EMPLOYEES  
3 WHERE LAST_NAME LIKE CONCAT(:NAME, '%')  
4 ORDER BY LAST_NAME;  
5
```

Results

NAME	LENGTH OF NAME
Harsh	5

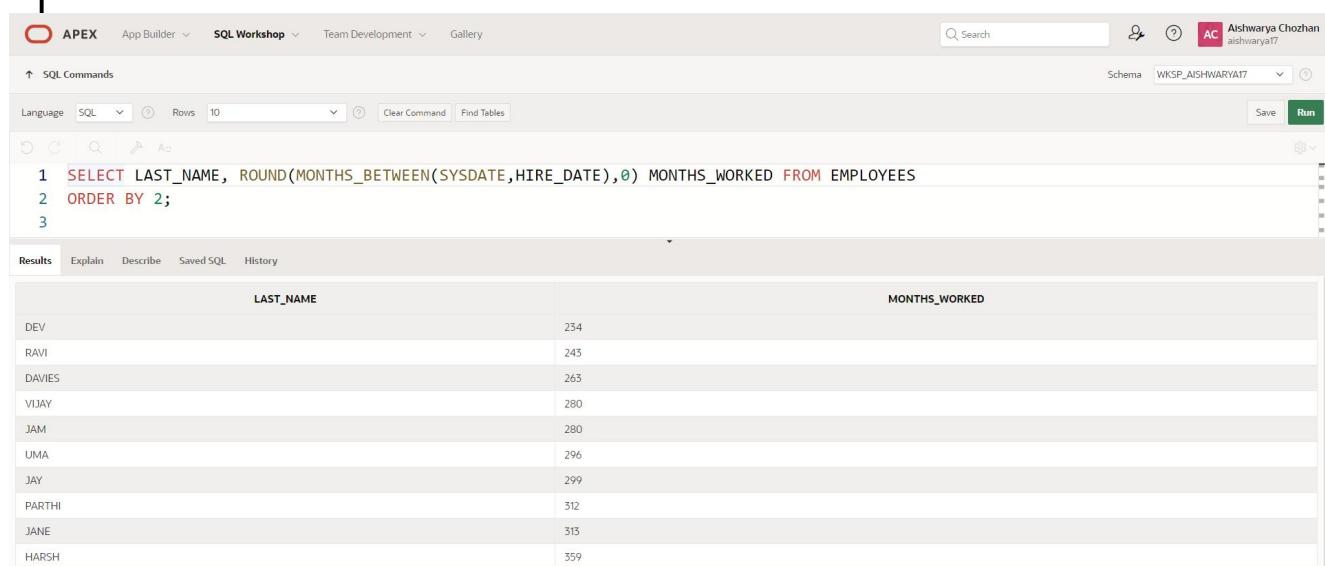
1 rows returned in 0.01 seconds Download

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, and Gallery. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
1 SELECT LAST_NAME, ROUND(MONTHS_BETWEEN(SYSDATE,HIRE_DATE),0) MONTHS_WORKED FROM EMPLOYEES
2 ORDER BY 2;
3
```

The Results tab displays the output of the query:

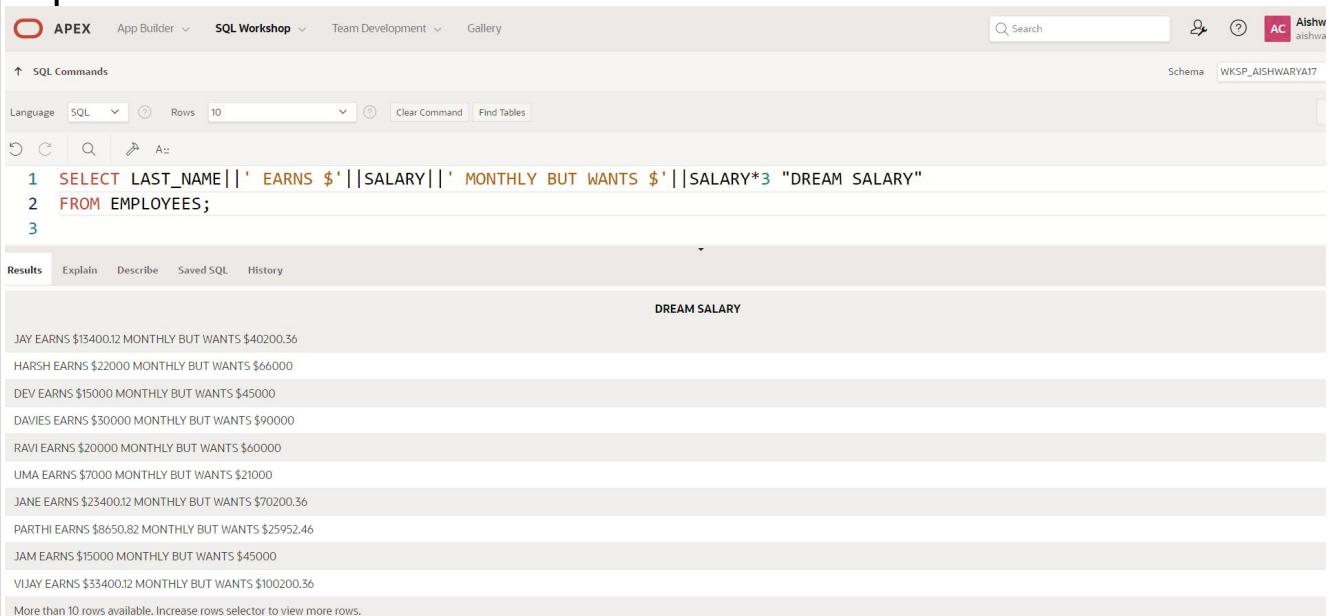
LAST_NAME	MONTHS_WORKED
DEV	234
RAVI	243
DAVIES	265
VIJAY	280
JAM	280
UMA	296
JAY	299
PARTHI	312
JANE	313
HARSH	359

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, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya17', and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands'. It shows the following code:

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

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

DREAM SALARY
JAY EARNS \$15400.12 MONTHLY BUT WANTS \$40200.36
HARSH EARNS \$22000 MONTHLY BUT WANTS \$66000
DEV EARNS \$15000 MONTHLY BUT WANTS \$45000
DAVIES EARNS \$30000 MONTHLY BUT WANTS \$90000
RAVI EARNS \$20000 MONTHLY BUT WANTS \$60000
UMA EARNS \$7000 MONTHLY BUT WANTS \$21000
JANE EARNS \$23400.12 MONTHLY BUT WANTS \$70200.36
PARTHI EARNS \$8650.82 MONTHLY BUT WANTS \$25952.46
JAM EARNS \$15000 MONTHLY BUT WANTS \$45000
VIJAY EARNS \$33400.12 MONTHLY BUT WANTS \$100200.36

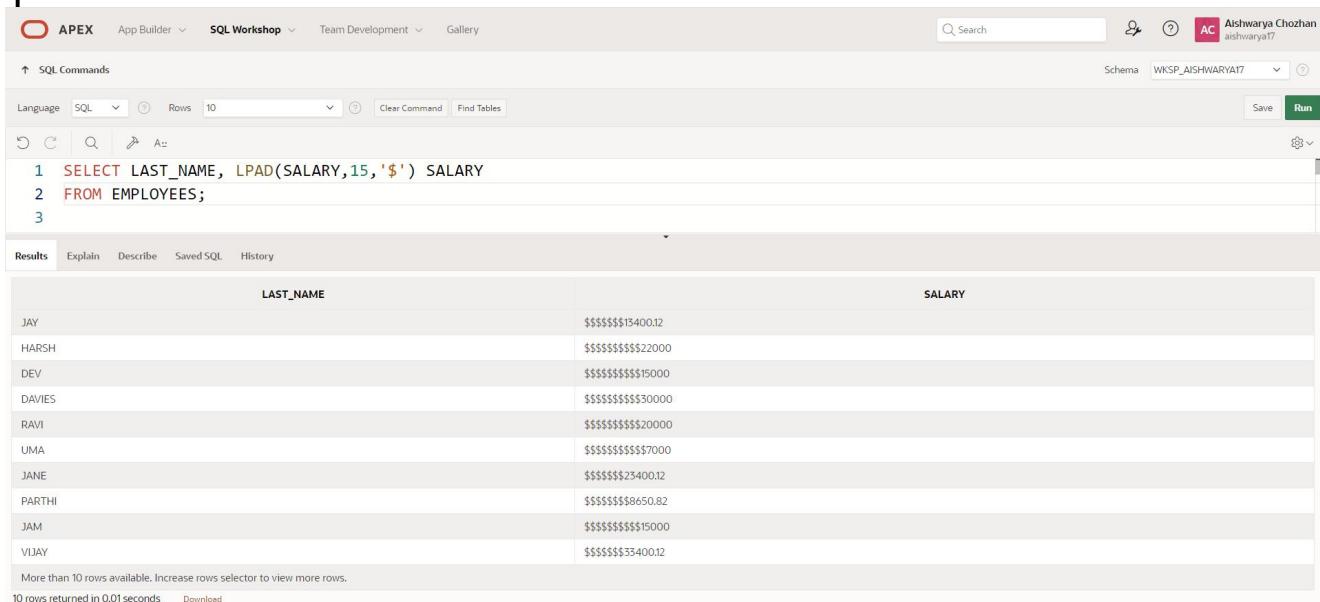
A note at the bottom says 'More than 10 rows available. Increase rows selector to view more rows.'

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, and Gallery. The right side shows the user profile 'Aishwarya Chozhan' and the schema 'WKSP_AISHWARYA17'. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the executed query:

```
1 SELECT LAST_NAME, LPAD(SALARY,15,'$') SALARY  
2 FROM EMPLOYEES;  
3
```

The Results tab shows the output:

LAST_NAME	SALARY
JAY	\$\$\$\$\$\$15400.12
HARSH	\$\$\$\$\$\$522000
DEV	\$\$\$\$\$\$15000
DAVIES	\$\$\$\$\$\$30000
RAVI	\$\$\$\$\$\$20000
UMA	\$\$\$\$\$\$7000
JANE	\$\$\$\$\$\$23400.12
PARTHI	\$\$\$\$\$\$8650.82
JAM	\$\$\$\$\$\$15000
VIJAY	\$\$\$\$\$\$33400.12

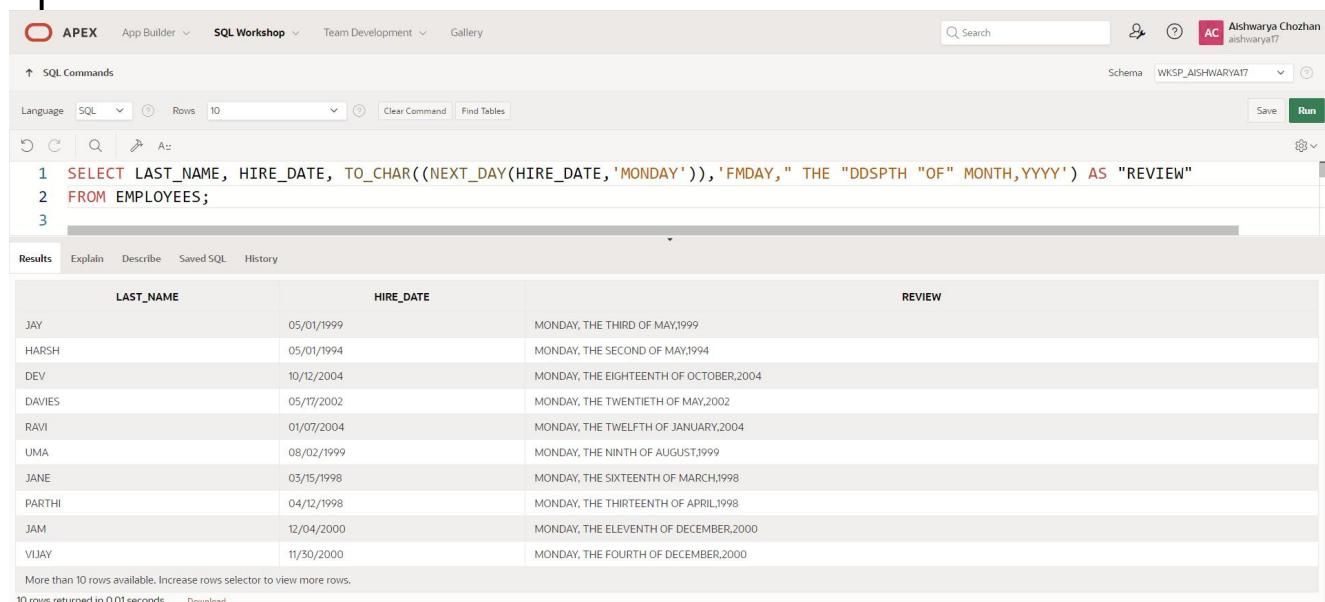
Below the table, a message says 'More than 10 rows available. Increase rows selector to view more rows.' and '10 rows returned in 0.01 seconds'.

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 "DDSPTH "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, and Gallery. On the right, there is a search bar, a user icon for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARVAT'. Below the toolbar, the language is set to SQL, rows are limited to 10, and there are buttons for Clear Command, Find Tables, Save, and Run.

The SQL command entered is:

```
1 SELECT LAST_NAME, HIRE_DATE, TO_CHAR((NEXT_DAY(HIRE_DATE,'MONDAY')),'FMDAY," THE "DDSPTH "OF" MONTH,YYYY') AS "REVIEW"  
2 FROM EMPLOYEES;  
3
```

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

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

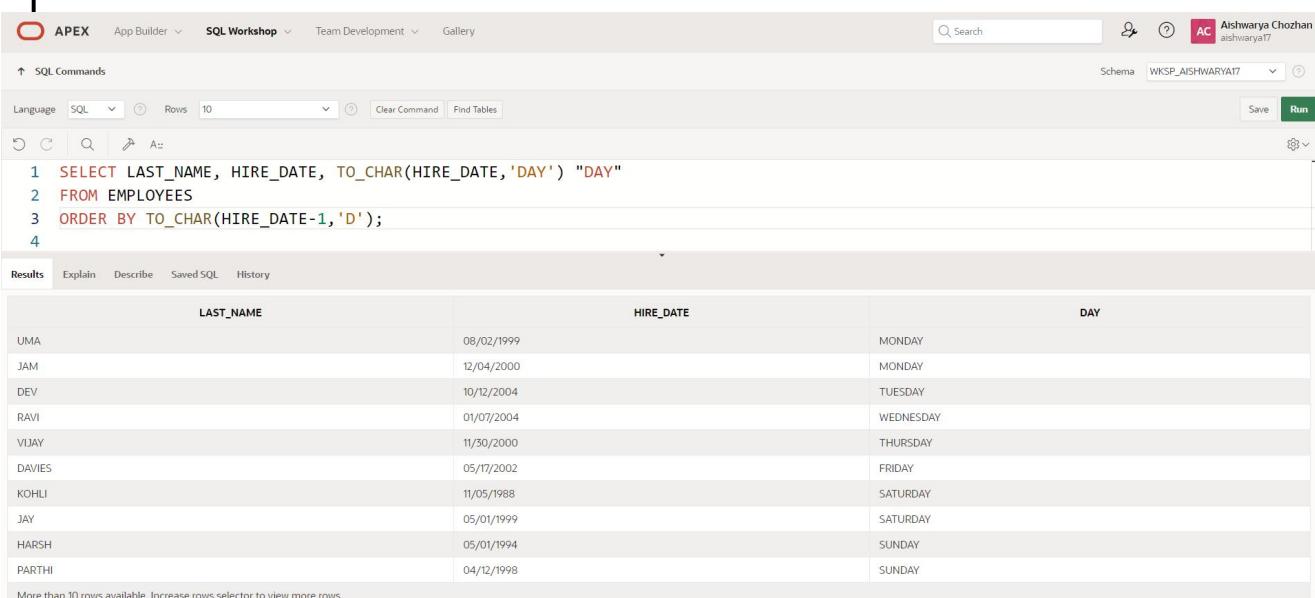
Below the table, a message indicates "More than 10 rows available. Increase rows selector to view more rows." and "10 rows returned in 0.01 seconds". There is also a "Download" link.

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 (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and a 'Run' button. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the executed query:

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

The Results tab displays the output in a table:

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

A note at the bottom of the results table says: "More than 10 rows available. Increase rows selector to view more rows."

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

RESULT :

DISPLAYING DATA FROM MULTIPLE TABLES

EX_NO:7

DATE:19/03/2024

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_17 D  
WHERE E.DEPARTMENT_ID = D.DEPT_ID;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, it shows the user's name, Aishwarya Chozhan, and schema, WKSP_AISHWARYA17. The main area is titled 'SQL Commands' and contains the following SQL code:

```
1 SELECT E.LAST_NAME,E.DEPARTMENT_ID,D.DEPT_ID FROM EMPLOYEES E,DEPARTMENT_17 D
2 WHERE E.DEPARTMENT_ID=D.DEPT_ID;
3
```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected and displays the following data:

LAST_NAME	DEPARTMENT_ID	DEPT_ID
JAY	10	10
HARSH	30	30
DEV	80	80
DAVIES	30	30
RAVI	10	10
UMA	20	20
JANE	50	50
PARTHI	20	20
JAM	80	80
VIJAY	40	40

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 E,DEPARTMENT_17 D
WHERE E.DEPARTMENT_ID=D.DEPT_ID AND E.DEPARTMENT_ID=80;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Aishwarya Chozhan' (aishwarya17), and a schema dropdown set to 'WKSP_AISHWARYA17'. Below the header, the SQL command window displays the following code:

```
1 SELECT DISTINCT JOB_ID, LOCATION_ID
2 FROM EMPLOYEES E,DEPARTMENT_17 D
3 WHERE E.DEPARTMENT_ID=D.DEPT_ID AND E.DEPARTMENT_ID=80;
4
```

The results window shows a table with two columns: 'JOB_ID' and 'LOCATION_ID'. The data is as follows:

JOB_ID	LOCATION_ID
DESIGNER	6
SUPERVISOR	6
ENGINEER	6

Below the table, it says '3 rows returned in 0.03 seconds' and has a 'Download' link.

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, E.DEPARTMENT_ID, D.DEPARTMENT_NAME, D.LOCATION_ID,
       L.CITY
  FROM EMPLOYEES E, DEPARTMENT_17 D, LOCATION L
 WHERE E.DEPARTMENT_ID = D.DEPARTMENT_ID AND D.LOCATION_ID = L.LOCATION_ID
   AND E.COMMISSION_PCT IS NOT NULL;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery Search

↑ SQL Commands Schema WKSP_AISHWARYA17 Save Run

Language SQL Rows 10 Clear Command Find Tables

```

1 SELECT E.LAST_NAME,E.DEPARTMENT_ID,D.DEPT_NAME,D.LOCATION_ID,
2 L.CITY FROM EMPLOYEES E,DEPARTMENT_17 D,LOCATION L
3 WHERE E.DEPARTMENT_ID=D.DEPT_ID AND D.LOCATION_ID=L.LOCATION_ID AND E.COMMISSION_PCT IS NOT NULL;
4

```

Results Explain Describe Saved SQL History

LAST_NAME	DEPARTMENT_ID	DEPT_NAME	LOCATION_ID	CITY
JAY	10	MARKETING	1	CHENNAI
HARSH	30	FINANCE	3	VALHALLA
DEV	80	MANUFACTURING	6	TORONTO
DAVIES	30	FINANCE	3	VALHALLA
RAVI	10	MARKETING	1	CHENNAI
JANE	50	HR	5	LONDON
JAM	80	MANUFACTURING	6	TORONTO
VIJAY	40	MANAGEMENT	4	DC
KOHLI	80	MANUFACTURING	6	TORONTO

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_17
WHERE EMPLOYEES.DEPARTMENT_ID=DEPARTMENT_17.DEPT_ID AND
LAST_NAME LIKE '%a%';

```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. Below the header, the 'SQL Commands' tab is selected, showing the following SQL code:

```

1 SELECT LAST_NAME,DEPT_NAME
2 FROM EMPLOYEES,DEPARTMENT_17
3 WHERE EMPLOYEES.DEPARTMENT_ID=DEPARTMENT_17.DEPT_ID AND LAST_NAME LIKE '%A%';
4

```

Below the code, the 'Results' tab is active, displaying the output in a grid format:

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

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_17 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. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. Below the header, the SQL editor contains the following query:

```

1 SELECT LAST_NAME, JOB_ID, DEPARTMENT_ID, DEPT_NAME
2 FROM EMPLOYEES JOIN DEPARTMENT_17 D
3 ON (DEPARTMENT_ID = DEPT_ID)
4 JOIN LOCATION L
5 ON (D.LOCATION_ID = L.LOCATION_ID)
6 WHERE LOWER(L.CITY) = 'toronto';

```

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

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

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:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```

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

```

Results Explain Describe Saved SQL History

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

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:

```

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

```

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

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:

```

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

```

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

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:

```

SELECT E.LAST_NAME, E.JOB_ID, D.DEPT_NAME,
E.SALARY, J.GRADE_LEVEL
FROM EMPLOYEES E, DEPARTMENT_17 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 APEX SQL Workshop interface. At the top, there are navigation links for App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. Below the header, a toolbar includes options for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run.

The main area displays a SQL query:

```

1 SELECT E.LAST_NAME, E.JOB_ID, D.DEPT_NAME,
2 E.SALARY, J.GRADE_LEVEL FROM EMPLOYEES E, DEPARTMENT_17 D, JOB_GRADES J
3 WHERE E.DEPARTMENT_ID = D.DEPT_ID AND E.SALARY BETWEEN J.LOWEST_SAL AND J.HIGHEST_SAL;

```

Below the query, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing a table with the following data:

LAST_NAME	JOB_ID	DEPT_NAME	SALARY	GRADE_LEVEL
HARSH	FI_MANAGER	FINANCE	22000	D
DAVIES	AUDITOR	FINANCE	30000	E
DEV	ENGINEER	MANUFACTURING	15000	C
JAM	DESIGNER	MANUFACTURING	15000	C
KOHLI	SUPERVISOR	MANUFACTURING	20000	D
VIJAY	MANAGER	MANAGEMENT	33400.12	E
JAY	SL_REP	MARKETING	13400.12	B
RAVI	MK_MANAGER	MARKETING	20000	D
UMA	ST_CLERK	STOCK	7000	A
PARTHI	ST_CLERK	STOCK	8650.82	A

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. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Aishwarya Chozhan' (aishwarya17), and a schema dropdown set to 'WKSP_AISHWARYA17'. Below the header, the SQL Commands tab is active, showing the following query:

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

Below the query, there are tabs for Results (selected), Explain, Describe, Saved SQL, and History. The Results tab displays the output:

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

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

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:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

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

Results Explain Describe Saved SQL History

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

6 rows returned in 0.01 seconds Download

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

RESULT

AGGREGATING DATA USING GROUP FUNCTIONS

EX_NO : 8

DATE:26/03/2024

1. 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.

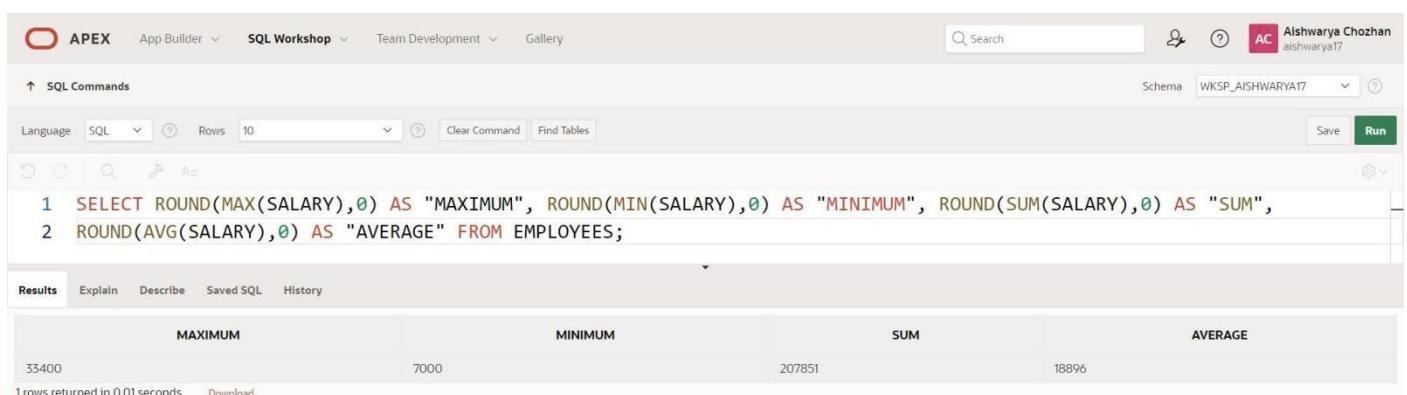
True/False

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;
```



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 user profile for 'Aishwarya Chozhan' (aishwaryachozhan17). The main workspace is titled 'SQL Commands'. It features a toolbar with icons for Run, Save, and other functions. Below the toolbar, there are dropdown menus for Language (set to SQL), Rows (set to 10), and a search bar. The SQL editor contains the following code:

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

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

	MAXIMUM	MINIMUM	SUM	AVERAGE
	33400	7000	207851	18896

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

OUTPUT:

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

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 GROUP BY JOB_ID;
```

OUTPUT:

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

```
1 SELECT ROUND(MAX(SALARY),0) AS "MAXIMUM", ROUND(MIN(SALARY),0) AS "MINIMUM", ROUND(SUM(SALARY),0) AS "SUM",  
2 ROUND(AVG(SALARY),0) AS "AVERAGE" FROM EMPLOYEES GROUP BY JOB_ID;  
3
```

The Results tab displays the output of the query:

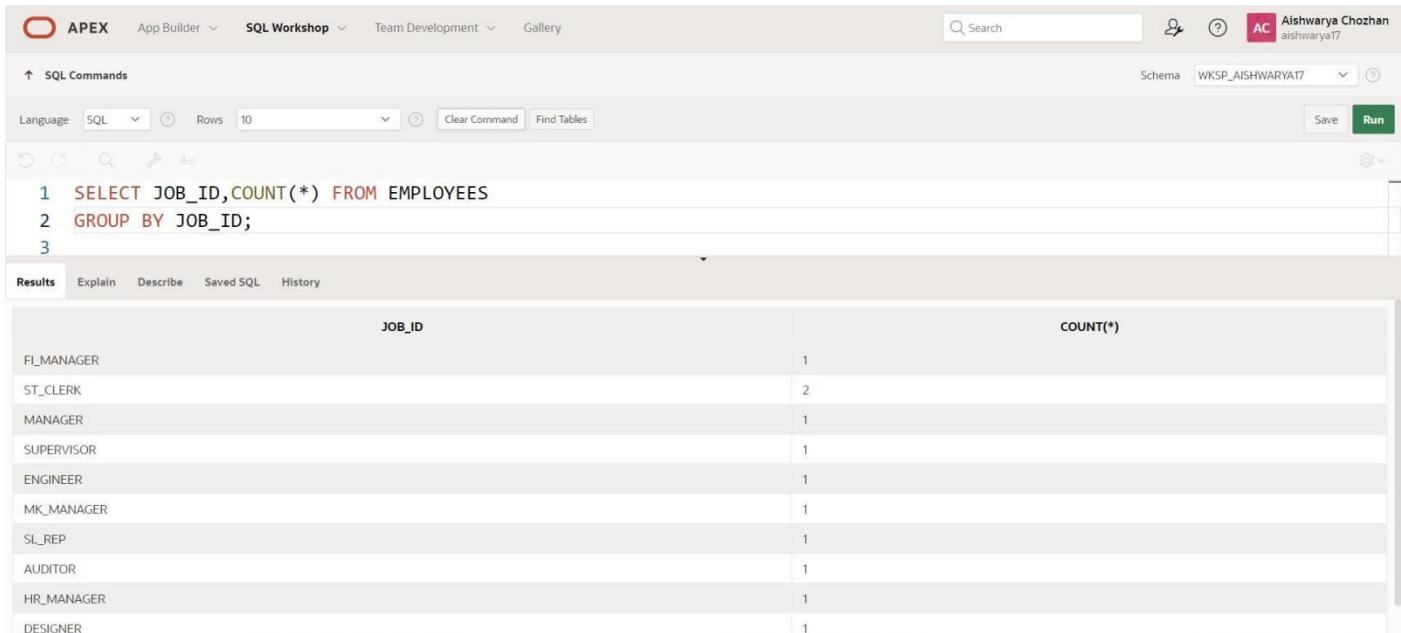
	MAXIMUM	MINIMUM	SUM	AVERAGE
22000	22000	22000	22000	22000
8651	7000	15651	7825	
33400	33400	33400	33400	33400
20000	20000	20000	20000	20000
15000	15000	15000	15000	15000
20000	20000	20000	20000	20000
13400	13400	13400	13400	13400
30000	30000	30000	30000	30000
23400	23400	23400	23400	23400
15000	15000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

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(*) 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, Team Development, and Gallery. On the right, there's a search bar, user information for 'Aishwarya Chozhan' (aishwarya17), and a 'Run' button. Below the toolbar, the schema is set to 'WKSP_AISHWARYA17'. The main area shows the SQL command being run:

```
1 SELECT JOB_ID,COUNT(*) FROM EMPLOYEES  
2 GROUP BY JOB_ID;  
3
```

The results tab is selected, displaying the output:

JOB_ID	COUNT(*)
FL_MANAGER	1
ST_CLERK	2
MANAGER	1
SUPERVISOR	1
ENGINEER	1
MK_MANAGER	1
SL REP	1
AUDITOR	1
HR_MANAGER	1
DESIGNER	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 APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user information for 'Aishwarya Chozhan' (aishwarya17), and a 'Run' button. The main area is titled 'SQL Commands' with a 'Results' tab selected. The query entered is:

```
1 SELECT COUNT(DISTINCT MANAGER_ID) "NUMBER OF MANAGERS"
2 FROM EMPLOYEES;
3
```

The results table has a single row with the header 'NUMBER OF MANAGERS' and the value '4'. Below the table, it says '1 rows returned in 0.01 seconds'.

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, Team Development, and Gallery. On the right, there's a search bar, user information for 'Aishwarya Chozhan' (aishwarya17), and a 'Run' button. The main area is titled 'SQL Commands' with a 'Results' tab selected. The query entered is:

```
1 SELECT MAX(SALARY)-MIN(SALARY) AS "DIFFERENCE" FROM EMPLOYEES;
2
```

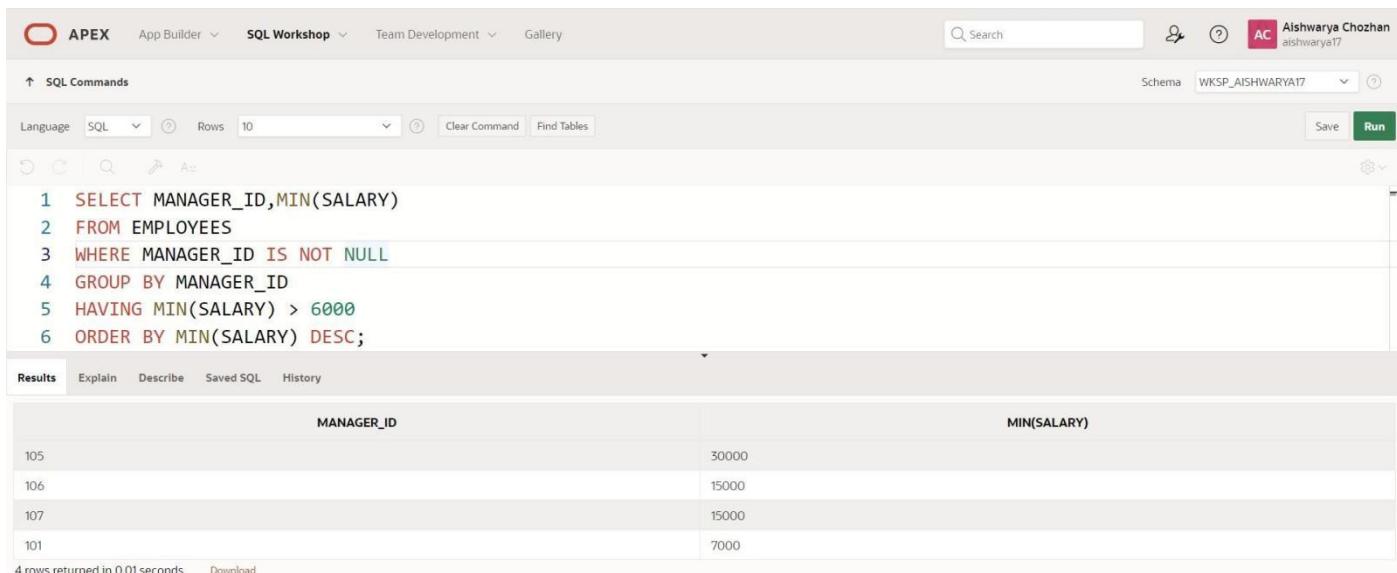
The results table has a single row with the header 'DIFFERENCE' and the value '26400.12'. Below the table, it says '1 rows returned in 0.01 seconds'.

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 APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwaryat17). The main workspace displays the SQL command entered by the user, followed by the results table.

```
1 SELECT MANAGER_ID,MIN(SALARY)
2 FROM EMPLOYEES
3 WHERE MANAGER_ID IS NOT NULL
4 GROUP BY MANAGER_ID
5 HAVING MIN(SALARY) > 6000
6 ORDER BY MIN(SALARY) DESC;
```

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

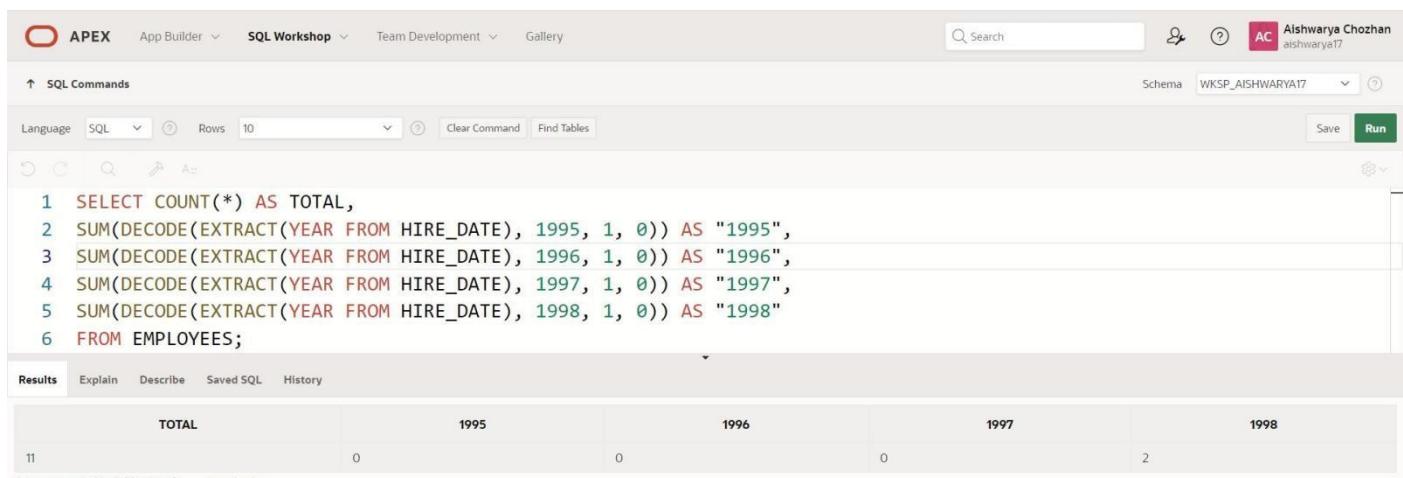
4 rows returned in 0.01 seconds [Download](#)

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 query has been run, and the results are displayed in a table. The table has five columns: TOTAL, 1995, 1996, 1997, and 1998. The data row shows values: 11, 0, 0, 0, and 2 respectively. The table header also includes the column names.

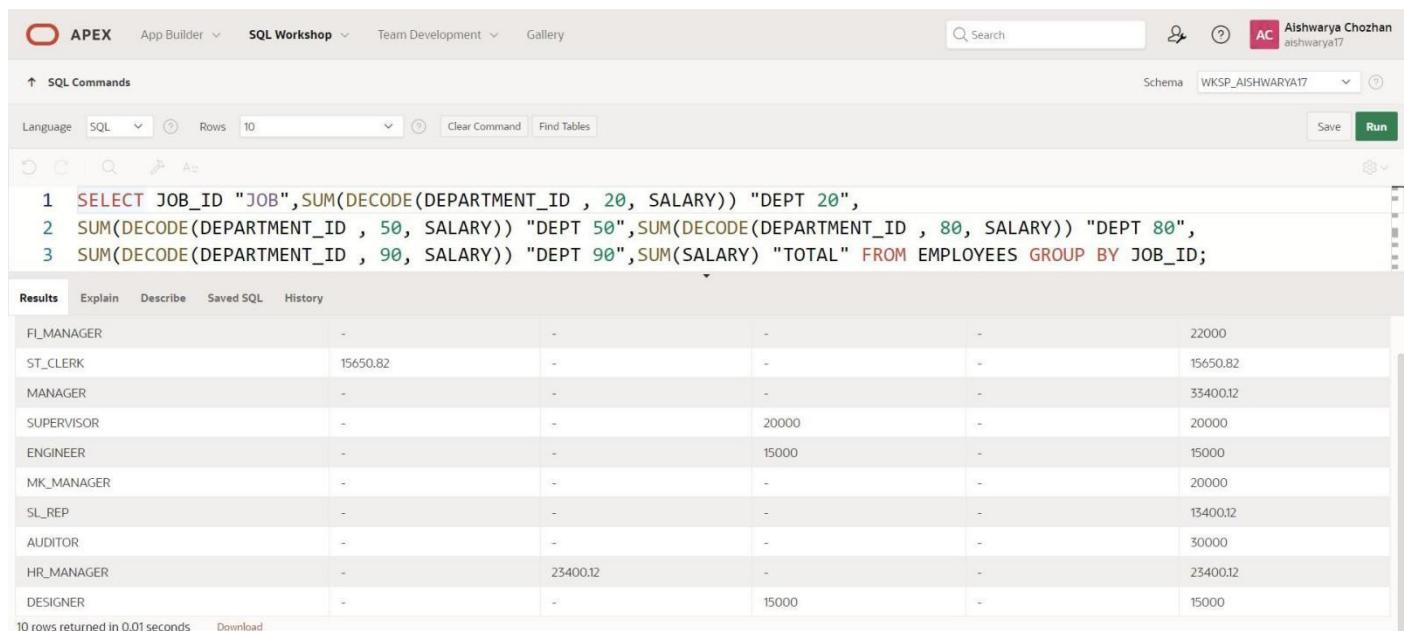
TOTAL	1995	1996	1997	1998
11	0	0	0	2

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 query is executed successfully, returning 10 rows of data. The results are displayed in a table with columns: JOB_ID, DEPT 20, DEPT 50, DEPT 80, DEPT 90, and TOTAL.

JOB_ID	DEPT 20	DEPT 50	DEPT 80	DEPT 90	TOTAL
FI_MANAGER	-	-	-	-	22000
ST_CLERK	15650.82	-	-	-	15650.82
MANAGER	-	-	-	-	33400.12
SUPERVISOR	-	-	20000	-	20000
ENGINEER	-	-	15000	-	15000
MK_MANAGER	-	-	-	-	20000
SL_REP	-	-	-	-	13400.12
AUDITOR	-	-	-	-	30000
HR_MANAGER	-	23400.12	-	-	23400.12
DESIGNER	-	-	15000	-	15000

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. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run.

The SQL command entered is:

```
1 SELECT D.DEPT_NAME "NAME",
2 D.LOCATION_ID "LOCATION ", COUNT(*) "NUMBER OF PEOPLE",
3 ROUND(AVG(SALARY),2) "SALARY" FROM EMPLOYEES E,
4 DEPARTMENT_17 D WHERE E.DEPARTMENT_ID = D.DEPT_ID
5 GROUP BY D.DEPT_NAME, D.LOCATION_ID;
6
```

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

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

At the bottom left, it says '6 rows returned in 0.06 seconds'. There is also a 'Download' link.

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

RESULT:

SUB QUERIES

EX_NO : 9

DATE:11/04/2024

Find the Solution for the following:

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:

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

The results pane shows a single row of data:

LAST_NAME	HIRE_DATE
PARTHI	12-APR-1998

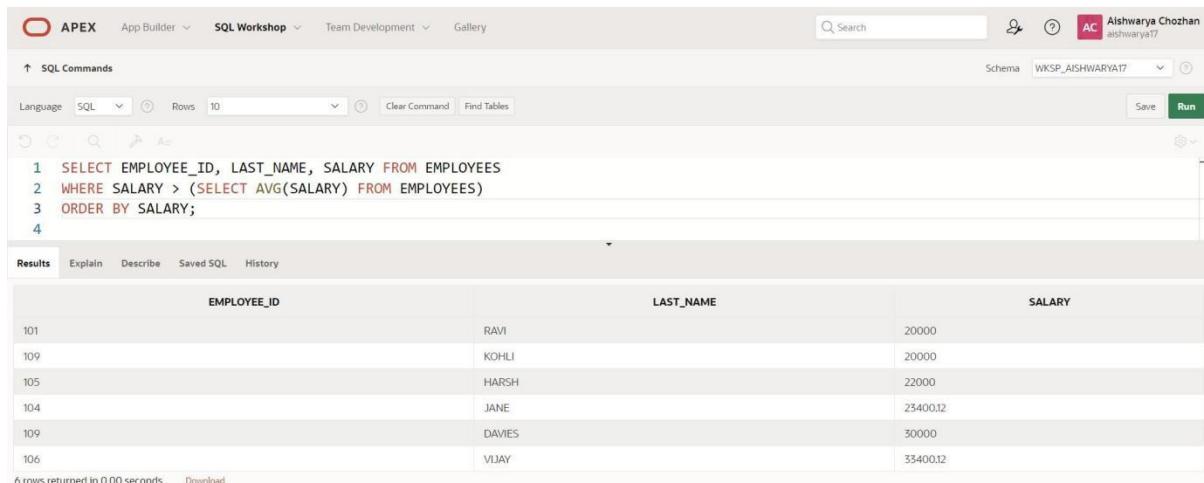
1 rows returned in 0.00 seconds

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 SQL Commands tab contains the following query:

```
1 SELECT EMPLOYEE_ID, LAST_NAME, SALARY FROM EMPLOYEES  
2 WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES)  
3 ORDER BY SALARY;  
4
```

The Results tab displays the output:

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

6 rows returned in 0.00 seconds

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 top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area has tabs for 'SQL Commands' (selected) and 'Results'. In the 'SQL Commands' tab, the following query is entered:

```
1 SELECT EMPLOYEE_ID, LAST_NAME
2 FROM EMPLOYEES A
3 JOIN (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME LIKE '%U%') B
4 ON A.DEPARTMENT_ID = B.DEPARTMENT_ID;
5
```

The 'Results' tab displays the output of the query:

EMPLOYEE_ID	LAST_NAME
102	UMA
103	PARTHI

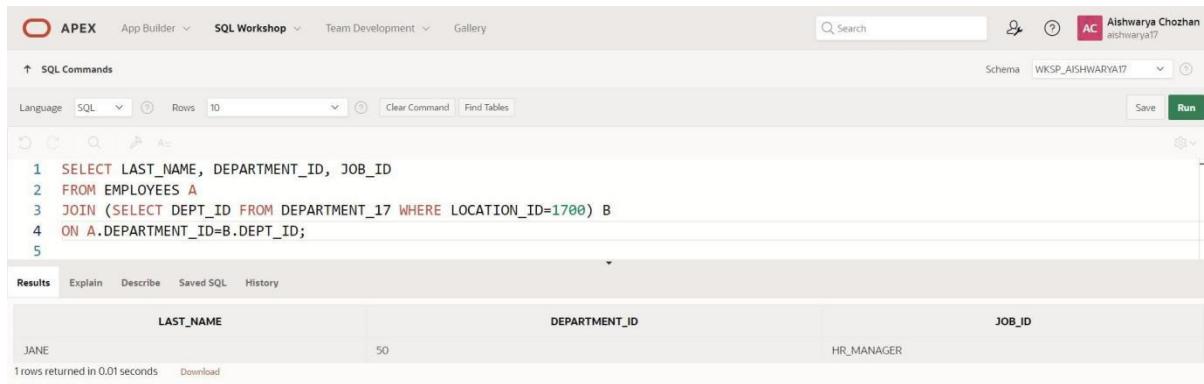
Below the table, it says '2 rows returned in 0.01 seconds'.

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. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID  
2 FROM EMPLOYEES A  
3 JOIN (SELECT DEPT_ID FROM DEPARTMENT_17 WHERE LOCATION_ID=1700) B  
4 ON A.DEPARTMENT_ID=B.DEPT_ID;  
5
```

Below the code, the 'Results' tab is selected, showing the output of the query:

LAST_NAME	DEPARTMENT_ID	JOB_ID
JANE	50	HR_MANAGER

At the bottom left, it says '1 rows returned in 0.01 seconds'. There are also 'Download' and 'Run' buttons at the bottom right of the results panel.

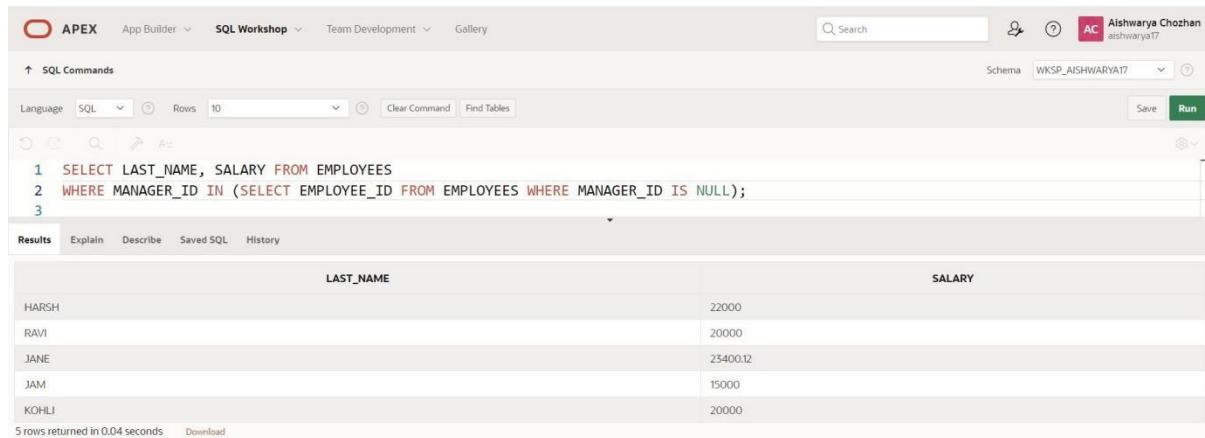
71.

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', and 'Gallery'. On the right, there's a user icon for 'Aishwarya Chozhan' and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area has tabs for 'SQL Commands' and 'Results' (which is selected). The SQL command entered is:

```
1 SELECT LAST_NAME, SALARY FROM EMPLOYEES  
2 WHERE MANAGER_ID IN (SELECT EMPLOYEE_ID FROM EMPLOYEES WHERE MANAGER_ID IS NULL);  
3
```

The results table shows the following data:

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

At the bottom, it says '5 rows returned in 0.04 seconds' and has 'Download' and 'Run' buttons.

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_17 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. The right side shows a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the executed query:

```
1 SELECT DEPARTMENT_ID, LAST_NAME, JOB_ID FROM EMPLOYEES  
2 WHERE DEPARTMENT_ID=(SELECT DEPT_ID FROM DEPARTMENT_17 WHERE DEPT_NAME='EXECUTIVE');  
3
```

The Results tab displays the output in a table:

DEPARTMENT_ID	LAST_NAME	JOB_ID
60	HARSH	CEO
60	SIDDHU	COO

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

73.

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', and 'Gallery'. The right side shows a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area has a toolbar with 'SQL Commands' (selected), 'Language' (set to SQL), 'Rows' (set to 10), and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. The SQL editor contains the following code:

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

The 'Results' tab is selected, showing a table with three columns: 'EMPLOYEE_ID', 'LAST_NAME', and 'SALARY'. The data returned is:

EMPLOYEE_ID	LAST_NAME	SALARY
113	HARSH	45000
114	SIDDHU	40000

At the bottom left, it says '2 rows returned in 0.01 seconds'. At the bottom right, there are 'Download' and 'Copy' buttons.

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

RESULT:

USING THE SET OPERATORS

EX.NO:10

DATE:15/04/2024

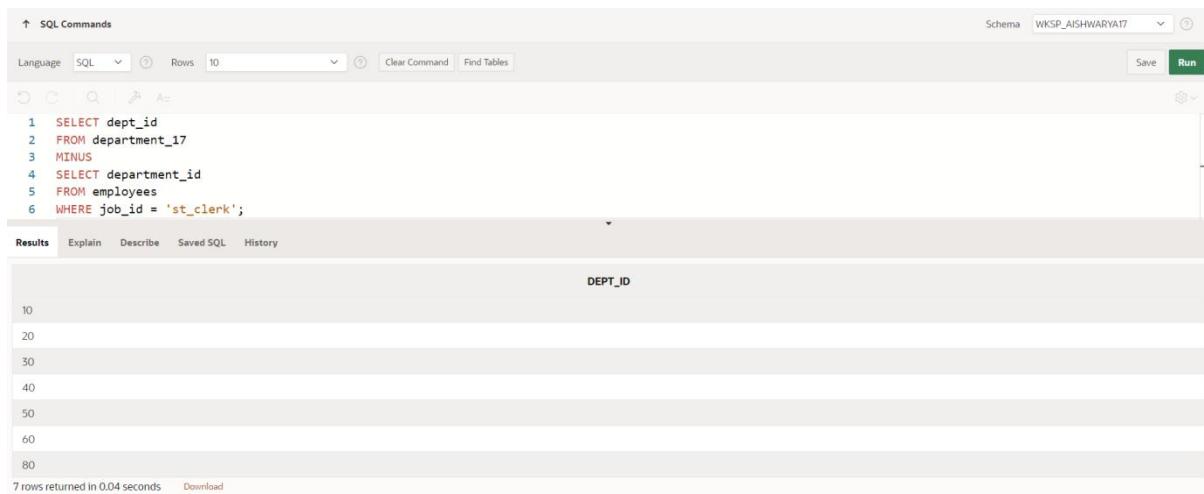
Find the Solution for the following:

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 DEPT_ID
FROM EMPLOYEES
WHERE JOB_ID = 'ST_CLERK';
```

OUTPUT:



The screenshot shows a SQL command window with the following details:

- Schema:** WKSP_AISHWARYA17
- Language:** SQL
- Rows:** 10
- Commands:** Clear Command, Find Tables
- Run Button:** A green "Run" button is visible in the top right corner.
- SQL Query:**

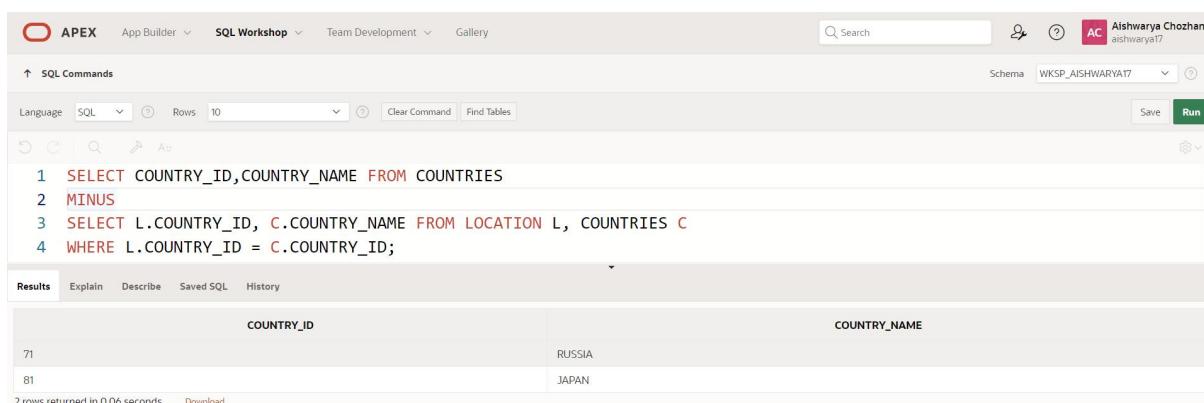
```
1 SELECT dept_id
2 FROM department_17
3 MINUS
4 SELECT department_id
5 FROM employees
6 WHERE job_id = 'st_clerk';
```
- Results:** The results are displayed in a table with a single column labeled "DEPT_ID". The values listed are 10, 20, 30, 40, 50, 60, and 80.
- Timing:** 7 rows returned in 0.04 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 locations l, countries c  
WHERE l.country_id = c.country_id;
```

OUTPUT:



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

```
1 SELECT COUNTRY_ID,COUNTRY_NAME FROM COUNTRIES  
2 MINUS  
3 SELECT L.COUNTRY_ID, C.COUNTRY_NAME FROM LOCATION L, COUNTRIES C  
4 WHERE L.COUNTRY_ID = C.COUNTRY_ID;
```

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

COUNTRY_ID	COUNTRY_NAME
71	RUSSIA
81	JAPAN

Below the table, it says "2 rows returned in 0.06 seconds".

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_no, dept_id  
FROM employees  
WHERE dept_id = 10
```

UNION ALL

```
SELECT DISTINCT job_no, dept_id  
FROM employees  
WHERE dept_id = 50
```

UNION ALL

```
SELECT DISTINCT job_no, dept_id  
FROM employees  
WHERE dept_id = 20;
```

OUTPUT:

The screenshot shows a SQL command window with the following details:

- Language:** SQL
- Schema:** WKSP_AISHWARYA17
- Commands:** 5 SQL statements are listed:
 - 1 SELECT DISTINCT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 10
 - 2 UNION ALL
 - 3 SELECT DISTINCT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 50
 - 4 UNION ALL
 - 5 SELECT DISTINCT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 20;
- Results:** A table is displayed with the following data:

JOB_ID	DEPARTMENT_ID
SL_REP	10
MK_MANAGER	10
HR_MANAGER	50
ST_CLERK	20

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).

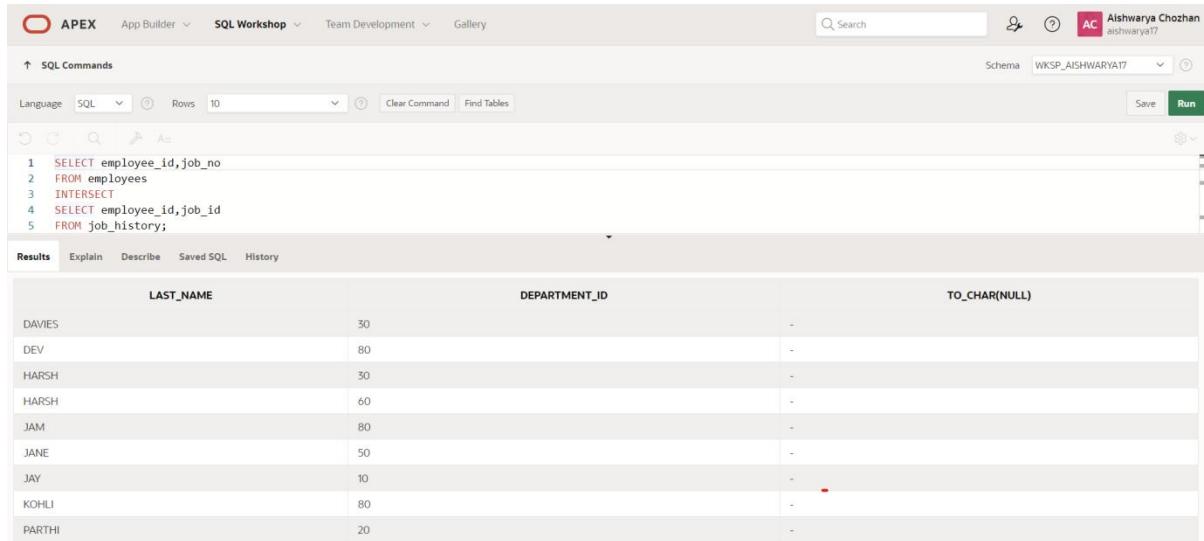
QUERY:

```

SELECT employee_id, job_no
FROM employees
INTERSECT
SELECT employee_id, job_no
FROM job_history;

```

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 'Aishwarya Chozhan' and schema 'WKSP_AISHWARYA17'. The main area has tabs for 'SQL Commands' and 'Results'. Under 'Results', there are tabs for 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the output of the executed query.

```

1 SELECT employee_id, job_no
2 FROM employees
3 INTERSECT
4 SELECT employee_id, job_id
5 FROM job_history;

```

LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
DAVIES	30	-
DEV	80	-
HARSH	30	-
HARSH	60	-
JAM	80	-
JANE	50	-
JAY	10	-
KOHLI	80	-
PARTHI	20	-

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

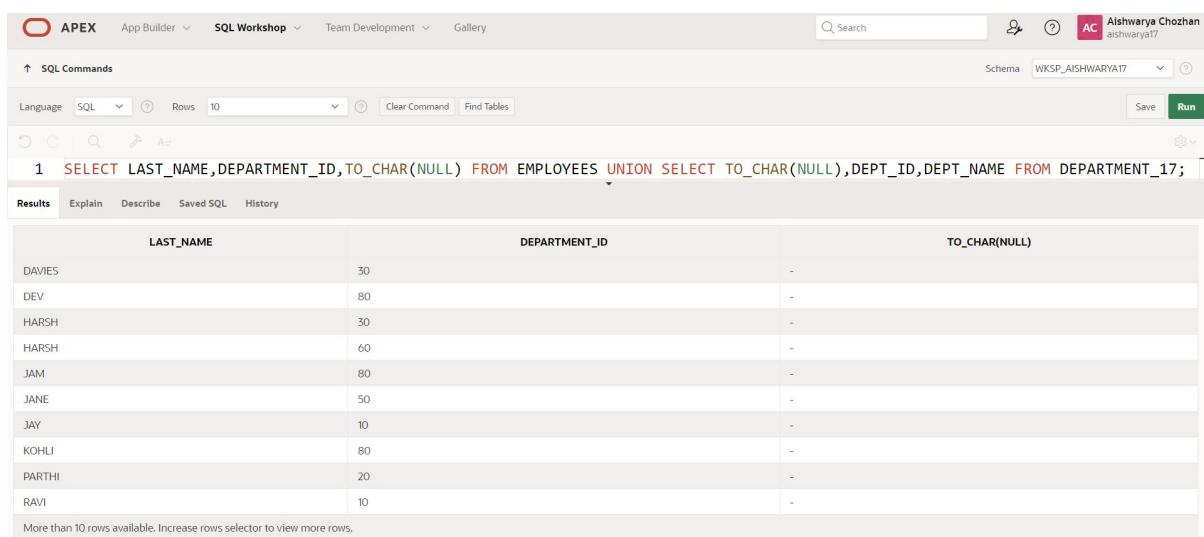
- Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department.

- 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,dept_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 top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area has tabs for 'SQL Commands' (selected), 'Results' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL command entered is:

```
1 SELECT LAST_NAME,DEPARTMENT_ID,TO_CHAR(NULL) FROM EMPLOYEES UNION SELECT TO_CHAR(NULL),DEPT_ID,DEPT_NAME FROM DEPARTMENT_17;
```

The results section displays the output of the query:

LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
DAVIES	50	-
DEV	80	-
HARSH	30	-
HARSH	60	-
JAM	80	-
JANE	50	-
JAY	10	-
KOHLI	80	-
PARTHI	20	-
RAVI	10	-

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

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

RESULT:

Creating views

EX_NO : 11

DATE:16/04/2024

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. In the central workspace, a SQL command is being entered to create a view:

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

Below the code, the results of the execution are displayed:

View created.
0.03 seconds

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 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user icon for 'Aishwarya Chozhan' and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands' with a sub-section 'EMPLOYEE_VU'. A single row of code is shown: '1 SELECT * FROM EMPLOYEE_VU;'. Below this is a table with three columns: 'EMPLOYEE_ID', 'EMPLOYEE', and 'DEPARTMENT_ID'. The data is as follows:

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
100	SAHANA JAY	10
113	KHAN HARSH	60
105	SANA HARSH	50
108	MIKA DEV	80
109	BEN DAVIES	30
101	ASHA RAVI	10
102	TARA UMA	20
104	MARY JANE	50
103	SANJ PARTHI	20
107	JIM JAM	80

A note at the bottom says '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 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user icon for 'Aishwarya Chozhan' and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands' with a sub-section 'USER.Views'. A single row of code is shown: '1 SELECT VIEW_NAME,TEXT FROM USER.Views;'. Below this is a table with two columns: 'VIEW_NAME' and 'TEXT'. The data is as follows:

VIEW_NAME	TEXT
EMPLOYEE_VU	SELECT EMPLOYEE_ID,FIRST_NAME '' LAST_NAME,DEPARTMENT_ID FROM EMPLOYEES

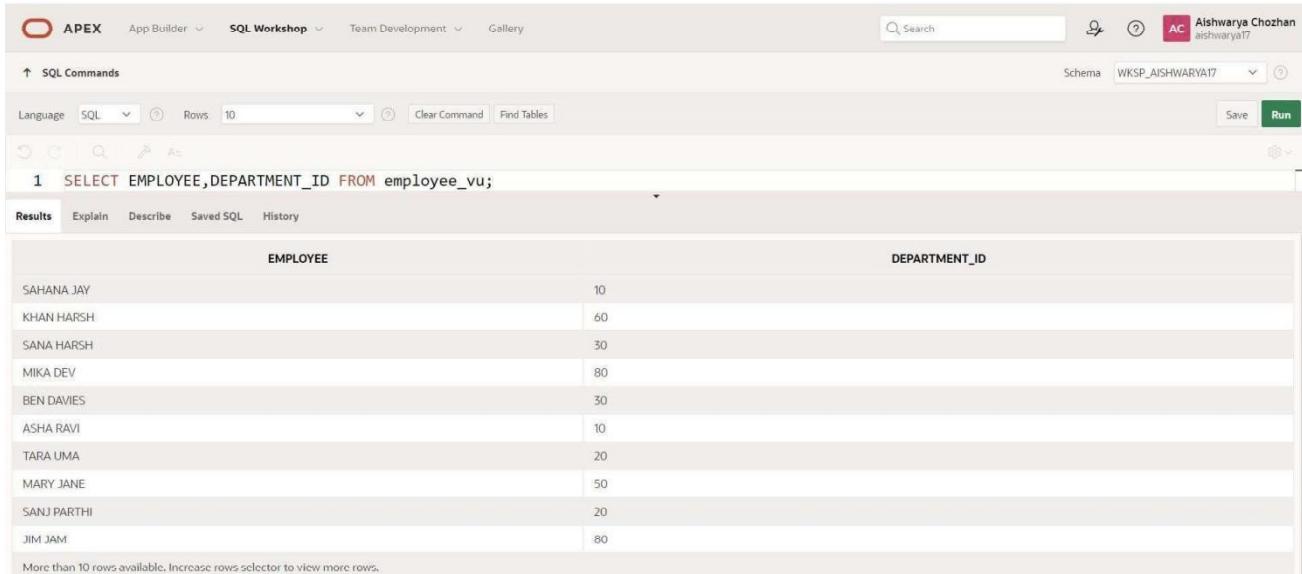
A note at the bottom says '1 rows returned in 0.05 seconds'.

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

QUERY:

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

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 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area displays a SQL command and its results.

SQL Command:

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

Results:

EMPLOYEE	DEPARTMENT_ID
SAHANA JAY	10
KHAN HARSH	60
SANA HARSH	30
MIKA DEV	80
BEN DAVIES	30
ASHA RAVI	10
TARA UMA	20
MARY JANE	50
SANJ PARTHI	20
JIM JAM	80

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

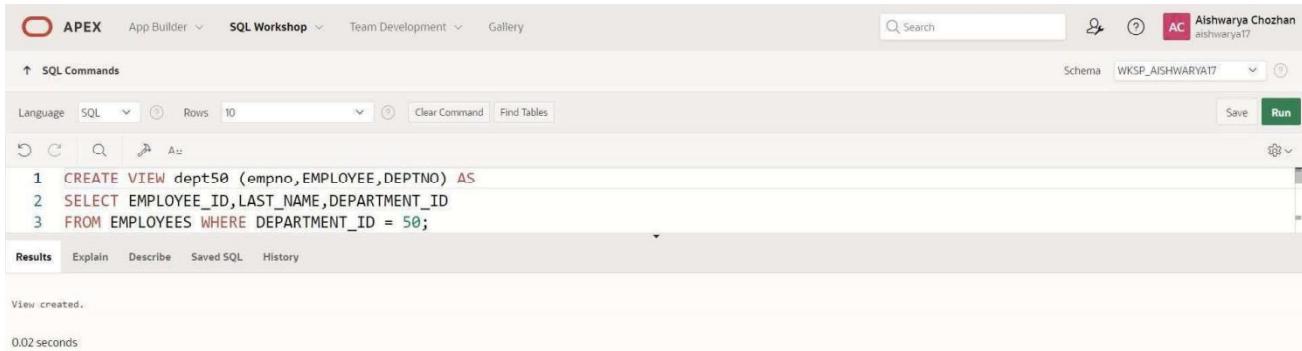
5. Create a view named DEPT50 that contains the employee number, employee last names and department numbers for all employees in department 50. Label the view columns EMPNO, EMPLOYEE and DEPTNO. Do not allow an employee to be reassigned to another department through the view.

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', and 'Gallery'. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. The main workspace displays the following SQL code:

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

Below the code, the 'Results' tab is selected, showing the message 'View created.' and a execution time of '0.02 seconds'.

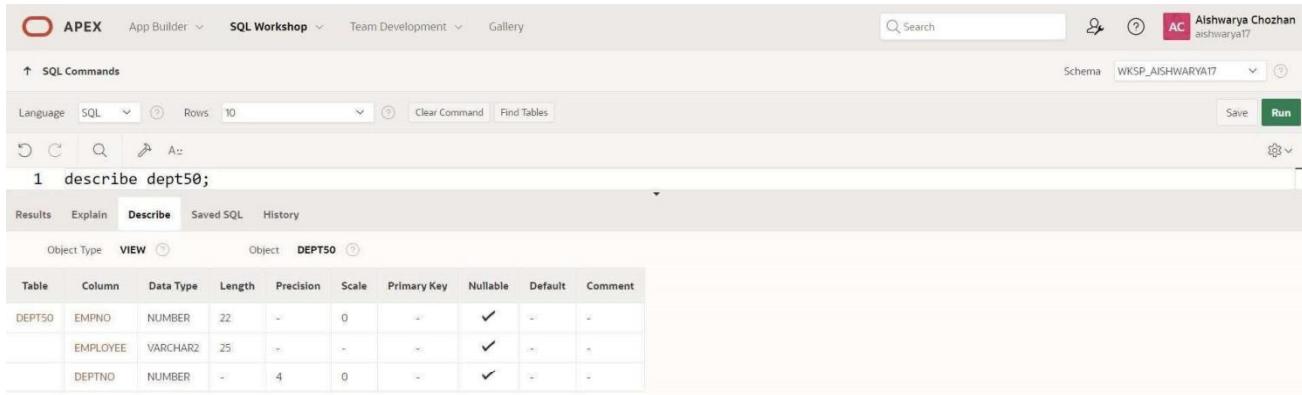
6. Display the structure and contents of the DEPT50 view.

QUERY:

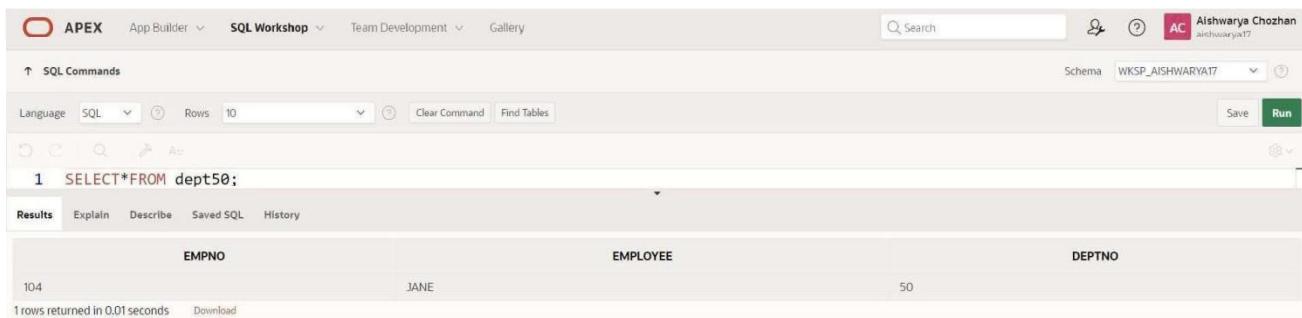
```
describe dept50;
```

```
SELECT*FROM dept50;
```

OUTPUT:



```
1 describe dept50;
Object Type: VIEW
Table Column Data Type Length Precision Scale Primary Key Nullable Default Comment
DEPT50 EMPNO NUMBER 22 - 0 - ✓ - -
EMPLOYEE VARCHAR2 25 - - - ✓ - -
DEPTNO NUMBER - 4 0 - - ✓ - -
```



```
1 SELECT*FROM dept50;
EMPNO          EMPLOYEE        DEPTNO
104             JANE            50
1 rows returned in 0.01 seconds   Download
```

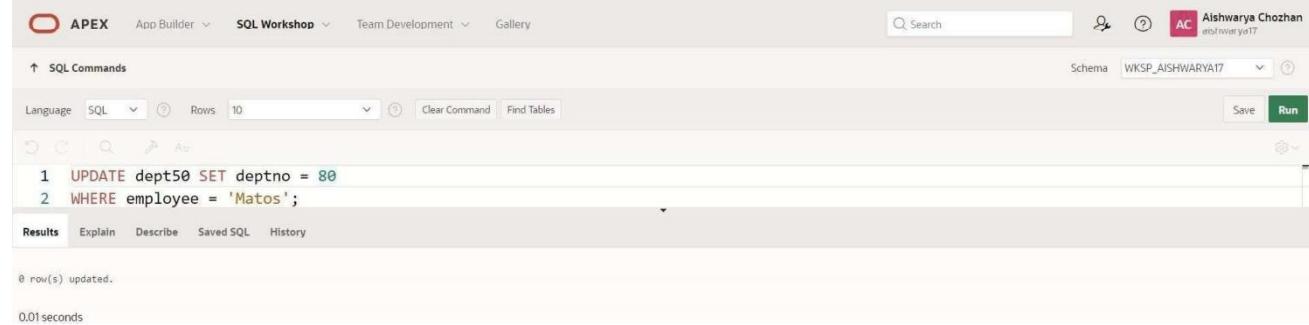
7. Attempt to reassign Matos to department 80.

QUERY:

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

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. Below the input area, the results show the executed SQL code:

```
1 UPDATE dept50 SET deptno = 80
2 WHERE employee = 'Matos';
```

Under the 'Results' tab, the output is displayed as:

```
0 row(s) updated.
```

Execution time is listed as 0.01 seconds.

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', and 'Gallery'. The right side shows the user 'Aishwarya Chozhan' (aishwaryach17) and the schema 'WKSP_AISHWARYA17'. The main area displays the following SQL code:

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

Below the code, the 'Results' tab is selected, showing the message 'View created.' and a execution time of '0.02 seconds'.

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

Faculty Signature	
-------------------	--

RESULT:

EXERCISE 12

Intro to Constraints; NOT NULL and UNIQUE Constraint

EXERCISE NO:12

DATE:23/04/2024

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.

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

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
Id	PK				NO	
name						
date_opened					NO	
address					NO	
city					NO	
zip/postal code						
phone						
email	UK					
manager_id						
Emergency contact						

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

Database can be as reliable as the data in it, and database rules are implemented as Constraint to maintain data integrity. For example these constraints may prohibit deletion of a table or some row when insertion, updation or deletion is executed.

Type of constraints:

- PRIMARY KEY Constraint
- UNIQUE Constraint
- FOREIGN KEY Constraint
- CHECK Constraint with condition applied on the column/columns (they work at row level)

- NOT NULL Constraint (implemented at row level using special CHECK Constraint having condition IS NOT NULL for single column)

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

- 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.
- If word CONSTRAINT is used in a CREATE TABLE statement, I must specify constraint name. Also, that is why, Table level constraint must be user-named.

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

- 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.
- Handling production issues may be faster with user-named constraints

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.

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
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.

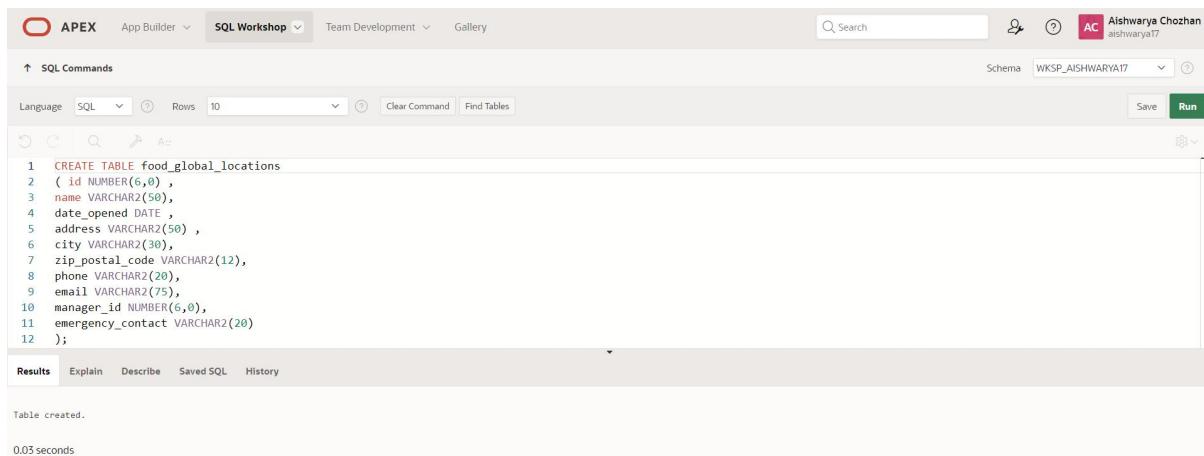
Global Fast Foods global_locations Table							
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT	
Id	PK	NUMBER	6	0	NO		
name		VARCHAR2	50		YES		
date_opened		DATE			NO		
address		VARCHAR2	50		NO		
city		VARCHAR2	30		NO		
zip/postal code		VARCHAR2	12		YES		
phone		VARCHAR2	20		YES		
email	UK	VARCHAR2	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.

QUERY:

```
CREATE TABLE food_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)
);
```

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 user icon for 'Aishwarya Chozhan' and a schema dropdown set to 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands' with a 'Run' button. Below it, the SQL code for creating the 'food_global_locations' table is pasted. The code defines the table with columns: id (NUMBER(6,0)), name (VARCHAR2(50)), date_opened (DATE), address (VARCHAR2(50)), city (VARCHAR2(30)), zip_postal_code (VARCHAR2(12)), phone (VARCHAR2(20)), email (VARCHAR2(75) with a unique constraint), manager_id (NUMBER(6,0)), and emergency_contact (VARCHAR2(20)). The table is created successfully, as indicated by the message 'Table created.' and a execution time of '0.03 seconds'.

```
CREATE TABLE food_global_locations
( id NUMBER(6,0) ,
name VARCHAR2(50),
date_opened DATE ,
address VARCHAR2(50) ,
city VARCHAR2(30) ,
zip_postal_code VARCHAR2(12) ,
phone VARCHAR2(20) ,
email VARCHAR2(75) ,
manager_id NUMBER(6,0),
emergency_contact VARCHAR2(20)
);
```

Results Explain Describe Saved SQL History

Table created.
0.03 seconds

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

The screenshot shows the Oracle Application Express interface with the 'Results' tab selected. The output area displays the message 'Table created.' and '0.05 seconds'. The top navigation bar includes tabs for Results, Explain, Describe, Saved SQL, and History.

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

QUERY:

```
DESCRIBE f_global_locations;
```

OUTPUT:

The screenshot shows the Oracle Application Express interface with the 'Describe' tab selected. The output area displays the SQL command '1 DESCRIBE f_global_locations;'. Below it is a detailed table summary for the 'F_GLOBAL_LOCATIONS' table, showing columns, data types, lengths, and constraints.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
F_GLOBAL_LOCATIONS	ID	NUMBER	-	6	0	-	✓	-	-
	NAME	VARCHAR2	50	-	-	-	✓	-	-
	DATE_OPENED	DATE	7	-	-	-	✓	-	-
	ADDRESS	VARCHAR2	50	-	-	-	✓	-	-
	CITY	VARCHAR2	30	-	-	-	✓	-	-
	ZIP_POSTAL_CODE	VARCHAR2	12	-	-	-	✓	-	-
	PHONE	VARCHAR2	20	-	-	-	✓	-	-
	EMAIL	VARCHAR2	75	-	-	-	✓	-	-
	MANAGER_ID	NUMBER	-	6	0	-	✓	-	-

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	

QUERY:

```
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)

);
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is active. The main area displays a SQL command to create a table:

```

1 CREATE TABLE f_global_locations
2 ( id NUMBER(6,0) ,
3   name VARCHAR2(50),
4   date_opened DATE ,
5   address VARCHAR2(50) ,
6   city VARCHAR2(30) ,
7   zip_postal_code VARCHAR2(12),
8   phone VARCHAR2(20),
9   email VARCHAR2(75) ,
10  manager_id NUMBER(6,0),
11  emergency_contact VARCHAR2(20));

```

Below the code, the results show 'Table created.' and a execution time of '0.02 seconds'. The tabs at the bottom are 'Results' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'.

PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

1. What is the purpose of a
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK CONSTRAINT

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.

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**

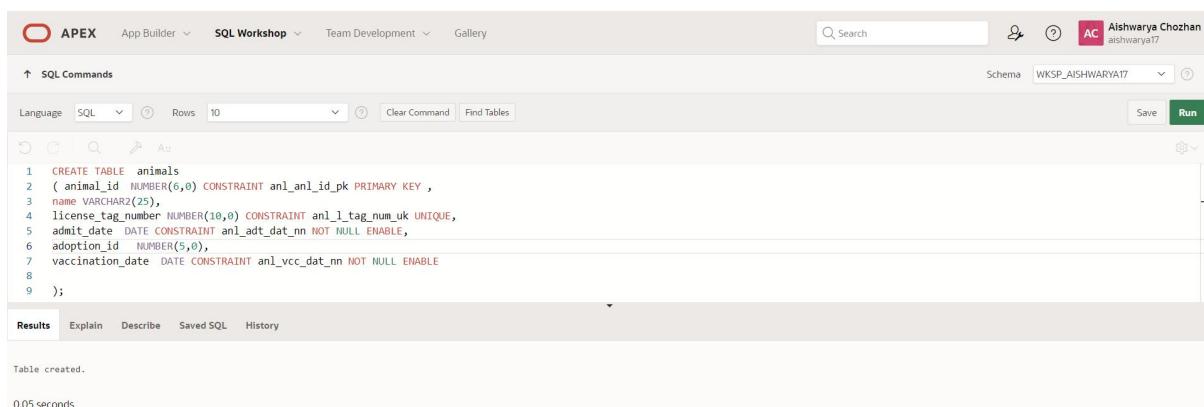
3. Create the animals table. Write the syntax you will use to create the table.

QUERY:

```
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  
);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there's a user profile for 'Aishwarya Chozhan' (ashwarya17). The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below the command input field, there are buttons for 'Save' and 'Run'. The command itself is displayed in the input field:

```
1 CREATE TABLE animals  
2 ( animal_id NUMBER(6,0) CONSTRAINT anl_anl_id_pk PRIMARY KEY ,  
3 name VARCHAR2(25),  
4 license_tag_number NUMBER(10,0) CONSTRAINT anl_l_tag_num_uk UNIQUE,  
5 admit_date DATE CONSTRAINT anl_adt_dat_nn NOT NULL ENABLE,  
6 adoption_id NUMBER(5,0),  
7 vaccination_date DATE CONSTRAINT anl_vcc_dat_nn NOT NULL ENABLE  
8 );
```

Below the command, the 'Results' tab is selected, showing the output: 'Table created.' and '0.05 seconds'.

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

QUERY:

```
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;
```

OUTPUT:

The screenshot shows the Oracle APEX 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 user profile for 'Aishwarya Chozhan' (aishwarya17). The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. The query editor contains the following code:

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

Below the code, the results show:

1 row(s) inserted.
0.05 seconds

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.

QUERY:

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

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

- a. ON DELETE CASCADE
- b. ON DELETE SET NULL

- a. Enables to automatically delete all related records in the child table when the corresponding record in the parent table is deleted. The deleted values will result in a cascade deletion in the animal table.
- b. If a corresponding record in the parent table is deleted, the child table corresponding record will be set to null.

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

- a. Not allowed to use subqueries.
- b. Each row in the table must make the condition True or unknown
- c. Must only be on the row where the constraint is defined

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

RESULT:

Creating views

EX_NO : 13

DATE:16/05/2024

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

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
ID	Song Title	ARTIST
47	Hurrah for Today	The Jubilant Trio
49	Lets Celebrate	The Celebrants

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:16/05/2024

1. Group functions work across many rows to produce one result per group. 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, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and session information 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands' with a 'Run' button. Below it, a code editor shows the SQL command to create a sequence:

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

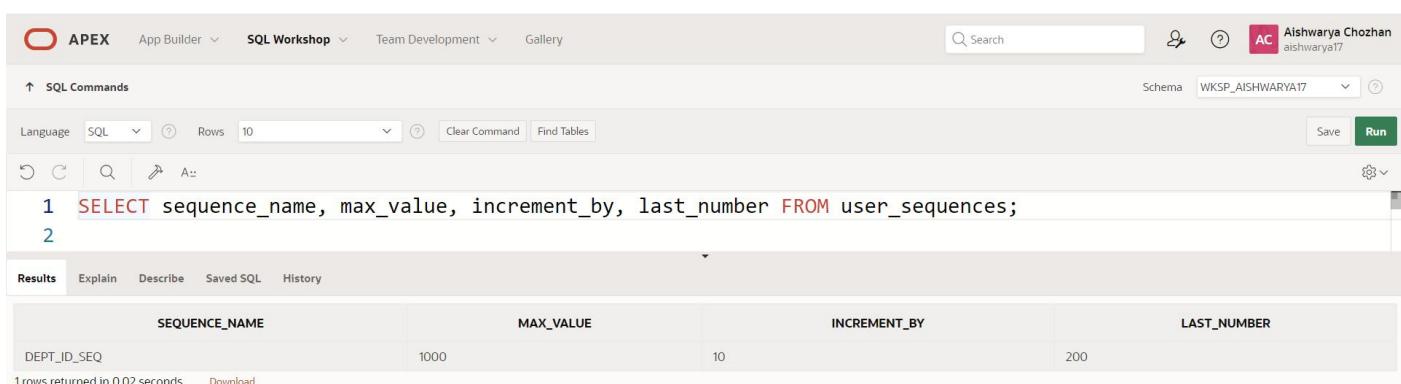
The results tab shows the output: 'Sequence created.' and '0.01 seconds' execution time.

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 same navigation and session context as the previous screenshot. The main area displays the query:

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

The results tab shows the output of the query:

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_ID_SEQ	1000	10	200

At the bottom, it says '1 rows returned in 0.02 seconds' and has a 'Download' link.

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 APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area is titled 'SQL Commands' with a 'Run' button at the bottom right. The command entered is:

```
1 INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education');
2
```

The results tab shows the output: '1 row(s) inserted.' and '0.03 seconds'. The interface has standard toolbar icons for copy, cut, search, and refresh.

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 APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area is titled 'SQL Commands' with a 'Run' button at the bottom right. The command entered is:

```
1 CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);
2
```

The results tab shows the output: 'Index created.' and '0.03 seconds'. The interface has standard toolbar icons for copy, cut, search, and refresh.

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 (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Aishwarya Chozhan', and session information 'Schema WKSP_AISHWARYA17'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';
2
```

Below the code, the results tab is selected, showing a table with three columns: INDEX_NAME, TABLE_NAME, and UNIQUENESS. The data row is:

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

At the bottom left, it says '1 rows returned in 0.08 seconds' and there's a 'Download' link.

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

RESULT:

CONTROLLING USER ACCESS

EX_NO:15

DATE:16/05/2024

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:16

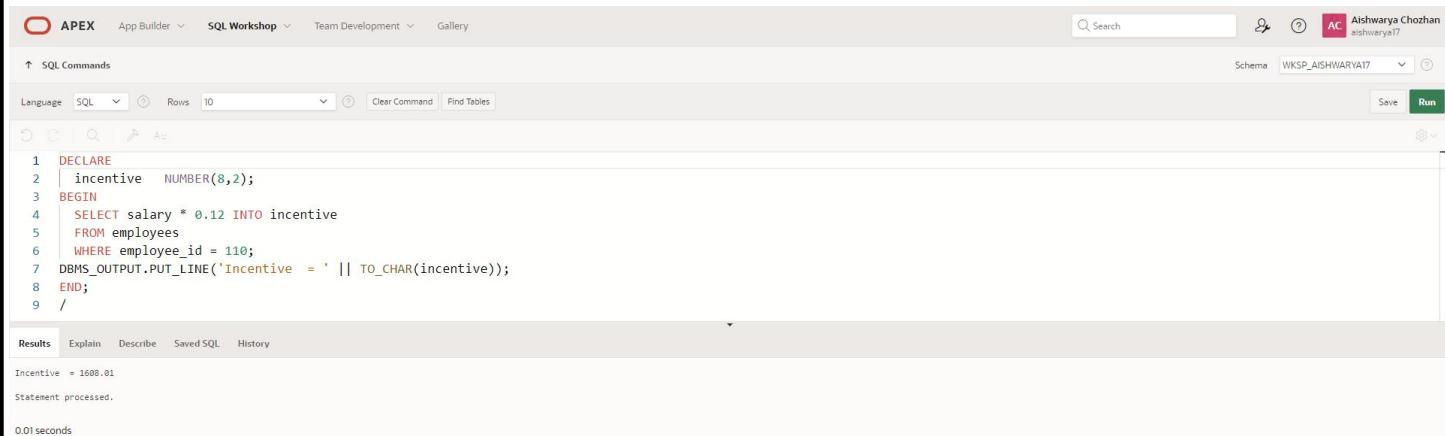
DATE:21/05/2024

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. In the SQL Commands pane, a PL/SQL block is written and executed. The output pane displays the result of the DBMS_OUTPUT.PUT_LINE statement.

```
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;
9  /
```

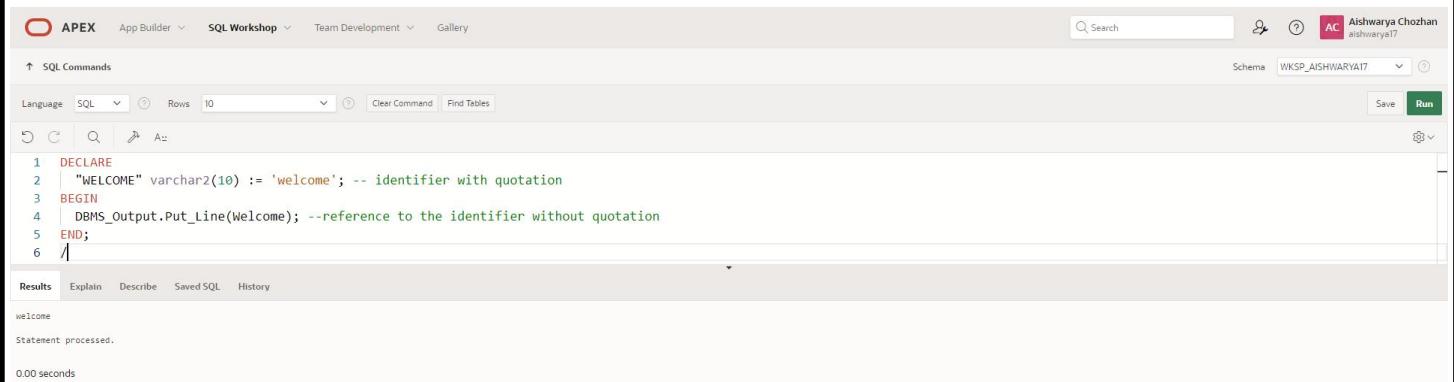
Incentive = 1608.01
Statement processed.
0.01 seconds

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;
/
```

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: Aishwarya Chozhan (aishwaryai7). The main area is titled 'SQL Commands' with tabs for Language (SQL selected), Explain, Describe, Saved SQL, and History. The SQL editor contains the following code:

```
1 DECLARE
2   "WELCOME" varchar2(10) := 'welcome'; -- identifier with quotation
3 BEGIN
4   DBMS_Output.Put_Line(Welcome); --reference to the identifier without quotation
5 END;
6 /
```

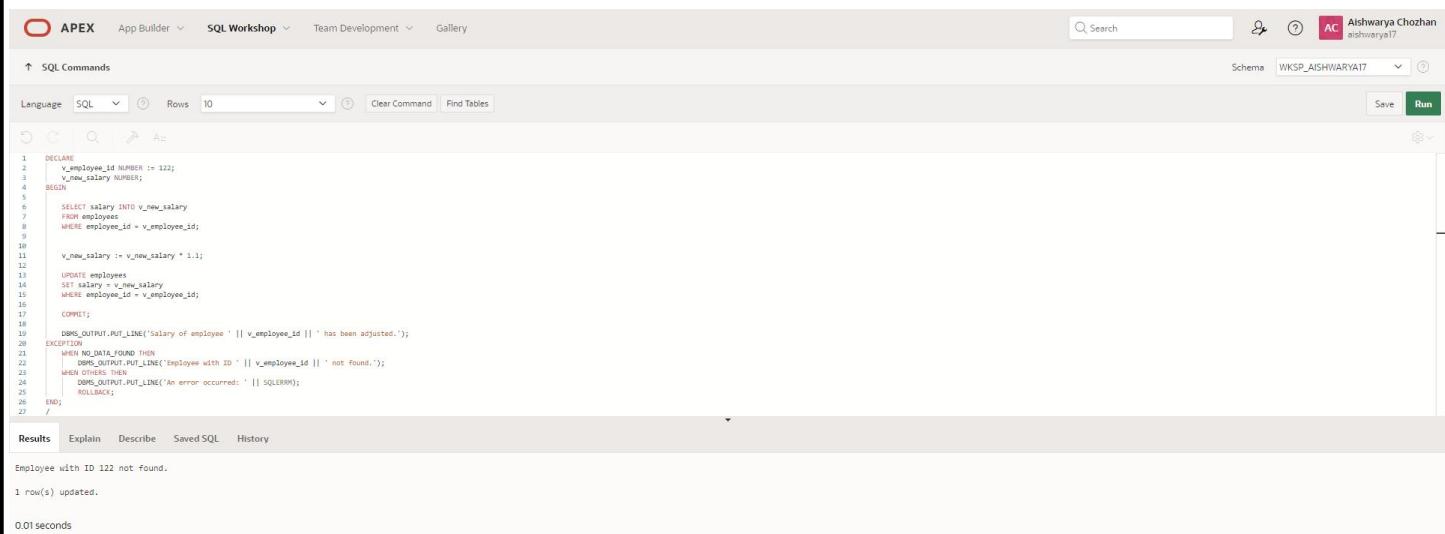
The results tab shows the output: 'welcome'. Below it, a message says 'Statement processed.' and '0.00 seconds'.

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

QUERY:

```
DECLARE
    v_employee_id NUMBER := 122;
    v_new_salary NUMBER;
BEGIN
    SELECT salary INTO v_new_salary
    FROM employees
    WHERE employee_id = v_employee_id;
    v_new_salary := v_new_salary * 1.1;
    UPDATE employees
    SET salary = v_new_salary
    WHERE employee_id = v_employee_id;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Salary of employee ' || v_employee_id || ' has been adjusted.');
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Employee with ID ' || v_employee_id || ' not found.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        ROLLBACK;
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'. On the right, there's a search bar, a schema dropdown set to 'WKSP_AISHWARYA17', and a 'Run' button. The main workspace displays the PL/SQL code from the previous step. The code is executed successfully, with the output showing 'Employee with ID 122 not found.' and '1 row(s) updated.' at the bottom. The status bar at the bottom indicates '0.01 seconds'.

```
1  DECLARE
2      v_employee_id NUMBER := 122;
3      v_new_salary NUMBER;
4  BEGIN
5
6      SELECT salary INTO v_new_salary
7      FROM employees
8      WHERE employee_id = v_employee_id;
9
10     v_new_salary := v_new_salary * 1.1;
11
12     UPDATE employees
13     SET salary = v_new_salary
14     WHERE employee_id = v_employee_id;
15
16     COMMIT;
17
18     DBMS_OUTPUT.PUT_LINE('Salary of employee ' || v_employee_id || ' has been adjusted.');
19
20  EXCEPTION
21      WHEN NO_DATA_FOUND THEN
22          DBMS_OUTPUT.PUT_LINE('Employee with ID ' || v_employee_id || ' not found.');
23      WHEN OTHERS THEN
24          DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
25          ROLLBACK;
26  END;
27 /
```

Results Explain Describe Saved SQL History

Employee with ID 122 not found.
1 row(s) updated.
0.01 seconds

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;
/

DECLARE
  PROCEDURE pri_not_m (
    m BOOLEAN
  ) IS
BEGIN
  pri_bool ('m', m);
  pri_bool ('NOT m', NOT m);
END pri_not_m;

BEGIN
  DBMS_OUTPUT.PUT_LINE('----- FOR m TRUE -----');
  pri_not_m (TRUE);
  DBMS_OUTPUT.PUT_LINE('----- FOR m FALSE -----');
  pri_not_m (FALSE);
  DBMS_OUTPUT.PUT_LINE('----- FOR m NULL -----');
  pri_not_m (NULL);
END;
/
```

OUTPUT:

A screenshot of 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 'Aishwarya Chozhan', and a 'Run' button. The main area shows the following SQL code:

```
1 CREATE OR REPLACE PROCEDURE pri_bool(
2   | boo_name  VARCHAR2,
3   | boo_val    BOOLEAN
4 ) IS
5 BEGIN
6   IF boo_val IS NULL THEN
7     DBMS_OUTPUT.PUT_LINE( boo_name || ' = NULL');
8   ELSIF boo_val = TRUE THEN
9     DBMS_OUTPUT.PUT_LINE( boo_name || ' = TRUE');
10  ELSE
11    DBMS_OUTPUT.PUT_LINE( boo_name || ' = FALSE');
12  END IF;
13 END;
14 /
```

Below the code, the 'Results' tab is selected, showing the message 'Procedure created.' and a execution time of '0.03 seconds'.

A screenshot of the Oracle SQL Workshop interface, similar to the first one but with different code. The top navigation bar and user information are identical. The main area shows the following SQL code:

```
1 DECLARE
2   PROCEDURE pri_not_m (
3     m  BOOLEAN
4   ) IS
5 BEGIN
6   pri_bool ('m', m);
7   pri_bool ('NOT m', NOT m);
8 END pri_not_m;
9
10 BEGIN
11   DBMS_OUTPUT.PUT_LINE('----- FOR m TRUE -----');
12   pri_not_m (TRUE);
13   DBMS_OUTPUT.PUT_LINE('----- FOR m FALSE -----');
14   pri_not_m (FALSE);
15   DBMS_OUTPUT.PUT_LINE('----- FOR m NULL -----');
16   pri_not_m (NULL);
17 END;
18 /
```

Below the code, the 'Results' tab is selected, showing the output of the procedure calls and a execution time of '0.00 seconds'.

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:

```

1  DECLARE
2      PROCEDURE pat_match (
3          test_string    VARCHAR2,
4          pattern        VARCHAR2
5      ) IS
6      BEGIN
7          IF test_string LIKE pattern THEN
8              DBMS_OUTPUT.PUT_LINE ('TRUE');
9          ELSE
10             DBMS_OUTPUT.PUT_LINE ('FALSE');
11         END IF;
12     END;
13 BEGIN
14     pat_match('Blweate', 'B%a_e');
15     pat_match('Blweate', 'B%A_E');
16 END;
17 /

```

Results Explain Describe Saved SQL History

TRUE
FALSE

Statement processed.

0.01 seconds

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:

```

1 DECLARE
2   num_small NUMBER := 8;
3   num_large NUMBER := 5;
4   num_temp NUMBER;
5   BEGIN
6
7   IF num_small > num_large THEN
8     num_temp := num_small;
9     num_small := num_large;
10    num_large := num_temp;
11  END IF;
12
13  DBMS_OUTPUT.PUT_LINE ('num_small = '||num_small);
14  DBMS_OUTPUT.PUT_LINE ('num_large = '||num_large);
15
16 /

```

Results Explain Describe Saved SQL History

```

num_small = 5
num_large = 8
Statement processed.

0.00 seconds

```

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_AISHWARYA17 Run
SQL Commands Language SQL Rows 10 Clear Command Find Tables Save Run
1  DECLARE
2      PROCEDURE test1(
3          sal_achieve NUMBER,
4          target_qty NUMBER,
5          emp_id NUMBER
6      )
7  IS
8      incentive NUMBER := 0;
9      updated VARCHAR2(10) := 'No';
10     BEGIN
11         IF sal_achieve > (target_qty + 200) THEN
12             incentive := (sal_achieve - target_qty)/4;
13         END IF;
14         UPDATE employees
15             SET salary = salary + incentive
16             WHERE employee_id = emp_id;
17         updated := 'Yes';
18     END IF;
19
20     DBMS_OUTPUT.PUT_LINE (
21         'Table updated? ' || updated || ',' ||
22         'incentive = ' || incentive || '!'
23     );
24  END test1;
25
26  test1(2300, 2000, 144);
27  test1(3600, 3000, 145);
28  /
29
30
Results Explain Describe Saved SQL History
Table updated? Yes, incentive = 75.
Table updated? Yes, incentive = 150.
1 row(s) updated.
0.01 seconds

```

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:

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 a schema dropdown ('WKSP_ARTHEY20') are also present. The main area is titled 'SQL Commands' and contains the following PL/SQL code:

```

171 DECLARE
172     PROCEDURE test1 (sal_achieve NUMBER)
173     IS
174         incentive NUMBER := 0;
175     BEGIN
176         IF sal_achieve > 44000 THEN
177             incentive := 1000;
178             DBMS_OUTPUT.PUT_LINE (
179                 'Sale achieved : ' || sal_achieve || ', incentive : ' || incentive || !
180 );
181         ELSEIF
182             incentive < 800;
183             DBMS_OUTPUT.PUT_LINE (
184                 'Sale achieved : ' || sal_achieve || ', incentive : ' || incentive || !
185 );
186         ELSE
187             incentive := 500;
188         END IF;
189     END test1;
190     BEGIN
191         test1(45000);
192         test1(36000);
193         test1(28000);
194     END;

```

Below the code, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab displays the output of the executed code:

```

Sale achieved : 45000, incentive : 1000.
Sale achieved : 36000, incentive : 000.
Sale achieved : 28000, incentive : 500.

```

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:

```

DECLARE
    v_emp_count NUMBER;
    v_vacancies NUMBER := 45;
BEGIN
    -- Count the number of employees in department 50
    SELECT COUNT(*)
    INTO v_emp_count
    FROM employees
    WHERE department_id = 50;

    -- Display the number of employees in department 50
    DBMS_OUTPUT.PUT_LINE('Number of employees in department 50: ' || v_emp_count);

    -- Check if there are any vacancies
    IF v_emp_count < v_vacancies THEN
        DBMS_OUTPUT.PUT_LINE('There are vacancies in department 50.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('There are no vacancies in department 50.');
    END IF;
END;
/

```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, tabs for 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' are visible. A search bar and a schema dropdown ('WKSP_ARTHEY28') are also present. The main area is titled 'SQL Commands'. The code entered is identical to the one above. In the 'Results' tab, the output is displayed:

```

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.

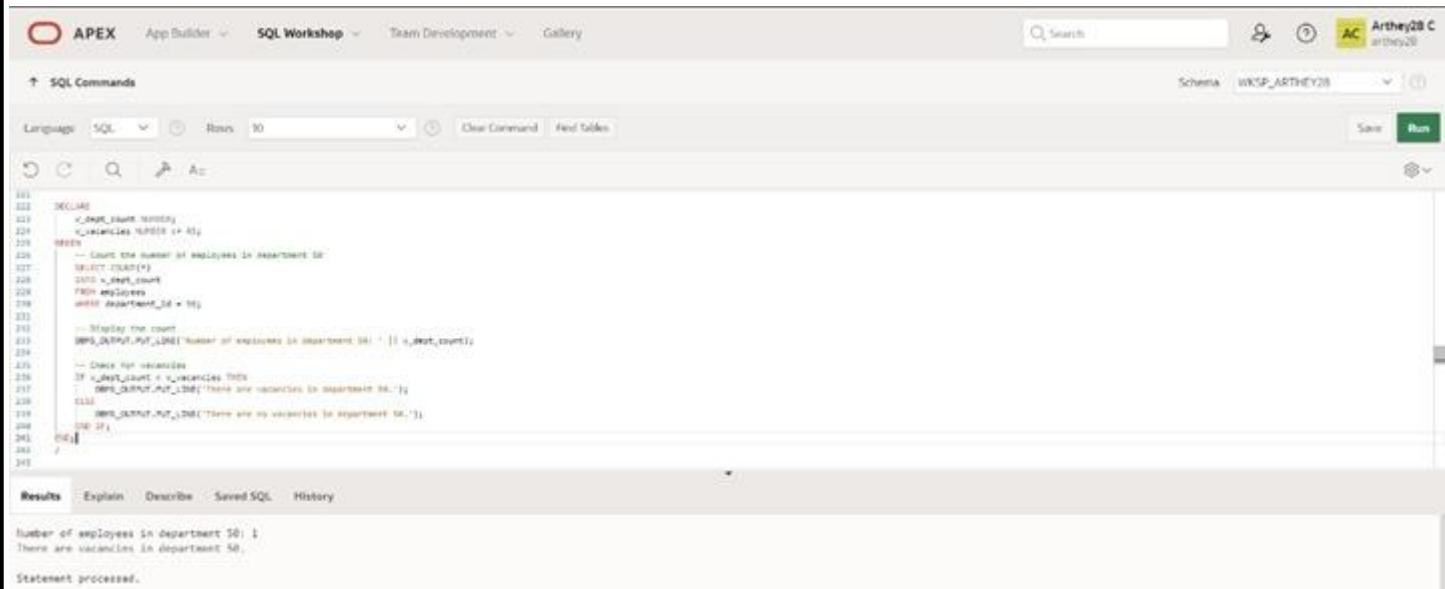
QUERY:

```
DECLARE
    v_dept_count NUMBER;
    v_vacancies NUMBER := 45;
BEGIN
    -- Count the number of employees in department 50
    SELECT COUNT(*)
    INTO v_dept_count
    FROM employees
    WHERE department_id = 50;

    -- Display the count
    DBMS_OUTPUT.PUT_LINE('Number of employees in department 50: ' || v_dept_count);

    -- Check for vacancies
    IF v_dept_count < v_vacancies THEN
        DBMS_OUTPUT.PUT_LINE('There are vacancies in department 50.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('There are no vacancies in department 50.');
    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', and 'Gallery'. The right side shows a user profile 'Arthey28 C' and a schema dropdown 'WKSP_ARTHEY28'. The main area has tabs for 'SQL Commands' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL editor contains the provided PL/SQL code. The results tab shows the output of the executed code, which includes the employee count and a message indicating the presence of vacancies.

```
111  DECLARE
112      v_dept_count NUMBER;
113      v_vacancies NUMBER := 45;
114  BEGIN
115      -- Count the number of employees in department 50
116      SELECT COUNT(*)
117      INTO v_dept_count
118      FROM employees
119      WHERE department_id = 50;

120      -- Display the count
121      DBMS_OUTPUT.PUT_LINE('Number of employees in department 50: ' || v_dept_count);

122      -- Check for vacancies
123      IF v_dept_count < v_vacancies THEN
124          DBMS_OUTPUT.PUT_LINE('There are vacancies in department 50.');
125      ELSE
126          DBMS_OUTPUT.PUT_LINE('There are no vacancies in department 50.');
127      END IF;
128  END;
129  /
```

Results

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

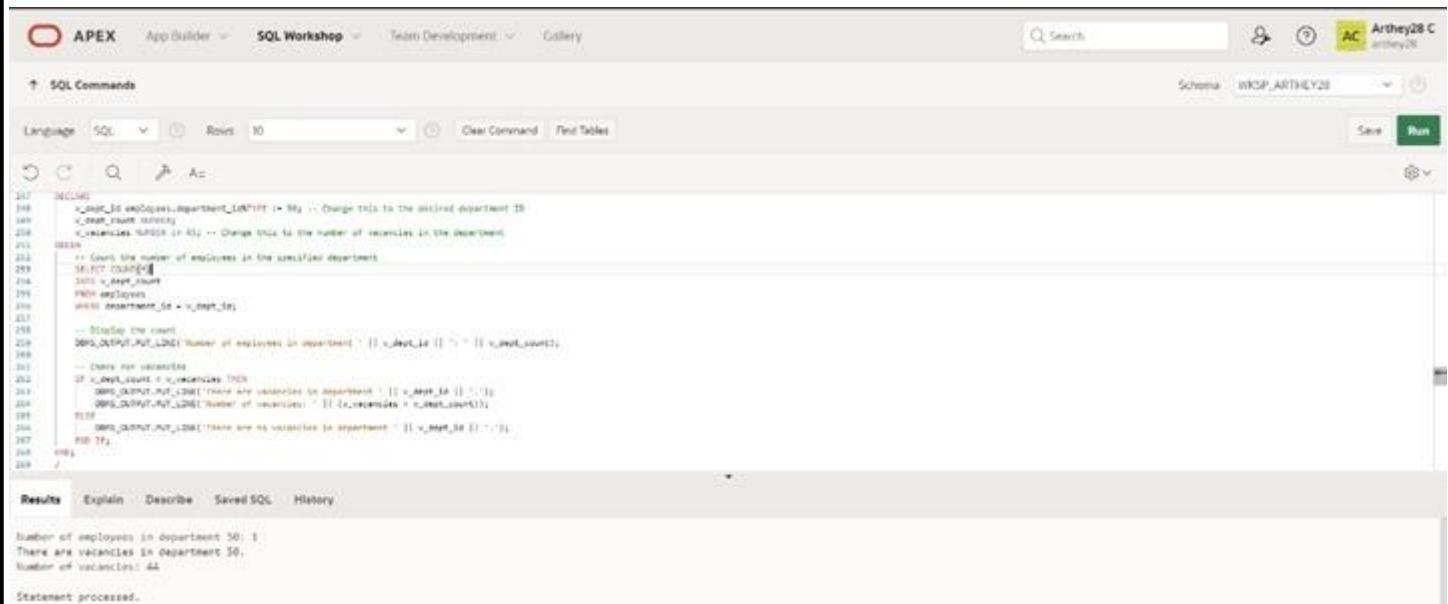
Statement processed.
```

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

QUERY:

```
DECLARE
    v_dept_id employees.department_id%TYPE := 50;
    v_dept_count NUMBER;
    v_vacancies NUMBER := 45;
BEGIN
    SELECT COUNT(*)
    INTO v_dept_count
    FROM employees
    WHERE department_id = v_dept_id;
    DBMS_OUTPUT.PUT_LINE('Number of employees in department ' || v_dept_id || ':' || v_dept_count);
    IF v_dept_count < v_vacancies THEN
        DBMS_OUTPUT.PUT_LINE('There are vacancies in department ' || v_dept_id || '!');
        DBMS_OUTPUT.PUT_LINE('Number of vacancies: ' || (v_vacancies - v_dept_count));
    ELSE
        DBMS_OUTPUT.PUT_LINE('There are no vacancies in department ' || v_dept_id || '!');
    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', and 'Gallery'. The 'Search' field is empty. The 'Schema' dropdown is set to 'WKSP_ARTHEY28'. Below the toolbar, there are buttons for 'Save' and 'Run'. The main workspace displays the PL/SQL code from the previous step. The code uses dynamic SQL to count employees in a specified department and then compares that count against a fixed number of vacancies (45). If the department has more employees than vacancies, it outputs a message indicating the number of vacancies. If the department has fewer or equal employees, it outputs a message indicating there are no vacancies. The code is annotated with comments explaining its purpose. The 'Results' tab at the bottom shows the output of the executed code, which includes the department ID (50), the count of employees (50), and the count of vacancies (44). The status bar at the bottom right indicates 'Statement processed.'

```
107    v_dept_id employees.department_id%TYPE := 50; -- Change this to the desired department
108    v_dept_count NUMBER;
109    v_vacancies NUMBER := 45; -- Change this to the number of vacancies in the department
110
111    -- Count the number of employees in the specified department
112    SELECT COUNT(*)@
113    INTO v_dept_count
114    FROM employees
115    WHERE department_id = v_dept_id;
116
117    ... Display the count
118    DBMS_OUTPUT.PUT_LINE('Number of employees in department ' || v_dept_id || ':' || v_dept_count);
119
120    ... Check for vacancies
121    IF v_dept_count < v_vacancies THEN
122        DBMS_OUTPUT.PUT_LINE('There are vacancies in department ' || v_dept_id || '!');
123        DBMS_OUTPUT.PUT_LINE('Number of vacancies: ' || (v_vacancies - v_dept_count));
124    ELSE
125        DBMS_OUTPUT.PUT_LINE('There are no vacancies in department ' || v_dept_id || '!');
126    END IF;
127
128    /
129
```

Results Explain Describe Saved SQL History

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

Statement processed.
```

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

QUERY:

```
DECLARE
  CURSOR employee_cursor IS
    SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
      FROM employees;
BEGIN
  -- Loop through the cursor and display employee information
  FOR employee_rec IN employee_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
    DBMS_OUTPUT.PUT_LINE('Job Title: ' || employee_rec.job_id);
    DBMS_OUTPUT.PUT_LINE('Hire Date: ' || TO_CHAR(employee_rec.hire_date, 'DD-MON-YYYY'));
    DBMS_OUTPUT.PUT_LINE('Salary: ' || employee_rec.salary);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
END;
/
DECLARE
  CURSOR employee_cursor IS
    SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, d.dept_name
      FROM employees e
        INNER JOIN department d ON e.department_id = d.dept_id;
BEGIN
  FOR employee_rec IN employee_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
    DBMS_OUTPUT.PUT_LINE('Department Name: ' || employee_rec.dept_name);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
END;
/
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables Save Run

```
1 DECLARE
2   CURSOR emp_cursor IS
3     SELECT e.employee_id, e.first_name, e.last_name AS manager_name
4     FROM employees e
5     LEFT JOIN employees m ON e.manager_id = m.employee_id;
6   emp_record emp_cursor%ROWTYPE;
7
8 BEGIN
9
10  OPEN emp_cursor;
11  FETCH emp_cursor INTO emp_record;
12  WHILE emp_cursor%FOUND LOOP
13    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.employee_id);
14    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_record.first_name);
15    DBMS_OUTPUT.PUT_LINE('Manager Name: ' || emp_record.manager_name);
16    DBMS_OUTPUT.PUT_LINE('-----');
17    FETCH emp_cursor INTO emp_record;
18  END LOOP;
19  CLOSE emp_cursor;
20 END;
21 /
22
```

Results Explain Describe Saved SQL History

```
Employee ID: 3
Employee Name: KAVITHA
Manager Name: KUMAR
-----
Employee ID: 2
Employee Name: AMU
Manager Name: AMU
```

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables Save Run

```
294
295  DECLARE
296    CURSOR employee_cursor IS
297      SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, d.dept_name
298      FROM employees e
299      INNER JOIN department d ON e.department_id = d.dept_id;
300
301 BEGIN
302   -- Loop through the cursor and display employee information
303   FOR employee_rec IN employee_cursor LOOP
304     DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
305     DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
306     DBMS_OUTPUT.PUT_LINE('Department Name: ' || employee_rec.dept_name);
307     DBMS_OUTPUT.PUT_LINE('-----');
308   END LOOP;
309 END;
310 /
```

Results Explain Describe Saved SQL History

```
Employee ID: 1
Employee Name: KUMAR DAVIES
Department Name: eee
-----
Employee ID: 110
Employee Name: MARY JAAGADEESH
Department Name: b
```

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 job_id, MIN(salary) AS min_salary
    FROM employees
    GROUP BY job_id;
BEGIN
  -- Loop through the cursor and display job information
  FOR job_rec IN job_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_rec.job_id);
    DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_rec.min_salary);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
END;
/
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'Arthey28' are also present. The main area is titled 'SQL Commands' and contains a code editor with the following PL/SQL script:

```

1  DECLARE
2    CURSOR job_cursor IS
3      SELECT job_id, min(salary) AS min_salary
4      FROM employees
5      GROUP BY job_id;
6  BEGIN
7    -- Loop through the cursor and display job information
8    FOR job_rec IN job_cursor LOOP
9      DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_rec.job_id);
10     DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_rec.min_salary);
11     DBMS_OUTPUT.PUT_LINE('-----');
12   END LOOP;
13 END;
14 /

```

Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying the output of the executed code:

```

Job ID: ST_CLERK
Minimum Salary: 1801.42
-----
Job ID: SA_MGR
Minimum Salary: 3400.52
-----
Statement processed.

```

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

QUERY:

```

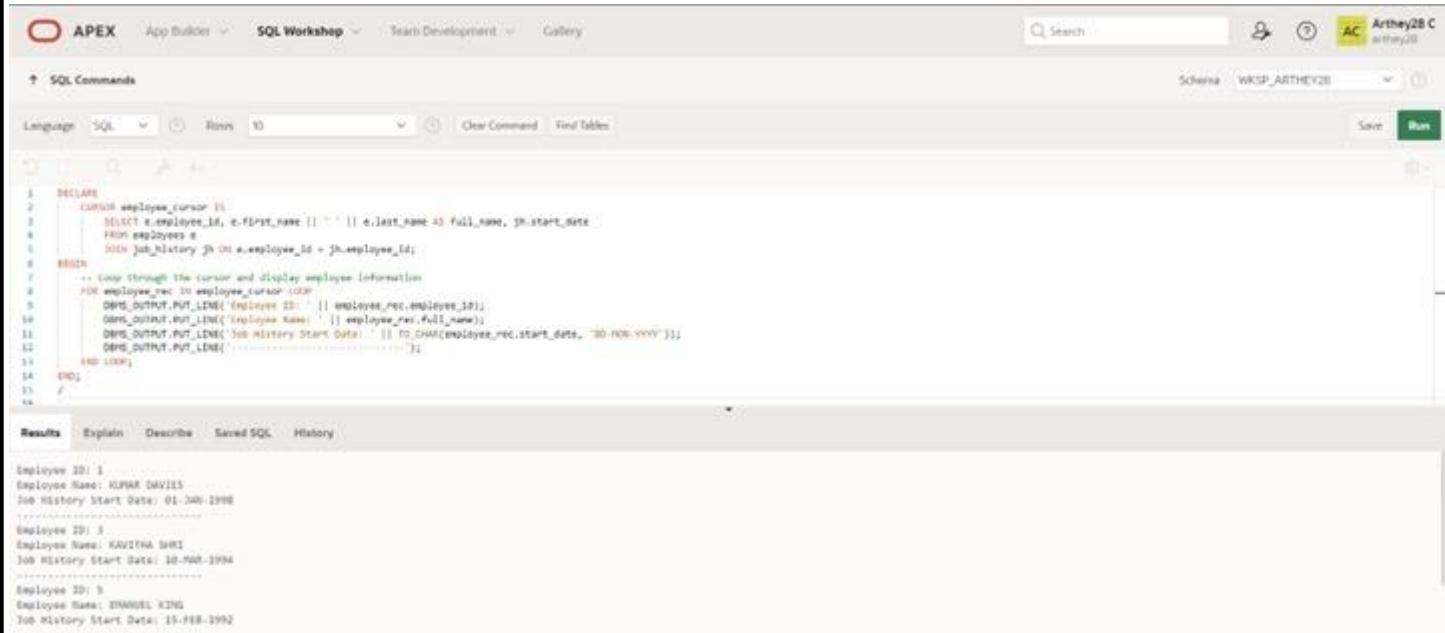
DECLARE
  CURSOR employee_cursor IS
    SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.start_date
    FROM employees e
    JOIN job_history jh ON e.employee_id = jh.employee_id;
BEGIN
  -- Loop through the cursor and display employee information
  FOR employee_rec IN employee_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
    DBMS_OUTPUT.PUT_LINE('Job History Start Date: ' || TO_CHAR(employee_rec.start_date, 'DD-MON-YYYY'));
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;

```

END;

/

OUTPUT:



```
1 DECLARE
2   CURSOR employee_cursor IS
3     SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.start_date
4       FROM employees e
5         JOIN job_history jh ON e.employee_id = jh.employee_id;
6 BEGIN
7   -- Loop through the cursor and display employee information
8   FOR employee_rec IN employee_cursor LOOP
9     DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
10    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
11    DBMS_OUTPUT.PUT_LINE('Job History Start Date: ' || TO_CHAR(employee_rec.start_date, 'DD-MON-YYYY'));
12    DBMS_OUTPUT.PUT_LINE('-----');
13  END LOOP;
14 END;
15 /
```

Employee ID: 1
Employee Name: KUMAR DAVIES
Job History Start Date: 01-JAN-1998

Employee ID: 3
Employee Name: KAVITHA SHRI
Job History Start Date: 18-MAR-1994

Employee ID: 5
Employee Name: DANIEL KING
Job History Start Date: 15-FEB-1992

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

QUERY:

DECLARE

```
  CURSOR employee_cursor IS
    SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.end_date
      FROM employees e
        JOIN job_history jh ON e.employee_id = jh.employee_id;
```

BEGIN

```
  FOR employee_rec IN employee_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);

    IF employee_rec.end_date IS NULL THEN
```

```

    DBMS_OUTPUT.PUT_LINE('Job History End Date: (Still Employed)');
ELSE
    DBMS_OUTPUT.PUT_LINE('Job History End Date: ' || TO_CHAR(employee_rec.end_date, 'DD-MON-
YYYY'));
END IF;

    DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;
END;
/

```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The code in the editor is a PL/SQL block that retrieves employee information from the employees and job_history tables and prints it using DBMS_OUTPUT.PUT_LINE. The results tab shows the output for two employees: employee ID 1 (KUMAR, still employed) and employee ID 3 (KAVITHA, whose history end date is 30-JUN-1995).

```

1  DECLARE
2      CURSOR employee_cursor IS
3          SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.end_date
4          FROM employees e
5              JOIN job_history jh ON e.employee_id = jh.employee_id;
6
7  BEGIN
8      -- Loop through the cursor and display employee information
9      FOR employee_rec IN employee_cursor LOOP
10          DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
11          DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
12
13          -- Check if the end date is null (meaning the employee is currently in the job)
14          IF employee_rec.end_date IS NULL THEN
15              DBMS_OUTPUT.PUT_LINE('Job History End Date: (Still Employed)');
16          ELSE
17              DBMS_OUTPUT.PUT_LINE('Job History End Date: ' || TO_CHAR(employee_rec.end_date, 'DD-MON-YYYY'));
18          END IF;
19
20          DBMS_OUTPUT.PUT_LINE('-----');
21      END LOOP;
22  END;

```

Employee ID	Employee Name	Job History End Date
1	KUMAR, DINESH	01-MAY-1998
3	KAVITHA, SHRI	30-JUN-1995

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

RESULT:

PROCEDURES AND FUNCTIONS

EX_NO: 17

DATE:21/05/2024

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, and Gallery. On the right, there's a user profile for 'Aishwarya Chozhan' (aishwarye17). The main area is titled 'SQL Commands'. The code editor contains a PL/SQL block to calculate the factorial of 10. The results tab shows the output '3628800' and a message 'Statement processed.' The bottom status bar indicates '0.00 seconds'.

```
1  DECLARE
2      fac NUMBER := 1;
3      n NUMBER := :1;
4  BEGIN
5      WHILE n > 0 LOOP
6          fac := n * fac;
7          n := n - 1;
8      END LOOP;
9      DBMS_OUTPUT.PUT_LINE(fac);
10 END;
```

Results Explain Describe Saved SQL History

3628800
Statement processed.
0.00 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 code in the SQL Commands pane is as follows:

```

1  DECLARE
2      v_book_id NUMBER := 1; -- Example book ID
3      v_title VARCHAR2(100);
4      v_author VARCHAR2(100);
5      v_publisher VARCHAR2(100);
6      v_year_published NUMBER;
7      v_status VARCHAR2(20) := 'INITIAL';
8
9      BEGIN
10         -- Call the procedure
11         GET_BOOK_INFO(v_book_id, v_title, v_author, v_publisher, v_year_published, v_status);
12
13         -- Display the results
14         IF v_status = 'FOUND' THEN
15             DBMS_OUTPUT.PUT_LINE('Book Found:');
16             DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
17             DBMS_OUTPUT.PUT_LINE('Author: ' || v_author);
18             DBMS_OUTPUT.PUT_LINE('Publisher: ' || v_publisher);
19             DBMS_OUTPUT.PUT_LINE('Year Published: ' || v_year_published);
20         ELSE
21             DBMS_OUTPUT.PUT_LINE('No book found with ID ' || v_book_id);
22         END IF;
23     END;
24

```

The Results pane displays the output:

```

Book Found:
Title: 1984
Author: George Orwell
Publisher:
Year Published: 1949

Statement processed.

```

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

RESULT:

TRIGGER

EX_NO: 18

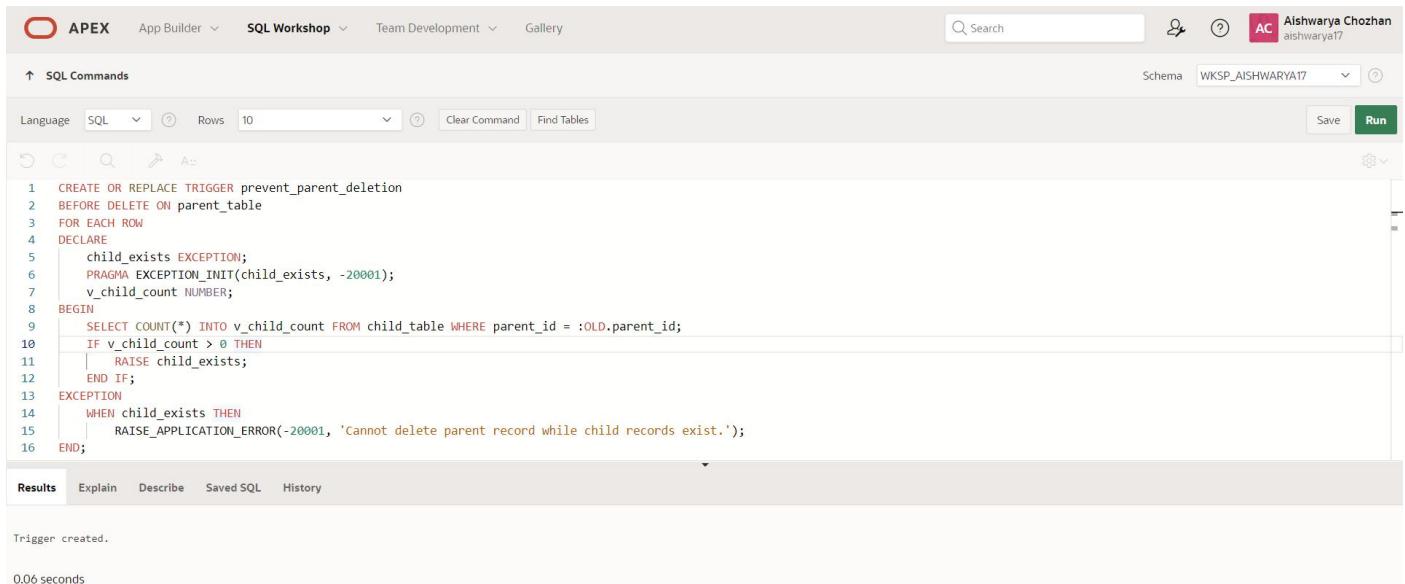
DATE:23/05/2024

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:



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 user icon for 'Aishwarya Chozhan' and a schema dropdown set to 'WKSP_AISHWARYA17'. The main workspace displays the PL/SQL code for the trigger. Below the code, the 'Results' tab is active, showing the message 'Trigger created.' and a execution time of '0.06 seconds'.

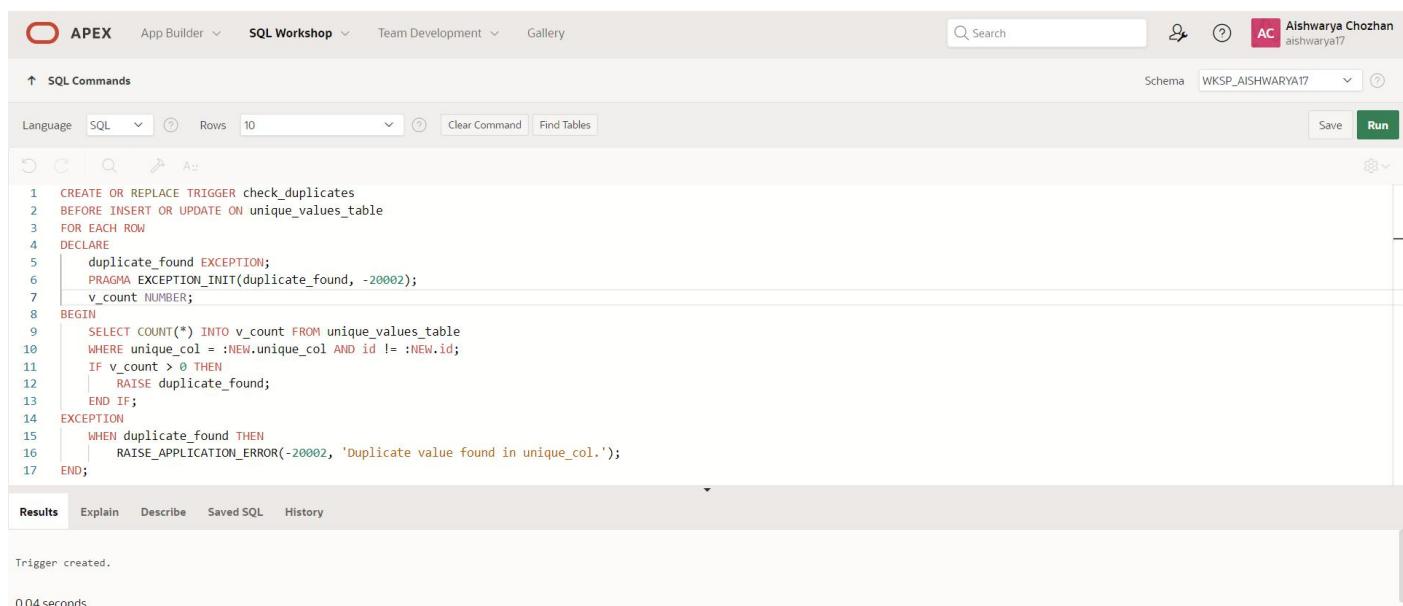
```
1 CREATE OR REPLACE TRIGGER prevent_parent_deletion
2 BEFORE DELETE ON parent_table
3 FOR EACH ROW
4 DECLARE
5     child_exists EXCEPTION;
6     PRAGMA EXCEPTION_INIT(child_exists, -20001);
7     v_child_count NUMBER;
8 BEGIN
9     SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id = :OLD.parent_id;
10    IF v_child_count > 0 THEN
11        RAISE child_exists;
12    END IF;
13 EXCEPTION
14    WHEN child_exists THEN
15        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child records exist.');
16 END;
```

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:



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 user profile for 'Aishwarya Chozhan' (aishwarya17). The main workspace displays the SQL command for creating the trigger, which is identical to the one provided in the text above. Below the command, the 'Results' tab is selected, showing the message 'Trigger created.' and a execution time of '0.04 seconds'. Other tabs like Explain, Describe, Saved SQL, and History are also visible.

```
1 CREATE OR REPLACE TRIGGER check_duplicates
2 BEFORE INSERT OR UPDATE ON unique_values_table
3 FOR EACH ROW
4 DECLARE
5     duplicate_found EXCEPTION;
6     PRAGMA EXCEPTION_INIT(duplicate_found, -20002);
7     v_count NUMBER;
8 BEGIN
9     SELECT COUNT(*) INTO v_count FROM unique_values_table
10    WHERE unique_col = :NEW.unique_col AND id != :NEW.id;
11    IF v_count > 0 THEN
12        RAISE duplicate_found;
13    END IF;
14 EXCEPTION
15    WHEN duplicate_found THEN
16        RAISE_APPLICATION_ERROR(-20002, 'Duplicate value found in unique_col.');
17 END;
```

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:

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, it shows the user 'Aishwarya Chozhan' (aishwarya17) and the schema 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands' and contains the following PL/SQL code:

```

1 CREATE OR REPLACE TRIGGER check_threshold
2 BEFORE INSERT OR UPDATE ON threshold_table
3 FOR EACH ROW
4 DECLARE
5     threshold_exceeded EXCEPTION;
6     PRAGMA EXCEPTION_INIT(threshold_exceeded, -20003);
7     v_sum NUMBER;
8     v_threshold NUMBER := 10000; -- Set your threshold here
9 BEGIN
10    SELECT SUM(value_col) INTO v_sum FROM threshold_table;
11    v_sum := v_sum + :NEW.value_col;
12    IF v_sum > v_threshold THEN
13        RAISE threshold_exceeded;
14    END IF;
15 EXCEPTION
16    WHEN threshold_exceeded THEN
17        RAISE_APPLICATION_ERROR(-20003, 'Threshold exceeded for value_col.');
18 END;

```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab shows the message "Trigger created." and a execution time of "0.04 seconds".

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 APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is highlighted. The main area is titled 'SQL Commands'. The code entered is:

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

The 'Results' tab is selected, showing the output: 'Trigger created.' and '0.04 seconds'. The schema dropdown shows 'WKSP_AISHWARYA17'.

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. In the top navigation bar, 'APEX' is selected. The main area displays the SQL code for creating a trigger:

```

1 CREATE OR REPLACE TRIGGER log_user_activity
2 AFTER INSERT OR UPDATE OR DELETE ON activity_table
3 FOR EACH ROW
4 BEGIN
5   IF INSERTING THEN
6     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
7       VALUES (activity_log_seq.NEXTVAL, 'INSERT', 'activity_table', :NEW.id, SYSTIMESTAMP);
8   ELSIF UPDATING THEN
9     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
10    VALUES (activity_log_seq.NEXTVAL, 'UPDATE', 'activity_table', :NEW.id, SYSTIMESTAMP);
11   ELSIF DELETING THEN
12     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
13       VALUES (activity_log_seq.NEXTVAL, 'DELETE', 'activity_table', :OLD.id, SYSTIMESTAMP);
14   END IF;
15 END;

```

Below the code, the 'Results' tab is selected, showing the message "Trigger created." and a execution time of "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:

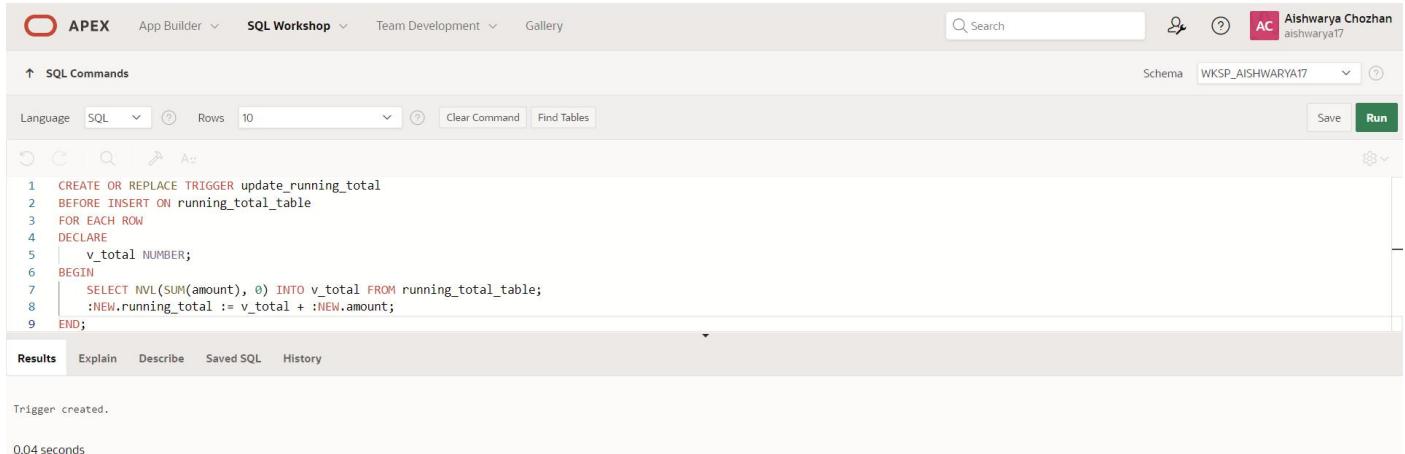
```

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. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop' (which is highlighted in blue), 'Team Development', and 'Gallery'. On the right side, there's a user profile for 'Aishwarya Chozhan' (aishwarya17). The main area is titled 'SQL Commands' and contains the following PL/SQL code:

```
1 CREATE OR REPLACE TRIGGER update_running_total
2 BEFORE INSERT ON running_total_table
3 FOR EACH ROW
4 DECLARE
5     v_total NUMBER;
6 BEGIN
7     SELECT NVL(SUM(amount), 0) INTO v_total FROM running_total_table;
8     :NEW.running_total := v_total + :NEW.amount;
9 END;
```

Below the code, the 'Results' tab is selected, showing the message 'Trigger created.' and a execution time of '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:

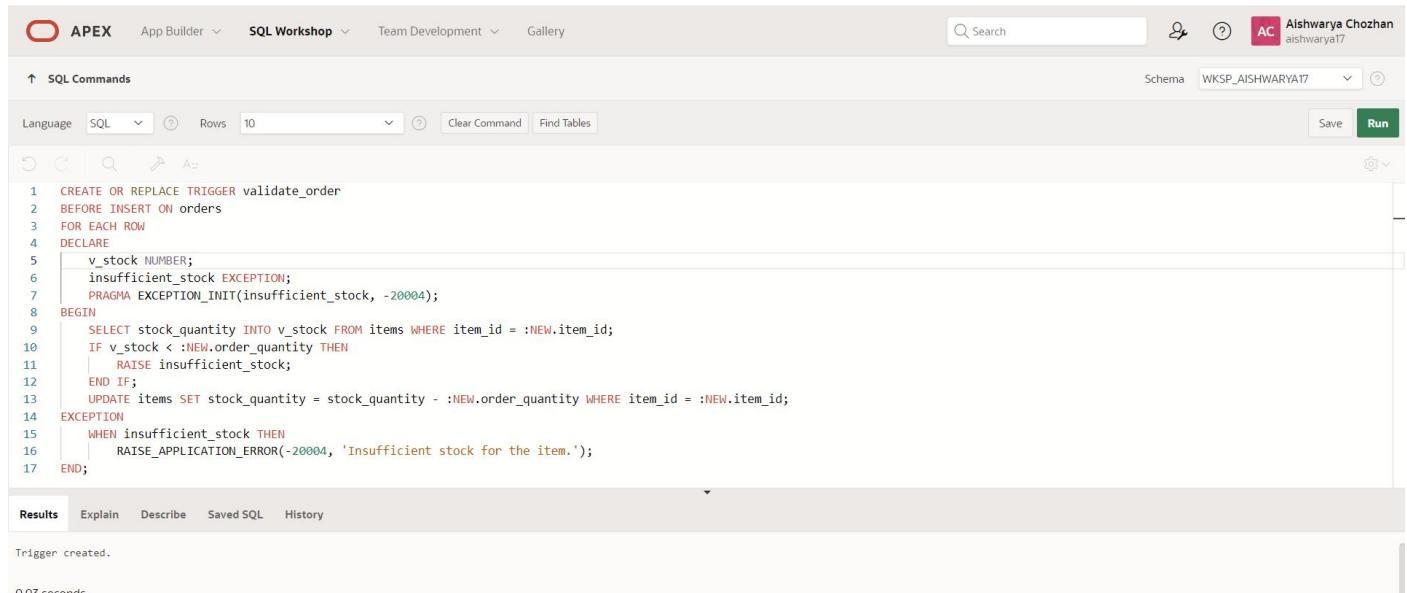
```
CREATE OR REPLACE TRIGGER validate_order
BEFORE INSERT ON orders
FOR EACH ROW
DECLARE
    v_stock NUMBER;
```

```

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:



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 the user profile 'Aishwarya Chozhan' and the schema 'WKSP_AISHWARYA17'. The main area is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is numbered from 1 to 17. The 'Run' button is highlighted in green at the top right of the code editor. Below the code, the 'Results' tab is selected, showing the message 'Trigger created.' and a execution time of '0.03 seconds'.

```

1  CREATE OR REPLACE TRIGGER validate_order
2  BEFORE INSERT ON orders
3  FOR EACH ROW
4  DECLARE
5      v_stock NUMBER;
6      insufficient_stock EXCEPTION;
7      PRAGMA EXCEPTION_INIT(insufficient_stock, -20004);
8  BEGIN
9      SELECT stock_quantity INTO v_stock FROM items WHERE item_id = :NEW.item_id;
10     IF v_stock < :NEW.order_quantity THEN
11         RAISE insufficient_stock;
12     END IF;
13     UPDATE items SET stock_quantity = stock_quantity - :NEW.order_quantity WHERE item_id = :NEW.item_id;
14 EXCEPTION
15     WHEN insufficient_stock THEN
16         RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');
17 END;

```

Trigger created.
0.03 seconds

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

RESULT:

MONGO DB

EX_NO: 19

DATE:24/05/2024

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:

```
aishwarya_17> db.restaurants.find( { $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'
},
{
  _id: ObjectId('564c2d949eb21ad392f1d6ec'),
  borough: 'Brooklyn',
  cuisine: 'Other',
  name: '',
  restaurant_id: '50017910'
},
{
  _id: ObjectId('564c2d949eb21ad392f1d6ed'),
  borough: 'Manhattan',
  cuisine: 'Other',
  name: '',
  restaurant_id: '50017912'
},
{
  _id: ObjectId('564c2d949eb21ad392f1d6f5'),
  borough: 'Brooklyn',
  cuisine: 'Other',
  name: '',
  restaurant_id: '50017925'
}]
```

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:

```
aishwarya_17> db.restaurants.find( { grades: { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 } );
aishwarya_17>
```

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:

```
aishwarya_17> 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 } );
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1})
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({}, { _id: 0 }).sort({ name: 1 });
[{"address: {
  building: '150',
  coord: [ -73.9189864, 40.8654529 ],
  street: 'Post Ave',
  zipcode: '10034'
},
borough: 'Manhattan',
cuisine: 'Other',
grades: [],
name: '',
restaurant_id: '500017887'
},
{"address: {
  building: '508',
  coord: [ -73.999813, 40.683876 ],
  street: 'Henry St',
  zipcode: '11231'
},
borough: 'Brooklyn',
cuisine: 'Other',
grades: [],
name: '',
restaurant_id: '500017910'
},
{"address: {
  building: '18',
  coord: [ -73.9966882, 40.7139264 ],
  street: 'Division St',
  zipcode: '10002'
},
borough: 'Manhattan',
cuisine: 'Other',
grades: [],
name: '',
restaurant_id: '500017912'
},
{"address: {
  building: '4704',
  coord: [ -74.013391, 40.64943 ],
  street: '3rd Ave',
  zipcode: '11228'
},
borough: 'Brooklyn',
cuisine: 'Other',
grades: [],
name: '',
restaurant_id: '500017925'
}]
```

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:

```
aishwarya_17> 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:

```
ashwarya_17> db.restaurants.find({}, { _id: 0 }).sort({ cuisine: 1, borough: -1 })
[
  {
    address: {
      building: '184',
      coord: [ -73.9189864, 40.8654829 ],
      street: 'West 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: '588',
      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: '11228'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017925'
  }
]
```

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:

```

]
aishwarya_17> db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })
[
  {
    _id: ObjectId('564c2d949eb21ad392fld6de'),
    address: {
      building: '15U',
      coord: [ -73.9189064, 40.8654529 ],
      street: 'Post Ave',
      zipcode: '10034'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017887'
  },
  {
    _id: ObjectId('564c2d949eb21ad392fld6ec'),
    address: {
      building: '588',
      coord: [ -73.999813, 40.683876 ],
      street: 'Henry St',
      zipcode: '11231'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017910'
  },
  {
    _id: ObjectId('564c2d949eb21ad392fld6ed'),
    address: {
      building: '15',
      coord: [ -73.9966882, 40.7139264 ],
      street: 'Division St',
      zipcode: '10002'
    },
    borough: 'Manhattan',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017912'
  },
  {
    _id: ObjectId('564c2d949eb21ad392fld6f5'),
    address: {
      building: '4704',
      coord: [ -74.013391, 40.64943 ],
      street: '3rd Ave',
      zipcode: '11228'
    },
    borough: 'Brooklyn',
    cuisine: 'Other',
    grades: [],
    name: '',
    restaurant_id: '50017925'
  }
]
aishwarya_17>

```

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:

```

aishwarya_17> db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
[ {
  _id: ObjectId('564c2d949eb21ad392f1d6de'),
  address: {
    building: '154',
    coord: [ -73.9189864, 40.8654529 ],
    street: 'Post Ave',
    zipcode: '10034'
  },
  borough: 'Manhattan',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017987'
},
{
  _id: ObjectId('564c2d949eb21ad392f1d6ec'),
  address: {
    building: '888',
    coord: [ -73.999813, 40.683876 ],
    street: 'Henry St',
    zipcode: '11231'
  },
  borough: 'Brooklyn',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017910'
},
{
  _id: ObjectId('564c2d949eb21ad392f1d6ed'),
  address: {
    building: '18',
    coord: [ -73.9966882, 40.7139264 ],
    street: 'Division St',
    zipcode: '10082'
  },
  borough: 'Manhattan',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017912'
},
{
  _id: ObjectId('564c2d949eb21ad392f1d6ef5'),
  address: {
    building: '4784',
    coord: [ -74.013391, 40.64943 ],
    street: '88d Ave',
    zipcode: '11220'
  },
  borough: 'Brooklyn',
  cuisine: 'Other',
  grades: [],
  name: '',
  restaurant_id: '50017925'
}
]

```

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:

```

aishwarya_17> db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 });
aishwarya_17>

```

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:

```
aishwarya_17> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
aishwarya_17> |
```

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:

```
aishwarya_17> 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"] } })  
aishwarya_17>
```

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:

```
aishwarya_17> db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })  
aishwarya_17> |
```

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

RESULT:

MONGO DB

EX_NO: 20

DATE:25/05/2024

1.) Find all movies with full information from the 'movies' collection that released in the year 1893.

QUERY:

```
db.movies.find({ year: 1893 })
```

OUTPUT:

```
aishwarya_17> 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')
    }
  }
]
aishwarya_17> |
```

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:

```
aishwarya_17> db.movies.find({ runtime: { $gt: 120 } })
[
  {
    _id: ObjectId('6650f4791f549008efcdcdf6'),
    id: ObjectId('573a1390f22313caabed5967'),
    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', 'édouardMathè', 'Marcel Lévesque', 'Jean Aymé' ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTc1NTY3NDIzNL5BML5BanBnXkFtZTgwNTIyODg5MTE@._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
    }
  }
]
aishwarya_17>
```

3.) Find all movies with full information from the 'movies' collection that have "Short" genre.

QUERY:

```
db.movies.find({ genres: 'Short' })
```

OUTPUT:

```
aishwarya_17> 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')
  }
}
]
aishwarya_17>
```

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:

```
aishwarya_17> 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')
  }
}
]
aishwarya_17> |
```

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:

```
[aishwarya_17]> db.movies.find({ countries: 'USA' })
[{"_id": ObjectId("073a1390f29313caabcd4135"),
  plot: "Three men hammer on an anvil and pass a bottle of beer around.",
  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("1893-05-07T00:00:00.000Z"),
  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("073a1390f29313caabcd4005"),
  id: ObjectId("073a1390f29313caabcd4005"),
  plot: "A married farmer falls under the spell of a stately woman from the city, who tries to convince him to drown his wife.",
  genres: ["Drama", "Romance"],
  runtime: 90,
  cast: ["George O'Brien",
    "Janet Gaynor",
    "Margaret Livingston",
    "Woolf Brining"],
  num_mflix_comments: 1,
  title: "Song of Two Humans",
  fullplot: "In this fable-morality subtitled "A Song of Two Humans", the "evil" temptress is a city woman who bewitches Farmer Anse and tries to convince him to murder his neglected wife, Indra.",
  countries: ["U.S.A."],
  released: ISODate("1927-11-06T00:00:00.000Z"),
  directors: ["F.W. Murnau"],
  writers: [
    "F.W. Murnau (scenario)",
    "Hermann Sudermann (from an original theme by)",
    "Walter Röhr (titles)",
    "H.W. Caldwell (titles)"
  ],
  awards: {
    wins: 0,
    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: 24086, id: 18455 },
  tomatoes: {},
  viewer: { rating: 8.4, numReviews: 9130, meter: 92 },
  critic: { rating: 8.9, numReviews: 68, meter: 98 },
  lastupdated: ISODate("2015-09-16T19:15:02.000Z"),
  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."}]
```

6.) Retrieve all movies from the 'movies' collection that have complete information and are rated as "UNRATED".

QUERY:

```
db.movies.find({ rated: 'UNRATED' })
```

OUTPUT:

```
[aishwarya_17]> 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:

```
aishwarya_17> db.movies.find({ 'imdb.votes': { $gt: 1000 } })  
[  
  {  
    _id: ObjectId('573a1390f29313caabed4138'),  
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',  
    genres: [ 'Short' ],  
    runtime: 1,  
    cast: [ 'Charles Kaysen', 'John Ott' ],  
    num_wFlix_comments: 1,  
    title: 'Hammer 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('1893-05-09T00:00:00Z'),  
    directors: [ 'William K.L. Dickson' ],  
    type: 'UNRATED',  
    awards: { wins: 1, nominations: 0, text: '1 win.' },  
    lastUpdated: '2015-08-26 00:03:50.130000000',  
    year: 1893,  
    imbd: { rating: 6.2, votes: 1189, id: 5 },  
    type: ' ',  
    tomatoes: {  
      viewer: { rating: 8, numReviews: 184, meter: 32 },  
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')  
    },  
    _id: ObjectId('6658f4791f549008efcdcf6'),  
    id: ObjectId('573a1390f21313caabed599e7'),  
    plot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',  
    genres: [ 'Action', 'Adventure', 'Crime' ],  
    runtime: 110,  
    rated: 'NOT RATED',  
    cast: [ 'Musidora', 'édouard Mathé', 'Jean Aymé' ],  
    poster: 'https://m.media-amazon.com/images/MV5BNTc1NTY3NDIzMjE5MjQyNzE0LVEtSY1000_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.',  
    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 00:24:27.330000000',  
    year: 1916,  
    imbd: { rating: 6.8, votes: 2878, id: 6206 },  
    countries: [ 'France' ],  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 3.8, numReviews: 2118, meter: 82 },  
      dvd: ISODate('2000-05-16T00:00:00Z'),  
      critic: { rating: 8.8, numReviews: 13, meter: 100 },  
      lastUpdated: ISODate('2015-09-15T17:02:33.000Z'),  
      rotten: 0,  
      fresh: 13  
    }  
]
```

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:

```
aishwarya_17> db.movies.find({ 'imdb.rating': { $gt: 7 } })  
[  
  {  
    _id: ObjectId('5656f4f21f549008efcdcd7'),  
    id: ObjectId('573a1391f29313caabed8945'),  
    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",  
      "Basil Roscoe"  
    ],  
    num_wFlix_comments: 1,  
    poster: 'https://m.media-amazon.com/images/MV5BNDVkyMwR2ItNzRiMy00IMQ4LTlHmjMzNDI1ZDYyOGVmMzJjXkEyXkFqcGdeQXVyNTgzMzU5MDI0LVEtSY1000_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 Arnes 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)',  
      'Alexander Willoughby (titles)',  
      'W.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,  
    imbd: { rating: 8.4, votes: 24488, id: 18455 },  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 4.4, numReviews: 9134, meter: 92 },  
      dvd: ISODate('2008-12-09T00:00:00Z'),  
      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  
    },  
    _id: ObjectId('573a1391f29313caabed599e7'),  
    id: ObjectId('573a1390f21313caabed599e7'),  
    plot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',  
    genres: [ 'Action', 'Adventure', 'Crime' ],  
    runtime: 110,  
    rated: 'NOT RATED',  
    cast: [ 'Musidora', 'édouard Mathé', 'Jean Aymé' ],  
    poster: 'https://m.media-amazon.com/images/MV5BNTc1NTY3NDIzMjE5MjQyNzE0LVEtSY1000_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.',  
    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 00:24:27.330000000',  
    year: 1916,  
    imbd: { rating: 6.8, votes: 2878, id: 6206 },  
    countries: [ 'France' ],  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 3.8, numReviews: 2118, meter: 82 },  
      dvd: ISODate('2000-05-16T00:00:00Z'),  
      critic: { rating: 8.8, numReviews: 13, meter: 100 },  
      lastUpdated: ISODate('2015-09-15T17:02:33.000Z'),  
      rotten: 0,  
      fresh: 13  
    }  
}]  
aishwarya_17>
```

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:

```
aishwarya_17> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
[
  {
    _id: ObjectId('6650f4f21f549008efcdcdf7'),
    id: ObjectId('973a1391f29313caabcd8945'),
    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/MVSBNDVkyMwM2ItNzRimy00NWQHLTlHMyMtNDIIZDyOGVmMzJjXhEyXhFqcGdeQXVyNTgzMzUSMDI0.._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('1927-11-04T00:00:00Z'),
    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: 2480, id: 18485 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 4.4, numReviews: 9134, meter: 92 },
      dvd: ISODate('1998-12-09T00:00:00.000Z'),
      critic: { rating: 8.3, 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
    }
  }
]
aishwarya_17>
```

10.) Retrieve all movies from the 'movies' collection that have received an award.

QUERY:

```
db.movies.find({ 'awards.wins': { $gt: 0 } })
```

OUTPUT:

```
aishwarya_17> db.movies.find({ 'awards.wins': { $gt: 0 } })
[
  {
    _id: ObjectId('973a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around',
    genres: ['Short'],
    runtime: 10,
    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 rounds, the metal goes back into 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('1892-05-09T00:00:00Z'),
    directors: ['William H.L. Dickson'],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:05:00.133000000',
    year: 1892,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 104, meter: 32 },
      lastupdated: ISODate('2015-08-28T16:34:09.000Z')
    }
  },
  {
    _id: ObjectId('6650f4f21f549008efcdcdf7'),
    id: ObjectId('973a1391f29313caabcd8945'),
    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/MVSBNDVkyMwM2ItNzRimy00NWQHLTlHMyMtNDIIZDyOGVmMzJjXhEyXhFqcGdeQXVyNTgzMzUSMDI0.._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('1927-11-04T00:00:00Z'),
    directors: ['F.W. Murnau'],
    writers: [
      'Carl Mayer (scenario)',
      'Hermann Sudermann (from an original theme by)'
    ],
    awards: {
      wins: 5,
      nominations: 1,
      text: 'Won 3 Oscars. Another 2 wins & 1 nomination.'
    }
  }
]
```

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:

```
aishwarya_17> 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('6650f4791f549008efcdcdf6'),
    genres: [ 'Action', 'Adventure', 'Crime' ],
    runtime: 399,
    cast: [ 'Musidora', 'édouardMathé', 'Marcel Lèvesque', 'Jean Aymé' ],
    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('6650f4f21f549008efcdcdf7'),
    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
  }
]
```

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:

```
aishwarya_17> 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
  }
]
aishwarya_17>
```

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:

```
aishwarya_17> 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' ]
}
]
aishwarya_17>
```

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:

```
aishwarya_17> 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' ]
}
]
aishwarya_17>
```

Evaluation Procedure	Marks awarded
----------------------	---------------

Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT: