

Creating and Managing Tables

EX_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 DEPT30(id number(7),name varchar2(25));

OUTPUT:

```
SQL Commands
Language: SQL Rows: 10
Clear Command Find Tables
Save Run
1 Create table DEPT30(
2   ID number(7),
3   NAME Varchar(25)
4 );
```

Results Explain Describe Saved SQL History

Table created.
0.04 seconds

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

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

QUERY:

Create table DEPT2(ID number(7),Last_Name varchar2(25),First_Name varchar2(25),Dept_id number(7));

OUTPUT:

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

```
1 Create table DEPT2(
2   ID number(7),
3   LAST_NAME Varchar(25),
4   FIRST_NAME Varchar(25),
5   DEPT_ID Number(7)
6 );
```

In the Results tab, the output shows:

```
Table created.  
0.05 seconds
```

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

QUERY:

Alter table DEPT2 modify(Last_Name varchar2(25));

OUTPUT:

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

```
1 Create table DEPT2(
2   ID number(7),
3   FIRST_NAME Varchar(25),
4   LAST_NAME Varchar(25),
5   DEPT_ID Number(7),
6 );
7 alter table DEPT2
8   modify LAST_NAME Varchar(50);
```

In the Results tab, the output shows:

```
Table altered.  
0.04 seconds
```

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

QUERY:

Create table EMP2(id number(7),first_name varchar2(25),Last_name varchar2(25),Salary number(10),Dept_id number(7));

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_ARVINDH30'. The main area contains the following SQL code:

```
1 Create Table DEPT2(
2     ID number(7),
3     FIRST_NAME Varchar(25),
4     LAST_NAME Varchar(25),
5     DEPT_ID Number(7),
6 );
7 alter table DEPT2
8 modify LAST_NAME Varchar(50);
9
10 create table EMP2(
11     Id number(10),
12     FIRST_NAME Varchar(25),
13     LAST_NAME Varchar(25),
14     SALARY number(10),
15     DEPT_ID Number(7)
16 );
```

The results pane at the bottom displays the message "Table created." and "0.02 seconds".

5.Drop the EMP table.

QUERY:

Drop table DEPT2;

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_ARVINDH30'. The main area contains the following SQL code:

```
1 Create Table DEPT2(
2     ID number(7),
3     FIRST_NAME Varchar(25),
4     LAST_NAME Varchar(25),
5     DEPT_ID Number(7),
6 );
7 alter table DEPT2
8 modify LAST_NAME Varchar(50);
9
10 create table EMP2(
11     Id number(10),
12     FIRST_NAME Varchar(25),
13     LAST_NAME Varchar(25),
14     SALARY number(10),
15     DEPT_ID Number(7)
16 );
17 Drop table DEPT2;
```

The results pane at the bottom displays the message "Table dropped." and "0.07 seconds".

6.Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename EMP2 to DEPT2;

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 (Arvindh.V 50) and the schema (WKSP_ARVINDH30). The main area is a SQL editor with the following code:

```
1 Create table DEPT2(
2     ID number(7),
3     FIRST_NAME Varchar(25),
4     LAST_NAME Varchar(25),
5     DEPT_ID Number(7),
6 );
7 alter table DEPT2
8     modify LAST_NAME Varchar(50);
9
10 create Table EMP2(
11     Id number(10),
12     FIRST_NAME Varchar(25),
13     LAST_NAME Varchar(25),
14     SALARY number(18),
15     DEPT_ID Number(7)
16 );
17 Drop table DEPT2;
18 Rename EMP2 to DEPT2;
```

The results panel at the bottom shows "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 DEPT1 is 'Department info';

comment on table DEPT2 is Employee info';

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 (Arvindh.V 50) and the schema (WKSP_ARVINDH30). The main area is a SQL editor with the following code:

```
1 comment on table DEPT1 is 'Department info';
2
3 
```

The results panel at the bottom shows "Statement processed." and "0.01 seconds". A watermark for "Activate Windows" is visible in the bottom right corner.

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. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for Arvindh.V0 with session ID arvindh50. The main area is titled 'SQL Commands' with tabs for Language (SQL selected), Rows (10), Clear Command, and Find Tables. Below these are icons for Undo, Redo, Search, and Run. The schema dropdown is set to WKSP_ARVINDH50. The code editor contains a script with several lines of SQL, including table creation, modification, and alteration statements, followed by a 'Drop column FIRST_NAME;' command at line 28. The results tab is active, showing the output: 'Table altered.' and a duration of '0.06 seconds'. Other tabs available are Explain, Describe, Saved SQL, and History.

```
1 Create table DEPT2(
2     ID number(7),
3     FIRST_NAME Varchar(25),
4     LAST_NAME Varchar(25),
5     DEPT_ID Number(7),
6 );
7 alter table DEPT2
8     modify LAST_NAME Varchar(50);
9
10 create table EMP2(
11     ID number(10),
12     FIRST_NAME Varchar(25),
13     LAST_NAME Varchar(25),
14     SALARY number(10),
15     DEPT_ID Number(7)
16 );
17 Drop table DEPT2;
18 Rename EMP2 to DEPT2;
19 Alter_table DEPT2
20 Drop column FIRST_NAME;
```

Results | Explain | Describe | Saved SQL | History

Table altered.
0.06 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

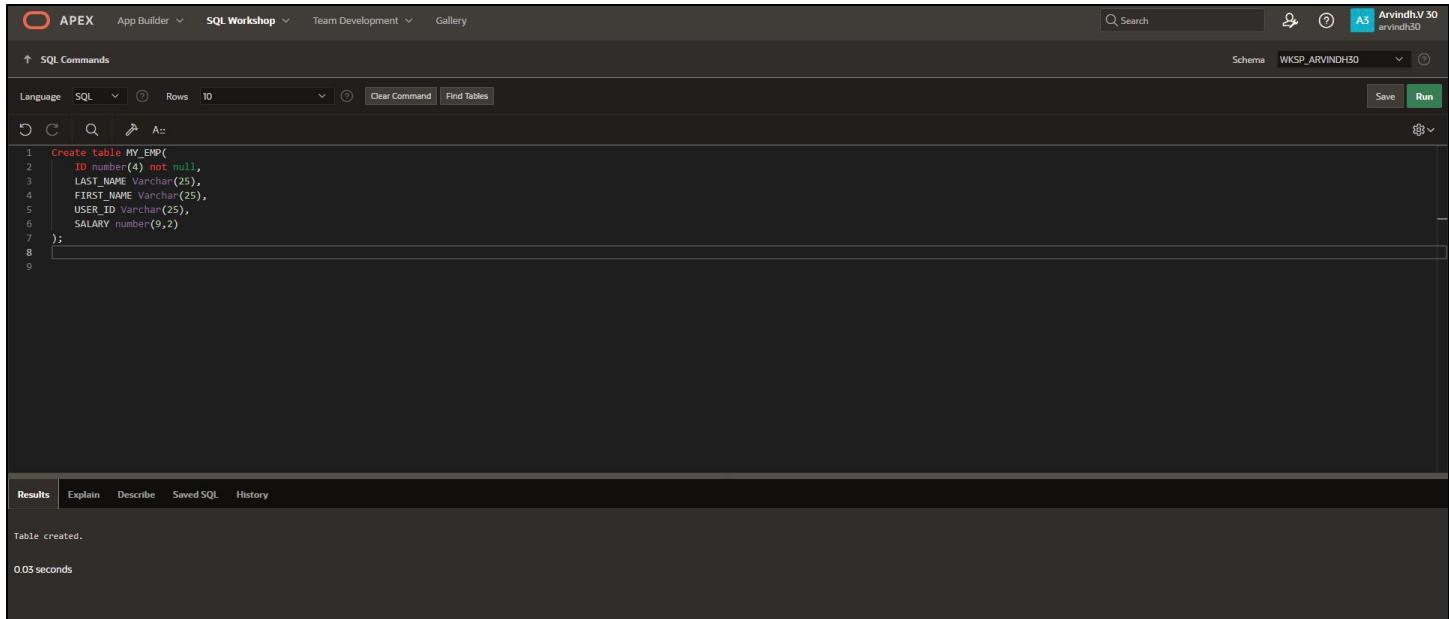
1.Create MY_EMPLOYEE table with the following structure

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_EMP(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 SQL Commands pane, the following SQL code is entered:

```
1 Create Table MY_EMP(
2   ID number(4) not null,
3   LAST_NAME Varchar(25),
4   FIRST_NAME Varchar(25),
5   USER_ID Varchar(25),
6   SALARY number(9,2)
7 );
```

In the Results pane, the output is:

```
Table created.  
0.03 seconds
```

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

ID	Last_name	First_name	Userid	salary
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	Cnewman	750
5	Ropebur	Audrey	arophebur	1550

QUERY:

```
Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
```

```
Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows the user 'Arvindh.V30' and workspace 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), and Clear Command. There are buttons for Save and Run. Below the tabs, there are icons for Undo, Redo, Find, and Replace. The SQL editor contains the following code:

```
1 Create table MY_EMP(
2   ID number(4) not null,
3   LAST_NAME Varchar(25),
4   FIRST_NAME Varchar(25),
5   USER_ID Varchar(25),
6   SALARY number(9,2)
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
10 Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','arophebur',1550);
13 Select * from MY_EMP
14
15
```

The results section shows the message '1 row(s) inserted.' and '0.00 seconds'.

3. Display the table with values.

QUERY:

Select * from MY_EMP;

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 contains the following SQL code:

```
1 Create table MY_EMP(
2     ID number(4) not null,
3     LAST_NAME Varchar(25),
4     FIRST_NAME Varchar(25),
5     USER_ID Varchar(25),
6     SALARY number(9,2)
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
10 Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
13 Select * From MY_EMP
14
```

Below the code, there is a results grid showing the data inserted into the table:

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

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_EMP values(3,'Biri','Ben','bbiri',1100);

Insert into MY_EMP values(4,'Newman','chad','Cnewman',750);

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 contains the following SQL code:

```
1 Create table MY_EMP(
2     ID number(4) not null,
3     LAST_NAME Varchar(25),
4     FIRST_NAME Varchar(25),
5     USER_ID Varchar(25),
6     SALARY number(9,2)
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
10 Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
13 Select * from MY_EMP
14
15
```

Below the code, the results show the insertion of a new row:

1 row(s) inserted.

0.00 seconds

5. Make the data additions permanent.

QUERY:

Select * from MY_EMP;

OUTPUT:

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

```
1 Create Table MY_EMP(
2   ID number(4) not null,
3   LAST_NAME Varchar(25),
4   FIRST_NAME Varchar(25),
5   USER_ID Varchar(25),
6   SALARY number(9,2),
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
10 Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','arophebur',1550);
13 Select * from MY_EMP
```

Below the code, the 'Results' tab is selected, showing the following table:

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

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

QUERY:

Update MY_EMP set last_name = 'Drexler' where ID = '3'

OUTPUT:

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

```
1 Create Table MY_EMP(
2   ID number(4) not null,
3   LAST_NAME Varchar(25),
4   FIRST_NAME Varchar(25),
5   USER_ID Varchar(25),
6   SALARY number(9,2),
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
10 Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','arophebur',1550);
13 Select * from MY_EMP
14 Update MY_EMP set LAST_NAME = 'Drexler' where ID = '3'
```

Below the code, the 'Results' tab is selected, showing the message:

1 row(s) updated.
0.01 seconds

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

QUERY:

Update MY_EMP 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 SQL Workshop tab is selected. The schema dropdown shows 'WKSP_ARVINDH30'. The main area displays the following SQL code:

```
1 Create table MY_EMP(
2   ID number(4) not null,
3   LAST_NAME Varchar(25),
4   FIRST_NAME Varchar(25),
5   USER_ID Varchar(25),
6   SALARY number(9,2)
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
10 Insert into MY_EMP values(3,'Biril','Ben','bbiril',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
13 Select * from MY_EMP
14 Update MY_EMP set LAST_NAME = 'Drexler' where ID = '3';
15 Update MY_EMP set SALARY = '10000' where SALARY<900
16
```

Below the code, the results section shows:

3 row(s) updated.
0.00 seconds

8.Delete Betty dancs from MY_EMPLOYEE table.

QUERY:

Delete from MY_EMP where LAST_NAME = 'Dancs'

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_ARVINDH30'. The main area displays the following SQL code:

```
1 Create table MY_EMP(
2   ID number(4) not null,
3   LAST_NAME Varchar(25),
4   FIRST_NAME Varchar(25),
5   USER_ID Varchar(25),
6   SALARY number(9,2)
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
10 Insert into MY_EMP values(3,'Biril','Ben','bbiril',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
13 Select * from MY_EMP
14 Update MY_EMP set LAST_NAME = 'Drexler' where ID = '3';
15 Update MY_EMP set SALARY = '10000' where SALARY<900
16 Delete from MY_EMP where LAST_NAME = 'Dancs'
```

Below the code, the results section shows:

1 row(s) deleted.
0.01 seconds

9.Empty the fourth row of the emp table.

QUERY:

Delete from MY_EMP where ID = '4'

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays a block of SQL code. The code starts with creating a table 'MY_EMP' with columns ID, LAST_NAME, FIRST_NAME, USER_ID, and SALARY. It then inserts five rows of data. Following the insertions, it performs several updates: changing the last name of employee ID 3 to 'Drexler', increasing the salary of employees with IDs 4 and 5 by 10000, and deleting employees with last names 'Dances' and '4'. Finally, it deletes the employee with ID 4. The results section shows that 1 row was deleted in 0.01 seconds.

```
1 Create Table MY_EMP(
2     ID number(4) not null,
3     LAST_NAME Varchar(25),
4     FIRST_NAME Varchar(25),
5     USER_ID Varchar(25),
6     SALARY number(9,2)
7 );
8 Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
9 Insert into MY_EMP values(2,'Dances','Betty','bdances',860);
10 Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
11 Insert into MY_EMP values(4,'Newman','Chad','cnewman',750);
12 Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
13 Select * from MY_EMP;
14 Update MY_EMP set LAST_NAME = 'Drexler' where ID = '3';
15 Update MY_EMP set SALARY = '10000' where SALARY<900;
16 Delete from MY_EMP where LAST_NAME = 'Dances';
17 Delete from MY_EMP where ID = '4';

```

Results | Explain | Describe | Saved SQL | History

1 row(s) deleted.
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:

Alter table DEPT2 add Constraint my_emp_id_pk Primary key(id);

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'alter table DEPT2 add Constraint my_emp_id_pk Primary key(id);'. Below the code, the 'Results' tab is active, showing the output: 'Table altered.' and '0.07 seconds'. The schema is set to 'WKSP_ARVINDH30'.

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 DEPT1 add Constraint my_dept_id_pk Primary key(id);

OUTPUT:

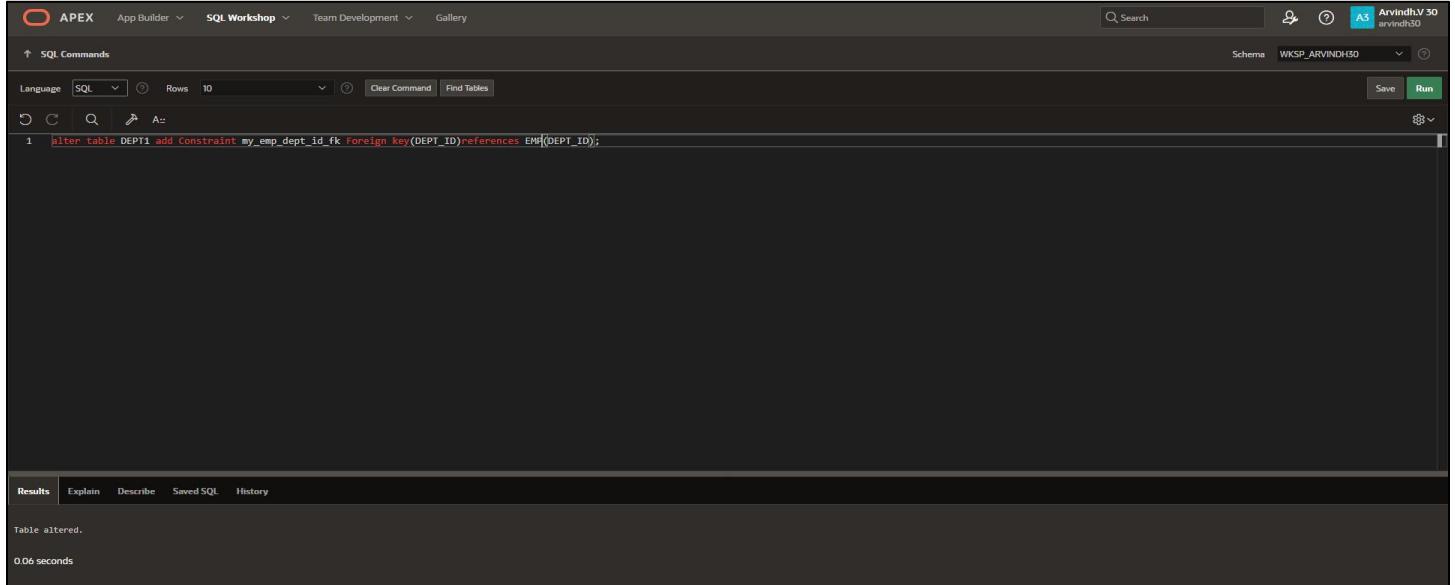
The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'alter table DEPT1 add constraint my_dept_id_pk Primary key(ID);'. Below the code, the 'Results' tab is active, showing the output: 'Table altered.' and '0.08 seconds'. The schema is set to 'WKSP_ARVINDH30'.

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 DEPT1 add Constraint my_emp_dept_id_fk Foreign key(DEPT_ID) references DEPT2(DEPT_ID);

OUTPUT:



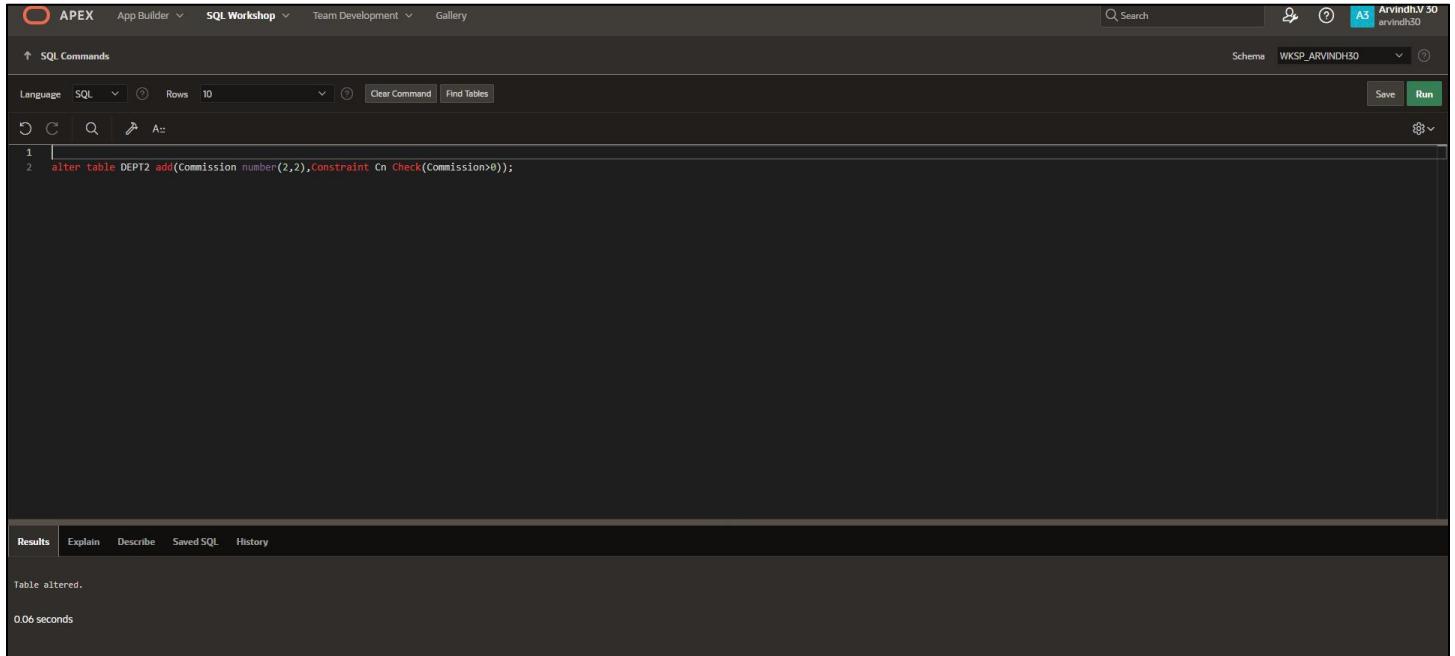
The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'alter table DEPT1 add Constraint my_emp_dept_id_fk Foreign key(DEPT_ID) references DEPT2(DEPT_ID);'. Below the code, the 'Results' tab is active, showing the output: 'Table altered.' and '0.06 seconds'. The schema is set to 'WKSP_ARVINDH30'.

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 DEPT2 add (Commission number(2,2),Constraint Cn Check(Commission>0));

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. Two lines of SQL code are entered: '1 alter table DEPT2 add(Commission number(2,2),Constraint Cn Check(Commission>0));'. Below the code, the 'Results' tab is active, showing the output: 'Table altered.' and '0.06 seconds'. The schema is set to 'WKSP_ARVINDH30'.

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

RESULT:

Writing Basic SQL SELECT Statements

EX_NO:4

DATE:03/03/2024

- 1.The following statement executes successfully.

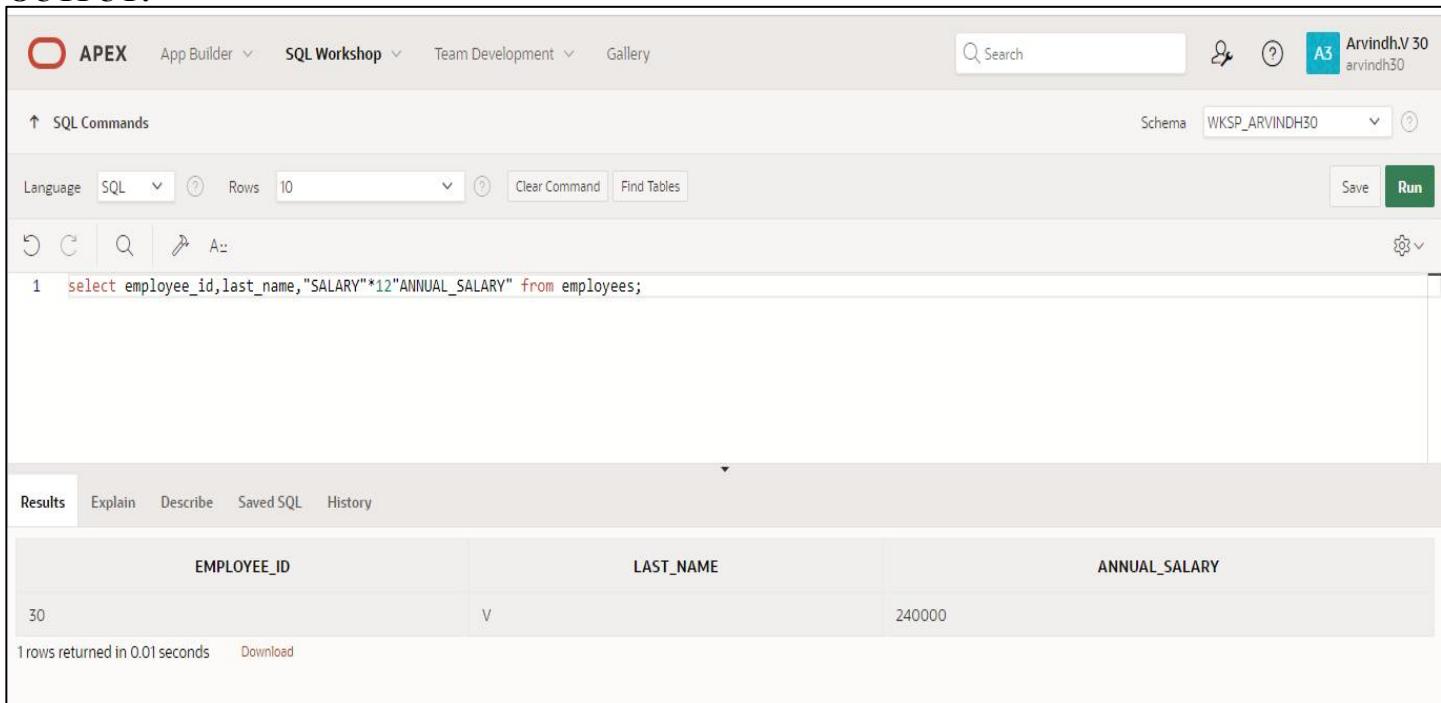
Identify the Errors

```
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. 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 'Arvindh.V 30 arvindh30', and a 'Run' button. The main workspace is titled 'SQL Commands' with a schema dropdown set to 'WKSP_ARVINDH30'. The SQL editor contains the following code:

```
1 select employee_id, last_name, "SALARY"*12"ANNUAL_SALARY" from employees;
```

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

EMPLOYEE_ID	LAST_NAME	ANNUAL_SALARY
30	V	240000

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

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

QUERY:

select * from employees;

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands pane, the query `select * from employees;` is entered. The Results pane displays the output in a grid format. The columns are labeled: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, MANAGER_ID, DEPARTMENT_ID, and ANNUAL_SALARY. There is one row returned, corresponding to employee ID 30, whose first name is Ms and last name is Dhoni. The salary is 20000, manager ID is 8, department ID is 7, and annual salary is null.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID	ANNUAL_SALARY
30	Ms	Dhoni	30@gmail.com	9820930434	09/07/2004	07	20000	8	7	-

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 as employee_number, last_name, job_id, hire_date from employees;

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the SQL Commands pane, the query `Select employee_id as employee_number, last_name, job_id, hire_date from employees;` is entered. The Results pane displays the output in a grid format. The columns are labeled: EMPLOYEE_NUMBER, LAST_NAME, JOB_ID, and HIRE_DATE. There is one row returned, corresponding to employee ID 30, whose last name is Dhoni, job ID is 07, and hire date is 09/07/2004.

EMPLOYEE_NUMBER	LAST_NAME	JOB_ID	HIRE_DATE
30	Dhoni	07	09/07/2004

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 search bar, user profile, and schema information (WKSP_ARVINDH30). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the query: "Select hire_date as startdate from employees;". The Results tab displays the output: "STARTDATE" followed by the value "09/07/2004". Below the results, it says "1 rows returned in 0.00 seconds" and has a "Download" link.

```
1 Select hire_date as startdate from employees;
2
```

STARTDATE
09/07/2004

1 rows returned in 0.00 seconds Download

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 search bar, user profile, and schema information (WKSP_ARVINDH30). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the query: "Select distinct job_id from employees;". The Results tab displays the output: "JOB_ID" followed by the value "07". Below the results, it says "1 rows returned in 0.00 seconds" and has a "Download" link.

```
1 Select distinct job_id from employees;
```

JOB_ID
07

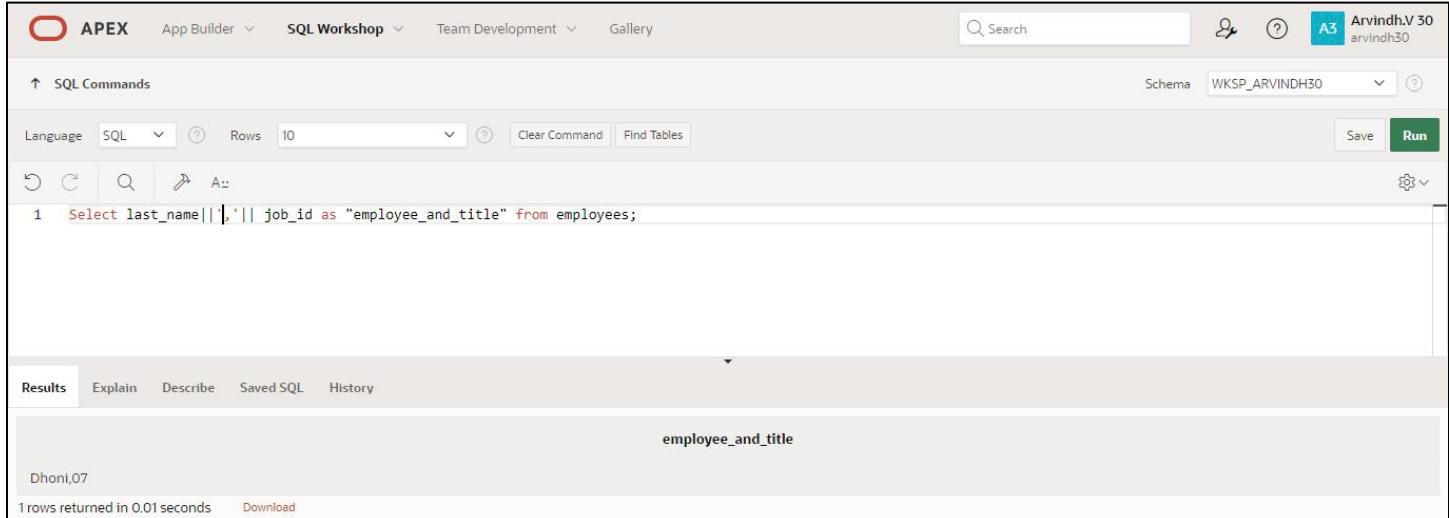
1 rows returned in 0.00 seconds Download

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:



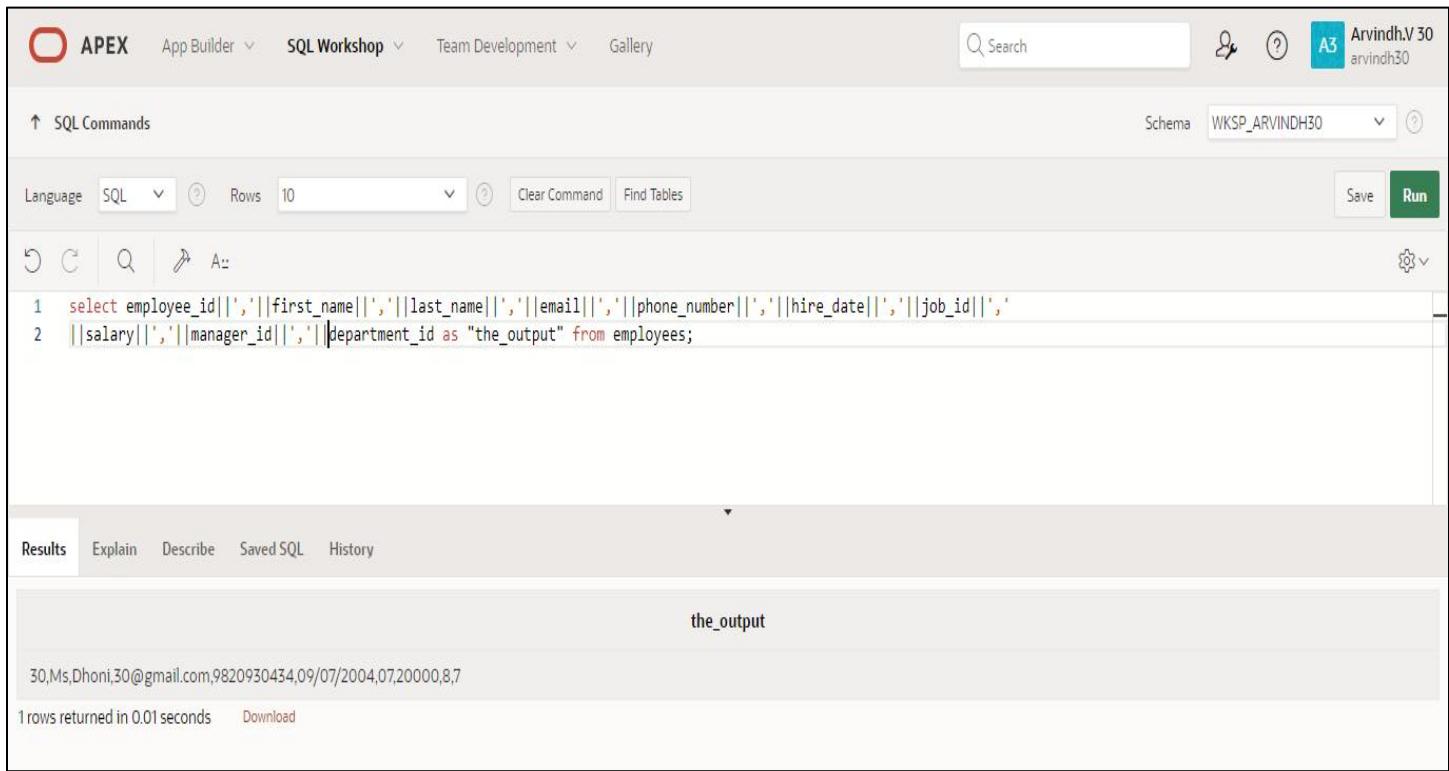
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arvindh.V 30 arvindh30', and a schema dropdown set to 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a language dropdown set to 'SQL' and a row limit of '10'. Below the command input field, there are icons for refresh, search, and run. The command entered is: 'Select last_name||','|| job_id as "employee_and_title" from employees;'. The results tab is selected, showing a single row with the value 'Dhoni,07' under the column 'employee_and_title'. At the bottom, it says '1 rows returned in 0.01 seconds' and has a 'Download' link.

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||','||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 'Arvindh.V 30 arvindh30', and a schema dropdown set to 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a language dropdown set to 'SQL' and a row limit of '10'. Below the command input field, there are icons for refresh, search, and run. The command entered is: '1 select employee_id||','||first_name||','||last_name||','||email||','||phone_number||','||hire_date||','||job_id||','||salary||','||manager_id||','||department_id as "the_output" from employees;'. The results tab is selected, showing a single row with the value '30,Ms,Dhoni,30@gmail.com,9820930434,09/07/2004,07,20000,8.7' under the column 'the_output'. At the bottom, it says '1 rows returned in 0.01 seconds' and has a 'Download' link.

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

RESULT:

RESTRICTING AND SORTING DATA

EX_NO:5

DATE:05/03/2024

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 APEX SQL Workshop interface. The SQL command entered is:

```
1 Select last_name,salary
2 From EMPLOYEES
3 Where salary>12000;
```

The results table displays the following data:

LAST_NAME	SALARY
Dhoni	20000
kohli	14000
raina	15000

3 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 APEX SQL Workshop interface. The SQL command entered is:

```
1 Select last_name,department_id
2 From EMPLOYEES
3 Where employee_id=176;
```

The results table displays the following data:

LAST_NAME	DEPARTMENT_ID
smith	97

1 rows returned in 0.01 seconds

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

QUERY:

```
select last_name,salary from employees where salary not between 5000 and 12000;
```

OUTPUT:

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

```
1 Select last_name,salary
2 From EMPLOYEES
3 | where salary not between 5000 and 12000;
```

The results section displays the following data:

LAST_NAME	SALARY
Dhoni	20000
kohli	14000
raina	15000

3 rows returned in 0.02 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 '02/20/1998' and '05/01/1998';
```

OUTPUT:

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

```
1 Select last_name,job_id,hire_date
2 From EMPLOYEES
3 | where hire_date between '02/20/1998' and '05/01/1998';
```

The results section displays the following data:

LAST_NAME	JOB_ID	HIRE_DATE
gyle	244	05/21/1998

1 rows returned in 0.01 seconds

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 SQL command entered is:

```
1 select last_name,department_id from employees where department_id in(20,50) order by last_name;
```

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

LAST_NAME	DEPARTMENT_ID
gayle	50

1 rows returned in 0.03 seconds

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",salary as "MONTHLY SALARY" from employees where (salary between 5000 and 12000) and (department_id in(20,50)) order by last_name asc;
```

OUTPUT:

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

```
1 select last_name as "EMPLOYEE",salary as "MONTHLY SALARY" from employees where (salary between 5000  
2 and 12000) and (department_id in(20,50)) order by last_name asc;
```

The results table has two columns: EMPLOYEE and MONTHLY SALARY. The single row returned is:

EMPLOYEE	MONTHLY SALARY
gayle	12000

1 rows returned in 0.04 seconds

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

QUERY:

```
select last_name,hire_date from employees where hire_date like '%1994';
```

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 Arvindh.V 30 (arvindh30). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL tab is selected, showing the query: `select last_name,hire_date from employees where hire_date like '%1994';`. The Results tab displays the output:

LAST_NAME	HIRE_DATE
smith	01/01/1994
raina	02/05/1994

Below the results, it says "2 rows returned in 0.00 seconds". There is also a "Download" link and a note about activating Windows.

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 top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for Arvindh.V 30 (arvindh30). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL tab is selected, showing the query: `select last_name,job_id from employees where manager_id is null;`. The Results tab displays the output:

LAST_NAME	JOB_ID
duplessis	6

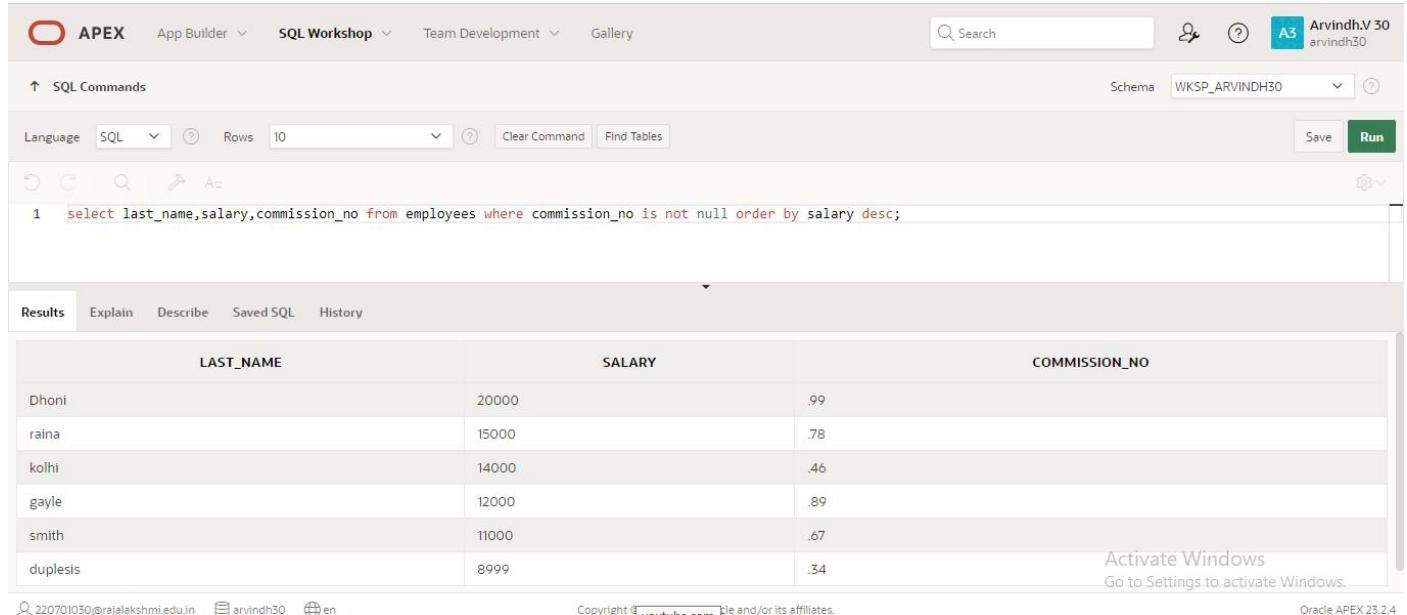
Below the results, it says "1 rows returned in 0.01 seconds". There is also a "Download" link and a note about activating Windows.

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_no from employees where commission_no is not null order by salary desc;
```

OUTPUT:



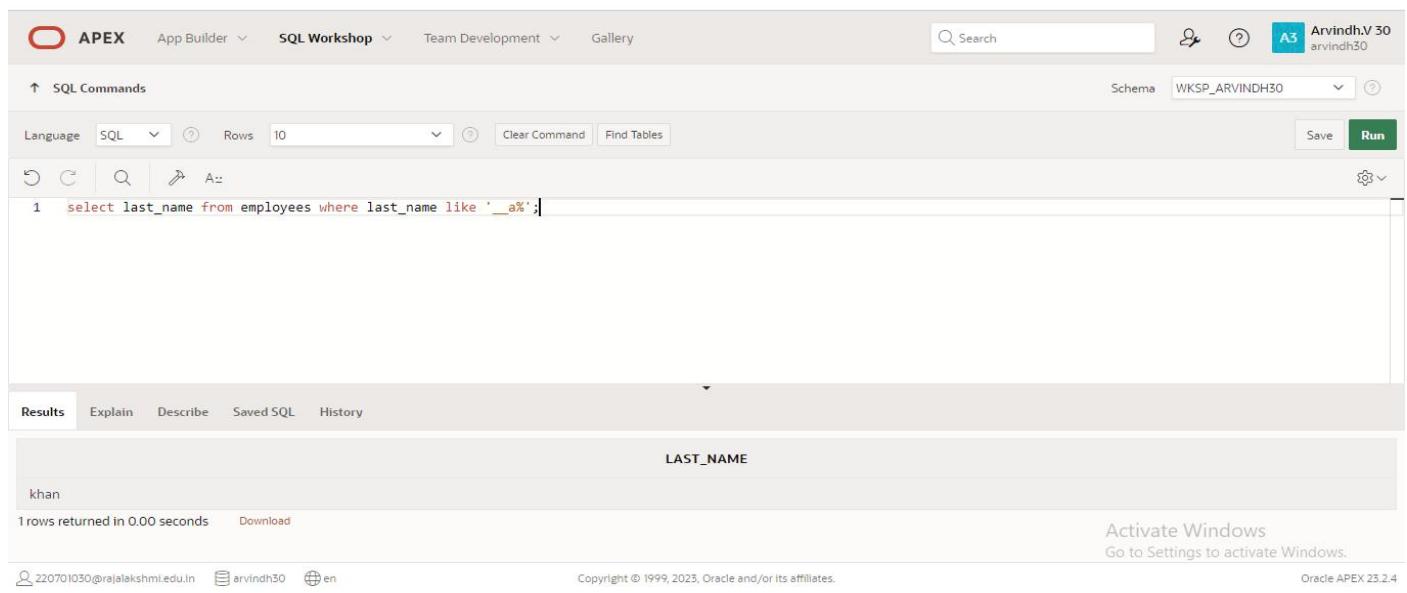
LAST_NAME	SALARY	COMMISSION_NO
Dhoni	20000	.99
raina	15000	.78
kolhi	14000	.46
gayle	12000	.89
smith	11000	.67
duplesis	8999	.34

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:



LAST_NAME
khan

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 SQL command entered is:

```
1 select last_name from employees where last_name like '%a%' and last_name like '%e%';
```

The results section displays a single row:

LAST_NAME
gayle

1 rows returned in 0.01 seconds

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

QUERY:

```
select last_name,job_id,salary from employees where job_id in ('SA_REP','ST_CL') and salary not in(2500,3500,7000);
```

OUTPUT:

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

```
1 select last_name,job_id,salary from employees where job_id in ('SA_REP','ST_CL') and
2 salary not in(2500,3500,7000);
```

The results section displays a single row:

LAST_NAME	JOB_ID	SALARY
jadeja	SA_REP	31004

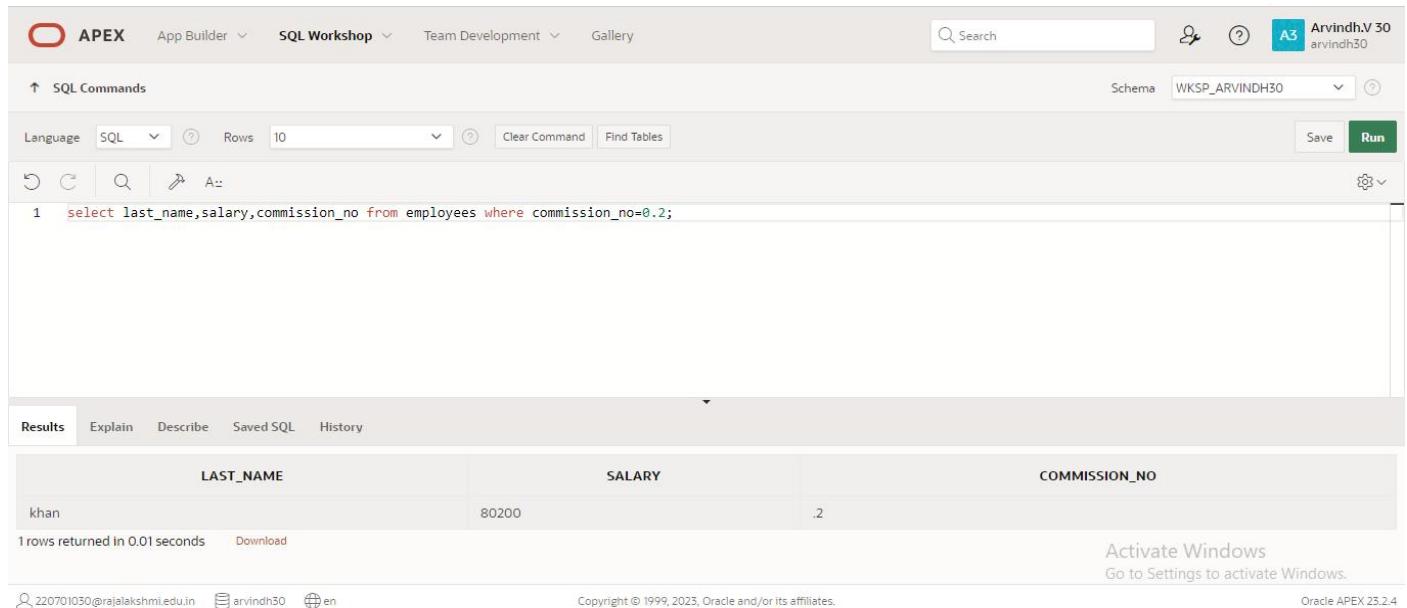
1 rows returned in 0.00 seconds

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

QUERY:

```
select last_name,salary,commission_no from employees where commission_no=0.2;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows the user 'Arvindh.V 30' and schema 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a search bar and filter options for rows (set to 10). The SQL editor contains the query: 'select last_name,salary,commission_no from employees where commission_no=0.2;'. Below the editor, the results tab is selected, showing a single row: khan, 80200, .2. The results table has columns 'LAST_NAME', 'SALARY', and 'COMMISSION_NO'. At the bottom, it says '1 rows returned in 0.01 seconds' and provides download and explain options.

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:

```
char(sysdate,'mm-dd-yy') as "date" from dual;
```

OUTPUT:

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

```
1 select to_char(sysdate,'mm-dd-yy') as "date"
2 from dual;
```

The results show a single row with the date 03-12-24.

date
03-12-24

Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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 NewSalary.

Query :

```
select employee_id,last_name,salary,salary+(15.5/100*salary) "new_salary"from employees;
```

OUTPUT:

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

```
1 select employee_id,last_name,salary,round(salary+salary * 0.155) as "new salary"
2 from employees;
```

The results show the employee ID, last name, salary, and the new salary after a 15.5% increase.

EMPLOYEE_ID	LAST_NAME	SALARY	new salary
30	Dhoni	20000	23100
35	Duplessis	8999	10394
48	Kolhi	14000	16170
776	Smith	11000	12705
333	Gayle	12000	13860
43	Khan	80200	92631
5	Raina	15000	17325
899	Jadeja	31004	35810

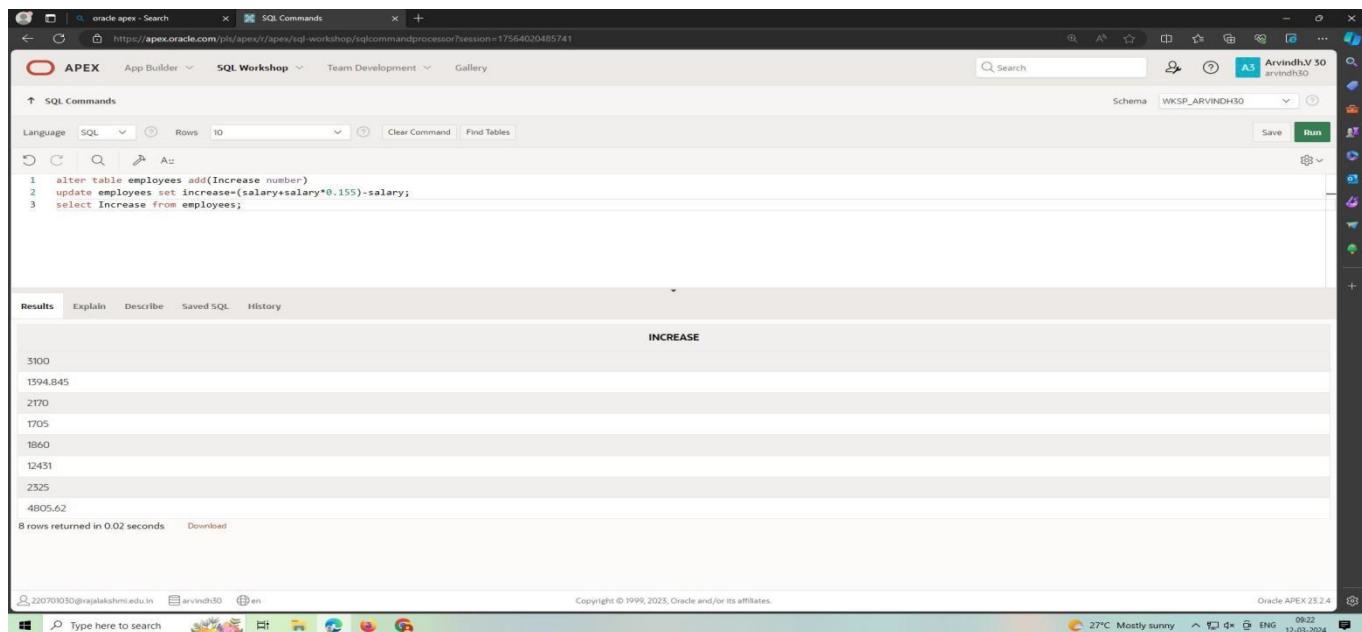
8 rows returned in 0.01 seconds Download

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

Query:

```
select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary", new_salary-salary as "Increase" from employees;
```

Output:



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

```
1 alter table employees add(Increase number)
2 update employees set increase=(salary+salary*0.155)-salary;
3 select Increase from employees;
```

The Results tab shows the output of the query:

INCREASE
3100
1594.845
2170
1705
1860
12431
2325
4805.62

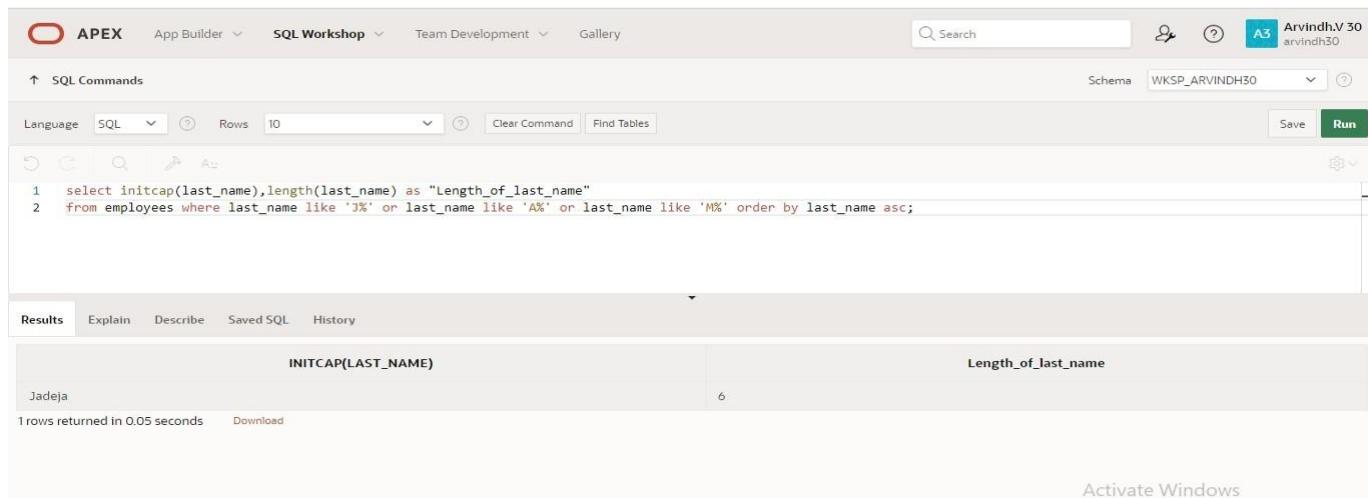
8 rows returned in 0.02 seconds

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), length(last_name) as "Length_of_last_name" from employees where last_name like 'J%' or last_name like 'A%' or last_name like 'M%' order by last_name asc;
```

Output:



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

```
1 select initcap(last_name), length(last_name) as "Length_of_last_name"
2 from employees where last_name like 'J%' or last_name like 'A%' or last_name like 'M%' order by last_name asc;
```

The Results tab shows the output of the query:

INITCAP(LAST_NAME)	Length_of_last_name
Jadeja	6

1 rows returned in 0.05 seconds

Activate Windows
Go to Settings to activate Windows.

Oracle APEX 23.2.4



Edit with WPS Office

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),length(last_name) as "Length_of_last_name" from employees where last_name like 'H%' order by last_name;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a session profile for 'Arvindh.V 30' (arvindh30). The main area has tabs for 'SQL Commands' and 'Results'. Under 'Results', the query is displayed:

```
1 select initcap(last_name)"NAME",length(last_name) as "Length_of_last_name" from employees where last_name like 'H%' order by last_name;
```

The results table has two columns: 'NAME' and 'Length_of_last_name'. One row is returned:

NAME	Length_of_last_name
Hardik	6

Below the table, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. The bottom of the page includes standard links like 'Activate Windows', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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((sysdate-hire_date)/30,0) as "MONTHS_WORKED" from employees order by round((sysdate-hire_date)/30,0) asc;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a session profile for 'Arvindh.V 30' (arvindh30). The main area has tabs for 'SQL Commands' and 'Results'. Under 'Results', the query is displayed:

```
1 select last_name,round((sysdate-hire_date)/30,0) as "MONTHS_WORKED" from employees order by round((sysdate-hire_date)/30,0) asc;
```

The results table has two columns: 'LAST_NAME' and 'MONTHS_WORKED'. Multiple rows are returned:

LAST_NAME	MONTHS_WORKED
kohli	199
Jadeja	222
Dhoni	239
Guptil	253
Hardik	277
duplesis	287
gayle	317

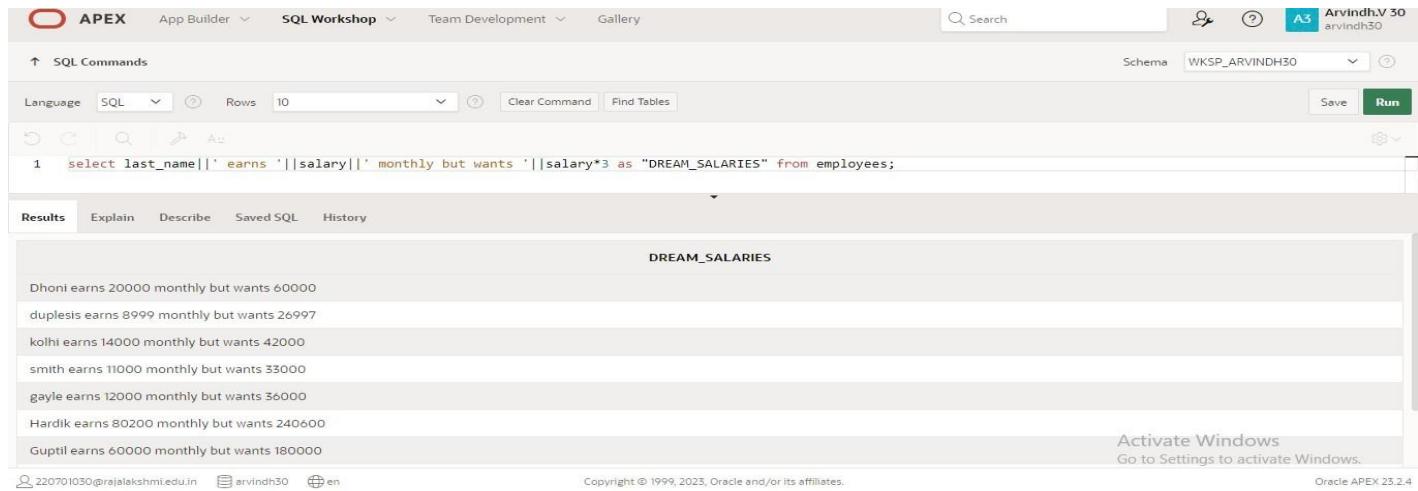
Below the table, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. The bottom of the page includes standard links like 'Activate Windows', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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

QUERY:

```
select last_name||' earns'||salary||' monthly but wants'||salary*3 as "DREAM_SALARIES" from employees;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The query is executed and the results are displayed in a table titled 'DREAM_SALARIES'. The results show the employee's last name followed by their current salary and three times their salary.

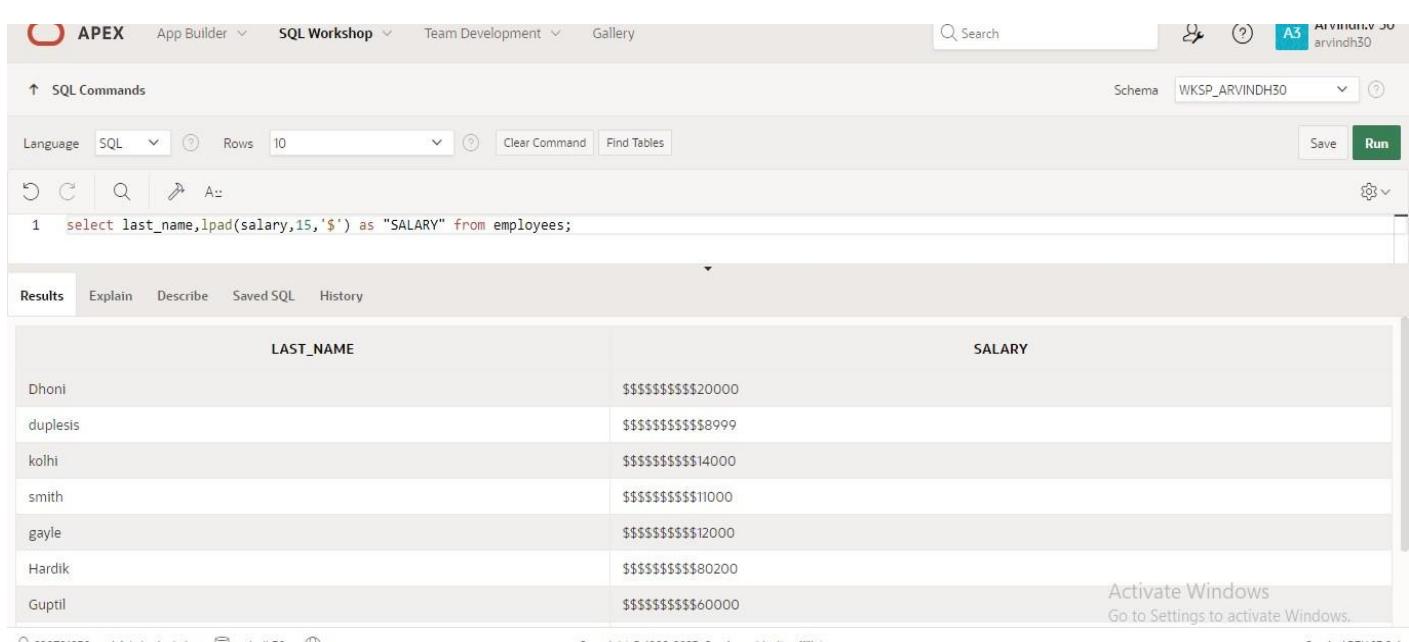
DREAM_SALARIES	
Dhoni	earns 20000 monthly but wants 60000
duplesis	earns 8999 monthly but wants 26997
kolhi	earns 14000 monthly but wants 42000
smith	earns 11000 monthly but wants 33000
gayle	earns 12000 monthly but wants 36000
Hardik	earns 80200 monthly but wants 240600
Guptil	earns 60000 monthly but wants 180000

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,'$') as "SALARY" from employees;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The query is executed and the results are displayed in a table with columns 'LAST_NAME' and 'SALARY'. The salary values are left-padded with '\$' to have a total width of 15 characters.

LAST_NAME	SALARY
Dhoni	\$\$\$\$\$\$\$\$\$\$20000
duplesis	\$\$\$\$\$\$\$\$\$\$8999
kolhi	\$\$\$\$\$\$\$\$\$\$14000
smith	\$\$\$\$\$\$\$\$\$\$11000
gayle	\$\$\$\$\$\$\$\$\$\$12000
Hardik	\$\$\$\$\$\$\$\$\$\$80200
Guptil	\$\$\$\$\$\$\$\$\$\$60000

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(ADD_MONTHS(hire_date, 6),  
'MONDAY'),'FMDay, "the "FMDD "of "FMMonth, YYYY') AS REVIEW FROM employees;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there is a user profile for 'ArvindhV30' and a search bar. Below the tabs, there are buttons for Language (SQL selected), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The main area contains the SQL command:

```
1 SELECT last_name,hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY'),'FMDay, "the "FMDD "of "FMMonth, YYYY') AS REVIEW FROM employees;
```

Below the command, there are tabs for Results (selected), Explain, Describe, Saved SQL, and History. The results section displays a table with three columns: LAST_NAME, HIRE_DATE, and REVIEW. The data is as follows:

LAST_NAME	HIRE_DATE	REVIEW
Dhoni	09/07/2004	Monday, the 14 of March, 2005
duplesis	09/08/2000	Monday, the 12 of March, 2001
kolhi	12/07/2007	Monday, the 09 of June, 2008
smith	01/01/1994	Monday, the 04 of July, 1994
gayle	03/21/1998	Monday, the 28 of September, 1998
Hardik	07/07/2001	Monday, the 14 of January, 2002
Guptil	07/03/2003	Monday, the 05 of January, 2004

At the bottom left, there are links for email (220701030@rajalakshmi.edu.in) and user (arvindh30). On the right, there is a watermark for 'Edit with WPS Office' and copyright information: Copyright © 1999-2025 Oracle and/or its affiliates.

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') as Day from employees order by TO_CHAR(hire_date,'Day');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop, Team Development, and Gallery. The right side shows a user profile for 'Arvindh.V 30' (arvindh30). The main workspace has tabs for SQL Commands, SQL, and Run. The SQL tab contains the query: 'SELECT last_name,hire_date,TO_CHAR(hire_date,'Day') as Day from employees order by TO_CHAR(hire_date,'Day');'. The Run button is highlighted in green. Below the query, the Results tab is selected, displaying the output in a grid format:

LAST_NAME	HIRE_DATE	DAY
duplesis	09/08/2000	Friday
kolhi	12/07/2007	Friday
Jadeja	01/09/2006	Monday
smith	01/01/1994	Saturday
gayle	03/21/1998	Saturday
raina	02/05/1994	Saturday
Hardik	07/07/2001	Saturday

At the bottom of the results grid, there is a watermark: 'Activate Windows Go to Settings to activate Windows.' The footer of the page includes the URL '220701030@rajalakshmi.edu.in', the user name 'arvindh30', and the language 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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.deptid from employees e,MYDEPT d where  
e.department_id=d.deptid;
```

OUTPUT:

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

```
1 Select e.last_name,e.department_id,d.deptid from employees e,MYDEPT d where  
2 e.department_id=d.deptid;
```

The results section displays the following data:

LAST_NAME	DEPARTMENT_ID	DEPTID
duplesis	80	80
kolhi	80	80
gayle	79	79
Hardik	78	78
raina	78	78

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,mydept d where e.department_id=d.deptid and  
e.department_id=80;
```

OUTPUT:

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

```
1 select distinct job_id,location_id from employees e,mydept d where e.department_id=d.deptid and e.department_id=80;
```

The results section displays the following data:

JOB_ID	LOCATION_ID
1700	5
1	5

2 rows returned in 0.02 seconds Download

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.dept_name,d.loc_id,l.city from employees e,mydept d,location l  
where e.department_id=d.deptid and d.loc_id=l.locationid and e.commission is not null;
```

OUTPUT:

The screenshot shows the Oracle APEX 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 'ArvindhV 30' and a schema dropdown set to 'WKSP_ARVINDH30'. The main area has tabs for 'SQL Commands', 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'SQL Commands' tab contains the following SQL code:

```
1 Select e.last_name,e.department_id,d.dept_name,d.loc_id,l.city from employees e,mydept d,location l  
2 where e.department_id=d.deptid and d.loc_id=l.locationid and e.commission is not null;
```

The 'Results' tab displays the query output as a table:

LAST_NAME	DEPARTMENT_ID	DEPT_NAME	LOC_ID	CITY
gayle	79	CSBS	3	toronto
duplesis	80	CSE	1	Chennai
kohli	80	CSE	1	Chennai
duplesis	80	CSE	1	Chennai
kohli	80	CSE	1	Chennai
gayle	79	CSBS	3	Chennai

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

QUERY:

```
Select employees.last_name,mydept.dept_name from employees,mydept where  
employees.department_id=mydept.deptid and last_name like '%a%';
```

OUTPUT:

The screenshot shows the Oracle APEX 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 'ArvindhV 30' and a schema dropdown set to 'WKSP_ARVINDH30'. The main area has tabs for 'SQL Commands', 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'SQL Commands' tab contains the following SQL code:

```
1 Select employees.last_name,mydept.dept_name from employees,mydept where  
2 employees.department_id=mydept.deptid and last_name like '%a%';
```

The 'Results' tab displays the query output as a table:

LAST_NAME	DEPT_NAME
gayle	CSBS
Hardik	Executive
raina	Executive

At the bottom of the page, there are footer links for 'Activate Windows', 'Go to Settings to activate Windows.', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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

QUERY: Select e.last_name,e.department_id,e.job_id,d.dept_name from employees e join mydept d on(e.department_id=d.deptid) join location on (d.loc_id=location.locationid) where lower(location.city)='toronto';

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'ArvindhV 30' and session ID 'arvindh30'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL tab contains the following query:

```
1 Select e.last_name,e.department_id,e.job_id,d.dept_name from employees e join mydept d
2 on(e.department_id=d.deptid) join location on (d.loc_id=location.locationid) where
3 lower(location.city)='toronto';
```

The Results tab displays the output:

LAST_NAME	DEPARTMENT_ID	JOB_ID	DEPT_NAME
gayle	79	3	CSBS

1 rows returned in 0.06 seconds. The bottom status bar shows the URL '220701030@rajalakshmi.edu.in', session 'arvindh30', and locale 'en'. It also includes copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and software version 'Oracle APEX 23.2.4'. An 'Activate Windows' watermark is visible.

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 last_name "Employee",employee_id "emp#",manager_name "manager",manager_id "Mgr#"  
from employees
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'ArvindhV 30' and session ID 'arvindh30'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL tab contains the following query:

```
1 Select last_name "Employee",employee_id "emp#",manager_name "manager",manager_id "Mgr#"  
2 from employees
```

The Results tab displays the output:

Employee	emp#	manager	Mgr#
Dhoni	30	King	8
duplesis	35	PARDIP	-
kolhi	18	VISHWA	6
smith	176	-	89
gayle	333	-	77
Hardik	43	-	

The bottom status bar shows the URL '220701030@rajalakshmi.edu.in', session 'arvindh30', and locale 'en'. It also includes copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and software version 'Oracle APEX 23.2.4'. An 'Activate Windows' watermark is visible.

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

QUERY: select e.last_name,e.manager_name,e.department_name,d.loc_id,d.loc,e.salary from employee e join mydept d on e.employee_id=d.empid order by e.employee_id

OUTPUT:

The screenshot shows the Oracle APEX 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 'Arvindh.v 30 arvindh30'. The main area has tabs for 'SQL Commands', 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL command entered is:

```
1 select e.last_name,e.manager_name,e.dept_name,d.loc_id,d.loc,e.salary from employees e join mydept d on e.employee_id=d.empid order by e.employee_id
```

The 'Results' tab displays the output:

LAST_NAME	MANAGER_NAME	DEPT_NAME	LOC_ID	LOC	SALARY
smith	king	eee	3	Bangaluru	11000

Below the table, it says '1 rows returned in 0.01 seconds' and there is a 'Download' link. The bottom of the page includes standard links like 'Activate Windows', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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.department_id "Dept",e.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,c.last_name
```

OUTPUT:

The screenshot shows the Oracle APEX 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 'Arvindh.v 30 arvindh30'. The main area has tabs for 'SQL Commands', 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL command entered is:

```
1 select e.department_id "Dept",e.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,c.last_name
```

The 'Results' tab displays the output:

Dept	colleague
78	Hardik
78	raina
80	duplesis
80	kolhi

Below the table, it says '4 rows returned in 0.01 seconds' and there is a 'Download' link. The bottom of the page includes standard links like 'Activate Windows', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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
JOIN mydept d ON (e.department_id = d.deptid) JOIN job_grades j ON (e.salary BETWEEN
j.lowest_sal AND j.highest_sal);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command is displayed in the editor:

```
1 SELECT e.last_name, e.job_id, d.dept_name, e.salary, j.grade_level FROM employees e
2 JOIN mydept d ON (e.department_id = d.deptid) JOIN job_grades j ON (e.salary BETWEEN
3 j.lowest_sal AND j.highest_sal);
```

The results pane shows the following data:

LAST_NAME	JOB_ID	DEPT_NAME	SALARY	GRADE_LEVEL
duplesis	1	CSE	8999	5th
kolhi	1700	CSE	14000	4th
raina	43	Executive	15000	4th
Hardik	9	Executive	80200	1st
gayle	3	CSBS	12000	4th

5 rows returned in 0.01 seconds

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, employees davies WHERE davies.last_name
='Davies' AND davies.hire_date < e.hire_date;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command is displayed in the editor:

```
1 SELECT e.last_name, e.hire_date FROM employees e, employees davies WHERE davies.last_name
='Davies' AND davies.hire_date < e.hire_date;
```

The results pane shows the following data:

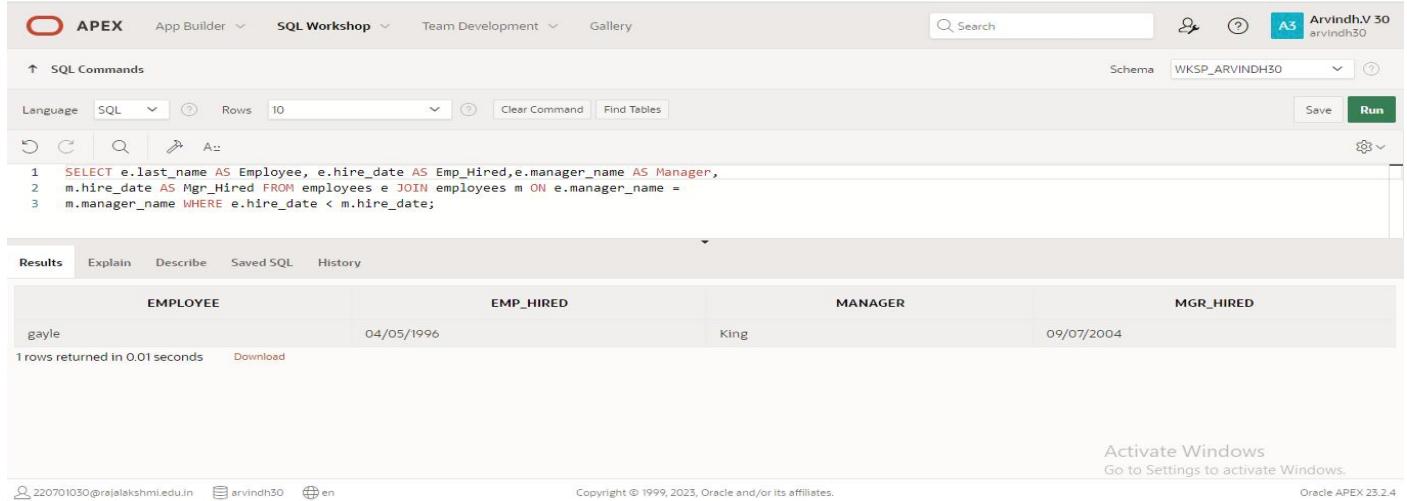
LAST_NAME	HIRE_DATE
Dhoni	09/07/2004
duplesis	09/08/2000
kolhi	12/07/2007
gayle	03/21/1998
Hardik	07/07/2001
Guptil	07/03/2003

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 e.last_name AS Employee, e.hire_date AS Emp_Hired,e.manager_name AS Manager,
m.hire_date AS Mgr_Hired FROM employees e JOIN employee m ON e.manager_name =
m.manager_name WHERE e.hire_date < m.hire_date;
```

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 is a user profile for 'Arvindh.v 30' with the ID 'arvindh30'. The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to 'WKSP_ARVINDH30'. Below the search bar are buttons for 'Save' and 'Run'. The SQL command entered is:

```
1 SELECT e.last_name AS Employee, e.hire_date AS Emp_Hired,e.manager_name AS Manager,
2 m.hire_date AS Mgr_Hired FROM employees e JOIN employee m ON e.manager_name =
3 m.manager_name WHERE e.hire_date < m.hire_date;
```

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

EMPLOYEE	EMP_HIRED	MANAGER	MGR_HIRED
gayle	04/05/1996	King	09/07/2004

Below the table, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes copyright information for Oracle and a link to activate Windows.

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

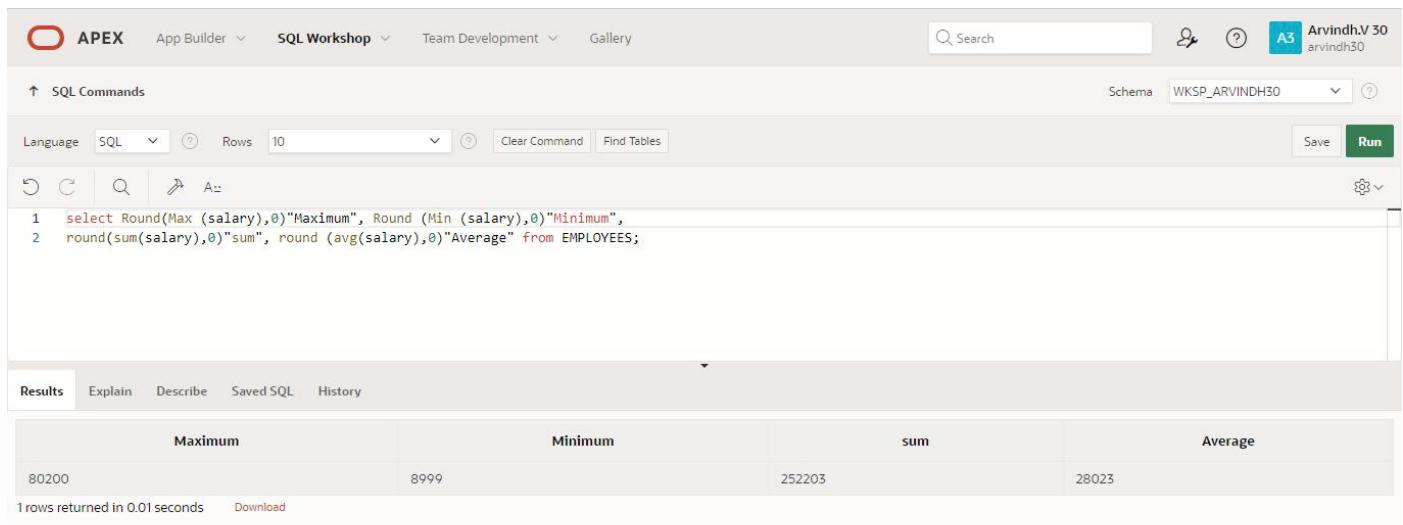
FALSE

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)"Maximum", Round (Min (salary),0)"Minimum",
round(sum(salary),0)"sum", round (avg(salary),0)"Average" 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. The user is signed in as Arvindh.v 30 (arvindh30). The SQL Commands tab is selected, showing the following SQL code:

```
1 select Round(Max (salary),0)"Maximum", Round (Min (salary),0)"Minimum",
2 round(sum(salary),0)"sum", round (avg(salary),0)"Average" from EMPLOYEES;
```

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

	Maximum	Minimum	sum	Average
	80200	8999	252203	28023

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

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

QUERY:

```
select job_id ,Round(MAX(salary),0) "MAXIMUM",Round (Min(salary),0)"Minimum",Round  
(SUM(Salary),0)"sum" ,Round (AVG (salary),0)"average" from EMPLOYEES group by job_id;
```

OUTPUT:

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

```
1 select job_id ,Round(MAX(salary),0) "MAXIMUM",Round (Min(salary),0)"Minimum",Round  
2 (SUM(Salary),0)"sum" ,Round (AVG (salary),0)"average" from EMPLOYEES group by job_id;
```

The results are displayed in a table:

JOB_ID	MAXIMUM	MINIMUM	sum	average
244	12000	12000	12000	12000
63	60000	60000	60000	60000
99	11000	11000	11000	11000
8	14000	14000	14000	14000
6	8999	8999	8999	8999
07	20000	20000	20000	20000

An 'Activate Windows' watermark is visible across the results.

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 where job_id='47'group by job_id
```

OUTPUT:

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

```
1 select job_id,count(*) from employees where job_id='47'group by job_id
```

The results are displayed in a table:

JOB_ID	COUNT(*)
47	2

1 rows returned in 0.01 seconds

An 'Activate Windows' watermark is visible across the results.

Activate Windows
Go to Settings to activate Windows.

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 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon for 'Arvindh.V 30 arvindh30', and a 'Schema' dropdown set to 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. A 'Rows' dropdown is set to '10'. Below the input field, the command is written: '1 select count(distinct manager_id)"Number of managers" from employees;'. The results tab is selected, showing a single row with the value '7' under the heading 'Number of managers'. Below the results, it says '1 rows returned in 0.00 seconds' and has a 'Download' link.

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

QUERY:select max(salary)-min(salary) difference 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 'Arvindh.V 30 arvindh30', and a 'Schema' dropdown set to 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. A 'Rows' dropdown is set to '10'. Below the input field, the command is written: '1 select max(salary)-min(salary) difference from employees;'. The results tab is selected, showing a single row with the value '71201' under the heading 'DIFFERENCE'. Below the results, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. A watermark at the bottom right reads 'Activate Windows Go to Settings to activate Windows'.

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 APEX SQL Workshop interface. The SQL command entered is:

```
1 select manager_id ,MIN(salary) from employees where manager_id is not null group by
2 manager_id having min(salary)>6000 order by min(salary) desc;
```

The results table has two columns: MANAGER_ID and MIN(SALARY). The data is:

MANAGER_ID	MIN(SALARY)
98	60000
24	31004
8	20000
2	15000
6	14000
77	12000

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(*)total,sum(decode(to_char(hire_date,'YYYY'),1995,1,0))"1995",sum(decode(to_char
(hire_date,'YYYY'),1996,1,0))"1996",sum(decode(to_char(hire_date,'YYYY'),1997,1,0))"1997",sum(
decode(to_char(hire_date,'YYYY'),1998,1,0)) "1998" from employees;
```

OUTPUT:

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

```
1 Select count(*)total,sum(decode(to_char(hire_date,'YYYY'),1995,1,0))"1995",sum(decode(to_char
2 (hire_date,'YYYY'),1996,1,0))"1996",sum(decode(to_char(hire_date,'YYYY'),1997,1,0))"1997",sum(
3 decode(to_char(hire_date,'YYYY'),1998,1,0)) "1998" from employees;
```

The results table has five columns: TOTAL, 1995, 1996, 1997, and 1998. The data is:

TOTAL	1995	1996	1997	1998
9	0	0	0	1

1 rows returned in 0.01 seconds

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

QUERY:

```
select job_id "job", sum(decode(department_id,20,salary))"Dept20",sum (decode(department_id ,50, salary)) "dept50",sum (decode(department_id ,80, salary)) "dept80",sum (decode(department_id ,90,salary)) "dept90",sum(salary) "TOTAL" from employees group by job_id
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The SQL command is displayed in the editor:

```
1 select job_id "job", sum(decode(department_id,20,salary))"Dept20",sum (decode(department_id ,50, salary)) "dept50",sum (decode(department_id ,80, salary)) "dept80",sum (decode(department_id ,90,salary)) "dept90",sum(salary) "TOTAL" from employees group by job_id
```

The results are presented in a grid:

job	Dept20	dept50	dept80	dept90	TOTAL
244	-	12000	-	-	12000
63	-	-	-	-	60000
99	-	-	-	-	11000
8	-	-	-	-	14000
6	-	-	8999	-	8999
07	-	-	-	-	20000

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 as "dept_name",d.loc as "department location", count(*) "Number of people",round(avg(salary),2) "salary" from mydept d inner join employees e on(d.deptid =e.department_id ) group by d.dept_name ,d.loc;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The SQL command is displayed in the editor:

```
1 select d.dept_name as "dept_name",d.loc as "department location", count(*) "Number of people",round(avg(salary),2) "salary" from mydept d inner join employees e on(d.deptid =e.department_id ) group by d.dept_name ,d.loc;
```

The results are presented in a grid:

dept_name	department location	Number of people	salary
CSE	Chennai	1	8999

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

RESULT:

SUB QUERIES

EX_NO:9

DATE:11/04/2024

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,hire_date from employees where department_id=(select department_id from employees where last_name='raina') and last_name not in('raina');
```

OUTPUT:

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

```
1 select last_name,hire_date from employees where department_id=(select department_id from employees where last_name='raina') and last_name not in('raina');
```

The results window displays the output:

LAST_NAME	HIRE_DATE
Dhoni	09/07/2004
Guptil	07/03/2003

2 rows returned in 0.01 seconds

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

QUERY: select employee_id,last_name,salary from employees where salary>(select avg(salary) from employees) order by salary;

OUTPUT:

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

```
1 select employee_id,last_name,salary from employees where salary>(select avg(salary) from employees) order by salary;
```

The results window displays the output:

EMPLOYEE_ID	LAST_NAME	SALARY
899	Jadeja	31004
309	Guptil	60000
43	khan	80200

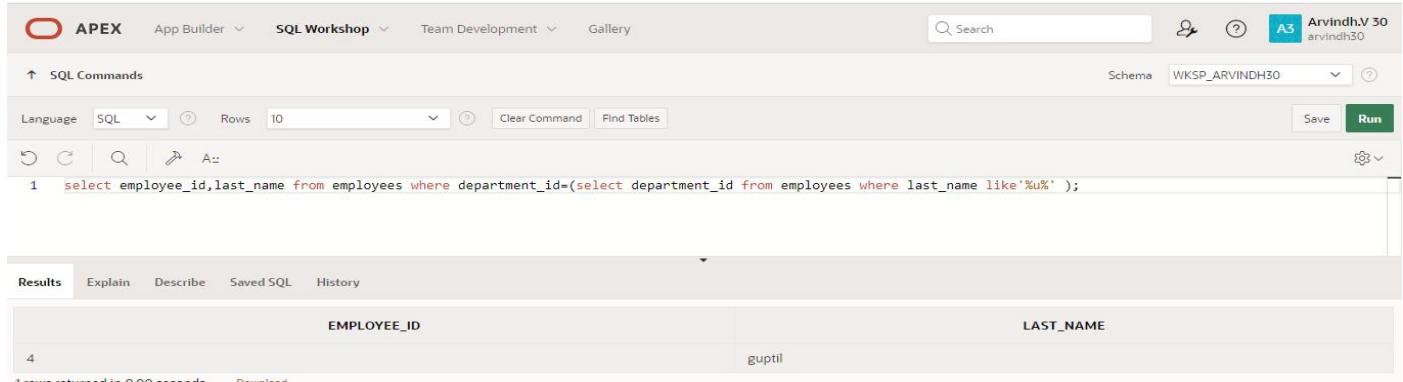
3 rows returned in 0.01 seconds

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 where department_id=(select department_id from employees where last_name like'%u%');
```

OUTPUT:



The screenshot shows the Oracle APEX 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 'Arvindh.V 30 arvindh30'. The main area has tabs for 'SQL Commands', 'Schema' set to 'WKSP_ARVINDH30', and 'Run' button. The SQL command entered is:

```
1 select employee_id,last_name from employees where department_id=(select department_id from employees where last_name like'%u%');
```

The results tab is selected, showing a single row:

EMPLOYEE_ID	LAST_NAME
4	guptil

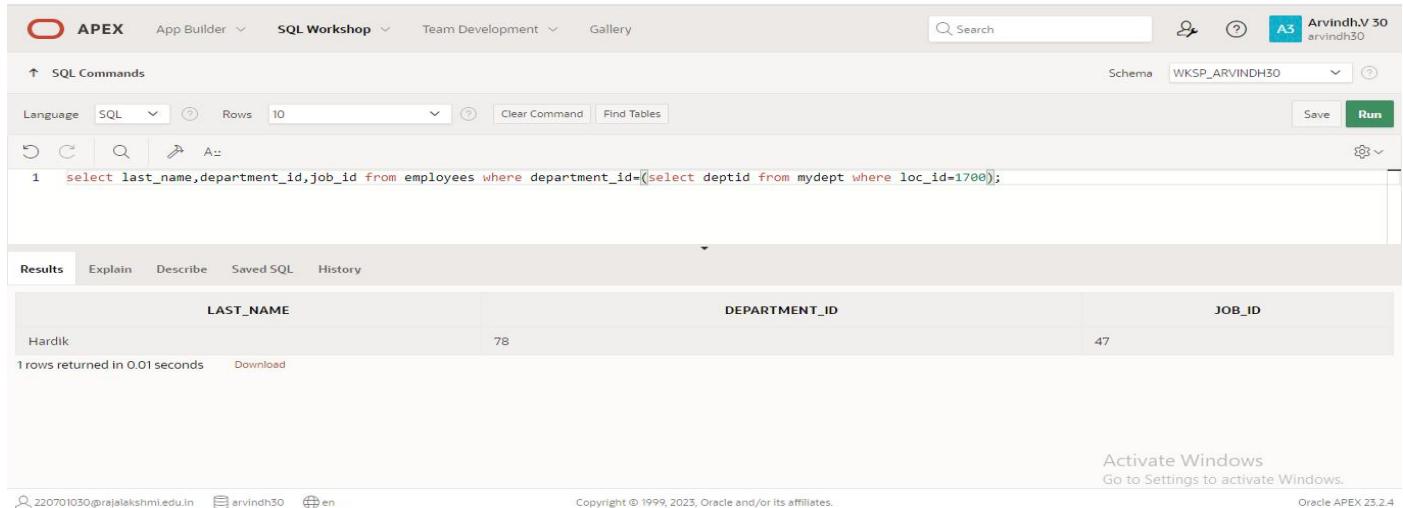
1 rows returned in 0.00 seconds. There are 'Download' and 'Run' buttons at the bottom.

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

QUERY:

```
select last_name,department_id,job_id from employees where department_id=(select dept_id from departments where loc_id=1700);
```

OUTPUT:



The screenshot shows the Oracle APEX 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 'Arvindh.V 30 arvindh30'. The main area has tabs for 'SQL Commands', 'Schema' set to 'WKSP_ARVINDH30', and 'Run' button. The SQL command entered is:

```
1 select last_name,department_id,job_id from employees where department_id=(select dept_id from departments where loc_id=1700);
```

The results tab is selected, showing a single row:

LAST_NAME	DEPARTMENT_ID	JOB_ID
Hardik	78	47

1 rows returned in 0.01 seconds. There are 'Download' and 'Run' buttons at the bottom. A watermark for 'Activate Windows Go to Settings to activate Windows.' is visible in the bottom right.

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

QUERY:

```
select last_name,salary from employees where manager_id=(select manager_id from employees  
where manager_name='King');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'ArvindhV 30' and schema 'WKSP_ARVINDH30'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is: `select last_name,salary from employees where manager_id=(select manager_id from employees where manager_name='King');`. The results section shows a single row: Dhoni with a salary of 20000. A message at the bottom right says 'Activate Windows'.

LAST_NAME	SALARY
Dhoni	20000

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

QUERY:

```
select department_id,last_name,job_id from employees where department_id in (select dept_id  
from departments where dept_name='Executive');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'ArvindhV 30' and schema 'WKSP_ARVINDH30'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is: `select department_id,last_name,job_id from employees where department_id in (select dept_id from departments where dept_name='Executive');`. The results section shows a single row: Hardik with department_id 78 and job_id 47. A message at the bottom right says 'Activate Windows'.

DEPARTMENT_ID	LAST_NAME	JOB_ID
78	Hardik	47

Activate Windows
Go to Settings to activate Windows.

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

QUERY:

```
select employee_id, last_name, salary from employees where salary > (select avg(salary) from employees where last_name like '%u%');
```

OUTPUT:

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

```
1 select employee_id, last_name, salary from employees where salary > (select avg(salary) from employees where last_name like '%u%');
```

The Results pane displays the output:

EMPLOYEE_ID	LAST_NAME	SALARY
43	Hardik	80200
309	Guptil	60000

2 rows returned in 0.00 seconds

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

RESULT:

USING THE SET OPERATORS

EX_NO:10

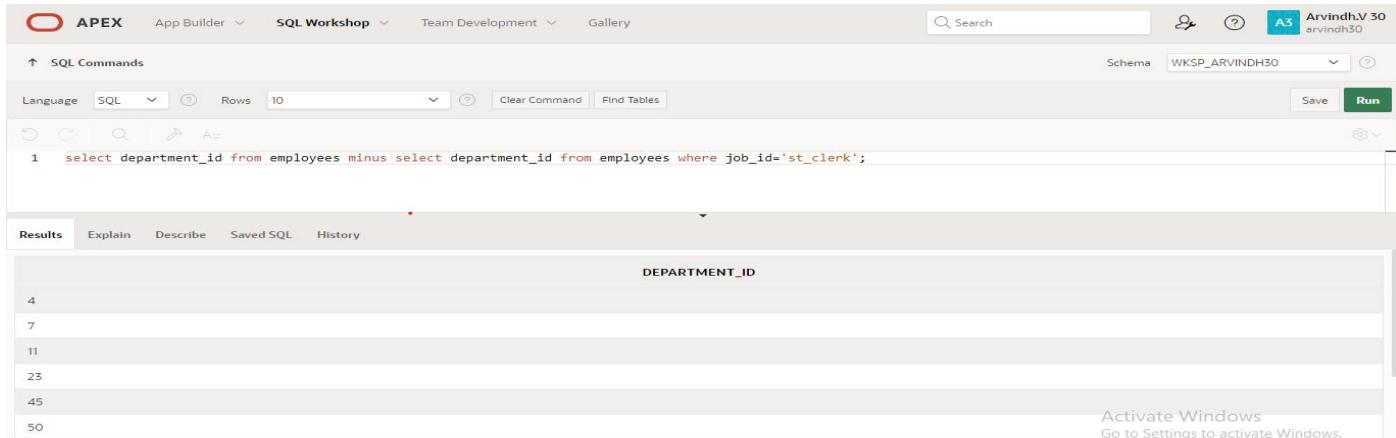
DATE:15/04/2024

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 department_id from employees minus select department_id from employees where job_id='st_clerk';
```

OUTPUT:



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

```
1 select department_id from employees minus select department_id from employees where job_id='st_clerk';
```

The results show the following department IDs:

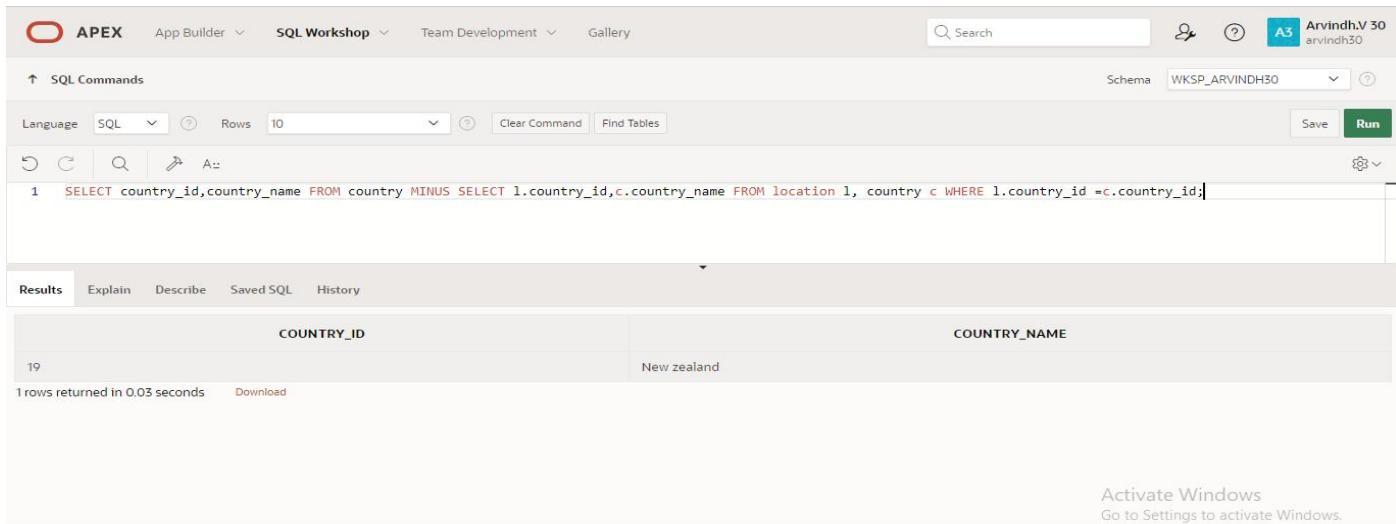
DEPARTMENT_ID
4
7
11
23
45
50

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,state_province from location minus select country_id,state_province from location,departments where location.location_id=departments.location_id;
```

OUTPUT:



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

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

The results show one row:

COUNTRY_ID	COUNTRY_NAME
19	New zealand

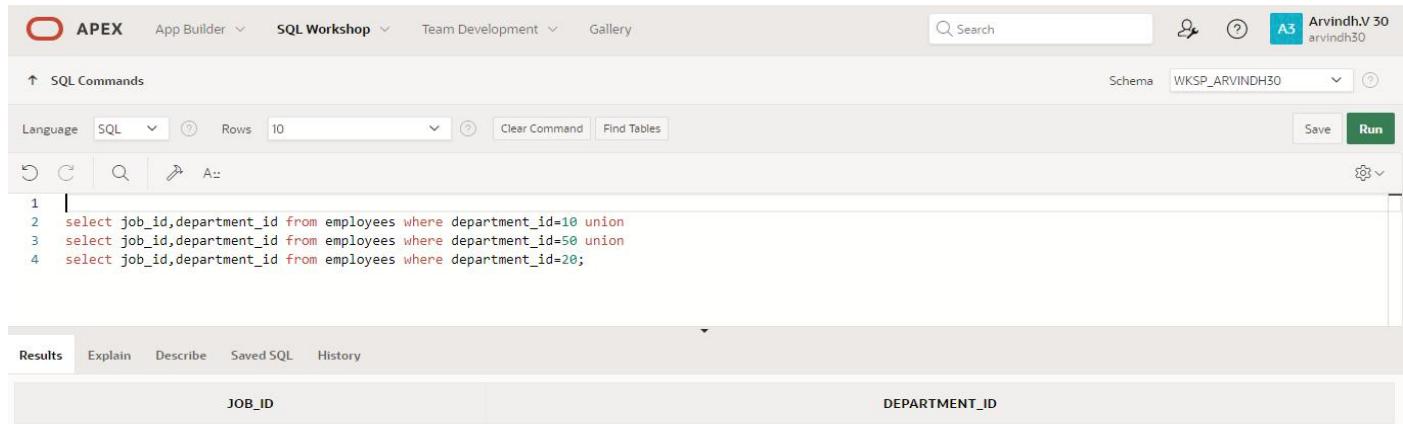
1 rows returned in 0.03 seconds

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 job_id,department_id from employees where department_id=10 union  
select job_id,department_id from employees where department_id=50 union  
select job_id,department_id from employees where department_id=20;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema is set to WKSP_ARVINDH30. The SQL command window contains the following code:

```
1 |  
2 | select job_id,department_id from employees where department_id=10 union  
3 | select job_id,department_id from employees where department_id=50 union  
4 | select job_id,department_id from employees where department_id=20;
```

The results section shows the output:

JOB_ID	DEPARTMENT_ID
244	50

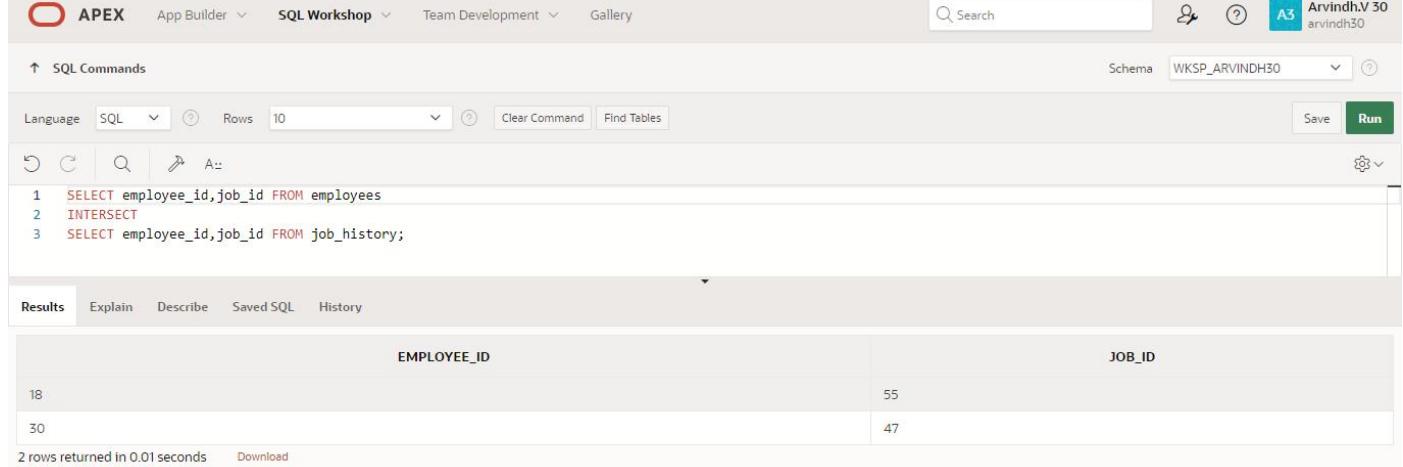
1 rows returned in 0.01 seconds [Download](#)

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_id FROM employees  
INTERSECT  
SELECT employee_id,job_id FROM job_history;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema is set to WKSP_ARVINDH30. The SQL command window contains the following code:

```
1 | SELECT employee_id,job_id FROM employees  
2 | INTERSECT  
3 | SELECT employee_id,job_id FROM job_history;
```

The results section shows the output:

EMPLOYEE_ID	JOB_ID
18	55
30	47

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

5.)The HR department needs a report with the following specifications: - Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department. - Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them Write a compound query to accomplish this.

QUERY:

```
SELECT last_name,department_id,TO_CHAR(null) FROM employees  
UNION SELECT TO_CHAR(null),deptid,dept_name FROM mydept
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL editor contains the following code:

```
1  SELECT last_name,department_id,TO_CHAR(null) FROM employees  
2  UNION SELECT TO_CHAR(null),deptid,dept_name FROM mydept  
3
```

The results tab displays the output of the query:

Jadeja	245	-
gayle	79	-
guptil	4	-
kolhi	87	-
plesis	80	-
raina	84	-
-	78	Executive

At the bottom right of the results table, there is a watermark: "Activate Windows Go to Settings to activate Windows".

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

RESULT:

CREATING VIEWS

EX_NO:11

DATE:16/04/20

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 OR REPLACE VIEW employees_vu AS SELECT employee_id, last_name employee,
department_id FROM employees;
```

OUTPUT:

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

```
CREATE OR REPLACE VIEW employees_vu AS SELECT employee_id, last_name employee,
department_id FROM employees;
```

Below the command, the results tab is active, showing the message "View created." and a execution time of "0.05 seconds".

2.) Display the contents of the EMPLOYEES_VU view.

QUERY:

```
select * from employees_vu;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface with the results of the previous query. The results table has three columns: EMPLOYEE_ID, EMPLOYEE, and DEPARTMENT_ID. The data is as follows:

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
30	Dhoni	7
35	plesis	80
18	kolhi	87
50	Davies	97
333	gayle	79
43	Hardik	78
4	guptil	4

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 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 search bar, a user icon, and a session identifier "Arvindh.V 30 arvindh30". The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the query: "1 SELECT view_name, text FROM user_views;". The Results tab displays the output:

VIEW_NAME	TEXT
DEPT50	SELECT employee_id empno, last_name employee, department_id deptno FROM employees WHERE department_id = 50 WITH CHECK OPTION
EMPLOYEES_VU	SELECT employee_id, last_name employee, department_id FROM employees

Below the table, it says "2 rows returned in 0.00 seconds" and has a "Download" link.

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

QUERY:

```
SELECT employee, department_id FROM employees_vu;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one but with a different query. The top navigation bar and session information are the same. The SQL Commands tab is active, showing the query: "1 SELECT employee, department_id FROM employees_vu;". The Results tab displays the output:

EMPLOYEE	DEPARTMENT_ID
Dhoni	7
plesis	80
kolhi	87
Davies	97
gayle	79
Hardik	78

At the bottom of the results table, there's a watermark: "Activate Windows Go to Settings to activate Windows." The footer also includes copyright information: "Copyright © 1999, 2023, Oracle and/or its affiliates." and "Oracle APEX 23.2.4".

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 AS SELECT employee_id empno, last_name employee, department_id deptno
FROM employees WHERE department_id = 50 WITH CHECK OPTION CONSTRAINT emp_dept_50;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Arvindhv 30' and schema 'WKSP_ARVINDH30'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL Commands tab contains the following code:

```
1 CREATE VIEW dept50 AS SELECT employee_id empno, last_name employee, department_id deptno
2 FROM employees
3 WHERE department_id = 50 WITH CHECK OPTION CONSTRAINT emp_dept_50;
```

The Results tab shows the message 'View created.' and execution time '0.05 seconds'. The bottom status bar includes user information (220701030@rajalakshmi.edu.in, arvindh30, en), copyright (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version (Oracle APEX 23.2.4). An 'Activate Windows' watermark is present.

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

QUERY:

```
Describe dept50;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface with the 'Describe' tab selected in the results panel. The SQL Commands tab contains the code '1 Describe dept50;'. The results show the structure of the DEPT50 view:

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

The bottom status bar includes user information (220701030@rajalakshmi.edu.in, arvindh30, en), copyright (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version (Oracle APEX 23.2.4). An 'Activate Windows' watermark is present.

7.) Attempt to reassign Matos to department 80

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows a user profile for 'Arvindh 30' and the schema 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below it, there are buttons for 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. The SQL command entered is: `1 UPDATE dept50 SET deptno=80 WHERE employee='Matos';`. The results section shows the output: `1 row(s) updated.` and `0.03 seconds`. The bottom right corner displays a promotional message: `Activate Windows
Go to Settings to activate Windows.`

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_grade 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. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows a user profile for 'Arvindh.V 30' and the schema 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below it, there are buttons for 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. The SQL command entered is: `1 create or replace view salary_vu as select e.last_name "Employee",d.dept_name Department, e.salary "Salary",j.grade_level "Grades"
2 from employees e,mydept d,job_grades j
3 where e.department_id=d.deptid and e.salary between j.lowest_sal and j.highest_sal;`. The results section shows the output: `View created.` and `0.02 seconds`. The bottom right corner displays a promotional message: `Activate Windows
Go to Settings to activate Windows.`

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

RESULT:

EXERCISE 12

PRACTICE QUESTIONS

EX NO : 12

DATE:23/04/2024

Intro to Constraints; NOT NULL and UNIQUE Constraints

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Table Created.

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

```
DESCRIBE f_global_locations;
```

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

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

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

PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

1. What is the purpose of a

- PRIMARY KEY
- FOREIGN KEY
- CHECK CONSTRAINT

a. PRIMARY KEY

Uniquely identify each row in table.

b. FOREIGN KEY

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

c. CHECK CONSTRAINT

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

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

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.

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

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

ANIMAL_ID	NAME	LICENSE_TAG_NUMBER	ADMIT_DATE	ADOPTION_ID	VACCINATION_DATE
101	Spot	35540	10-Oct-2004	205	12-Oct-2004

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

```
SELECT * FROM animals;
```

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

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

a. ON DELETE CASCADE

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

b. ON DELETE SET NULL

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

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

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

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

RESULT:

PRACTICE PROBLEM

Managing Constraints

Using Oracle Application Express, click the SQL Workshop tab in the menu bar. Click the Object Browser and verify that you have a table named copy_d_clients and a table named copy_d_events. If you don't have these tables in your schema, create them before completing the exercises below. Here is how the original tables are related. The d_clients table has a primary key client_number. This has a primary-key constraint and it is referenced in the foreign-key constraint on the d_events table.

NOTE: The practice exercises use the d_clients and d_events tables in the DJs on Demand database. Students will work with copies of these two tables named copy_d_clients and copy_d_events. Make sure they have new copies of the tables (without changes made from previous exercises). Remember, tables copied using a subquery do not have the integrity constraints as established in the original tables. When using the SELECT statement to view the constraint name, the tablename must be all capital letters.

1. What are four functions that an ALTER statement can perform on constraints?

- ADD
- DROP
- ENABLE
- DISABLE

2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy_d_clients table. Name the primary key copy_d_clients_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy_d_clients.table?

```
ALTER TABLE copy_d_clients
ADD CONSTRAINT copy_d_clt_client_number_pk PRIMARY KEY (client_number);
```

3. Create a FOREIGN KEY constraint in the copy_d_events table. Name the foreign key copy_d_events_fk. This key references the copy_d_clients table client_number column. What is the syntax you used to create the FOREIGN KEY constraint in the copy_d_events table?

```
ALTER TABLE copy_d_events
ADD CONSTRAINT copy_d_eve_client_number_fk FOREIGN KEY (client_number) REFERENCES
copy_d_clients (client_number) ENABLE;
```

4. Use a SELECT statement to verify the constraint names for each of the tables. Note that the tablename must be capitalized.

```
SELECT constraint_name, constraint_type, table_name
FROM user_constraints
WHERE table_name = UPPER('copy_d_events');
```

a. The constraint name for the primary key in the copy_d_clients table is_____.

COPY_D_CLT_CLIENT_NUMBER_PK

5. Drop the PRIMARY KEY constraint on the copy_d_clients table. Explain your results.

```
ALTER TABLE copy_d_clients
DROP CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK CASCADE ;
```

6. Add the following event to the copy_d_events table. Explain your results.

ID	NAME	EVENT_DATE	DESCRIPTION	COST	VENUE_ID	PACKAGE_CODE	THEME_CODE	CLIENT_NUMBER
140	Cline Bas Mitzvah	15-Jul-2004	Church and Private Home formal	4500	105	87	77	7125

```
INSERT INTO
copy_d_events(client_number,id,name,event_date,description,cost,venue_id,package_code,theme_code)
VALUES(7125,140,'Cline Bas Mitzvah',TO_DATE('15-Jul-2004','dd-Mon-yyyy'),'Church and Private Home formal',4500,105,87,77);
```

RESULT: ORA-02291: integrity constraint (HKUMAR.COPY_D_EVE_CLIENT_NUMBER_FK) violated - parent key not found

7. Create an ALTER TABLE query to disable the primary key in the copy_d_clients table. Then add the values from #6 to the copy_d_events table. Explain your results.

```
ALTER TABLE copy_d_clients
DISABLE CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK CASCADE;
```

8. Repeat question 6: Insert the new values in the copy_d_events table. Explain your results.

```
INSERT INTO
copy_d_events(client_number,id,name,event_date,description,cost,venue_id,package_code,theme_code)
VALUES(7125,140,'Cline Bas Mitzvah',TO_DATE('15-Jul-2004','dd-Mon-yyyy'),'Church and Private Home formal',4500,105,87,77);
```

1 row(s) inserted.

9. Enable the primary-key constraint in the copy_d_clients table. Explain your results.

```
ALTER TABLE copy_d_clients
ENABLE CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK ;
```

10. If you wanted to enable the foreign-key column and reestablish the referential integrity between these two tables, what must be done?

```
DELETE FROM copy_d_events WHERE
client_number NOT IN ( SELECT client_number FROM copy_d_clients);
```

1 row(s) deleted.

```
ALTER TABLE copy_d_events
ENABLE CONSTRAINT COPY_D_EVE_CLIENT_NUMBER_FK;
```

Table altered.

11. Why might you want to disable and then re-enable a constraint?

Generally to make bulk operations fast, where my input data is diligently sanitized and I am sure, it is safe to save some time in this clumsy process.

12. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?

Queries are same as in point 2,3, 4 above.

- C - Check constraint
Sub-case - if I see SEARCH_CONDITION something like "FIRST_NAME" IS NOT NULL , its a NOT NULL constraint.
- P - Primary key
- R - Referential integrity (fk)
- U - Unique key

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

EXERCISE 13

Creating Views

EX NO:13

DATE:16/05/2024

- 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.**

- 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';
```

- 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](#)

- 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';
```

- 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

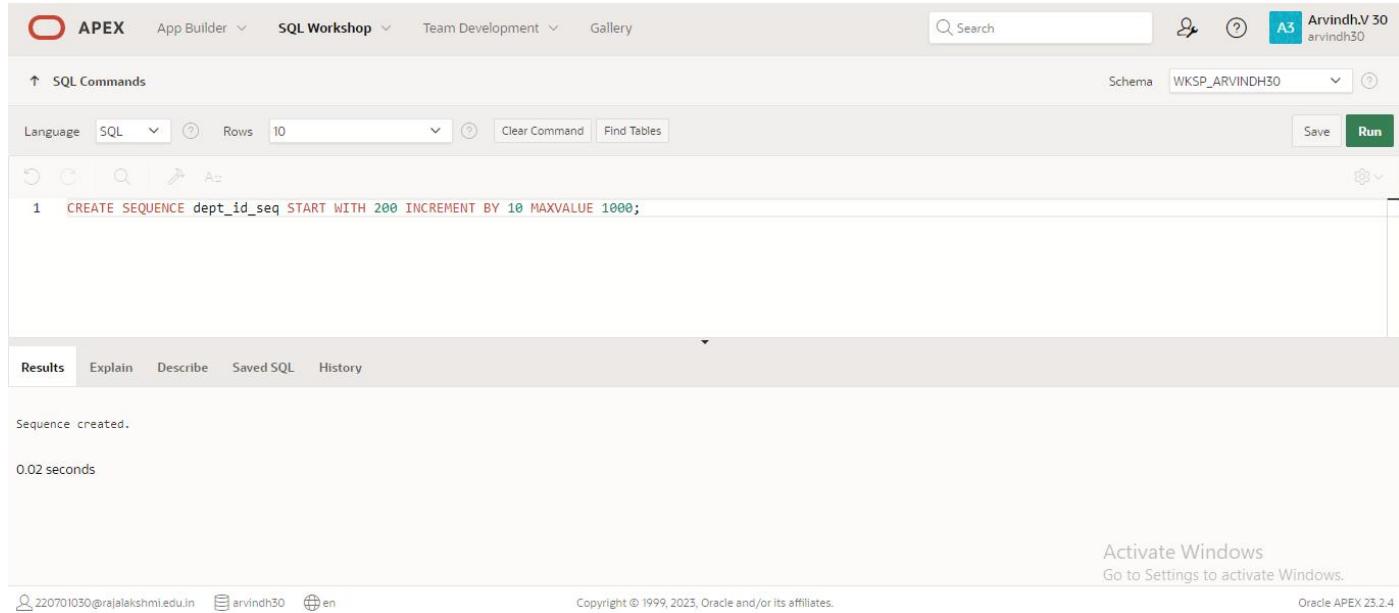
DATE:16/05/2024

1.)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:



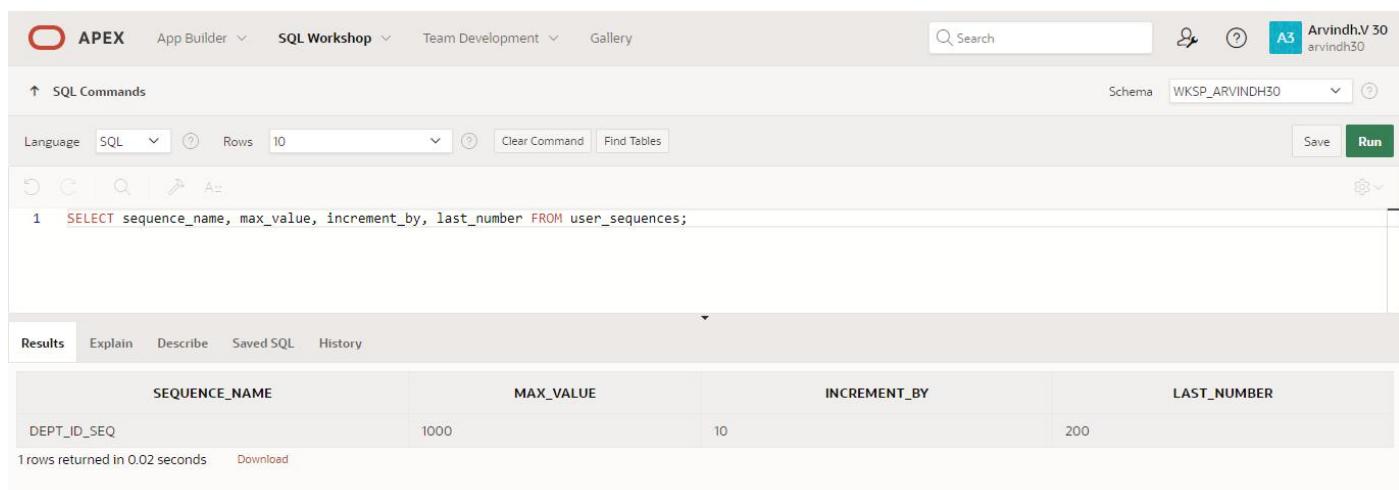
The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The search bar contains 'Search'. On the right, there's a user icon 'Arvindh.V 30 arvindh30' and a 'Save' button. The main workspace has tabs for 'SQL Commands' (selected), 'Language' (set to 'SQL'), 'Rows' (set to 10), and buttons for 'Clear Command' and 'Find Tables'. Below this, a code editor shows the SQL command: 'CREATE SEQUENCE dept_id_seq START WITH 200 INCREMENT BY 10 MAXVALUE 1000;'. The results tab shows the output: 'Sequence created.' and '0.02 seconds'. At the bottom, it displays the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

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 APEX SQL Workshop interface, similar to the previous one but with a different query. The top navigation bar and workspace are identical. The code editor now contains the query: 'SELECT sequence_name, max_value, increment_by, last_number FROM user_sequences;'. The results tab displays a table with four columns: 'SEQUENCE_NAME', 'MAX_VALUE', 'INCREMENT_BY', and 'LAST_NUMBER'. A single row is shown: 'DEPT_ID_SEQ', '1000', '10', and '200'. Below the table, it says '1 rows returned in 0.02 seconds' and has a 'Download' link. The bottom of the screen shows the copyright notice and version information.

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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as Arvindh.v 30 (arvindh30). The SQL Commands tab is active, showing the following SQL code:

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

The Results tab displays the output of the query:

ID	NAME
200	Education

Below the results, it says "1 rows returned in 0.01 seconds". The bottom of the page includes copyright information and a link to activate Windows.

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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as Arvindh.v 30 (arvindh30). The SQL Commands tab is active, showing the following SQL code:

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

The Results tab displays the output of the query:

Index created.
0.03 seconds

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The top right corner shows the user's name, Arvindh.V 30, and session ID, arvindh30. The main workspace has a search bar and a schema dropdown set to WKSP_ARVINDH30. Below the search bar are buttons for Save and Run. The SQL Commands section contains the query: `1 SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';`. The Results tab is selected, displaying a table with three columns: INDEX_NAME, TABLE_NAME, and UNIQUENESS. The table shows one row: EMP_DEPT_ID_IDX, EMPLOYEES, and NONUNIQUE. At the bottom left, it says "1 rows returned in 0.04 seconds". The bottom right of the page displays a watermark: "Activate Windows Go to Settings to activate Windows." and the copyright notice "Copyright © 1999, 2023, Oracle and/or its affiliates." The footer also includes the URL "220701030@rajalakshmi.edu.in" and the session ID "arvindh30".

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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there are user profile icons for 'Arvindh,V 30' and 'arvindh30'. The main workspace is titled 'SQL Commands'. It features a toolbar with buttons for Undo, Redo, Find, Replace, and Run. Below the toolbar, the code editor displays the following PL/SQL block:

```
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;
```

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

```
Incentive = 7200
Statement processed.

0.01 seconds
```

At the bottom of the interface, there are footer links for Activate Windows, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.

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 APEX SQL Workshop interface. In the SQL Commands tab, a PL/SQL block is entered:

```
1 DECLARE
2 WELCOME varchar2(10) := 'welcome';
3 BEGIN
4 DBMS_Output.Put_Line("Welcome");
5 END;
6 /
```

The output pane shows the following error message:

```
Error at line 4/23: ORA-06550: line 4, column 23:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.MV_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored
```

Below the error message, the original code is repeated:

```
2. WELCOME varchar2(10) := 'welcome';
3. BEGIN
4. DBMS_Output.Put_Line("Welcome");
5. END;
6. /
```

At the bottom of the page, there are footer links for user information, copyright notice, and Oracle APEX version.

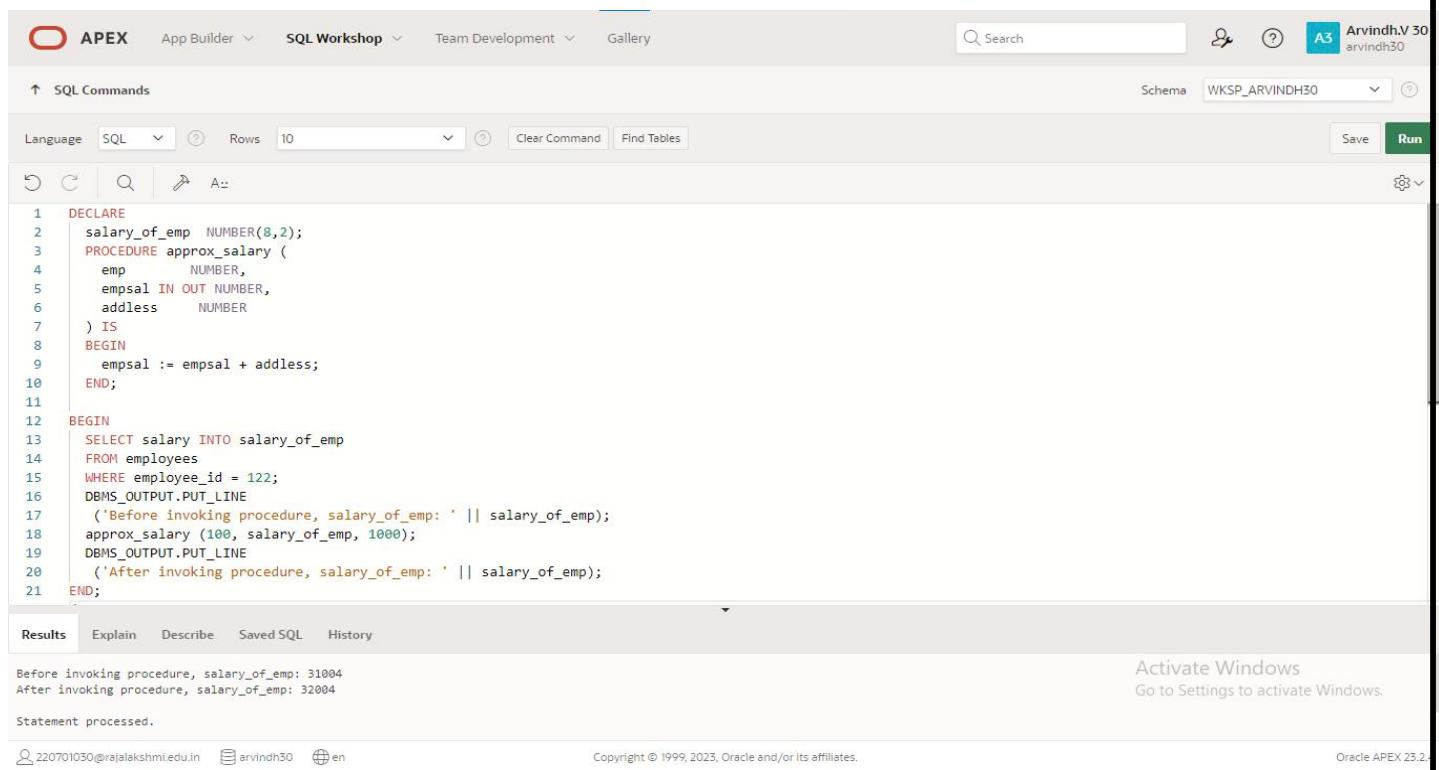
3.) Write a PL/SQL block to adjust the salary of the employee whose ID 122.

QUERY:

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

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

OUTPUT:



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. The right side shows the user 'Arvindh.V 30' and schema 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' and contains the PL/SQL code from the previous section. The code is numbered from 1 to 21. The 'Run' button is visible at the bottom right of the command input field. Below the code, the 'Results' tab is selected, showing the output of the executed code. The output consists of two lines: 'Before invoking procedure, salary_of_emp: 31000' and 'After invoking procedure, salary_of_emp: 32000'. At the very bottom, a message says 'Statement processed.'

```
1  DECLARE
2      salary_of_emp NUMBER(8,2);
3      PROCEDURE approx_salary (
4          emp      NUMBER,
5          empsal IN OUT NUMBER,
6          addless  NUMBER
7      ) IS
8      BEGIN
9          empsal := empsal + addless;
10     END;
11
12 BEGIN
13     SELECT salary INTO salary_of_emp
14     FROM employees
15     WHERE employee_id = 122;
16     DBMS_OUTPUT.PUT_LINE
17     ('Before invoking procedure, salary_of_emp: ' || salary_of_emp);
18     approx_salary (100, salary_of_emp, 1000);
19     DBMS_OUTPUT.PUT_LINE
20     ('After invoking procedure, salary_of_emp: ' || salary_of_emp);
21 END;
```

Before invoking procedure, salary_of_emp: 31000
After invoking procedure, salary_of_emp: 32000

Statement processed.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX 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 'ArvindhV 30' (arvindh30). The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_ARVINDH30'. The code editor contains the PL/SQL procedure definition. The bottom section shows the results of the execution, displaying the message 'Procedure created.' and a timestamp of '0.03 seconds'.

```
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;
```

Results Explain Describe Saved SQL History

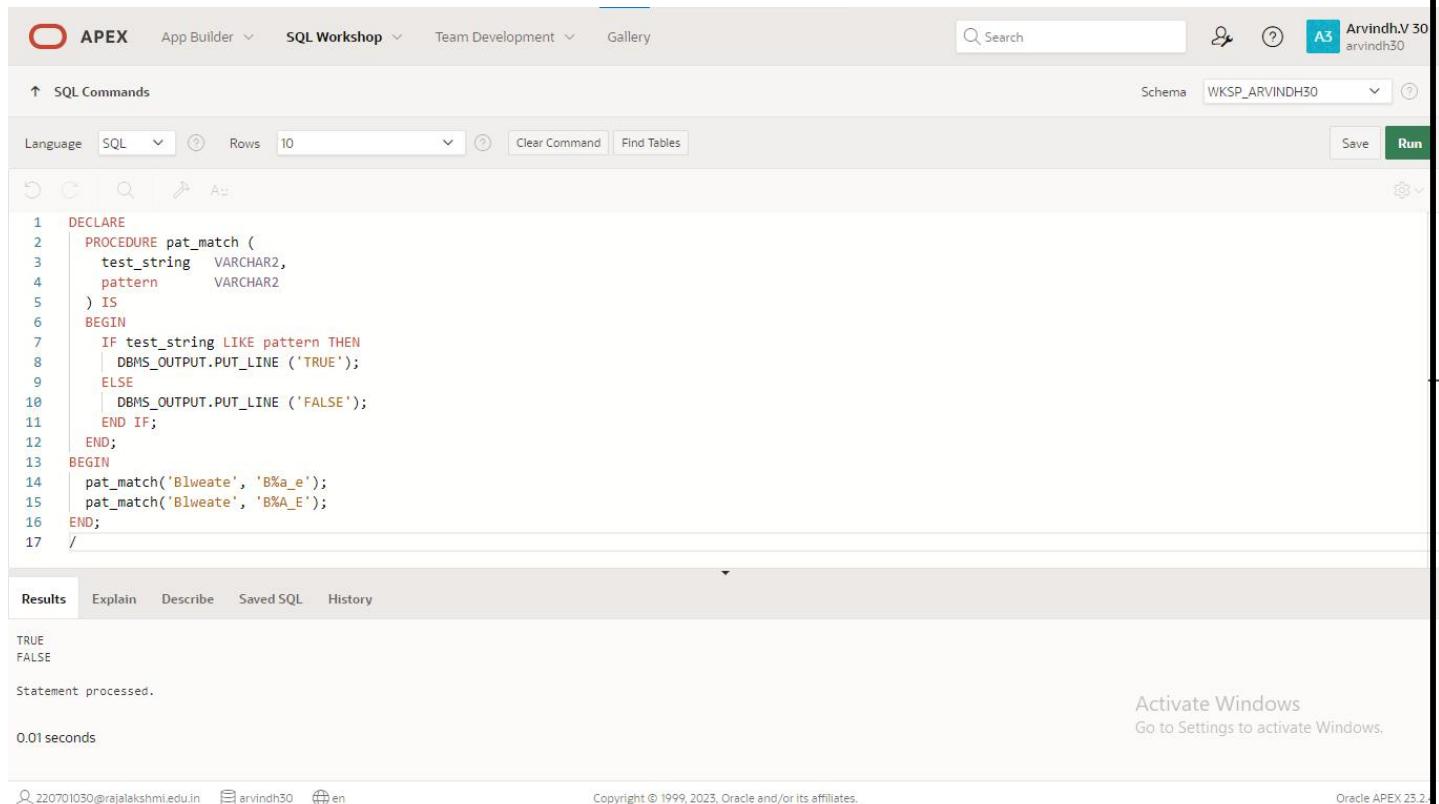
Procedure created.
0.03 seconds

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:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for Arvindh.V 30 (arvindh30). The main workspace is titled 'SQL Commands' and contains the PL/SQL code from the previous section. Below the code, the 'Results' tab is selected, displaying the output: 'TRUE' and 'FALSE'. The status bar at the bottom indicates the statement was processed in 0.01 seconds.

```
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

Activate Windows
Go to Settings to activate Windows.

220701030@rajalakshmi.edu.in arvindh30 en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.

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

QUERY:

DECLARE

```
num_small NUMBER := 8;
```

```
num_large NUMBER := 5;
```

```
num_temp NUMBER;
```

```
BEGIN
```

```
IF num_small > num_large THEN
```

```
    num_temp := num_small;
```

```
    num_small := num_large;
```

```
    num_large := num_temp;
```

```
END IF;
```

```
DBMS_OUTPUT.PUT_LINE ('num_small = '||num_small);
```

```
DBMS_OUTPUT.PUT_LINE ('num_large = '||num_large);
```

```
END;
```

```
/
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side of the header shows a user profile for 'ArvindhV 30' and a schema dropdown set to 'WKSP_ARVINDH30'. The main workspace is titled 'SQL Commands' and contains the following PL/SQL code:

```
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 END;
16 /
```

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

```
num_small = 5
num_large = 8

Statement processed.

0.01 seconds
```

On the right side of the results, there is a promotional message for activating Windows:

Activate Windows
Go to Settings to activate Windows.

At the bottom of the page, there are footer links for copyright information and Oracle APEX version:

Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 25.2.

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:

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. The SQL Workshop tab is selected. The main area displays a PL/SQL code editor with the following content:

```
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(3) := 'No';
10     BEGIN
11         IF sal_achieve > (target_qty + 200) THEN
12             incentive := (sal_achieve - target_qty)/4;
13             UPDATE employees
14             SET salary = salary + incentive
15             WHERE employee_id = emp_id;
16             updated := 'Yes';
17         END IF;
18         DBMS_OUTPUT.PUT_LINE (
19             'Table updated? ' || updated || ', '
20             'incentive = ' || incentive || '.'
21         );
22     END test1;
```

The results pane shows the output of the procedure execution:

```
Table updated? Yes, incentive = 75.
Table updated? Yes, incentive = 150.
```

Below the results, it says "1 row(s) updated." and shows the user information: 220701050@rajalakshmi.edu.in and arvindh30. The copyright notice at the bottom is "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version is "Oracle APEX 23.2".

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:

```
1  DECLARE
2    PROCEDURE test1 (sal_achieve NUMBER)
3    IS
4      incentive NUMBER := 0;
5    BEGIN
6      IF sal_achieve > 44000 THEN
7        | incentive := 1800;
8      ELSIF sal_achieve > 32000 THEN
9        | incentive := 800;
10     ELSE
11        | incentive := 500;
12     END IF;
13   DBMS_OUTPUT.NEW_LINE;
14   DBMS_OUTPUT.PUT_LINE (
15     'Sale achieved : ' || sal_achieve || ', incentive : ' || incentive || '.'
16   );
17   END test1;
18   BEGIN
19     test1(45000);

```

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

Statement processed.

Activate Windows
Go to Settings to activate Windows.

Copyright © 1999, 2023, Oracle and/or its affiliates.

Oracle APEX 23.2

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

 tot_emp NUMBER;

BEGIN

 SELECT Count(*)

 INTO tot_emp

 FROM employees e

 join mydept d

 ON e.department_id = d.deptid

 WHERE e.department_id = 50;

 dbms_output.Put_line ('The employees are in the department 50: '

 ||To_char(tot_emp));

 IF tot_emp >= 45 THEN

 dbms_output.Put_line ('There are no vacancies in the department 50.');

 ELSE

 dbms_output.Put_line ('There are some vacancies in department 50.');

 END IF;

```
END;
```

```
/
```

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. The right side of the header shows the schema 'WKSP_ARVINDH30' and a user icon 'Arvindh30'. The main workspace is titled 'SQL Commands' and contains the following PL/SQL code:

```
1  DECLARE
2  |      tot_emp NUMBER;
3  BEGIN
4      SELECT Count(*)
5          INTO tot_emp
6          FROM employees e
7              JOIN mydept d
8                  ON e.department_id = d.deptid
9      WHERE e.department_id = 50;
10
11     dbms_output.Put_line ('The employees are in the department 50: '
12                           || To_char(tot_emp));
13
14     IF tot_emp >= 45 THEN
15         dbms_output.Put_line ('There are no vacancies in the department 50.');
16     ELSE
17         dbms_output.Put_line ('There are some vacancies in department 50.');
18     END IF;
19 END;
20 /
```

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

```
The employees are in the department 50: 0
There are some vacancies in department 50.

Statement processed.

0.03 seconds
```

On the right side of the results area, there is a message: "Activate Windows" with the subtext "Go to Settings to activate Windows." At the bottom of the page, there are copyright notices for Oracle and APEX, along with a footer link for Oracle APEX 23.2.

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

QUERY:

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

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

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

```
END IF;
```

```
END;
```

```
/
```

OUTPUT:

The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. The code area contains a PL/SQL block that counts employees in department 80 and prints the result. The results pane shows the output: "The employees are in the department 80 is: 2" and "There are 43 vacancies in department 80".

```
1  DECLARE
2      tot_emp NUMBER;
3      get_dep_id NUMBER;
4
5  BEGIN
6      get_dep_id := 80;
7      SELECT Count(*)
8          INTO  tot_emp
9         FROM employees e
10            JOIN mydept d
11              ON e.department_id = d.department_id
12     WHERE e.department_id = get_dep_id;
13
14     dbms_output.Put_line ('The employees are in the department'||get_dep_id||' is: '
15                           ||To_char(tot_emp));
16
17     IF tot_emp >= 45 THEN
18         dbms_output.Put_line ('There are no vacancies in the department'||get_dep_id);
19     ELSE
20         dbms_output.Put_line ('There are'||to_char(45-tot_emp)||' vacancies in department'|| get_dep_id );
21     END IF;
22 END;
23 /
```

Results Explain Describe Saved SQL History

The employees are in the department 80 is: 2
There are 43 vacancies in department 80
statement processed.
0.02 seconds

Activate Windows
Go to Settings to activate Windows.

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

QUERY:

```
DECLARE
```

```
v_employee_id employees.employee_id%TYPE;
```

```
v_full_name employees.first_name%TYPE;
```

```
v_job_id employees.job_id%TYPE;
```

```
v_hire_date employees.hire_date%TYPE;
```

```
v_salary employees.salary%TYPE;
```

```
CURSOR c_employees IS
```

```
    SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
    FROM employees;
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('Employee ID | Full Name | Job Title | Hire Date | Salary');
```

```
    DBMS_OUTPUT.PUT_LINE('-----');
```

```
    OPEN c_employees;
```

```
    FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
```

```
    WHILE c_employees%FOUND LOOP
```

```
        DBMS_OUTPUT.PUT_LINE(v_employee_id || ' ' || v_full_name || ' ' || v_job_id || ' ' ||
v_hire_date || ' ' || v_salary);
```

```
        FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
```

```
    END LOOP;
```

```
    CLOSE c_employees;
```

```
END;/
```

OUTPUT:

The screenshot shows the Oracle APEX 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 'Arvindh.V 30' with the schema 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The code editor contains a PL/SQL block:

```
1  DECLARE
2    v_employee_id employees.employee_id%TYPE;
3    v_full_name employees.first_name%TYPE;
4    v_job_id employees.job_id%TYPE;
5    v_hire_date employees.hire_date%TYPE;
6    v_salary employees.salary%TYPE;
7    CURSOR c_employees IS
8      SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
9        FROM employees;
10   BEGIN
11     DBMS_OUTPUT.PUT_LINE('Employee ID | Full Name | Job Title | Hire Date | Salary');
12     DBMS_OUTPUT.PUT_LINE('-----');
13     OPEN c_employees;
14     FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
15     WHILE c_employees%FOUND LOOP
16       DBMS_OUTPUT.PUT_LINE(v_employee_id || ' ' || v_full_name || ' ' || v_job_id || ' ' || v_hire_date || ' ' || v_salary);
17   END;
18
19  
```

The results tab displays the output of the program, listing employee details:

Employee ID	Full Name	Job Title	Hire Date	Salary
30	Ma Dhoni	47	09/07/2004	20000
35	faf de plesis	58	09/08/2000	8999
18	virat kolhi	55	12/07/2007	14000
50	steve Davies	1700	05/03/1997	11000
333	chris gayle	1701	04/05/1996	12000
43	Z Hardik	3	07/07/2001	80200
110	Martin Matos	47	07/03/2003	60000
3	suresh raina	49	02/05/1994	15000
122	Ravindra Jadeja	43	01/09/2006	31004

Below the results, a message says 'Activate Windows' with a link to 'Go to Settings to activate Windows.' The bottom status bar shows the statement was processed and includes the user's email (220701030@rajalakshmi.edu.in) and the Oracle APEX version (23.2).

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile 'Arvindh.V 30 arvindh30' are also present. The main area is titled 'SQL Commands' with tabs for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run.

```
1  DECLARE
2      CURSOR emp_cursor IS
3          SELECT e.employee_id, e.first_name, m.first_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  BEGIN
8      OPEN emp_cursor;
9      FETCH emp_cursor INTO emp_record;
10     WHILE emp_cursor%FOUND LOOP
11         DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.employee_id);
12         DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_record.first_name);
13         DBMS_OUTPUT.PUT_LINE('Manager Name: ' || emp_record.manager_name);
14         DBMS_OUTPUT.PUT_LINE('-----');
15     FETCH emp_cursor INTO emp_record;
16 END LOOP;
17 CLOSE emp_cursor;
```

The 'Results' tab is selected, displaying the output of the PL/SQL code:

```
Employee ID: 18
Employee Name: virat
Manager Name:
-----
Employee ID: 110
Employee Name: Martin
Manager Name:
-----
Employee ID: 30
Employee Name: Ms
Manager Name:
-----
```

On the right side of the results, there is an 'Activate Windows' message: "Activate Windows Go to Settings to activate Windows." At the bottom, the footer includes the URL "220701030@rajalakshmi.edu.in", the user "arvindh30", and the language "en". Copyright information "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version "Oracle APEX 23.2" are also present.

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

QUERY:

DECLARE

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

OUTPUT:

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. A search bar and user profile 'Arvindh.V 30 arvindh30' are also present. The main area is titled 'SQL Commands' with tabs for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run.

```
1  DECLARE
2      CURSOR job_cursor IS
3          SELECT e.job_id, j.lowest_sal
4              FROM job_grades j,employees e;
5      job_record job_cursor%ROWTYPE;
6
7      BEGIN
8          OPEN job_cursor;
9          FETCH job_cursor INTO job_record;
10         WHILE job_cursor%FOUND LOOP
11             DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_record.job_id);
12             DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_record.lowest_sal);
13             DBMS_OUTPUT.PUT_LINE('-----');
14             FETCH job_cursor INTO job_record;
15         END LOOP;
16         CLOSE job_cursor;
17     END;
18 /
```

The results pane displays the output of the PL/SQL code:

```
Job ID: 47
Minimum Salary: 15001
-----
Job ID: 50
Minimum Salary: 15001
-----
Job ID: 55
Minimum Salary: 15001
-----
Job ID: 1700
Minimum Salary: 15001
-----
```

On the right side of the results pane, there is an 'Activate Windows' message: 'Activate Windows Go to Settings to activate Windows.'

At the bottom, the footer includes the URL '220701030@rajalakshmi.edu.in', the schema 'arvindh30', and the language 'en'. Copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2' are also present.

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

QUERY:

DECLARE

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

```

    || Rpad(emp_sal_rec.last_name, 25)
    || Rpad(emp_sal_rec.job_id, 35)
    || To_char(emp_start_date, 'dd-mon-yyyy'));

END LOOP;
END;
/

```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the code editor, a PL/SQL block is written to print employee details. The results tab shows the output for two employees.

```

1  DECLARE
2      CURSOR employees_cur IS
3          SELECT employee_id,
4                  last_name,
5                  job_id,
6                  start_date
7              FROM employees
8          NATURAL JOIN job_history;
9      emp_start_date DATE;
10 BEGIN
11     dbms_output.Put_line(Rpad('Employee ID', 15)
12                           || Rpad('Last Name', 25)
13                           || Rpad('Job Id', 35)
14                           || 'Start Date');
15
16     dbms_output.Put_line('-----');
17

```

Employee ID	Last Name	Job Id	Start Date
125	Johnson	hr_rep	22-apr-1999
125	Johnson	hr_rep	22-apr-1999

Statement processed.

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

QUERY:

```

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

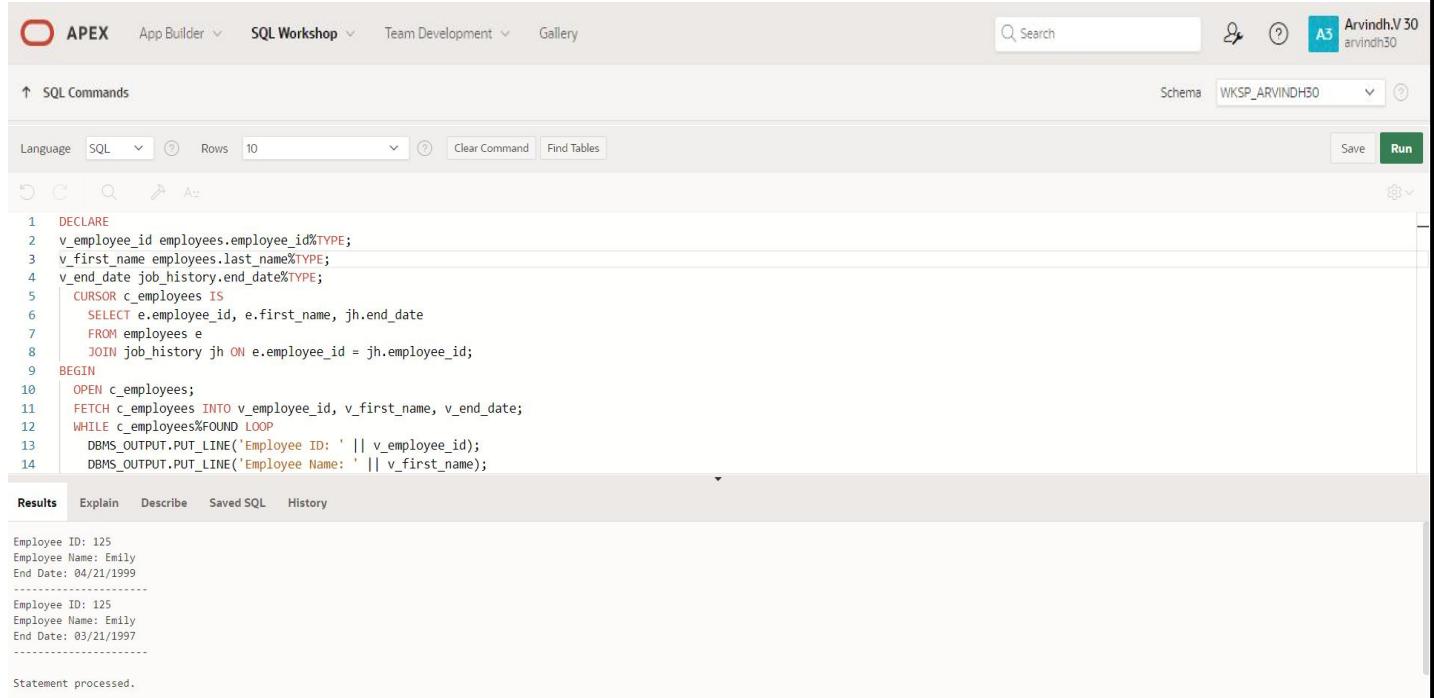
```

```

    FETCH c_employees INTO v_employee_id, v_first_name, v_end_date;
END LOOP;
CLOSE c_employees;
END;

```

OUTPUT:



The screenshot shows the Oracle 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 (Arvindh.V30), and a schema dropdown set to WKSP_ARVINDH50. Below the toolbar, the main area has tabs for Language (SQL selected), Rows (10), Clear Command, and Find Tables. The code editor contains a PL/SQL block. The results tab displays the output of the query.

```

1  DECLARE
2      v_employee_id employees.employee_id%TYPE;
3      v_first_name employees.last_name%TYPE;
4      v_end_date job_history.end_date%TYPE;
5      CURSOR c_employees IS
6          SELECT e.employee_id, e.first_name, jh.end_date
7              FROM employees e
8                 JOIN job_history jh ON e.employee_id = jh.employee_id;
9  BEGIN
10     OPEN c_employees;
11     FETCH c_employees INTO v_employee_id, v_first_name, v_end_date;
12     WHILE c_employees%FOUND LOOP
13         DBMS_OUTPUT.PUT_LINE('Employee ID: ' || v_employee_id);
14         DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_first_name);

```

Results

```

Employee ID: 125
Employee Name: Emily
End Date: 04/21/1999
-----
Employee ID: 125
Employee Name: Emily
End Date: 03/21/1997
-----
Statement processed.

```

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 search bar, a user icon for 'Arvindh.V 30 arvindh30', and a toolbar with Save and Run buttons. The main workspace is a SQL editor with the following content:

```
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;
```

Below the editor, a results panel shows the output of the executed code:

Results	Explain	Describe	Saved SQL	History
720				
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 APEX 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 'Arvindh.V 30' and the schema 'WKSP_ARVINDH30'. The main area is titled 'SQL Commands' and contains the following PL/SQL code:

```
1 CREATE OR REPLACE PROCEDURE get_book_info (
2     p_book_id IN NUMBER,
3     p_title IN OUT VARCHAR2,
4     p_author OUT VARCHAR2,
5     p_year_published OUT NUMBER
6 )
7 AS
8 BEGIN
9     SELECT title, author, year_published INTO p_title, p_author, p_year_published
10    FROM books
11   WHERE book_id = p_book_id;
12
13   p_title := p_title || ' - Retrieved';
14 EXCEPTION
15     WHEN NO_DATA_FOUND THEN
16         p_title := NULL;
17         p_author := NULL;
18         p_year_published := NULL;
19 END;
20
21
22 DECLARE
23     v_book_id NUMBER := 1;
24     v_title VARCHAR2(100);
```

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

```
Title: To Kill a Mockingbird - Retrieved
Author: Harper Lee
Year Published: 1960

Statement processed.
```

At the bottom, the footer includes user information (220701030@rajalakshmi.edu.in, arvindh30, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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

RESULT:

TRIGGER

EX_NO: 18

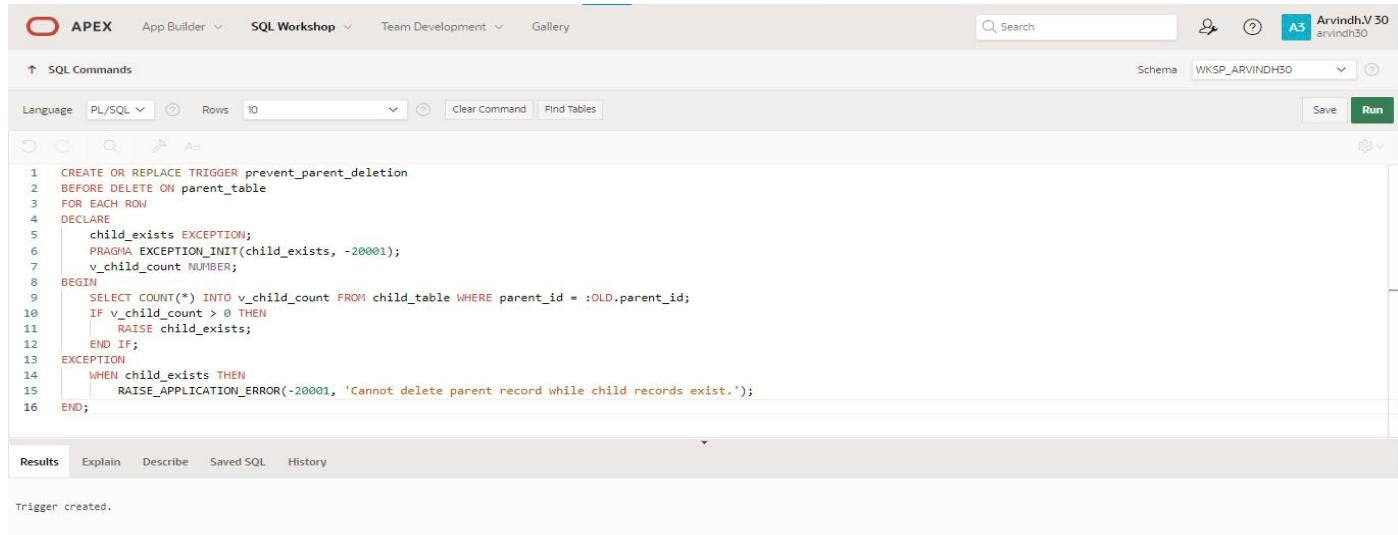
DATE:23/05/2024

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 with the following details:

- Top navigation bar: APEX, App Builder, SQL Workshop, Team Development, Gallery.
- Toolbar: Search, Refresh, Help, A3 icon, Arvindh.V 30, Schema: WKSP_ARVINDH30, Run button.
- Query editor:
 - Language: PL/SQL.
 - Rows: 10.
 - Buttons: Clear Command, Find Tables.
- Code area:

```
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;
```
- Results tab:

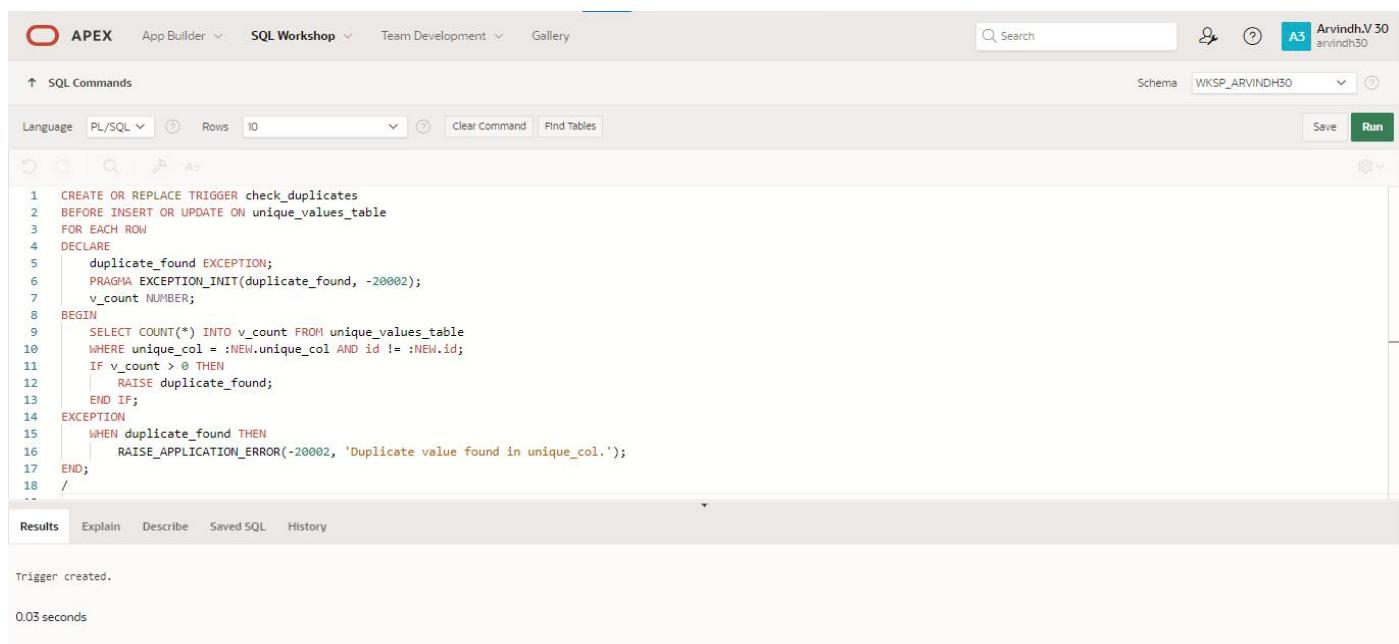
Trigger created.
0.04 seconds

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. The right side shows the schema WKS_ARVINDH30 and a run button. The main area displays the PL/SQL code for the trigger. The code is highlighted in blue and black, with syntax errors in red. The output pane at the bottom shows the message "Trigger created." and a execution time of "0.03 seconds".

```
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;
18 /
```

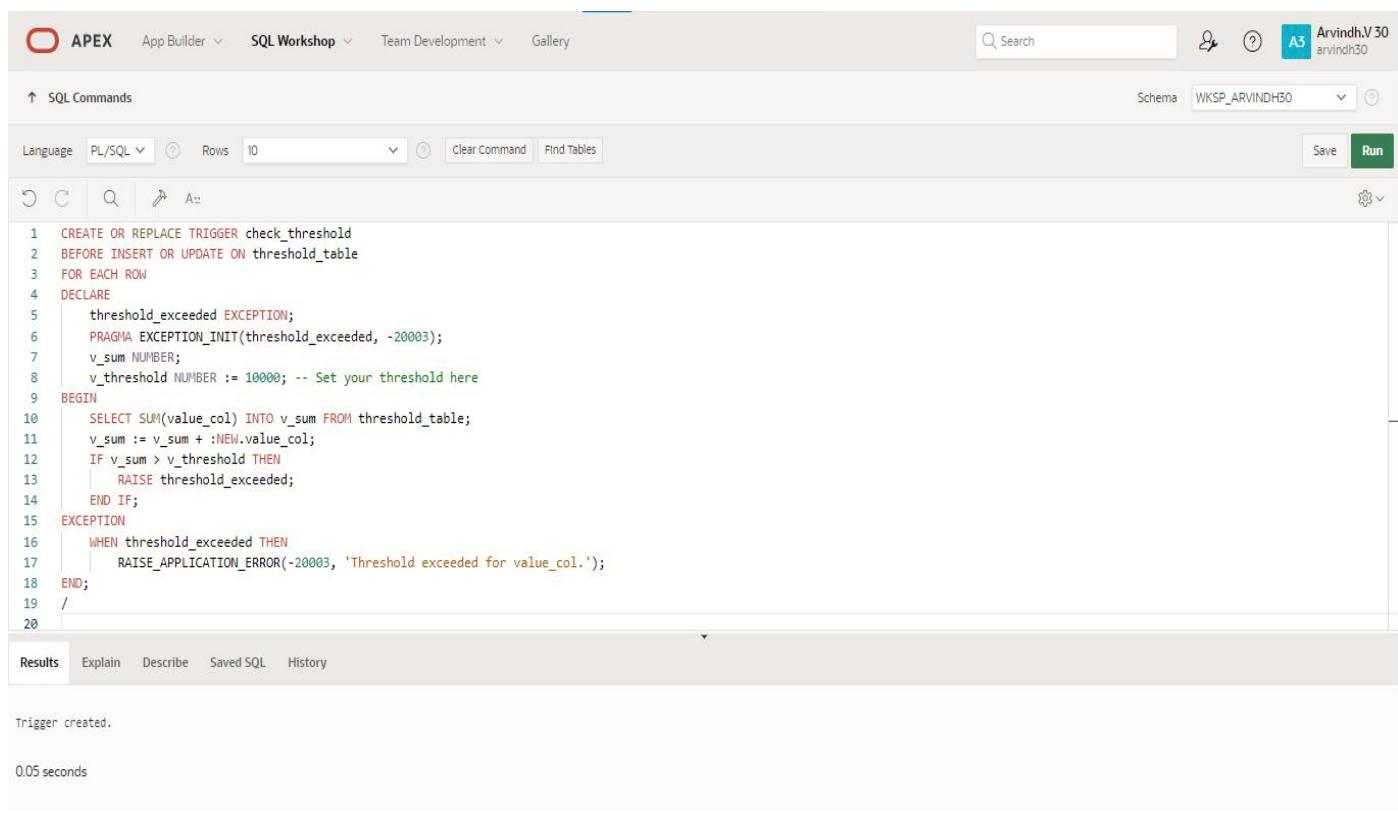
Trigger created.
0.03 seconds

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

QUERY:

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

OUTPUT:



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, and the schema name WKSP_ARVINDH30. The main workspace displays the PL/SQL code for the trigger. The code is numbered from 1 to 20. Lines 1 through 18 are the trigger definition, and lines 19 and 20 show the successful creation of the trigger. Below the code, the results tab shows the message "Trigger created." and a execution time of "0.05 seconds".

```
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;
19 /
20
```

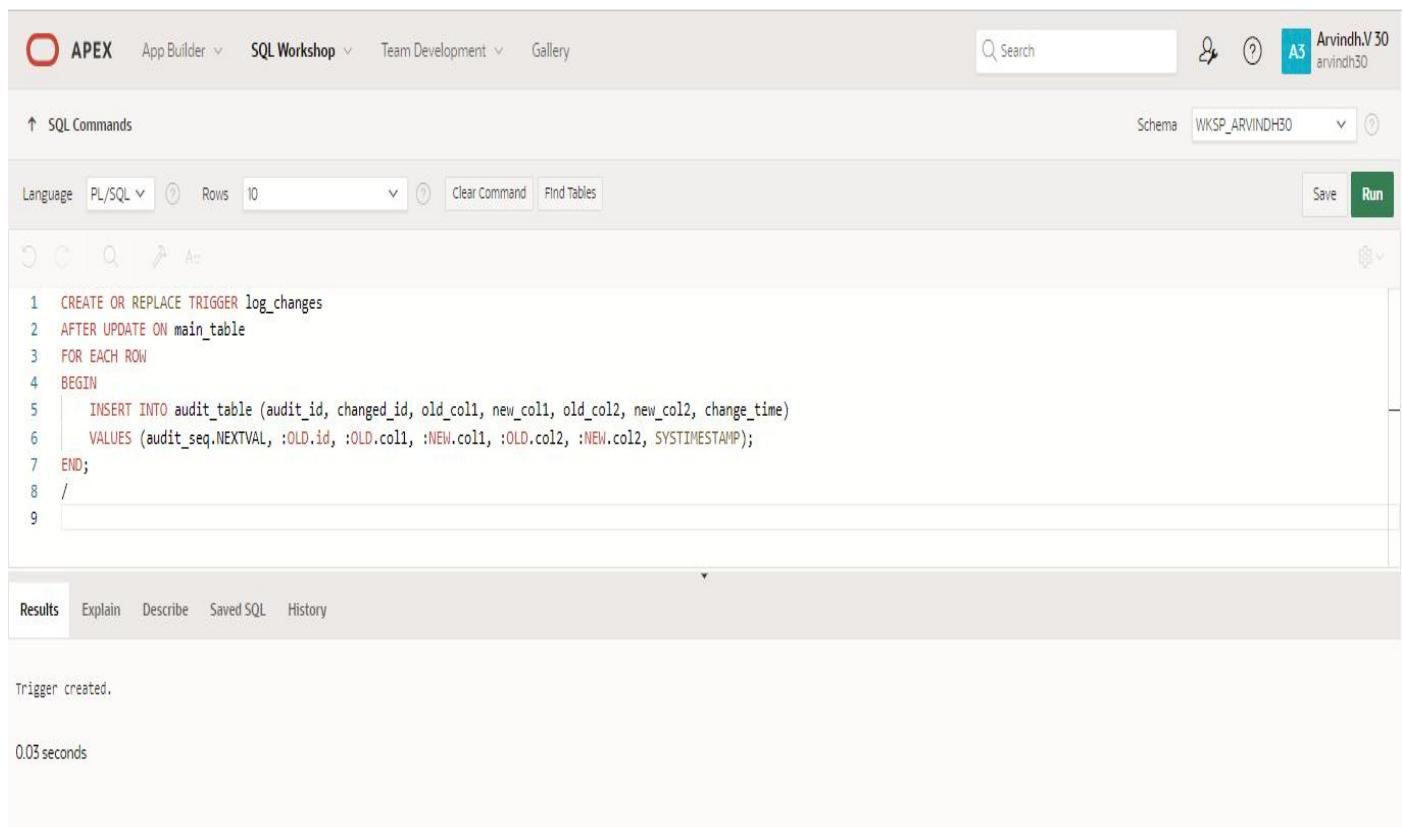
Trigger created.
0.05 seconds

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 SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays the PL/SQL code for the trigger 'log_changes'. The code is as follows:

```
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;
8 /
9 
```

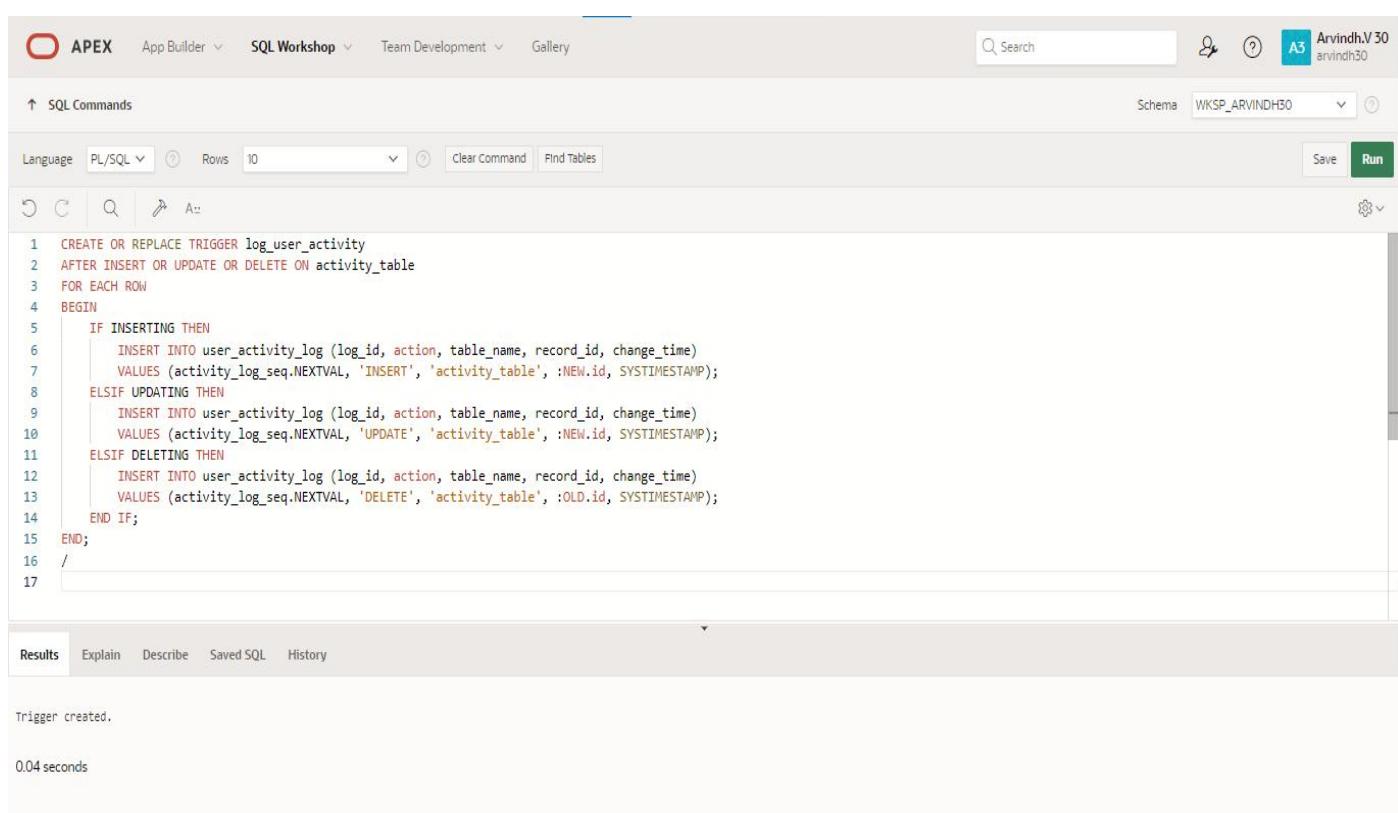
Below the code, the 'Results' tab is active, showing the message "Trigger created." and a execution time of "0.03 seconds".

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for Arvindh.V 30 (arvindh30). The main workspace displays the PL/SQL code for the trigger definition. The code is as follows:

```
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;
16 /
17
```

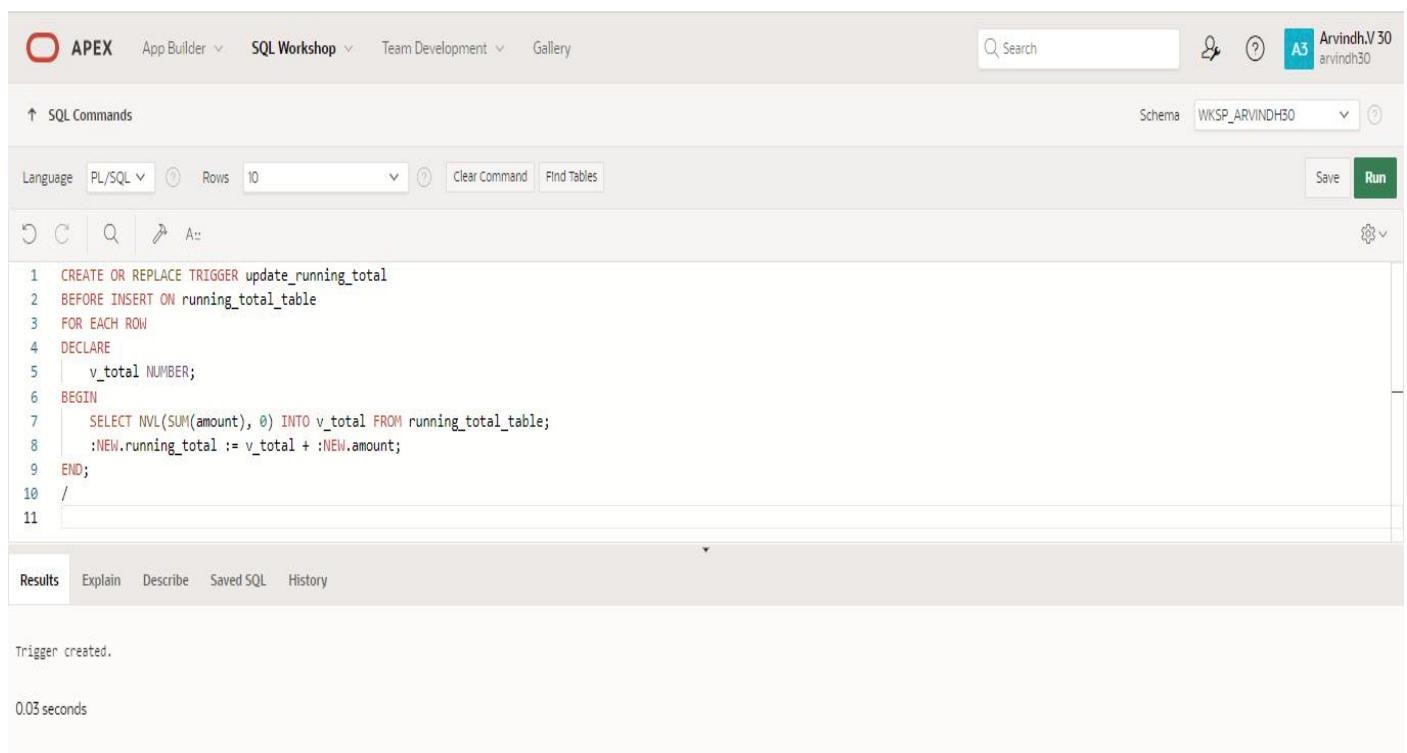
Below the code, the Results tab shows the output: "Trigger created." and "0.04 seconds".

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for Arvindh.V30, and a schema dropdown set to WKSP_ARVINDH30. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is numbered from 1 to 11. The 'Run' button is highlighted in green at the bottom right of the command input area. Below the workspace, tabs for Results, Explain, Describe, Saved SQL, and History are visible. The results section displays the message 'Trigger created.' and a execution time of '0.03 seconds'.

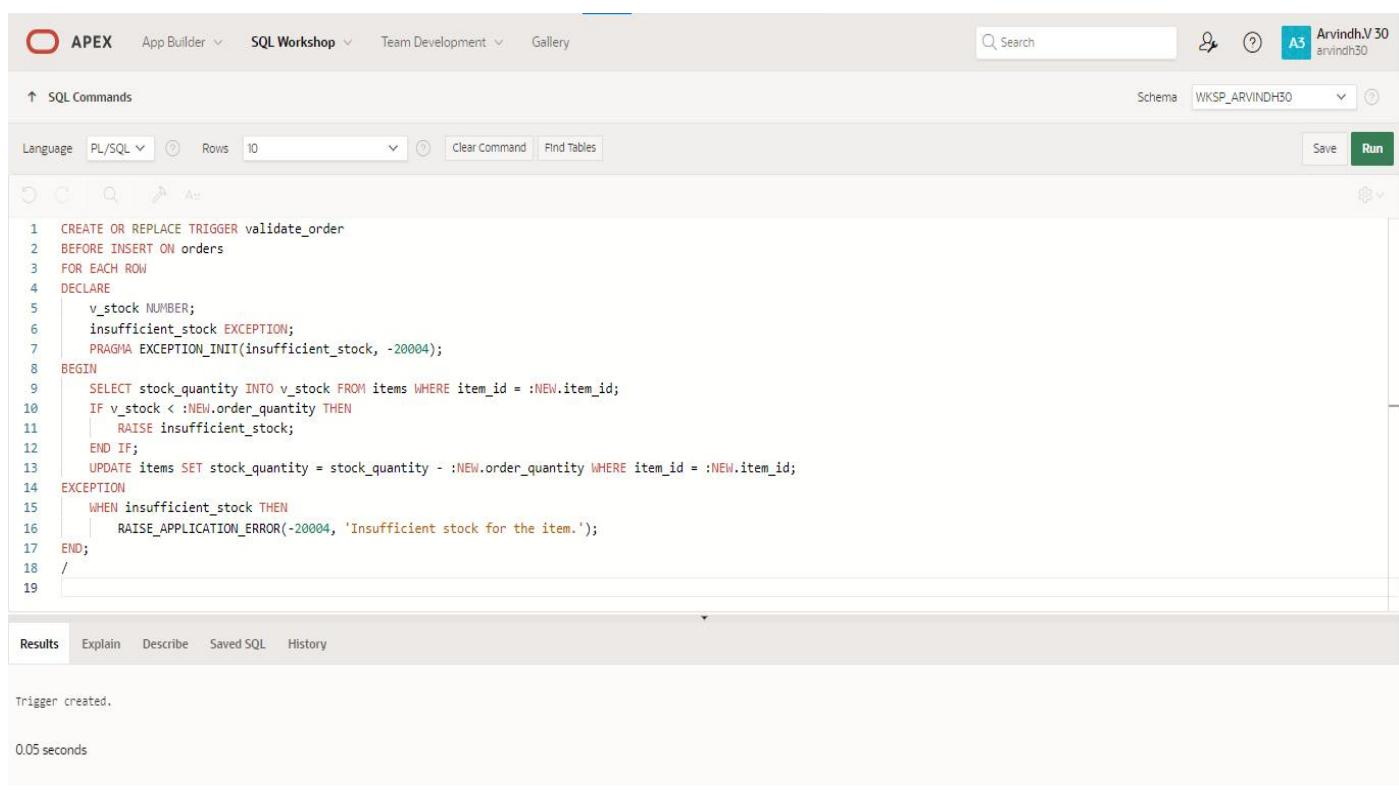
```
1 CREATE OR REPLACE TRIGGER update_running_total
2 BEFORE INSERT ON running_total_table
3 FOR EACH ROW
4 DECLARE
5     v_total NUMBER;
6 BEGIN
7     SELECT NVL(SUM(amount), 0) INTO v_total FROM running_total_table;
8     :NEW.running_total := v_total + :NEW.amount;
9 END;
10 /
11 
```

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 with the following details:

- Toolbar:** APEX, App Builder, SQL Workshop (selected), Team Development, Gallery.
- Header:** Search bar, Help icon, Session ID: Arvindh.V30 arvindh50.
- Schema:** WKSP_ARVINDH30.
- Language:** PL/SQL selected.
- Text Area:** The trigger code is pasted here, numbered from 1 to 19.
- Buttons:** Save, Run, Clear Command, Find Tables.
- Results Tab:** Shows the message "Trigger created." and a execution time of "0.05 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;
18 /
19 
```

Trigger created.
0.05 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 'Chinese' 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:

```
arvindh_30> db.restaurants.find({$and:[{ $or: [{cuisine: { $nin: ["American", "Chinese"] }}], {name: /Wil/}]}}}, {restaurant_id: 1, name: 1, borough: 1, cuisine: 1})
[
  {
    _id: ObjectId('66534fe9209d88ed59cdcf6'),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
arvindh_30>
```

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:

```
arvindh_30> db.restaurants.find( { grades: { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 } )
[
  {
    _id: ObjectId('66536722ff6e99f562cdcdf6'),
    restaurant_id: '123456',
    name: 'Restaurant A',
    grades: [
      {
        grade: 'A',
        score: 11,
        date: ISODate('2014-08-11T00:00:00.000Z')
      },
      {
        grade: 'B',
        score: 9,
        date: ISODate('2013-05-01T00:00:00.000Z')
      }
    ],
    {
      _id: ObjectId('66536722ff6e99f562cdcdf7'),
      restaurant_id: '789012',
      name: 'Restaurant B',
      grades: [
        {
          grade: 'A',
          score: 11,
          date: ISODate('2014-08-11T00:00:00.000Z')
        },
        {
          grade: 'C',
          score: 5,
          date: ISODate('2012-07-14T00:00:00.000Z')
        }
      ],
    }
]
```

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:

```
arvindh_30> 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 } );
[
  {
    _id: ObjectId('6653700f4c308e631cccdce0d'),
    restaurant_id: '101',
    name: 'Restaurant One',
    grades: [
      {
        grade: 'B',
        score: 8,
        date: ISODate('2013-05-01T00:00:00.000Z')
      },
      {
        grade: 'A',
        score: 9,
        date: ISODate('2014-08-11T00:00:00.000Z')
      }
    ]
  },
  {
    _id: ObjectId('6653700f4c308e631cccdce0e'),
    restaurant_id: '102',
    name: 'Restaurant Two',
    grades: [
      {
        grade: 'C',
        score: 7,
        date: ISODate('2012-07-14T00:00:00.000Z')
      },
      {
        grade: 'A',
        score: 9,
        date: ISODate('2014-08-11T00:00:00.000Z')
      }
    ]
  }
]
```

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:

```
arvindh_30> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1})
[
  {
    restaurant_id: '100001',
    name: 'Restaurant X',
    address: {
      street: '101 First Ave',
      city: 'City A',
      zipcode: '12345',
      coord: [ 40.7128, 43 ]
    }
  },
  {
    restaurant_id: '100002',
    name: 'Restaurant Y',
    address: {
      street: '202 Second St',
      city: 'City B',
      zipcode: '67890',
      coord: [ 34.0522, 50 ]
    }
  },
  {
    restaurant_id: '100003',
    name: 'Restaurant Z',
    address: {
      street: '303 Third Blvd',
      city: 'City C',
      zipcode: '13579',
      coord: [ 41.8781, 45.5 ]
    }
  }
]
arvindh_30>
```

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:

```
arvindh_30> db.restaurants.find({}, { _id: 0 }).sort({ name: 1 });
[
  {
    address: {
      building: '1007',
      coord: [ -73.856677, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
arvindh_30>
```

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:

```
] arvindh_30> db.restaurants.find({}, { _id: 0 }).sort({ name: -1 })
[
  {
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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:

```
] arvindh_30> db.restaurants.find({}, { _id: 0 }).sort({ cuisine: 1, borough: -1 })
[
  {
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
arvindh_30>
```

Ad
Go

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:

```
arvindh_30> db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })
[ {
  _id: ObjectId('66534fe9209d88ed59cdcdf6'),
  address: {
    building: '1007',
    coord: [ -73.856077, 40.848447 ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: ISODate('2014-03-03T00:00:00.000Z'),
      grade: 'A',
      score: 2
    },
    {
      date: ISODate('2013-09-11T00:00:00.000Z'),
      grade: 'A',
      score: 6
    },
    {
      date: ISODate('2013-01-24T00:00:00.000Z'),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate('2011-11-23T00:00:00.000Z'),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate('2011-03-10T00:00:00.000Z'),
      grade: 'B',
      score: 14
    }
  ],
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
}
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
[ {
  _id: ObjectId('66534fe9209d88ed59cdcdf6'),
  address: {
    building: '1007',
    coord: [ -73.856077, 40.848447 ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: ISODate('2014-03-03T00:00:00.000Z'),
      grade: 'A',
      score: 2
    },
    {
      date: ISODate('2013-09-11T00:00:00.000Z'),
      grade: 'A',
      score: 6
    },
    {
      date: ISODate('2013-01-24T00:00:00.000Z'),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate('2011-11-23T00:00:00.000Z'),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate('2011-03-10T00:00:00.000Z'),
      grade: 'B',
      score: 14
    }
  ],
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
}
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 });
[
  {
    _id: ObjectId('66534fe9209d88ed59cdcdf6'),
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
arvindh_30>
```

Activate Windows
Go to Settings to activate V

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:

```
arvindh_30> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
[
  {
    _id: ObjectId('66536a454c308e631cccdcf9'),
    name: 'Monsoon',
    borough: 'Manhattan',
    address: { coord: [ 40.7128, -74.006 ] },
    cuisine: 'Indian'
  },
  {
    _id: ObjectId('66536a454c308e631cccdcf9'),
    name: 'Harmony Cafe',
    borough: 'Brooklyn',
    address: { coord: [ 40.6782, -73.9442 ] },
    cuisine: 'Fusion'
  },
  {
    _id: ObjectId('66536a454c308e631cccdcfb'),
    name: 'Le Monde',
    borough: 'Manhattan',
    address: { coord: [ 40.8054, -73.9662 ] },
    cuisine: 'French'
  },
  {
    _id: ObjectId('66536a454c308e631cccdfdc'),
    name: 'Monarch',
    borough: 'Queens',
    address: { coord: [ 40.7282, -73.7949 ] },
    cuisine: 'American'
  }
]
arvindh_30>
```

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:

```
arvindh_30> db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
[
  {
    _id: ObjectId('66536c4c4c308e631cccdcdfd'),
    name: 'Madison Square Diner',
    borough: 'Manhattan',
    address: { coord: [ 40.7425, -73.9903 ] },
    cuisine: 'American'
  },
  {
    _id: ObjectId('66536c4c4c308e631cccdcdfe'),
    name: 'Mad for Chicken',
    borough: 'Brooklyn',
    address: { coord: [ 40.6872, -73.9677 ] },
    cuisine: 'Korean'
  },
  {
    _id: ObjectId('66536c4c4c308e631cccdcff'),
    name: 'Madiba Restaurant',
    borough: 'Brooklyn',
    address: { coord: [ 40.6839, -73.9653 ] },
    cuisine: 'South African'
  },
  {
    _id: ObjectId('66536c4c4c308e631cccdce0'),
    name: 'Madame Vo',
    borough: 'Manhattan',
    address: { coord: [ 40.7295, -73.988 ] },
    cuisine: 'Vietnamese'
  }
]
arvindh_30> -
```

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:

```
arvindh_30> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
[
  {
    _id: ObjectId('66534fe9209d88ed59cdcdfe'),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
[
  {
    _id: ObjectId('66536cbc4c308e631cccdce01'),
    restaurant_id: '101010',
    name: 'Low Score Diner',
    borough: 'Manhattan',
    address: {
      coord: [ 40.73061, -73.935242 ],
      street: '123 Example St',
      zipcode: '10001'
    },
    cuisine: 'American',
    grades: [
      {
        grade: 'B',
        score: 4,
        date: ISODate('2022-01-01T00:00:00.000Z')
      },
      {
        grade: 'A',
        score: 8,
        date: ISODate('2021-01-01T00:00:00.000Z')
      }
    ]
  },
  {
    _id: ObjectId('66536cbc4c308e631cccdce02'),
    restaurant_id: '202020',
    name: 'Tiny Scores Eatery',
    borough: 'Manhattan',
    address: {
      coord: [ 40.742054, -73.973968 ],
      street: '456 Sample Blvd',
      zipcode: '10002'
    },
    cuisine: 'Italian',
    grades: [
      {
        grade: 'C',
        score: 3,
        date: ISODate('2023-02-15T00:00:00.000Z')
      }
    ]
  }
]
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
[
  {
    _id: ObjectId('66536cbc4c308e631cccdce01'),
    restaurant_id: '101010',
    name: 'Low Score Diner',
    borough: 'Manhattan',
    address: {
      coord: [ 40.73061, -73.935242 ],
      street: '123 Example St',
      zipcode: '10001'
    },
    cuisine: 'American',
    grades: [
      {
        grade: 'B',
        score: 4,
        date: ISODate('2022-01-01T00:00:00.000Z')
      },
      {
        grade: 'A',
        score: 8,
        date: ISODate('2021-01-01T00:00:00.000Z')
      }
    ]
  },
  {
    _id: ObjectId('66536cbc4c308e631cccdce02'),
    restaurant_id: '202020',
    name: 'Tiny Scores Eatery',
    borough: 'Manhattan',
    address: {
      coord: [ 40.742054, -73.973968 ],
      street: '456 Sample Blvd',
      zipcode: '10002'
    },
    cuisine: 'Italian',
    grades: [
      {
        grade: 'C',
        score: 3,
        date: ISODate('2023-02-15T00:00:00.000Z')
      }
    ]
  }
]
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } }, { $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
[
  {
    _id: ObjectId('66536cbc4c308e631cccdce02'),
    restaurant_id: '202020',
    name: 'Tiny Scores Eatery',
    borough: 'Manhattan',
    address: {
      coord: [ 40.742054, -73.973968 ],
      street: '456 Sample Blvd',
      zipcode: '10002'
    },
    cuisine: 'Italian',
    grades: [
      {
        grade: 'C',
        score: 3,
        date: ISODate('2023-02-15T00:00:00.000Z')
      },
      {
        grade: 'B',
        score: 6,
        date: ISODate('2021-05-20T00:00:00.000Z')
      }
    ]
  },
  {
    _id: ObjectId('66536cbc4c308e631cccdce03'),
    restaurant_id: '303030',
    name: 'Minimal Marks Cafe',
    borough: 'Manhattan',
    address: {
      coord: [ 40.75917, -73.984703 ],
      street: '789 Demo Rd',
      zipcode: '10003'
    },
    cuisine: 'French',
    grades: [
      {
        grade: 'A',
        score: 2,
        date: ISODate('2024-03-10T00:00:00.000Z')
      }
    ]
  }
]
```

Activate Windows
Go to Settings to activate Windows.
Copilot (preview)

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:

```
arvindh_30> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } }, { $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
[
  {
    _id: ObjectId('66536cbc4c308e631cccdce02'),
    restaurant_id: '202020',
    name: 'Tiny Scores Eatery',
    borough: 'Manhattan',
    address: {
      coord: [ 40.742054, -73.973968 ],
      street: '456 Sample Blvd',
      zipcode: '10002'
    },
    cuisine: 'Italian',
    grades: [
      {
        grade: 'C',
        score: 3,
        date: ISODate('2023-02-15T00:00:00.000Z')
      },
      {
        grade: 'B',
        score: 6,
        date: ISODate('2021-05-20T00:00:00.000Z')
      }
    ]
  },
  {
    _id: ObjectId('66536cbc4c308e631cccdce03'),
    restaurant_id: '303030',
    name: 'Minimal Marks Cafe',
    borough: 'Manhattan',
    address: {
      coord: [ 40.75917, -73.984703 ],
      street: '789 Demo Rd',
      zipcode: '10003'
    },
    cuisine: 'French',
    grades: [
      {
        grade: 'A',
        score: 2,
        date: ISODate('2024-03-10T00:00:00.000Z')
      }
    ]
  }
]
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }]} )  
[  
  {  
    _id: ObjectId('66534fe9209d88ed59cdcdf6'),  
    address: {  
      building: '1007',  
      coord: [ -73.856077, 40.848447 ],  
      street: 'Morris Park Ave',  
      zipcode: '10462'  
    },  
    borough: 'Bronx',  
    cuisine: 'Bakery',  
    grades: [  
      {  
        date: ISODate('2014-03-03T00:00:00.000Z'),  
        grade: 'A',  
        score: 2  
      },  
      {  
        date: ISODate('2013-09-11T00:00:00.000Z'),  
        grade: 'A',  
        score: 6  
      },  
      {  
        date: ISODate('2013-01-24T00:00:00.000Z'),  
        grade: 'A',  
        score: 10  
      },  
      {  
        date: ISODate('2011-11-23T00:00:00.000Z'),  
        grade: 'A',  
        score: 9  
      },  
      {  
        date: ISODate('2011-03-10T00:00:00.000Z'),  
        grade: 'B',  
        score: 14  
      }  
    ],  
    name: 'Morris Park Bake Shop',  
    restaurant_id: '30075445'  
  }  
]  
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows

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:

```
arvindh_30> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })  
[  
  {  
    _id: ObjectId('66536e2b4c308e631cccdce04'),  
    restaurant_id: '404040',  
    name: 'Top Graded Diner',  
    borough: 'Manhattan',  
    address: {  
      coord: [ 40.741895, -73.989308 ],  
      street: '123 Main St',  
      zipcode: '10010'  
    },  
    cuisine: 'American',  
    grades: [  
      {  
        grade: 'A',  
        score: 2,  
        date: ISODate('2024-01-15T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 6,  
        date: ISODate('2023-05-10T00:00:00.000Z')  
      },  
      {  
        grade: 'B',  
        score: 10,  
        date: ISODate('2022-01-15T00:00:00.000Z')  
      }  
    ]  
  },  
  {  
    _id: ObjectId('66536e2b4c308e631cccdce05'),  
    restaurant_id: '505050',  
    name: 'High Scorers Cafe',  
    borough: 'Manhattan',  
    address: {  
      coord: [ 40.750742, -73.989698 ],  
      street: '456 Park Ave',  
      zipcode: '10011'  
    },  
    cuisine: 'French',  
    grades: [  
      {  
        grade: 'A',  
        score: 2,  
        date: ISODate('2024-01-15T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 6,  
        date: ISODate('2023-05-10T00:00:00.000Z')  
      },  
      {  
        grade: 'B',  
        score: 10,  
        date: ISODate('2022-01-15T00:00:00.000Z')  
      }  
    ]  
  }  
]  
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
[  
  {  
    _id: ObjectId('66536e2b4c308e631cccdce04'),  
    restaurant_id: '404040',  
    name: 'Top Graded Diner',  
    borough: 'Manhattan',  
    address: {  
      coord: [ 40.741895, -73.989308 ],  
      street: '123 Main St',  
      zipcode: '10010'  
    },  
    cuisine: 'American',  
    grades: [  
      {  
        grade: 'A',  
        score: 2,  
        date: ISODate('2024-01-15T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 6,  
        date: ISODate('2023-05-10T00:00:00.000Z')  
      },  
      {  
        grade: 'B',  
        score: 10,  
        date: ISODate('2022-01-15T00:00:00.000Z')  
      }  
    ],  
    _id: ObjectId('66536e2b4c308e631cccdce05'),  
    restaurant_id: '505050',  
    name: 'High Scorers Café',  
    borough: 'Manhattan',  
    address: {  
      coord: [ 40.750742, -73.989698 ],  
      street: '456 Park Ave',  
      zipcode: '10011'  
    },  
  },  
  {  
    _id: ObjectId('66536e2b4c308e631cccdce06'),  
    restaurant_id: '606060',  
    name: 'Excellent Eats',  
    borough: 'Manhattan',  
    address: {  
      coord: [ 40.732013, -73.996155 ],  
      street: '789 Broadway',  
      zipcode: '10012'  
    },  
    cuisine: 'Italian',  
    grades: [  
      {  
        grade: 'A',  
        score: 2,  
        date: ISODate('2024-02-20T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 6,  
        date: ISODate('2023-06-18T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 7,  
        date: ISODate('2022-03-15T00:00:00.000Z')  
      }  
    ]  
  }  
]
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
[  
  {  
    _id: ObjectId('66536e2b4c308e631cccdce05'),  
    restaurant_id: '505050',  
    name: 'High Scorers Café',  
    borough: 'Manhattan',  
    address: {  
      coord: [ 40.750742, -73.989698 ],  
      street: '456 Park Ave',  
      zipcode: '10011'  
    },  
    cuisine: 'French',  
    grades: [  
      {  
        grade: 'A',  
        score: 2,  
        date: ISODate('2024-02-20T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 6,  
        date: ISODate('2023-06-18T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 7,  
        date: ISODate('2022-03-15T00:00:00.000Z')  
      }  
    ]  
  },  
  {  
    _id: ObjectId('66536e2b4c308e631cccdce06'),  
    restaurant_id: '606060',  
    name: 'Excellent Eats',  
    borough: 'Manhattan',  
    address: {  
      coord: [ 40.732013, -73.996155 ],  
      street: '789 Broadway',  
      zipcode: '10012'  
    },  
    cuisine: 'Italian',  
    grades: [  
      {  
        grade: 'A',  
        score: 2,  
        date: ISODate('2024-02-20T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 6,  
        date: ISODate('2023-06-18T00:00:00.000Z')  
      },  
      {  
        grade: 'A',  
        score: 7,  
        date: ISODate('2022-03-15T00:00:00.000Z')  
      }  
    ]  
  }  
]
```

Activate Windows
Go to Settings to activate Windows.

No new notifications

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:

```
arvindh_30> 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"] } })
[
  {
    _id: ObjectId('66536e2b4c308e631cccdce05'),
    restaurant_id: '505050',
    name: 'High Scorers Cafe',
    borough: 'Manhattan',
    address: {
      coord: [ 40.750742, -73.989698 ],
      street: '456 Park Ave',
      zipcode: '10011'
    },
    cuisine: 'French',
    grades: [
      {
        grade: 'A',
        score: 2,
        date: ISODate('2024-02-20T00:00:00.000Z')
      },
      {
        grade: 'A',
        score: 6,
        date: ISODate('2023-06-18T00:00:00.000Z')
      },
      {
        grade: 'A',
        score: 7,
        date: ISODate('2022-03-15T00:00:00.000Z')
      }
    ],
    _id: ObjectId('66536e2b4c308e631cccdce06'),
    restaurant_id: '606060',
    name: 'Excellent Eats',
    borough: 'Manhattan',
    address: {
      coord: [ 40.732013, -73.996155 ],
      street: '789 Broadway',
      zipcode: '10012'
    },
    cuisine: 'Italian',
    grades: [
      ...
    ]
  }
]

```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
[
  {
    _id: ObjectId('66534fe9209d88ed59cdcdf6'),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
```

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:

```
arvindh_30> db.movies.find({ year: 1893 })
[
  {
    _id: ObjectId('665364cd209d88ed59cdce1e'),
    title: 'Movie from 1893',
    year: 1893,
    languages: [ 'English' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'Director One' ],
    writers: [ 'Writer One' ],
    countries: [ 'USA' ]
  },
  {
    _id: ObjectId('665364cd209d88ed59cdce1f'),
    title: 'Another 1893 Movie',
    year: 1893,
    languages: [ 'English' ],
    released: ISODate('1893-07-22T00:00:00.000Z'),
    directors: [ 'Director Two' ],
    writers: [ 'Writer Two' ],
    countries: [ 'France' ]
  }
]
arvindh_30>
```

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:

```
arvindh_30> db.movies.find({ runtime: { $gt: 120 } })
[
  {
    _id: ObjectId('665360bd209d88ed59cdce05'),
    title: 'Example Movie 3',
    languages: [ 'Spanish' ],
    released: ISODate('2010-11-10T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Four' ],
    awards: { wins: 5, nominations: 1, text: '5 wins & 1 nomination' },
    year: 2010,
    genres: [ 'Action', 'Thriller' ],
    runtime: 130,
    cast: [ 'Charles Kayser', 'Actor Four' ],
    countries: [ 'Mexico' ]
  },
  {
    _id: ObjectId('665364cd209d88ed59cdce20'),
    title: 'Long Movie 1',
    runtime: 150,
    languages: [ 'English' ],
    released: ISODate('2001-12-19T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Three' ],
    countries: [ 'USA' ]
  },
  {
    _id: ObjectId('665364cd209d88ed59cdce21'),
    title: 'Long Movie 2',
    runtime: 150,
    languages: [ 'French' ],
    released: ISODate('2005-07-22T00:00:00.000Z'),
    directors: [ 'Director Four' ],
    writers: [ 'Writer Four' ],
    countries: [ 'Canada' ]
  }
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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

QUERY:

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

OUTPUT:

```
arvindh_30> db.movies.find({ genres: 'Short' })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYYNzQ0MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.17700000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
arvindh_30>
```

Activate Windows

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:

```
arvindh_30> db.movies.find({ directors: 'William K.L. Dickson' })
[
  {
    _id: ObjectId('665364d2209d88ed59cdce22'),
    title: 'Dickson Movie 1',
    directors: [ 'William K.L. Dickson' ],
    languages: [ 'English' ],
    released: ISODate('1894-01-07T00:00:00.000Z'),
    writers: [ 'Writer Five' ],
    countries: [ 'USA' ]
  },
  {
    _id: ObjectId('665364d2209d88ed59cdce23'),
    title: 'Dickson Movie 2',
    directors: [ 'William K.L. Dickson' ],
    languages: [ 'French' ],
    released: ISODate('1895-03-22T00:00:00.000Z'),
    writers: [ 'Writer Six' ],
    countries: [ 'France' ]
  }
]
arvindh_30> s
```

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:

```
arvindh_30> db.movies.find({ countries: 'USA' })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYYNS00MDVmLWIwYjgtMmYvYWIXZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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

QUERY:

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

OUTPUT:

```
arvindh_30> db.movies.find({ rated: 'UNRATED' })
[
  {
    _id: ObjectId('6653639f209d88ed59cdce0f'),
    title: 'Unrated Movie 1',
    languages: [ 'English' ],
    released: ISODate('2001-12-19T00:00:00.000Z'),
    directors: [ 'Director One' ],
    writers: [ 'Writer One' ],
    countries: [ 'USA' ],
    rated: 'UNRATED'
  },
  {
    _id: ObjectId('6653639f209d88ed59cdce10'),
    title: 'Unrated Movie 2',
    languages: [ 'English', 'French' ],
    released: ISODate('2005-07-22T00:00:00.000Z'),
    directors: [ 'Director Two' ],
    writers: [ 'Writer Two', 'Writer Three' ],
    countries: [ 'France', 'Canada' ],
    rated: 'UNRATED'
  },
  {
    _id: ObjectId('6653639f209d88ed59cdce11'),
    title: 'Unrated Movie 3',
    languages: [ 'Spanish' ],
    released: ISODate('2010-11-10T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Four' ],
    countries: [ 'Mexico' ],
    rated: 'UNRATED'
  }
]
arvindh_30>
```

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:

```
arvindh_30> db.movies.find({ 'imdb.votes': { $gt: 1000 } })
[ {
  _id: ObjectId('573a1390f29313caabcd42e8'),
  plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
  genres: [ 'Short', 'Western' ],
  runtime: 11,
  cast: [
    'A.C. Abadie',
    "Gilbert M. 'Broncho Billy' Anderson",
    'George Barnes',
    'Justus D. Barnes'
  ],
  poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYYNS00MDVmLWIwYjgtMmYwYWIxZDYYNzU2XkEyXkFqcGdeQXVyNzQzNzQzI@._V1_SY1000_SX677_AL_.jpg',
  title: 'The Great Train Robbery',
  fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
  languages: [ 'English' ],
  released: ISODate('1903-12-01T00:00:00.000Z'),
  directors: [ 'Edwin S. Porter' ],
  rated: 'TV-G',
  awards: { wins: 1, nominations: 0, text: '1 win.' },
  lastupdated: '2015-08-13 00:27:59.177000000',
  year: 1903,
  imdb: { rating: 7.4, votes: 9847, id: 439 },
  countries: [ 'USA' ],
  type: 'movie',
  tomatoes: {
    viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
    fresh: 6,
    critic: { rating: 7.6, numReviews: 6, meter: 100 },
    rotten: 0,
    lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
  }
}
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.movies.find({ 'imdb.rating': { $gt: 7 } })
[ {
  _id: ObjectId('573a1390f29313caabcd42e8'),
  plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
  genres: [ 'Short', 'Western' ],
  runtime: 11,
  cast: [
    'A.C. Abadie',
    "Gilbert M. 'Broncho Billy' Anderson",
    'George Barnes',
    'Justus D. Barnes'
  ],
  poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYYNS00MDVmLWIwYjgtMmYwYWIxZDYYNzU2XkEyXkFqcGdeQXVyNzQzNzQzI@._V1_SY1000_SX677_AL_.jpg',
  title: 'The Great Train Robbery',
  fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
  languages: [ 'English' ],
  released: ISODate('1903-12-01T00:00:00.000Z'),
  directors: [ 'Edwin S. Porter' ],
  rated: 'TV-G',
  awards: { wins: 1, nominations: 0, text: '1 win.' },
  lastupdated: '2015-08-13 00:27:59.177000000',
  year: 1903,
  imdb: { rating: 7.4, votes: 9847, id: 439 },
  countries: [ 'USA' ],
  type: 'movie',
  tomatoes: {
    viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
    fresh: 6,
    critic: { rating: 7.6, numReviews: 6, meter: 100 },
    rotten: 0,
    lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
  }
}
]
arvindh_30>
```

Activate Windows
Go to Settings to activate Windows.

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:

```
arvindh_30> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
[
  {
    _id: ObjectId('665362b4209d88ed59cdce0c'),
    title: 'High Rated Movie 1',
    languages: [ 'English' ],
    released: ISODate('2001-12-19T00:00:00.000Z'),
    directors: [ 'Director One' ],
    writers: [ 'Writer One' ],
    countries: [ 'USA' ],
    tomatoes: { viewer: { rating: 4.5, numReviews: 120, meter: 85 } }
  },
  {
    _id: ObjectId('665362b4209d88ed59cdce0d'),
    title: 'High Rated Movie 2',
    languages: [ 'English', 'French' ],
    released: ISODate('2005-07-22T00:00:00.000Z'),
    directors: [ 'Director Two' ],
    writers: [ 'Writer Two', 'Writer Three' ],
    countries: [ 'France', 'Canada' ],
    tomatoes: { viewer: { rating: 4.8, numReviews: 250, meter: 90 } }
  },
  {
    _id: ObjectId('665362b4209d88ed59cdce0e'),
    title: 'High Rated Movie 3',
    languages: [ 'Spanish' ],
    released: ISODate('2010-11-10T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Four' ],
    countries: [ 'Mexico' ],
    tomatoes: { viewer: { rating: 4.7, numReviews: 180, meter: 88 } }
  }
]
arvindh_30> -
```

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

QUERY:

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

OUTPUT:

```
arvindh_30> db.movies.find({ 'awards.wins': { $gt: 0 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      'Gilbert M. 'Broncho Billy' Anderson',
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MVSBMTU3NjE5NzYtYTYYNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2550, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
arvindh_30>
```

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:

```
arvindh_30> 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('665360bd209d88ed59cdce03'),
    title: 'Example Movie 1',
    languages: [ 'English' ],
    released: ISODate('2001-12-19T00:00:00.000Z'),
    directors: [ 'Director One' ],
    writers: [ 'Writer One' ],
    awards: { wins: 3, nominations: 2, text: '3 wins & 2 nominations' },
    year: 2001,
    genres: [ 'Drama' ],
    runtime: 120,
    cast: [ 'Charles Kayser', 'Actor Two' ],
    countries: [ 'USA' ]
  },
  {
    _id: ObjectId('665360bd209d88ed59cdce04'),
    title: 'Example Movie 2',
    languages: [ 'English', 'French' ],
    released: ISODate('2005-07-22T00:00:00.000Z'),
    directors: [ 'Director Two' ],
    writers: [ 'Writer Two', 'Writer Three' ],
    awards: { wins: 1, nominations: 4, text: '1 win & 4 nominations' },
    year: 2005,
    genres: [ 'Comedy', 'Romance' ],
    runtime: 95,
    cast: [ 'Charles Kayser', 'Actor Three' ],
    countries: [ 'France', 'Canada' ]
  },
  {
    _id: ObjectId('665360bd209d88ed59cdce05'),
    title: 'Example Movie 3',
    languages: [ 'Spanish' ],
    released: ISODate('2010-11-10T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Four' ],
    awards: { wins: 5, nominations: 1, text: '5 wins & 1 nomination' },
    year: 2010,
    genres: [ 'Action', 'Thriller' ],
    runtime: 130,
    cast: [ 'Charles Kayser', 'Actor Four' ],
    countries: [ 'Mexico' ]
  }
]

Activate Windows
Go to Settings to activate Windows.
```

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:

```
arvindh_30> 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('665360bd209d88ed59cdce03'),
    title: 'Example Movie 1',
    languages: [ 'English' ],
    released: ISODate('2001-12-19T00:00:00.000Z'),
    directors: [ 'Director One' ],
    writers: [ 'Writer One' ],
    awards: { wins: 3, nominations: 2, text: '3 wins & 2 nominations' },
    year: 2001,
    genres: [ 'Drama' ],
    runtime: 120,
    cast: [ 'Charles Kayser', 'Actor Two' ],
    countries: [ 'USA' ]
  },
  {
    _id: ObjectId('665360bd209d88ed59cdce04'),
    title: 'Example Movie 2',
    languages: [ 'English', 'French' ],
    released: ISODate('2005-07-22T00:00:00.000Z'),
    directors: [ 'Director Two' ],
    writers: [ 'Writer Two', 'Writer Three' ],
    awards: { wins: 1, nominations: 4, text: '1 win & 4 nominations' },
    year: 2005,
    genres: [ 'Comedy', 'Romance' ],
    runtime: 95,
    cast: [ 'Charles Kayser', 'Actor Three' ],
    countries: [ 'France', 'Canada' ]
  },
  {
    _id: ObjectId('665360bd209d88ed59cdce05'),
    title: 'Example Movie 3',
    languages: [ 'Spanish' ],
    released: ISODate('2010-11-10T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Four' ],
    awards: { wins: 5, nominations: 1, text: '5 wins & 1 nomination' },
    year: 2010,
    genres: [ 'Action', 'Thriller' ],
    runtime: 130,
    cast: [ 'Charles Kayser', 'Actor Four' ],
    countries: [ 'Mexico' ]
  }
]

Activate Windows
Go to Settings to activate Windows.
```

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:

```
arvindh_30> 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('6653615a209d88ed59cdce06'),
    title: 'Example Movie 1',
    languages: [ 'English' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'Director One' ],
    writers: [ 'Writer One' ],
    countries: [ 'USA' ]
  },
  {
    _id: ObjectId('6653615a209d88ed59cdce07'),
    title: 'Example Movie 2',
    languages: [ 'French' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'Director Two' ],
    writers: [ 'Writer Two' ],
    countries: [ 'France' ]
  },
  {
    _id: ObjectId('6653615a209d88ed59cdce08'),
    title: 'Example Movie 3',
    languages: [ 'Spanish' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Three' ],
    countries: [ 'Spain' ]
  }
]
arvindh_30> -
```

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:

```
arvindh_30> db.movies.find( { title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )
[
  {
    _id: ObjectId('665361e6209d88ed59cdce09'),
    title: 'The Scene of the Crime',
    languages: [ 'English' ],
    released: ISODate('2001-12-19T00:00:00.000Z'),
    directors: [ 'Director One' ],
    writers: [ 'Writer One' ],
    countries: [ 'USA' ]
  },
  {
    _id: ObjectId('665361e6209d88ed59cdce0a'),
    title: 'Behind the Scenes',
    languages: [ 'English', 'French' ],
    released: ISODate('2005-07-22T00:00:00.000Z'),
    directors: [ 'Director Two' ],
    writers: [ 'Writer Two', 'Writer Three' ],
    countries: [ 'France', 'Canada' ]
  },
  {
    _id: ObjectId('665361e6209d88ed59cdce0b'),
    title: 'Scene Stealer',
    languages: [ 'Spanish' ],
    released: ISODate('2010-11-10T00:00:00.000Z'),
    directors: [ 'Director Three' ],
    writers: [ 'Writer Four' ],
    countries: [ 'Mexico' ]
  }
]
arvindh_30> -
```

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

RESULT: