

Creating and Managing Tables

EX_NO:1

DATE:

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

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

QUERY:

```
Create table DEPT(dept_id number(7),name varchar2(25));
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands section, the following command is entered and executed:

```
create table DEPT (dept_id number(7) not null, Name varchar2(20) );
```

The Results tab shows the output:

```
Table created.  
0.04 seconds
```

At the bottom, the footer includes:

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

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

QUERY:

```
Create table EMP(dept_id number(7) not null,Last_Name varchar2(25),First_Name varchar2(25),Dept_id number(7));
```

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 signed in as 'Dhanya Vanisha' with the session ID 'euphoria'. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_EUPHORIA'. The main area displays the SQL command for creating the EMP table:

```
1 Create table EMP(ID number(7) not null,Last_Name varchar2(25),First_Name varchar2(25),Dept_id number(7));
2
```

The 'Results' tab is active, showing the output: 'Table created.' and '0.03 seconds'. The bottom footer includes the user's email '220701062@rajalakshmi.edu.in', the application name 'euphoria', the language 'en', copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile for 'Dhanya Vanisha euphoria', and a schema dropdown set to 'WKSP_EUPHORIA'. Below the tabs, there's a toolbar with icons for Undo, Redo, Find, Replace, and others. The main area is titled 'SQL Commands' and contains a text input field with the SQL command: '1 Alter table EMP modify(Last_Name varchar2(50));'. Below the input field, there are buttons for 'Save' and 'Run'. The results section is active, showing the output of the command: 'Table altered.' and '0.04 seconds'. The bottom of the screen displays copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

```
Create table EMPLOYEES2(emp_id number(7),first_name varchar2(25),Last_name varchar2(25),Salary int(8,2),Dept_id number(7));
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, user profile information for 'Dhanya Vanisha euphoria', and various icons. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), and Schema (set to WKSP_EUPHORIA). There are buttons for Save and Run. Below the tabs, there are icons for Undo, Redo, Find, and Paste. The SQL command entered is:

```
1 Create table EMPLOYEES2(emp_id number(7),first_name varchar2(25),Last_name varchar2(25),Salary number(8,2),Dept_id number(7));  
2
```

Below the command, the 'Results' tab is selected, showing the output: 'Table created.' and '0.04 seconds'. The bottom of the screen displays copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

5.Drop the EMP table.

QUERY:

Drop table EMP;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, a user profile for 'Dhanya Vanisha' (euphoria) is displayed. The main area is titled 'SQL Commands'. The 'Language' dropdown is set to 'SQL'. Below it, there are buttons for 'Clear Command' and 'Find Tables'. The results pane shows the output of the executed SQL command:

```
1 Drop table EMP;
2 |
```

The 'Results' tab is selected, showing the message 'Table dropped.' and a time of '0.09 seconds'. The bottom of the page includes copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

6.Rename the EMPLOYEES2 table as EMP. QUERY:

Rename EMPLOYEES2 to EMP;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, user profile information for 'Dhanya Vanisha euphoria', and a 'Run' button. The main area is titled 'SQL Commands' and contains a code editor with the following content:

```
1 Rename EMPLOYEES2 to EMP;
2
```

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

Statement processed.
0.05 seconds

At the bottom, there are footer links for user profile, language, and help, along with copyright and version information: Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4.

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

QUERY:

comment on table DEPT is 'Department info'; comment
on table EMP is 'Employee info';

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, user profile information for 'Dhanya Vanisha euphoria', and a 'Run' button. The main area is titled 'SQL Commands'. It includes a toolbar with icons for Undo, Redo, Search, Find Tables, and Run. Below the toolbar, there are dropdown menus for Language (SQL), Rows (10), and a schema dropdown set to 'WKSP_EUPHORIA'. The SQL editor contains the following code:

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

Below the editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. The output section shows the message 'Statement processed.' and a time of '0.01 seconds'. At the bottom, there are footer links for user email (220701062@rajalakshmi.edu.in), workspace (euphoria), and language (en). The copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also present.

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

QUERY:

```
Alter table EMP drop column first_name;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Schema WKSP_EUPHORIA Save Run

SQL Commands:

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

Results Explain Describe Saved SQL History

Table altered.
0.06 seconds

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

RESULT:

MANIPULATING DATA

EX_NO:2

DATE:

1.Create MY_EMPLOYEE table with the following structure

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

QUERY:

CREATE TABLE MY_EMPLOYEE (id number(4),Last_name varchar(25),First_name varchar(25),Userid varchar(25),Salary number(9,2);

OUTPUT

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Dhanya Vanisha' and a 'Search' bar. The main workspace is titled 'SQL Commands'. It features a toolbar with icons for Undo, Redo, Find, Replace, and Save/Run. Below the toolbar, the schema is set to 'WKSP_EUPHORIA'. The SQL editor contains the command to create the 'MY_EMPLOYEE' table. The results tab at the bottom shows the output: 'Table created.' and '0.03 seconds'. The footer includes copyright information for Oracle and the APEX version.

```
1 CREATE TABLE MY_EMPLOYEE (id number(4),Last_name varchar(25),First_name varchar(25),Userid varchar(25),Salary number(9,2));
```

Results Explain Describe Saved SQL History

Table created.
0.03 seconds

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

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

QUERY:

```
insert into MY_EMPLOYEE values(1,'Patel','Ralph','rpatel',895); insert  
into MY_EMPLOYEE values(2,'Danes','Betty','bdancs',860);
```

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 SQL Workshop tab is selected. The schema dropdown is set to WKSP_EUPHORIA. The main area contains the following SQL code:

```
1  insert into MY_EMPLOYEE values(1,'Patel','Ralph','rpatel',895);  
2 |
```

Below the code, the results section shows the output:

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

Execution time is listed as 0.04 seconds. The bottom footer includes user information (220701062@rajalakshmi.edu.in), session details (euphoria, en), copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the APEX version (Oracle APEX 23.2.4).

3.Display the table with values.

QUERY:

```
select * from MY_EMPLOYEE;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. On the right, there's a search bar, user profile information for 'Dhanya Vanisha euphoria', and a 'Run' button. The main area is titled 'SQL Commands' with a dropdown for 'Language' set to 'SQL'. Below this, there are buttons for Undo, Redo, Find, Replace, and a refresh icon. The SQL command entered is 'select * from MY_EMPLOYEE;'. The results tab is selected, showing a table with columns ID, LAST_NAME, FIRST_NAME, USERID, and SALARY. The data returned is:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Danes	Betty	bdanes	860

Below the table, it says '2 rows returned in 0.03 seconds' and has a 'Download' link. At the bottom, there are footer links for user email, site name, language, copyright notice, and Oracle APEX version.

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

QUERY:

```
insert into MY_EMPLOYEE values (3,'Biri','Ben','bbiri',1100); insert  
into MY_EMPLOYEE values (4,'Newman','Chad','Cnewman',750);
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands Schema WKSP_EUPHORIA

Language SQL Rows 10 Clear Command Find Tables Save Run

SQL Commands

```
1 insert into MY_EMPLOYEE values (4,'Newman','Chad','Cnewman',750);
```

Results Explain Describe Saved SQL History

2 weeks ago	update MY_EMPLOYEE set Salary=1000 where Salary<=900; select * from MY_EMPLOYEE;	WKSP_EUPHORIA
2 weeks ago	update MY_EMPLOYEE set Salary=1000 where Salary<=900;	WKSP_EUPHORIA
2 weeks ago	update MY_EMPLOYEE set Last_Name='Drexler' where ID=3;	WKSP_EUPHORIA
2 weeks ago	update table MY_EMPLOYEE set Last_Name='Drexler' where ID=3;	WKSP_EUPHORIA
2 weeks ago	select * from MY_EMPLOYEE;	WKSP_EUPHORIA
2 weeks ago	insert into MY_EMPLOYEE values (4,'Newman','Chad','Cnewman',750);	WKSP_EUPHORIA

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

5. Make the data additions permanent.

QUERY:

select * from MY_EMPLOYEE;

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands Schema WKSP_EUPHORIA

Language SQL Rows 10 Clear Command Find Tables Save Run

SQL Commands

```
1 select * from MY_EMPLOYEE;
```

Results Explain Describe Saved SQL History

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
3	Biri	Ben	bbiri	1100
2	Danes	Betty	bdanes	860

3 rows returned in 0.00 seconds Download

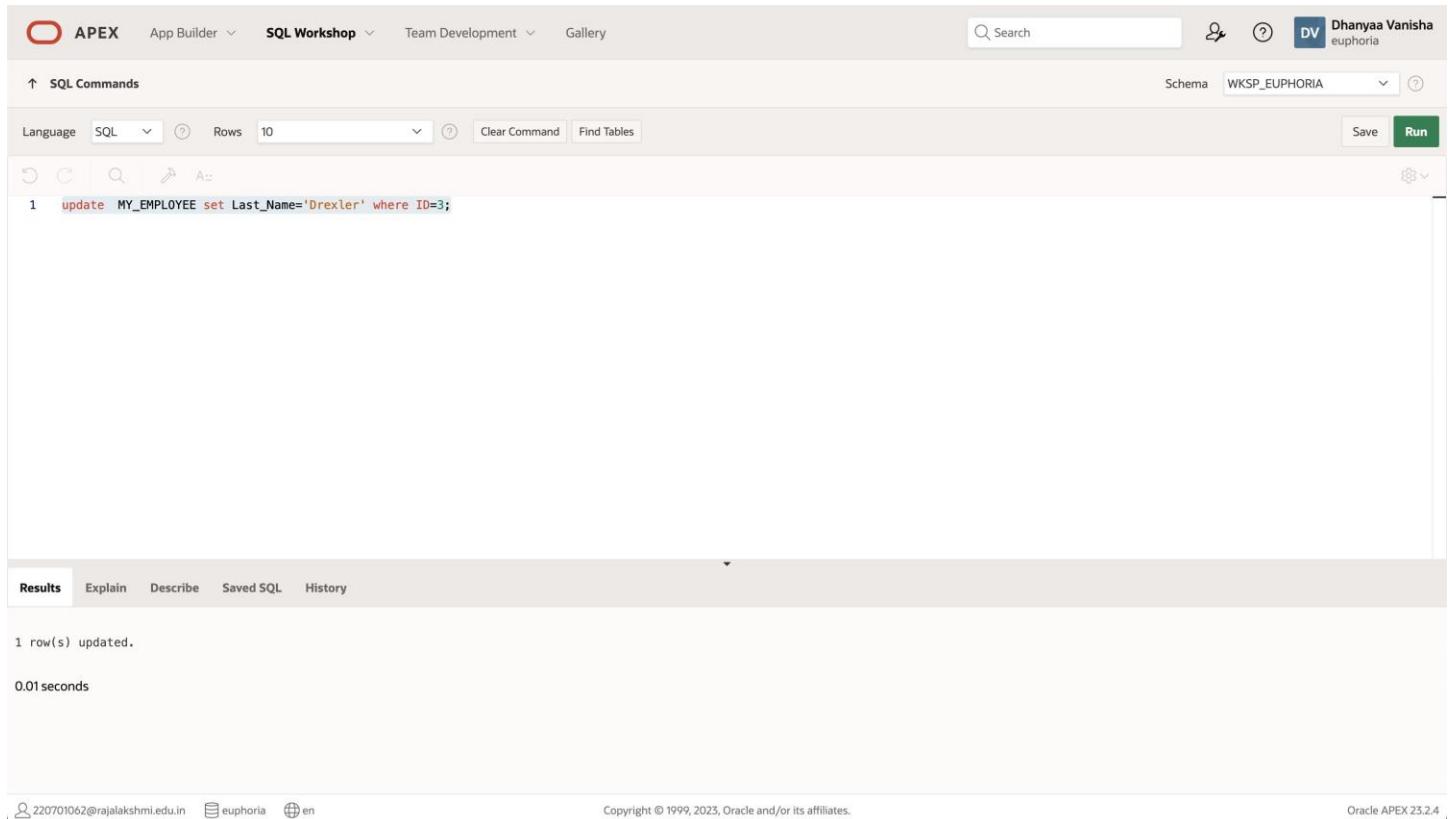
javascript:sc_postHistory('69277145321096050100','WKSP_EUPHORIA');

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

OUTPUT:

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

QUERY: update MY_EMPLOYEE set Last_Name='Drexler'
where ID=3; 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's a search bar, user profile for 'Dhanya Vanisha euphoria', and a schema dropdown set to 'WKSP_EUPHORIA'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code: 'update MY_EMPLOYEE set Last_Name='Drexler' where ID=3;'. Below the code, the results tab is selected, showing the output: '1 row(s) updated.' and '0.01 seconds'. The bottom footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

```
update MY_EMPLOYEE set Last_Name='Drexler' where ID=3;
```

Results Explain Describe Saved SQL History

1 row(s) updated.
0.01 seconds

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

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

QUERY:

```
update MY_EMPLOYEE set Salary=1000 where Salary<=900;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Search

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Schema WKSP_EUPHORIA Save Run

1 update MY_EMPLOYEE set Salary=1000 where Salary<=900;

Results Explain Describe Saved SQL History

2 row(s) updated.
0.01 seconds

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile information for 'Dhanya Vanisha euphoria', and a 'Run' button. The main area is titled 'SQL Commands'. It has dropdowns for 'Language' (set to SQL), 'Rows' (set to 10), and buttons for 'Clear Command' and 'Find Tables'. Below this is a code editor with a single line of SQL: 'update MY_EMPLOYEE set Salary=1000 where Salary<=900;'. To the left of the code editor are icons for refresh, undo, redo, search, and a dropdown menu. To the right are buttons for 'Save' and 'Run'. Below the code editor is a results panel with tabs for 'Results' (which is selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. The results section displays the output of the executed query: '2 row(s) updated.' and '0.01 seconds'. At the bottom of the page, there are footer links for user ID, session name, language, and copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4'.

8.Delete Betty dancs from MY_EMPLOYEE table.

QUERY: delete from MY_EMPLOYEE where

First_Name='Betty'; OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, a user profile for 'Dhanya Varsha euphoria' is displayed. The main area is titled 'SQL Commands'. It includes a search bar, schema selection ('WKSP_EUPHORIA'), and buttons for Save and Run. Below this, there are icons for Undo, Redo, Find, and Copy. The SQL editor contains the following command:

```
1 Delete from MY_EMPLOYEE where First_Name='Betty';
```

Below the editor, a results panel shows the output of the query:

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

Execution time is listed as 0.01 seconds. The bottom of the page includes footer information: a user icon with '220701062@rajalakshmi.edu.in', a location icon with 'euphoria', a language icon with 'en', copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the text 'Oracle APEX 23.2.4'.

9. Empty the fourth row of the emp table.

QUERY:

delete from MY_EMPLOYEE where ID=4; OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, user profile information for 'Dhanya Vanisha euphoria', and a schema dropdown set to 'WKSP_EUPHORIA'. Below the header, the SQL Commands tab is selected. The SQL editor shows the following command:

```
1 delete from MY_EMPLOYEE where ID=4;
```

Below the editor, the Results tab is selected. The output shows:

```
0 row(s) deleted.
```

Execution time: 0.01 seconds

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

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

RESULT:

INCLUDING CONSTRAINTS

EX_NO:3

DATE:

1.Add a table-level PRIMARY KEY constraint to the EMP table on the ID column.The constraint should be named at creation. Name the constraint my_emp_id_pk.

QUERY:

```
create table EMP(id number(6),last_name varchar2(25),email varchar2(25),salary
number(8,2),constraint my_emp_id_pk primary key(id));
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, user profile for 'Dhanya Vanisha euphoria', and a schema dropdown set to 'WKSP_EUPHORIA'. Below the header, the main area is titled 'SQL Commands' with a sub-header 'Language: SQL'. It shows a single line of SQL code: 'create table EMP(id number(6),last_name varchar2(25),email varchar2(25),salary number(8,2),constraint my_emp_id_pk primary key(id));'. Below the code, there are several icons for operations like Run, Save, and Undo. At the bottom of the SQL area, there are tabs for Results, Explain, Describe, Saved SQL, and History. The 'Results' tab is selected. The results section displays the message 'Table created.' and '0.05 seconds' execution time. The footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

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:

create table DEPT(id number(6),last_name varchar2(25) not null,first_name varchar2(25) not null,email varchar2(25),constraint my_dept_id_pk primary key(id)); OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, user profile for 'Dhanya Vanisha euphoria', and a schema dropdown set to 'WKSP_EUPHORIA'. Below the header, the 'SQL Commands' tab is selected. The main area contains a code editor with the following SQL command:

```
1 create table DEPT(id number(6),last_name varchar2(25) not null,first_name varchar2(25) not null,email varchar2(25),constraint my_dept_id_pk primary key(id));
```

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active and displays the output of the query: "Table created." It also shows a execution time of "0.05 seconds". At the bottom of the page, there are footer links for user information (220701062@rajalakshmi.edu.in, euphoria, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

3.Add a column DEPT_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to nonexistent department. Name the constraint my_emp_dept_id_fk.

QUERY:

```
alter table EMP add dept_id number(5);
alter table EMP add constraint my_emp_dept_id_fk foreign key(dept_id) references EMP(id);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, user profile for 'Dhanya Vanisha euphoria', and a schema dropdown set to 'WKSP_EUPHORIA'. Below the header, the 'SQL Commands' tab is selected. The main area contains a code editor with the following SQL command:

```
1 alter table EMP add constraint my_emp_dept_id_fk foreign key(dept_id) references EMP(id);
```

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

Table altered.
0.06 seconds

At the bottom of the page, there are footer links for user information (220701062@rajalakshmi.edu.in, euphoria, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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

QUERY:

```
alter table EMP add commision number(5) check (commision>0);
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Search DV Dhanya Vanisha euphoria

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Schema WKSP_EUPHORIA Save Run

1 alter table EMP add commision number(5) check (commision>0);

Results Explain Describe Saved SQL History

Table altered.
0.06 seconds

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

RESULT:

Writing Basic SQL SELECT Statements

EX_NO:4

DATE:

-
- 1.The following statement executes successfully.

Identify the Errors

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

QUERY:

```
Select employee_id, last_name, sal*12 as ANNUALSALARY from EMPLOYEES;
```

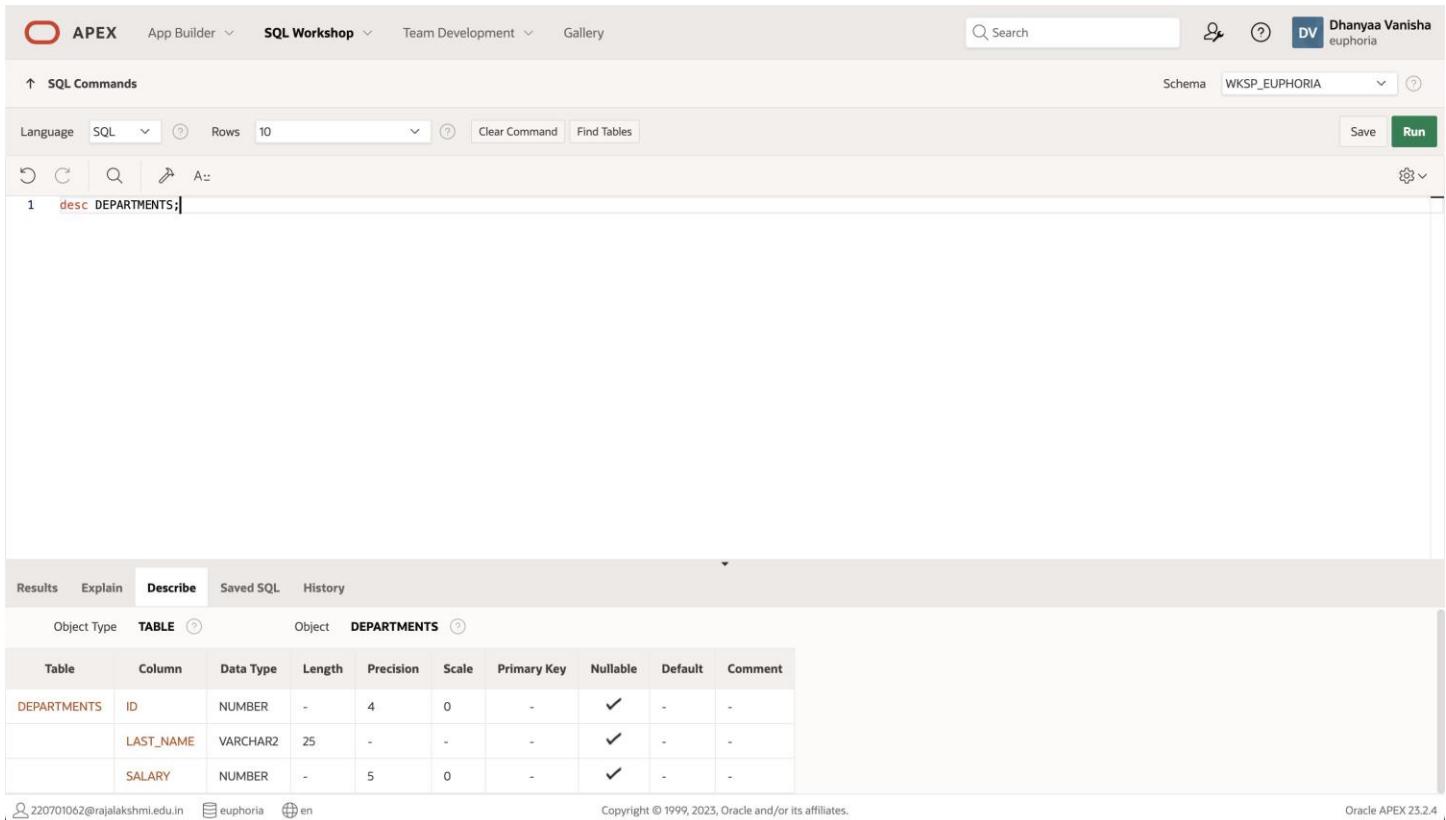
2. Show the structure of departments the table. Select all the data from it.

QUERY:

```
desc DEPARTMENTS; select *
```

```
from DEPARTMENTS;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. The main workspace has a toolbar with various icons. Below the toolbar, the SQL command 'desc DEPARTMENTS;' is entered in the text area. The results pane shows the description of the DEPARTMENTS table, including columns ID, LAST_NAME, and SALARY with their respective data types and constraints.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENTS	ID	NUMBER	-	4	0	-	✓	-	-
	LAST_NAME	VARCHAR2	25	-	-	-	✓	-	-
	SALARY	NUMBER	-	5	0	-	✓	-	-

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

```
QUERY: select employee_number, last_name, job_code, hire_date from
```

```
EMPLOYEES; OUTPUT:
```

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for App Builder, SQL Workshop, Team Development, and Gallery. On the right, a user profile for 'Dhanya Vanisha euphoria' is displayed. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The schema is set to 'WKSP_EUPHORIA'. A search bar is at the top right. Below the command input field, there are buttons for 'Save' and 'Run'. The command entered is 'select employee_number, last_name, job_code, hire_date from EMPLOYEES;'. The results section shows a table with four columns: EMPLOYEE_NUMBER, LAST_NAME, JOB_CODE, and HIRE_DATE. The data returned is:

	EMPLOYEE_NUMBER	LAST_NAME	JOB_CODE	HIRE_DATE
1		adam	123	05/10/2020
2		eve	234	02/20/2021
3		joe	345	03/11/2024

Below the table, it says '3 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom, there are footer links for email, language, and help, along with copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

4. Provide an alias STARTDATE for the hire date.

QUERY: select hire_date as START_DATE from
EMPLOYEES;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, a user profile for 'Dhanya Vanisha' is shown. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, displaying the following SQL code:

```
1 select hire_date as START_DATE from EMPLOYEES;
```

The Results tab shows the output of the query:

START_DATE
05/10/2020
02/20/2021
03/11/2024

Below the results, it says '3 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes copyright information for Oracle and a note about Oracle APEX version 23.2.4.

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

QUERY:

select distinct job_code from EMPLOYEES;

The screenshot shows the Oracle APEX SQL Workshop interface, identical to the previous one but with a different query. The SQL Commands tab is active, displaying the following SQL code:

```
1 select distinct job_code from EMPLOYEES;
```

The Results tab shows the output of the query:

JOB_CODE
345
234
123

Below the results, it says '3 rows returned in 0.00 seconds' and has a 'Download' link. The bottom of the page includes copyright information for Oracle and a note about Oracle APEX version 23.2.4.

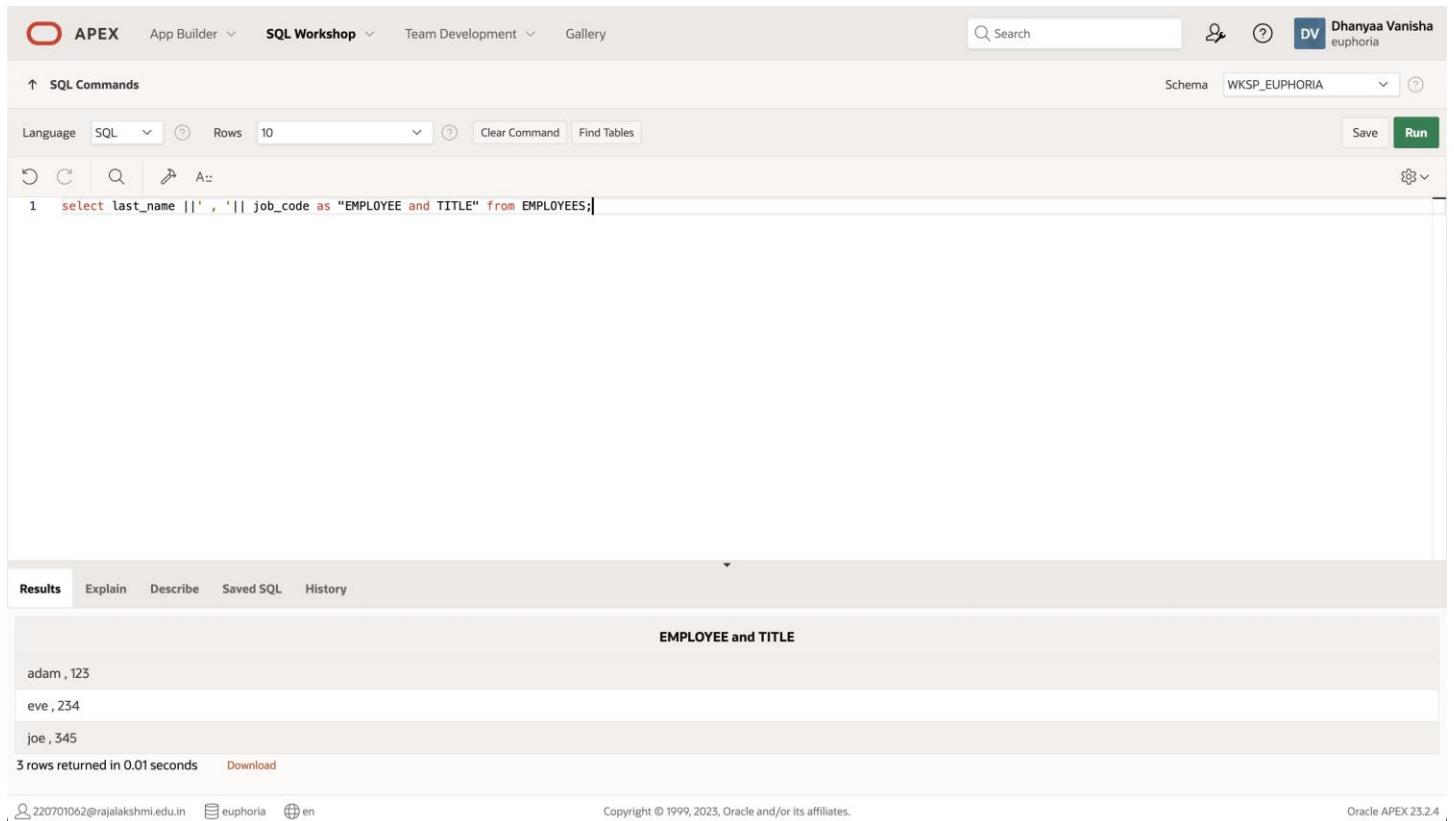
OUTPUT:

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_code as "EMPLOYEE and TITLE" from 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. A search bar and user profile 'Dhanya Vanisha euphoria' are also present. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select last_name ||' , '|| job_code as "EMPLOYEE and TITLE" from EMPLOYEES;
```

The results tab is selected, displaying the output:

EMPLOYEE and TITLE
adam , 123
eve , 234
joe , 345

Below the results, it says '3 rows returned in 0.01 seconds' and provides a 'Download' link. The bottom footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

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 last_name ||' , '|| job_code ||' , '|| employee_number||' , '|| hire_date as "The Output" from EMPLOYEES;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Search

Schema WKSP_EUPHORIA

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 select last_name ||' ,'|| job_code || ' ,'||employee_number|| ' ,'|| hire_date as "The Output" from EMPLOYEES;
```

Results Explain Describe Saved SQL History

The Output

```
adam , 123 , 1 , 05/10/2020
eve , 234 , 2 , 02/20/2021
joe , 345 , 3 , 03/11/2024
```

3 rows returned in 0.01 seconds Download

✉ 220701062@rajalakshmi.edu.in 📄 euphoria 🌐 en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

RESULT:

RESTRICTING AND SORTING DATA

EX.NO.5

DATE:

Find the Solution for the following:

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

QUERY:

Select last_name ,Salary from EMP;

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's a user profile for 'Dhanyaa Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains a command line with the following text:

```
1 Select last_name ,Salary from EMP;
2 |
```

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

LAST_NAME	SALARY
Adam	12000
Joel	12000
Henry	2000
Min	30000
eve	10000

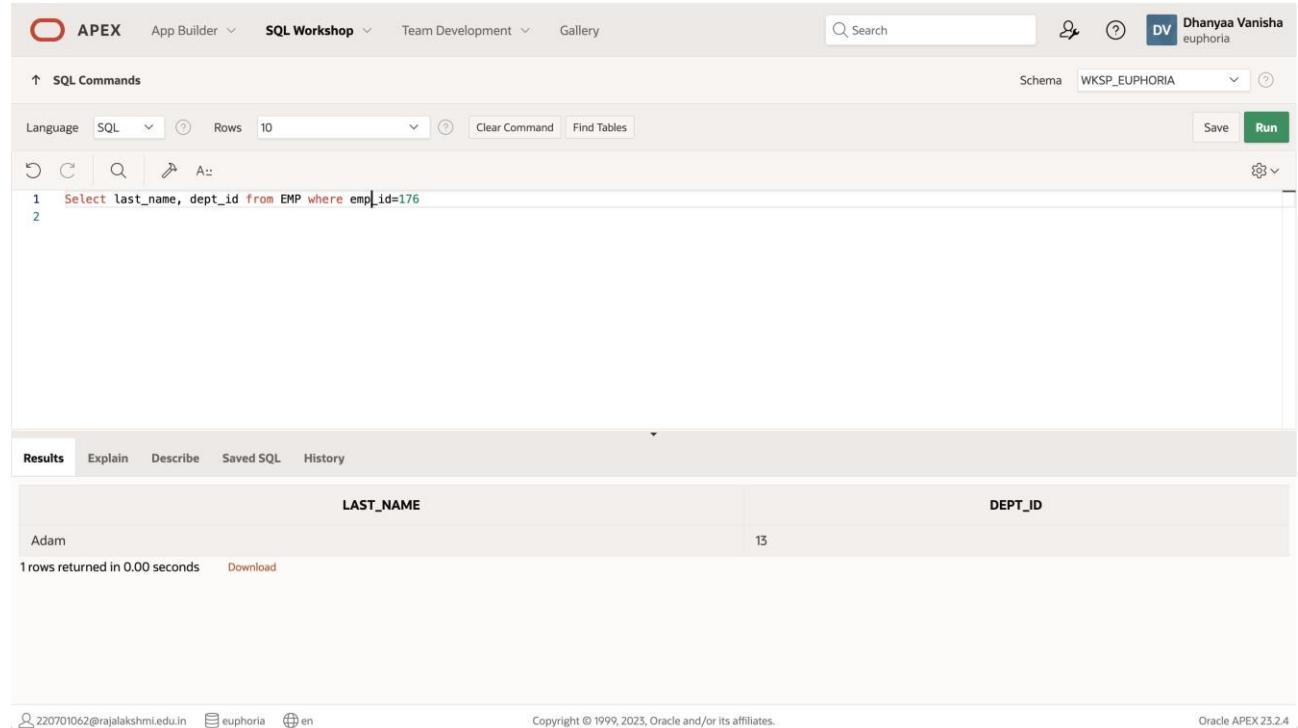
At the bottom of the results pane, it says '5 rows returned in 0.01 seconds'. The footer of the page includes copyright information for Oracle and links for user profile, euphoria, and language selection (en).

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

QUERY:

Select last_name, dept_id from EMP where e_id=176;

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's a user profile for 'Dhanya Vanisha euphoria'. The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the query: 'Select last_name, dept_id from EMP where emp_id=176'. The results tab shows a single row: Adam, 13. The bottom footer includes copyright information and a link to Oracle APEX 23.2.4.

LAST_NAME	DEPT_ID
Adam	13

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 , dept_id from EMP Where
Salary NOT BETWEEN 5000 AND 12000;**

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, a user profile for 'Dhanya Vanisha euphoria' is shown. The main area is titled 'SQL Commands' with a 'Run' button. The code entered is:

```
1 SELECT last_name , dept_id from EMP
2 Where Salary NOT BETWEEN 5000 AND 12000;
3
```

Below the code, the 'Results' tab is selected, showing a table with two rows:

LAST_NAME	DEPT_ID
Henry	12
Min	23

Text at the bottom indicates '2 rows returned in 0.00 seconds' and a 'Download' link.

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, emp_id, hire_date from EMP where hire_date  
between 'February 20,1998' AND 'May 1,1998'.
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for App Builder, SQL Workshop, Team Development, and Gallery. On the right, a user profile for 'Dhanya Vanisha euphoria' is displayed. The main area is titled 'SQL Commands' with a 'Run' button. The schema is set to 'WKSP_EUPHORIA'. The code entered is:

```
1 Select last_name, emp_id, hire_date from EMP
2 where hire_date between 'February 20,1998' AND 'May 1,1998';
3
4
```

The results section shows a table with three columns: LAST_NAME, EMP_ID, and HIRE_DATE. The data returned is:

LAST_NAME	EMP_ID	HIRE_DATE
Adam	176	02/21/1998
Joel	170	04/20/1998

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

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,Salary from EMP where (Salary BETWEEN 5000 AND 12000) AND (dept_id IN(20,50)) order by last_name AS EMPLOYEE, MONTHLY SALARY;

OUTPUT:

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 Employee  
where (Salary BETWEEN 5000 AND 12000) AND (dept_id IN(20,50)) order by last_name ;
```

OUTPUT:

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 EMPLOYEE where hire_date like '%1994';

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's a search bar, user profile (Dhanya Vanisha euphoria), and a schema dropdown set to WKSP_EUPHORIA. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 Select last_name ,hire_date from EMPLOYEE where hire_date like '%1994';
2
```

Below the code, the results tab is selected, showing the output:

LAST_NAME	HIRE_DATE
Min	09/30/1994

1 rows returned in 0.00 seconds [Download](#)

At the bottom, the footer includes user information (220701062@rajalakshmi.edu.in, euphoria, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 23.2.4).

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_title from EMPLOYEE where manager_id is NULL;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for 'APEX', 'App Builder', 'SQL Workshop' (which is selected), 'Team Development', and 'Gallery'. On the right, a user profile for 'Dhanyaa Vanisha euphoria' is shown. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. It displays a query: 'Select last_name ,job_title from EMPLOYEE where manager_id is NULL;'. Below the query, the results are shown in a table:

LAST_NAME	JOB_TITLE
Joel	INTERN
Min	TESTER

Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes standard footer links like '220701062@rajalakshmi.edu.in', 'euphoria', 'en', and copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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,order by)

QUERY:

Select last_name ,Salary,commission from EMPLOYEE where commission is not

null order by Salary,commission DESC; OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile for 'Dhanyaa Vanisha euphoria' are also at the top right. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. It displays a query:

```
1 Select last_name ,Salary,commision from EMPLOYEE where commision is not null order by Salary,commision DESC;
2
```

Below the command area, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing a table with three columns: LAST_NAME, SALARY, and COMMISION. The data returned is:

LAST_NAME	SALARY	COMMISION
Henry	2000	1000
eve	10000	5000
Adam	12000	3000
Joel	12000	500

Below the table, it says '4 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for user profile, workspace, and language, along with copyright information and the Oracle APEX version.

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

QUERY:

Select last_name from EMPLOYEE where last_name LIKE

'__A%';

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. A search bar and user profile for 'Dhanya Vanisha euphoria' are also present. The main area is titled 'SQL Commands' and contains a code editor with the following SQL query:

```
1 Select last_name from EMPLOYEE where last_name LIKE '_A%';
```

The results tab shows the message "no data found".

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 ,job_title ,Salary from EMPLOYEE where last_name LIKE '%A%'

AND last_name LIKE '%E%'; 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. A search bar and user profile for 'Dhanya Vanisha euphoria' are also present. The main area is titled 'SQL Commands' and contains a code editor with the following SQL query:

```
1 Select last_name ,job_title ,Salary from EMPLOYEE where last_name LIKE '%A%' AND last_name LIKE '%E%';
```

The results tab shows the message "no data found".

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 AS "EMPLOYEE",Salary AS "MONTHLY SALARY" from EMPLOYEE  
where (Salary BETWEEN 5000 AND 12000) AND (dept_id IN(20,50)) 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', 'Gallery', a search bar, and a user profile for 'Dhanya Vanisha euphoria'. The main area has tabs for 'SQL Commands' and 'Results'. In the 'SQL Commands' tab, the query is displayed:

```
1 Select last_name AS "EMPLOYEE",Salary AS "MONTHLY SALARY" from EMPLOYEE where (Salary BETWEEN 5000 AND 12000) AND (dept_id IN(20,50)) order by last_name ;  
2  
3  
4  
5
```

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

EMPLOYEE	MONTHLY SALARY
eva	10000

Below the table, it says '1 rows returned in 0.00 seconds' and has a 'Download' link.

1. 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 from EMPLOYEE where commission=Salary*0.20;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Schema WKSP_EUPHORIA Save Run

1 Select last_name ,Salary,commision from EMPLOYEE where commision=Salary*.20;
2

Results Explain Describe Saved SQL History

LAST_NAME	SALARY	COMMISION
Adam	12000	2400

1 rows returned in 0.01 seconds Download

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

RESULT:

SINGLE ROW FUNCTIONS

EX.NO.6.

DATE:

Find the Solution for the following:

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

QUERY:

SELECT SYSDATE AS “DATE” FROM DUAL; OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands pane, the query `select sysdate from dual;` is entered. The results pane displays a single row with the column `SYSDATE` containing the value `03/09/2024`. The status bar at the bottom indicates "1 rows returned in 0.02 seconds".

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

QUERY:

**SELECT emp_number, last_name, Salary, Salary+(15.5/100*Salary) “NEW SALARY”
From EMP;**

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands pane, the query `select emp_number, last_name, salary, salary+(salary*15.5/100) "New Salary" from emp;` is entered. The results pane displays a table with columns `EMP_NUMBER`, `LAST_NAME`, `SALARY`, and `New Salary`. The data shows two rows: one for employee 123 with a new salary of 13860, and one for employee 254 with a new salary of 11550. The status bar at the bottom indicates "2 rows returned in 0.02 seconds".

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 emp_number, last_name, Salary, Salary+(Salary*15/100))-Salary "Increase"
From EMP;

OUTPUT:

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

```
1 SELECT emp_number , last_name , salary , (salary+(salary *15/100))-salary "Increase"
2 from emp;
```

The results section displays the following data:

EMP_NUMBER	LAST_NAME	SALARY	Increase
123	adam	12000	1800
254	eve	10000	1500

2 rows returned in 0.01 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) "Name", length(last_name) "Length of Name"
from EMP
where last_name like 'J%' or last_name like 'A%' or last_name like 'M%' order
by last_name; **OUTPUT:**

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

```
1 Select initcap(last_name) "Name", length(last_name) "Length of Name"
2 from EMP
3 where last_name like 'J%' or last_name like 'A%' or last_name like 'M%' ;
4 order by last_name;
```

The results section displays the following data:

Name	Length of Name
Adam	4
Joel	4

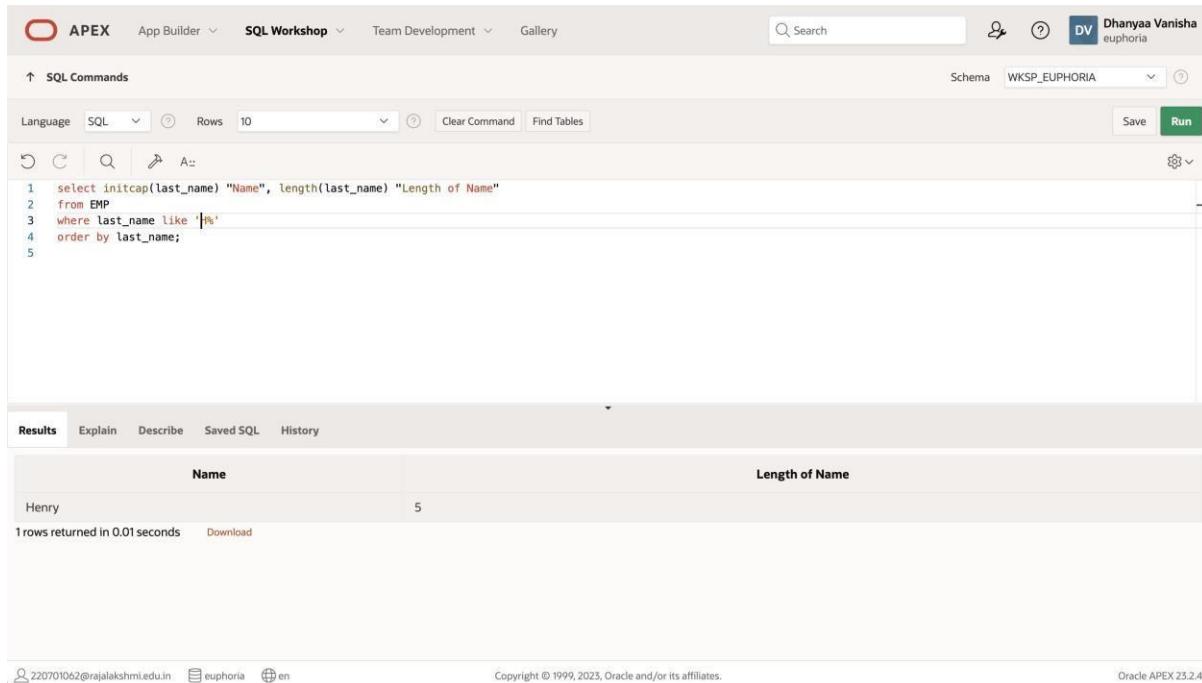
2 rows returned in 0.01 seconds

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) "Name", length(last_name) "Length of Name" from  
EMP where last_name like '&name%' order by last_name;
```

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 'Dhanya Vanisha euphoria'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab displays the following SQL code:

```
1 select initcap(last_name) "Name", length(last_name) "Length of Name"  
2 from EMP  
3 where last_name like '&name%'  
4 order by last_name;  
5
```

The Results tab shows the output of the query:

Name	Length of Name
Henry	5

Below the results, it says '1 rows returned in 0.01 seconds' and there is a 'Download' link. The bottom of the page includes copyright information for Oracle and the APEX version.

6. The HR department wants to find the length of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column MONTHS_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.

QUERY:

```
select last_name, round(months_between(sysdate,hire_date),0) Months_worked from EMP order by 2;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for 'APEX', 'App Builder', 'SQL Workshop' (which is selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. Below the tabs, there's a search bar and some schema-related buttons. The main area has a toolbar with icons for refresh, search, and run. The SQL command input field contains the following code:

```
1 select last_name, round(months_between(sysdate,hire_date),0) Months_worked
2 from EMP order by 2;
3
```

Below the code, the 'Results' tab is selected, showing a table with two columns: 'LAST_NAME' and 'MONTHS_WORKED'. The data is as follows:

LAST_NAME	MONTHS_WORKED
Henry	175
eve	228
Adam	229
min	233
Joel	358

At the bottom of the results panel, it says '5 rows returned in 0.01 seconds'. There are also download and history buttons.

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 "Dream Salary"
from EMP;
```

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, it shows the user 'Dhanya Vanisha' and the schema 'WKSP_EUPHORIA'. The main area is titled 'SQL Commands' with a sub-section 'Results'. The SQL code entered is:

```
1 Select last_name||' earns $'||salary||' monthly but wants $'||salary*3 "Dream Salary"
2 from EMP;
3
```

The results section displays the output:

Dream Salary	
Adam	earns \$12000 monthly but wants \$36000
Joel	earns \$12000 monthly but wants \$36000
Henry	earns \$20000 monthly but wants \$60000
min	earns \$30000 monthly but wants \$90000
eve	earns \$10000 monthly but wants \$30000

Below the results, it says '5 rows returned in 0.01 seconds'.

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

QUERY:

Select last_name, lpad(salary,15,'\$') Salary

from EMP;

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, it shows the user 'Dhanya Vanisha' and the schema 'WKSP_EUPHORIA'. The main area is titled 'SQL Commands' with a sub-section 'Results'. The SQL code entered is:

```
1 Select last_name, lpad(salary,15,'$') Salary
2 from emp;
3
```

The results section displays the output:

LAST_NAME	SALARY
Adam	\$\$\$\$\$\$\$\$\$\$12000
Joel	\$\$\$\$\$\$\$\$\$\$12000
Henry	\$\$\$\$\$\$\$\$\$\$20000
min	\$\$\$\$\$\$\$\$\$\$30000
eve	\$\$\$\$\$\$\$\$\$\$10000

Below the results, it says '5 rows returned in 0.01 seconds'.

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

QUERY:

**select last_name, hire_date, to_char((next_day(hire_date,'Monday')),'fmday," the
"ddspth "of" month,yyyy') "REVIEW" from EMP;**

OUTPUT

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 select last_name, hire_date, to_char(next_day(hire_date,'Monday'),'fmday," the "ddspth "of" month,yyyy') "REVIEW" from EMP;
```

Results Explain Describe Saved SQL History

LAST_NAME	HIRE_DATE	REVIEW
Adam	02/11/2005	monday, the fourteenth of february,2005
Joel	05/20/1994	monday, the twenty-third of may,1994
Henry	08/11/2009	monday, the seventeenth of august,2009
min	09/30/2004	monday, the fourth of october,2004
eve	03/11/2005	monday, the fourteenth of march,2005

5 rows returned in 0.00 seconds Download

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

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

QUERY:

Select Last_name, hire_date, to_char(hire_date,'Day') "Day" from EMP
order by to_char(hire_date-1,'d') OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 Select Last_name, hire_date, to_char(hire_date,'Day') "Day"
2 from EMP
3 order by to_char(hire_date-1,'d')
```

Results Explain Describe Saved SQL History

LAST_NAME	HIRE_DATE	Day
Henry	08/11/2009	Tuesday
min	09/30/2004	Thursday
Joel	05/20/1994	Friday
eve	03/11/2005	Friday
Adam	02/11/2005	Friday

5 rows returned in 0.01 seconds Download

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

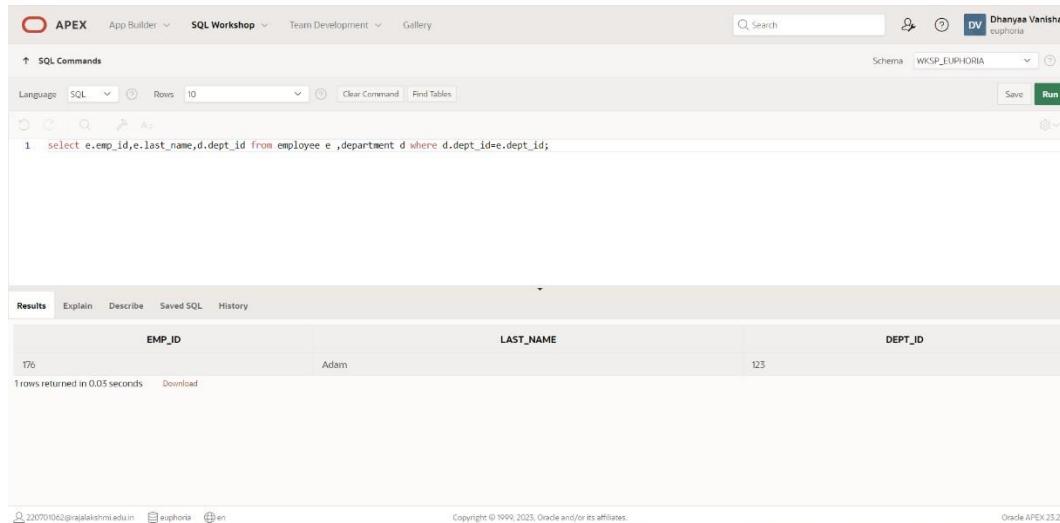
DATE:

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

QUERY:

```
Select e.emp_id,e.last_name,d.dept_id from employee e,department d where d.dept_id=e.dept_id;
```

OUTPUT:



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

```
1 select e.emp_id,e.last_name,d.dept_id from employee e ,department d where d.dept_id=e.dept_id;
```

The results table has three columns: EMP_ID, LAST_NAME, and DEPT_ID. One row is returned, showing Adam in department 123.

EMP_ID	LAST_NAME	DEPT_ID
176	Adam	123

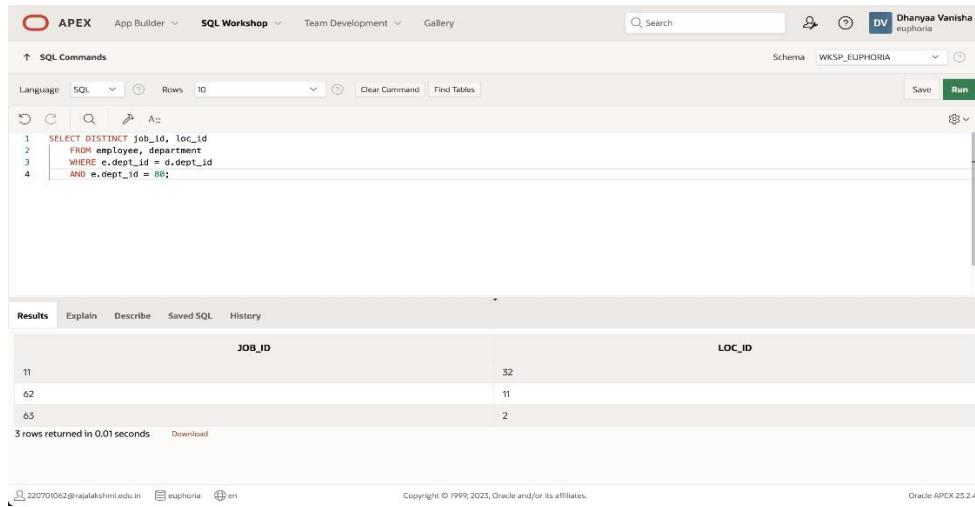
1 rows returned in 0.05 seconds

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, loc_id FROM employee e, department d WHERE e.dept_id = d.dept_id AND e.dept_id = 80;
```

OUTPUT:



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

```
1 SELECT DISTINCT job_id, loc_id
2   FROM employee, department
3  WHERE e.dept_id = d.dept_id
4    AND e.dept_id = 80;
```

The results table has two columns: JOB_ID and LOC_ID. Three rows are returned, showing (32, 11), (62, 11), and (63, 2).

JOB_ID	LOC_ID
32	11
62	11
63	2

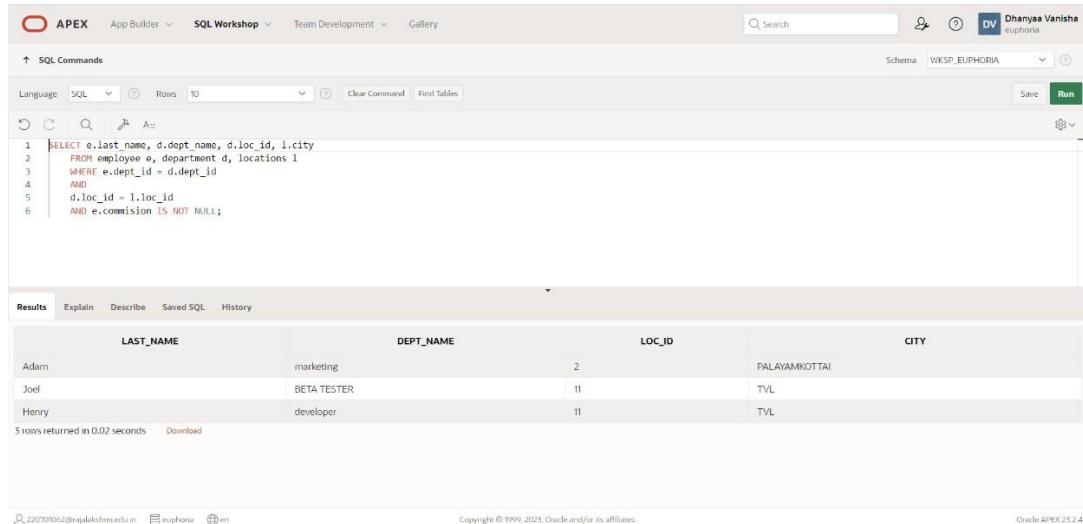
3 rows returned in 0.01 seconds

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

QUERY:

```
SELECT e.last_name, d.dept_name, d.loc_id, l.city FROM employee e, department d, locations l WHERE e.dept_id = d.dept_id AND d.loc_id = l.loc_id AND e.commission IS NOT NULL;
```

OUTPUT:



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

```
1 SELECT e.last_name, d.dept_name, d.loc_id, l.city
2   FROM employee e, department d, locations l
3  WHERE e.dept_id = d.dept_id
4    AND
5    d.loc_id = l.loc_id
6   AND e.commission IS NOT NULL;
```

The results section displays the following data:

LAST_NAME	DEPT_NAME	LOC_ID	CITY
Adam	marketing	2	PALAYAMKOTTAI
Joel	BETA TESTER	11	TVL
Henry	developer	11	TVL

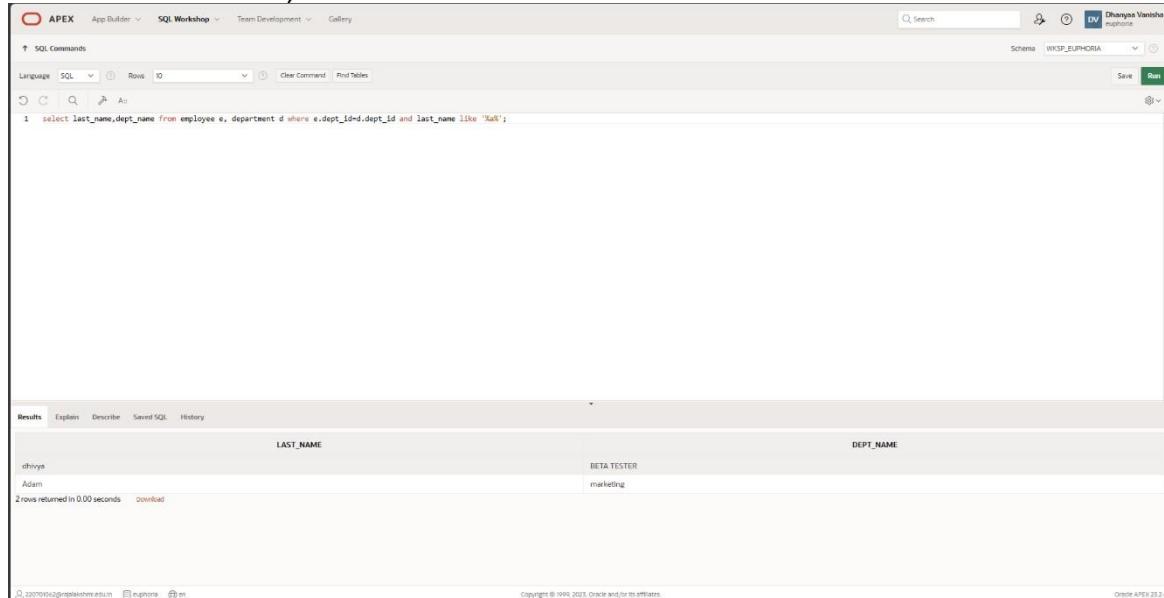
3 rows returned in 0.02 seconds. [Download](#)

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

QUERY:

```
SELECT last_name, dept_name FROM employee e, department d WHERE e.dept_id = d.dept_id AND last_name LIKE '%a%';
```

OUTPUT:



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

```
1 select last_name,dept_name from employee e, department d where e.dept_id=d.dept_id and last_name like '%a%';
```

The results section displays the following data:

LAST_NAME	DEPT_NAME
dhanya	BETA TESTER
Adam	marketing

2 rows returned in 0.00 seconds. [Download](#)

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.job_id, e.dept_id, d.dept_name
FROM employee e JOIN department d ON (e.dept_id = d.dept_id)
JOIN locations l ON (d.loc_id = l.loc_id)
WHERE LOWER(l.city) = 'toronto';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command is pasted into the editor, and the results are displayed in a grid format. The result shows one row for Henry, with columns LAST_NAME, JOB_ID, DEPT_ID, and DEPT_NAME. The DEPT_NAME is developer.

LAST_NAME	JOB_ID	DEPT_ID	DEPT_NAME
Henry	11	80	developer

1 rows returned in 0.00 seconds [Download](#)

6. Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, Respectively

QUERY:

```
SELECT e.last_name "Employee", e.employee_id "EMP#", m.last_name "Manager", m.employee_id "Mgr#"
FROM employee e join managers m ON (e.mgr_id = m.emp_id);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command is pasted into the editor, and the results are displayed in a grid format. The result shows three rows for Adam, Eve, and King, with columns Employee, EMP#, Manager, and Mgr#. The Manager column contains the manager's last name, and the Mgr# column contains the manager's employee ID.

Employee	EMP#	Manager	Mgr#
Adam	176	daisy	62
Adam	176	eve	62
Joe	234	king	11

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

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 "Employee", e.emp_id "EMP#", m.last_name "Manager", m.emp_id "Mgr#"  
FROM employee e LEFT OUTER JOIN managers m ON (e.mgr_id = m.emp_id);
```

OUTPUT:

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

```
1 SELECT e.last_name "Employee", e.emp_id "EMP#",  
2      m.last_name "Manager", m.emp_id "Mgr#"  
3   FROM employee e  
4     LEFT OUTER JOIN managers m  
5       ON (e.mgr_id = m.emp_id);
```

The Results tab displays the output of the query:

Employee	EMP#	Manager	Mgr#
Dhanya	170	-	-
Adam	176	-	-
Joe	254	-	-
Min	46	-	-
Eva	170	-	-

Rows returned in 0.02 seconds

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.dept_id department, e.last_name employee, c.last_name colleague  
FROM employee e JOIN employee c  
ON (e.dept_id = c.dept_id)  
WHERE e.emp_id <> c.emp_id ORDER BY e.dept_id, e.last_name,  
c.last_name;
```

OUTPUT:

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

```
1 SELECT e.dept_id department, e.last_name employee,  
2        c.last_name colleague  
3   FROM employee e JOIN employee c  
4     ON (e.dept_id = c.dept_id)  
5    WHERE e.emp_id <> c.emp_id  
6    ORDER BY e.dept_id, e.last_name, c.last_name;
```

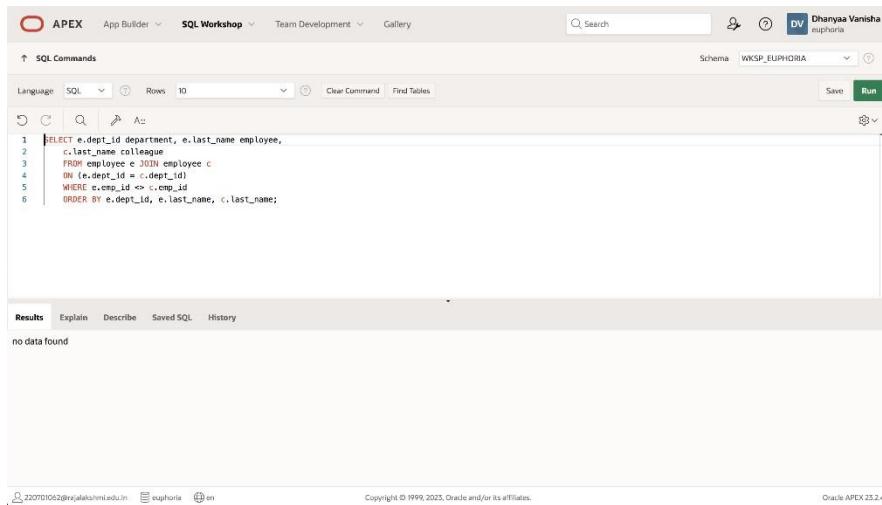
The Results tab shows the message: "no data found".

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_lvl FROM employee e JOIN department d  
ON (e.dept_id = d.dept_id) JOIN Grade j  
ON (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 right side shows the user 'Dhanya Vanisha euphoria'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is:

```
1 SELECT e.dept_id department, e.last_name employee,  
2       c.last_name colleague  
3  FROM employee e JOIN employee c  
4    ON (e.dept_id = c.dept_id)  
5   WHERE e.emp_id >= c.emp_id  
6 ORDER BY e.dept_id, e.last_name, c.last_name;
```

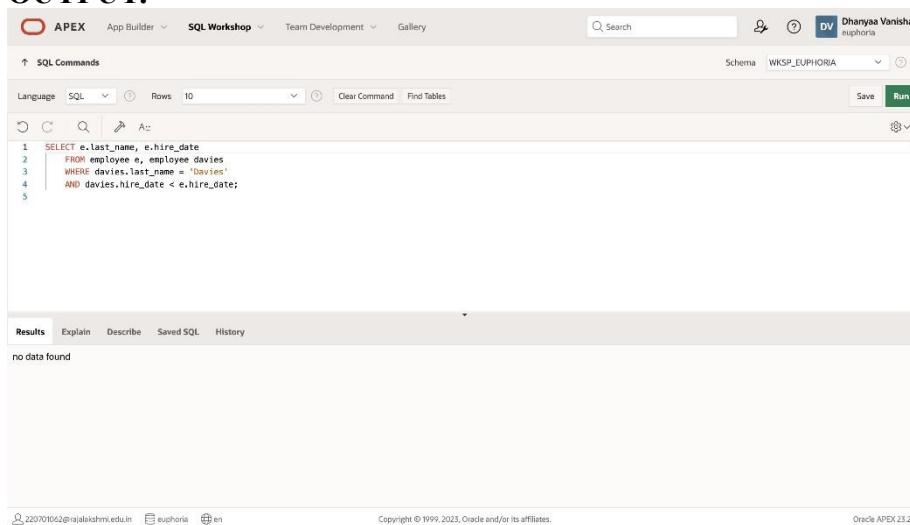
The results section below shows the message 'no data found'.

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

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 the user 'Dhanya Vanisha euphoria'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is:

```
1 SELECT e.last_name, e.hire_date  
2  FROM employee e, employee davies  
3 WHERE davies.last_name = 'Davies'  
4   AND davies.hire_date < e.hire_date;  
5
```

The results section below shows the message 'no data found'.

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, e.hire_date, m.last_name, m.hire_date    FROM employee e, managers m    WHERE e.mgr_id = m.emp_id AND 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. The user is logged in as 'Dhanya Vanisha' with the schema 'WKSP_EUPHORIA'. The SQL Commands tab is selected, showing the following SQL code:

```
1 SELECT e.last_name, e.hire_date, m.last_name, m.hire_date
2   FROM employee e, managers m
3  WHERE e.mgr_id = m.emp_id
4  AND e.hire_date < m.hire_date;
5
```

Below the code, the 'Run' button is highlighted. The results section displays the output:

LAST_NAME	HIRE_DATE	LAST_NAME	HIRE_DATE
Adam	02/21/1998	daisy	11/03/2005

1 rows returned in 0.01 seconds. There is a 'Download' link next to the results.

Evaluation Procedure	Marks awarded
Query(5)	

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

RESULT:

AGGREGATING DATA USING GROUP FUNCTIONS

EX_NO:8

DATE:

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 employee;
```

OUTPUT:

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

```
1 SELECT ROUND(MAX(salary),0) "Maximum",  
2 ROUND(MIN(salary),0) "Minimum",  
3 ROUND(SUM(salary),0) "Sum",  
4 ROUND(AVG(salary),0) "Average"  
5 FROM employee;
```

Below the code, the 'Results' tab is selected, displaying the output:

Maximum	Minimum	Sum	Average
30000	2000	66000	13200

The results table shows one row returned in 0.00 seconds. The bottom of the page includes footer links for 220701062@rajalakshmi.edu.in, euphoria, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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 employee
 GROUP BY job_id;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. On the right, there's a user profile for 'Dhanyaa Varisha euphoria'. Below the tabs, there's a search bar and a schema dropdown set to 'WKSP_EUPHORIA'. The main area is a code editor with the following SQL query:

```
1  SELECT job_id, ROUND(MAX(salary),0) "Maximum",
2  ROUND(MIN(salary),0) "Minimum",
3  ROUND(SUM(salary),0) "Sum",
4  ROUND(AVG(salary),0) "Average"
5  FROM employee|
6  GROUP BY job_id;
```

Below the code editor, there are buttons for Save and Run. The 'Run' button is highlighted in green. The results section shows a table with the following data:

JOB_ID	Maximum	Minimum	Sum	Average
62	12000	12000	12000	12000
63	12000	12000	12000	12000
11	2000	2000	2000	2000
-	30000	10000	40000	20000

At the bottom left, it says '4 rows returned in 0.01 seconds' and there's a 'Download' link. At the bottom right, it says 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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.

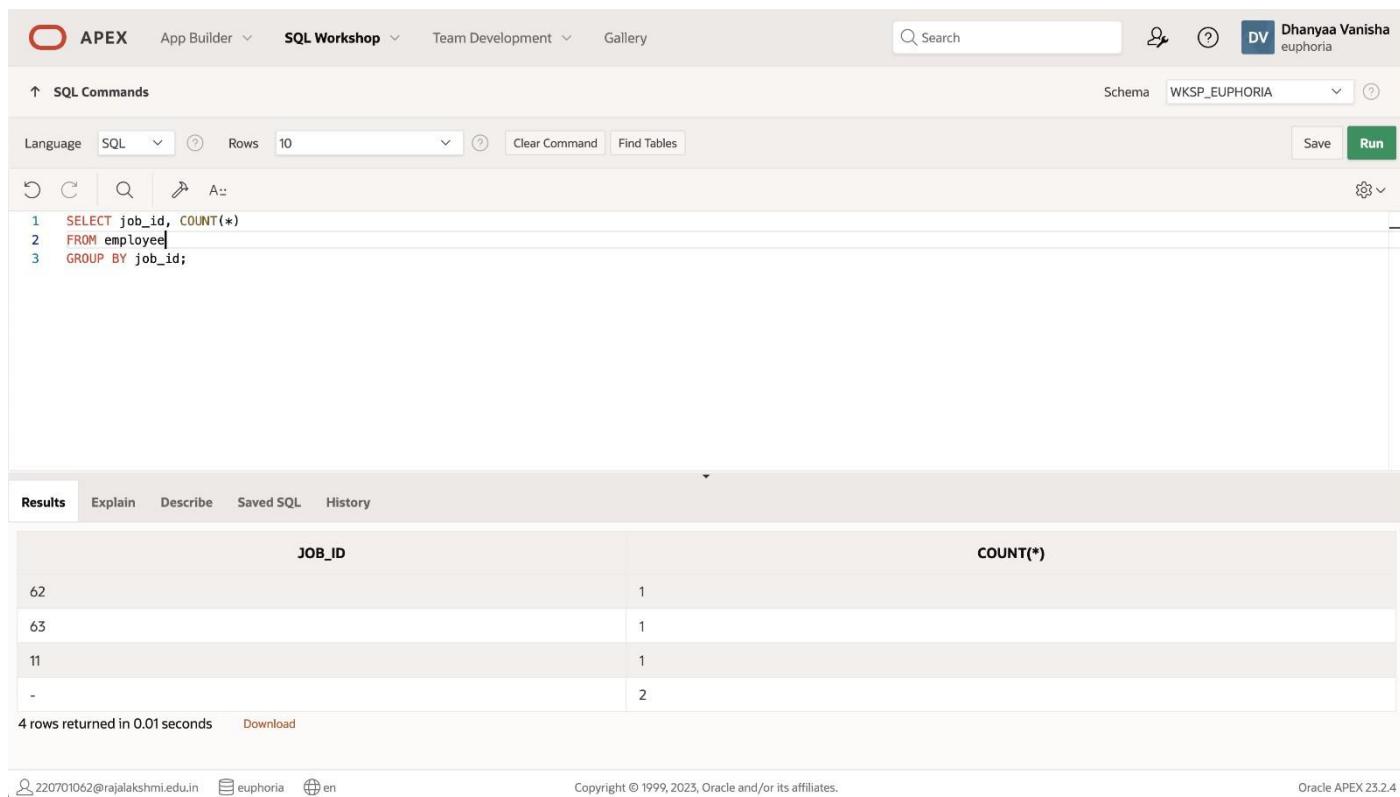
QUERY:

```
SELECT job_id, COUNT(*)
```

```
FROM employee
```

```
GROUP BY job_id;
```

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 the user profile "Dhanya Vanisha euphoria". The main workspace is titled "SQL Commands". The SQL editor contains the following code:

```
1 SELECT job_id, COUNT(*)
2 FROM employee
3 GROUP BY job_id;
```

The "Results" tab is selected at the bottom left. The results table has two columns: "JOB_ID" and "COUNT(*)". The data returned is:

JOB_ID	COUNT(*)
62	1
63	1
11	1
-	2

Below the table, it says "4 rows returned in 0.01 seconds" and there is a "Download" link. The footer includes the email "220701062@rajalakshmi.edu.in", the brand "euphoria", and the language "en". Copyright information "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version "Oracle APEX 23.2.4" are also present.

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 mgr_id) "Number of Managers"FROM employee;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'Dhanyaa Varsha euphoria'. Below the tabs, there's a search bar and a toolbar with icons for Undo, Redo, Refresh, and Help. The main area is titled 'SQL Commands' and shows the following content:

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

Query entered:

```
1 SELECT COUNT(DISTINCT mgr_id) "Number of Managers"FROM employee;
```

The results section shows a single row with the value '3' under the heading 'Number of Managers'. Below the results, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for email (220701062@rajalakshmi.edu.in), session (euphoria), and language (en). The copyright notice reads 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the page is identified as 'Oracle APEX 23.2.4'.

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

QUERY:

```
SELECT MAX(salary) - MIN(salary) DIFFERENCE FROM employee;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for 'APEX' (selected), 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user profile for 'Dhanya Vanisha euphoria', and a 'Save' button. The main area is titled 'SQL Commands' and contains a language selector ('SQL'), row limit ('Rows 10'), and buttons for 'Clear Command' and 'Find Tables'. Below this is a toolbar with icons for refresh, search, and other functions. A code editor window displays the following SQL query:

```
1 SELECT MAX(salary) - MIN(salary) DIFFERENCE FROM employee;
```

Below the code editor, there are tabs for 'Results' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. The results section shows a single row with the value '28000' under the heading 'DIFFERENCE'. Below the result, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for email, user profile, and language, along with copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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 mgr_id, MIN(salary)FROM employee
```

```
WHERE mgr_id IS NOT NULL
```

```
GROUP BY mgr_id
```

```
HAVING MIN(salary) > 6000
```

```
ORDER BY MIN(salary) DESC;
```

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 the user profile 'Dhanya Vanisha euphoria'. The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following query:

```
1 SELECT mgr_id, MIN(salary)FROM employee
2 WHERE mgr_id IS NOT NULL
3 GROUP BY mgr_id
4 HAVING MIN(salary) > 6000
5 ORDER BY MIN(salary) DESC;
```

Below the query, the Results tab is selected, displaying the following table:

MGR_ID	MIN(SALARY)
63	12000
64	12000

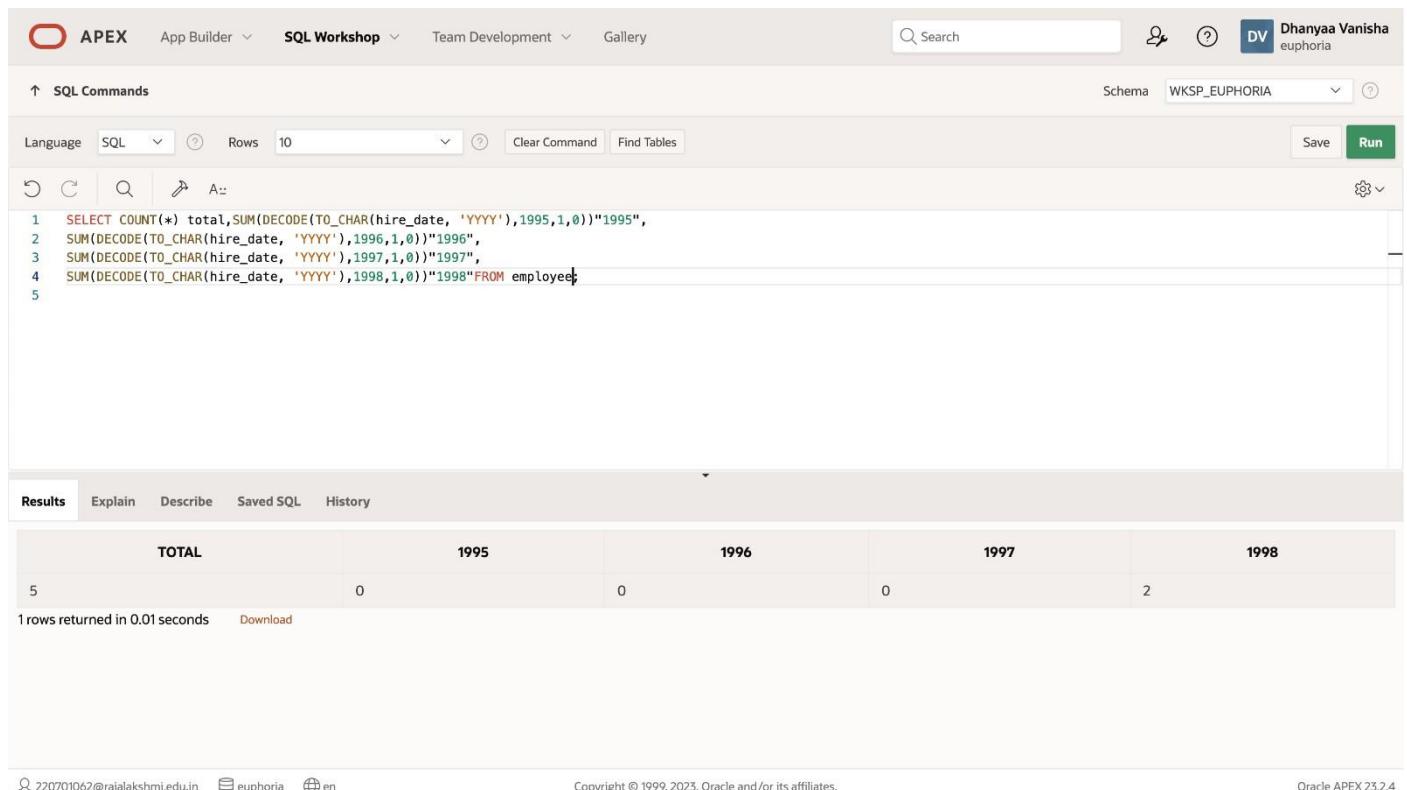
At the bottom, it says '2 rows returned in 0.00 seconds' and has a 'Download' link. The footer includes the email '220701062@rajalakshmi.edu.in', the theme 'euphoria', the language 'en', copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

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 employee;
```

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 the user profile "Dhanya Vanisha euphoria". The main workspace has tabs for SQL Commands and Results. In the SQL Commands tab, the query is pasted and executed. The Results tab displays the output in a grid format:

TOTAL	1995	1996	1997	1998
5	0	0	0	2

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

Page footer: 220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

```
SELECT job_id "Job",
       SUM(DECODE(department_id , 20, salary)) "Dept 20",
       SUM(DECODE(department_id , 50, salary)) "Dept 50",
       SUM(DECODE(department_id , 80, salary)) "Dept 80",
       SUM(DECODE(department_id , 90, salary)) "Dept 90",SUM(salary) "Total"FROM employee
GROUP BY job_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 for 'Dhanya Vanisha euphoria'. The main area is titled 'SQL Commands' with tabs for Language (SQL selected), Rows (10), Clear Command, Find Tables, Save, and Run.

The SQL command entered is:

```
1 SELECT job_id "Job",
2    SUM(DECODE(dept_id , 20, salary)) "Dept 20",
3    SUM(DECODE(dept_id , 50, salary)) "Dept 50",
4    SUM(DECODE(dept_id , 80, salary)) "Dept 80",
5    SUM(DECODE(dept_id , 90, salary)) "Dept 90",SUM(salary) "Total"FROM employee
6 GROUP BY job_id;
```

The results section displays the output of the query:

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
62	-	-	-	-	12000
63	-	-	-	-	12000
11	-	-	2000	-	2000
-	-	-	-	-	40000

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

Page footer: 220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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.

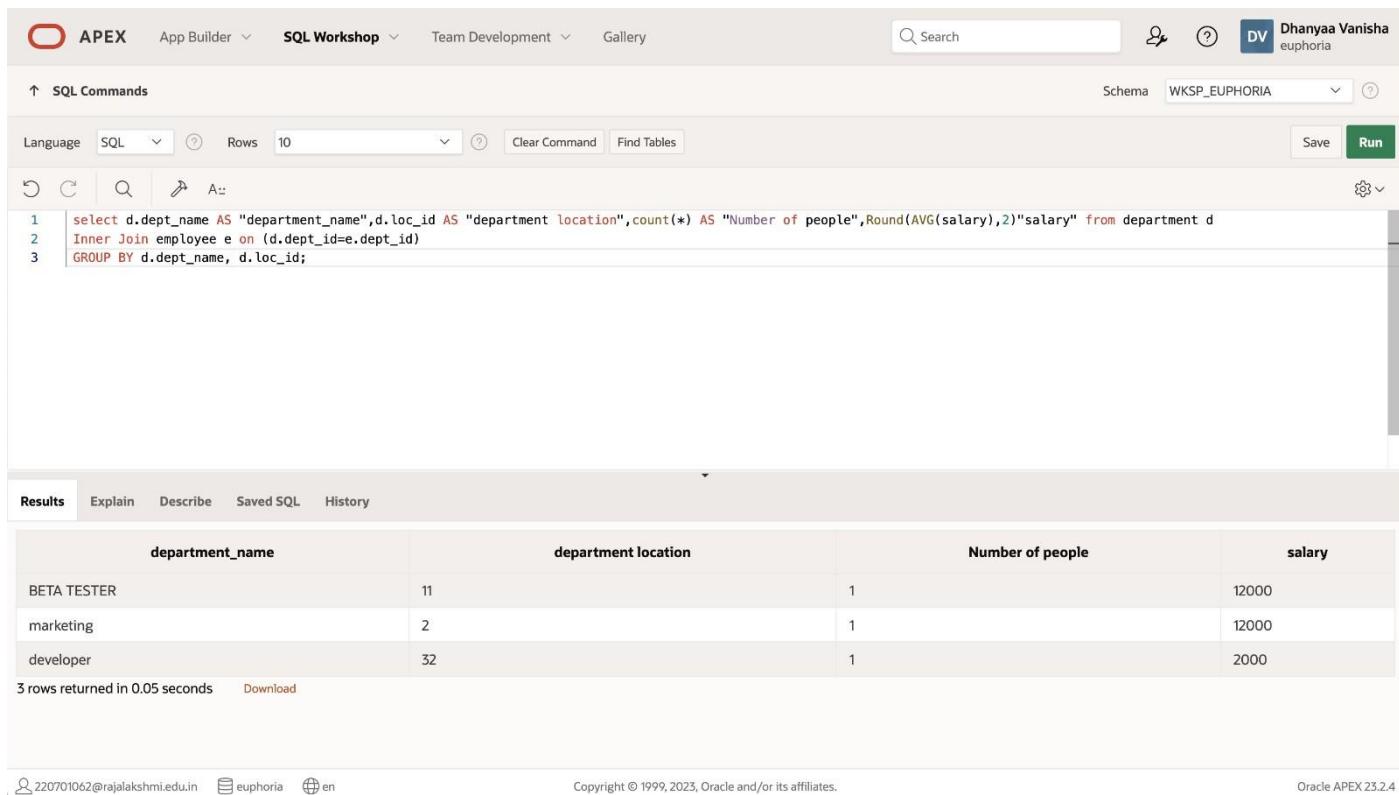
QUERY:

```
select d.dept_name AS "department_name",d.loc_id AS "department location",count(*) AS "Number of people",Round(AVG(salary),2)"salary" from department d
```

```
Inner Join employee e on (d.dept_id=e.dept_id)
```

```
GROUP BY d.dept_name, d.loc_id;
```

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 is a user profile for 'Dhanya Vanisha' and a schema dropdown set to 'WKSP_EUPHORIA'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 | select d.dept_name AS "department_name",d.loc_id AS "department location",count(*) AS "Number of people",Round(AVG(salary),2)"salary" from department d
2 | Inner Join employee e on (d.dept_id=e.dept_id)
3 | GROUP BY d.dept_name, d.loc_id;
```

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

department_name	department location	Number of people	salary
BETA TESTER	11	1	12000
marketing	2	1	12000
developer	32	1	2000

At the bottom left, it says '3 rows returned in 0.05 seconds' and 'Download'. At the bottom right, it says '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:

SUB-QUERIES

EX.NO:9

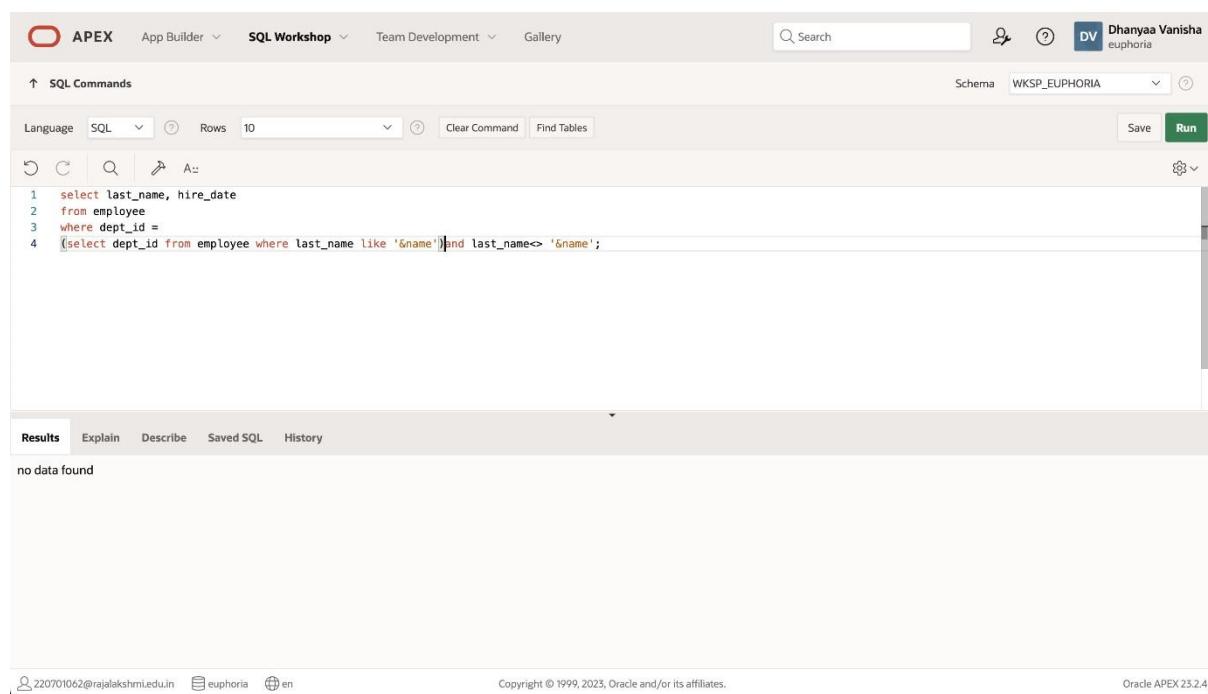
DATE:

Find the Solution for the following:

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

QUERY:

```
select last_name, hire_date  
from employee where  
dept_id =  
(select dept_id from employee where last_name like '&name')and  
last_name<>'&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'. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), and 'Clear Command'. Below this is a toolbar with icons for Undo, Redo, Search, and Run. The code editor contains the following SQL query:

```
1 select last_name, hire_date  
2 from employee  
3 where dept_id =  
4 (select dept_id from employee where last_name like '&name')and last_name<>'&name';
```

The results section below the editor is currently empty, displaying 'no data found'. At the bottom of the page, there are footer links for '220701062@rajalakshmi.edu.in', 'euphoria', and 'en', along with copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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 emp_id, last_name, salary from employee  
where salary > (select avg(salary) from employee)  
order by salary;
```

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's a user profile for 'Dhanya Vanisha euphoria' and a search bar. The main workspace has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
1 select emp_id, last_name, salary  
2 from employee  
3 where salary > (select avg(salary) from employee)  
4 order by salary;
```

The Results tab displays the output of the query:

EMP_ID	LAST_NAME	SALARY
46	Min	30000

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

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

QUERY:

```
select employee_id, last_name  
from employees  
where dept_id in (select dept_id from employees where last_name like  
'%u%');
```

OUTPUT:

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. In the SQL Commands editor, the following query is entered:

```
1 select emp_id, last_name
2 from employee
3 where dept_id in (select dept_id from employee where last_name like '%u%');
```

The Results tab displays the output:

EMP_ID	LAST_NAME
170	puma

1 rows returned in 0.01 seconds

Copyright © 1999, 2025, Oracle and/or its affiliates.

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, dept_id, job_id
from employees
where dept_id in (select dept_id from department where loc_id =1700);
```

OUTPUT:

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. In the SQL Commands editor, the following query is entered:

```
1 select last_name, dept_id, job_id
2 from employee
3 where dept_id in (select dept_id from department where loc_id =1700);
```

The Results tab displays the output:

LAST_NAME	DEPT_ID	JOB_ID
Joe	80	11

1 rows returned in 0.01 seconds

Copyright © 1999, 2025, Oracle and/or its affiliates.

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_no in (select employee_id from employees where
last_name='King');
```

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 'Dhanya Vanisha euphoria'. 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 last_name, salary
2   from employee
3  where mgr_id in (select emp_id from employee where last_name='King');
```

The Results tab displays the output of the query:

LAST_NAME	SALARY
king	12000

Below the table, it says '1 rows returned in 0.01 seconds' and there is a 'Download' link. The bottom footer includes user information (220701062@rajalakshmi.edu.in, euphoria, en), copyright notice (Copyright © 1999, 2025, Oracle and/or its affiliates.), and system information (Oracle APEX 23.2.4).

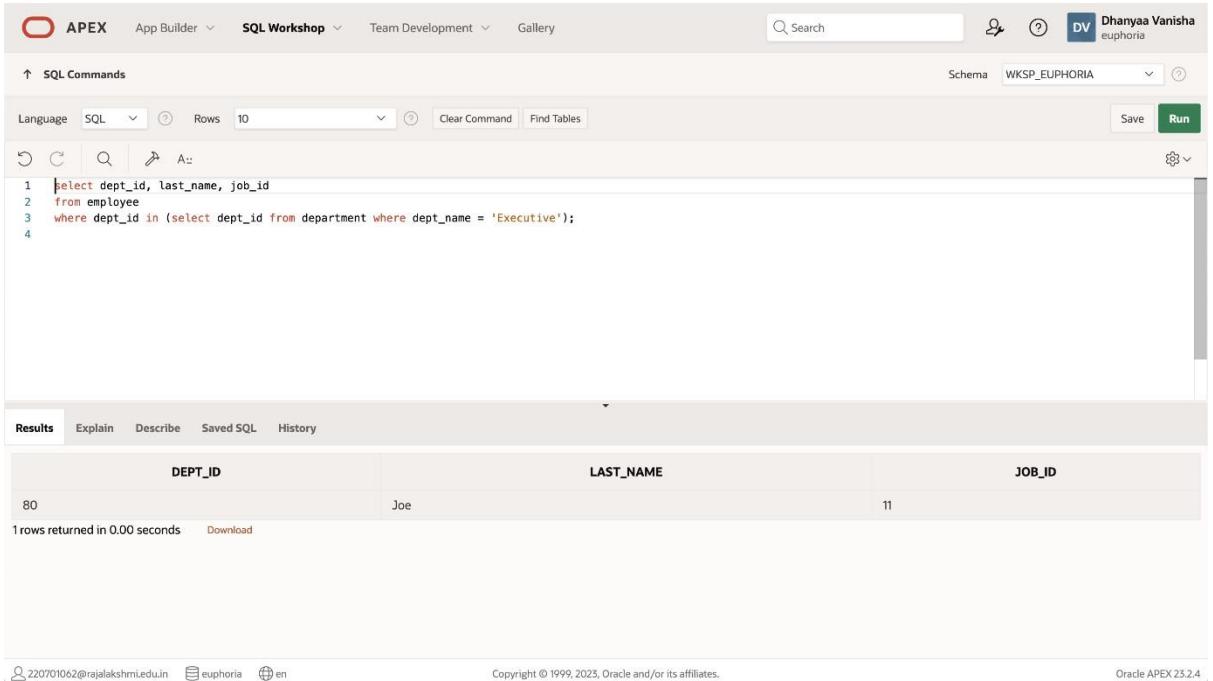
6. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

QUERY:

```
select dept_id, last_name, job_id
from employees
```

```
where dept_id in (select dept_id from department where dept_name = 'Executive');
```

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. A search bar and user profile 'Dhanya Vanisha euphoria' are also present. The main area displays a SQL command window with the following code:

```
1 select dept_id, last_name, job_id
2   from employee
3  where dept_id in (select dept_id from department where dept_name = 'Executive');
4
```

Below the code, the results tab is selected, showing the output:

DEPT_ID	LAST_NAME	JOB_ID
80	Joe	11

1 rows returned in 0.00 seconds [Download](#)

At the bottom, the footer includes user information (220701062@rajalakshmi.edu.in, euphoria, en), copyright notice (Copyright © 1999, 2025, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Schema WKSP_EUPHORIA Save Run

```

1 select emp_id, last_name, salary
2 from employee
3 where salary > (select avg(salary) from employee) and
4 dept_id in (select dept_id from employee where last_name like '%u%');
5

```

Results Explain Describe Saved SQL History

EMP_ID	LAST_NAME	SALARY
170	puma	20000

1 rows returned in 0.01 seconds Download

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 25.2.4

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

RESULT:

USING THE SET OPERATORS

EX.NO:10

DATE:

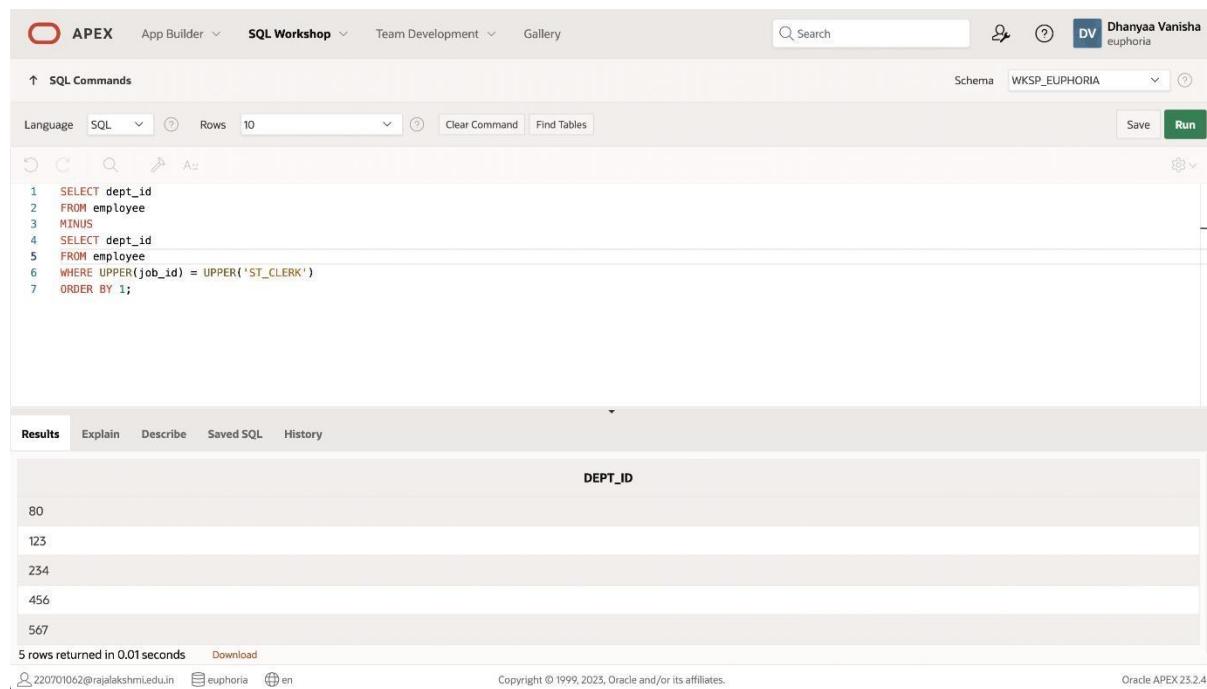
Find the Solution for the following:

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

QUERY:

```
SELECT dept_id
FROM department
MINUS
SELECT dept_id
FROM employees
WHERE job_id = 'st_clerk';
```

OUTPUT:



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

```
1 SELECT dept_id
2 FROM employee
3 MINUS
4 SELECT dept_id
5 FROM employee
6 WHERE UPPER(job_id) = UPPER('ST_CLERK')
7 ORDER BY 1;
```

The results section displays the following data:

DEPT_ID
80
123
234
456
567

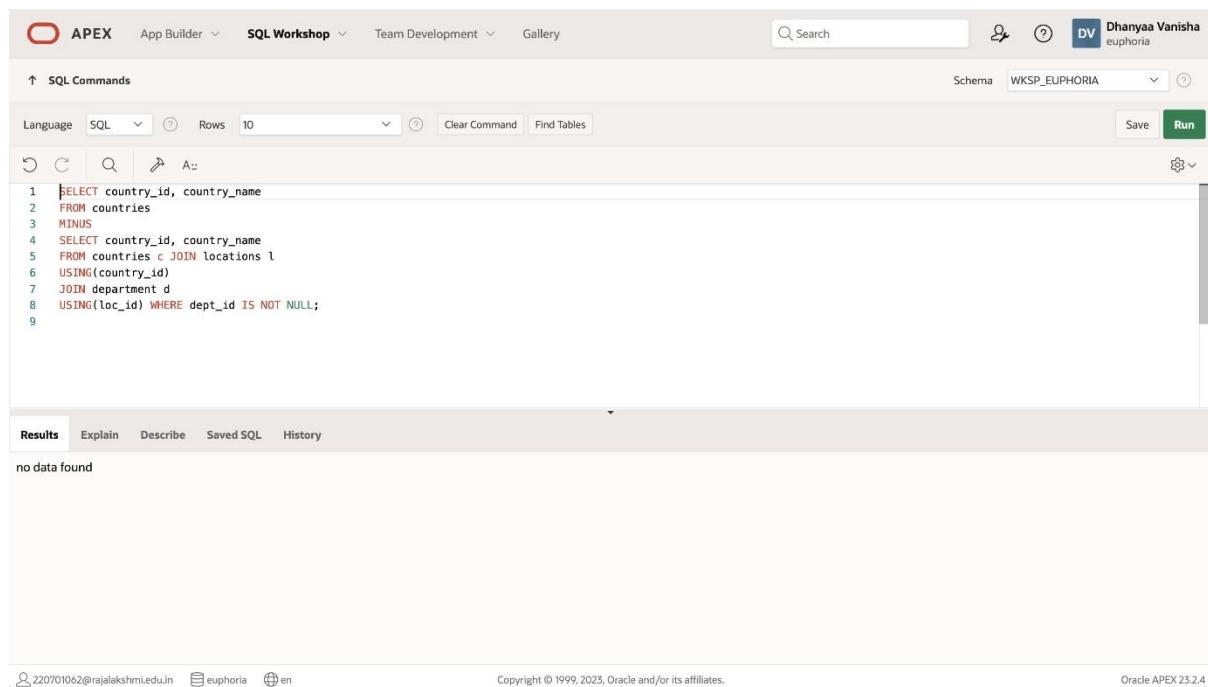
Below the table, it says "5 rows returned in 0.01 seconds".

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

QUERY:

```
SELECT country_id, country_name  
FROM countries  
MINUS  
SELECT l.country_id, c.country_name  
FROM locations l, countries c  
WHERE l.country_id = c.country_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 the user 'Dhanya Vanisha euphoria'. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. Below these are icons for Undo, Redo, Search, and Paste. The SQL code is listed in a code editor:

```
1 SELECT country_id, country_name  
2 FROM countries  
3 MINUS  
4 SELECT country_id, country_name  
5 FROM countries c JOIN locations l  
6 USING(country_id)  
7 JOIN department d  
8 USING(loc_id) WHERE dept_id IS NOT NULL;
```

The results tab is selected at the bottom, showing the message 'no data found'.

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

QUERY:

```
SELECT DISTINCT job_no, dept_id  
FROM employees
```

```
WHERE dept_id = 10
```

UNION ALL

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

UNION ALL

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

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user's name, "Dhanya Vanisha euphoria", is displayed in the top right corner. The main workspace is titled "SQL Commands". The code area contains the following SQL query:

```
1 SELECT DISTINCT job_id, dept_id  
2 FROM employee  
3 WHERE dept_id = 10 UNION ALL  
4 SELECT DISTINCT job_id, dept_id  
5 FROM employee  
6 WHERE dept_id = 50 UNION ALL  
7 SELECT DISTINCT job_id, dept_id  
8 FROM employee  
9 WHERE dept_id = 20;
```

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

JOB_ID	DEPT_ID
62	10
63	50
11	20

The results indicate that there are three rows returned in 0.01 seconds. The bottom of the page shows the user's email (220701062@rajalakshmi.edu.in), the workspace name (euphoria), and the copyright information (Copyright © 1999, 2025, Oracle and/or its affiliates). The Oracle APEX version is also mentioned as 23.2.4.

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

QUERY:

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

OUTPUT:

```

1  SELECT emp_id, job_id
2  FROM employee INTERSECT
3  SELECT emp_id, job_id
4  FROM job_histories|
5  ORDER BY 1;

```

EMP_ID	JOB_ID
170	65
176	62

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,dept_id,TO_CHAR(null)
FROM employees
UNION
SELECT TO_CHAR(null),dept_id,dept_name
FROM department; OUTPUT:

```

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 SELECT last_name, dept_id, TO_CHAR('null')
2 FROM employee UNION
3 SELECT TO_CHAR('null'), dept_id, dept_name
4 FROM department|
5 ORDER BY 1;
```

Results Explain Describe Saved SQL History

LAST_NAME	DEPT_ID	TO_CHAR('NULL')
Adam	10	null
Joe	20	null
Min	456	null
davies	567	null
null	80	Executive
null	177	marketing

220701062@rajalakshmi.edu.in euphoria Oracle APEX 23.2.4

Copyright © 1999, 2025, Oracle and/or its affiliates.

RESULT:

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

CREATING VIEWS

EX_NO:11

DATE:

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

QUERY:

```
CREATE 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. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile for 'Dhanyaa Vanisha euphoria' are also present. The main workspace is titled 'SQL Commands' and contains a code editor with the following SQL statement:

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

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The results section displays the message 'View created.' and a execution time of '0.03 seconds'. At the bottom, footer information includes the user email '220701062@rajalakshmi.edu.in', the workspace name 'euphoria', the language 'en', copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

2.) Display the contents of the EMPLOYEES_VU view.

QUERY:

```
select * from employees_vu;
```

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. A search bar and user profile for 'Dhanyaa Vanisha euphoria' are also present. The main workspace displays a SQL command line with the query 'select * from employees_vu;' and a results grid showing employee data. The bottom footer includes copyright information and the version 'Oracle APEX 23.2.4'.

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
176	Adam	10
170	puma	50
234	Joe	20
46	Min	456

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 (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and session information for "Dhanyaa Vanisha euphoria". The main workspace has tabs for SQL Commands and Results. In the SQL Commands tab, the query `SELECT view_name, text FROM user_views;` is entered. The Results tab displays the output in a table format:

VIEW_NAME	TEXT
EMPLOYEES_VU	SELECT employee_id, last_name employee, department_id FROM employees
VIEW_D_SONGS	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'

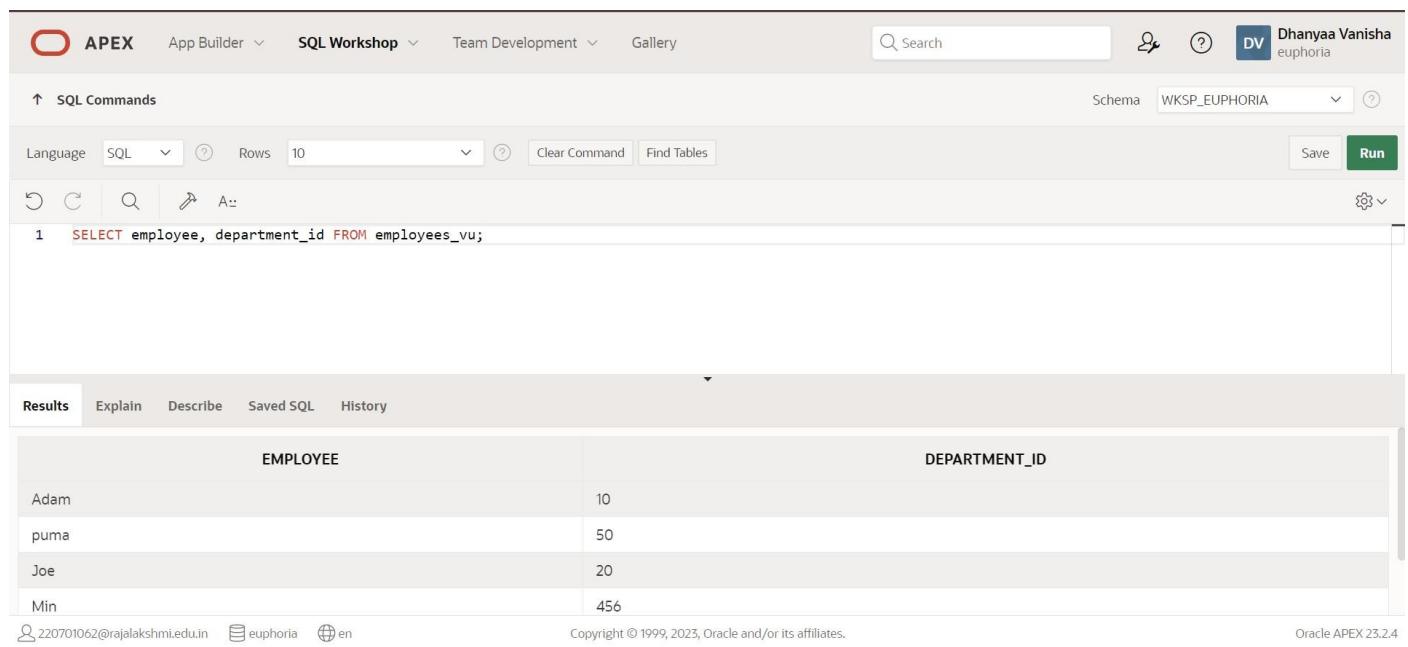
Below the table, it says "2 rows returned in 0.03 seconds" and provides a "Download" link. The bottom of the page shows user information (220701062@rajalakshmi.edu.in, euphoria, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 23.2.4).

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. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile for 'Dhanya Vanisha euphoria', and a 'Run' button. The main workspace is titled 'SQL Commands' and contains the query: 'SELECT employee, department_id FROM employees_vu;'. Below the query, the results are displayed in a table with two columns: 'EMPLOYEE' and 'DEPARTMENT_ID'. The data rows are Adam (10), puma (50), Joe (20), and Min (456). At the bottom left, there are user profile icons and language settings. The bottom right corner indicates the version: Oracle APEX 23.2.4.

EMPLOYEE	DEPARTMENT_ID
Adam	10
puma	50
Joe	20
Min	456

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 links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is signed in as 'Dhanyaa Vanisha' with the profile icon 'DV'. The main workspace is titled 'SQL Commands' and shows the following SQL code:

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

Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the message 'View created.' and a execution time of '0.02 seconds'. At the bottom of the page, there are footer links for '220701062@rajalakshmi.edu.in', 'euphoria', 'en', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

6.)

Display the structure and contents of the DEPT50 view.

QUERY:

Describe dept50;

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. A search bar and user profile 'Dhanya Vanisha euphoria' are also present. The main workspace shows a SQL command entry area with the query '1 Describe dept50;'. Below this, the 'Describe' tab is selected in a tab bar along with Explain, Saved SQL, and History. The object type is set to VIEW. The results table displays three columns: EMPNO, EMPLOYEE, and DEPTNO, each with their respective data types, lengths, precision, scale, primary key status, nullable, default, and comments.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT50	EMPNO	NUMBER	-	7	0	-	✓	-	-
	EMPLOYEE	VARCHAR2	25	-	-	-	✓	-	-
	DEPTNO	NUMBER	-	6	0	-	✓	-	-

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 links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user information for 'Dhanyaa Vanisha euphoria', and a 'Run' button. The main workspace is titled 'SQL Commands' and contains the following content:

```
Language SQL Rows 10 Clear Command Find Tables
1 UPDATE dept50 SET deptno=80 WHERE employee='Matos';
```

Below the command input area, there are several icons: Undo, Redo, Search, Insert, and others. The results section at the bottom shows the output of the query:

0 row(s) updated.
0.01 seconds

At the very bottom of the page, there are footer links for user information (220701062@rajalakshmi.edu.in), a logo for 'euphoria', and copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates). The footer also indicates the version 'Oracle APEX 23.2.4'.

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 links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile 'Dhanyaa Vanisha euphoria' are also present. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 create or replace view salary_vu as select e.last_name "Employee",d.dept_name Department,
2 e.salary "Salary",j.grade_level "Grades"
3 from employees e,departments d,job_grade j where e.department_id=d.dept_id and e.salary
4 between j.lowest_sal and j.highest_sal;
```

Below the code, the 'Results' tab is selected, showing the output: 'View created.' and '0.02 seconds'. The bottom footer includes user information (220701062@rajalakshmi.edu.in, euphoria, en) and copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates). The version 'Oracle APEX 23.2.4' is also mentioned.

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

RESULT:

INTRO TO CONSTRAINTS: NOT NULL AND UNIQUE CONSTRAINTS

EX-NO : 12

DATE:

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

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

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

Ans:

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

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

Ans:

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

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

Ans:

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

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

Ans:

Global Fast Foods global_locations Table							
NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE	
id	pk	NUMBER	6	0	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	0		
emergency_contact		VARCHAR2	20				

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

Ans:

Global Fast Foods global_locations Table							
NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE	
id	pk	NUMBER	6	0	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	0	Yes	
emergency_contact		VARCHAR2	20			Yes	

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

Ans:

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

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

Ans:

Table Created.

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

Ans:

```
DESCRIBE f_global_locations;
```

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

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

Ans:

```

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

```

PRIMARY KEY, FOREIGN KEY, AND CHECK CONSTRAINTS

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

Ans:

a. PRIMARY KEY

Uniquely identify each row in table.

b. FOREIGN KEY

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

c. CHECK CONSTRAINT

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

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

Ans:

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

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

Ans:

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

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

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

Ans:

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

```
SELECT * FROM animals;
```

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

Ans:

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

a. ON DELETE CASCADE

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

ON DELETE SET NULL

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

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

Ans:

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

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

RESULT :

INTRO TO CONSTRAINTS: NOT NULL AND UNIQUE CONSTRAINTS

EX-NO : 12

DATE:

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

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

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

Ans:

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

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

Ans:

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

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

Ans:

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

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

Ans:

Global Fast Foods global_locations Table							
NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE	
id	pk	NUMBER	6	0	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	0		
emergency_contact		VARCHAR2	20				

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

Ans:

Global Fast Foods global_locations Table							
NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE	
id	pk	NUMBER	6	0	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	0	Yes	
emergency_contact		VARCHAR2	20			Yes	

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

Ans:

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

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

Ans:

Table Created.

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

Ans:

```
DESCRIBE f_global_locations;
```

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

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

Ans:

```

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

```

PRIMARY KEY, FOREIGN KEY, AND CHECK CONSTRAINTS

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

Ans:

a. PRIMARY KEY

Uniquely identify each row in table.

b. FOREIGN KEY

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

c. CHECK CONSTRAINT

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

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

Ans:

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

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

Ans:

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

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

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

Ans:

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

```
SELECT * FROM animals;
```

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

Ans:

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

a. ON DELETE CASCADE

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

ON DELETE SET NULL

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

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

Ans:

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

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

RESULT :

EXERCISE 13

Creating Views

1. What are three uses for a view from a DBA's perspective?
 - Restrict access and display selective columns
 - Reduce complexity of queries from other internal systems. So, providing a way to view same data in a different manner.
 - Let the app code rely on views and allow the internal implementation of tables to be modified later.
2. Create a simple view called view_d_songs that contains the ID, title and artist from the DJs on Demand table for each "New Age" type code. In the subquery, use the alias "Song Title" for the title column.

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

3. SELECT * FROM view_d_songs. What was returned?

Results	Explain	Describe	Saved SQL	History
ID	Song Title	ARTIST		
47	Hurrah for Today	The Jubilant Trio		
49	Lets Celebrate	The Celebrants		

2 rows returned in 0.00 seconds [Download](#)

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

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

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

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

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

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

DML Operations and Views

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

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

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

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

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

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

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

```
SELECT * FROM view_copy_d_songs;
```

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

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

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

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

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

```
SELECT *
FROM copy_d_cds
WHERE year = '2000'
WITH READ ONLY ;
```

```
SELECT * FROM read_copy_d_cds;
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

DELETE,INSERT,MODIFY restricted if it contains:

Group functions
GROUP BY CLAUSE DISTINCT
pseudocolumn ROWNUM Keyword

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

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

13. What is the “singularity” in terms of computing?

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

Managing Views

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

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

```
SELECT * FROM view_copy_d_songs;
```

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

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

ORA-00942: table or view does not exist

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

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

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

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

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

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

Indexes and Synonyms

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

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

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

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

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

3. When will an index be created automatically?

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

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

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

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

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

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

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

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

```
CREATE SYNONYM dj_tracks FOR d_track_listings;
```

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

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

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

```
CREATE SYNONYM dj_tracks2 FOR d_track_listings;
```

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

10. Drop the synonym that you created in question

```
DROP SYNONYM dj_tracks2;
```

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

RESULT:

OTHER DATABASE OBJECTS

EX_NO:14

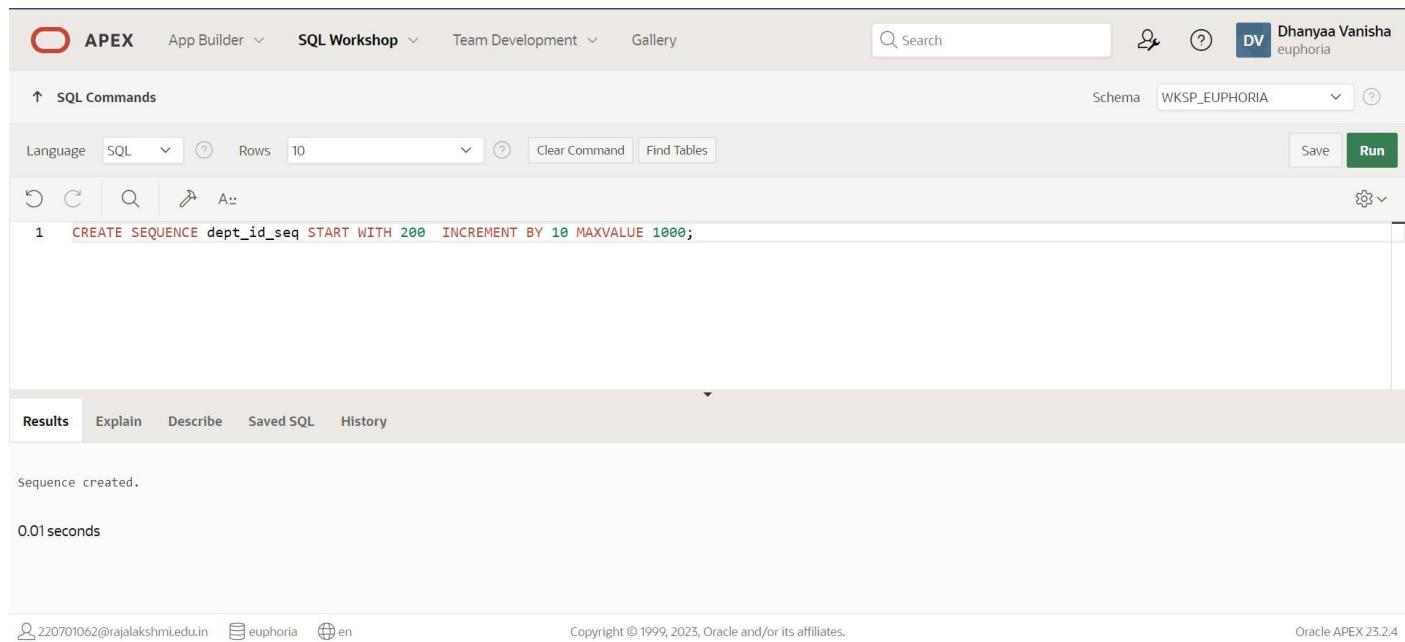
DATE:

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 links for APEX, App Builder, SQL Workshop, Team Development, and Gallery, along with a search bar and user information for 'Dhanyaa Vanisha'.

In the SQL Commands tab, the schema is set to 'WKSP_EUPHORIA'. The query entered is:

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

The results section shows the output:

Sequence created.
0.01 seconds

At the bottom, footer information includes the user email '220701062@rajalakshmi.edu.in', the workspace name 'euphoria', the language 'en', 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. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', 'Gallery', a search bar, and a user profile for 'Dhanya Vanisha euphoria'. Below the navigation is a toolbar with icons for 'SQL Commands', 'Language' (set to SQL), 'Rows' (set to 10), 'Clear Command', 'Find Tables', 'Save', and 'Run'. The main area contains the SQL command: '1 SELECT sequence_name, max_value, increment_by, last_number FROM user_sequences;'. The results section shows a single row of data:

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_ID_SEQ	1000	10	200

Below the results, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom, it shows the user's email (220701062@rajalakshmi.edu.in), the workspace ('euphoria'), and the language ('en'). Copyright information and the APEX version (23.2.4) are also at the bottom.

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, identical to the previous one but with an error message. The results section displays the following error:

```
Error at line 1/13: ORA-00947: not enough values
1. INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education');
```

At the bottom, it shows '0.01seconds', the user's email, workspace, language, copyright information, and the APEX version (23.2.4).

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 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows the user 'Dhanya Vanisha' and the schema 'WKSP_EUPHORIA'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below the dropdown are buttons for 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. To the right are 'Save' and 'Run' buttons. The SQL command 'CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);' is entered in the text area. The results section below shows the output: 'Index created.', execution time '0.03 seconds', and copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' The bottom right corner indicates 'Oracle APEX 23.2.4'.

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 and user information are the same as the previous screenshot. The main area shows the SQL command 'SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';' in the 'SQL Commands' section. The results section displays a table with three columns: 'INDEX_NAME', 'TABLE_NAME', and 'UNIQUENESS'. The data row is 'EMP_DEPT_ID_IDX', 'EMPLOYEES', and 'NONUNIQUE'. Below the table, it says '1 rows returned in 0.00 seconds'. The bottom of the page includes copyright information and 'Oracle APEX 23.2.4'.

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

RESULT:

USER

EX_NO:15

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

**CONTROLLING
ACCESS**

DATE:

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

The CREATE SESSION system privilege

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

The CREATE TABLE privilege

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

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

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

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

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

The ALTER USER statement

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

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

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

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

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

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

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

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

SELECT table_name FROM user_tables;

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

Team 1 revokes the privilege.

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

Team 2 revokes the privilege.

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

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

Team 1 executes this INSERT statement.

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

Team 2 executes this INSERT statement.

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

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

RESULT:

PL/SQL

CONTROL STRUCTURES

EX_NO:

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Dhanya Vanisha' and a schema dropdown set to 'WKSP_EUPHORIA'. The main workspace is titled 'SQL Commands' and contains a code editor with 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;
9
```

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

```
Incentive = 1440
Statement processed.
```

At the bottom, the footer includes copyright information for Oracle and links for user profile, language, and help.

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. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there is a search bar, user profile for 'Dhanya Vanisha euphoria', and a 'Run' button. The main workspace displays the following PL/SQL code:

```
1  DECLARE
2  WELCOME varchar2(10) := 'welcome';
3  BEGIN
4  DBMS_OUTPUT.PUT_LINE('Welcome');
5  END;
6
```

Below the code, the results tab is selected, showing the output:

```
Welcome
Statement processed.
```

Execution time is listed as 0.01 seconds. The bottom footer includes copyright information and links for user profile, language, and help.

3.)

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

QUERY:

DECLARE

```
salary_of_emp NUMBER(8,2);
```

```
PROCEDURE approx_salary (
```

```
    emp      NUMBER,
```

```
    empsal IN OUT NUMBER,
```

```
    addless   NUMBER
```

```
) IS
```

```
BEGIN
```

```
    empsal := empsal + addless;
```

```
END;
```

BEGIN

```
SELECT salary INTO salary_of_emp
```

```
FROM employees
```

```
WHERE employee_id = 122;
```

```
DBMS_OUTPUT.PUT_LINE
```

```
('Before invoking procedure, salary_of_emp: ' || salary_of_emp);
```

```
approx_salary (100, salary_of_emp, 1000);
```

```
DBMS_OUTPUT.PUT_LINE
```

```
('After invoking procedure, salary_of_emp: ' || salary_of_emp);
```

```
END;
```

```
/
```

OUTPUT:

```
APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_EUPHORIA Run
↑ SQL Commands Language PL/SQL Rows 10 Clear Command Find Tables Save Run
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
    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;
/

```

Results Explain Describe Saved SQL History

Before invoking procedure, salary_of_emp: 12000
After invoking procedure, salary_of_emp: 13000

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 links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile 'Dhanya Vanisha euphoria' are also present. The main workspace is titled 'SQL Commands' and shows the PL/SQL code for the 'pri_bool' procedure. The code uses the DBMS_OUTPUT package to print the value of 'boo_name'. The procedure checks if 'boo_val' is NULL, TRUE, or FALSE and prints the corresponding message. The code is numbered from 1 to 14. Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The results tab shows the message 'Procedure created.' and execution details: 0.02 seconds, user 220701062@rajalakshmi.edu.in, session euphoria, and language en. The footer includes copyright information for Oracle and the APEX version 23.2.4.

```
3 |   boo_val   BOOLEAN
4 | ) IS
5 | BEGIN
6 |   IF boo_val IS NULL THEN
7 |     DBMS_OUTPUT.PUT_LINE( boo_name || ' = NULL');
8 |   ELSIF boo_val = TRUE THEN
9 |     DBMS_OUTPUT.PUT_LINE( boo_name || ' = TRUE');
10 |   ELSE
11 |     DBMS_OUTPUT.PUT_LINE( boo_name || ' = FALSE');
12 |   END IF;
13 | END;
14 |
```

Procedure created.

0.02 seconds
220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

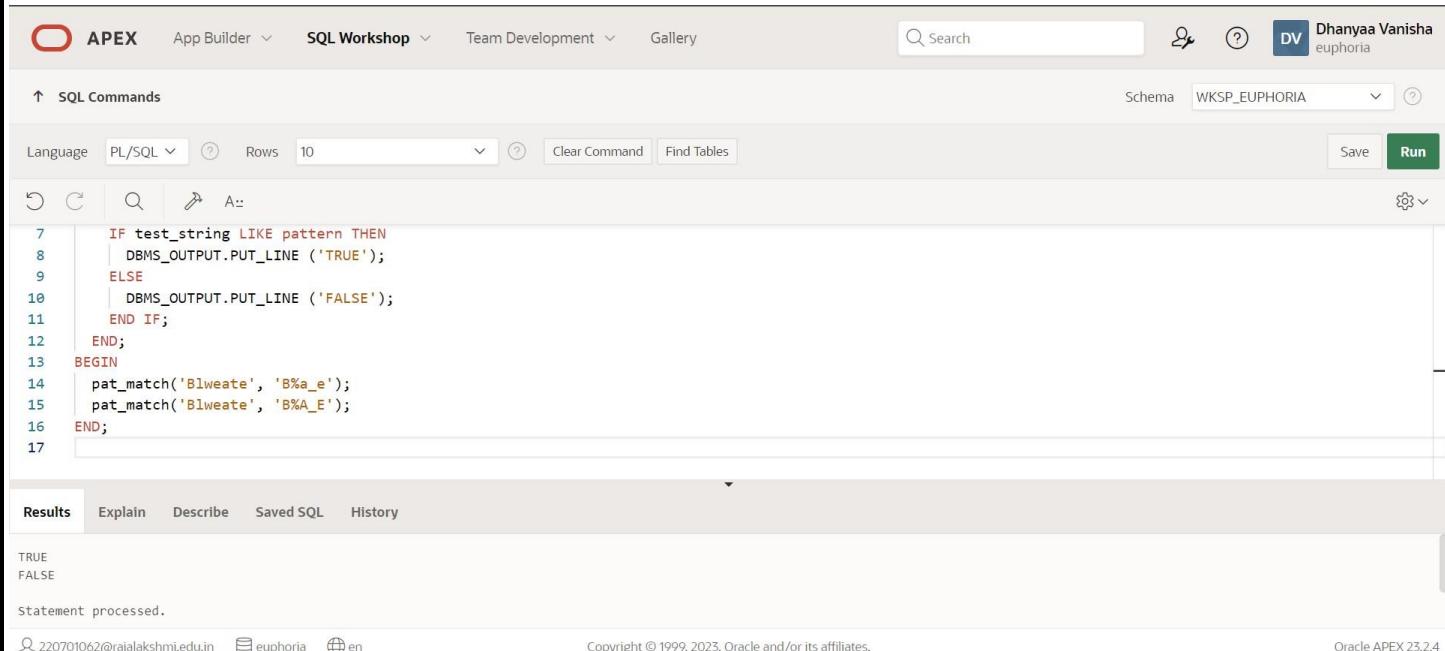
5.)

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, Team Development, and Gallery. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. The main area is titled 'SQL Commands' and contains a code editor with the following PL/SQL block:

```
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
```

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

```
TRUE
FALSE
```

At the bottom, it says 'Statement processed.' and includes copyright information for Oracle and links for user 220701062@rajalakshmi.edu.in, euphoria, and en.

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. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains a code editor with the following PL/SQL block:

```
6
7  IF num_small > num_large THEN
8      num_temp := num_small;
9      num_small := num_large;
10     num_large := num_temp;
11  END IF;
12
13  DBMS_OUTPUT.PUT_LINE ('num_small = '||num_small);
14  DBMS_OUTPUT.PUT_LINE ('num_large = '||num_large);
15
16
```

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

```
num_small = 5
num_large = 8

Statement processed.
```

At the bottom, the footer includes copyright information for Oracle and links for user profile, language, and help.

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:

```

15   WHERE employees_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;
23 BEGIN
24   test1(2300, 144);
25   test1(3600, 3000, 145);
26 END;

```

Results Explain Describe Saved SQL History

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

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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;
/

```

9.)

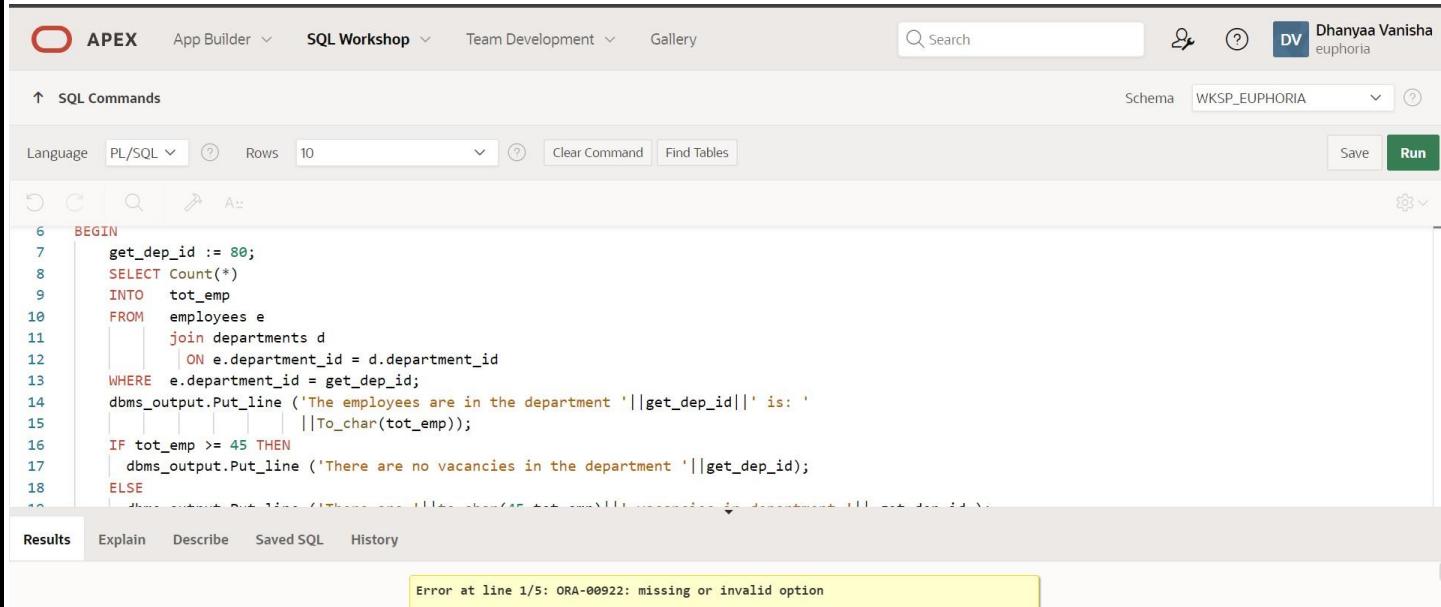
Write a PL/SQL program to count number of employees in department 50 and check whether this department have any vacancies or not. There are 45 vacancies in this department.

QUERY:

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

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains a code editor with the following PL/SQL script:

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

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The 'Results' tab is selected. A yellow box highlights an error message: 'Error at line 1/5: ORA-00992: missing or invalid option'. At the bottom of the page, there are footer links for user information (220701062@rajalakshmi.edu.in), language (euphoria), and page footer (Copyright © 1999, 2023, Oracle and/or its affiliates). The page also indicates it is Oracle APEX 23.2.4.

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 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 'Dhanya Vanisha' with the schema 'WKSP_EUPHORIA'. The main workspace is titled 'SQL Commands' and contains the following PL/SQL code:

```
7  SELECT Count(*)  
8  INTO tot_emp  
9  FROM employees e  
10 join departments 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
```

The code is executed, and the results pane displays the output:

```
The employees are in the department 80 is: 0  
There are 45 vacancies in department 80  
Statement processed.
```

11.)

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

QUERY:

DECLARE

v_employee_id employees.employee_id%TYPE;

v_full_name employees.first_name%TYPE;

v_job_id employees.job_id%TYPE;

v_hire_date employees.hire_date%TYPE;

v_salary employees.salary%TYPE;

CURSOR c_employees IS

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

BEGIN

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

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

OPEN c_employees;

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

WHILE c_employees%FOUND LOOP

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

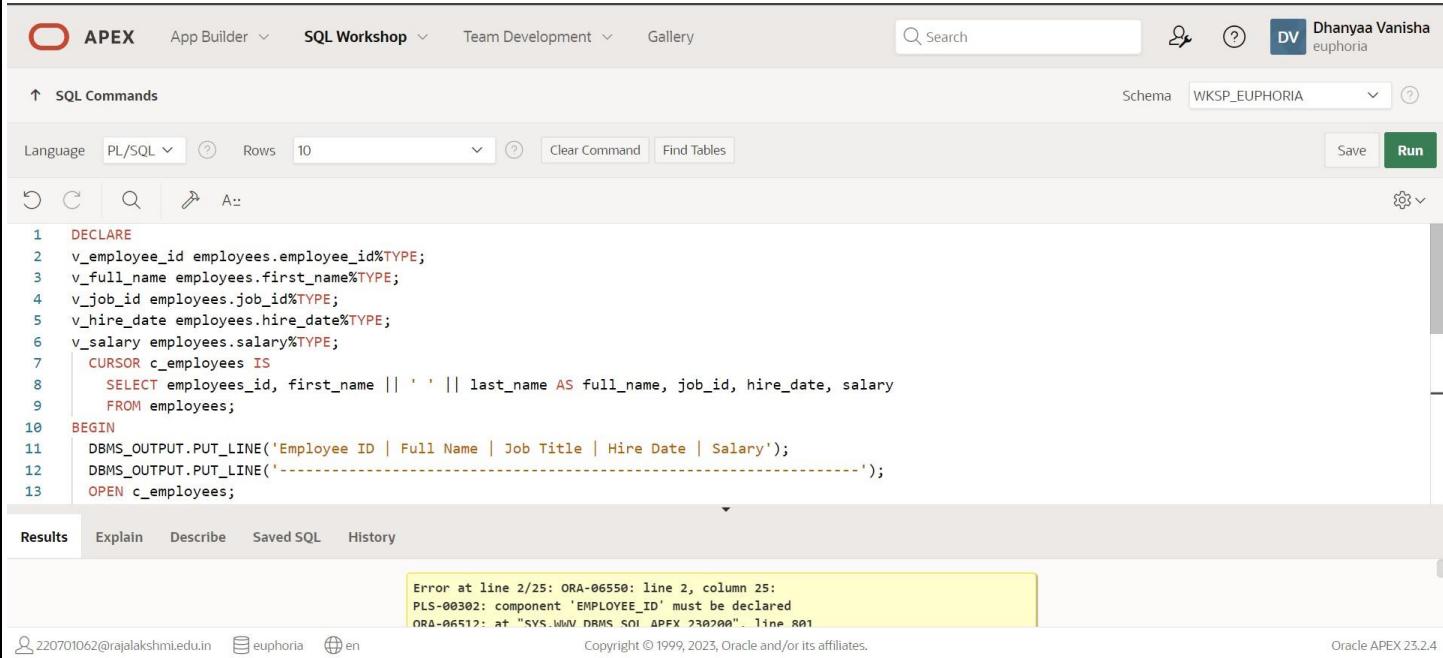
FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;

END LOOP;

CLOSE c_employees;

END;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The code area contains the PL/SQL block provided above. Line 25, which contains the declaration of 'v_employee_id', is highlighted in yellow, indicating an error. A tooltip at the bottom of the code area displays the error message: "Error at line 2/25: ORA-06550: line 2, column 25: PLS-00302: component 'EMPLOYEE_ID' must be declared ORA-06512: at "SYS.WWV_DRMS_SOI_APFX_23020". line 801". The results tab at the bottom shows the error message again.

```
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;
```

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 (selected), Team Development, and Gallery. A search bar and user profile 'Dhanya Vanisha euphoria' are also present. The main workspace displays a PL/SQL code editor with the following content:

```
6  BEGIN
7    OPEN job_cursor;
8    FETCH job_cursor INTO job_record;
9    WHILE job_cursor%FOUND LOOP
10      DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_record.job_id);
11      DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_record.lowest_sal);
12      DBMS_OUTPUT.PUT_LINE('-----');
13      FETCH job_cursor INTO job_record;
14    END LOOP;
15    CLOSE job_cursor;
16  END;
17  |
```

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

```
Job ID: 62
Minimum Salary:
-----
```

At the bottom of the interface, footer information includes the URL '220701062@rajalakshmi.edu.in', the theme 'euphoria', language 'en', copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

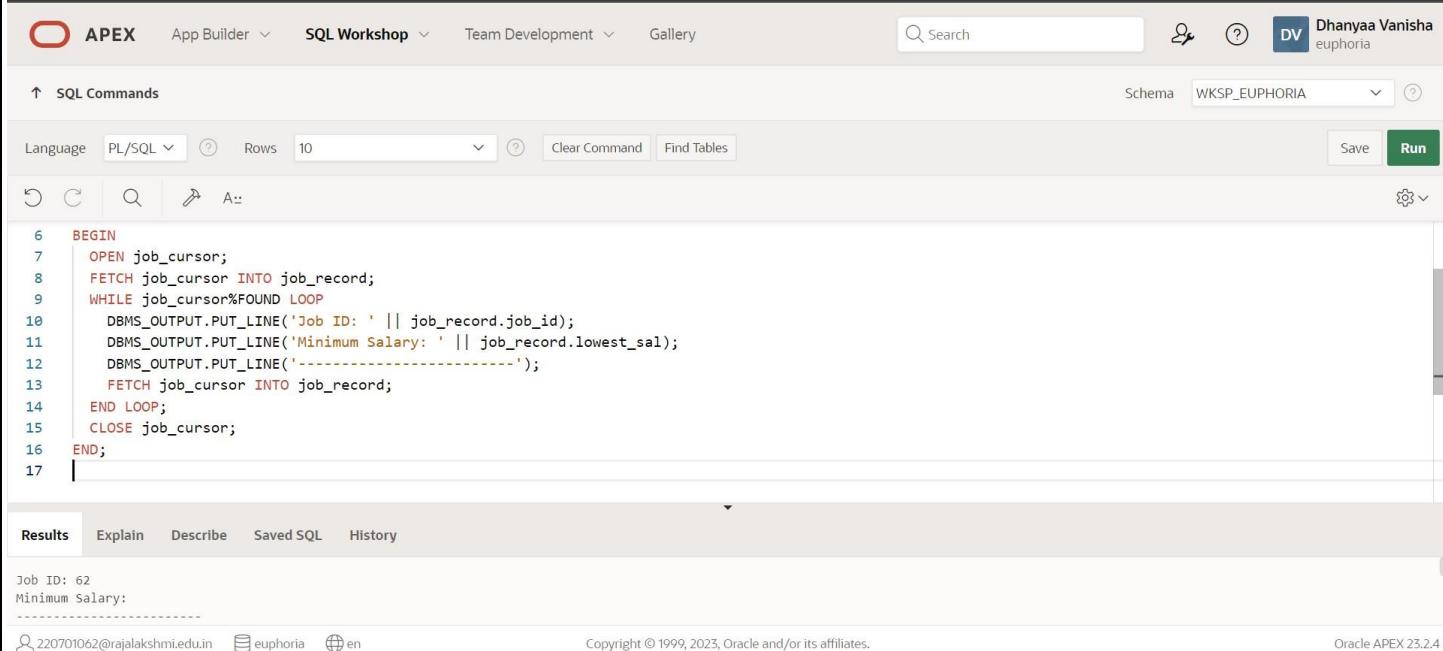
13.)

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

QUERY:

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

OUTPUT:



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's a user profile for 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code from the previous section. The code is numbered 6 to 17. Below the code, the 'Results' tab is selected, showing the output: 'Job ID: 62' and 'Minimum Salary:'. The bottom footer includes copyright information for Oracle APEX 23.2.4 and user details.

```
6 BEGIN
7   OPEN job_cursor;
8   FETCH job_cursor INTO job_record;
9   WHILE job_cursor%FOUND LOOP
10     DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_record.job_id);
11     DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_record.lowest_sal);
12     DBMS_OUTPUT.PUT_LINE('-----');
13     FETCH job_cursor INTO job_record;
14   END LOOP;
15   CLOSE job_cursor;
16 END;
17 |
```

Results Explain Describe Saved SQL History

Job ID: 62
Minimum Salary:

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

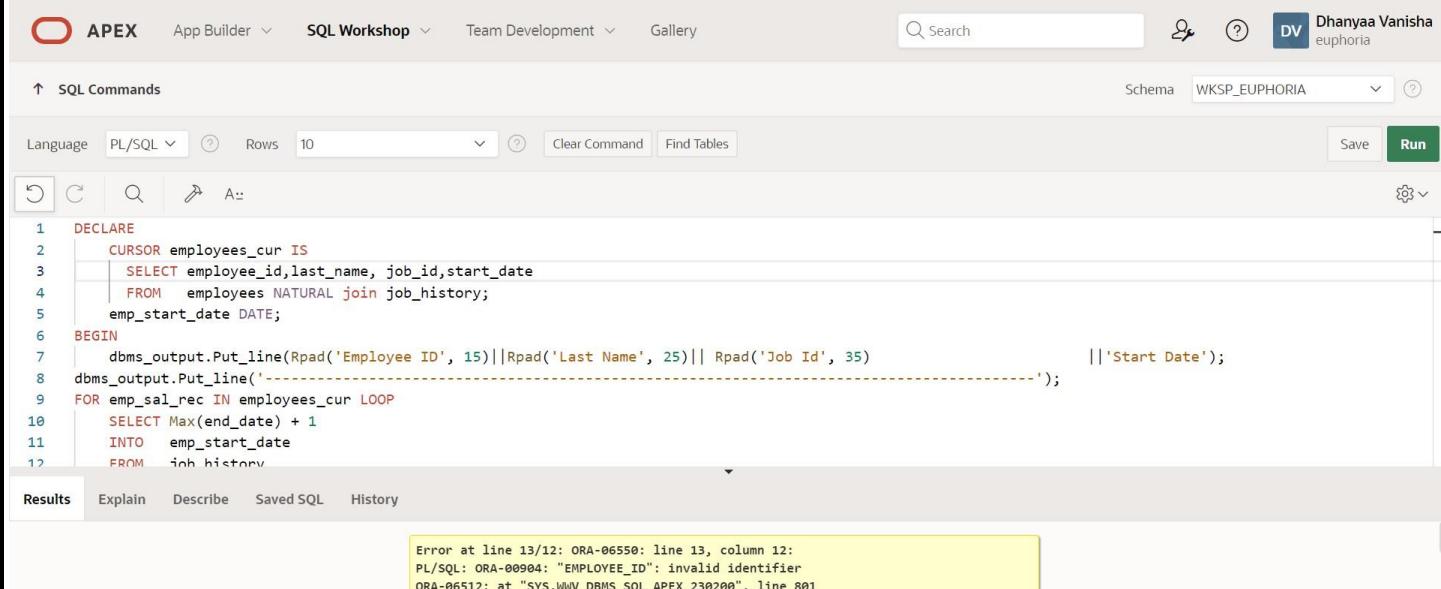
14.) Write a PL/SQL program to display the employee IDs, names, and job history start dates of all employees.

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 APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (which is selected), 'Team Development', 'Gallery', and a user profile for 'Dhanya Varsha euphoria'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code provided above. Below the code, the 'Results' tab is active, displaying an error message in a yellow box: 'Error at line 13/12: ORA-06550: line 13, column 12: PL/SQL: ORA-00904: "EMPLOYEE_ID": invalid identifier ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_230200", line 801 ORA-06550: line 10, column 5:'. The bottom of the screen shows the footer with user information '220701062@rajalakshmi.edu.in euphoria en', copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and version 'Oracle APEX 23.2.4'.

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:

```
APEX App Builder SQL Workshop Team Development Gallery Search DV Dhanya Vanisha euphoria
↑ SQL Commands Schema WKSP_EUPHORIA Save Run
Language PL/SQL Rows 10 Clear Command Find Tables
① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬
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 employees_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;
```

Error at line 2/25: ORA-06550: line 2, column 25:
PLS-00302: component 'EMPLOYEE_ID' must be declared
ORA-06512: at "SYS.WWV_NRMS_S01_APPFX_230700", line 801

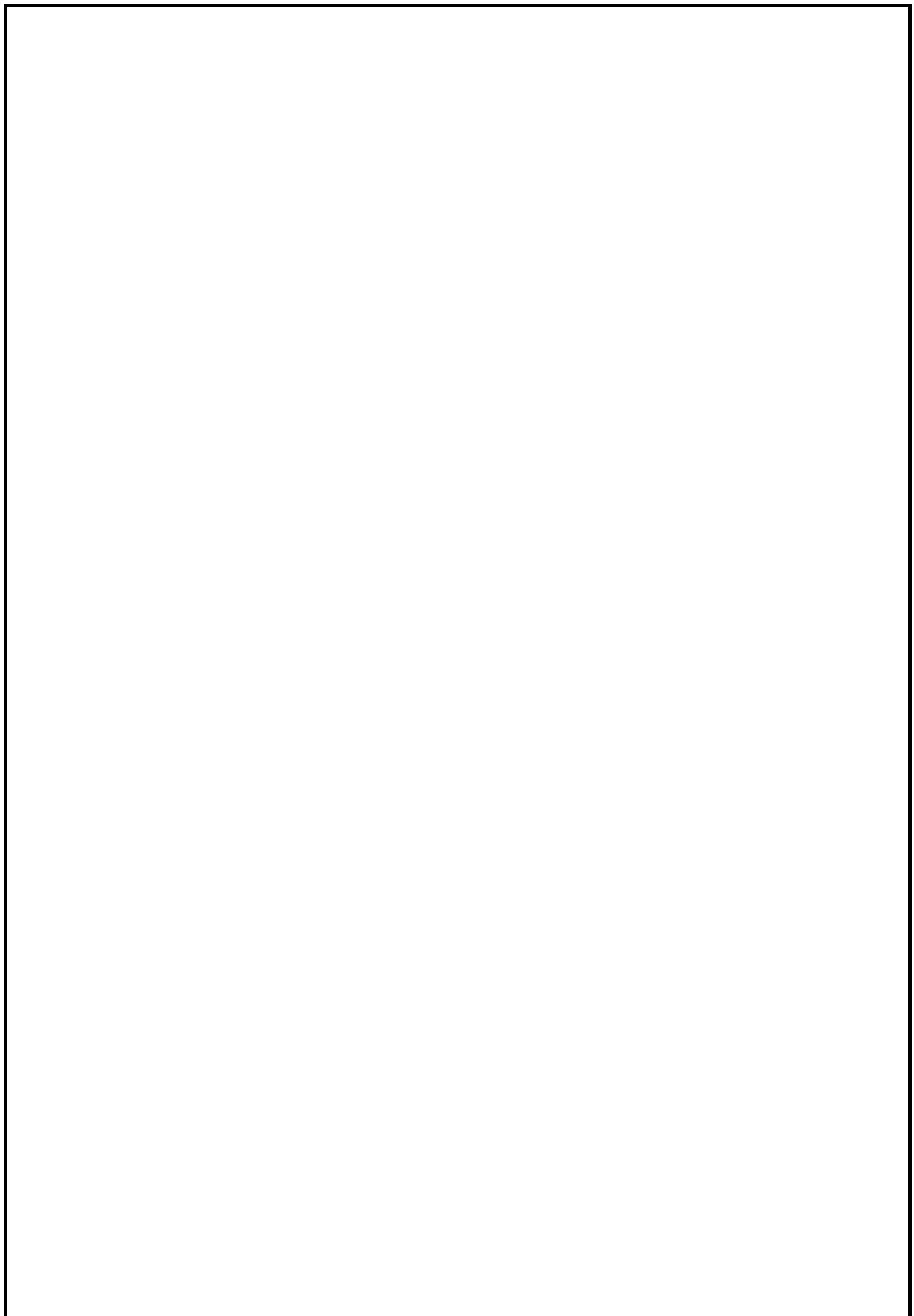
Results Explain Describe Saved SQL History

220701062@rajalakshmi.edu.in euphoria en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4



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

RESULT:





PROCEDURES AND FUNCTIONS

EX_NO: 17

DATE:

1.) Factorial of a number using function.

QUERY:

DECLARE

```
1 fac NUMBER := 1;
```

```
2 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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is identified as 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains the following PL/SQL code:

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

```
479001600
Statement processed.
```

At the bottom of the page, footer information includes the email '220701062@rajalakshmi.edu.in', the domain 'euphoria', and the language 'en'. Copyright information from Oracle is also present.

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:

Screenshot of Oracle APEX SQL Workshop interface showing a PL/SQL procedure code and its execution results.

The code in the editor is:

```
21  v_title VARCHAR2(100);
22  v_author VARCHAR2(100);
23  v_year_published NUMBER;
24  BEGIN
25    v_title := 'Initial Title';
26    get_book_info(p_book_id => v_book_id, p_title => v_title, p_author => v_author,
27    p_year_published => v_year_published);
28    DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
29    DBMS_OUTPUT.PUT_LINE('Author: ' || v_author);
30    DBMS_OUTPUT.PUT_LINE('Year Published: ' || v_year_published);
31  END;
```

The results tab shows the following output:

```
Error at line 19: PLS-00103: Encountered the symbol "DECLARE"
1. CREATE OR REPLACE PROCEDURE get_book_info (
```

Details from the bottom of the page:

- User: 220701062@rajalakshmi.edu.in
- Schema: euphoria
- Language: en
- Copyright © 1999, 2023, Oracle and/or its affiliates.
- Oracle APEX 23.2.4

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

RESULT:

Faculty Signature

TRIGGER

EX_NO: 18

DATE:

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

QUERY:

```
CREATE OR REPLACE TRIGGER prevent_parent_deletion
```

```
BEFORE DELETE ON parent_table
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    child_exists EXCEPTION;
```

```
    PRAGMA EXCEPTION_INIT(child_exists, -20001);
```

```
    v_child_count NUMBER;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id =  
    :OLD.parent_id;
```

```
    IF v_child_count > 0 THEN
```

```
        RAISE child_exists;
```

```
    END IF;
```

```
EXCEPTION
```

```
    WHEN child_exists THEN
```

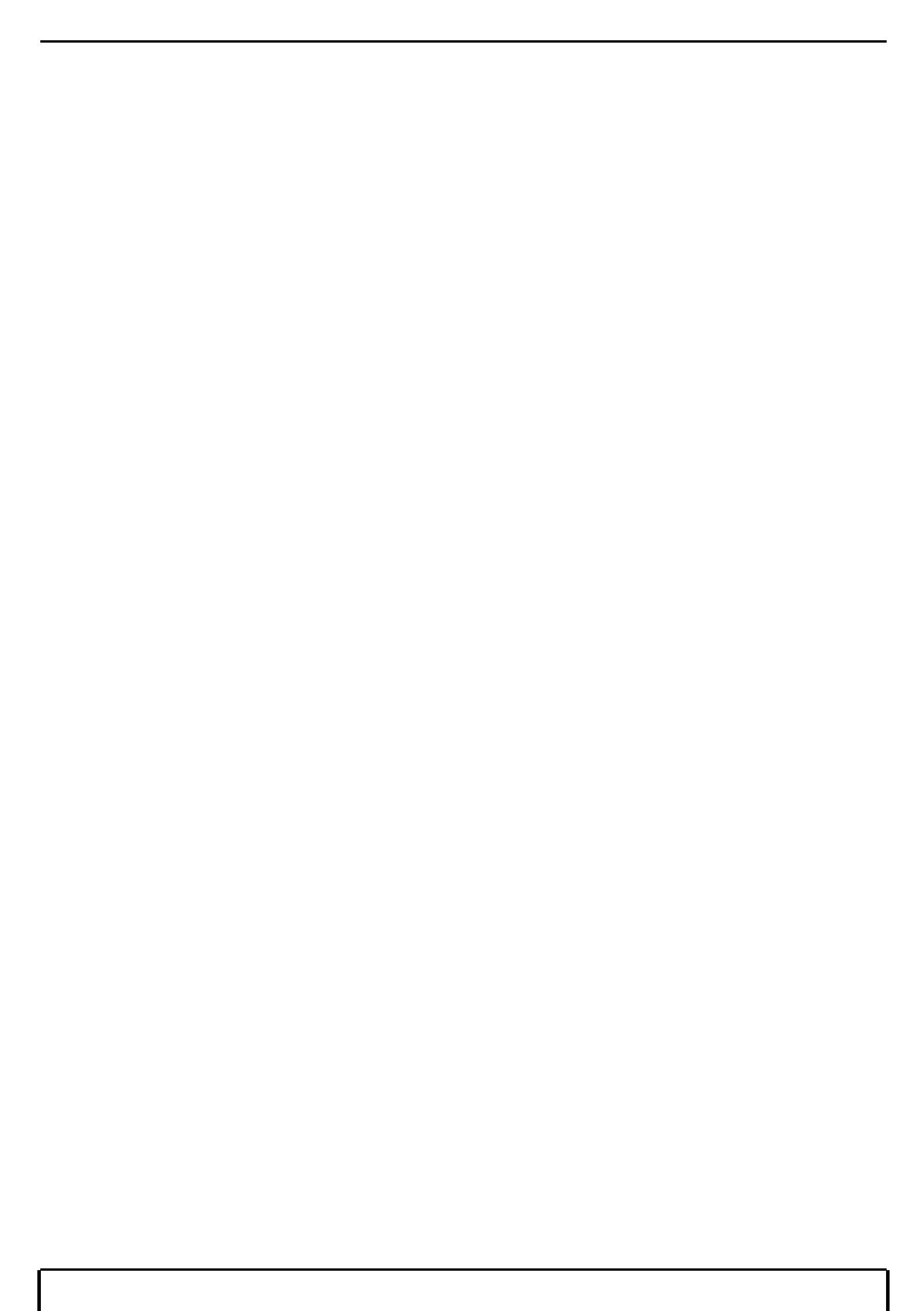
```
        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child records  
exist.');
```

```
END;
```

OUTPUT:

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 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is highlighted in red and green. Below the code, the 'Results' tab is selected, showing the output: 'Trigger created.' and '0.04 seconds'. The bottom footer includes copyright information for Oracle and the APEX version.

```
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;
17 /
```



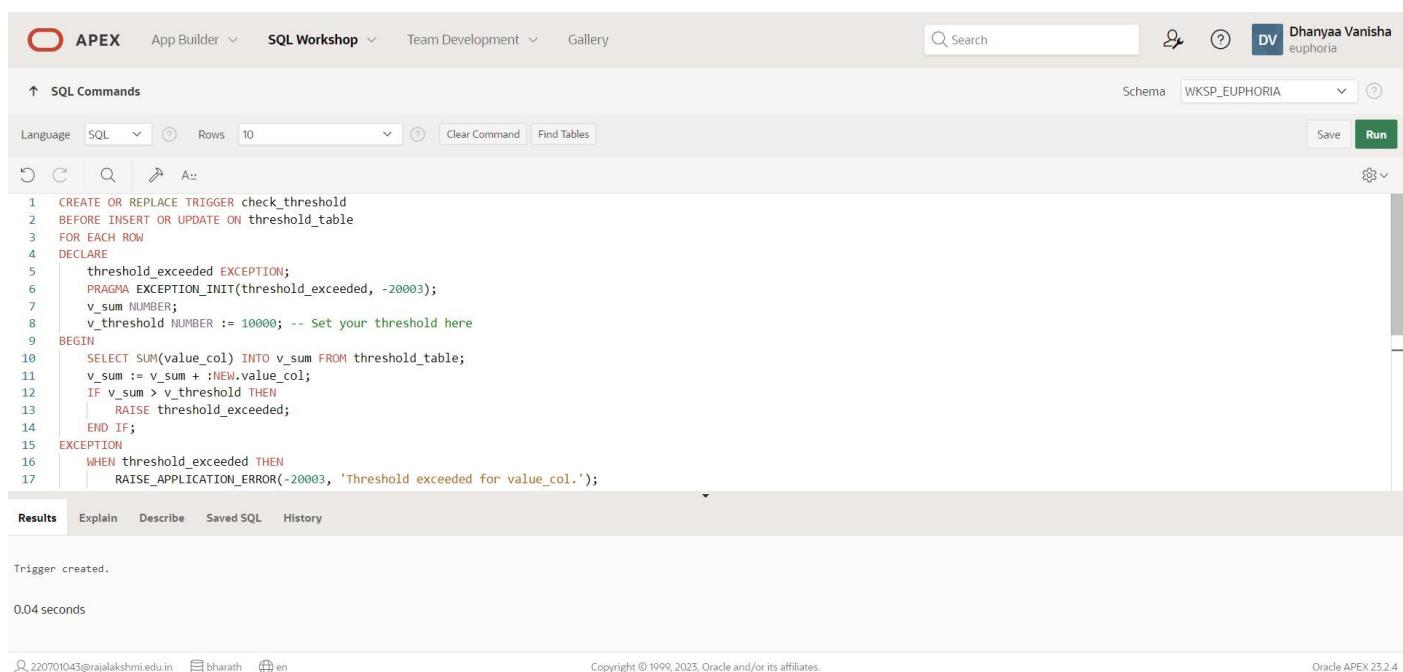
3.)

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 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 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is highlighted in red and orange. Below the code, the 'Results' tab is selected, showing the output: 'Trigger created.' and '0.04 seconds'. At the bottom, footer information includes the URL '220701043@rajalakshmi.edu.in', the name 'bharath', and the page number 'en'. The copyright notice reads 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

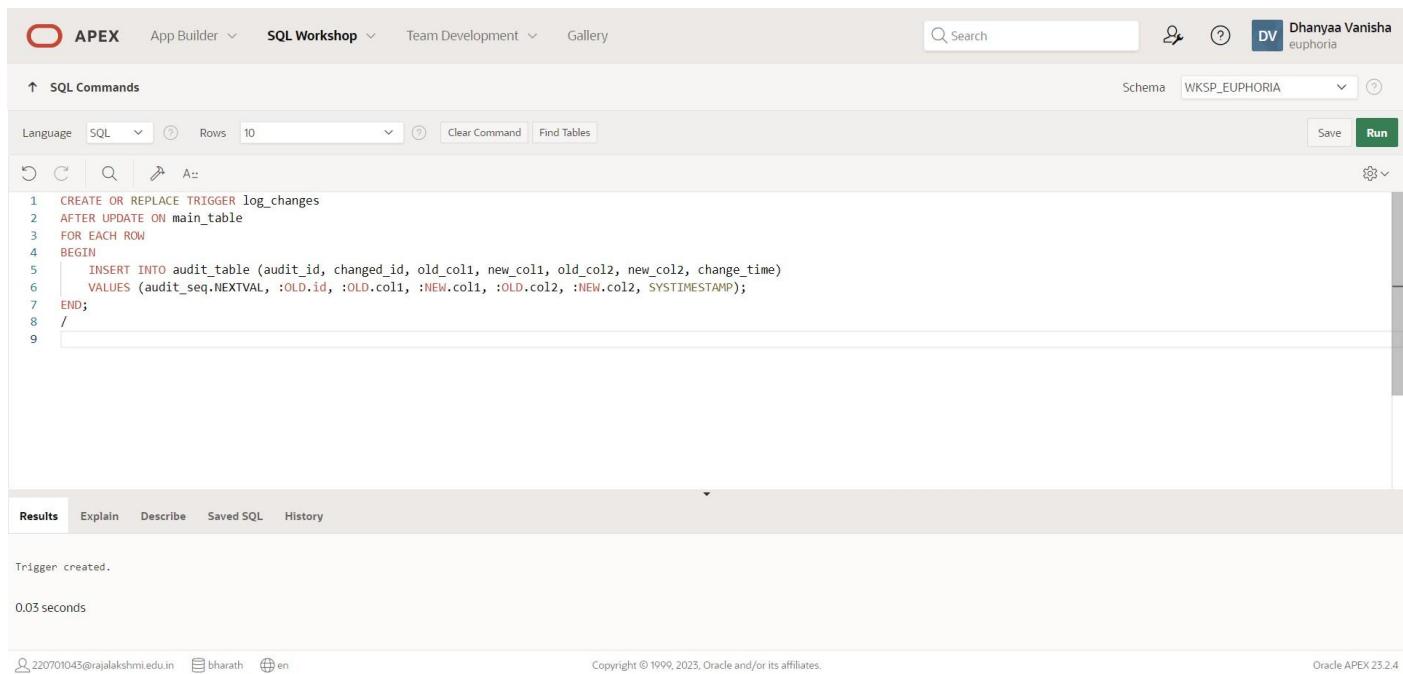
```
1 CREATE OR REPLACE TRIGGER check_threshold
2 BEFORE INSERT OR UPDATE ON threshold_table
3 FOR EACH ROW
4 DECLARE
5     threshold_exceeded EXCEPTION;
6     PRAGMA EXCEPTION_INIT(threshold_exceeded, -20003);
7     v_sum NUMBER;
8     v_threshold NUMBER := 10000; -- Set your threshold here
9 BEGIN
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.');
```

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for creating the 'log_changes' trigger. 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,
6 change_time)
7     VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2, :NEW.col2,
8 SYSTIMESTAMP);
9 END;
9 /
```

Below the code, the 'Results' tab is selected, showing the message 'Trigger created.' and a execution time of '0.03 seconds'. The bottom footer includes copyright information for Oracle and the APEX version.

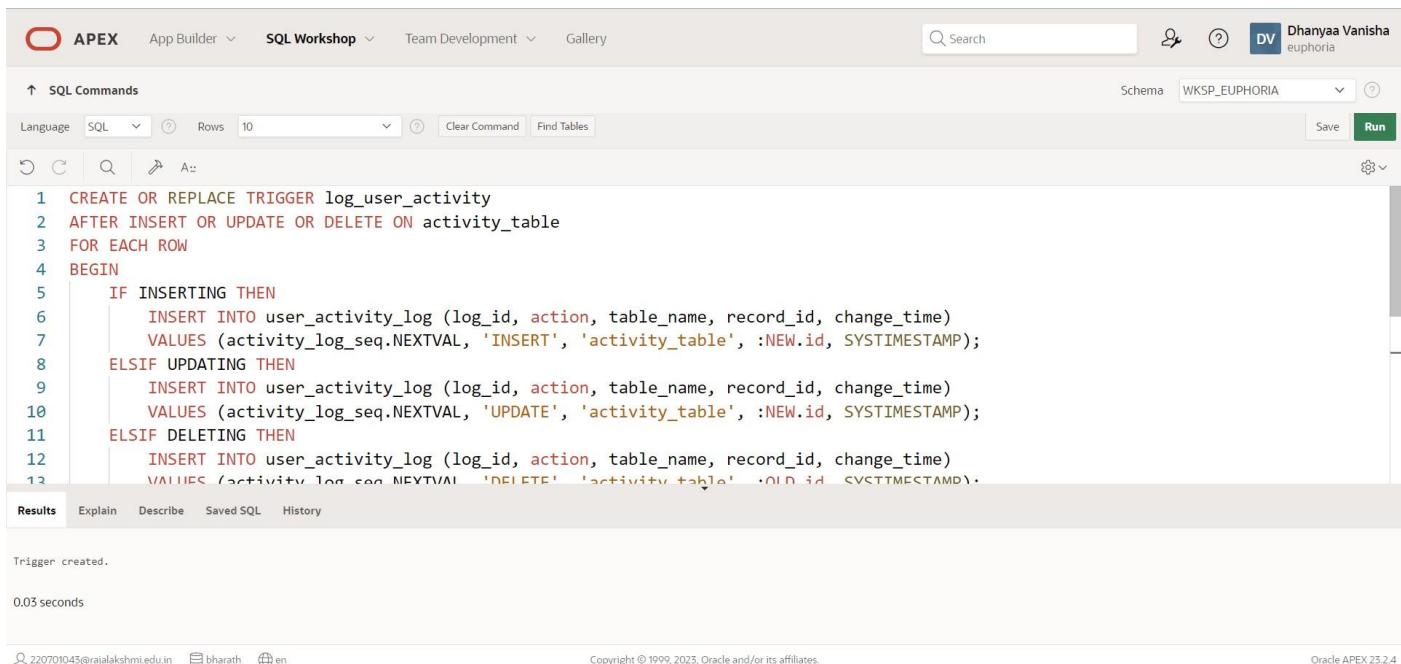
5.)

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 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 'Dhanya Varisha euphoria'. The main area is titled 'SQL Commands' and contains the following PL/SQL code:

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

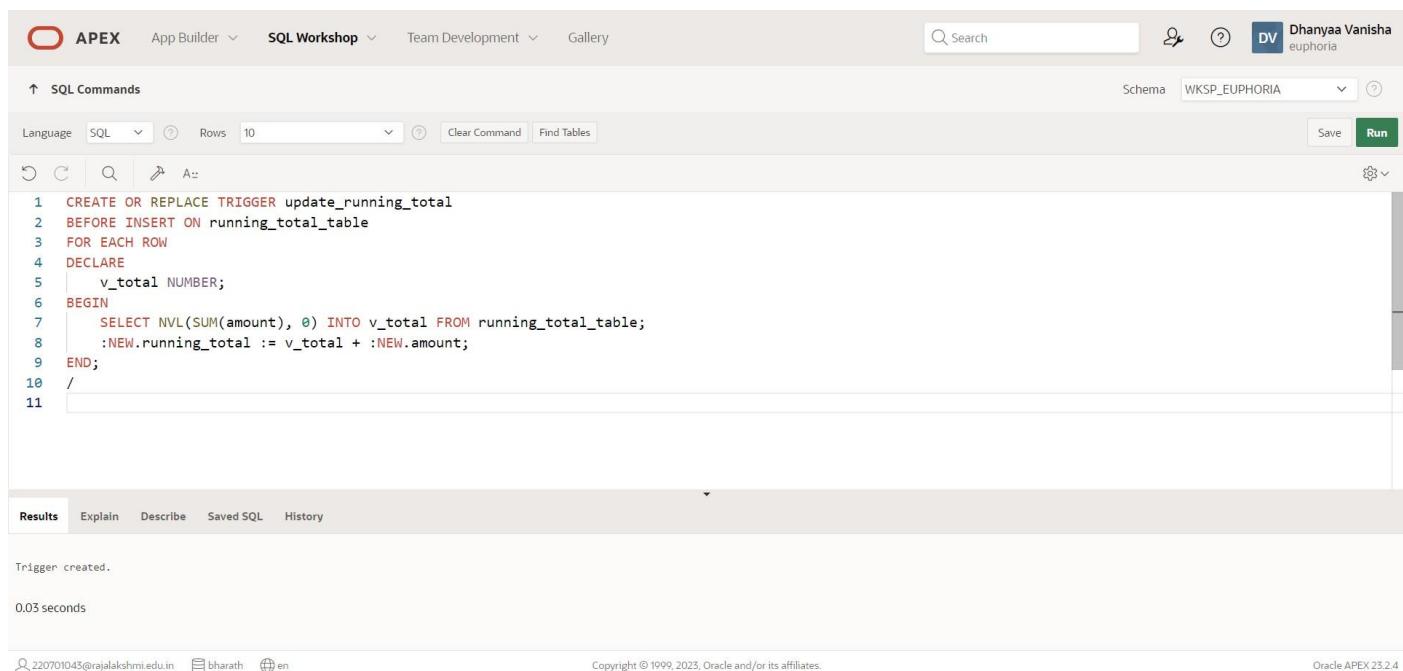
Below the code, the 'Results' tab is selected, showing the message 'Trigger created.' and a timestamp '0.05 seconds'. The bottom footer includes copyright information for Oracle and links to user profile and history.

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 APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'Dhanya Vanisha euphoria'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is numbered from 1 to 11. Below the code, the 'Results' tab is selected, showing the output: 'Trigger created.' and '0.03 seconds'. At the bottom, there are footer links for user information and copyright notice.

```
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 
```

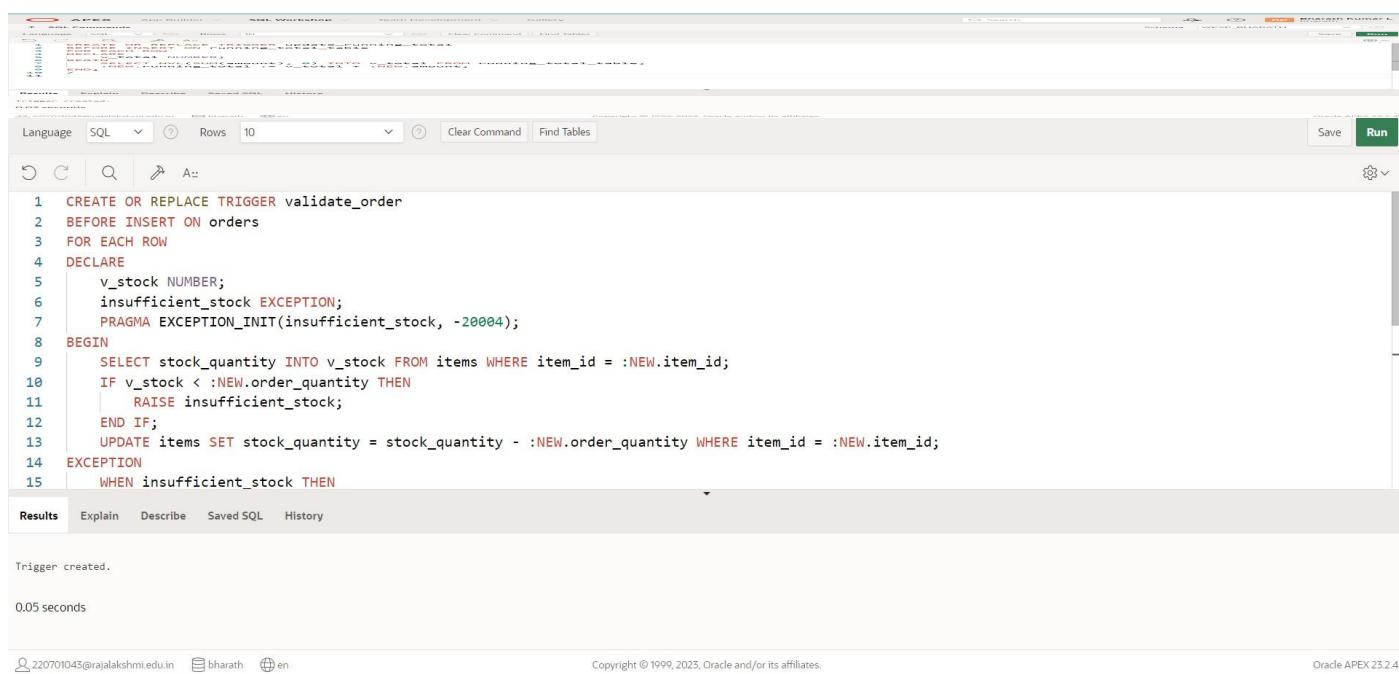
7.)

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 APEX SQL workspace interface. The SQL tab is active, displaying the PL/SQL code for the trigger. The code is identical to the one provided in the question, including the trigger name, arguments, declaration section, BEGIN block with a SELECT statement and IF condition, an UPDATE statement, and an EXCEPTION section with a WHEN clause. The code is numbered from 1 to 15. Below the code, the results pane shows the message "Trigger created." and a execution time of "0.05 seconds". At the bottom, there are footer links for user information and copyright notice.

```
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
        RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');

Trigger created.

0.05 seconds
```

RESULT:

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

MONGO DB

EX_NO: 19

DATE:

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

QUERY:

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

OUTPUT:

```
Dhanyaa Vanisha_62> db.restaurants.find( { $or: [ { name: /^Wil/ }, { cuisine: { $nin: ['American', 'Chinese'] } } ] , { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 } } );
{
  "_id": ObjectId('664f3c798752f54dc3cdcdf7'),
  "borough": "Bronx",
  "cuisine": "Bakery",
  "name": "Morris Park Bake Shop",
  "restaurant_id": '30075445'
}
Dhanyaa Vanisha_62> |
```

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:

```
Dhanyaa Vanisha_62> db.restaurants.find( { grades: { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 } );
Dhanyaa Vanisha_62> |
```

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:

```
Dhanyaa Vanisha_62> 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 } );
Dhanyaa Vanisha_62> |
```

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:

```
Dhanyaa Vanisha_62> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, { _id:0, restaurant_id:1, name:1, address:1})
Dhanyaa Vanisha_62>
```

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:

```
Dhanayaa Vanisha_62>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'
  }
]
Dhanayaa Vanisha_62>
```

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:

```
Dhanayaa Vanisha_62>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'
  }
]
Dhanayaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>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"}]
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({"address.street": {$exists: true, $ne: ""}})
[{"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"}]
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
[
  {
    _id: ObjectId('664f3c798752f54dc3cdcdf7'),
    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'
  }
]
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 })
[
  {
    _id: ObjectId('664f3c798752f54dc3cdcdf7'),
    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'
  }
]
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })  
[  
  {  
    _id: ObjectId('664f3c798752f54dc3cdcdf7'),  
    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'  
  }  
]  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })  
[  
  {  
    _id: ObjectId('664f3c798752f54dc3cdcdf7'),  
    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'  
  }  
]  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
Dhanyaa Vanisha_62>
```

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:

```
Dhanyaa Vanisha_62> db.restaurants.find({$and:[{"grades.grade": "A", "grades.score": 2}, {"grades.grade": "A", "grades.score": 6}], $or:[{"borough": "Manhattan"}, {"borough": "Brooklyn"}], "cuisine": {$ne: "American"}})
```

Dhanyaa Vanisha_62>

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:

```
Dhanyaa Vanisha_62> 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"]}})
```

Dhanyaa Vanisha_62>

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:

```
Dhanyaa Vanisha_62>db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
[ {
  _id: ObjectId('664f3c798752f54dc3cdcdf7'),
  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'
}
]
Dhanyaa Vanisha_62>
```

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

RESULT:

MONGO DB

EX_NO: 20

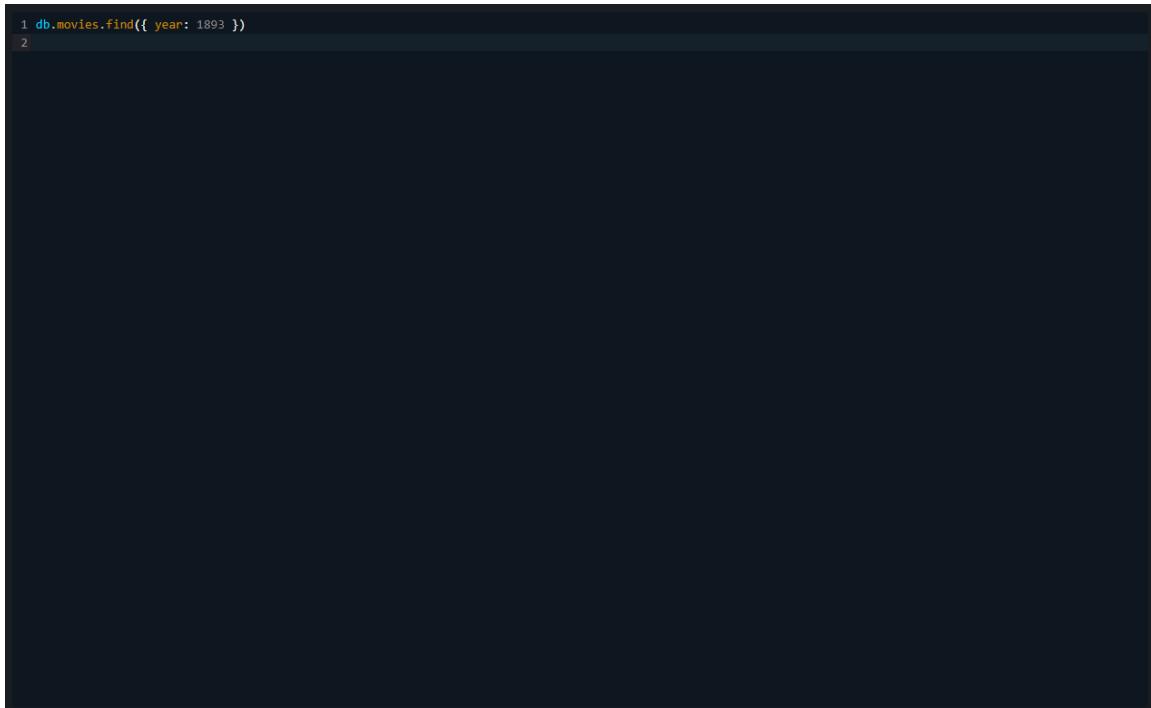
DATE:

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

QUERY:

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

OUTPUT:



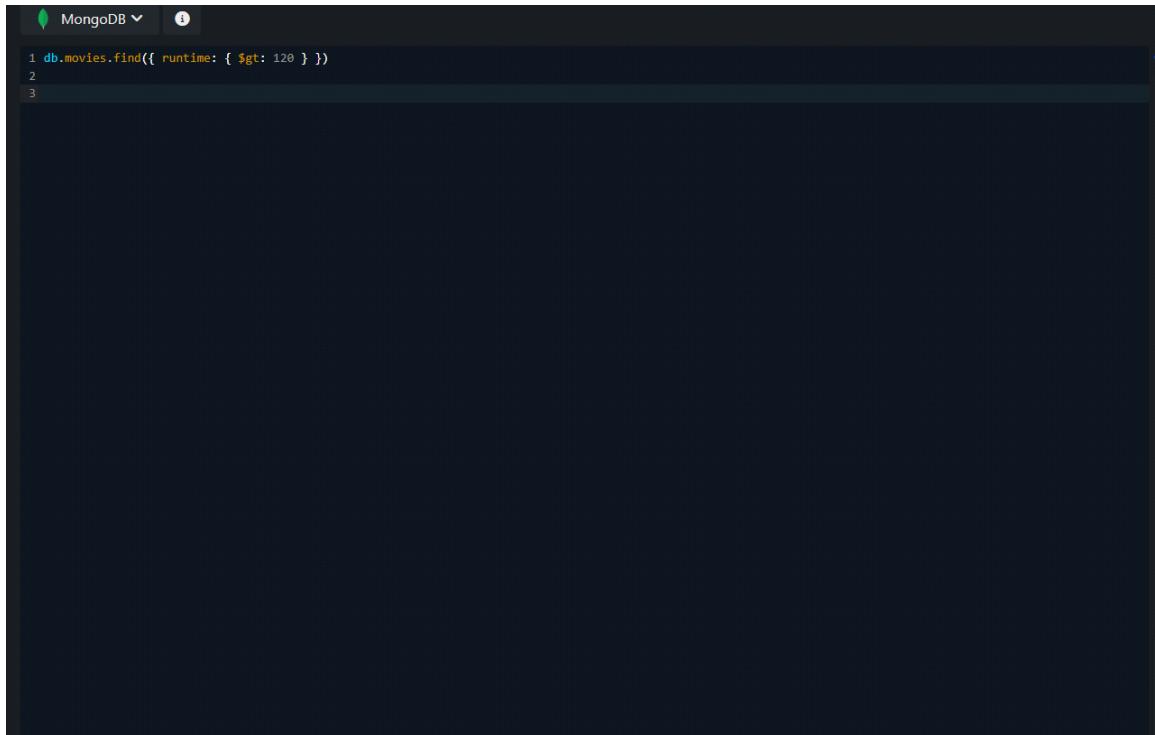
```
1 db.movies.find({ year: 1893 })
2
```

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:



A screenshot of a MongoDB shell window. The title bar says "MongoDB". The code area contains three lines of code:

```
1 db.movies.find({ runtime: { $gt: 120 } })  
2  
3
```

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

QUERY:

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

OUTPUT:



A screenshot of a MongoDB terminal window. The title bar says "MongoDB". The main area shows two lines of code:

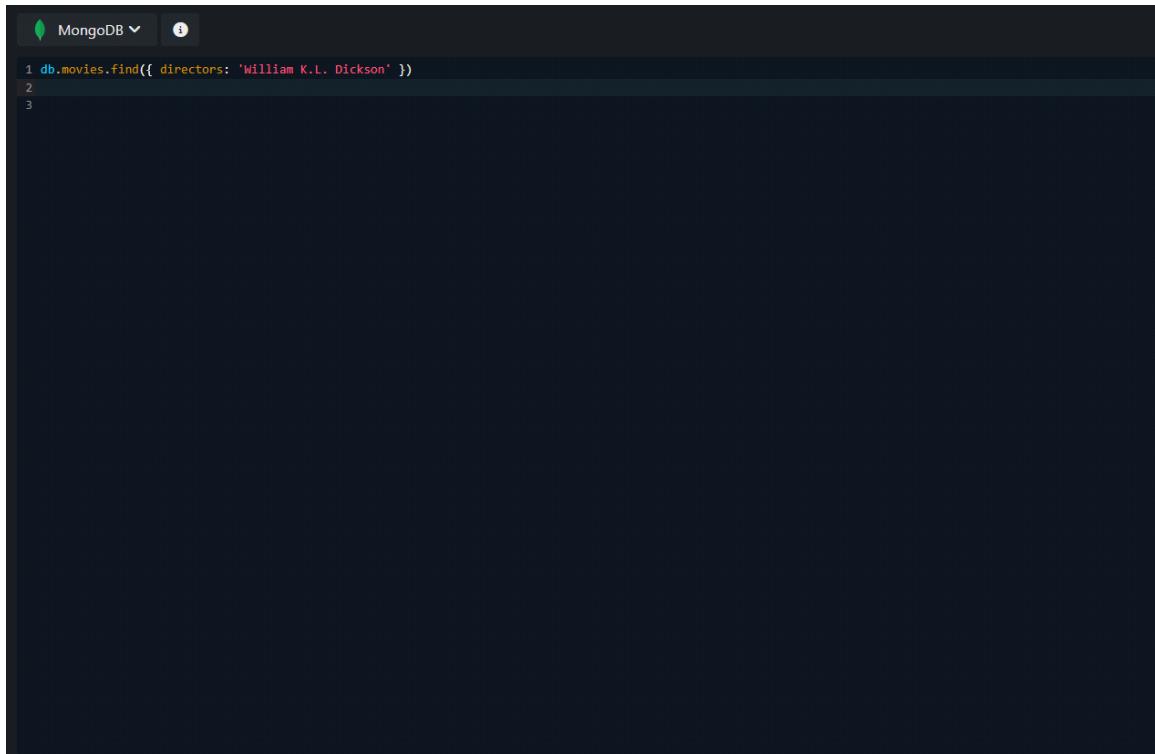
```
1 db.movies.find({ genres: 'Short' })  
2
```

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:



A screenshot of a MongoDB terminal window. The title bar says "MongoDB". The main area shows a single line of code: "db.movies.find({ directors: 'William K.L. Dickson' })". The code is color-coded: "db" and "movies" are blue, "find" is orange, and the query condition is in pink.

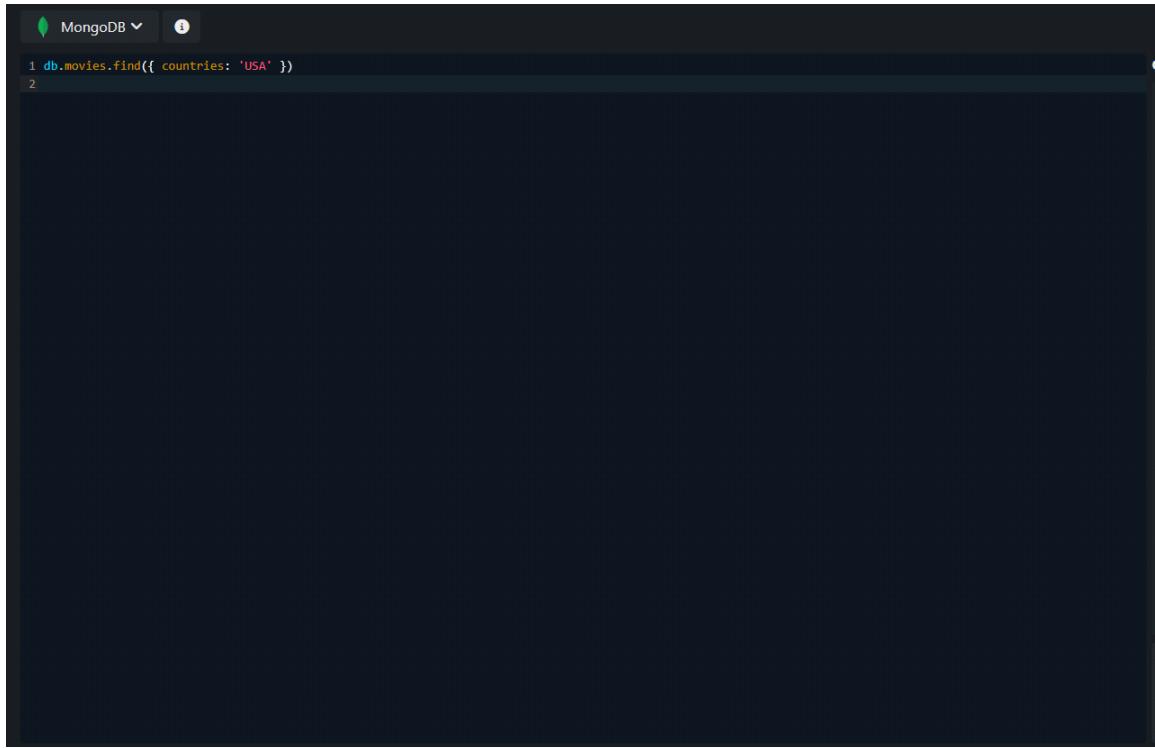
```
1 db.movies.find({ directors: 'William K.L. Dickson' })
```

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:



A screenshot of a MongoDB terminal window. The title bar says "MongoDB". The main area shows two lines of code:

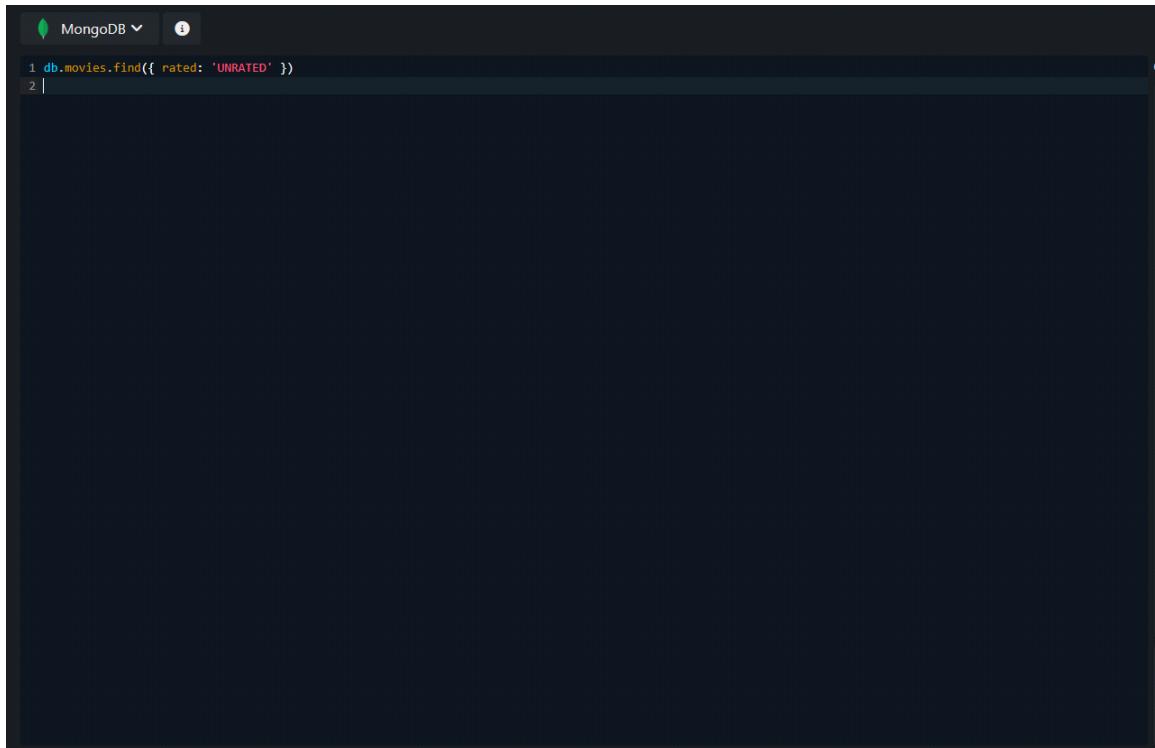
```
1 db.movies.find({ countries: 'USA' })  
2
```

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

QUERY:

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

OUTPUT:



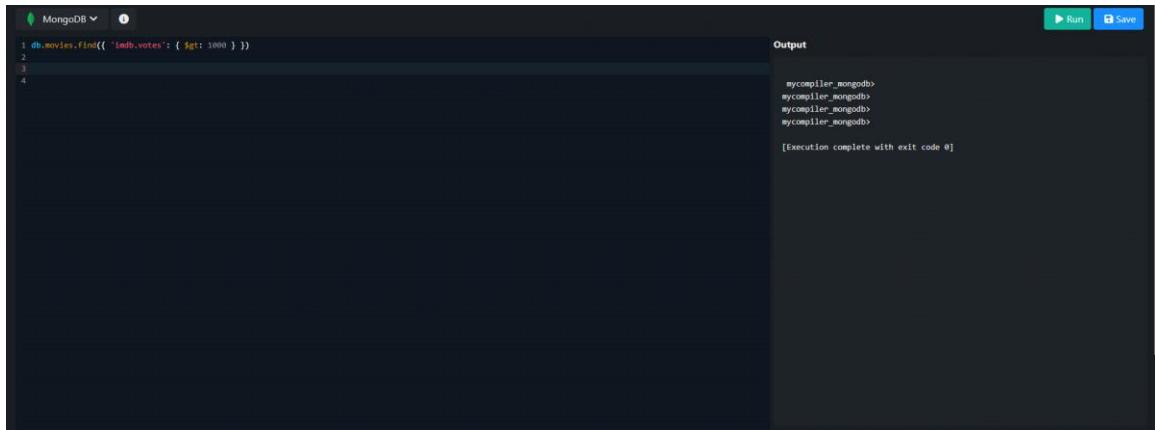
```
MongoDB > db.movies.find({ rated: 'UNRATED' })
```

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:



```
MongoDB > db.movies.find({ "imdb.votes": { $gt: 1000 } })
```

Output

```
mycompiler_mongodb>
mycompiler_mongodb>
mycompiler_mongodb>
mycompiler_mongodb>
```

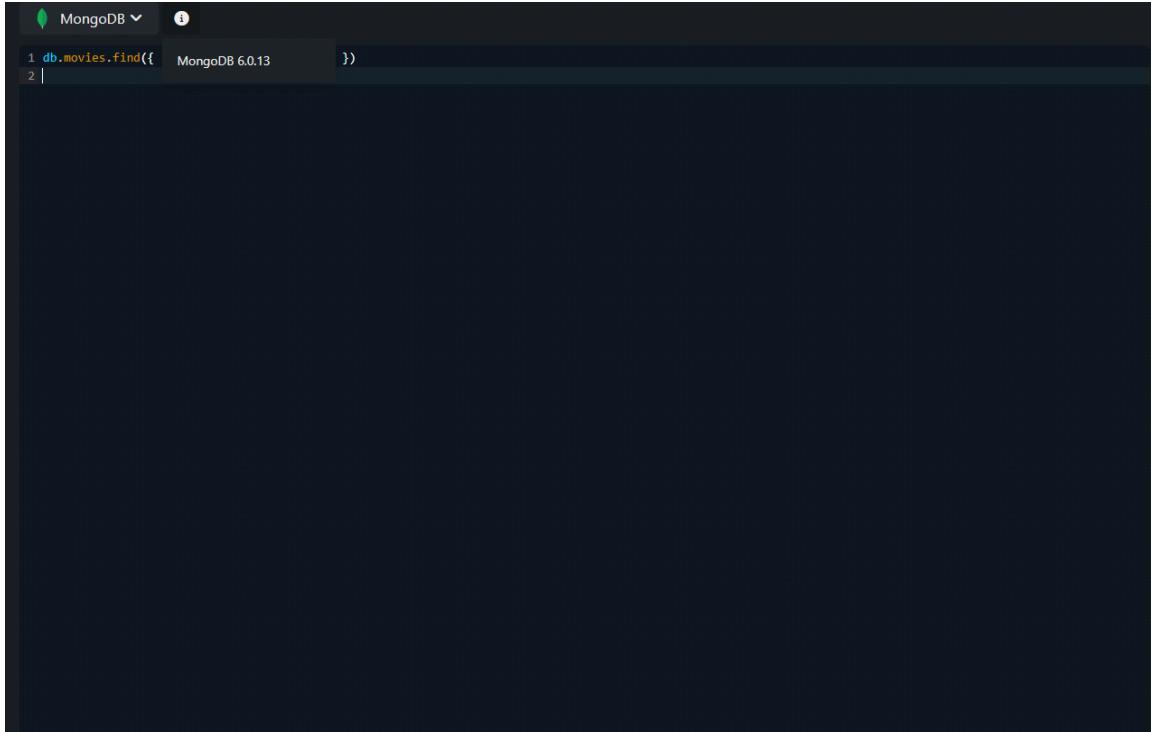
[Execution complete with exit code 0]

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:

A screenshot of a MongoDB shell window. The title bar says "MongoDB". The main area shows a command line with two lines of text:
1 db.movies.find({ 'imdb.rating': { \$gt: 7 } })
2 |
The window has a dark background and light-colored text.

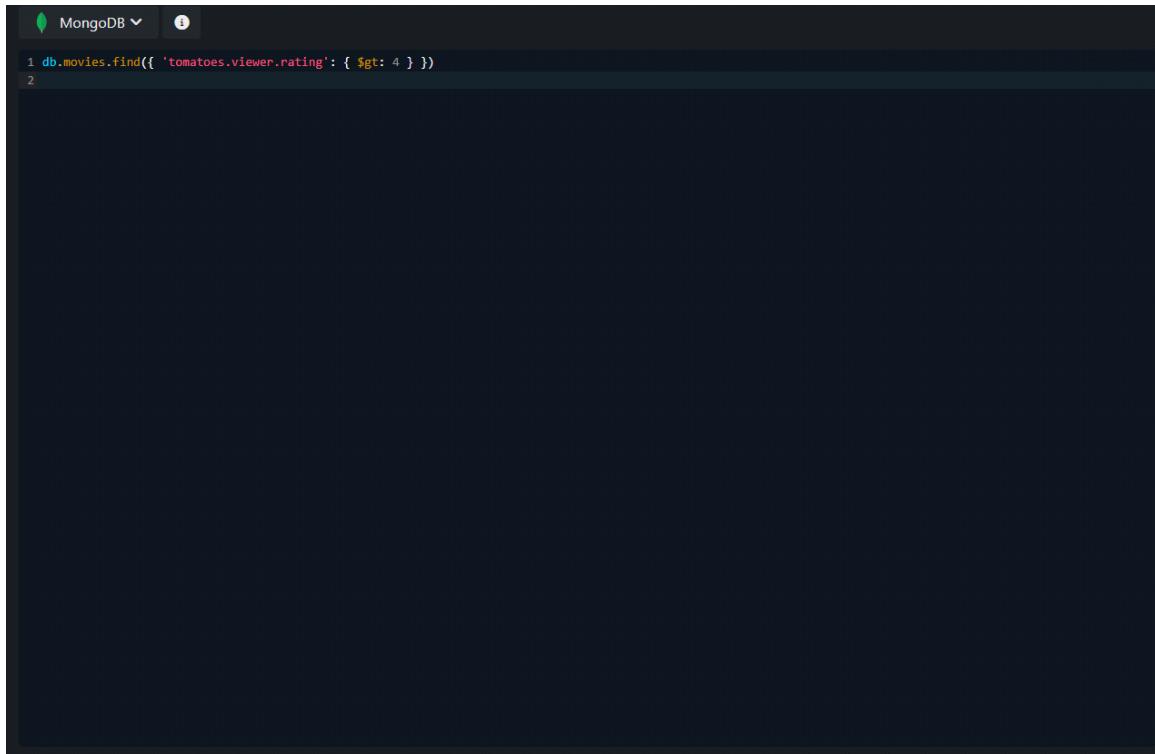
Line	Content
1	db.movies.find({ 'imdb.rating': { \$gt: 7 } })
2	

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:



A screenshot of a MongoDB shell window. The title bar says "MongoDB". The code input field contains two lines of MongoDB queries:

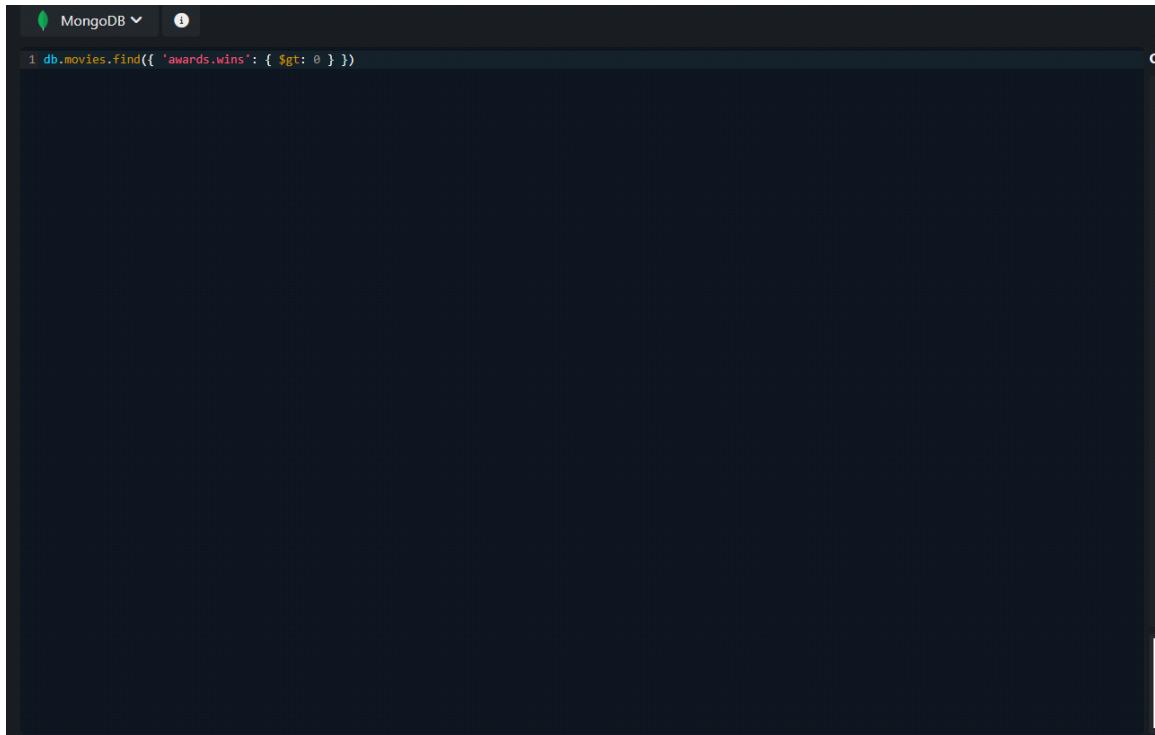
```
1 db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })  
2
```

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

QUERY:

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

OUTPUT:

A screenshot of a MongoDB shell window. The title bar says "MongoDB". The main area shows a single line of code: "1 db.movies.find({ 'awards.wins': { \$gt: 0 } })". The rest of the window is blank, indicating no results have been returned yet.

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:

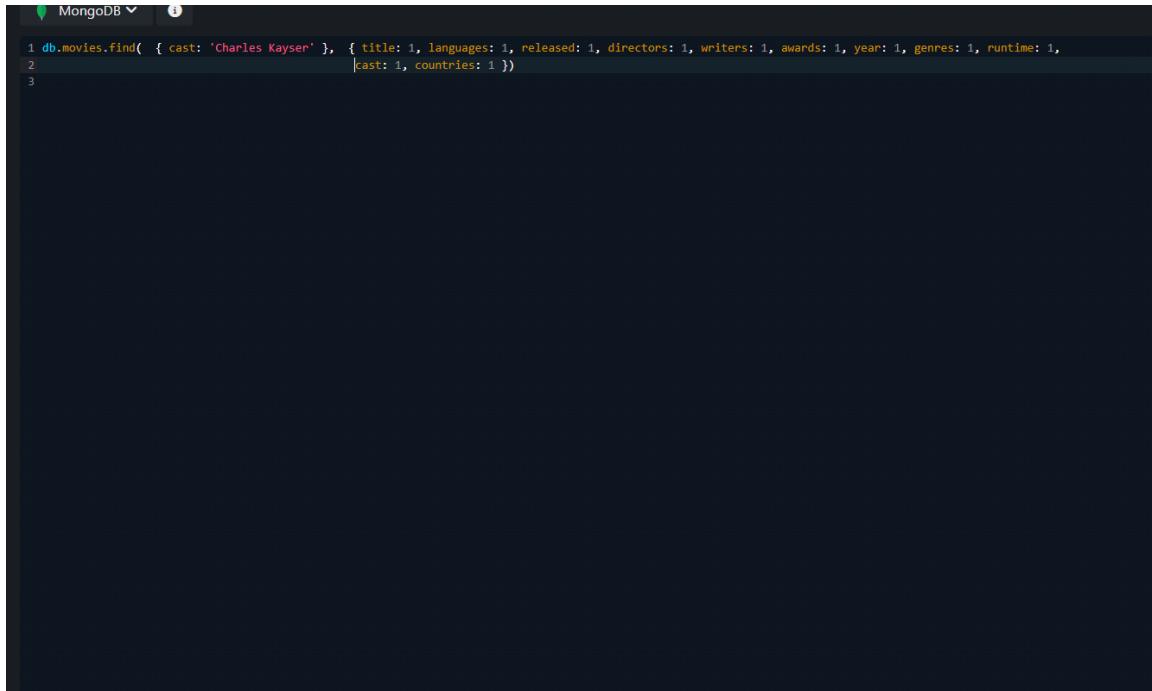
```
MongoDB> 1 db.movies.find( { 'awards.nominations': { $gt: 0 } }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })
2
3
```

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:

A screenshot of a MongoDB shell window titled "MongoDB". The code entered is:

```
1 db.movies.find( { cast: 'Charles Kaysen' }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1,
2   |cast: 1, countries: 1 })
3
```

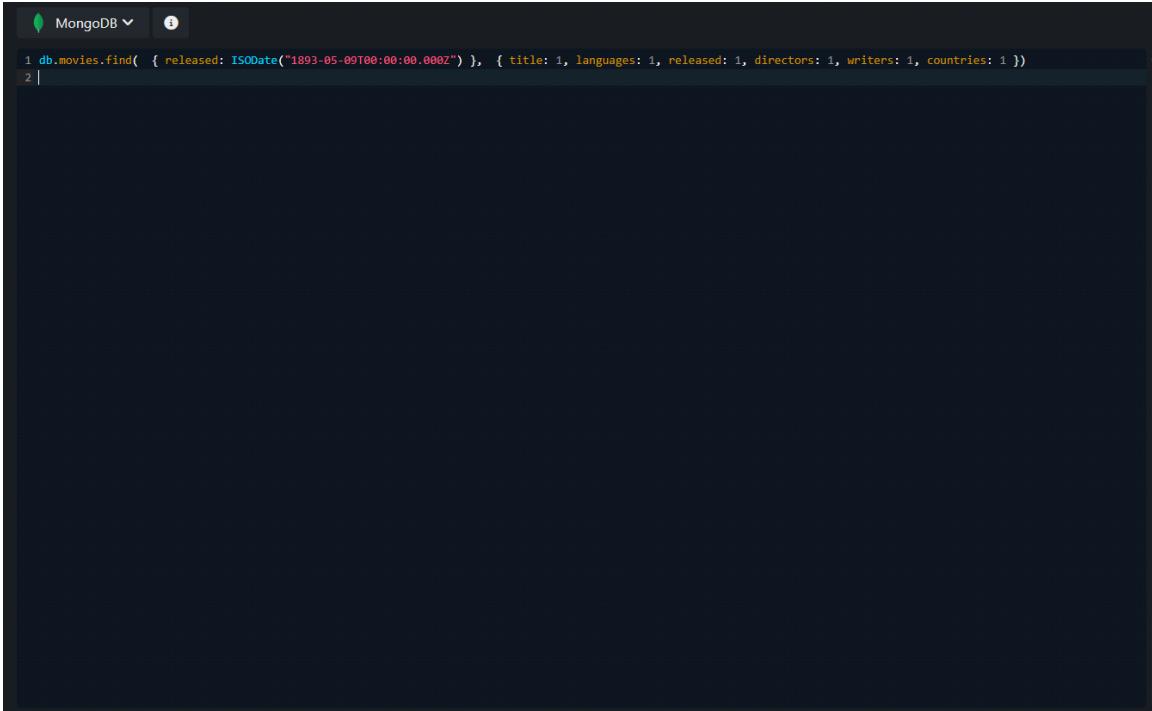
The window has a dark background and light-colored text.

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:



A screenshot of a MongoDB shell interface. The title bar says "MongoDB". The command line shows two lines of code:

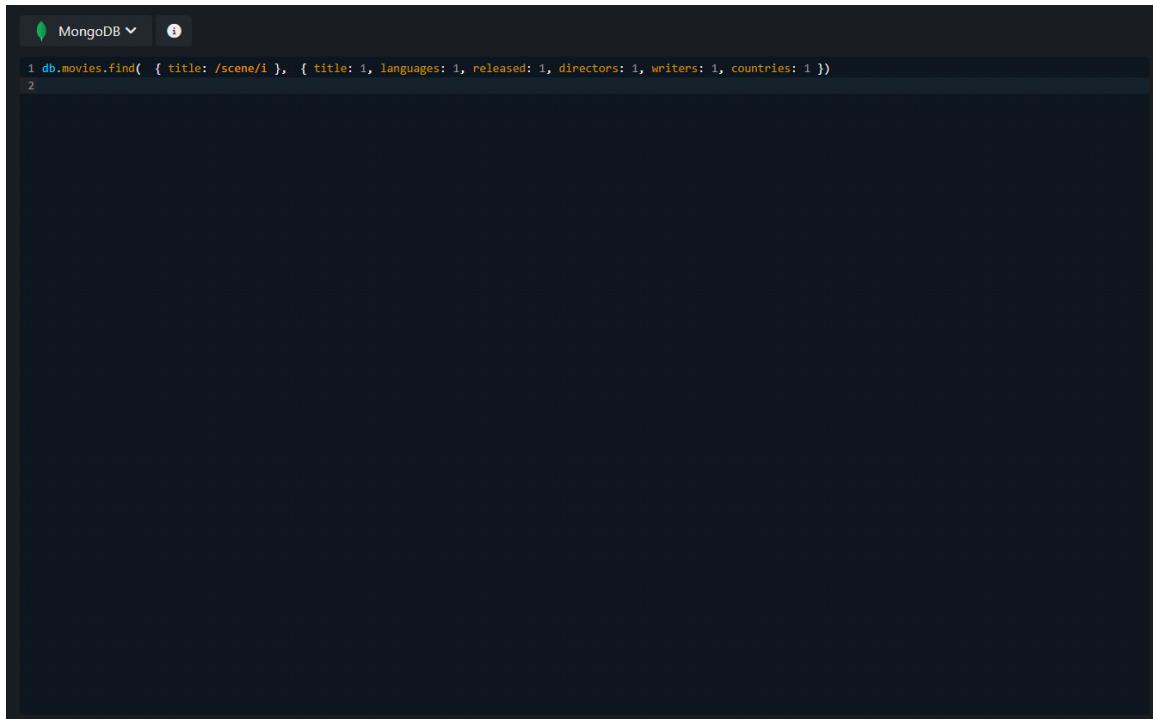
```
1 db.movies.find( { released: ISODate("1893-05-09T00:00:00.000Z") }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })  
2 |
```

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:



A screenshot of a MongoDB shell window. The title bar says "MongoDB". The main area contains the following code:

```
1 db.movies.find( { title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })
```

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

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

RESULT: