

Creating and Managing Tables

EX_NO:1

DATE:

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

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

QUERY:

```
Create table dept(id number(7),name varchar2(25))
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'SQL Commands' is selected. The schema dropdown shows 'WKSP_BRON'. The main area contains the SQL command: 'Create table dept(id number(7),name varchar2(25));'. Below the command, the results section displays the message 'Table created.' and '3.59 seconds'. At the bottom, it shows the user's email '220701039@rajalakshmi.edu.in', the session ID 'bron', and the language 'en'. The copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also visible.

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 emp2(id number(7),Last_Name varchar2(25),First_Name varchar2(25),Dept_id number(7));
```

OUTPUT:

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

```
1 Create table emp2(id number(7),Last_Name varchar2(25),First_Name varchar2(25),Dept_id number(7));
```

In the Results tab, the output shows:

```
Table created.
```

Execution time: 0.03 seconds

Page footer information:

- User: 220701059@rajalakshmi.edu.in
- Session: bron
- Language: en
- Copyright: Copyright © 1999, 2023, Oracle and/or its affiliates.
- Version: Oracle APEX 25.2.4

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

QUERY:

```
Alter table emp2 modify(Last_Name varchar2(25));
```

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, a user icon for 'bavesh karthik bron', and a 'Run' button. The main workspace is titled 'SQL Commands'. It shows the following command in the editor:

```
1 Alter table emp2 modify(Last_Name varchar2(25));|
```

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

```
Table altered.
```

Execution details: 0.05 seconds. The bottom of the screen displays copyright information and the version '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 emp2(id number(7),first_name varchar2(25),Last_name varchar2(25),Salary int,Dept_id number(7));
```

OUTPUT:

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

```
1 Create table employees2(id number(7),first_name varchar2(25),Last_name varchar2(25),Salary int,Dept_id
2 number(7));
```

The 'Run' button is highlighted in green. Below the editor, the 'Results' tab is selected. The output shows:

Table created.
0.03 seconds

At the bottom, user information and copyright details are visible:

220701039@rajalakshmi.edu.in bron en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

5.Drop the EMP table.

QUERY:

Drop table emp;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, along with 'SQL Workshop'. The schema is set to 'WKSP_BRON'. The SQL editor contains the command 'Drop table emp;'. The results tab shows the output: 'Table dropped.' and a execution time of '0.80 seconds'. The bottom footer includes user information and copyright details.

APEX App Builder SQL Workshop Team Development Gallery

Search

Schema WKSP_BRON

Language SQL Rows 10 Clear Command Find Tables Save Run

SQL Commands

1 Drop table emp;

Results Explain Describe Saved SQL History

Table dropped.

0.80 seconds

Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 25.2.4

6.Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename emp to emp2;

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 'bavesh karthik' are also present. The main area is titled 'SQL Commands' and contains a code editor with the following content:

```
1 Rename emp to emp2;
2
```

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

Statement processed.
0.04 seconds

At the bottom of the page, footer information includes the URL '220701039@rajalakshmi.edu.in', session ID 'bron', and a copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' followed by '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. In the top navigation bar, the 'SQL Workshop' tab is selected. The main area is titled 'SQL Commands'. The schema dropdown shows 'WKSP_BRON'. The code entered is:

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

Below the code, the 'Results' tab is active. The output shows:

```
Statement processed.
0.03 seconds
```

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

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

QUERY:

```
Alter table emp drop column first_name;
```

OUTPUT:

The screenshot shows the Oracle 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 'bavesh karthik' and the schema 'WKSP_BRON'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 Alter table emp2 drop column first_name;
```

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

```
Table altered.
```

Execution time: 0.07 seconds

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

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 myemp(ID Number (4) Not null, 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. The SQL Workshop tab is selected. The schema dropdown is set to 'WKSP_BRON'. The SQL command input field contains the following code:

```
1 create table myemp(ID Number (4) Not null, Last_name Varchar (25), First_name Varchar (25),
2 Userid Varchar (25), Salary Number (9,2));
```

The results section below the code shows the output of the command:

Table created.
0.04 seconds

At the bottom of the page, there are footer links for user information (220701039@rajalakshmi.edu.in, bron, en), copyright (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version (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 myemp values(2, 'Dancs', 'Betty', 'bdancs' ,860);
```

OUTPUT:

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

```
1 insert into myemp values(2, 'Dancs', 'Betty', 'bdancs' ,860);
```

In the Results tab, the output is displayed as:

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

Execution details at the bottom show:

```
0.01 seconds
```

Page footer information includes:

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

```
Oracle APEX 23.2.4
```

3. Display the table with values.

QUERY:

```
select*from myemp;
```

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, a user profile for 'bavesh karthik bron', and a 'Run' button. The main workspace displays the SQL command 'select*from myemp;' in the SQL Commands section. Below it, the Results section shows a single row of data from the 'myemp' table:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
2	Dancs	Betty	bdancs	860

Below the results, a message indicates '1 rows returned in 0.04 seconds' and a 'Download' link is available. The bottom footer includes the URL '220701039@rajalakshmi.edu.in', session information 'bron en', copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

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 myemp values (3, 'Biri', 'Ben', 'bbiri' ,1100);
```

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 'bavesh karthik bron'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), and buttons for Clear Command and Find Tables. Below these is a toolbar with icons for Undo, Redo, Search, and Paste. The command input field contains the SQL statement: `1 insert into myemp values (3, 'Biri', 'Ben', 'bbiri' ,1100);`. To the right of the input field are Save and Run buttons. The results section below shows the output of the query: `1 row(s) inserted.` and `0.00 seconds`. The bottom of the page displays copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. and Oracle APEX 23.2.4.

5. Make the data additions permanent.

QUERY:

```
select*from myemp;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands panel, the command `select*from myemp;` is entered. The Results panel displays the output of the query:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
3	Biri	Ben	bbiri	1100
2	Dancs	Betty	bdancs	860

Below the table, it says "2 rows returned in 0.01 seconds". The bottom of the page includes user information (220701039@rajalakshmi.edu.in, bron, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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

QUERY:

```
update myemp 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, and session information (Schema: WKSP_BRON, User: bavesh_karthik). The main workspace is titled 'SQL Commands' and contains a command line with the following SQL statement:

```
1 update myemp set last_name ='Drexler'where id=3;
```

Below the command line, the results section displays the output of the query:

1 row(s) updated.
0.02 seconds

At the bottom, the footer includes user details (User ID: 220701039@rajalakshmi.edu.in, Language: en, Locale: en), copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the software version (Oracle APEX 23.2.4).

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

QUERY:

```
update myemp set Salary =1000 where Salary<900
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop' (which is currently active), 'Team Development', and 'Gallery'. On the right side, there's a search bar, user profile information for 'bavesh karthik bron', and a 'Run' button. The main workspace is titled 'SQL Commands' and contains the following content:

```
1 update myemp set Salary =1000 where Salary<900;
```

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

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

At the bottom of the page, there are footer links for user information (220701059@rajalakshmi.edu.in, bron, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and system information (Oracle APEX 23.2.4).

8.Delete Betty dancs from MY_EMPLOYEE table.

QUERY:

```
delete from myemp where First_name= 'Betty';
```

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, a user profile for 'bavesh karthik bron', and a schema dropdown set to 'WKSP_BRON'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code: '1 delete from myemp where First_name= 'Betty'';. Below the code, the results tab is selected, showing the output: '1 row(s) deleted.' and '0.01 seconds'. The bottom footer displays user information ('220701039@rajalakshmi.edu.in', 'bron', 'en') and copyright information ('Copyright © 1999, 2023, Oracle and/or its affiliates.', 'Oracle APEX 23.2.4').

```
1 delete from myemp where First_name= 'Betty';
```

Results Explain Describe Saved SQL History

1 row(s) deleted.
0.01 seconds

220701039@rajalakshmi.edu.in bron en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

9.Empty the fourth row of the emp table.

QUERY:

delete from myemp where id=4

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 (Schema: WKSP_BRON, User: bavesh karthik bron). The main workspace is titled "SQL Commands". It features a toolbar with icons for Undo, Redo, Find, Replace, and Run. Below the toolbar, the command input field contains the SQL statement: "1 delete from myemp where id=4". The results section is active, showing the output of the query: "0 row(s) deleted." and "0.01 seconds". At the bottom, footer information includes the user's email (220701039@rajalakshmi.edu.in), session ID (bron), and language (en). Copyright information and the version "Oracle APEX 23.2.4" are also present.

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) not null, 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. In the SQL Commands tab, a single SQL command is entered to create a table 'EMP39' with columns 'ID', 'LAST_NAME', 'EMAIL', and 'SALARY', and a primary key constraint 'MY_EMP_ID_PK' on the 'ID' column. The command is:

```
1 CREATE TABLE EMP39(ID NUMBER(6),LAST_NAME VARCHAR(25) NOT NULL,EMAIL VARCHAR(25),SALARY NUMBER(8,2),CONSTRAINT MY_EMP_ID_PK PRIMARY KEY (ID));
```

In the Results tab, the output shows the table was created successfully, and it took 0.04 seconds. The page footer includes copyright information for Oracle and the APEX version.

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

QUERY:

TABLE NAME MATUM MATHIKO EMP39

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, a user profile for 'bavesh karthik bron', and a schema dropdown set to 'WKSP_BRON'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 CREATE TABLE DEPT9(ID NUMBER(6),LAST_NAME VARCHAR2(25) NOT NULL,EMAIL VARCHAR2(25),CONSTRAINT MY_DEPT9_ID_PK PRIMARY KEY (ID));
```

Below the code, the 'Results' tab is selected, showing the output: 'Table created.' and '0.06 seconds'. The bottom footer includes copyright information for Oracle and the APEX version '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:

```
CREATE TABLE DEPTS(ID number (6), last name varchar2(25) not null,email varchar2(25),constraint my_dept5_id_pk PRIMARY KEY(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's a search bar, user profile, and session information (Schema: WKSP_BRON, User: bavesh karthik). The main area is titled 'SQL Commands' with tabs for Language (SQL selected), Rows (10), Clear Command, and Find Tables. Below this is a toolbar with icons for Undo, Redo, Search, and Run. The SQL editor contains the following code:

```
1 ALTER TABLE EMP39 ADD CONSTRAINT MY_EMP_DEPT_TD_FK FOREIGN KEY(DEPT_ID)
2 REFERENCES EMP39(ID);
3
```

Below the editor, the 'Results' tab is active, showing the output of the command: "Table altered." and "0.06 seconds". At the bottom, footer information includes the URL 220701039@rajalakshmi.edu.in, session ID bron, language en, copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version 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 CONSTRAINT MY_EMP_DEPT_TD_FK FOREIGN  
KEY(DEPT_ID) REFERENCES EMP(ID);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop' (which is currently active), 'Team Development', and 'Gallery'. On the right side, there is a user profile for 'bavesh karthik bron'. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. Below the input field, there are buttons for 'Save' and 'Run'. The SQL command entered is:

```
1 ALTER TABLE EMP39 ADD COMMISSION NUMBER(2,2) CHECK(COMMISSION>0);  
2
```

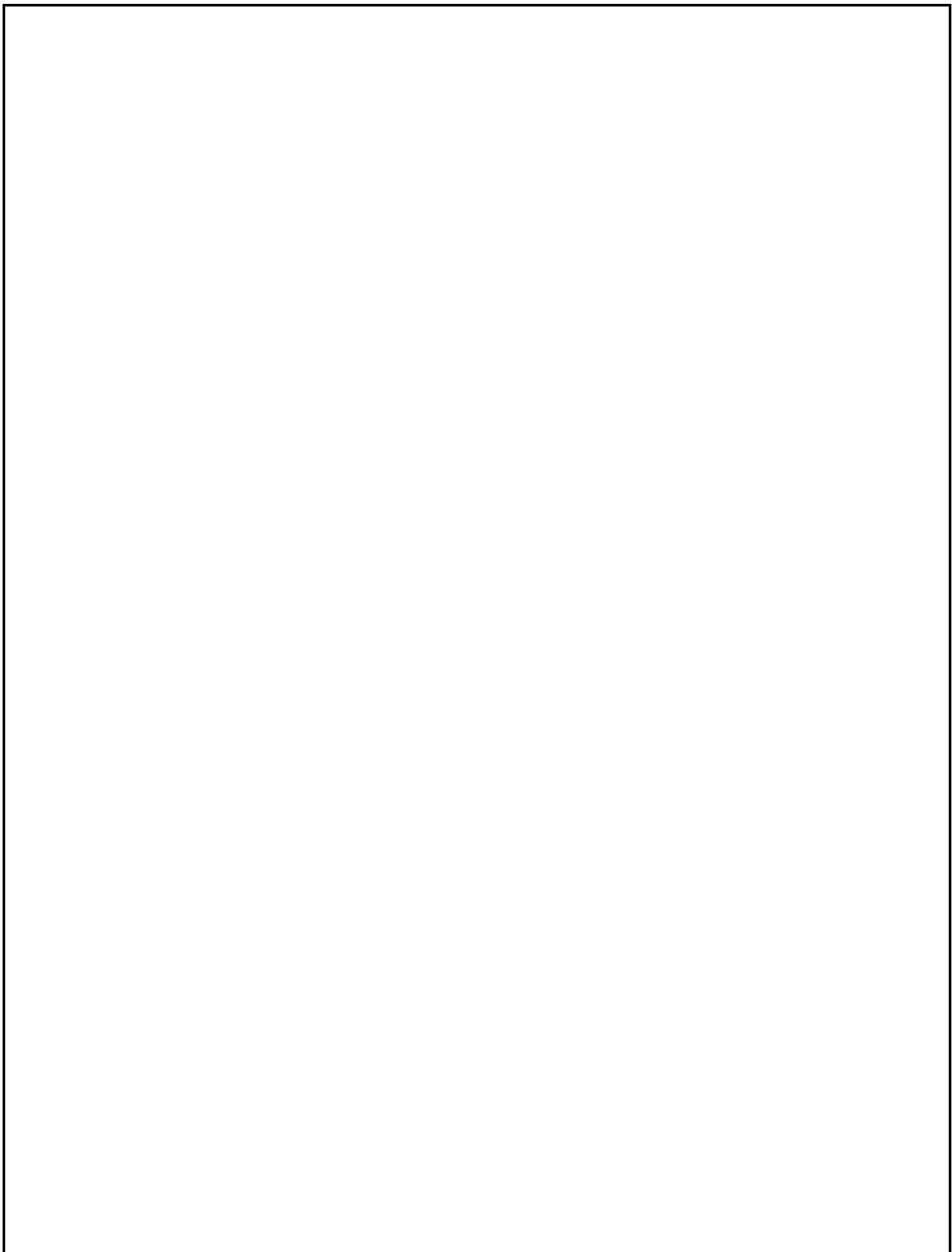
After running the command, the results tab shows the output:

```
Table altered.  
0.05 seconds
```

At the bottom of the page, there are footer links for user information (220701039@rajalakshmi.edu.in, bron, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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

RESULT:



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 emp_id, last_name, salary*12 as "ANNUAL SALARY" 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. The right side shows a user profile for 'bavesh karthik' with the ID 'bron'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The command entered is:

```
1 select emp_id, last_name, salary*12 as "ANNUAL SALARY" from employees;
```

The results section shows the output of the query:

EMP_ID	LAST_NAME	ANNUAL SALARY
114	raphealy	132000
113	popp	82800

Below the table, it says '2 rows returned in 0.03 seconds' and has a 'Download' link. The bottom footer includes copyright information for Oracle and the APEX version '23.2.4'.

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

QUERY:

```
Desc dept;
```

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 'bavesh karthik' with the schema 'WKSP_BRON'. The main area is titled 'SQL Commands' and contains the query 'Desc dept;'. Below the query, the 'Describe' tab is selected, showing the structure of the 'DEPT' table. The table has two columns: 'ID' (NUMBER, precision 7, scale 0, nullable yes) and 'NAME' (VARCHAR2, length 25, nullable yes). The bottom of the screen displays copyright information for Oracle and the APEX version.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT	ID	NUMBER	-	7	0	-	✓	-	-
	NAME	VARCHAR2	25	-	-	-	✓	-	-

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 emp_id,last_name,job_id,hire_date from employees;
```

OUTPUT:

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. The schema is set to WKSP_BRON. The SQL command entered is:

```
1 Select emp_id,last_name,job_id,hire_date from employees;
2
```

The results section displays the following data:

EMP_ID	LAST_NAME	JOB_ID	HIRE_DATE
114	raphealy	ac_account	02/03/1999
115	popp	ac_account	03/08/2024

2 rows returned in 0.01 seconds

At the bottom, the footer includes:

220701039@rajalakshmi.edu.in bron en Copyright © 1999-2023, Oracle and/or its affiliates. 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. 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 'bavesh karthik' and the schema 'WKSP_BRON'. The main area is titled 'SQL Commands' and contains a single line of SQL code: '1 select hire_date as "START_DATE" from employees;'. Below this, the 'Results' tab is selected, displaying the output of the query. The results show two rows: '02/03/1999' and '03/08/2024', both under the column header 'START_DATE'. At the bottom left, it says '2 rows returned in 0.00 seconds' and has a 'Download' link. The bottom right corner indicates the version 'Oracle APEX 23.2.4'.

START_DATE
02/03/1999
03/08/2024

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

QUERY:

```
select distinct job_id from employees;
```

OUTPUT:

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

```
1 select distinct job_id from employees;
2 |
```

The Results tab is selected, showing the output:

JOB_ID
ac_account

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

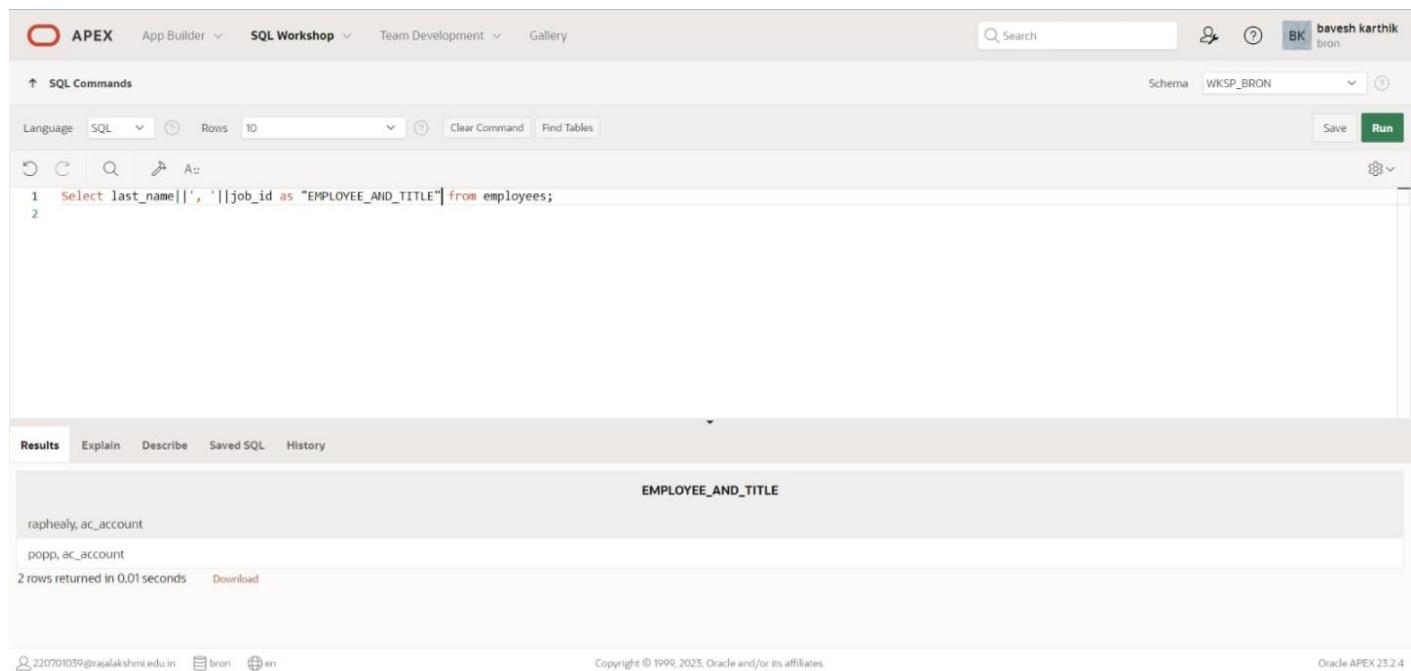
At the bottom, the footer includes user information (220701039@rajalakshmi.edu.in, bron, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version (Oracle APEX 23.2.4).

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

QUERY:

Select last_name||', '||job_id as "EMPLOYEE_AND_TITLE" from employees;

OUTPUT:



The screenshot shows the Oracle 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, a user profile for 'bavesh karthik bron', and session information. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 Select last_name||', '||job_id as "EMPLOYEE_AND_TITLE" from employees;
2
```

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

EMPLOYEE_AND_TITLE
raphealy, ac_account
popp, ac_account

The results indicate that 2 rows were returned in 0.01 seconds. At the bottom, there are footer links for Explain, Describe, Saved SQL, History, and a copyright notice: Copyright © 1999, 2023, Oracle and/or its affiliates. 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 emp_id||','||first_name||','||last_name||','||email||','||phone_no||','||hire_date||','||job_id||','||salary  
||','||commission_pct||','||manager_id||','||department_id as "THE_OUTPUT" from employees;
```

OUTPUT:

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

```
1 select emp_id||','||first_name||','||last_name||','||email||','||phone_no||','||hire_date||','||job_id||','||salary  
2 ||','||commission_pct||','||manager_id||','||department_id as "THE_OUTPUT" from employees;  
3
```

Below the code, the 'Results' tab is selected. The output section displays the results under the heading 'THE_OUTPUT':

THE_OUTPUT
114,den,raphealy,drapheal,5151274561,02/03/1999,ac_account,11000,100,30
113,louis,popp,lpopp,5151244567,03/08/2024,ac_account,6900,,205,100

At the bottom of the results, it says '2 rows returned in 0.01 seconds' and has a 'Download' link. The footer of the page includes the URL '220701039@rajalakshmi.edu.in', the session ID 'bron', the language 'en', and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' followed by '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

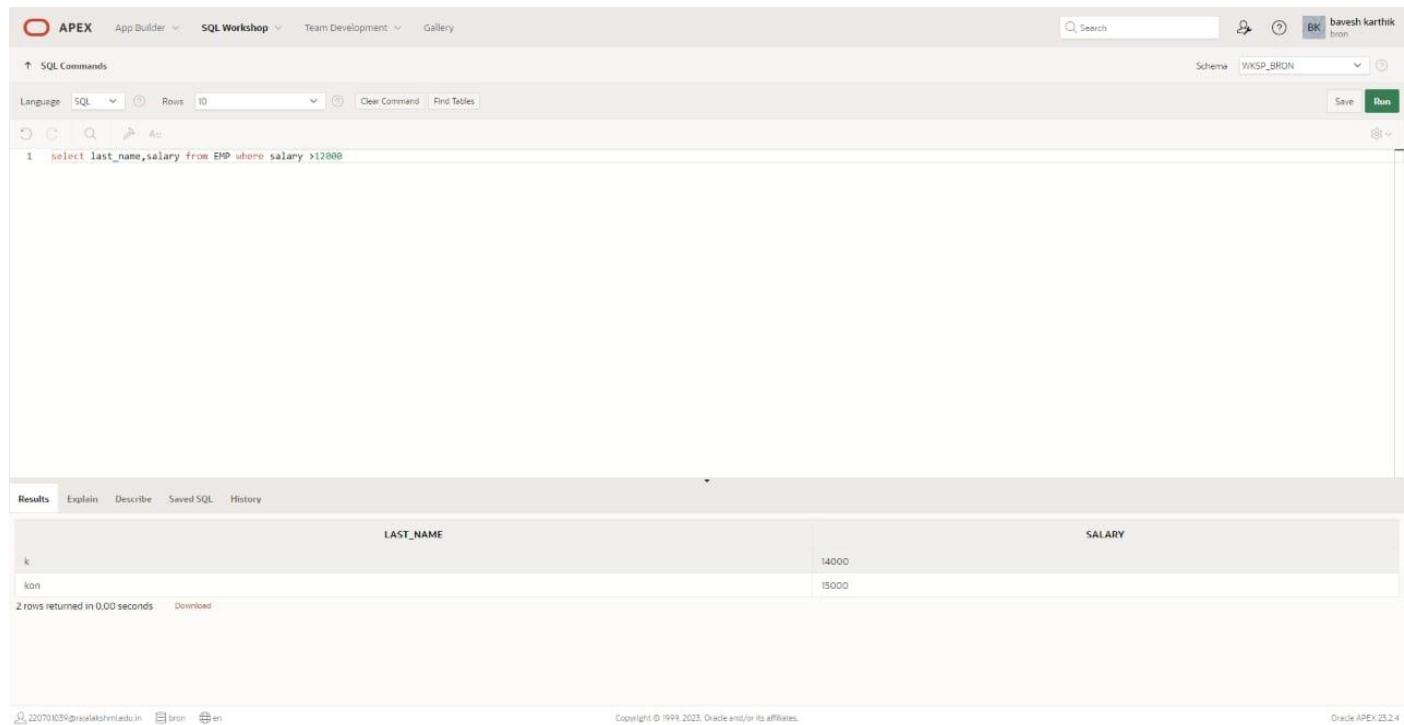
DATE:

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

QUERY:

```
select last_name,salary from EMP where salary >12000;
```

OUTPUT:



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

```
1 select last_name,salary from EMP where salary >12000
```

The results section displays the following data:

LAST_NAME	SALARY
K	14000
KAN	15000

Below the table, it says "2 rows returned in 0.00 seconds".

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

QUERY:

```
select last_name,department_number from EMP where emp_id=176;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown shows 'WkSP_BRON' and the user 'bavesh.karthik'. The SQL command input field contains the query: 'select last_name , dept_no from EMP where emp_id=176'. The results section displays the output:

LAST_NAME	DEPT_NO
ROW1	

1 rows returned in 0.01 seconds. The bottom of the page includes copyright information: Copyright © 1999-2023, Oracle and/or its affiliates. and Oracle APEX 23.2.4.

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

QUERY:

```
select last_name ,salary from 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 'APEX' (selected), 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right side, there's a search bar, a user icon for 'bavesh.karthik', and a schema dropdown set to 'WKSP_BRON'. Below the tabs, there are buttons for 'Save' and 'Run'. The main area has a toolbar with icons for Undo, Redo, Find, Replace, and others. The SQL command input field contains the query: 'select last_name ,salary from EMP where salary not between 5000 and 12000'. The results section shows a table with two rows:

LAST_NAME	SALARY
k	14000
kon	15000

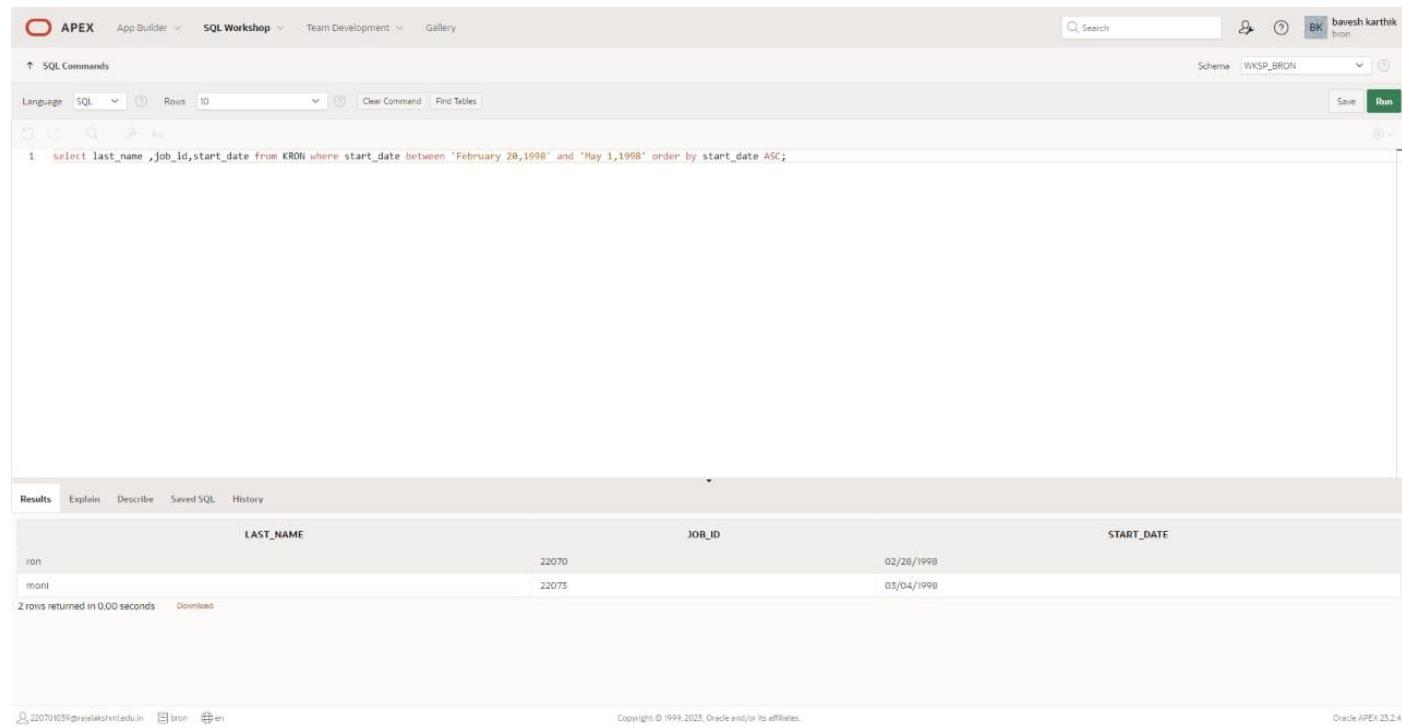
Below the table, it says '2 rows returned in 0.00 seconds' and has a 'Download' link. The bottom of the page includes a footer with copyright information: 'Copyright © 1999-2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

4. Display the employee last name, job ID, and start date of employees hired between February 20,1998 and May 1,1998.order the query in ascending order by start date.(hints: between)

QUERY:

```
select last_name,job_id,Hire_Date from empo21 where Hire_Date between '02/20/1998' and '05/01/1998' order by Hire_date asc;
```

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 'bavesh karthik' and a schema dropdown set to 'WKSP_BRON'. The main workspace contains a SQL command line with the following query:

```
1 select last_name ,job_id,start_date from KROB where start_date between 'February 20,1998' and "May 1,1998" order by start_date ASC;
```

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

LAST_NAME	JOB_ID	START_DATE
iron	22070	02/20/1998
moni	22073	03/04/1998

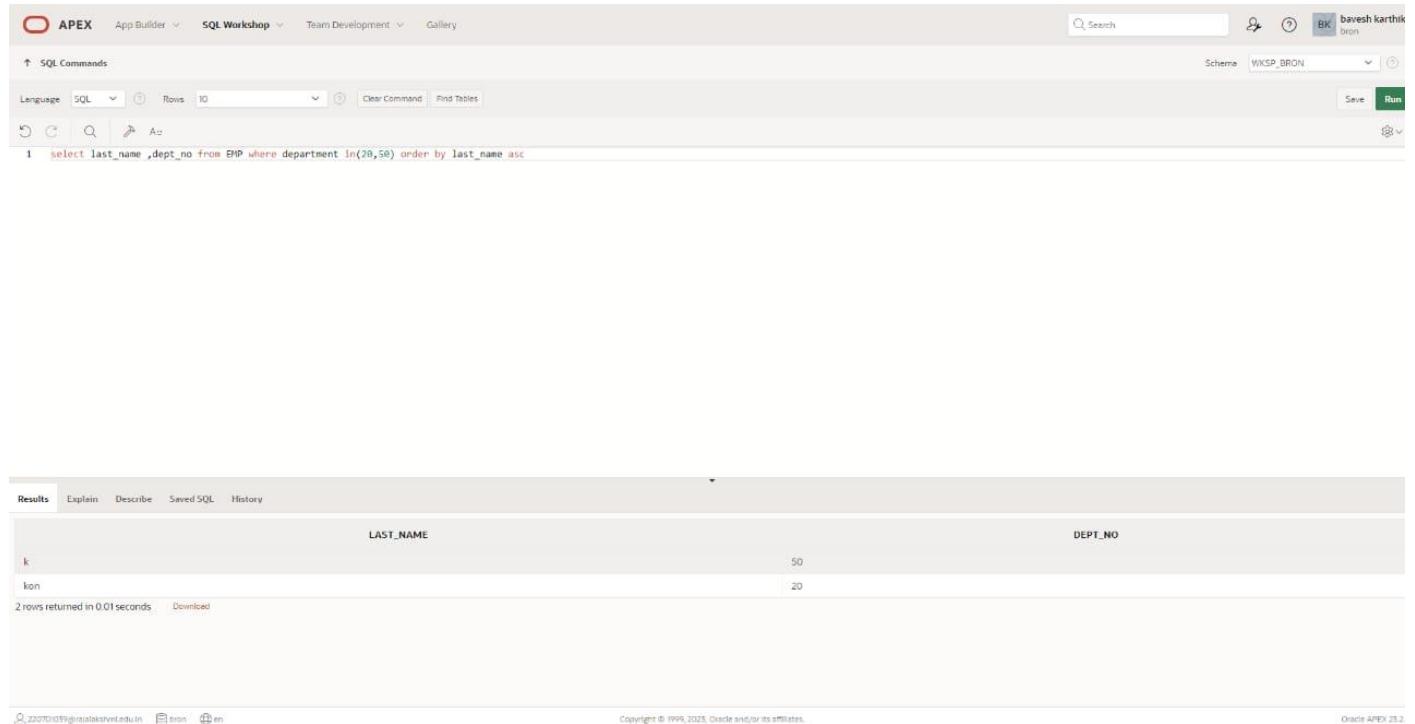
At the bottom, it says '2 rows returned in 0.00 seconds' and provides download options.

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

QUERY:

```
select last_name, department_number from EMP where department in (20,50) order by last_name asc;
```

OUTPUT:



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

```
1 select last_name ,dept_no from EMP where department in(20,50) order by last_name asc
```

The results section displays the following data:

LAST_NAME	DEPT_NO
K	50
KON	20

2 rows returned in 0.01 seconds. Downloaded.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle 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, a user profile for 'bavesh karthik', and a schema dropdown set to 'WKSP_BRON'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

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

Below the code, the 'Results' tab is selected, showing the message 'no data found'. The bottom of the page displays user information (email: 220701039@rajalakshmi.edu.in, session ID: bron, locale: en), a copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the software version (Oracle APEX 23.2.4).

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

QUERY:

```
select last_name,hire_date from employees where hire_date like '%94';
```

OUTPUT:

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

```
1 select last_name,job_name,salary from EMP where job_name in ('hr','manager') and salary not in (2500,3500,7000);
```

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

LAST_NAME	JOB_NAME	SALARY
aero	manager	10000
aes	manager	12000

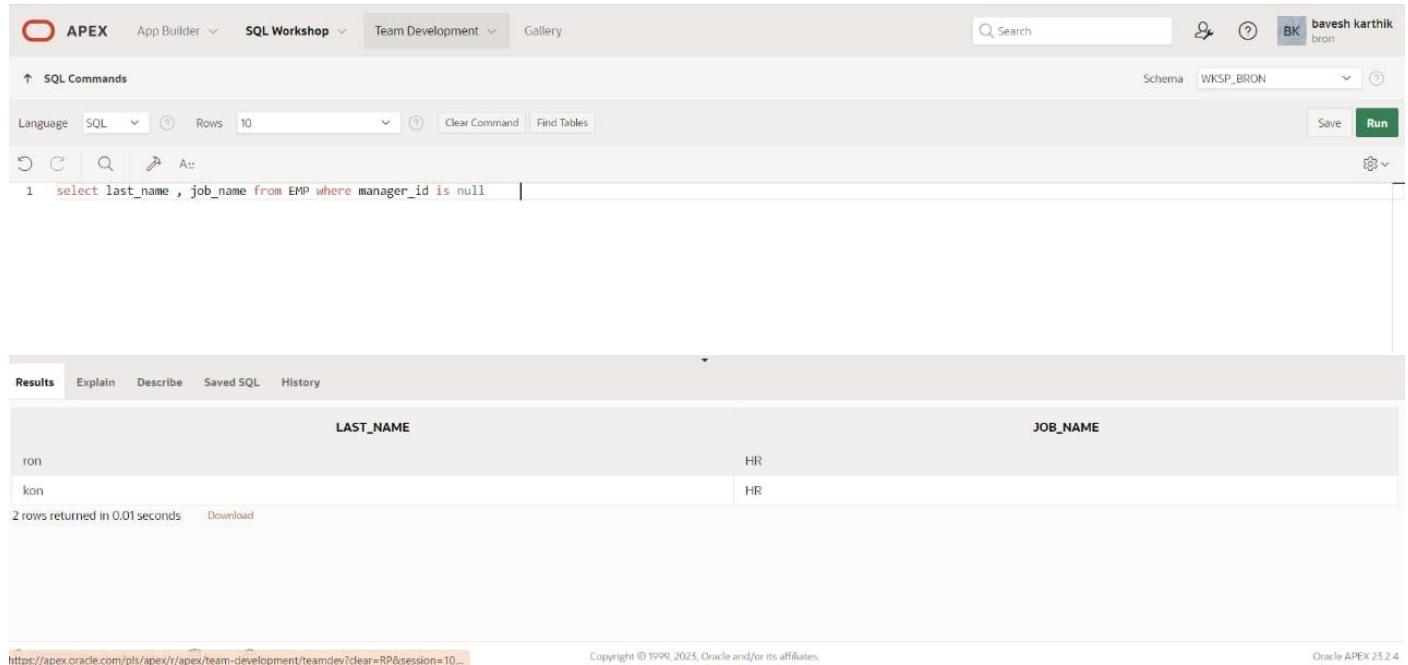
At the bottom left, it says '2 rows returned in 0.01 seconds'. At the bottom right, it says 'Download'. The footer includes links for user information (220701039@rajalakshmi.edu.in, bron, en) and copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates). On the far right, it says 'Oracle 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_id  
FROM EMP  
WHERE manager_id IS NULL;
```

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 'bavesh.karthik' with session ID 'bron'. The schema selected is 'WKSP_BRON'. The SQL command entered is:

```
1 select last_name , job_name from EMP where manager_id is null
```

The results section displays the output of the query:

LAST_NAME	JOB_NAME
jon	HR
kon	HR

2 rows returned in 0.01 seconds. There is a 'Download' link below the results.

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 nul,orderby)

QUERY:

```
SELECT last_name, salary, commission_pct  
FROM EMP  
WHERE commission_pct IS NOT NULL  
ORDER BY salary DESC, commission_pct 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 a user profile for 'bavesh karthik bron'. The main workspace has tabs for SQL Commands, SQL (selected), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The SQL command entered is:

```
1 select last_name,salary,commission_pct from EMP where commission_pct is not null order by salary desc;
```

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

LAST_NAME	SALARY	COMMISSION_PCT
kon	11000	53

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

The bottom footer includes user information (220701039@rjsjalakshmi.edu.in, bron, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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

QUERY:

```
SELECT last_name  
FROM EMP  
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, Team Development, and Gallery. On the right side, there is a user profile for 'bavesh.karthik' and a schema dropdown set to 'WKSP_BRON'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select last_name from KRON where last_name like '__a'
```

The results tab is selected, displaying the output:

LAST_NAME
bhanu

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

At the bottom of the page, there are footer links for '220701039@rajalakshmi.edu.in', 'bron', and 'en'. The copyright notice reads 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version is 'Oracle APEX 23.2.4'.

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

QUERY:

```
SELECT last_name  
FROM EMP  
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, Team Development, and Gallery. On the right, there's a user profile for 'bavesh karthik' and a search bar. The main workspace is titled 'SQL Commands'. It has dropdown menus for Language (set to SQL), Rows (set to 10), and Schema (set to WKSP_BRON). Below these are standard database navigation icons (refresh, search, etc.). A code editor window contains the following SQL query:

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

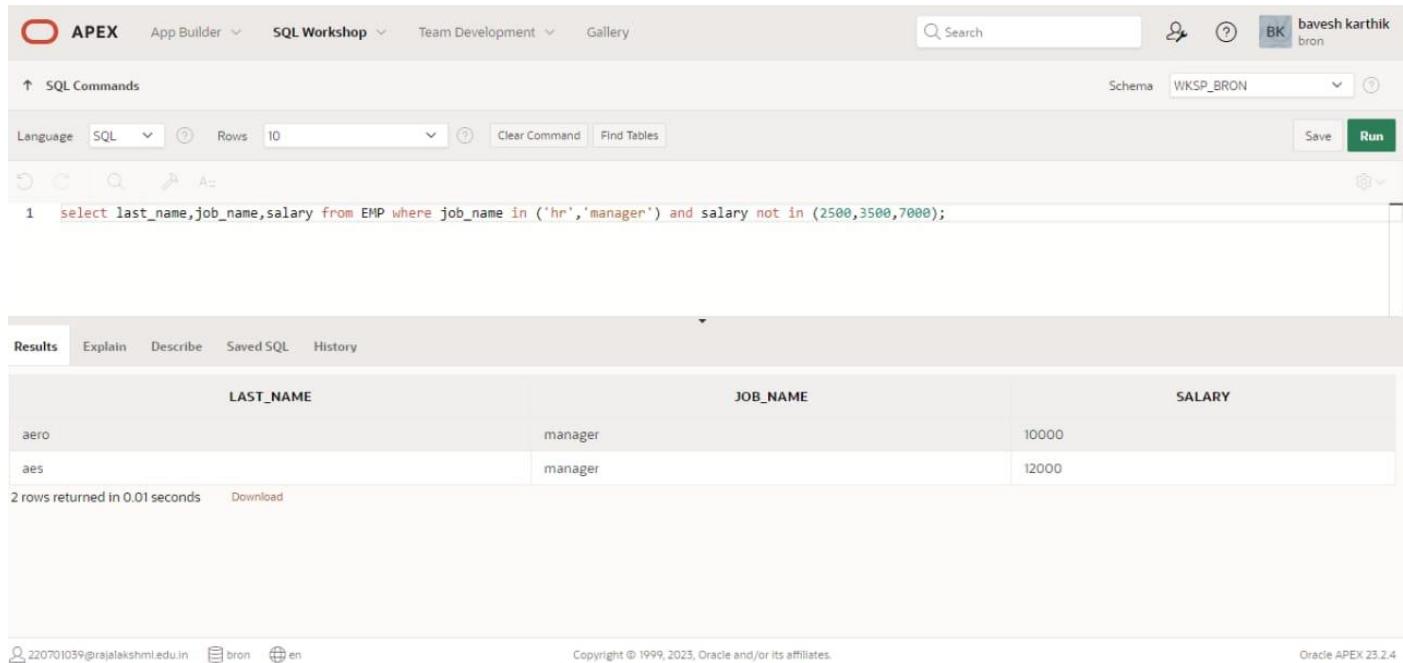
Below the code editor is a results grid. The first row is a header labeled 'LAST_NAME'. The second row contains the value 'aero'. The third row contains the value 'aes'. At the bottom left, it says '2 rows returned in 0.01 seconds'. At the bottom right, there are download and refresh buttons. The footer of the page includes the URL '220701039@rajalakshmi.edu.in', session information 'bron', and language 'en'. It also includes copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

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

QUERY:

```
select last_name,job_name,salary from EMP where job_name in ('hr', 'manager') and salary not in (2500,3500,7000);
```

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 'bavesh karthik' with session ID 'br0n'. The schema selected is 'WKSP_BRON'. The SQL command entered is:

```
1 select last_name,job_name,salary from EMP where job_name in ('hr', 'manager') and salary not in (2500,3500,7000);
```

The results section displays the following data:

LAST_NAME	JOB_NAME	SALARY
aero	manager	10000
aes	manager	12000

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

At the bottom of the page, there is footer information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.', 'Oracle APEX 23.2.4', and user details: '220701039@rajalakshmi.edu.in' and 'br0n en'.

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

QUERY:

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

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 'bavesh karthik bron'. The main area is titled 'SQL Commands' with tabs for Language (SQL selected), Rows (10), Clear Command, Find Tables, Save, and Run. The command input field contains the following SQL query:

```
1 select last_name,salary,commission_pct from employees where commission_pct=0.2;
2
```

The 'Results' tab is selected, showing the message 'no data found'.

At the bottom, the footer includes the URL '220701059@irajalakshmi.edu.in', session information 'bron', and page number 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

RESULT:

SINGLE ROW FUNCTIONS

EX_NO:6

DATE:

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

QUERY:

```
select sysdate from dual;
```

OUTPUT:

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

```
1 select sysdate from dual;
```

In the results section, there is one row returned, showing the current date as 03/12/2024. The results table has a single column labeled 'SYSDATE'.

SYSDATE
03/12/2024

At the bottom of the page, there are footer links for copyright information and Oracle APEX version 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. In the SQL Commands pane, a query is entered:

```
1 select id, last_name, salary, salary+(15.5/100*salary) "new_salary" from my_employee;
```

The Results pane displays the output of the query:

ID	LAST_NAME	SALARY	new_salary
1	Patel	1000	1155
6	Raavi	1122	1295.91
5	Ropebur	12000	13860
3	Drexler	1100	1270.5

Below the results, it says "4 rows returned in 0.05 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 employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary", new_salary-salary as "Increase" from employees;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'bavesh.karthik' with session ID 'WKSP_FROZEN'. The SQL Commands tab is selected, showing the following SQL code:

```
1 select id, last_name, salary, salary+(15.5/100*salary) "new_salary", new_salary-salary as "Increase" from my_employee;
```

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

ID	LAST_NAME	SALARY	new_salary	Increase
1	Patel	1000	1155	-500
6	Raavi	1122	1295.91	-522
5	Ropebur	12000	13860	-4000
3	Drexler	1100	1270.5	-600

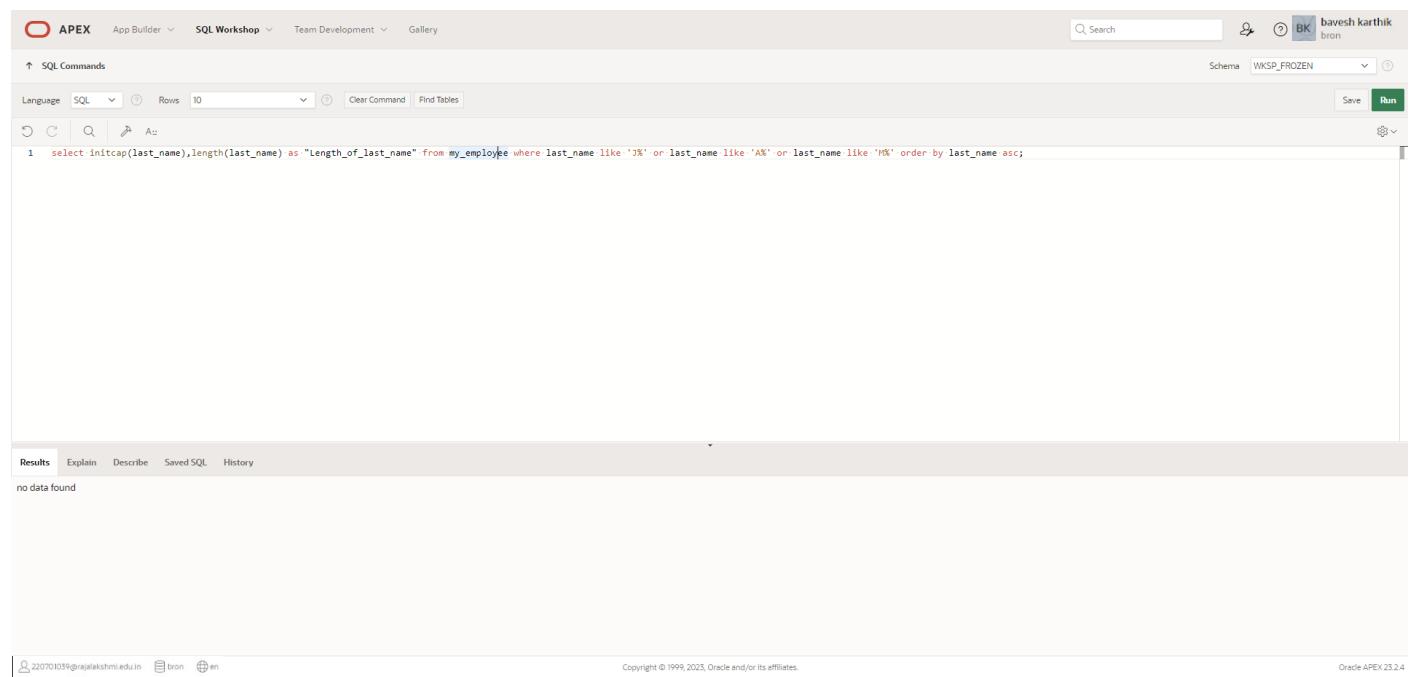
Below the results, it says '4 rows returned in 0.01 seconds' and there is a 'Download' link. The bottom of the page includes copyright information and a footer with user details.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon for 'bavesh karthik', and a schema dropdown set to 'WKSP_FROZEN'. The main area has tabs for 'SQL Commands' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. Below the tabs, there are buttons for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', 'Find Tables', 'Save', and 'Run'. The SQL command entered is:

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

The 'Results' tab is selected, and the message 'no data found' is displayed. At the bottom, the footer includes the URL '220701039@rajalakshmi.edu.in', the session ID 'bron', and the page number '1 of 1'. The copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also present.

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.

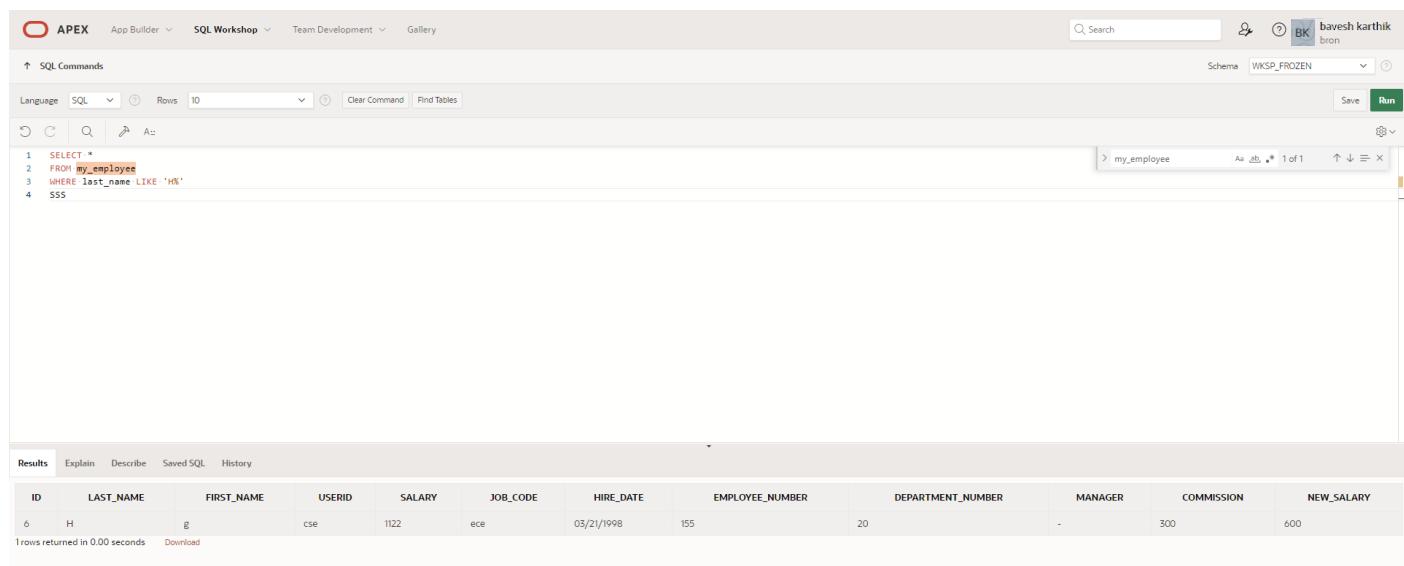
QUERY:

```
SELECT *
```

```
FROM my_employee
```

```
WHERE last_name LIKE 'H%'
```

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 'bavesh karthik' and the workspace 'WKSP_FROZEN'. The main area has tabs for SQL Commands, SQL (selected), Rows (set to 10), and Clear Command/Find Tables. Below these are icons for Undo, Redo, Search, and Autocomplete. The SQL editor contains the following code:

```
1 SELECT *
2 FROM my_employee
3 WHERE last_name LIKE 'H%'
4 SSS
```

The results pane shows a single row from the 'my_employee' table:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY	JOB_CODE	HIRE_DATE	EMPLOYEE_NUMBER	DEPARTMENT_NUMBER	MANAGER	COMMISSION	NEW_SALARY
6	H	G	cse	1122	ece	03/21/1998	155	20	-	300	600

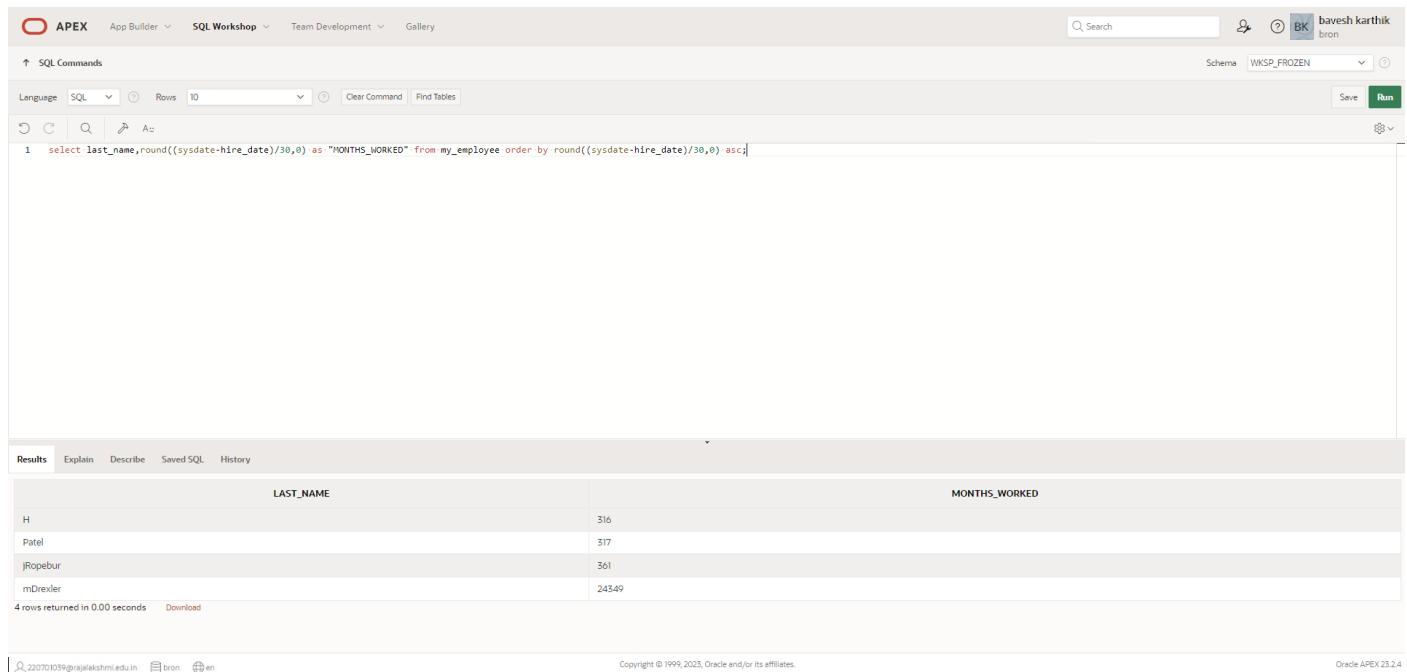
Below the results, it says '1 rows returned in 0.00 seconds' and there is a 'Download' link. The bottom of the page includes copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:



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

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

The results section displays the following data:

LAST_NAME	MONTHS_WORKED
H	316
Patel	317
iRopebur	361
mDrexler	24349

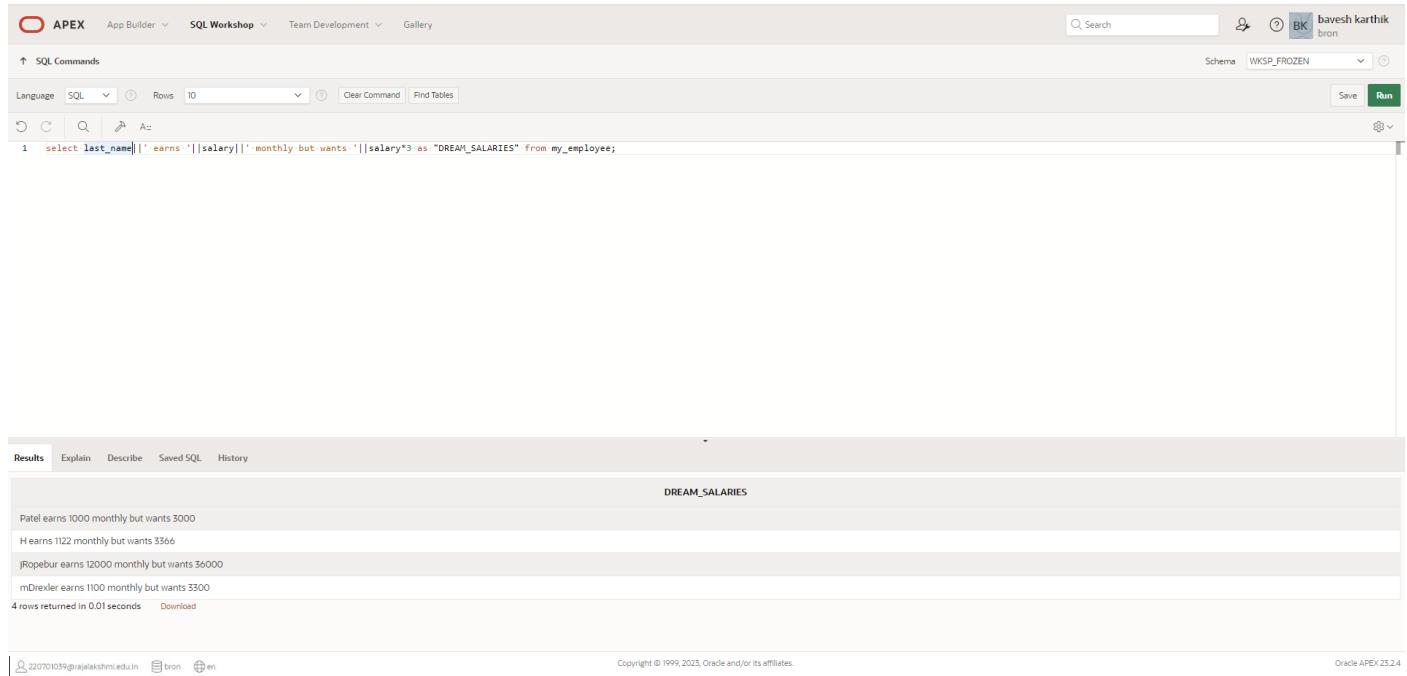
4 rows returned in 0.00 seconds [Download](#)

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The top right shows the user 'bavesh_karthik' and the schema 'WKSP_FROZEN'. The main workspace has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the following SQL code:

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

The results section shows the output of the query:

DREAM_SALARIES
Patel earns 1000 monthly but wants 3000
H earns 1122 monthly but wants 3366
JRopebur earns 12000 monthly but wants 36000
mDrexler earns 1100 monthly but wants 3300

Below the results, it says '4 rows returned in 0.01 seconds' and provides a 'Download' link. The bottom of the page includes copyright information and user details.

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

QUERY:

```
select last_name, lpad(salary,15,'$') as "SALARY" from employees;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'bavesh karthik', and a schema dropdown set to 'WKSP_FROZEN'. Below the toolbar, the SQL editor shows the query: 'select last_name, lpad(salary,15,'\$') as "SALARY" from employees;'. The results section displays the output in a table format:

LAST_NAME	SALARY
Patel	\$\$\$\$\$\$\$\$\$\$1000
H	\$\$\$\$\$\$\$\$\$\$1122
Jopebur	\$\$\$\$\$\$\$\$\$\$12000
mDrexler	\$\$\$\$\$\$\$\$\$\$1100

Below the table, it says '4 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

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

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

The results section displays the following data:

LAST_NAME	HIRE_DATE	REVIEW
Patel	02/20/1998	Monday, the 24 of August, 1998
H	03/21/1998	Monday, the 28 of September, 1998
Ropebur	07/08/1994	Monday, the 09 of January, 1995
mDrexler	04/02/0024	Monday, the 09 of October, 24

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

At the bottom, it shows the user information: 220701039@rajalakshmi.edu.in, bron, en. 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') as Day from employees order by TO_CHAR(hire_date,'Day');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user 'bavesh karthik bron' and the schema 'WKSP_FROZEN'. The main workspace has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, displaying the following SQL query:

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

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

LAST_NAME	HIRE_DATE	DAY
Patel	02/20/1998	Friday
JRopebur	07/08/1994	Friday
H	03/21/1998	Saturday
mDrexler	04/02/0024	Sunday

At the bottom left, it says '4 rows returned in 0.00 seconds'. At the bottom right, it says 'Download' and '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 last_name,department_name, department_number from my_employee
```

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 'bavesh karthik' and a schema dropdown set to 'WKSP_FROZEN'. The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', a single line of SQL is entered: 'select last_name,department_number,department_name from my_employee'. The 'Results' tab displays the output in a grid format with columns 'LAST_NAME', 'DEPARTMENT_NUMBER', and 'DEPARTMENT_NAME'. The data returned is as follows:

LAST_NAME	DEPARTMENT_NUMBER	DEPARTMENT_NAME
Patel	50	CSE
H	20	eee
Ropebur	20	mech
mDrexler	1122	it

Below the results, it says '4 rows returned in 0.00 seconds' and provides a 'Download' link. At the bottom, there are footer links for '22070103%@rajalakshmi.edu.in', 'bron', 'en', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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_code ,department_location_id from my_employees ,departments d where department _number like '%80%';
```

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, a user icon for 'bavesh.karthik', and a schema dropdown set to 'WKSP_FROZEN'. Below the header, the SQL Commands section contains a code editor with the following SQL query:

```

1 select distinct job_code,department_location from my_employee
2 where department_number like '%80%';
3
4

```

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

JOB_CODE	DEPARTMENT_LOCATION
ece	ready
mech	tha

Below the table, it says "2 rows returned in 0.01 seconds". There are also "Download" and "Print" buttons.

At the bottom of the page, there is footer information: "Copyright © 1999-2023, Oracle and/or its affiliates", "Oracle APEX 23.2.4", and a user profile for "bavesh.karthik" with the email "220701039@rajalakshmi.edu.in".

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.location_id,l.city from employees e,departments d,locations l
where e.dept_id = d.dept_id and d.location_id=location_id and e.commission_pct is not null;
```

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, a user icon for 'bavesh karthik', and a schema dropdown set to 'WKSP_FROZEN'. Below the header, the SQL Commands section contains the following code:

```
1 select last_name,department_name,location_id,city from my_employee
2 where commission is not null;
```

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

LAST_NAME	DEPARTMENT_NAME	LOCATION_ID	CITY
Patel	cse	600001	chennai
H	eee	600002	chennai
JRopebur	mech	600003	chennai
mDrexler	it	600004	chennai

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

At the bottom, there are footer links for 220701039@rajalakshmi.edu.in, bron, and en, along with copyright information: Copyright © 1999-2023, Oracle and/or its affiliates. Oracle APEX 23.2.4.

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

QUERY:

Select last_name,dept_name from employees,departments where employees.dept_id=departments.dept_id
And last_name like '%a%';

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables

```
1 select last_name,department_name from my_employee
2 where last_name like '%p%';
```

Save Run

Results Explain Describe Saved SQL History

LAST_NAME	DEPARTMENT_NAME
patel	cse
JRopebur	mech

2 rows returned in 0.01 seconds Download

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

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

QUERY:

```
Select e.last_name,e.job_id,e.dept_id,d.dept_name from employees e join departments d on (e.dept_id=d.dept_id)
Join locations l on(d.location_id=l.location_id) where lower(l.city)='toronto';
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables

```
1 select last_name,job_code,department_number,department_name from my_employee
2 where work like '%toronto%';
```

Save Run

Results Explain Describe Saved SQL History

LAST_NAME	JOB_CODE	DEPARTMENT_NUMBER	DEPARTMENT_NAME
patel	CSE	50	CSE
mDrexler	eee	1122	IT

2 rows returned in 0.01 seconds Download

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

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

QUERY:

```
Select w.last_name "Employee",w.emp_id "EMP#",m.last_name "Manager",m.emp_id "Mgr#"  
From employees w join employees m on (w.manager_id=m.emp_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 search bar, a user icon, and a session identifier 'bavesh karthik bron'. Below the toolbar, the schema is set to 'WKSP_FROZEN'. The main workspace contains a SQL command editor with the following code:

```
1 select last_name,job_code,department_number,department_name from my_employee  
2 where work like '%toronto%';
```

Below the editor, the results tab is selected, displaying the output of the query:

LAST_NAME	JOB_CODE	DEPARTMENT_NUMBER	DEPARTMENT_NAME
patel	cse	50	cse
mDrexler	eee	1122	it

2 rows returned in 0.01 seconds. There are also 'Download' and 'History' buttons.

At the bottom of the page, there are footer links for 220701039@rajalakshmi.edu.in, bron, en, and en. The copyright notice reads 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version is 'Oracle APEX 23.2.4'.

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

QUERY:

```
Select last_name"Employee",employee-id"EMP#",last_name "Manager",employee_id "Mgr#" from my_employee  
Left outer join employees m on(w.manager_id=m.emp_id);
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Language SQL Rows 10 Clear Command Find Tables

```

1 SELECT w.last_name "Employee", w.emp_id "EMP#",
2      m.last_name "Manager", m.emp_id "Mgr#"
3   FROM employees w
4  LEFT OUTER JOIN employess m
5    ON (w.manager_id = m.emp_id);
6
7

```

Results Explain Describe Saved SQL History

Employee	EMP#	Manager	Mgr#
vegan	3	roy	2
roy	1	rayan	1
rayan	2	rayan	1

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

220701039@rajalakshmi.edu.in bron en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

8.Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label

QUERY:

```
Select dept_id department,e.last_name employee,c.last_name colleague from employees e join employees c
On (e.deot_id=c.dept_id) where e.emp_id<>c.emp_id order by e.dept_id,e.last_name,c.last.name;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Language SQL Rows 10 Clear Command Find Tables

```

1 Select e.dept_id department,e.last_name employee,c.last_name colleague from employees e join employees c
2 On (e.deot_id=c.dept_id) where e.emp_id<>c.emp_id order by e.dept_id,e.last_name,c.last.name;
3

```

Results Explain Describe Saved SQL History

DEPARTMENT_NUMBER	LAST_NAME	DEPARTMENT_NUMBER
50	patel	50
80	H	80
80	jRopebur	80
112	mDrexler	112

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

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

QUERY:

```
Select e.last_name,e.job_id,d.dept_name,e.salary,j.grade_level from employees e join departments d  
on (e.dept_id=d.dept_id) join job_grades j on (e.salary between j.low_sal and j.high_sal);
```

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 'bavesh karthik' and a schema dropdown set to 'WKSP_FROZEN'. The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the following SQL code:

```
1 Select e.last_name,e.job_id,d.dept_name,e.salary,j.grade_level from employees e join departments d  
2 on (e.dept_id=d.dept_id) join job_grades j on (e.salary between j.low_sal and j.high_sal);  
3
```

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

LAST_NAME	JOB_CODE	DEPARTMENT_NAME	SALARY	GRADE_LEVEL
patel	cse	cse	1000	2
H	ece	eee	1122	42
Ropebur	mech	mech	12000	4
mDrexler	eee	it	1100	63

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

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 employes e join employes.davies on davies.hire_date < e.hire_date

Where davies.last_name = 'Davies';

OUTPUT:

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

```
1 Select e.last_name,e.hire_date from employes e join employes.davies on davies.hire_date < e.hire_date
2 Where davies.last_name = 'Davies';
3
```

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

LAST_NAME	HIRE_DATE
H	03/21/1998
davies	07/08/1994
mDrexler	04/02/0024
patel	02/20/1998

At the bottom left, it says '4 rows returned in 0.01 seconds'. At the bottom right, it says 'Download'.

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

QUERY:

```
select last-name as Employee,e.hire_date as Emp_hired,e.manager_name as manager,m.hire_date as
mgr_hired from my_employee where e.hire_date < m.hire_date;
```

OUTPUT:

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables Schema: WKSP_DATABASE04 Save Run

```

1 SELECT e.last_name AS Employee, e.hire_date AS Emp_Hired,
2 e.man_name AS Manager, m.hire_date AS Mgr_Hired
3 FROM employees e
4 JOIN employees m ON e.man_name = m.last_name
5 WHERE e.hire_date < m.hire_date;

```

Results Explain Describe Saved SQL History

EMPLOYEE	EMP_HIRED	MANAGER	MGR_HIRED
davies	02/26/1998	davies	02/05/1999

1 rows returned in 0.00 seconds Download

Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 25.2.

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

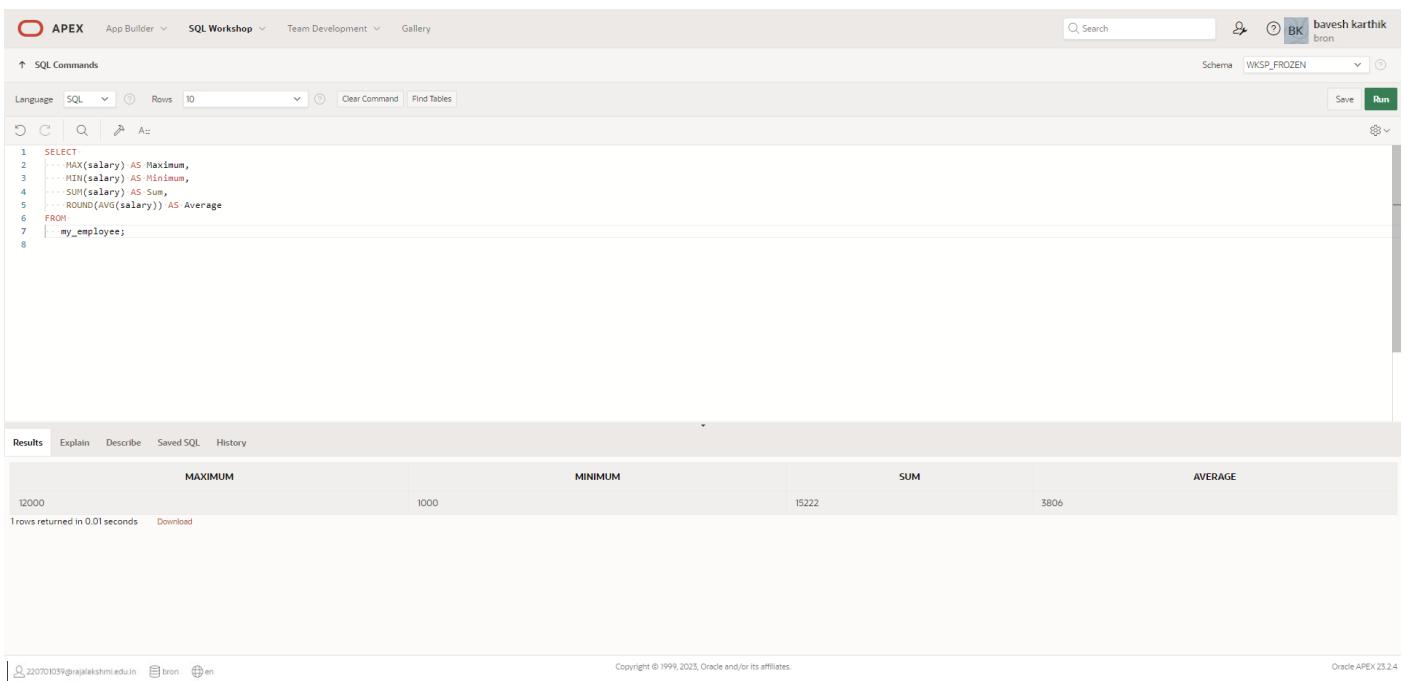
4. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number

QUERY:

SELECT

```
MAX(salary) AS Maximum, MIN(salary) AS Minimum,
SUM(salary) AS Sum, ROUND(AVG(salary)) AS Average FROM my_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. The right side shows the user's name 'bavesh karthik' and session information 'WKSP_FROZEN'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
1 SELECT
2   MAX(salary) AS Maximum,
3   MIN(salary) AS Minimum,
4   SUM(salary) AS Sum,
5   ROUND(AVG(salary)) AS Average
6 FROM
7   my_employee;
```

The Results tab displays the output of the query:

	MAXIMUM	MINIMUM	SUM	AVERAGE
12000	1000	15222	3806	

Below the table, it says '1 rows returned in 0.01 seconds' and there is a 'Download' link. At the bottom, there are footer links for copyright (Oracle 1999-2023), user information ('220701039@rajalakshmi.edu.in'), and language ('en').

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

QUERY:

SELECT

```
    job_code, MAX(salary) AS Maximum, MIN(salary) AS Minimum, SUM(salary) AS Sum  
    ROUND(AVG(salary)) AS Average FROM my_employee GROUP BY job_code;
```

OUTPUT:

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

```
1 SELECT  
2     job_code,  
3     MAX(salary) AS Maximum,  
4     MIN(salary) AS Minimum,  
5     SUM(salary) AS Sum,  
6     ROUND(AVG(salary)) AS Average  
7  FROM  
8      my_employee  
9  GROUP BY  
10     job_code;
```

The results section displays the following table:

JOB_CODE	MAXIMUM	MINIMUM	SUM	AVERAGE
CSE	1000	1000	1000	1000
EEE	1100	1100	1100	1100
Mech	12000	12000	12000	12000
ECE	1122	1122	1122	1122

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

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

QUERY:

```
Select job_code, count(*) from my_employee where job_code='cse' group by job_code;
```

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, a user icon for 'bavesh karthik bron', and a dropdown for 'Schema' set to 'WKSP_FROZEN'. Below the tabs, there are buttons for Language (SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The main area contains a SQL command: 'Select job_code, count(*) from my_employee where job_code='cse' group by job_code;'. The results section shows a single row with 'CSE' in the 'JOB_CODE' column and '1' in the 'COUNT(*)' column. Below the results, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. The bottom of the page includes copyright information ('Copyright © 1999, 2023, Oracle and/or its affiliates.') and a footer with user details ('220701039@rajalakshmi.edu.in', 'bron', 'en').

JOB_CODE	COUNT(*)
CSE	1

7.Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers.

QUERY:

```
SELECT COUNT(DISTINCT manager_id) AS "Number of managers" FROM my_employee;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables

```
1 SELECT COUNT(DISTINCT manager_id) AS "Number of managers" FROM my_employee;
2
```

Results Explain Describe Saved SQL History

Number of managers
4

1 rows returned in 0.02 seconds Download

220701039@rajalakshmi.edu.in bron en Copyright © 1999, 2023, Oracle and/or its affiliates. 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 my_employee;

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables

```
1 Select max(salary)-min(salary) difference from my_employee;
```

Results Explain Describe Saved SQL History

DIFFERENCE
11000

1 rows returned in 0.01 seconds Download

220701039@rajalakshmi.edu.in bron en Copyright © 1999, 2023, Oracle and/or its affiliates. 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 manager_id,min(salary) from my_employee where manager_id is not null group by manager_id having min(salary)>6000 order by min(salary) desc
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'bavesh.karthik' with schema 'WKSP_FROZEN'. The SQL tab is selected, displaying the following SQL code:

```
1 Select manager_id,min(salary) from my_employee where manager_id is not null group by manager_id
2 having min(salary)>6000 order by min(salary) desc
3
```

The results section shows a single row of data:

MANAGER_ID	MIN(SALARY)
456	12000

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

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

QUERY:

```
Select count(*) as
total,sum(decode(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 empa;
```

OUTPUT:

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

- SQL Commands**: The tab is selected.
- Language**: SQL
- Rows**: 10
- Schema**: WKSP_DATABASE04
- Run** button

The SQL code entered is:

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

The results section displays the following data:

	TOTAL	1995	1996	1997	1998
5		1	1	1	1

1 rows returned in 0.01 seconds

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

QUERY:

```
Select job_id "job", sum(decode(dept_id,20,salary))"dept20", sum(decode(dept_id,50,salary))"dept50",
sum(decode(dept_id,80,salary))"dept80", sum(decode(dept_id,90,salary))"dept90" from empa;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

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

```

4   SUM(CASE WHEN department_number = 50 THEN salary ELSE 0 END) AS "Department 50 Salary",
5   SUM(CASE WHEN department_number = 80 THEN salary ELSE 0 END) AS "Department 80 Salary",
6   SUM(CASE WHEN department_number = 90 THEN salary ELSE 0 END) AS "Department 90 Salary",
7   SUM(salary) AS "Total Salary"
8 FROM my_employee
9 WHERE
10    department_number IN (20, 50, 80, 90)
11 GROUP BY
12    job_code;
13
14

```

Results Explain Describe Saved SQL History

Job	Department 20 Salary	Department 50 Salary	Department 80 Salary	Department 90 Salary	Total Salary
Cse	1000	0	0	0	1000
eee	0	0	0	1100	1100
mech	0	0	12000	0	12000
ece	0	1122	0	0	1122

4 rows returned in 0.01 seconds Download

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 "dept_name",d.loc as "department location",count(*) as "Number of people",round(avg(e.salary),2) as "salary" from departments d inner join emp a on (d.dept_id=e.dept_id)
Group by d.dept_name,d.loc;
```

OUTPUT:

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables Schema: WKSP_DATABASE04 Save Run

```

1 SELECT
2     d.dept_name AS "dept_name",
3     d.loc AS "department location",
4     COUNT(*) AS "Number of people",
5     ROUND(AVG(e.salary), 2) AS "salary"
6 FROM departments d
7 INNER JOIN
8     emp a ON (d.dept_id = e.dept_id)
9 GROUP BY
10    d.dept_name,
11    d.loc;
12

```

Results Explain Describe Saved SQL History

dept_name	department location	Number of people	salary
finance	chennai	2	52500
marketing	bangalore	1	70000
sales	vellore	2	52500

3 rows returned in 0.05 seconds Download

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

RESULT:

SUB QUERIES

EX_NO:9

DATE:

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

QUERY:

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

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 search bar, a user icon for 'bavesh karthik', and a schema dropdown set to 'VKSP_FROZEN'. The main workspace is a SQL editor with the following content:

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

The results tab at the bottom shows the message 'no data found'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. 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, a user icon for 'bavesh karthik', and a schema dropdown set to 'WKSP_FROZEN'. Below the header, the 'SQL Commands' tab is selected. The SQL editor contains the following query:

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

Below the editor, there are buttons for Save and Run. The results section shows a single row of data:

EMPLOYEE_ID	LAST_NAME	SALARY
78797987	davies	12000

Below the table, it says '1 rows returned in 0.02 seconds' and has a 'Download' link. The bottom of the page includes copyright information and a footer for Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon (BK), and the name 'bavesh karthik'. Below the tabs, there are buttons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run.

The main area contains a code editor with the following SQL query:

```

1 SELECT Employee_id, Last_Name
2 FROM my_Employee
3 WHERE Department_id IN (
4 ...  SELECT Department_id
5 ... FROM my_Employeed
6 ... WHERE Last_Name LIKE '%u%'
7 );
8

```

Below the code editor is a results grid with two rows. The columns are labeled 'EMPLOYEE_ID' and 'LAST_NAME'. The first row has values 8786547564 and U. The second row has values 78779987 and daviesu.

At the bottom left, it says '2 rows returned in 0.01 seconds' and there's a 'Download' link. At the bottom right, it says 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there's a search bar, user information for 'VG vaitheswaran g a frozen', and a 'Run' button. Below the navigation, the 'SQL Commands' tab is active, showing the following SQL code:

```
1 SELECT Last_Name, Department_ID, Job_code
2 FROM my_Employee
3 WHERE Department_ID = '1700';
4
```

Under the 'Results' tab, the output is displayed in a table:

LAST_NAME	DEPARTMENT_ID	JOB_CODE
U	1700	ece

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

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

QUERY:

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

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, a user icon for 'bavesh karthik', and a session identifier 'WKSP_FROZEN'. Below the header, the SQL Commands tab is selected. The SQL editor contains the following code:

```
1 select last_name,salary from my_employee where manager_id=(select manager_id from
2 my_employee where manager_lastName='King');
```

The results section below the editor shows the message "no data found".

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. 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, a user icon for 'bavesh karthik', and a session identifier 'WKSP_FROZEN'. Below the header, the SQL Commands tab is selected. The SQL editor contains the following query:

```
1 select department_id, last_name, job_code from my_employee where department_id in (select department_id from departments where dept_name='Executive');
```

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

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables

```
1 select department_id, last_name, job_code from my_employee where department_id in (select department_id from departments where dept_name='Executive');
```

Save Run

Schema: WKSP_FROZEN

Results Explain Describe Saved SQL History

no data found

220701039@rajalakshmi.edu.in bron 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:

USING THE SET OPERATORS

EX_NO:10

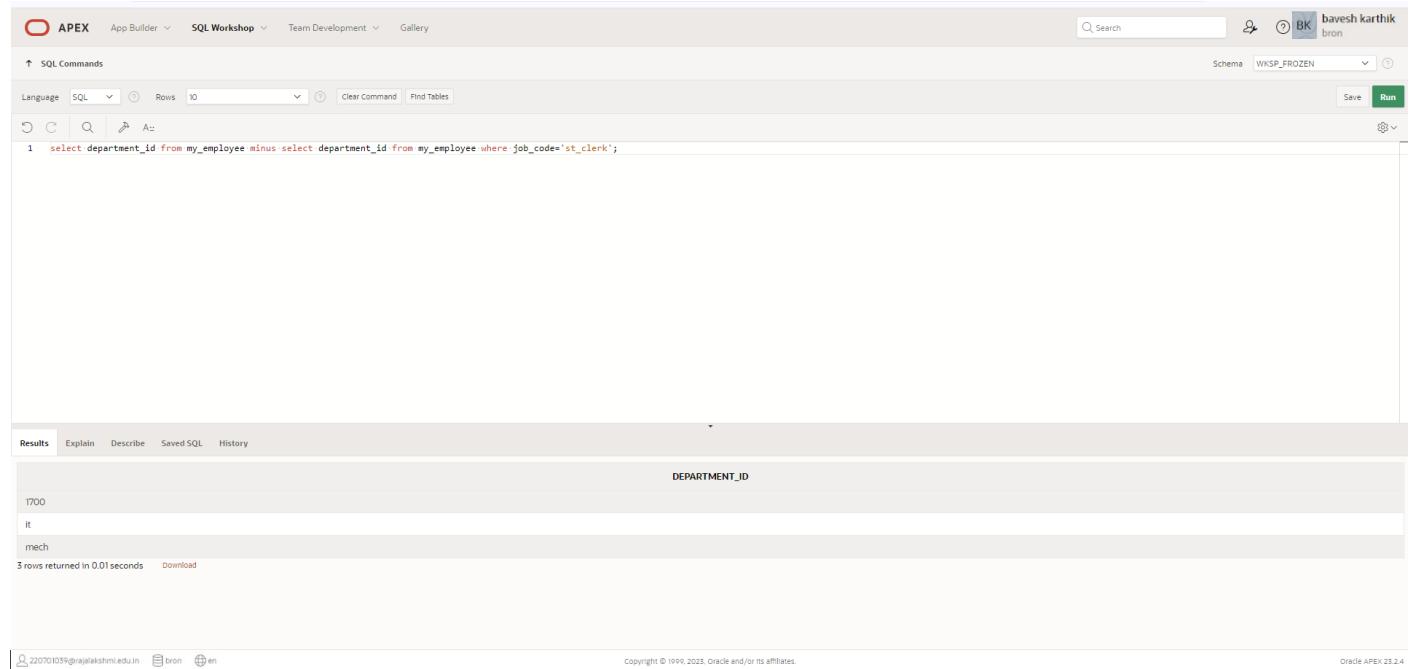
DATE:

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

QUERY:

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

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 'bavesh karthik' and a search bar. The main workspace has a toolbar with various icons. Below the toolbar, the SQL command is entered:

```
1 select department_id from my_employee minus select department_id from my_employee where job_code='st_clerk';
```

The results section shows the output:

DEPARTMENT_ID
1700
IT
MECH

Below the table, it says "3 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,state_province from location minus select country_id,state_province from  
location,departments where location.location_id=departments.location_id;
```

OUTPUT:

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

QUERY:

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

OUTPUT:

- 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 job_id,employee_id from employees intersect select e.job_id,e.employee_id from  
employees e,job_history j where e.job_id=j.old_job_id;
```

OUTPUT:

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 first_name||' '||last_name as "Name",department_id from employees union all select  
dept_name,dept_id from departments;
```

OUTPUT:

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

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

RESULT:

CREATING VIEWS

EX_NO:11

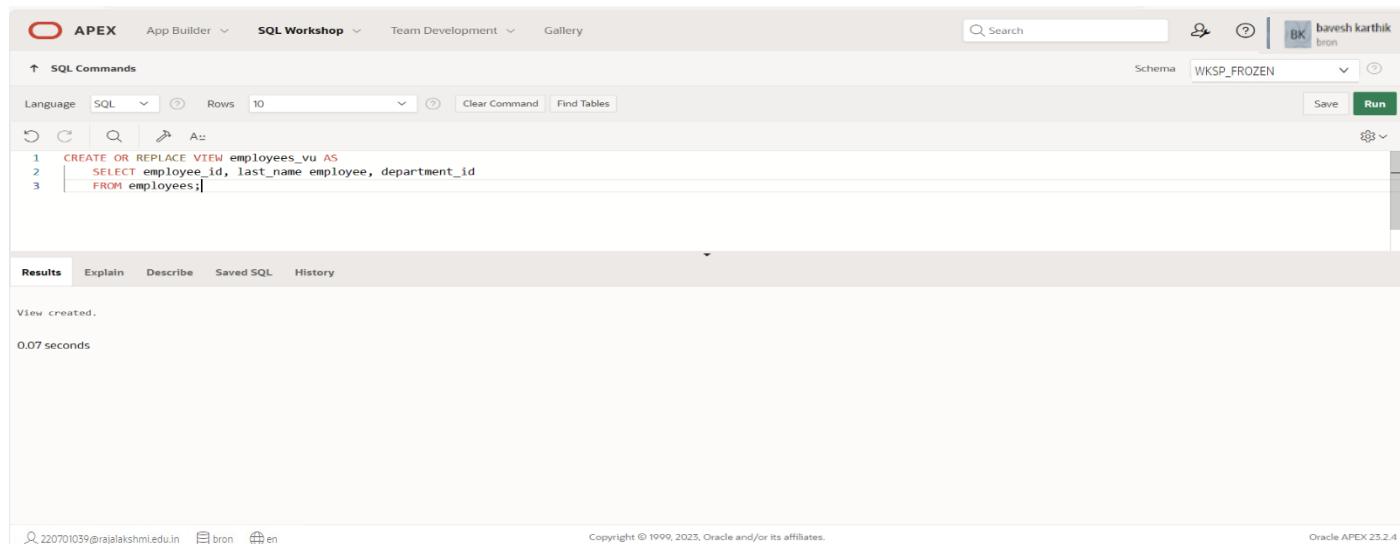
DATE:

- 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. In the top navigation bar, 'SQL Workshop' is selected. The main area displays the following SQL code:

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

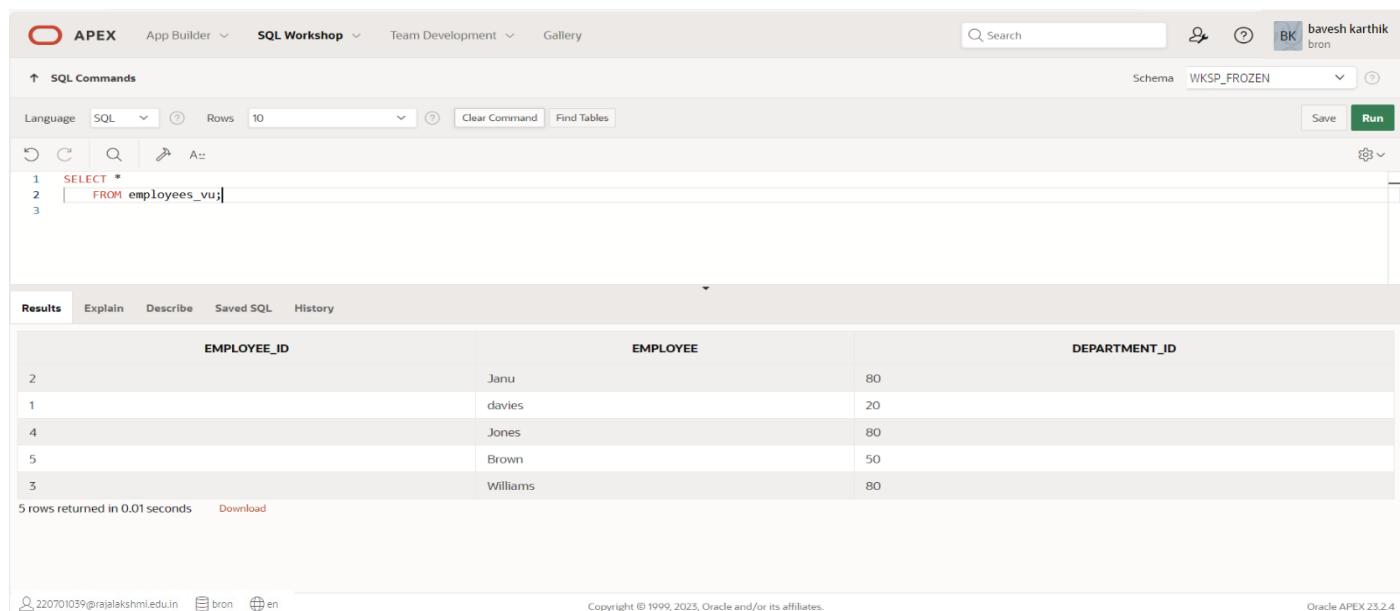
Below the code, the 'Results' tab is active, showing the message "View created." and a execution time of "0.07 seconds". The bottom status bar indicates the user is connected to "220701059@rajalakshmi.edu.in" as "bron" and the session is "en".

- 2.) Display the contents of the EMPLOYEES_VU view.

QUERY:

```
select * from employees_vu;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface with the 'Results' tab active. The results table displays the following data:

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
2	Janu	80
1	davies	20
4	Jones	80
5	Brown	50
3	Williams	80

Below the table, it says "5 rows returned in 0.01 seconds". The bottom status bar indicates the user is connected to "220701059@rajalakshmi.edu.in" as "bron" and the session is "en".

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 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The schema dropdown is set to 'WKSP_FROZEN'. The main area has tabs for 'SQL Commands' and 'Results'. In the 'SQL Commands' tab, the query is displayed:

```
1 | SELECT view_name, text
2 | FROM user_views;
3 |
```

In the 'Results' tab, the output is shown in two columns: 'VIEW_NAME' and 'TEXT'. The single row returned is:

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

Below the table, it says '1 rows returned in 0.04 seconds' and there is a 'Download' link.

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

QUERY:

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

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 schema dropdown is set to 'WKS' and the user is 'baivesh karthik'. The main area has tabs for 'SQL Commands' and 'Results'. In the 'SQL Commands' tab, the query is displayed:

```
1 | SELECT employee, department_id
2 | FROM employees_vu;
3 |
```

In the 'Results' tab, the output is shown in two columns: 'EMPLOYEE' and 'DEPARTMENT_ID'. The five rows returned are:

EMPLOYEE	DEPARTMENT_ID
Janu	80
davies	20
Jones	80
Brown	50
Williams	80

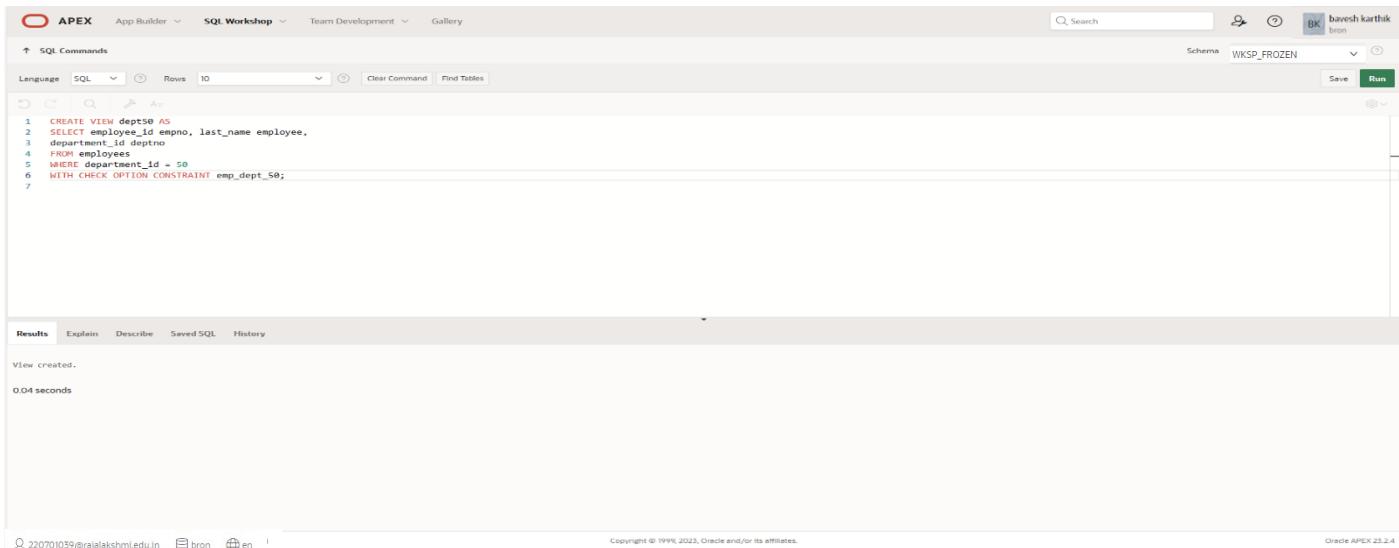
Below the table, it says '5 rows returned in 0.01 seconds' and there is a 'Download' link.

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. In the top navigation bar, 'SQL Workshop' is selected. The schema dropdown shows 'WKSP_FROZEN'. The main area displays the SQL command for creating the view:

```
1 CREATE VIEW dept50 AS
2  SELECT employee_id empno, last_name employee,
3  department_id deptno
4  FROM employees
5  WHERE department_id = 50
6  WITH CHECK OPTION CONSTRAINT emp_dept_50;
7
```

Below the command, the results show:

View created.
0.04 seconds

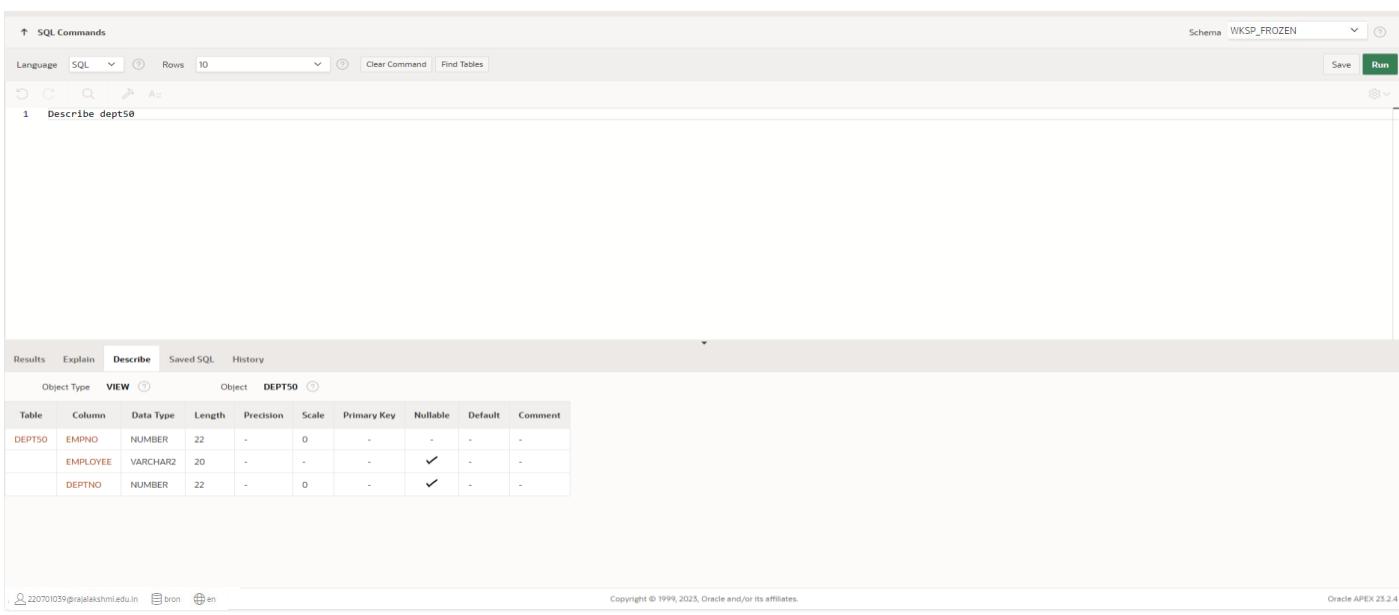
At the bottom, the URL is 220701039@rajalakshmi.edu.in, and the page footer indicates Copyright © 1999, 2023, Oracle and/or its affiliates.

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

QUERY:

```
Describe dept50;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The schema dropdown shows 'WKSP_FROZEN'. The main area displays the SQL command for describing the view:

```
1 Describe dept50
```

Below the command, the results show:

Object Type: VIEW Object: DEPT50

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT50	EMPNO	NUMBER	22	-	0	-	-	-	-
	EMPLOYEE	VARCHAR2	20	-	-	-	✓	-	-
	DEPTNO	NUMBER	22	-	0	-	✓	-	-

At the bottom, the URL is 220701039@rajalakshmi.edu.in, and the page footer indicates Copyright © 1999, 2023, Oracle and/or its affiliates.

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

```
1 UPDATE dept50
2 SET deptno=80
3 WHERE employee='Matos';
4
```

Below the command, the results section shows:

0 row(s) updated.
0.05 seconds

At the bottom of the interface, the copyright notice reads "Copyright © 1999, 2023, Oracle and/or its affiliates." and 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. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is active. The main area contains the following SQL command:

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

Below the command, the results section shows:

View created.
0.04 seconds

At the bottom of the interface, the copyright notice reads "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version "Oracle APEX 23.2.4".

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

RESULT:

EXERCISE 12

PRACTICE QUESTIONS

Intro to Constraints; NOT NULL and UNIQUE Constraints

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

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

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

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

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

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

3. Why is it important to give meaningful names to constraints?
- If a constraint is violated in a SQL statement execution, it is easy to identify the cause with user-named constraints.
 - It is easy to alter names/drop constraint.
4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.

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

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

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

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

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

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

Table Created.

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

```
DESCRIBE f_global_locations;
```

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

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

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

PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

1. What is the purpose of a
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK CONSTRAINT
 - a. **PRIMARY KEY**
Uniquely identify each row in table.
 - b. **FOREIGN KEY**
Referential integrity constraint links back parent table's primary/unique key to child table's column.
 - c. **CHECK CONSTRAINT**
Explicitly define condition to be met by each row's fields. This condition must be returned as true or unknown.
-
2. Using the column information for the animals table below, name constraints where applicable at the table level, otherwise name them at the column level. Define the primary key (animal_id). The license_tag_number must be unique. The admit_date and vaccination_date columns cannot contain null values.

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

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

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

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

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

```
SELECT * FROM animals;
```

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

ANIMAL_ID	NAM E	LICENSE_TAG_NUMBE R	ADMIT_DATE	ADOPTION_ID	VACCINATION_DATE
101	Spot	35540	10-Oct-2004	205	12-Oct-2004

added to the animals table.

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

- a. ON DELETE CASCADE

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

- b. ON DELETE SET NULL

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

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

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

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

PRACTICE PROBLEM

Managing Constraints

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

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

1. What are four functions that an ALTER statement can perform on constraints?
 - ADD
 - DROP
 - ENABLE
 - DISABLE
2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy_d_clients table. Name the primary key copy_d_clients_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy_d_clients.table?

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

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

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

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

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

- a. The constraint name for the primary key in the copy_d_clients table is _____.

COPY_D_CLT_CLIENT_NUMBER_PK

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

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

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

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

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

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

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

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

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

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

1 row(s) inserted.

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

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

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

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

1 row(s) deleted.

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

Table altered.

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

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

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

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

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

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

EXERCISE 13

Creating Views

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 ASSELECT title, artistFROM 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, salaryFROM 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_idFROM(SELECT dpt.department_id,MAX(NVL(emp.salary,0)) max_dpt_salFROM departments dpt LEFT OUTER JOIN employees empON dpt.department_id = emp.department_idGROUP BY dpt.department_id) dptmx LEFT OUTER JOINemployees empm ON dptmx.department_id = empm.department_idWHERE 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, salaryFROM(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_idxON 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

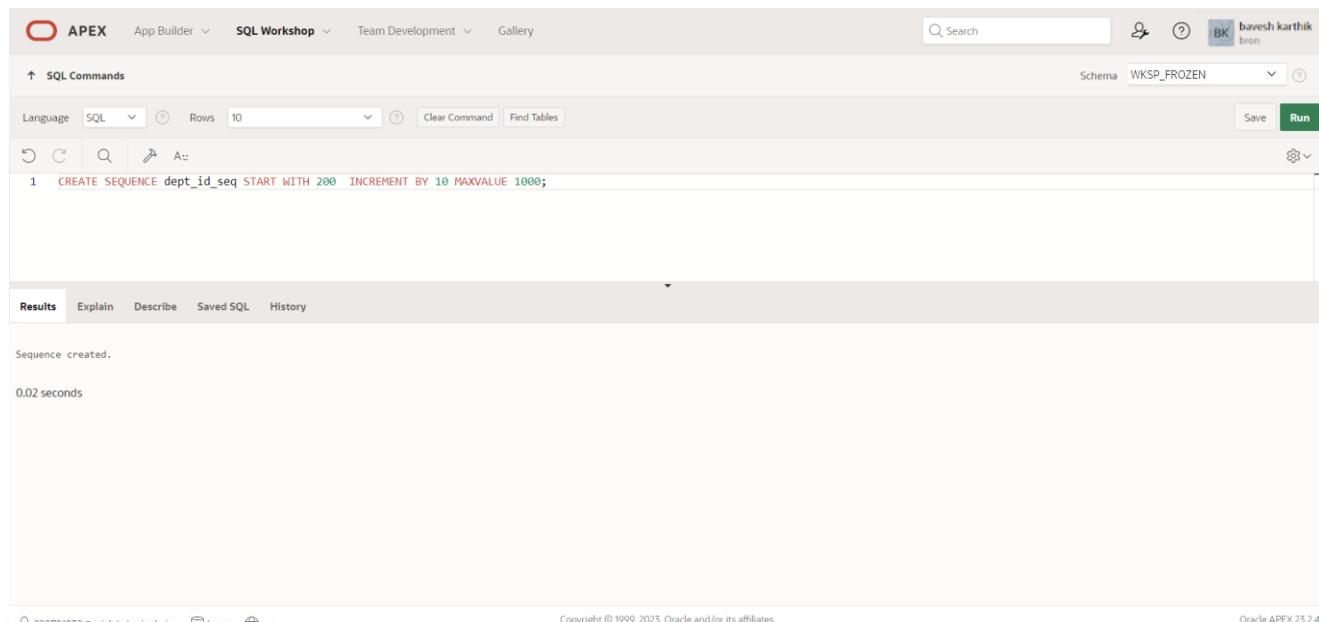
DATE:

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. In the top navigation bar, 'APEX' is selected. The main area contains the following SQL command:

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

Below the command, the results pane displays the output:

```
Sequence created.
```

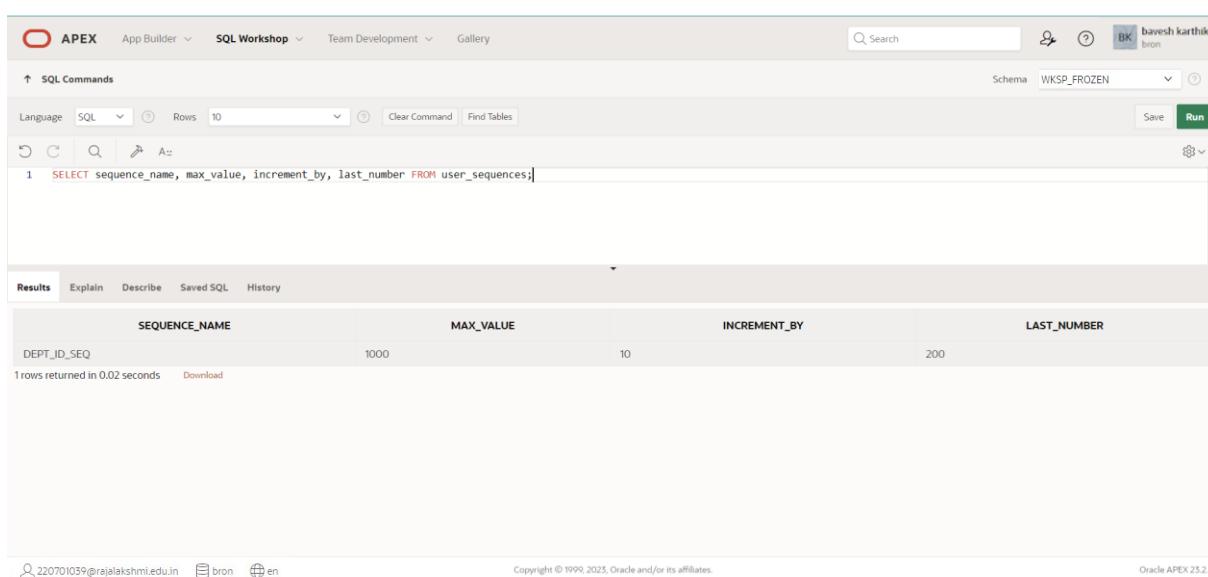
At the bottom of the interface, it shows the user information: 220701039@rajalakshmi.edu.in, bron, en, and the copyright notice: Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4.

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. In the top navigation bar, 'APEX' is selected. The main area contains the following SQL command:

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

Below the command, the results pane displays the output:

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_ID_SEQ	1000	10	200

At the bottom of the interface, it shows the user information: 220701039@rajalakshmi.edu.in, bron, en, and the copyright notice: Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4.

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

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 search bar, a user icon for 'bavesh karthik bron', and a sign-out link. The main area is titled 'SQL Scripts \ Results'. It displays the execution of a script named 'ex14' which is marked as 'Complete'. The summary table shows one row inserted with an elapsed time of 0.02 seconds. Below this, three summary metrics are shown: 1 statement processed, 1 successful, and 0 with errors. At the bottom, it shows the user's details (220701039@rajalakshmi.edu.in, bron, en), the copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 25.2.4).

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:

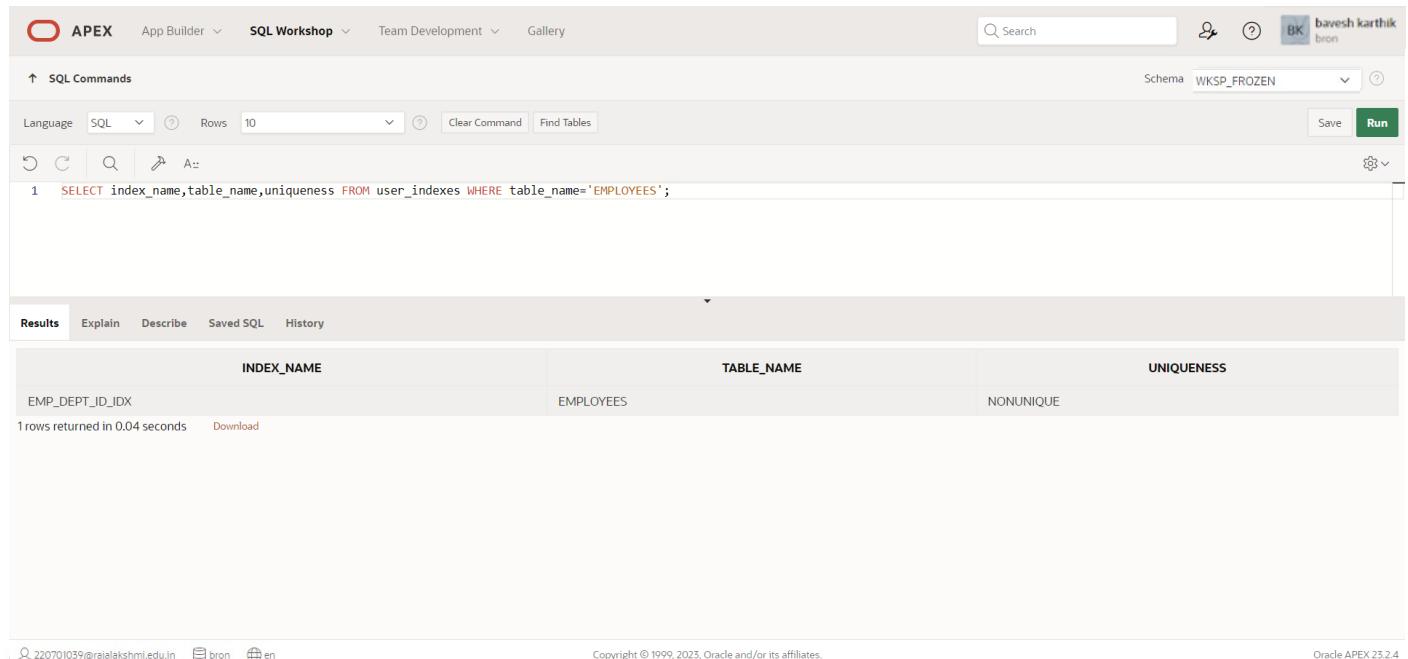
The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, a schema dropdown set to 'WKSP_FROZEN', and a sign-out link. The main area is titled 'SQL Commands'. A single command is entered in the text area: 'CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);'. Below the command, the results tab is selected, showing the message 'Index created.' and a timestamp of '0.03 seconds'. At the bottom, it shows the user's details (220701039@rajalakshmi.edu.in, bron, en), the copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 25.2.4).

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 includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'bavesh karthik bron', and a schema dropdown set to 'WKSP_FROZEN'. The main workspace contains a SQL command line with the query: 'SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';'. Below the command line is a results grid with three columns: INDEX_NAME, TABLE_NAME, and UNIQUENESS. The single row returned is 'EMP_DEPT_ID_IDX' under INDEX_NAME, 'EMPLOYEES' under TABLE_NAME, and 'NONUNIQUE' under UNIQUENESS. The bottom of the page shows copyright information and the version 'Oracle APEX 23.2.4'.

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

1 rows returned in 0.04 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:

CONTROLLING USER ACCESS

EX_NO:15

DATE:

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

The CREATE SESSION system privilege

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

The CREATE TABLE privilege

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

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

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

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

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

The ALTER USER statement

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

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

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

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

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

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

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

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

```
SELECT table_name FROM user_tables;
```

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

Team 1 revokes the privilege.

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

Team 2 revokes the privilege.

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

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

Team 1 executes this INSERT statement.

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

Team 2 executes this INSERT statement.

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

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

RESULT:

PL/SQL CONTROL STRUCTURES

EX NO:16

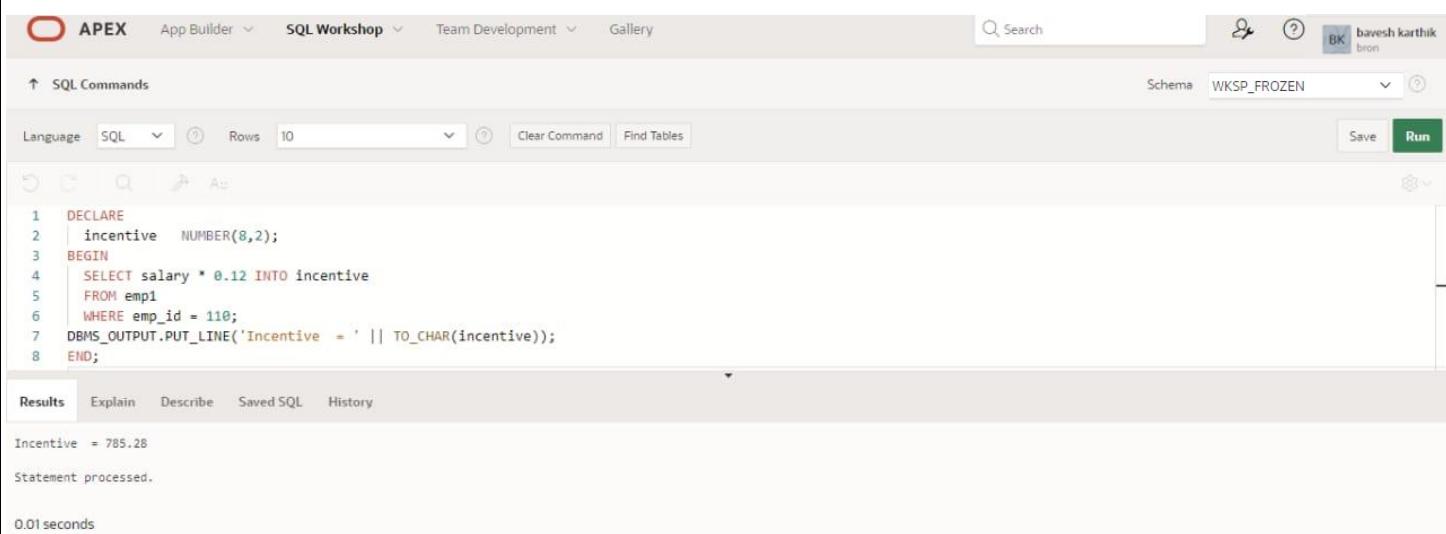
DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a user profile for 'bavesh karthik bron'. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_FROZEN'. The code editor contains the provided PL/SQL block. The results tab shows the output: 'Incentive = 785.28', 'Statement processed.', and a execution time of '0.01 seconds'.

```
1  DECLARE
2      incentive NUMBER(8,2);
3  BEGIN
4      SELECT salary * 0.12 INTO incentive
5      FROM emp1
6      WHERE emp_id = 110;
7      DBMS_OUTPUT.PUT_LINE('Incentive = ' || TO_CHAR(incentive));
8  END;
```

Incentive = 785.28
Statement processed.
0.01 seconds

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

QUERY:

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

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

OUTPUT:

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

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

An error message is displayed in a yellow box:

```
Error at line 4/25: ORA-06550: line 4, column 25:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 3:
PL/SQL: Statement ignored
```

The error message points to line 4, column 25, where the identifier 'Welcome' is used without being declared.

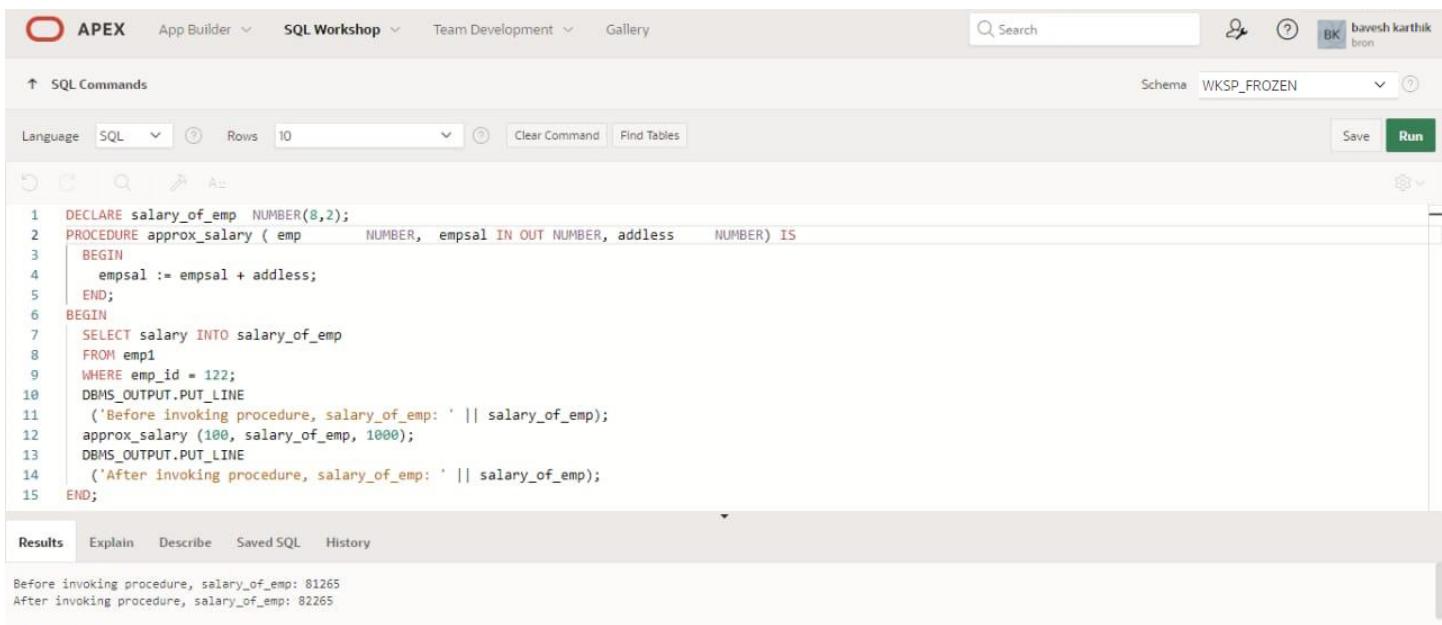
3.) Write a PL/SQL block to adjust the salary of the employee whose ID 122.

QUERY:

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

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

OUTPUT:



The screenshot shows the Oracle 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 'bavesh karthik bron'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The code area contains the PL/SQL block from the previous step. The results tab at the bottom shows the output of the DBMS_OUTPUT.PUT_LINE statements.

```
1 DECLARE salary_of_emp NUMBER(8,2);
2 PROCEDURE approx_salary ( emp      NUMBER, empsal IN OUT NUMBER, addless  NUMBER) IS
3 BEGIN
4     empsal := empsal + addless;
5 END;
6 BEGIN
7     SELECT salary INTO salary_of_emp
8     FROM emp1
9     WHERE emp_id = 122;
10    DBMS_OUTPUT.PUT_LINE
11    ('Before invoking procedure, salary_of_emp: ' || salary_of_emp);
12    approx_salary (100, salary_of_emp, 1000);
13    DBMS_OUTPUT.PUT_LINE
14    ('After invoking procedure, salary_of_emp: ' || salary_of_emp);
15 END;
```

Results Explain Describe Saved SQL History

```
Before invoking procedure, salary_of_emp: 81265
After invoking procedure, salary_of_emp: 82265

Statement processed.
```

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

QUERY:

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

OUTPUT:

The screenshot shows a database query results window with the following details:

- Results Tab:** The active tab, indicating the output of the query.
- Navigation:** Explain, Describe, Saved SQL, History.
- Output Content:**

```
m = TRUE
m AND n = TRUE
----- FOR m TRUE AND n NULL -----
m = TRUE
n = NULL
m AND n = NULL
----- FOR m FALSE AND n NULL -----
m = FALSE
n = NULL
m AND n = FALSE
----- FOR m NULL AND n TRUE -----
m = NULL
n = TRUE
m AND n = NULL
----- FOR m NULL AND n FALSE -----
m = NULL
n = FALSE
m AND n = FALSE

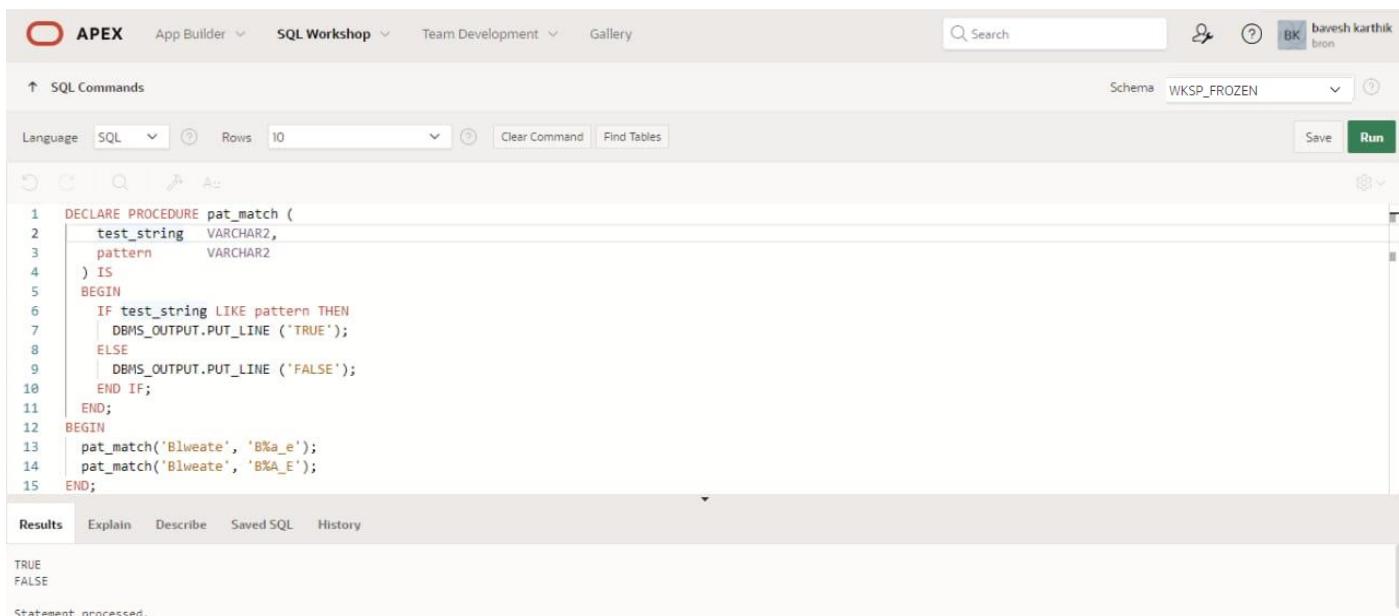
Statement processed.
```
- Timing:** 0.01 seconds.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown shows WKSP_FROZEN. The main area displays the PL/SQL code for the pat_match procedure. The code uses the LIKE operator with wildcards (%) to match strings. The output section at the bottom shows the results of executing the code: TRUE and FALSE, indicating the matches found for the two patterns provided.

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

Results Explain Describe Saved SQL History

TRUE
FALSE

Statement processed.

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

QUERY:

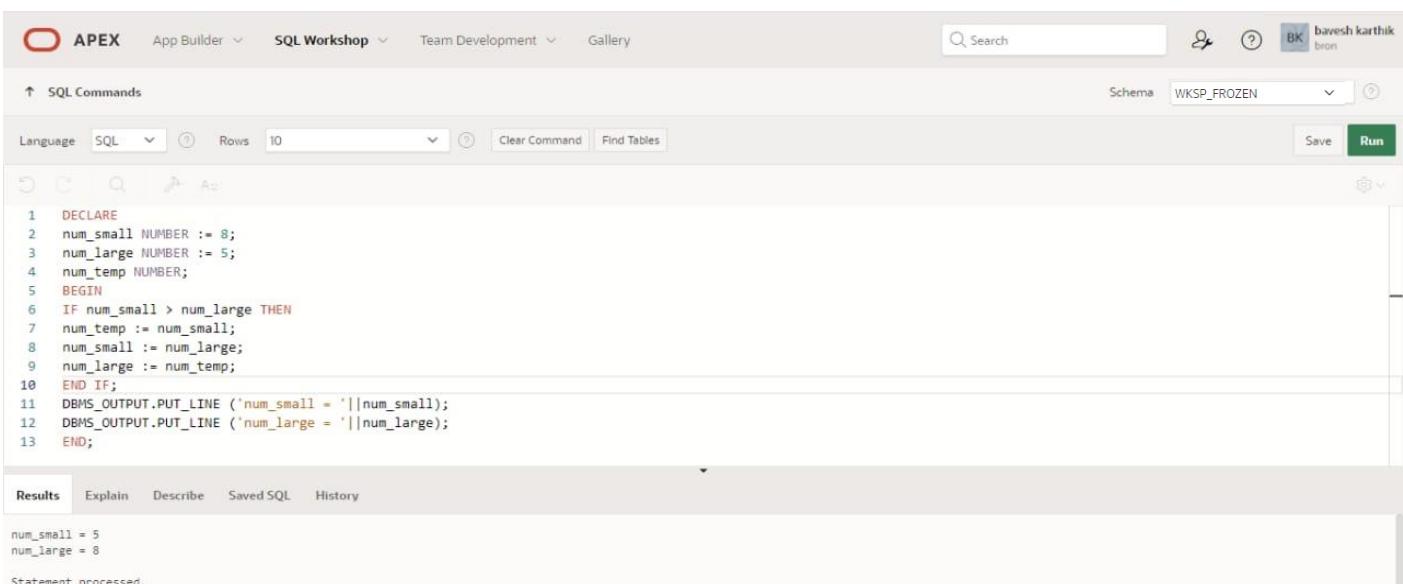
DECLARE

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

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

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'bavesh karthik' with a 'bron' role. The main area is titled 'SQL Commands'. The code editor contains the provided PL/SQL block. The results tab at the bottom shows the output: 'num_small = 5' and 'num_large = 8', followed by a message 'Statement processed.'

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

Results Explain Describe Saved SQL History

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

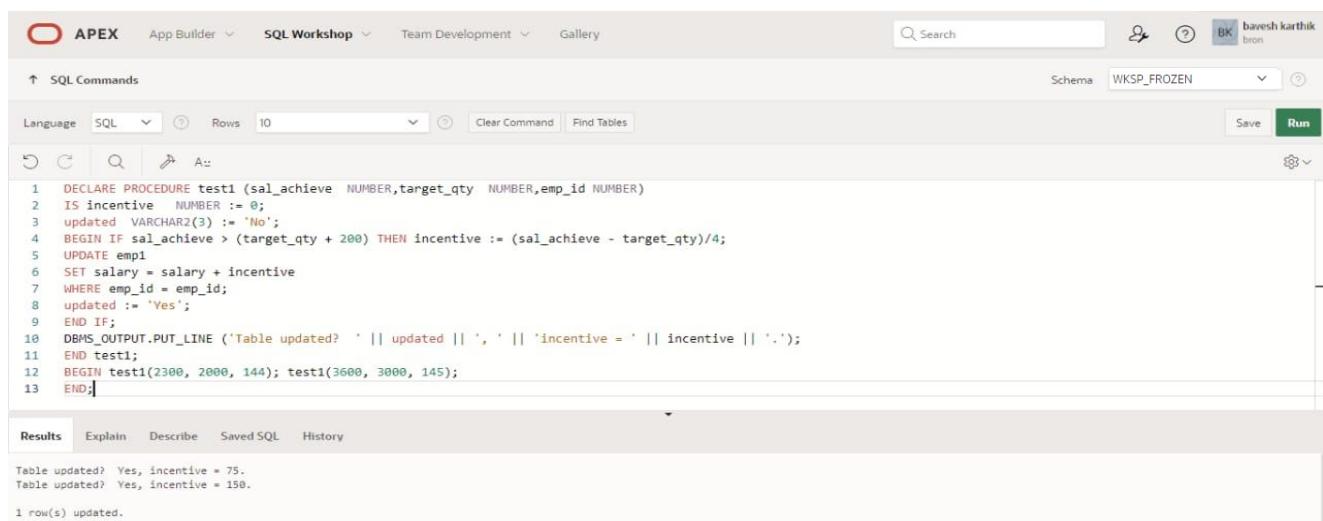
7.) Write a PL/SQL procedure to calculate the incentive on a target achieved and display the message either the record updated or not.

QUERY:

DECLARE

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The main area displays the PL/SQL code for the test1 procedure. Below the code, the Results tab is active, showing the output of the executed code. The output indicates that two rows were updated, with incentives of 75 and 150 respectively, and both records were updated successfully.

```
1  DECLARE PROCEDURE test1 (sal_achieve NUMBER,target_qty NUMBER,emp_id NUMBER)
2  IS incentive NUMBER := 0;
3  updated VARCHAR2(3) := 'No';
4  BEGIN IF sal_achieve > (target_qty + 200) THEN incentive := (sal_achieve - target_qty)/4;
5  UPDATE emp1
6  SET salary = salary + incentive
7  WHERE emp_id = emp_id;
8  updated := 'Yes';
9  END IF;
10 DBMS_OUTPUT.PUT_LINE ('Table updated? ' || updated || ',' ||
11 'incentive = ' || incentive || '.');
12 END test1;
13 BEGIN test1(2300, 2000, 144); test1(3600, 3000, 145);
14 END;
```

Results

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

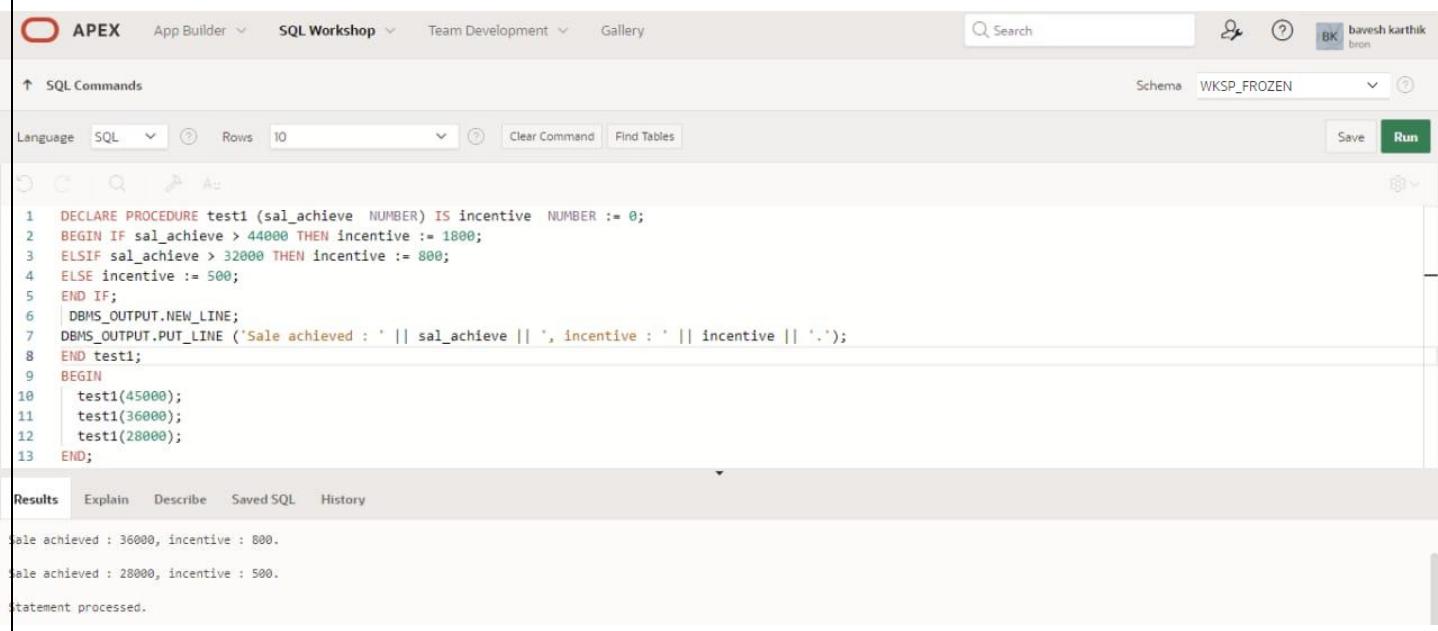
1 row(s) updated.
```

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'bavesh karthik' with a 'WKSP_FROZEN' schema selected. The main area is titled 'SQL Commands' and contains the PL/SQL code from the previous block. Below the code, the 'Results' tab is active, displaying the output of the executed code. The output shows three rows of results: 'Sale achieved : 36000, incentive : 800.', 'Sale achieved : 28000, incentive : 500.', and 'statement processed.'

```
1  DECLARE PROCEDURE test1 (sal_achieve NUMBER) IS incentive NUMBER := 0;
2  BEGIN IF sal_achieve > 44000 THEN incentive := 1800;
3  ELSIF sal_achieve > 32000 THEN incentive := 800;
4  ELSE incentive := 500;
5  END IF;
6  |DBMS_OUTPUT.NEW_LINE;
7  DBMS_OUTPUT.PUT_LINE ('Sale achieved : ' || sal_achieve || ', incentive : ' || incentive || '.');
8  END test1;
9  BEGIN
10    test1(45000);
11    test1(36000);
12    test1(28000);
13  END;
```

Results Explain Describe Saved SQL History

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

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

QUERY:

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
    tot_emp NUMBER;  
    get_dep_id NUMBER;
```

```
BEGIN
```

```
    get_dep_id := 80;
```

```
    SELECT Count(*)
```

```
    INTO tot_emp
```

```
    FROM employees e
```

```
        join departments d
```

```
            ON e.department_id = d.department_id
```

```
    WHERE e.department_id = get_dep_id;
```

```
    dbms_output.Put_line ('The employees are in the department'||get_dep_id||' is: '
```

```
        ||To_char(tot_emp));
```

```
    IF tot_emp >= 45 THEN
```

```
        dbms_output.Put_line ('There are no vacancies in the department'||get_dep_id);
```

```
    ELSE
```

```
        dbms_output.Put_line ('There are'||to_char(45-tot_emp)||' vacancies in department'||get_dep_id
```

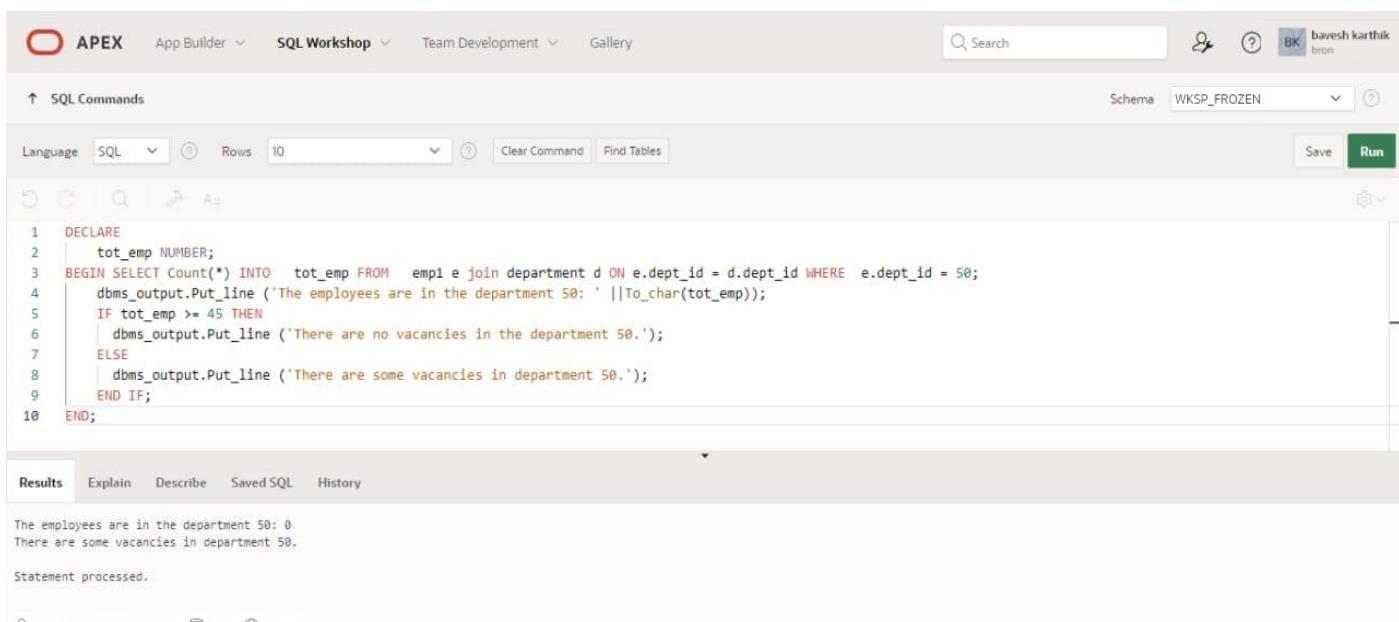
```
);
```

```
    END IF;
```

```
END;
```

```
/
```

OUTPUT:



```
APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_FROZEN Run  
SQL Commands Language SQL Rows 10 Clear Command Find Tables Save Run  
1 DECLARE  
2     tot_emp NUMBER;  
3 BEGIN SELECT Count(*) INTO tot_emp FROM emp1 e join department d ON e.dept_id = d.dept_id WHERE e.dept_id = 50;  
4 dbms_output.Put_line ('The employees are in the department 50: '||To_char(tot_emp));  
5 IF tot_emp >= 45 THEN  
6     dbms_output.Put_line ('There are no vacancies in the department 50.');
```

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

220701039@rajalakshmi.edu.in bron en

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:

```
APPEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_FROZEN Run  
↑ SQL Commands Language SQL Rows 10 Clear Command Find Tables Save Run  
11 WHERE e.department_id = get_dep_id;  
12 dbms_output.Put_line ('The employees are in the department'||get_dep_id||' is: '  
13 ||To_char(tot_emp));  
14 IF tot_emp >= 45 THEN  
15 dbms_output.Put_line ('There are no vacancies in the department'||get_dep_id);  
16 ELSE  
17 dbms_output.Put_line ('There are'||to_char(45-tot_emp)||' vacancies in department'||  
18 get_dep_id );  
19 END IF;  
20 END;  
21 /
```

The employees are in the department 80 is: 6
There are 39 vacancies in department 80
Statement processed.
0.01 seconds

Copyright © 1999-2023, Oracle and/or its affiliates. Oracle APFX 25.2.4

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

QUERY:

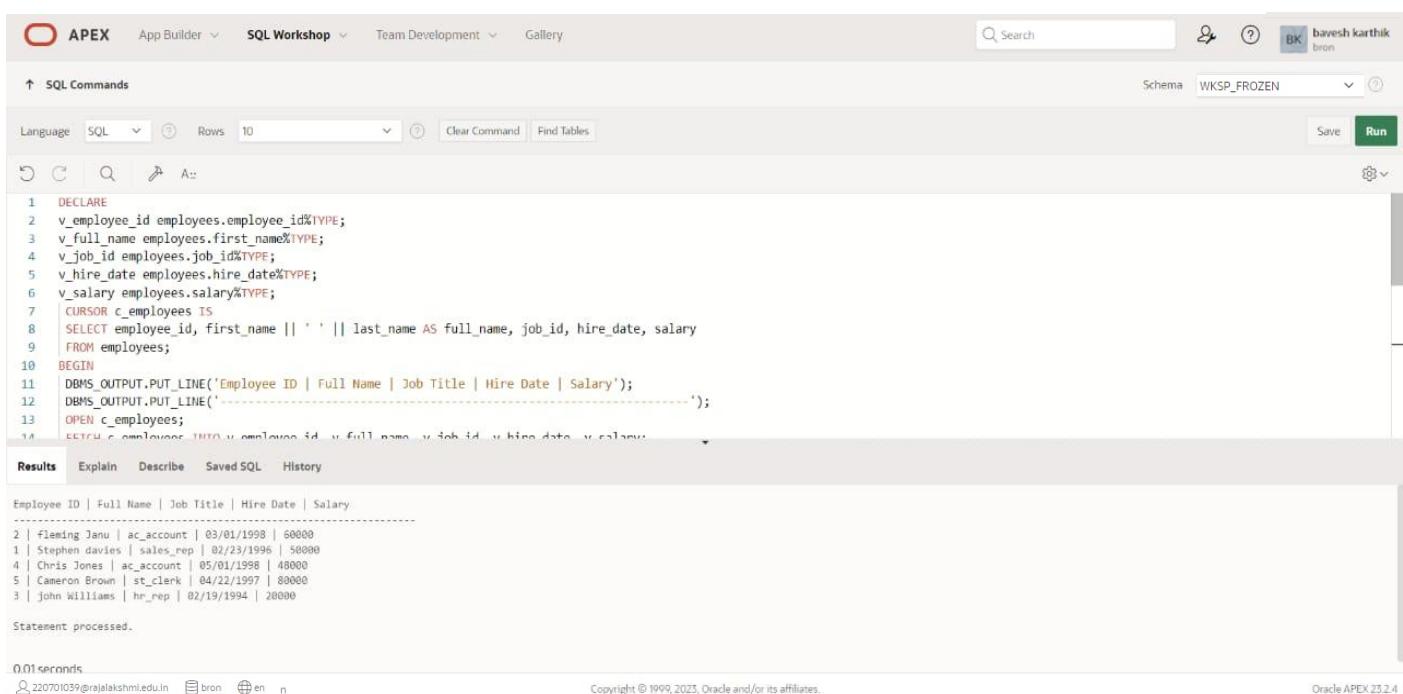
DECLARE

```
v_employee_id employees.employee_id%TYPE;
v_full_name employees.first_name%TYPE;
v_job_id employees.job_id%TYPE;
v_hire_date employees.hire_date%TYPE;
v_salary employees.salary%TYPE;

CURSOR c_employees IS
  SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
  FROM employees;

BEGIN
  DBMS_OUTPUT.PUT_LINE('Employee ID | Full Name | Job Title | Hire Date | Salary');
  DBMS_OUTPUT.PUT_LINE('-----');
  OPEN c_employees;
  FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
  WHILE c_employees%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(v_employee_id || ' ' || v_full_name || ' ' || v_job_id || ' ' ||
v_hire_date || ' ' || v_salary);
    FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
  END LOOP;
  CLOSE c_employees;
END;
/
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a user profile for 'bavesh karthik bron'. The main area has tabs for 'SQL Commands' (selected), 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL command entered is the PL/SQL code provided above. The 'Results' tab displays the output:

Employee ID	Full Name	Job Title	Hire Date	Salary
2	Fleming Janu	ac_account	03/01/1998	60000
1	Stephen davies	sales_rep	02/23/1996	50000
4	Chris Jones	ac_account	05/01/1998	48000
5	Cameron Brown	st_clerk	04/22/1997	80000
3	John Williams	hr_rep	02/19/1994	20000

Below the results, it says 'Statement processed.' and '0.01 seconds.'

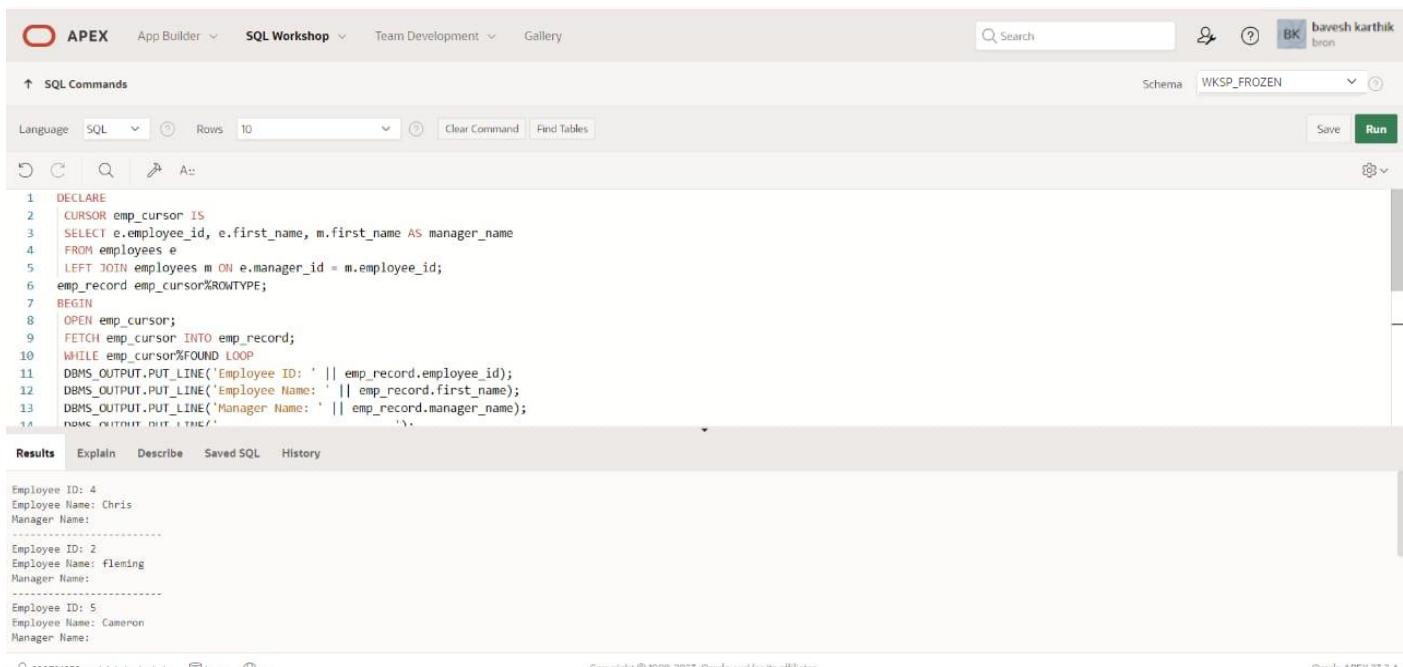
12.) Write a PL/SQL program to display the employee IDs, names, and department names of all employees.

QUERY:

DECLARE

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is signed in as 'bavesh karthik' with the session ID 'bron'. The schema is set to 'WKSP_FROZEN'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code from the previous section. Below the code, the 'Results' tab is selected, displaying the output of the program. The output shows three records from the employees table, each with its Employee ID, First Name, and Manager Name. The output is formatted with a header and a separator line between records.

Employee ID	Employee Name	Manager Name
4	Chris	
2	fleming	
5	Cameron	

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 APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'bavesh karthik bron'. The main workspace has tabs for SQL Commands, SQL, Rows (10), Clear Command, Find Tables, Save, and Run. Below this is a code editor window displaying the PL/SQL code from above. The code is numbered 1 to 13. The 'Run' button is highlighted in green. At the bottom, the 'Results' tab is selected, showing the output of the program. The output consists of four rows of data, each starting with 'Job ID:' followed by a job title and its minimum salary. The job titles listed are ac_account, sales_rep, st_clerk, and another ac_account entry.

Job ID	Title	Minimum Salary
ac_account	Account Executive	40000
sales_rep	Sales Representative	40000
ac_account	Account Executive	40000
st_clerk	Stock Clerk	40000

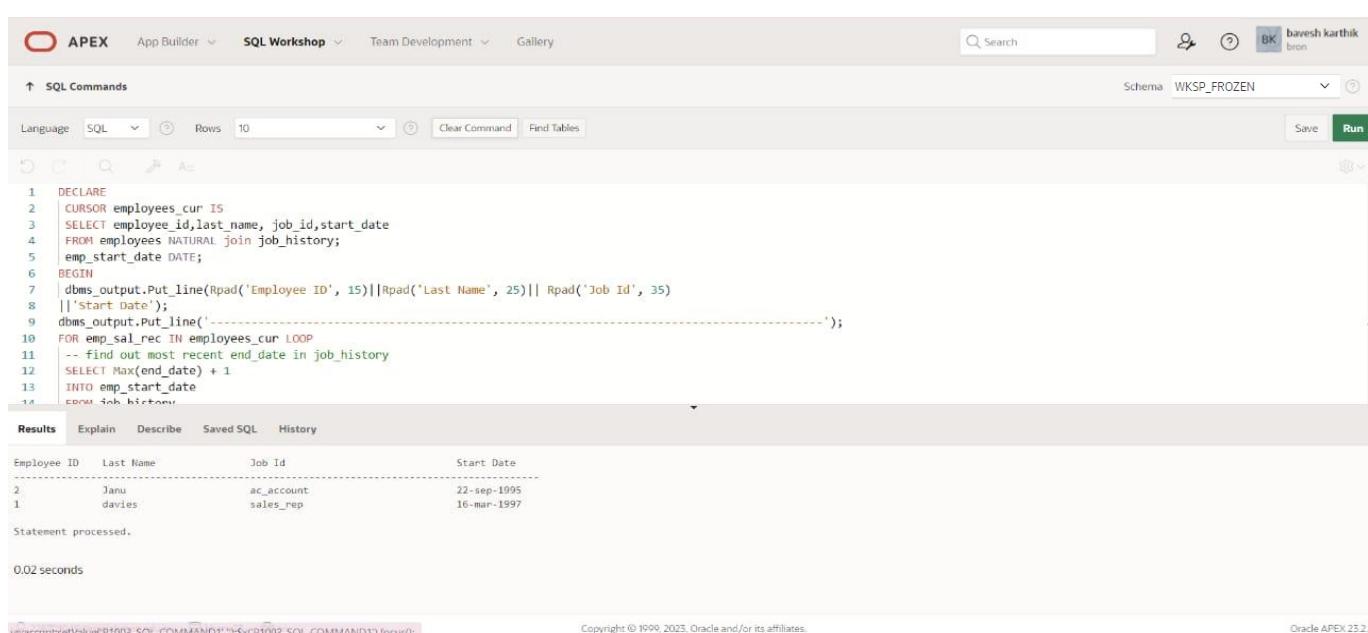
14.) Write a PL/SQL program to display the employee IDs, names, and job history start dates of all employees.

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:



Employee ID	Last Name	Job Id	Start Date
2	Janet	ac_account	22-sep-1995
1	Davies	sales_rep	16-mar-1997

Statement processed.
0.02 seconds

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

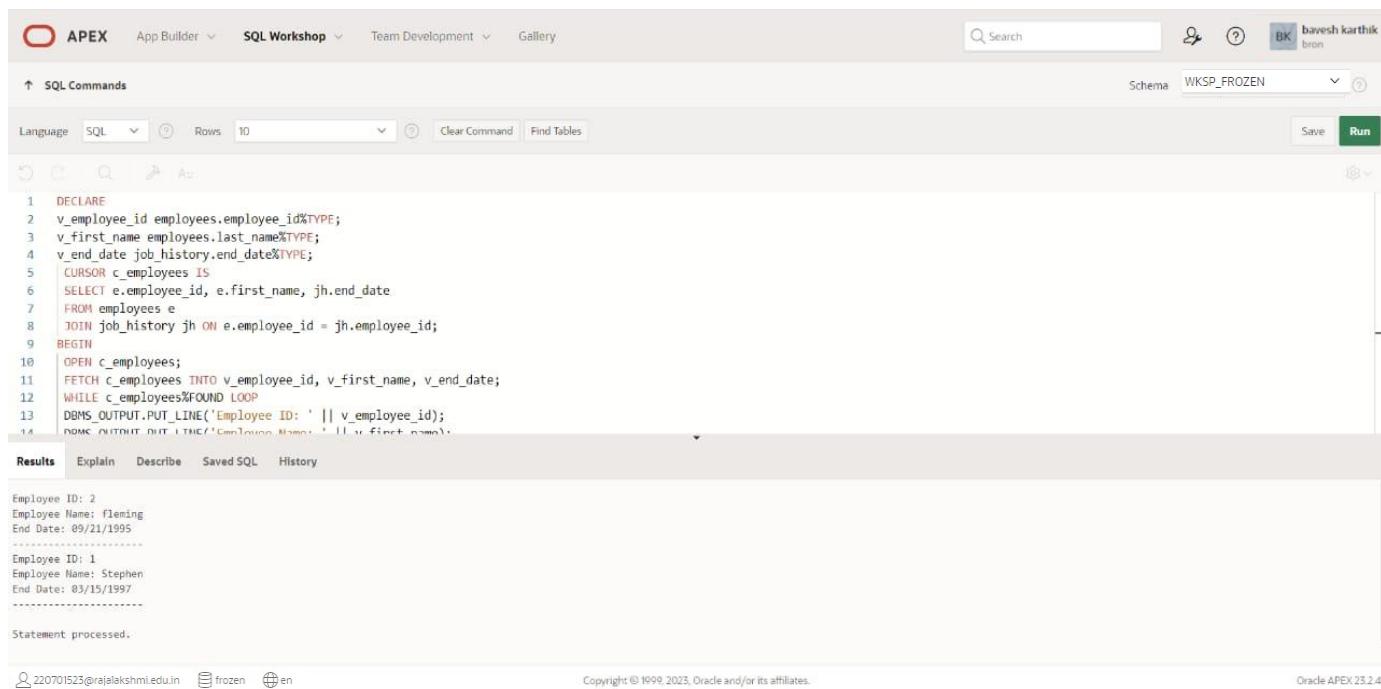
Oracle APEX 23.2.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is active. The right side of the header shows the user 'bavesh karthik' and the schema 'WKSP_FROZEN'. The main area is titled 'SQL Commands' with a 'Run' button. The code area contains the PL/SQL block from above. The results area displays the output of the program, which shows two rows of employee information separated by a horizontal line. The bottom status bar shows the user '220701523@rajalakshmi.edu.in', the schema 'frozen', and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' The version 'Oracle APEX 23.2.4' is also visible.

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

Employee ID	Employee Name	End Date
2	Fleming	09/21/1995
1	Stephen	03/15/1997

Statement processed.

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

RESULT:

PROCEDURES AND FUNCTIONS

EX_NO: 17

DATE:

1.)Factorial of a number using function.

QUERY:

DECLARE

fac NUMBER := 1;

n NUMBER := :1;

BEGIN

WHILE $n > 0$ LOOP

fac := n * fac;

$n := n - 1;$

END LOOP;

DBMS_OUTPUT.PUT_LINE(fac);

— 11 —

END.

OUTPUT:

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

QUERY:

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

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

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

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

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

OUTPUT:

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

```
1 CREATE OR REPLACE PROCEDURE get_book_info (
2     p_book_id IN NUMBER,
3     p_title OUT VARCHAR2,
4     p_author OUT VARCHAR2,
5     p_year_published OUT NUMBER
6 )
7 AS
8 BEGIN
9     SELECT title, author, year_published INTO p_title, p_author, p_year_published
10    FROM books
11   WHERE book_id = p_book_id;

```

The 'Results' tab is selected, showing the output of the command: "Procedure created." Below the results, it indicates "0.04 seconds". At the bottom left, there are user profile icons for '220701039@rajalakshmi.edu.in' and 'bron'. The bottom right corner shows the copyright notice: "Copyright © 1999, 2022, 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:

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
```

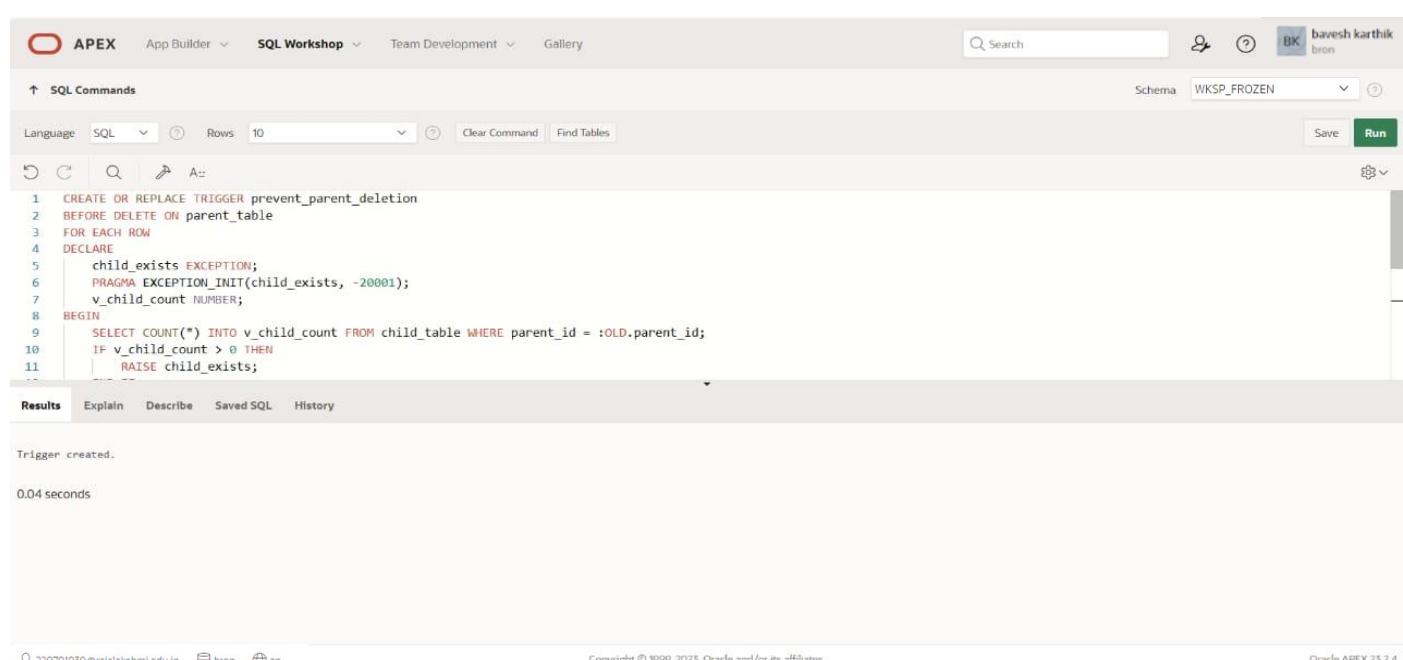
```
        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child
```



```
records exist.');
```

```
END;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The main workspace is titled "SQL Commands". The schema is set to "WKSP_FROZEN". The code area contains the PL/SQL trigger definition provided in the previous text. The "Results" tab at the bottom shows 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 END;
```

Trigger created.
0.04 seconds

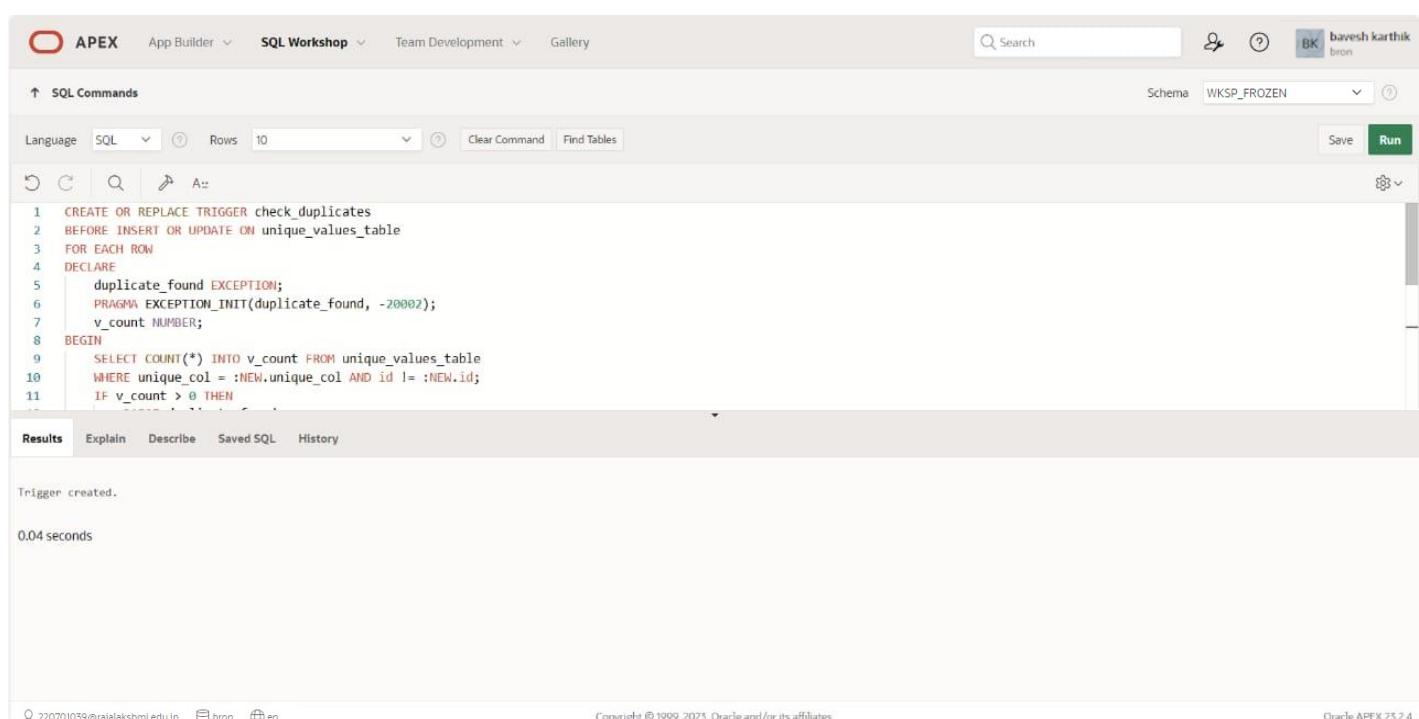
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 25.2.4

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is active. The main area is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is as follows:

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

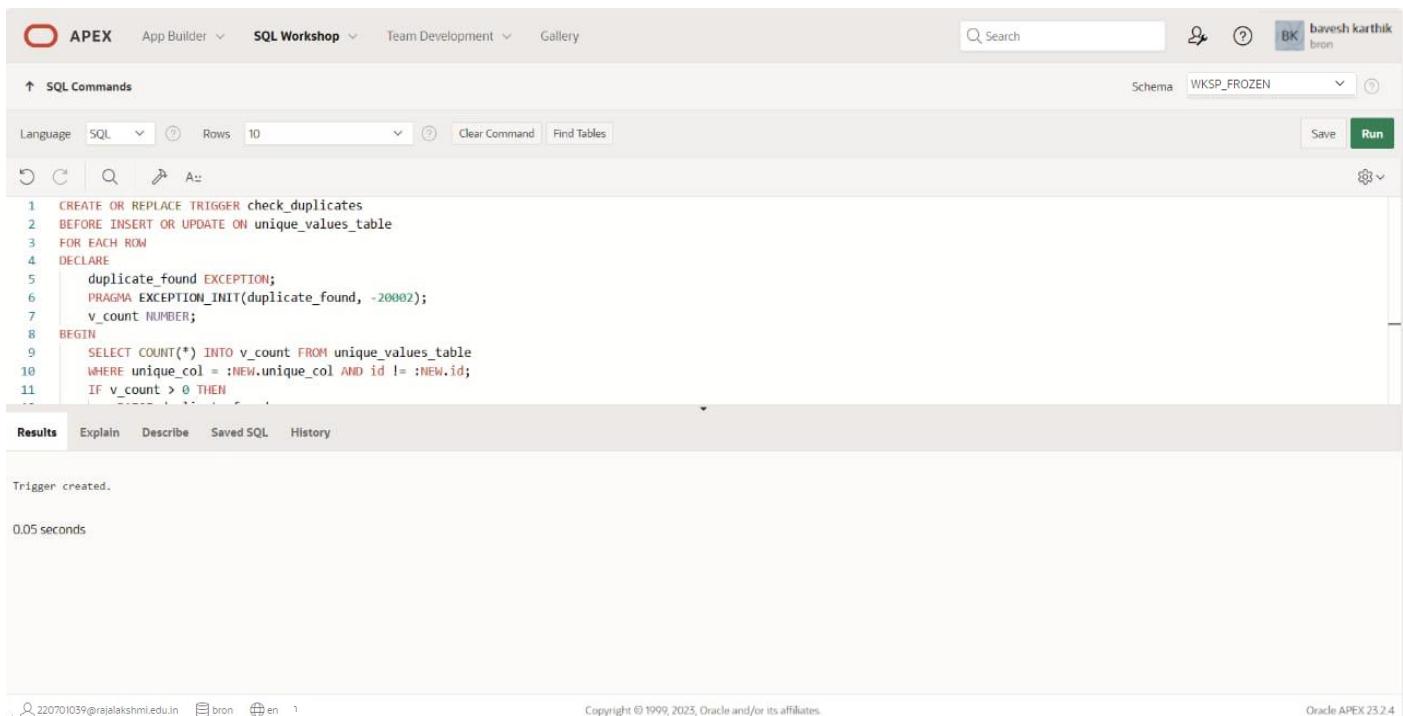
Below the code, the 'Results' tab is selected, showing the message 'Trigger created.' and a execution time of '0.04 seconds'. The bottom of the screen displays user information and copyright notices.

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 (selected), Team Development, and Gallery. The search bar contains the placeholder 'Search'. On the right, there are user profile icons for 'bavesh karthik' and 'bron'.

The main workspace is titled 'SQL Commands'. It features a toolbar with icons for Undo, Redo, Find, Replace, and Run. Below the toolbar, the language is set to 'SQL', rows are set to 10, and there are buttons for 'Clear Command' and 'Find Tables'. The 'Run' button is highlighted in green.

The SQL command area contains the PL/SQL code for creating the 'check_threshold' trigger. The code defines a trigger on the 'threshold_table' that fires before insert or update. It declares a local exception 'threshold_exceeded' and initializes it with code -20003. It also declares variables 'v_sum' and 'v_threshold' with a default value of 10000. The trigger body uses a SELECT statement to get the current sum of 'value_col' from the table, adds the new value, and then checks if the sum exceeds the threshold. If it does, it raises the declared exception. An exception block handles this raise by calling 'RAISE_APPLICATION_ERROR' with the same error code and a descriptive message.

The results tab shows the output of the command: 'Trigger created.' and a execution time of '0.05 seconds'. The bottom of the page includes copyright information for Oracle and the APEX 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     duplicate_found EXCEPTION;
6     PRAGMA EXCEPTION_INIT(duplicate_found, -20002);
7     v_count NUMBER;
8 BEGIN
9     SELECT COUNT(*) INTO v_count FROM unique_values_table
10    WHERE unique_col = :NEW.unique_col AND id != :NEW.id;
11    IF v_count > 0 THEN
12        RAISE duplicate_found;
13    END IF;
14 END;
```

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right side, there's a user profile for 'bavesh karthik' and a schema dropdown set to 'WKSP_FROZEN'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below the title are buttons for 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. On the far right are 'Save' and 'Run' buttons. The code editor contains the PL/SQL trigger creation script. The results tab shows the output: 'Trigger created.' and '0.05 seconds'. At the bottom, it shows the user information '220701039@rajalakshmi.edu.in' and 'bron', along with copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

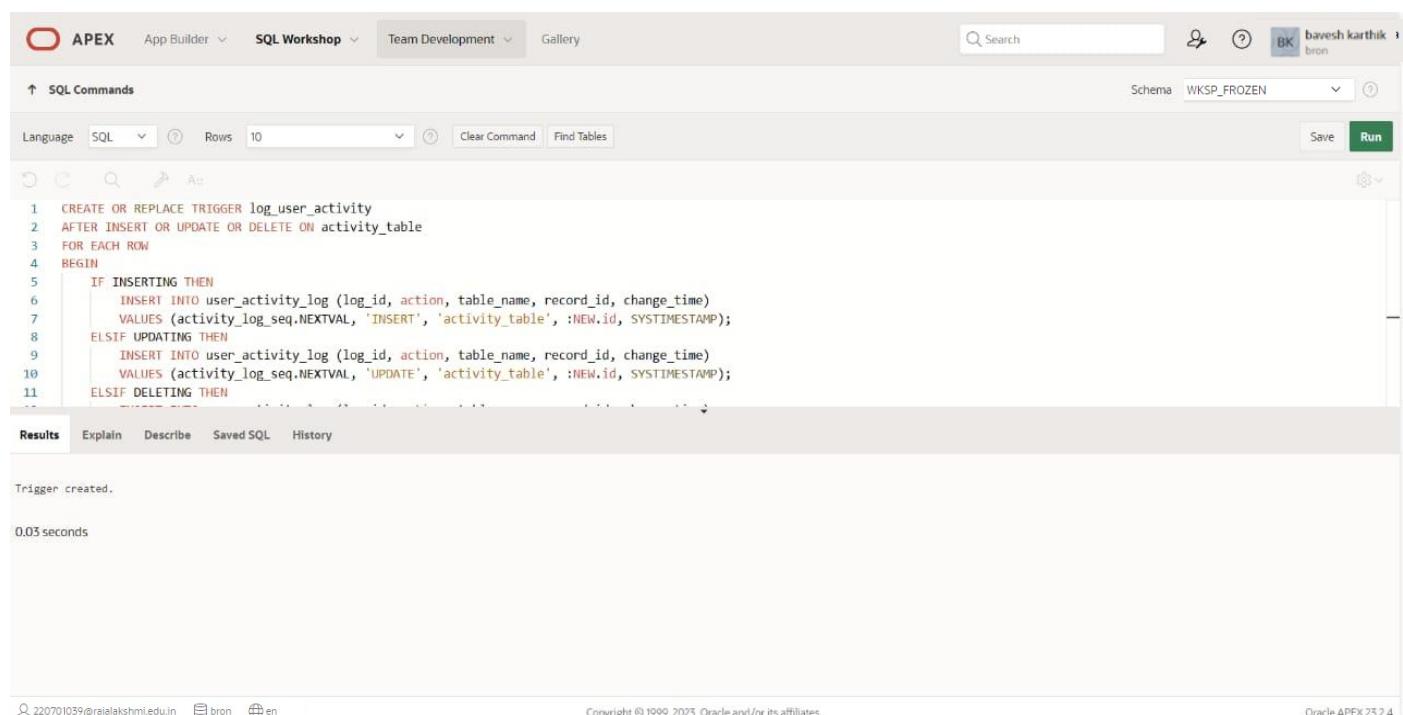
```
1 CREATE OR REPLACE TRIGGER log_changes
2 AFTER UPDATE ON main_table
3 FOR EACH ROW
4 BEGIN
5     INSERT INTO audit_table (audit_id, changed_id, old_col1, new_col1, old_col2, new_col2, change_time)
6     VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2, :NEW.col2, SYSTIMESTAMP);
7 END;
8
```

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 links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side of the header shows a user profile for 'bavesh.karthik' and a session ID 'bron'. The main workspace is titled 'SQL Commands' and contains a code editor with 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

```

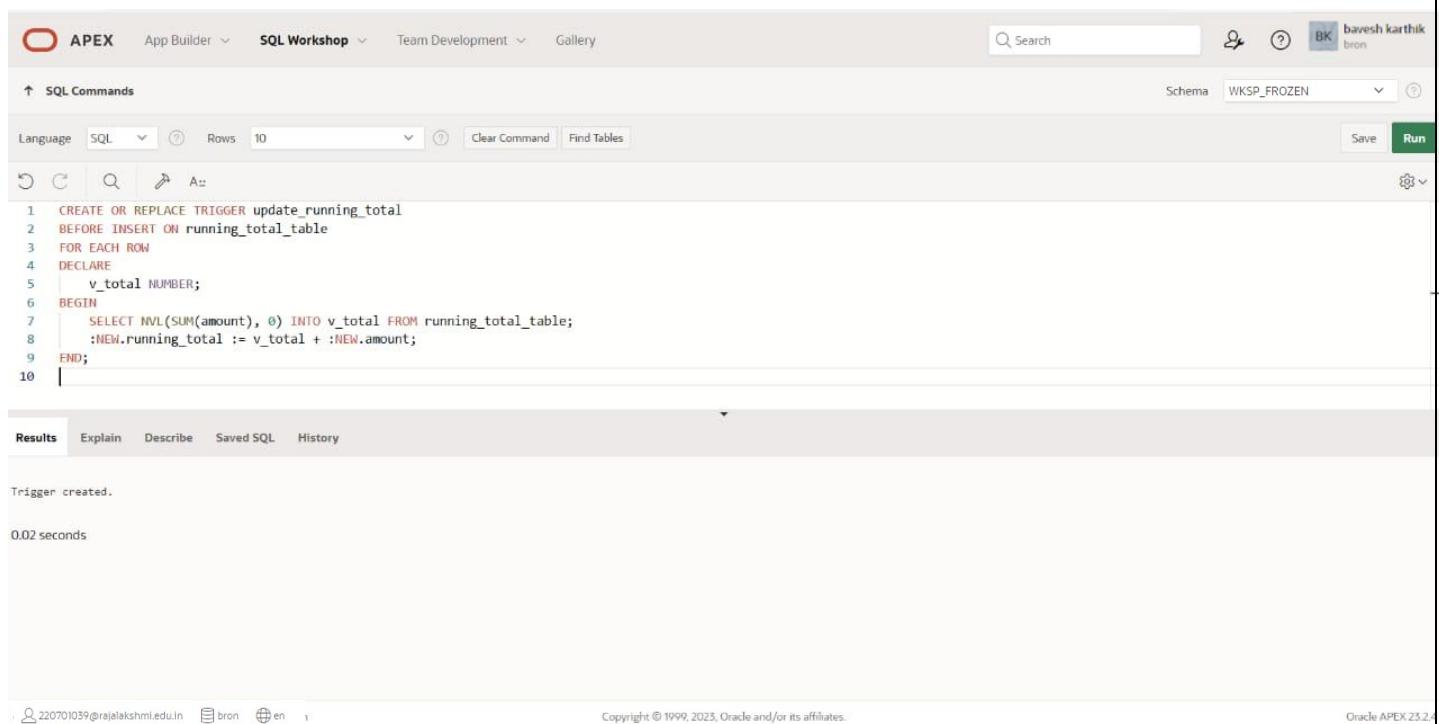
Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab displays the output: 'Trigger created.' and '0.03 seconds'. At the bottom of the page, there are footer links for user information ('220701039@rajalakshmi.edu.in', 'bron', 'en') and copyright information ('Copyright © 1999, 2023, Oracle and/or its affiliates.', 'Oracle APEX 23.2.4').

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 'APEX', 'App Builder', 'SQL Workshop' (which is selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon for 'bavesh karthik bron', and a 'Schema' dropdown set to 'WKSP_FROZEN'. Below the navigation is a toolbar with icons for Undo, Redo, Find, Replace, and Run. The main area is a code editor with the following SQL code:

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

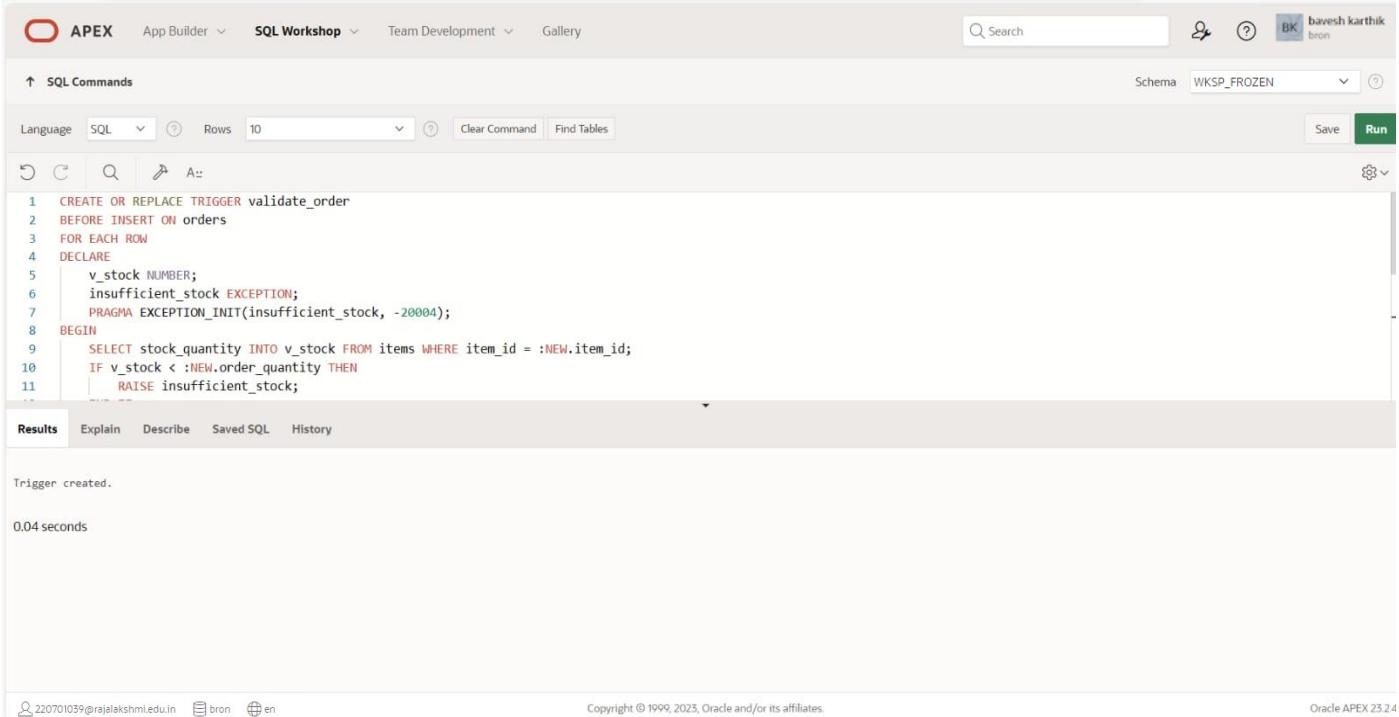
Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active and displays the output: 'Trigger created.' and '0.02 seconds'. At the bottom, the footer includes the URL '220701039@rajalakshmi.edu.in', the schema 'bron', and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.0'.

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 Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. The right side of the header shows the user 'bavesh karthik' and the schema 'WKSP_FROZEN'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the 'validate_order' trigger. The code is syntax-highlighted, with keywords in red and identifiers in blue. The bottom of the screen displays the 'Results' tab, which shows the message 'Trigger created.' and a execution time of '0.04 seconds'. The footer includes copyright information for Oracle and the version 'Oracle APEX 23.2.4'.

```
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;

```

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

RESULT:

MONGO DB

EX_NO: 19

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows a MongoDB query editor interface. On the left, there is a text input field labeled "Enter a title...". Below it are two dropdown menus: one for "MongoDB" and another with a question mark icon. To the right is a "Run" button with a play icon. The main area contains the MongoDB query:1: { \$or: [{ name: /^Wil/ }, { cuisine: { \$nin: ['American', 'Chinese'] } }] , { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 } }); Output

To the right of the query is a terminal window showing the command-line interface:

```
mycompiler_mongodb>
mycompiler_mongodb>
```

At the bottom of the terminal window, it says "[Execution complete with exit code 0]".

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:

myCompiler

English Recent

Enter a title...

MongoDB

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

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

Enter a title...

MongoDB

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

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. At the top, there is a search bar labeled "Enter a title..." and a dropdown menu set to "MongoDB". Below the search bar are two buttons: a blue one with a gear icon and a red one with a trash bin icon. To the right is a green "Run" button with a play icon. The main area is titled "Output" and contains the following text:

```
1{$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1}]] Output
```

```
mycompiler_mongodb>
mycompiler_mongodb>
```

[Execution complete with exit code 0]

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:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field labeled "Enter a title...". Below it are two buttons: one with a green icon and the text "MongoDB" and another with a blue icon. To the right of these buttons are two small circular icons. At the top right, there is a "Ctrl+" key combination and a green play button icon. In the main area, a command is entered in a light blue input field:

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

To the right of the input field is a "Output" section with a dark grey header. The output text is displayed in white:

```
mycompiler_mongodb>
mycompiler_mongodb>
```

At the bottom of the output section, a message indicates the execution status:

```
[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field labeled "Enter a title...". Below it are two buttons: one with a green icon and the text "MongoDB" and another with a blue icon. To the right of these buttons are two small circular icons. At the top right, there is a "Ctrl+" key combination and a green play button icon. In the main area, a command is entered in a light blue input field:

```
1{{$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1}}| Output
```

To the right of the input field is a "Output" section with a dark grey header. The output text is displayed in white:

```
mycompiler_mongodb>
mycompiler_mongodb>
```

At the bottom of the output section, a message indicates the execution status:

```
[Execution complete with exit code 0]
```

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:

Enter a title...

MongoDB Run

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

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

Enter a title...

MongoDB Run

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

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled 'Enter a title...' and a dropdown menu set to 'MongoDB'. Below the search bar is a toolbar with a green play button icon and a red 'Run' button. The main area contains a code editor with the following content:

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

To the right of the code editor is a panel titled 'Output' containing the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

Enter a title...

MongoDB Run

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

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

Enter a title...

MongoDB Run

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

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text 'Enter a title...'. Below it are two buttons: 'MongoDB' with a dropdown arrow and a small info icon. To the right is a green 'Run' button with a play icon. The main area contains a code editor with the following content:

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

Below the code editor, the number '2' is followed by '3'. To the right, under the heading 'Output', the terminal session shows:

```
mycompiler_mongodb>
mycompiler_mongodb>
```

At the bottom of the output window, the message '[Execution complete with exit code 0]' is displayed.

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:

Enter a title...

MongoDB ▾



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

2

3

Output

```
mycompiler_mongodb>
mycompiler_mongodb>
```

[Execution complete with exit code 0]

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:

Enter a title...

MongoDB ▾



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

2

3

Output

```
mycompiler_mongodb>
mycompiler_mongodb>
```

[Execution complete with exit code 0]

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:

The screenshot shows a MongoDB shell interface. At the top, there is a search bar labeled "Enter a title...". Below it, a dropdown menu says "MongoDB" with a "▼" icon, and a small info icon. On the right, there is a green "Run" button with a play icon. The main area contains the following text:

```
1.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] }) Output
2
3
```

On the right side of the interface, there is a terminal window with the following output:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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:

```
Enter a title...
MongoDB ▾ ⓘ Run
1{ "score": { $lt: 5 } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" }}] Output
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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:

```
Enter a title...
MongoDB ▾ ⓘ Run
1{ $lt: 5 } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] })] Output
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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:



The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text 'Enter a title...'. Below it are two buttons: 'MongoDB' with a dropdown arrow and a small info icon. To the right is a green 'Run' button with a play icon. The main area contains a code editor with the following content:

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

Next to the code editor is a 'Output' tab. The output window shows the command prompt 'mycompiler_mongodb>' followed by '[Execution complete with exit code 0]'. There is also a small note at the bottom right of the output window: 'mycompiler_mongodb> mycompiler_mongodb> [Execution complete with exit code 0]'.

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:

Enter a title...

MongoDB ▾



Run

```
1 $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" }]} Output
```

```
mycompiler_mongodb>
```

```
mycompiler_mongodb>
```

```
[Execution complete with exit code 0]
```

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:

Enter a title...

MongoDB ▾



Run

```
1.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] }]} Output
```

```
mycompiler_mongodb>
```

```
mycompiler_mongodb>
```

```
[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB query editor interface. At the top, there is a search bar labeled "Enter a title...". Below it are two dropdown menus: one for "MongoDB" and another with a question mark icon. To the right are two buttons: a green "Run" button and a blue "Copy" button. The main area contains the MongoDB query and its output. The query is:
1 'A", "grades.score": 6 }], \$or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { \$ne: "American" } })The output shows the command prompt:
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

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:

The screenshot shows a MongoDB query editor interface. At the top, there is a search bar labeled "Enter a title...". Below it are two dropdown menus: one for "MongoDB" and another with a question mark icon. To the right are two buttons: a green "Run" button and a blue "Copy" button. The main area contains the MongoDB query and its output. The query is:
1 "core": 6 }], \$or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { \$nin: ["American", "Chinese"] } })The output shows the command prompt:
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

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:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text 'Enter a title...'. Below it are two buttons: one with a green icon labeled 'MongoDB' and another with a blue icon. To the right are two buttons: a teal 'Run' button and a blue 'Save' button. The main area contains a code editor with the following content:

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

To the right of the code editor is a light gray panel titled 'Output' containing the results of the query execution:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. On the left, there is an input field labeled "Enter a title...". Below it, a dropdown menu is set to "MongoDB" and a help icon is visible. On the right, a "Run" button is shown with a play icon. The code input area contains the command: `1 db.movies.find({ year: 1893 })`. To the right, under the heading "Output", the response is displayed as follows:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. On the left, there is an input field labeled "Enter a title...". Below it, a dropdown menu is set to "MongoDB" and a help icon is visible. On the right, a "Run" button is shown with a play icon. The code input area contains the command: `1 db.movies.find({ runtime: { $gt: 120 } })`. To the right, under the heading "Output", the response is displayed as follows:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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

QUERY:

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

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon and the text "Ctrl+Enter". The main area contains a code block with the command `1 db.movies.find({ genres: 'short' })`. To the right, under the heading "Output", the results are displayed: `mycompiler_mongodb>`, `mycompiler_mongodb>`, and the message "[Execution complete with exit code 0]".

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:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon and the text "Ctrl+Enter". The main area contains a code block with the command `1 db.movies.find({ directors: 'William K.L. Dickson' })`. To the right, under the heading "Output", the results are displayed: `mycompiler_mongodb>`, `mycompiler_mongodb>`, and the message "[Execution complete with exit code 0]".

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:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: one with a green leaf icon labeled "MongoDB" and another with an info icon. To the right is a "Run" button with a play icon. The main area contains the command: `1 db.movies.find({ countries: 'USA' })`. On the far right, under the heading "Output", the response is shown: `mycompiler_mongodb>`, `mycompiler_mongodb>`, and [Execution complete with exit code 0].

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

QUERY:

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

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: one with a green leaf icon labeled "MongoDB" and another with an info icon. To the right is a "Run" button with a play icon. The main area contains the command: `1 db.movies.find({ rated: 'UNRATED' })`. On the far right, under the heading "Output", the response is shown: `mycompiler_mongodb>`, `mycompiler_mongodb>`, and [Execution complete with exit code 0].

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:

Enter a title...

MongoDB ▾

Ctrl+Enter

Run

```
1 db.movies.find({ 'imdb.votes': { $gt: 1000 } })
```

Output

```
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:

Enter a title...

MongoDB ▾

Run

```
1 db.movies.find({ 'imdb.rating': { $gt: 7 } })
```

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon. In the main area, a command is entered: `db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })`. The output window on the right shows the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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

QUERY:

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

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon. In the main area, a command is entered: `db.movies.find({ 'awards.wins': { $gt: 0 } })`. The output window on the right shows the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

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:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled "Enter a title..." and a dropdown menu set to "MongoDB". Below the search bar are two buttons: a green one with a gear icon and a blue one with a question mark icon. To the right of these buttons is a teal "Run" button. The main area contains a command line and its output. The command is:

```
1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })
```

The output shows the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

At the bottom right of the interface, there is a small advertisement for Figma.

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:

MongoDB?Run

```
1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })|| Output
```

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:

MongoDB?Run

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

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

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:



The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a "Run" button with a play icon. The main area contains a code block with the query: "1 db.movies.find({ title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })". To the right of the code is a "Output" section. The output shows the command "mycompiler_mongodb>" followed by "mycompiler_mongodb>". At the bottom of the output section, the message "[Execution complete with exit code 0]" is displayed.

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

RESULT: