

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. 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 'Ashwin V ashwin' with the ID '220701518'. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_220701518'. Below the title are buttons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run. The SQL editor contains the following code:

```
1 create table Dep (Dept_ID Number(6)Not Null,Dept_Name Varchar(20), Manager_ID VARCHAR(6), Location_ID Number(4));
```

The 'Results' tab is selected at the bottom, showing the message 'Table created.'.

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop, Team Development, and Gallery. The user is signed in as 'Ashwin V ashwin' with ID '220701518'. The main workspace is titled 'SQL Commands' and shows the following SQL code:

```
1 create table Dep (Dept_ID Number(6)Not Null,Dept_Name Varchar(20), Manager_ID VARCHAR(6), Location_ID Number(4));
2 
```

The 'Run' button is highlighted in green at the bottom right of the command input area. Below the workspace, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying the message 'Table created.'.

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

QUERY:

```
Alter table emp modify(Last_Name varchar2(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 profile for 'Ashwin V ashwin 220701518', and a toolbar with icons for Save and Run.

The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 alter table employees modify(Last_Name varchar(50));
```

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

Table altered.

At the bottom, the footer displays copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates, 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 columns. Name the columns Id, First_name, Last_name, salary and Dept_id respectively.

QUERY:

Create table employees2(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. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile for Ashwin V ashwin (220701518), and a Run button. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 create table employees2(id number(6) not null,first_name varchar(20),last_name varchar(25)not null,salary number(8,2),dept_id number(6)not null);
```

Below the code, the 'Results' tab is selected, showing the message "Table created." The bottom footer includes copyright information for Oracle and the APEX version (23.2.4).

5.Drop the EMP table.

QUERY:

Drop table emp;

OUTPUT:

The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. In the top navigation bar, 'APEX' is highlighted, along with 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, the user 'Ashwin V ashwin' is logged in, with the session ID '220701518'. The main area is titled 'SQL Commands' and contains a single line of code: '1 drop table employees;'. Below the code, the results tab is active, showing the message 'Table dropped.' at the top. At the bottom of the results pane, there are links for 'Explain', 'Describe', 'Saved SQL', and 'History'. The footer of the page includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

```
1 drop table employees;
```

Results Explain Describe Saved SQL History

Table dropped.

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

6.Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename employees2 to emp;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is active. The main area is titled 'SQL Commands'. A single line of SQL code is entered: '1 rename employees2 to emp;'. The 'Run' button is visible at the top right. In the bottom results pane, the message 'Statement processed.' is displayed, along with a timestamp '0.05 seconds' and copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' The interface includes various tabs like 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'.

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. 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 help icon, and a user profile for 'Ashwin V ashwin' (ID: 220701518). The main workspace is titled 'SQL Commands'. It features a toolbar with icons for Undo, Redo, Search, 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 command area contains the following SQL code:

```
1 comment on table dept is 'Department info';
2 comment on table emp is Employee info';
3
4
```

Below the command area, there is a results section with tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. A message at the bottom of the results section says: "Enter SQL statement or PL/SQL command and click Run to see the results." At the very bottom of the page, there are footer links for 220701518@rajalakshmi.edu.in, 220701518, and en, along with copyright information: "Copyright © 1999, 2023, Oracle and/or its affiliates." and "Oracle APEX 23.2.4".

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, Team Development, and Gallery. On the right, there is a search bar, a user profile for 'Ashwin V ashwin' (ID 220701518), and a schema dropdown set to 'WKSP_220701518'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 alter table emp2 drop column firstname
```

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

```
Table altered.
```

Execution time is listed as 0.05 seconds. The bottom of the page displays 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:

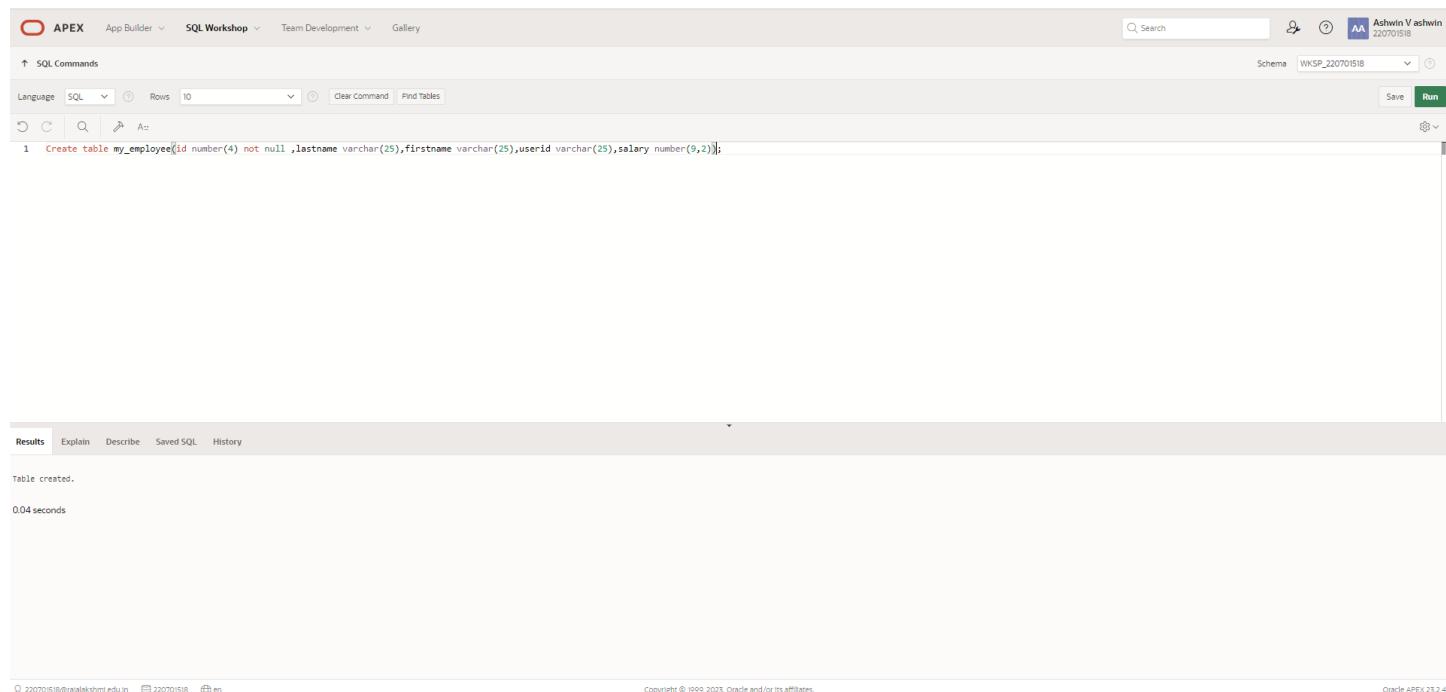
1.Create MY_EMPLOYEE table with the following structure

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 my_employee(id number(4) not null ,lastname varchar(25),firstname varchar(25),userid varchar(25),salary number(9,2));
```

OUTPUT:



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

```
1 Create table my_employee(id number(4) not null ,lastname varchar(25),firstname varchar(25),userid varchar(25),salary number(9,2));
```

In the Results tab, the output shows:

```
Table created.  
0.04 seconds
```

At the bottom of the page, there are footer links: 220701518@rajalakshmi.edu.in, 220701518, en, Copyright © 1994-2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, tabs for 'App Builder', 'SQL Workshop' (which is active), 'Team Development', and 'Gallery' are visible. On the right side, the user 'Ashwin Vashwin' and the schema 'WKSP_220701518' are displayed. The main area is titled 'SQL Commands'. The language is set to 'SQL'. The command entered is:

```
1 Insert into my_employee  
2 values(1,'Patel','Ralph','rpatel',895)  
3
```

Below the command, the results section shows:

Results Explain Describe Saved SQL History

1 row(s) inserted.
0.04 seconds

At the bottom, footer information includes:

220701518@rajalakshmi.edu.in 220701518 en Copyright © 1999-2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

3.Display the table with values.

QUERY:

```
select * from my_employee
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as Adhwin Vashwin (220701518). The SQL Commands tab is selected, showing the query: `select * from my_employee`. The results section displays a table with the following data:

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

Below the table, it says "5 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.

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

QUERY:

```
Insert into my_employee values (4,'csk','rcb','mi',860);
```

OUTPUT:

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

```
1 Insert into my_employee values (6,'csk','rcb','mi',860);
2
```

Below the command, the results section displays:

1 row(s) inserted.
0.00 seconds

At the bottom of the page, there are footer links: '22070151@rajalakshmi.edu.in', '22070151', 'en', 'Copyright © 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

5. Make the data additions permanent.

QUERY:

Select * from my_employee

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there is a user profile for 'Ashwin V ashwin' and a schema dropdown set to 'WKSP_220701518'. The main area has a search bar and a toolbar with icons for Undo, Redo, Find, Copy, Paste, and Run. Below the toolbar, the SQL command 'Select * from my_employee' is entered. The results section shows a table with the following data:

ID	LASTNAME	FIRSTNAME	USERID	SALARY
2	Danics	Betty	bdanics	860
5	Ropebur	Audrey	aropibur	1550
6	csk	rcb	mi	860
1	Patel	Ralph	rpatel	895
4	Newman	Chad	Cnewman	750
3	Birki	Ben	bbirki	1100

Below the table, it says '6 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom, there are footer links for email, session ID, and help, along with copyright information and the version 'Oracle APEX 23.2.4'.

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

QUERY:

```
UPDATE my_employee SET lastname ='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, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_22070518'. The main area displays the SQL command:

```
1 UPDATE my_employee SET lastname ='drexler' where ID=3;
2
```

Below the command, the results section shows:

Results Explain Describe Saved SQL History

1 row(s) updated.

0.01 seconds

At the bottom of the page, there are footer links for '22070518@rataleshmi.edu.in', '22070518', and 'en'. The copyright notice reads 'Copyright © 1999-2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

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

QUERY:

UPDATE my_employee SET salary=1000 where salary<900;

OUTPUT:

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

```
1 UPDATE my_employee SET salary=1000 where salary<900;
2 |
```

Below the command, the results section displays the output:

4 row(s) updated.
0.01 seconds

At the bottom of the page, there are footer links: '22070168@nraalakshmi.edu.in', '22070168', 'en', 'Copyright © 1999-2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

8.Delete Betty dancs from MY _EMPLOYEE table.

QUERY:

Delete from my_employee where firstname='Betty' and last name = Dancs;

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_220701S1B'. The main area displays the following SQL command:

```
1 Delete from my_employee where lastname = 'Dancs' and firstname = 'Betty';
2
3
```

Below the command, the results section shows the output:

0 row(s) deleted.
0.02 seconds

At the bottom of the page, there are footer links for user information (220701S1B@rajalakshmi.edu.in), session ID (220701S1B), and language (en). The copyright notice reads "Copyright © 1999-2023, Oracle and/or its affiliates." and the page version is "Oracle APEX 23.2.4".

9.Empty the fourth row of the emp table.

QUERY:

**Delete from employee
Where empid = 4;**

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 Delete from my_employee
2 Where ID = 4;
3
```

Below the command, the results section shows:

1 row(s) deleted.
0.001seconds

At the bottom of the page, there is footer information: '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:

INCLUDING CONSTRAINTS

EX_NO:3

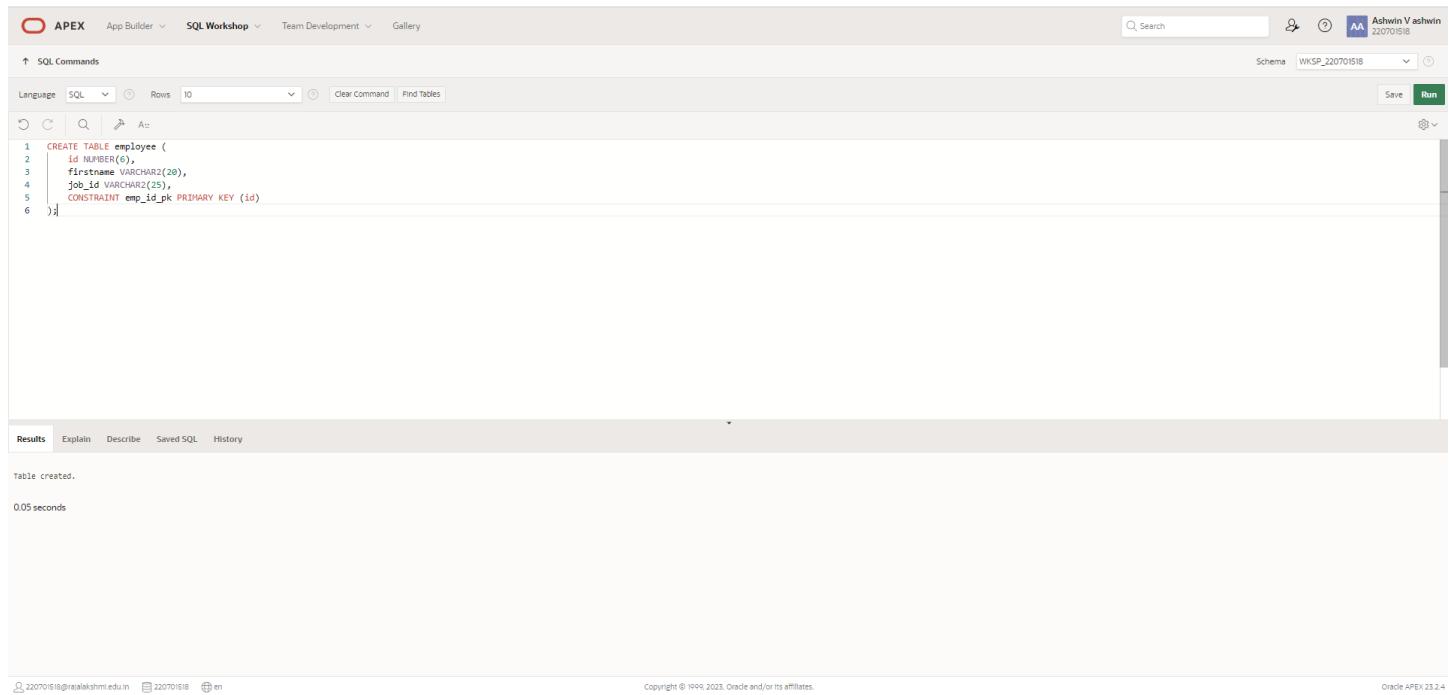
DATE:

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 employee ( id NUMBER(6), firstname VARCHAR2(20), job_id VARCHAR2(25),
CONSTRAINT emp_id_pk PRIMARY KEY (id)
);
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, a CREATE TABLE statement is entered:

```
1 CREATE TABLE employee (
2   id NUMBER(6),
3   firstname VARCHAR2(20),
4   job_id VARCHAR2(25),
5   CONSTRAINT emp_id_pk PRIMARY KEY (id)
6 );
```

The statement is highlighted with syntax coloring. Below the command, the results show:

Table created.
0.05 seconds

At the bottom, the footer includes the URL <https://220701518@rajalakshmi.edu.in>, session information, and copyright notice: Copyright © 1999-2023, Oracle and/or its affiliates.

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

QUERY:

```
CREATE TABLE DEPT ( id NUMBER(6), firstname VARCHAR2(20), job_id VARCHAR2(25),
CONSTRAINT dept_id_pk PRIMARY KEY (id)
);
```

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 code:

```
1 CREATE TABLE DEPT (
2   | id NUMBER(6),
3   | firstname VARCHAR2(20),
4   | job_id VARCHAR2(25),
5   | CONSTRAINT dept_id_pk PRIMARY KEY (id)
6 );
```

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

```
Table created.  
0.05 seconds
```

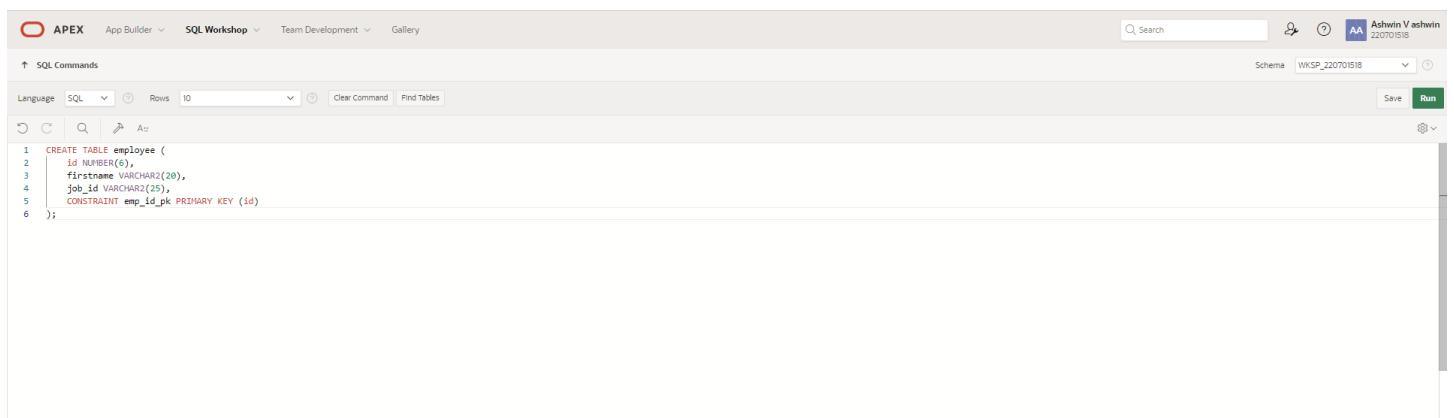
At the bottom of the page, there are footer links: 'Copyright © 1999-2023, Oracle and/or its affiliates.', 'Oracle APEX 23.2.4', and language selection 'en'.

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

QUERY:

Alter table emp add constraint my_emp_dept_id_fk foreign key(deptid) references emp(employeeid);

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is active. The main area is titled 'SQL Commands'. A dropdown menu 'Language' is set to 'SQL'. The 'Rows' dropdown is set to '10'. There are buttons for 'Clear Command' and 'Find Tables'. On the right side, there's a 'Search' bar, a user profile for 'Ashwin Vashwin 220701518', and buttons for 'Save' and 'Run'. The code area contains the following SQL command:

```
1 CREATE TABLE employee (
2     id NUMBER(6),
3     firstname VARCHAR2(20),
4     job_id VARCHAR2(25),
5     CONSTRAINT emp_id_pk PRIMARY KEY (id)
6 );
```

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.

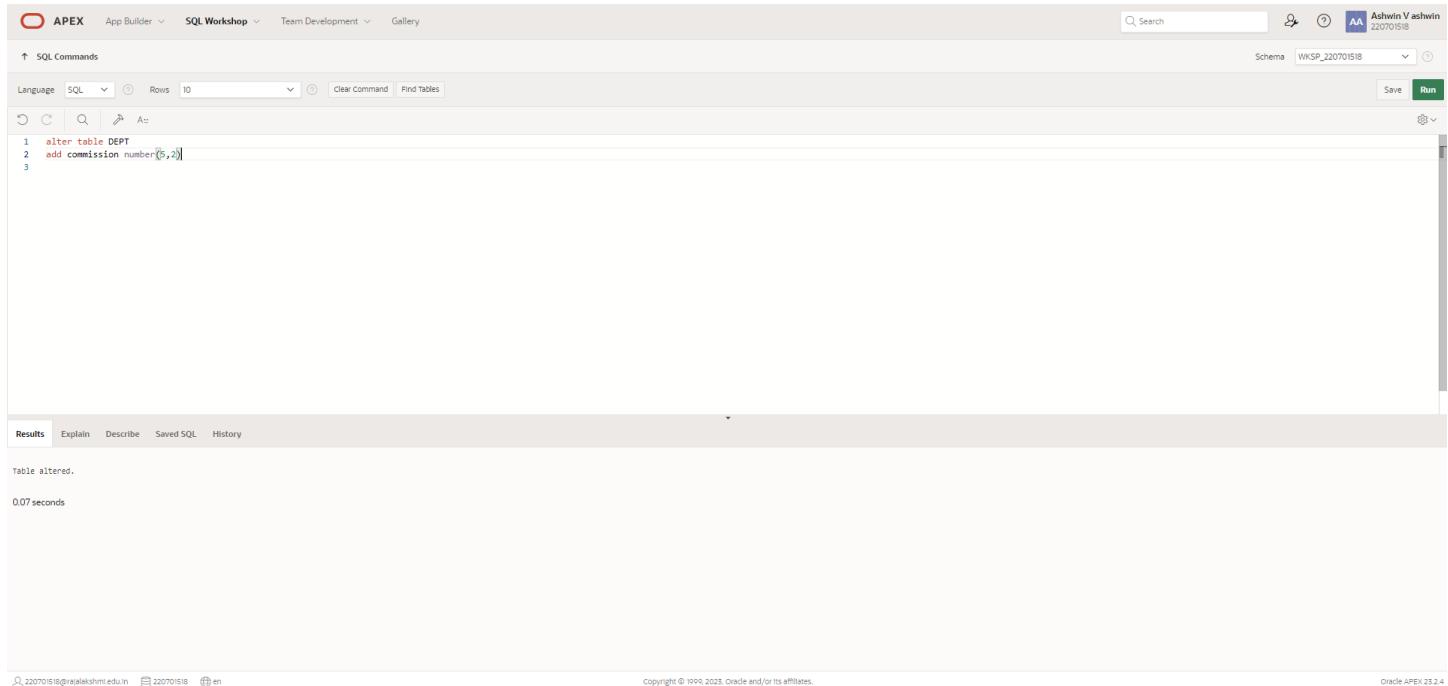
QUERY:

alter table DEPT

add commission number(5,2)

select * from DEPT where commissionn > 0;

OUTPUT:



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

```
1 alter table DEPT
2 add commission number(5,2)
3
```

The 'Run' button is highlighted in green at the top right of the SQL entry area. Below the command area, the Results tab is selected, showing the output:

Table altered.
0.07 seconds

At the bottom of the page, there are footer links: 22070158@rajalakshmi.edu.in, 22070158, en, 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:

Writing Basic SQL SELECT Statements

EX_NO:4

DATE:

- 1.The following statement executes successfully.

Identify the Errors

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

QUERY:

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

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

QUERY:

```
Describe 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 SQL Workshop tab is selected. The schema dropdown shows 'WKSP_220701518'. The main area displays the following SQL command:

```
1 Describe Dept;  
2
```

Below the command, the results are shown under the 'Describe' tab. The results table has columns: Object Type, TABLE, Object, DEPT. It lists the columns of the DEPT table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT	ID	NUMBER	-	6	0	1	-	-	-
	FIRSTNAME	VARCHAR2	20	-	-	-	✓	-	-
	JOB_ID	VARCHAR2	25	-	-	-	✓	-	-
	COMMISSION	NUMBER	-	5	2	-	✓	-	-

At the bottom of the page, there are footer links: 220701518@rajalakshmi.edu.in, 220701518, en, Copyright © 1999-2023, Oracle and/or its affiliates., and Oracle APEX 23.2.4.

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 employeeid,lastname,jobid,hiredate 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 Ashwin V ashwin (220701518). The SQL Commands tab is selected, showing the following SQL code:

```
1 Select id,lastname,job_code,hire_date from my_employee;
2
3
```

The Results tab displays the output of the query:

ID	LASTNAME	JOB_CODE	HIRE_DATE
5	Ropebur	cse	01/02/0024
6	csk	ece	02/02/0024
1	Patel	mech	03/02/0024
3	drexler	eee	04/02/0024

Below the table, it says "4 rows returned in 0.01 seconds" and there is a "Download" link. The bottom of the page includes standard footer links and copyright information.

4. Provide an alias STARTDATE for the hire date.

QUERY:

Select hiredate as StartDate 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 right side of the header shows the user's name, "Ashwin V ashwin 220701518", and a "Run" button. The main workspace is titled "SQL Commands". The SQL editor contains the following query:

```
1 Select hire_date as StartDate from my_employee;
```

Below the editor, the "Results" tab is selected. The output shows a single column named "STARTDATE" with four rows of data:

STARTDATE
01/02/0024
02/02/0024
03/02/0024
04/02/0024

At the bottom of the results pane, it says "4 rows returned in 0.01 seconds" and has a "Download" link. The footer of the page includes copyright information: "Copyright © 1999-2023, Oracle and/or its affiliates." and "Oracle APEX 23.2.4".

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

QUERY:

Select Unique jobid 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 right side of the header shows the user's name, "Ashwin V ashwin 220701518", and a "Run" button. The main workspace is titled "SQL Commands" and contains the following SQL code:

```
1 Select Unique job_code from my_employee;
2
```

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

JOB_CODE
cse
eee
mech
ece

At the bottom left, it says "4 rows returned in 0.00 seconds". At the bottom right, there are links for "Download", "Copyright © 1999-2023, Oracle and/or its affiliates.", and "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 lastname || ',' || job_code as Title from employee;

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_220701518'. The main area contains the following SQL code:

```
1 Select lastname || ',' || job_code as Title from MY_employee;
2
```

The results section displays the output of the query:

TITLE
Ropebur,cse
csk,ce
Patel,mech
drexler,eee

Below the results, it says "4 rows returned in 0.00 seconds".

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 employeeid ||','|| lastname ||','|| jobid ||','|| email ||','|| salary ||','|| hiredate 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 a SQL command window with the following content:

```
1 Select id ||','|| lastname ||','|| job_code ||','|| email ||','|| salary ||','|| hire_date as "the_output" from
2 my_employees;
```

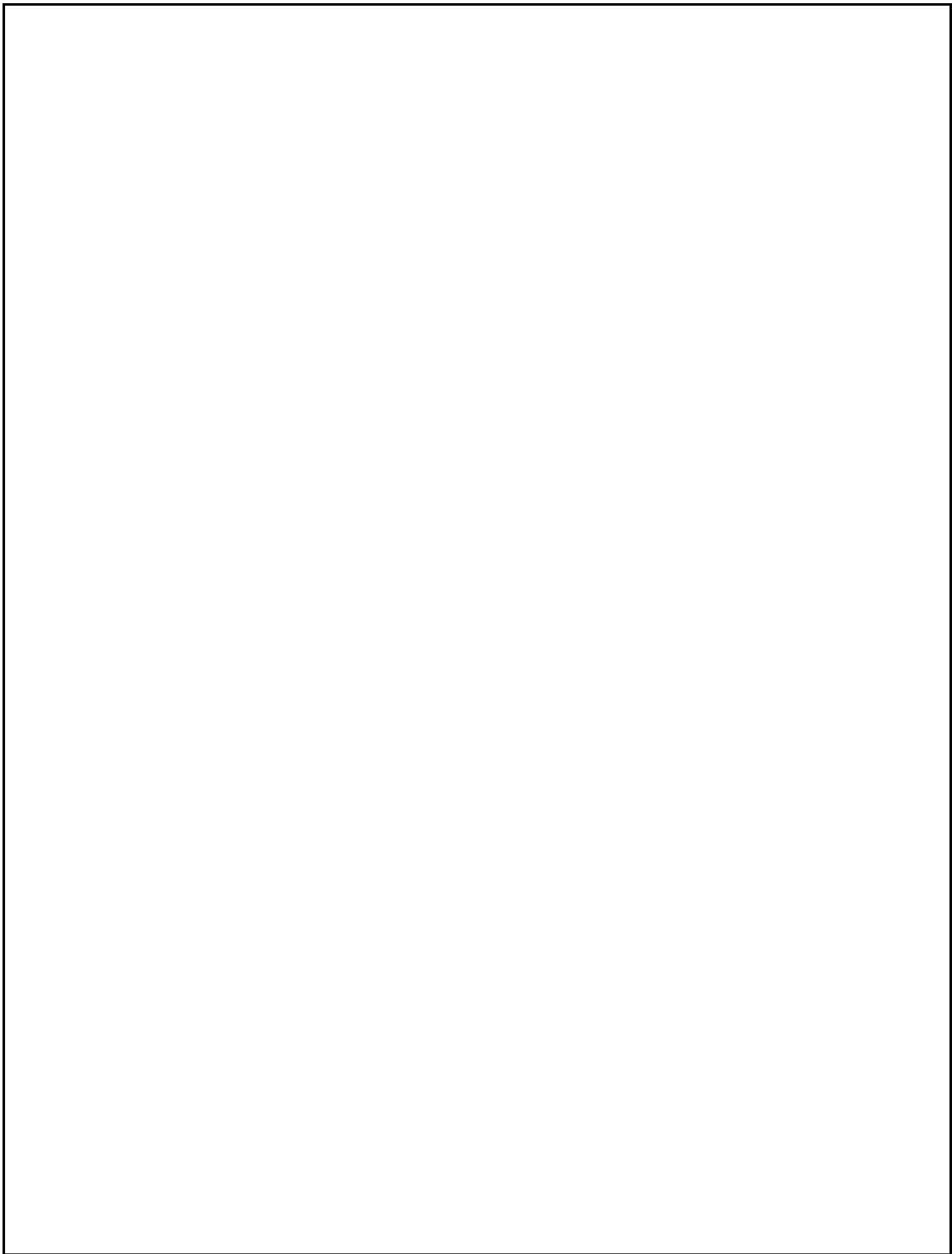
Below the command window, the results are displayed in a table:

the_output
5.Ropeburuse.cse@gmail.com,1550,01/02/0024
6.cskece.ece@gmail.com,1000,02/0024
1.Patel.mech.mech@gmail.com,1000,03/02/0024
3.drexler_eee.eee@gmail.com,1000,04/02/0024

At the bottom of the results pane, it says "4 rows returned in 0.01 seconds".

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

RESULT:



RESTRICTING AND SORTING DATA

EX_NO:5

DATE:

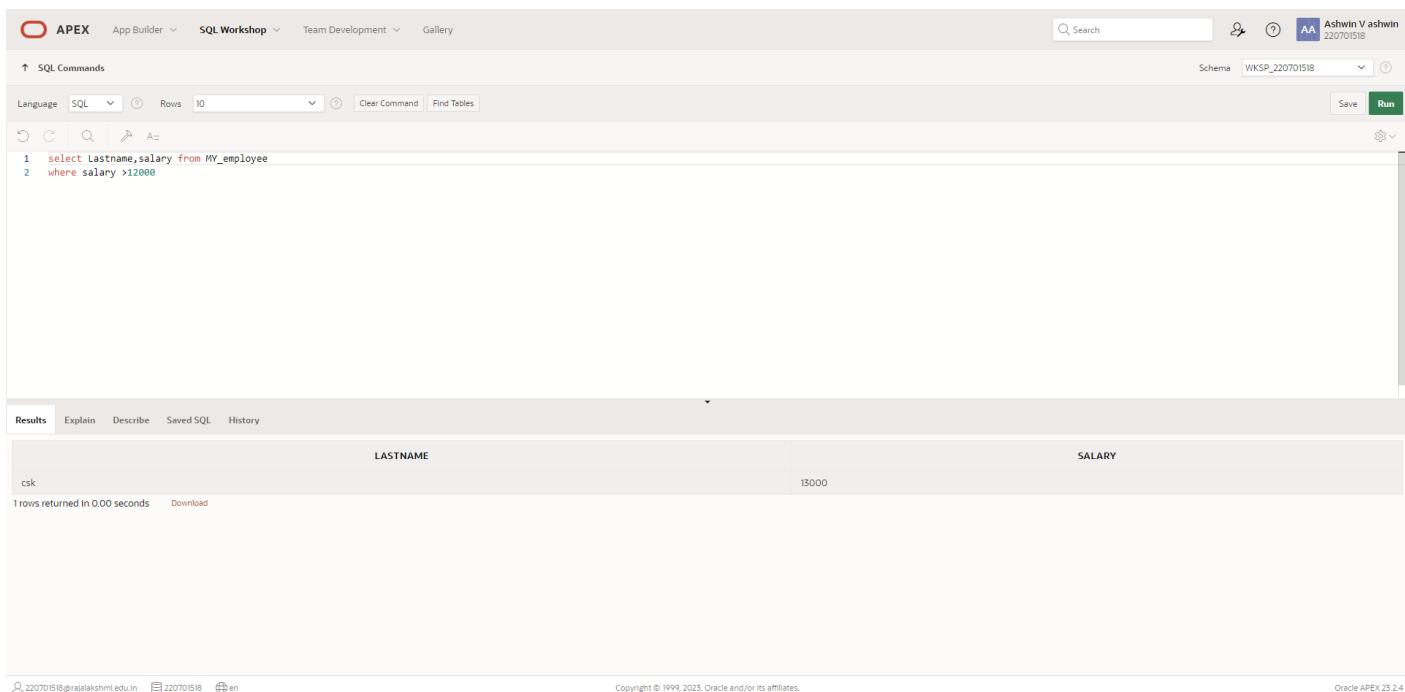
- 1.Create a query to display the last name and salary of employees earning more than 12000.

QUERY:

Select last_name,salary from MY_employee

Where salary > 12000

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 'Ashwin V ashwin' and a schema dropdown set to 'WKSP_220701518'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select Lastname,salary from MY_employee
2 where salary >12000
```

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

LASTNAME	SALARY
csk	13000

At the bottom of the results pane, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. The footer of the page includes copyright information for Oracle and the APEX version.

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

QUERY:

Select last_name, department_number from MY_employee

Where department_number = 176

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and session information (Schema: WKSP_220701518, User: Ashwin V ashwin 220701518). The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select lastname,DEPARTMENT_NUMBER from My_employee
2 where EMPLOYEE_NO =176
```

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

LASTNAME	DEPARTMENT_NUMBER
drexler	25

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

At the bottom, the footer includes links for 220701518@rajalakshmi.edu.in, 220701518, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

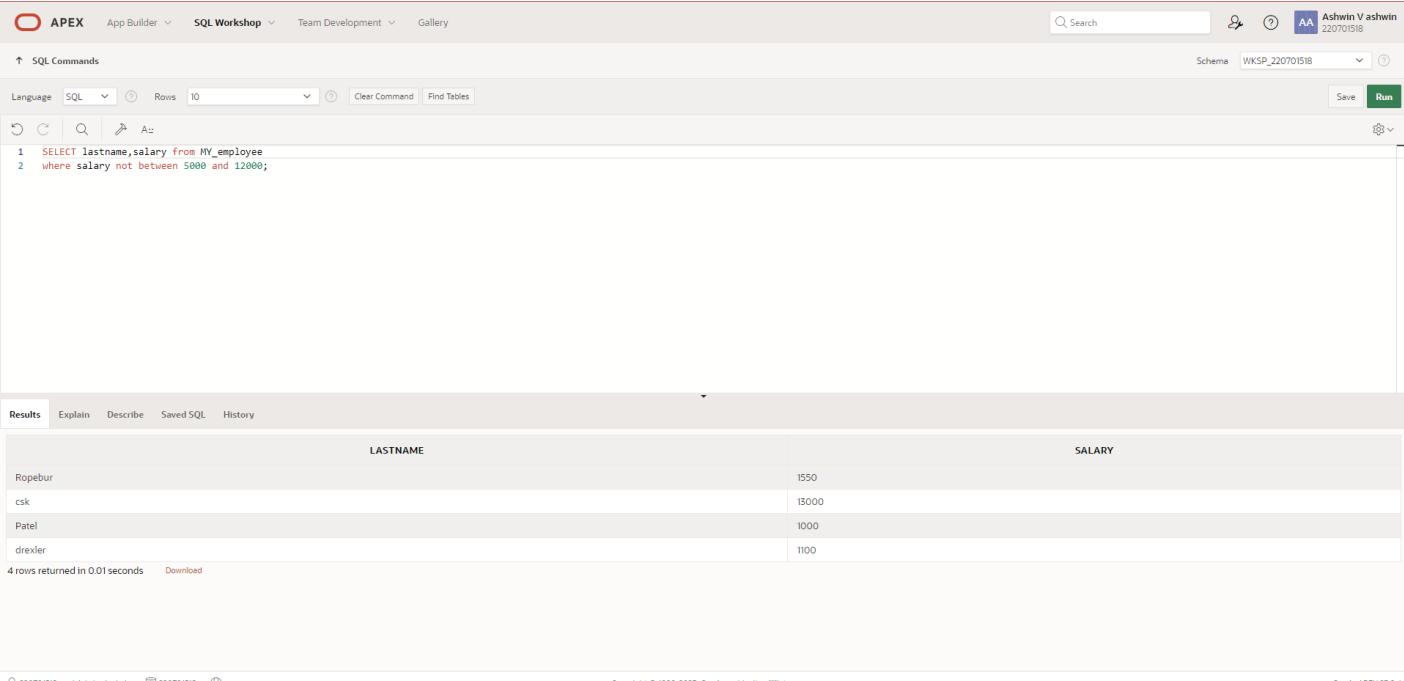
3. Create a query to display the last name and salary of employees whose salary is not in the range of 5000 and 12000. (hints: not between)

QUERY:

Select last_name, salary from MY_employee

Where salary not between 5000 and 12000;

OUTPUT:



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

```
1 SELECT lastname,salary from MY_employee
2 where salary not between 5000 and 12000;
```

The results section displays the following data:

LASTNAME	SALARY
Ropebur	1550
csk	13000
Patel	1000
drexler	1100

4 rows returned in 0.01 seconds

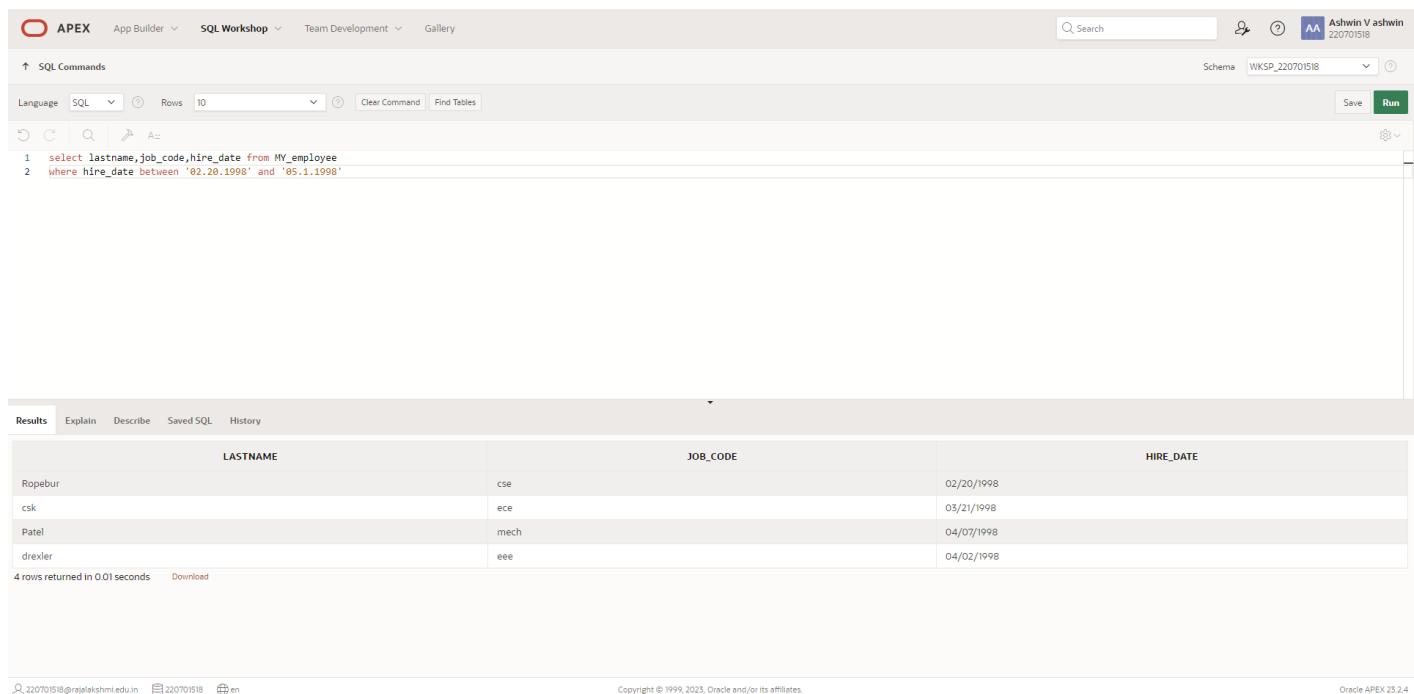
4. Display the employee last name, job ID, and start date of employees hired between February 20,1998 and May 1,1998.order the query in ascending order by start date.(hints: between)

QUERY:

Select last_name, job_code,hire_date from MY_employee

Where hire_date between '2.20.1998' and '5.1.1998';

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a user profile for 'Ashwin V ashwin' with the ID '220701518'. The main workspace is titled 'SQL Commands' with a schema of 'WKSP_220701518'. The SQL editor contains the following code:

```
1 select lastname,job_code,hire_date from MY_employee
2 where hire_date between '02.20.1998' and '05.1.1998';
```

The results section displays the output of the query:

LASTNAME	JOB_CODE	HIRE_DATE
Ropebur	cse	02/20/1998
csk	ece	03/21/1998
Patel	mech	04/07/1998
drexler	eee	04/02/1998

Below the results, it says '4 rows returned in 0.01 seconds'.

At the bottom of the page, there are footer links for '220701518@rajkalakshmi.edu.in', '220701518', and 'en'. The copyright notice reads 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the page is identified as 'Oracle APEX 23.2.4'.

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 MY_employee

Where department_number in (20,50) order by last_name;

OUTPUT:

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

```
1 select lastname,DEPARTMENT_NUMBER FROM MY_EMPLOYEE
2 WHERE DEPARTMENT_NUMBER IN(20,50)
3 ORDER BY lastname;
```

The results section displays the following data:

LASTNAME	DEPARTMENT_NUMBER
Ropebur	20
csk	50

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

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

QUERY:

```
SELECT last_name,salary  
FROM MY_employee  
WHERE salary BETWEEN 5000 AND 12000  
AND department_number IN (20, 50)
```

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's name, schema (WKSP_220701518), and a 'Run' button.

In the main workspace, the SQL Commands tab is active. The query entered is:

```
1 select lastname,salary FROM MY_EMPLOYEE  
2 WHERE salary between 5000 and 12000 AND  
3 DEPARTMENT_NUMBER IN (20,50);
```

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

LASTNAME	SALARY
Ropebur	5000
csk	7000

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

At the bottom of the page, there are footer links for 220701518@rajalakshmi.edu.in, 220701518, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and 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 lastname,hire_date from MY_employee  
where hire_date like '1994%';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Ashwin V ashwin 220701518', and a toolbar with Save and Run buttons. The main workspace displays the SQL command:

```
1 select lastname,hire_date from MY_employee where hire_date like '1994%';
```

Below the command, the Results tab is selected, showing the message "no data found". The bottom footer contains copyright information: "Copyright © 1999, 2025, Oracle and/or its affiliates." and "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_code  
FROM MY_employee  
WHERE manager IS NULL;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile "Ashwin V ashwin" and the schema "WKSP_220701518". The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
1 SELECT last_name, job_code  
2 FROM MY_employee  
3 WHERE manager IS NULL;  
4
```

The Results tab displays the output of the query:

LASTNAME	JOB_CODE
Ropebur	cse
drexler	eee

Below the table, it says "2 rows returned in 0.01 seconds" and there is a "Download" link.

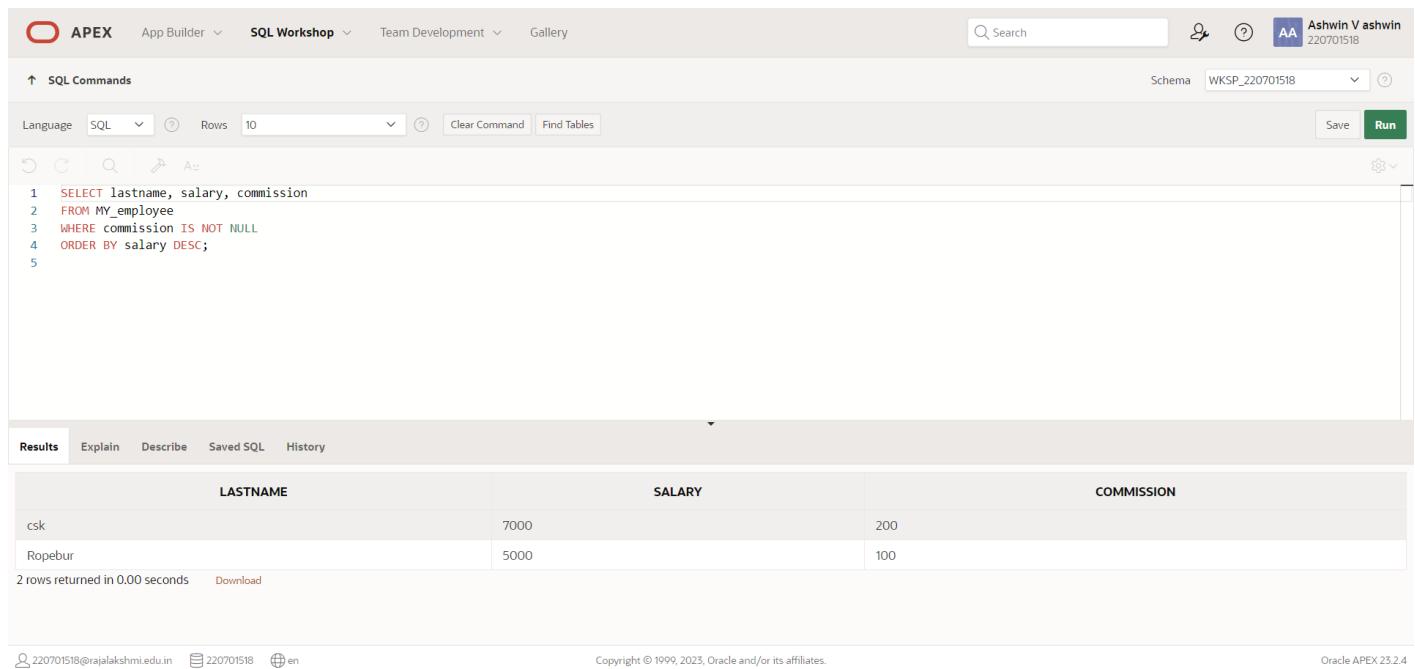
At the bottom, the footer includes links for 220701518@rajalakshmi.edu.in, 220701518, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

9. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.(hints: is not nul,orderby)

QUERY:

```
SELECT lastname, salary, commission  
FROM MY_employee  
WHERE commission IS NOT NULL  
ORDER BY salary DESC;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Ashwin V ashwin 220701518', and buttons for Save and Run.

The SQL Commands tab is active, displaying the following SQL code:

```
1 SELECT lastname, salary, commission  
2 FROM MY_employee  
3 WHERE commission IS NOT NULL  
4 ORDER BY salary DESC;  
5
```

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

LASTNAME	SALARY	COMMISSION
csk	7000	200
Ropebur	5000	100

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

At the bottom, the footer includes links for 220701518@rajalakshmi.edu.in, 220701518, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and 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 lastname  
FROM MY_employee  
WHERE lastname 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, there's a user profile for 'Ashwin V ashwin' and a schema dropdown set to 'WKSP_220701518'. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. Below this, the SQL query is displayed:

```
1 SELECT lastname  
2 FROM MY_employee  
3 WHERE lastname LIKE '__a%';  
4
```

The results tab is selected, showing a single row of output:

LASTNAME
RAAVI

Below the results, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for '220701518@rajalakshmi.edu.in', '220701518', and 'en', along with copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and '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 MY_employee  
WHERE last_name LIKE '%a%' AND last_name LIKE '%e%';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and session information for "Ashwin V ashwin 220701518". The main area is titled "SQL Commands" with tabs for Language (SQL selected), Rows (10), Clear Command, Find Tables, Save, and Run.

The SQL code entered is:

```
1 SELECT lastname  
2 FROM MY_employee  
3 WHERE lastname LIKE '%a%' AND lastname LIKE '%e%';  
4
```

The results tab shows the output:

LASTNAME
Patel

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

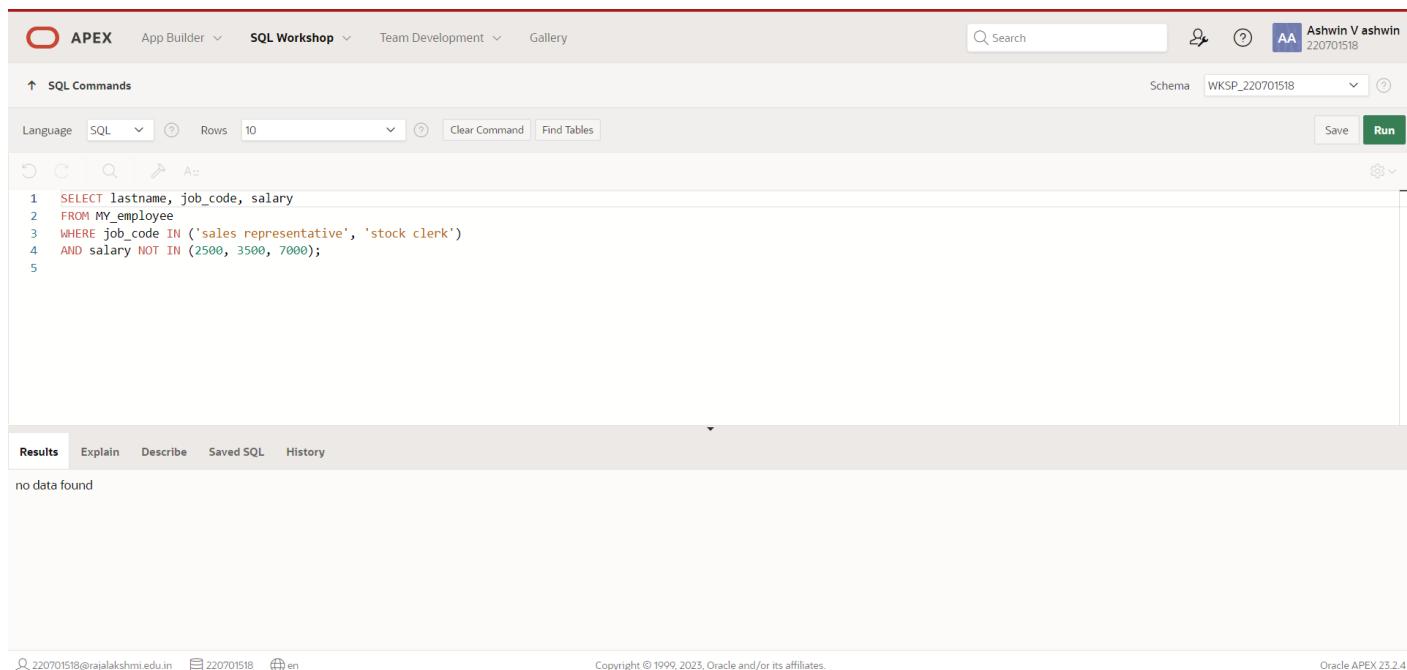
At the bottom, the footer includes links for 220701518@rajalakshmi.edu.in, 220701518, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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 lastname, job_code, salary  
FROM MY_employee  
WHERE job_code IN ('sales representative', 'stock clerk')  
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 (selected), Team Development, and Gallery. On the right, there's a search bar, user profile for Ashwin V ashwin, and session information (Schema: WKSP_220701518). Below the toolbar, the SQL command input area contains the query provided in the question. The results tab is selected, showing the message "no data found". At the bottom, footer information includes the user's email (220701518@rajalakshmi.edu.in), session ID (220701518), language (en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 23.4).

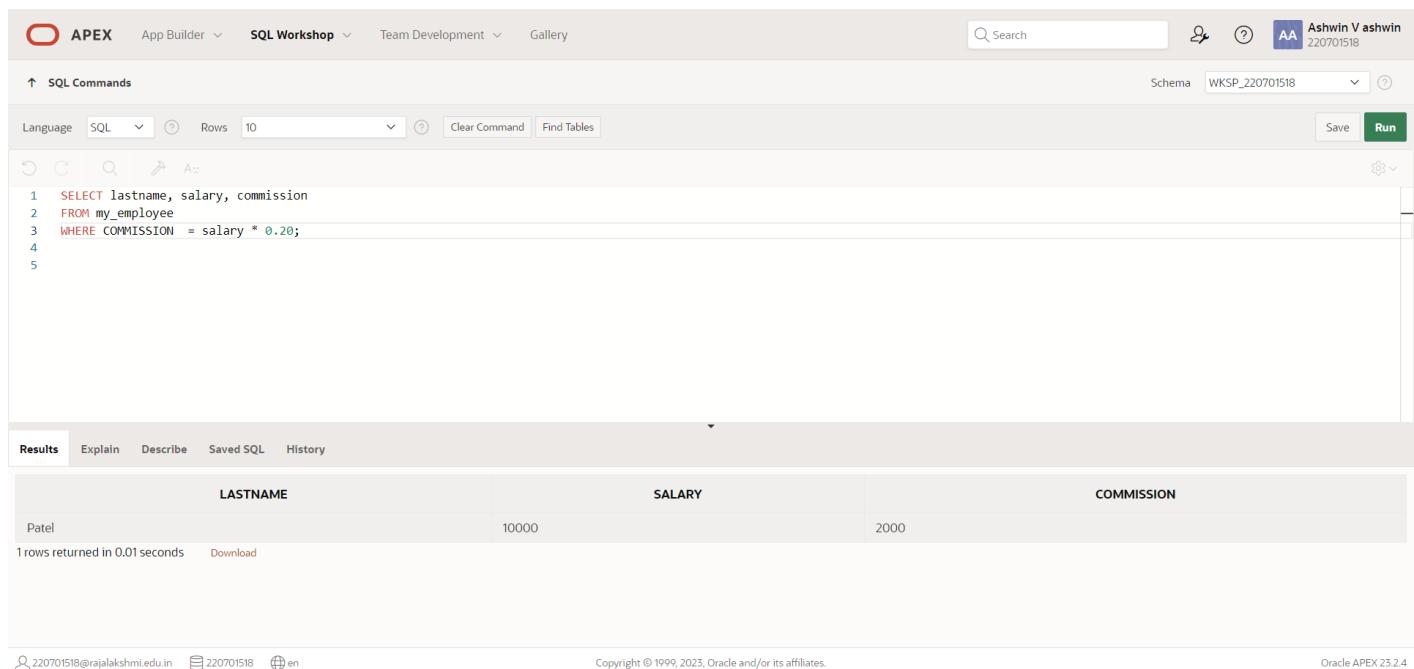
```
1 SELECT lastname, job_code, salary  
2 FROM MY_employee  
3 WHERE job_code IN ('sales representative', 'stock clerk')  
4 AND salary NOT IN (2500, 3500, 7000);  
5
```

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

QUERY:

```
SELECT lastname, salary, commission  
FROM MY_employee  
WHERE commission = salary * 0.20;
```

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 'Ashwin V ashwin' and a schema dropdown set to 'WKSP_220701518'. The main area has tabs for SQL Commands and Results. Under SQL Commands, the query is displayed:

```
1 SELECT lastname, salary, commission  
2 FROM my_employee  
3 WHERE COMMISSION = salary * 0.20;  
4  
5
```

Under Results, the output is shown in a table:

LASTNAME	SALARY	COMMISSION
Patel	10000	2000

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

At the bottom, the footer includes links for 220701518@rajalakshmi.edu.in, 220701518, and en, along with copyright information: '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:

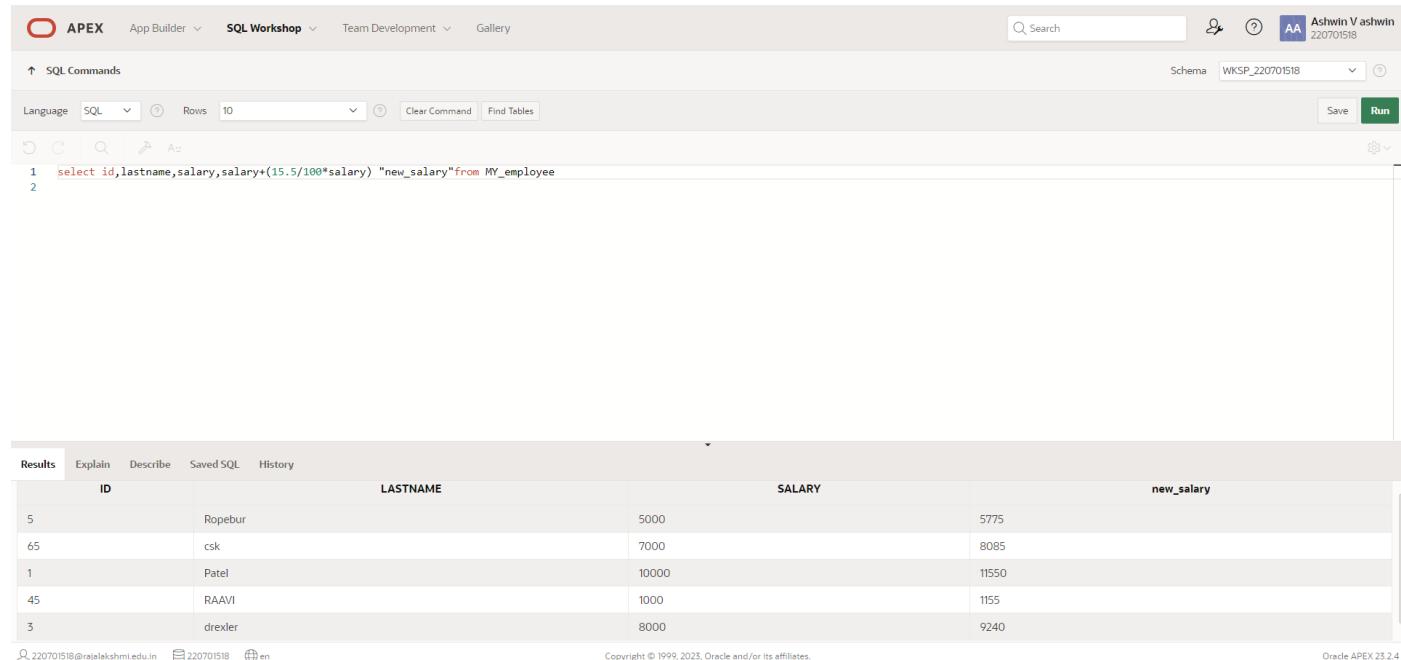
The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is active. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'select sysdate from dual;'. The 'Run' button is visible at the bottom right of the command input field. In the results section, the output is displayed under the 'SYSDATE' column, showing the value '03/13/2024'. The status bar at the bottom indicates '1 rows returned in 0.02 seconds'.

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

QUERY:

```
select id,lastname,salary,salary+(15.5/100*salary) "new_salary"from MY_EMPLOYEE;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Ashwin V ashwin' (220701518), and a 'Run' button. The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the following code:

```
1 select id,lastname,salary,salary+(15.5/100*salary) "new_salary"from MY_employee
2
```

Below the code, the Results tab is selected, displaying a table with four columns: ID, LASTNAME, SALARY, and new_salary. The data is as follows:

ID	LASTNAME	SALARY	new_salary
5	Ropebur	5000	5775
65	csk	7000	8085
1	Patel	10000	11550
45	RAAVI	1000	1155
3	drexler	8000	9240

At the bottom, there are footer links for 220701518@rajalakshmi.edu.in, 220701518, and en, along with copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. and Oracle APEX 23.2.4.

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 id,lastname,salary,salary+(15.5/100*salary) "new_salary",new_salary-salary as "Increase" from  
MY_EMPLOYEE;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Ashwin V ashwin' (ID 220701518), and a 'Run' button. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

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

The Results tab displays the query output as a table:

ID	LASTNAME	SALARY	new_salary	Increase
5	Ropebur	5000	5775	775
65	csk	7000	8085	1085
1	Patel	10000	11550	1550
45	RAAVI	1000	1155	155
3	drexler	8000	9240	1240

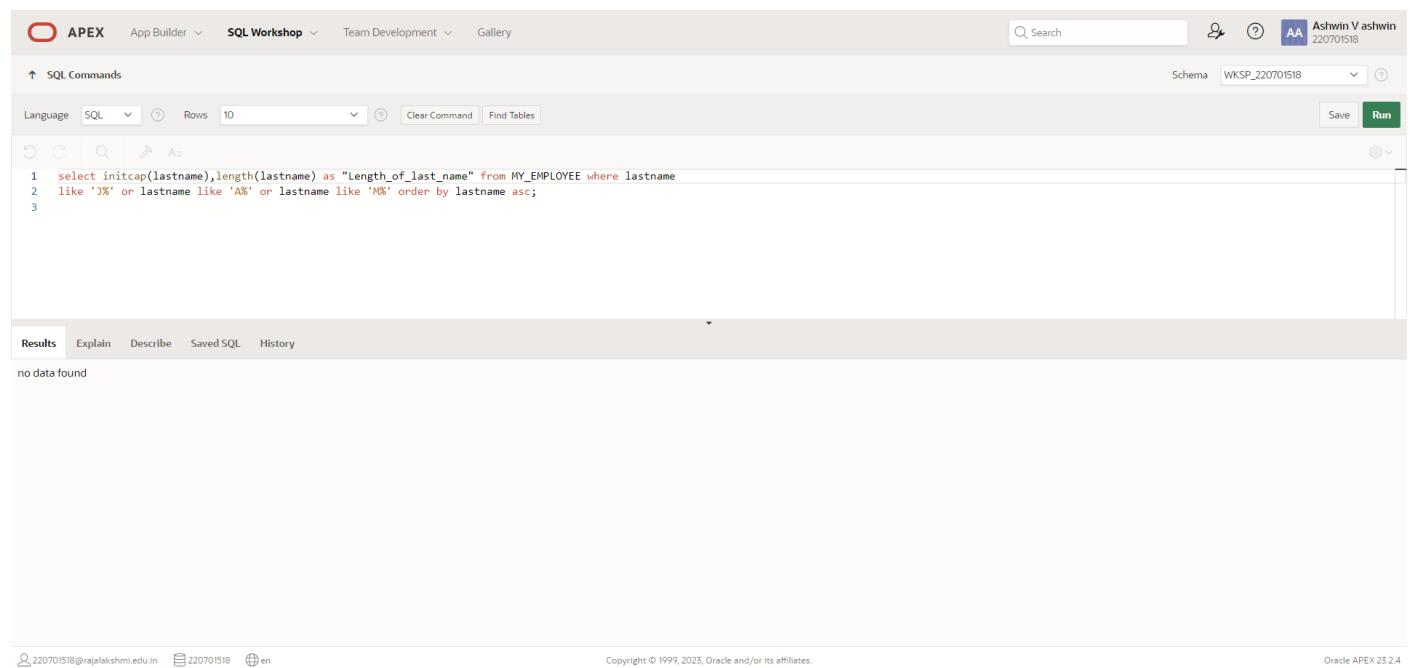
Below the table, it says '5 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom, there are footer links for user info (220701518@rajalakshmi.edu.in, 220701518, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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(lastname),length(lastname) as "Length_of_lastname" from MY_EMPLOYEE where lastname like 'J%' or lastname like 'A%' or lastname like 'M%' order by lastname 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 user profile for 'Ashwin V ashwin' and a timestamp '220701518'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select initcap(lastname),length(lastname) as "Length_of_lastname" from MY_EMPLOYEE where lastname
2 like 'J%' or lastname like 'A%' or lastname like 'M%' order by lastname asc;
3
```

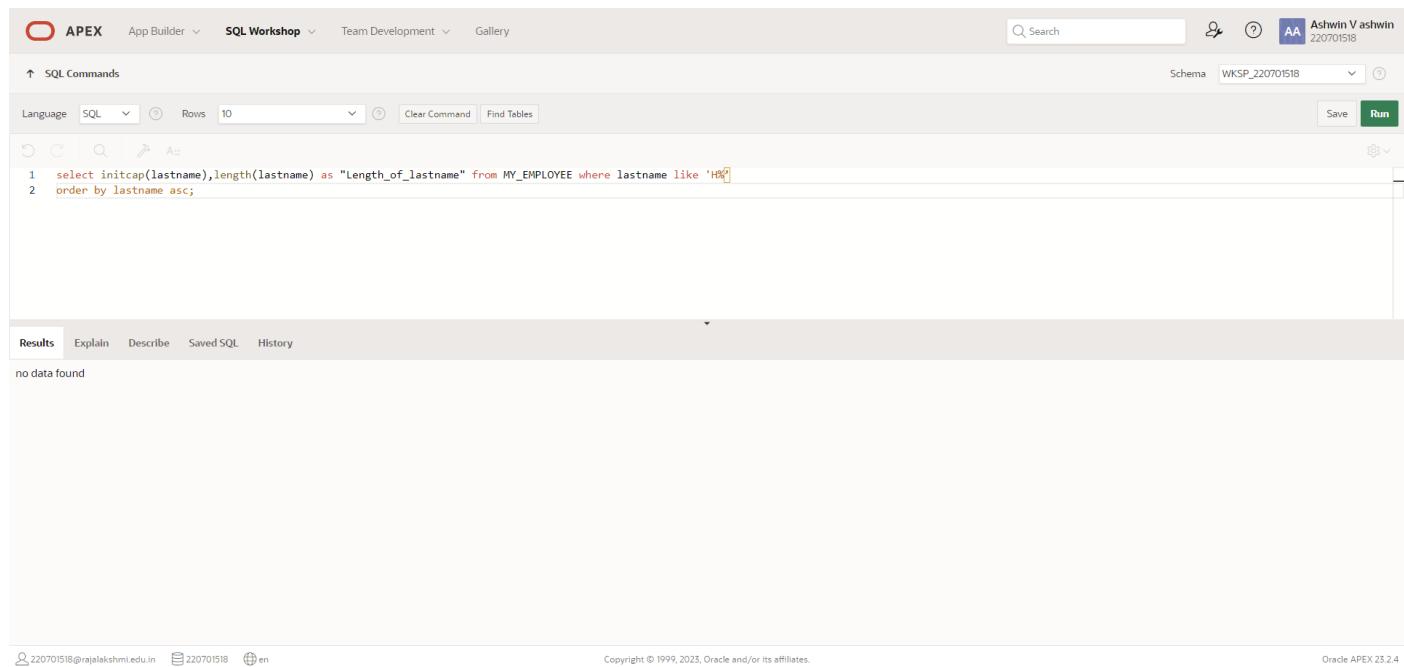
Below the code, the 'Results' tab is selected, showing the message 'no data found'. At the bottom of the page, there are footer links for '220701518@rajalakshmi.edu.in', '220701518', and 'en', along with copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

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

QUERY:

```
select initcap(last_name),length(last_name) as "Length_of_last_name" from MY_EMPLOYEE where last_name like 'H%' 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. The right side shows the user profile 'Ashwin V Ashwin' and the schema 'WKSP_220701518'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select initcap(lastname),length(lastname) as "Length_of_lastname" from MY_EMPLOYEE where lastname like 'H%'  
2 order by lastname asc;
```

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

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 lastname,round((sysdate-hire_date)/30,0) as "MONTHS_WORKED" from MY_EMPLOYEE order by round((sysdate-hire_date)/30,0) 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 the user profile 'Ashwin V ashwin' and the schema 'WKSP_220701518'. The main area has tabs for SQL Commands, Results (selected), Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
1 select lastname,round((sysdate-hire_date)/30,0) as "MONTHS_WORKED" from MY_EMPLOYEE order by
2 round((sysdate-hire_date)/30,0) asc;
3
```

The Results tab displays the output of the query:

LASTNAME	MONTHS_WORKED
Patel	316
csk	365
drexler	365
Ropebur	366
RAAVI	-

Below the table, it says '5 rows returned in 0.00 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for email, profile, language, copyright information ('Copyright © 1999, 2023, Oracle and/or its affiliates.'), and the version ('Oracle APEX 23.2.4').

7. Create a report that produces the following for each employee:

<employee last name> earns<salary>monthly but wants <3 times salary>.Label the column Dream Salaries.

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile of Ashwin V ashwin (220701518). The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to 'WKSP_220701518'. Below the search bar are buttons for Save and Run. The SQL editor contains the following code:

```
1 select lastname||' earns'||salary||' monthly but wants'||salary*3 as "DREAM_SALARIES" from  
2 MY_EMPLOYEE;
```

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

DREAM_SALARIES
Ropebur earns 5000 monthly but wants 15000
csk earns 7000 monthly but wants 21000
Patel earns 10000 monthly but wants 30000
RAAVI earns 1000 monthly but wants 3000
drexler earns 8000 monthly but wants 24000

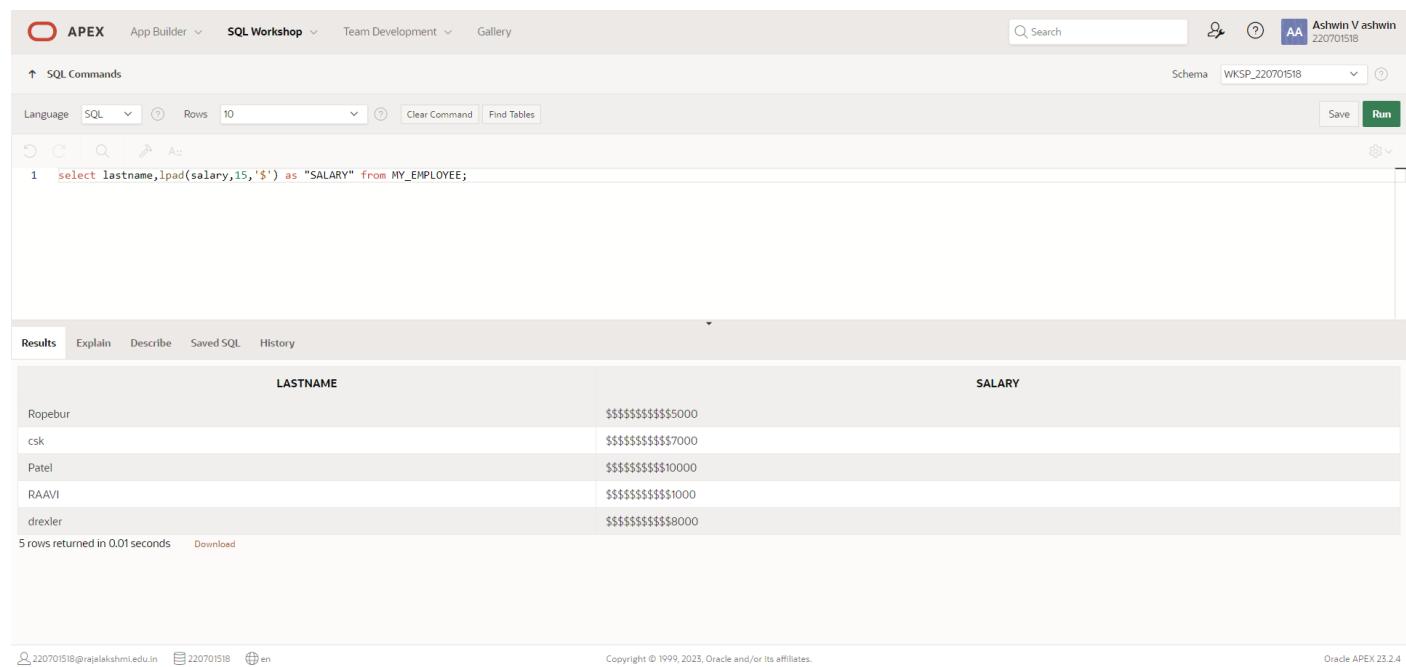
Below the table, it says '5 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. and Oracle APEX 23.2.4.

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 lastname,lpad(salary,15,'$') as "SALARY" from MY_EMPLOYEE;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'Ashwin V Ashwin' and a timestamp '220701518'. The main workspace has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the query: 'select lastname,lpad(salary,15,'\$') as "SALARY" from MY_EMPLOYEE;'. The Results tab displays the output in a table format:

LASTNAME	SALARY
Ropebur	\$\$\$\$\$\$\$\$\$\$5000
csk	\$\$\$\$\$\$\$\$\$\$7000
Patel	\$\$\$\$\$\$\$\$\$\$10000
RAAVI	\$\$\$\$\$\$\$\$\$\$1000
drexler	\$\$\$\$\$\$\$\$\$\$8000

Below the table, it says '5 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes footer links for 220701518@rajalakshmi.edu.in, 220701518, en, 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 lastname,hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY'),'FMDay, "the  
"FMDD " of "FMMonth, YYYY') AS REVIEW FROM MY_EMPLOYEE;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows the user profile of Ashwin V ashwin (220701518). The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, displaying the following SQL code:

```
1 SELECT lastname,hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY'),'FMDay, "the  
2 "FMDD " of "FMMonth, YYYY') AS REVIEW FROM MY_EMPLOYEE;  
3
```

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

LASTNAME	HIRE_DATE	REVIEW
Ropebur	02/20/1994	Monday, the 22 of August, 1994
csk	03/21/1994	Monday, the 26 of September, 1994
Patel	04/07/1998	Monday, the 12 of October, 1998
RAAVI	-	-
drexler	04/02/1994	Monday, the 03 of October, 1994

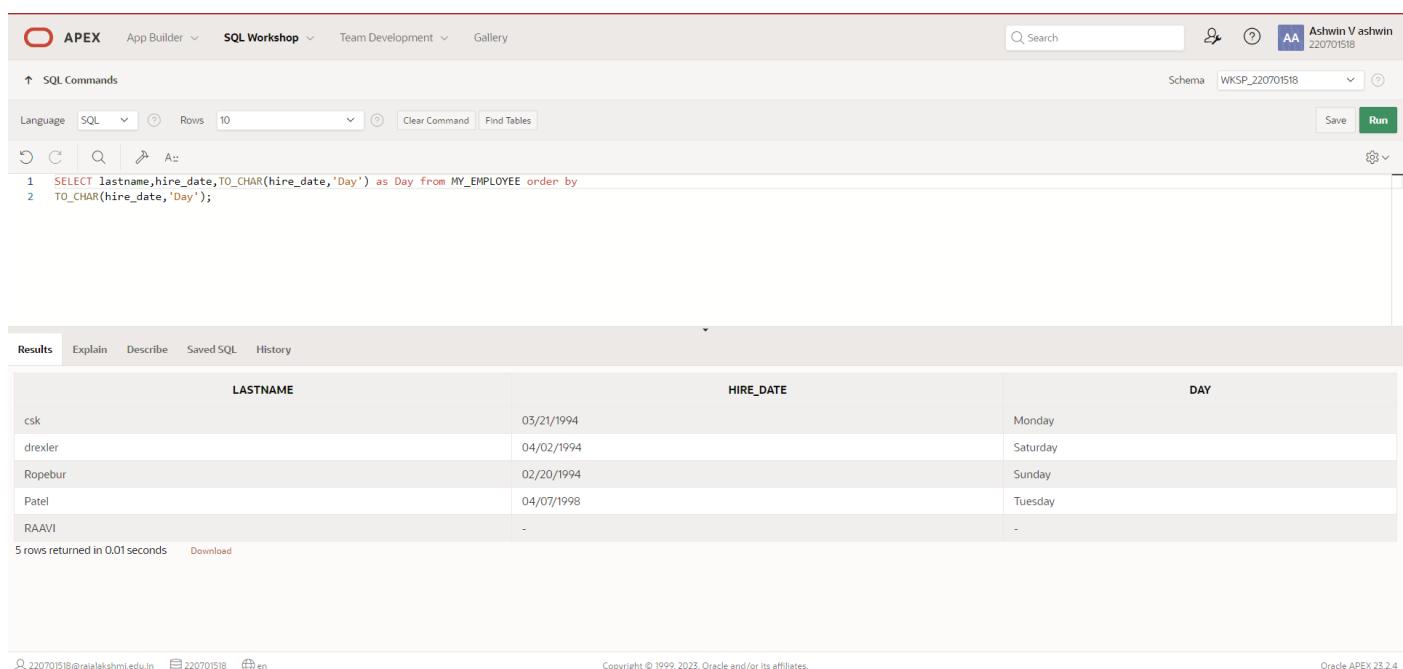
At the bottom left, it says "5 rows returned in 0.00 seconds" and there is a "Download" link. The bottom right corner indicates the version "Oracle APEX 23.2.4".

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

OUTPUT:



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

```
1 SELECT lastname,hire_date,TO_CHAR(hire_date,'Day') as Day from MY_EMPLOYEE order by
2 TO_CHAR(hire_date,'Day');
```

The results section displays the following data:

LASTNAME	HIRE_DATE	DAY
csk	03/21/1994	Monday
drexler	04/02/1994	Saturday
Ropebur	02/20/1994	Sunday
Patel	04/07/1998	Tuesday
RAAVI		-

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

At the bottom, the footer includes: 220701518@rajalakshmi.edu.in, 220701518, 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:

DISPLAYING DATA FROM MULTIPLE TABLES

EX_NO:7

DATE:

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

QUERY:

```
select e.name,e.dept_id,d.dept_name from employees e,departments d  
where e.dept_id=d.dept_id;
```

OUTPUT:

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

```
1 SELECT LASTNAME ,DEPARTMENT_NUMBER,DEPARTMENT_NAME FROM MY_EMPLOYEE
```

The results are displayed in a table:

LASTNAME	DEPARTMENT_NUMBER	DEPARTMENT_NAME
Ropebur	20	CSE
csk	50	IT
Patel	30	MECH
RAAVI	20	-
drexler	25	CSE

5 rows returned in 0.02 seconds

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

QUERY:

```
select distinct e.job_id,d.location_id from employees e,departments d where e.dept_id=d.dept_id  
and e.dept_id =80;
```

OUTPUT:

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

```
1 SELECT DISTINCT e.job_id, d.location_id  
2 FROM employees e, departments d  
3 WHERE e.dept_id = d.dept_id  
4 AND e.dept_id = 80;
```

The results are displayed in a table:

JOB_ID	LOCATION_ID
55	12
80	12

2 rows returned in 0.01 seconds

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

QUERY:

```
Select e.last_name,d.dept_name,d.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. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Ashwin V ashwin' and the schema 'WKSP_220701518'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL tab contains the following code:

```
1 SELECT lastname,department_name,LOCATION_ID,CITY FROM MY_EMPLOYEE  
2 WHERE COMMISSION IS NOT NULL
```

The Results tab displays the query results in a grid format:

LASTNAME	DEPARTMENT_NAME	LOCATION_ID	CITY
Ropebur	cse	101	CHENNAI
csk	IT	102	DELHI
Patel	MECH	103	CHENNAI

Below the grid, it says '3 rows returned in 0.00 seconds'.

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:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Ashwin V ashwin' and the schema 'WKSP_220701518'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL tab contains the following code:

```
1 SELECT LASTNAME,DEPARTMENT_NAME FROM MY_EMPLOYEE  
2 WHERE LASTNAME LIKE '%a%';
```

The Results tab displays the query results in a grid format:

LASTNAME	DEPARTMENT_NAME
Ropebur	cse
patel	MECH

Below the grid, it says '2 rows returned in 0.00 seconds'.

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:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows the user 'Ashwin V ashwin' and the schema 'WKSP_22070151B'. The main area has tabs for 'SQL Commands', 'Language' (set to 'SQL'), 'Rows' (set to 10), and 'Clear Command'. Below these are icons for 'Run', 'Save', and 'Run'. The SQL command entered is:

```
1 SELECT LASTNAME,JOB,DEPARTMENT_NUMBER,DEPARTMENT_NAME FROM MY_EMPLOYEE  
2 WHERE JOB LIKE 'TORONTO'
```

The results section shows a single row returned in 0.01 seconds:

LASTNAME	JOB	DEPARTMENT_NUMBER	DEPARTMENT_NAME
RAAVI	TORONTO	20	

At the bottom, there are links for '22070151B@rajalakshmi.edu.in', '22070151B', and 'en'. The footer indicates 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and '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 employes w join employes m on (w.manager_id=m.emp_id);
```

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 right side shows the user 'Ashwin V ashwin' and the schema 'WKSP_22070151B'. The main area has tabs for 'SQL Commands', 'Language' (set to 'SQL'), 'Rows' (set to 10), and 'Clear Command'. Below these are icons for 'Run', 'Save', and 'Run'. The SQL command entered is:

```
1 SELECT EMPLOYEE_ID AS EMP#,  
2 MANAGER_LASTNAME AS Manager,  
3 MANAGER_NUMBER AS Mgr#  
4 FROM MY_EMPLOYEE  
5
```

The results section shows 5 rows returned in 0.02 seconds:

EMP#	MANAGER	MGR#
136	CHOTU	1
146	DOLU	2
156	BOLU	3
-	KALIA	4
176	TUN	5

At the bottom, there are links for '22070151B@rajalakshmi.edu.in', '22070151B', and 'en'. The footer indicates 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and '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 w.last_name "Employee", w.emp_id "EMP#", m.last_name "Manager", m.emp_id "Mgr#" from employes w  
Left outer join employes m on(w.manager_id=m.emp_id);
```

OUTPUT:

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

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

The results window displays the following data:

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

3 rows returned in 0.01 seconds

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

QUERY:

```
Select e.dept_id department,e.last_name employee,c.last_name colleague from employes e join employes 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:

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

```
1 SELECT e.deptid dept, e.lastname employees,  
2       c.lastname colleague  
3   FROM employees e JOIN employees c  
4  ON (e.deptid = c.deptid)  
5 WHERE e.employeeid <> c.employeeid  
6 ORDER BY e.deptid, e.lastname, c.lastname;
```

The results window displays the following data:

DEPT	EMPLOYEES	COLLEAGUE
100	Garcia	Joan
100	Joan	Garcia

2 rows returned in 0.01 seconds

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 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'Ashwin V ashwin' with the ID '220701518'. The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the following query is displayed:

```
1 SELECT e.lastname, e.jobid, d.deptname, e.salary, j.gradelevel
2 FROM employees e JOIN dept d
3 ON (e.deptid = d.deptid)
4 JOIN jobgrade j
5 ON (e.salary BETWEEN j.lowestsal AND j.highestsal);
```

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

LASTNAME	JOBID	DEPTNAME	SALARY	GRADELEVEL
king	122	executive	1000	3
king	122	executive	1000	1
zlotkey	SALES	IT	10000	2
zlotkey	SALES	IT	10000	2

Below the table, it says '4 rows returned in 0.06 seconds'. The bottom of the page includes copyright information for Oracle and the APEX version '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 employees e join employees.davies on davies.hire_date < e.hire_date
Where davies.last_name = 'Davies';
```

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 'Ashwin V ashwin' with the ID '220701518'. The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the following query is displayed:

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

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

LAST_NAME	HIRE_DATE
roy	02/03/2020

Below the table, it says '1 rows returned in 0.01 seconds'. The bottom of the page includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

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

QUERY:

```
select e.last_name as Employee,e.hire_date as Emp_hired,e.man_name as manager,m.hire_date as mgr_hired from employees join employees m on e.man_name=m.last_name where e.hire_date<m.hire_date;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile "Ashwin V ashwin 220701518" and the schema "WKSP_220701518". The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query code. The Results tab displays the output in a grid format with four columns: EMPLOYEE, EMP_HIRED, MANAGER, and MGR_HIRED. The output shows one row for employee 'davies' with hire date 02/26/1998 and manager 'davies' with hire date 02/05/1999. The bottom of the page includes copyright information and a footer for Oracle APEX 25.2.4.

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

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

RESULT:

AGGREGATING DATA USING GROUP FUNCTIONS

EX_NO:8

DATE:

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

TRUE

2. Group functions include nulls in calculations.
True/False

FALSE

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

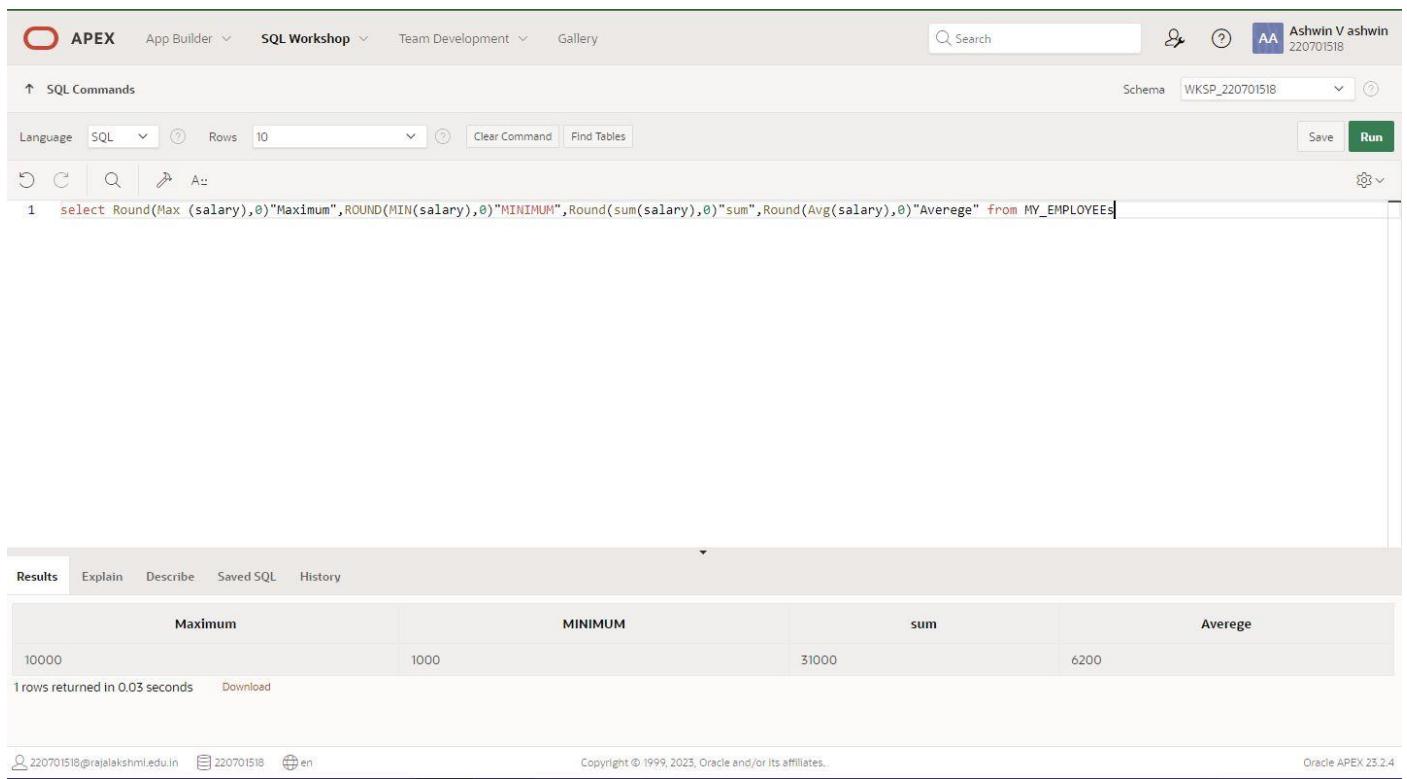
FALSE

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

QUERY:

```
Select
round(max(salary),0)"Maximum",round(min(salary),0)"Minimum",round(sum(salary),0)"sum",round(avg(salary),0)
"average" from EMPA;
```

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 Ashwin V ashwin (ID: 220701518). The SQL Commands tab is selected, showing the following query:

```
1 select Round(Max (salary),0)"Maximum",ROUND(MIN(salary),0)"MINIMUM",Round(sum(salary),0)"sum",Round(Avg(salary),0)"Average" from MY_EMPLOYEES
```

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

Maximum	MINIMUM	sum	Average
10000	1000	31000	6200

Below the table, it says "1 rows returned in 0.03 seconds". The bottom footer includes links for Explain, Saved SQL, History, and copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4.

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

QUERY:Select

```
job_id,round(max(salary),0)"MAXIMUM",round(Min(salary),0)"Minimun",round(sum(salary),0)"sum",round(Avg(sal  
ary),0)"average" from EMPA group by job_id;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows the user 'Ashwin V ashwin' and the schema 'WKSP_220701518'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The query entered is:

```
1 select Job_id, Round(Max (salary),0)"Maximum",ROUND(MIN(salary),0)"MINIMUM",Round(sum(salary),0)"sum",Round(Avg(salary),0)"Averge" from MY_EMPLOYEE GROUP BY Job_id
```

The results tab is selected, displaying the following data:

JOB_ID	Maximum	MINIMUM	sum	Average
cse	5000	5000	5000	5000
-	1000	1000	1000	1000
eee	8000	8000	8000	8000

At the bottom, it shows the connection details '220701518@rajalakshmi.edu.in' and 'en', the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

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

QUERY:Select job_id,count(*) from EMPA where job_id='47' group by job_id;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows the user 'Ashwin V ashwin' and the schema 'WKSP_220701518'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The query entered is:

```
1 select Job_id,Count(*)from MY_Employee group by Job_id
```

The results tab is selected, displaying the following data:

JOB_ID	COUNT(*)
cse	1
-	1
eee	1

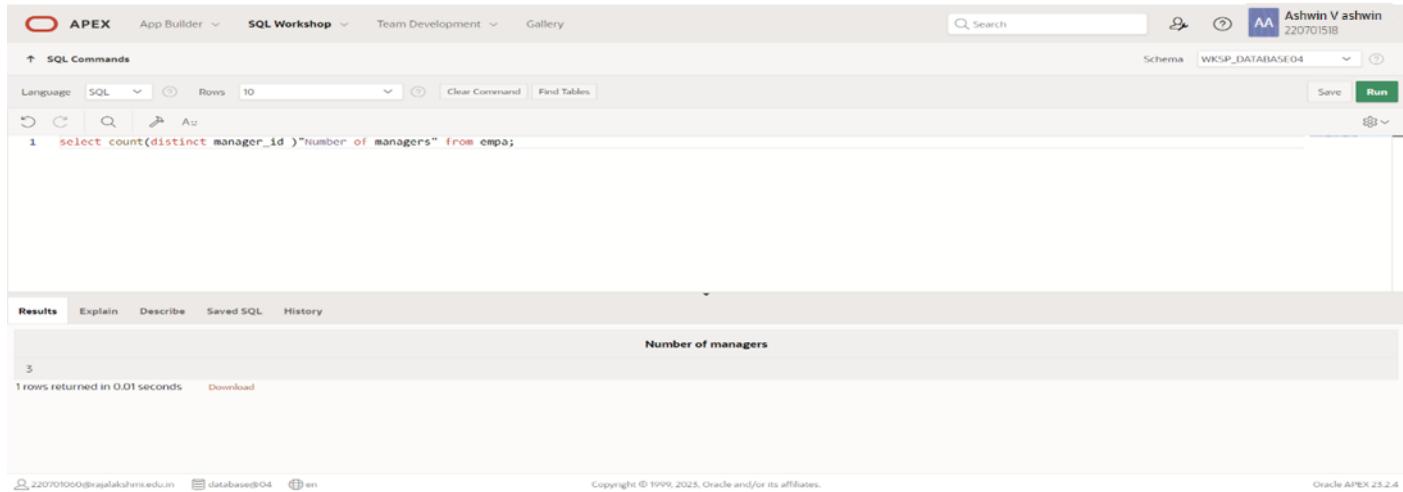
At the bottom, it shows the connection details '220701518@rajalakshmi.edu.in' and 'en', the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

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

QUERY:

Select count(distinct manager_id)"Number of managers" from empa;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is: `select count(distinct manager_id)"Number of managers" from empa;`. The results section displays a single row with the value 3, labeled "Number of managers". The output is returned in 0.01 seconds.

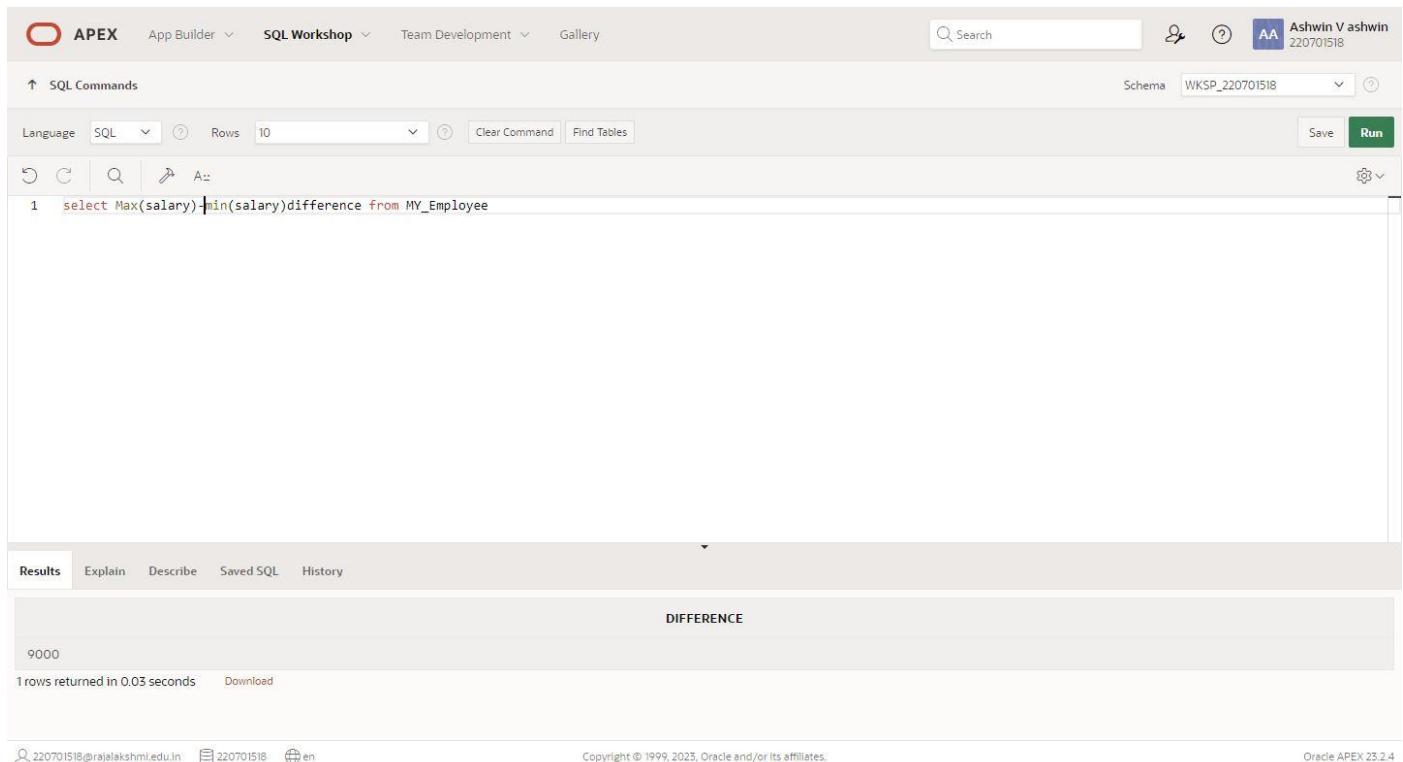
Number of managers
3

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

QUERY:

Select max(salary)-min(salary) difference from empa;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is: `select Max(salary)-min(salary)difference from MY_Employee`. The results section displays a single row with the value 9000, labeled "DIFFERENCE". The output is returned in 0.03 seconds.

DIFFERENCE
9000

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 empa 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 query executed is:

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

The results table has two columns: MANAGERID and MIN(SALARY). The single row returned is:

MANAGERID	MIN(SALARY)
10003	45000

1 rows returned in 0.02 seconds

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 the Oracle APEX SQL Workshop interface. The query executed 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 table has five columns: TOTAL, 1995, 1996, 1997, and 1998. The single row returned is:

TOTAL	1995	1996	1997	1998
5	1	1	1	1

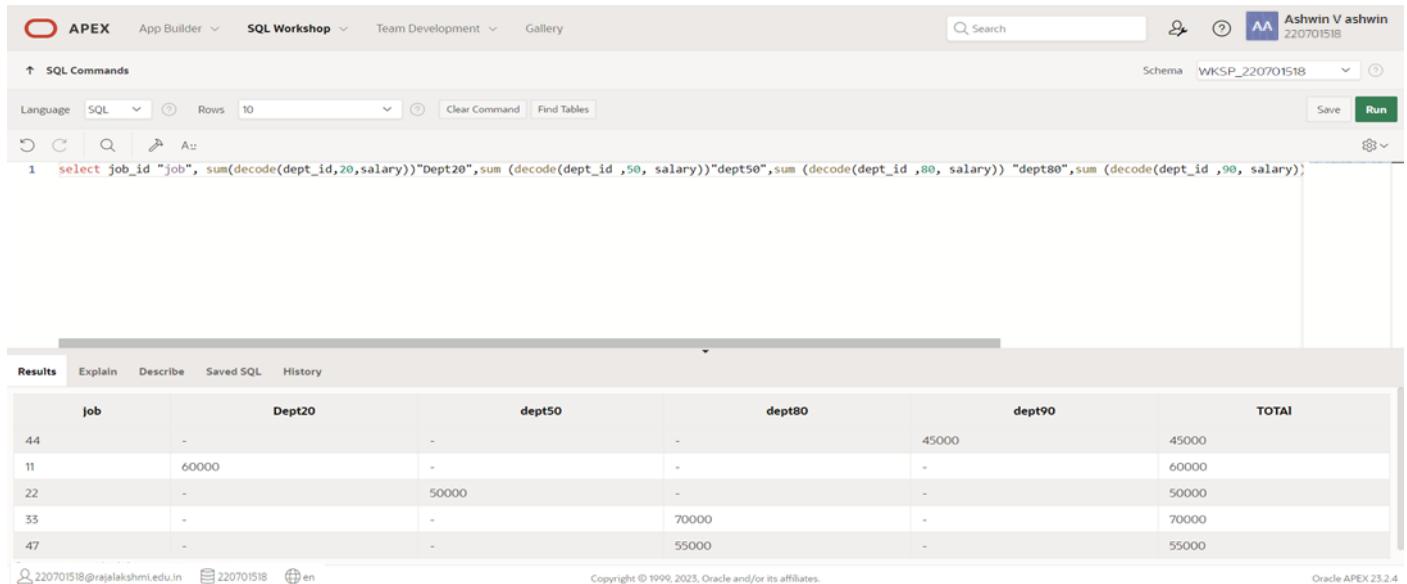
1 rows returned in 0.01 seconds

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:



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

```
1 select job_id "job", sum(decode(dept_id,20,salary))"Dept20", sum (decode(dept_id ,50, salary))"dept50",  
2 sum(decode(dept_id,80,salary))"dept80", sum (decode(dept_id ,90, salary)) "dept90" from empa;
```

The results window displays the output in a grid format:

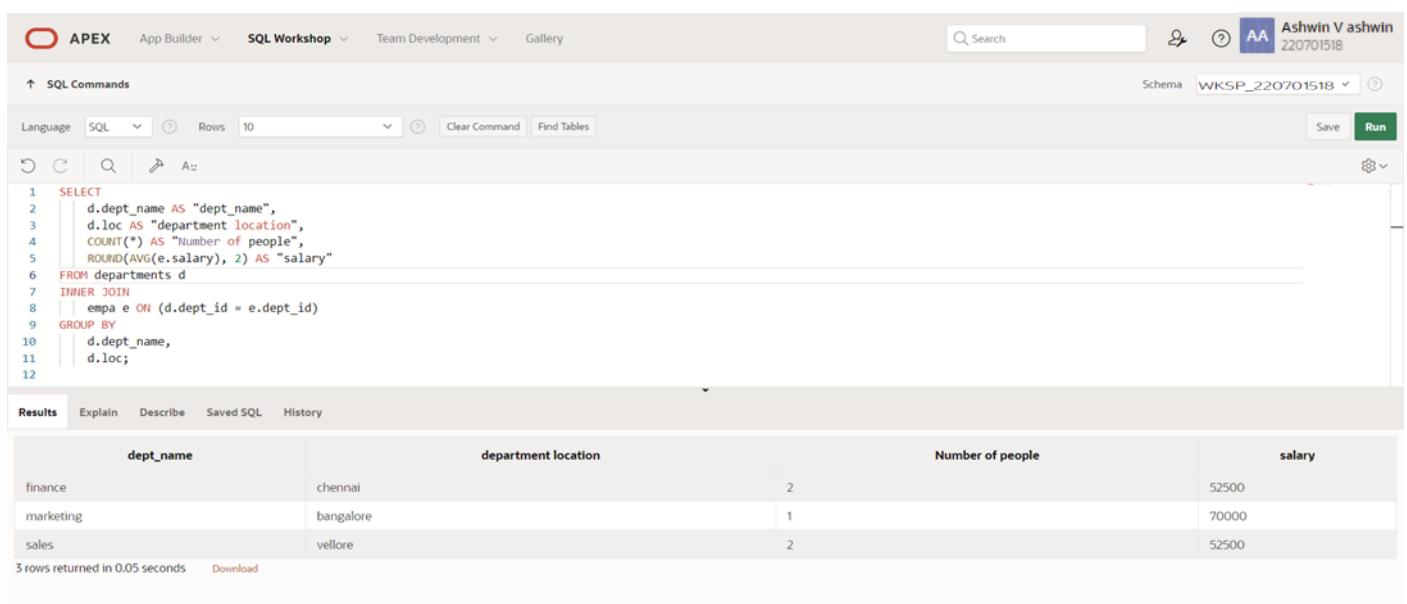
Job	Dept20	Dept50	Dept80	Dept90	Total
44	-	-	-	45000	45000
11	60000	-	-	-	60000
22	-	50000	-	-	50000
35	-	-	70000	-	70000
47	-	-	55000	-	55000

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 empa e on (d.dept_id=e.dept_id)  
Group by d.dept_name,d.loc;
```

OUTPUT:



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

```
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      | empa e ON (d.dept_id = e.dept_id)  
9  GROUP BY  
10    d.dept_name,  
11    d.loc;  
12
```

The results window displays the output in a grid format:

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

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 dept_id=(select dept_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 links for APEX, App Builder, SQL Workshop (selected), Team Development, Gallery, and a search bar. The right side shows the user profile "Ashwin V ashwin 220701518". The main workspace is titled "SQL Commands" and contains the following SQL code:

```
1 Select lastname, hiredate  
2 from employees  
3 where deptid =  
4 (select deptid from employees where lastname like 'zlotkey')and lastname <> 'zlotkey';  
5
```

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

LASTNAME	HIREDATE
Hamilton	10/12/2014

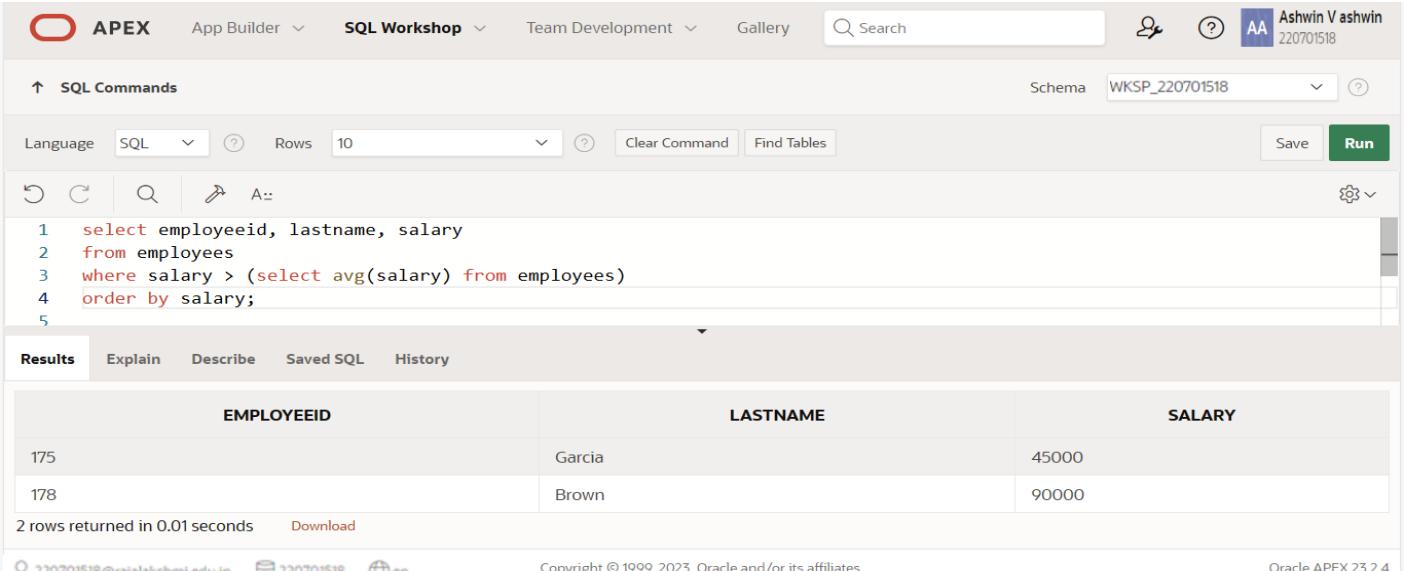
Below the table, it says "1 rows returned in 0.01 seconds" and has a "Download" link. At the bottom of the page, there are footer links for "220701518@rajalakshmi.edu.in", "220701518", "en", "Copyright © 1999, 2023, Oracle and/or its affiliates.", and "Oracle APEX 23.2.4".

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery, along with a search bar and user profile information for 'Ashwin V ashwin' (220701518). The main area is titled 'SQL Commands' and contains the following SQL code:

```
1 select employeeid, lastname, salary
2 from employees
3 where salary > (select avg(salary) from employees)
4 order by salary;
5
```

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

EMPLOYEEID	LASTNAME	SALARY
175	Garcia	45000
178	Brown	90000

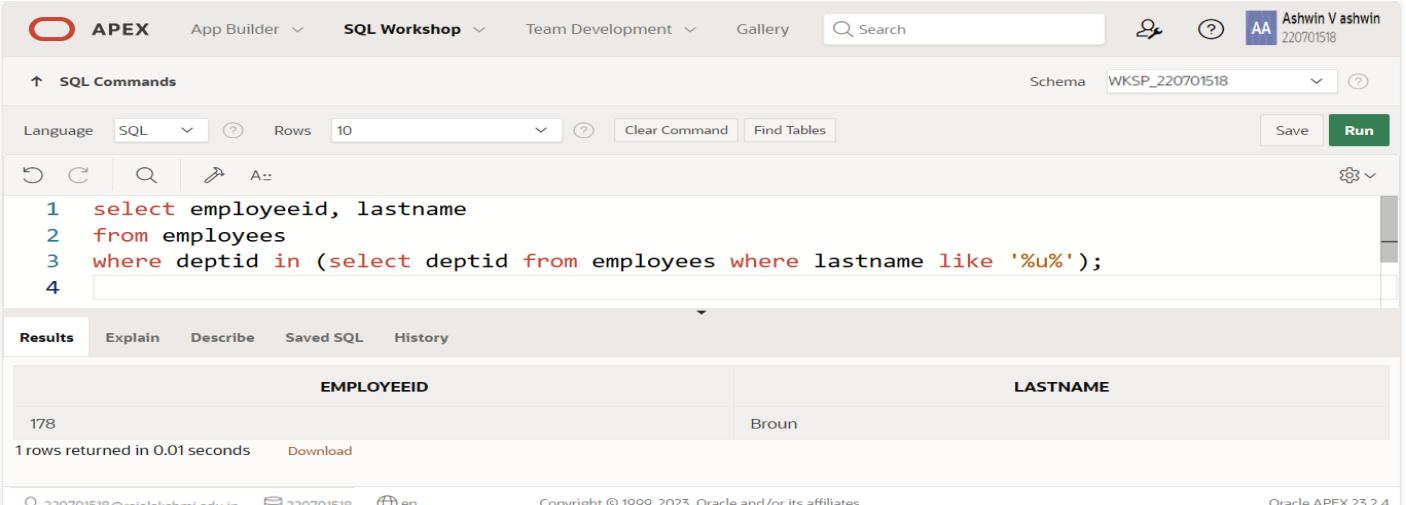
Below the table, it says '2 rows returned in 0.01 seconds' and provides a 'Download' link. The bottom footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

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 emp_id, last_name from employees where dept_id = (select dept_id from employees where last_name like '%u%');
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery, along with a search bar and user profile information for 'Ashwin V ashwin' (220701518). The main area is titled 'SQL Commands' and contains the following SQL code:

```
1 select employeeid, lastname
2 from employees
3 where deptid in (select deptid from employees where lastname like '%u%');
4
```

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

EMPLOYEEID	LASTNAME
178	Brown

Below the table, it says '1 rows returned in 0.01 seconds' and provides a 'Download' link. The bottom footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

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 APEX SQL Workshop interface. The SQL command entered is:

```
1 select lastname, deptid, jobid  
2 from employees  
3 where deptid in (select deptid from dept where locationid =1700);
```

The results table shows two rows:

LASTNAME	DEPTID	JOBID
davies	100	MARKETING
Garcia	100	HRREP

2 rows returned in 0.00 seconds

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

QUERY:

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

OUTPUT:

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

```
1 select lastname,salary from employees  
2 where managerid in (select employeeid from employees where lastname = 'king');
```

The results table shows no data found:

LASTNAME	SALARY
no data found	

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

OUTPUT:

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

```
1 select deptid, lastname, jobid
2 from employees
3 where deptid in (select deptid from dept where deptname = 'executive');
```

The results table shows two rows:

DEPTID	LASTNAME	JOBID
100	davies	MARKETING
100	king	HRREP

2 rows returned in 0.01 seconds

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

OUTPUT:

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

```
1 select employeeid, lastname, salary
2 from employees
3 where salary > (select avg(salary) from employees) and
4 deptid in (select deptid from employees where lastname like '%u%');
```

The results table shows one row:

EMPLOYEEID	LASTNAME	SALARY
178	Broun	90000

1 rows returned in 0.01 seconds

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

RESULT:

USING THE SET OPERATORS

EX_NO:10

DATE:

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

QUERY:

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

OUTPUT:



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

```
1 select dept_id from employees minus select dept_id from employees where job_id='st_clerk'
```

The results section displays the output:

DEPT_ID
45
90

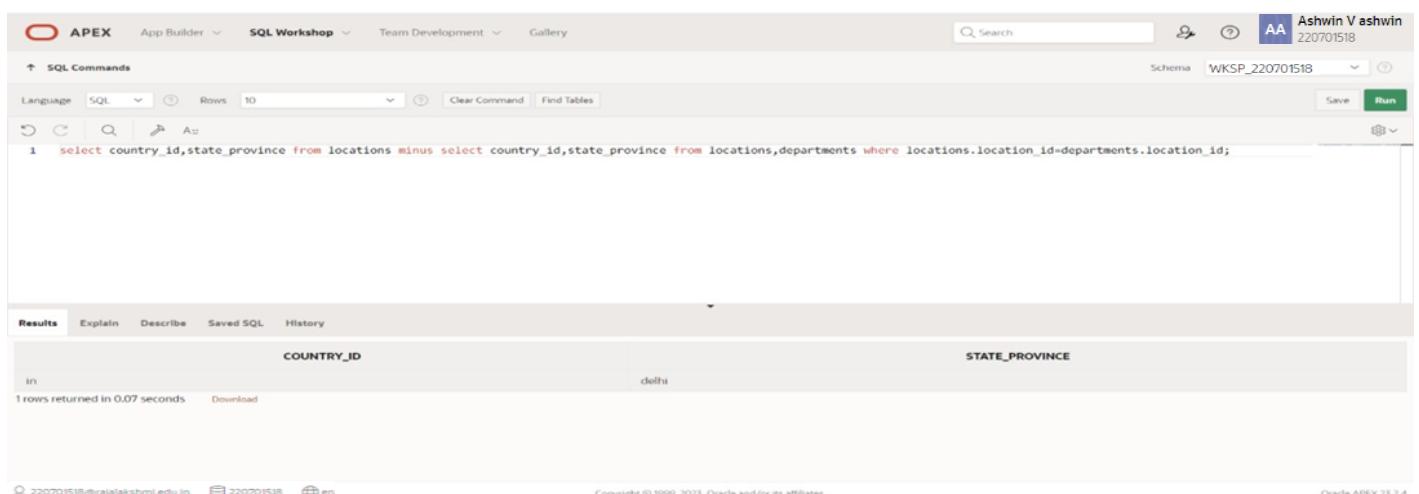
2 rows returned in 0.01 seconds

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

OUTPUT:



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

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

The results section displays the output:

COUNTRY_ID	STATE_PROVINCE
in	delhi

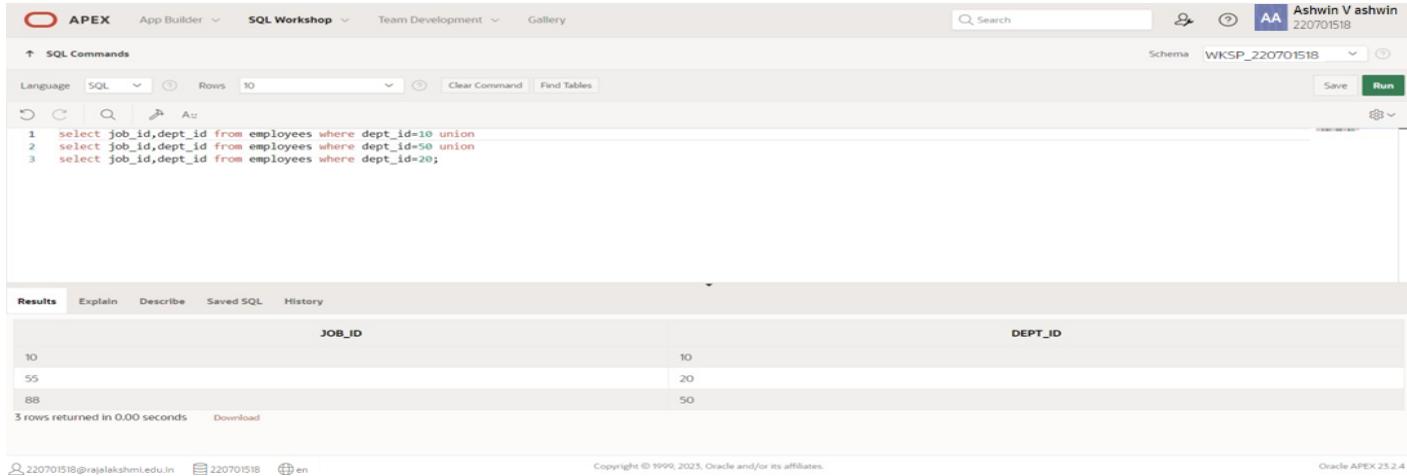
1 rows returned in 0.07 seconds

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command window contains three UNIONed SELECT statements. The results are displayed in a table with two columns: JOB_ID and DEPT_ID. The data is as follows:

JOB_ID	DEPT_ID
10	10
55	20
88	50

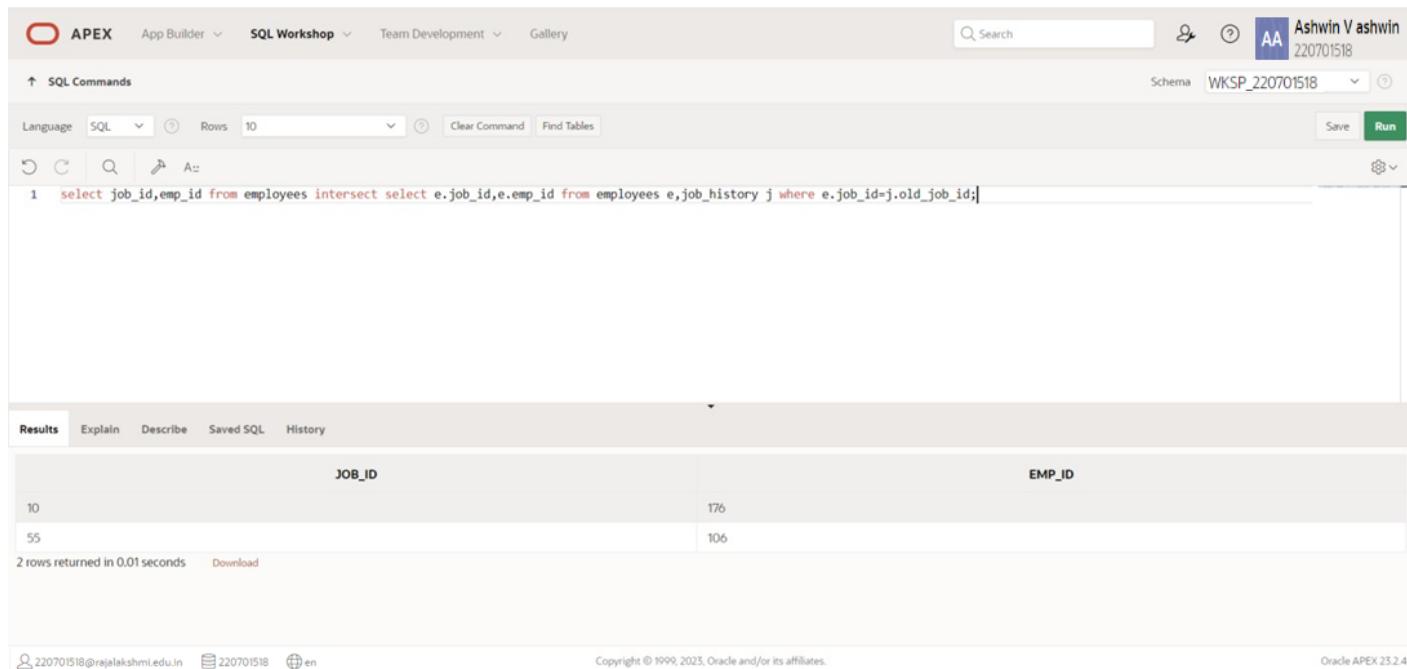
3 rows returned in 0.00 seconds

- 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:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command window contains a query using INTERSECT to find employees who currently have the same job title as their original hire. The results are displayed in a table with two columns: JOB_ID and EMP_ID. The data is as follows:

JOB_ID	EMP_ID
10	176
55	106

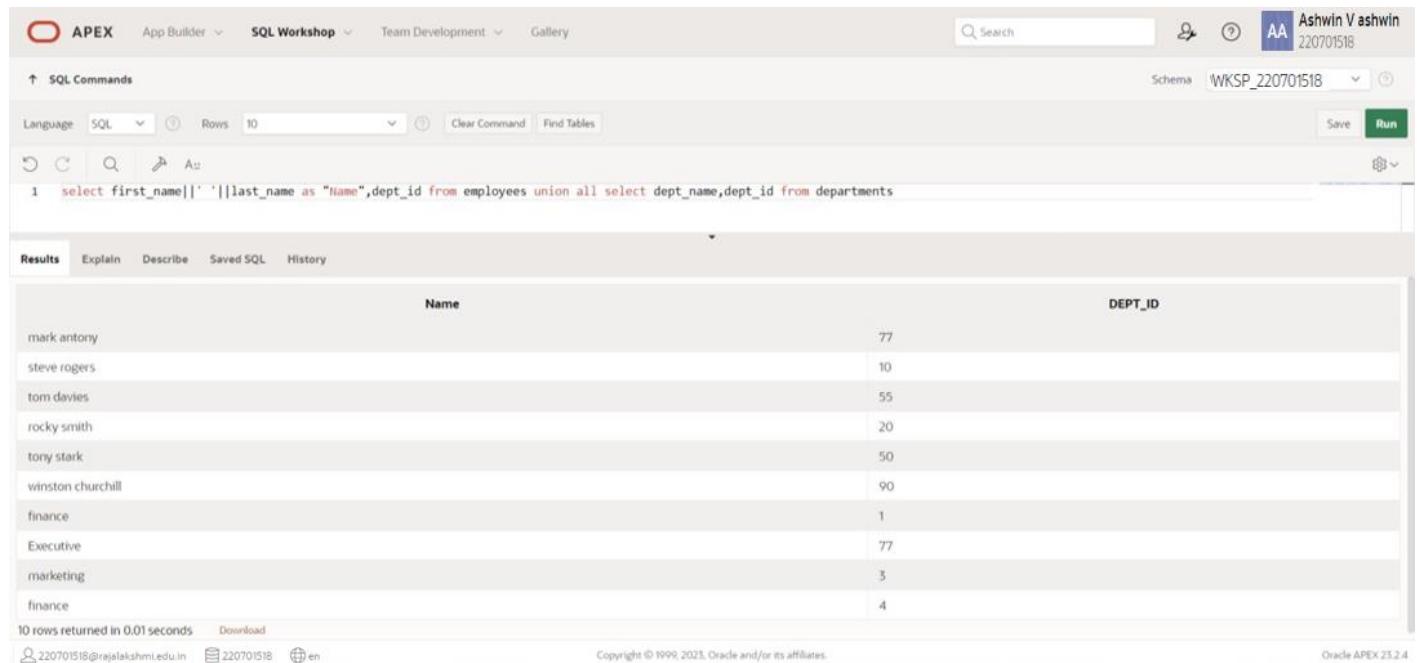
2 rows returned in 0.01 seconds

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:



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 'Ashwin V Ashwin' and a schema dropdown set to 'WKSP_220701518'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below it, a command window contains the following SQL code:

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

Below the command window is a results grid. The columns are labeled 'Name' and 'DEPT_ID'. The data returned is:

Name	DEPT_ID
mark antony	77
steve rogers	10
tom davies	55
rocky smith	20
tony stark	50
winston churchill	90
finance	1
Executive	77
marketing	3
finance	4

At the bottom of the results grid, it says '10 rows returned in 0.01 seconds'. The footer of the page includes links for 'Download', '220701518@rajalakshmi.edu.in', '220701518', and 'en'. It also features a copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and a version note 'Oracle APEX 23.2.4'.

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,
dept_id FROM employees;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' 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,
3       dept_id
  FROM employees;
```

Below the code, the 'Results' tab is active, showing the message "View created." and a execution time of "0.05 seconds". The bottom status bar includes the URL "220701518@rajalakshmi.edu.in", session ID "220701518", and the message "Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4".

2.) Display the contents of the EMPLOYEES_VU view.

QUERY:

```
select * from employees_vu;
```

OUTPUT:

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

```
1 SELECT *FROM EMPLOYEES_VU;
```

Below the code, the 'Results' tab is active, displaying a table with three columns: EMPLOYEE_ID, EMPLOYEE_LASTNAME, and DEPARTMENT_ID. The data is as follows:

EMPLOYEE_ID	EMPLOYEE_LASTNAME	DEPARTMENT_ID
5	Ropebur	80
65	csk	50
1	patel	80
45	RAAVI	80
3	drexler	80

At the bottom, it says "5 rows returned in 0.02 seconds" and "Download". The bottom status bar includes the URL "220701518@rajalakshmi.edu.in", session ID "220701518", and the message "Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4".

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

QUERY:

```
SELECT view_name, text FROM user_views;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Ashwin V ashwin' (220701518), and a schema dropdown set to 'WKSP_220701518'. The main area is titled 'SQL Commands' with a 'Run' button. Below it, a code editor contains the following SQL query:

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

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

VIEW_NAME	TEXT
EMPLOYEES_VU	SELECT EMPLOYEE_ID,EMPLOYEE_LASTNAME ,department_id FROM MY_EMPLOYEE

Below the table, it says '1 rows returned in 0.04 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for '220701518@rajalakshmi.edu.in', '220701518', 'en', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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

QUERY:

```
SELECT employee, dept_id FROM employees_vu;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar and user profile are identical. The main area is titled 'SQL Commands' with a 'Run' button. Below it, a code editor contains the following SQL query:

```
1  SELECT employee_lastname, department_id FROM employees_vu;
2
```

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

EMPLOYEE_LASTNAME	DEPARTMENT_ID
Ropebur	80
csk	50
patel	80
RAAVI	80
drexler	80

Below the table, it says '5 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for '220701518@rajalakshmi.edu.in', '220701518', 'en', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (which is selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'Ashwin V ashwin' and a search bar. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701518'. Below the schema is a toolbar with icons for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', 'Find Tables', 'Save', and 'Run'. The code editor contains the following SQL command:

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

Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active. The output shows the message 'View created.' and a execution time of '0.14 seconds'. At the bottom, it shows the user's email '220701518@rajalakshmi.edu.in', the session ID '220701518', and the page language 'en'. Copyright information and the APEX version 'Oracle APEX 23.2.4' are also present.

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

QUERY:

```
Describe dept50;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one but with the 'Describe' tab selected in the results panel. The top navigation bar and schema selection are identical. The code editor contains the 'Describe' command:

```
1 Describe dept50;
2 |
```

The results panel shows the structure of the 'DEPT50' view. It has two columns: 'EMPNO' and 'EMPLOYEE_LASTNAME'. The 'EMPNO' column is of type NUMBER with a precision of 4 and scale of 0. It is not a primary key, nullable, and has no default value. The 'EMPLOYEE_LASTNAME' column is of type VARCHAR2 with a length of 25, is nullable, and has a check constraint marked with a checkmark. The results panel also includes tabs for 'Object Type' (set to 'VIEW'), 'Object' (set to 'DEPT50'), and 'Comment'.

At the bottom, it shows the user's email '220701518@rajalakshmi.edu.in', the session ID '220701518', and the page language 'en'. Copyright information and the APEX version 'Oracle APEX 23.2.4' are also present.

7.) Attempt to reassign Matos to department 80

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Ashwin V ashwin' (ID 220701518), and a 'Run' button. The main workspace is titled 'SQL Commands' and contains the following SQL code:

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

Below the code, the 'Results' tab is selected, showing the output: '0 row(s) updated.' and '0.04 seconds'. The bottom of the screen displays copyright information for Oracle and the APEX version.

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, identical to the previous one but with a different query. The top navigation bar and user information are the same. The main workspace contains the following SQL code:

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

Below the code, the 'Results' tab is selected, showing '0 row(s) updated.' and '0.04 seconds'. The bottom of the screen displays copyright information for Oracle and the APEX version.

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.

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

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

```
SELECT * FROM animals;
```

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

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

a. ON DELETE CASCADE

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

b. ON DELETE SET NULL

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

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

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

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

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

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

```
CREATE OR REPLACE VIEW view_d_songs AS
```

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

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

```
CREATE OR REPLACE VIEW view_d_events_pkgs AS
```

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

6. It is company policy that only upper-level management be allowed access to individual employee salaries.

The department managers, however, need to know the minimum, maximum, and average salaries, grouped by department. Use the Oracle database to prepare a view that displays the needed information for department managers.

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

DML Operations and Views

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

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

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

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

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

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

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

```
SELECT * FROM view_copy_d_songs;
```

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

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

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

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

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

```
SELECT * FROM read_copy_d_cds;
```

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

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

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

```
CREATE OR REPLACE VIEW read_copy_d_cds AS
SELECT *
FROM copy_d_cds
WHERE year = '2000'
WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds;
```

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

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

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

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

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

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

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

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

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

DELETE,INSERT,MODIFY restricted if it contains:

Group functions
GROUP BY CLAUSE
DISTINCT
pseudocolumn ROWNUM Keyword

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

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

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

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

Managing Views

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

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

```
SELECT * FROM view_copy_d_songs;
```

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

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

ORA-00942: table or view does not exist

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

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

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

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

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

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

Indexes and Synonyms

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

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

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

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

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

3. When will an index be created automatically?

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

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

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

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

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

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

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

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

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

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

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

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

CREATE SYNONYM dj_tracks2 FOR d_track_listings;

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

10. Drop the synonym that you created in question

DROP SYNONYM dj_tracks2;

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

RESULT:

OTHER DATABASE OBJECTS

EX_NO:14

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is chosen from a dropdown menu. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'CREATE SEQUENCE dept_id_seq START WITH 200 INCREMENT BY 10 MAXVALUE 1000;'. The 'Run' button is visible on the right. Below the command, the results show the message 'Sequence created.' and a execution time of '0.02 seconds'. The bottom status bar includes the user's email and session ID.

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 with the 'SQL Scripts' tab selected. A summary table is displayed with one row of data. The table has columns: Number, Elapsed, Statement, Feedback, and Rows. The data row shows: 1, 0.02, 'SELECT sequence_name, max_value, increment_by, last_number F', 'Statement processed.', 1. Below the table, three status indicators are shown: 'Statements Processed' (1), 'Successful' (1), and 'With Errors' (0). The bottom status bar includes the user's email and session ID.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'Ashwin V ashwin' and a search bar. The main area displays the results of a script execution. The 'dept' script is listed as complete with 15 rows processed. The summary table shows 1 statement, 0.02 elapsed time, and 1 row inserted. Below the table, three status indicators are shown: 'Statements Processed' (1), 'Successful' (1), and 'With Errors' (0). The bottom of the page includes copyright information for Oracle and the APEX version.

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's a user profile for 'Ashwin V ashwin' and a search bar. The main area displays the results of a command execution. The command entered was 'CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);'. The results section shows the message 'Index created.' and a duration of '0.04 seconds'. The bottom of the page includes copyright information for Oracle and the APEX version.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name, "Ashwin V ashwin", and the schema name, "WKSP_220701518". The main workspace is titled "SQL Commands" and contains the following SQL query:

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

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

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

Below the table, it says "1 rows returned in 0.04 seconds" and provides a "Download" link. The bottom of the page includes standard Oracle copyright information and the text "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;
```

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

RESULT:

PL/SQL

CONTROL STRUCTURES

EX_NO:

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, tabs for 'App Builder', 'SQL Workshop' (which is active), 'Team Development', and 'Gallery' are visible. On the right side of the header, there is a search bar, a user profile for 'Ashwin Vashwin 220701518', and a 'Run' button. The main workspace is a code editor with the following content:

```
1 DECLARE
2   incentive NUMBER(8,2);
3 BEGIN
4   SELECT salary*0.12 INTO incentive
5   FROM employees
6   WHERE emp_id = 110;
7   DBMS_OUTPUT.PUT_LINE('Incentive = ' || TO_CHAR(incentive));
8 END;
9 |
```

Below the code editor, a results panel is shown with the following output:

Results
Incentive = 4800 Statement processed. 0.02 seconds

At the bottom of the page, there are footer links for '220701518@relaxashwin.edu.in', '220701518', and 'en'. The copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also present.

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 pane, the following PL/SQL code is entered:

```
1 DECLARE
2 "WELCOME" varchar2(10) := 'welcome';
3 BEGIN
4 DBMS_Output.Put_Line("Welcome");
5 END;
6 /
7
```

In the Results pane, an error message is displayed in a yellow box:

```
Error at line 4/23: ORA-06550: line 4, column 23:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.IMP_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored
```

Below the error message, the code is shown again:

```
2. "WELCOME" varchar2(10) := 'welcome';
3. BEGIN
4. DBMS_Output.Put_Line("Welcome");
5. END;
6. /
7.
```

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

```
1 DECLARE
2 WELCOME varchar2(10) := 'welcome';
3 BEGIN
4 DBMS_Output.Put_Line("Welcome");
5 END;
6 /
7
```

In the Results pane, an error message is displayed in a yellow box:

```
Error at line 4/23: ORA-06550: line 4, column 23:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.IMP_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored
```

Below the error message, the code is shown again:

```
2. WELCOME varchar2(10) := 'welcome';
3. BEGIN
4. DBMS_Output.Put_Line("Welcome");
5. END;
6. /
7.
```

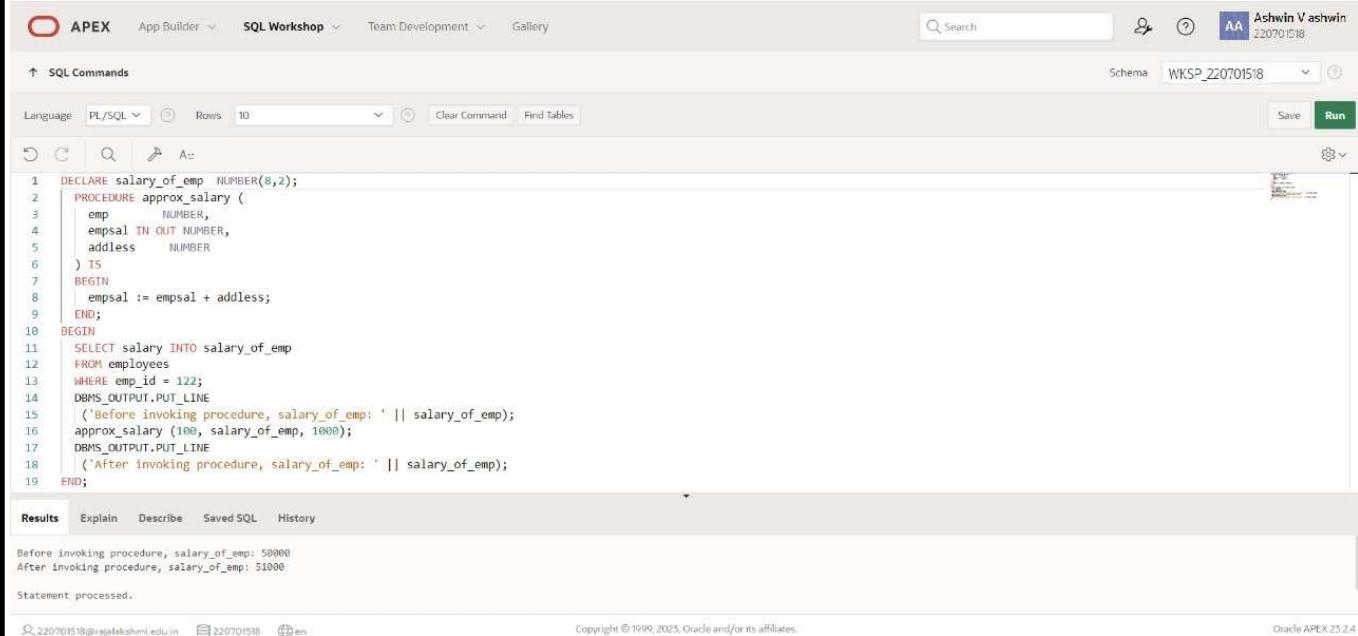
3.) Write a PL/SQL block to adjust the salary of the employee whose ID 122.

QUERY:

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

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there are user profile and session information: "Ashwin V ashwin 220701518". The main workspace is titled "SQL Commands" and has a schema dropdown set to "WKSP_220701518". The language is set to "PL/SQL". The code area contains the PL/SQL block provided above. The results section at the bottom displays the output of the DBMS_OUTPUT.PUT_LINE statements: "Before invoking procedure, salary_of_emp: 50000" and "After invoking procedure, salary_of_emp: 51000". The status bar at the bottom indicates "Statement processed."

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

Results Explain Describe Saved SQL History

Before invoking procedure, salary_of_emp: 50000
After invoking procedure, salary_of_emp: 51000

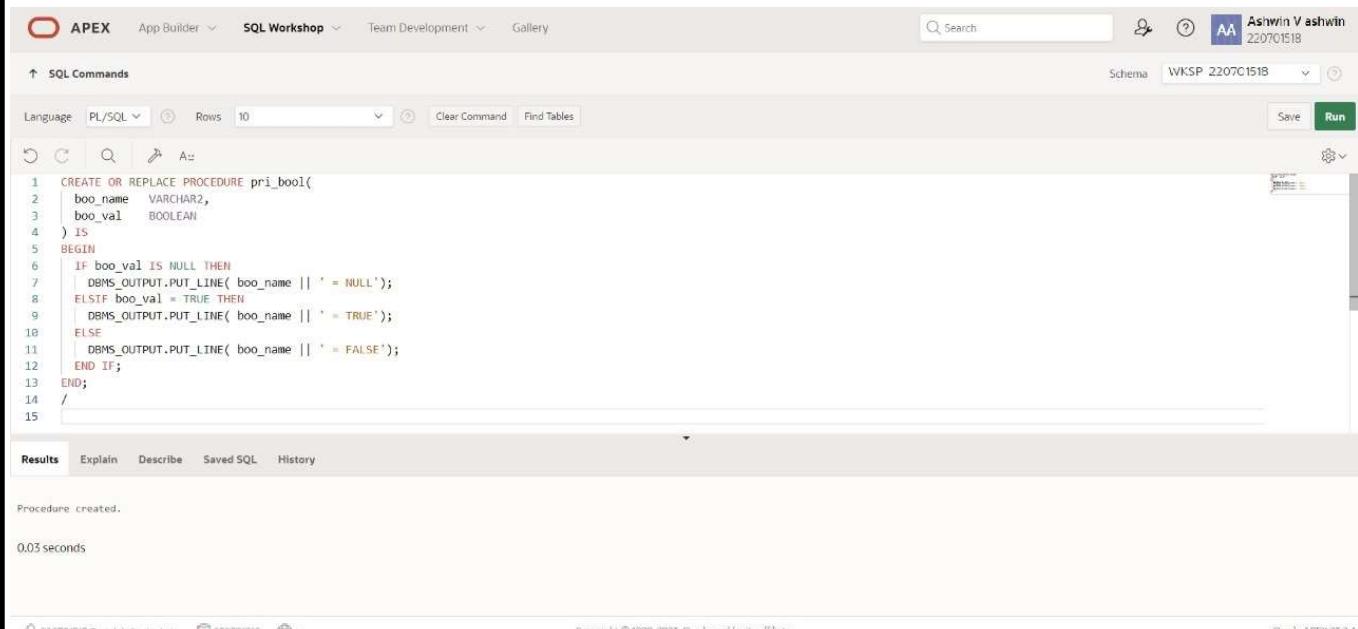
Statement processed.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there is a user profile for 'Ashwin Vashwin' (ID 220701518). The main area is titled 'SQL Commands' and contains the PL/SQL code for the 'pri_bool' procedure. The code is as follows:

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

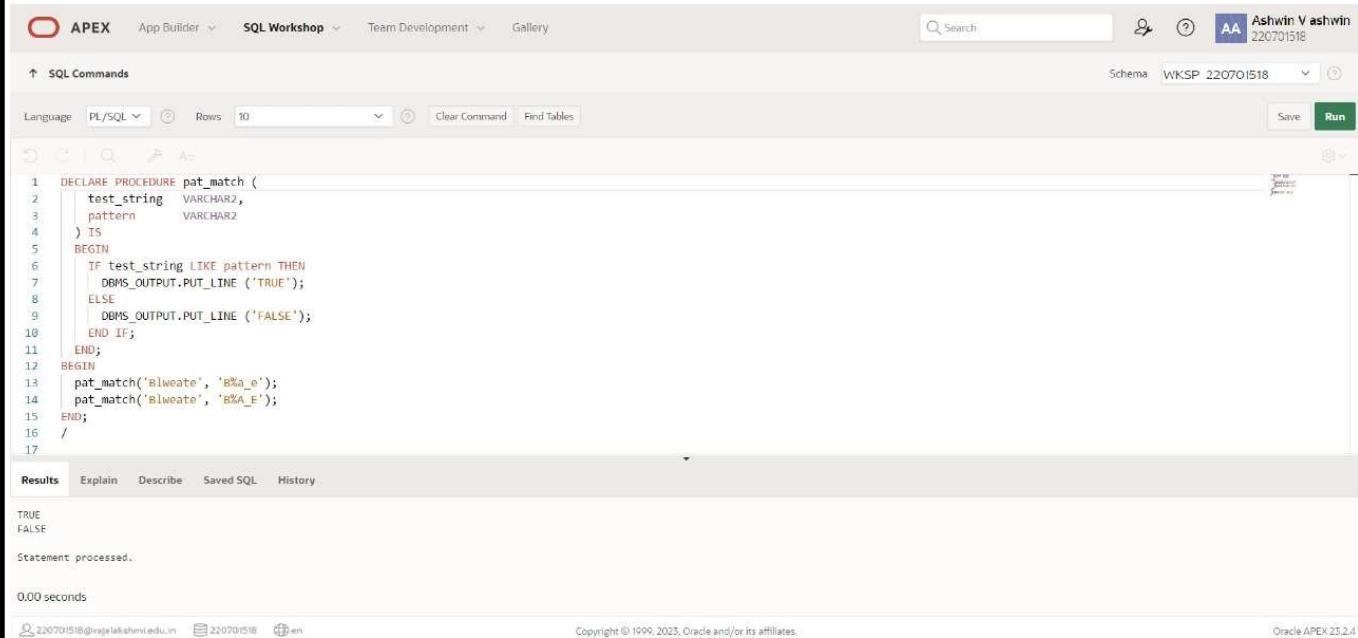
Below the code, the 'Results' tab is active, showing the message 'Procedure created.' and a execution time of '0.03 seconds'. At the bottom, there are footer links for 'Copyright © 1999, 2025, Oracle and/or its affiliates.', 'Oracle APEX 23.2.4', and language selection ('en').

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as Ashwin V ashwin (220701518). The SQL Workshop tab is selected. The query editor window displays the PL/SQL code provided above. The results tab is active, showing the output of the executed code:

```
TRUE
FALSE
Statement processed.
0.00 seconds
```

At the bottom, the footer includes copyright information for Oracle and the APEX version (23.2.4).

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

QUERY:

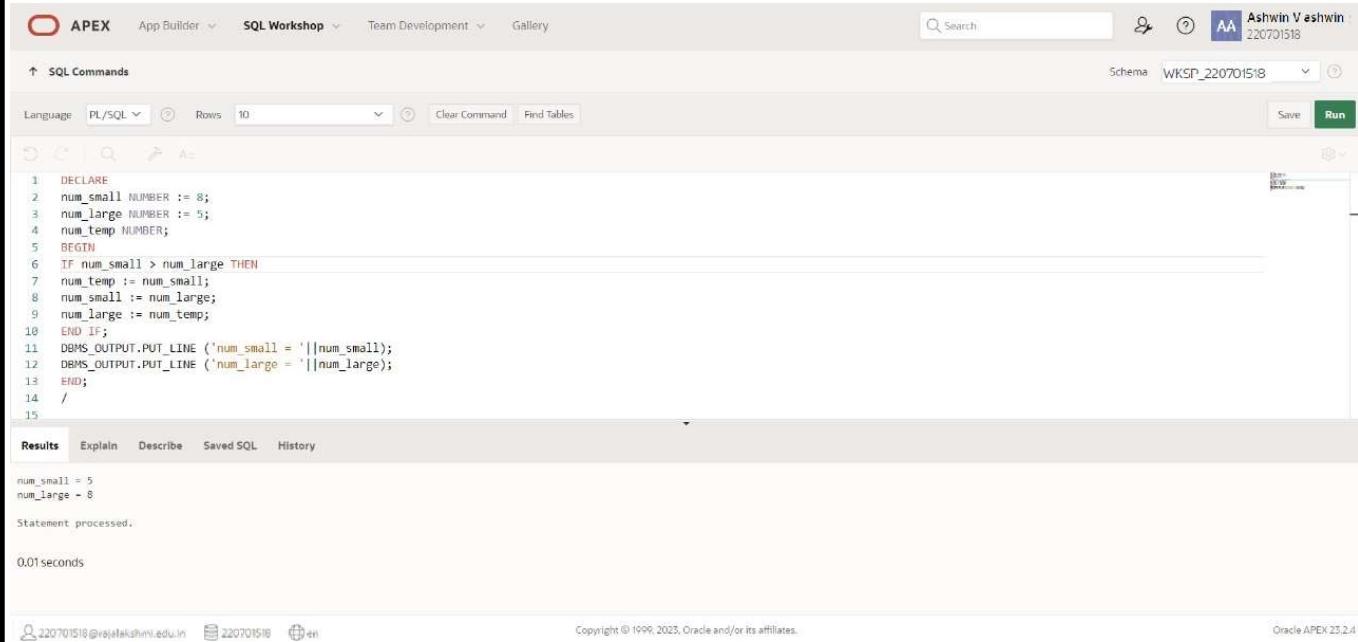
```
DECLARE
num_small NUMBER := 8;
num_large NUMBER := 5;
num_temp NUMBER;
BEGIN

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

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

```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Ashwin V Ashwin' and ID '220701518'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code from the previous section. The code is numbered from 1 to 15. Below the code, the 'Results' tab is selected, showing the output of the executed code. The output displays the values of num_small and num_large after the swap.

```
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;
14 /
15
```

Results

```
num_small = 5
num_large - 8

Statement processed.

0.01seconds
```

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

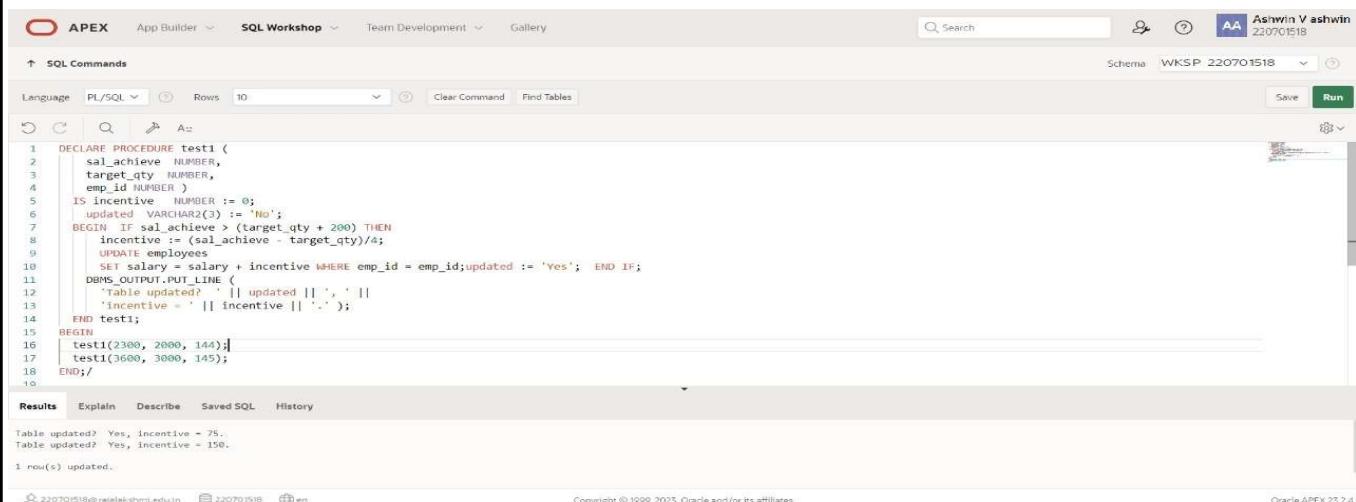
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The top right corner shows the user 'Ashwin V ashwin' and session ID '220701518'. 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 executed procedure calls. The output shows two rows of data, each indicating a table update and calculating an incentive based on the difference between achieved and target values.

```
1 DECLARE PROCEDURE test1 (
2   sal_achieve NUMBER,
3   target_qty NUMBER,
4   emp_id NUMBER
5 )
6 IS incentive NUMBER := 0;
7 updated VARCHAR2(3) := 'No';
8 BEGIN IF sal_achieve > (target_qty + 200) THEN
9   incentive := (sal_achieve - target_qty)/4;
10  UPDATE employees
11    SET salary = salary + incentive WHERE emp_id = emp_id;updated := 'Yes'; END IF;
12  DBMS_OUTPUT.PUT_LINE (
13    'Table updated? ' || updated || ',' ||
14    'incentive = ' || incentive || '.'
15 );
16 BEGIN
17   test1(2300, 2000, 144);
18   test1(3600, 3000, 145);
19 END;/
```

Results

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

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

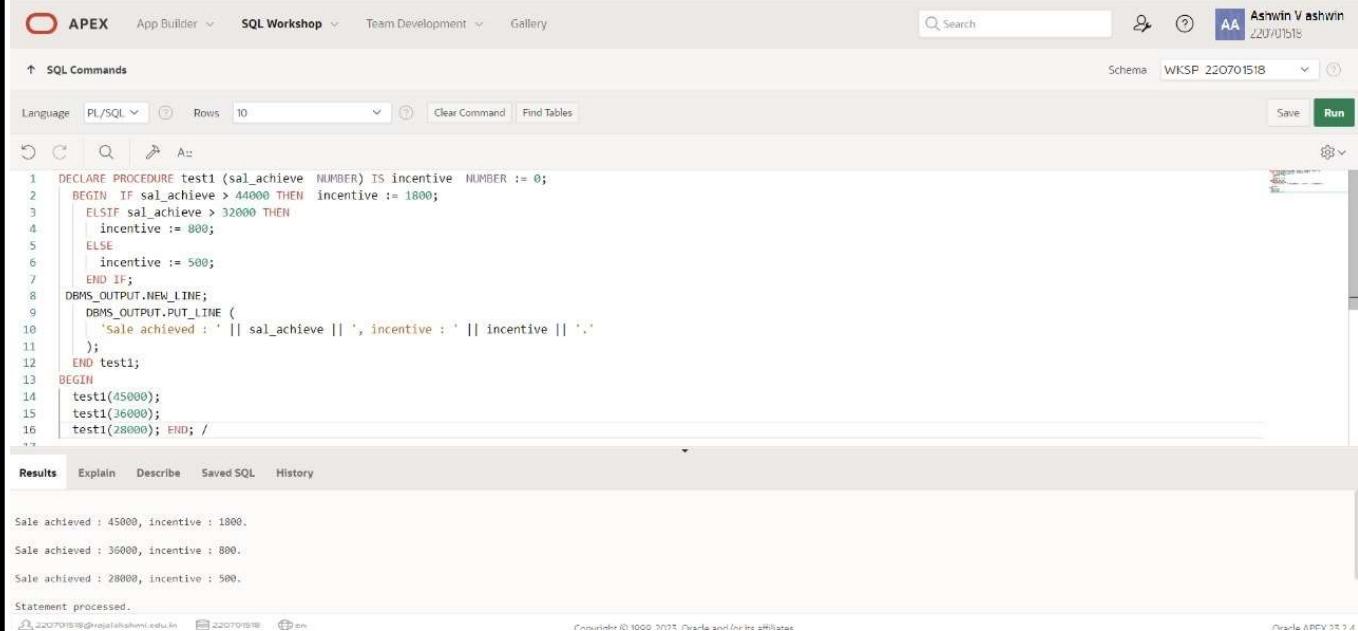
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'Ashwin V ashwin' and a session ID 'WKSP_220701518'. The main area is titled 'SQL Commands' and contains the PL/SQL code from the previous block. Below the code, the 'Results' tab is selected, showing the output of the executed statements:

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

At the bottom, it shows the connection details '220701518@rejasql1.shenzen.edu.in' and the Oracle version 'Oracle APPS 19.9.4'.

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

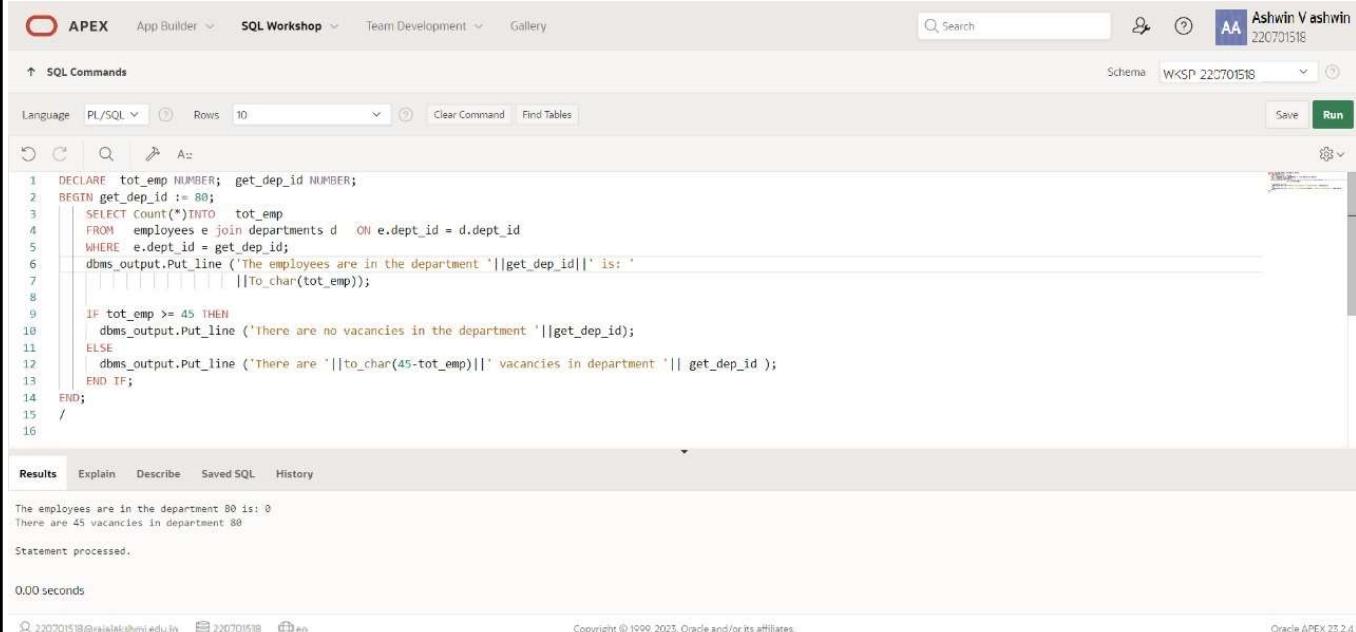
QUERY:

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

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

```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows the user profile of Ashwin V ashwin (220701518). The main area is titled 'SQL Commands' and contains the PL/SQL code from the previous section. The code is executed successfully, and the results pane displays the output: 'The employees are in the department 80 is: 0' and 'There are 45 vacancies in department 80'. The bottom status bar indicates 'Statement processed.' and '0.00 seconds'.

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

```

Results Explain Describe Saved SQL History

The employees are in the department 80 is: 0
There are 45 vacancies in department 80

Statement processed.
0.00 seconds

220701518@rajalakshmi.edu.in 220701518 en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

DECLARE

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

BEGIN

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

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

IF tot_emp >= 45 THEN

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

ELSE

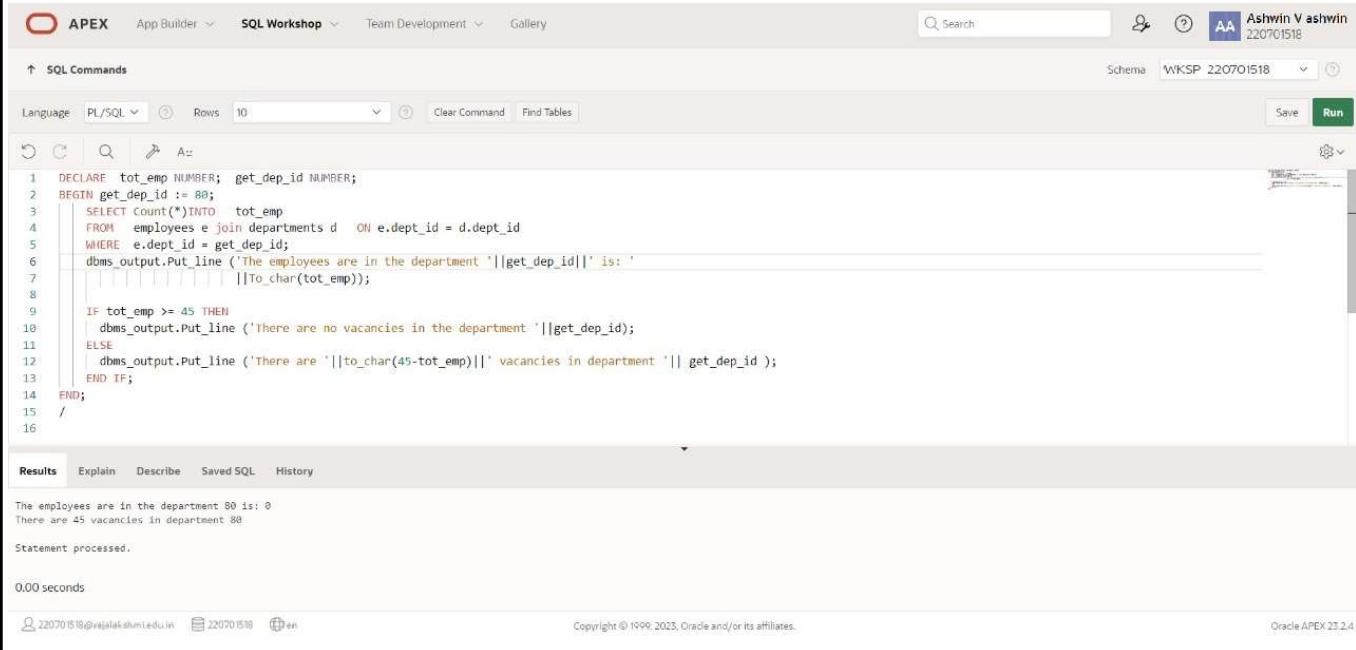
```
dbms_output.Put_line ('There are'||to_char(45-tot_emp)||' vacancies in department'||get_dep_id );
```

END IF;

END;

/

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The code has been entered into the SQL Commands editor. The output pane at the bottom shows the results of the execution, which include the total employee count (0) and the message indicating there are 45 vacancies in department 88.

```
1  DECLARE tot_emp NUMBER; get_dep_id NUMBER;
2  BEGIN get_dep_id := 88;
3  SELECT Count(*)INTO tot_emp
4  FROM employees e join departments d ON e.dept_id = d.dept_id
5  WHERE e.dept_id = get_dep_id;
6  dbms_output.Put_line ('The employees are in the department'||get_dep_id||' is:'
7  ||To_char(tot_emp));
8
9  IF tot_emp >= 45 THEN
10  dbms_output.Put_line ('There are no vacancies in the department'||get_dep_id);
11  ELSE
12  dbms_output.Put_line ('There are'||to_char(45-tot_emp)||' vacancies in department'||get_dep_id );
13  END IF;
14 END;
15 /
16
```

The employees are in the department 88 is: 0
There are 45 vacancies in department 88
Statement processed.
0.00 seconds

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

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

QUERY:

DECLARE

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

CURSOR c_employees IS

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

BEGIN

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

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

```
OPEN c_employees;
```

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

```
WHILE c_employees%FOUND LOOP
```

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

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

```
END LOOP;
```

```
CLOSE c_employees;
```

```
END;
```

```
/
```

OUTPUT:

```
APEX App Builder SQL Workshop Team Development Gallery Search Schema WwKSP 220701518
PL/SQL Rows 10 Save Run
SQL Commands
Language: PL/SQL Rows: 10 Clear Command Find Tables
1 DECLARE v_emp_id employees.emp_id%TYPE; v_full_name employees.first_name%TYPE; v_job_id employees.job_id%TYPE; v_hire_date employees.hire_date%TYPE;
2 v_salary employees.salary%TYPE;
3 CURSOR c_employees IS SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary FROM employees;
4 BEGIN
5 DBMS_OUTPUT.PUT_LINE('Employee ID | Full Name | Job Title | Hire Date | Salary');
6 DBMS_OUTPUT.PUT_LINE('-----');
7 OPEN c_employees;
8 FETCH c_employees INTO v_emp_id, v_full_name, v_job_id, v_hire_date, v_salary;
9 WHILE c_employees%FOUND LOOP
10 DBMS_OUTPUT.PUT_LINE(v_emp_id || ' ' || v_full_name || ' ' || v_job_id || ' ' || v_hire_date || ' ' || v_salary);
11 FETCH c_employees INTO v_emp_id, v_full_name, v_job_id, v_hire_date, v_salary; END LOOP; CLOSE c_employees; END;
12 
```

Results Explain Describe Saved SQL History

Employee ID	Full Name	Job Title	Hire Date	Salary
101	mark roy st_clerk	02/03/1998	18900	
110	steve davies 10	02/05/1999	40900	
122	tom janu st_charles	02/05/2000	50900	
106	rocky smith 55	03/25/1996	25900	
102	tony stark 88	05/01/1994	28900	
1212	winston churchill 22	01/01/2001	50900	

Statement processed.

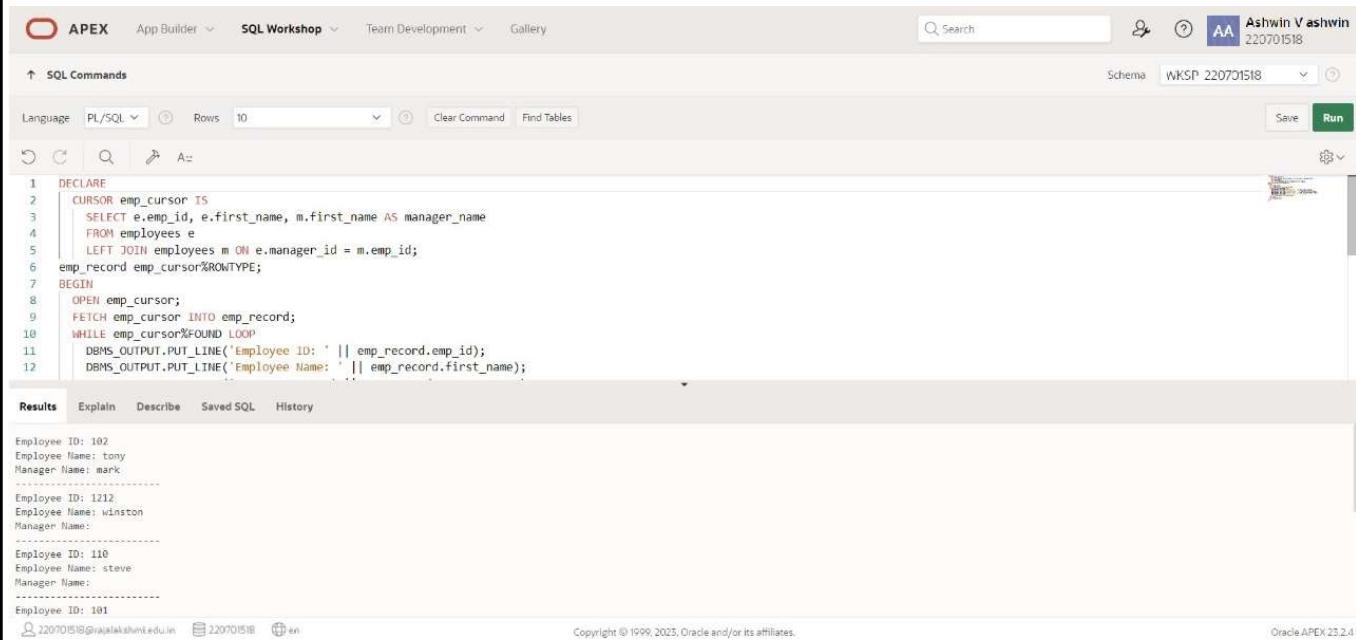
Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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 top right corner shows the user's name, Ashwin V Ashwin, and session ID, WKSP 220701518. The main workspace is titled "SQL Commands". The code area contains the PL/SQL block provided above. The "Results" tab is selected, displaying the output of the program. The output shows four records from the employees table, each with their employee ID, first name, and manager's first name. The output is formatted with a header and a separator line between records.

Employee ID	Employee Name	Manager Name
102	tony	mark
1212	winston	
110	steve	
101		

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile (AA, Ashwin V ashwin), and a timestamp (22/07/2018). The main workspace is titled "SQL Commands". It features a toolbar with Language (PL/SQL), Rows (10), Clear Command, Find Tables, Save, and Run buttons. Below the toolbar is a code editor with the following PL/SQL script:

```
1 DECLARE
2   CURSOR job_cursor IS
3     SELECT e.job_id, j.low_sal
4       FROM job_grades j,employees e;
5   job_record job_cursor%ROWTYPE;
6 BEGIN
7   OPEN job_cursor;
8   FETCH job_cursor INTO job_record;
9   WHILE job_cursor%FOUND LOOP
```

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

```
Job ID: st_clerk
Minimum Salary: 10000
-----
Job ID: 10
Minimum Salary: 10000
-----
Job ID: st_charles
Minimum Salary: 10000
-----
Job ID: 55
Minimum Salary: 10000
-----
Job ID: 88
Minimum Salary: 10000
-----
Job ID: 22
Minimum Salary: 10000
```

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

QUERY:

DECLARE

```
CURSOR employees_cur IS
  SELECT employee_id, last_name, job_id, start_date
  FROM employees NATURAL JOIN job_history;
  emp_start_date DATE;

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

/

OUTPUT:



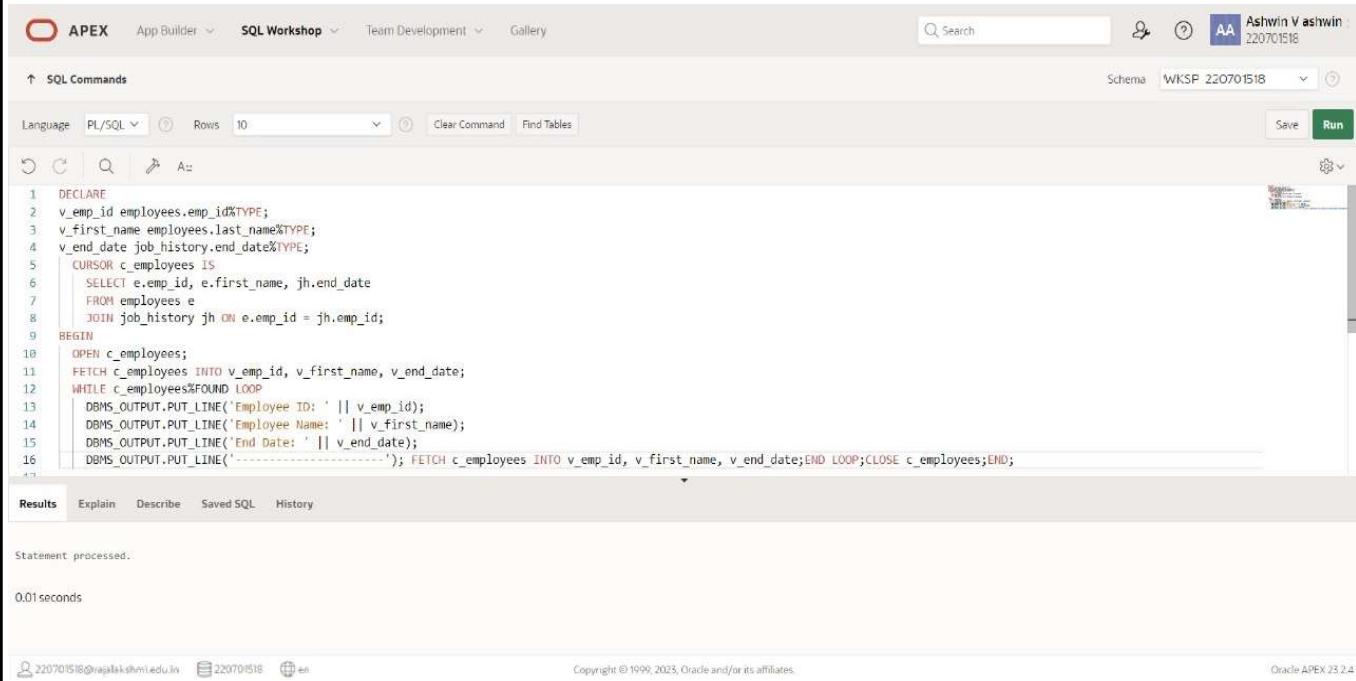
The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Ashwin V ashwin' and session information '220701518'. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'PL/SQL'), 'Rows' (set to 10), 'Clear Command', 'Find Tables', 'Schema' (set to 'WKSP 220701518'), 'Save', and 'Run'. The code area contains the PL/SQL block from above. Below the code, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The results section displays the output of the program, which consists of a single row with columns 'Employee ID', 'Last Name', 'Job Id', and 'Start Date'. The 'Start Date' column shows the value '-----'. At the bottom, it says 'Statement processed.' and '0.02 seconds'.

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:

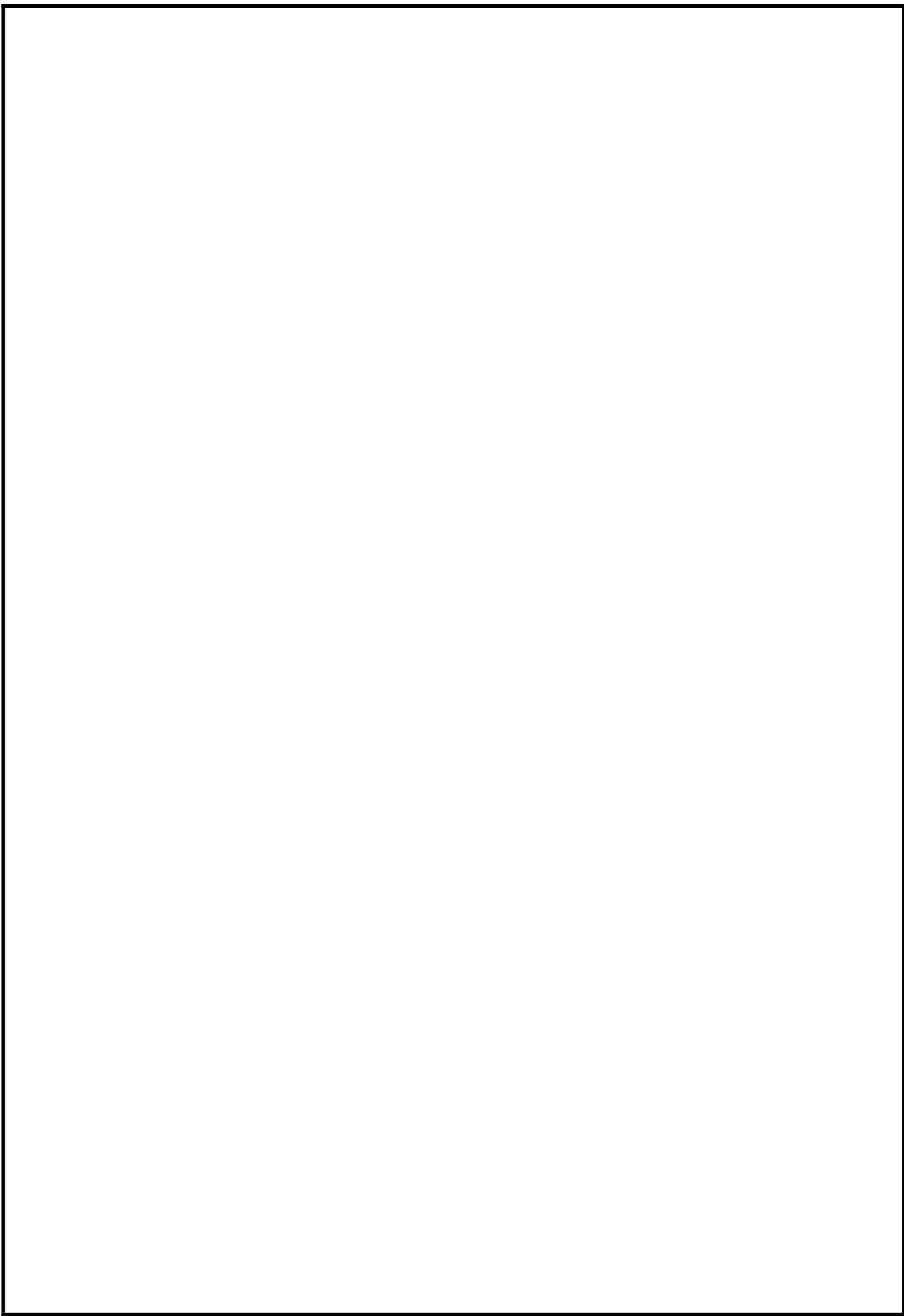


The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Ashwin V ashwin' and session ID '220701518'. The main area is titled 'SQL Commands' with a 'PL/SQL' language dropdown. The code area contains the PL/SQL block from above. The bottom section shows the results of the execution: 'Statement processed.' and '0.01 seconds'. The footer includes copyright information for Oracle and the APEX version '23.2.4'.

```
1  DECLARE
2    v_emp_id employees.emp_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.emp_id, e.first_name, jh.end_date
7        FROM employees e
8       JOIN job_history jh ON e.emp_id = jh.emp_id;
9    BEGIN
10      OPEN c_employees;
11      FETCH c_employees INTO v_emp_id, v_first_name, v_end_date;
12      WHILE c_employees%FOUND LOOP
13        DBMS_OUTPUT.PUT_LINE('Employee ID: ' || v_emp_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('-----'); FETCH c_employees INTO v_emp_id, v_first_name, v_end_date;END LOOP;CLOSE c_employees;END;
```

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

RESULT:



PROCEDURES AND FUNCTIONS

EX_NO: 17

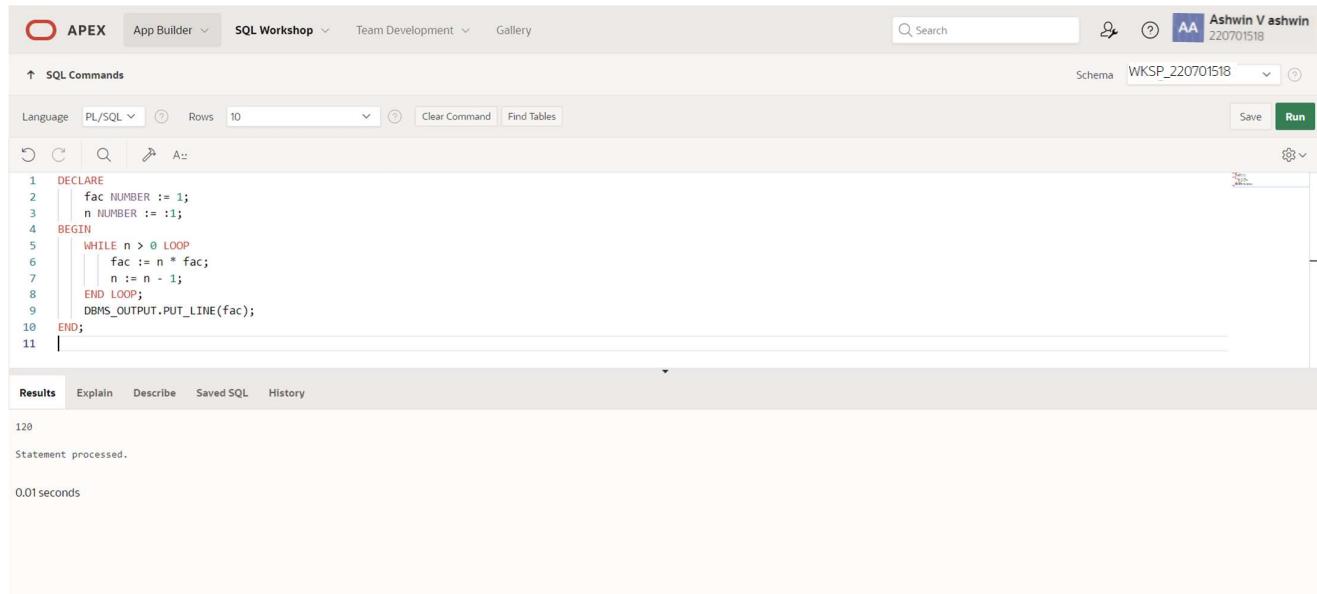
DATE:

1.) Factorial of a number using function.

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays a PL/SQL block for calculating factorial. The code is as follows:

```
1 DECLARE
2     |   fac NUMBER := 1;
3     |   n NUMBER := :1;
4 BEGIN
5     WHILE n > 0 LOOP
6         |   fac := n * fac;
7         |   n := n - 1;
8     END LOOP;
9     DBMS_OUTPUT.PUT_LINE(fac);
10 END;
11 |
```

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

```
128
Statement processed.

0.01seconds
```

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

QUERY:

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

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

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

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

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays a PL/SQL code editor with the following procedure definition:

```
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;
```

Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected. The output pane shows the message 'Procedure created.' and a execution time of '0.04 seconds'.

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

RESULT:

TRIGGER

EX_NO: 18

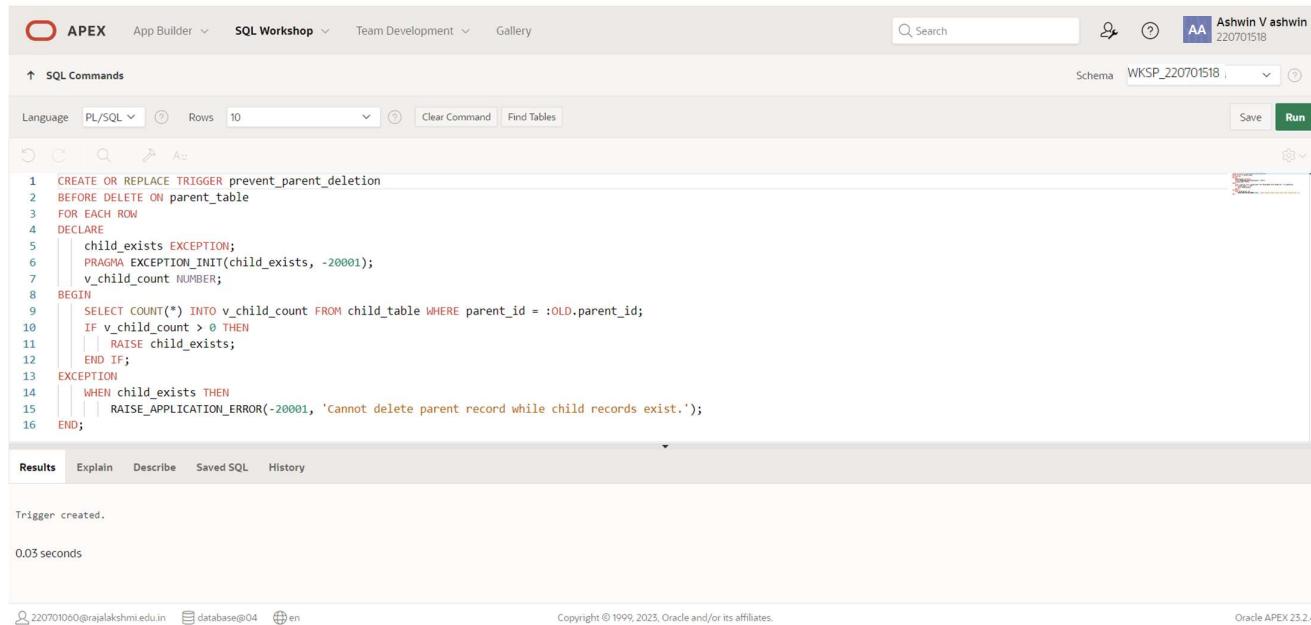
DATE:

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 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is active. The schema dropdown shows 'WKSP_220701518'. The main area displays the PL/SQL code for the trigger. The bottom navigation bar has tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The results section shows the message 'Trigger created.' and a execution time of '0.03 seconds'. The footer includes copyright information for Oracle and the APEX version.

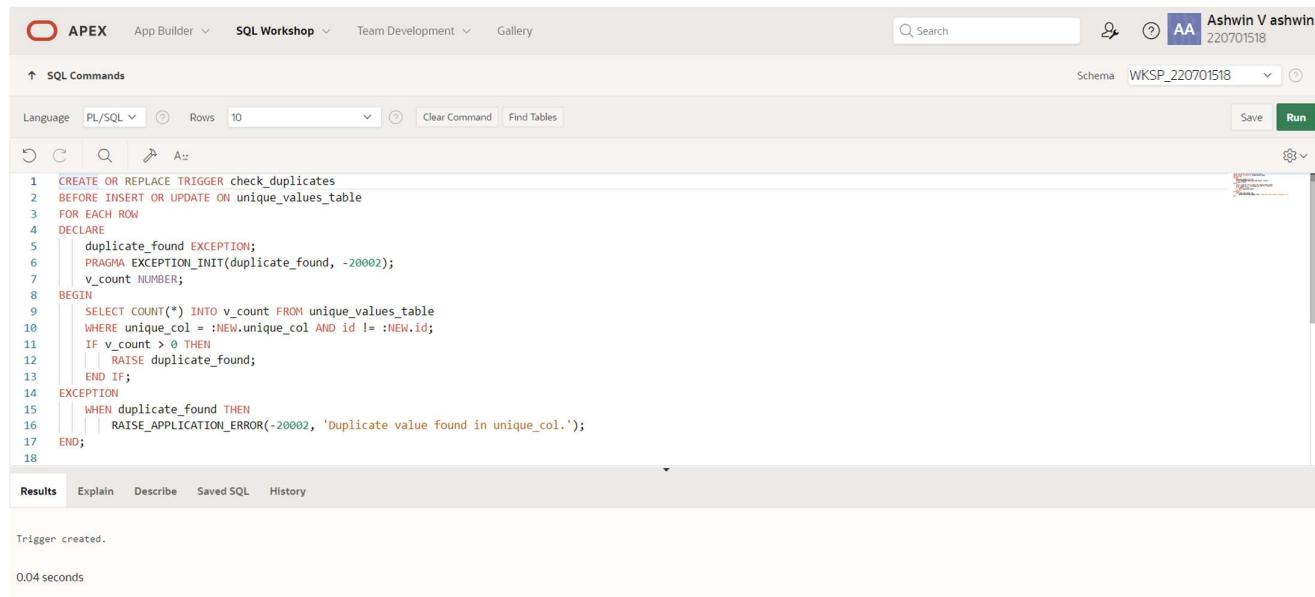
```
1 CREATE OR REPLACE TRIGGER prevent_parent_deletion
2 BEFORE DELETE ON parent_table
3 FOR EACH ROW
4 DECLARE
5     child_exists EXCEPTION;
6     PRAGMA EXCEPTION_INIT(child_exists, -20001);
7     v_child_count NUMBER;
8 BEGIN
9     SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id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

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The main area displays the PL/SQL code for the trigger. The code is syntax-highlighted, with 'CREATE OR REPLACE TRIGGER' in blue, 'unique_values_table' in green, and 'unique_col' in red. The trigger body includes declarations for 'duplicate_found' exception and variable 'v_count', a 'BEGIN' block with a select statement and an if condition, and an 'EXCEPTION' block with a raise application error. At the bottom of the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The results pane shows the message 'Trigger created.' and a execution time of '0.04 seconds'.

```
1 CREATE OR REPLACE TRIGGER check_duplicates
2 BEFORE INSERT OR UPDATE ON unique_values_table
3 FOR EACH ROW
4 DECLARE
5     duplicate_found EXCEPTION;
6     PRAGMA EXCEPTION_INIT(duplicate_found, -20002);
7     v_count NUMBER;
8 BEGIN
9     SELECT COUNT(*) INTO v_count FROM unique_values_table
10    WHERE unique_col = :NEW.unique_col AND id != :NEW.id;
11    IF v_count > 0 THEN
12        RAISE duplicate_found;
13    END IF;
14 EXCEPTION
15    WHEN duplicate_found THEN
16        RAISE_APPLICATION_ERROR(-20002, 'Duplicate value found in unique_col.');
17 END;
18
```

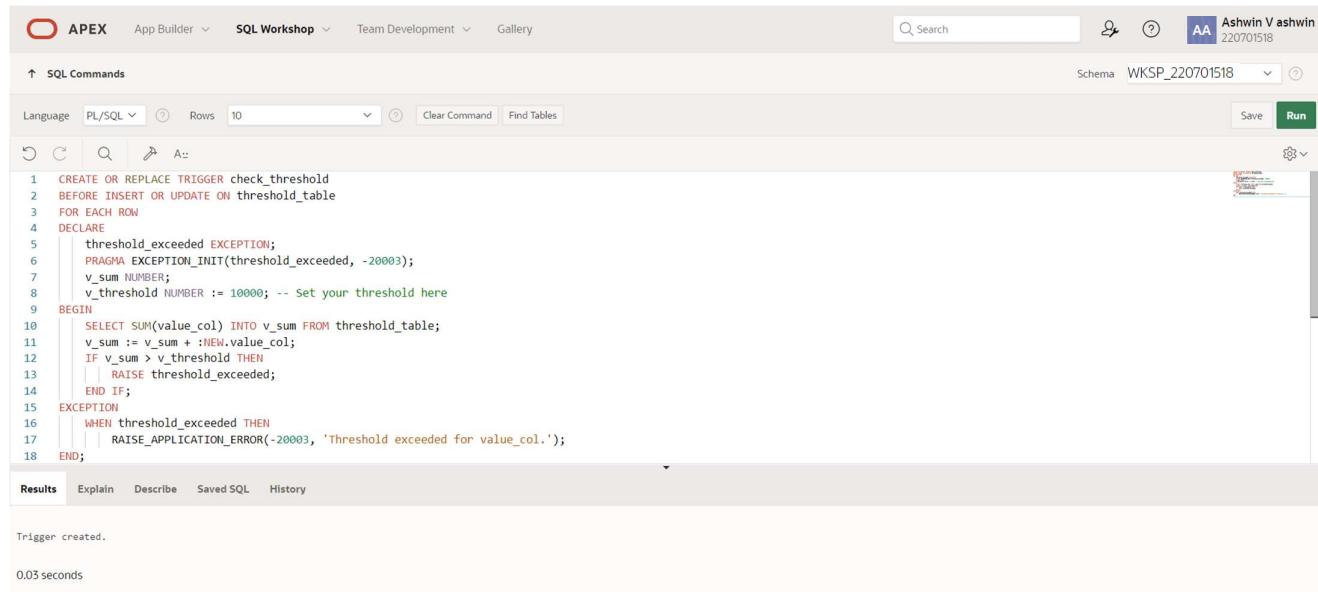
Trigger created.
0.04 seconds

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays the PL/SQL code for the 'check_threshold' trigger. The code is identical to the one provided in the question. Below the code, the 'Results' tab is active, showing the message 'Trigger created.' and a execution time of '0.03 seconds'. The schema is set to 'WKSP_220701518'.

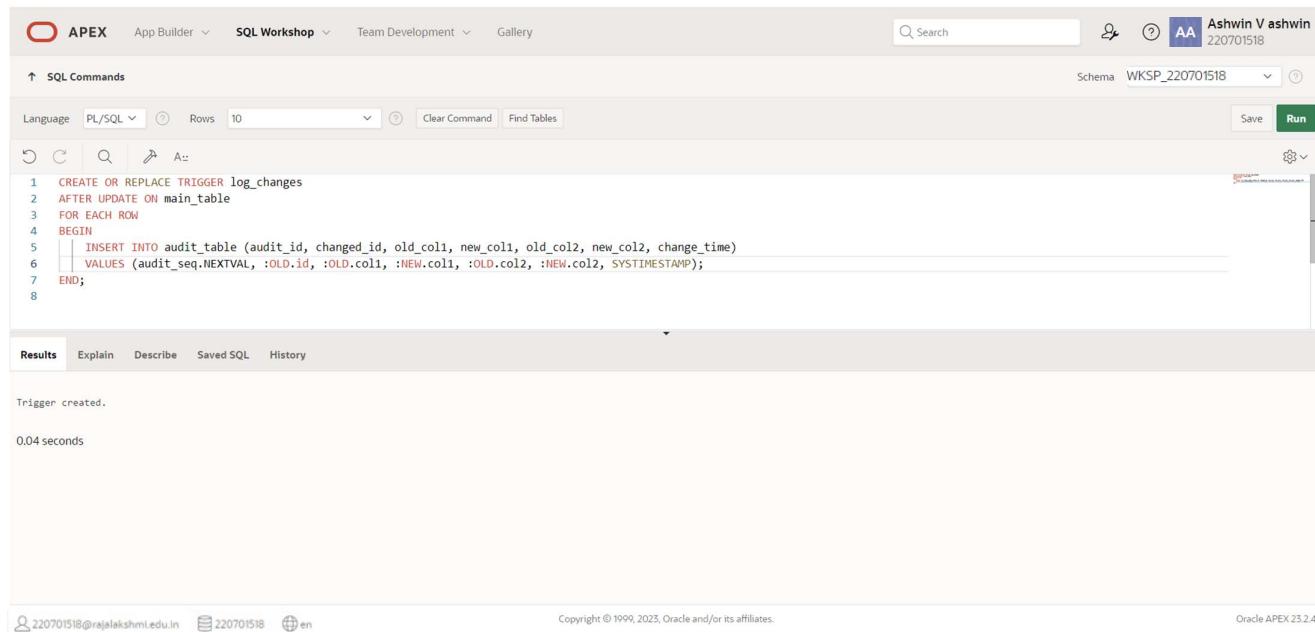
```
1 CREATE OR REPLACE TRIGGER check_threshold
2 BEFORE INSERT OR UPDATE ON threshold_table
3 FOR EACH ROW
4 DECLARE
5     threshold_exceeded EXCEPTION;
6     PRAGMA EXCEPTION_INIT(threshold_exceeded, -20003);
7     v_sum NUMBER;
8     v_threshold NUMBER := 10000; -- Set your threshold here
9 BEGIN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

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'Ashwin V ashwin' (ID 220701518). The main area is titled 'SQL Commands' with a 'PL/SQL' language dropdown. The schema is set to 'WKSP_220701518'. The code editor contains the PL/SQL trigger definition provided in the question. Below the code, the 'Results' tab is selected, showing the output: 'Trigger created.' and '0.04 seconds'. The bottom footer includes user information (220701518@rajalakshmi.edu.in, 220701518, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version 'Oracle APEX 23.2.4'.

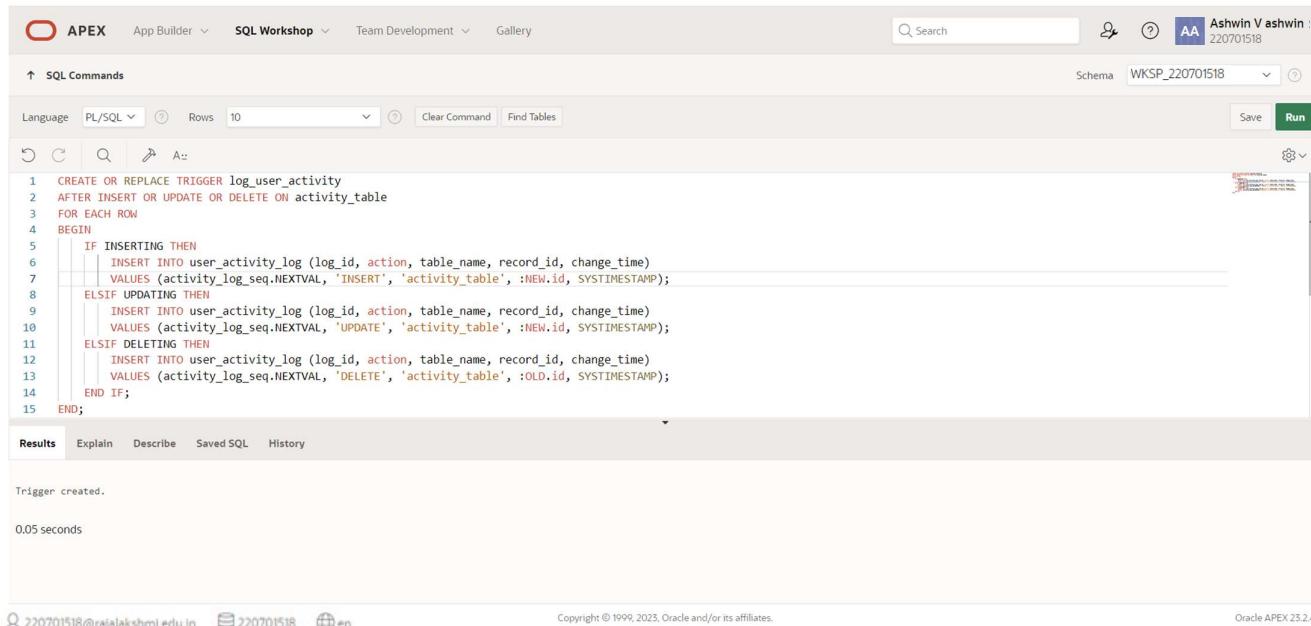
```
1 CREATE OR REPLACE TRIGGER log_changes
2 AFTER UPDATE ON main_table
3 FOR EACH ROW
4 BEGIN
5     INSERT INTO audit_table (audit_id, changed_id, old_col1, new_col1, old_col2, new_col2,
change_time)
6     VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2, :NEW.col2,
SYSTIMESTAMP);
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 APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The top right shows the user profile 'Ashwin V ashwin' and the schema 'WKSP_220701518'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is syntax-highlighted, showing keywords in blue, identifiers in black, and comments in green. The code itself is identical to the one provided in the question. Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The results section shows the message 'Trigger created.' and a execution time of '0.05 seconds'. At the bottom, there are footer links for user information, session details, copyright notice, and the version 'Oracle APEX 23.2.4'.

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

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' (selected), 'Team Development', and 'Gallery'. The right side shows the user 'Ashwin V ashwin' and the schema 'WKSP_220701518'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'PL/SQL'. Below the dropdown are buttons for 'Save' and 'Run'. The code area contains the PL/SQL trigger definition. The bottom section has tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab displays the message 'Trigger created.' and a execution time of '0.04 seconds'. The footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

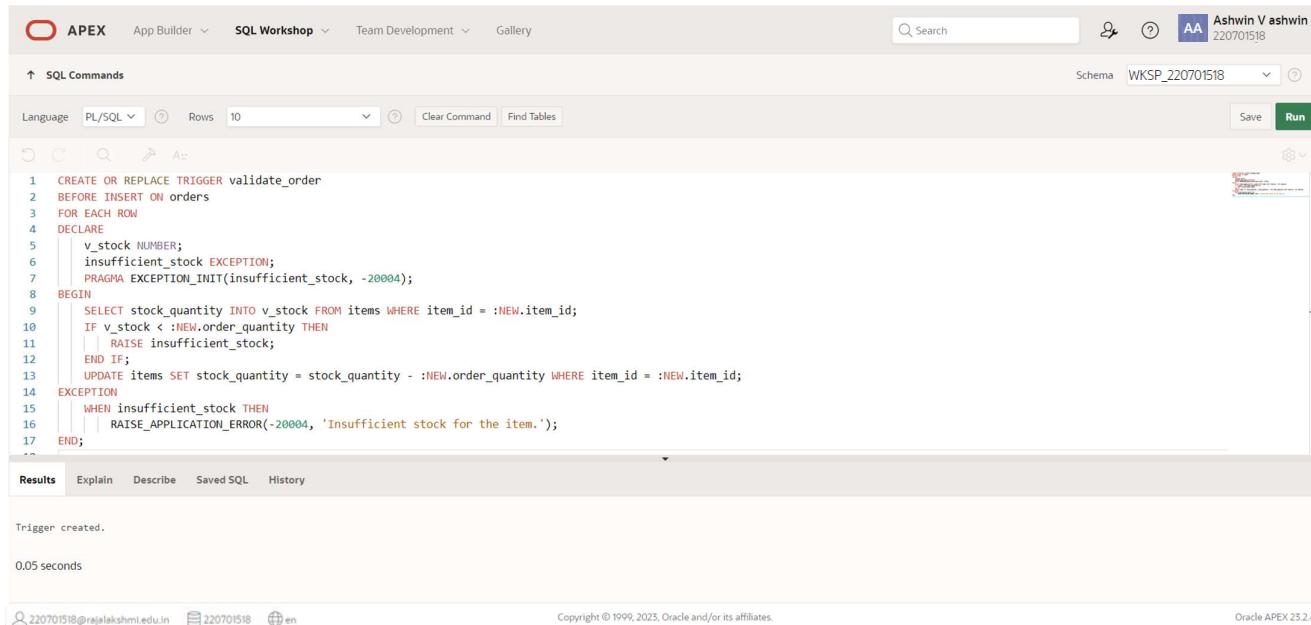
```
1 CREATE OR REPLACE TRIGGER update_running_total
2 BEFORE INSERT ON running_total_table
3 FOR EACH ROW
4 DECLARE
5     v_total NUMBER;
6 BEGIN
7     SELECT NVL(SUM(amount), 0) INTO v_total FROM running_total_table;
8     :NEW.running_total := v_total + :NEW.amount;
9 END;
```

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 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The user is logged in as 'Ashwin V ashwin' with ID '220701518'. The schema is set to 'WKSP_220701518'. The main area displays the PL/SQL code for the 'validate_order' trigger. The code is identical to the one provided in the question, including the declaration of 'v_stock' as a NUMBER, the definition of the 'insufficient_stock' exception, the BEGIN block with a SELECT statement to fetch the current stock quantity, an IF condition to check if it's less than the order quantity, an UPDATE statement to decrement the stock quantity, and finally an EXCEPTION block with a WHEN clause for the insufficient_stock exception that raises the application error with message 'Insufficient stock for the item.'. Below the code, the 'Results' tab is selected, showing the message 'Trigger created.' and a execution time of '0.05 seconds'. The bottom footer includes copyright information for Oracle and the APEX version '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;
12    END IF;
13    UPDATE items SET stock_quantity = stock_quantity - :NEW.order_quantity WHERE item_id
14 = :NEW.item_id;
15 EXCEPTION
16    WHEN insufficient_stock THEN
17        RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');
18 END;
```

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:

```
ashwin_518> db.restaurants.insertOne({  
...   "address": {  
...     "building": "1007",  
...     "coord": [ -73.856077, 40.848447 ],  
...     "street": "Morris Park Ave",  
...     "zipcode": "10462"  
...   },  
...   "borough": "Bronx",  
...   "cuisine": "Bakery",  
...   "grades": [  
...     { "date": new Date(1393804800000), "grade": "A", "score": 2 },  
...     { "date": new Date(1378857600000), "grade": "A", "score": 6 },  
...     { "date": new Date(1358985600000), "grade": "A", "score": 10 },  
...     { "date": new Date(1322006400000), "grade": "A", "score": 9 },  
...     { "date": new Date(1299715200000), "grade": "B", "score": 14 }  
...   ],  
...   "name": "Morris Park Bake Shop",  
...   "restaurant_id": "30075445"  
... });  
{  
  acknowledged: true,  
  insertedId: ObjectId('6655ef7d22e26a69a8cdcdf7')  
}  
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ "grades": { $elemMatch: { "grade": "A", "score": 11, "date": ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 })  
ashwin_518> |
```

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:

```
ashwin_518> 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 } )
ashwin_518>
```

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:

```
ashwin_518> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1})
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find( {}, { _id: 0 }).sort( { name: 1 } )
[ {
    address: {
        building: '1007',
        coord: [ -73.856077, 40.848447 ],
        street: 'Morris Park Ave',
        zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
        {
            date: ISODate('2014-03-03T00:00:00.000Z'),
            grade: 'A',
            score: 2
        },
        {
            date: ISODate('2013-09-11T00:00:00.000Z'),
            grade: 'A',
            score: 6
        },
        {
            date: ISODate('2013-01-24T00:00:00.000Z'),
            grade: 'A',
            score: 10
        },
        {
            date: ISODate('2011-11-23T00:00:00.000Z'),
            grade: 'A',
            score: 9
        },
        {
            date: ISODate('2011-03-10T00:00:00.000Z'),
            grade: 'B',
            score: 14
        }
    ],
    ...
}
```

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:

```
{
  address: {
    building: '1007',
    coord: [ -73.856077, 40.848447 ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: ISODate('2014-03-03T00:00:00.000Z'),
      grade: 'A',
      score: 2
    },
    {
      date: ISODate('2013-09-11T00:00:00.000Z'),
      grade: 'A',
      score: 6
    },
    {
      date: ISODate('2013-01-24T00:00:00.000Z'),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate('2011-11-23T00:00:00.000Z'),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate('2011-03-10T00:00:00.000Z'),
      grade: 'B',
      score: 14
    }
  ],
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
}
ashwin_518> |
```

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:

```
{
  address: {
    building: '1007',
    coord: [ -73.856077, 40.848447 ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: ISODate('2014-03-03T00:00:00.000Z'),
      grade: 'A',
      score: 2
    },
    {
      date: ISODate('2013-09-11T00:00:00.000Z'),
      grade: 'A',
      score: 6
    },
    {
      date: ISODate('2013-01-24T00:00:00.000Z'),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate('2011-11-23T00:00:00.000Z'),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate('2011-03-10T00:00:00.000Z'),
      grade: 'B',
      score: 14
    }
  ],
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
}
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })
[ {
    _id: ObjectId('6655ef5022e26a69a8cdcdf6'),
    address: {
        building: '1007',
        coord: [ -73.856077, 40.848447 ],
        street: 'Morris Park Ave',
        zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
        {
            date: ISODate('2014-03-03T00:00:00.000Z'),
            grade: 'A',
            score: 2
        },
        {
            date: ISODate('2013-09-11T00:00:00.000Z'),
            grade: 'A',
            score: 6
        },
        {
            date: ISODate('2013-01-24T00:00:00.000Z'),
            grade: 'A',
            score: 10
        },
        {
            date: ISODate('2011-11-23T00:00:00.000Z'),
            grade: 'A',
            score: 9
        },
        {
            date: ISODate('2011-03-10T00:00:00.000Z'),
            grade: 'B',
            score: 14
        }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
}
```

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:

```
ashwin_518> db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
[ {
    _id: ObjectId('6655f4bb69eceff92bcdcdf6'),
    address: {
        building: '1007',
        coord: [ -73.856077, 40.848447 ],
        street: 'Morris Park Ave',
        zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
        {
            date: ISODate('2014-03-03T00:00:00.000Z'),
            grade: 'A',
            score: 2
        },
        {
            date: ISODate('2013-09-11T00:00:00.000Z'),
            grade: 'A',
            score: 6
        },
        {
            date: ISODate('2013-01-24T00:00:00.000Z'),
            grade: 'A',
            score: 10
        },
        {
            date: ISODate('2011-11-23T00:00:00.000Z'),
            grade: 'A',
            score: 9
        },
        {
            date: ISODate('2011-03-10T00:00:00.000Z'),
            grade: 'B',
            score: 14
        }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
}
ashwin_518> S
```

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:

```
ashwin_518> db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 })
[{
  "_id": ObjectId('6655ef5022e26a69a8cdcdf6'),
  "grades": [
    {
      "date": ISODate('2014-03-03T00:00:00.000Z'),
      "grade": 'A',
      "score": 2
    },
    {
      "date": ISODate('2013-09-11T00:00:00.000Z'),
      "grade": 'A',
      "score": 6
    },
    {
      "date": ISODate('2013-01-24T00:00:00.000Z'),
      "grade": 'A',
      "score": 10
    },
    {
      "date": ISODate('2011-11-23T00:00:00.000Z'),
      "grade": 'A',
      "score": 9
    },
    {
      "date": ISODate('2011-03-10T00:00:00.000Z'),
      "grade": 'B',
      "score": 14
    }
  ],
  "name": 'Morris Park Bake Shop',
  "restaurant_id": '3007545'
},
{
  "_id": ObjectId('6655ef7d22e26a69a8cdcdf7'),
  "grades": [
    {
      "date": ISODate('2014-03-03T00:00:00.000Z'),
      "grade": 'A',
      "score": 2
    },
    {
      "date": ISODate('2013-09-11T00:00:00.000Z'),
      "grade": 'A',
      "score": 6
    },
    {
      "date": ISODate('2013-01-24T00:00:00.000Z'),
      "grade": 'A',
      "score": 10
    },
    {
      "date": ISODate('2011-11-23T00:00:00.000Z'),
      "grade": 'A',
      "score": 9
    },
    {
      "date": ISODate('2011-03-10T00:00:00.000Z'),
      "grade": 'B',
      "score": 14
    }
  ]
}].
```

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:

```
ashwin_518> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
ashwin_518>
```

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:

```
ashwin_518> db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 })
[
  {
    _id: ObjectId('6655ef5022e26a69a8cdcd46'),
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
]
```

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:

```
ashwin_518> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
ashwin_518>
```

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:

```
ashwin_518> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })  
ashwin_518>
```

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:

```
ashwin_518> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
[ {
  _id: ObjectId('6655ef5022e26a69a8cdcdf6'),
  address: {
    building: '1007',
    coord: [ -73.856077, 40.848447 ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: ISODate('2014-03-03T00:00:00.000Z'),
      grade: 'A',
      score: 2
    },
    {
      date: ISODate('2013-09-11T00:00:00.000Z'),
      grade: 'A',
      score: 6
    },
    {
      date: ISODate('2013-01-24T00:00:00.000Z'),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate('2011-11-23T00:00:00.000Z'),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate('2011-03-10T00:00:00.000Z'),
      grade: 'B',
      score: 14
    }
  ]
},
```

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:

```
ashwin_518> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
ashwin_518> |
```

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:

```
ashwin_518> 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"] } })  
ashwin_518> |
```

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:

```
ashwin_518> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
[ {
    _id: ObjectId('6655ef5022e26a69a8cdcdf6'),
    address: {
        building: '1007',
        coord: [ -73.856677, 40.848447 ],
        street: 'Morris Park Ave',
        zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
        {
            date: ISODate('2014-03-03T00:00:00.000Z'),
            grade: 'A',
            score: 2
        },
        {
            date: ISODate('2013-09-11T00:00:00.000Z'),
            grade: 'A',
            score: 6
        },
        {
            date: ISODate('2013-01-24T00:00:00.000Z'),
            grade: 'A',
            score: 10
        },
        {
            date: ISODate('2011-11-23T00:00:00.000Z'),
            grade: 'A',
            score: 9
        },
        {
            date: ISODate('2011-03-10T00:00:00.000Z'),
            grade: 'B',
            score: 14
        }
    ]
},
```

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:

```
ashwin_518> db.movies.find({ year: 1893 })
ashwin_518>
```

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:

```
ashwin_518> db.movies.find({ runtime: { $gt: 120 } })
ashwin_518> |
```

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

QUERY:

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

OUTPUT:

```
ashwin_518> db.movies.find({ genres: 'Short' })
ashwin_518> |
```

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:

```
ashwin_518> db.movies.find({ directors: 'William K.L. Dickson' })  
ashwin_518> |
```

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:

```
ashwin_518> db.movies.find({ countries: 'USA' })  
ashwin_518> |
```

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

QUERY:

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

OUTPUT:

```
ashwin_518> db.movies.find({ rated: 'UNRATED' })  
ashwin_518> |
```

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:

```
ashwin_518> db.movies.find({ 'imdb.votes': { $gt: 1000 } })
```

```
ashwin_518>
```

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:

```
ashwin_518> db.movies.find({ 'imdb.rating': { $gt: 7 } })
```

```
ashwin_518>
```

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:

```
ashwin_518> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
```

```
ashwin_518> |
```

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

QUERY:

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

OUTPUT:

```
ashwin_518> db.movies.find({ 'awards.wins': { $gt: 0 } })  
ashwin_518> |
```

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:

```
ashwin_518> 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 }  
... )  
ashwin_518> |
```

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:

```
ashwin_518> 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 }  
... )  
ashwin_518> |
```

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:

```
ashwin_518> db.movies.find(  
...   { released: ISODate("1893-05-09T00:00:00.000Z") },  
...   { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 }  
... )  
ashwin_518> |
```

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:

```
ashwin_518> db.movies.find(  
...   { title: /scene/i },  
...   { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 }  
... )  
ashwin_518> |
```

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

RESULT: