

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 dep(Dep_ID Number(6)Not Null, Dept_Name Varchar(20), Manager_ID VARCHAR(10), Location_ID Number(10));

OUTPUT:

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

```
1 create table Dep (Dep_ID Number(6)Not Null, Dept_Name Varchar(20), Manager_ID VARCHAR(10), Location_ID Number(10))
```

The Results tab displays the output of the query:

```
Table created.  
0.05 seconds
```

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

Activate Windows
Go to Settings to activate Windows.
Copyright © 1999, 2025, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

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

QUERY:

```
Create table Employeee (Employee_ID NUMBER(6)Not Null,First_Name Varchar(20),Last_Name  
Varchar(25) Not Null,Email Varchar(25) Not Null,PhoneNumber Varchar(20),Hire_Date Date Not Null,Job_ID  
Varchar(10)Not Null,Salary Number(8,2),Commission_pct Number(2,2),Manager_ID  
Number(6),Department_ID Number(4));
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. At the top, the URL is https://apex.oracle.com/pls/apex/r/apex/sql-workshop/sqlcommandprocessor?session=5779725054757. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, the user is logged in as DEVDHARSHAN SR, database@04. The main area is titled 'SQL Commands'. The code editor contains the following SQL command:

```
1 --create Table Dep (Dep_ID Number(4)Not Null,Dept_Name Varchar(20), Manager_ID Varchar(10), Location_ID Number(10))
2 Create Table Employeee(Employee_ID Number(6)Not Null,First_Name Varchar(20),Last_Name Varchar(25) Not Null,Email Varchar(25) Not Null,PhoneNumber Varchar(20) Not Null,Hire_Date Date Not Null,Job_ID Varchar(10) Not Null,Salary Number(8,2),Commission_pct Number(2,2),Manager_ID Number(6),Department_ID Number(4));
3
4
```

The results tab shows the message "Table created." and a execution time of "0.05 seconds". There is also a note to "Activate Windows Go to Settings to activate Windows."

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

QUERY:

```
Alter table Employeee modify(Last_Name varchar(50));
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, a user icon for 'DEVDHARSHAN SR database@04', and a schema dropdown set to 'WKSP DATABASE04'. Below the header, a toolbar includes Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run buttons. The main area contains the following SQL code:

```
1 --create table Dep (Dep_ID Number(6)Not Null, Dept_Name Varchar(20), Manager_ID VARCHAR(10), Location_ID Number(10))
2 --create Table Employee(Employee_ID NUMBER(6)Not Null,First_Name Varchar(20),Last_Name Varchar(25) Not Null,Email Varchar(25) Not Null,PhoneNumber Varchar(20),Hire_Date Date Not Null,Job_ID NUMBER(4)Not Null,Salary NUMBER(8,2),Dept_ID NUMBER(6));
3
4 alter table employee modify [Last_Name varchar(50)];
```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab shows the output: "Table altered." It also indicates a execution time of "0.05 seconds". At the bottom, it shows the user information "220701060@rajalakshmi.edu.in database@04 en", the copyright notice "Copyright © 1999, 2023, Oracle and/or its affiliates.", and the APEX 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 varchar(25)not null,Salary number(8,2),Dept_id number(6) not null);

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, it shows the user's name, DEVDHARSHAN SR, and the schema, WKSP_DATABASE04. The main area is titled "SQL Commands". The code editor contains the following SQL statements:

```
1 --create table Dep (Dep_ID Number(6)Not Null, Dept_Name Varchar(20), Manager_ID VARCHAR(10), Location_ID Number(10))
2 --create Table Employeee(Employee_ID NUMBER(6)Not Null,First_Name Varchar(20),Last_Name Varchar(25) Not Null,Email Varchar(25) Not Null,PhoneNumber Varchar(20),Hire_Date Date Not Null,Job_ID
3 --alter table employeee modify (Last_Name varchar(50));
4
5 create table employee2(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);
```

The "Results" tab is selected, showing the output: "Table created." Below it, the execution time is listed as "0.04 seconds". At the bottom, there are copyright notices for Oracle and the APEX version.

5. Drop the EMP table.

QUERY:

Drop table Employeee;

OUTPUT:

SQL Commands M Inbox (1,362) - 220701060@rajalakshmi.edu.in | II CSE A DBMS 2023 - 2024 | +

https://apex.oracle.com/pls/apex/l/apex/sql-workshop/sqlcommandprocessor?session=5779725054757

M Paused

APEX App Builder SQL Workshop Team Development Gallery Search DS DEVDHARSHAN SR database@04

↑ SQL Commands Schema WKSP_DATABASE04

Language SQL Rows 10 Clear Command Find Tables Save Run

Q C Q A:

```
1
2 --create table Dep (Dep_ID Number(6)Not Null, Dept_Name Varchar(20), Manager_ID VARCHAR(10), Location_ID Number(10))
3 --create Table Employee(Employee_ID NUMBER(6)Not Null,First_Name Varchar(20),Last_Name Varchar(25) Not Null,Email Varchar(25) Not Null,PhoneNumber Varchar(20),Hire_Date Date Not Null,Job_ID
4 --alter table employee modify (Last_Name varchar(50));
5 --create table employee2(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);
6 drop table employee;
```

Results Explain Describe Saved SQL History

Table dropped.

0.09 seconds

Activate Windows
Go to Settings to activate Windows.

220701060@rajalakshmi.edu.in database@04 en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

14:05 02-03-2024

6. Rename the EMPLOYEES2 table as EMP. **QUERY:**

Rename employees2 to emp;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side of the header shows the user's name, DEVDHARSHAN SR, and the schema, WKSP_DATABASE04. The main workspace is titled "SQL Commands". The SQL editor contains the command: "1 rename employee2 to emp;". Below the editor, the "Results" tab is selected, showing the output: "Statement processed." and "0.05 seconds". At the bottom, the footer includes copyright information, a Windows activation message, and system status icons.

```
1 rename employee2 to emp;
```

Statement processed.
0.05 seconds

Activate Windows
Go to Settings to activate Windows.

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

Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. A single command is entered in the text input field: 'comment on table dept is 'Department info'';'. Below the input field, there are buttons for 'Save' and 'Run'. The 'Run' button is highlighted in green. The results section at the bottom shows the output of the command: 'Statement processed.' and '0.04 seconds'. The status bar at the bottom right indicates 'Oracle APEX 23.2.4'.

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

Statement processed.
0.04 seconds

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. A single line of SQL code is entered: '1 Alter table emp drop column first_name;'. Below the command, the results section displays the output: 'Table altered.' and '0.05 seconds'. The bottom right corner of the interface shows '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:

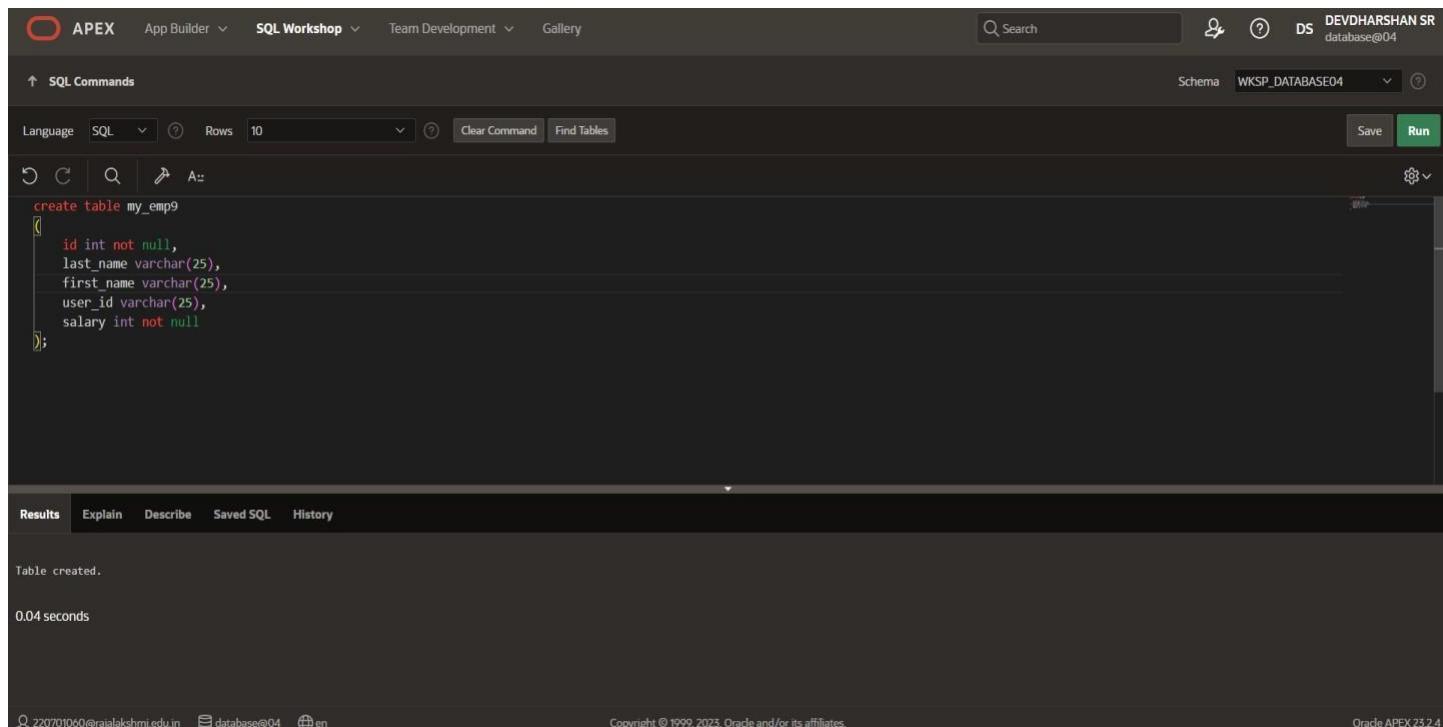
1. Create MY_EMPLOYEE table with the following structure

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_emp9(id int not null, last name varchar(25), first name varchar(25), user id varchar(25), salary int not null);

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 command:

```
create table my_emp9
(
    id int not null,
    last_name varchar(25),
    first_name varchar(25),
    user_id varchar(25),
    salary int not null
);
```

Below the code, the 'Results' tab is active, showing the output: "Table created." and "0.04 seconds". The bottom status bar includes the user information "220701060@rajalakshmi.edu.in" and "database@04", along with copyright and version details.

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_emp9 values(1,'Patel','Ralph','rpatel',895);
Insert into my_emp9 values(2,'Dancs','Betty','bdancs',860);
```

OUTPUT:

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

```
insert into my_emp9 values(4,'newman','chad','cnewman',730);
```

The Results tab is selected, showing the output:

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

Execution details:

- Time: 0.00 seconds
- User: 220701060@rajalakshmi.edu.in
- Session: database@04
- Language: SQL
- Rows: 10

Page footer:

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

3. Display the table with values.

QUERY:

Select * from my_emp9;order by id;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the user 'DEVDHARSHAN SR' connected to 'database@04'. Below the tabs, there's a search bar and some icons. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. It contains a command input field with the query 'select * from my_emp9 order by id;'. To the right of the input field are 'Save' and 'Run' buttons. The results section is active, showing a table with four columns: ID, LAST_NAME, FIRST_NAME, and USER_ID, with a corresponding SALARY column. The table has two rows. At the bottom of the results table, it says '2 rows returned in 0.00 seconds' and has a 'Download' link. The footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

ID	LAST_NAME	FIRST_NAME	USER_ID	SALARY
1	patel	ralph	rpatel	895
2	dancs	betty	bdancs	800

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_employees values (3,'biri','ben','bbiri',1100),
Values(4,'newman','chad','cnewman',750);

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Search DS DEVDA database

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

↻ ⌂ 🔍 A±

```
insert into my_emp9 values(4,'newman','Chad','cnewman',730);
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

220701060@rajalakshmi.edu.in database@04 en

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

Oracle Database

5. Make the data additions permanent.

QUERY:

Select * from my_emp9 order by Id;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the schema as WKSP_DATABASE04 and the connection as DEVDHARSHAN SR database@04. The main workspace is titled "SQL Commands" and contains the following SQL code:

```
select * from my_emp9 order by id;
```

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

ID	LAST_NAME	FIRST_NAME	USER_ID	SALARY
1	patel	ralph	rpatel	895
2	dancs	betty	bdancs	800
3	biri	ben	bbiri	830
4	newman	Chad	cnewman	750

At the bottom of the results pane, it says "4 rows returned in 0.00 seconds". The footer of the page includes copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. and Oracle APEX 23.3.4.

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

QUERY:

Update my_emp9 set last_name='DREXLER' where Id=3;

OUTPUT:

A screenshot of 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, it shows a user profile for 'DEVDHARSHAN SR' connected to 'database@04'. The main area is titled 'SQL Commands' with a sub-header 'Language: SQL, Rows: 10'. Below this is a text input field containing the SQL command: 'update my_emp9 set last_name='DREXLER' where Id=3;'. The results section shows the output: '1 row(s) updated.' and '0.01 seconds'. The bottom status bar displays the session ID '220701060@rajalakshmi.edu.in', the database connection 'database@04', and the environment 'en'. It also includes copyright information '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_emp9 set salary=1000 where salary<=900;
```

OUTPUT:

A screenshot of the Oracle APEX SQL Workshop interface, identical to the previous one but with a different SQL command. The top navigation bar, user profile, and interface layout are the same. The SQL command in the input field is: 'update my_emp9 set salary=1000 where salary<=900;'. The results section shows '4 row(s) updated.' and '0.01 seconds'. The bottom status bar and footer information are also identical to the first screenshot.

8. Delete Betty dancs from MY_EMPLOYEE table.

QUERY:

Delete from my_emp9 where first_name='betty';

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:

```
delete from my_emp9 where first_name='betty';
```

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

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

Execution time is listed as 0.01 seconds. At the bottom of the interface, the copyright notice reads "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version "Oracle APEX 23.2.4".

9. Empty the fourth row of the emp table.

QUERY:

Delete from my_emp9 where id=4;

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

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

delete from my_emp9 where id=4;

Results Explain Describe Saved SQL History

1 row(s) deleted.
0.01 seconds

220701060@rajalakshmi.edu.in database@04 en Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

RESULT:

INCLUDING CONSTRAINTS

EX_NO:3

DATE:

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

QUERY:

```
Create table emp19(Id int not null,last_name varchar2(25), email varchar(25),salary int not null, constraint my_emp_id_pk primary key(id));
```

OUTPUT:

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

```
create table emp19
(
    ID INT NOT NULL,
    LAST_NAME VARCHAR2(25),
    EMAIL VARCHAR2(25),
    SALARY INT NOT NULL,
    CONSTRAINT my_emp_id_pk PRIMARY KEY(ID)
);
```

The code is highlighted in green, indicating it is valid SQL. Below the code, the results of the execution are shown:

Table created.
0.06 seconds

The URL of the page is https://apex.oracle.com/pls/apex/r/apex/workspace/home?session=14349558133531

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 dep13(id int not null,last_name varchar2(25),first_name varchar2(25),email varchar2(25),salary int not null constraint my_dep_id_pk primary key(id));
```

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 displays the following SQL command:

```
create table dep19
(
    ID INT NOT NULL,
    LAST_NAME VARCHAR2(25),
    FIRST_NAME VARCHAR2(25),
    EMAIL VARCHAR2(25),
    SALARY INT NOT NULL,
    CONSTRAINT my_dep_id_mk PRIMARY KEY(ID)
);
```

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

```
Table created.
```

Execution time: 0.05 seconds

At the bottom, the session information is shown: 220701050@rajalakshmi.edu.in, database@04, en. Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 25.2.4

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

QUERY:

```
alter table emp19
add dept_id int not null;
```

```
alter table emp19
add constraint my_dept_id_mk Foreign key dept_id references emp19(Id);
```

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 displays the following SQL command:

```
alter table emp19
add dept_id int not null;
```

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

```
Table altered.
```

Execution time: 0.08 seconds

At the bottom, the session information is shown: 220701050@rajalakshmi.edu.in, database@04, en. Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 25.2.4

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

```
alter table emp19
add constraint my_dept_id_pk Foreign key (dept_id) references emp19(id);
```

The results section shows the output of the command:

```
Table altered.  
0.06 seconds
```

At the bottom, the footer includes the URL '220701060@rajalakshmi.edu.in', the database name 'database@04', and the language 'en'. It also mentions 'Copyright © 1999, 2025, Oracle and/or its affiliates.' and 'Oracle APEX 25.2.4'.

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

QUERY:

```
alter table emp19
add commissions number(2,2) check (commissions>0);
```

OUTPUT:

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

```
alter table emp19
add commissions number(2,2) check (commissions>0);
```

The results section shows the output of the command:

```
Table altered.  
0.06 seconds
```

At the bottom, the footer includes the URL '220701060@rajalakshmi.edu.in', the database name 'database@04', and the language 'en'. It also mentions 'Copyright © 1999, 2025, Oracle and/or its affiliates.' and 'Oracle APEX 25.2.4'.

:

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

RESULT:

Writing Basic SQL SELECT Statements

EX_NO:4

DATE:

1. The following statement executes successfully.

Identify the Errors

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

QUERY:

```
Select employee_id, last_name, sal*12 as annual salary from employees;
```

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

QUERY:

```
Create table Employees(emp_no number(5),emp_id number(20),first_name varchar2(20),last_name  
varchar2(20),salary number(8,2),job_id number(20),jobcoee number(20),hire_date varchar2(20));
```

OUTPUT:

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

```
1 create table employees(emp_no number(5),emp_id number(20),first_name varchar2(20),last_name varchar2(20),salary number(7,2),job_id number(20),jobcode number(20),hire_date varchar2(20));
```

The results tab is active, displaying the output:

```
Table created.
```

Execution time: 0.03 seconds

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

QUERY:

```
Select emp_no,last_name,jobcode,hire_date from Employees order by emp_no;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows a connection named DS DEVDHARSHAN SR database@04 and a schema WKSP_DATABASE04. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The code editor contains the following SQL:

```

1 --create table employees(emp_no number(5),emp_id number(20),first_name varchar2(20),last_name varchar2(20),salary number(7,2),job_id number(20),jobcode number(20),hire_date varchar2(20));
2 --insert into employees values('2','102','tony','stark','20000','002','401','15/01/2015');
3 --insert into employees values('3','103','bruce','banner','40000','003','402','17/10/2016');
4 --insert into employees values('4','105','bruce','wayne','50000','004','404','18/05/2015');
5 --insert into employees values('5','106','steve','smith','25000','005','405','21/11/2013');
6 --select emp_id,last_name,salary*12 from employees;
7 select emp_no,last_name,jobcode,hire_date from employees order by emp_no;
8
9
10

```

Below the code editor is a results grid with columns: EMP_NO, LAST_NAME, JOBCODE, and HIRE_DATE. The data is as follows:

EMP_NO	LAST_NAME	JOBCODE	HIRE_DATE
1	roger	400	14/01/2018
2	stark	401	15/01/2015
3	banner	402	17/10/2016
4	wayne	404	18/05/2015
5	smith	405	21/11/2013

At the bottom of the interface, there are status bars for the browser and operating system.

4. Provide an alias STARTDATE for the hire date.

QUERY:

Select hire_date as STARTDATE from emplo **OUTPUT:**

The screenshot shows the Oracle APEX SQL Workshop interface. The setup is identical to the previous one, with the SQL Workshop tab selected. The code editor contains the same SQL as the first screenshot, but the final line is modified to include an alias:

```

1 --create table employees(emp_no number(5),emp_id number(20),first_name varchar2(20),last_name varchar2(20),salary number(7,2),job_id number(20),jobcode number(20),hire_date varchar2(20));
2 --insert into employees values('2','102','tony','stark','20000','002','401','15/01/2015');
3 --insert into employees values('3','103','bruce','banner','40000','003','402','17/10/2016');
4 --insert into employees values('4','105','bruce','wayne','50000','004','404','18/05/2015');
5 --insert into employees values('5','106','steve','smith','25000','005','405','21/11/2013');
6 --select emp_id,last_name,salary*12 from employees;
7 select emp_no,last_name,jobcode,hire_date as STARTDATE from employees order by emp_no;
8
9
10

```

The results grid shows the data with the 'HIRE_DATE' column renamed to 'STARTDATE' in the output:

EMP_NO	LAST_NAME	JOBCODE	STARTDATE
1	roger	400	14/01/2018
2	stark	401	15/01/2015
3	banner	402	17/10/2016
4	wayne	404	18/05/2015
5	smith	405	21/11/2013

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

QUERY:

Select distinct jobcodes 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. On the right, it shows the schema as 'WKSP_DATABASE04' and the connection as 'DEVDHARSHAN SR database@04'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

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

The Results tab displays the output of the query:

JOB_ID
1
2
4
5
3

Below the results, it says '5 rows returned in 0.01 seconds' and provides download options. The bottom of the page includes copyright information for Oracle and the APEX version.

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

QUERY:

Select last_name||','||job_id as EMPLOYEE_TITLE from employees;

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, it shows the schema as 'WKSP_DATABASE04' and the connection as 'DEVDHARSHAN SR database@04'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

```
-- create table employees(emp_no number(5),emp_id number(20),last_name varchar2(20),first_name varchar2(20),middle_name varchar2(20),salary number(7,2),job_id number(20),job_code number(4),hire_date varchar2(20));  
-- insert into employees values('1','102','tomy','stark','20000','002','401','15/01/2015');  
-- insert into employees values('3','103','bruce','banner','40000','003','402','17/10/2016');  
-- insert into employees values('4','105','bruce','wayne','50000','004','404','18/05/2015');  
-- insert into employees values('5','106','steve','smith','25000','005','405','21/11/2013');  
-- select emp_id,last_name,salary*12 from employees;  
-- select emp_no,last_name,jobcode,hire_date from employees order by emp_no;  
-- select emp_no,last_name,job_code,hire_day from emp;  
-- select hire_date as STARTDATE from employees;  
-- select distinct jobcode from employees;  
select last_name||','||job_id as EMPLOYEE_TITLE from employees;
```

The Results tab displays the output of the query:

EMPLOYEE_TITLE
roger,1
banner, 3
wayne, 4
smith, 5
stark, 2

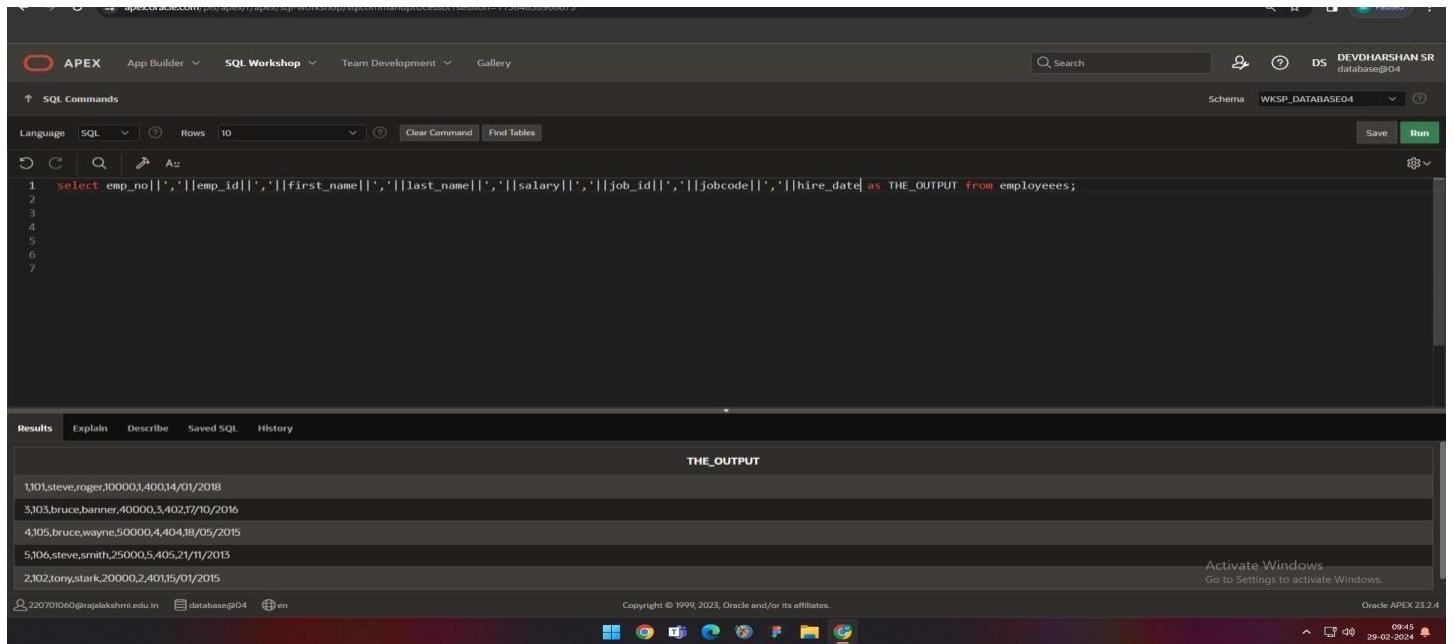
Below the results, it says '5 rows returned in 0.01 seconds' and provides download options. There is also a message about activating Windows.

OUTPUT:

7. Create a query to display all the data from the employees table. Separate each column by a comma. Name the column THE_OUTPUT.

QUERY:

Select emp_no||','||emp_id||','||first_name||','||last_name||','||salary||','||job_id||','||jobcode||','||hire_date as THE_OUTPUT from employees **OUTPUT**:



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

```
1 select emp_no||','||emp_id||','||first_name||','||last_name||','||salary||','||job_id||','||jobcode||','||hire_date||' as THE_OUTPUT from employees;
```

The results section displays the output as follows:

THE_OUTPUT
1,101,steve,roger,10000,1,400,14/01/2018
3,103,bruce,banner,40000,3,402,17/10/2016
4,105,bruce,wayne,50000,4,404,18/05/2015
5,106,steve,smith,25000,5,405,21/11/2013
2,102,tony,stark,20000,2,401,15/01/2015

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 from employees where salary>12000;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX' (with a red icon), 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows 'DS DEVDHARSHAN SR database@04'. The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the query is written:

```
1 Select last_name from employees where salary>12000;
2
```

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

LAST_NAME
banner
wayne
smith
stark

Below the table, it says '4 rows returned in 0.00 seconds' and has a 'Download' link. At the bottom, there are footer links for '220701060@rajalakshmi.edu.in', 'database@04', and 'en', along with copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

Select last_name,department_id from employees where employee_id=176;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar includes 'APEX' (with a red icon), 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows 'DS DEVDHARSHAN SR database@04'. The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the query is written:

```
1 Select last_name,dept_id from employees where emp_id=176;
2
```

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

LAST_NAME	DEPT_ID
banner	63

Below the table, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. At the bottom, there are footer links for '220701060@rajalakshmi.edu.in', 'database@04', and 'en', along with copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

```
select last_name,salary from employees where salary not between 5000 and 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, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR. The main area is titled "SQL Commands" with a search bar and a "Run" button. Below the command input, there are icons for Undo, Redo, Find, and Paste. The SQL command entered is:

```
1 select last_name,salary from employees where salary not between 5000 and 12000;
2
```

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

LAST_NAME	SALARY
banner	40000
wayne	50000
smith	25000
stark	20000

Below the table, it says "4 rows returned in 0.01 seconds" and has a "Download" link. At the bottom, it shows the session information "220701060@rajalakshmi.edu.in database@04 en" and the copyright notice "Copyright © 1999, 2023, Oracle and/or its affiliates." The footer also indicates "Oracle APEX 23.2.4".

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

QUERY:

```
Select last_name,job_id,hire_date from employees where hire_date between 'February,20,1998' and 'May,1,1998';
```

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, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR. The main area is titled "SQL Commands" with a search bar and a "Run" button. Below the command input, there are icons for Undo, Redo, Find, and Paste. The SQL command entered is:

```
1 Select last_name,job_id,hire_date from employees where hire_date between 'February,20,1998' and 'May,1,1998';
2
```

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

LAST_NAME	JOB_ID	HIRE_DATE
banner	3	02/26/1998

Below the table, it says "1 rows returned in 0.01 seconds" and has a "Download" link. At the bottom, it shows the session information "220701060@rajalakshmi.edu.in database@04 en" and the copyright notice "Copyright © 1999, 2023, Oracle and/or its affiliates." The footer also indicates "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_id from employees where department_id in(20,50) order by last_name;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, it shows the user 'DEVDHARSHAN SR' and the schema 'WKSP_DATABASE04'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. Below the command input, there are tabs for Results, Explain, Describe, Saved SQL, and History. The results table has columns 'LAST_NAME' and 'DEPT_ID'. The data returned is:

LAST_NAME	DEPT_ID
roger	56
wayne	25

Below the table, it says '2 rows returned in 0.00 seconds' and has a 'Download' link. The bottom footer includes the user '220701060@rajalakshmi.edu.in', the database 'database@04', and the language 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, it shows the user 'DEVDHARSHAN SR' and the schema 'WKSP_DATABASE04'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. Below the command input, there are tabs for Results, Explain, Describe, Saved SQL, and History. The results table has columns 'EMPLOYEE' and 'MONTHLY SALARY'. The data returned is:

EMPLOYEE	MONTHLY SALARY
roger	5000
wayne	12000

Below the table, it says 'no data found'. The bottom footer includes the user '220701060@rajalakshmi.edu.in', the database 'database@04', and the language 'en'. It also states '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 last_name,hire_date from employees where hire_date like '1994';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab contains the query: `select last_name,hire_date from employees where hire_date like '1994';`. The Results tab displays the output in a table:

LAST_NAME	HIRE_DATE
stark	05/01/1994

1 rows returned in 0.00 seconds

8. Display the last name and job title of all employees who do not have a manager.(hints: is null)

QUERY:

```
select last_name,job_id from employees where manager_id is null;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab contains the query: `select last_name,job_id from employees where manager_id is null;`. The Results tab displays the output in a table:

LAST_NAME	JOB_ID
roger	1
banner	3
wayne	4
smith	5
stark	2

5 rows returned in 0.01 seconds

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

QUERY:

```
select last_name,salary,commission_pct from employees where commission_pct 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, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR database@04. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query: `select last_name,salary,commission_pct from employees where commission_pct is not null order by salary desc;`. The Results tab displays the output in a grid format:

LAST_NAME	SALARY	COMMISSION_PCT
wayne	50000	50
roger	10000	25

Below the grid, it says "2 rows returned in 0.01 seconds". The bottom of the page includes copyright information for Oracle and APEX, and the URL 220701060@rajalakshmi.edu.in.

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

QUERY:

```
select last_name from employees where last_name like '__a%';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR database@04. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query: `select last_name from employees where last_name like '__a%';`. The Results tab displays the output in a grid format:

LAST_NAME
stark

Below the grid, it says "1 rows returned in 0.00 seconds". The bottom of the page includes copyright information for Oracle and APEX, and the URL 220701060@rajalakshmi.edu.in.

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

QUERY:

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

OUTPUT:

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

```
1 select last_name from employees where last_name like '%a%' and last_name like '%e%';
2
```

The results table shows two rows:

LAST_NAME
banner
wayne

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

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

QUERY:

```
select last_name,job_id,salary from employees where job_id in ('sales representative','stock clerk') and salary not in(2500,3500,7000);
```

OUTPUT:

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

```
1 select last_name,job_id,salary from employees where job_id in ('sales representative','stock clerk') and salary not in(2500,3500,7000);
```

The results table shows one row:

LAST_NAME	JOB_ID	SALARY
banner	sales representative	40000

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. On the right, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR database@04. The main area is titled "SQL Commands" and contains a code editor with the following SQL statement:

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

Below the code editor, there are buttons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run. The "Run" button is highlighted in green. The results section below shows the message "no data found". At the bottom, it displays the user's email (220701060@rajalakshmi.edu.in), the database connection (database@04), and the session language (en). It also includes copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates) and the software version (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. The SQL command is:

```
1 select sysdate from dual;
2
```

The results show a single row with the column name 'SYSDATE' and the value '03/13/2024'.

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

QUERY:

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

OUTPUT:

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

```
1 select emp_id, last_name, salary, salary+(15.5/100*salary) "new_salary" from employees;
2
```

The results show five rows of data with columns: EMP_ID, LAST_NAME, SALARY, and new_salary. The data is as follows:

EMP_ID	LAST_NAME	SALARY	new_salary
101	roger	10000	11550
103	banner	40000	46200
105	wayne	50000	57750
106	smith	25000	28875
102	stark	20000	23100

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The query is executed successfully, and the results are displayed in a table format. The table has columns: EMP_ID, LAST_NAME, SALARY, new_salary, and Increase. The data shows salary increases for five employees: roger, banner, wayne, smith, and stark.

EMP_ID	LAST_NAME	SALARY	new_salary	Increase
101	roger	10000	11550	1550
103	banner	40000	46200	6200
105	wayne	50000	57750	7750
106	smith	25000	28875	3875
102	stark	20000	23100	3100

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

QUERY:

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

OUTPUT:

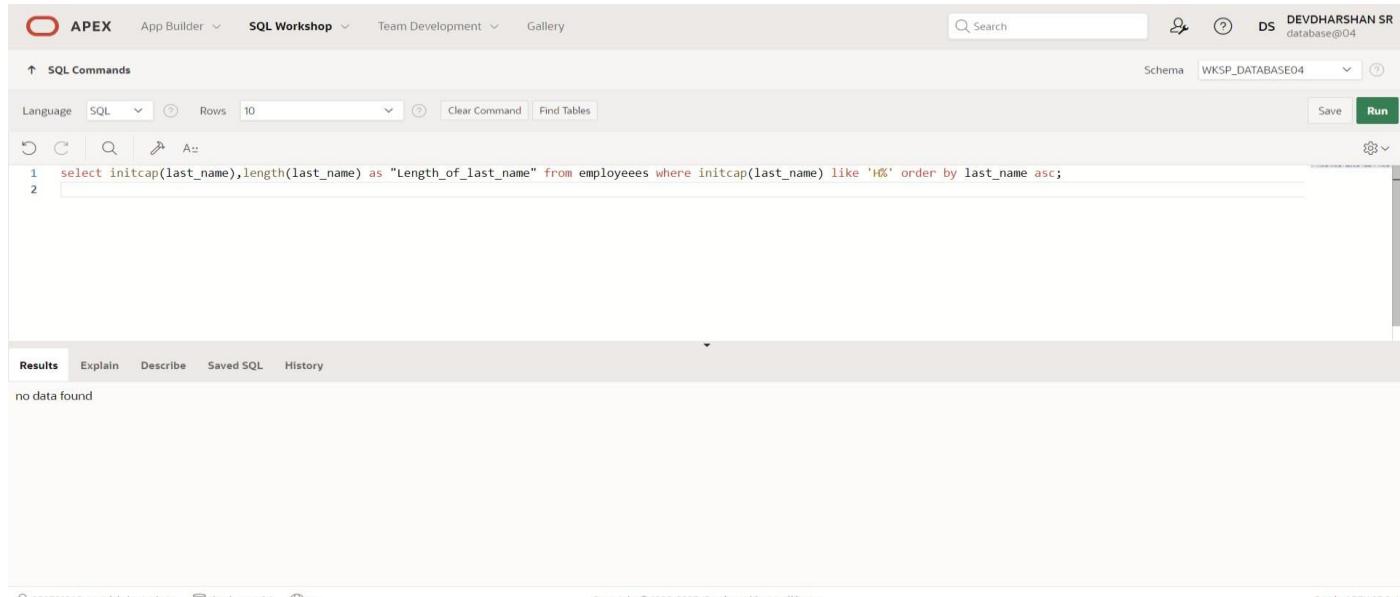
The screenshot shows the Oracle APEX SQL Workshop interface. The query is executed successfully, but the results table is empty, indicating no data found that matches the specified criteria.

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

QUERY:

```
select initcap(last_name),length(last_name) as "Length_of_last_name" from employees where last_name like 'H%'  
order by last_name 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, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the following SQL code:

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

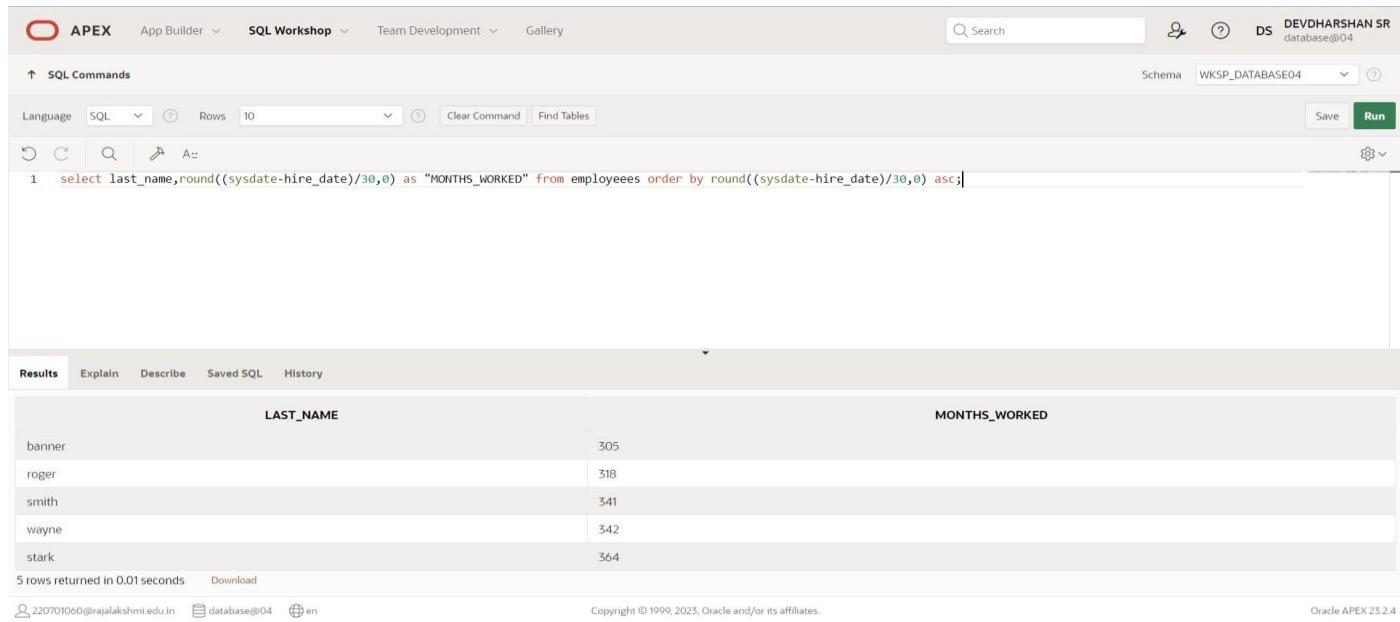
The Results tab is selected, displaying the message "no data found".

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the following SQL code:

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

The Results tab is selected, displaying a table with two columns: LAST_NAME and MONTHS_WORKED. The data is as follows:

LAST_NAME	MONTHS_WORKED
banner	305
roger	318
smith	341
wayne	342
stark	364

At the bottom, it says "5 rows returned in 0.01 seconds" and has a "Download" link. The footer includes the user information 220701060@rajalakshmi.edu.in, the schema database@04, and the language en. It also states 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 last_name||' earns'||salary||' monthly but wants'||salary*3 as "DREAM_SALARIES" from employees;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the schema as 'DEVDHARSHAN SR database@04'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query:

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

The Results tab displays the output:

DREAM_SALARIES	
roger	earns 10000 monthly but wants 30000
banner	earns 40000 monthly but wants 120000
wayne	earns 50000 monthly but wants 150000
smith	earns 25000 monthly but wants 75000
stark	earns 20000 monthly but wants 60000

Below the table, it says '5 rows returned in 0.00 seconds' and has a 'Download' link. The bottom footer includes copyright information and the version '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 last_name, lpad(salary,15,'$') as "SALARY" from employees;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar and schema are identical. The SQL Commands tab contains the query:

```
1 select last_name, lpad(salary,15,'$') as "SALARY" from employees;
2
```

The Results tab displays the output:

LAST_NAME	SALARY
roger	\$\$\$\$\$\$\$\$\$\$10000
banner	\$\$\$\$\$\$\$\$\$\$40000
wayne	\$\$\$\$\$\$\$\$\$\$50000
smith	\$\$\$\$\$\$\$\$\$\$25000
stark	\$\$\$\$\$\$\$\$\$\$20000

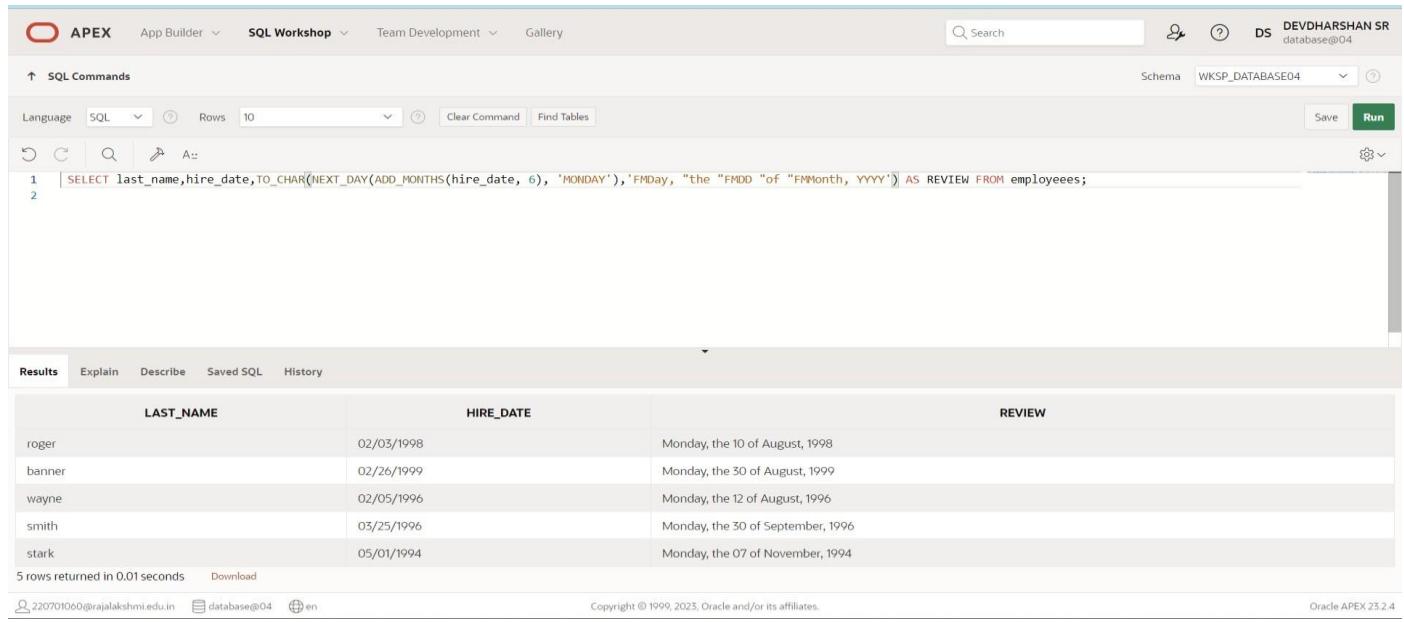
Below the table, it says '5 rows returned in 0.00 seconds' and has a 'Download' link. The bottom footer includes copyright information and the version 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab is active, displaying the query from question 9. The results tab shows the output for five employees: roger, banner, wayne, smith, and stark. The results are as follows:

LAST_NAME	HIRE_DATE	REVIEW
roger	02/03/1998	Monday, the 10 of August, 1998
banner	02/26/1999	Monday, the 30 of August, 1999
wayne	02/05/1996	Monday, the 12 of August, 1996
smith	03/25/1996	Monday, the 30 of September, 1996
stark	05/01/1994	Monday, the 07 of November, 1994

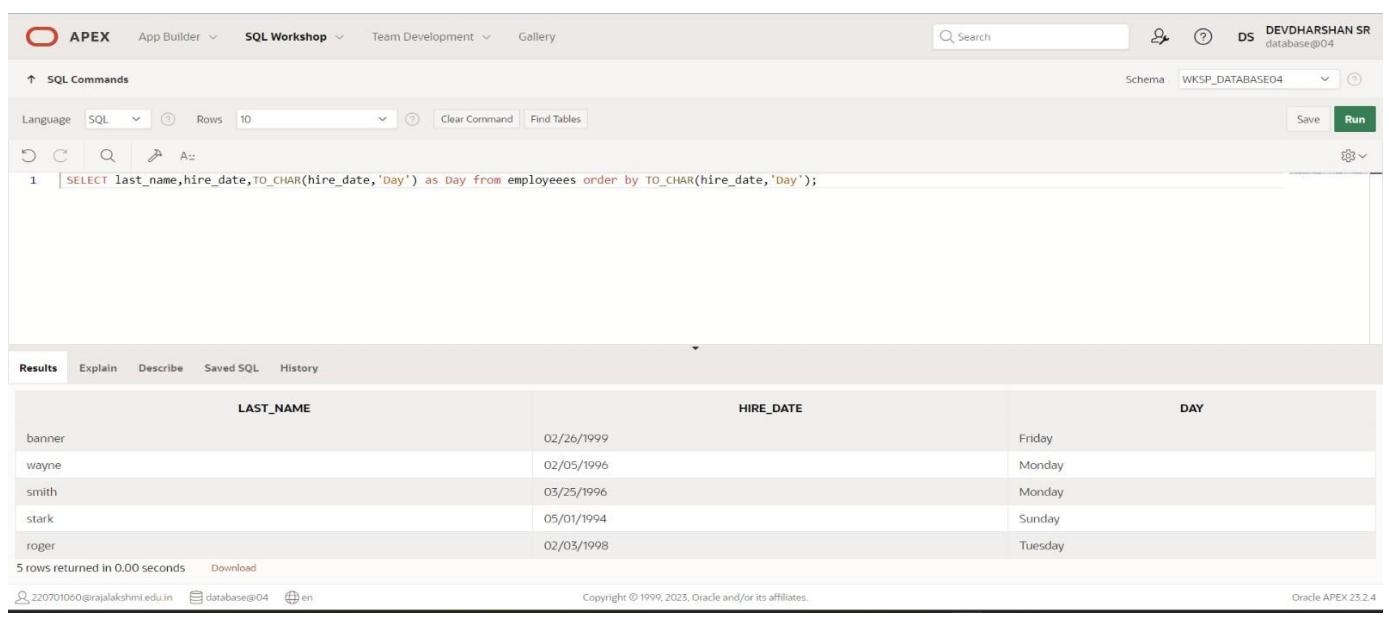
5 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab is active, displaying the query from question 10. The results tab shows the output for five employees: banner, wayne, smith, stark, and roger. The results are as follows:

LAST_NAME	HIRE_DATE	DAY
banner	02/26/1999	Friday
wayne	02/05/1996	Monday
smith	03/25/1996	Monday
stark	05/01/1994	Sunday
roger	02/03/1998	Tuesday

5 rows returned in 0.00 seconds

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

```
1 | SELECT e.last_name, e.dept_id, d.dept_name  
2 | FROM employees e, departments d  
3 | WHERE e.dept_id = d.dept_id;  
4 |
```

The results section displays the output:

LAST_NAME	DEPT_ID	DEPT_NAME
banner	22	csd

1 rows returned in 0.00 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 APEX SQL Workshop interface. The SQL command window contains the following query:

```
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 section displays the output:

JOB_ID	LOCATION_ID
55	12
80	12

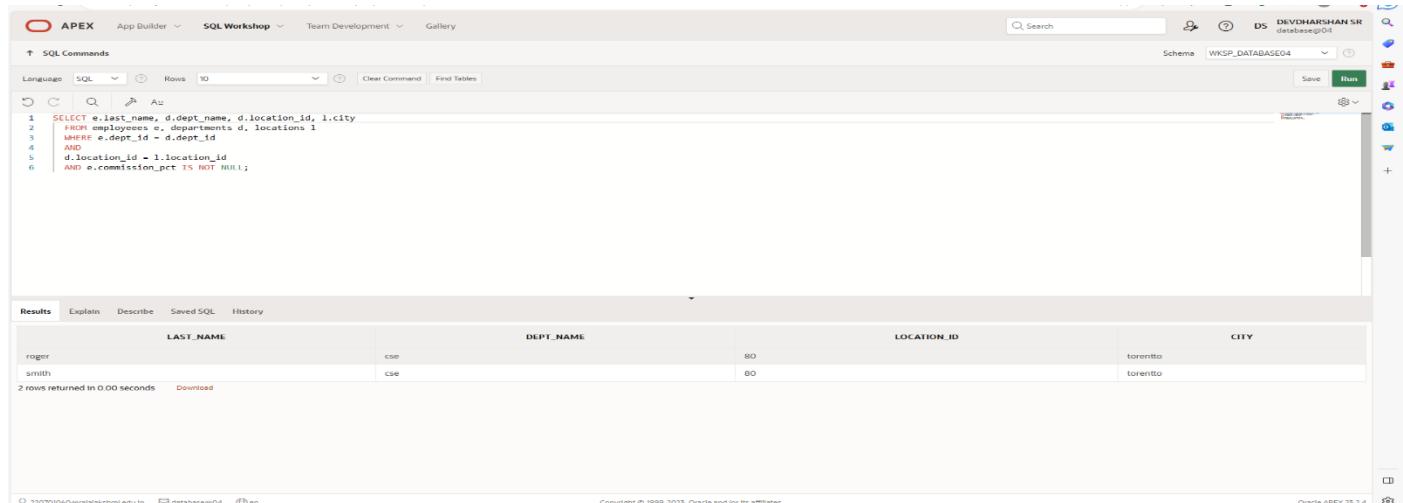
2 rows returned in 0.01 seconds

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 SQL editor contains the following 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 = l.location_id  
AND e.commission_pct IS NOT NULL;
```

The results pane displays the following data:

LAST_NAME	DEPT_NAME	LOCATION_ID	CITY
roger	cse	80	toronto
smith	cse	80	toronto

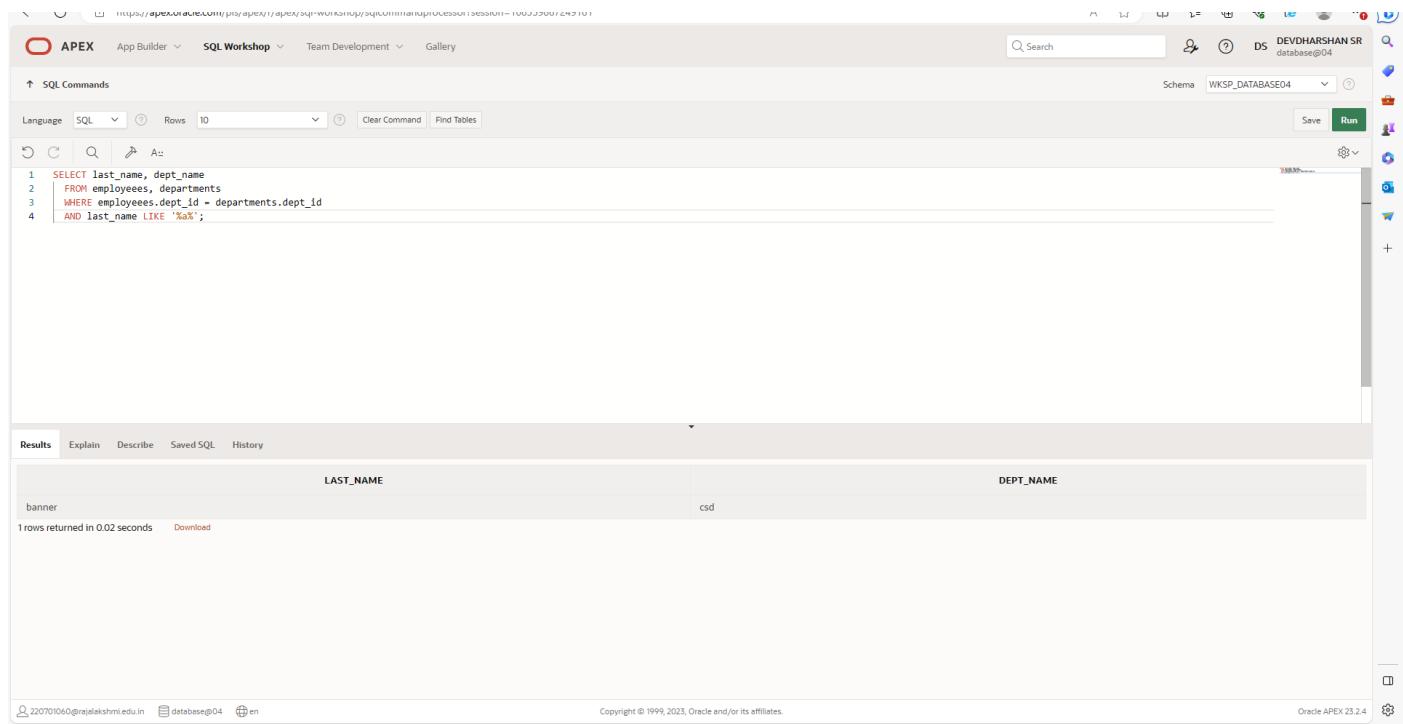
2 rows returned in 0.00 seconds

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 SQL editor contains the following query:

```
SELECT last_name, dept_name  
FROM employees, departments  
WHERE employees.dept_id = departments.dept_id  
AND last_name LIKE '%a%';
```

The results pane displays the following data:

LAST_NAME	DEPT_NAME
banner	csd

1 rows returned in 0.02 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 SQL command window contains the following query:

```
1 SELECT e.last_name, e.job_id, e.dept_id, d.dept_name
2 FROM employees e JOIN departments d
3 ON (e.dept_id = d.dept_id)
4 JOIN locations l
5 ON (d.location_id = l.location_id)
6 WHERE LOWER(l.city) = 'toronto';
```

The results section displays the following data:

LAST_NAME	JOB_ID	DEPT_ID	DEPT_NAME
roger	80	80	cse
smith	55	80	cse

2 rows returned in 0.00 seconds

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

QUERY:

```
Select w.last_name "Employee",w.emp_id "EMP#",m.last_name "Manager",m.emp_id "Mgr#"
From employees w join employees m on (w.manager_id=m.emp_id);
```

OUTPUT:

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

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

The results section displays the following data:

Employee	EMP#	Manager	Mgr#
roy	101	roy	101

1 rows returned in 0.01 seconds

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 is:

```
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);  
6  
7
```

The results table has four columns: Employee, EMP#, Manager, and Mgr#. The data is:

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 is:

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

The results table has three columns: DEPARTMENT, EMPLOYEE, and COLLEAGUE. The data is:

DEPARTMENT	EMPLOYEE	COLLEAGUE
11	rayan	rayan
22	roy	roy

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, it shows the user 'DEVDHARSHAN SR' and the schema 'WKSP_DATABASE04'. The main area has tabs for 'SQL Commands' (selected), 'Language' (SQL), 'Rows' (10), 'Clear Command', and 'Find Tables'. Below these are icons for Undo, Redo, Search, and Run. The SQL command entered is:

```
1 SELECT e.last_name, e.job_id, d.dept_name, e.salary, j.grade_level
2 FROM employees e JOIN departments d
3 ON (e.dept_id = d.dept_id)
4 JOIN job_grades j
5 ON (e.salary BETWEEN j.low_sal AND j.high_sal);
```

The results tab is selected, displaying the following data:

LAST_NAME	JOB_ID	DEPT_NAME	SALARY	GRADE_LEVEL
roger	80	cse	10000	b
roger	80	cse	10000	a
banner	10	csd	40000	b
banner	10	csd	40000	a
banner	10	csd	40000	a++

At the bottom, it shows the connection details '220701060@rajalakshmi.edu.in database@04 en', copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

10. Create a query to display the name and hire date of any employee hired after employee Davies.

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows the user 'DEVDHARSHAN SR' and the schema 'WKSP_DATABASE04'. The main area has tabs for 'SQL Commands' (selected), 'Language' (SQL), 'Rows' (10), 'Clear Command', and 'Find Tables'. Below these are icons for Undo, Redo, Search, and Run. The SQL command entered is:

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

The results tab is selected, displaying the following data:

LAST_NAME	HIRE_DATE
roy	02/03/2020

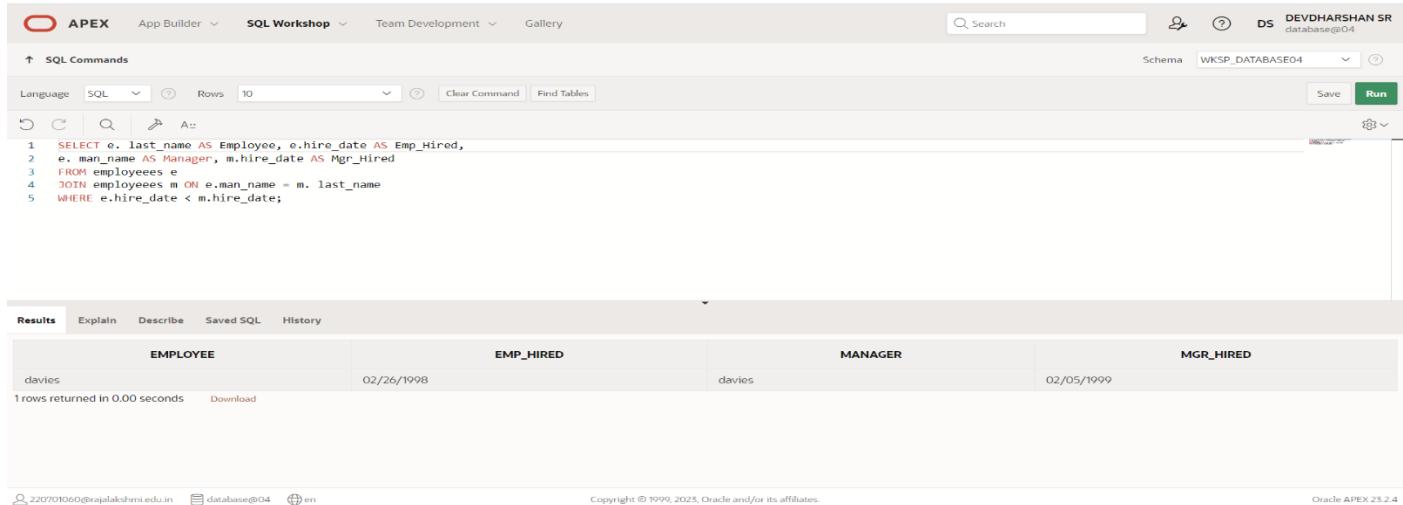
At the bottom, it shows the connection details '220701060@rajalakshmi.edu.in database@04 en', copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and the version 'Oracle APEX 23.2.4'.

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 a connection named DEVDHARSHAN SR connected to database@O4. The SQL Commands tab is active, displaying the following SQL code:

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

The Results tab shows the output of the query:

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

1 rows returned in 0.00 seconds

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. On the right, there's a user profile for DEVDHARSHAN SR and a database connection to WKSP_DATABASE04. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select round(max(salary),0)"Maximum",  
2      round(min(salary),0)"Minimum",  
3      round(sum(salary),0)"sum",  
4      round(avg(salary),0) "Average"  
from EMPA;
```

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

	Maximum	Minimum	sum	Average
	70000	45000	280000	56000

The results show that 1 row was returned in 0.01 seconds. At the bottom of the page, there are footer links for copyright information and Oracle APEX version 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(salary),0)"average" from EMPA group by job_id;

OUTPUT:

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

```
1 select job_id ,Round(MAX(salary),0) "MAXIMUM",Round (Min(salary),0)"Minimun",Round
2 (SUM(Salary),0)"sum",Round (Avg (salary),0)"average" from EMPA group by job_id;
```

The results are displayed in a table:

JOB_ID	MAXIMUM	MINIMUM	sum	average
44	45000	45000	45000	45000
11	60000	60000	60000	60000
22	50000	50000	50000	50000
33	70000	70000	70000	70000
55	55000	55000	55000	55000

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 query entered is:

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

The results are displayed in a table:

JOB_ID	COUNT(*)
47	1

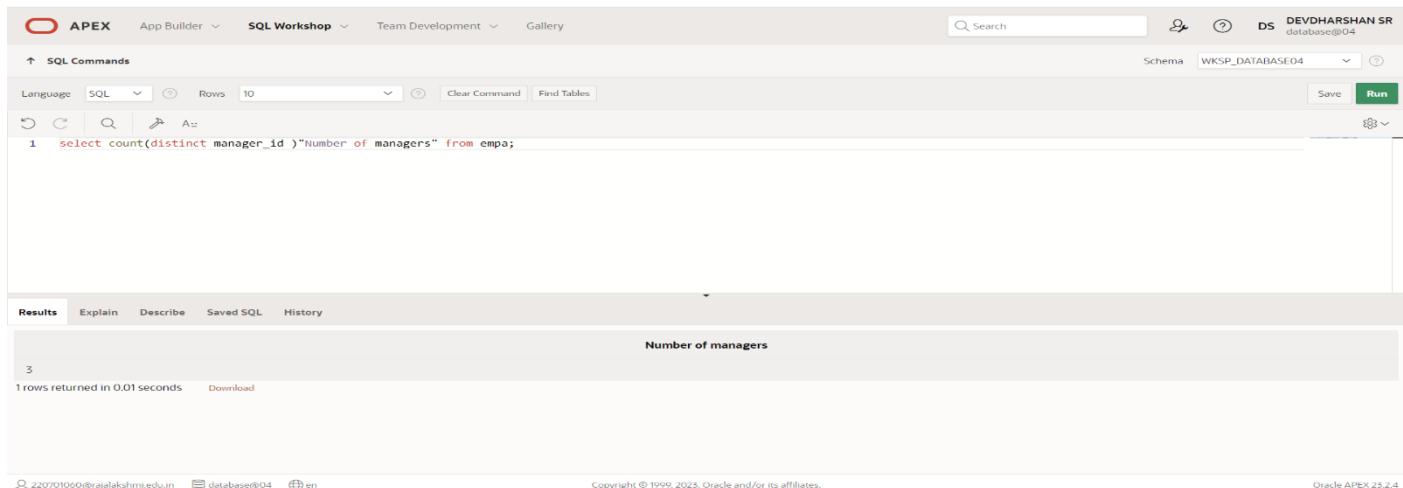
1 rows returned in 0.00 seconds [Download](#)

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".

Number of managers
3

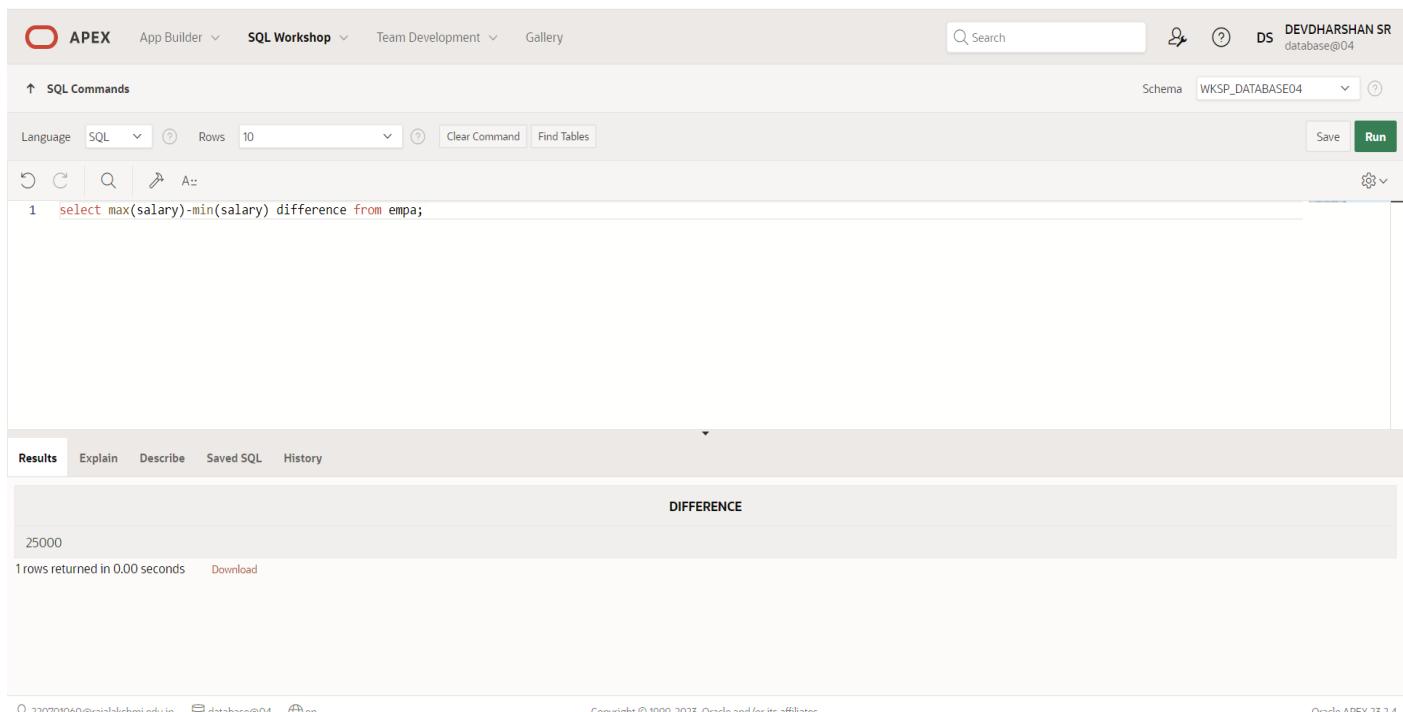
1 rows returned in 0.01 seconds [Download](#)

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 empa;`. The results section displays a single row with the value 25000, labeled "DIFFERENCE".

DIFFERENCE
25000

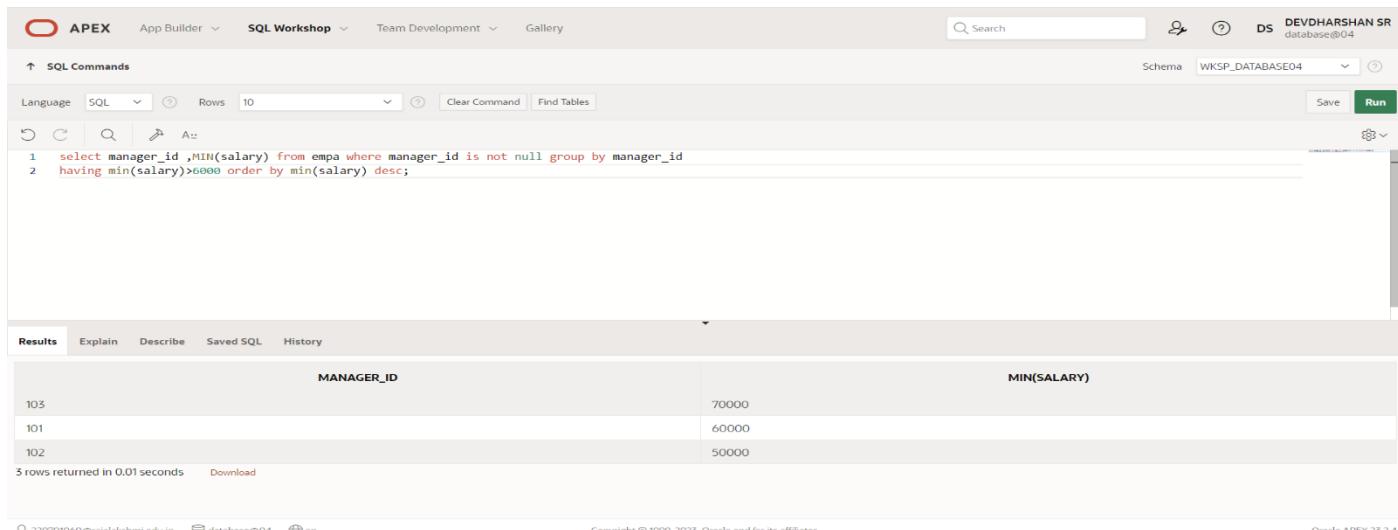
1 rows returned in 0.00 seconds [Download](#)

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 SQL command entered is:

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

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

MANAGER_ID	MIN(SALARY)
103	70000
101	60000
102	50000

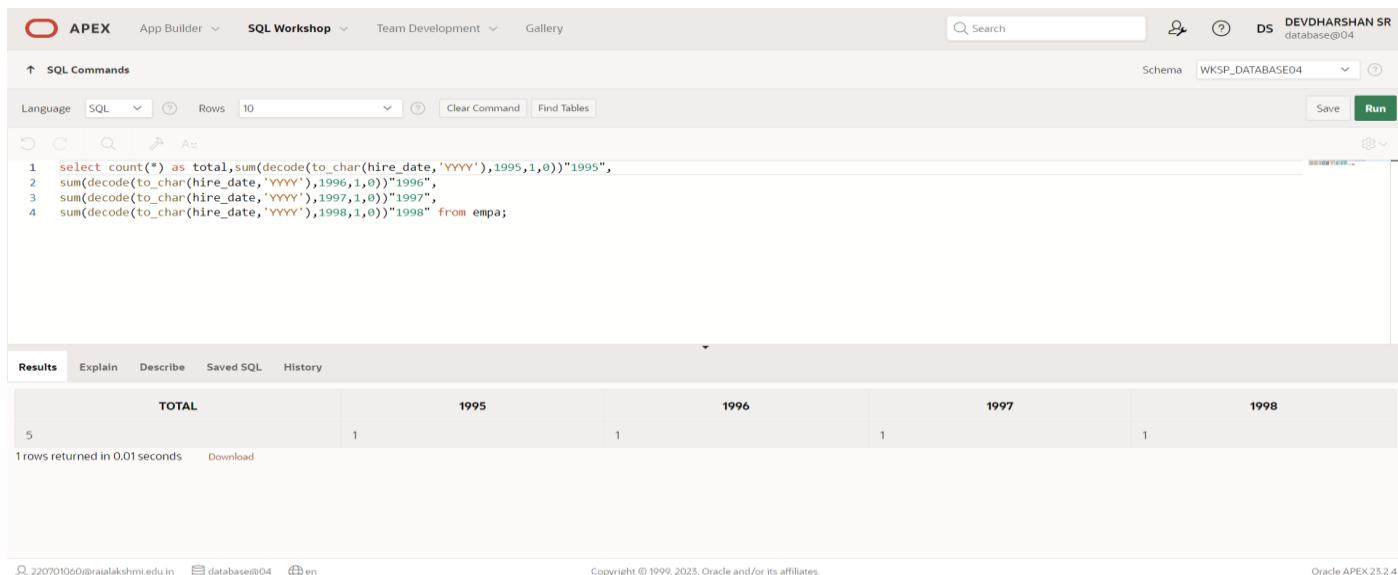
3 rows returned in 0.01 seconds

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 SQL command entered is:

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

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

```
1 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;
```

The results are displayed in a matrix format:

job	Dept20	dept50	dept80	dept90	TOTAL
44	-	-	-	45000	45000
11	60000	-	-	-	60000
22	-	50000	-	-	50000
33	-	-	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 is:

```
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;
```

The results are displayed in a table:

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

3 rows returned in 0.05 seconds

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

RESULT:

SUB QUERIES

EX_NO:9

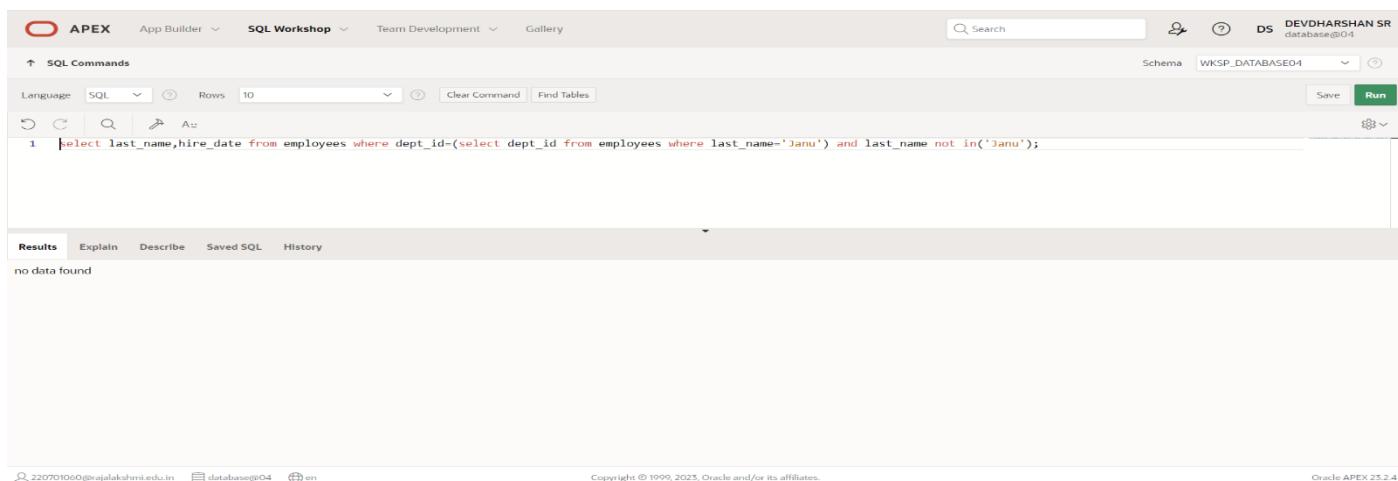
DATE:

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. In the SQL Commands pane, the following query is entered:

```
1 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');
```

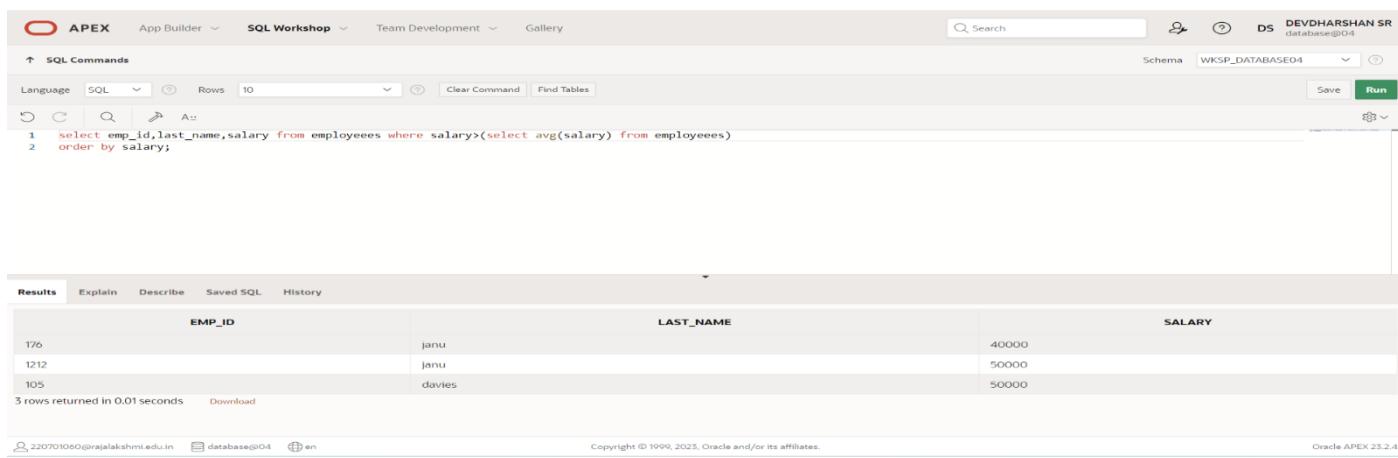
The Results pane shows the output: "no data found".

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

QUERY:

```
select 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. In the SQL Commands pane, the following query is entered:

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

The Results pane displays a table with the following data:

EMP_ID	LAST_NAME	SALARY
176	janu	40000
1212	janu	50000
105	davies	50000

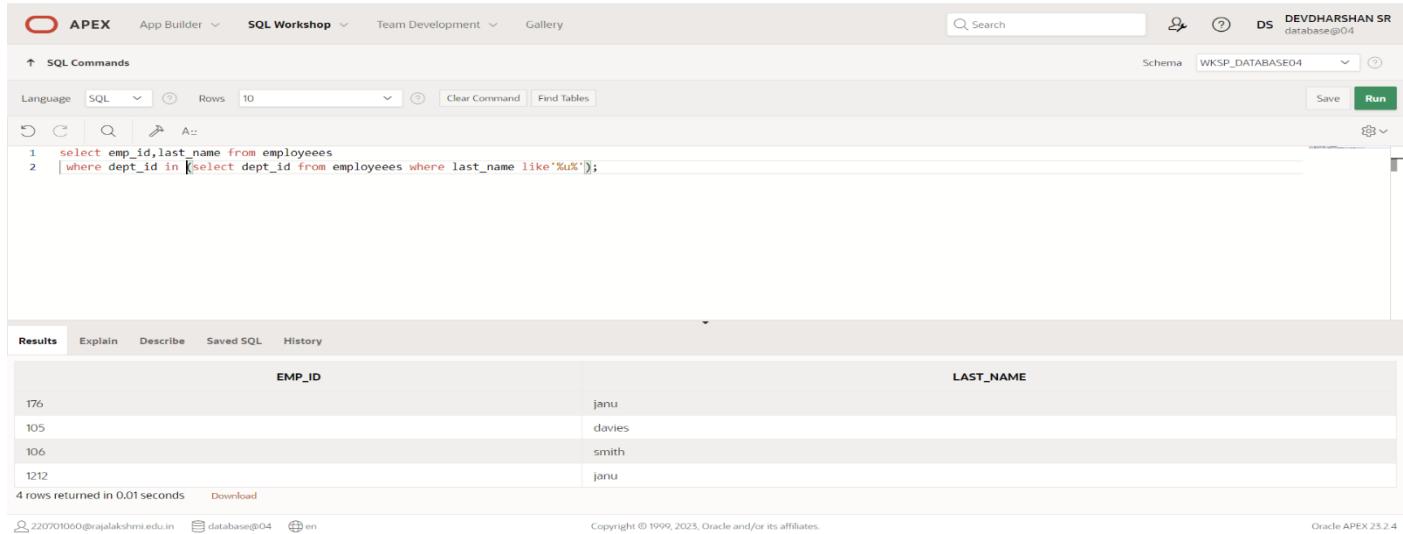
3 rows returned in 0.01 seconds

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

QUERY:

```
select 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 SQL Commands tab is active, displaying the following query:

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

The Results tab shows the output of the query:

EMP_ID	LAST_NAME
176	janu
105	davies
106	smith
1212	janu

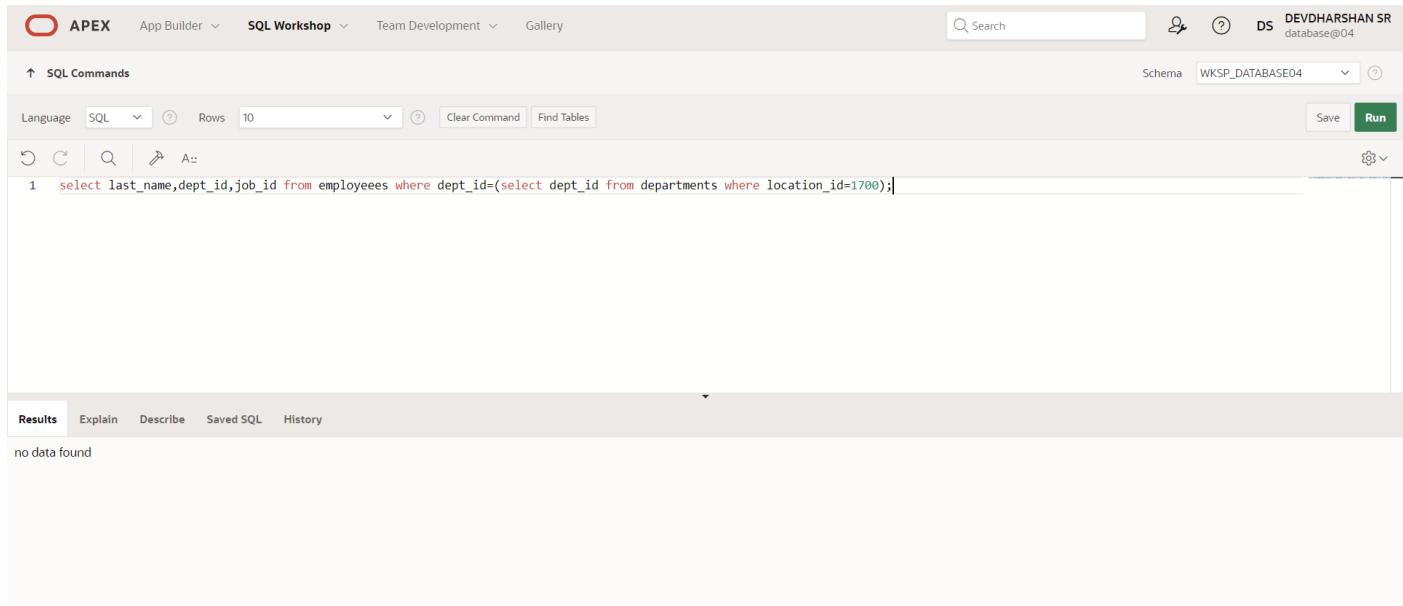
4 rows returned in 0.01 seconds

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 Commands tab is active, displaying the following query:

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

The Results tab shows the output of the query:

LAST_NAME	DEPT_ID	JOB_ID
janu	10	14

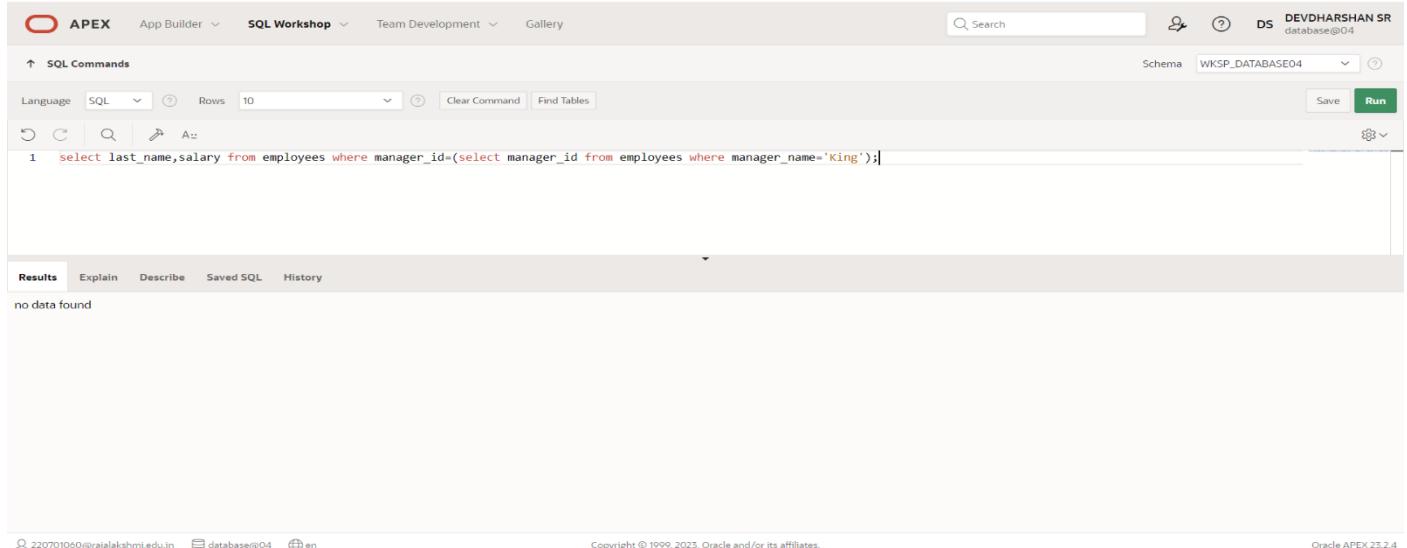
no data found

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows 'DS DEVDHARSHAN SR database@O4'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is:

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

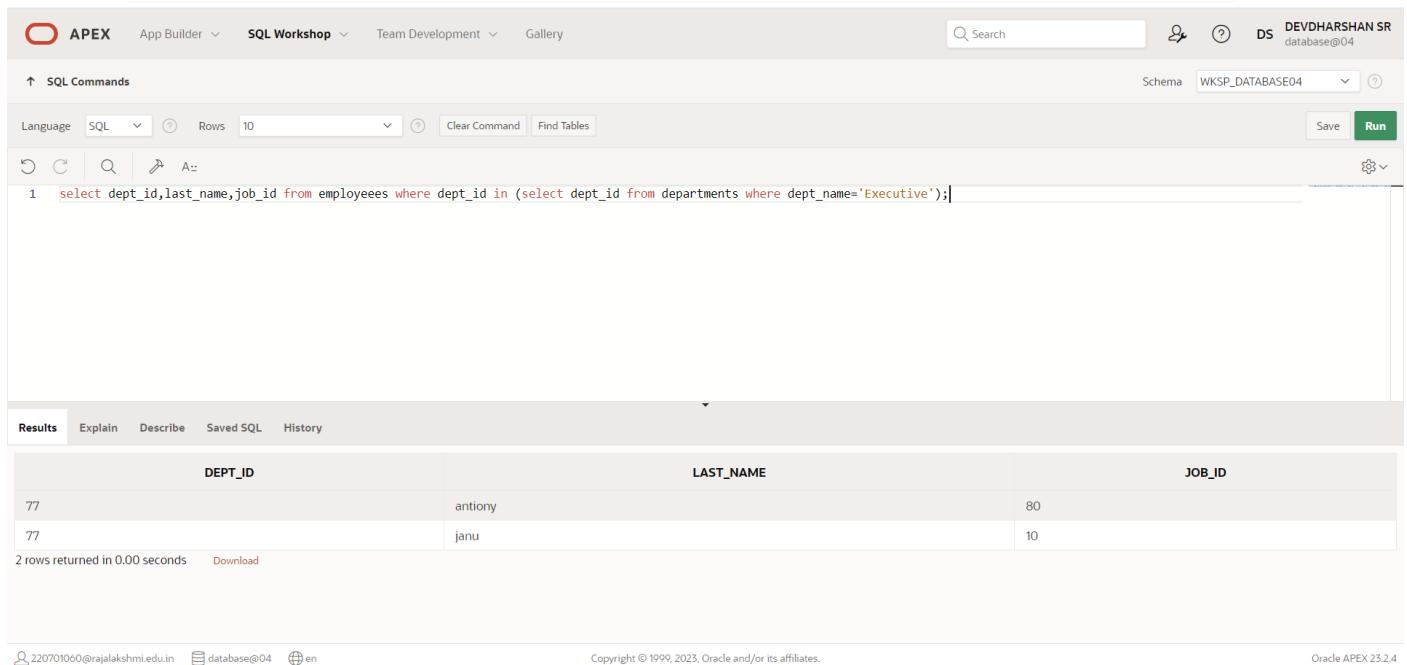
The results tab shows the message 'no data found'.

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

QUERY:

```
select 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 top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows 'DS DEVDHARSHAN SR database@O4'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is:

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

The results tab displays a table with three columns: DEPT_ID, LAST_NAME, and JOB_ID. The data is:

DEPT_ID	LAST_NAME	JOB_ID
77	antony	80
77	janu	10

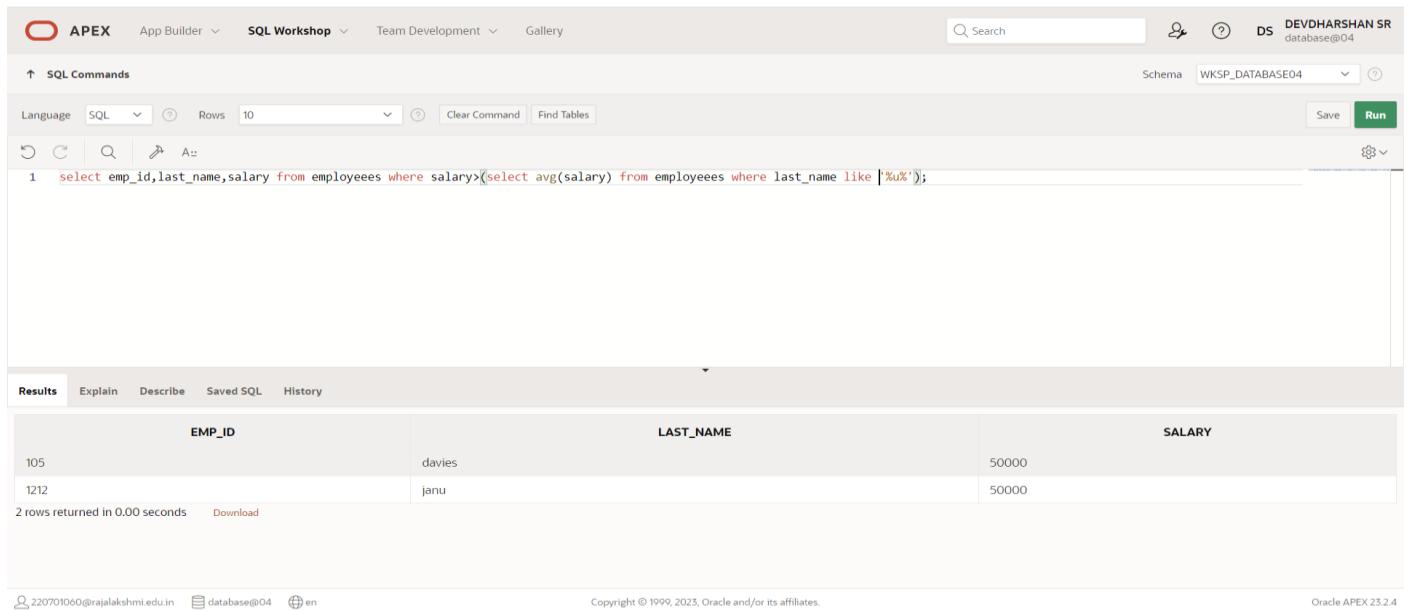
Below the table, it says '2 rows returned in 0.00 seconds' and has a 'Download' link. The bottom footer includes user information and copyright details.

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 top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows a user profile for 'DEVDHARSHAN SR' connected to 'database@i04'. The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, displaying the query: 'select emp_id,last_name,salary from employees where salary>(select avg(salary) from employees where last_name like '%u%');'. Below the query, the Results tab is selected, showing a table with three columns: EMP_ID, LAST_NAME, and SALARY. The data returned is:

EMP_ID	LAST_NAME	SALARY
105	davies	50000
1212	janu	50000

Below the table, it says '2 rows returned in 0.00 seconds' and there is a 'Download' link. The bottom of the page shows the URL '220701060@rajalakshmi.edu.in', the database connection 'database@i04', and the language 'en'. It also includes copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the software version 'Oracle APEX 25.2.4'.

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

RESULT:

USING THE SET OPERATORS

EX_NO:10

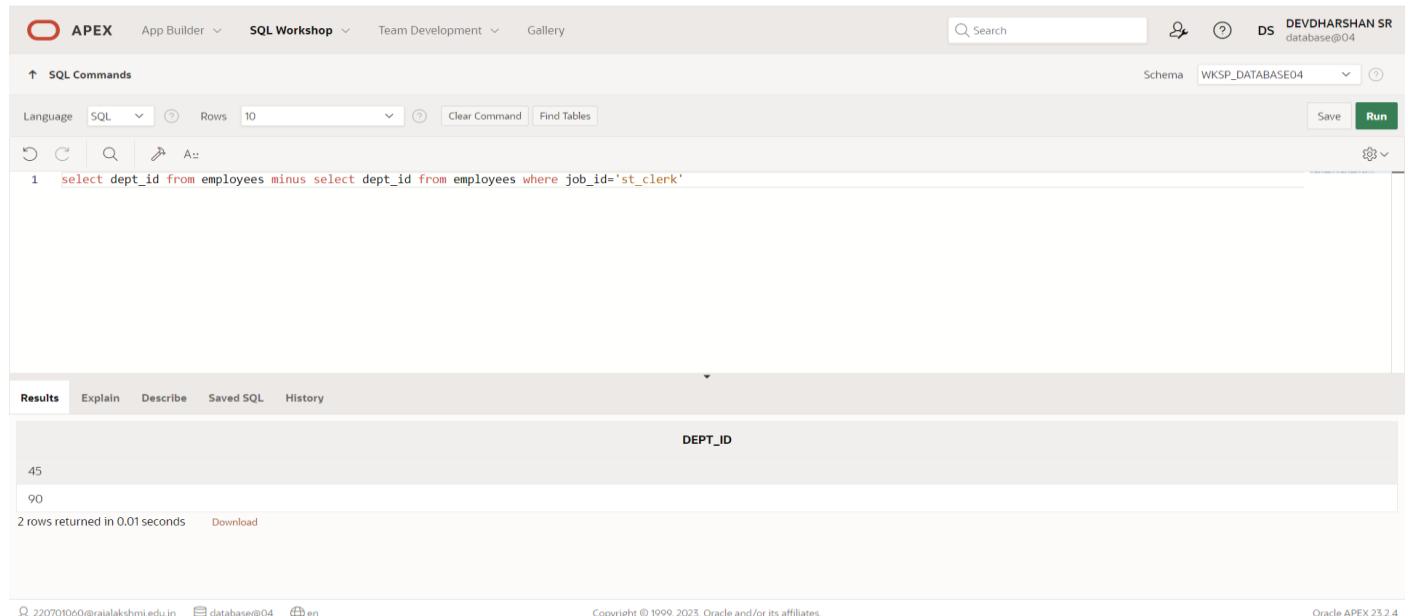
DATE:

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 top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows 'DS DEVDHARSHAN SR database@04'. The main area has a 'Search' bar and a toolbar with 'Save' and 'Run' buttons. 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 shows the output:

DEPT_ID
45
90

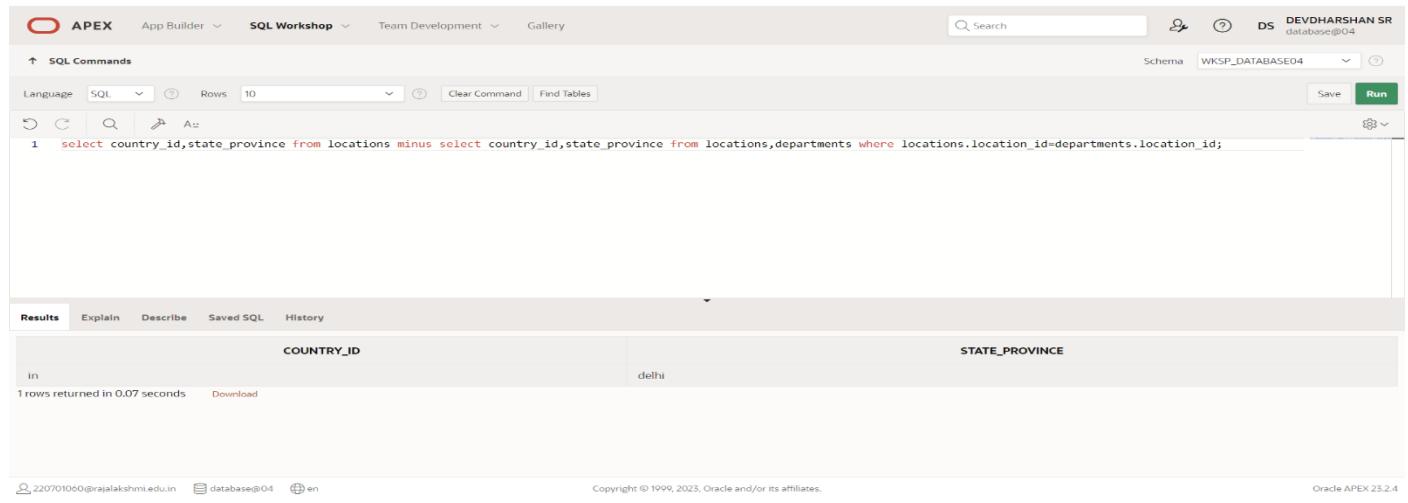
Below the results, it says '2 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom, it shows the user '220701060@rajalakshmi.edu.in', the schema 'database@04', and the language 'en'. Copyright information and the version 'Oracle APEX 23.2.4' are also present.

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 top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows 'DS DEVDHARSHAN SR database@04'. The main area has a 'Search' bar and a toolbar with 'Save' and 'Run' buttons. 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 shows the output:

COUNTRY_ID	STATE_PROVINCE
in	delhi

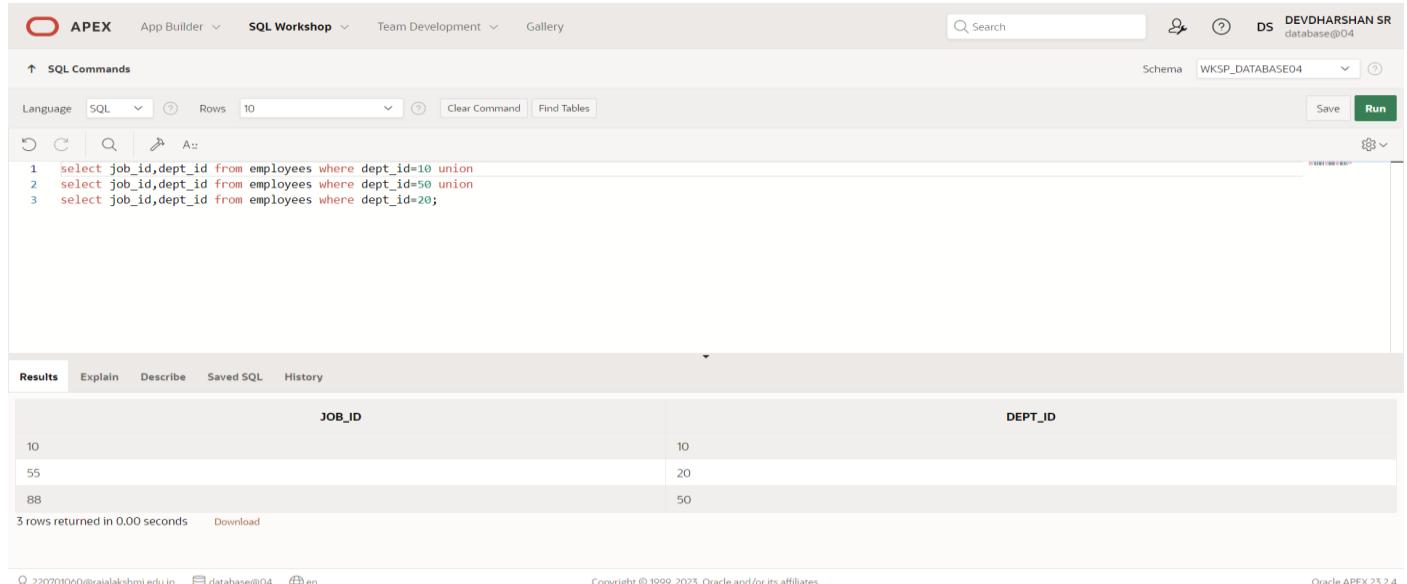
Below the results, it says '1 rows returned in 0.07 seconds' and has a 'Download' link. At the bottom, it shows the user '220701060@rajalakshmi.edu.in', the schema 'database@04', and the language 'en'. Copyright information and the version 'Oracle APEX 23.2.4' are also present.

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 columns 'JOB_ID' and 'DEPT_ID'. The data is as follows:

JOB_ID	DEPT_ID
10	10
55	20
88	50

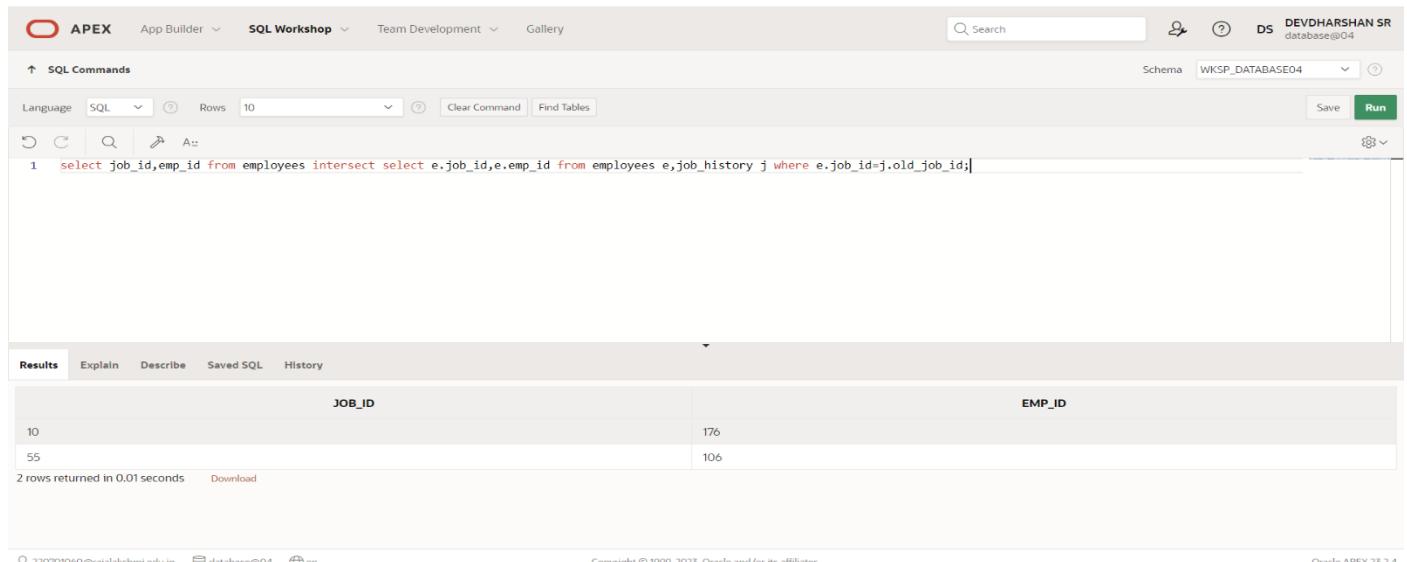
3 rows returned in 0.00 seconds

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. The results are displayed in a table with columns 'JOB_ID' and 'EMP_ID'. The data is as follows:

JOB_ID	EMP_ID
10	176
55	106

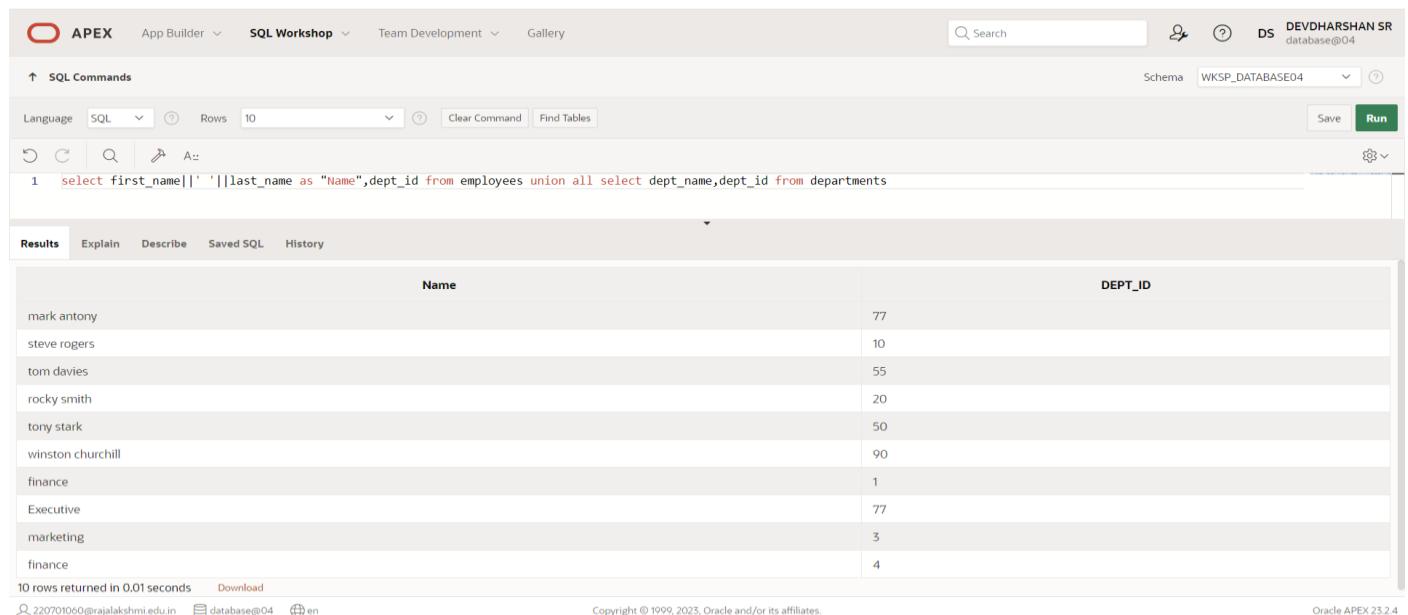
2 rows returned in 0.01 seconds

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. The right side shows the schema as 'DEVDHARSHAN SR database@O4'. The main area displays the SQL command:

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

The results section shows the output of the query:

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

Below the table, it says '10 rows returned in 0.01 seconds'. The bottom of the page includes links for Download, 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:

CREATING VIEWS

EX_NO:11

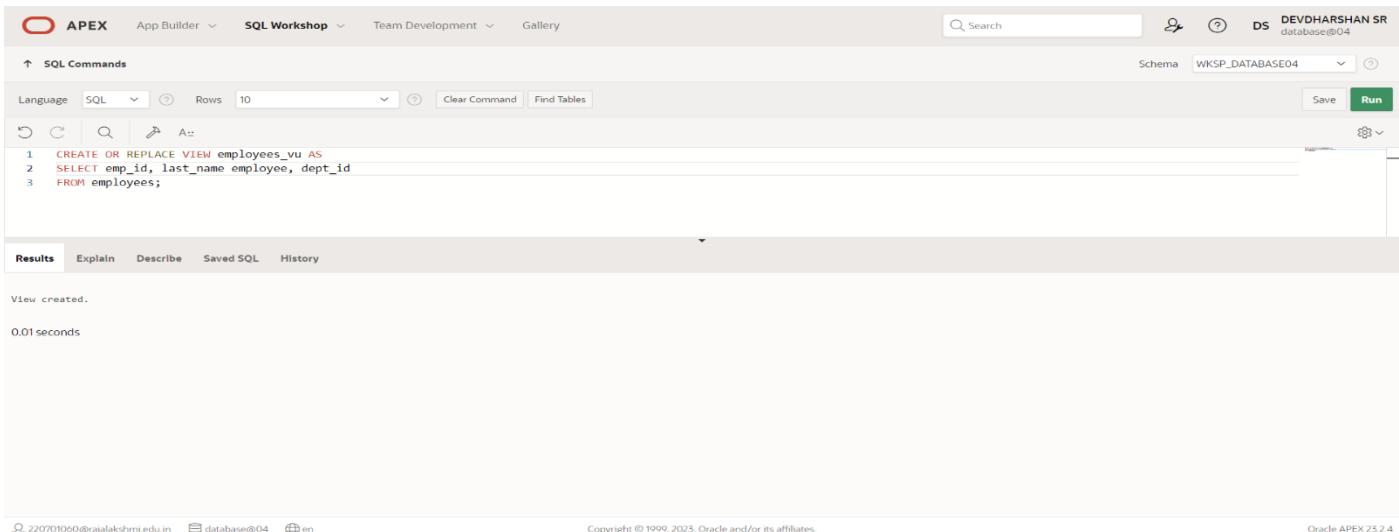
DATE:

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. Under the 'SQL Workshop' dropdown, 'SQL Commands' is chosen. The main area contains the following SQL code:

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

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

```
View created.
0.01 seconds
```

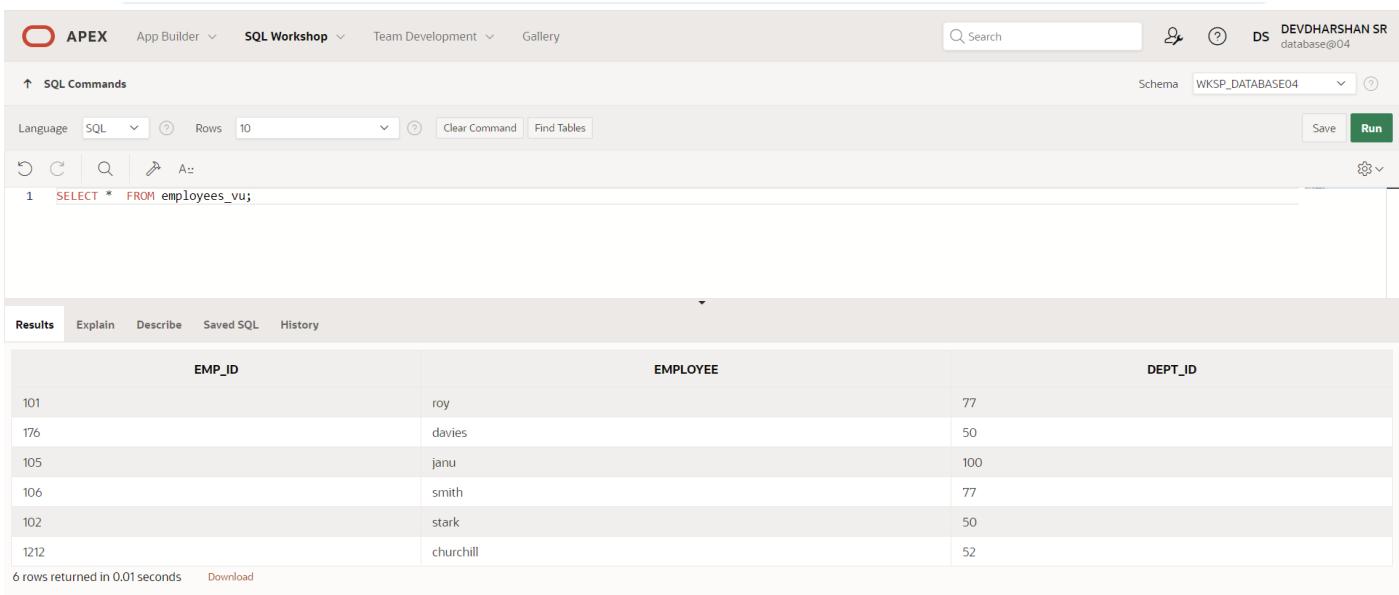
At the bottom of the page, the footer includes the URL '220701060@rajalakshmi.edu.in', the database name 'database@04', and the language 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 25.2.4'.

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. Under the 'SQL Workshop' dropdown, 'SQL Commands' is chosen. The main area contains the following SQL code:

```
1 SELECT * FROM employees_vu;
```

Below the code, the 'Results' tab is selected. The output displays a table with three columns: 'EMP_ID', 'EMPLOYEE', and 'DEPT_ID'. The data is as follows:

EMP_ID	EMPLOYEE	DEPT_ID
101	roy	77
176	davies	50
105	janu	100
106	smith	77
102	stark	50
1212	churchill	52

At the bottom left, it says '6 rows returned in 0.01 seconds'. At the bottom right, there is a 'Download' button. The footer includes the URL '220701060@rajalakshmi.edu.in', the database name 'database@04', and the language 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 25.2.4'.

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 SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area has a 'SQL Commands' tab open. The SQL command entered is:

```
1 select view_name, text
2 from user_views;
```

In the results section, there is one row returned:

VIEW_NAME	TEXT
EMPLOYEES_VU	SELECT emp_id, last_name employee, dept_id FROM employees

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

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 SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area has a 'SQL Commands' tab open. The SQL command entered is:

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

In the results section, there are six rows returned:

EMPLOYEE	DEPT_ID
roy	77
davies	50
janu	100
smith	77
stark	50
churchill	52

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

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

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

Below the code, the results show:

View created.
0.07 seconds

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

QUERY:

```
Describe dept50;
```

OUTPUT:

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

```
1 DESCRIBE dept50 ;
```

Below the code, the results show the description of the view:

Object Type: VIEW
Object: DEPT50

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

7.) Attempt to reassign Matos to department 80

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows a user profile for DEVDHARSHAN SR and the schema WKSP_DATABASE04. The main area is titled "SQL Commands" and contains the following SQL code:

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

Below the code, the "Results" tab is selected, showing the output: "0 row(s) updated." and "0.04 seconds".

8.) Create a view called SALARY_VU based on the employee last names, department names, salaries, and salary grades for all employees. Use the Employees, DEPARTMENTS and JOB_GRADE tables. Label the column Employee, Department, salary, and Grade respectively.

QUERY:

```
create or replace view salary_vu as select e.last_name "Employee",d.dept_name Department,
e.salary "Salary",j.grade_level "Grades" from employees e,departments d,job_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 SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows a user profile for DEVDHARSHAN SR and the schema WKSP_DATABASE04. The main area is titled "SQL Commands" and contains the following SQL code for creating a view:

```
1 CREATE OR REPLACE VIEW salary_vu
2 AS
3 SELECT e.last_name "Employee",
4       d.dept_name "Department",
5       e.salary "Salary",
6       j.grade_level "Grades"
7  FROM employees e,
8       departments d,
9       job_grades j
10 WHERE e.dept_id = d.dept_id
11   AND e.salary BETWEEN j.low_sal and j.high_sal;
```

Below the code, the "Results" tab is selected, showing the output: "View created." and "0.00 seconds".

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		
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

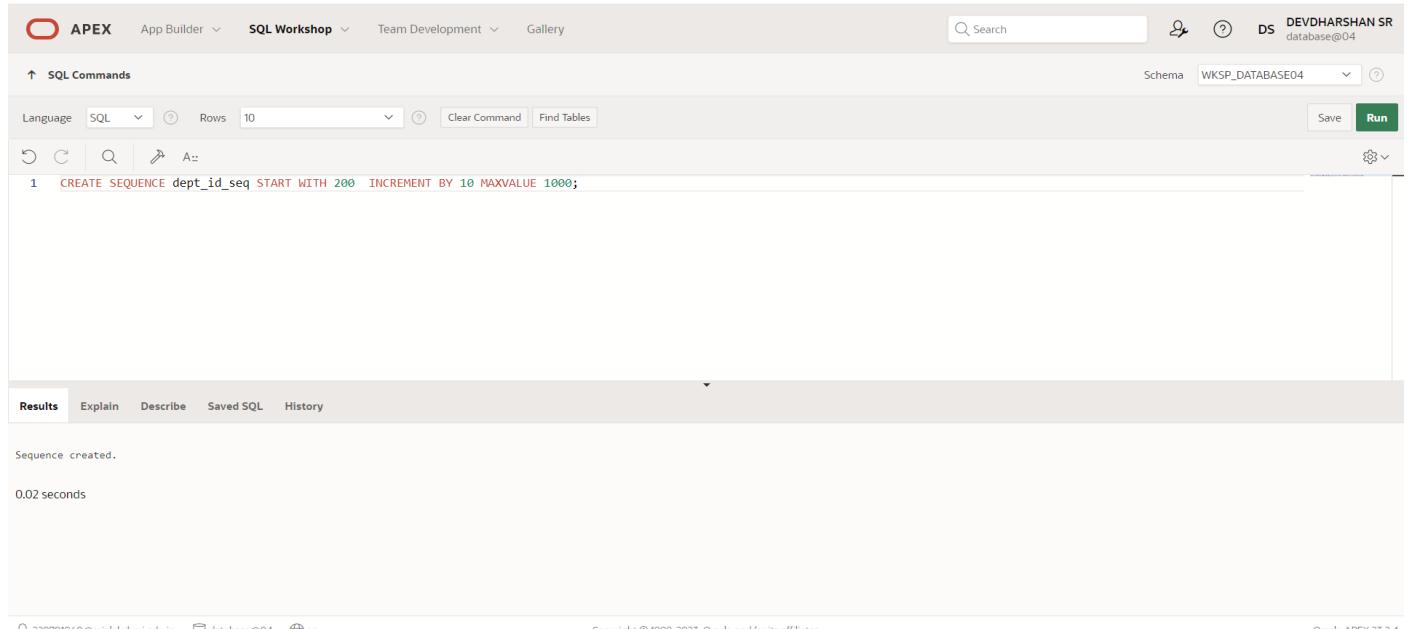
DATE:

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:



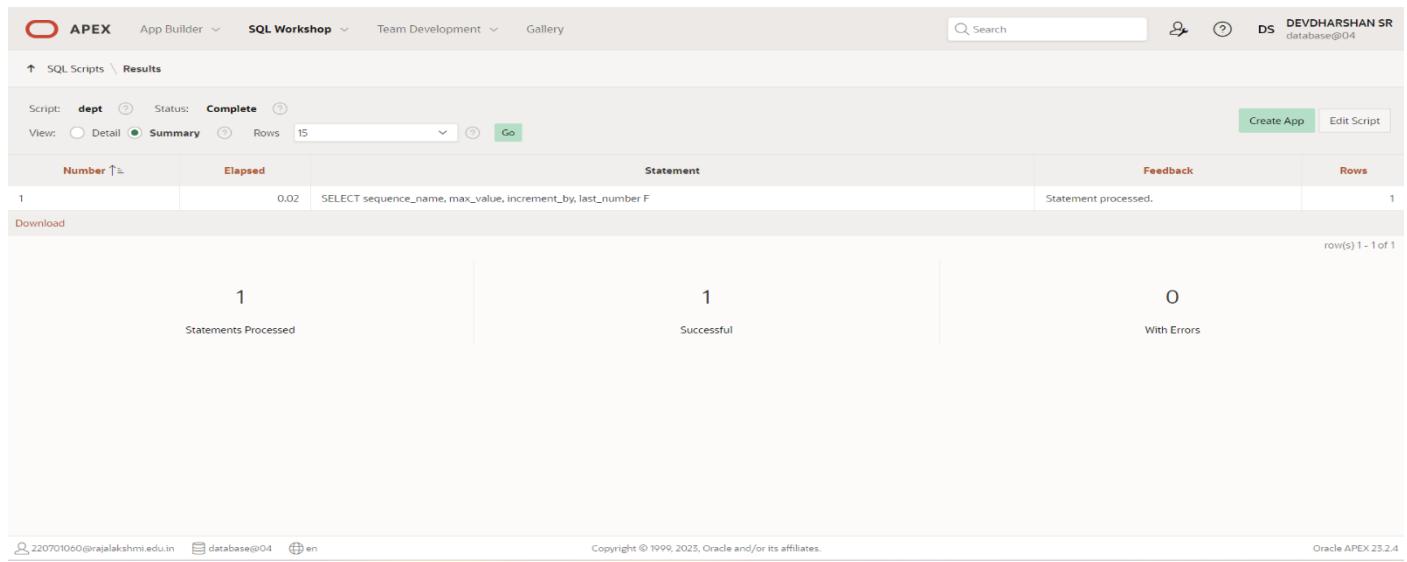
The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The main area contains the SQL command: 'CREATE SEQUENCE dept_id_seq START WITH 200 INCREMENT BY 10 MAXVALUE 1000;'. Below the command, the output shows: 'Sequence created.' and '0.02 seconds'. The bottom status bar includes the user information '220701060@rajalakshmi.edu.in database@04 en' and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.'

2.) Write a query in a script to display the following information about your sequences: sequence name, maximum value, increment size, and last number

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The main area shows the results of the query: 'SELECT sequence_name, max_value, increment_by, last_number F'. The results table has columns: Number, Elapsed, Statement, Feedback, and Rows. The data row shows: Number 1, Elapsed 0.02, Statement 'SELECT sequence_name, max_value, increment_by, last_number F', Feedback 'Statement processed.', and Rows 1. The bottom status bar includes the user information '220701060@rajalakshmi.edu.in database@04 en' and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.'

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 (selected), Team Development, and Gallery. On the right, there are search, help, and user profile icons. The main area displays the results of a script run. The 'Script' dropdown shows 'lab12_3' and the status is 'Complete'. The 'View' dropdown has 'Summary' selected. The results table has columns for Number, Elapsed, Statement, Feedback, and Rows. One row is listed: '1' in Number, '0.03' in Elapsed, 'INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education')' in Statement, '1 row(s) inserted.' in Feedback, and '1' in Rows. Below the table, it says 'Download' and 'row(s) 1 - 1 of 1'. At the bottom, it shows 'Statements Processed' (1), 'Successful' (1), and 'With Errors' (0). The footer includes copyright information for Oracle and the APEX version (23.2.4).

4.) Create a nonunique index on the foreign key column (DEPT_ID) in the EMP table.

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Commands interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there are search, help, and user profile icons. The main area shows a command line with the schema 'WKSP_DATABASE04'. The command entered is 'CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);'. Below the command line, the results tab is selected, showing the message 'Index created.' and a time of '0.04 seconds'. The footer includes copyright information for Oracle and the APEX version (23.2.4).

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows the user information: DEVDHARSHAN SR database@04. The main area is titled "SQL Commands" with tabs for Language (set to SQL), Rows (set to 10), and a "Run" button. The SQL command entered is: `SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';`. Below the command, the results tab is selected, showing a table with three columns: INDEX_NAME, TABLE_NAME, and UNIQUENESS. The single row returned is: INDEX_NAME is EMP_DEPT_ID_IDX, TABLE_NAME is EMPLOYEES, and UNIQUENESS is NONUNIQUE. The status message at the bottom indicates 1 row returned in 0.04 seconds with a "Download" link.

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

RESULT:

CONTROLLING USER ACCESS

EX_NO:15

DATE:

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

The CREATE SESSION system privilege

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

The CREATE TABLE privilege

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

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

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

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

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

The ALTER USER statement

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

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

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

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

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

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

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

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

SELECT table_name FROM user_tables;

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

Team 1 revokes the privilege.

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

Team 2 revokes the privilege.

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

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

Team 1 executes this INSERT statement.

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

Team 2 executes this INSERT statement.

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

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

RESULT:

PL/SQL

CONTROL STRUCTURES

EX_NO:16

DATE:

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, along with 'SQL Workshop'. The schema is set to 'WKSP_DATABASE04'. The main area is titled 'SQL Commands' and contains the following PL/SQL code:

```
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, the 'Results' tab is active, showing the output of the query:

```
Incentive  = 4800
Statement processed.

0.02 seconds
```

At the bottom of the page, there are footer links: '220701060@rajalakshmi.edu.in', 'database@04', 'en', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, the schema is set to 'DEVDHARSHAN SR database@04'. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. Below this, there are buttons for 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. The SQL editor contains the following code:

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

Below the editor, the 'Results' tab is selected. It displays the error message:

```
Error at line 4/23: ORA-06550: line 4, column 23:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.0MW_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored
```

Following the error message, the remaining code is listed:

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

At the bottom of the results pane, the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are visible.

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

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

The 'Results' tab is selected, displaying the same error message as the previous screenshot:

```
Error at line 4/23: ORA-06550: line 4, column 23:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.0MW_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored
```

Following the error message, the remaining code is listed:

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

At the bottom of the results pane, the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are visible.

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

QUERY:

DECLARE

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

```
PROCEDURE approx_salary (
```

```
    emp      NUMBER,
```

```
    empsal IN OUT NUMBER,
```

```
    addless   NUMBER
```

```
) IS
```

```
BEGIN
```

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

```
END;
```

BEGIN

```
SELECT salary INTO salary_of_emp
```

```
FROM employees
```

```
WHERE employee_id = 122;
```

```
DBMS_OUTPUT.PUT_LINE
```

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

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

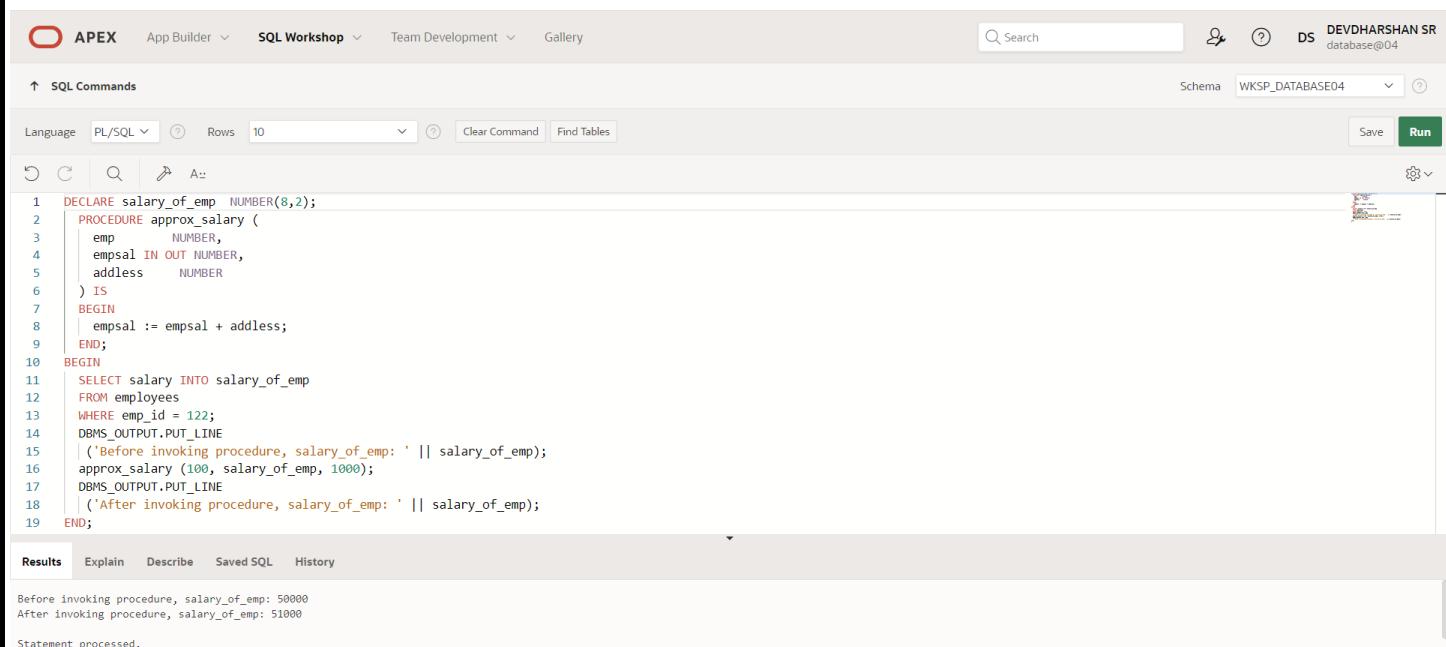
```
DBMS_OUTPUT.PUT_LINE
```

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

```
END;
```

```
/
```

OUTPUT:



```
APEX App Builder SQL Workshop Team Development Gallery DS DEVDHARSHAN SR database@04
↑ SQL Commands Schema WKSP_DATABASE04 Run
Language PL/SQL Rows 10 Clear Command Find Tables
SQL Commands
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 BEGIN
11     SELECT salary INTO salary_of_emp
12     FROM employees
13     WHERE emp_id = 122;
14     DBMS_OUTPUT.PUT_LINE
15     ('Before invoking procedure, salary_of_emp: ' || salary_of_emp);
16     approx_salary (100, salary_of_emp, 1000);
17     DBMS_OUTPUT.PUT_LINE
18     ('After invoking procedure, salary_of_emp: ' || salary_of_emp);
19 END;
Results Explain Describe Saved SQL History
Before invoking procedure, salary_of_emp: 50000
After invoking procedure, salary_of_emp: 51000
Statement processed.
220701060@rajalakshmi.edu.in database@04 en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 25.2.4
```

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, along with 'SQL Workshop'. The schema is set to 'WKSP_DATABASE04'. The main area is titled 'SQL Commands' and contains the PL/SQL code for the 'pri_bool' procedure. The code is numbered from 1 to 15. The 'Run' button is visible at the top right of the command input field. Below the command input, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is currently selected. The output section displays the message 'Procedure created.' and a execution time of '0.03 seconds'. At the bottom, the footer includes copyright information for Oracle and the APEX version '23.2.4'.

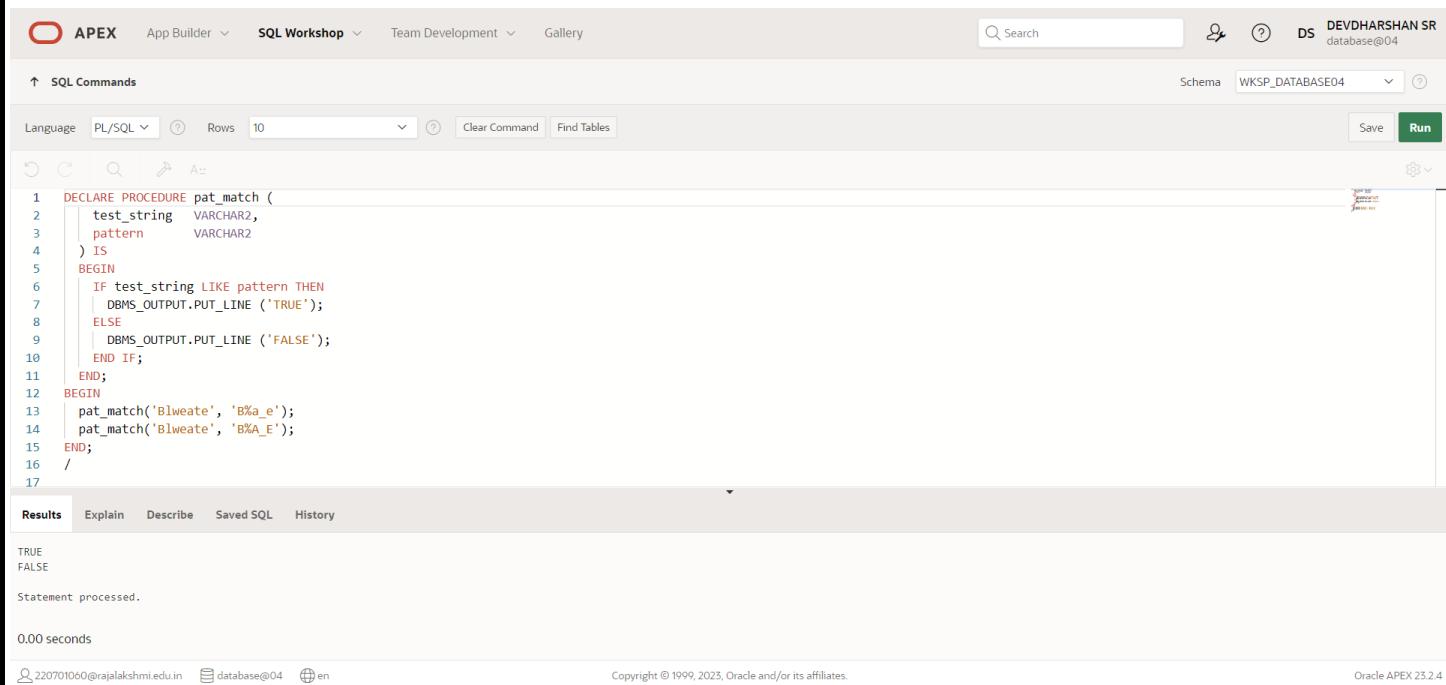
```
1 CREATE OR REPLACE PROCEDURE pri_bool(
2   boo_name  VARCHAR2,
3   boo_val   BOOLEAN
4 ) IS
5 BEGIN
6   IF boo_val IS NULL THEN
7     DBMS_OUTPUT.PUT_LINE( boo_name || ' = NULL');
8   ELSIF boo_val = TRUE THEN
9     DBMS_OUTPUT.PUT_LINE( boo_name || ' = TRUE');
10  ELSE
11    DBMS_OUTPUT.PUT_LINE( boo_name || ' = FALSE');
12  END IF;
13 END;
14 /
15
```

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 right side of the header shows the user 'DEVDHARSHAN SR' and the schema 'WKSP_DATABASE04'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code from the previous section. The code is syntax-highlighted, with 'DECLARE' in blue, 'PROCEDURE' in red, and 'IF', 'THEN', 'ELSE', 'DBMS_OUTPUT.PUT_LINE' in black. The output pane at the bottom shows the results of the execution: 'TRUE' and 'FALSE' on separate lines, followed by the message 'Statement processed.' and '0.00 seconds' execution time. The bottom status bar displays the user '220701060@rajalakshmi.edu.in', the database 'database@04', and the language 'en'. The copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also visible.

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

Results Explain Describe Saved SQL History

TRUE
FALSE
Statement processed.
0.00 seconds

220701060@rajalakshmi.edu.in database@04 en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 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', 'Team Development', and 'Gallery'. The right side shows the connection details 'DS DEVDHARSHAN SR database@04'. The main area is titled 'SQL Commands' with a 'PL/SQL' dropdown. The code area contains a numbered PL/SQL block. The results tab at the bottom shows the output of the executed code.

```
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
```

Footer: 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;
```

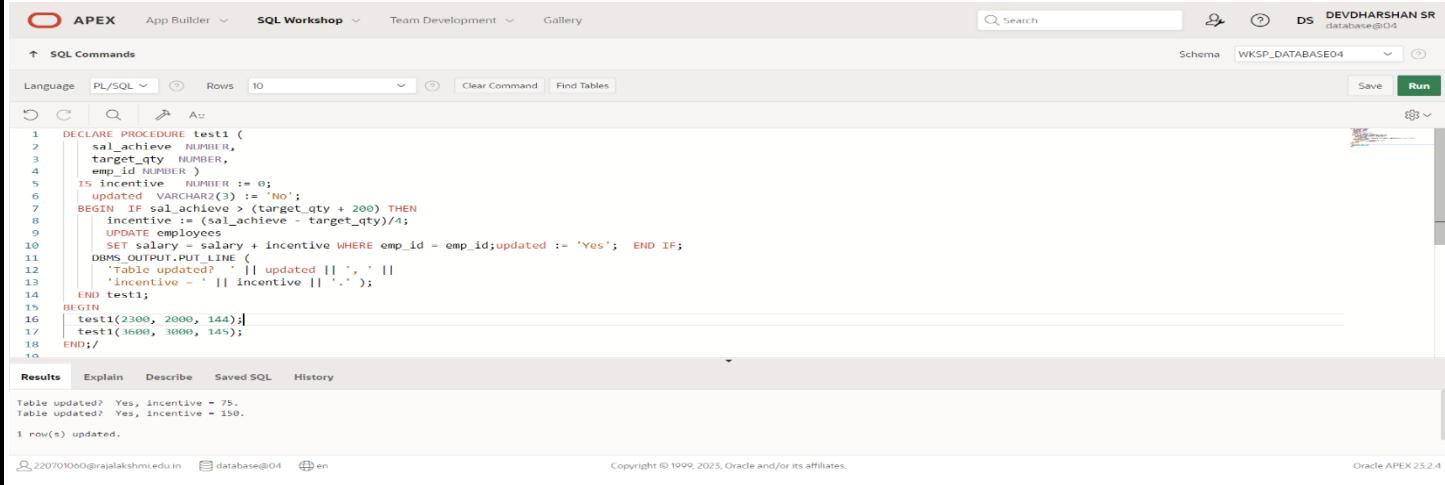
BEGIN

```
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' (selected), 'Team Development', and 'Gallery'. The right side shows the schema 'DEVDHARSHAN SR' and workspace 'WKSP_DATABASE04'. The main area displays the PL/SQL code for the 'test1' procedure and its execution. The code is as follows:

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

The results section shows the output of the executed code:

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

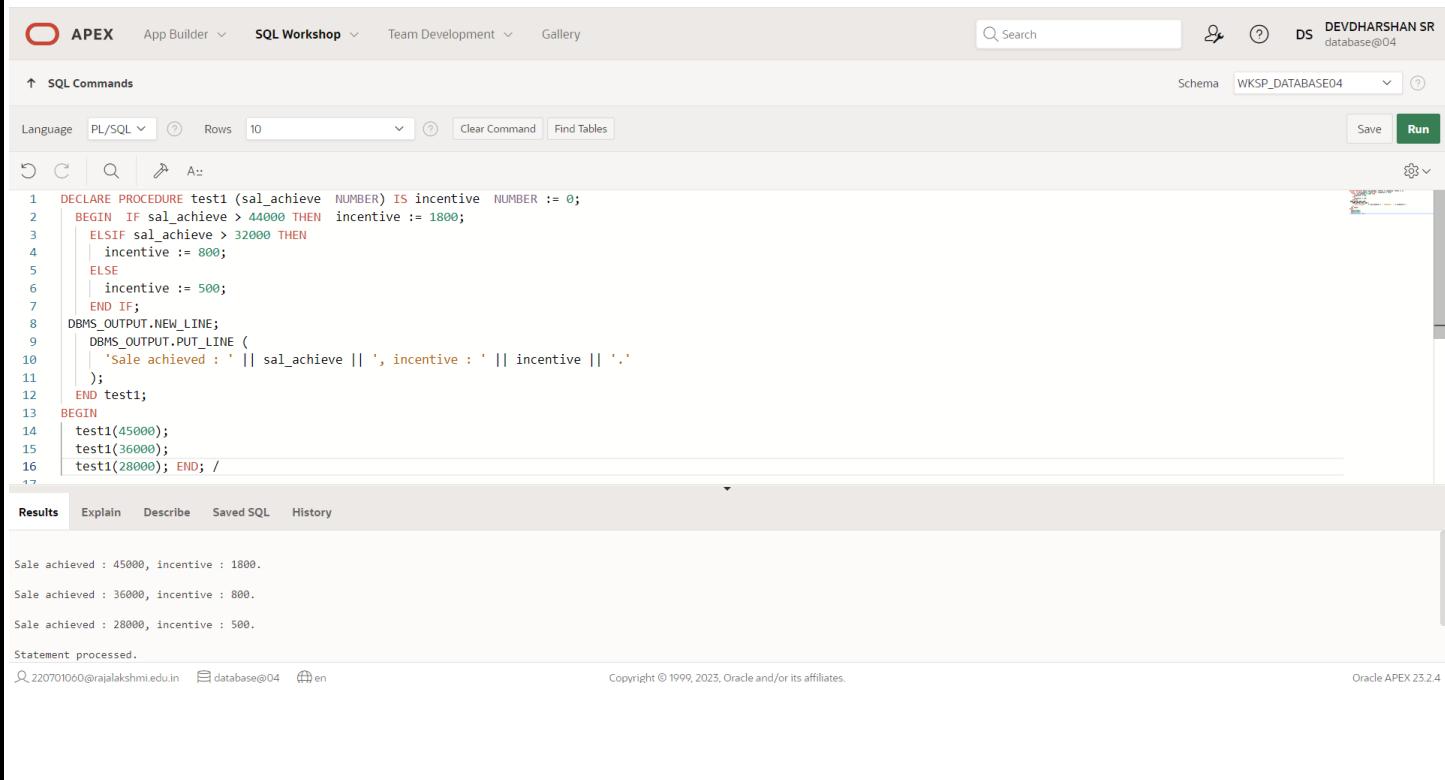
At the bottom, the footer includes the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

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 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 connection information: DS DEVDHARSHAN SR database@04. The main workspace is titled "SQL Commands". The query editor contains the provided PL/SQL code. The results pane at the bottom displays the output of the executed code, showing three rows of results: "Sale achieved : 45000, incentive : 1800.", "Sale achieved : 36000, incentive : 800.", and "Sale achieved : 28000, incentive : 500.". The status bar at the bottom indicates "Statement processed." and shows the user's email and the database connection.

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

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

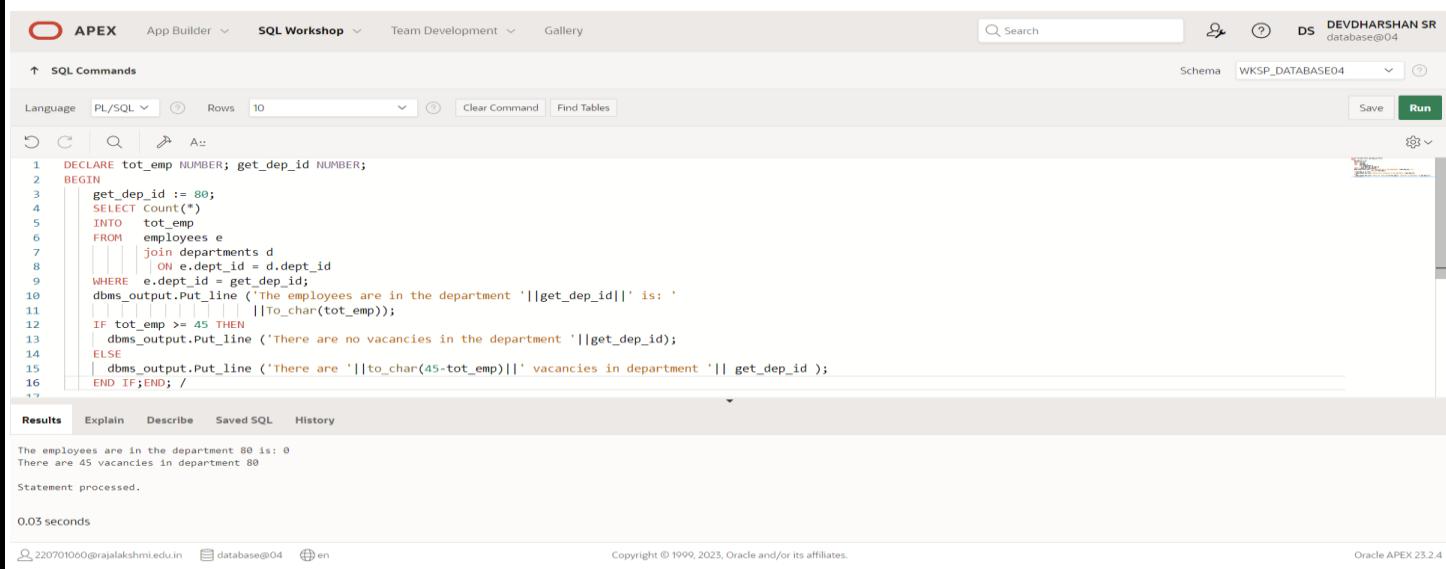
9.) Write a PL/SQL program to count number of employees in department 50 and check whether this department have any vacancies or not. There are 45 vacancies in this department.

QUERY:

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

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

OUTPUT:



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_DATABASE04'. The main area displays the PL/SQL code. The results tab at the bottom shows the output of the executed code:

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

Statement processed.

0.03 seconds
```

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

QUERY:

DECLARE

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

BEGIN

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

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

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

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

```
ELSE
```

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

```
END IF;
```

```
END;
```

```
/
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', 'Gallery', 'Search', and 'DS DEVDHARSHAN SR database@04'. 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_DATABASE04), 'Save', and 'Run'. The code area contains a numbered PL/SQL block. The output pane at the bottom shows the results of the execution: 'The employees are in the department 80 is: 0' and 'There are 45 vacancies in department 80'. The status bar at the bottom 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
15 /
16
```

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

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:



```
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 emp_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;
12 CLOSE c_employees; END;/
```

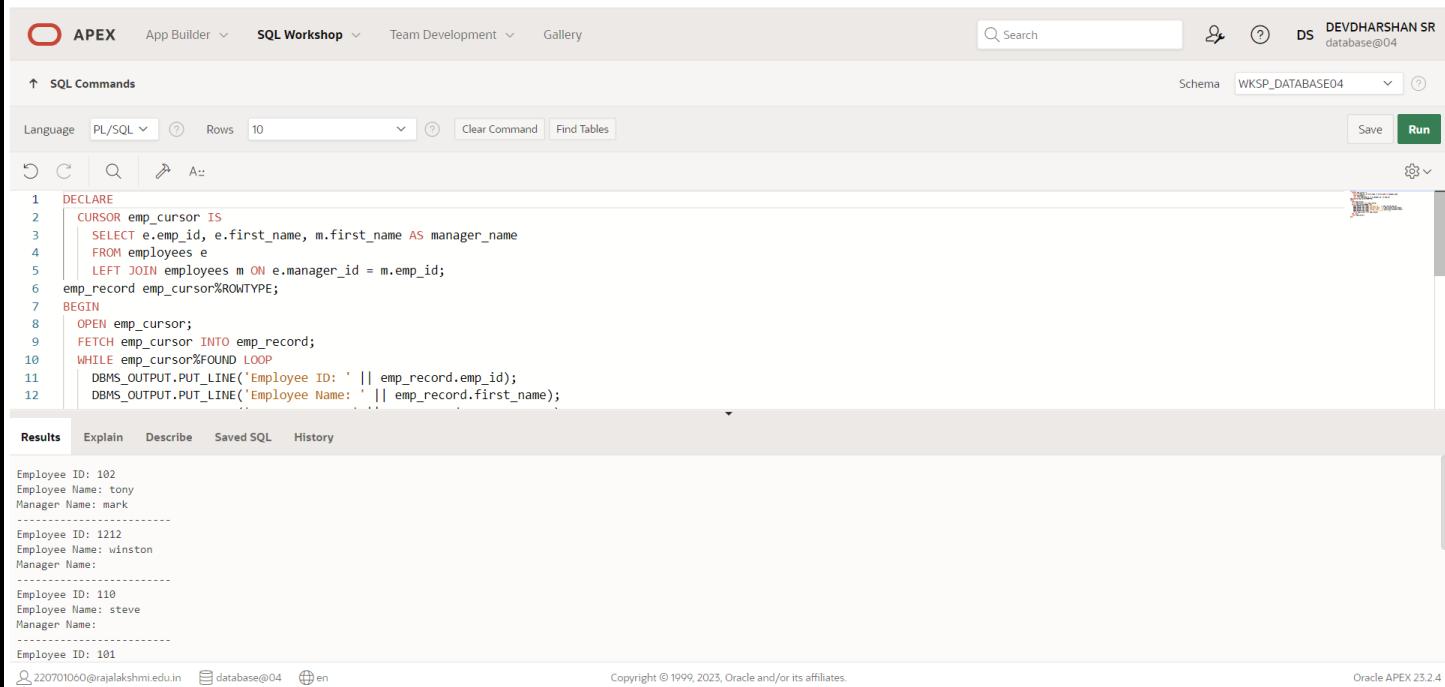
Employee ID	Full Name	Job Title	Hire Date	Salary
101	mark roy	st_clerk	02/03/1998	10900
110	steve davis	10	02/05/1999	40900
122	tom janu	st_charles	02/05/2000	50900
106	rocky smith	55	03/25/1996	25900
102	tony stark	88	05/01/1994	20900
1212	winston churchill	22	01/01/2001	50900

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 SQL Workshop tab is selected. The schema dropdown shows 'WKSP_DATABASE04'. The main area displays the PL/SQL code for the cursor and loop, with line numbers 1 through 12. Below the code, the 'Results' tab is active, showing the output of the program. The output consists of several lines of text, each starting with 'Employee ID:' followed by a value like 102, 1212, 110, or 101, and ending with 'Manager Name:' followed by a name like tony, mark, winston, or steve. There are also separator lines consisting of four dashes ('-----'). At the bottom of the results pane, there is footer information including the user's email (220701060@rajalakshmi.edu.in), the database connection (database@04), and the session identifier (en). The copyright notice at the bottom reads 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

```
1  DECLARE
2    CURSOR emp_cursor IS
3      SELECT e.emp_id, e.first_name, m.first_name AS manager_name
4      FROM employees e
5      LEFT JOIN employees m ON e.manager_id = m.emp_id;
6    emp_record emp_cursor%ROWTYPE;
7  BEGIN
8    OPEN emp_cursor;
9    FETCH emp_cursor INTO emp_record;
10   WHILE emp_cursor%FOUND LOOP
11     DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.emp_id);
12     DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_record.first_name);
```

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a connection to 'DEVDHARSHAN SR database@04'. The main area has tabs for 'SQL Commands' (selected), 'Results' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'SQL Commands' tab contains the PL/SQL code from the previous block. The 'Results' tab displays the output of the code execution, listing job IDs and their minimum salaries.

Job ID	Job Title	Min. Salary
st_clerk	Sales Representative	10000
10	Sales Representative	10000
st_charles	Sales Representative	10000
55	Sales Representative	10000
88	Sales Representative	10000
22	Sales Representative	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', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is selected. The main area displays the PL/SQL code for the query. Below the code, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing the output of the query. The output table has columns: Employee ID, Last Name, Job Id, and Start Date. The data is as follows:

Employee ID	Last Name	Job Id	Start Date
101	Alonso	IT_PROG	2019-01-01
102	Bates	SA_REP	2019-01-01
103	Carter	SA_REP	2019-01-01
104	Dodger	SA_REP	2019-01-01
105	Ford	SA_REP	2019-01-01
106	Garcia	SA_REP	2019-01-01
107	Haze	SA_REP	2019-01-01
108	Kingsley	SA_REP	2019-01-01
109	Leverett	SA_REP	2019-01-01
110	Martinez	SA_REP	2019-01-01
111	Plum	SA_REP	2019-01-01
112	Randall	SA_REP	2019-01-01
113	Solomon	SA_REP	2019-01-01
114	Tucker	SA_REP	2019-01-01
115	Waterson	SA_REP	2019-01-01
116	Zlot	SA_REP	2019-01-01

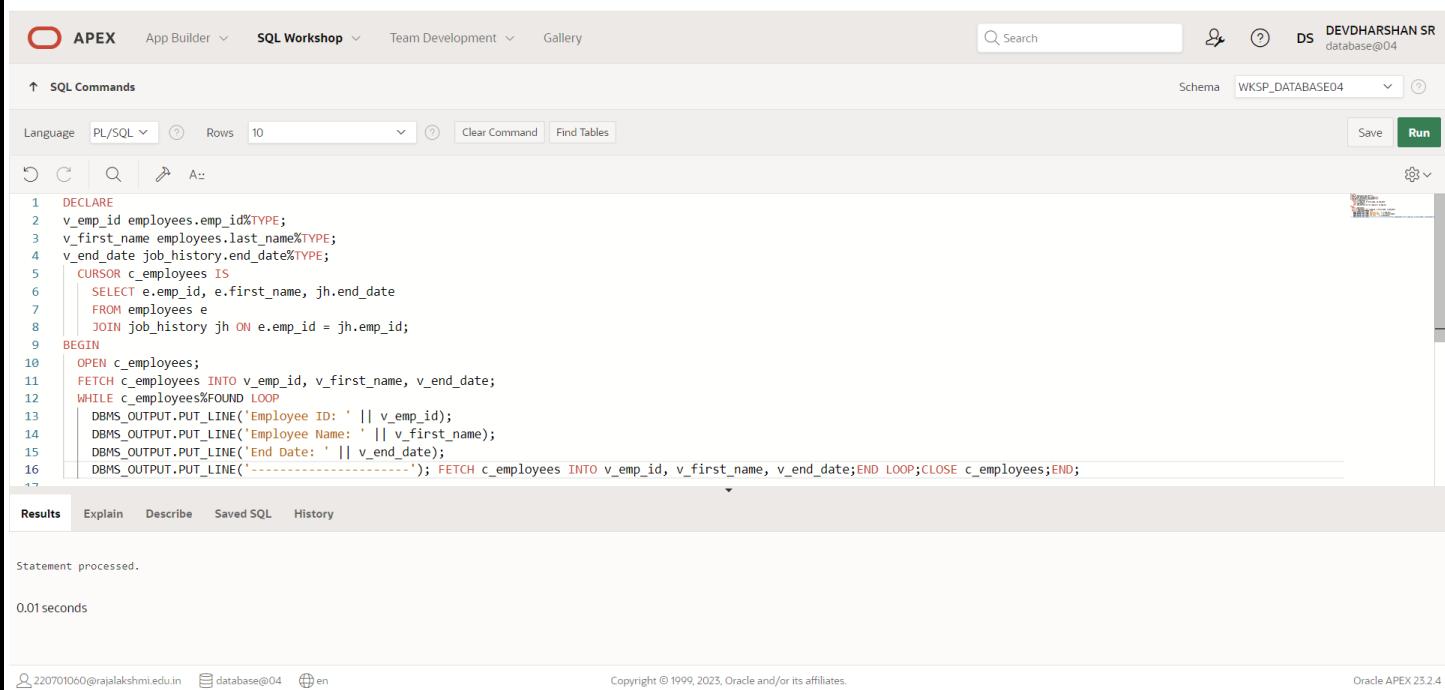
Below the table, a message states 'Statement processed.' and '0.02 seconds'.

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 links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the connection information: DS DEVDHARSHAN SR database@04. The main workspace is titled "SQL Commands". The command input area contains the provided PL/SQL code. The "Run" button is visible at the top right of the command input. Below the command input, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is selected, displaying the output: "Statement processed." and "0.01 seconds". At the bottom, the footer includes user information (220701060@rajalakshmi.edu.in), a database link (database@04), and copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates). The Oracle APEX version is also mentioned as 23.2.4.

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

RESULT:

PROCEDURES AND FUNCTIONS

EX_NO: 17

DATE:

1.) Factorial of a number using function.

QUERY:

DECLARE

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

BEGIN

```
    WHILE n > 0 LOOP
```

```
        fac := n * fac;
```

```
        n := n - 1;
```

```
    END LOOP;
```

```
    DBMS_OUTPUT.PUT_LINE(fac);
```

END;

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the user DEVDHARSHAN SR and the schema WKSP_DATABASE04. The main area is titled 'SQL Commands' and contains the following PL/SQL code:

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

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

```
120
Statement processed.
0.01 seconds
```

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

QUERY:

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

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

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

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

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

OUTPUT:

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

```
1 CREATE OR REPLACE PROCEDURE get_book_info (
2     p_book_id IN NUMBER,
3     p_title IN OUT VARCHAR2,
4     p_author OUT VARCHAR2,
5     p_year_published OUT NUMBER
6 )AS BEGIN
7     SELECT title, author, year_published INTO p_title, p_author, p_year_published
8     FROM books
9     WHERE book_id = p_book_id;
10    p_title := p_title || ' - Retrieved';
11    EXCEPTION WHEN NO_DATA_FOUND THEN p_title := NULL; p_author := NULL; p_year_published := NULL;
12 END;
```

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

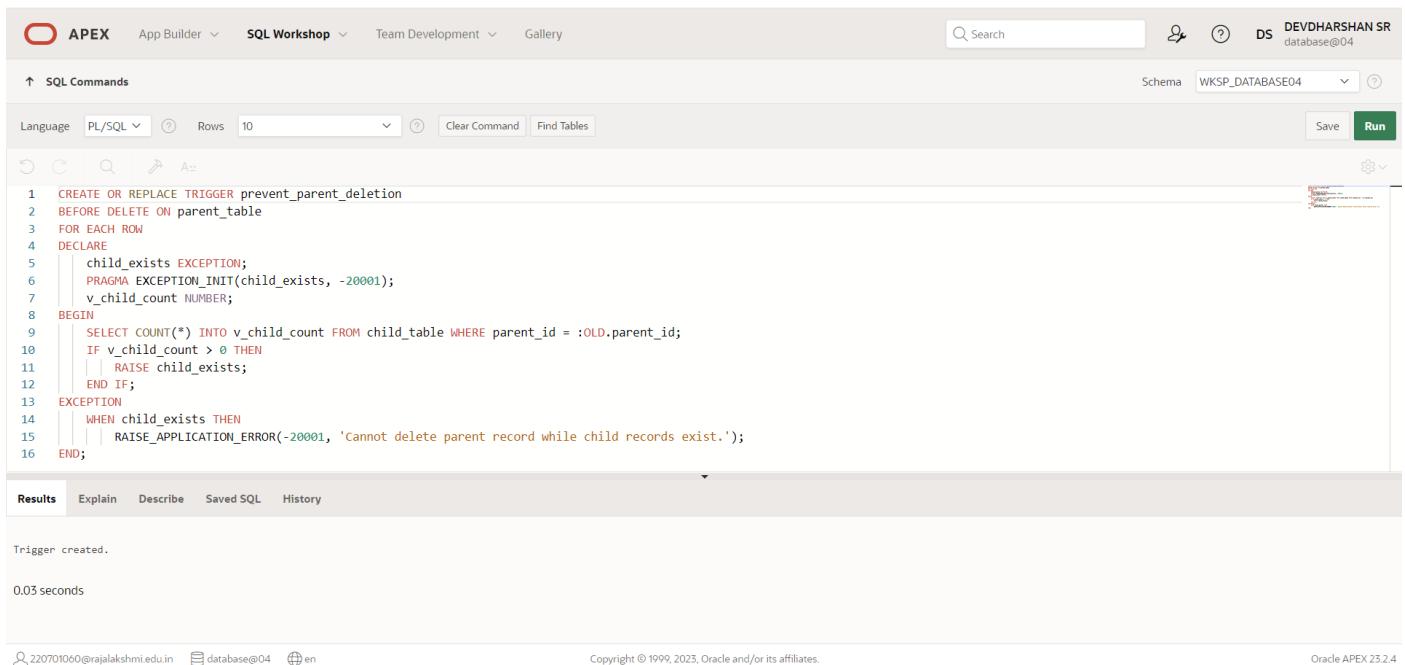
DATE:

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:



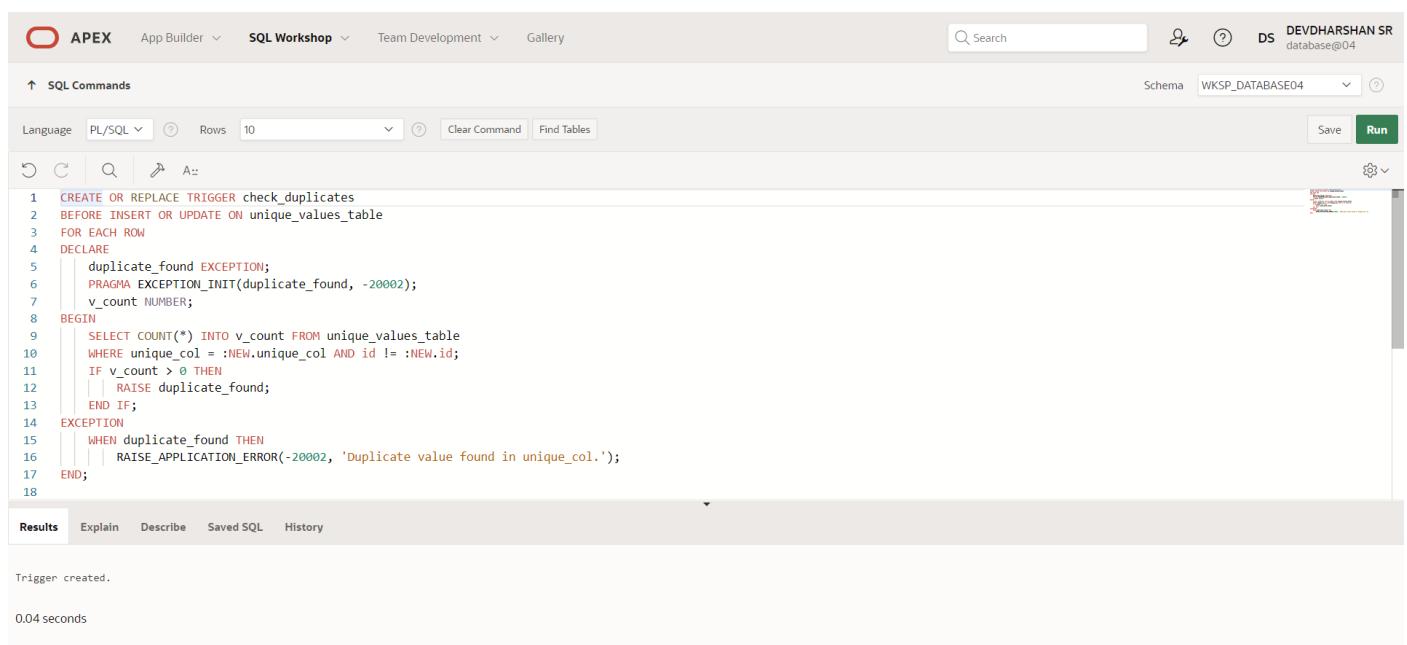
The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, it shows the schema as WKSP_DATABASE04 and the user as DEVDHARSHAN SR database@04. The main workspace is titled "SQL Commands". The SQL editor contains the PL/SQL code for the trigger. The code is highlighted in blue and red, indicating syntax and comments. The "Run" button at the bottom right of the editor is highlighted in green. Below the editor, the "Results" tab is selected, showing the output: "Trigger created." and "0.05 seconds". At the very bottom, there is footer information including the user's email (220701060@rajalakshmi.edu.in), the database name (database@04), and the language (en). The copyright notice is "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version is "Oracle APEX 23.2.4".

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays the PL/SQL code for the trigger. The code is highlighted with syntax coloring. The bottom section shows the results of the execution, indicating that the trigger was successfully created.

```
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
```

Results

```
Trigger created.

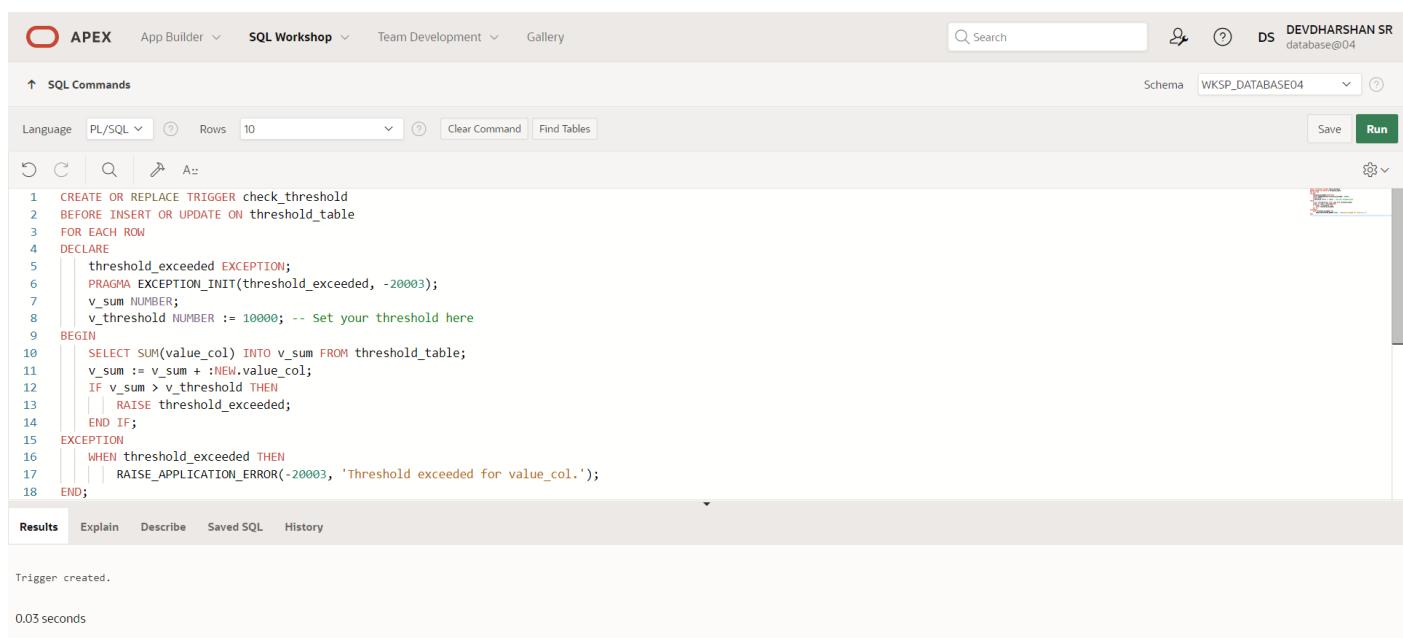
0.04 seconds
```

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. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user DEVDHARSHAN SR and the schema WKSP_DATABASE04. The main area displays the PL/SQL code for the trigger. The code is highlighted in blue and orange, corresponding to keywords and comments. Below the code, the Results tab is selected, showing the message "Trigger created." and a execution time of "0.03 seconds".

```
1 CREATE OR REPLACE TRIGGER 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'. The right side shows the user 'DEVDHARSHAN SR database@04'. The main area is titled 'SQL Commands' with a 'PL/SQL' language dropdown. The code area contains the PL/SQL trigger definition. The bottom section shows the results of the command execution.

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

Results

Trigger created.

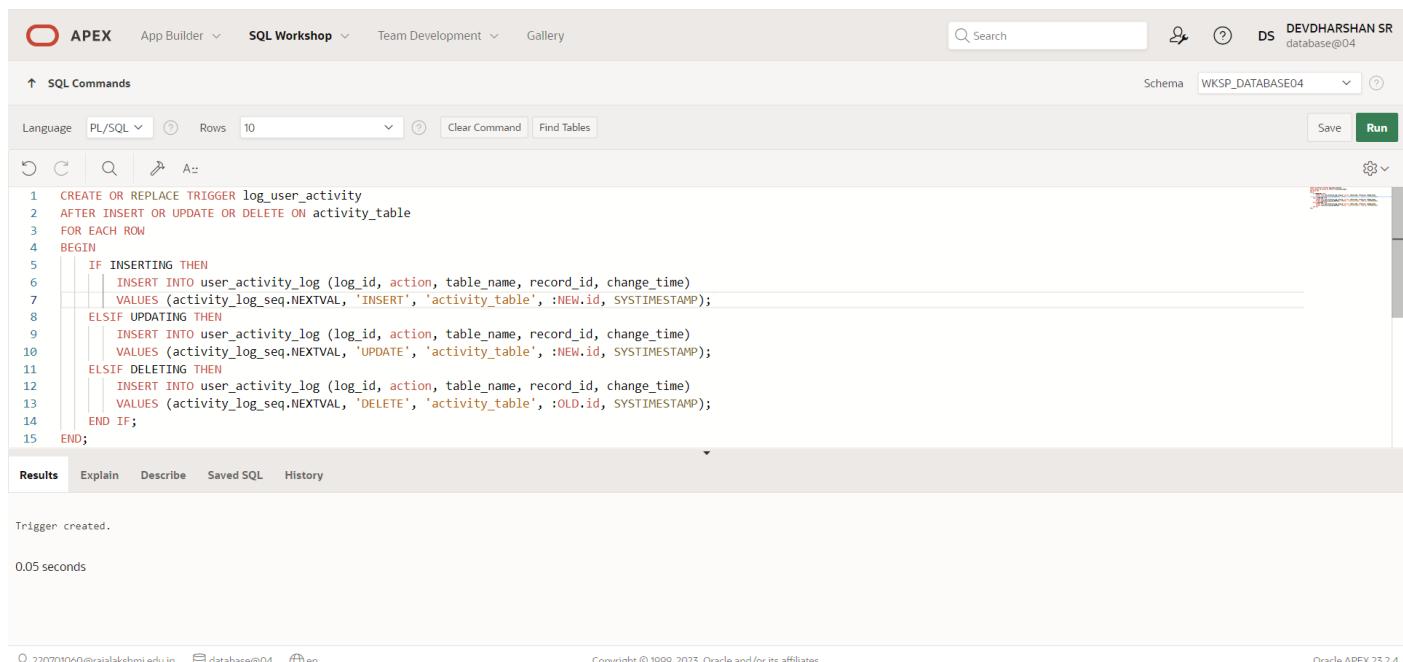
0.04 seconds

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there are search, refresh, and user profile icons, along with the schema information: DS DEVDHARSHAN SR database@04. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is syntax-highlighted, with keywords in blue and identifiers in black. The bottom section displays the results of the command execution, showing the message 'Trigger created.' and a execution time of '0.05 seconds'. The footer includes copyright information for Oracle and the APEX version.

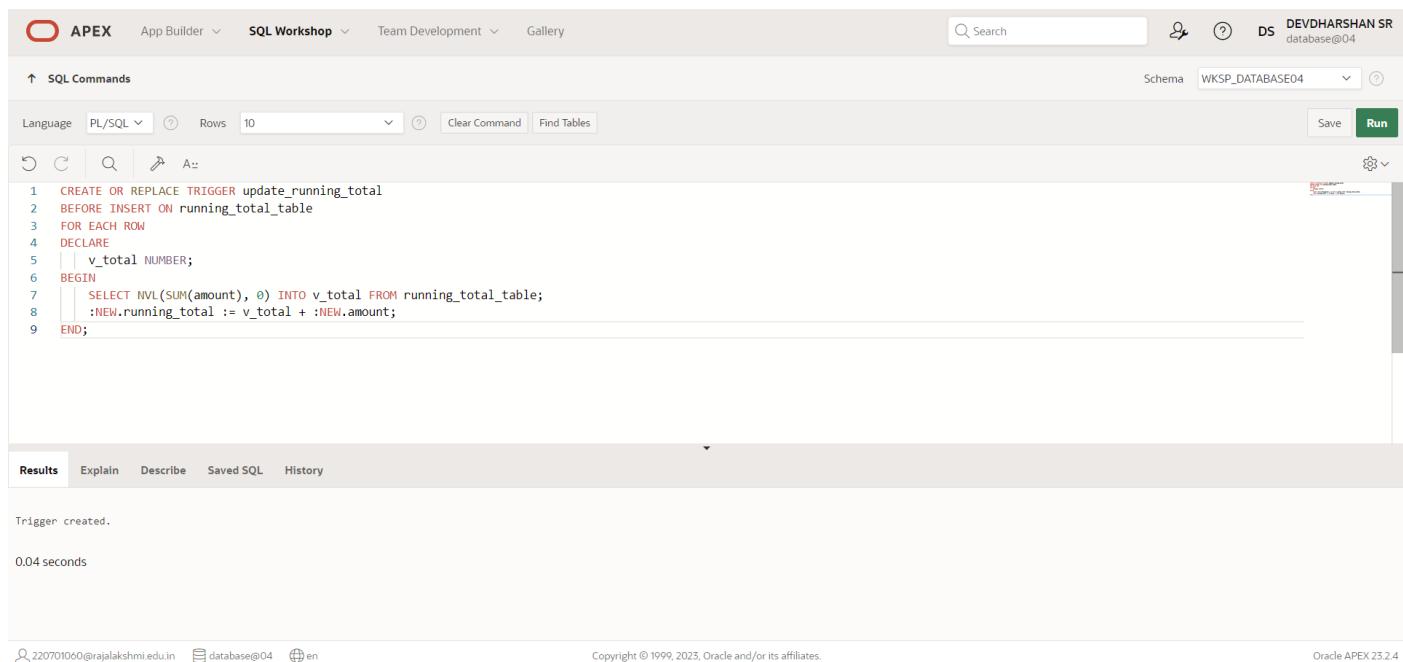
```
1 CREATE OR REPLACE TRIGGER log_user_activity
2 AFTER INSERT OR UPDATE OR DELETE ON activity_table
3 FOR EACH ROW
4 BEGIN
5     IF INSERTING THEN
6         INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
7         VALUES (activity_log_seq.NEXTVAL, 'INSERT', 'activity_table', :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 links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the user 'DEVDHARSHAN SR' and the schema 'WKSP_DATABASE04'. The main workspace is titled 'SQL Commands' and contains a PL/SQL editor. The code for the trigger is pasted into the editor, and the 'Run' button is highlighted. Below the editor, the 'Results' tab is selected, showing the output: 'Trigger created.' and '0.04 seconds'. At the bottom, there are footer links for copyright information and the version 'Oracle APEX 23.2.4'.

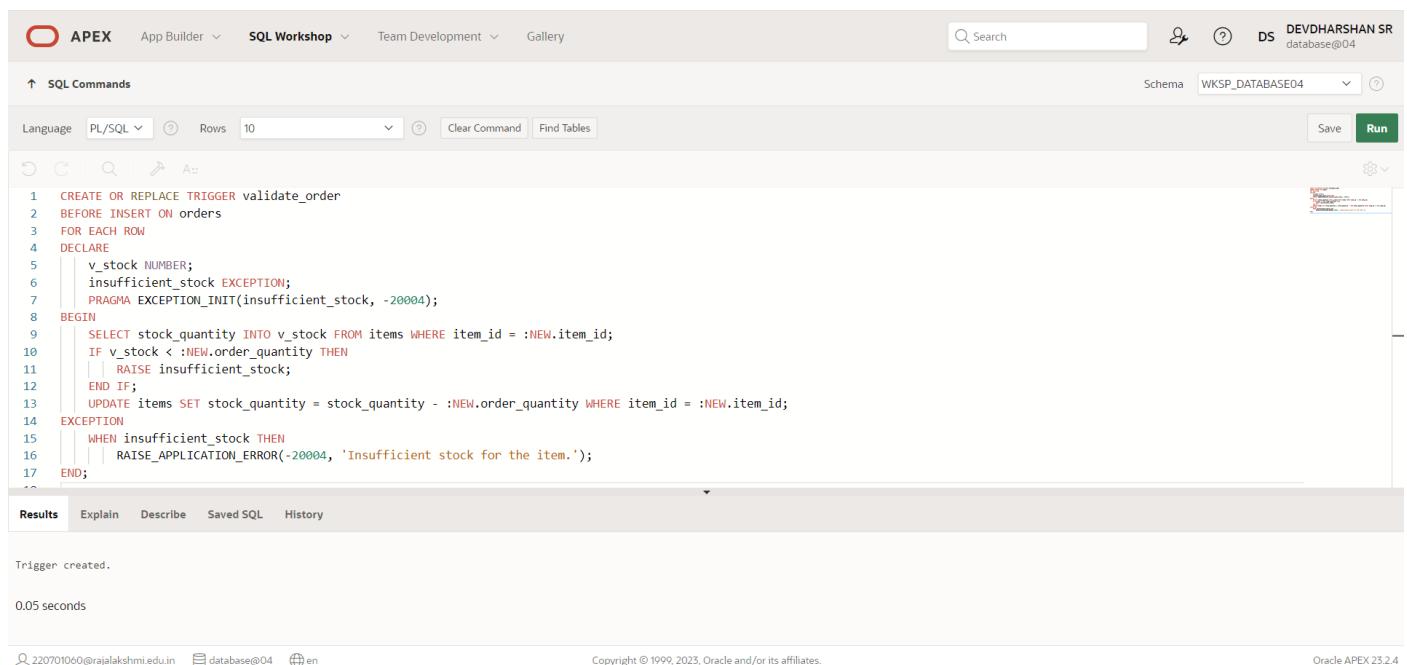
```
1 CREATE OR REPLACE TRIGGER update_running_total
2 BEFORE INSERT ON running_total_table
3 FOR EACH ROW
4 DECLARE
5     v_total NUMBER;
6 BEGIN
7     SELECT NVL(SUM(amount), 0) INTO v_total FROM running_total_table;
8     :NEW.running_total := v_total + :NEW.amount;
9 END;
```

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 links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is connected to the schema 'WKSP_DATABASE04' as 'DEVDHARSHAN SR'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the trigger. The code is highlighted in red and blue, indicating syntax. The bottom pane shows the results of the command execution, displaying the message 'Trigger created.' and a execution time of '0.05 seconds'. The footer includes copyright information for Oracle and the APEX version.

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

Trigger created.
0.05 seconds

220701060@rajalakshmi.edu.in database@04 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:

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:

```
devdharshan_60> db.restaurants.find(
...   {
...     $or: [
...       { name: /^Wil/ },
...       { cuisine: { $nin: ['American', 'Chinese'] } }
...     ]
...   },
...   {
...     restaurant_id: 1,
...     name: 1,
...     borough: 1,
...     cuisine: 1
...   }
... );
devdharshan_60>
```

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:

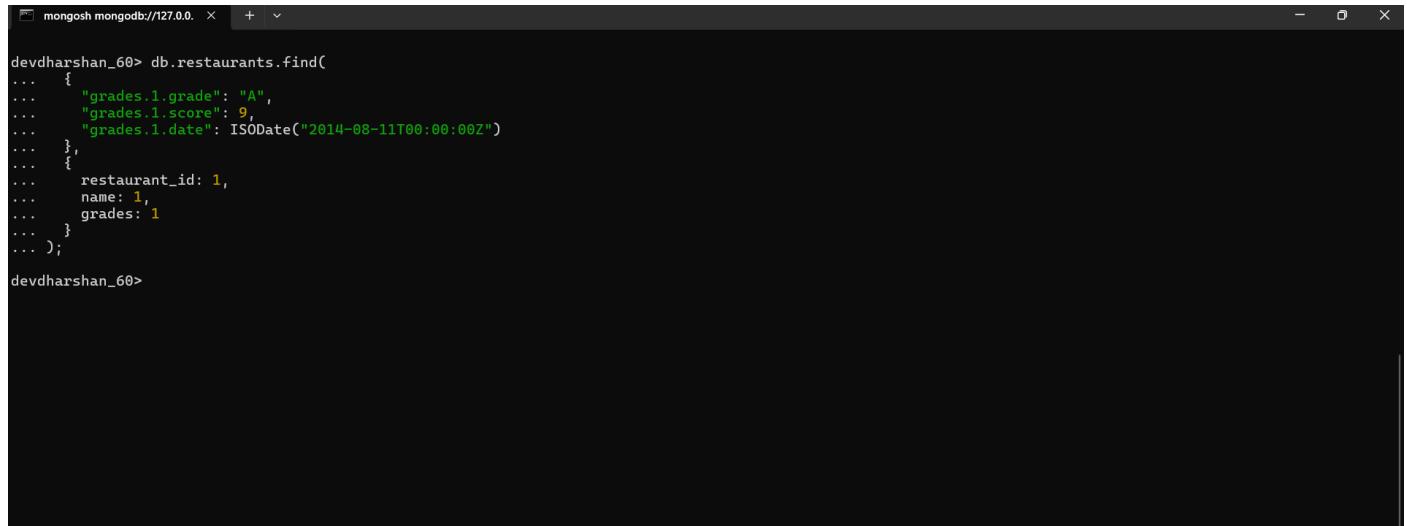
```
devdharshan_60> db.restaurants.find(
...   {
...     grades: {
...       $elemMatch: {
...         grade: "A",
...         score: 11,
...         date: ISODate("2014-08-11T00:00:00Z")
...       }
...     }
...   },
...   {
...     restaurant_id: 1,
...     name: 1,
...     grades: 1
...   }
... );
devdharshan_60>
```

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:



```
mongosh mongodb://127.0.0.1:27017
devdharshan_60> 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
...   }
... );
devdharshan_60>
```

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:



```
devdharshan_60> db.restaurants.find(
...   {
...     "address.coord.1": {
...       $gt: 42,
...       $lte: 52
...     }
...   },
...   {
...     _id: 0,
...     restaurant_id: 1,
...     name: 1,
...     address: 1
...   }
... );
devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find(
...   {},
...   { _id: 0 }
... ).sort(
...   { name: 1 }
... );

devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find(
...   {},
...   { _id: 0 }
... ).sort(
...   { name: -1 }
... );

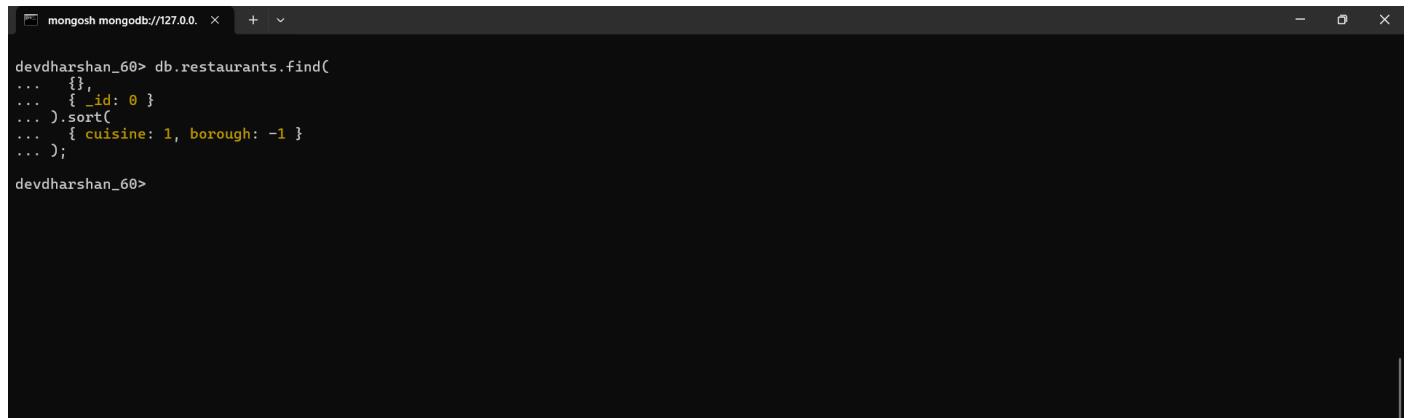
devdharshan_60> |
```

7.) Write a MongoDB query to arrange 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:



```
mongosh mongodb://127.0.0.1:27017
devdharshan_60> db.restaurants.find(
...   {},
...   { _id: 0 }
... ).sort(
...   { cuisine: 1, borough: -1 }
... );
devdharshan_60>
```

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:



```
mongosh mongodb://127.0.0.1:27017
devdharshan_60> db.restaurants.find(
...   { "address.street": { $exists: true, $ne: "" } }
... );
devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find(
...   { "address.coord": { $elemMatch: { $type: "double" } } }
... );
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find(
...   { "grades.score": { $mod: [7, 0] } },
...   { restaurant_id: 1, name: 1, grades: 1 }
... );
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })  
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })  
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })  
devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
devdharshan_60>
```

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:

```
devdharshan_60> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
devdharshan_60>
```

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:

```
devdharshan_60> 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"] } })  
devdharshan_60> |
```

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:

```
devdharshan_60> db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })  
devdharshan_60>
```

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:

```
devdharshan_60> db.movies.find({ year: 1893 })
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find({ runtime: { $gt: 120 } })
devdharshan_60> |
```

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

QUERY:

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

OUTPUT:

```
devdharshan_60> db.movies.find({ genres: 'Short' })
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find({ directors: 'William K.L. Dickson' })
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find({ countries: 'USA' })
devdharshan_60> |
```

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

QUERY:

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

OUTPUT:

```
devdharshan_60> db.movies.find({ rated: 'UNRATED' })
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find({ 'imdb.votes': { $gt: 1000 } })  
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find({ 'imdb.rating': { $gt: 7 } })  
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })  
devdharshan_60> |
```

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

QUERY:

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

OUTPUT:

```
devdharshan_60> db.movies.find({ 'awards.wins': { $gt: 0 } })  
devdharshan_60> |
```

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:

```
devdharshan_60> 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 })
```

```
devdharshan_60> |
```

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:

```
devdharshan_60> 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 })
```

```
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find( { released: ISODate("1893-05-09T00:00:00.000Z") }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )
devdharshan_60> |
```

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:

```
devdharshan_60> db.movies.find( { title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )
devdharshan_60> |
```

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

RESULT: