Creating and Managing Tables

EX_NO:1 **DATE:**

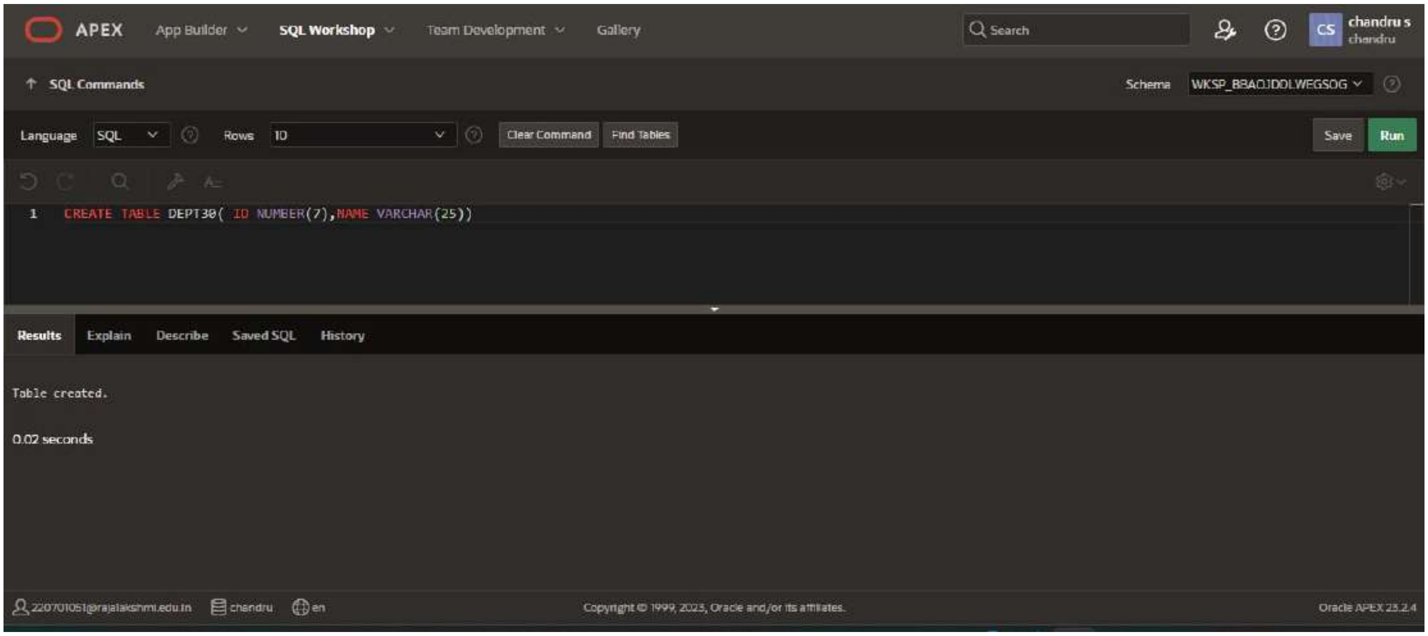
1.Create the DEPT table based on the DEPARTMENT following the table instance chart below. Confirm that the table is created.

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

QUERY:

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

OUTPUT:



```
CREATE TABLE DEPT30( ID NUMBER(7),NAME VARCHAR(25))
```

```
Table created.

0.02 seconds
```

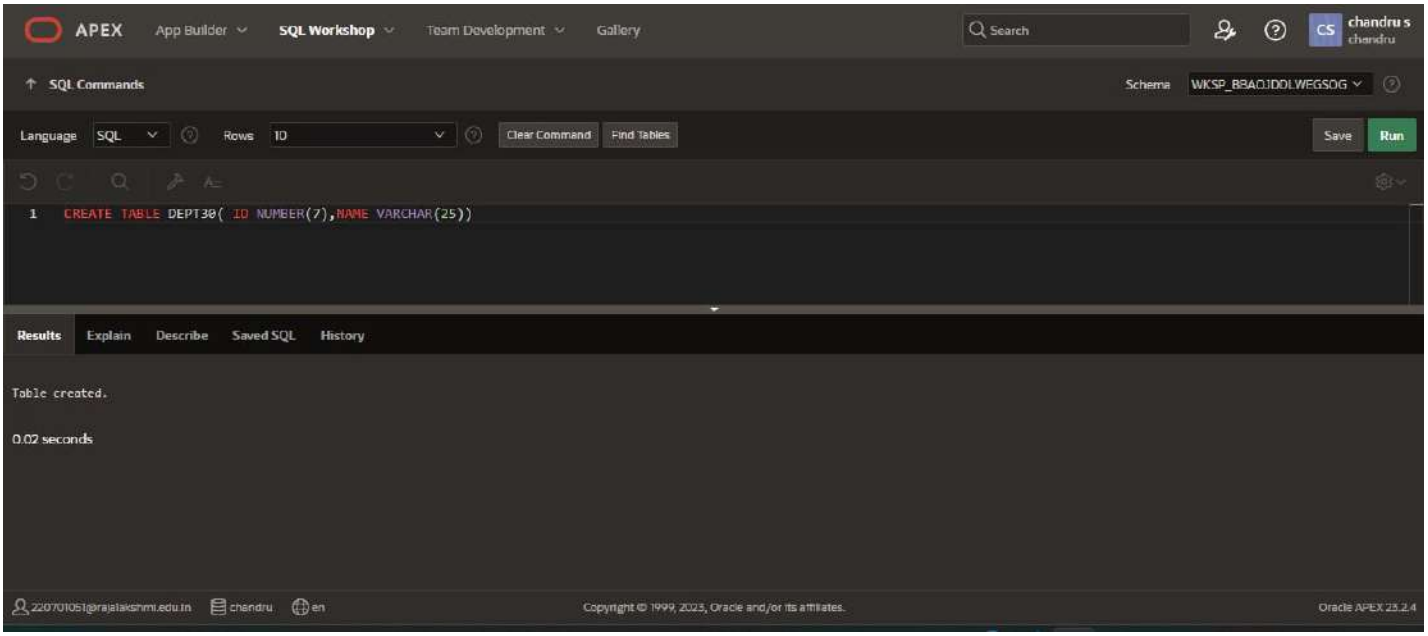
2.Create the EMP table based on the following instance chart. Confirm that the table is created.

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

QUERY:

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

OUTPUT:



```
CREATE TABLE DEPT30( ID NUMBER(7),NAME VARCHAR(25))
```

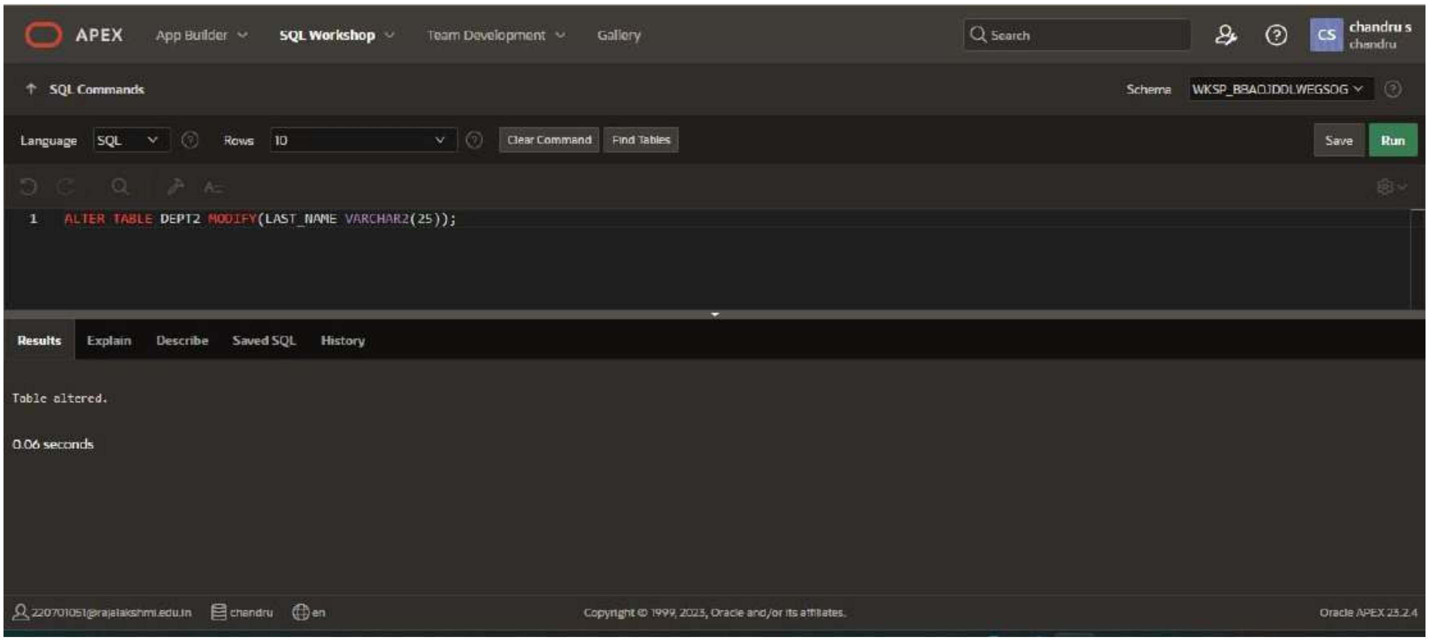
```
Table created.

0.02 seconds
```

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

QUERY:

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

OUTPUT:



```
ALTER TABLE DEPT2 MODIFY(LAST_NAME VARCHAR2(25));
```

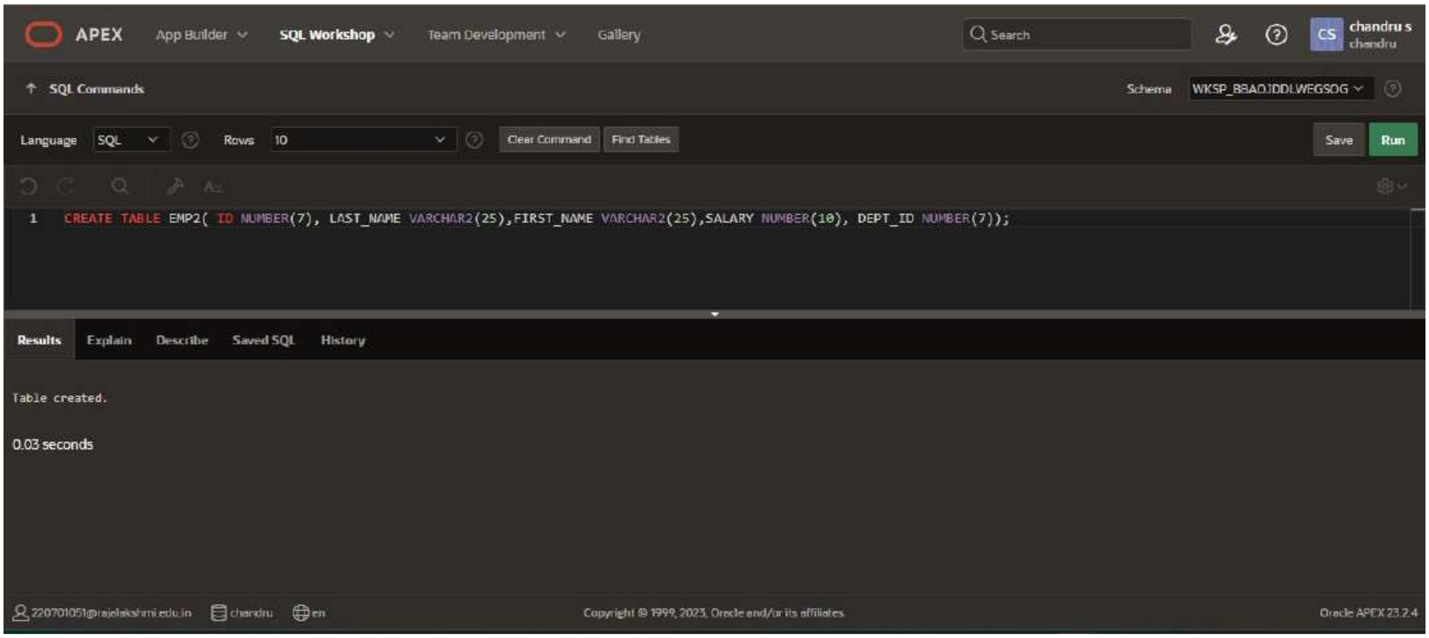
```
Table altered.

0.06 seconds
```

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

QUERY:

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

OUTPUT:



```
CREATE TABLE EMP2( ID NUMBER(7), LAST_NAME VARCHAR2(25),FIRST_NAME VARCHAR2(25),SALARY NUMBER(10), DEPT_ID NUMBER(7));
```

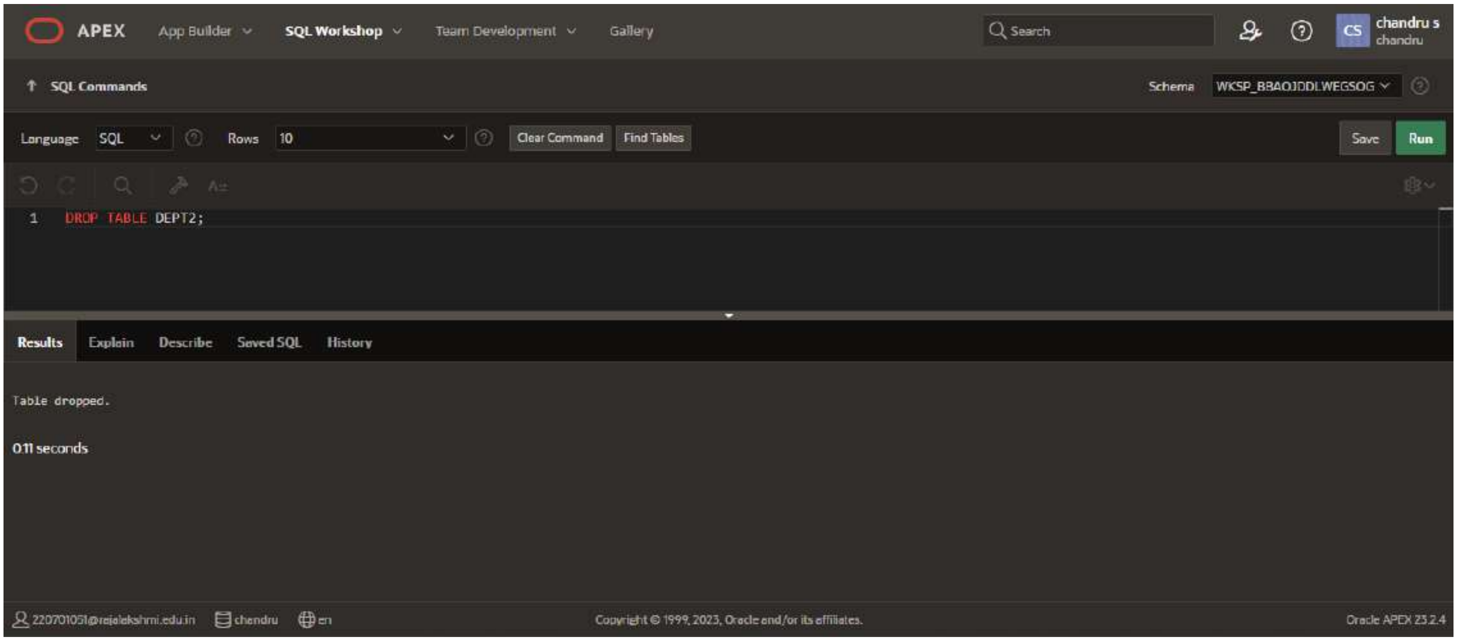
```
Table created.

0.03 seconds
```

5.Drop the EMP table.

QUERY:

Drop table emp;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
DROP TABLE DEPT2;
```

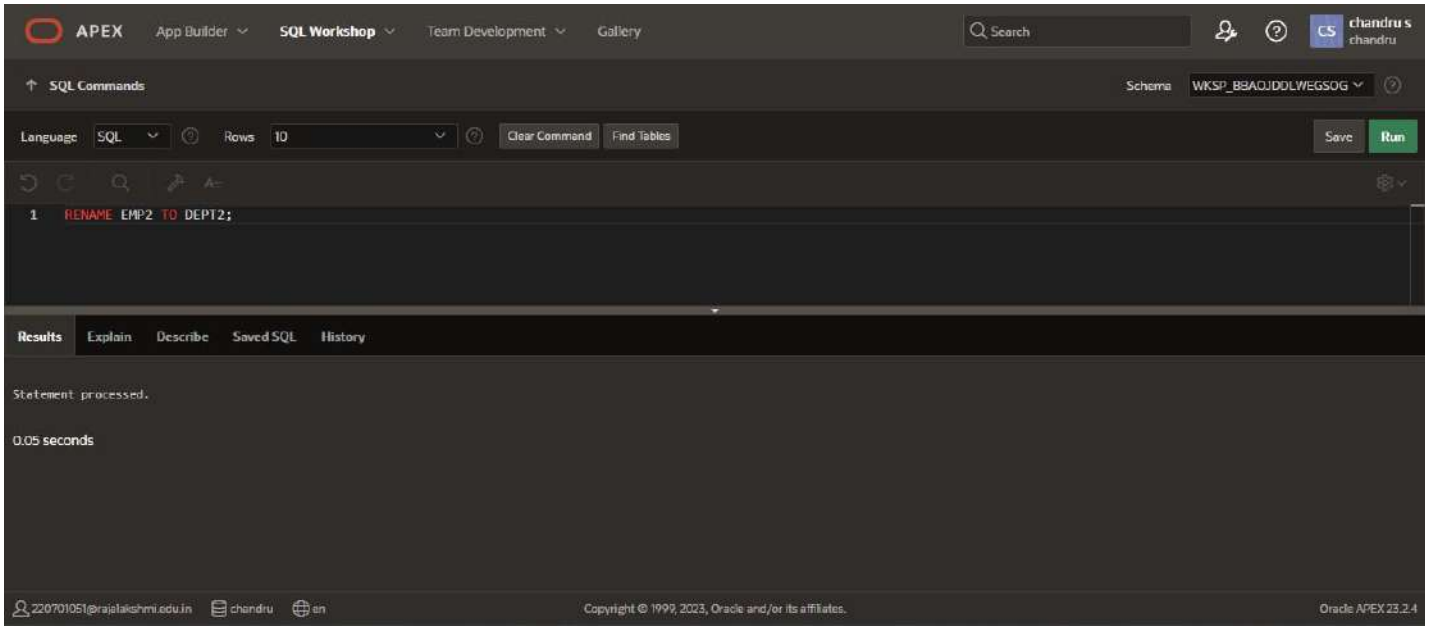
```
Table dropped.

0.11 seconds
```

6.Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename employees2 to emp;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
RENAME EMP2 TO DEPT2;
```

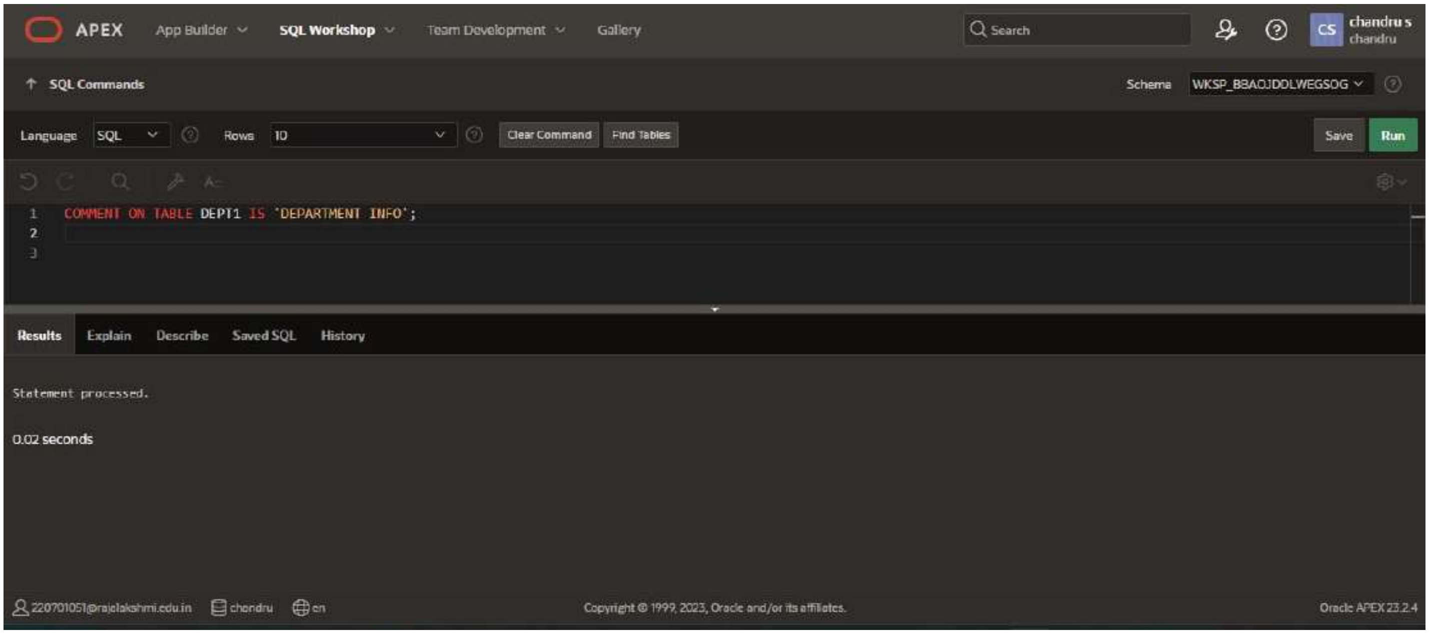
```
Statement processed.

0.05 seconds
```

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:



```
COMMENT ON TABLE DEPT1 IS 'DEPARTMENT INFO';
```

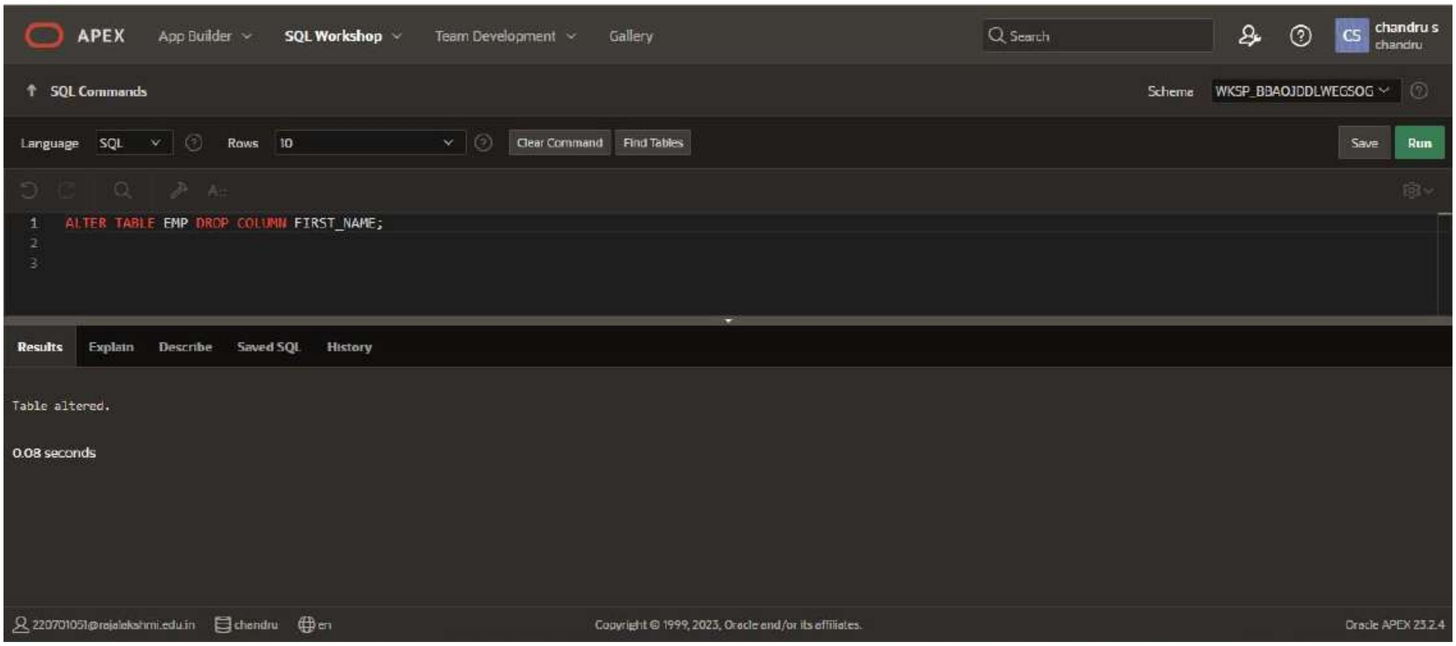
```
Statement processed.

0.02 seconds
```

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

QUERY:

Alter table emp drop column first_name;

OUTPUT:



```
ALTER TABLE EMP DROP COLUMN FIRST_NAME;
```

```
Table altered.

0.08 seconds
```

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

RESULT:

MANIPULATING DATA

EX_NO:2 **DATE:**

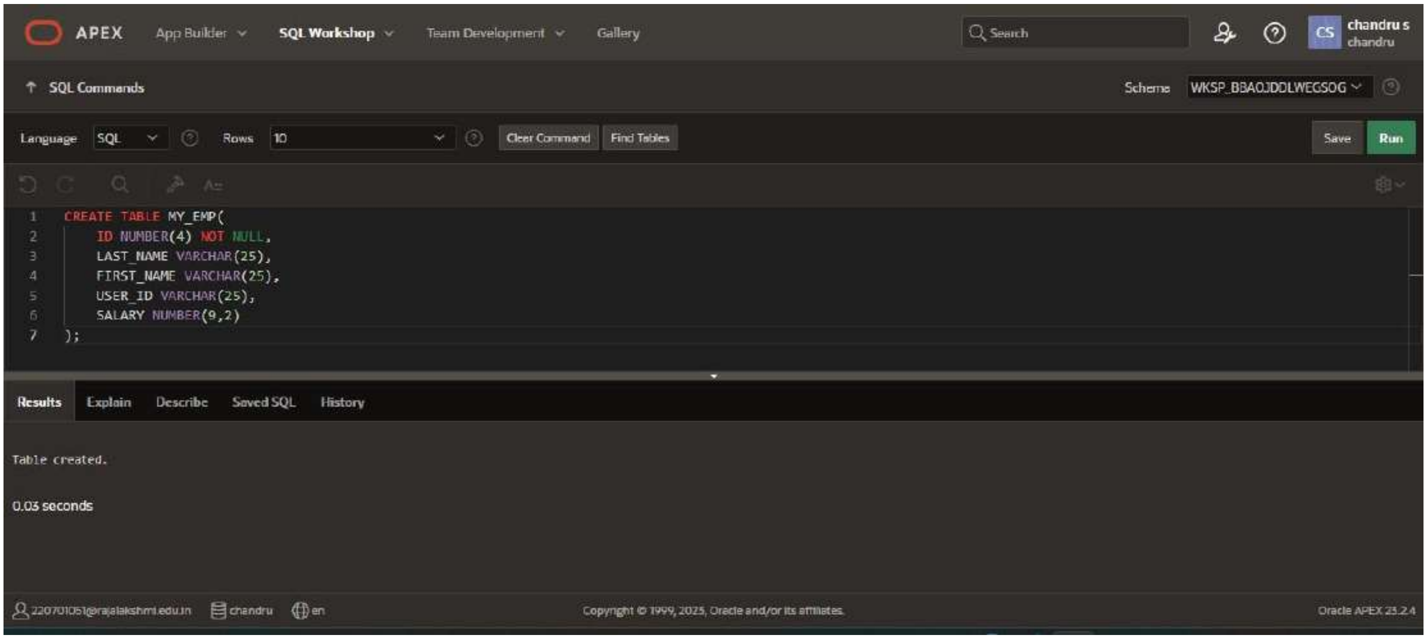
1.Create MY_EMPLOYEE table with the following structure

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_emp(id number(4) not null,last_name varchar(25),
First_name varchar(25),userid varchar(25),salary number(9,2));

OUTPUT:



```
CREATE TABLE MY_EMP(
    ID NUMBER(4) NOT NULL,
    LAST_NAME VARCHAR(25),
    FIRST_NAME VARCHAR(25),
    USER_ID VARCHAR(25),
    SALARY NUMBER(9,2)
);
```

```
Table created.

0.03 seconds
```

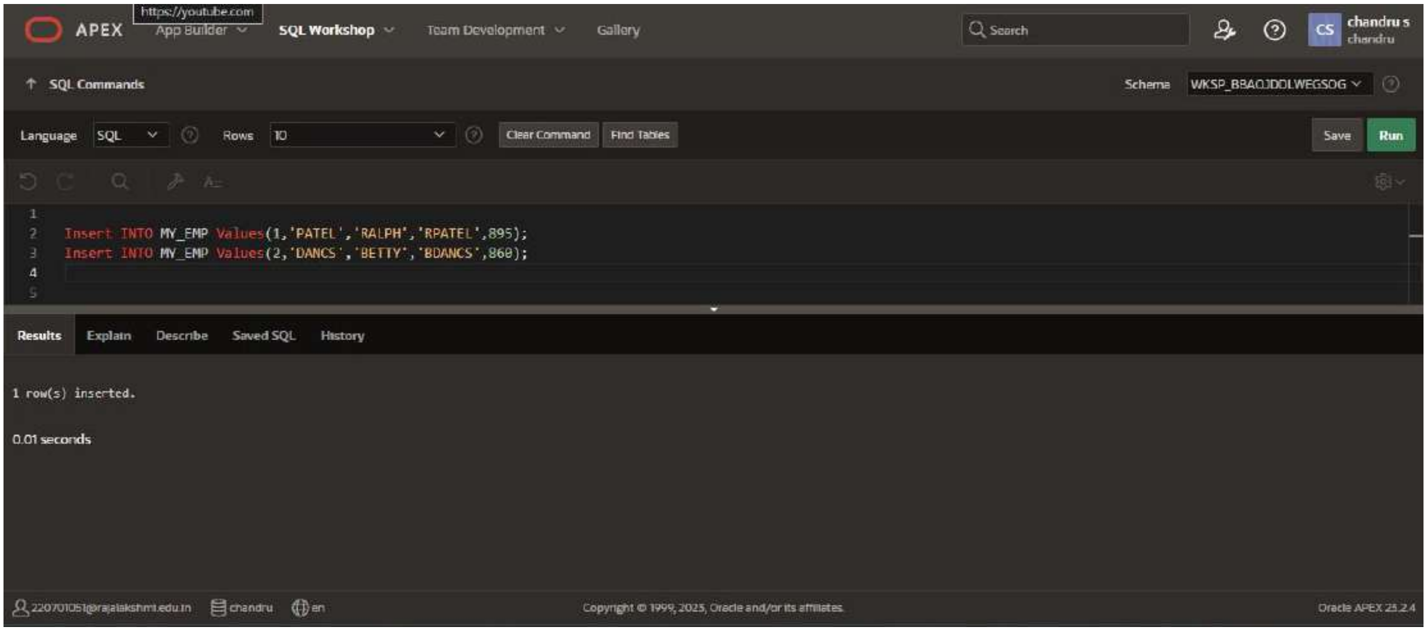
2.Add the first and second rows data to MY_EMPLOYEE table from the following sample data.

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

QUERY:

Insert into my_emp values(1,’Patel’,’Ralph’,’rpatel’,895);
Insert into my_emp values(1,’Dancs’,’Betty’,’bdancs’,860);

OUTPUT:



```
Insert INTO MY_EMP Values(1,'PATEL','RALPH','RPATEL',895);
Insert INTO MY_EMP Values(2,'DANCS','BETTY','BDANCS',860);
```

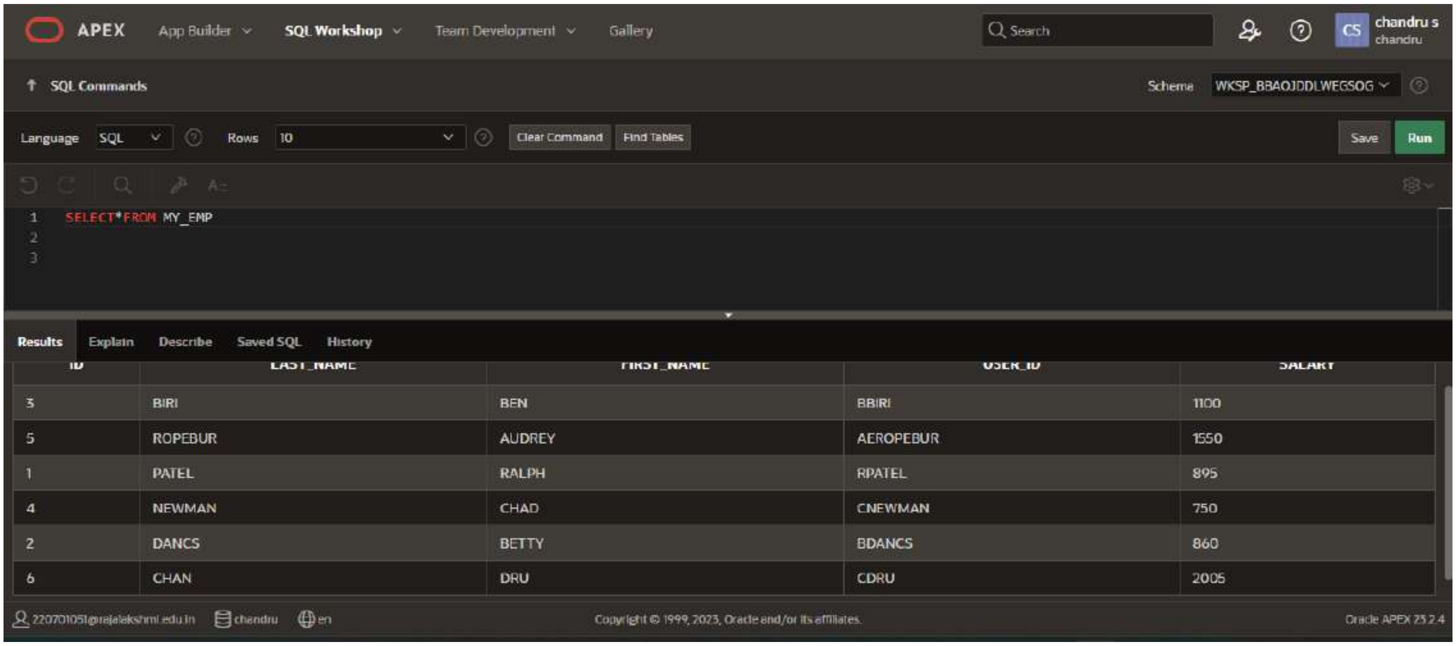
```
1 row(s) inserted.

0.01 seconds
```

3.Display the table with values.

QUERY:

Select*from my_emp;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
SELECT*FROM MY_EMP
```

ID	LAST_NAME	FIRST_NAME	USER_ID	SALARY
3	BIRI	BEN	BBIRI	1100
5	ROPEBUR	AUDREY	AEROPEBUR	1550
1	PATEL	RALPH	RPATEL	895
4	NEWMAN	CHAD	CNEWMAN	750
2	DANCS	BETTY	BDANCS	860
6	CHAN	DRU	CDRU	2005

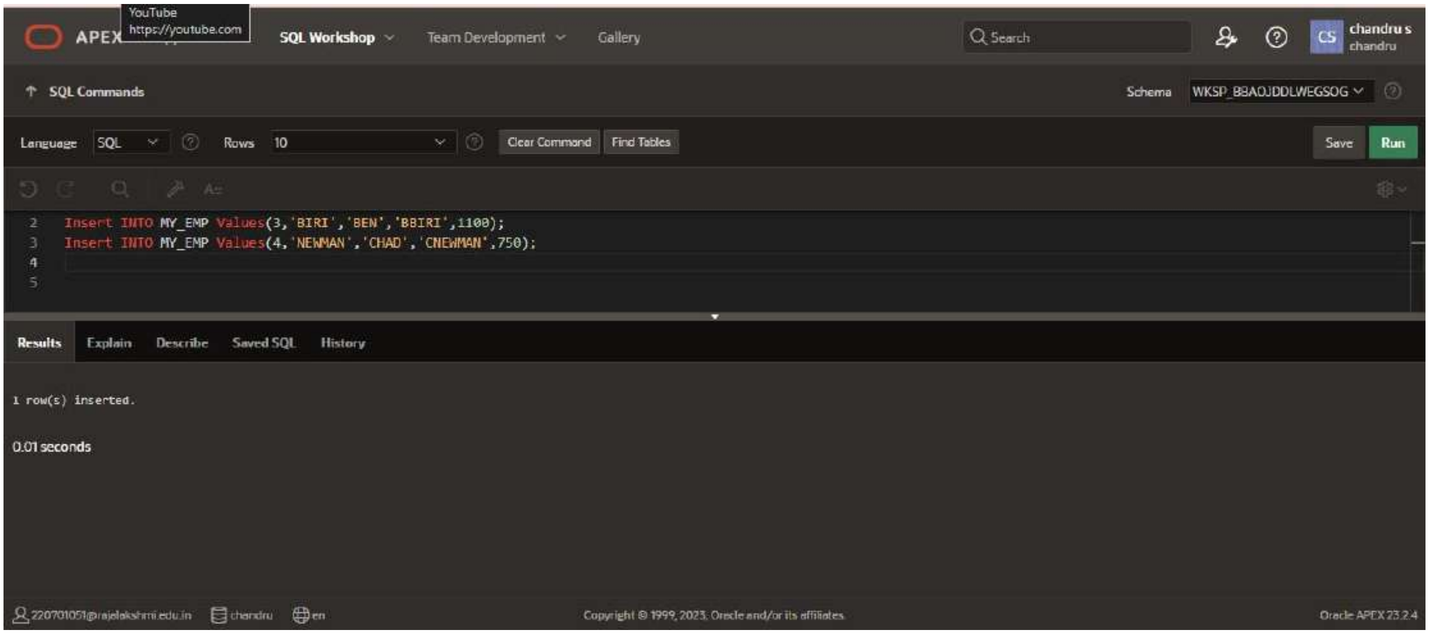
4.Populate the next two rows of data from the sample data. Concatenate the first letter of the first_name with the first seven characters of the last_name to produce Userid.

QUERY:

Insert into my_emp values(3,’biri’,’Ben’,’bbiri’,1100);

Insert into my_emp values(1,’newman’,’chad’,’cnewman’,750);

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
Insert INTO MY_EMP Values(3,'BIRI','BEN','BBIRI',1100);
Insert INTO MY_EMP Values(4,'NEWMAN','CHAD','CNEWMAN',750);
```

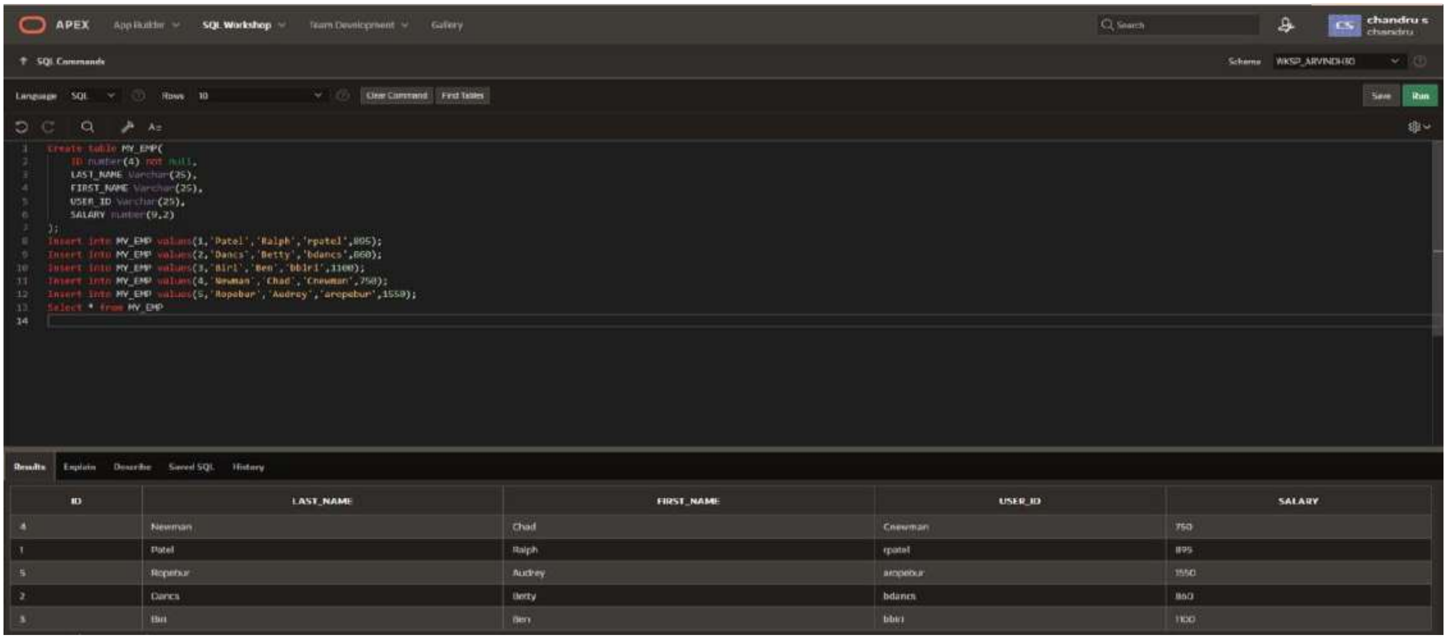
```
1 row(s) inserted.
```

0.01 seconds

5.Make the data additions permanent.

QUERY:
Select*from my_emp;

OUTPUT:



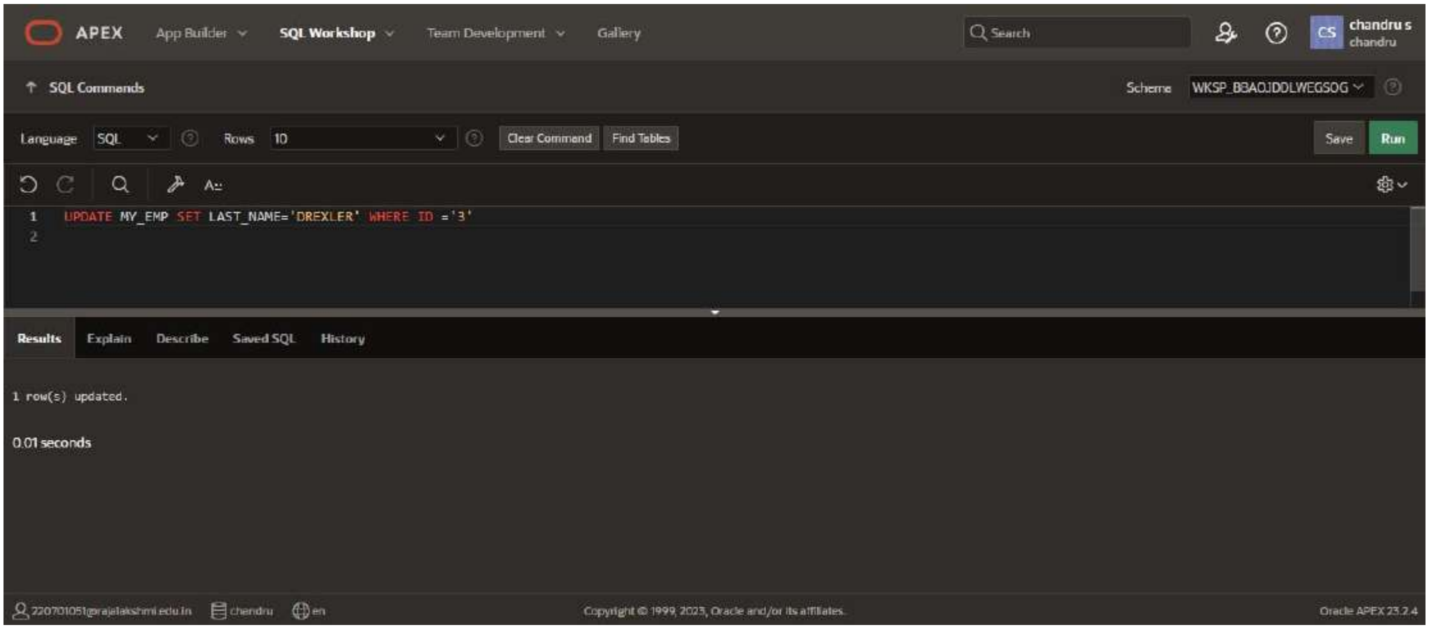
```
Create table MY_EMP(
    ID number(4) not null,
    LAST_NAME Varchar(25),
    FIRST_NAME Varchar(25),
    USER_ID Varchar(25),
    SALARY number(9,2)
);
Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);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

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

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

QUERY:
Update my_emp set last_name='drexler' where id=3;

OUTPUT:



```
UPDATE MY_EMP SET LAST_NAME='DREXLER' WHERE ID ='3'
```

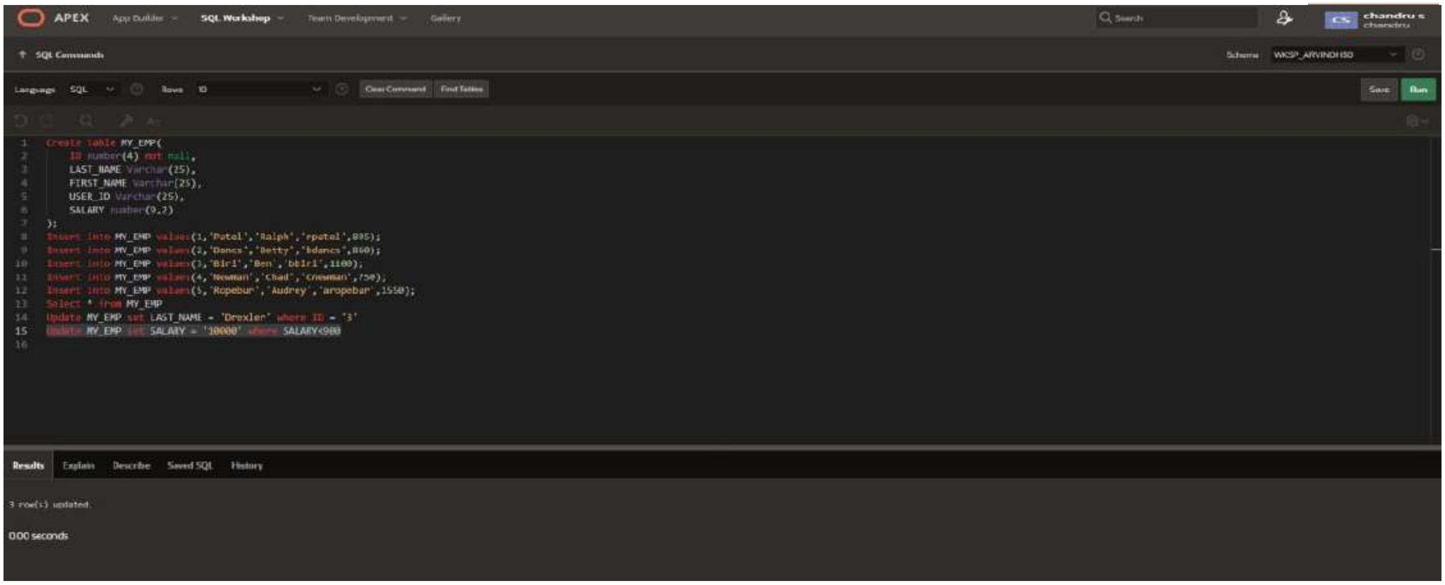
```
1 row(s) updated.

0.01 seconds
```

220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:
Update my_emp set salary=1000 where id=3;

OUTPUT:



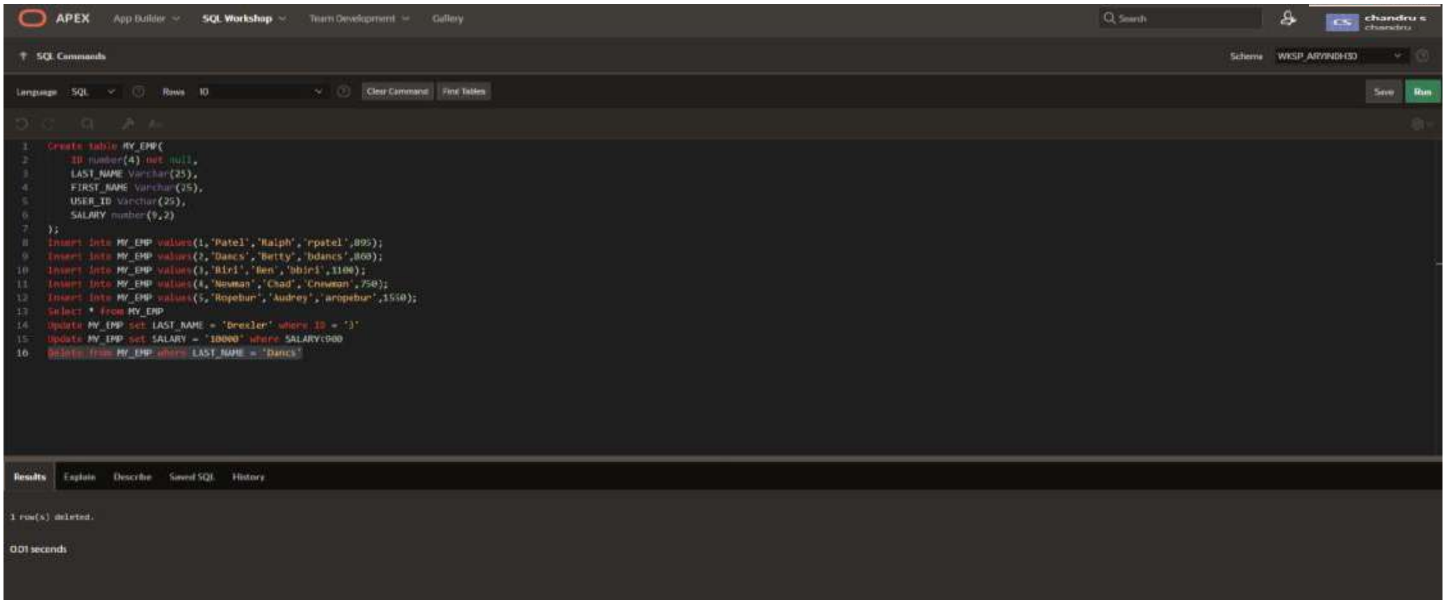
```
Create table MY_EMP(
    ID number(4) not null,
    LAST_NAME Varchar(25),
    FIRST_NAME Varchar(25),
    USER_ID Varchar(25),
    SALARY number(9,2)
);
Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
Select * from MY_EMP
Update MY_EMP set LAST_NAME = 'Drexler' where ID = '3'
Update MY_EMP set SALARY = '10000' where SALARY<900
```

```
3 row(s) updated.

0.00 seconds
```

8.Delete Betty dancs from MY _EMPLOYEE table.

QUERY:
Delete from my_emp where last_name=’Dancs’;

OUTPUT:



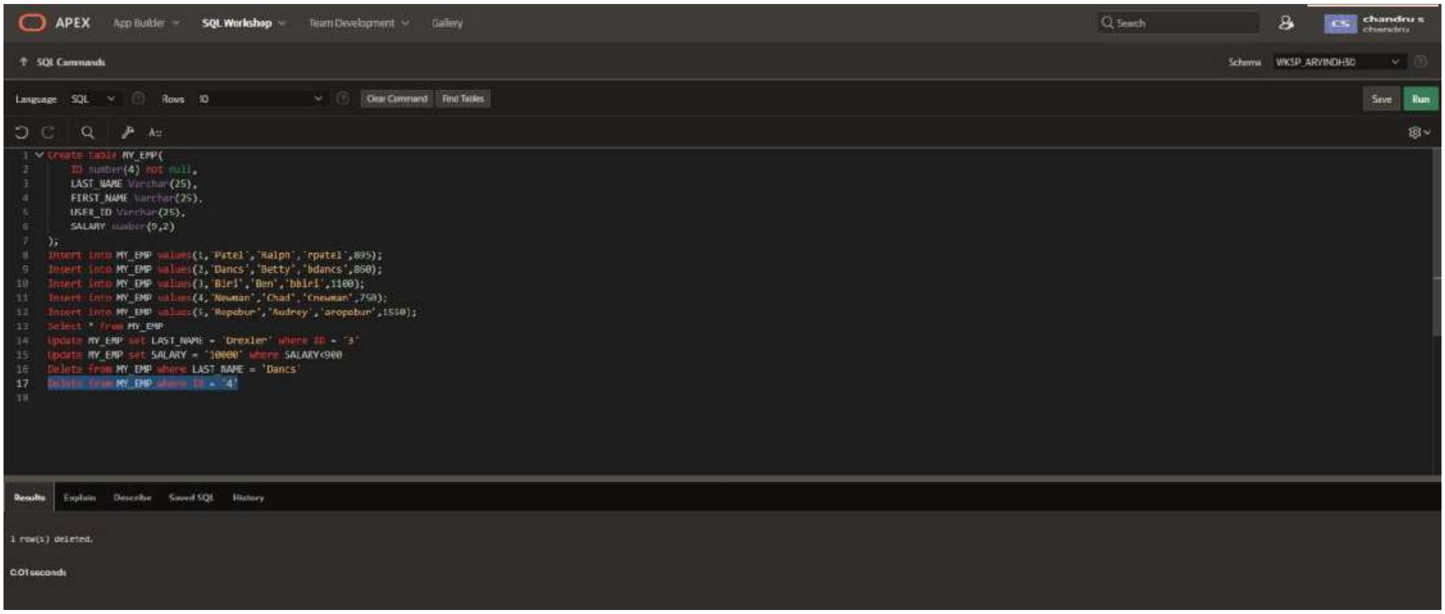
```
Create table MY_EMP(
    ID number(4) not null,
    LAST_NAME Varchar(25),
    FIRST_NAME Varchar(25),
    USER_ID Varchar(25),
    SALARY number(9,2)
);
Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
Select * from MY_EMP
Update MY_EMP set LAST_NAME = 'Drexler' where ID = '3'
Update MY_EMP set SALARY = '10000' where SALARY<900
Delete from MY_EMP where LAST_NAME = 'Dancs'
```

```
1 row(s) deleted.

0.01 seconds
```

9.Empty the fourth row of the emp table.

QUERY:
Delete from my_emp where id=4;

OUTPUT:



Schema WKSP_ARVINDH30

Language SQL Rows 10 Clear Command Find Tables

```
Create table MY_EMP(
    ID number(4) not null,
    LAST_NAME Varchar(25),
    FIRST_NAME Varchar(25),
    USER_ID Varchar(25),
    SALARY number(9,2)
);
Insert into MY_EMP values(1,'Patel','Ralph','rpatel',895);
Insert into MY_EMP values(2,'Dancs','Betty','bdancs',860);
Insert into MY_EMP values(3,'Biri','Ben','bbiri',1100);
Insert into MY_EMP values(4,'Newman','Chad','Cnewman',750);
Insert into MY_EMP values(5,'Ropebur','Audrey','aropebur',1550);
Select * from MY_EMP
Update MY_EMP set LAST_NAME = 'Drexler' where ID = '3'
Update MY_EMP set SALARY = '10000' where SALARY<900
Delete from MY_EMP where LAST_NAME = 'Dancs'
Delete from MY_EMP where ID = '4'
```

```
1 row(s) deleted.

0.01 seconds
```

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

RESULT:

INCLUDING CONSTRAINTS

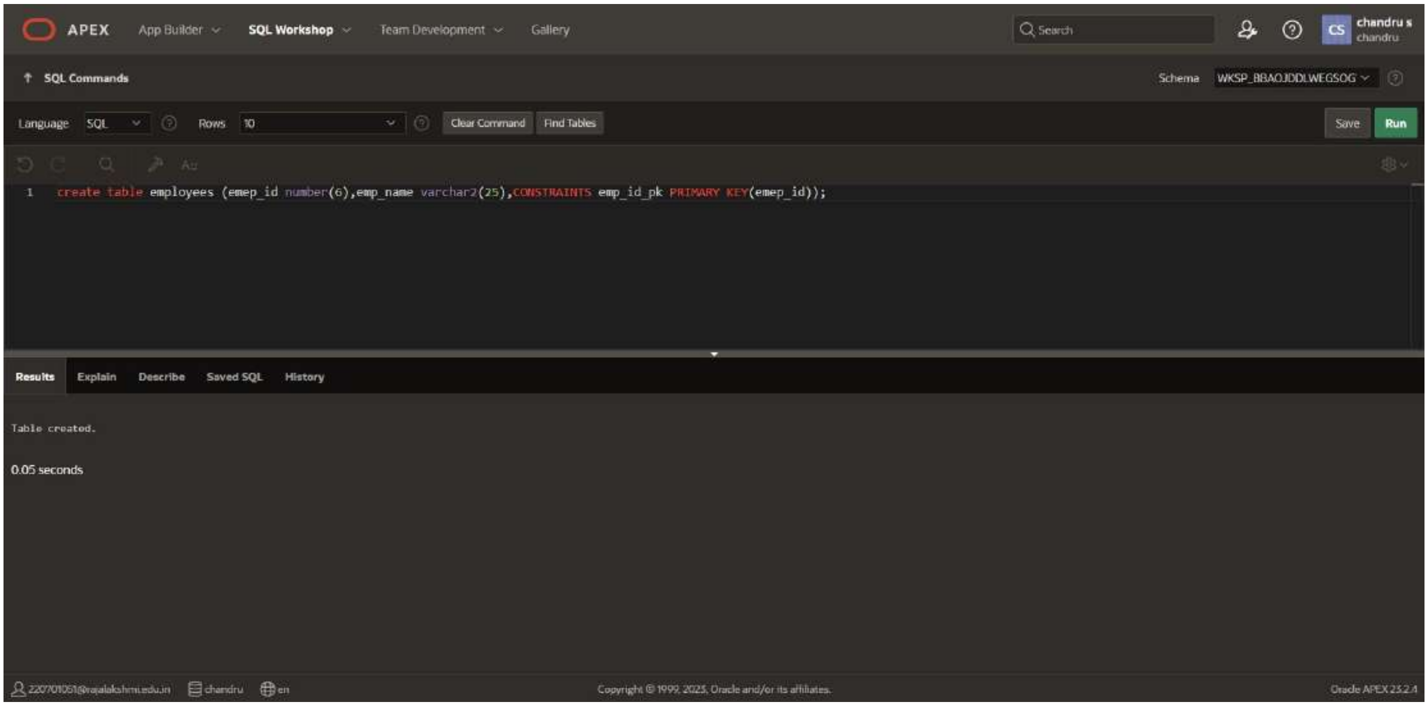
EX_NO:3 **DATE:**

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 department (dept_id number(6),dept_name varchar2(25),Constraints emp_id_pk primary key(emp_id);

OUTPUT:



```
create table employees (emep_id number(6),emp_name varchar2(25),CONSTRAINTS emp_id_pk PRIMARY KEY(emep_id));
```

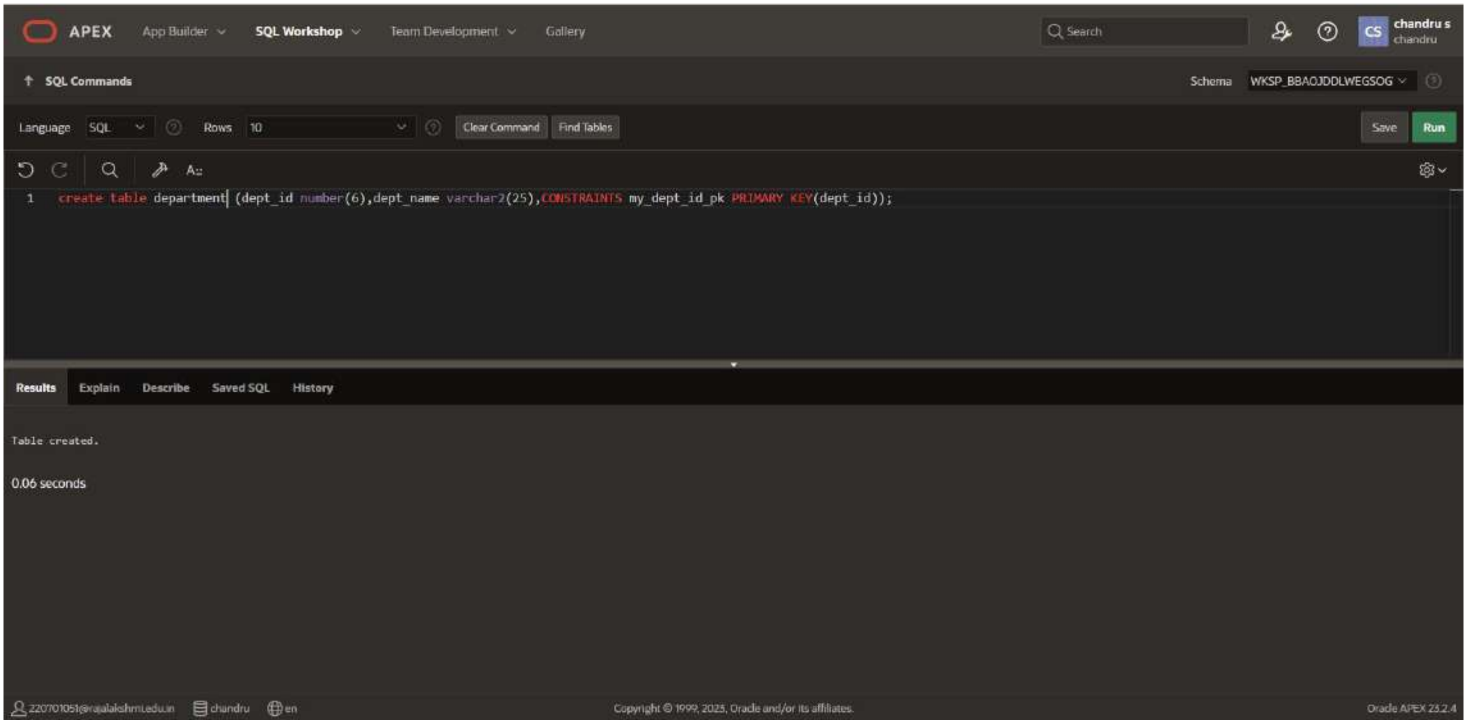
```
Table created.

0.05 seconds
```

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

QUERY:

Create table employees(dept_id number(6),dept_name varchar2(25),Constraints my_dept_id_pk primary key(dept_id);

OUTPUT:



```
create table department (dept_id number(6),dept_name varchar2(25),CONSTRAINTS my_dept_id_pk PRIMARY KEY(dept_id));
```

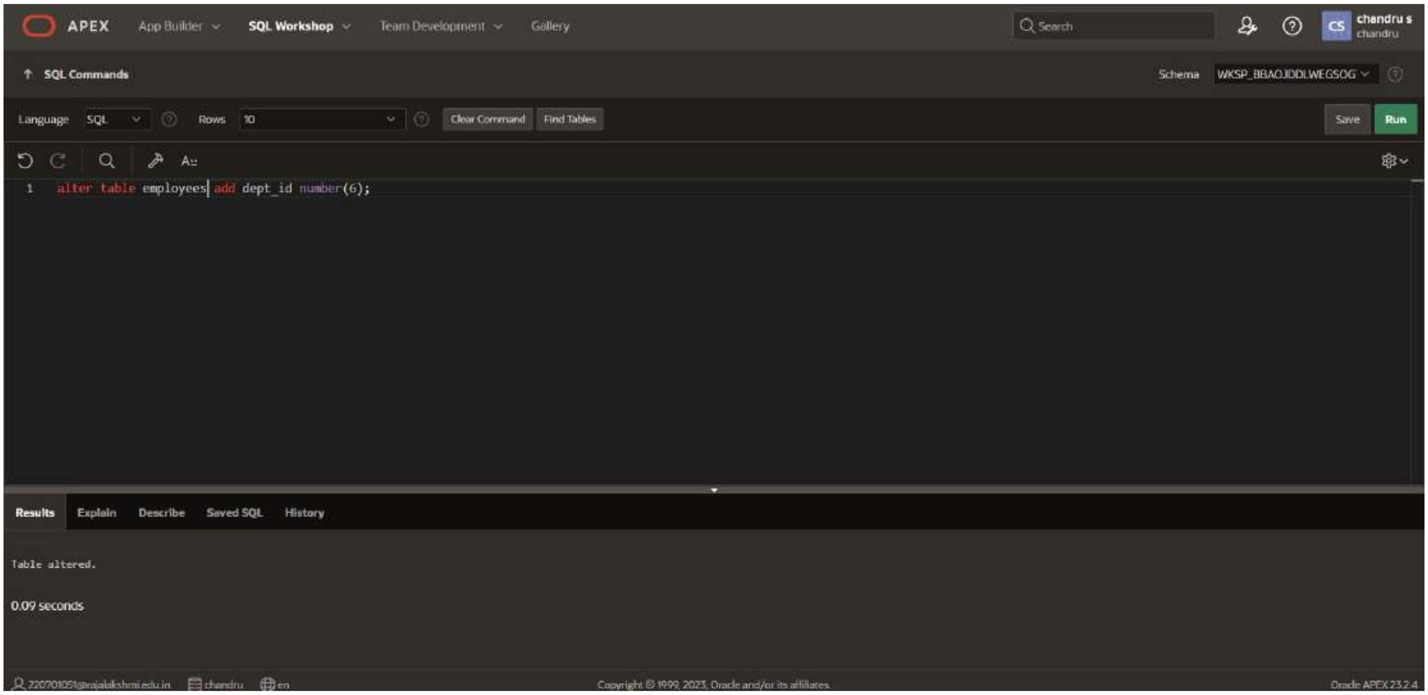
```
Table created.

0.06 seconds
```

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 deparment. Name the constraint my_emp_dept_id_fk.

QUERY:

alter table employees add dept_id number(6);

OUTPUT:



```
alter table employees add dept_id number(6);
```

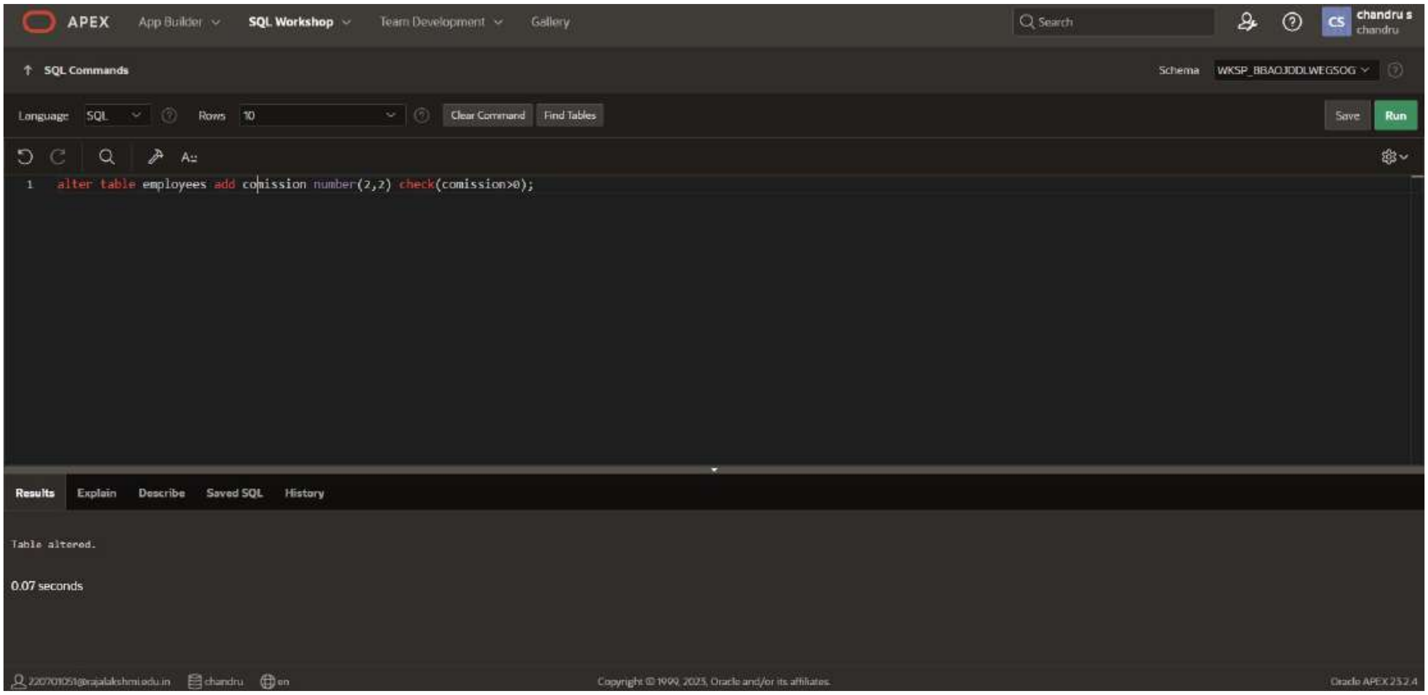
```
Table altered.

0.09 seconds
```

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 employees add commission number(2,2) check(commission>0);

OUTPUT:



```
alter table employees add comission number(2,2) check(comission>0);
```

```
Table altered.

0.07 seconds
```

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

RESULT:

Writing Basic SQL SELECT Statements

EX_NO:4 **DATE:**

1.The following statement executes successfully.

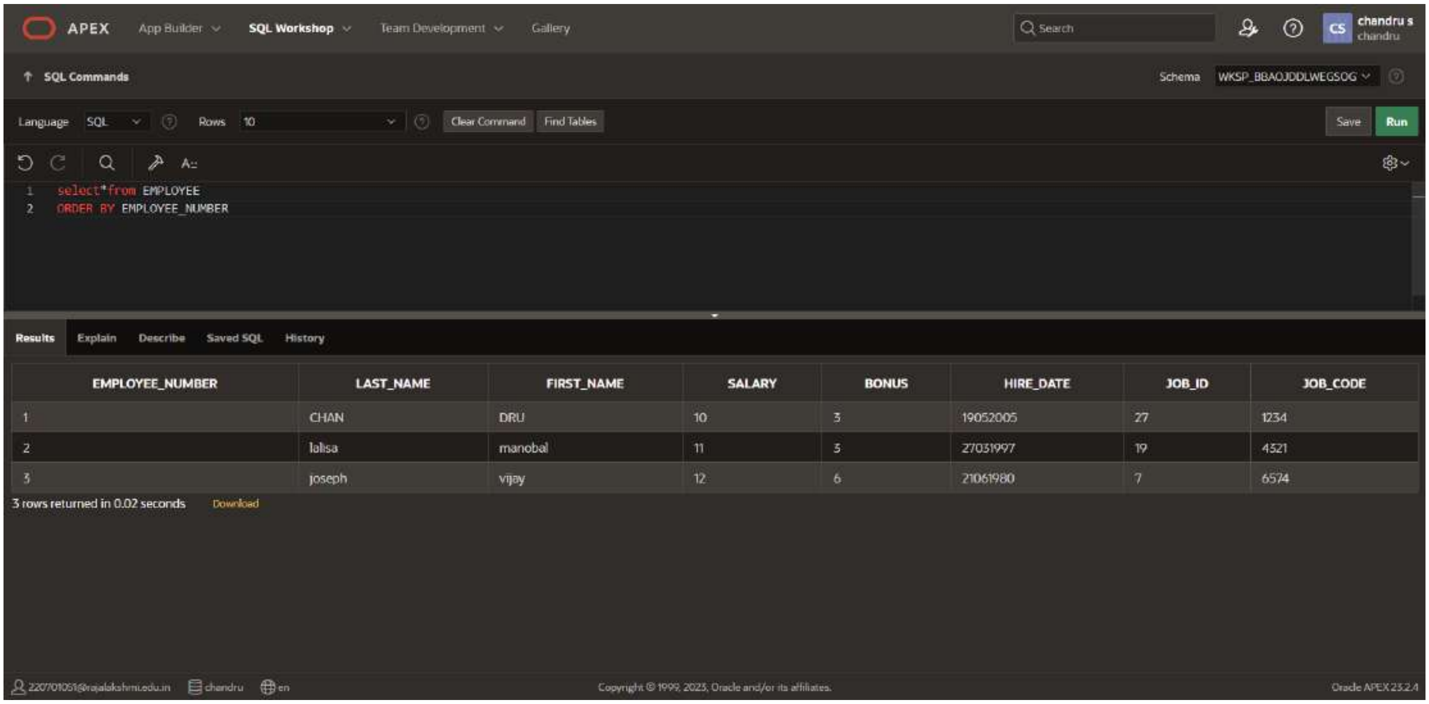
Identify the Errors

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

QUERY:

select employee_id,last_name sal*12 “ANNUAL SALARY” from employees;

OUTPUT:



```
select*from EMPLOYEE
ORDER BY EMPLOYEE_NUMBER
```

EMPLOYEE_NUMBER	LAST_NAME	FIRST_NAME	SALARY	BONUS	HIRE_DATE	JOB_ID	JOB_CODE
1	CHAN	DRU	10	3	19052005	27	1234
2	lalisa	manobal	11	3	27031997	19	4321
3	joseph	vijay	12	6	21061980	7	6574

3 rows returned in 0.02 seconds

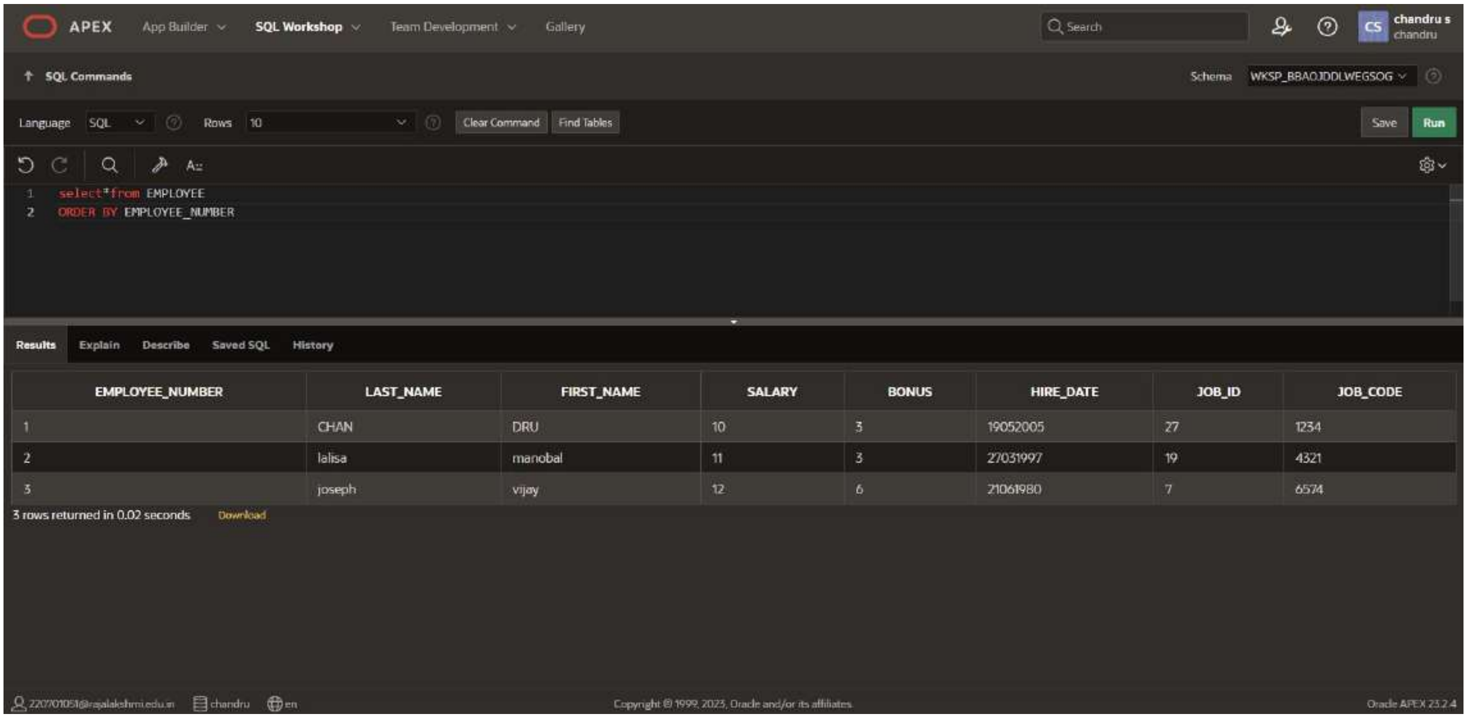
Download
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

Select*from employee order by employee_number;

OUTPUT:



```
select*from EMPLOYEE
ORDER BY EMPLOYEE_NUMBER
```

EMPLOYEE_NUMBER	LAST_NAME	FIRST_NAME	SALARY	BONUS	HIRE_DATE	JOB_ID	JOB_CODE
1	CHAN	DRU	10	3	19052005	27	1234
2	lalisa	manobal	11	3	27031997	19	4321
3	joseph	vijay	12	6	21061980	7	6574

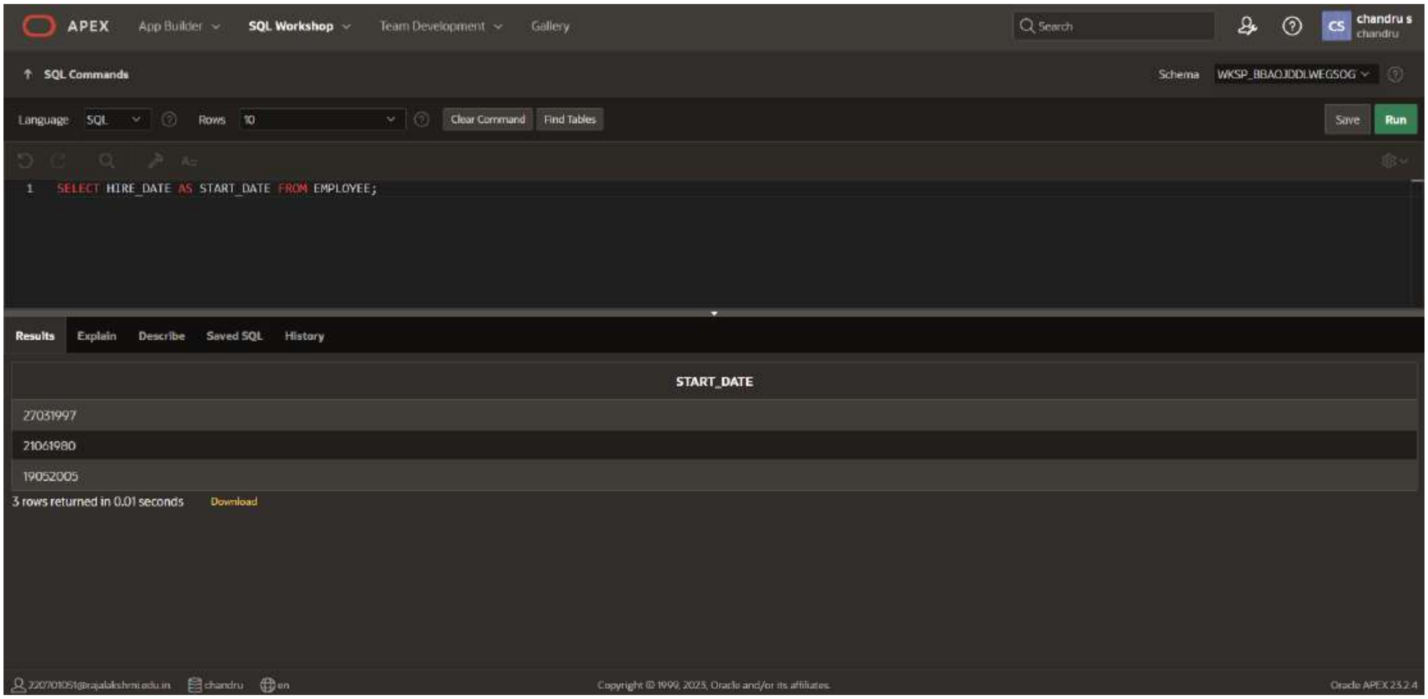
3 rows returned in 0.02 seconds

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 last_name,job_id,hire_date,emp_id from employees;

OUTPUT:



```
SELECT HIRE_DATE AS START_DATE FROM EMPLOYEE;
```

START_DATE
27031997
21061980
19052005

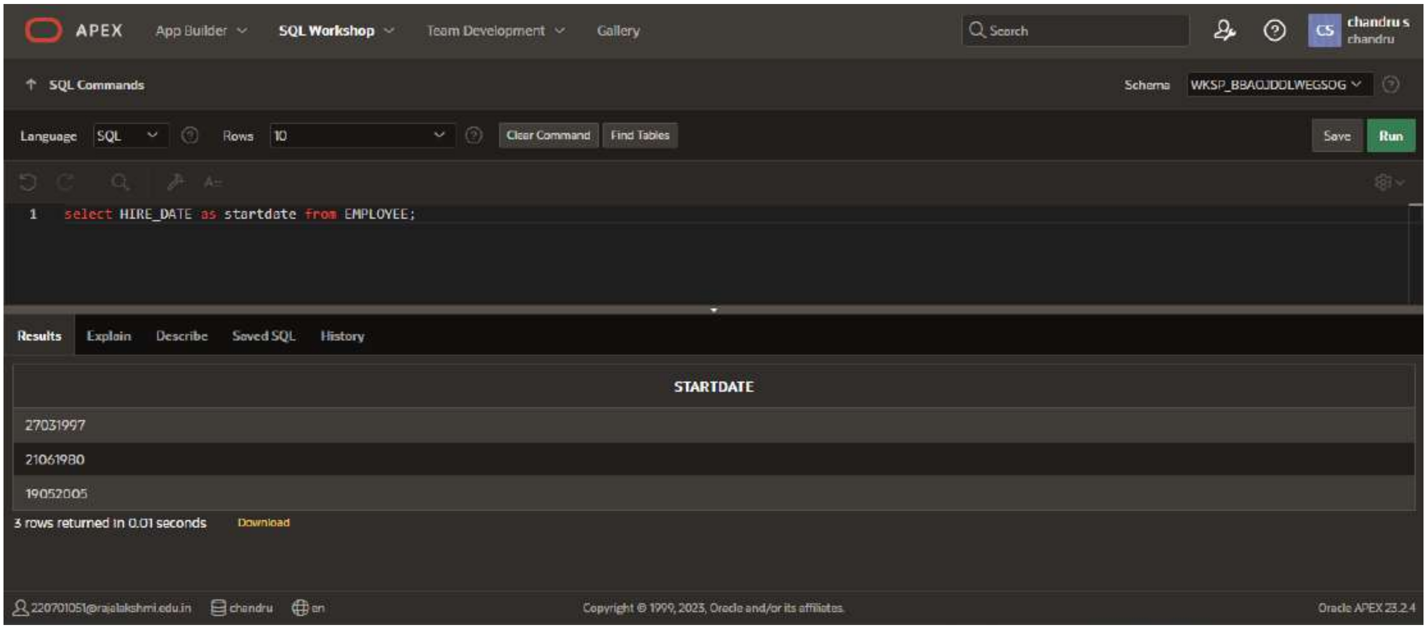
3 rows returned in 0.01 seconds

4.Provide an alias STARTDATE for the hire date.

QUERY:

Select hire_date as “STARTDATE” from employee;

OUTPUT:



```
select HIRE_DATE as startdate from EMPLOYEE;
```

STARTDATE
27031997
21061980
19052005

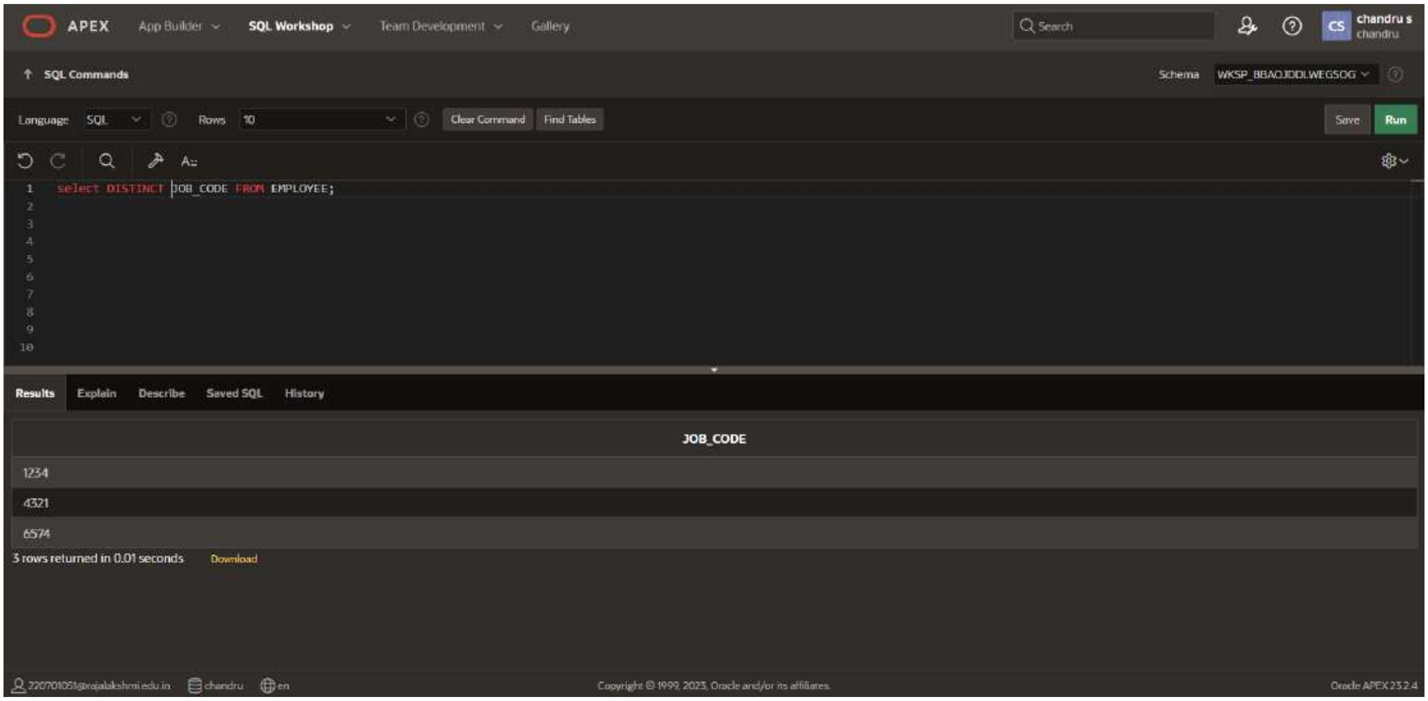
3 rows returned in 0.01 seconds

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

QUERY:

select distinct job_id from employee;

OUTPUT:



```
select DISTINCT JOB_CODE FROM EMPLOYEE;
```

JOB_CODE
1234
4321
6574

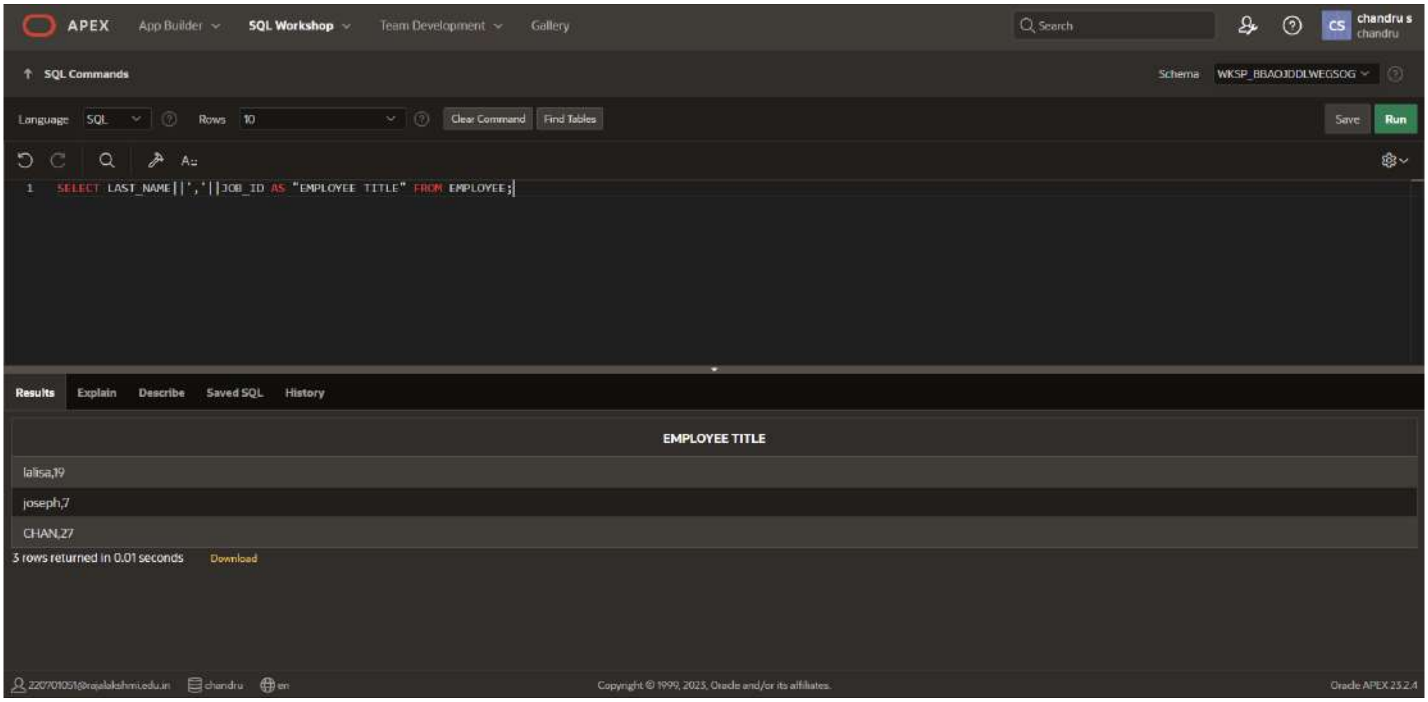
3 rows returned in 0.01 seconds

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

QUERY:

Select last_name||’,’||job_id as “EMPLOYEE_AND_TITLE” from employee;

OUTPUT:



```
SELECT LAST_NAME||','||JOB_ID AS "EMPLOYEE TITLE" FROM EMPLOYEE;
```

EMPLOYEE TITLE
lalisa,19
joseph,7
CHAN,27

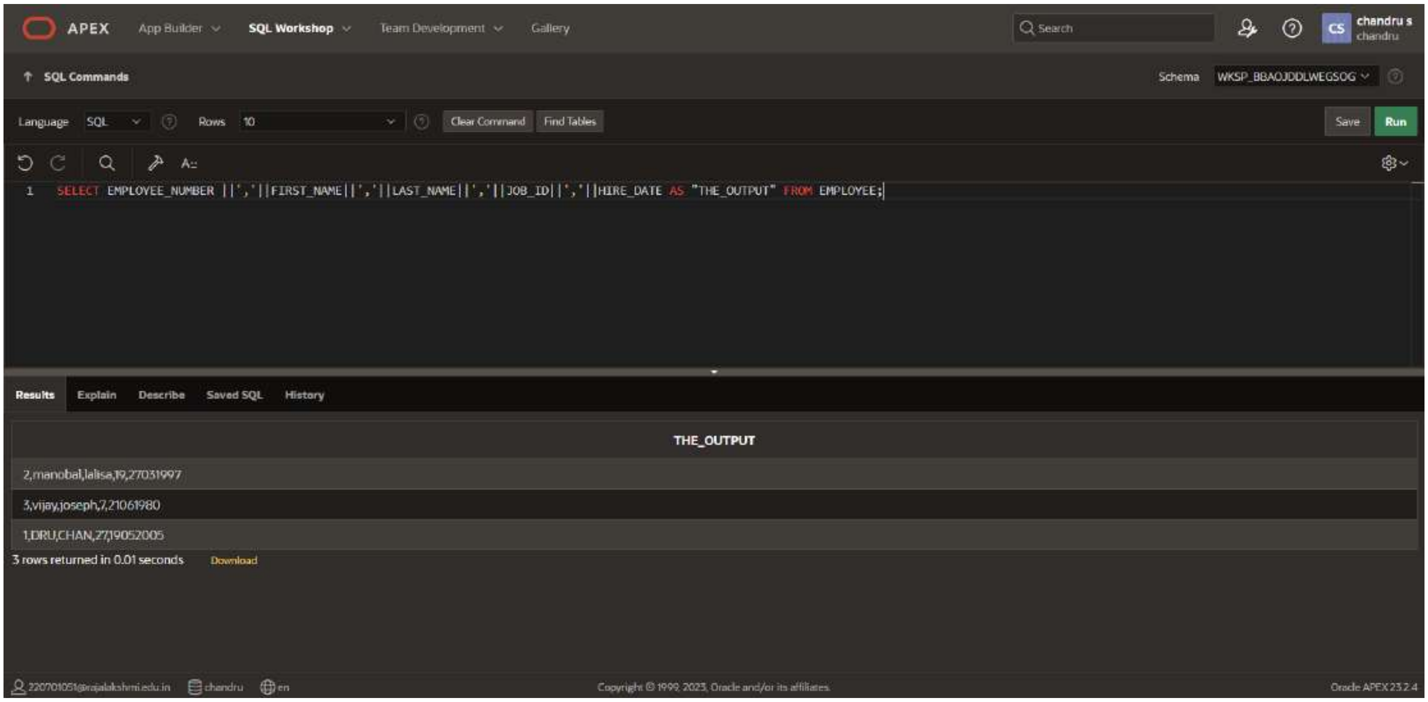
3 rows returned in 0.01 seconds

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

QUERY:

Select employee_id||','||first_name||','||last_name||','||email||','||phone_no||','||hire_date||','||job_id||','||salary||','||commission_pct||','||manager_id||','||department_id as THE_OUTPUT from employees;

OUTPUT:



Language: SQL Rows: 10

```
SELECT EMPLOYEE_NUMBER ||','||FIRST_NAME||','||LAST_NAME||','||JOB_ID||','||HIRE_DATE AS "THE_OUTPUT" FROM EMPLOYEE;
```

```
THE_OUTPUT
2,manobal,lalisa,19,27031997
3,vijay,joseph,7,21061980
1,DRU,CHAN,27,19052005
3 rows returned in 0.01 seconds
```

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

RESULT:

RESTRICTING AND SORTING DATA

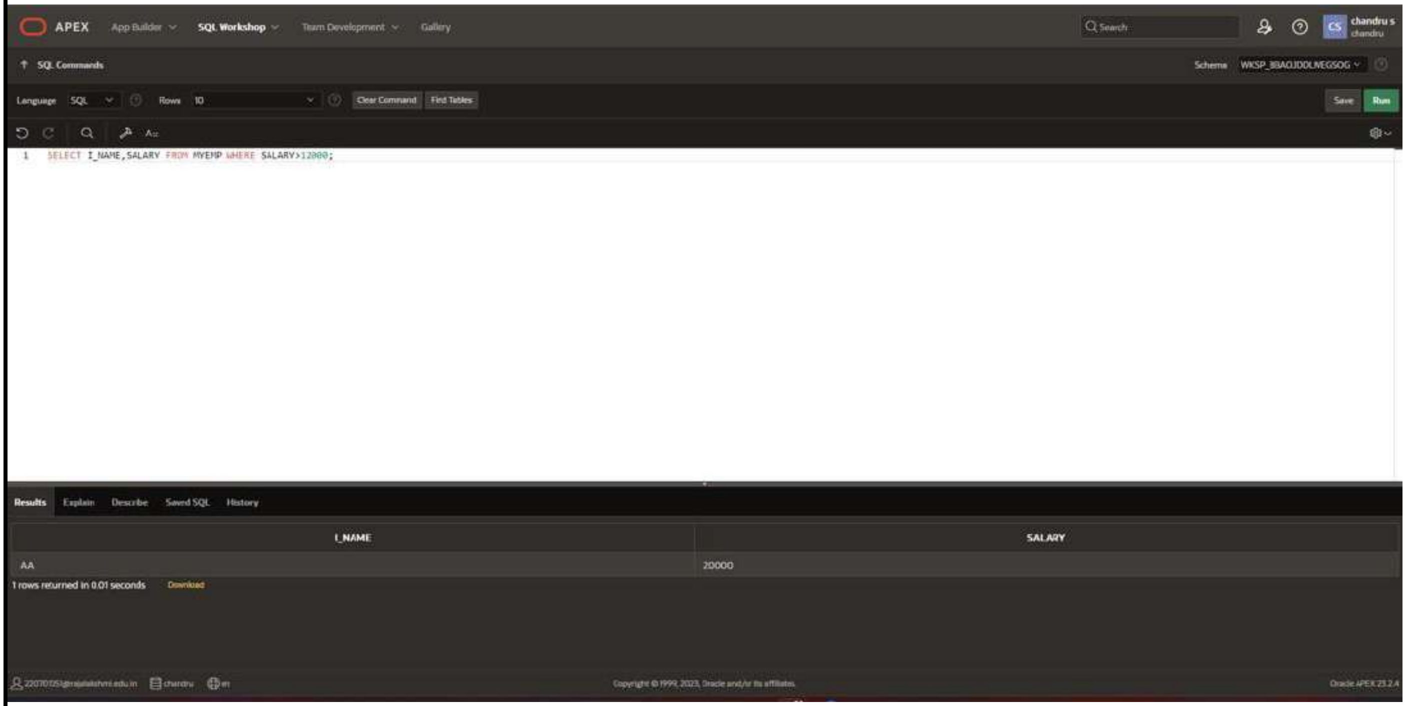
EX_NO:5 **DATE:**

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

QUERY:

select last_name,salary from myemp where salary >12000;

OUTPUT:



```
SELECT I_NAME,SALARY FROM MYEMP WHERE SALARY>12000;
```

I_NAME	SALARY
AA	20000

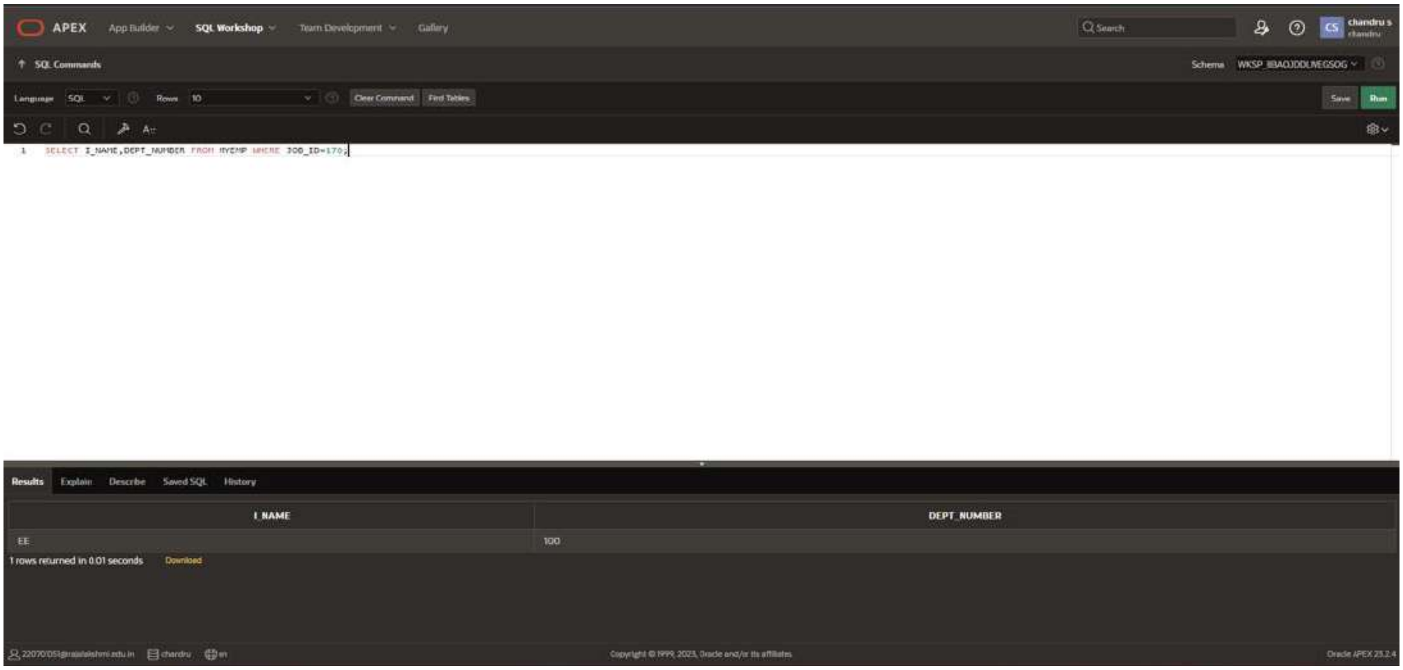
1 rows returned in 0.01 seconds

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

QUERY:

select last_name,department_number from myemp where emp_id=176;

OUTPUT:



```
SELECT I_NAME,DEPT_NUMBER FROM MYEMP WHERE JOB_ID=176;
```

I_NAME	DEPT_NUMBER
EE	100

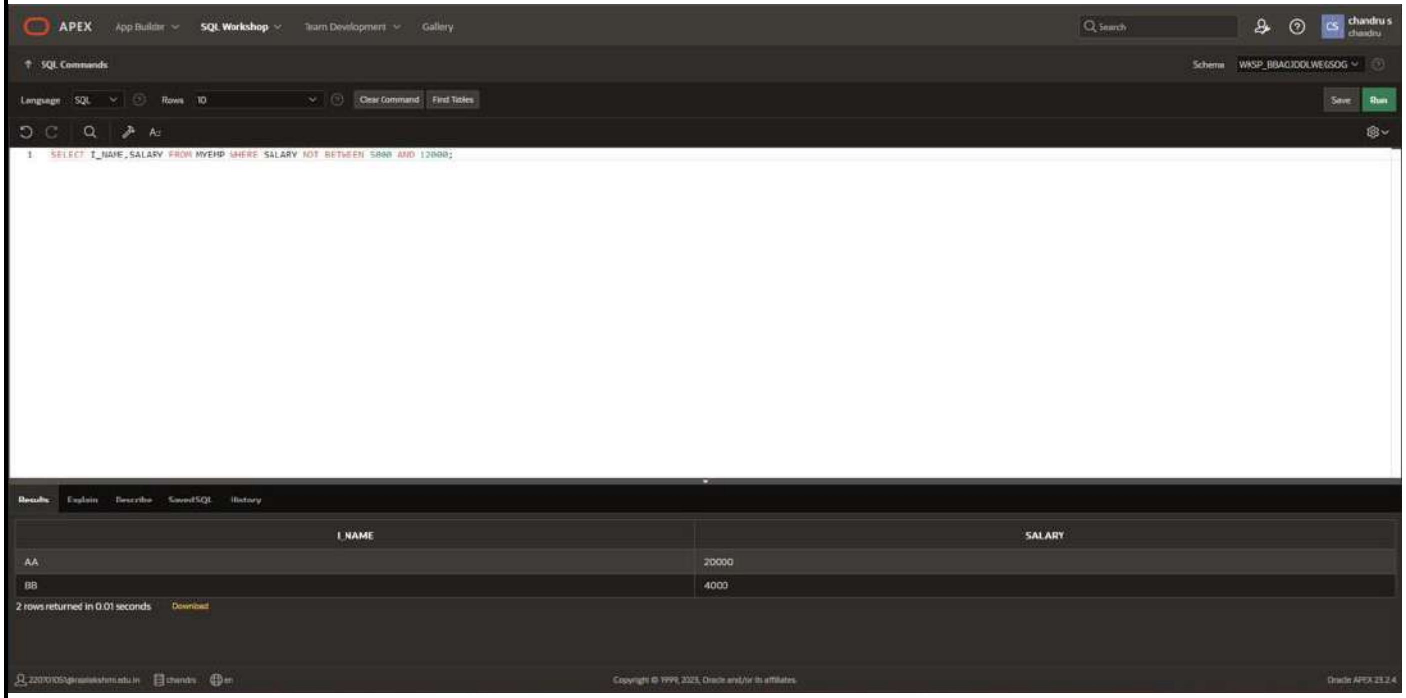
1 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



```
SELECT I_NAME,SALARY FROM MYEMP WHERE SALARY NOT BETWEEN 5000 AND 12000;
```

I_NAME	SALARY
AA	20000
BB	4000

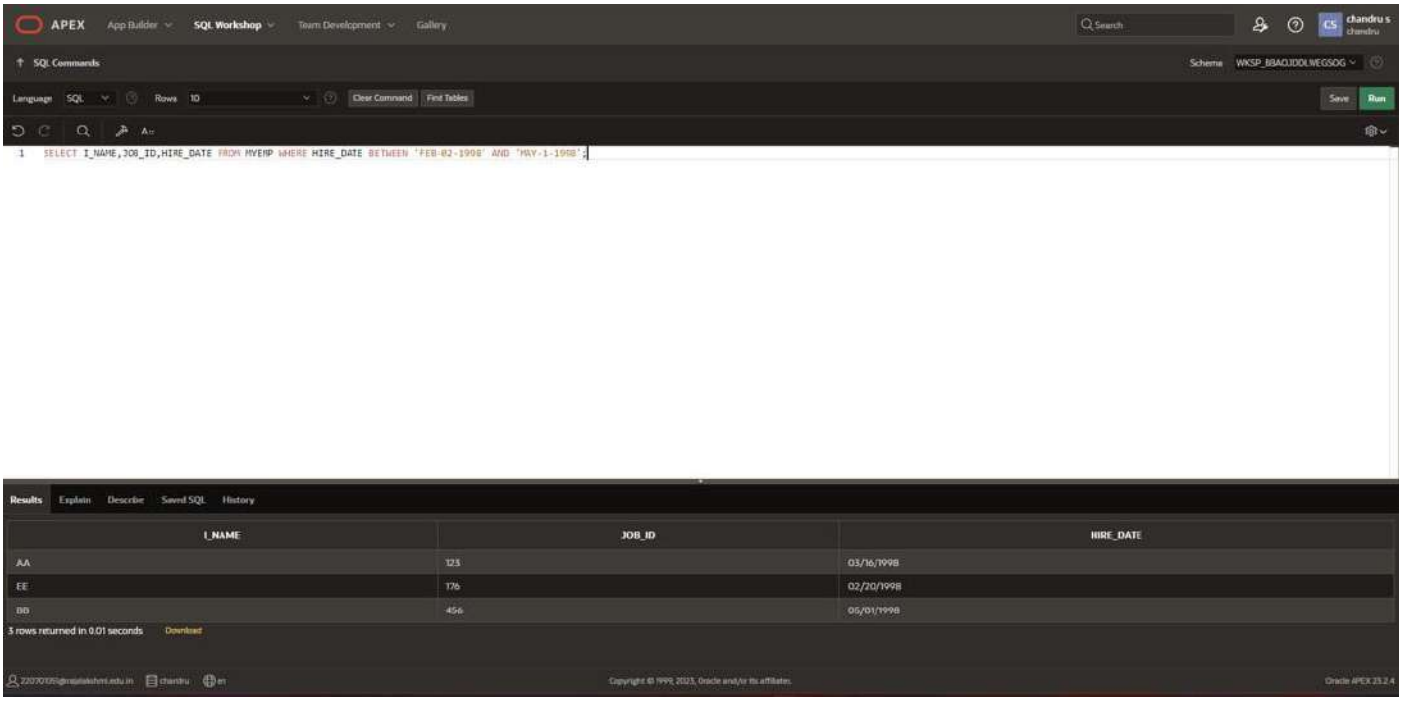
2 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



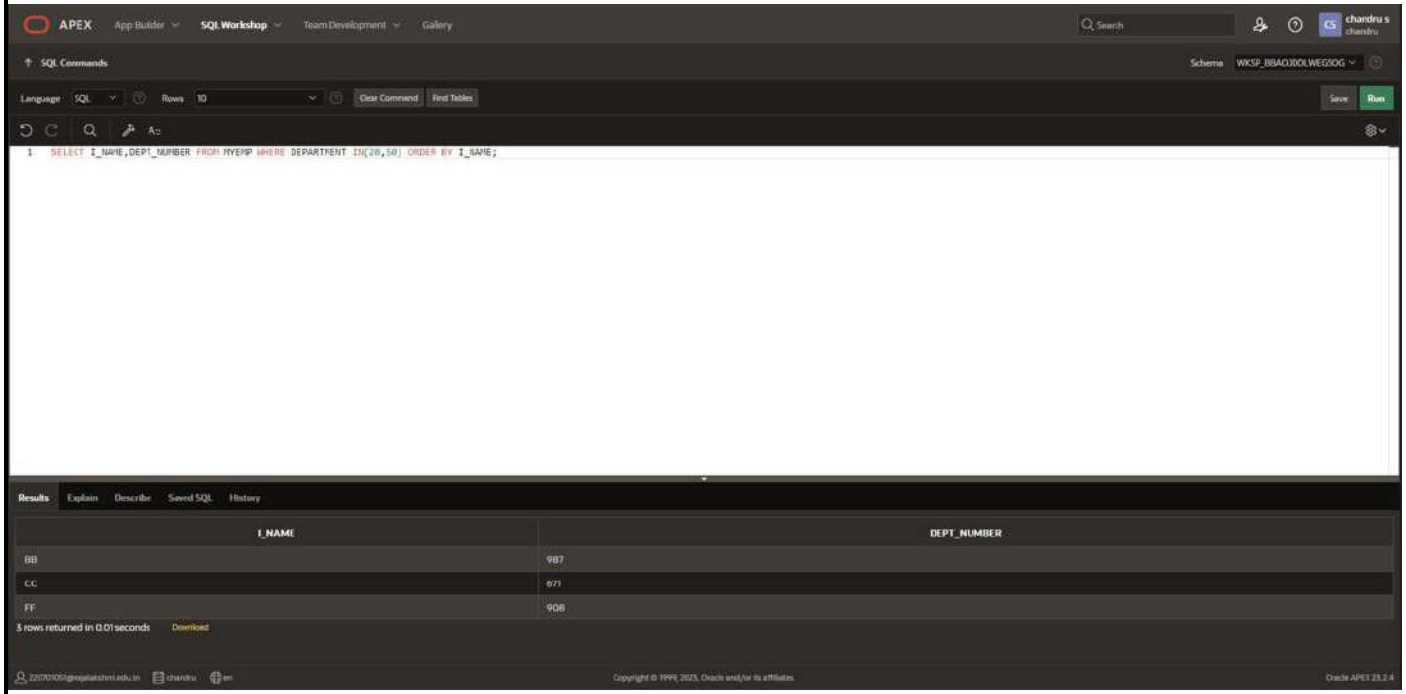
```
SELECT I_NAME,JOB_ID,HIRE_DATE FROM MYEMP WHERE HIRE_DATE BETWEEN 'FEB-02-1998' AND 'MAY-1-1998';
```

I_NAME	JOB_ID	HIRE_DATE
AA	123	03/16/1998
EE	176	02/20/1998
BB	456	05/01/1998

3 rows returned in 0.01 seconds

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

select last_name, department_number from myemp where department in (20,50) order by last_name asc; **OUTPUT:**



```
SELECT I_NAME,DEPT_NUMBER FROM MYEMP WHERE DEPARTMENT IN(20,50) ORDER BY I_NAME;
```

I_NAME	DEPT_NUMBER
BB	987
CC	671
FF	908

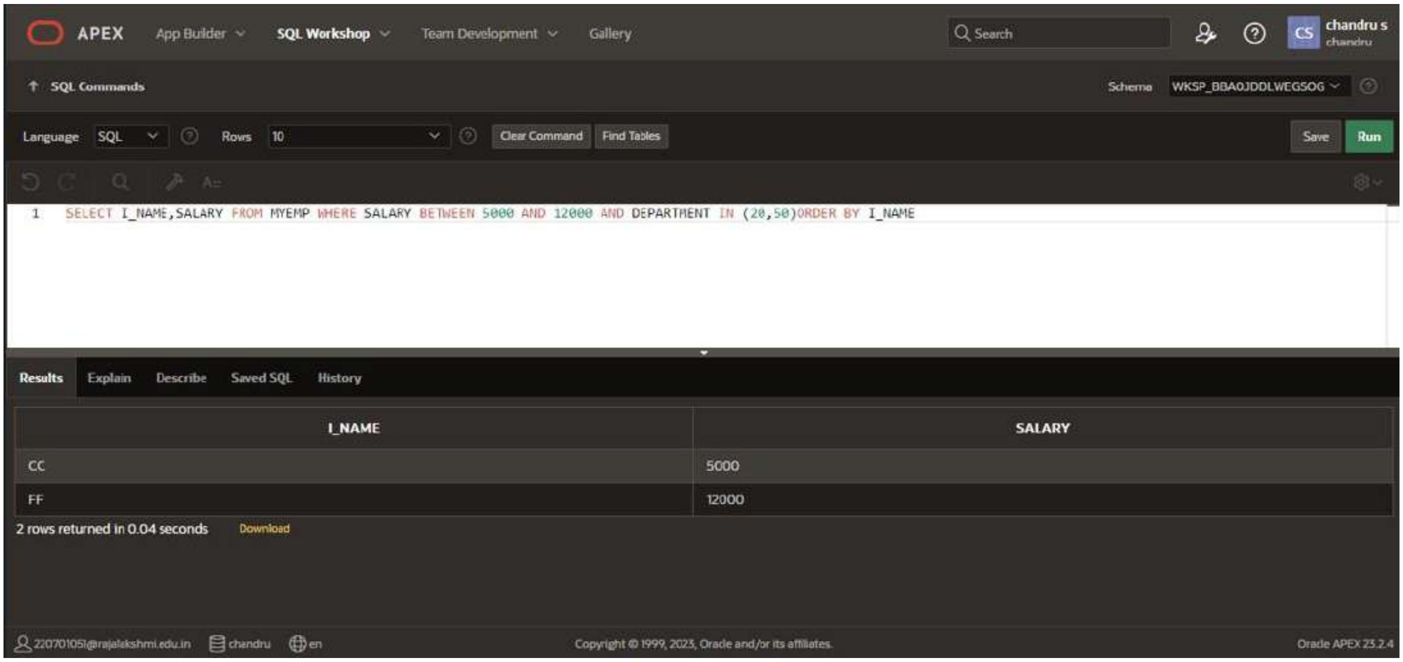
3 rows returned in 0.01 seconds

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

QUERY:

Select last_name as"EMPLOYEE",salary as "MONTHLY_SALARY" from myemp where salary between 5000 and 2000 and department in(20,50) order by Name_emp asc;

OUTPUT:



```
SELECT I_NAME,SALARY FROM MYEMP WHERE SALARY BETWEEN 5000 AND 12000 AND DEPARTMENT IN (20,50)ORDER BY I_NAME
```

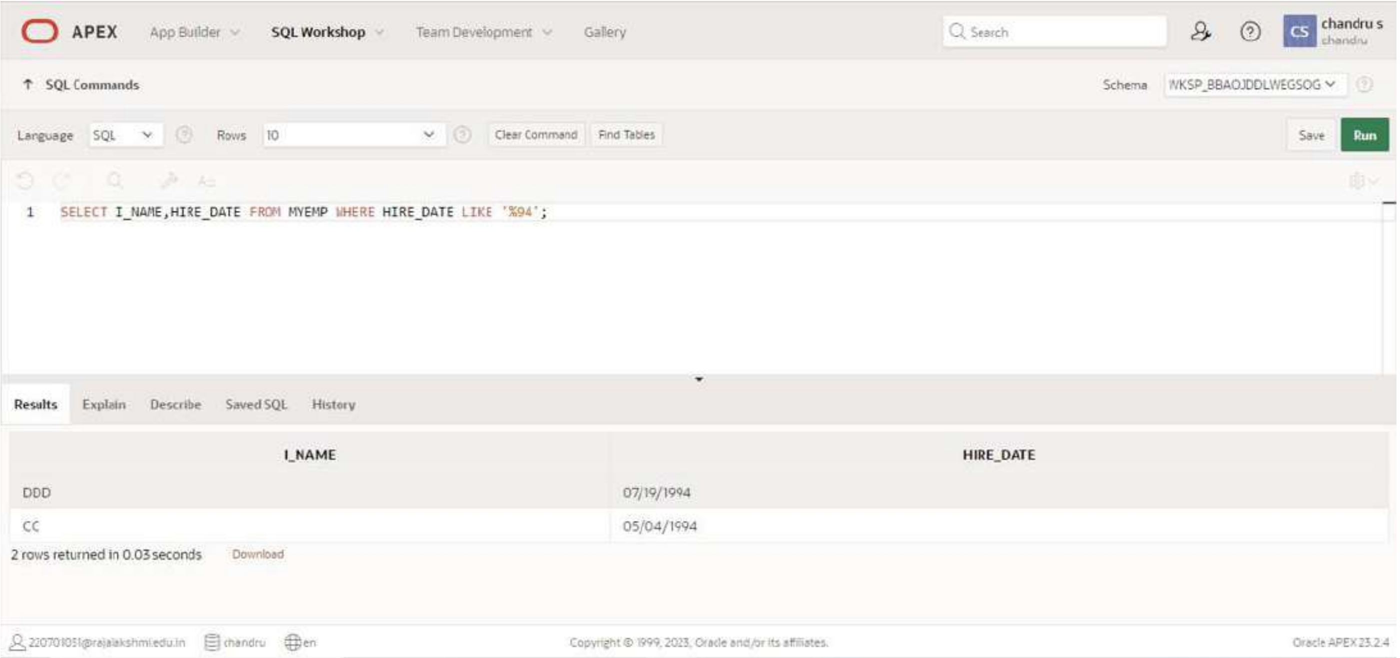
I_NAME	SALARY
CC	5000
FF	12000

2 rows returned in 0.04 seconds

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

QUERY:

SELECT last_name, hire_date FROM myemp WHERE hire_date like '%94'; **OUTPUT:**



```
SELECT I_NAME,HIRE_DATE FROM MYEMP WHERE HIRE_DATE LIKE '%94';
```

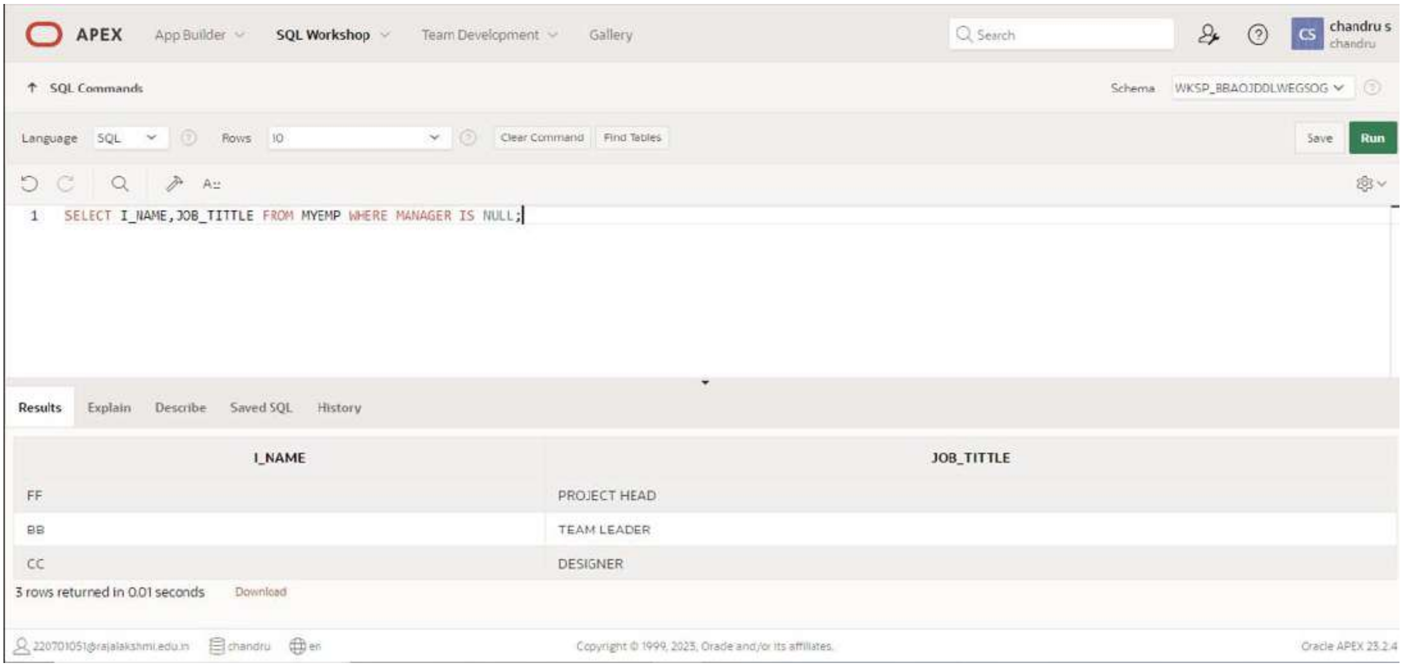
I_NAME	HIRE_DATE
DDD	07/19/1994
CC	05/04/1994

2 rows returned in 0.03 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 myemp WHERE manager_id IS NULL; **OUTPUT:**



```
SELECT I_NAME,JOB_TITTLE FROM MYEMP WHERE MANAGER IS NULL;
```

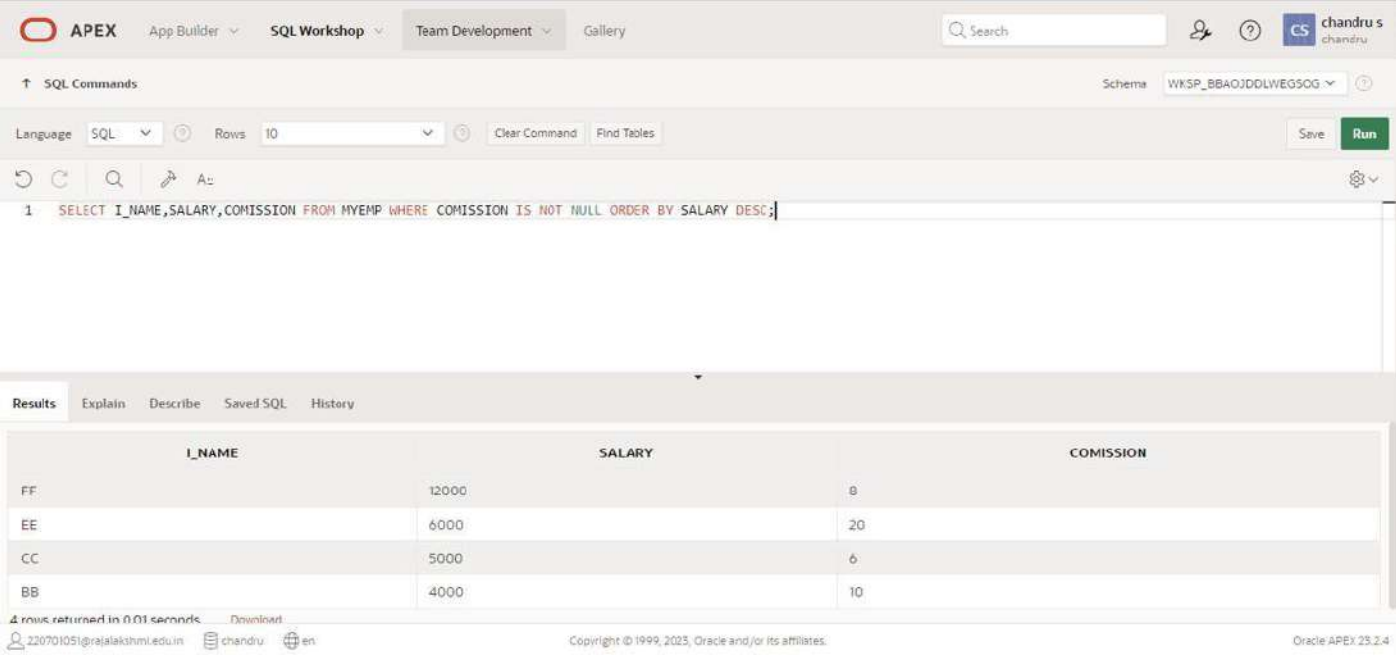
I_NAME	JOB_TITTLE
FF	PROJECT HEAD
BB	TEAM LEADER
CC	DESIGNER

3 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 myemp WHERE commission_pct IS NOT NULL ORDER BY salary DESC, commission_pct DESC; **OUTPUT:**



```
SELECT I_NAME,SALARY,COMISSION FROM MYEMP WHERE COMISSION IS NOT NULL ORDER BY SALARY DESC;
```

I_NAME	SALARY	COMISSION
FF	12000	8
EE	6000	20
CC	5000	6
BB	4000	10

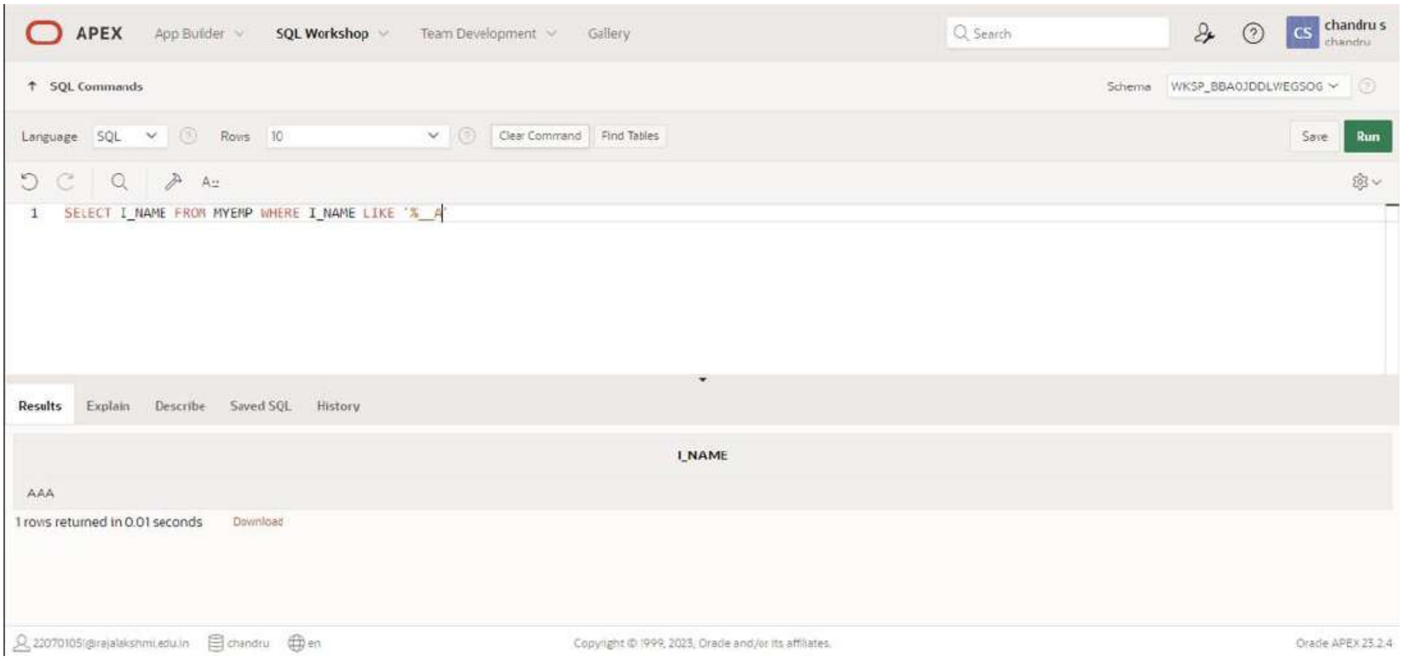
4 rows returned in 0.01 seconds Download

220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

SELECT last_name FROM myemp WHERE last_name LIKE '__a%'; **OUTPUT:**



```
SELECT I_NAME FROM MYEMP WHERE I_NAME LIKE '%__A'
```

I_NAME
AAA

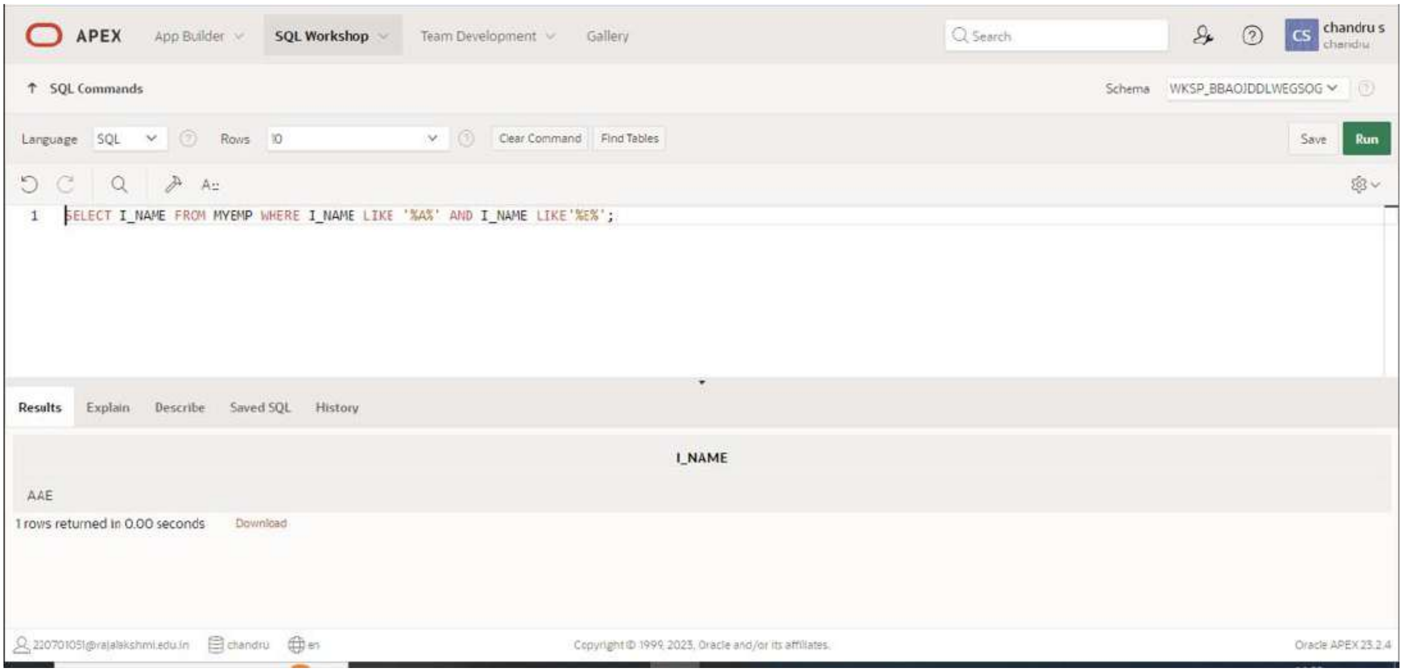
1 rows returned in 0.01 seconds Download

220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

SELECT last_name FROM myemp WHERE last_name LIKE '%a%' AND last_name LIKE '%e%'; **OUTPUT:**



```
SELECT I_NAME FROM MYEMP WHERE I_NAME LIKE '%A%' AND I_NAME LIKE'%E%';
```

I_NAME
AAE

1 rows returned in 0.00 seconds   Download

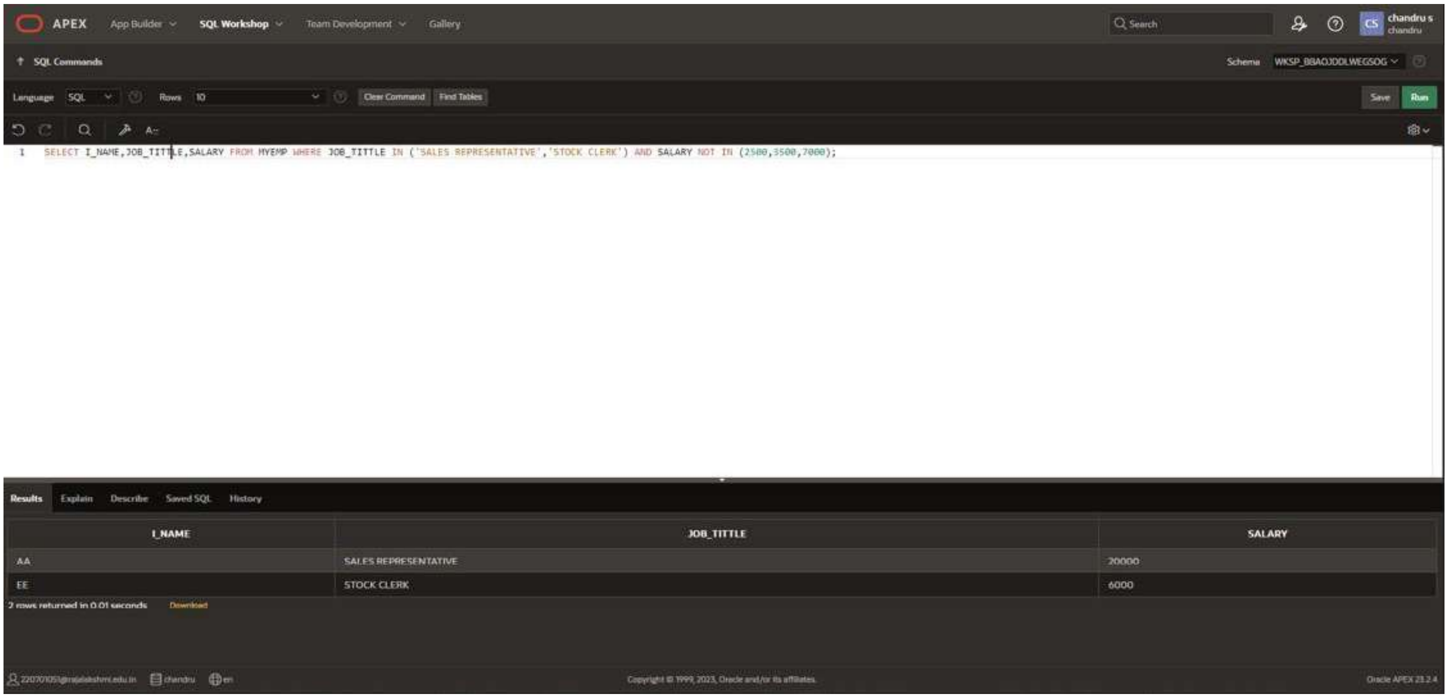
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



```
SELECT I_NAME,JOB_TITTLE,SALARY FROM MYEMP WHERE JOB_TITTLE IN ('SALES REPRESENTATIVE','STOCK CLERK') AND SALARY NOT IN (2500,3500,7000);
```

I_NAME	JOB_TITTLE	SALARY
AA	SALES REPRESENTATIVE	20000
EE	STOCK CLERK	6000

2 rows returned in 0.01 seconds   Download

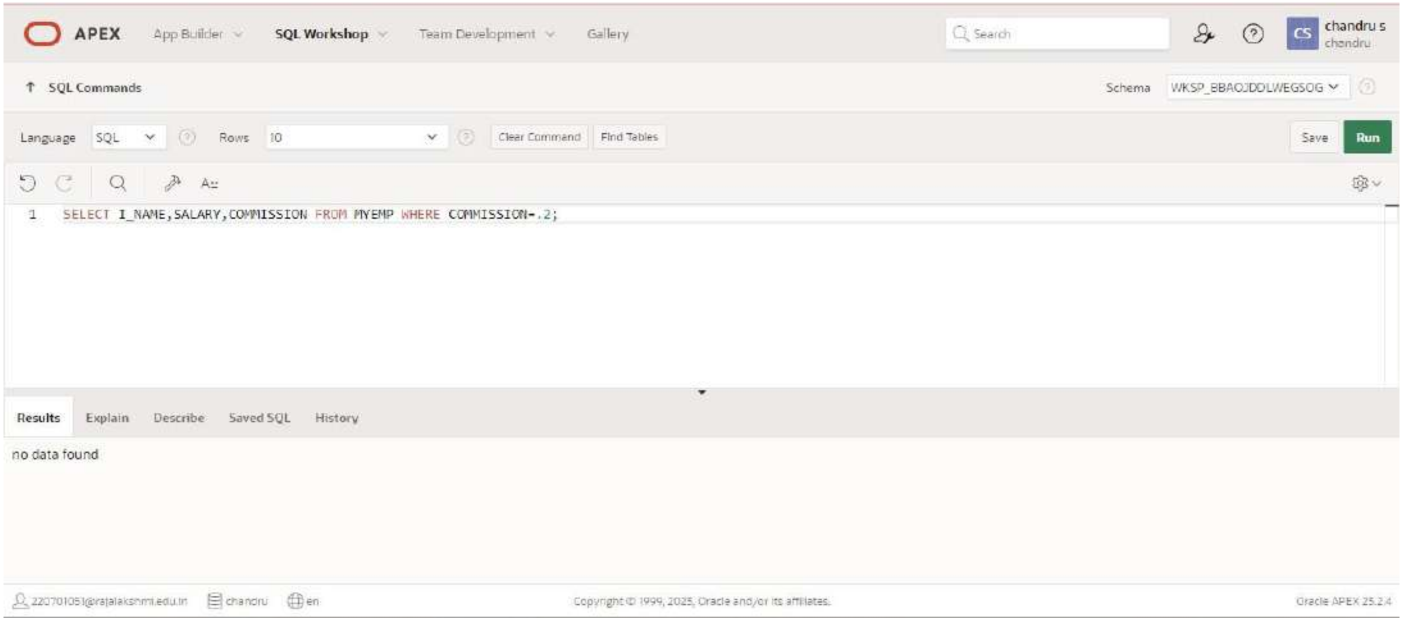
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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 myemp where commission_pct=0.2;

OUTPUT:



```
SELECT I_NAME,SALARY,COMMISSION FROM MYEMP WHERE COMMISSION=.2;
```

```
no data found
```

220701051@rajalakshmi.edu.in chandru en

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

Oracle APEX 25.2.4

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

RESULT:

SINGLE ROW FUNCTIONS

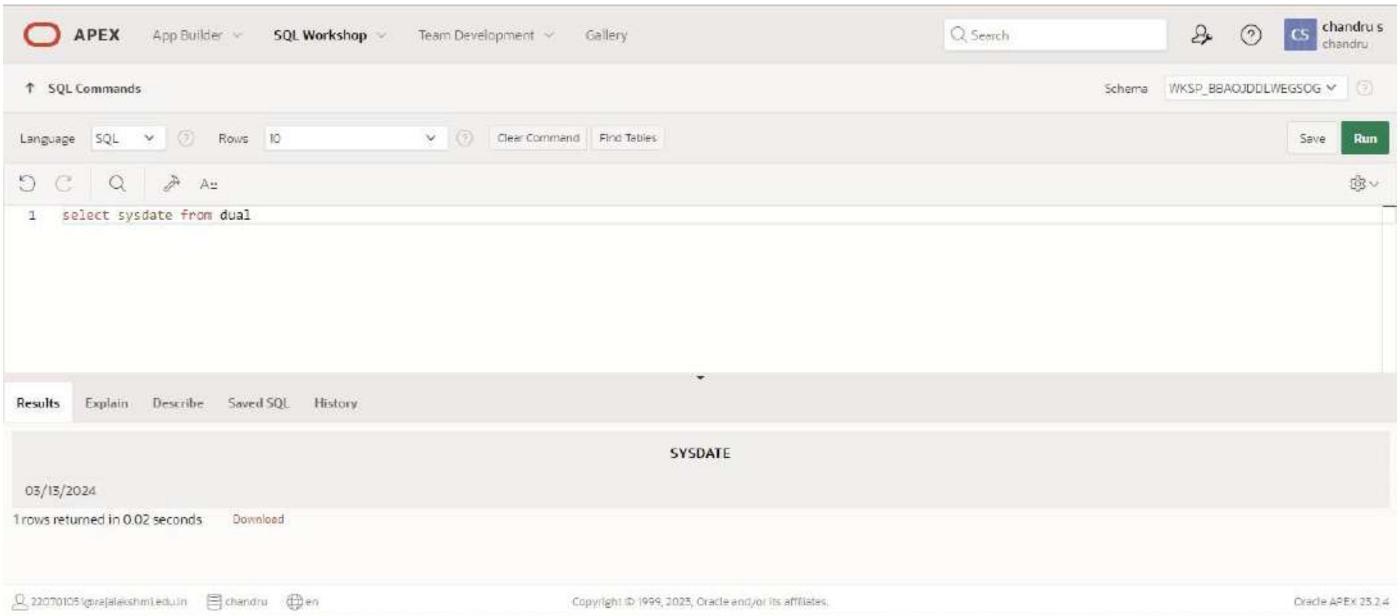
EX_NO:6 **DATE:**

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

QUERY:

select sysdate from dual;

OUTPUT:



```
select sysdate from dual
```

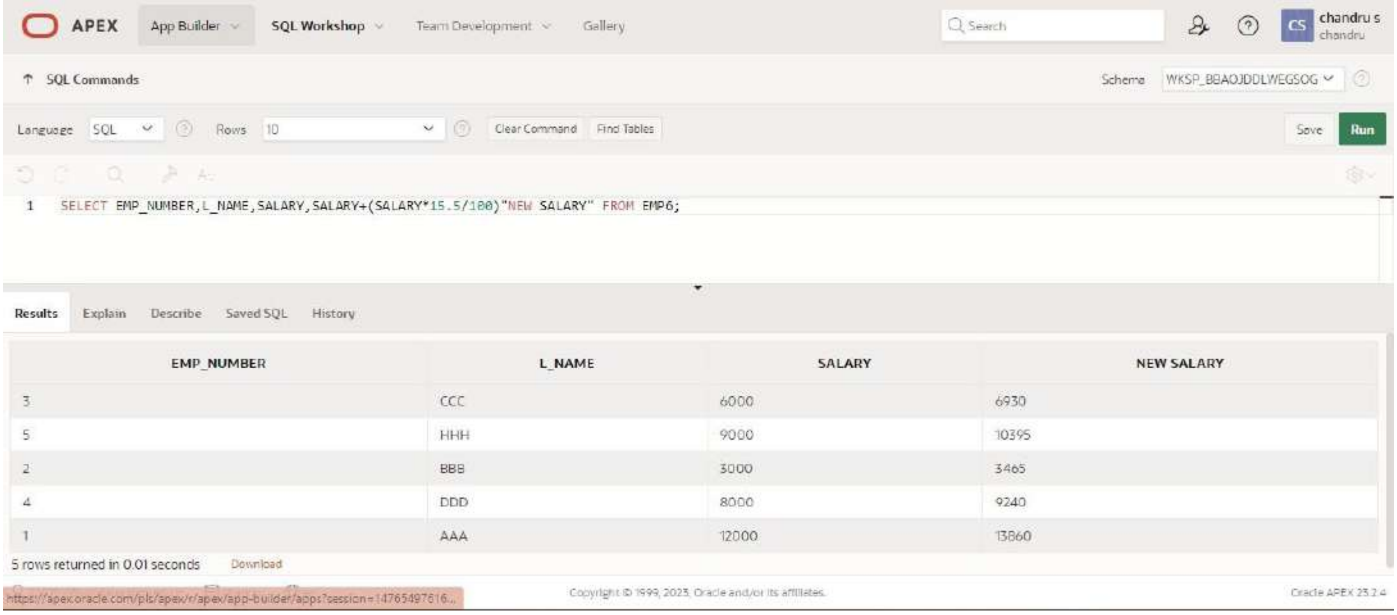
```
SYSDATE
03/13/2024
1 rows returned in 0.02 seconds
```

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

QUERY:

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

OUTPUT:



```
SELECT EMP_NUMBER,L_NAME,SALARY,SALARY+(SALARY*15.5/100)"NEW SALARY" FROM EMP6;
```

EMP_NUMBER	L_NAME	SALARY	NEW SALARY
3	CCC	6000	6930
5	HHH	9000	10395
2	BBB	3000	3465
4	DDD	8000	9240
1	AAA	12000	13860

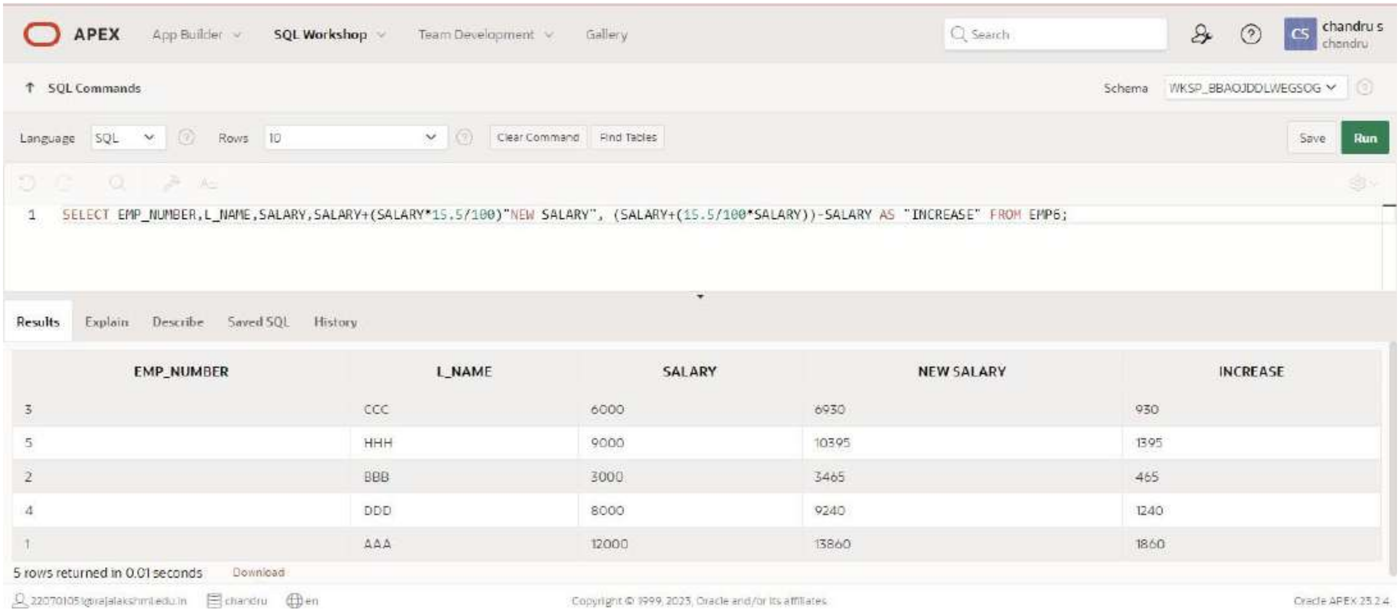
5 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



```
SELECT EMP_NUMBER,L_NAME,SALARY,SALARY+(SALARY*15.5/100)"NEW SALARY", (SALARY+(15.5/100*SALARY))-SALARY AS "INCREASE" FROM EMP6;
```

EMP_NUMBER	L_NAME	SALARY	NEW SALARY	INCREASE
3	CCC	6000	6930	930
5	HHH	9000	10395	1395
2	BBB	3000	3465	465
4	DDD	8000	9240	1240
1	AAA	12000	13860	1860

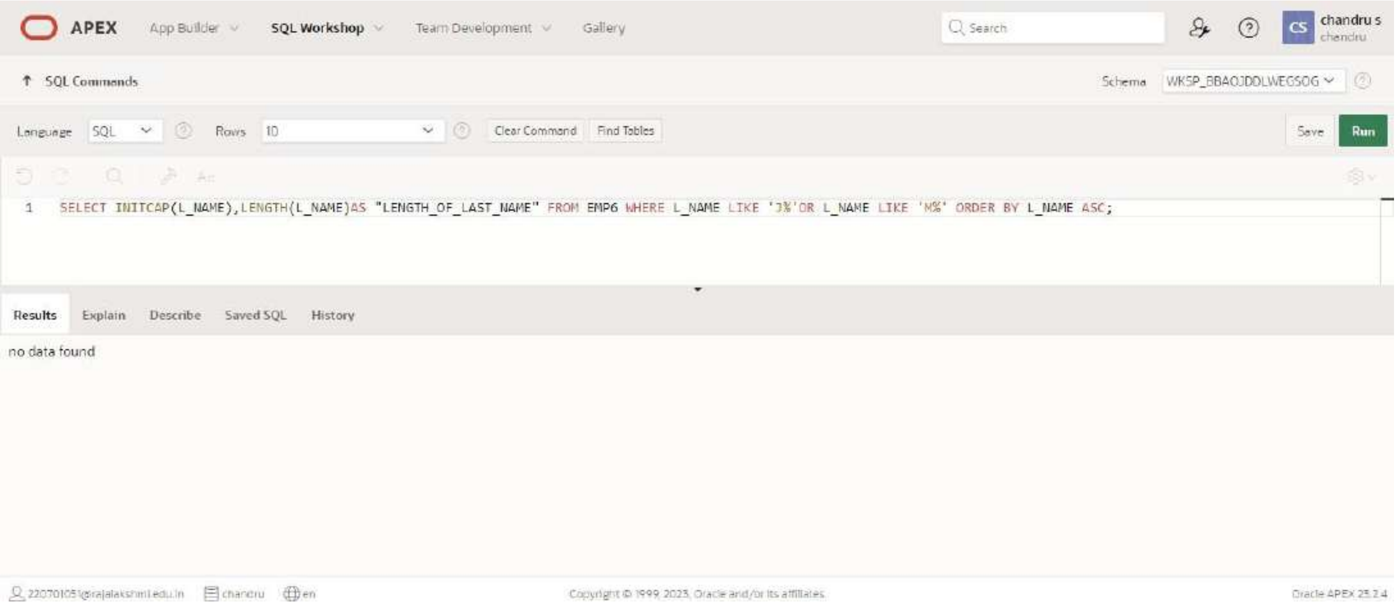
5 rows returned in 0.01 seconds

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

QUERY:

select initcap(last_name),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:



```
SELECT INITCAP(L_NAME),LENGTH(L_NAME)AS "LENGTH_OF_LAST_NAME" FROM EMP6 WHERE L_NAME LIKE 'J%'OR L_NAME LIKE 'M%' ORDER BY L_NAME ASC;
```

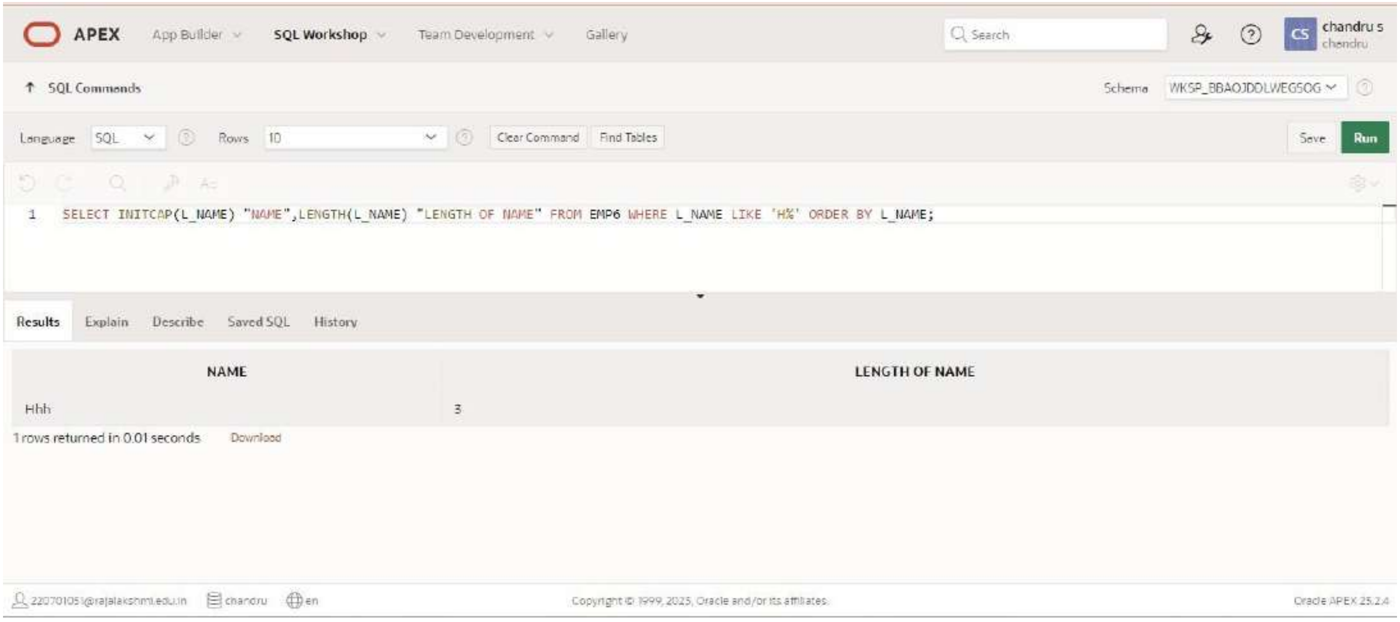
no data found

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:



```
SELECT INITCAP(L_NAME) "NAME",LENGTH(L_NAME) "LENGTH OF NAME" FROM EMP6 WHERE L_NAME LIKE 'H%' ORDER BY L_NAME;
```

NAME	LENGTH OF NAME
Hhh	3

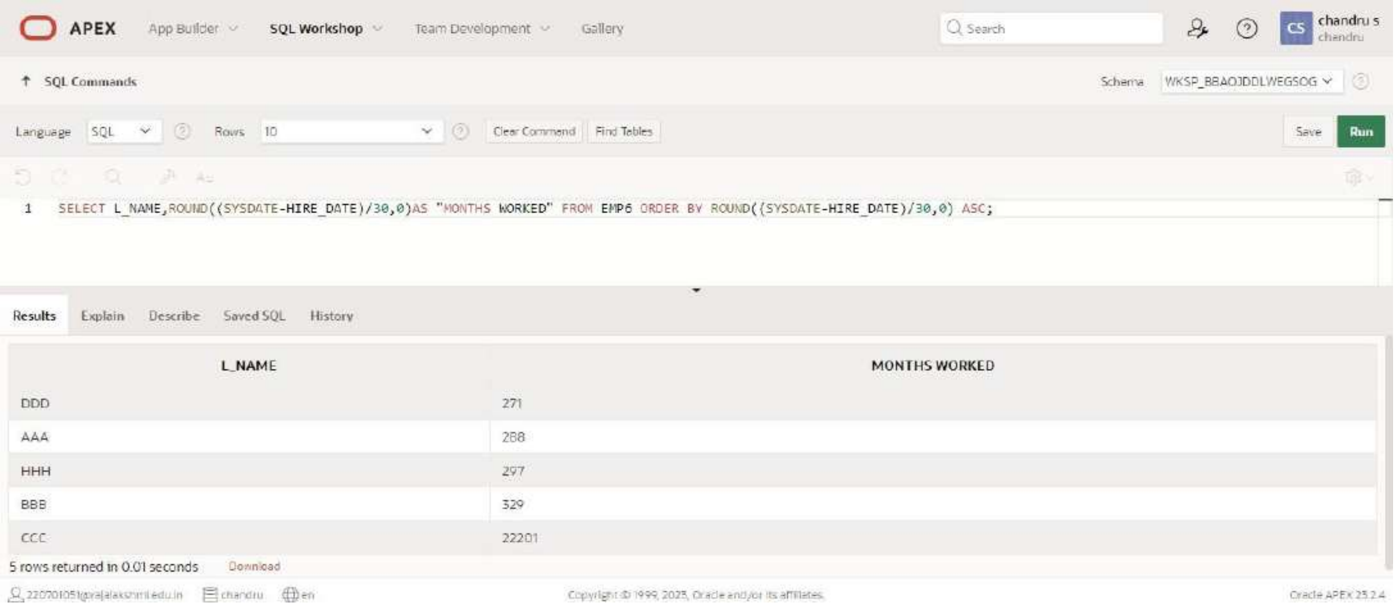
1 rows returned in 0.01 seconds Download

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 byround((sysdate-hire_date)/30,0) asc;

OUTPUT:



```
SELECT L_NAME,ROUND((SYSDATE-HIRE_DATE)/30,0)AS "MONTHS WORKED" FROM EMP6 ORDER BY ROUND((SYSDATE-HIRE_DATE)/30,0) ASC;
```

L_NAME	MONTHS WORKED
DDD	271
AAA	288
HHH	297
BBB	329
CCC	22201

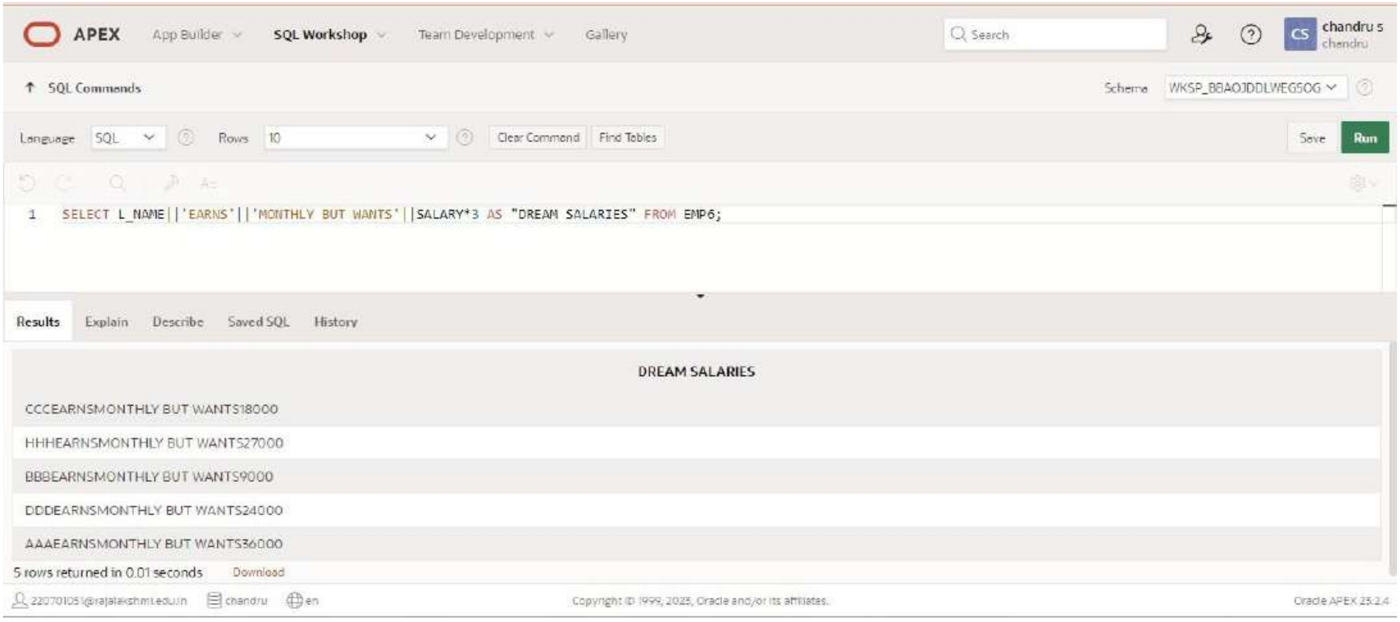
5 rows returned in 0.01 seconds Download

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

QUERY:

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

OUTPUT:



```
SELECT L_NAME||'EARNS'||'MONTHLY BUT WANTS'||SALARY*3 AS "DREAM SALARIES" FROM EMP6;
```

DREAM SALARIES
CCCEARNSMONTHLY BUT WANTS18000
HHHEARNSMONTHLY BUT WANTS27000
BBBEARNSMONTHLY BUT WANTS9000
DDDEARNSMONTHLY BUT WANTS24000
AAAEARNSMONTHLY BUT WANTS36000

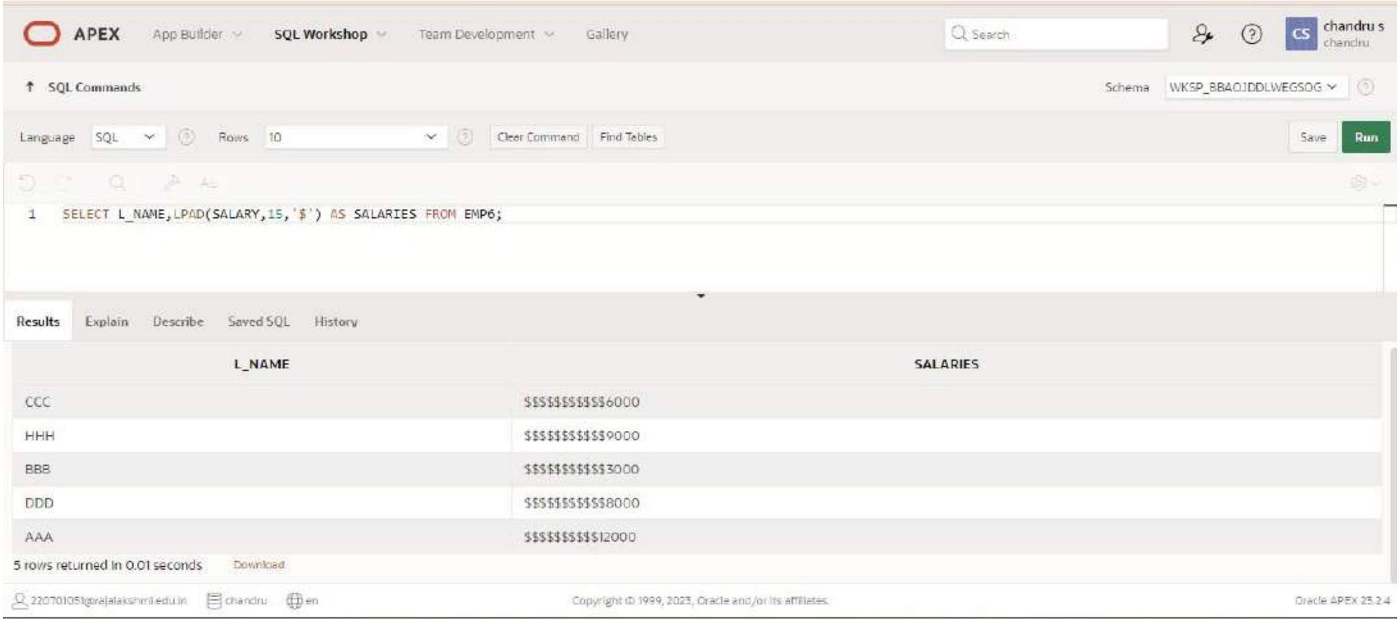
5 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



```
SELECT L_NAME,LPAD(SALARY,15,'$') AS SALARIES FROM EMP6;
```

L_NAME	SALARIES
CCC	$$$$$$$$$$$6000
HHH	$$$$$$$$$$$9000
BBB	$$$$$$$$$$$3000
DDD	$$$$$$$$$$$8000
AAA	$$$$$$$$$$12000

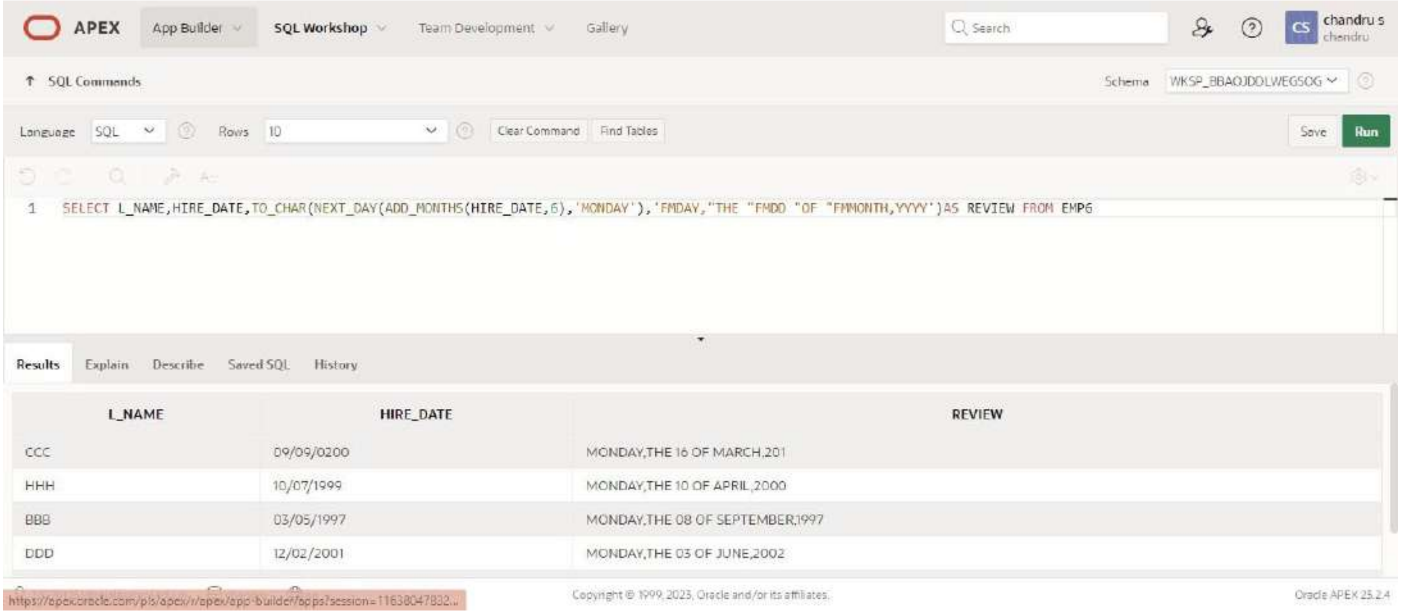
5 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



```
SELECT L_NAME,HIRE_DATE,TO_CHAR(NEXT_DAY(ADD_MONTHS(HIRE_DATE,6),'MONDAY'),'FMDAY,"THE "FMDD "OF "FMMONTH,YYYY')AS REVIEW FROM EMP6
```

L_NAME	HIRE_DATE	REVIEW
CCC	09/09/0200	MONDAY,THE 16 OF MARCH,201
HHH	10/07/1999	MONDAY,THE 10 OF APRIL,2000
BBB	03/05/1997	MONDAY,THE 08 OF SEPTEMBER,1997
DDD	12/02/2001	MONDAY,THE 03 OF JUNE,2002

https://apex.oracle.com/pls/apex/r/apex/app-builder/apps?session=11638047832...

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

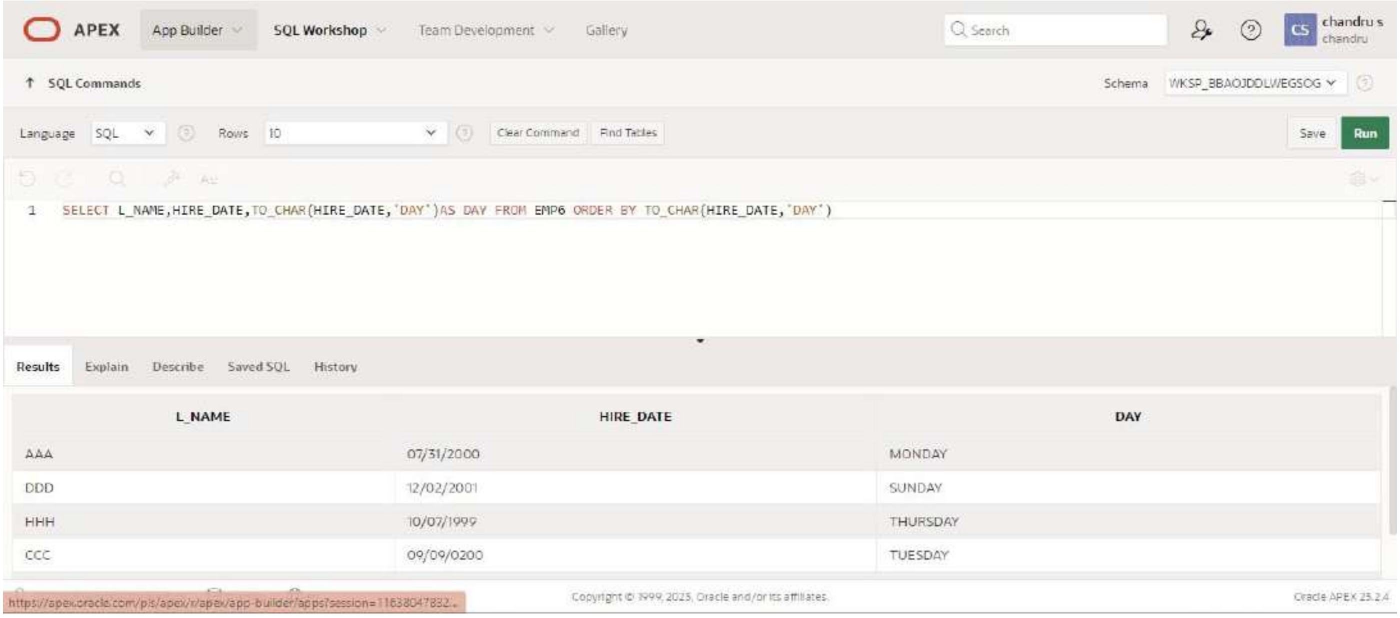
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



```
SELECT L_NAME,HIRE_DATE,TO_CHAR(HIRE_DATE,'DAY')AS DAY FROM EMP6 ORDER BY TO_CHAR(HIRE_DATE,'DAY')
```

L_NAME	HIRE_DATE	DAY
AAA	07/31/2000	MONDAY
DDD	12/02/2001	SUNDAY
HHH	10/07/1999	THURSDAY
CCC	09/09/0200	TUESDAY

https://apex.oracle.com/pls/apex/r/apex/app-builder/apps?session=11638047832...

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

Oracle APEX 23.2.4

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

RESULT:

DISPLAYING DATA FROM MULTIPLE TABLES

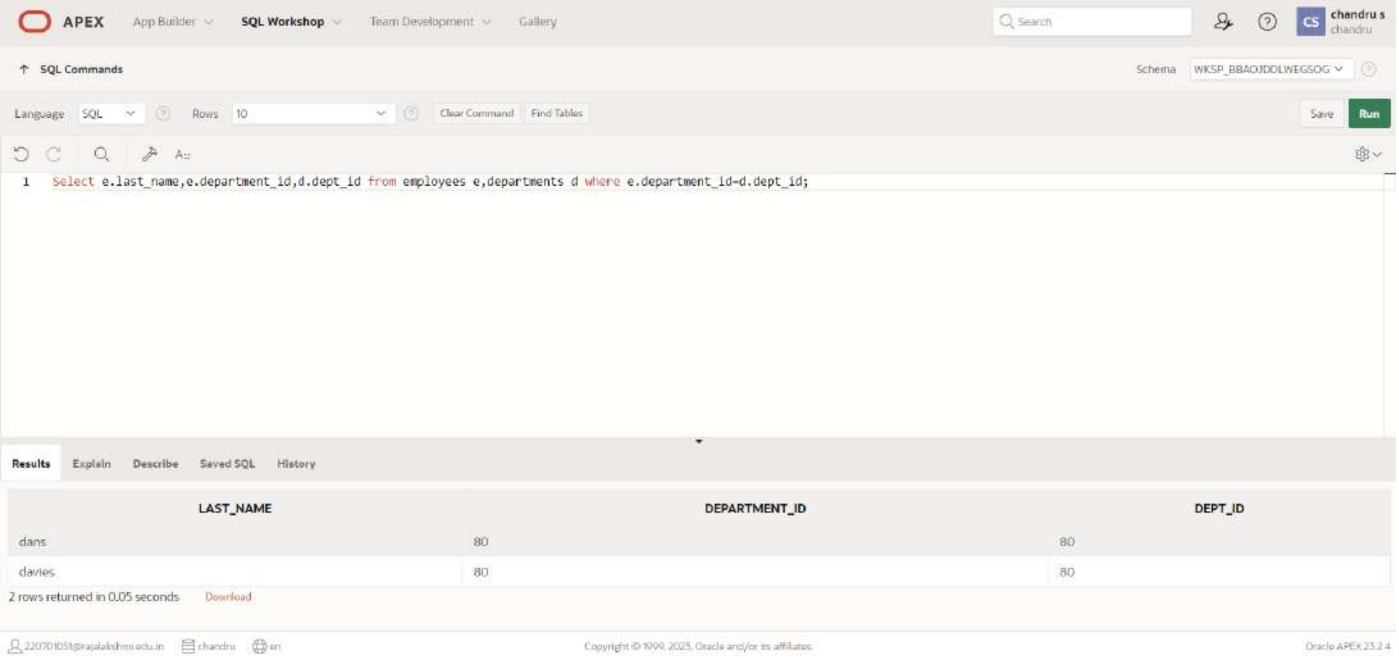
EX_NO:7 **DATE:**

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

QUERY:

Select e.last_name,e.department_id,d.department_id from employees e,departments d where e.department_id=d.department_id;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

```
Select e.last_name,e.department_id,d.dept_id from employees e,departments d where e.department_id=d.dept_id;
```

LAST_NAME	DEPARTMENT_ID	DEPT_ID
dans	80	80
davies	80	80

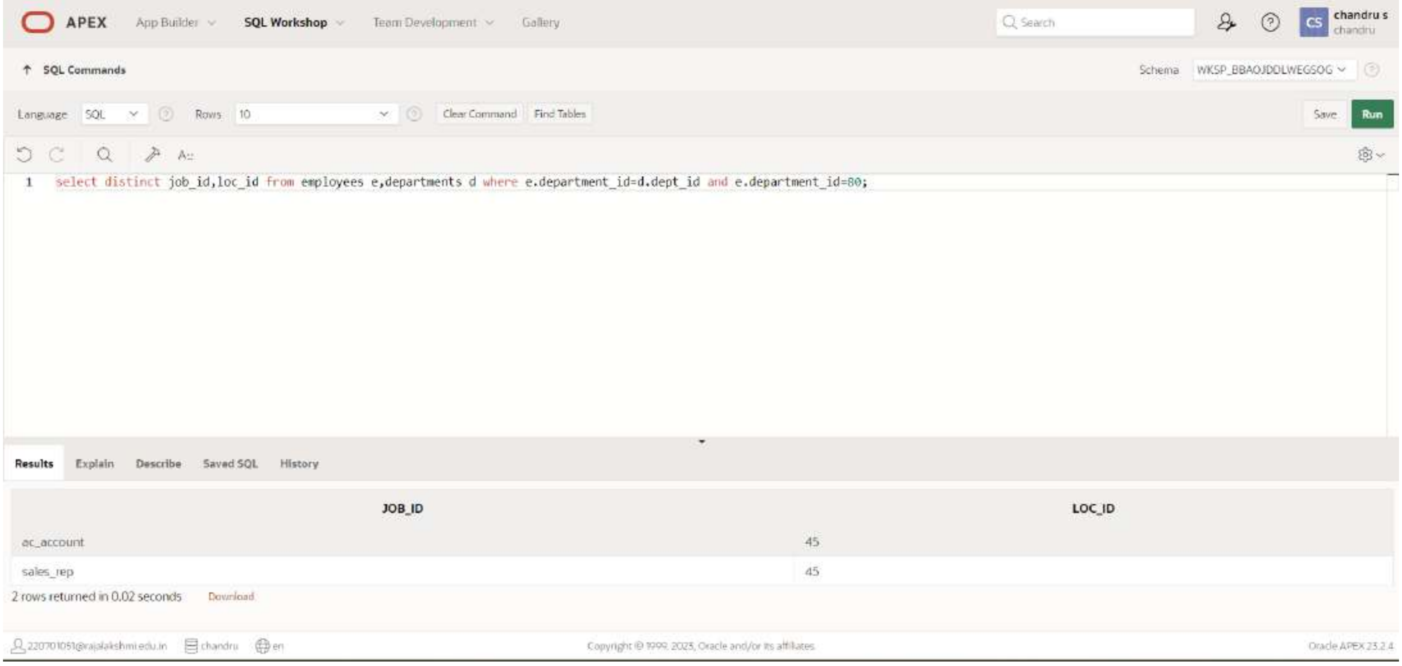
2 rows returned in 0.05 seconds

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

QUERY:

select distinct job_id,loc_id from employees e,departments d where e.department_id=d.department_id and e.department_id=80;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

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

JOB_ID	LOC_ID
ac_account	45
sales_rep	45

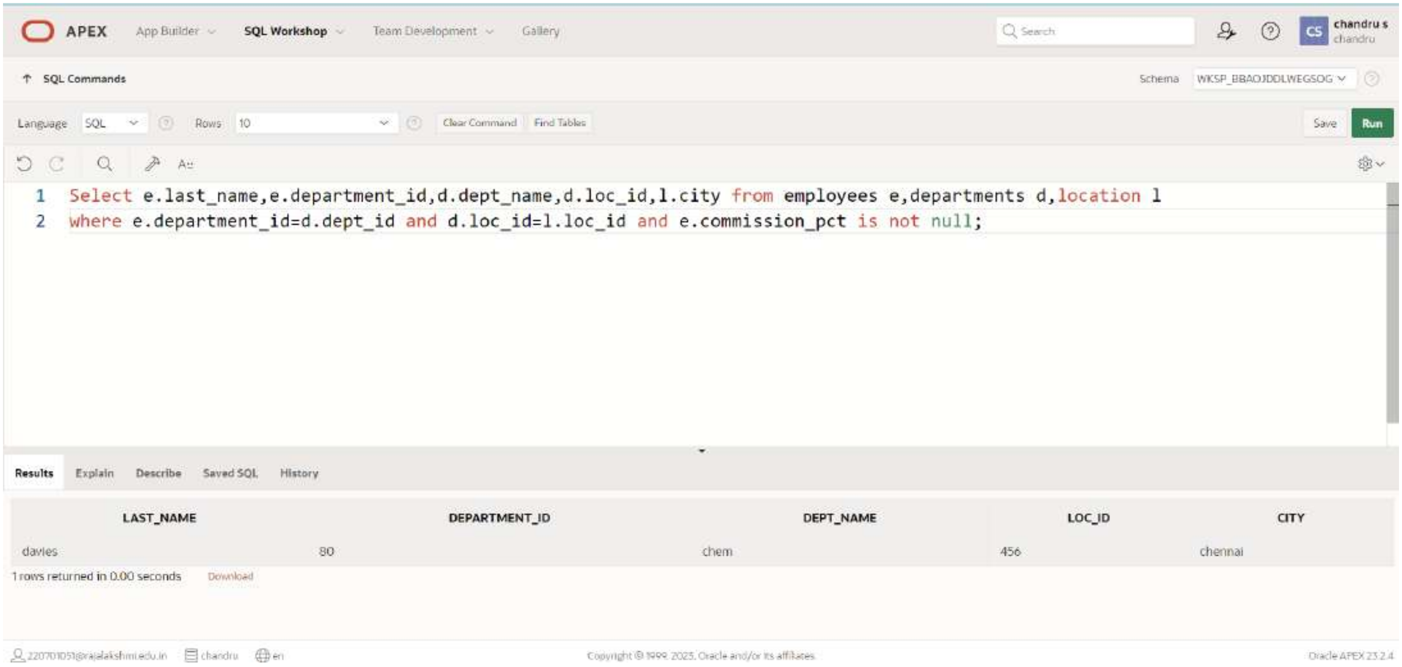
2 rows returned in 0.02 seconds

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,e.department_id,d.dept_name,d.loc_id,l.city from employees e,departments d,location l where e.department_id=d.department_id and d.loc_id=l.location_id and e.commission_pct is not null;

OUTPUT:



```
Select e.last_name,e.department_id,d.dept_name,d.loc_id,l.city from employees e,departments d,location l
where e.department_id=d.dept_id and d.loc_id=l.loc_id and e.commission_pct is not null;
```

LAST_NAME	DEPARTMENT_ID	DEPT_NAME	LOC_ID	CITY
davies	80	chem	456	chennai

1 rows returned in 0.00 seconds Download

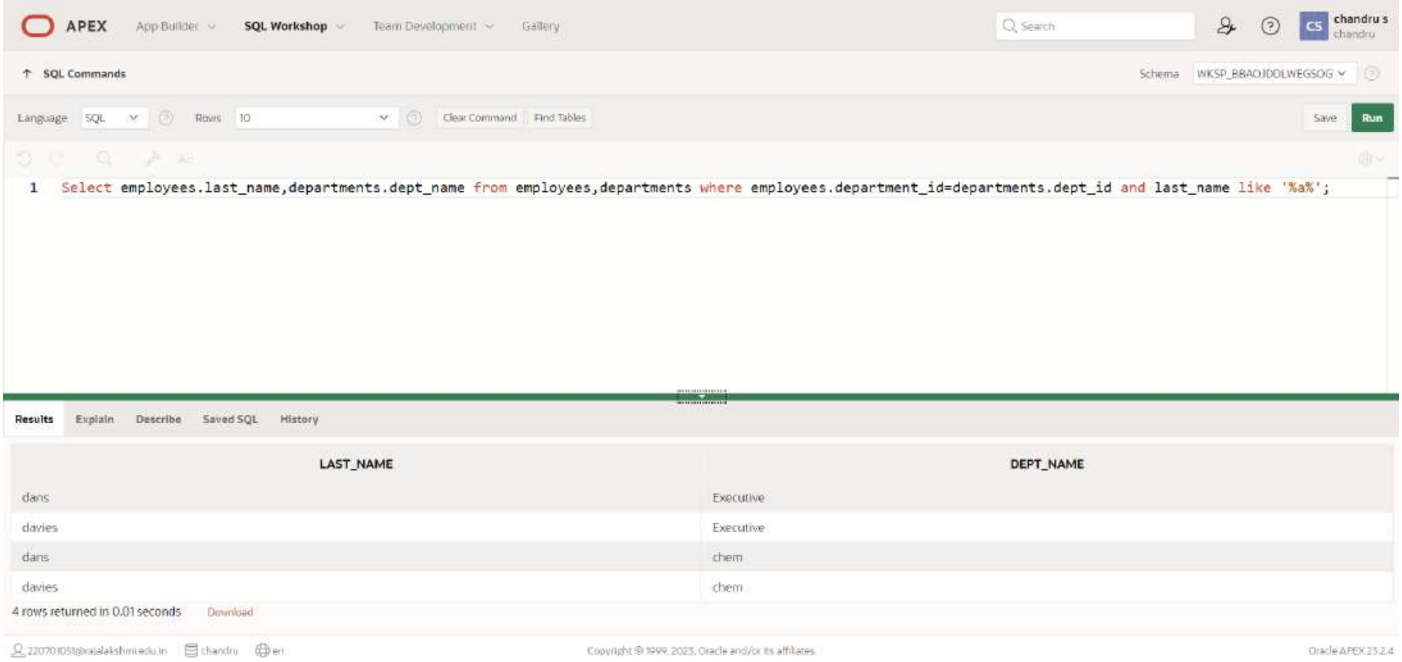
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

Select employees.last_name,departments.dept_name from employees,departmentswhere employees.department_id=departments.department_id and last_name like ‘%a%’;

OUTPUT:



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

LAST_NAME	DEPT_NAME
dans	Executive
davies	Executive
dans	chem
davies	chem

4 rows returned in 0.01 seconds Download

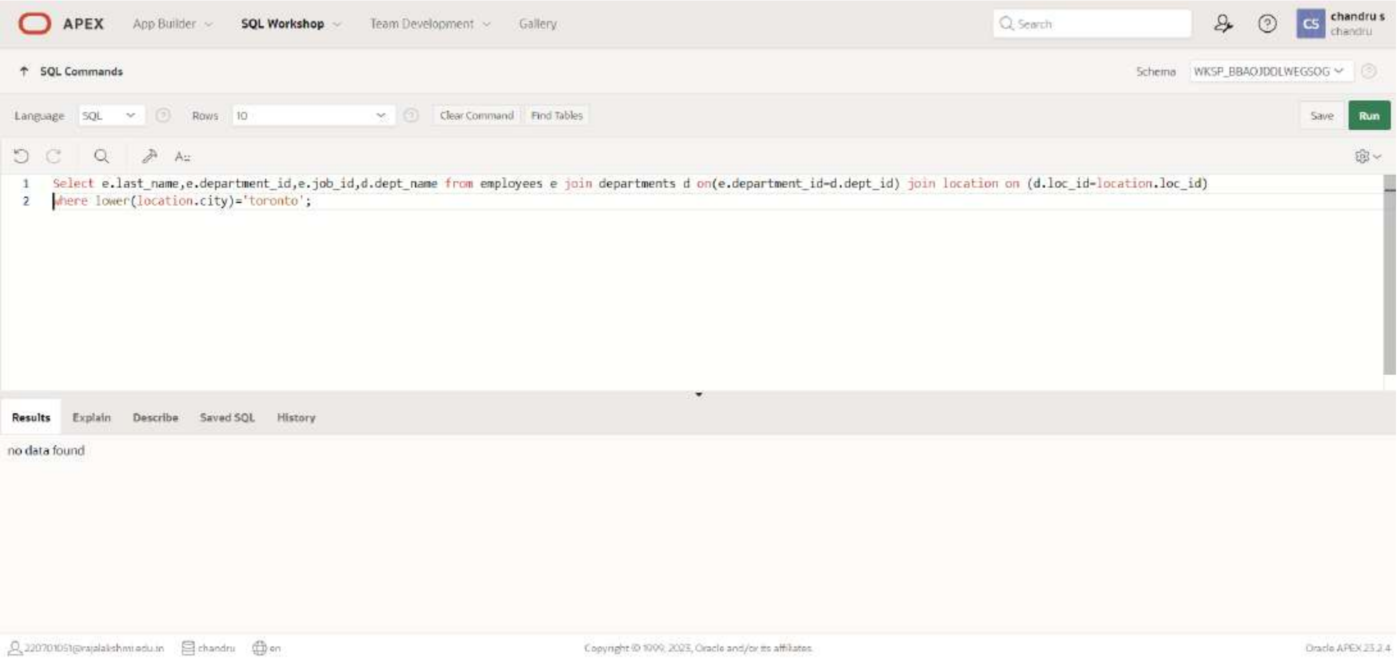
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

Select e.last_name,e.department_id,e.job_id,d.dept_name from employees e join departments d on(e.department_id=d.department_id) join location on (d.location_id=location.location_id) where lower(location.city)=’toronto’;

OUTPUT:



```
Select e.last_name,e.department_id,e.job_id,d.dept_name from employees e join departments d on(e.department_id=d.dept_id) join location on (d.loc_id=location.loc_id)
where lower(location.city)='toronto';
```

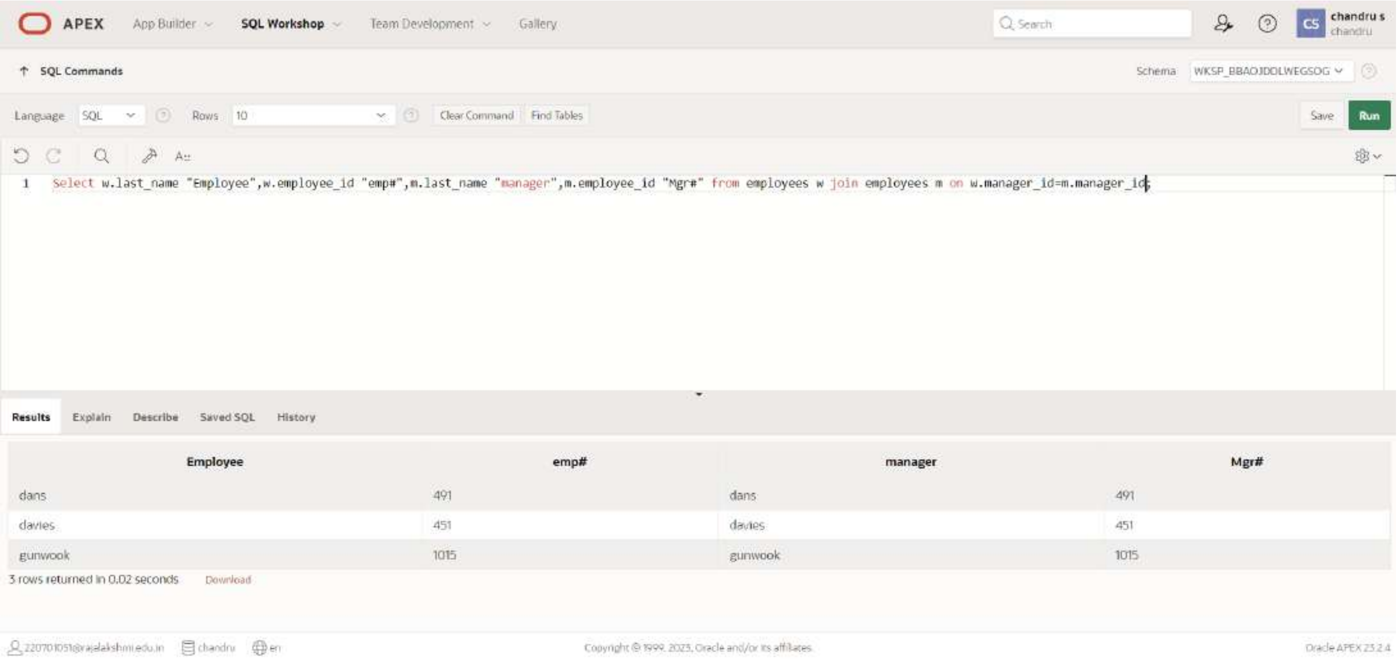
no data found

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.employee_id “emp#”,m.last_name ‘manager”,m.employee_id “Mgr#” from employees m on (w.manager_id=m.employee_id);

OUTPUT:



```
Select w.last_name "Employee",w.employee_id "emp#",m.last_name "manager",m.employee_id "Mgr#" from employees w join employees m on w.manager_id=m.manager_id;
```

Employee	emp#	manager	Mgr#
dans	491	dans	491
davies	451	davies	451
gunwook	1015	gunwook	1015

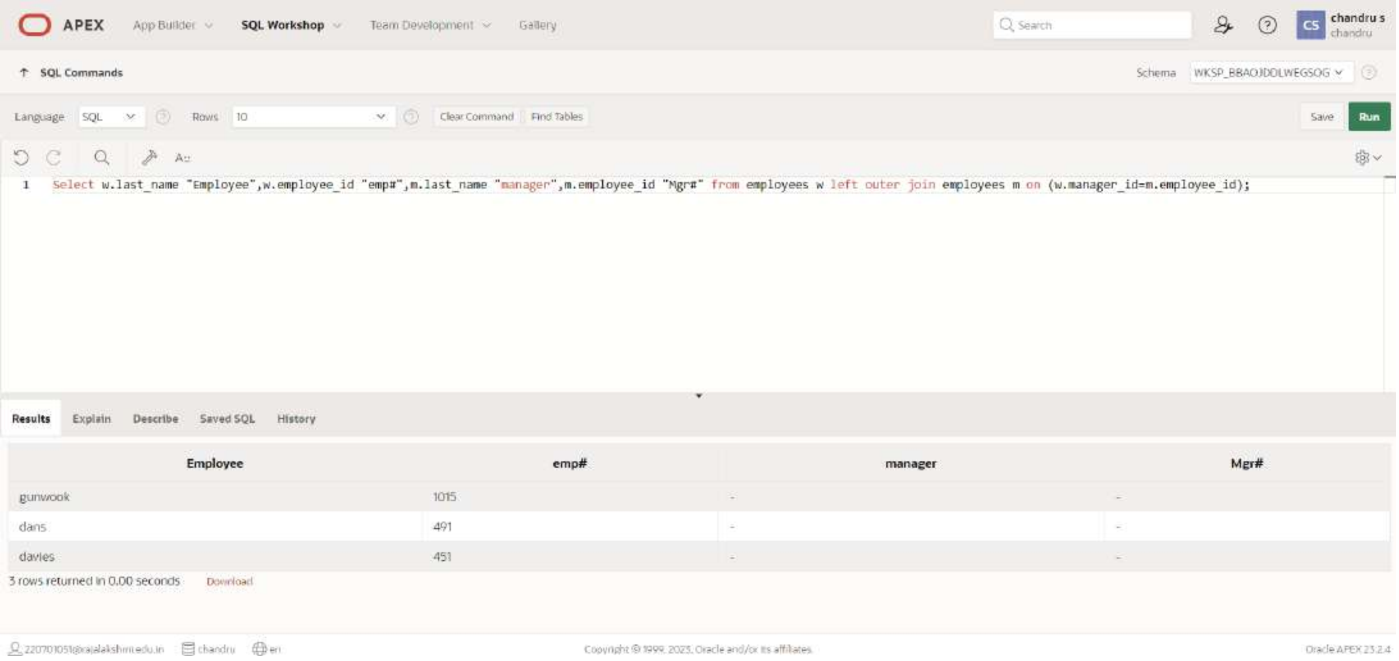
3 rows returned in 0.02 seconds

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.employee_id “emp#”,m.last_name ‘manager”,m.employee_id “Mgr#” from employees w left outer join employees m on (w.manager_id=m.employee_id);

OUTPUT:



```
Select w.last_name "Employee",w.employee_id "emp#",m.last_name "manager",m.employee_id "Mgr#" from employees w left outer join employees m on (w.manager_id=m.employee_id);
```

Employee	emp#	manager	Mgr#
gunwook	1015	-	-
dans	491	-	-
davies	451	-	-

3 rows returned in 0.00 seconds Download

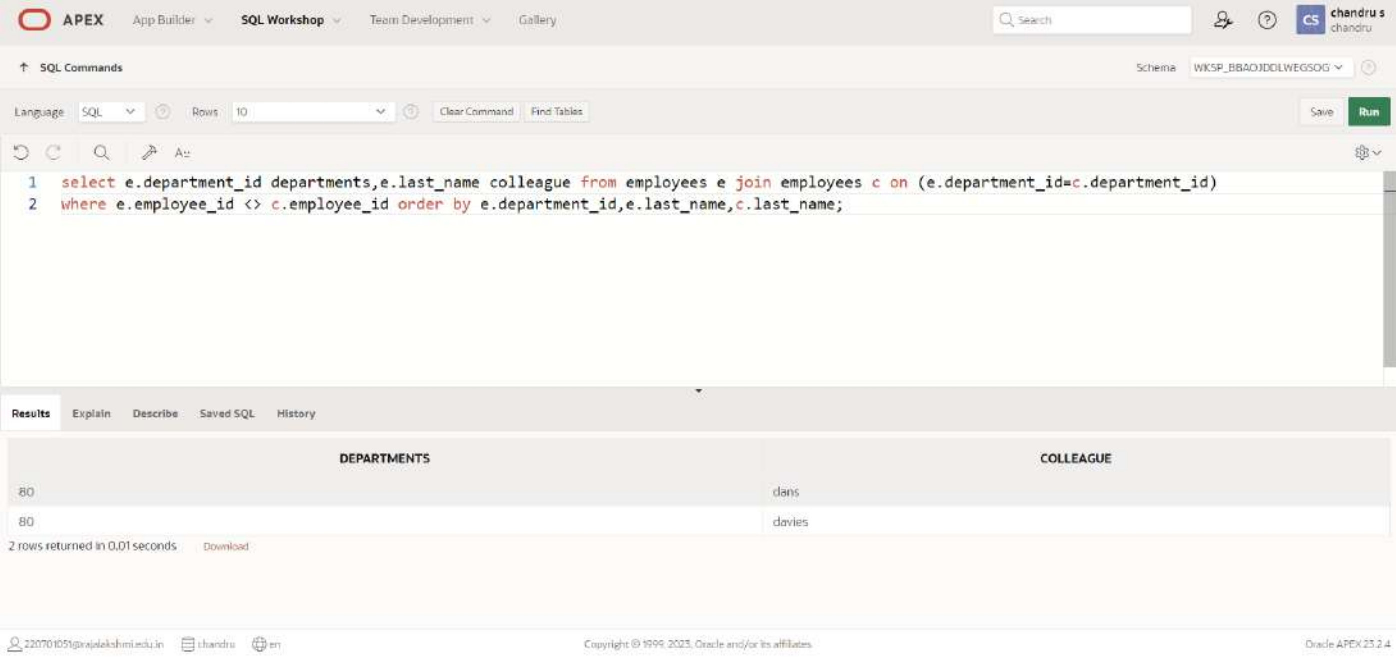
2207010151@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

select e.department_id departments,e.last_name colleague from employees e join employees c on (e.department_id=c.department_id) where e.employee_id <> c.employee_id order by e.department_id,e.last_name,c.last_name;

OUTPUT:



```
select e.department_id departments,e.last_name colleague from employees e join employees c on (e.department_id=c.department_id)
where e.employee_id <> c.employee_id order by e.department_id,e.last_name,c.last_name;
```

DEPARTMENTS	COLLEAGUE
80	dans
80	davies

2 rows returned in 0.01 seconds Download

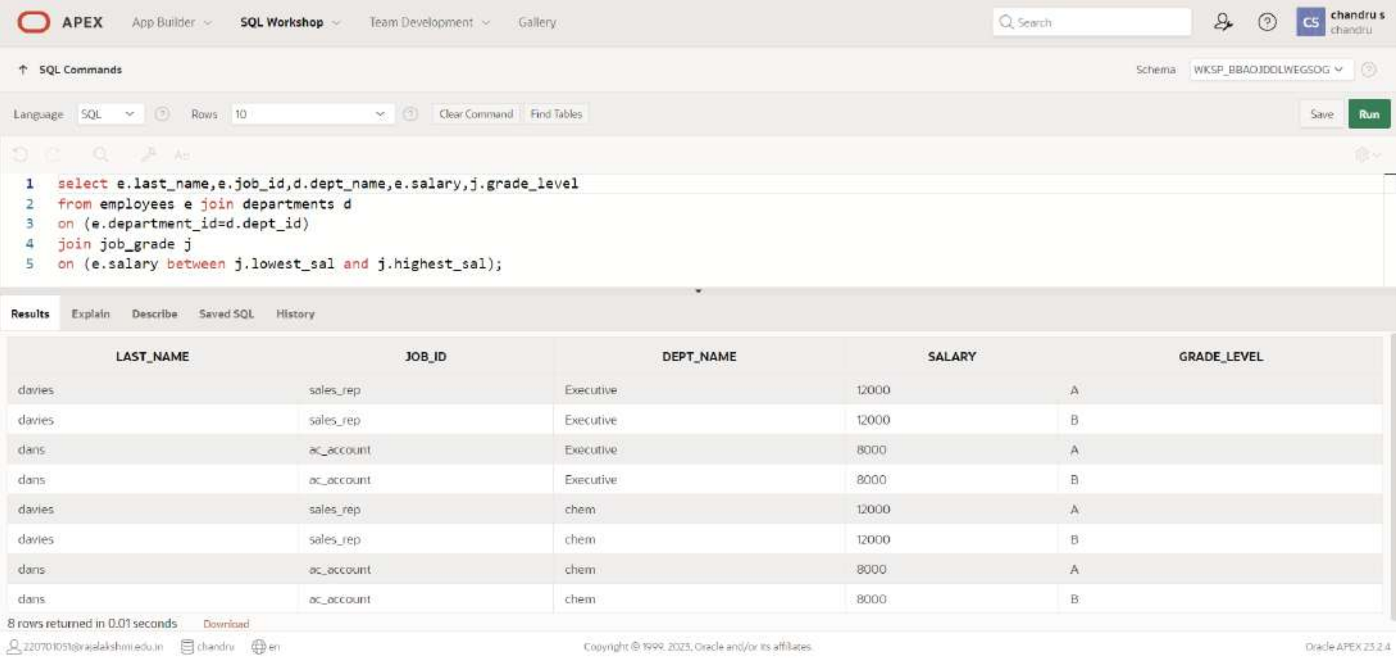
2207010151@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

SELECT e.last_name, e. job_id, d.dept_name,e.salary, j.grade_level
FROM employees e JOIN departments d
ON (e.department_id = d. dept_id)
JOIN job_grade j
ON (e.salary BETWEEN j. lowest_ sal AND j.highest_sal);

OUTPUT:



```
select e.last_name,e.job_id,d.dept_name,e.salary,j.grade_level
from employees e join departments d
on (e.department_id=d.dept_id)
join job_grade j
on (e.salary between j.lowest_sal and j.highest_sal);
```

LAST_NAME	JOB_ID	DEPT_NAME	SALARY	GRADE_LEVEL
davies	sales_rep	Executive	12000	A
davies	sales_rep	Executive	12000	B
dans	ac_account	Executive	8000	A
dans	ac_account	Executive	8000	B
davies	sales_rep	chem	12000	A
davies	sales_rep	chem	12000	B
dans	ac_account	chem	8000	A
dans	ac_account	chem	8000	B

8 rows returned in 0.01 seconds Download

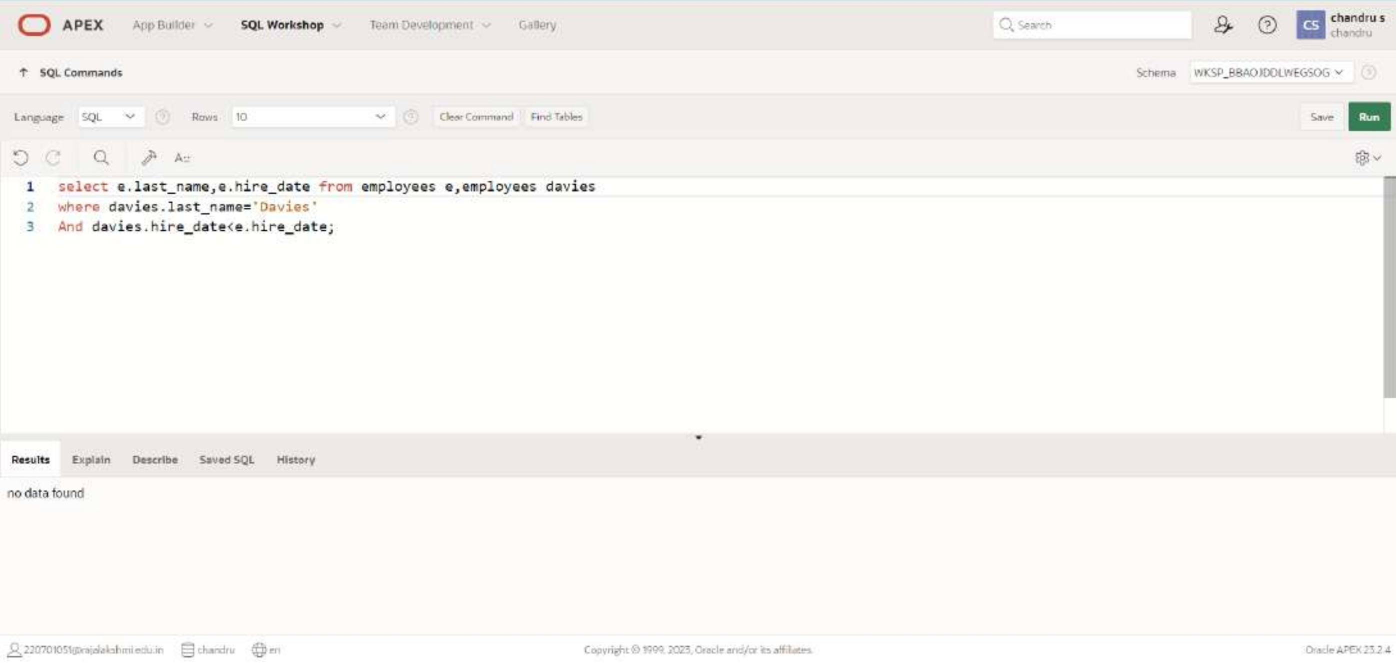
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

SELECT e. last_name, e. hire_date FROM employees e, employees davies
WHERE davies.last name = 'Davies'
AND davies.hire_date ‹ e.hire_date;

OUTPUT:



```
select e.last_name,e.hire_date from employees e,employees davies
where davies.last_name='Davies'
And davies.hire_date<e.hire_date;
```

no data found

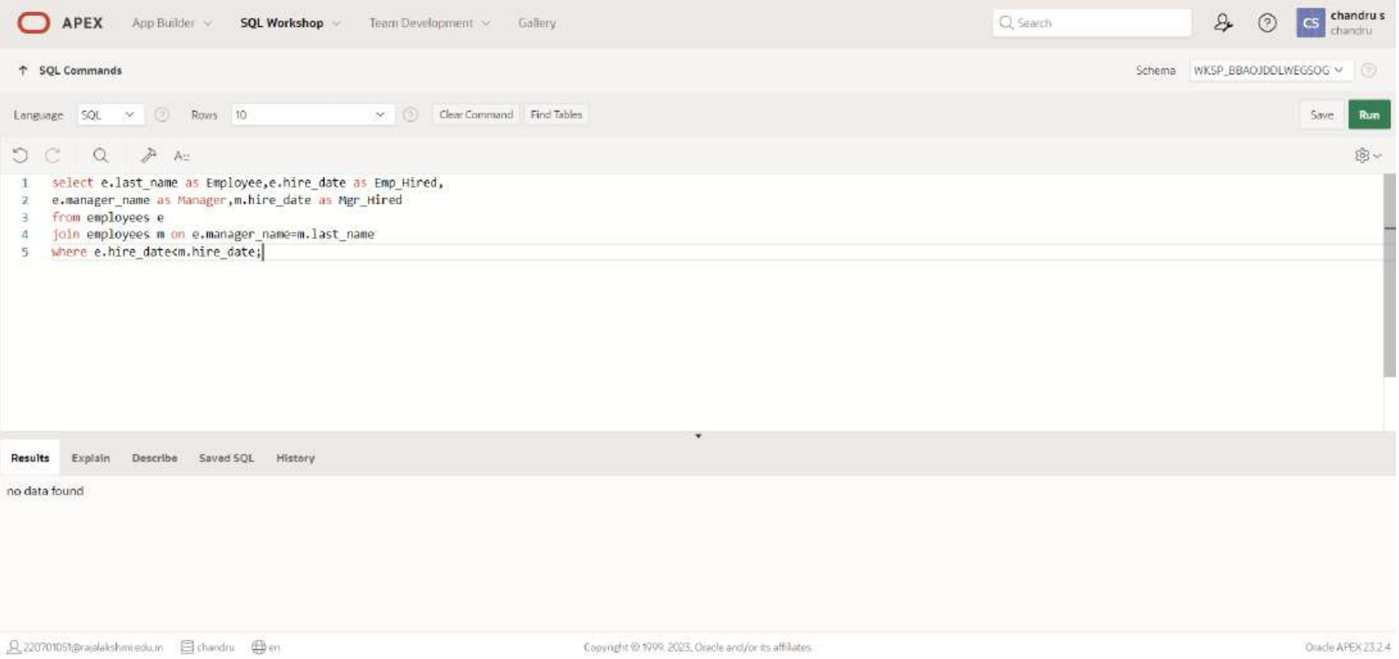
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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. manager_name AS Manager, m.hire_date AS Mgr_Hired
FROM employees e
JOIN employees|m ON e-manager _name = m. last_name
WHERE e.hire_date ‹ m.hire_date;
```

OUTPUT:



```
select e.last_name as Employee,e.hire_date as Emp_Hired,
e.manager_name as Manager,m.hire_date as Mgr_Hired
from employees e
join employees m on e.manager_name=m.last_name
where e.hire_date<m.hire_date;
```

```
no data found
```

220701051@rajalakshmi.edu.in
chandru
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:

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

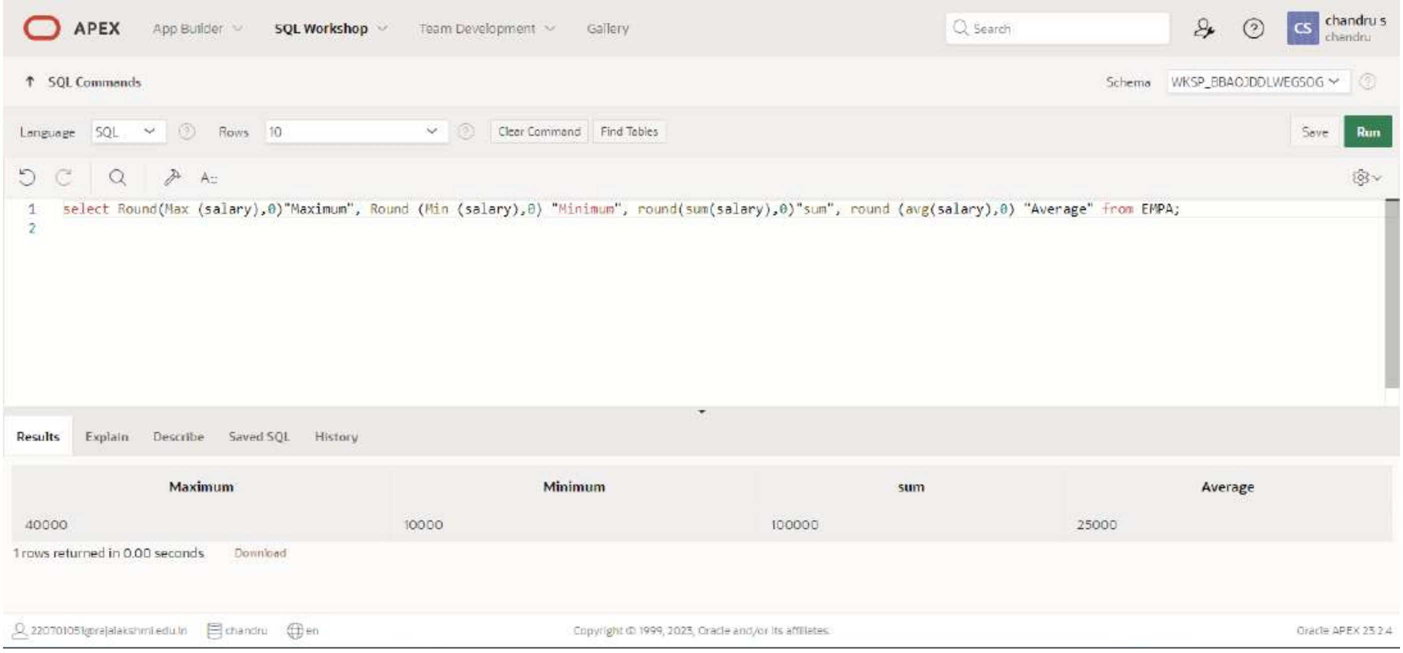
4.Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number

QUERY:

select Round(Max (salary),0)"Maximum", Round (Min (salary),0) "Minimum",

round(sum(salary),0)"sum", round (avg(salary),0) "Average" from EMPLOYEES;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

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

Maximum	Minimum	sum	Average
40000	10000	100000	25000

1 rows returned in 0.00 seconds

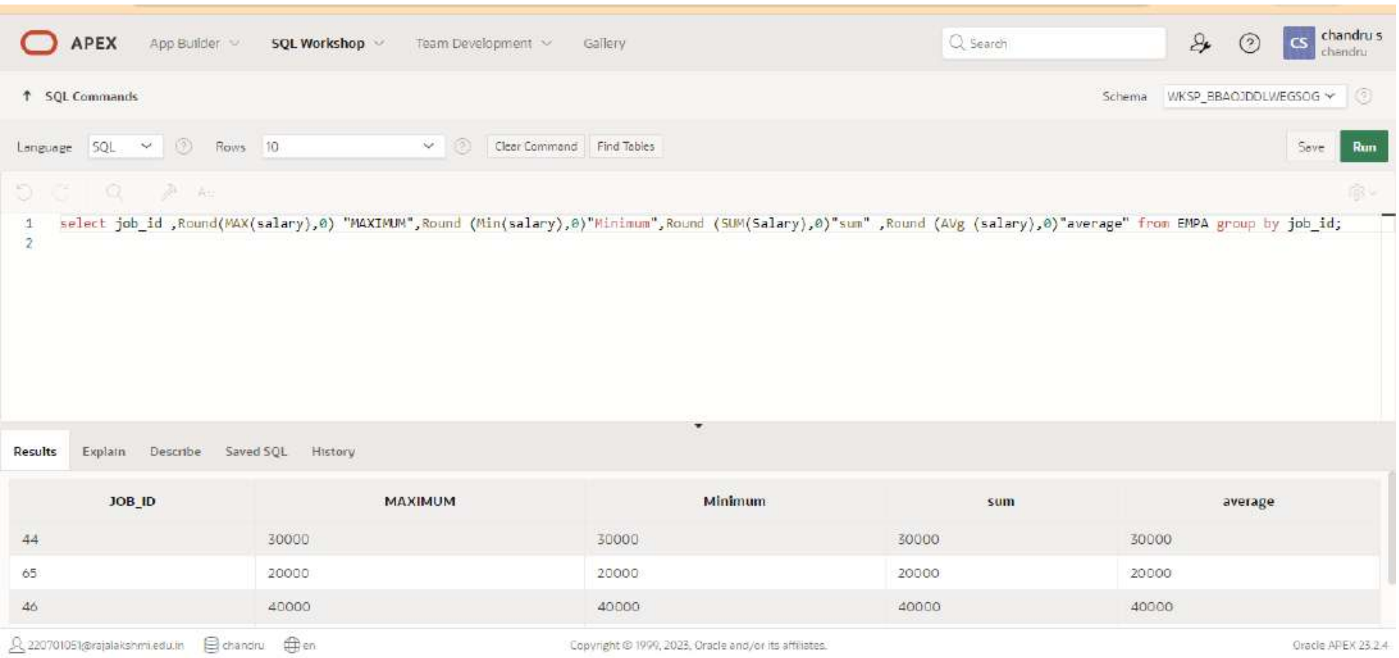
220701051@rajalakshmi.edu.in chandru en

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

QUERY:

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

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

JOB_ID	MAXIMUM	Minimum	sum	average
44	30000	30000	30000	30000
65	20000	20000	20000	20000
46	40000	40000	40000	40000

220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

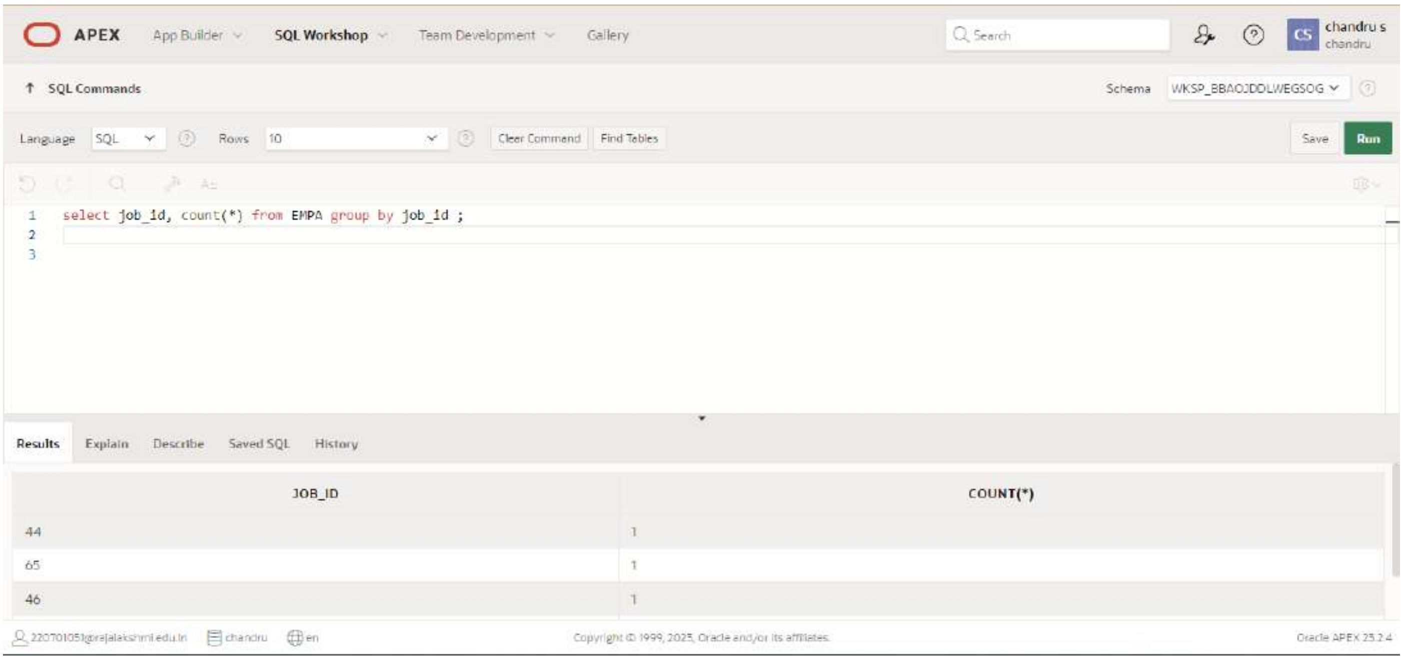
6.Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

QUERY:

select job_id, count(*) from EMPLOYEES group by job_id ;

select job_id, count(*) from EMPLOYEES where job_id='47' group by job_id ;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

```
select job_id, count(*) from EMPA group by job_id ;
```

JOB_ID	COUNT(*)
44	1
65	1
46	1

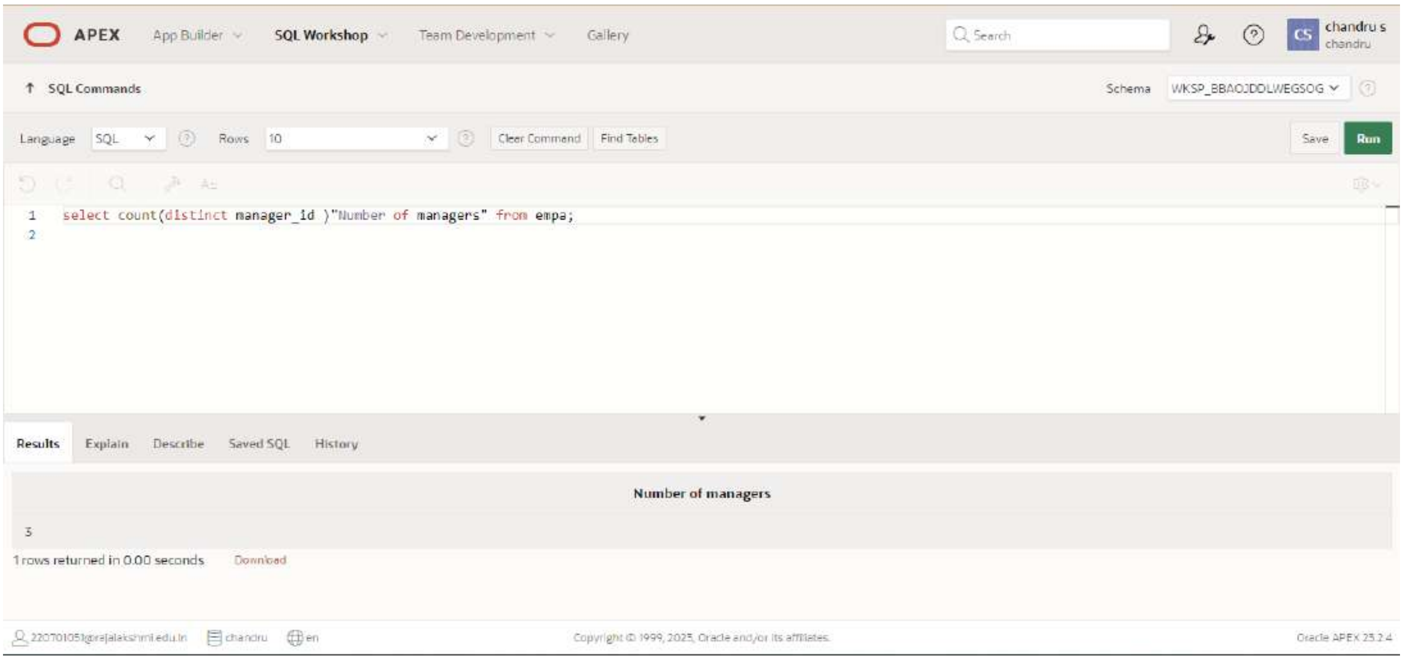
220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

select count(distinct manager_id)"Number of managers" from employees;

OUTPUT:



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

Number of managers
5

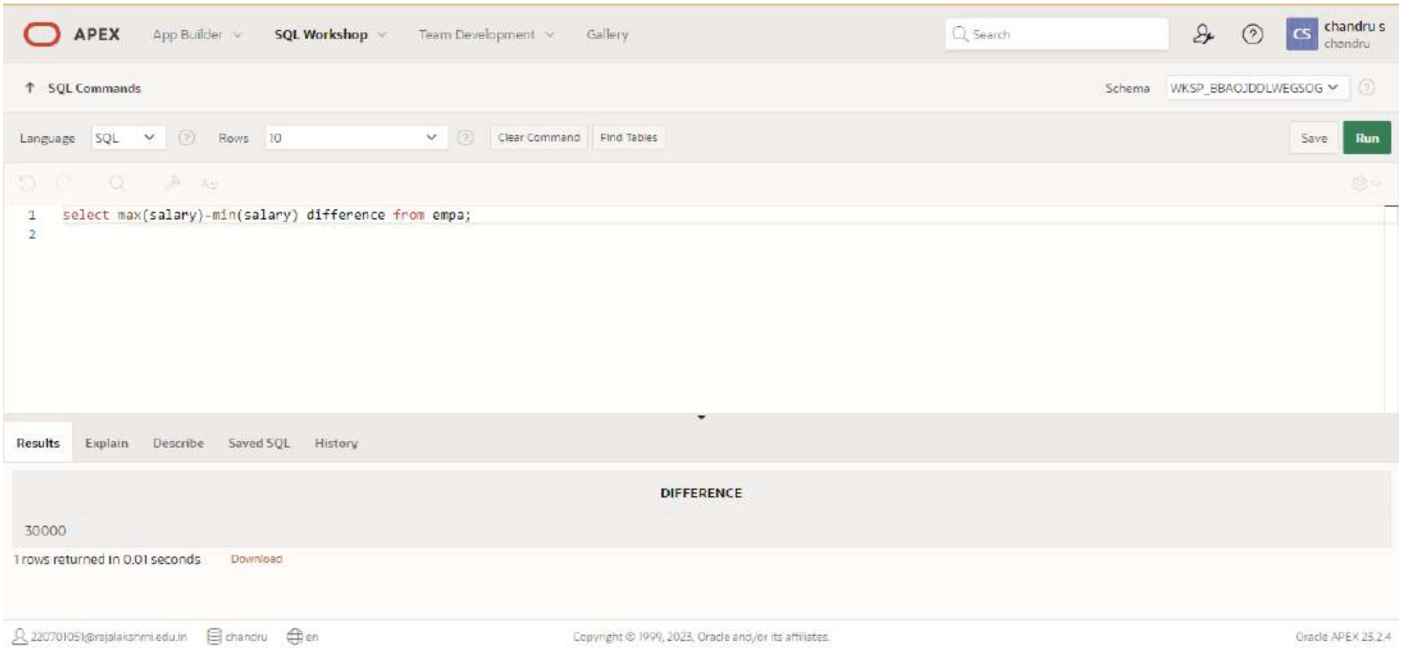
1 rows returned in 0.00 seconds

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

QUERY:

select max(salary)-min(salary) difference from employees;

OUTPUT:



```
select max(salary)-min(salary) difference from empa;
```

DIFFERENCE
30000

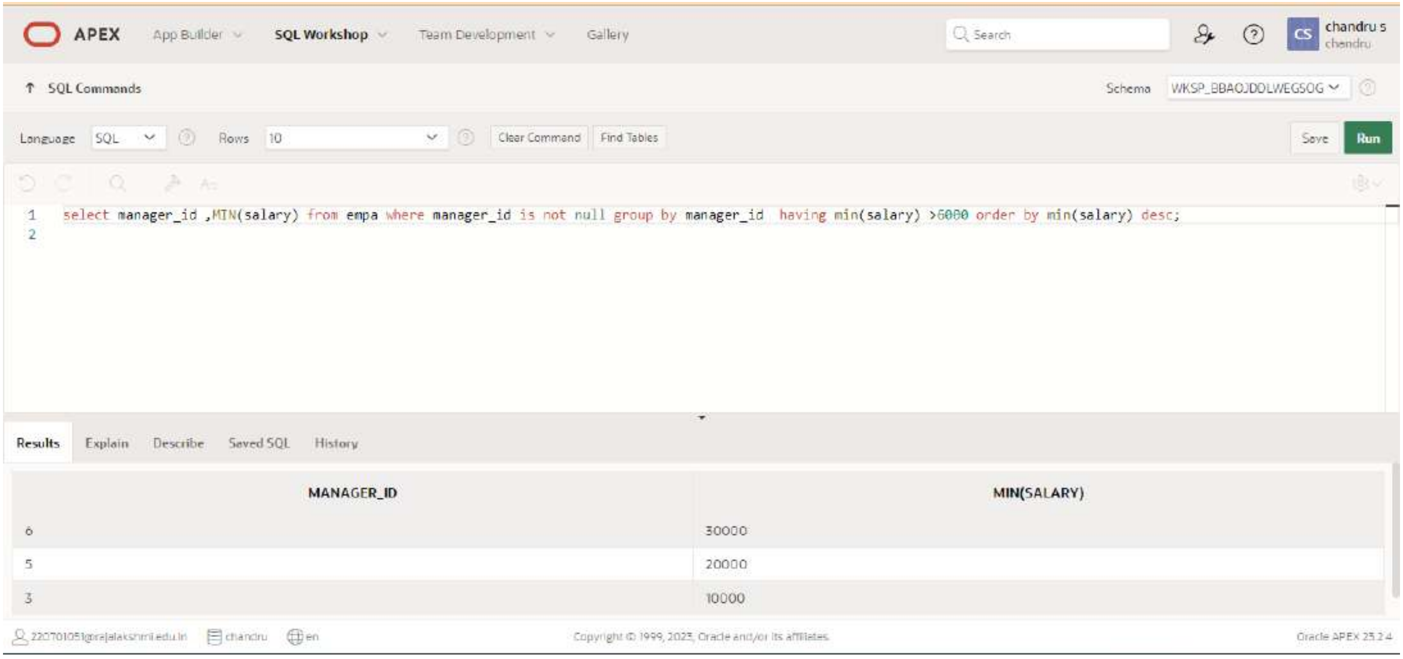
1 rows returned in 0.01 seconds

9.Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.

QUERY:

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

OUTPUT:



```
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;
```

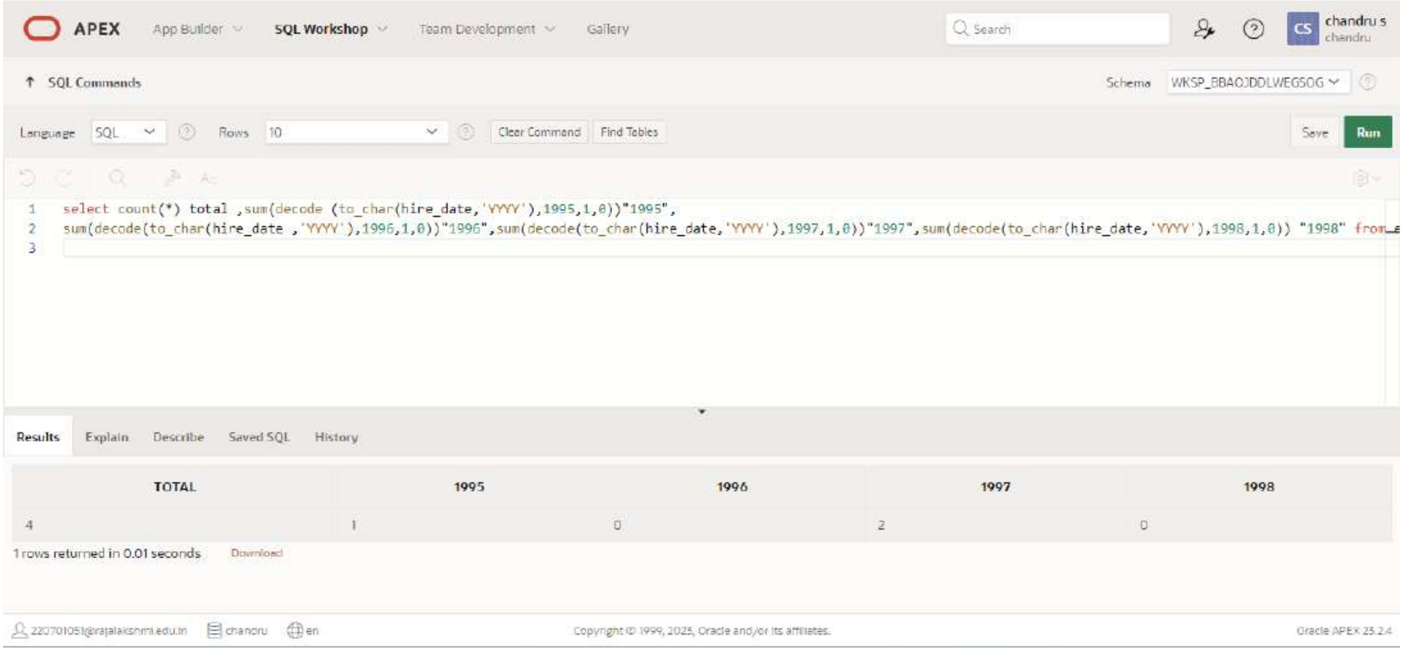
MANAGER_ID	MIN(SALARY)
6	30000
5	20000
3	10000

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:

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

OUTPUT:



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

TOTAL	1995	1996	1997	1998
4	1	0	2	0

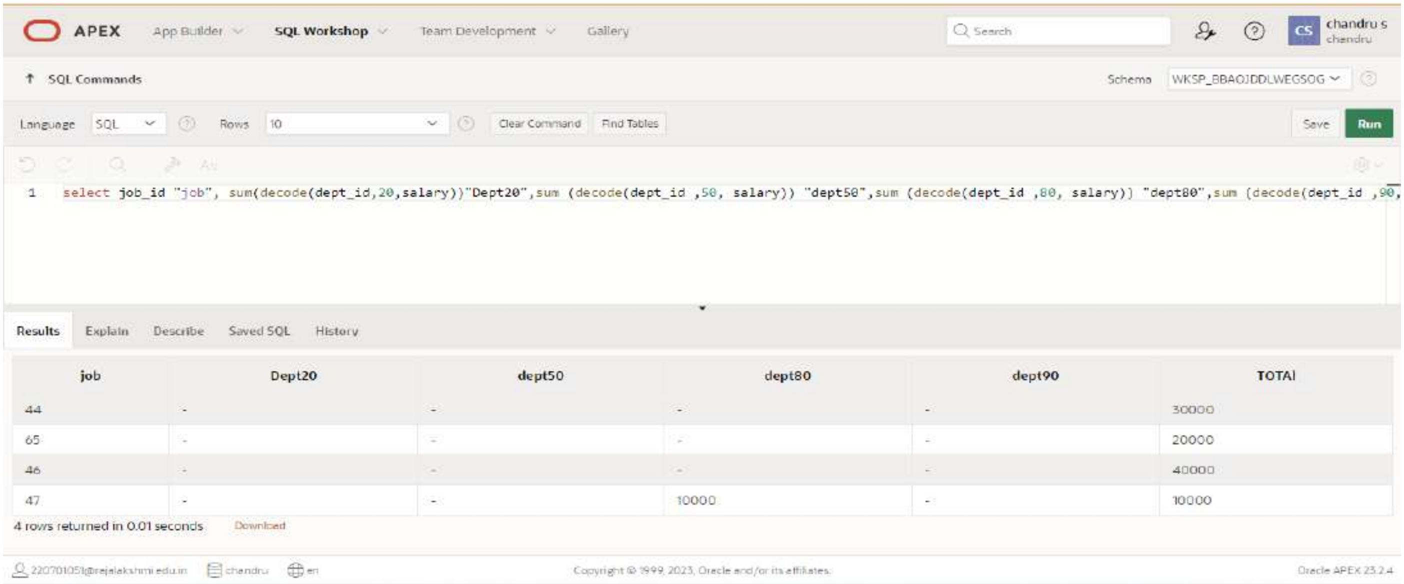
1 rows returned in 0.01 seconds

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 department_id,job_id,sum(salary) from employees where department_id in (20,50,80,90) group by rollup(department_id,job_id);

OUTPUT:



```
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,
```

job	Dept20	dept50	dept80	dept90	TOTAl
44	-	-	-	-	30000
65	-	-	-	-	20000
46	-	-	-	-	40000
47	-	-	10000	-	10000

4 rows returned in 0.01 seconds Download

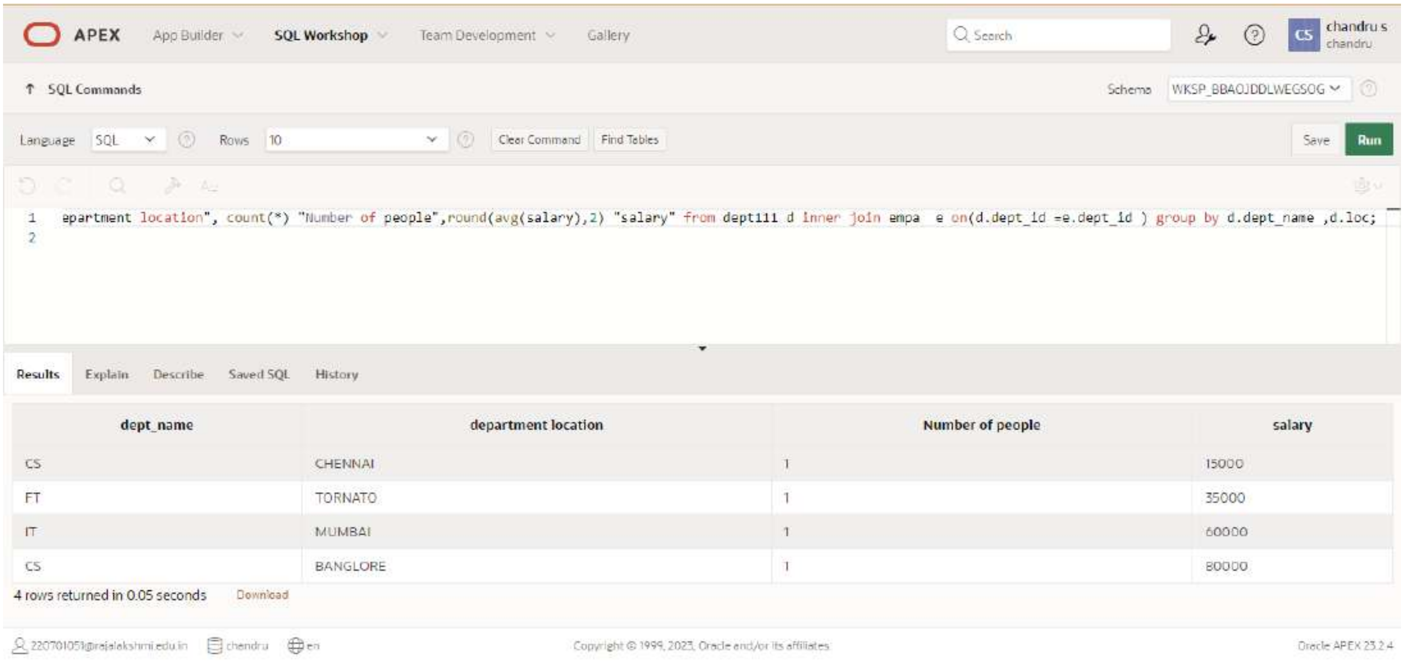
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

12. Write a query to display each department's name, location, number of employees, and the average salary for all the employees in that department. Label the column name-Location, Number of people, and salary respectively. Round the average salary to two decimal places.

QUERY:

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

OUTPUT:



```
epartment location", count(*) "Number of people",round(avg(salary),2) "salary" from dept111 d inner join empa  e on(d.dept_id =e.dept_id ) group by d.dept_name ,d.loc;
```

dept_name	department location	Number of people	salary
CS	CHENNAI	1	15000
FT	TORNATO	1	35000
IT	MUMBAI	1	60000
CS	BANGLORE	1	80000

4 rows returned in 0.05 seconds Download

220701051@rajalakshmi.edu.in
chandru
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:

SUB QUERIES

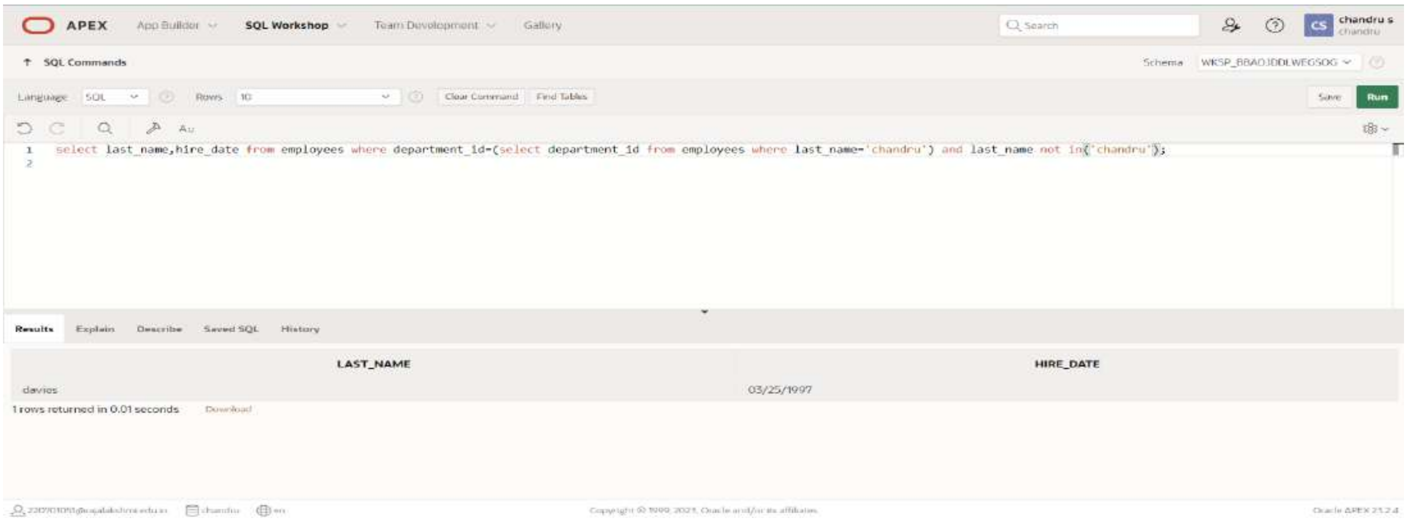
EX_NO:9 **DATE:**

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

QUERY:

select last_name,hire_date from employees where department_id=(select department_id from employees where last_name='chandru') and last_name not in(‘chandru’);

OUTPUT:



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

LAST_NAME	HIRE_DATE
davies	03/25/1997

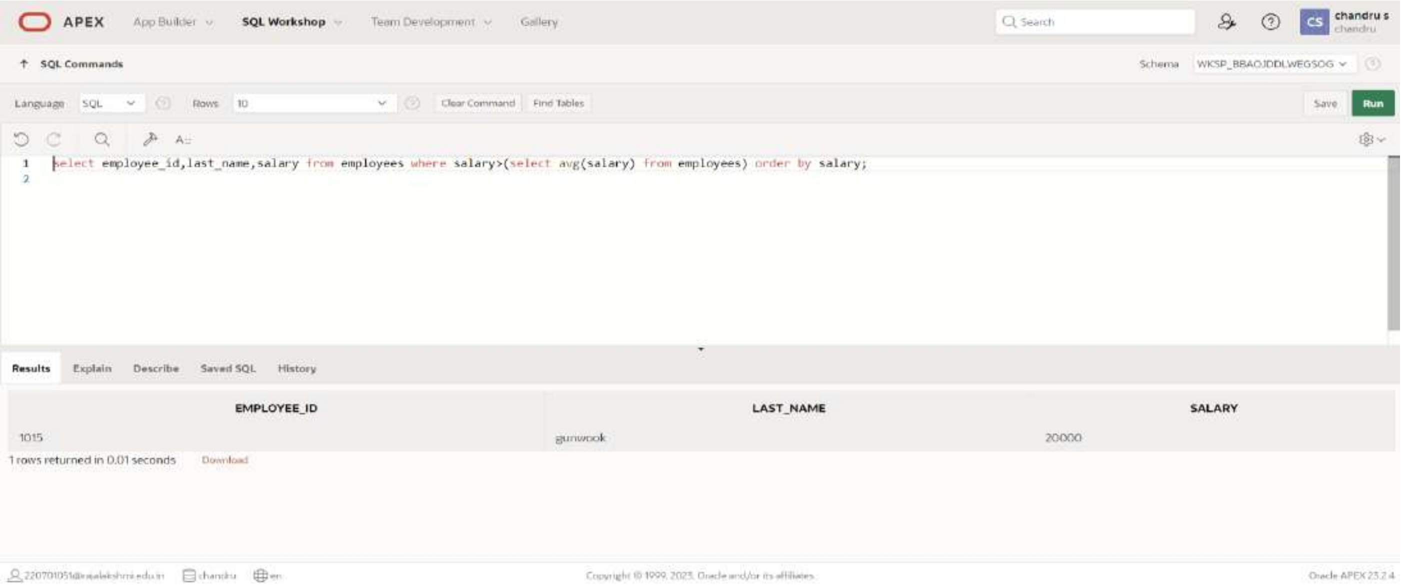
1 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



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

EMPLOYEE_ID	LAST_NAME	SALARY
1015	gunwook	20000

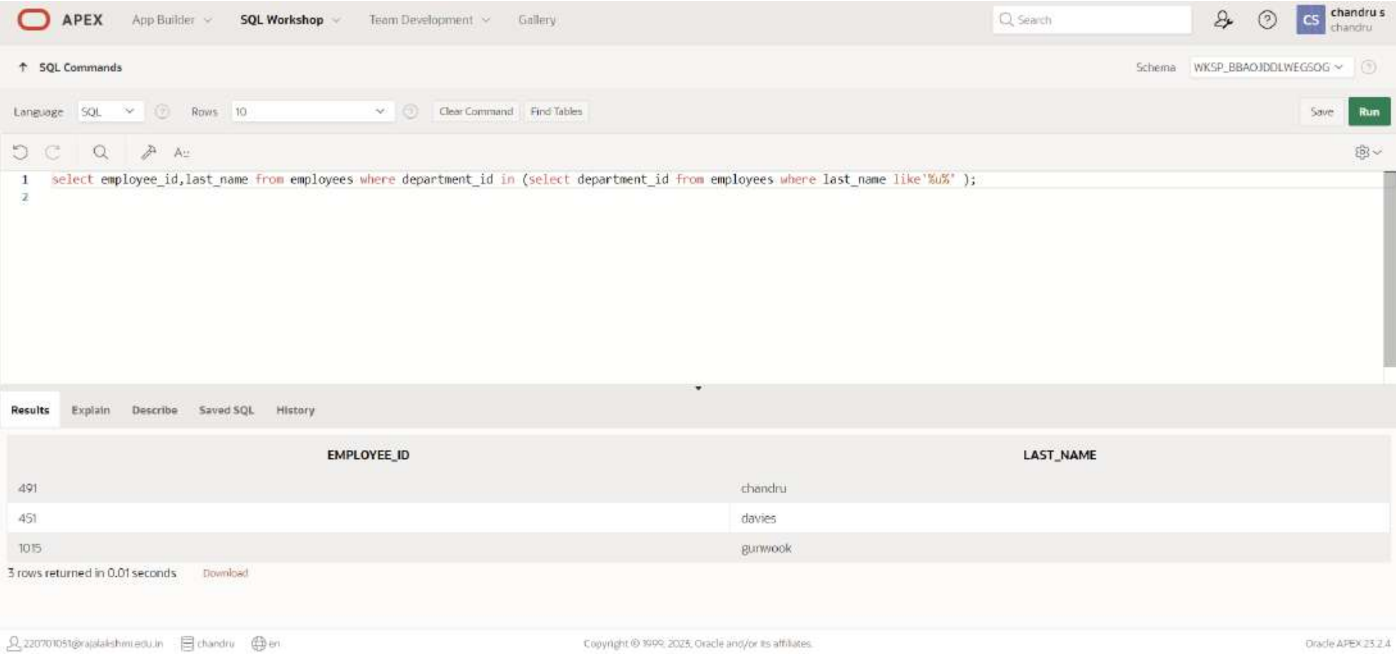
1 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



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

EMPLOYEE_ID	LAST_NAME
491	chandru
451	davies
1015	gurwook

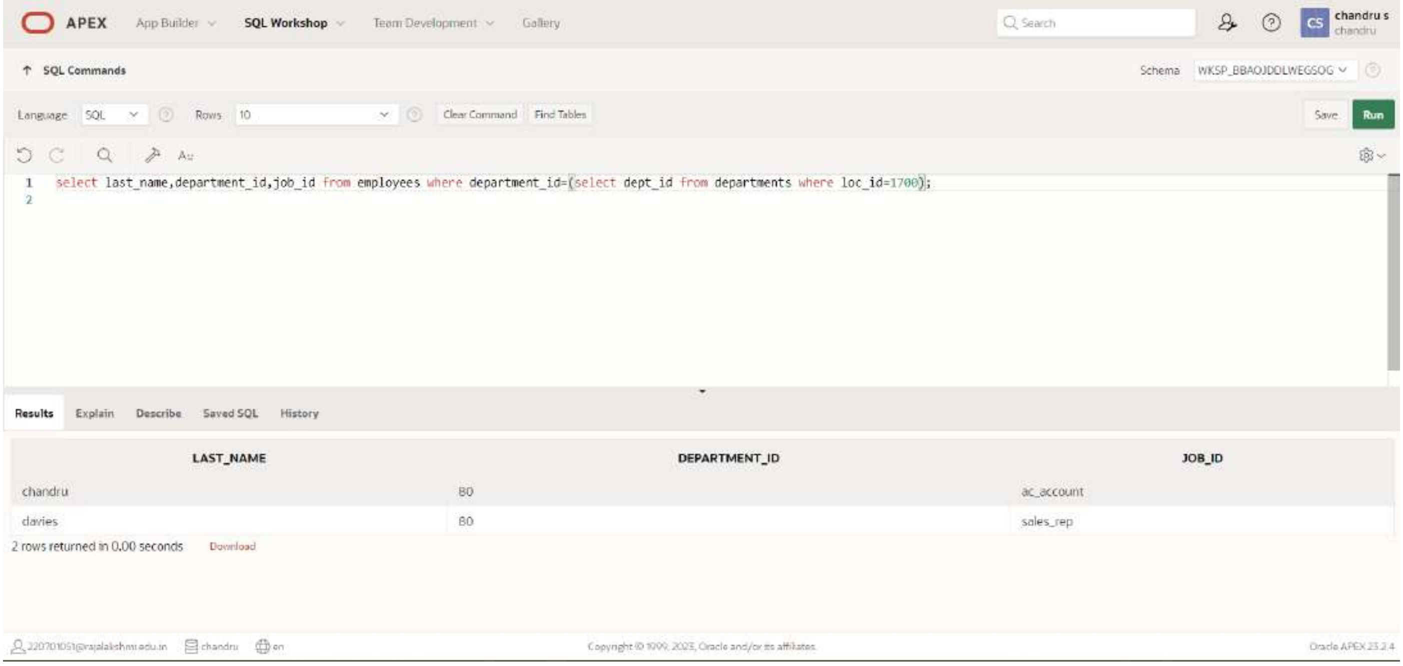
3 rows returned in 0.01 seconds

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 department_id from departments where location_id=1700);

OUTPUT:



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

LAST_NAME	DEPARTMENT_ID	JOB_ID
chandru	80	ac_account
davies	80	sales_rep

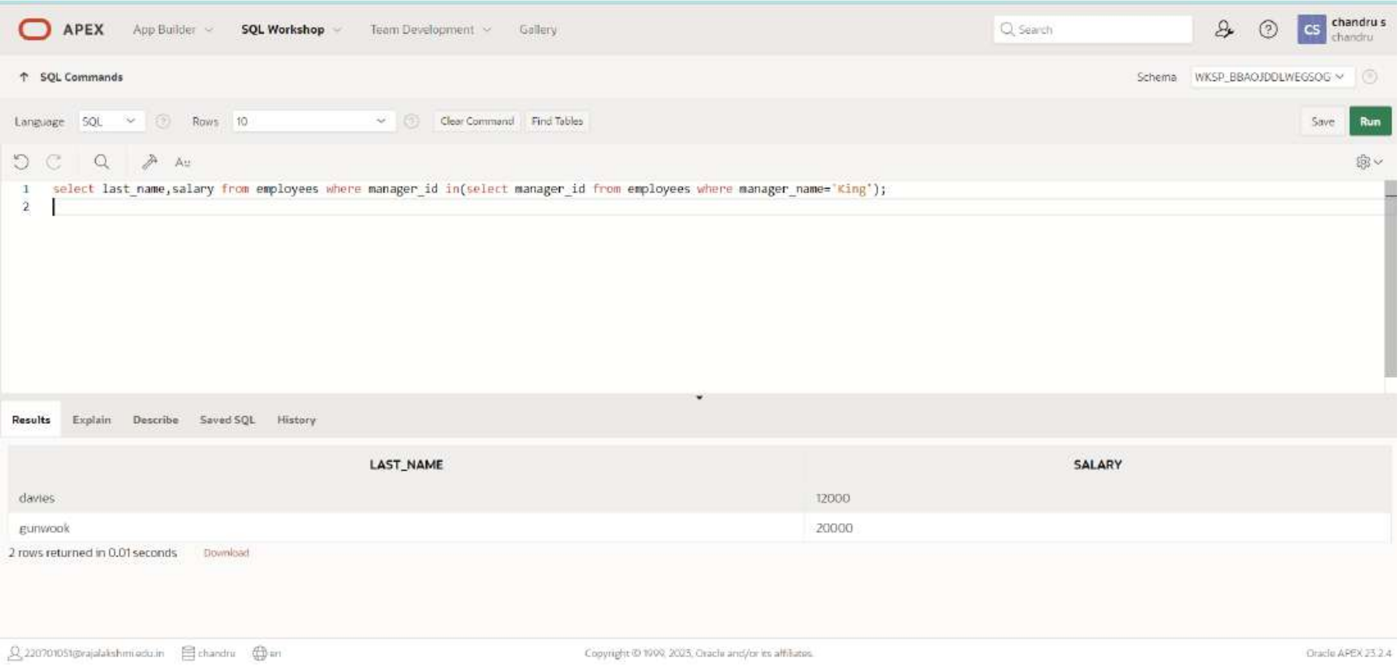
2 rows returned in 0.00 seconds

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

QUERY:

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

OUTPUT:



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

LAST_NAME	SALARY
davies	12000
gunwook	20000

2 rows returned in 0.01 seconds Download

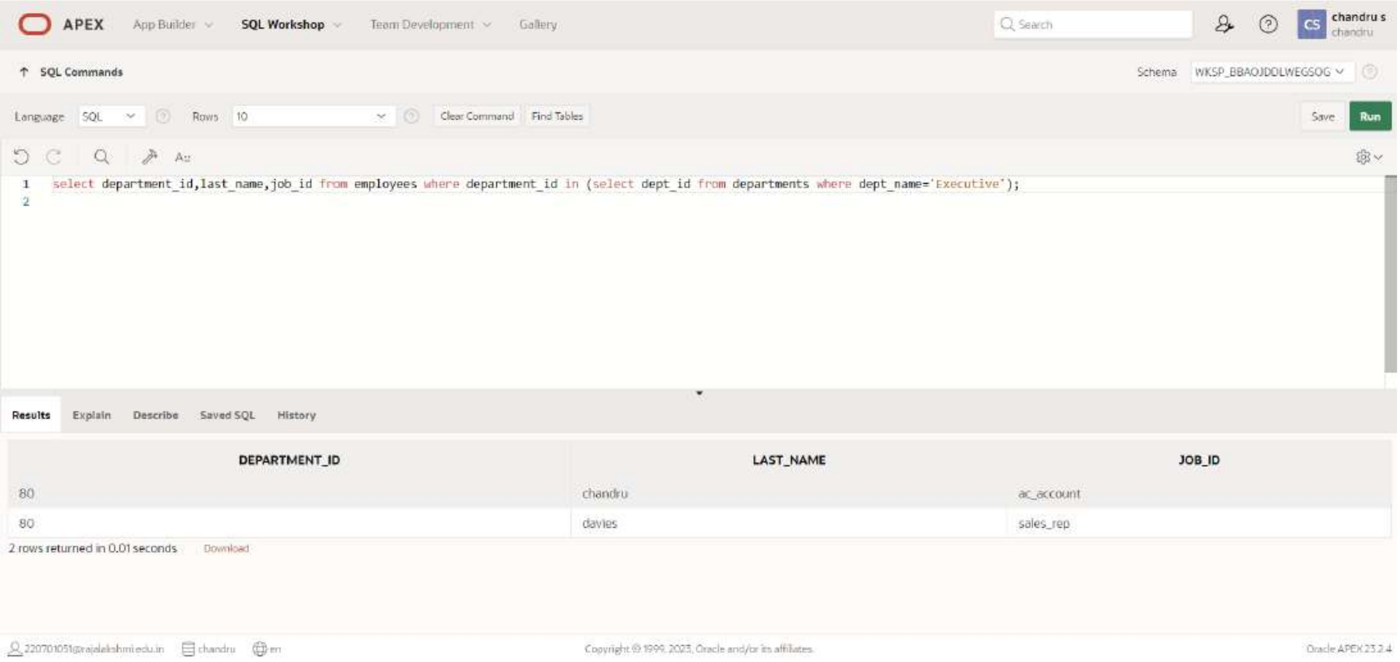
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2025, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



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

DEPARTMENT_ID	LAST_NAME	JOB_ID
80	chandru	ac_account
80	davies	sales_rep

2 rows returned in 0.01 seconds Download

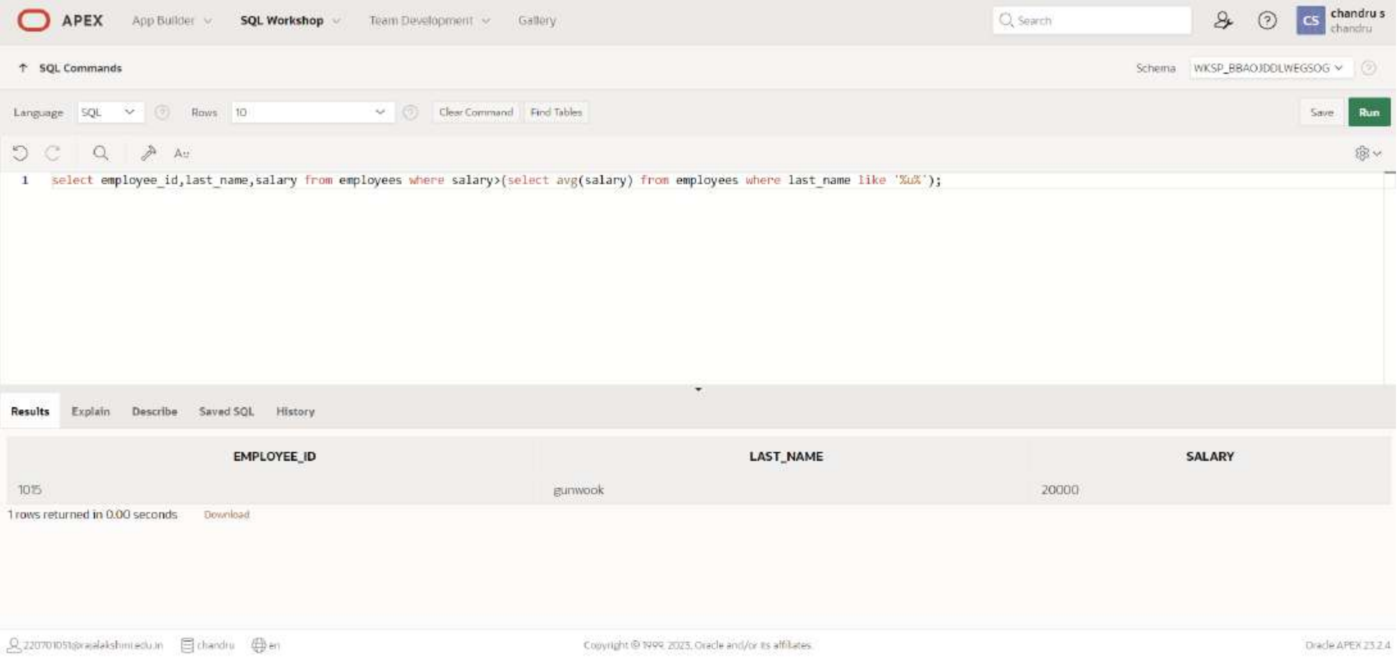
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2025, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



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

EMPLOYEE_ID	LAST_NAME	SALARY
1015	gunwook	20000

1 rows returned in 0.00 seconds

220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

RESULT:

USING THE SET OPERATORS

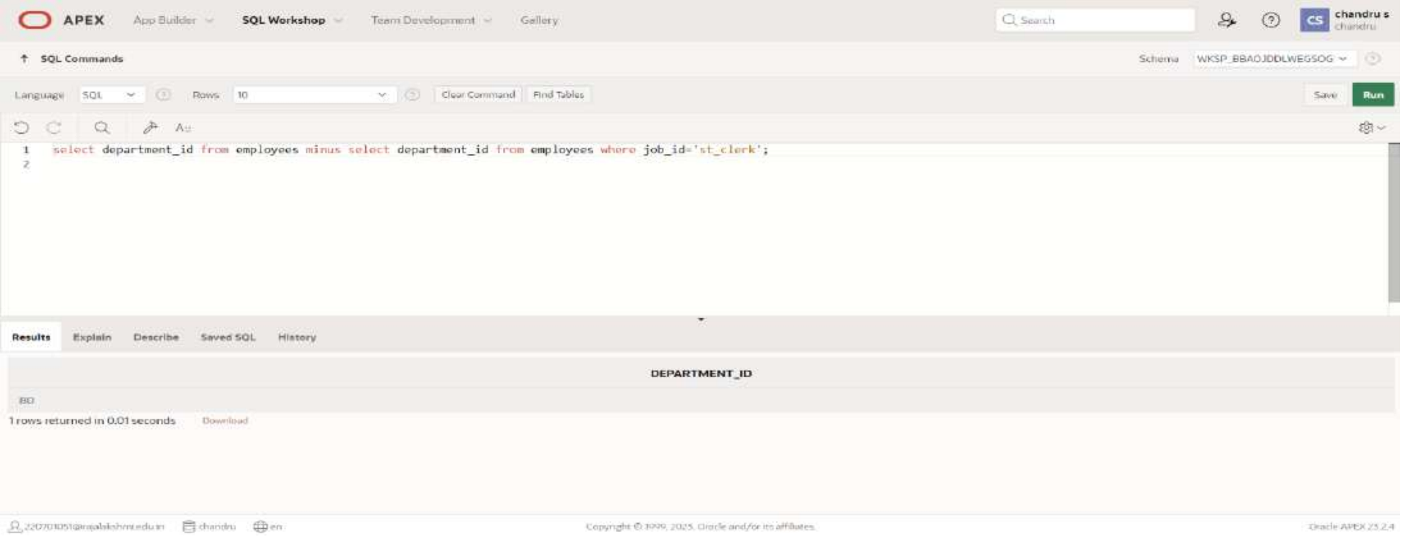
EX_NO:10 **DATE:**

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

QUERY:

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

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

DEPARTMENT_ID
80

1 rows returned in 0.01 seconds

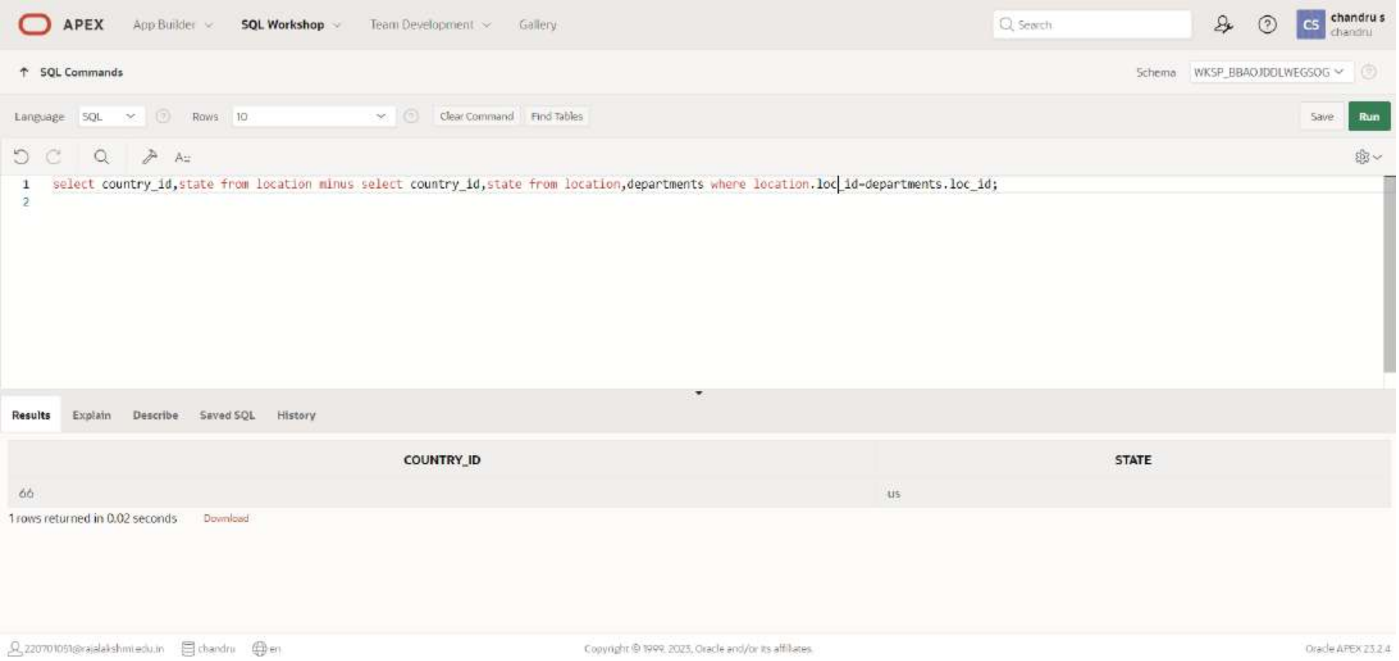
220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

```
select country_id,state from location minus select country_id,state from location,departments where location.loc_id=departments.loc_id;
```

COUNTRY_ID	STATE
66	us

1 rows returned in 0.02 seconds

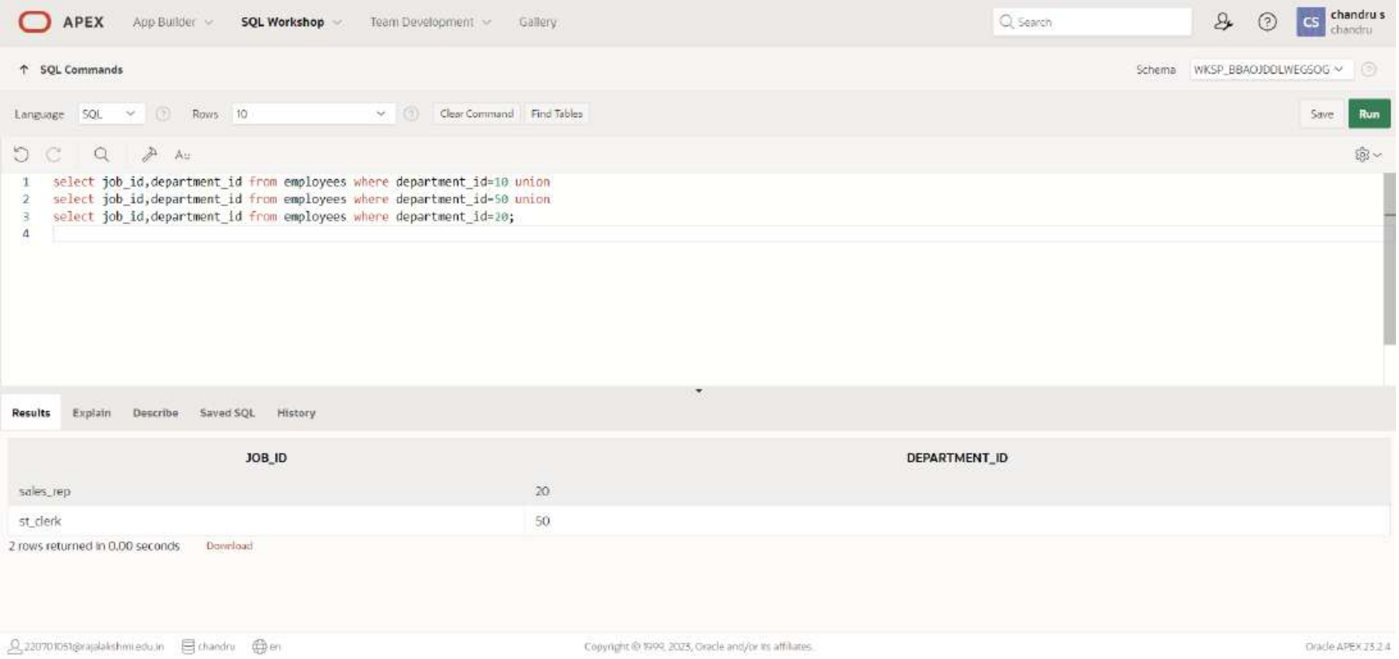
220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



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

JOB_ID	DEPARTMENT_ID
sales_rep	20
st_clerk	50

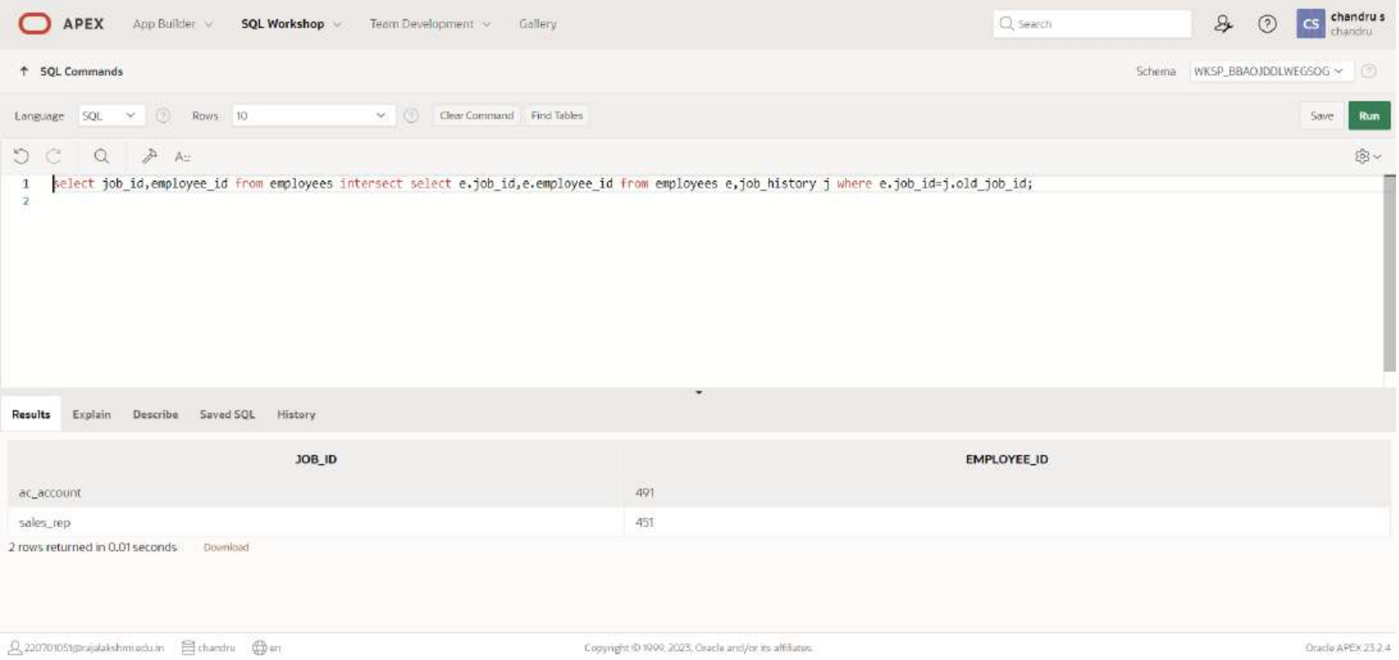
2 rows returned in 0.00 seconds

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:



```
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;
```

JOB_ID	EMPLOYEE_ID
ac_account	491
sales_rep	451

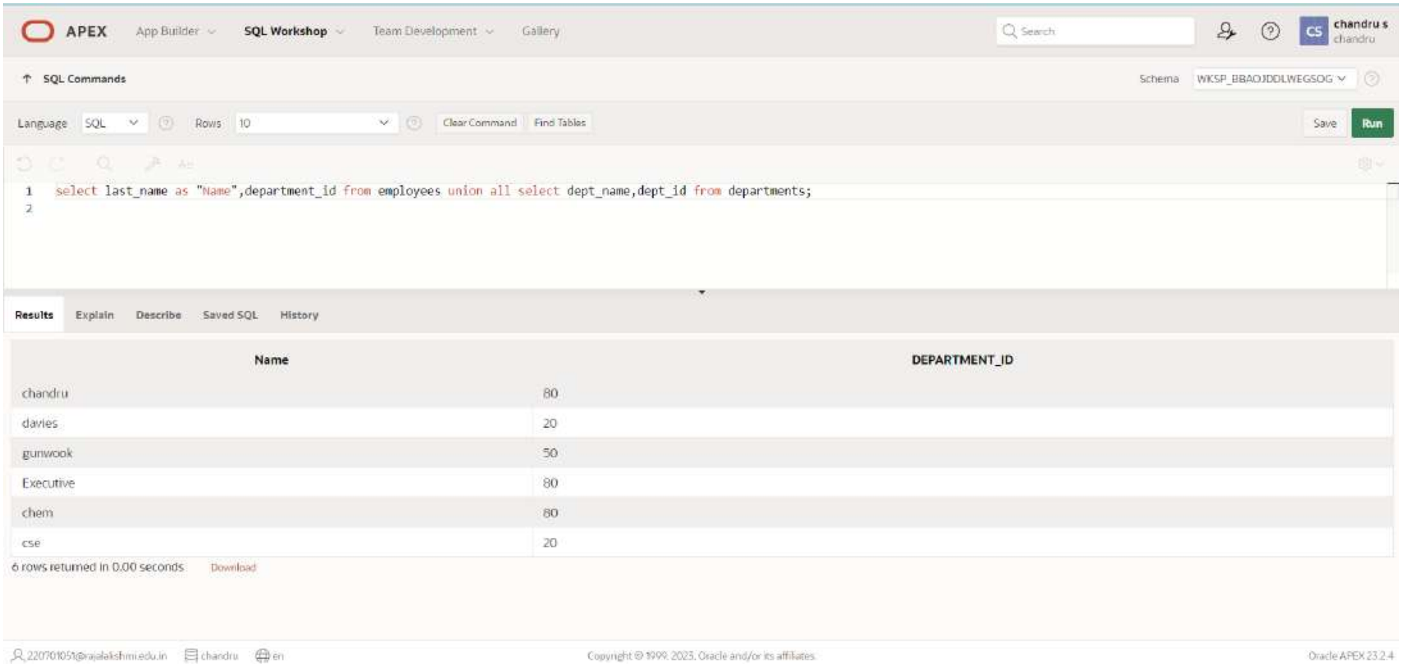
2 rows returned in 0.01 seconds

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,department_id from departments;

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

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

Name	DEPARTMENT_ID
chandru	80
davies	20
gunwook	50
Executive	80
chem	80
cse	20

6 rows returned in 0.00 seconds

220701051@rajalakshmi.edu.in chandru 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:

CREATING VIEWS

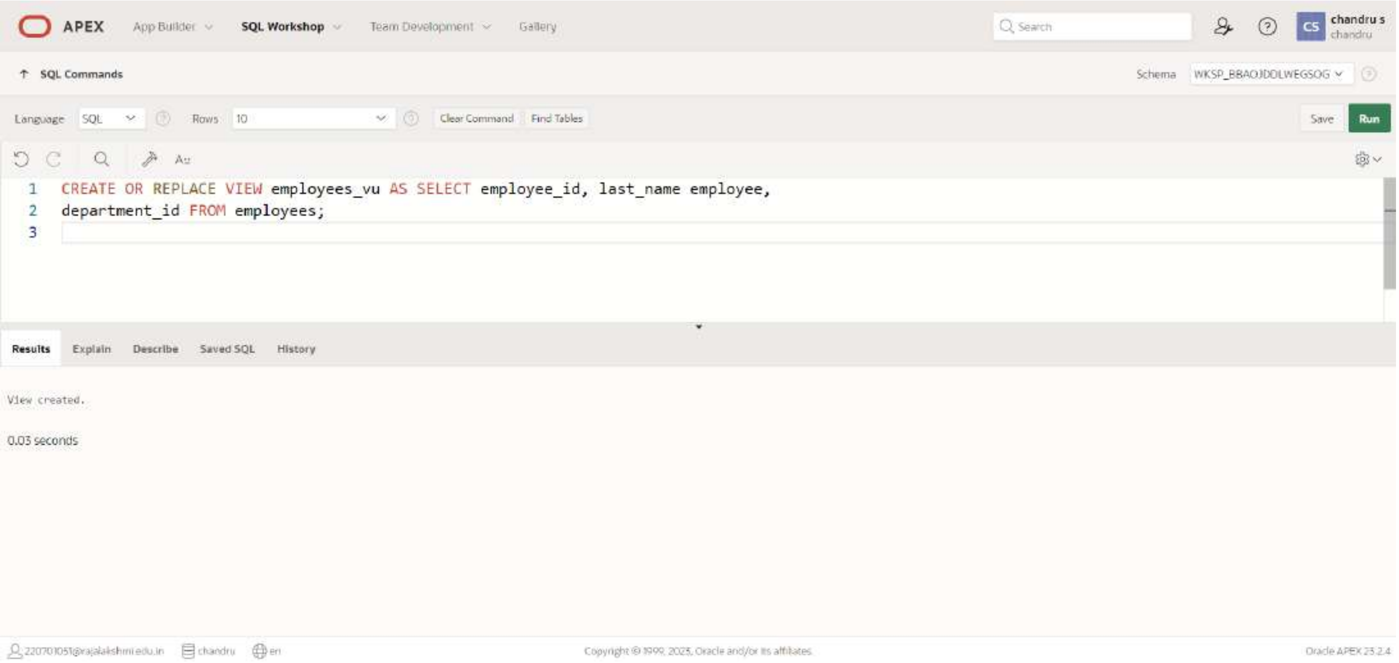
EX_NO:11 **DATE:**

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

QUERY:

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

OUTPUT:



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

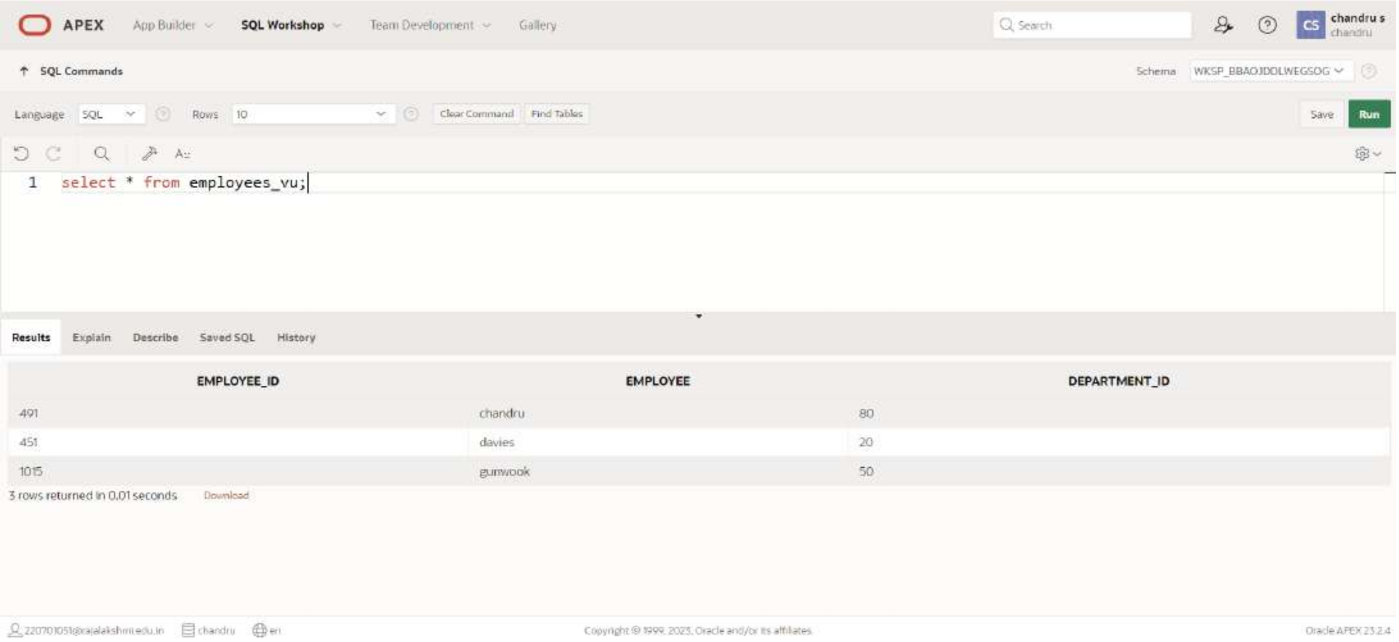
```
View created.

0.03 seconds
```

2.) Display the contents of the EMPLOYEES_VU view.

QUERY:

select * from employees_vu;

OUTPUT:



```
select * from employees_vu;
```

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
491	chandru	80
451	davies	20
1015	gunwook	50

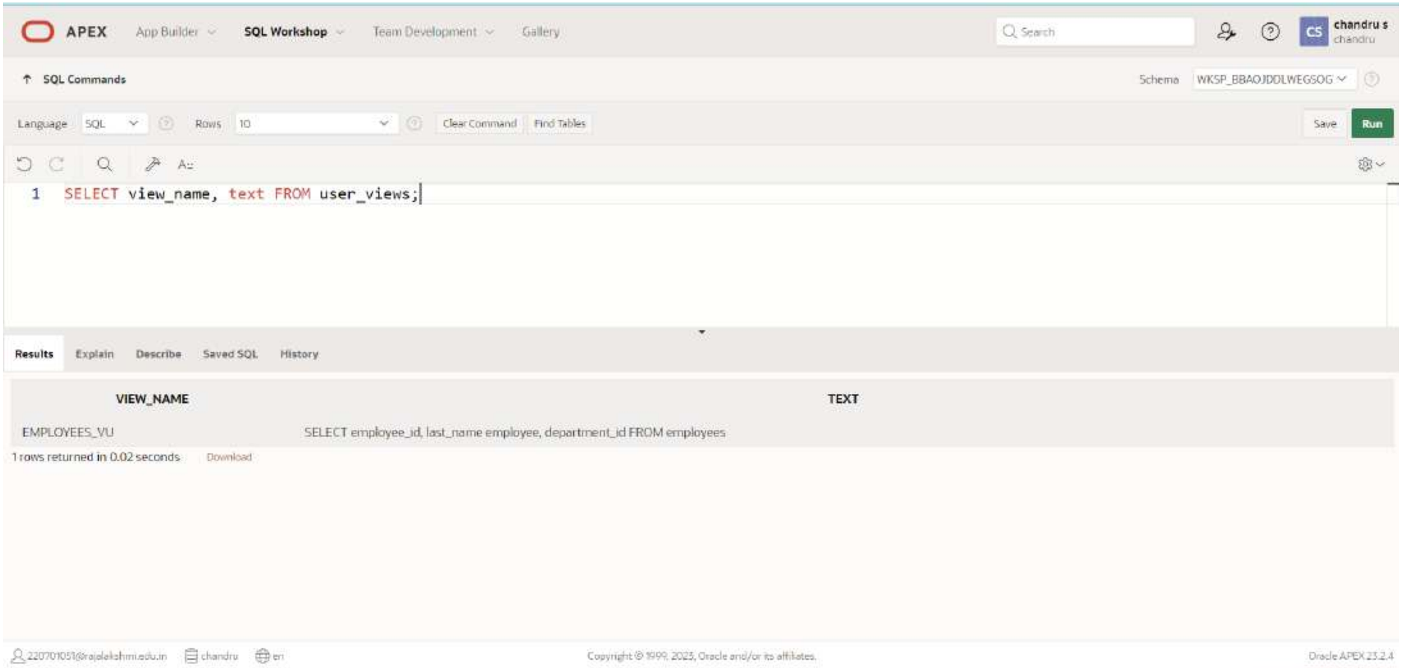
3 rows returned in 0.01 seconds

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

QUERY:

SELECT view_name, text FROM user_views;

OUTPUT:



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

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

1 rows returned in 0.02 seconds Download

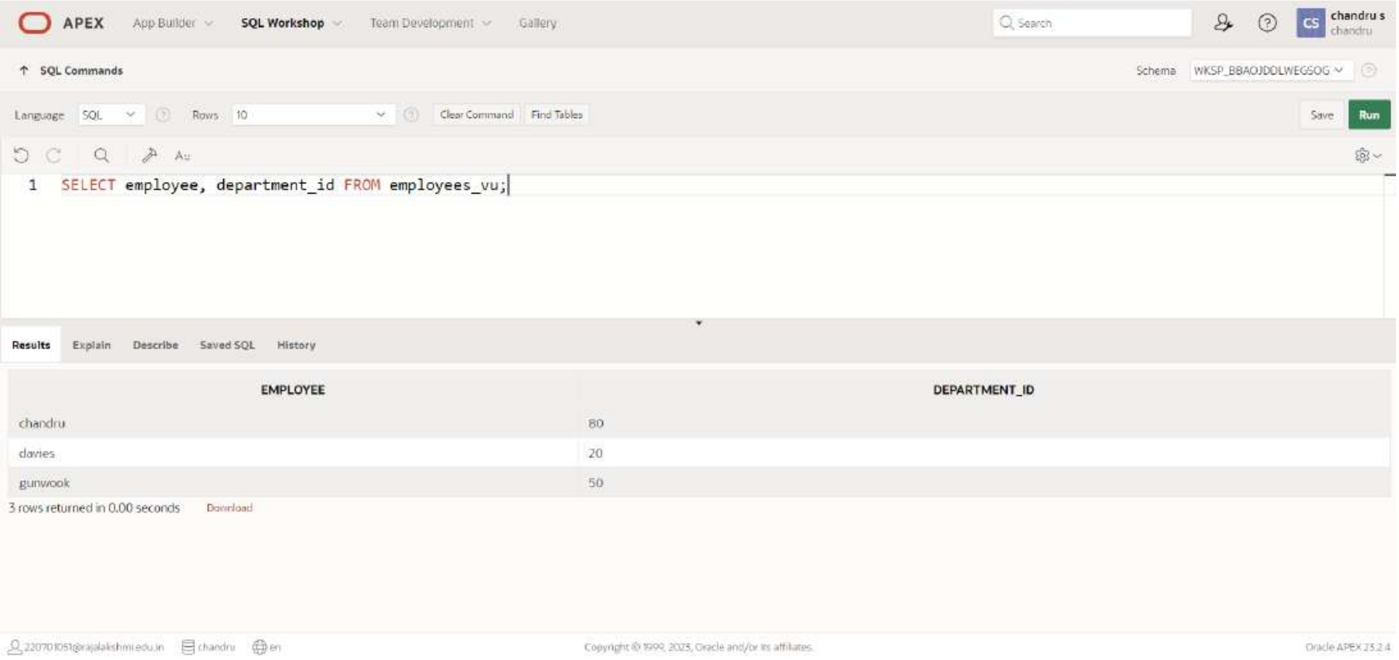
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

SELECT employee, department_id FROM employees_vu;

OUTPUT:



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

EMPLOYEE	DEPARTMENT_ID
chandru	80
davies	20
gunwook	50

3 rows returned in 0.00 seconds Download

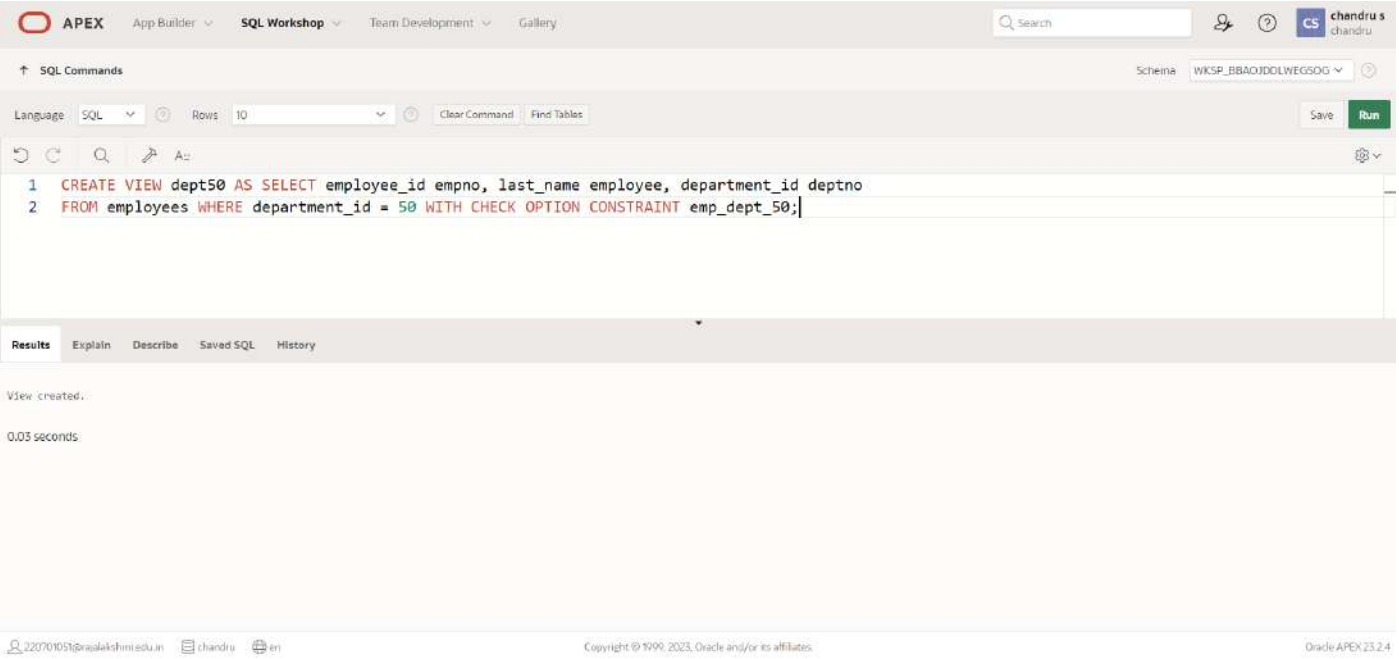
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



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

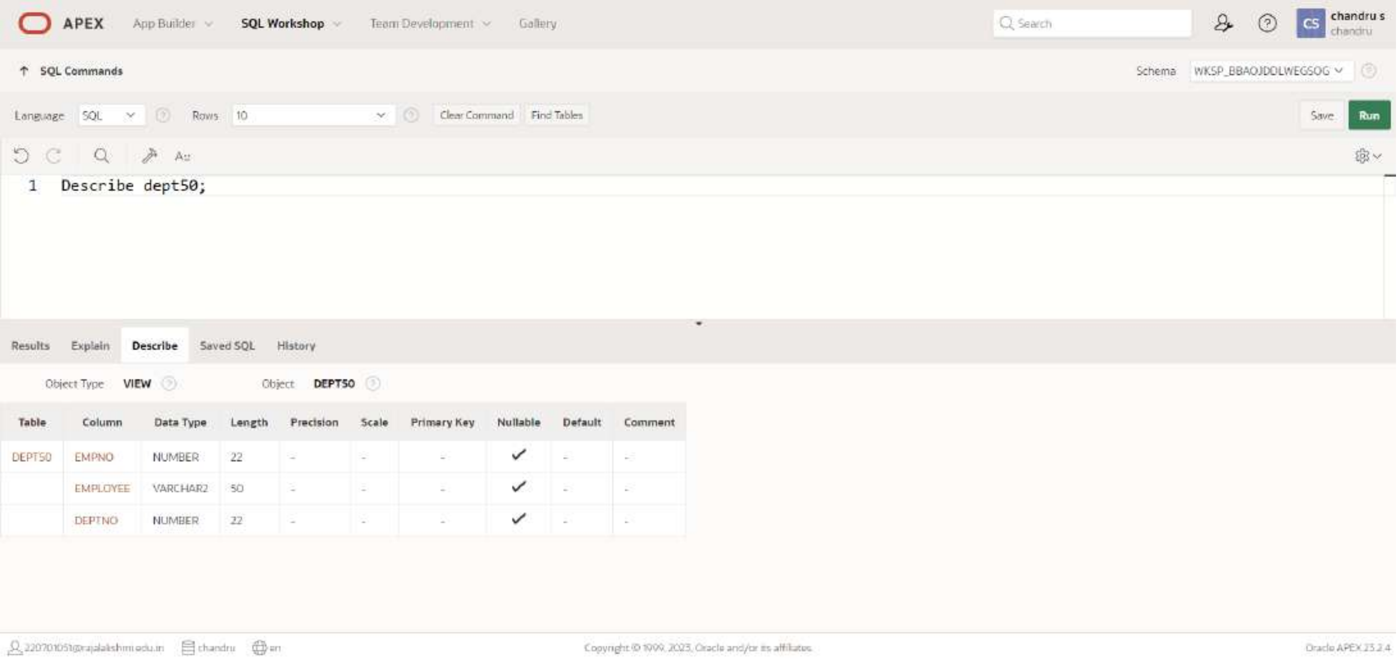
```
View created.

0.03 seconds
```

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

QUERY:

Describe dept50;

OUTPUT:



```
Describe dept50;
```

Object Type VIEW Object DEPT50

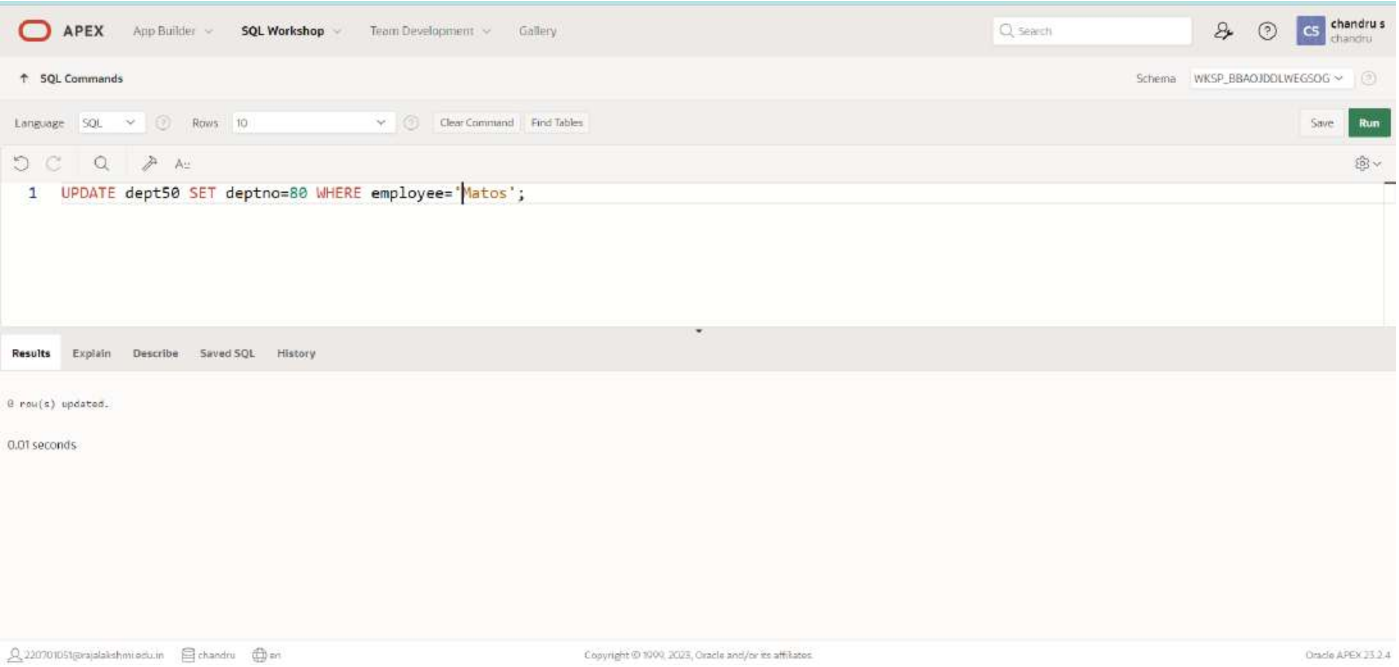
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT50	EMPNO	NUMBER	22	-	-	-	✓	-	-
	EMPLOYEE	VARCHAR2	50	-	-	-	✓	-	-
	DEPTNO	NUMBER	22	-	-	-	✓	-	-

7.) Attempt to reassign Matos to department 80

QUERY:

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

OUTPUT:



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

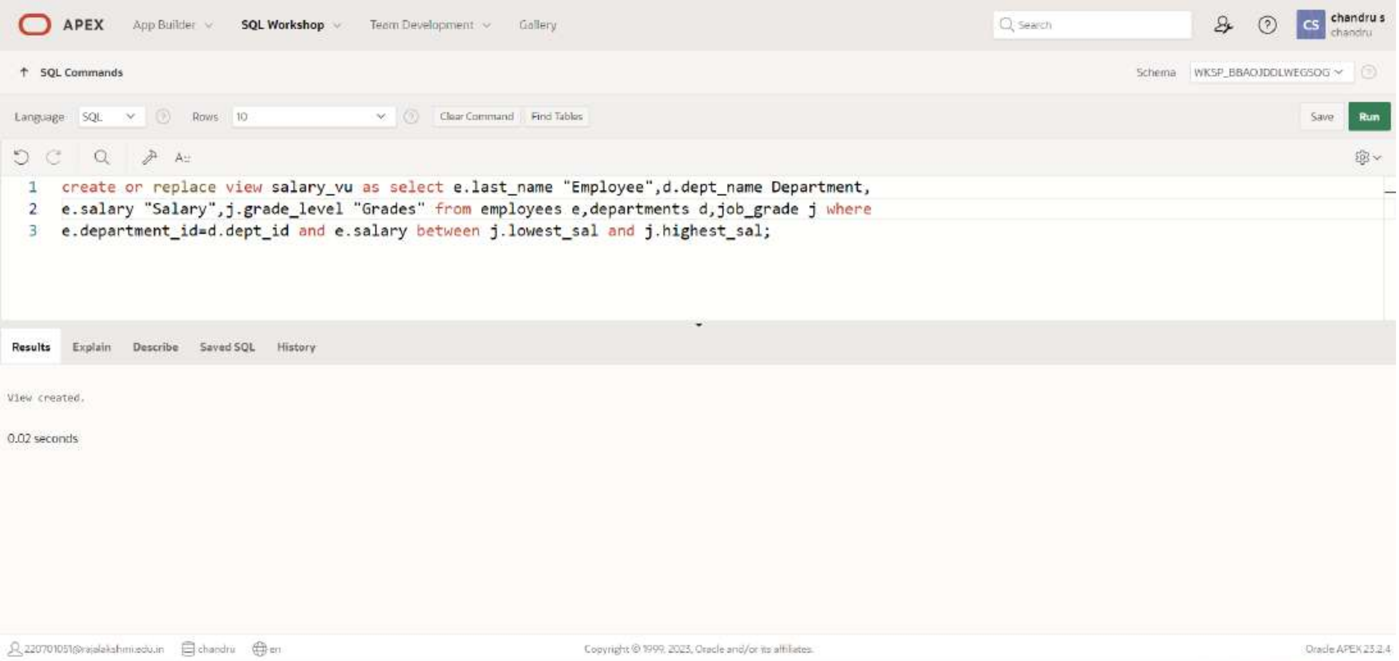
```
0 row(s) updated.

0.01 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:



```
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;
```

```
View created.

0.02 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)
);
```

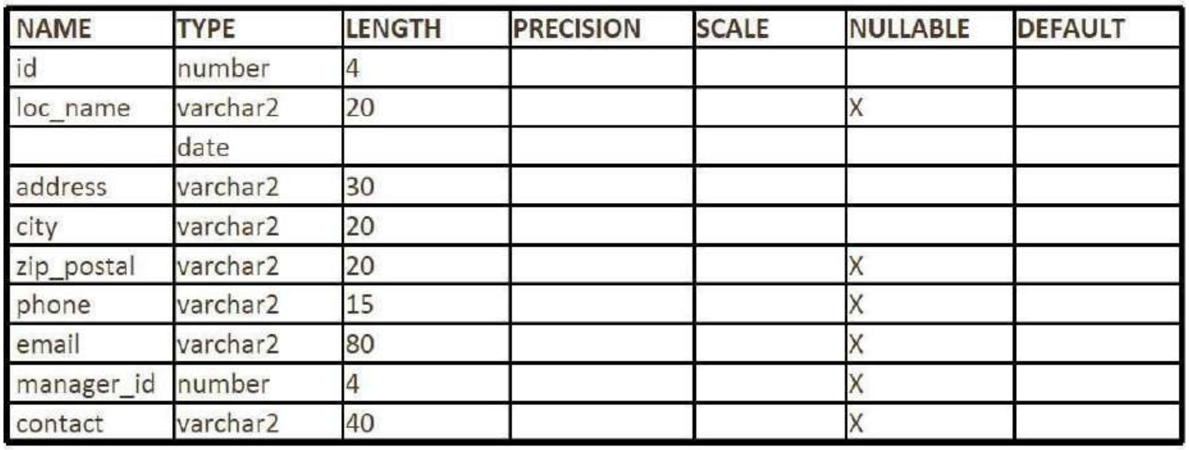
7. Execute the CREATE TABLE statement in Oracle Application Express.

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 tablenames 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 tablenames 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	

RESULT:

EXERCISE 13

Creating Views

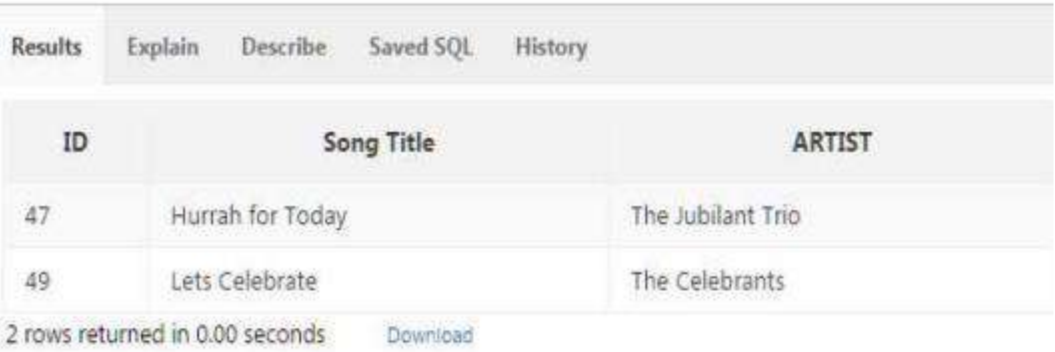
1. What are three uses for a view from a DBA's perspective?

- **Restrict access and display selective columns**
- **Reduce complexity of queries from other internal systems. So, providing a way to view same data in a different manner.**
- **Let the app code rely on views and allow the internal implementation of tables to be modified later.**

2. Create a simple view called view_d_songs that contains the ID, title and artist from the DJs on Demand table for each "New Age" type code. In the subquery, use the alias "Song Title" for the title column.

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

3. SELECT * FROM view_d_songs. What was returned?



ID	Song Title	ARTIST
47	Hurrah for Today	The Jubilant Trio
49	Lets Celebrate	The Celebrants

2 rows returned in 0.00 seconds

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?

P**rimary 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

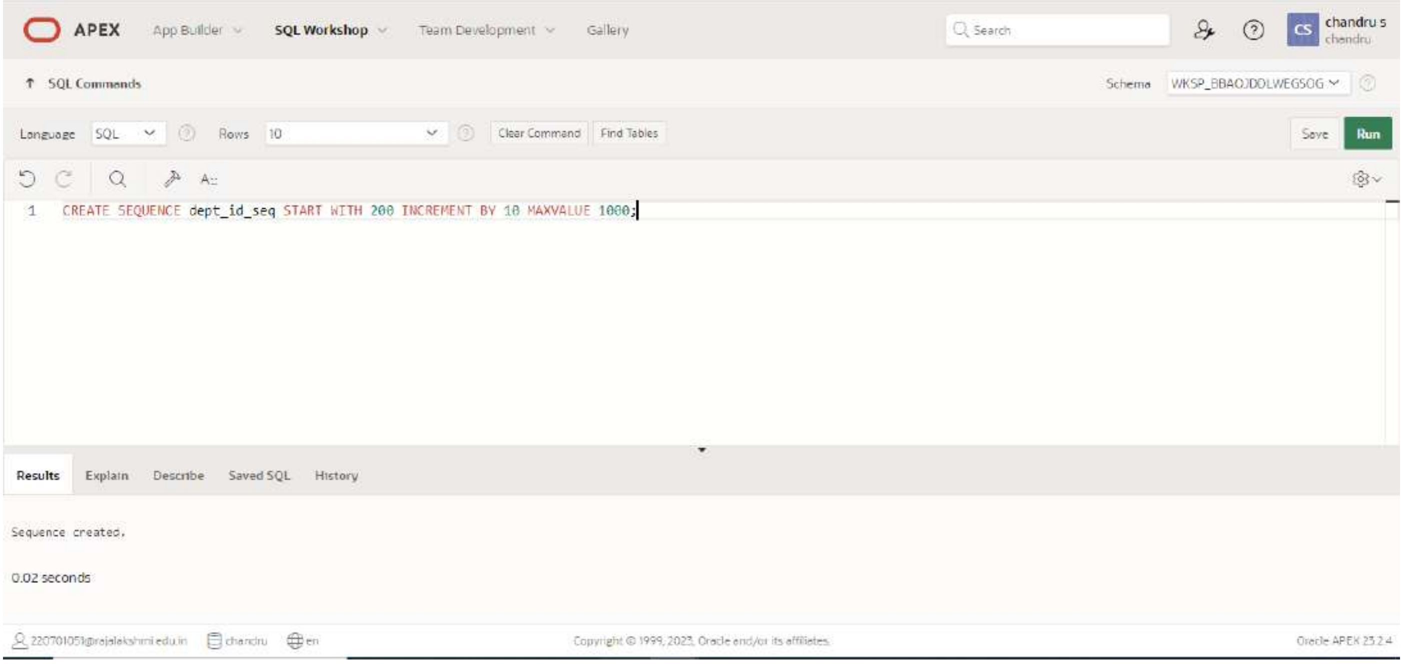
EX_NO:14 **DATE:**

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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

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

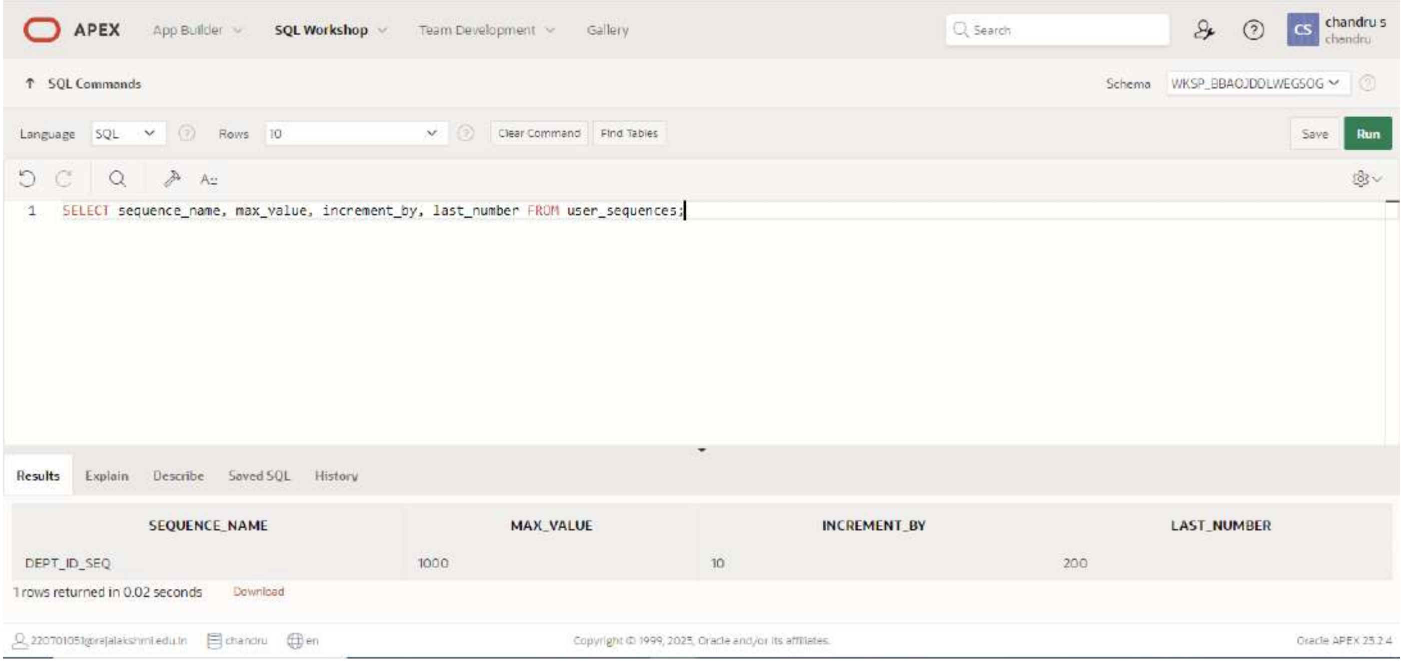
```
Sequence created.

0.02 seconds
```

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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

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

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_ID_SEQ	1000	10	200

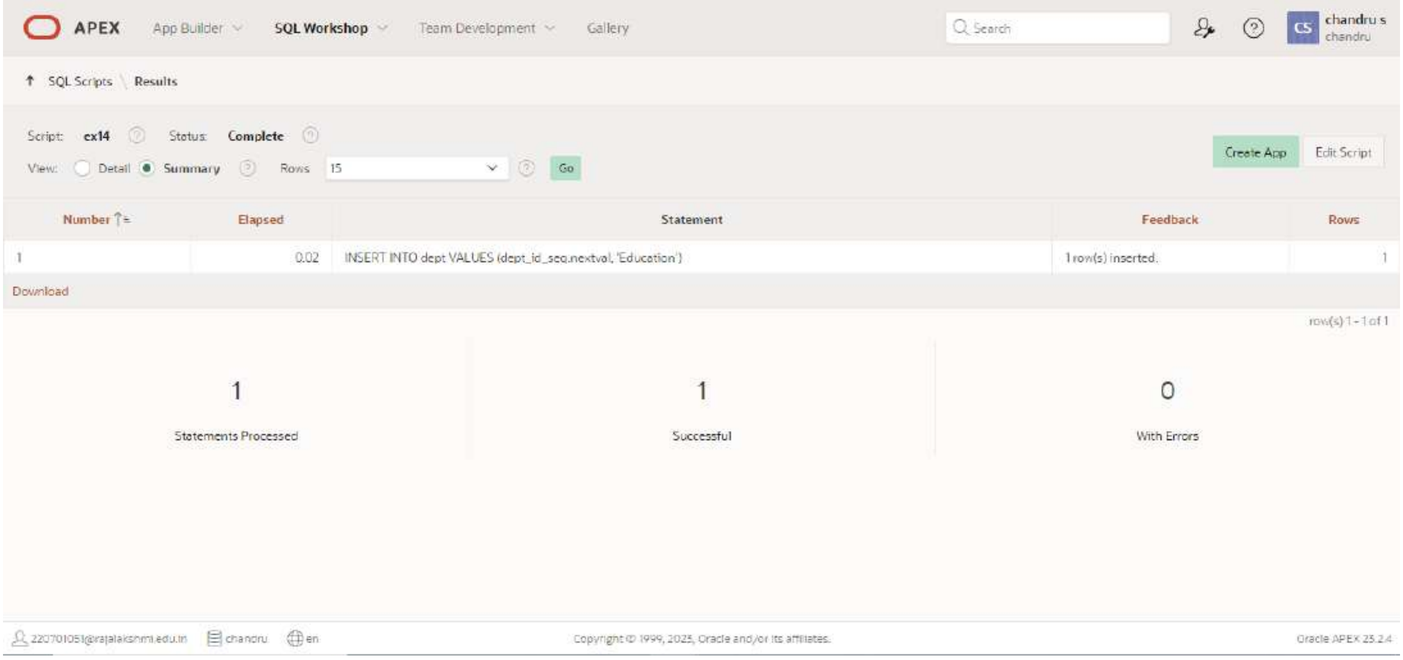
1 rows returned in 0.02 seconds

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

QUERY:

INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education');
INSERT INTO dept VALUES (dept_id_seq.nextval, ‘Administration’);

OUTPUT:



Script: ex14 Status: Complete

View: ○ Detail ● Summary Rows 15

Number	Elapsed	Statement	Feedback	Rows
1	0.02	INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education')	1 row(s) inserted.	1

row(s) 1 - 1 of 1

1 Statements Processed

1 Successful

0 With Errors

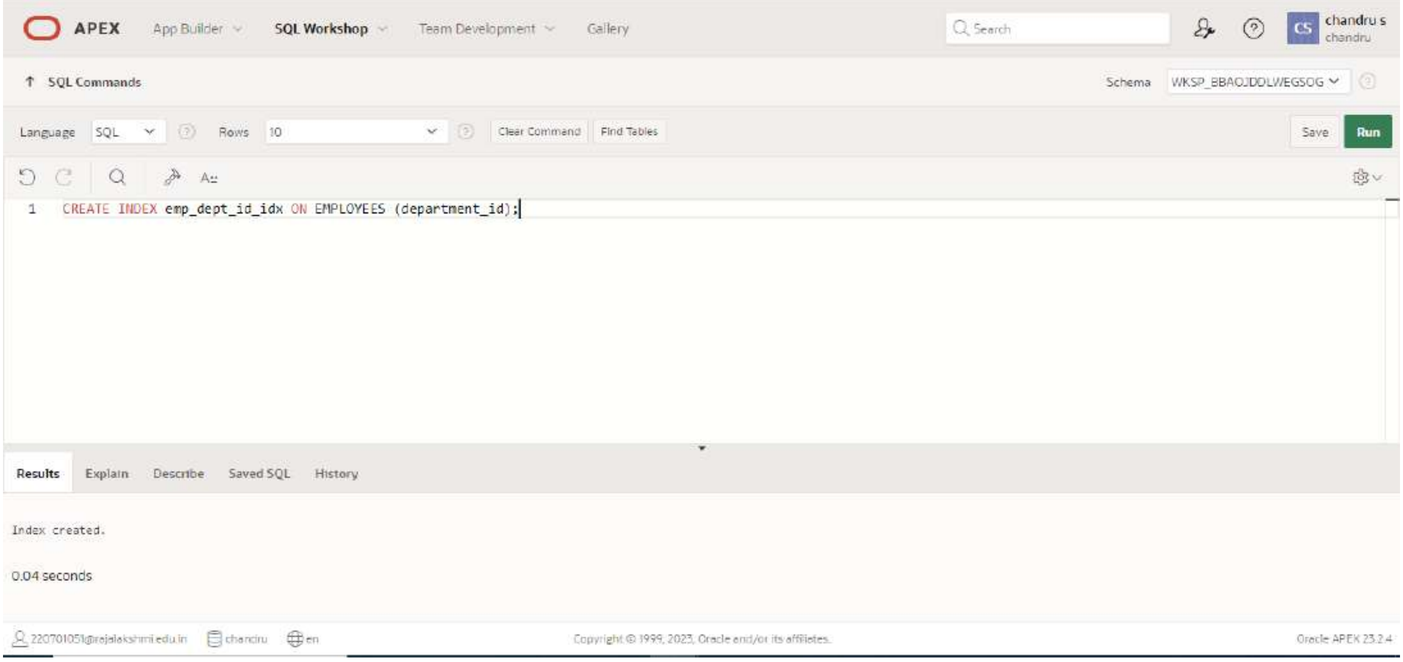
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

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

```
Index created.

0.04 seconds
```

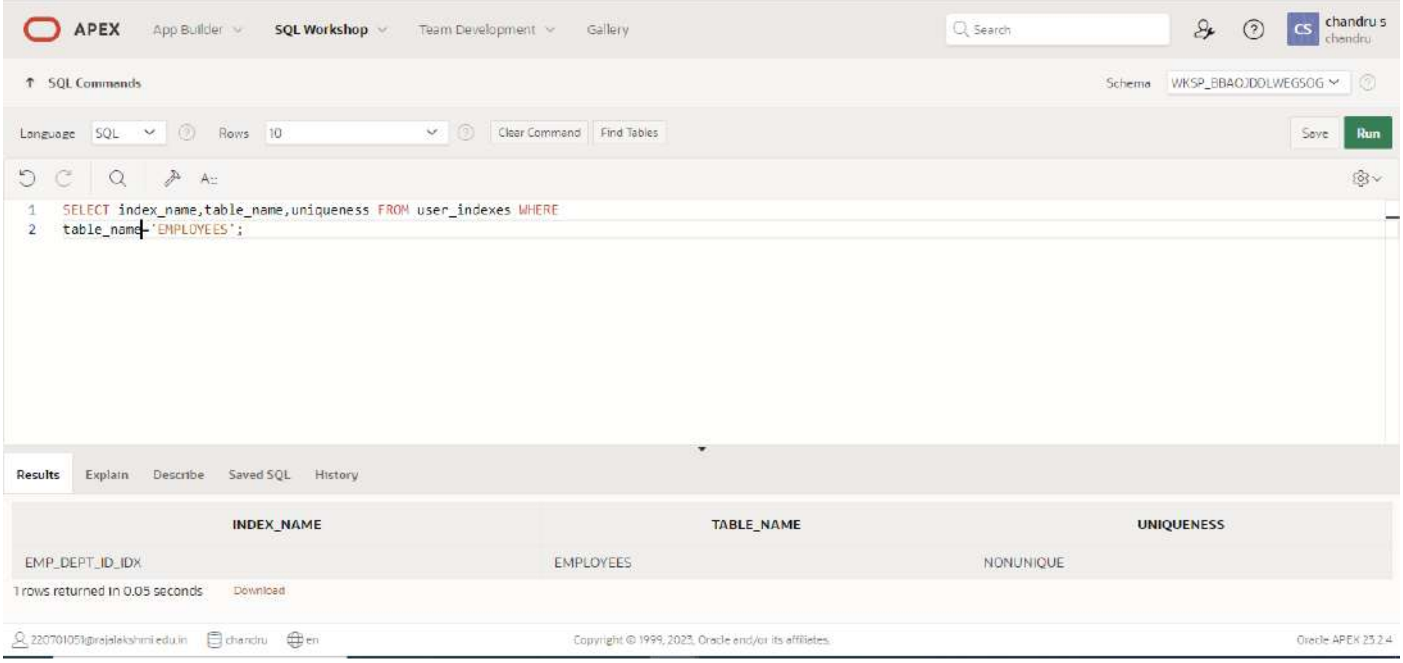
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

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

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

1 rows returned in 0.05 seconds Download

220701051@rajalakshmi.edu.in chandru 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:

CONTROLLING USER ACCESS

EX_NO:15 **DATE:**

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

The CREATE SESSION system privilege

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

The CREATE TABLE privilege

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

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

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

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

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

The ALTER USER statement

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

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

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

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

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

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

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

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

SELECT table_name FROM user_tables;

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

Team 1 revokes the privilege.

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

Team 2 revokes the privilege.

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

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

Team 1 executes this INSERT statement.

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

Team 2 executes this INSERT statement.

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

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	
Viva(5)	
Total (10)	
Faculty Signature	

RESULT:

PL/SQL

CONTROL STRUCTURES

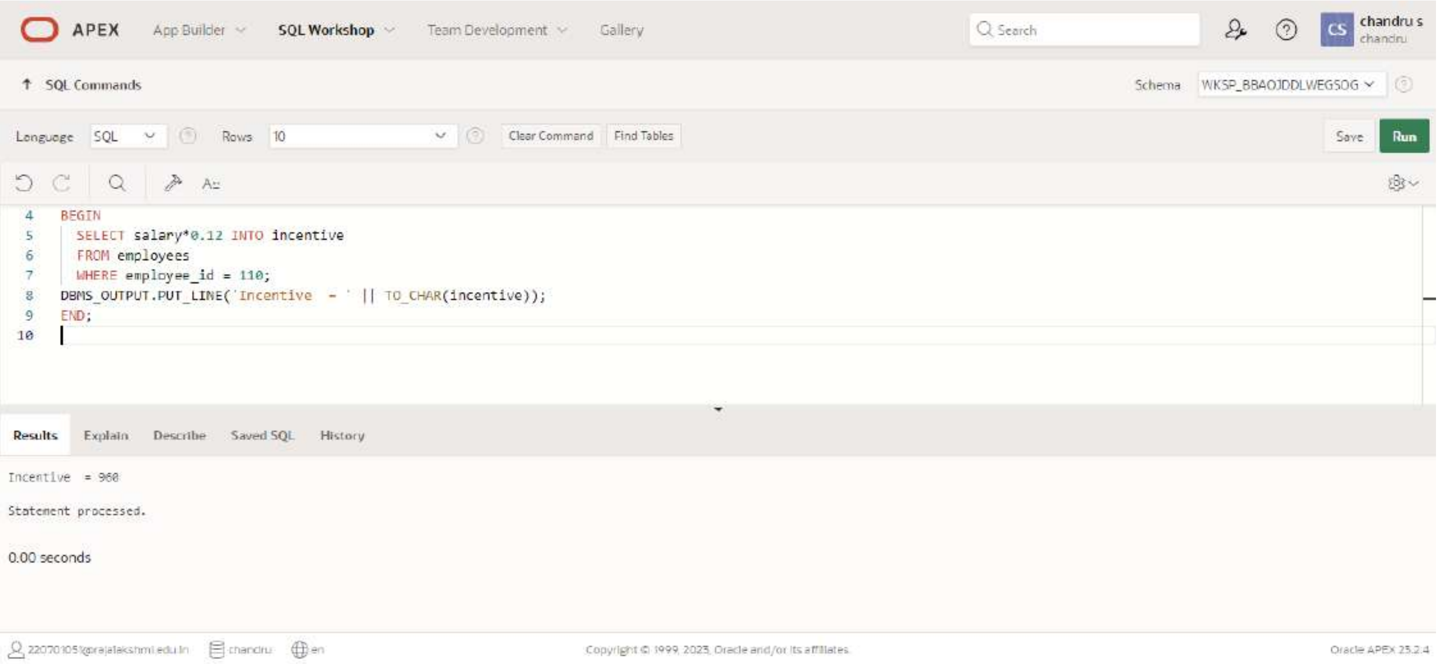
EX_NO:16 **DATE:**

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

QUERY:

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

OUTPUT:



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

```
Incentive  = 960

Statement processed.

0.00 seconds
```

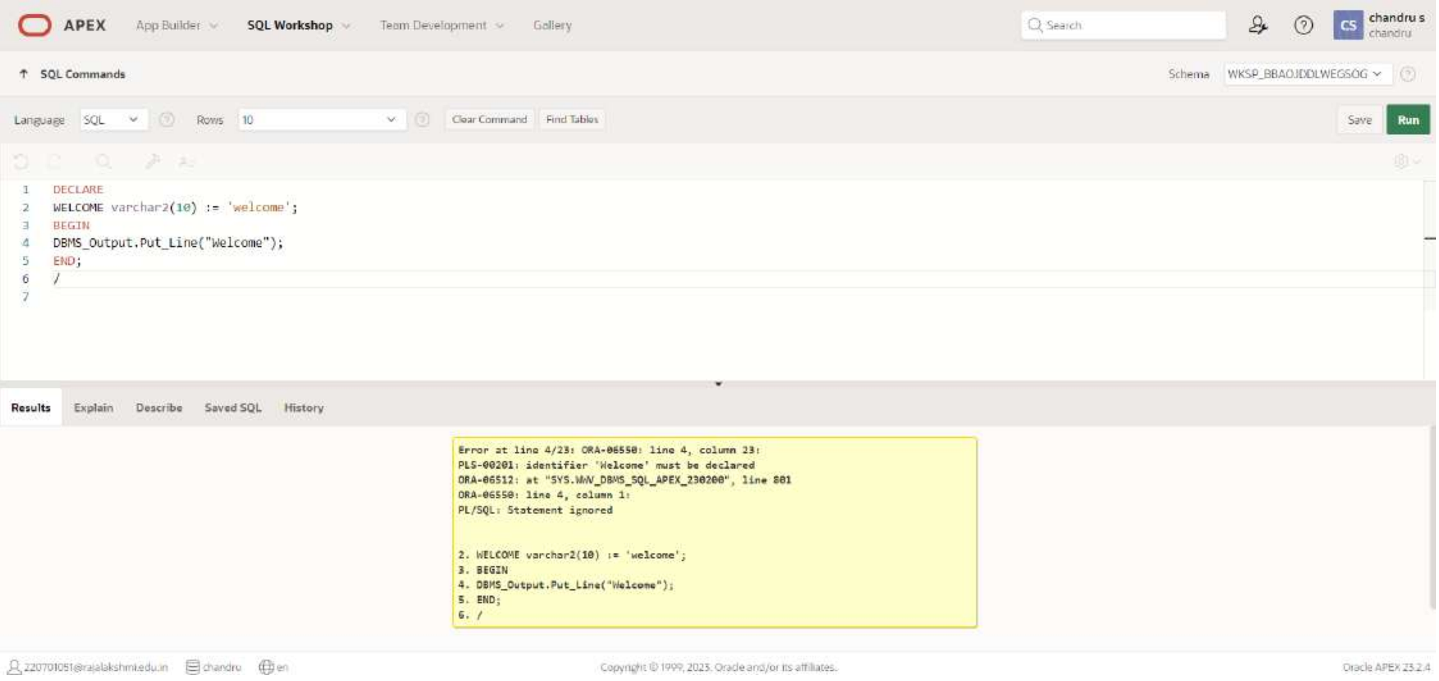
2.) Write a PL/SQL block to show an invalid case-insensitive reference to a quoted and without quoted user-defined identifier

QUERY:

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

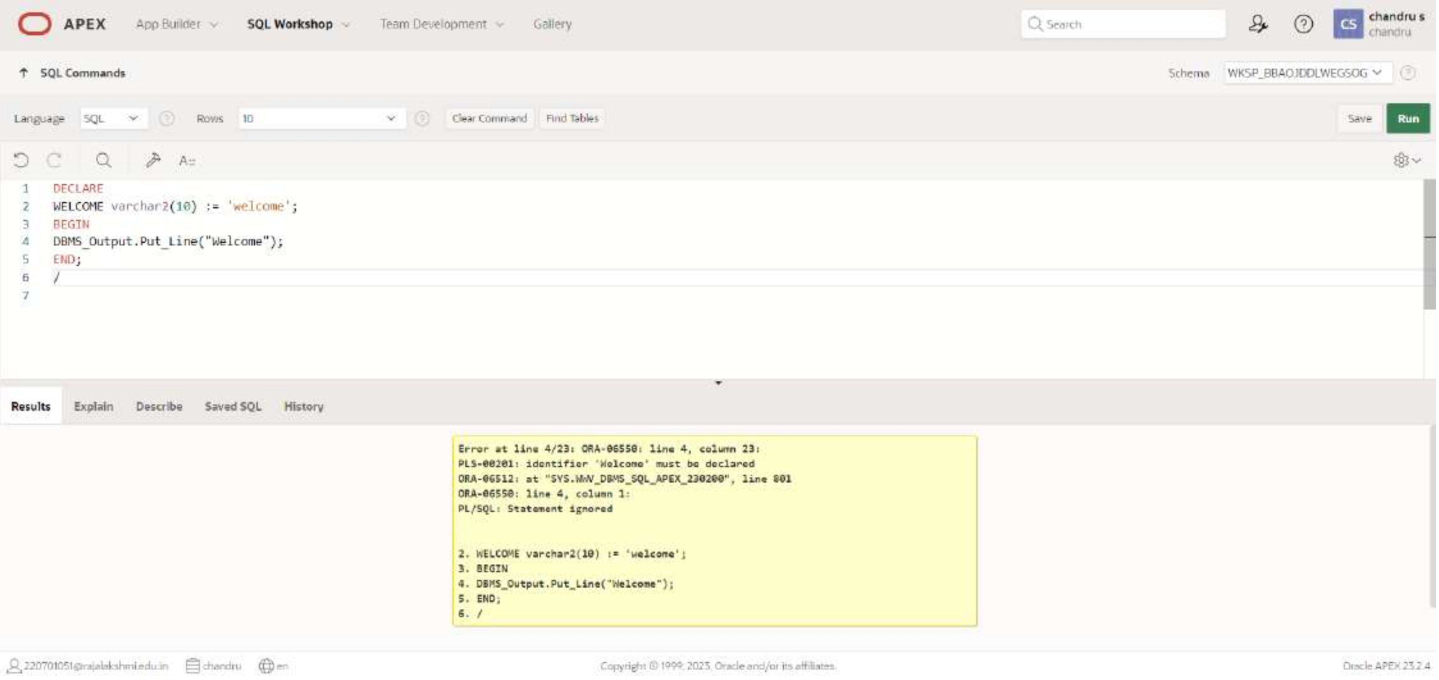
```
DECLARE
WELCOME varchar2(10) := 'welcome';
BEGIN
DBMS_Output.Put_Line("Welcome");
END;
/
```

OUTPUT:



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

```
Error at line 4/23: ORA-06550: line 4, column 23:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored


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



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

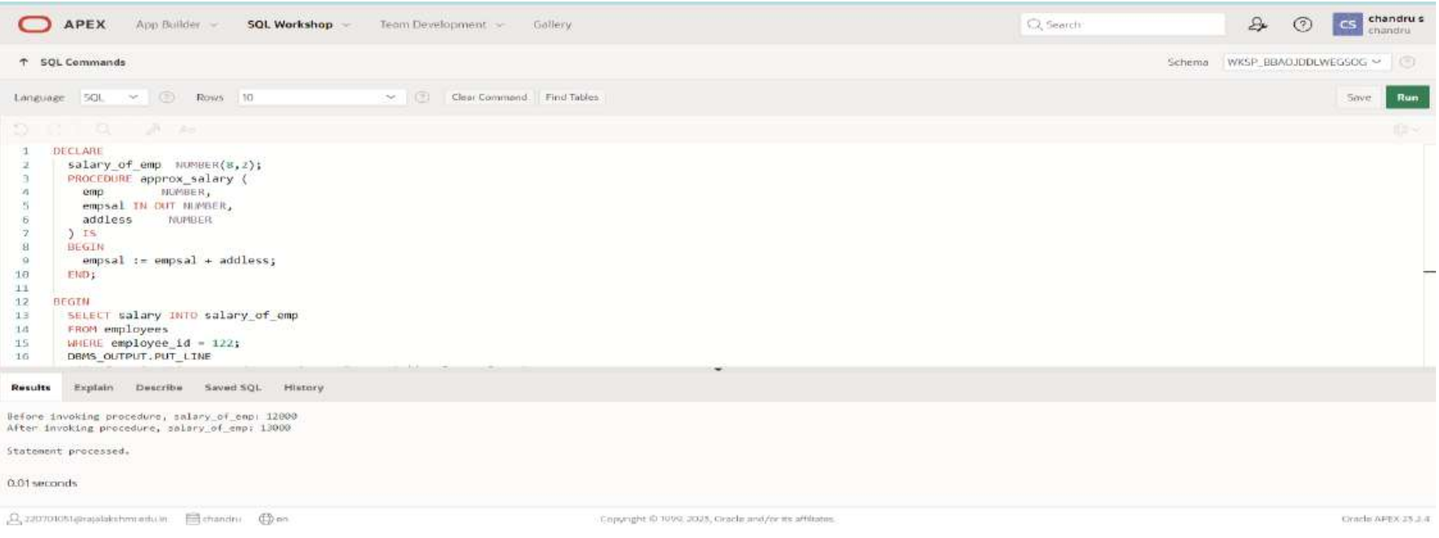
```
Error at line 4/23: ORA-06550: line 4, column 23:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_230200", line 801
ORA-06550: line 4, column 1:
PL/SQL: Statement ignored


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

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:



```
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
```

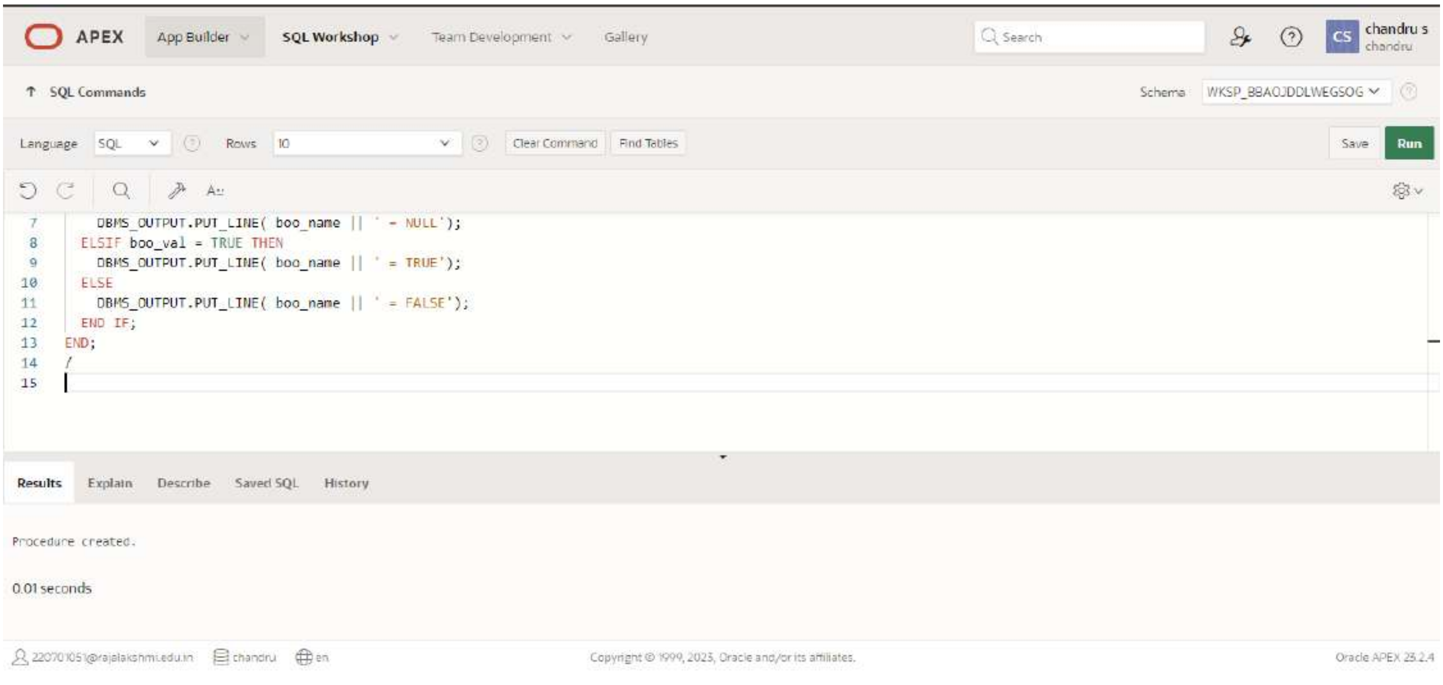
```
Before invoking procedure, salary_of_emp: 12000
After invoking procedure, salary_of_emp: 13000

Statement processed.

0.01 seconds
```

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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
    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;
/
```

```
Procedure created.
```

0.01 seconds

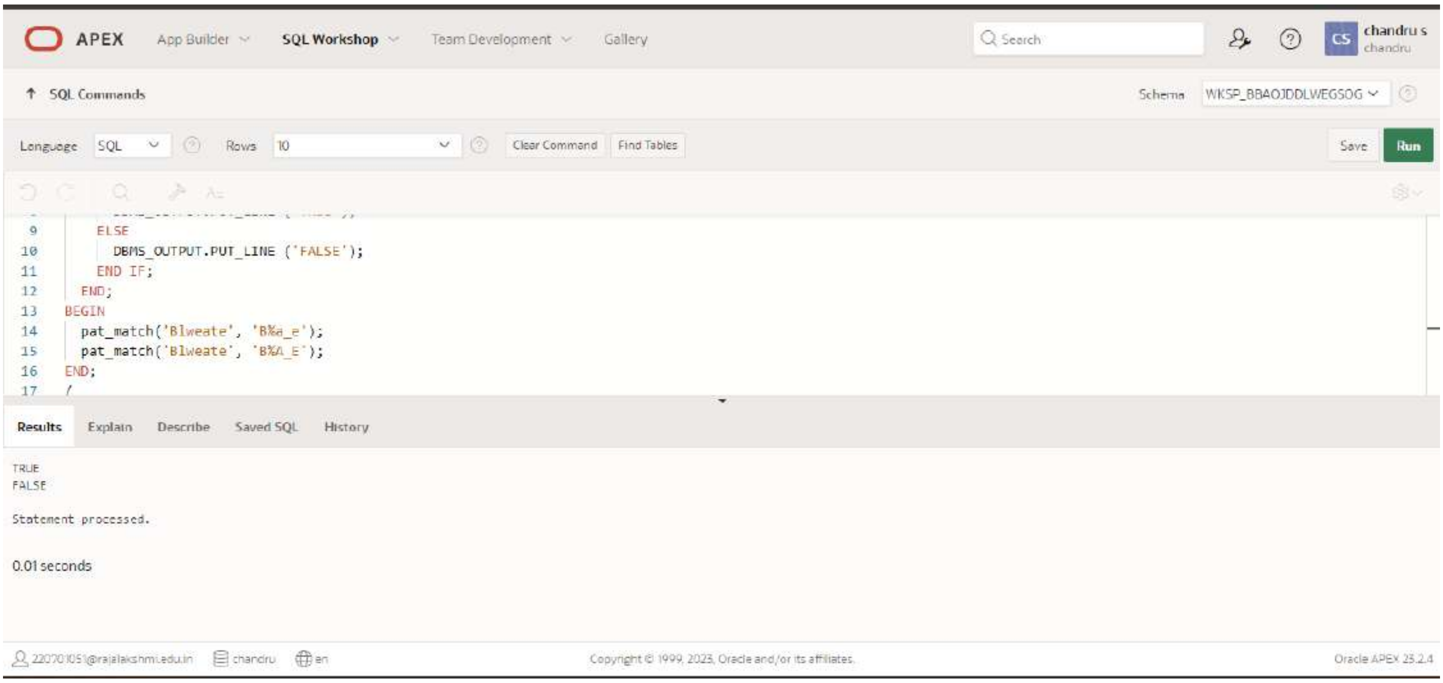
220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
    ELSE
      DBMS_OUTPUT.PUT_LINE ('FALSE');
    END IF;
  END;
BEGIN
  pat_match('Blweate', 'B%a_e');
  pat_match('Blweate', 'B%A_E');
END;
/
```

```
TRUE
FALSE

Statement processed.
```

0.01 seconds

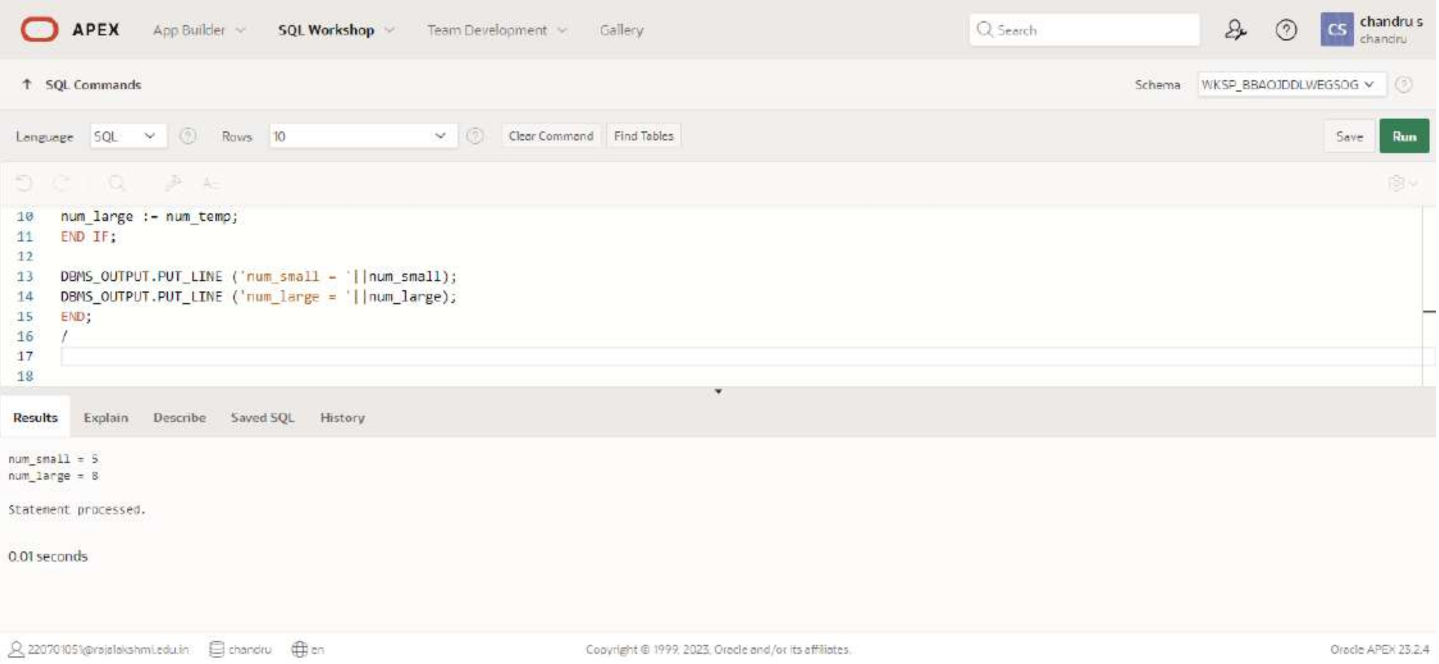
220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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:



Schema: WKSP_BBAOJDDLWEGSOG

Language: SQL

Rows: 10

```
num_large := num_temp;
END IF;

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

```
num_small = 5
num_large = 8

Statement processed.
```

0.01 seconds

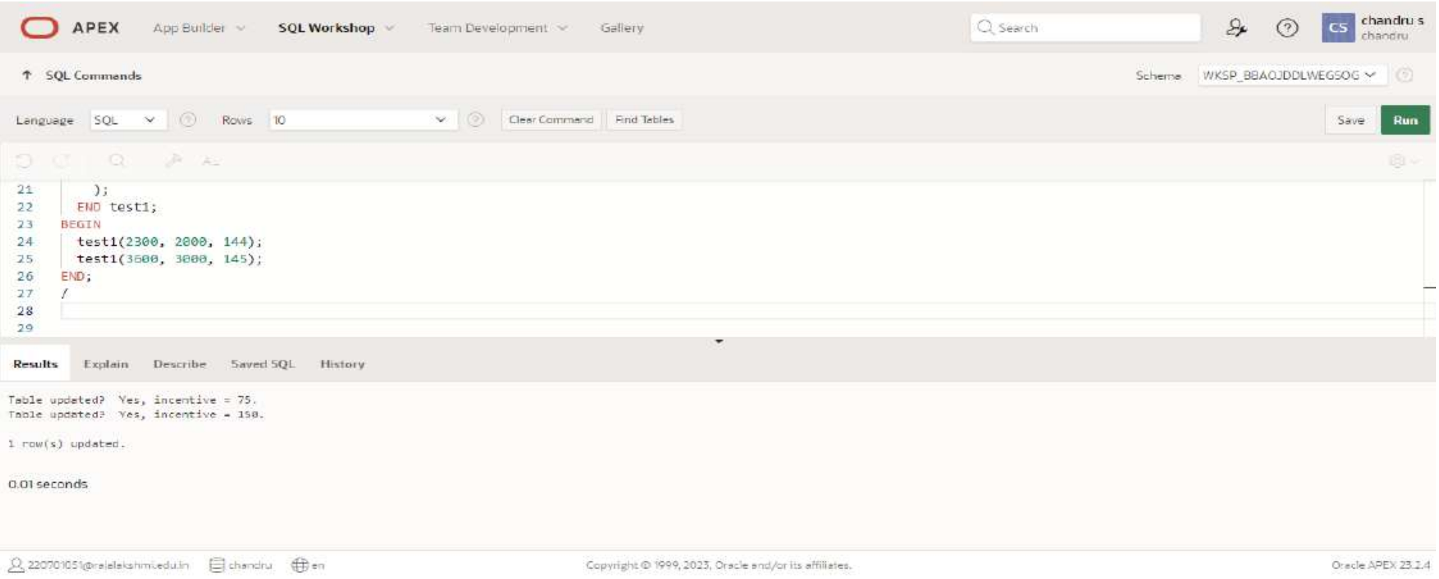
220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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:



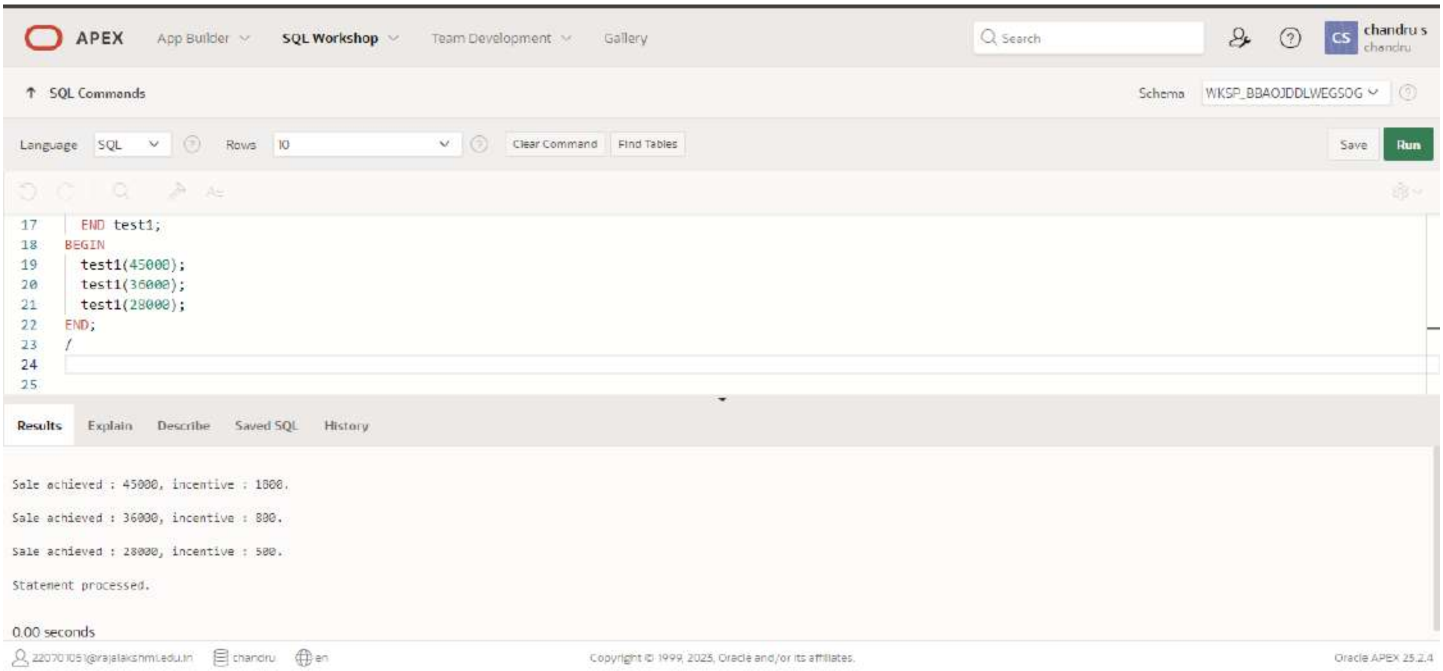
```
    );
  END test1;
BEGIN
  test1(2300, 2000, 144);
  test1(3600, 3000, 145);
END;
/
```

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

1 row(s) updated.

0.01 seconds
```

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;
/
```



```
  END test1;
BEGIN
  test1(45000);
  test1(36000);
  test1(28000);
END;
/
```

```
Sale achieved : 45000, incentive : 1800.

Sale achieved : 36000, incentive : 800.

Sale achieved : 28000, incentive : 500.

Statement processed.
```

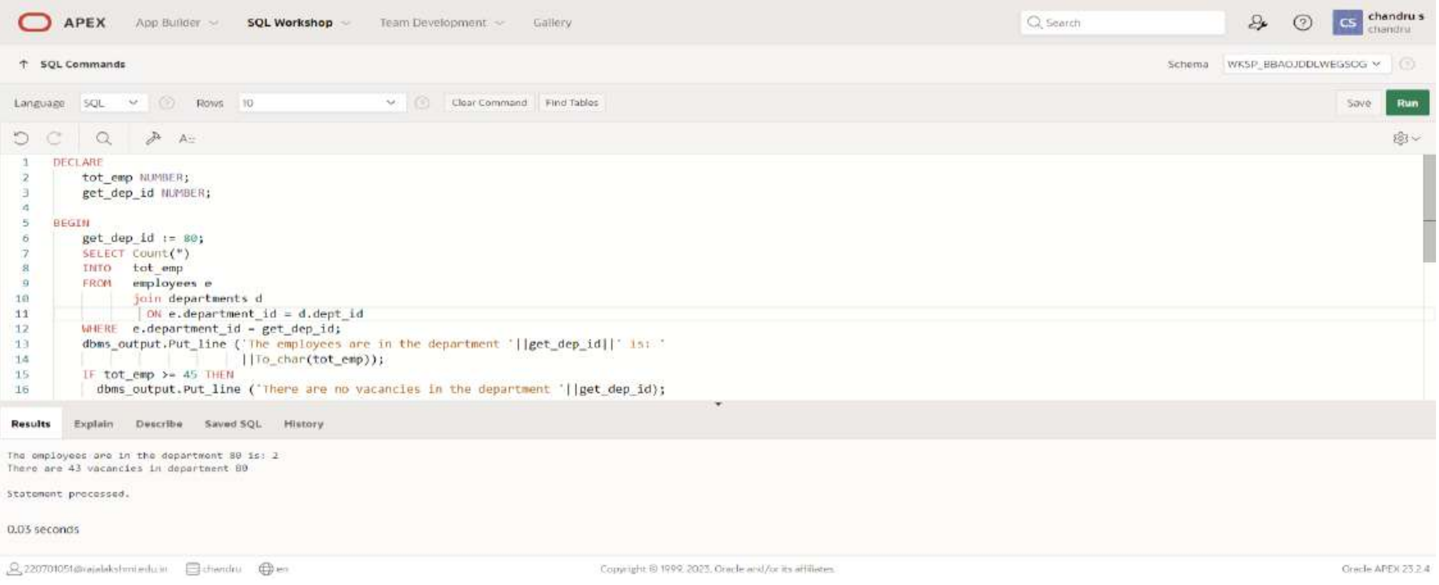
0.00 seconds

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

QUERY:

```
DECLARE
    tot_emp NUMBER;
           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:



```
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);
```

```
The employees are in the department 80 is: 2
There are 43 vacancies in department 80

Statement processed.
```

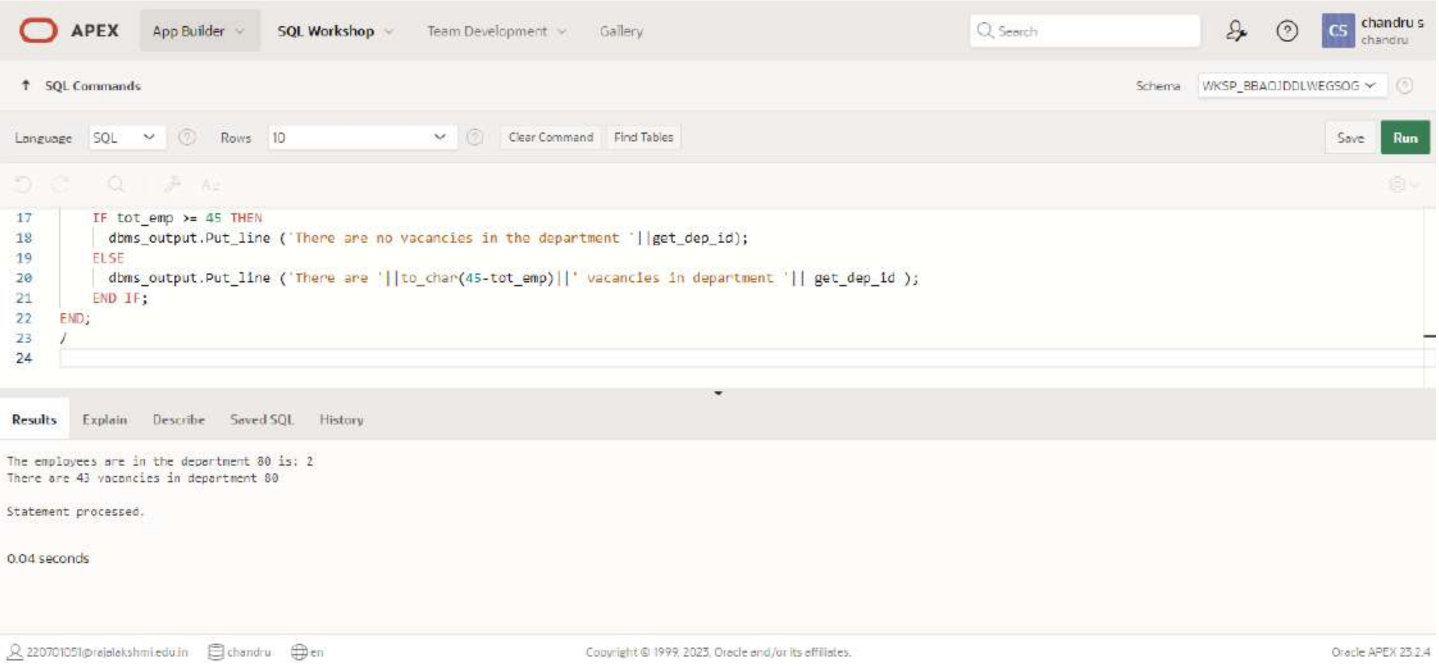
0.03 seconds

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:



```
    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;
/
```

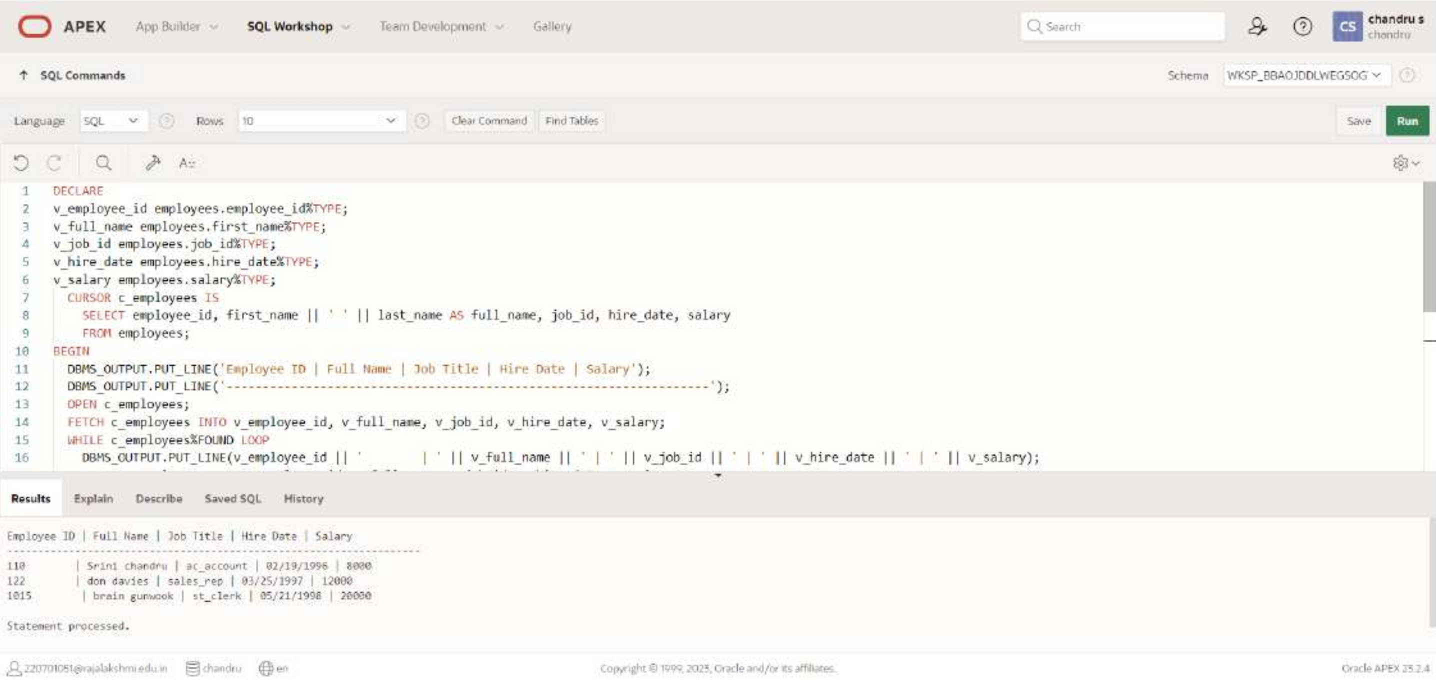
```
The employees are in the department 80 is: 2
There are 43 vacancies in department 80

Statement processed.

0.04 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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

```
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);
```

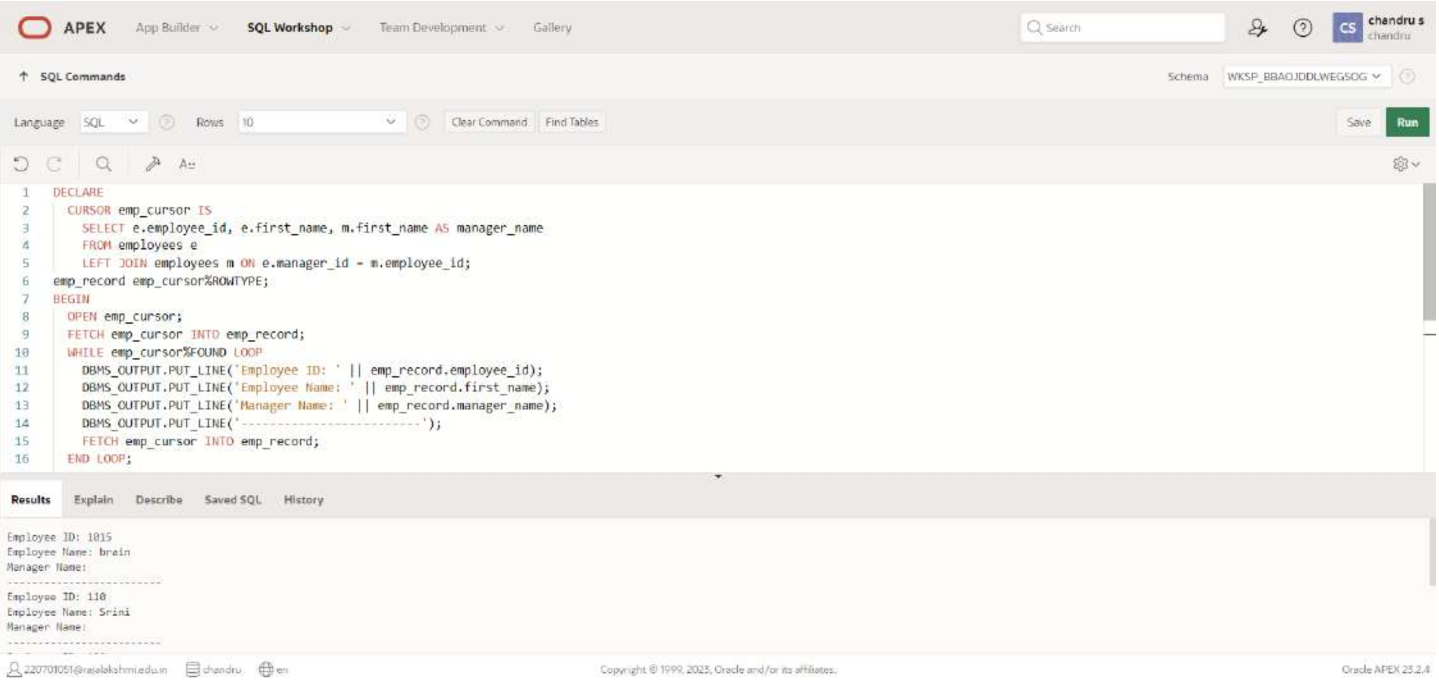
```
Employee ID | Full Name | Job Title | Hire Date | Salary
-----------------------------------------------------------------
110        | Srini chandru | ac_account | 02/19/1996 | 8000
122        | don davies | sales_rep | 03/25/1997 | 12000
1015        | brain gunwook | st_clerk | 05/21/1998 | 20000

Statement processed.
```

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:



```
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;
```

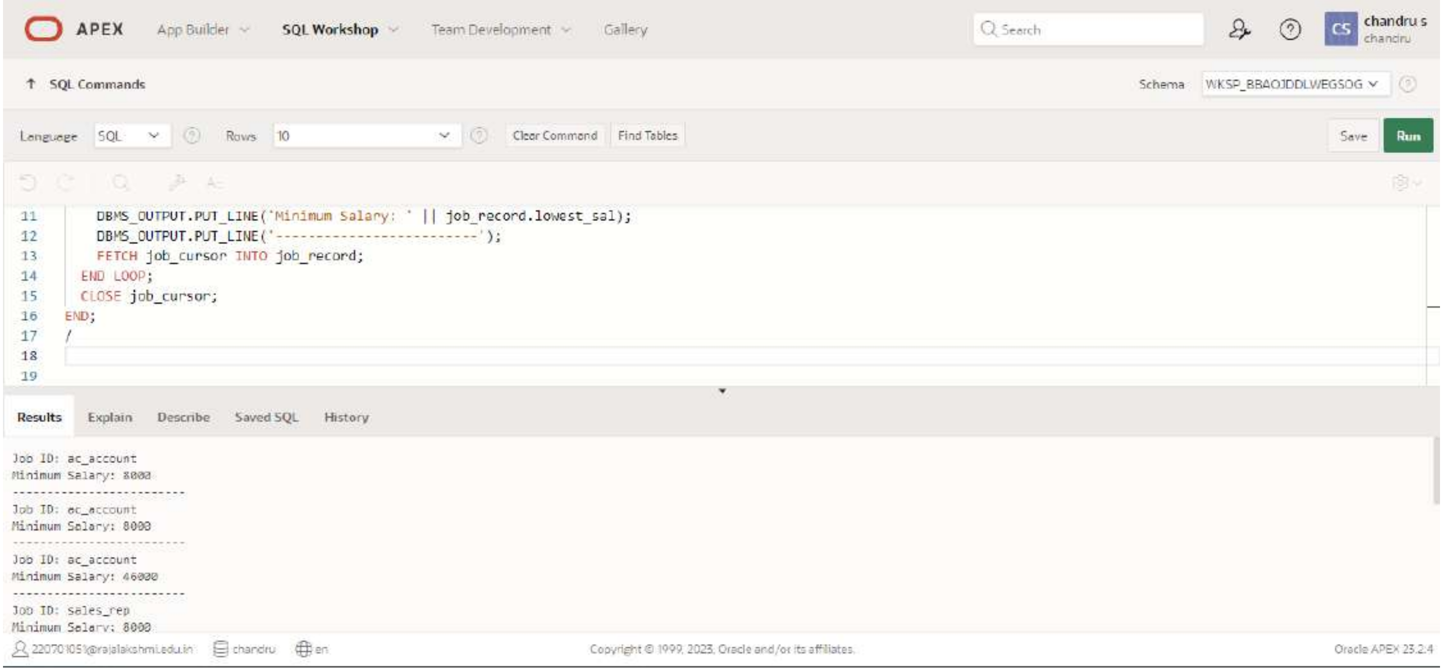
```
Employee ID: 1015
Employee Name: brain
Manager Name:
-------------------------
Employee ID: 110
Employee Name: Srini
Manager Name:
-------------------------
```

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

QUERY:

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

OUTPUT:



```
    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;
/
```

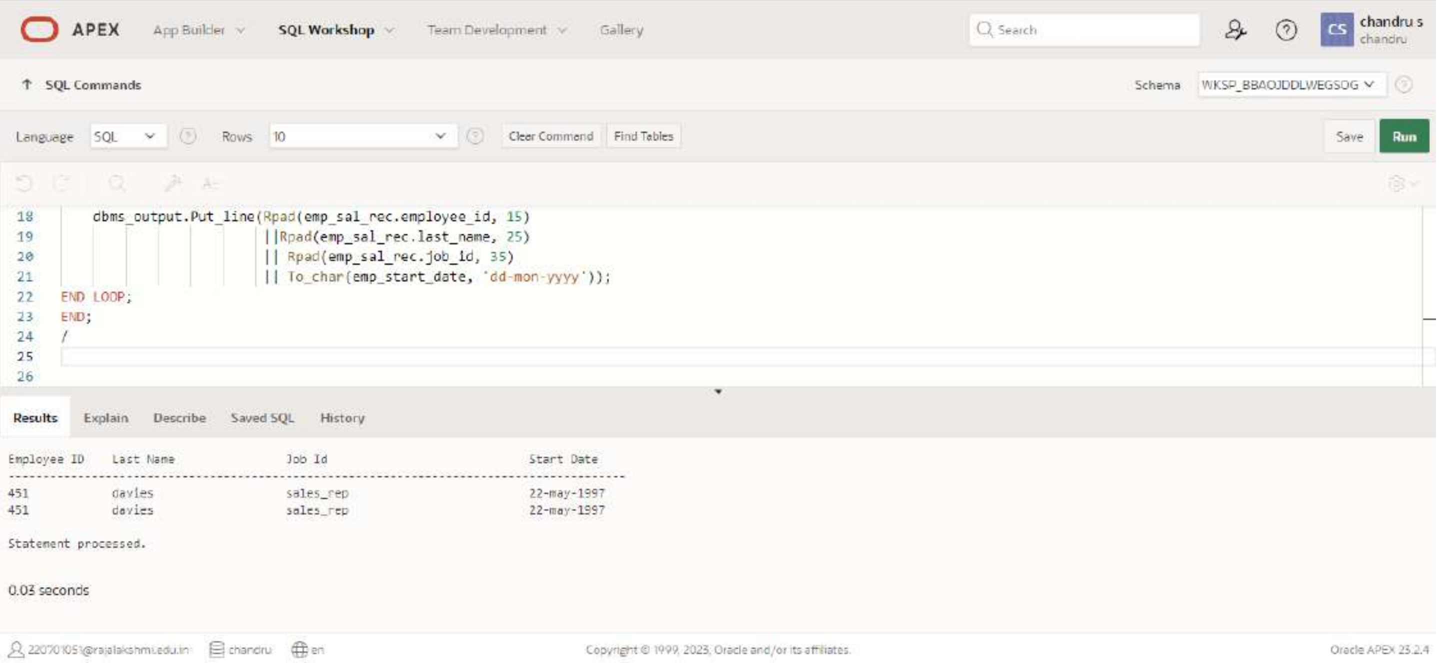
```
Job ID: ac_account
Minimum Salary: 8000
-------------------------
Job ID: ac_account
Minimum Salary: 8000
-------------------------
Job ID: ac_account
Minimum Salary: 46000
-------------------------
Job ID: sales_rep
Minimum Salary: 8000
```

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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
    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;
/
```

```
Employee ID    Last Name                Job Id                             Start Date
-----------------------------------------------------------------------------------------
451            davies                   sales_rep                          22-may-1997
451            davies                   sales_rep                          22-may-1997

Statement processed.
```

0.03 seconds

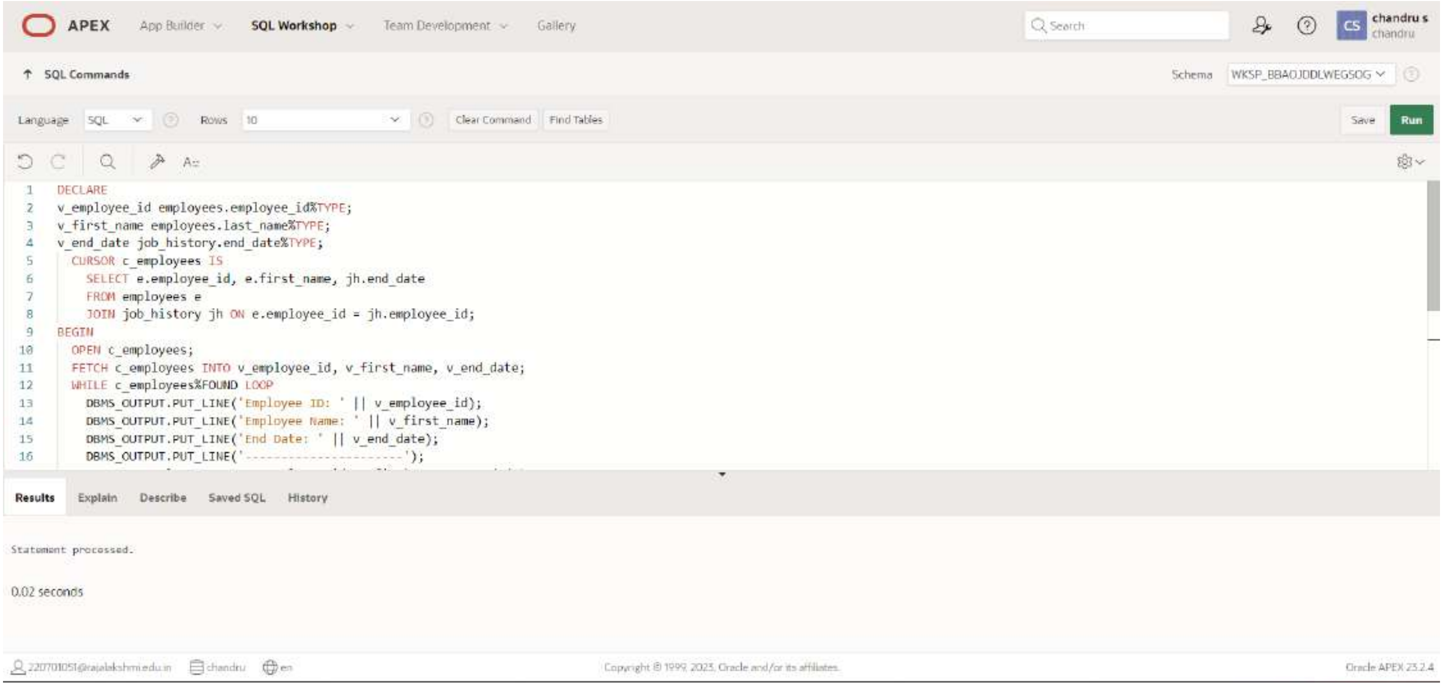
220701051@rajalakshmi.edu.in   chandru   en   Copyright © 1999, 2023, Oracle and/or its affiliates.   Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL Rows 10

```
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('----------------------');
```

```
Statement processed.

0.02 seconds
```

220701051@rajalakshmi.edu.in chandru en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

RESULT:

PROCEDURES AND FUNCTIONS

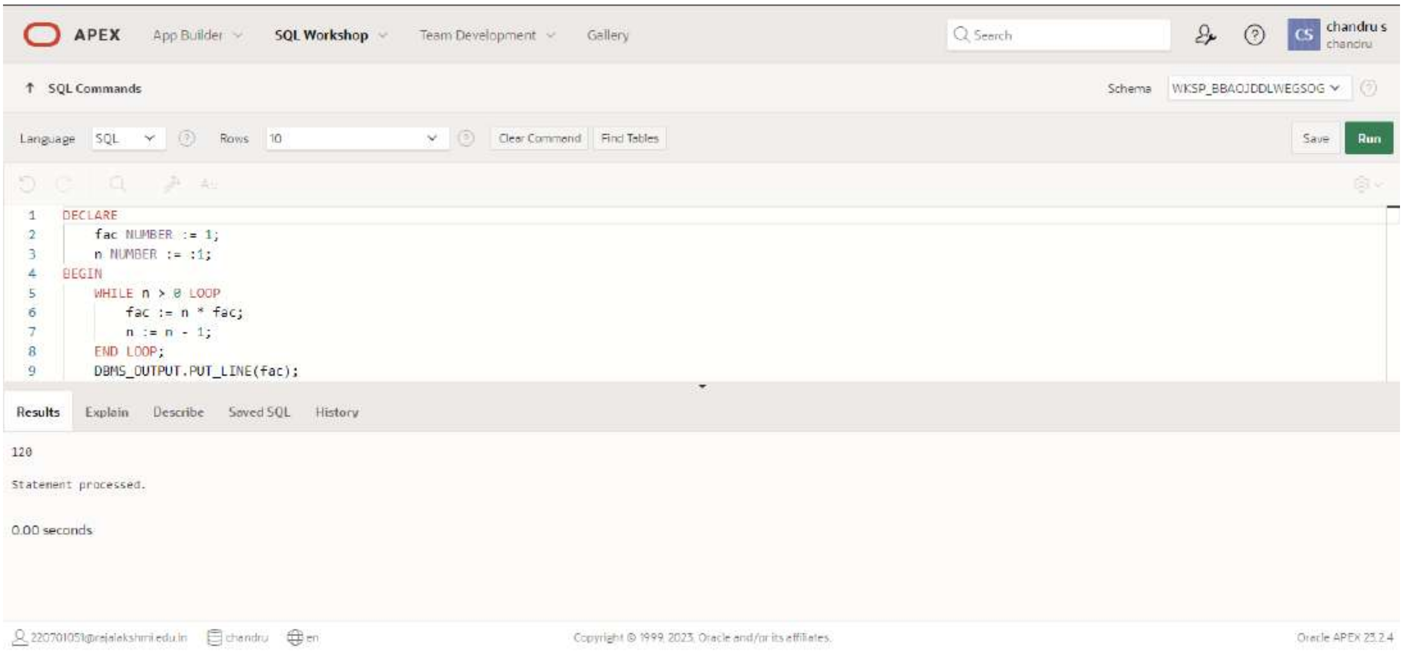
EX_NO: 17 **DATE:**

1.)Factorial of a number using function.

QUERY:

```
DECLARE
    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:



```
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);
```

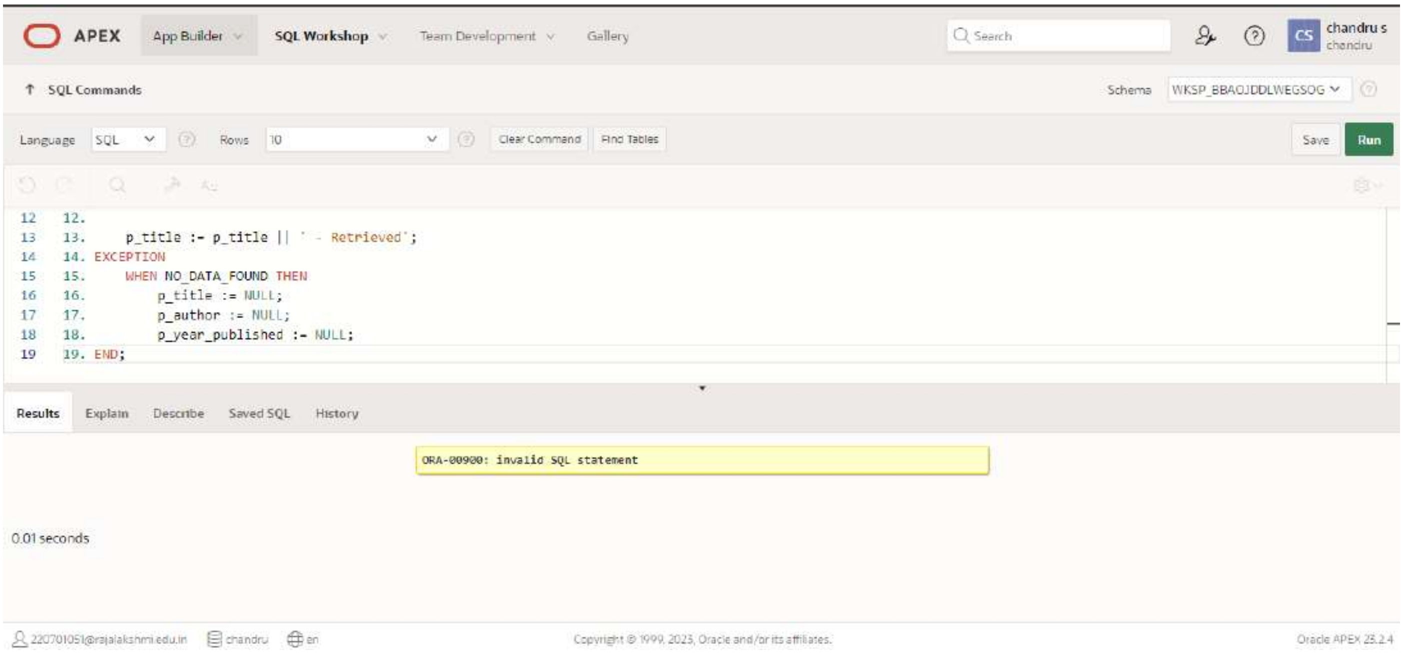
```
120

Statement processed.

0.00 seconds
```

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

QUERY:

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

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


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

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

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
12.
13.     p_title := p_title || ' - Retrieved';
14. EXCEPTION
15.     WHEN NO_DATA_FOUND THEN
16.         p_title := NULL;
17.         p_author := NULL;
18.         p_year_published := NULL;
19. END;
```

```
ORA-00900: invalid SQL statement
```

0.01 seconds

220701051@rajalakshmi.edu.in chandru en

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

RESULT:

TRIGGER

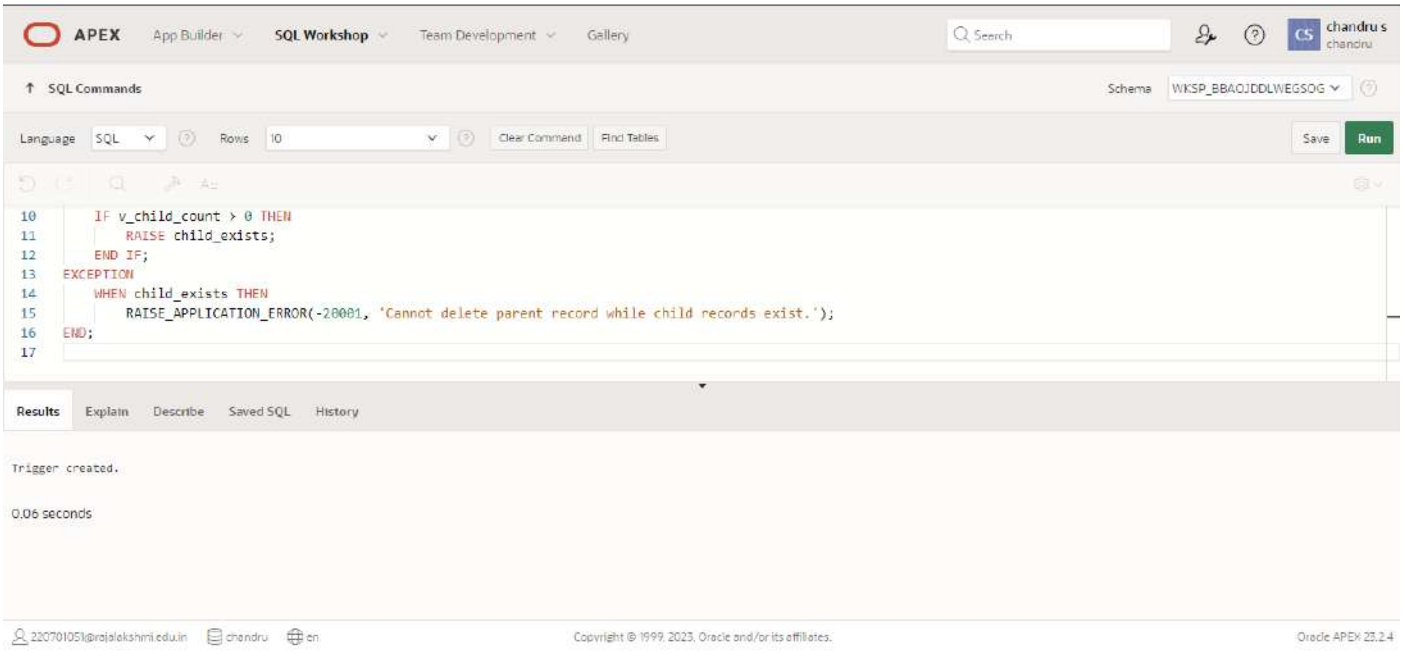
EX_NO: 18 **DATE:**

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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
    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;
```

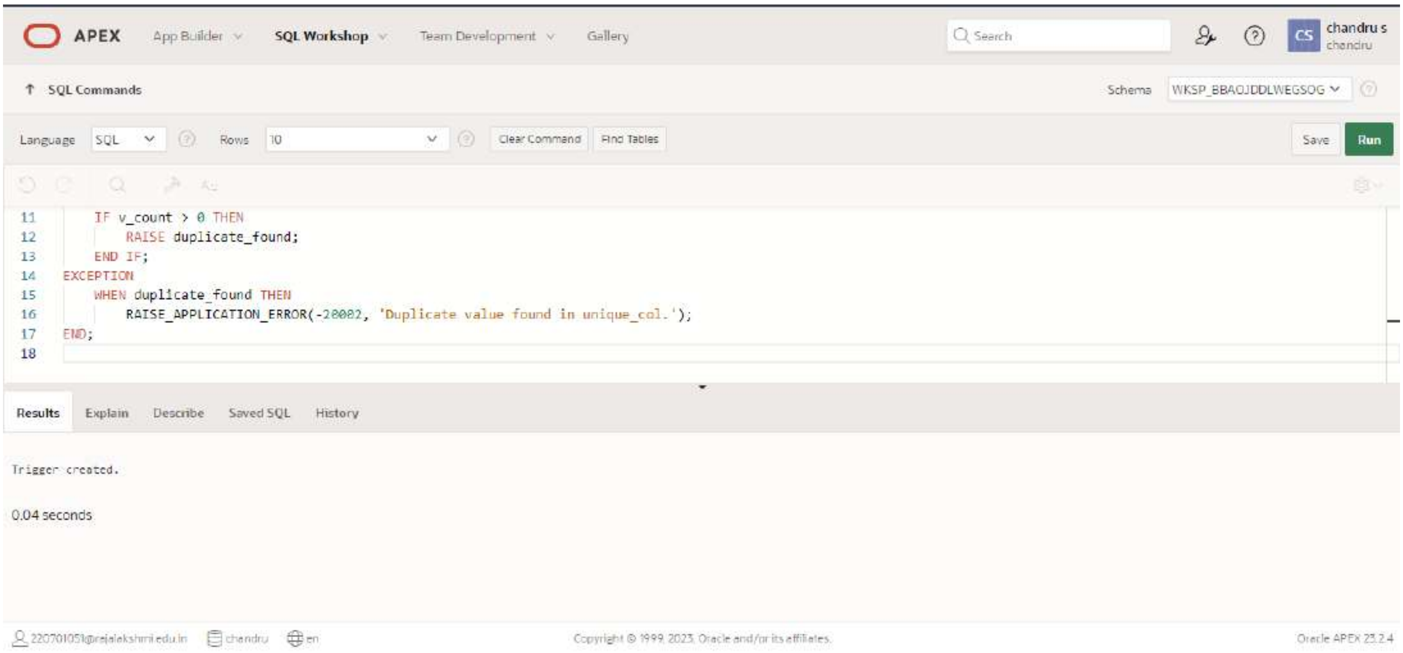
```
Trigger created.

0.06 seconds
```

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

QUERY:

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

OUTPUT:



```
    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;
```

```
Trigger created.

0.04 seconds
```

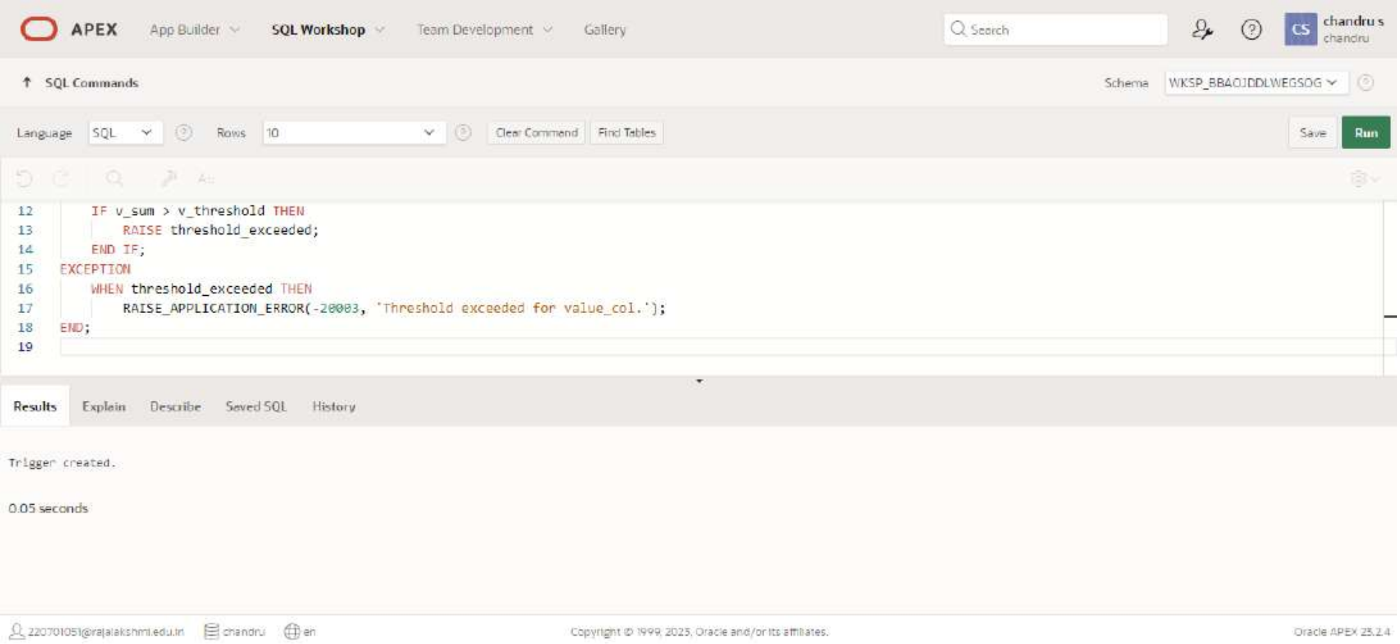
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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:



Schema: WKSP_BBAOJDDLWEGSOG

Language: SQL

Rows: 10

```
    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;
```

```
Trigger created.
```

0.05 seconds

220701051@rajalakshmi.edu.in chandru en

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

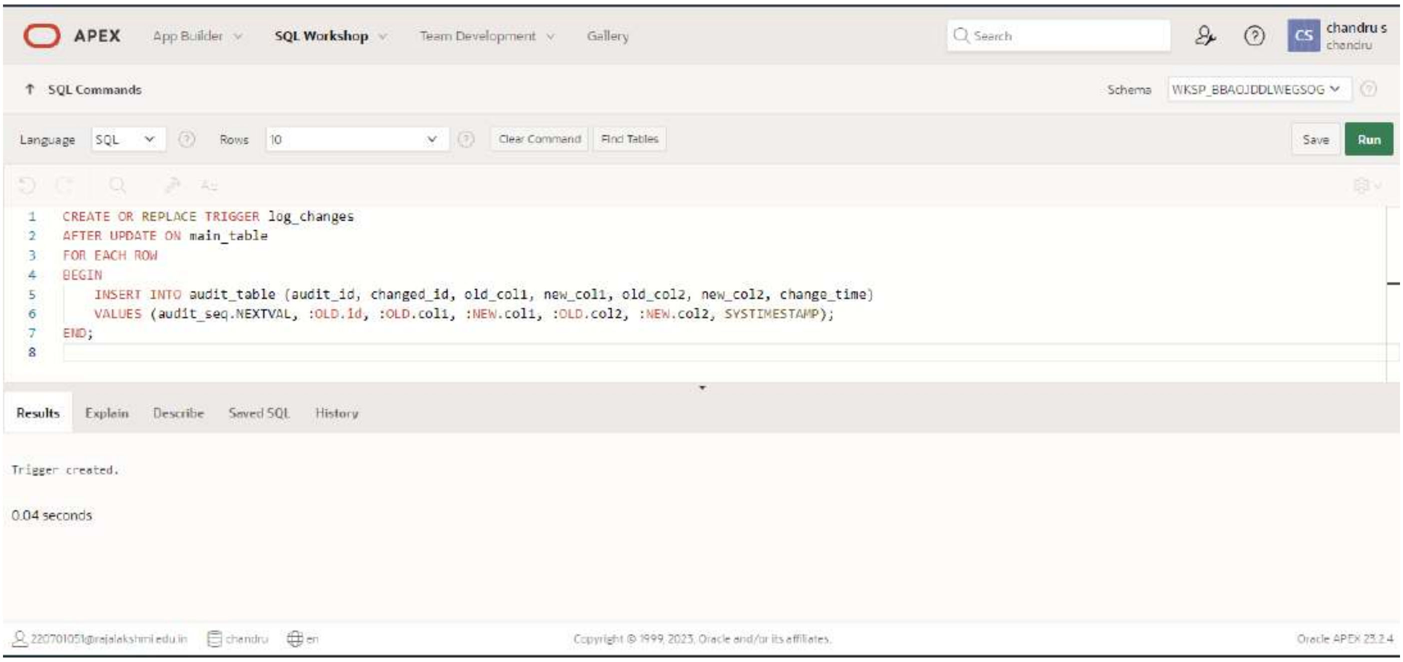
Oracle APEX 23.2.4

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:



```
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;
```

```
Trigger created.

0.04 seconds
```

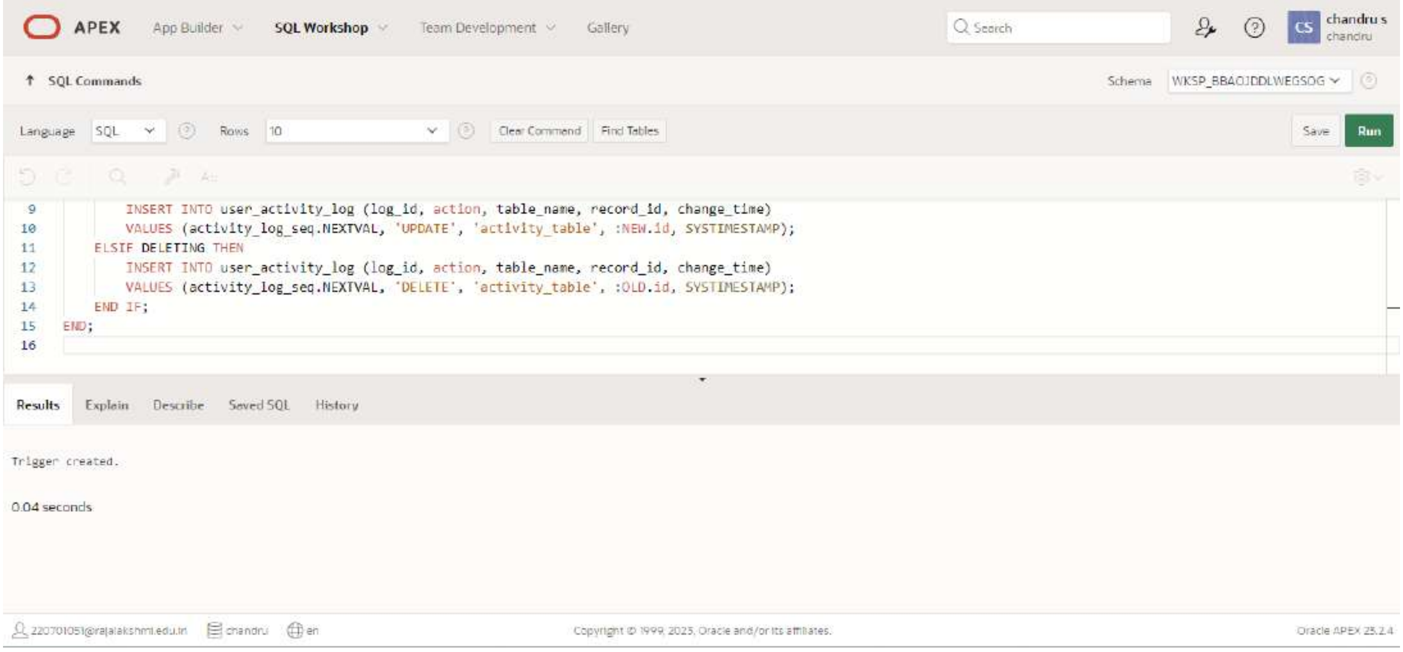
220701051@rajalakshmi.edu.in
chandru
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.4

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:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
        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;
```

```
Trigger created.
```

0.04 seconds

220701051@rajalakshmi.edu.in chandru en

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

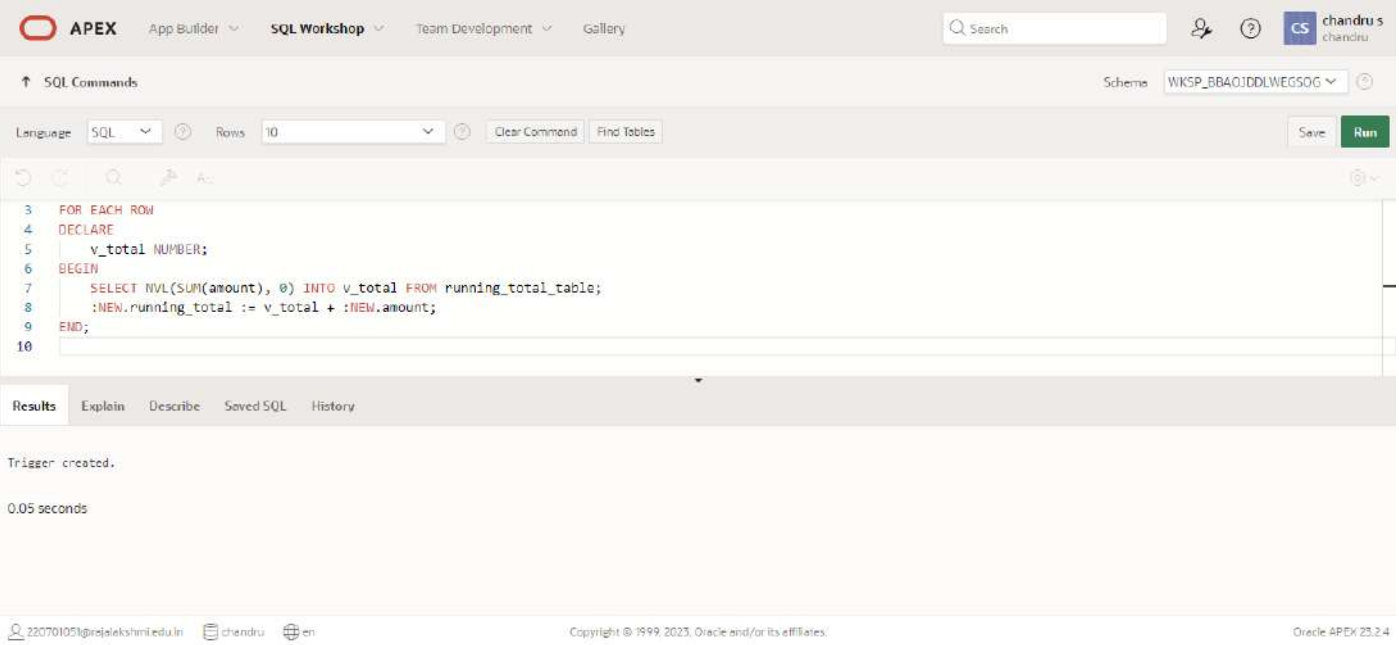
Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



Schema WKSP_BBAOJDDLWEGSOG

Language SQL

Rows 10

```
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;
```

```
Trigger created.

0.05 seconds
```

220701051@rajalakshmi.edu.in chandru en

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

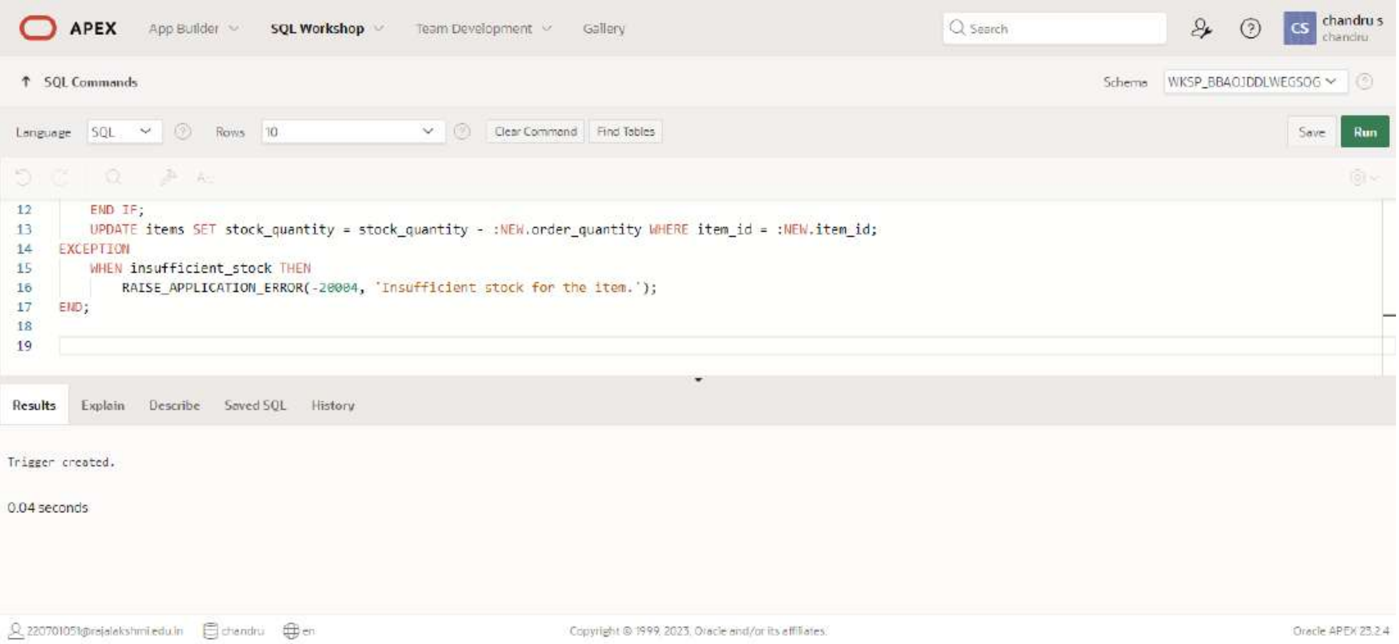
Oracle APEX 23.2.4

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:



```
    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;
```

```
Trigger created.

0.04 seconds
```

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

RESULT:

MONGO DB

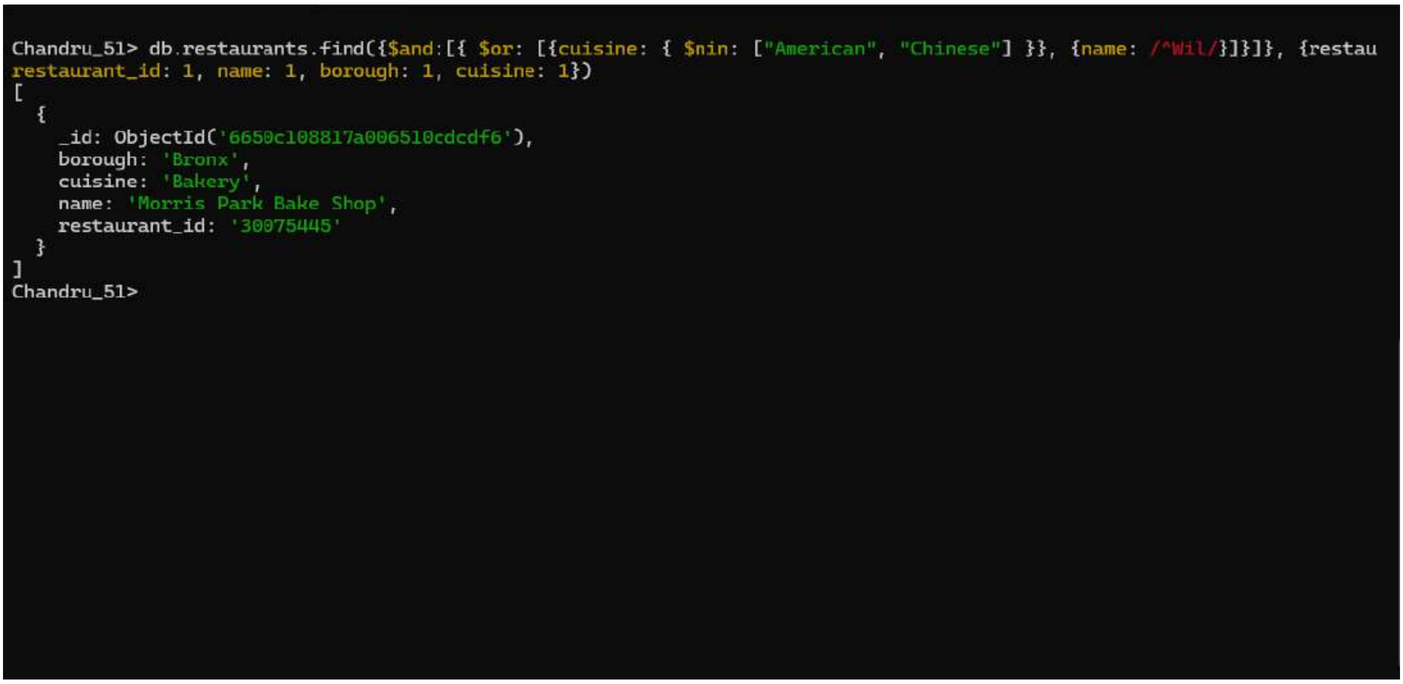
EX_NO: 19 **DATE:**

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:



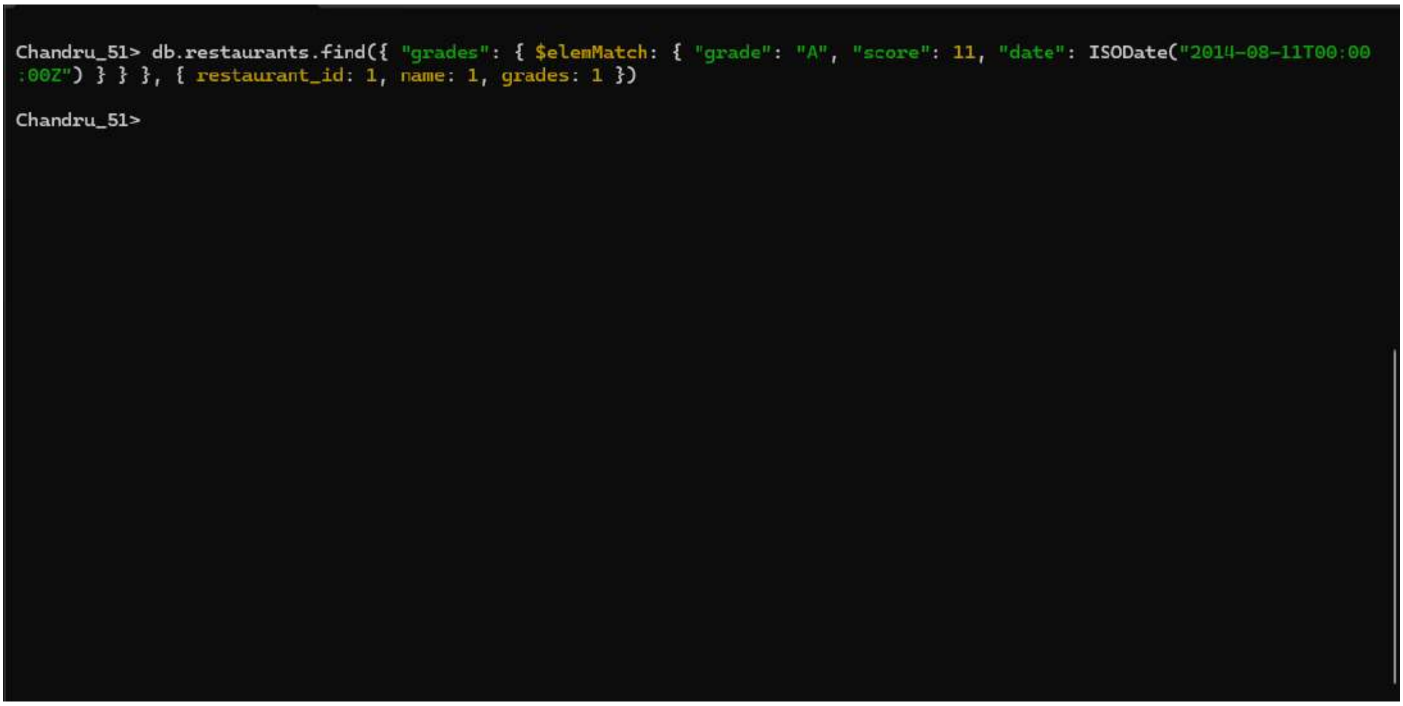
```
Chandru_51> db.restaurants.find({$and:[{ $or: [{cuisine: { $nin: ["American", "Chinese"] }}, {name: /^Wil/}]}]}, {restau
restaurant_id: 1, name: 1, borough: 1, cuisine: 1})
[
  {
    _id: ObjectId('6650c108817a006510cdcdf6'),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51>
```

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:



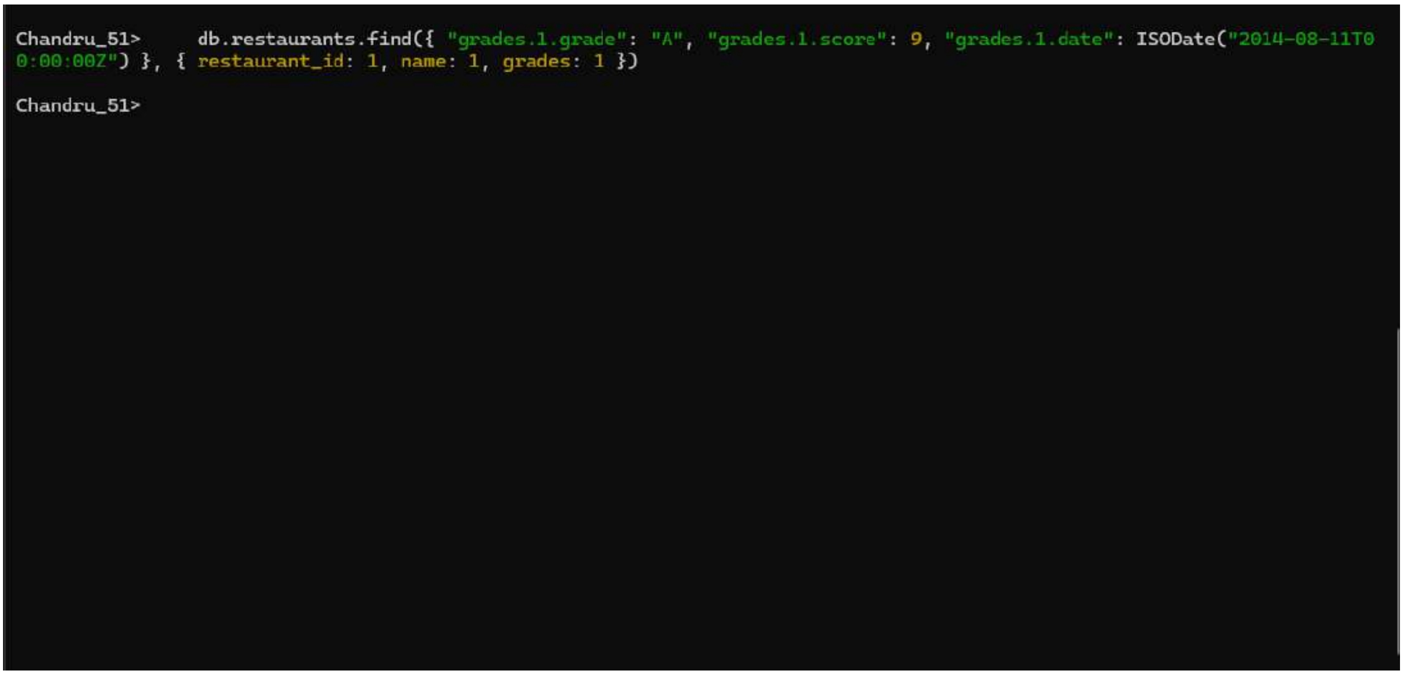
```
Chandru_51> db.restaurants.find({ "grades": { $elemMatch: { "grade": "A", "score": 11, "date": ISODate("2014-08-11T00:00
:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 })

Chandru_51>
```

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("2014-08-1T00:00:00Z") },{ restaurant_id: 1, name: 1, grades: 1 });

OUTPUT:



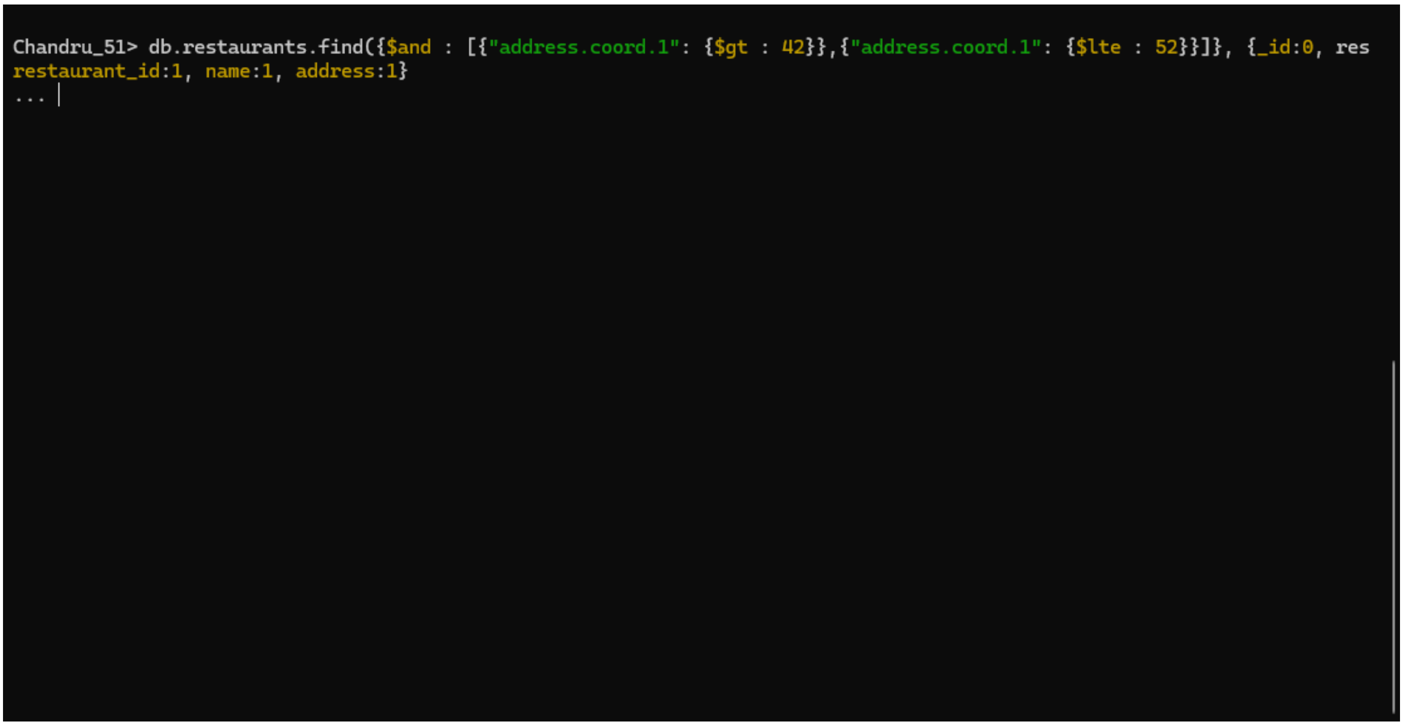
```
Chandru_51>     db.restaurants.find({ "grades.1.grade": "A", "grades.1.score": 9, "grades.1.date": ISODate("2014-08-11T0
0:00:00Z") }, { restaurant_id: 1, name: 1, grades: 1 })

Chandru_51>
```

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:



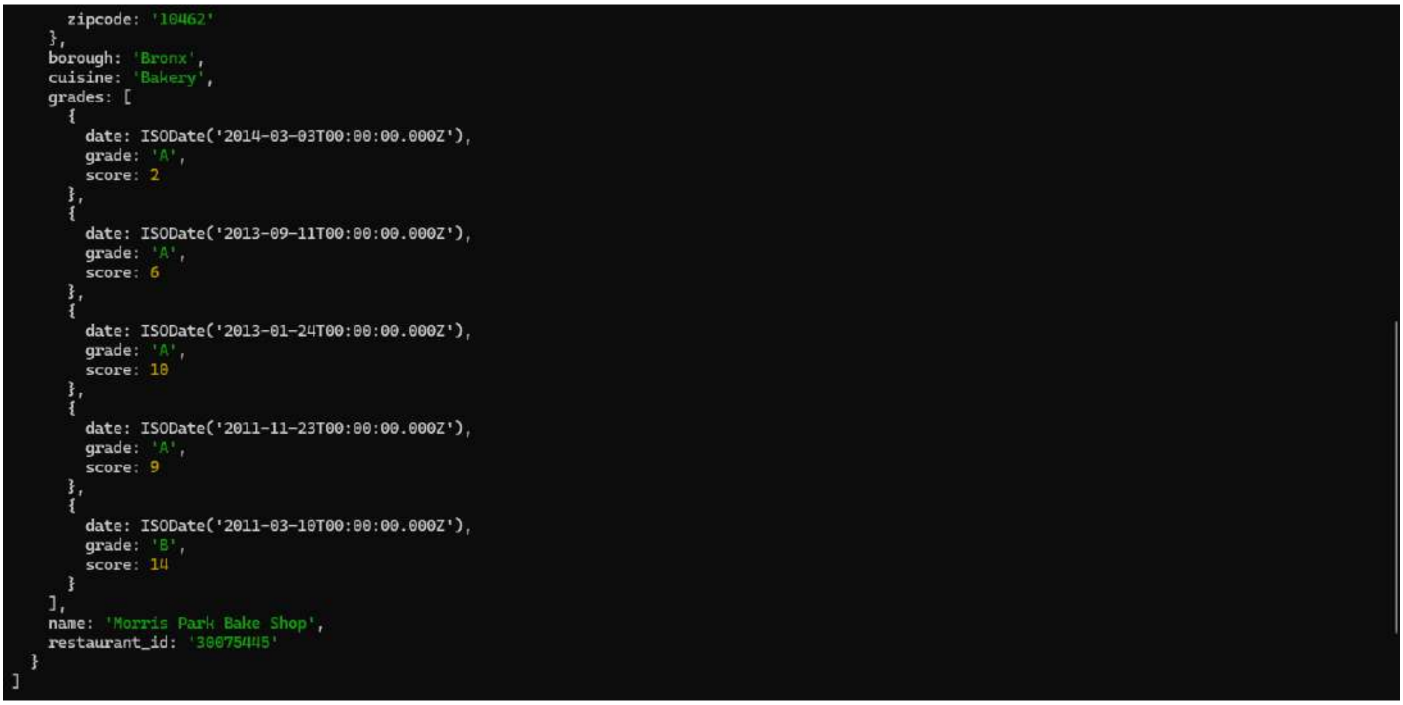
```
Chandru_51> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}},{"address.coord.1": {$lte : 52}}]}, {_id:0, res
restaurant_id:1, name:1, address:1}
...
```

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:



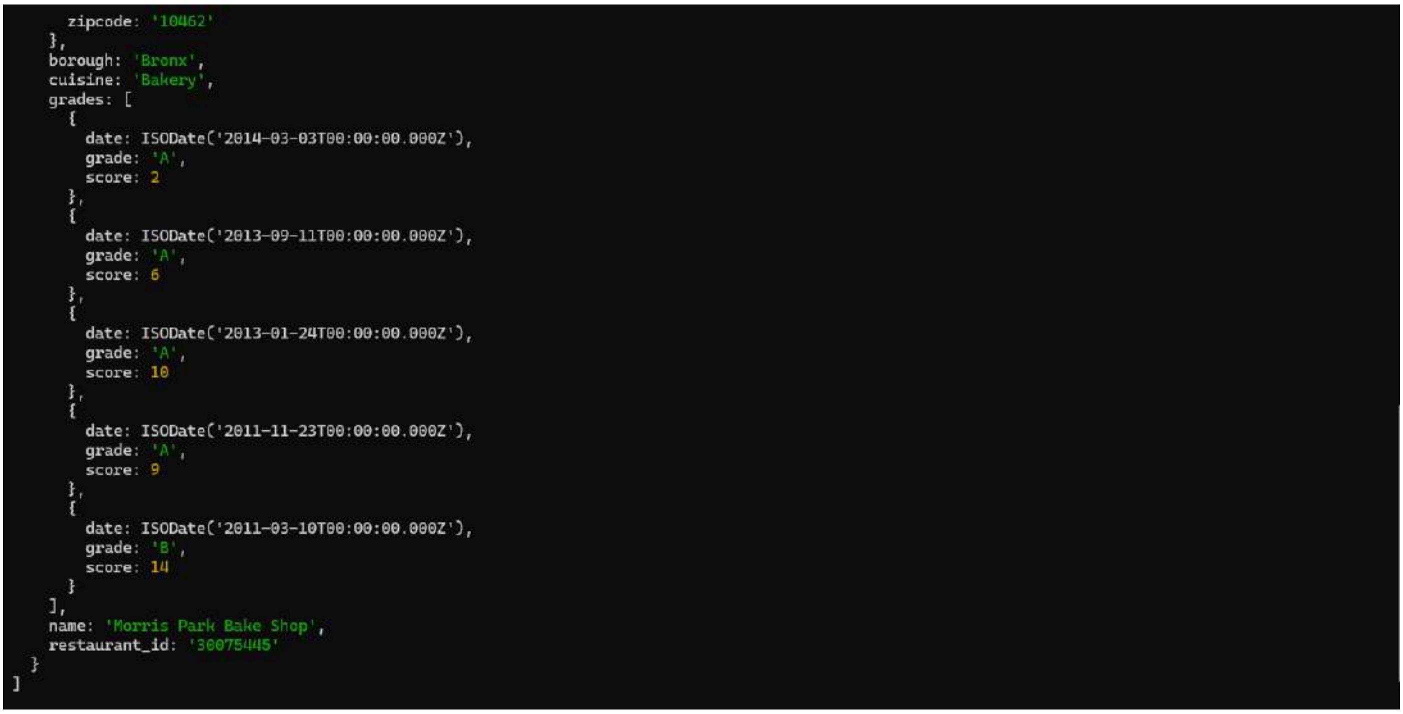
```
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },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

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:



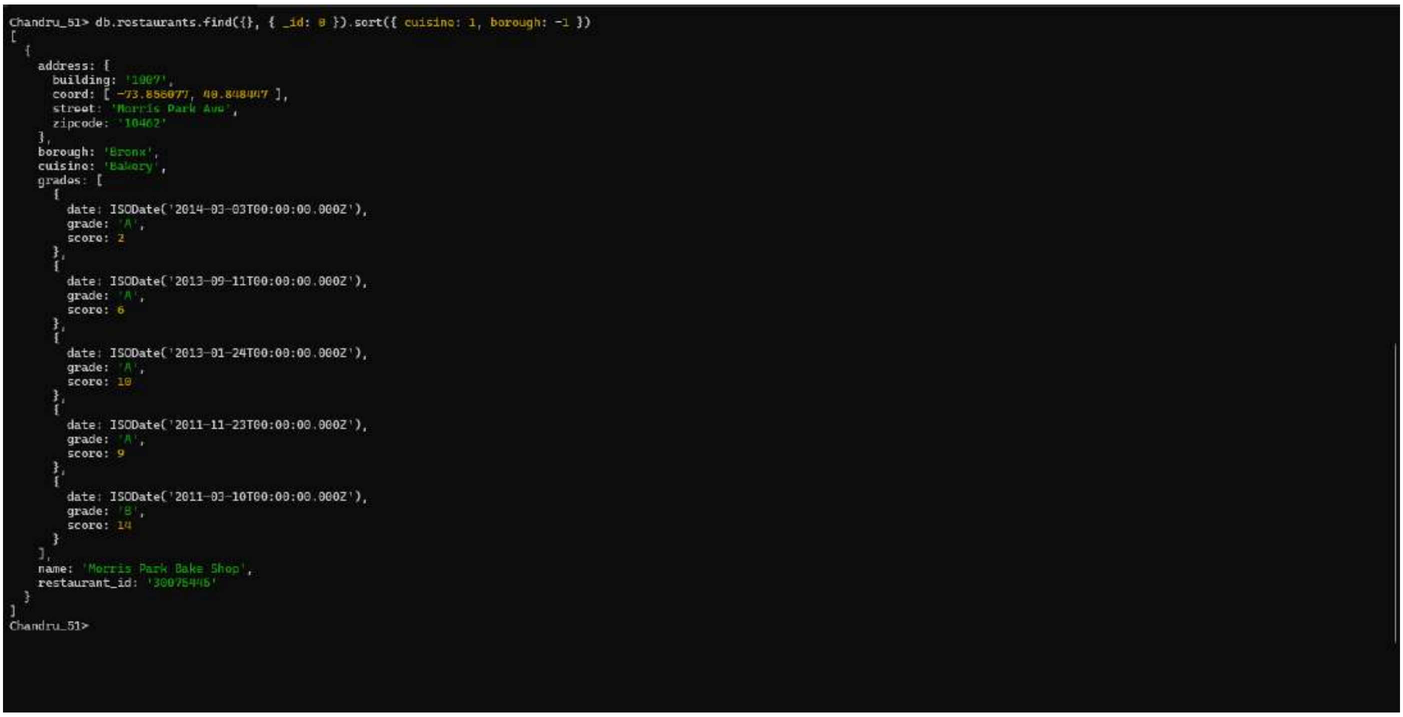
```
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },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

7.) Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.

QUERY:

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

OUTPUT:



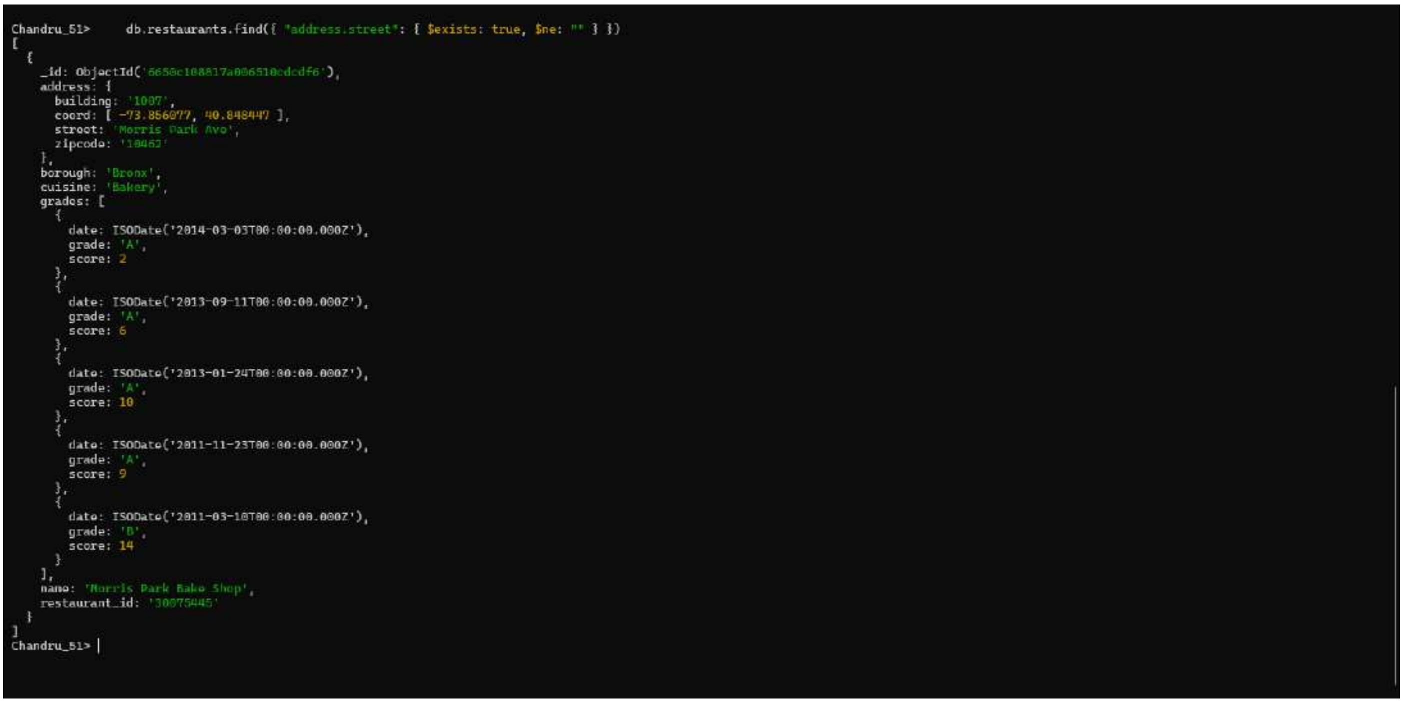
```
Chandru_51> db.restaurants.find({}, { _id: 0 }).sort({ cuisine: 1, borough: -1 })
[
  {
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51>
```

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:



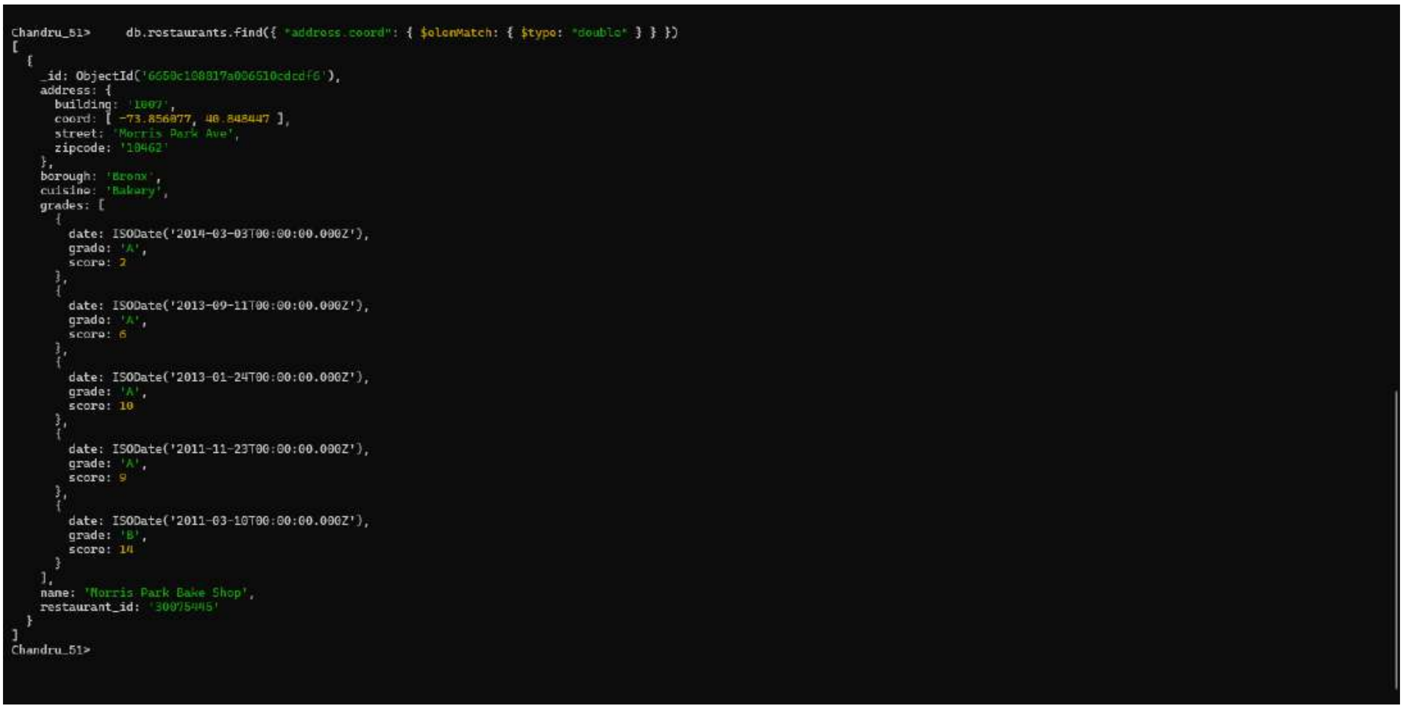
```
Chandru_51>    db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })
[
  {
    _id: ObjectId('6658c108817a006510cdcdf6'),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51> |
```

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:



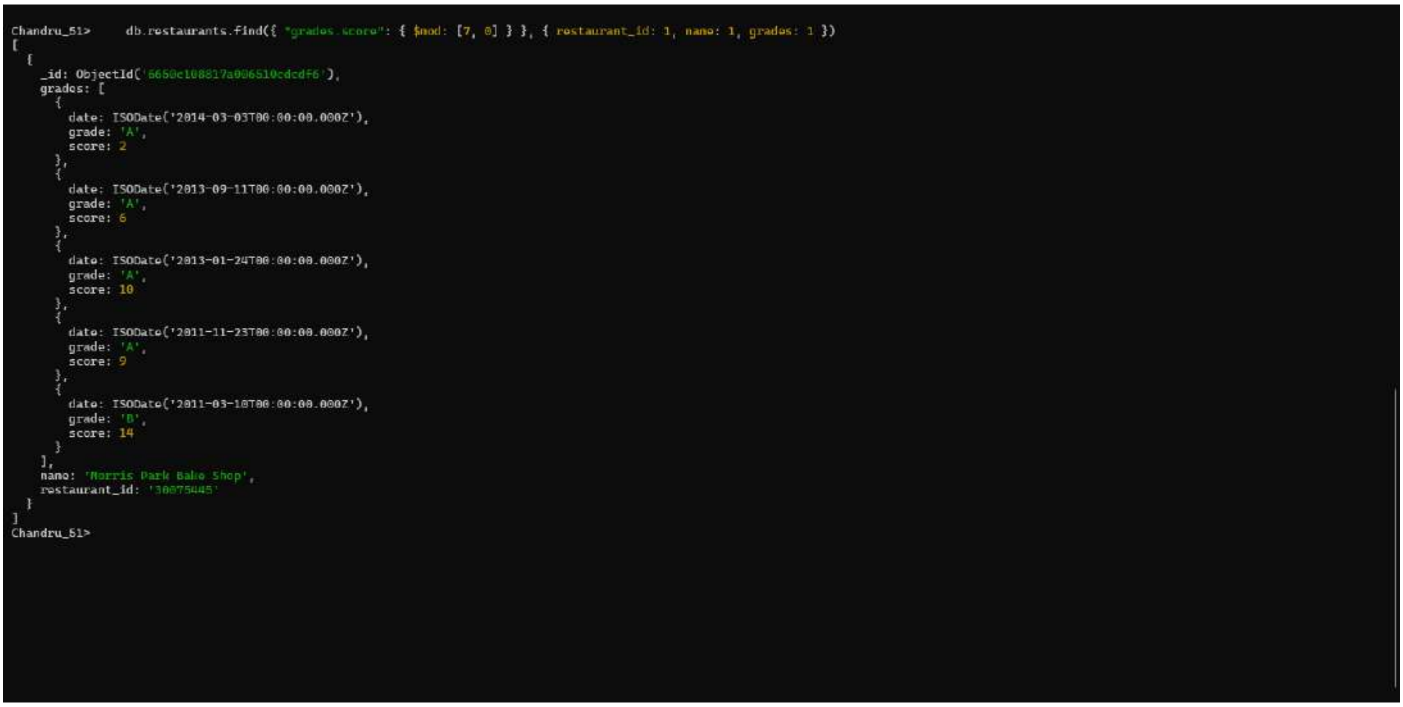
```
Chandru_51>     db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
[
  {
    _id: ObjectId('6650c108817a006510cdcdf6'),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51>
```

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:



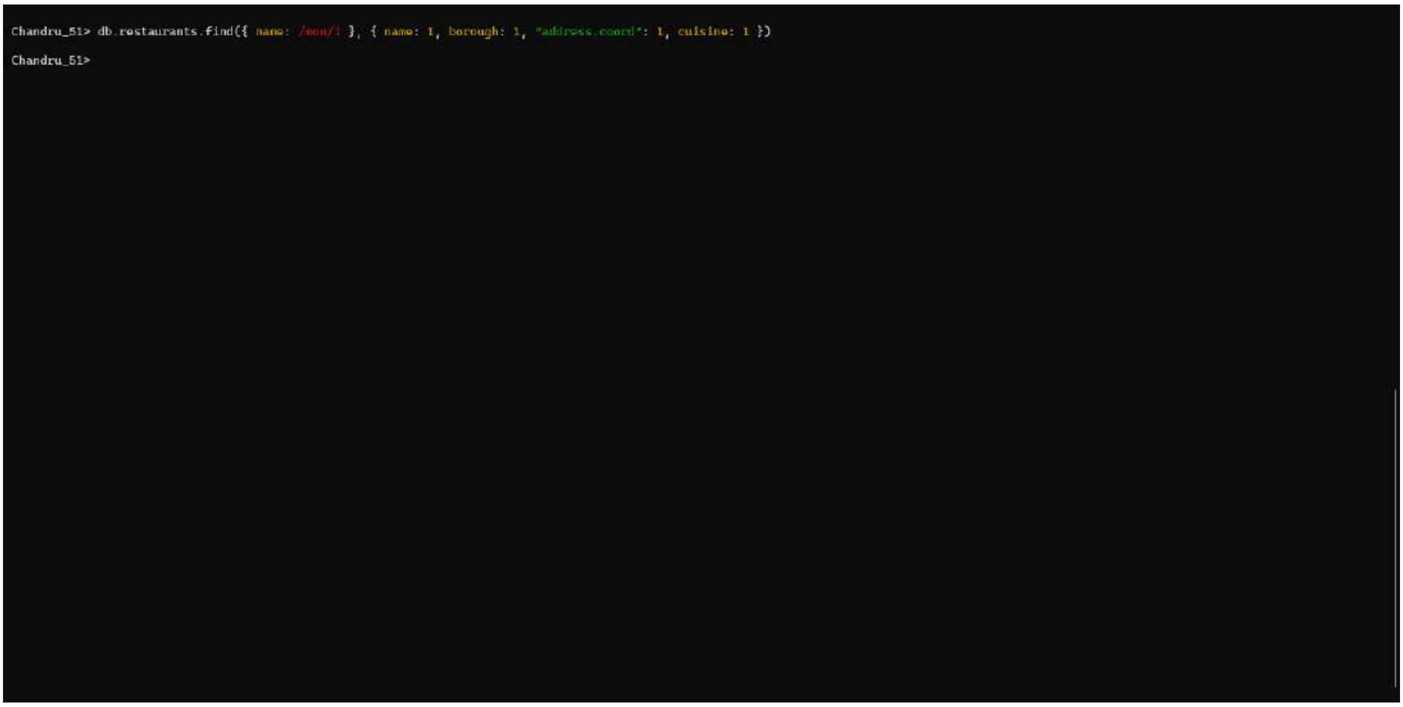
```
Chandru_51>     db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 })
[
  {
    _id: ObjectId('6650c108817a006510cdcdf6'),
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51>
```

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:



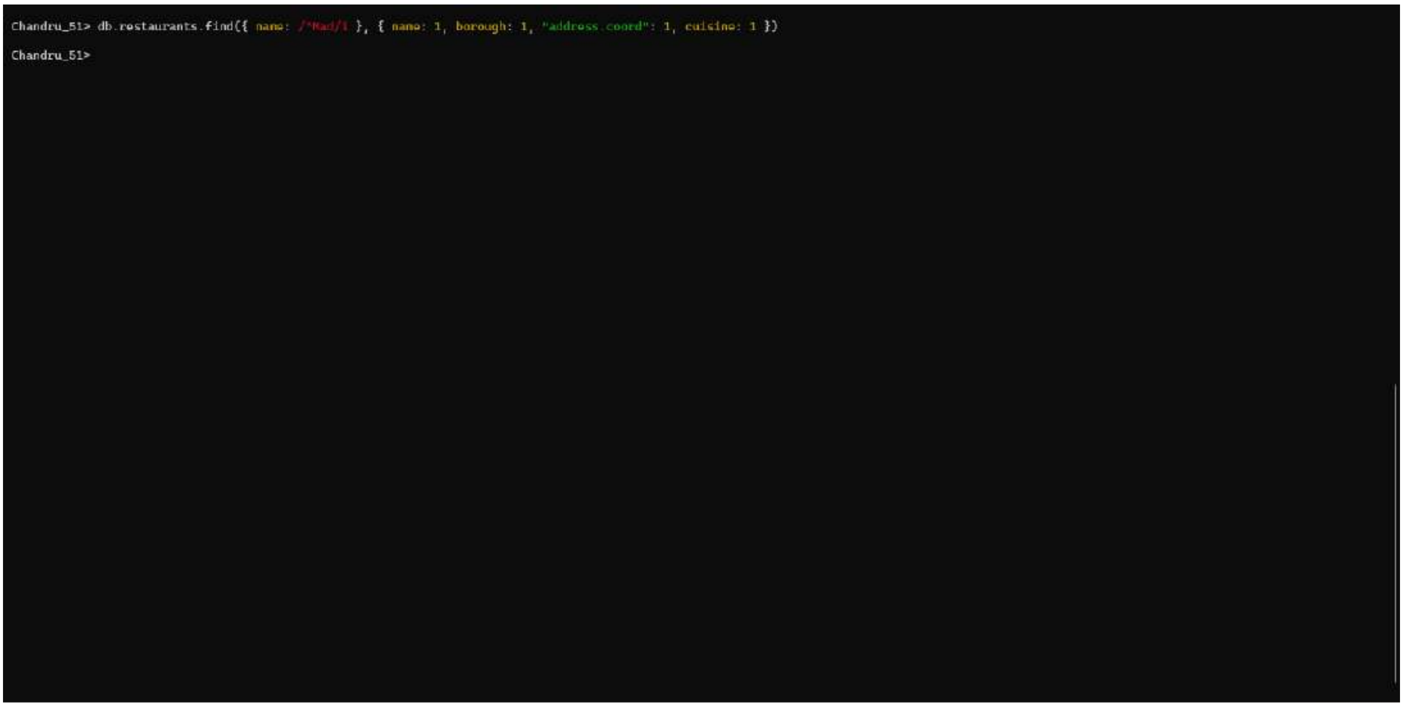
```
Chandru_51> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
Chandru_51>
```

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:

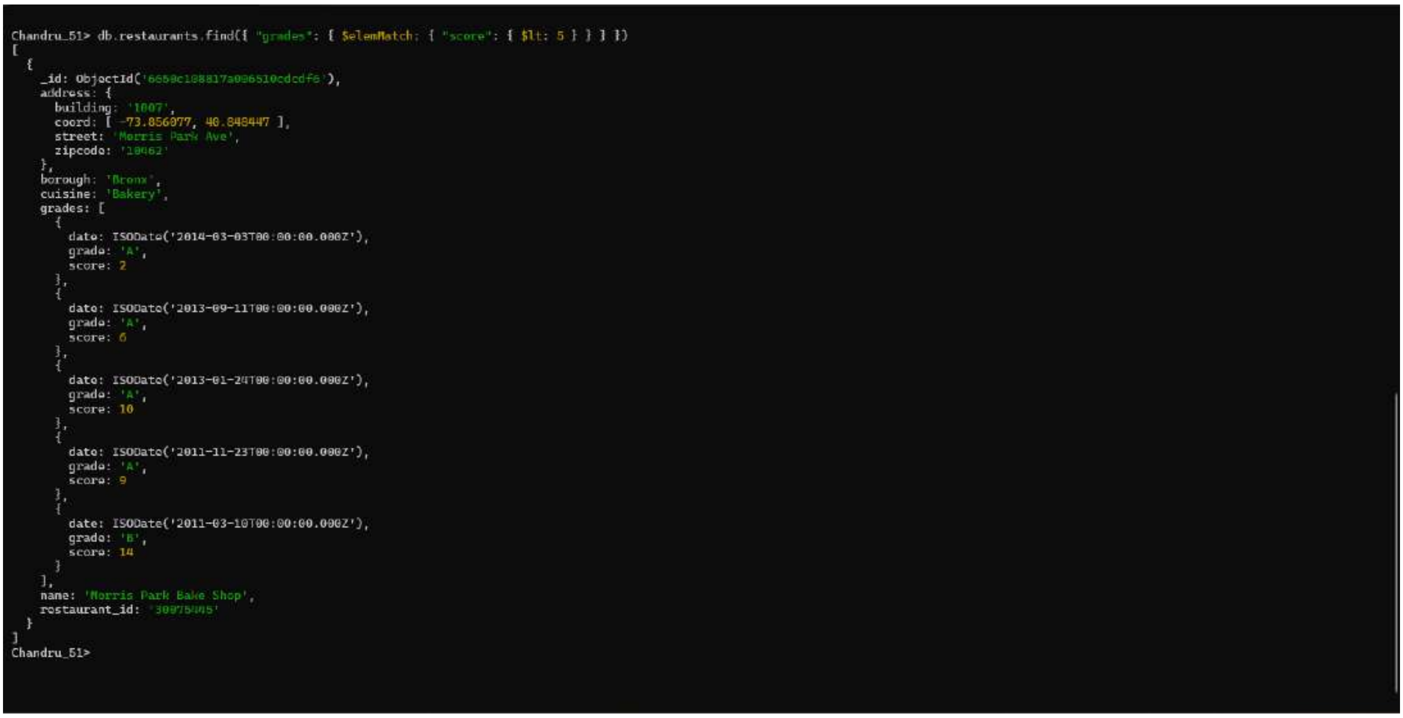


```
Chandru_51> db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
Chandru_51>
```

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:

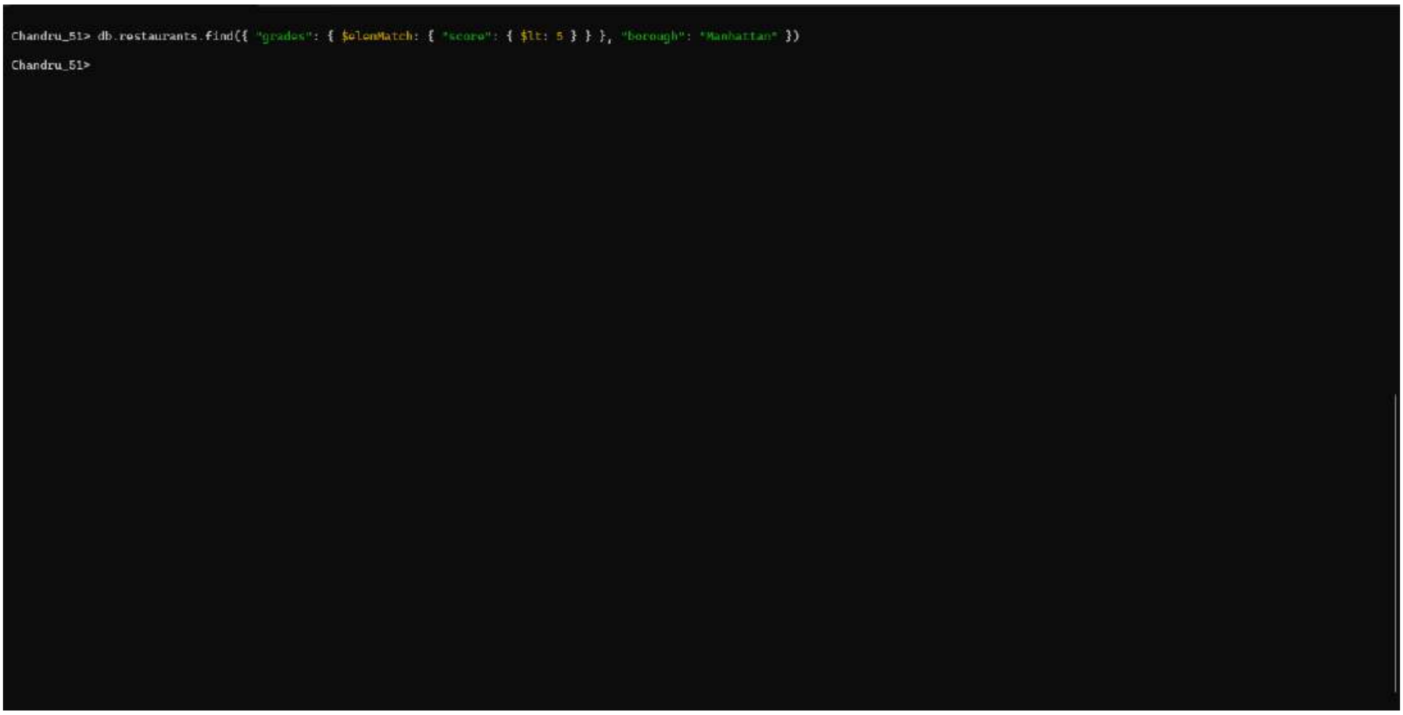


```
Chandru_51> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
[
  {
    _id: ObjectId('6658c108817a096510cdcdf6'),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51>
```

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:

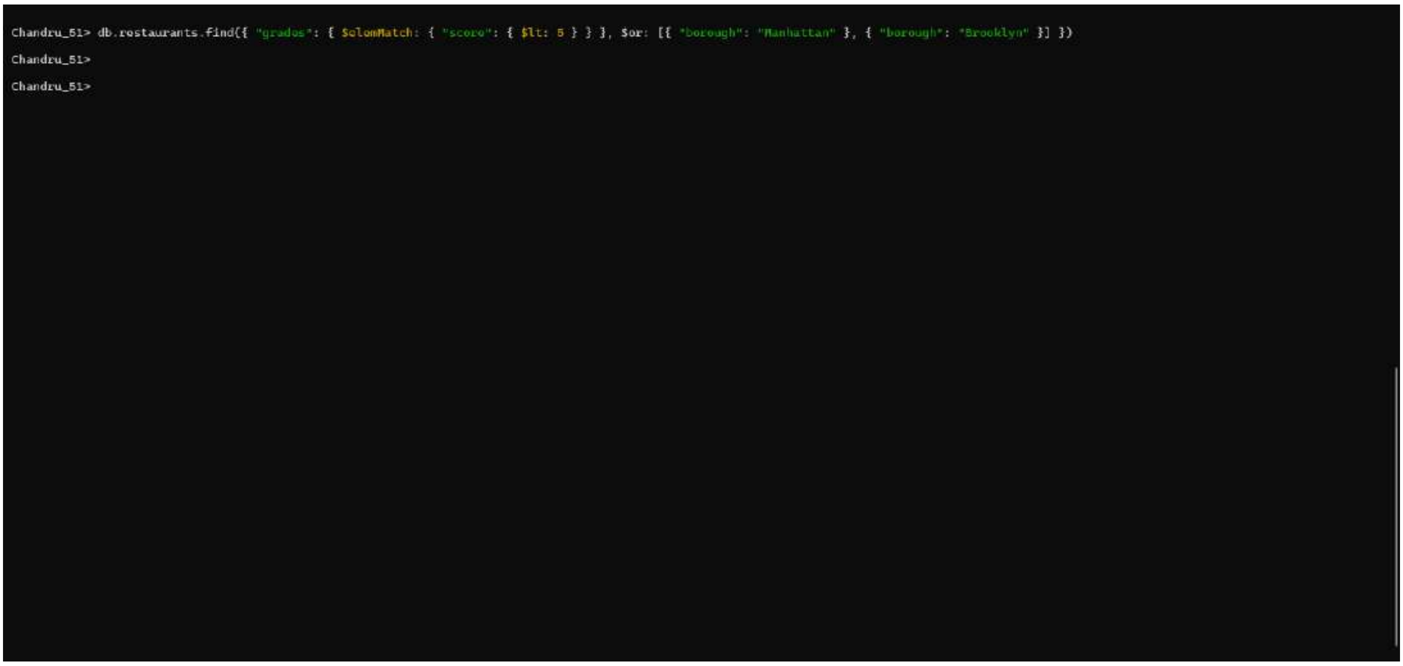


```
Chandru_51> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })

Chandru_51>
```

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:

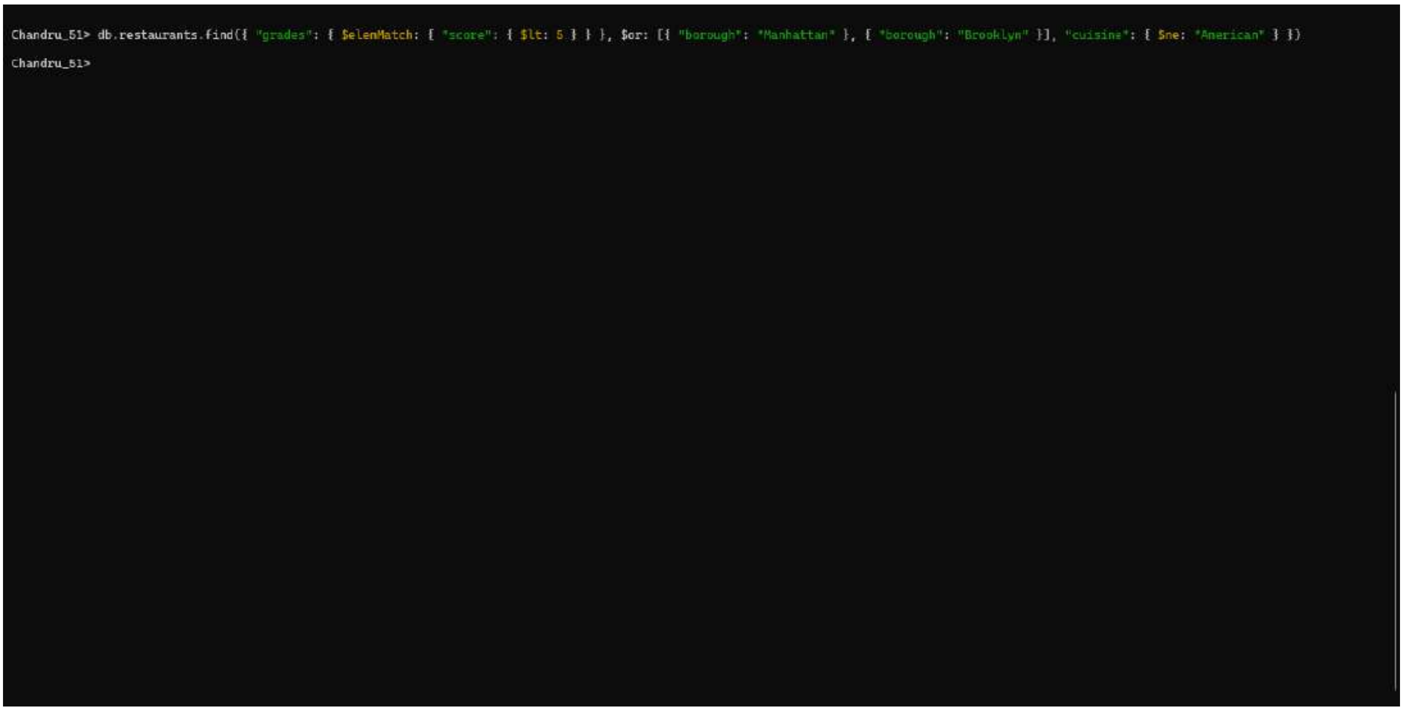


```
Chandru_51> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
Chandru_51>
Chandru_51>
```

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:

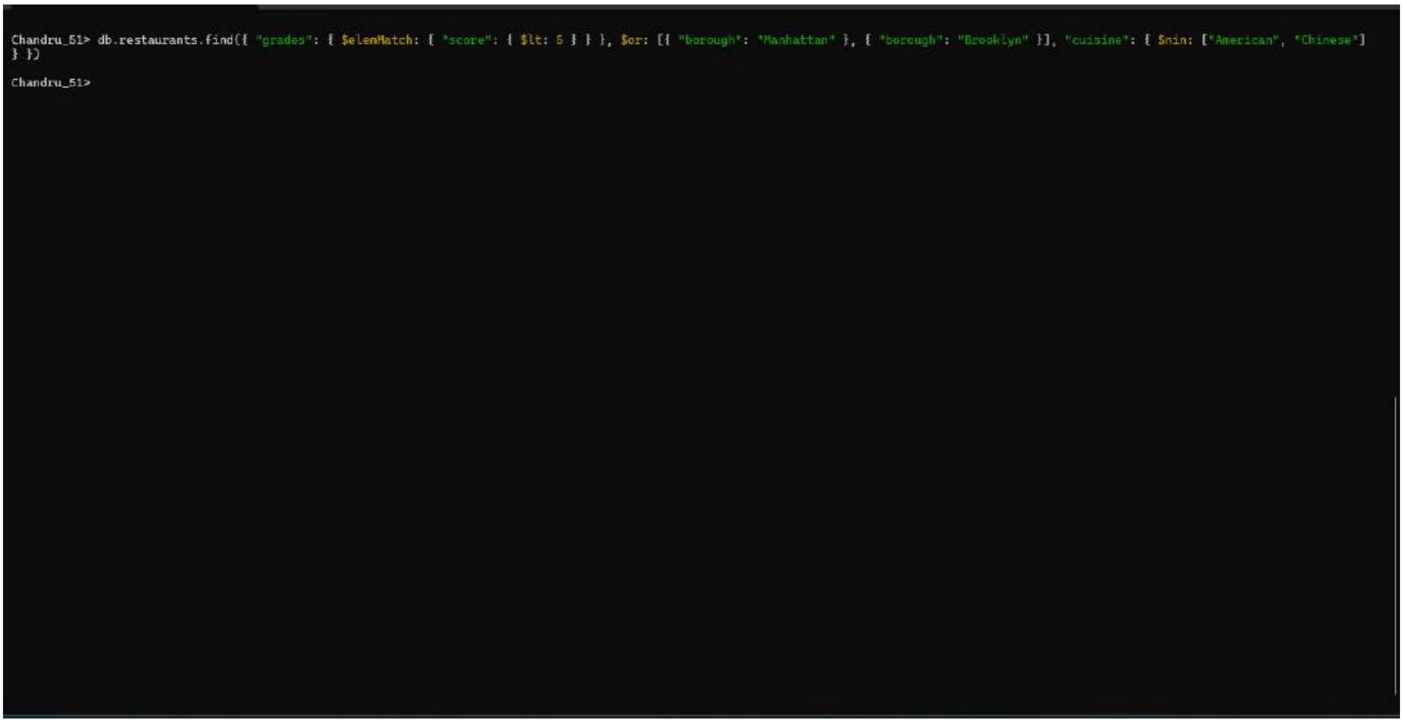


```
Chandru_51> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
Chandru_51>
```

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:

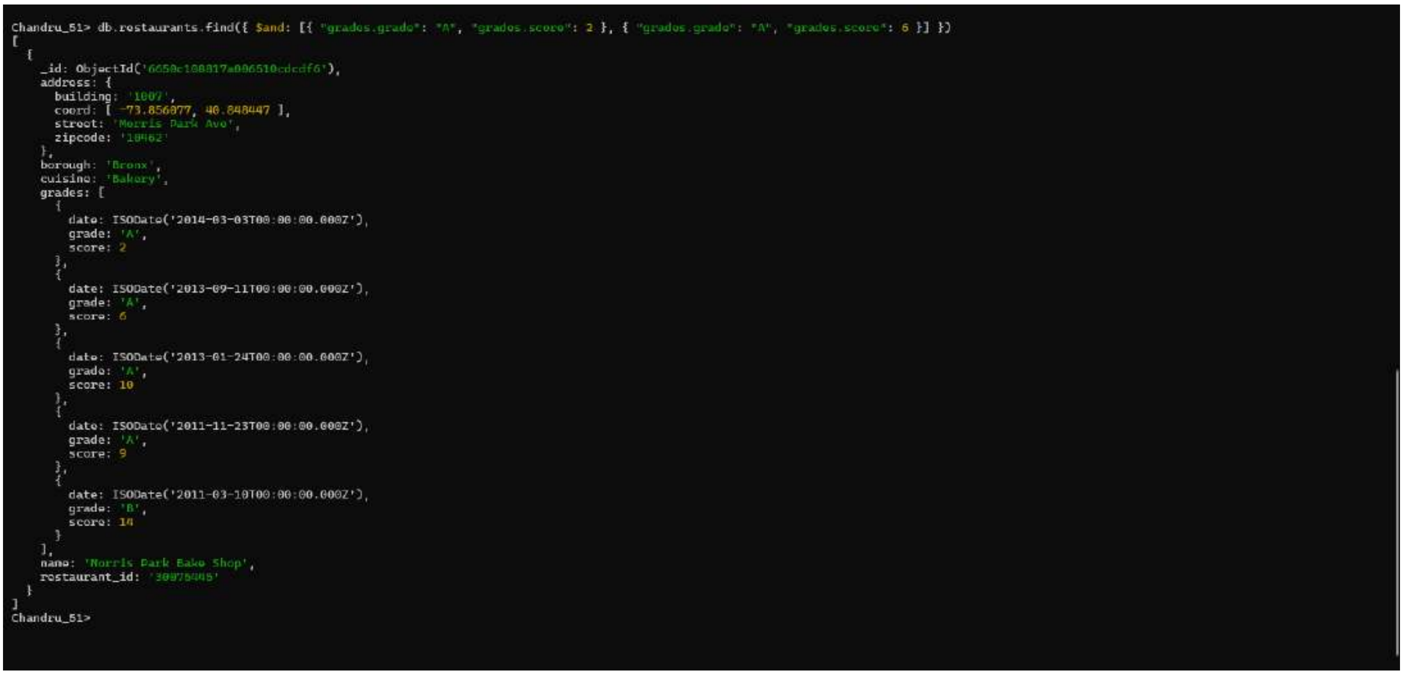


```
Chandru_51> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"]
} })

Chandru_51>
```

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:

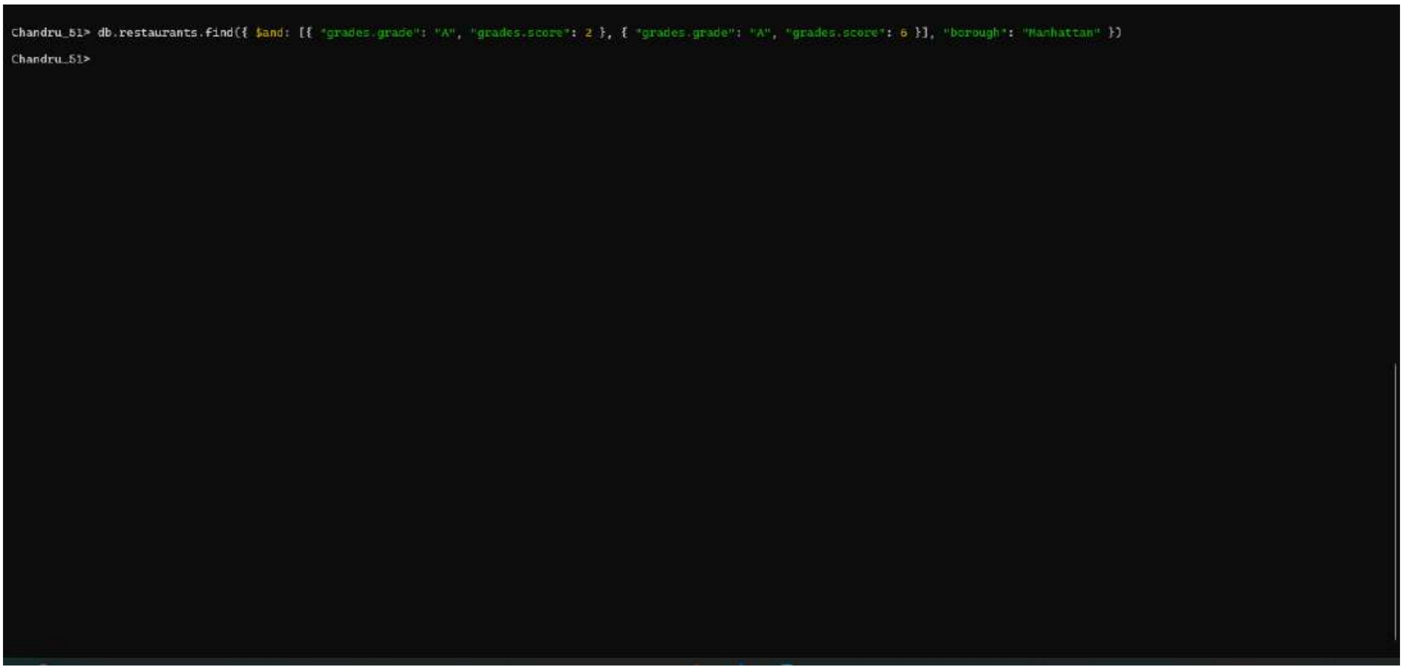


```
Chandru_51> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
[
  {
    _id: ObjectId('6650c108817a006510cdcdf6'),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51>
```

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:

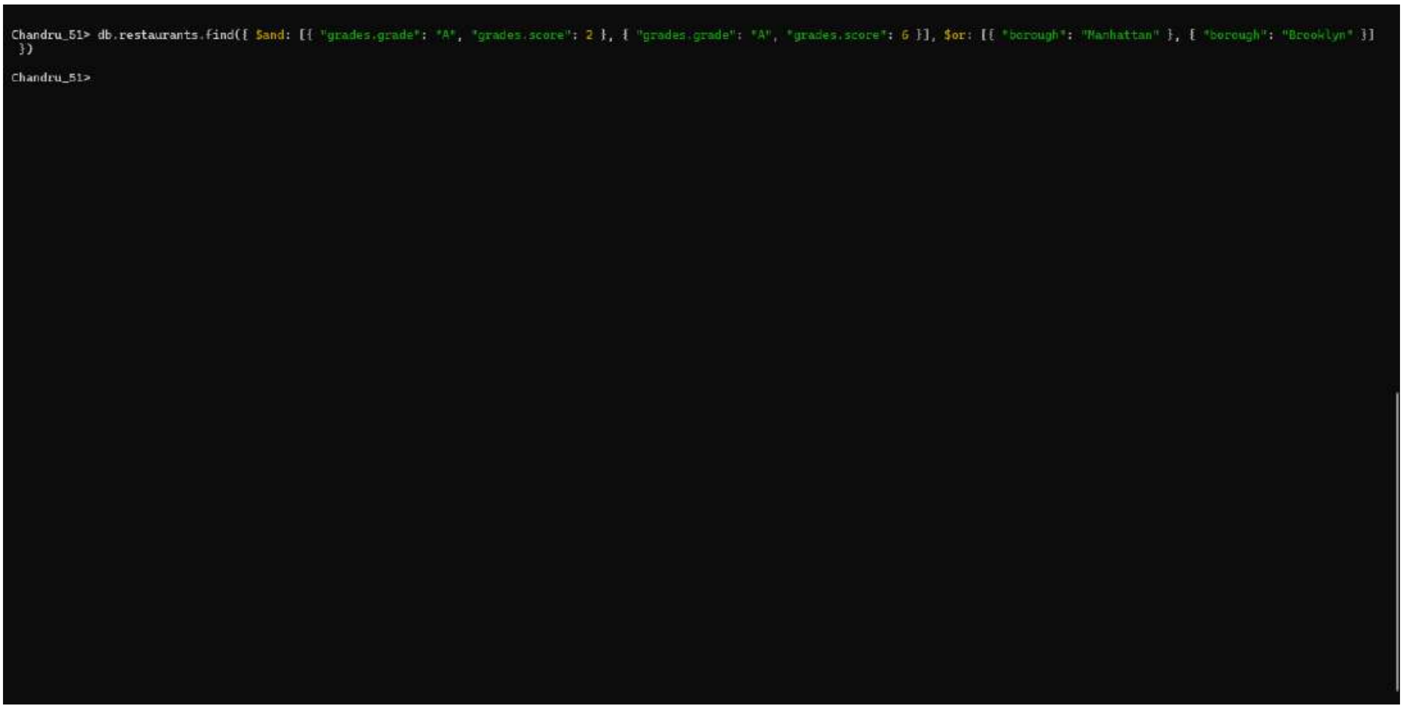


```
Chandru_51> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })
Chandru_51>
```

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:



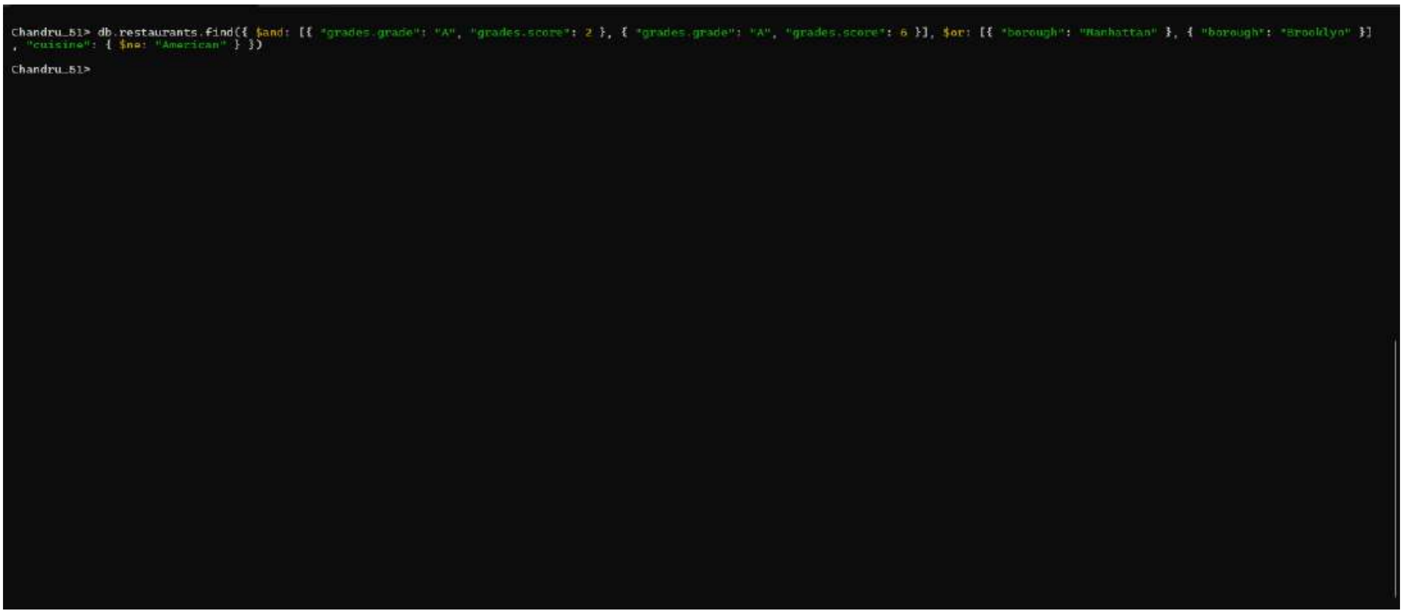
```
Chandru_51> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }]
 })
Chandru_51>
```

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:



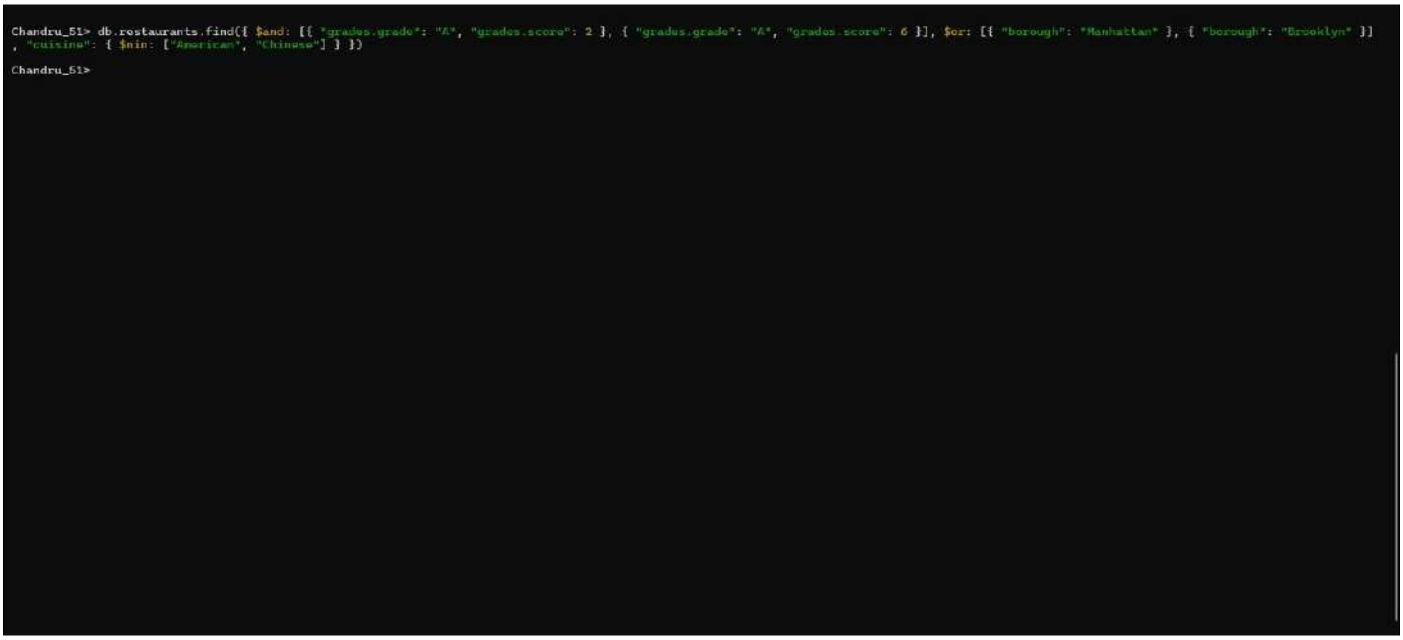
```
Chandru_51> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }]
, "cuisine": { $ne: "American" } })

Chandru_51>
```

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:



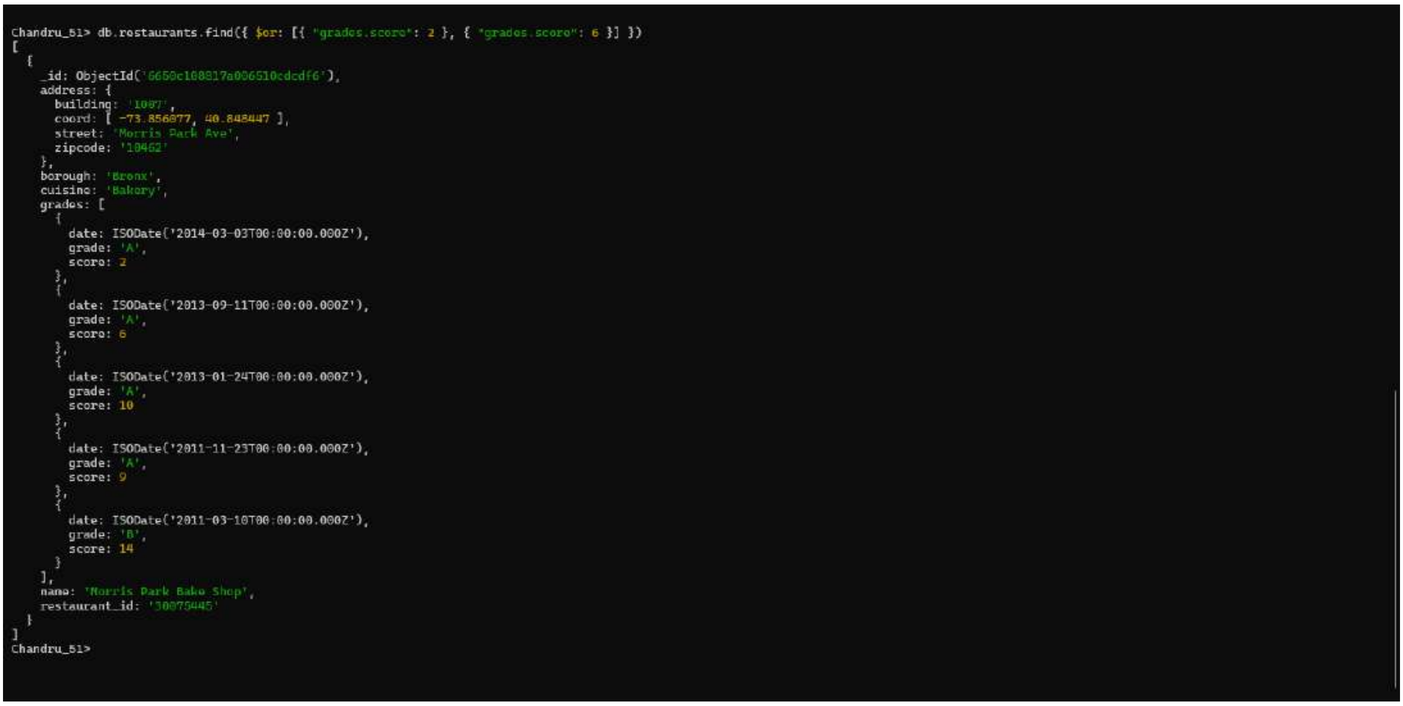
```
Chandru_51> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }]
, "cuisine": { $nin: ["American", "Chinese"] } })

Chandru_51>
```

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:



```
Chandru_51> db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
[
  {
    _id: ObjectId('6650c108817a006510cdcdf6'),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Chandru_51>
```

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

RESULT:

MONGO DB

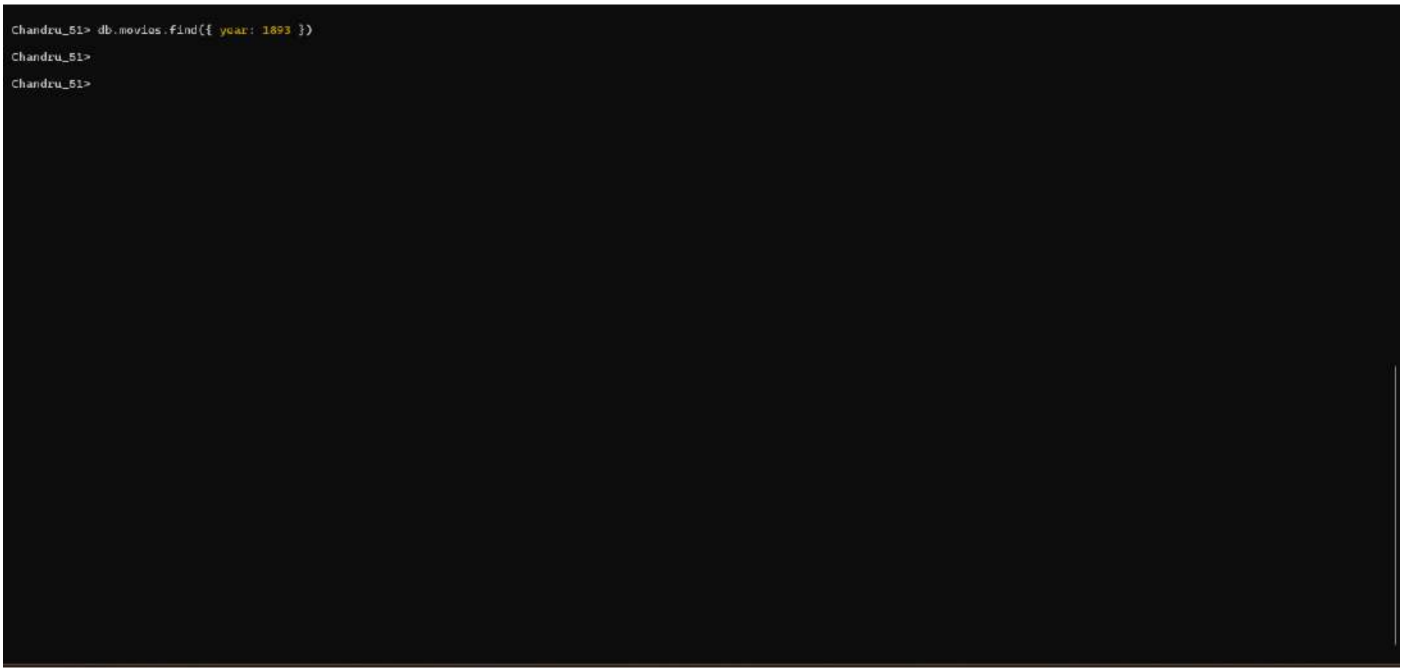
EX_NO: 20 **DATE:**

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

QUERY:

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

OUTPUT:



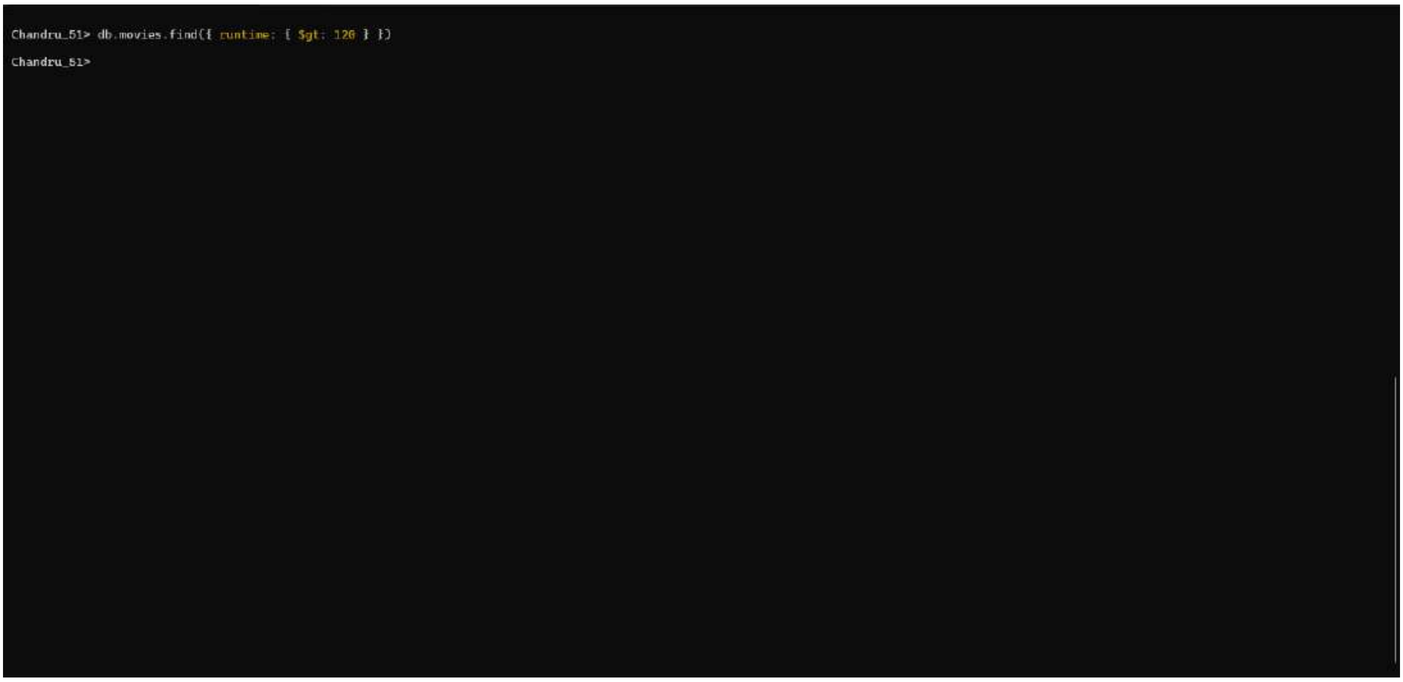
```
Chandru_51> db.movies.find({ year: 1893 })
Chandru_51>
Chandru_51>
```

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:



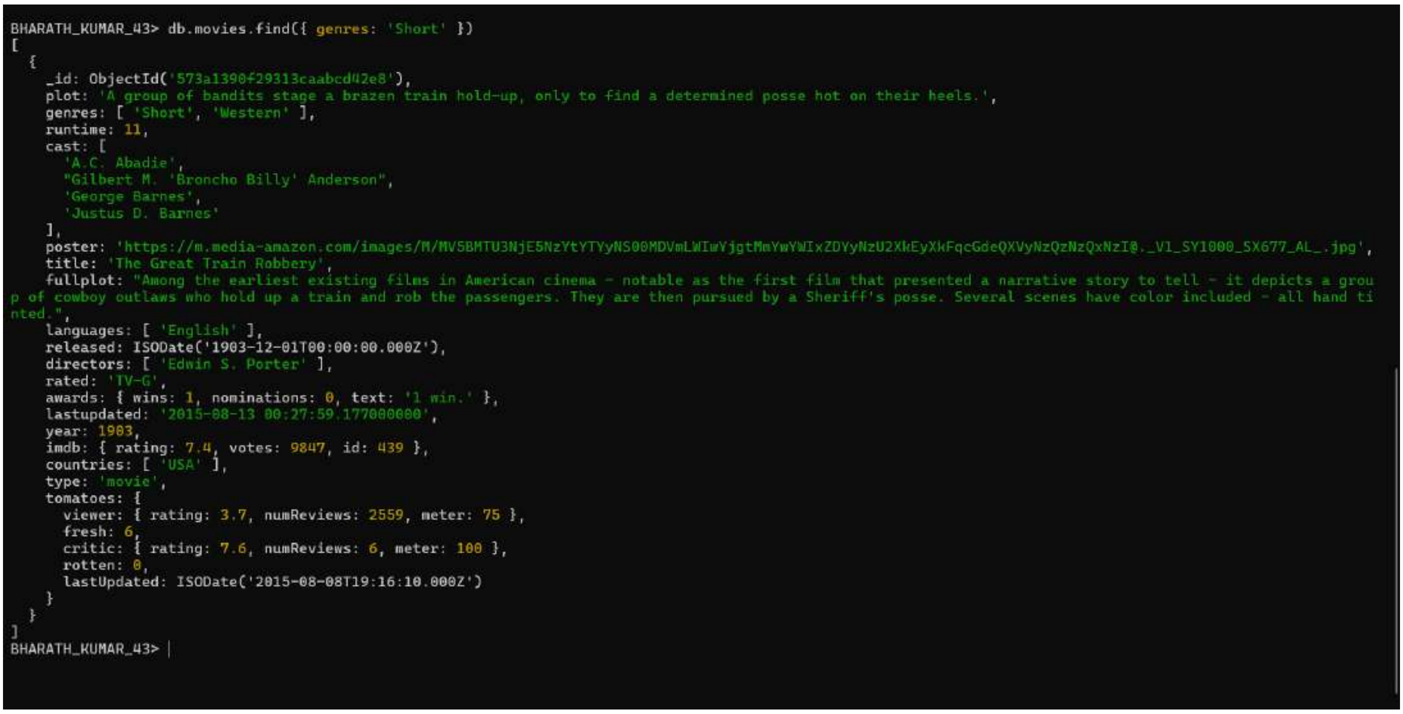
```
Chandru_51> db.movies.find({ runtime: { $gt: 120 } })
Chandru_51>
```

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

QUERY:

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

OUTPUT:



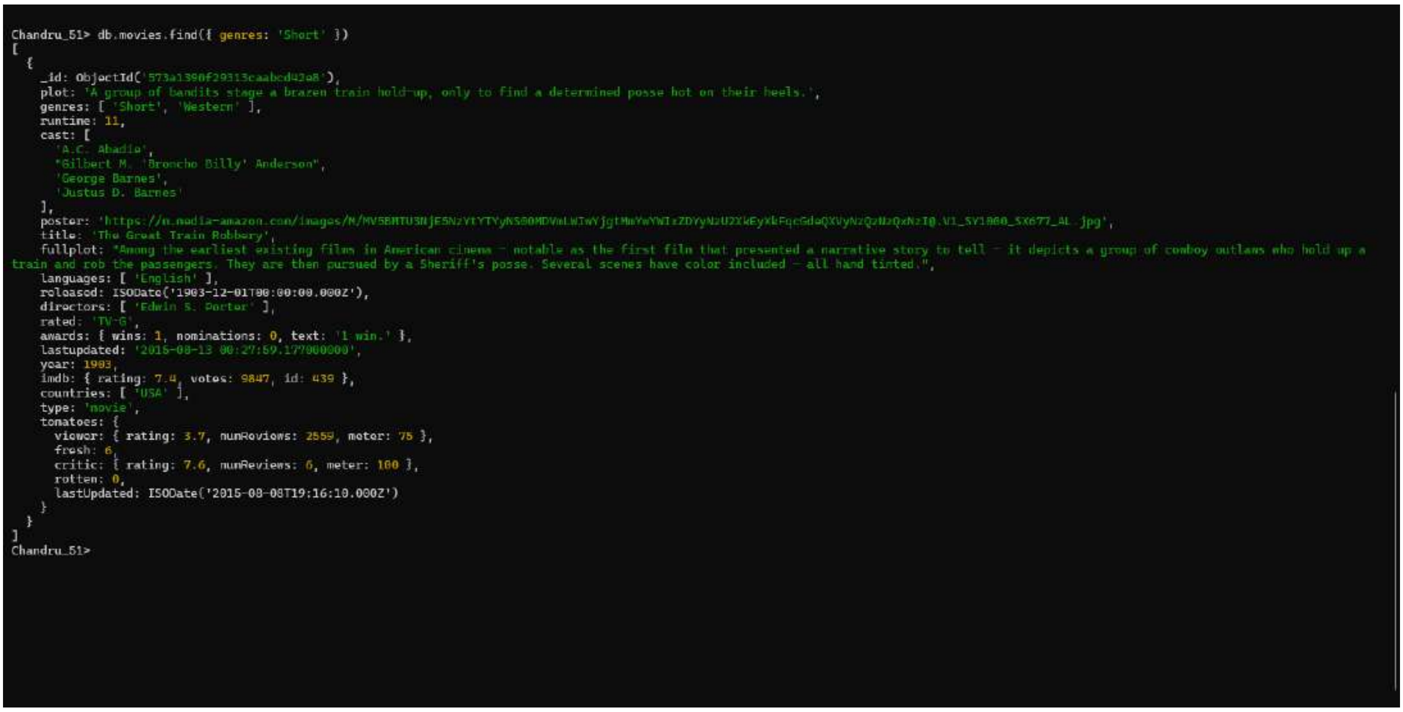
```
BHARATH_KUMAR_43> db.movies.find({ genres: 'Short' })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a grou
p of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand ti
nted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
BHARATH_KUMAR_43> |
```

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:



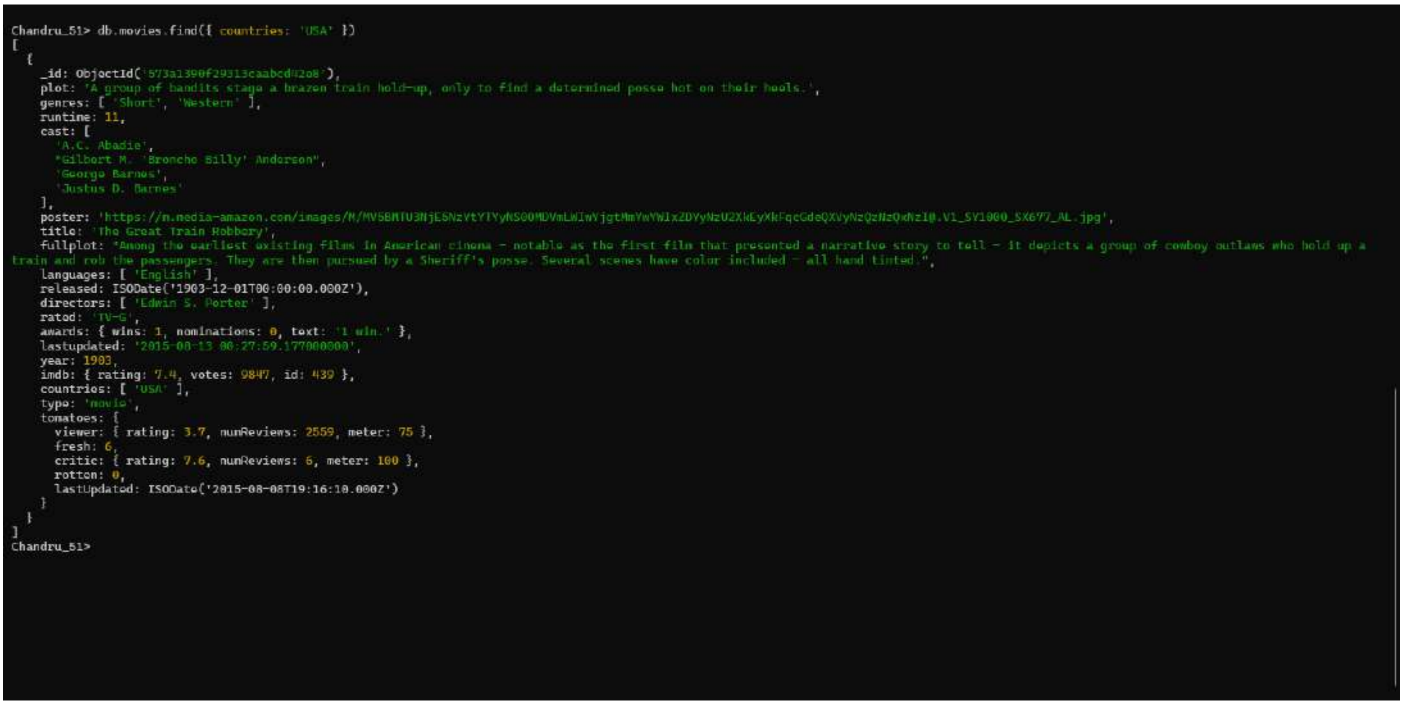
```
Chandru_51> db.movies.find({ genres: 'Short' })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a
train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Chandru_51>
```

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:



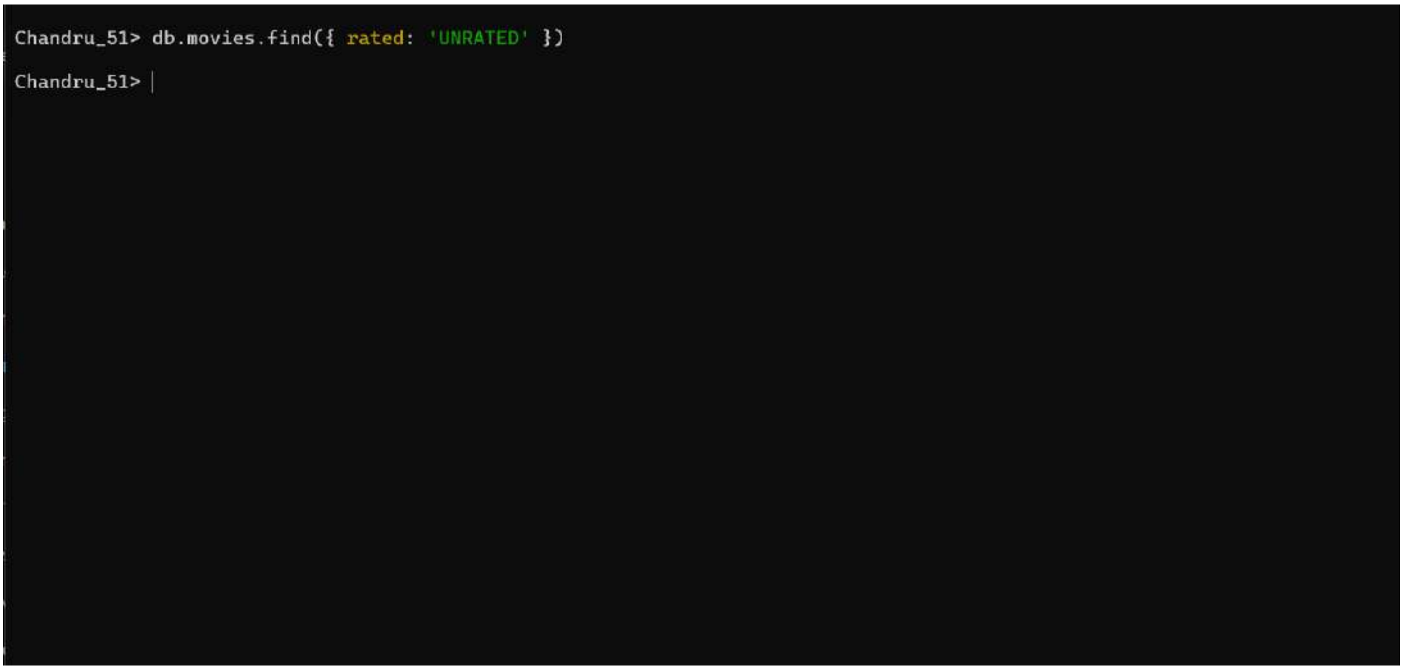
```
Chandru_51> db.movies.find({ countries: 'USA' })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@.V1_SY1000_SX677_AL.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a
train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Chandru_51>
```

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

QUERY:

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

OUTPUT:



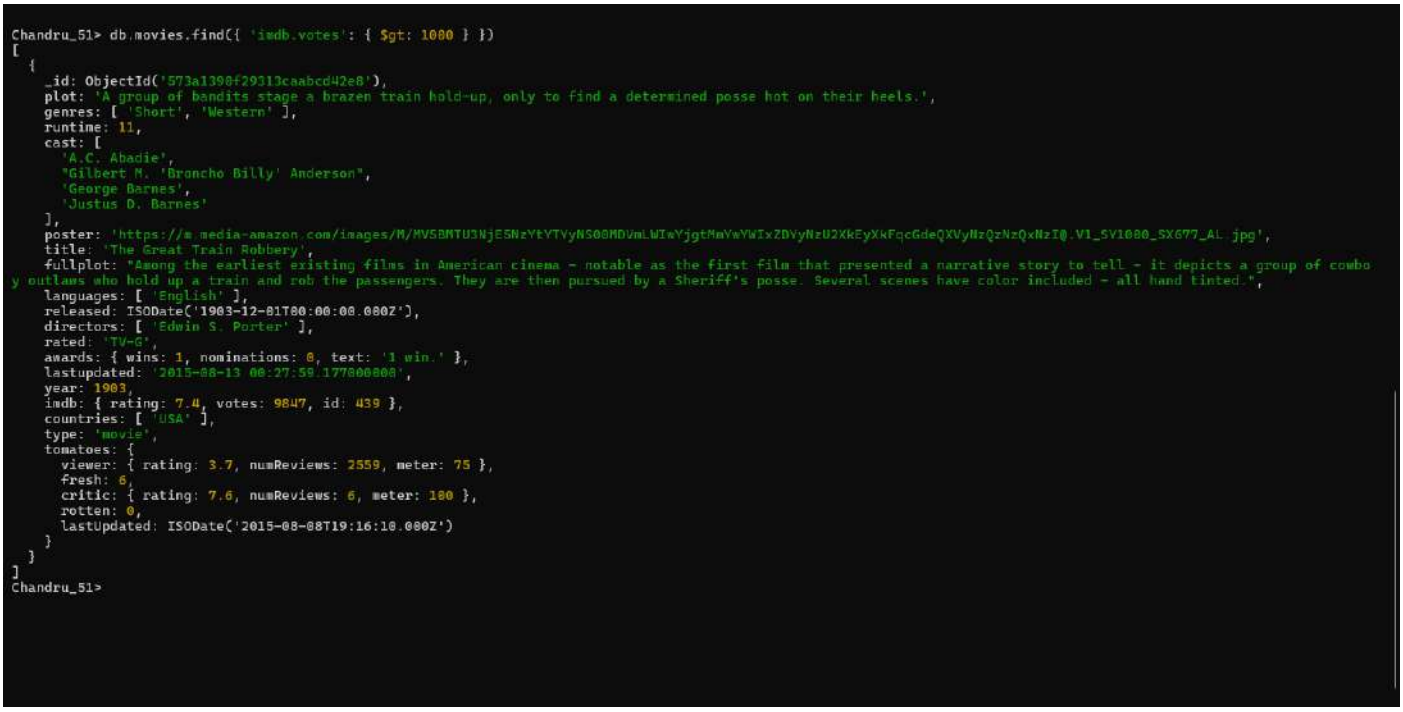
```
Chandru_51> db.movies.find({ rated: 'UNRATED' })

Chandru_51> |
```

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:



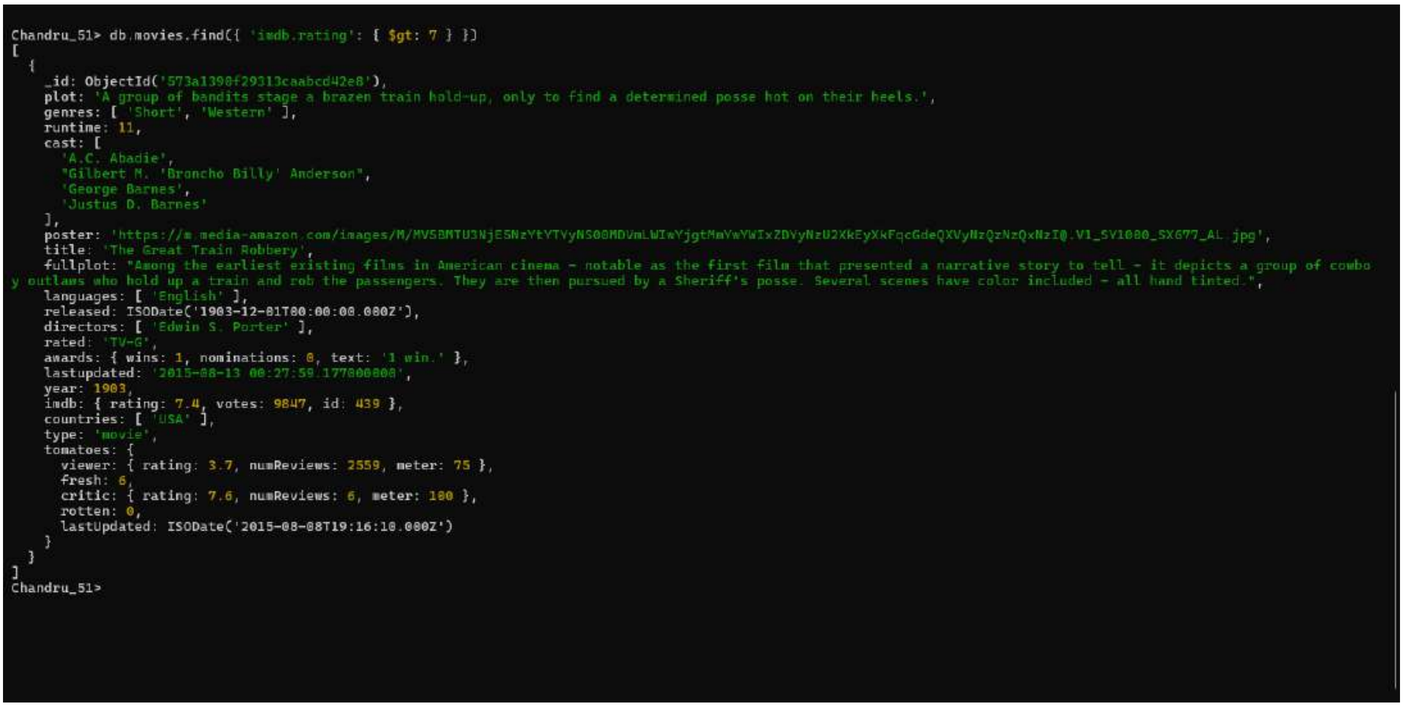
```
Chandru_51> db.movies.find({ 'imdb.votes': { $gt: 1000 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowbo
y outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Chandru_51>
```

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:



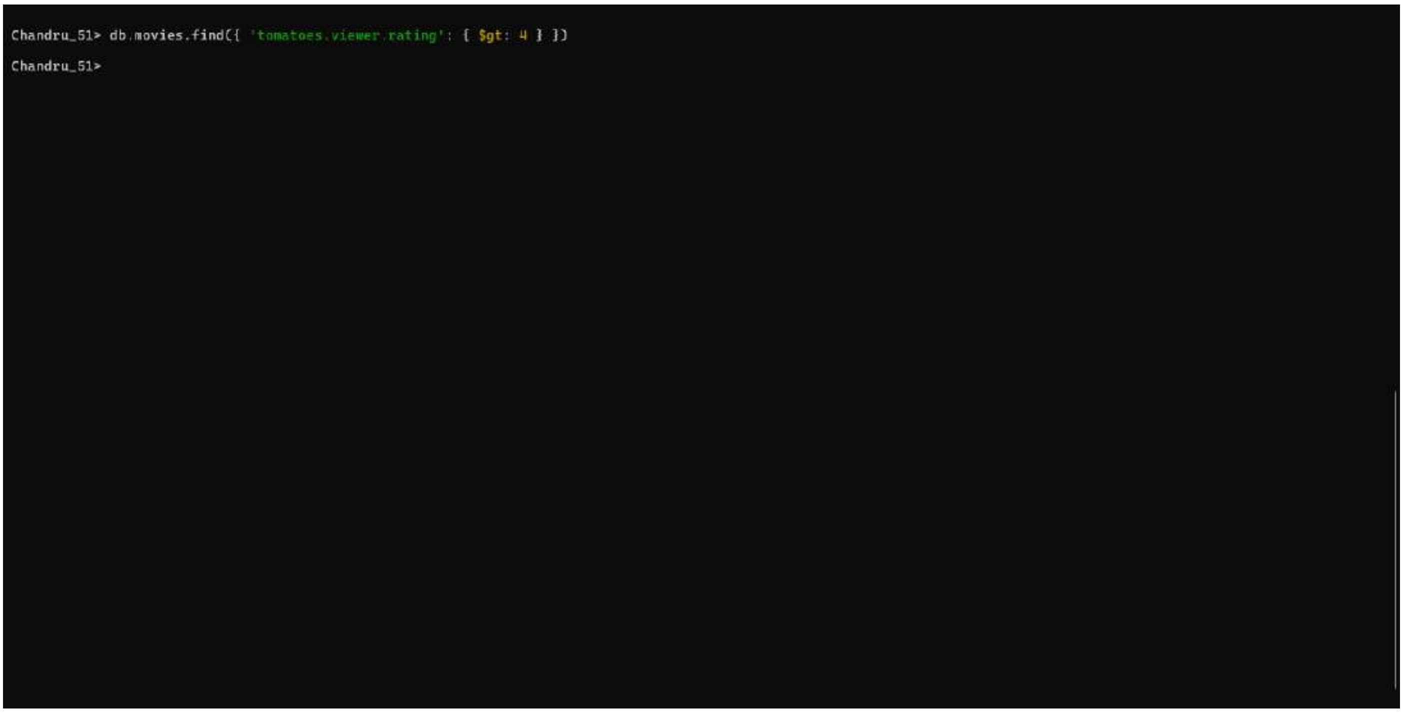
```
Chandru_51> db.movies.find({ 'imdb.rating': { $gt: 7 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowbo
y outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Chandru_51>
```

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:



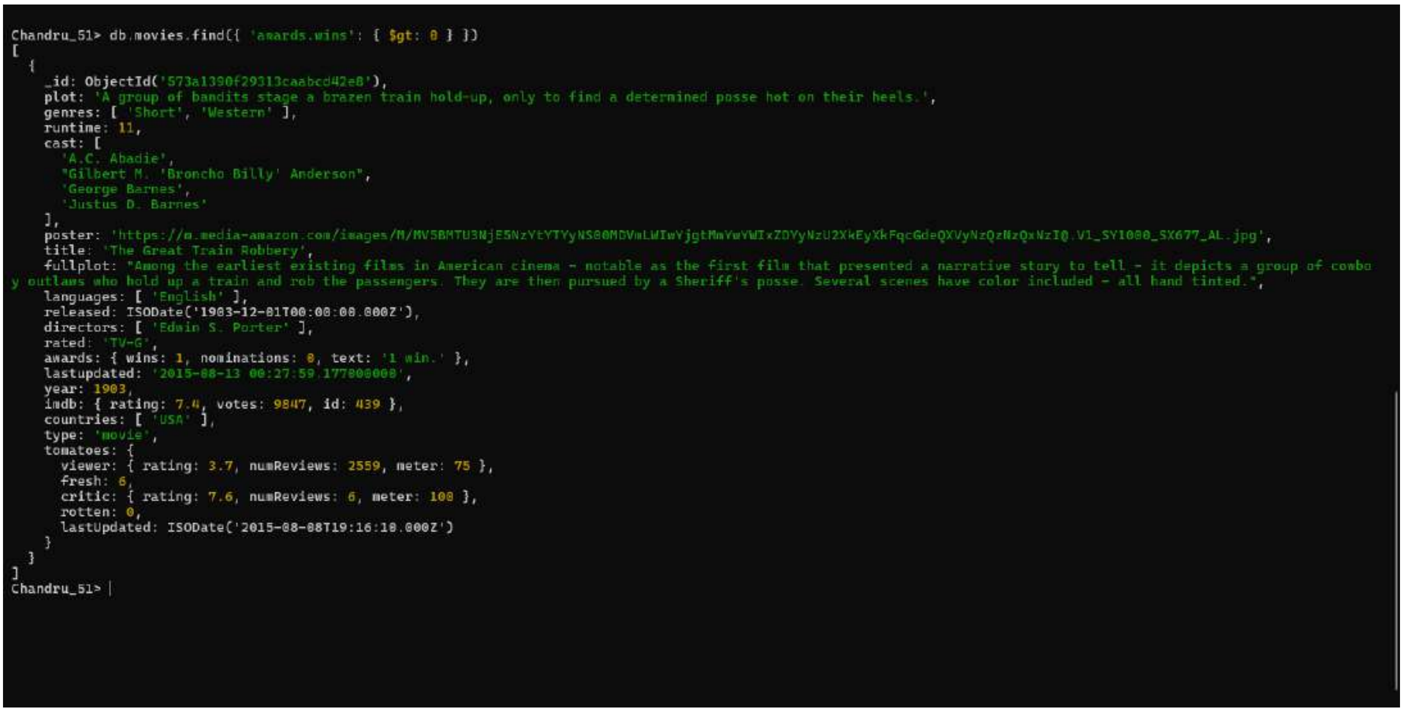
```
Chandru_51> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
Chandru_51>
```

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

QUERY:

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

OUTPUT:



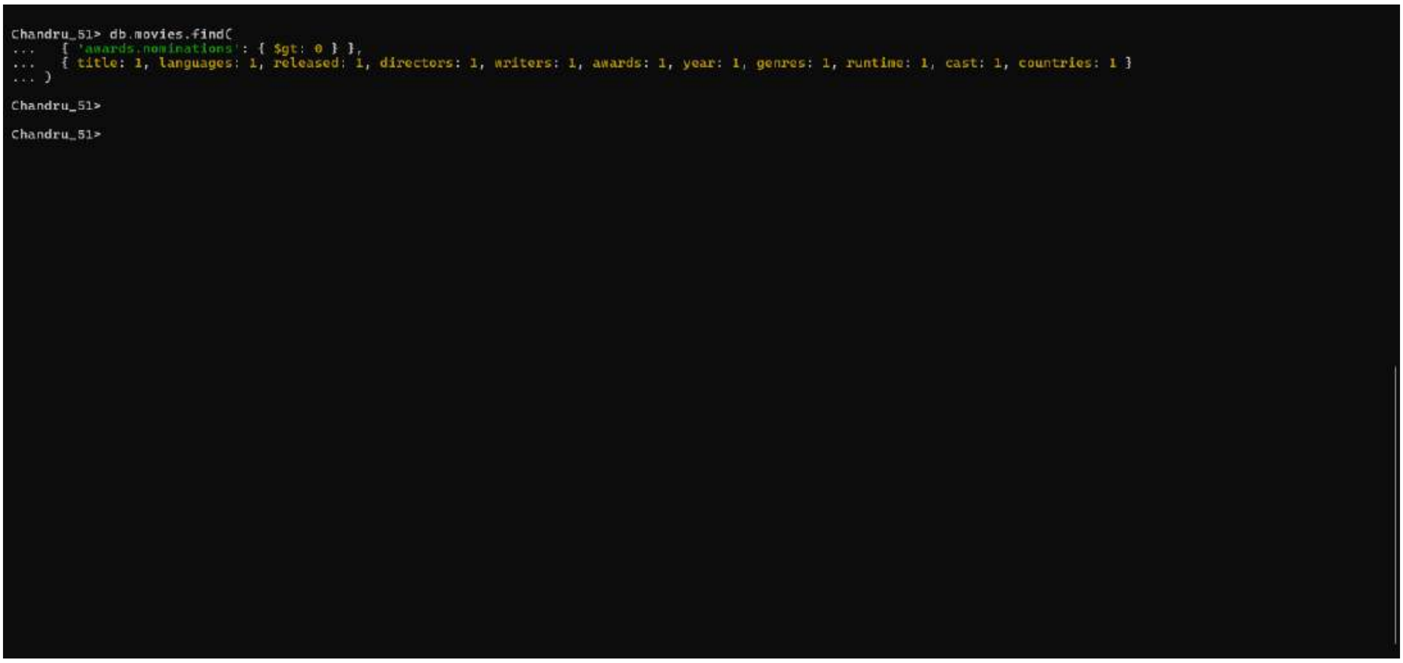
```
Chandru_51> db.movies.find({ 'awards.wins': { $gt: 0 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowbo
y outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Chandru_51> |
```

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:



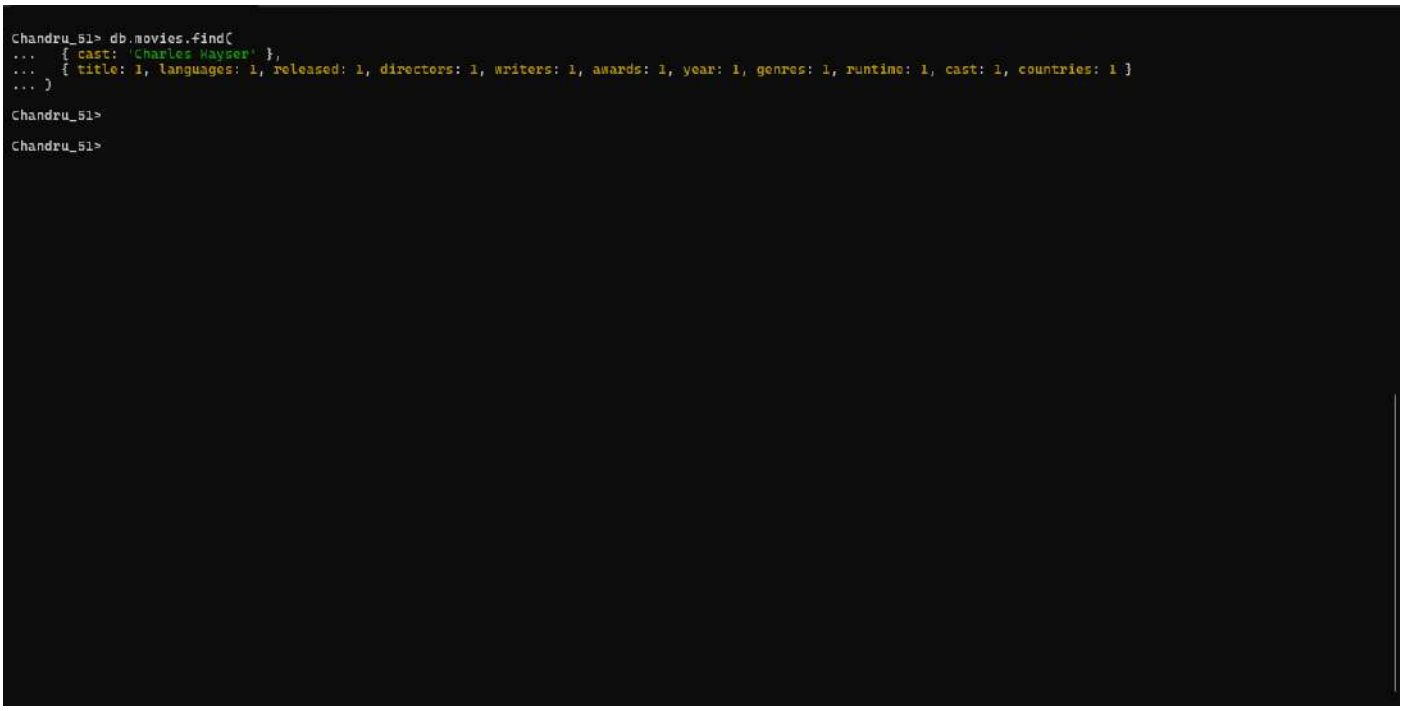
```
Chandru_51> db.movies.find(
...    { 'awards.nominations': { $gt: 0 } },
...    { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }
... )

Chandru_51>

Chandru_51>
```

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:



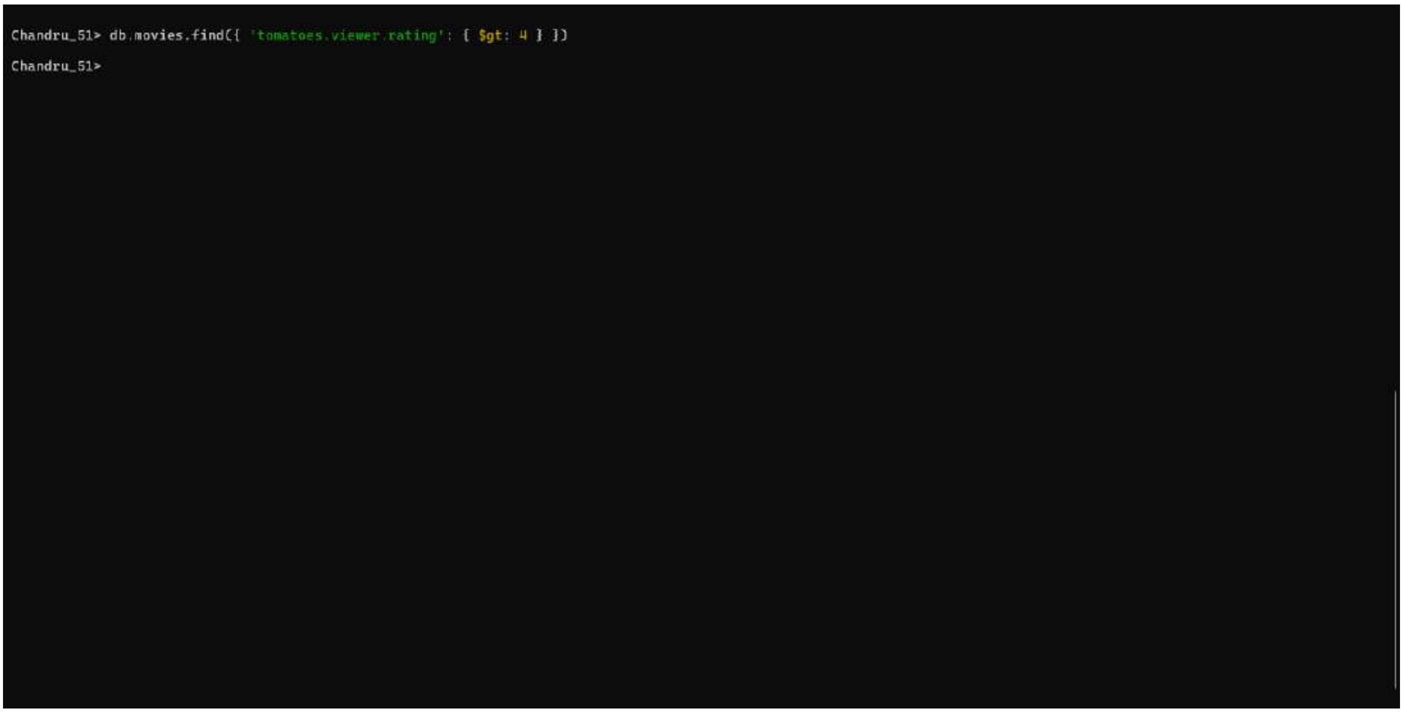
```
Chandru_51> db.movies.find(
...    { cast: 'Charles Kayser' },
...    { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }
... )

Chandru_51>

Chandru_51>
```

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:



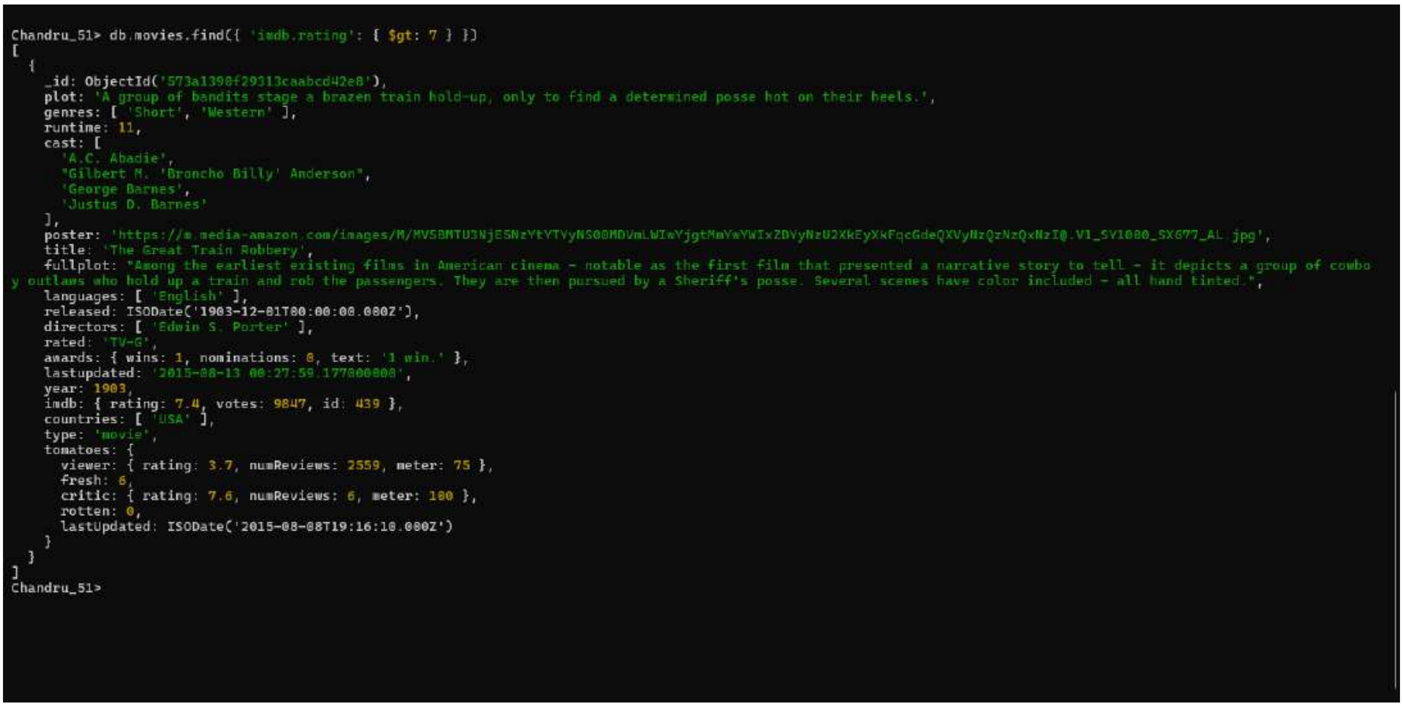
```
Chandru_51> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
Chandru_51>
```

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:



```
Chandru_51> db.movies.find({ 'imdb.rating': { $gt: 7 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      "Gilbert M. 'Broncho Billy' Anderson",
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYyNS00MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@.V1_SY1000_SX677_AL.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowbo
y outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Chandru_51>
```

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

RESULT: