

Creating and Managing Tables

EX_NO:1

DATE:

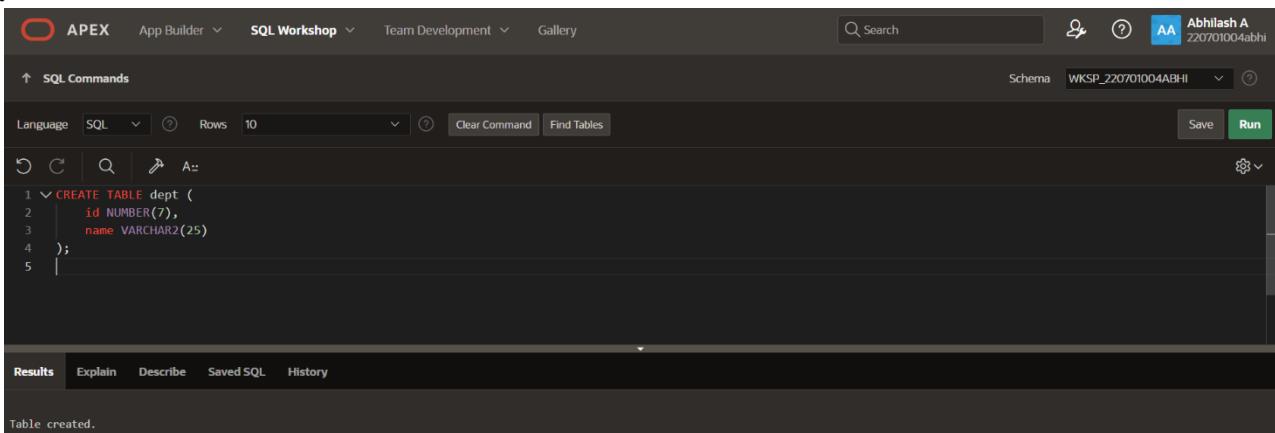
1. Create the DEPT table based on the DEPARTMENT following the table instance chart below. Confirm that the table is created.

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

QUERY:

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

OUTPUT:



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

```
1 ✓ CREATE TABLE dept (
2     id NUMBER(7),
3     name VARCHAR2(25)
4 );
5 |
```

The 'Results' tab shows the output: "Table created."

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

| | | | | |
|--------------|----|-----------|------------|---------|
| Column name | ID | LAST_NAME | FIRST_NAME | DEPT_ID |
| Key Type | | | | |
| Nulls/Unique | | | | |
| FK table | | | | |

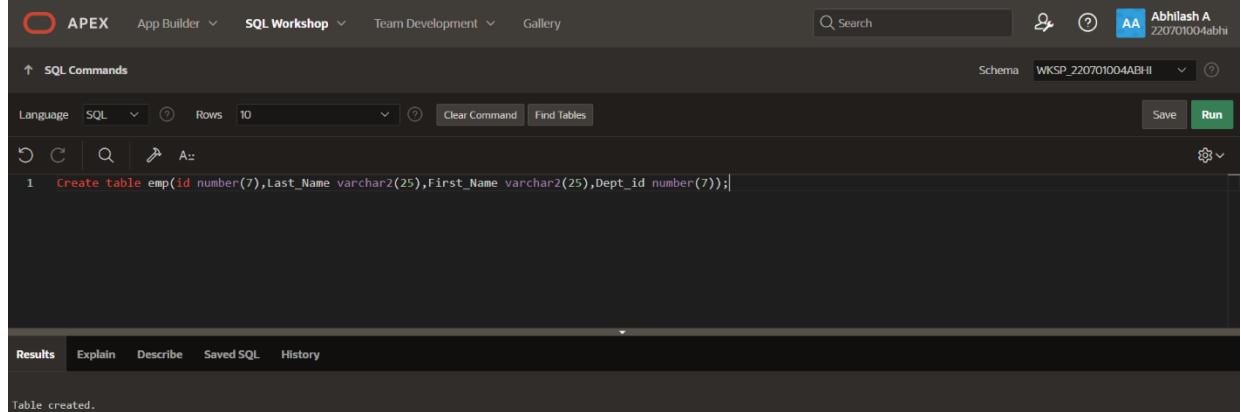
| | | | | |
|-----------|--|--|--|--|
| FK column | | | | |
|-----------|--|--|--|--|

| | | | | |
|------------------|--------|----------|----------|--------|
| Data Type | Number | Varchar2 | Varchar2 | Number |
| Length | 7 | 25 | 25 | 7 |

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema is set to WKSP_220701004ABHI. The SQL command window contains the following code:

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

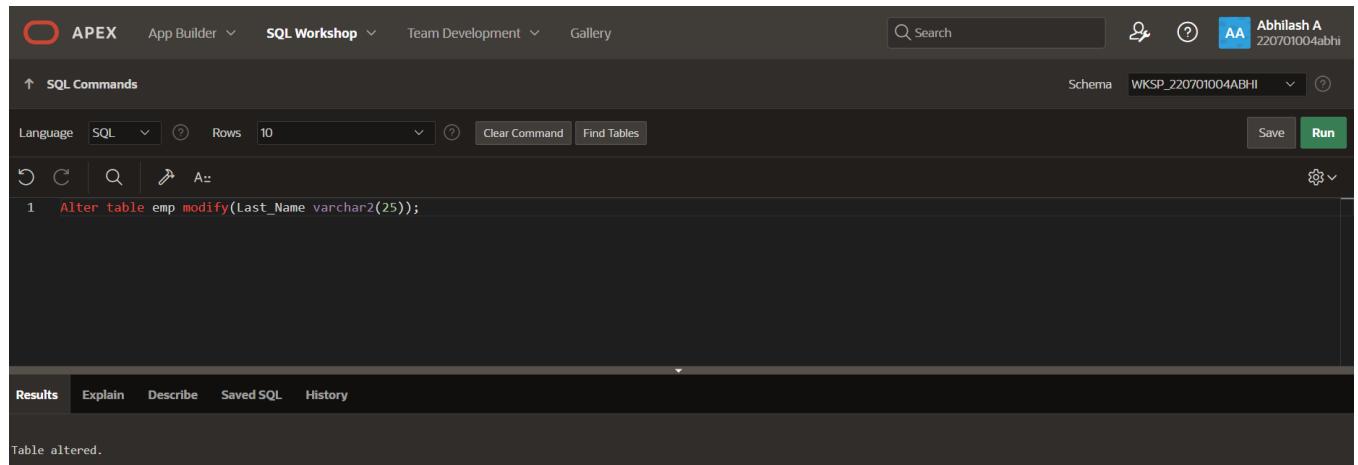
The results window shows the message "Table created." indicating the table was successfully created.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema is set to WKSP_220701004ABHI. The SQL command window contains the following code:

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

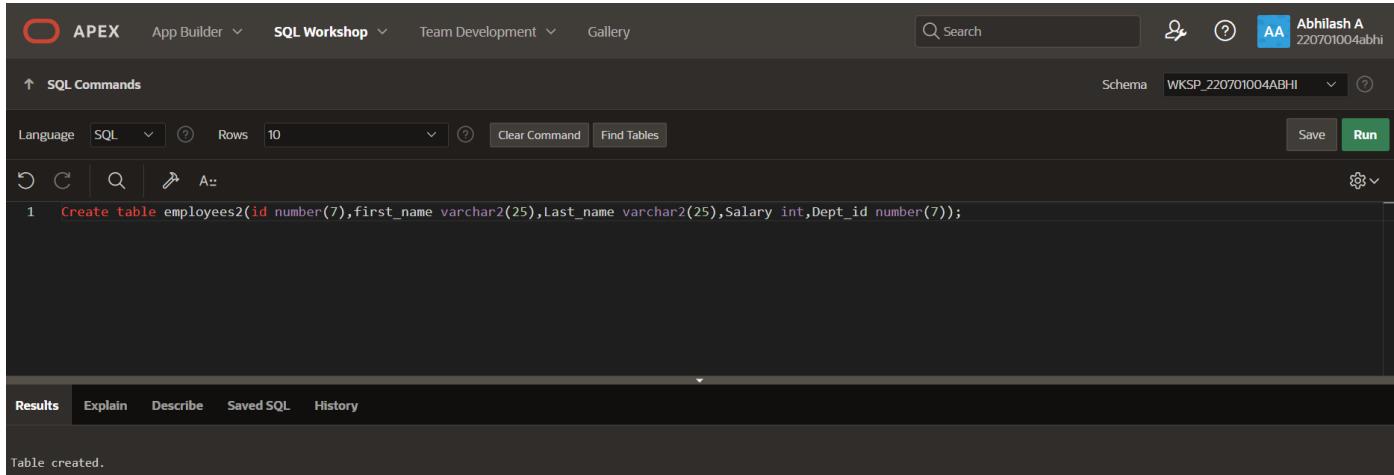
The results window shows the message "Table altered." indicating the table was successfully modified.

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:



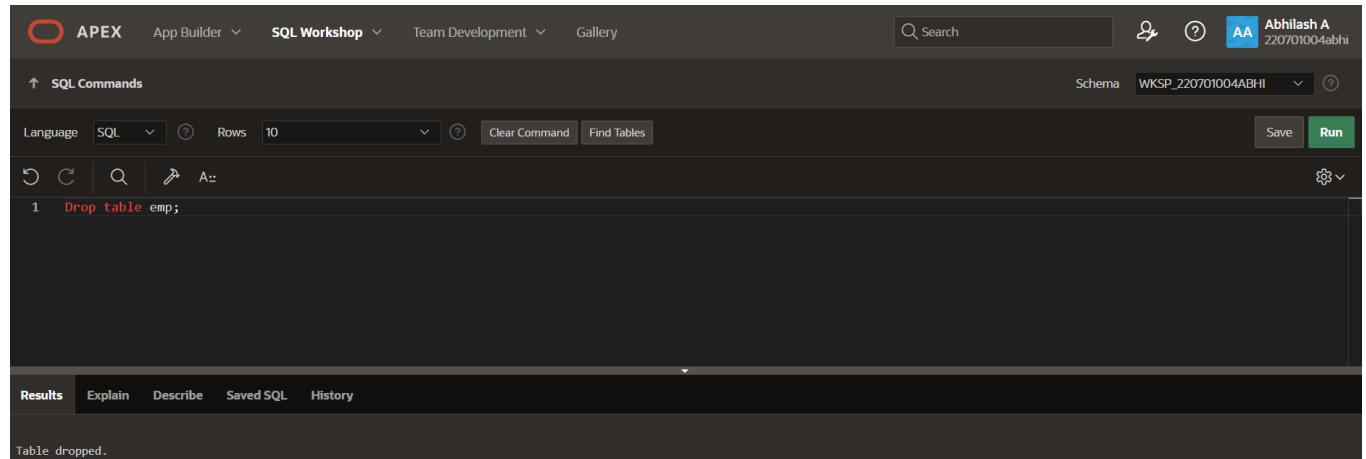
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The SQL editor contains the command: 'Create table employees2(id number(7),first_name varchar2(25),Last_name varchar2(25),Salary int,Dept_id number(7));'. The results tab shows the message 'Table created.'

5.Drop the EMP table.

QUERY:

Drop table emp;

OUTPUT:



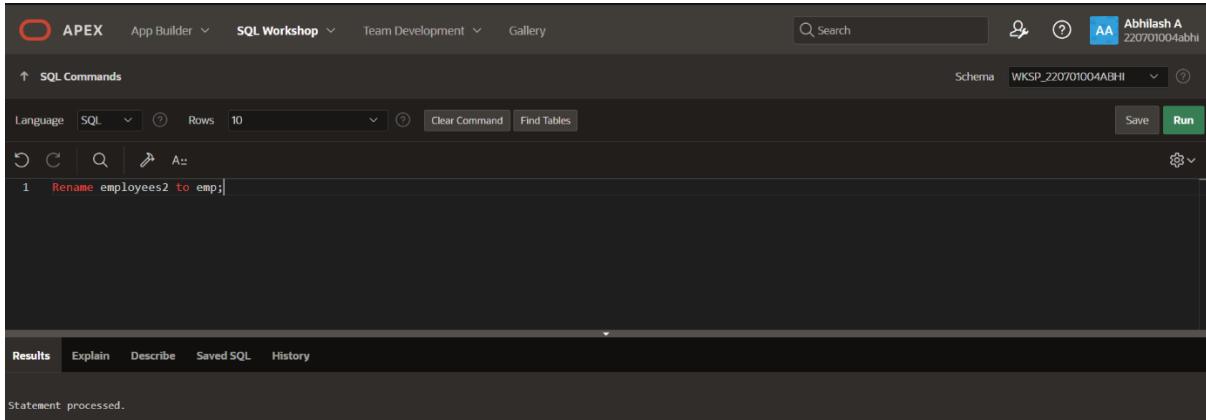
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The SQL editor contains the command: 'Drop table emp;'. The results tab shows the message 'Table dropped.'

6. Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename employees2 to emp;

OUTPUT:



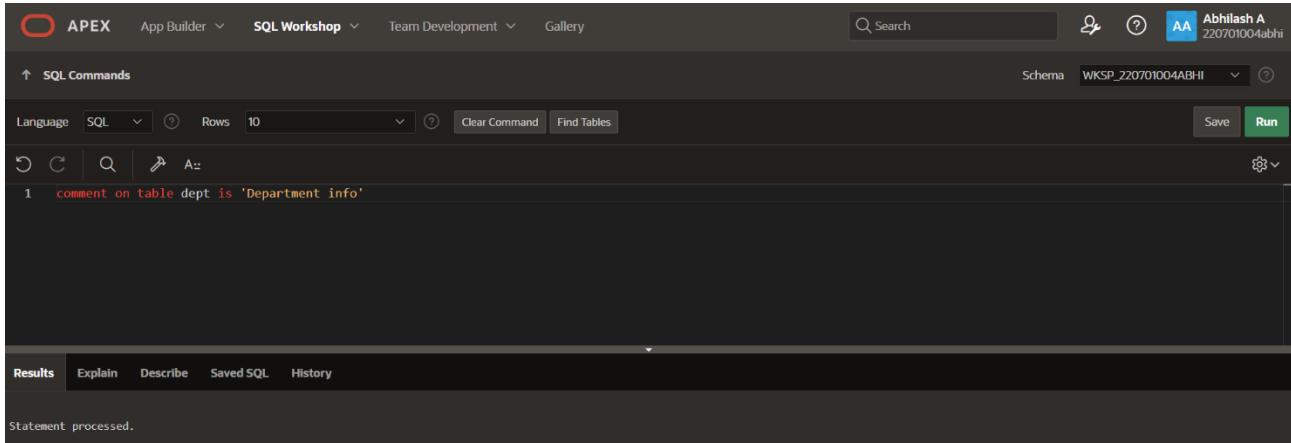
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. The command '1 Rename employees2 to emp;' is entered in the text area. Below the command, the status bar shows 'Statement processed.'

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

QUERY:

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

OUTPUT:



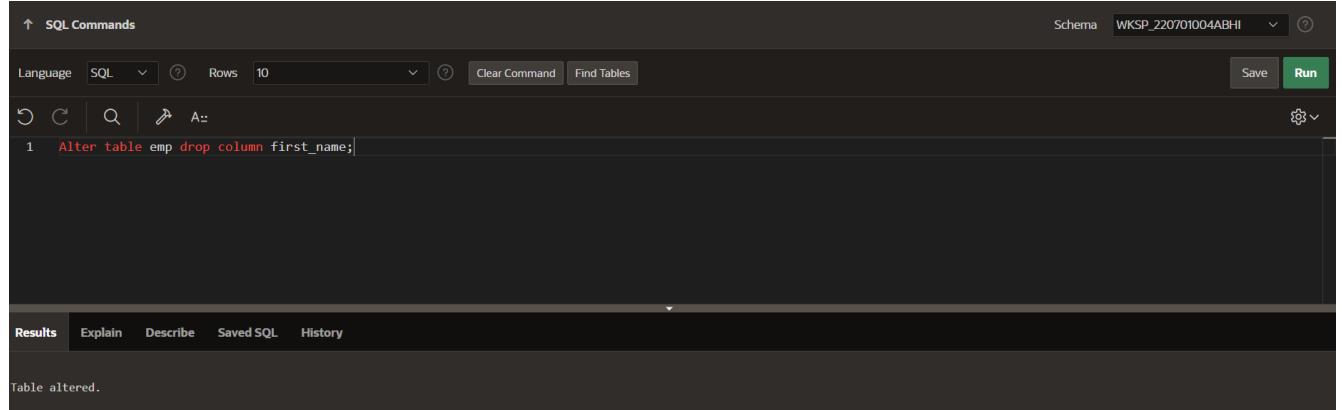
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. The command '1 comment on table dept is 'Department info'' is entered in the text area. Below the command, the status bar shows 'Statement processed.'

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

QUERY:

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

OUTPUT:



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

- Header: SQL Commands, Schema: WKSP_220701004ABHI, Save, Run.
- Toolbar: Language (SQL), Rows (10), Clear Command, Find Tables.
- Text Area: 1 Alter table emp drop column first_name; (highlighted in red).
- Bottom Bar: Results, Explain, Describe, Saved SQL, History.
- Status Bar: Table altered.

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

RESULT:

MANIPULATING DATA

EX_NO:2

DATE:

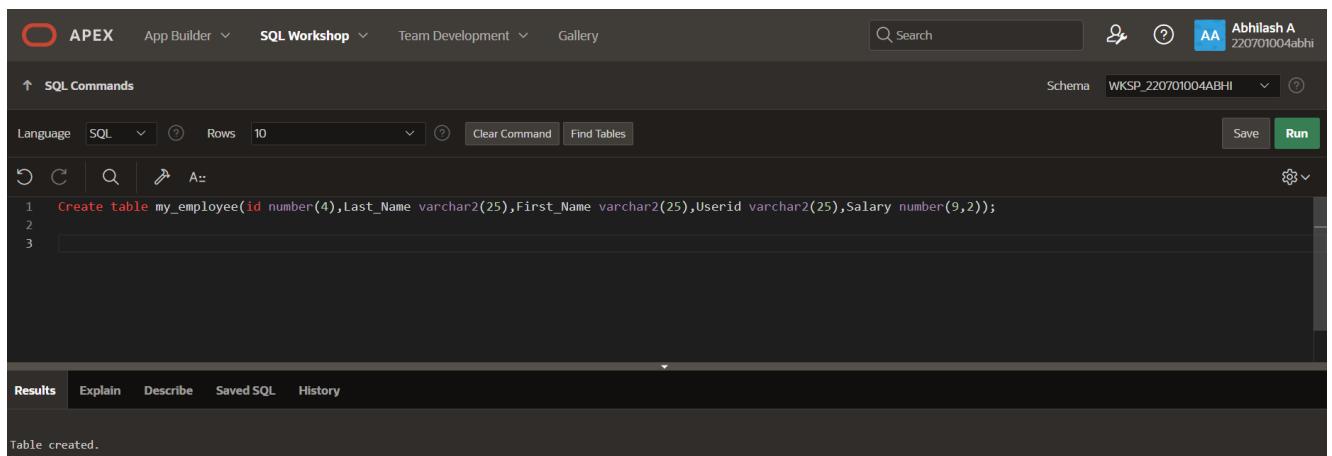
1.Create MY_EMPLOYEE table with the following structure

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

QUERY:

Create table my_employee(id number(4),Last_Name varchar2(25),First_Name varchar2(25),Userid varchar2(25),Salary number(9,2));

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' dropdown is open. The right side of the header shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands'. The command entered is:

```
1 Create table my_employee(id number(4),Last_Name varchar2(25),First_Name varchar2(25),Userid varchar2(25),Salary number(9,2));
```

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

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

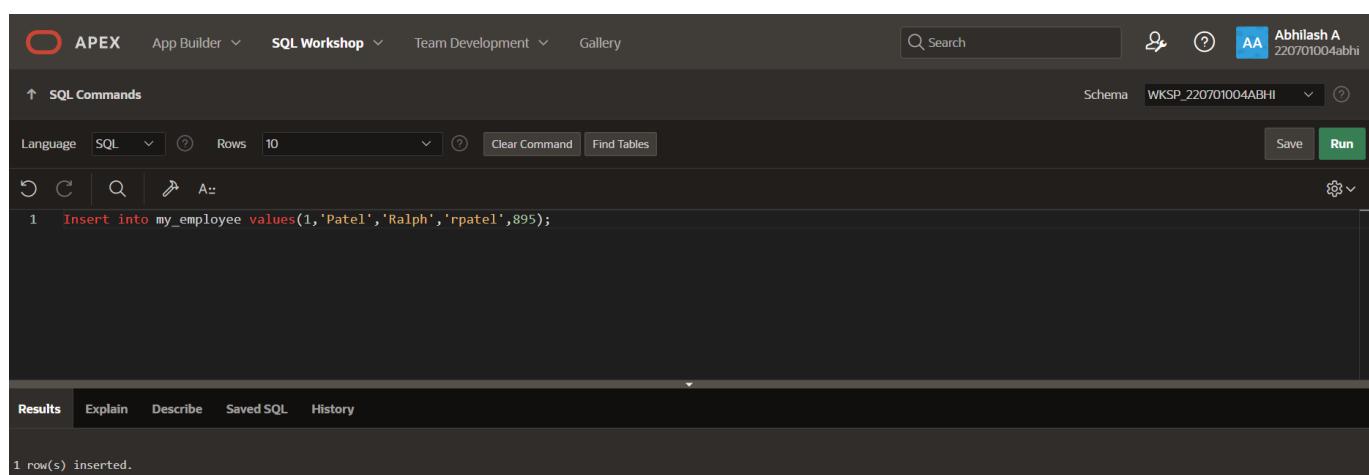
| ID | Last_name | First_name | Userid | salary |
|----|-----------|------------|----------|--------|
| 1 | Patel | Ralph | rpatel | 895 |
| 2 | Dancs | Betty | bdancs | 860 |
| 3 | Biri | Ben | bbiri | 1100 |
| 4 | Newman | Chad | Cnewman | 750 |
| 5 | Ropebur | Audrey | aropebur | 1550 |

QUERY:

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

```
Insert into my_employee values(1,'Dancs','Betty','bdancs',860);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL command:

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

The Results tab shows the output:

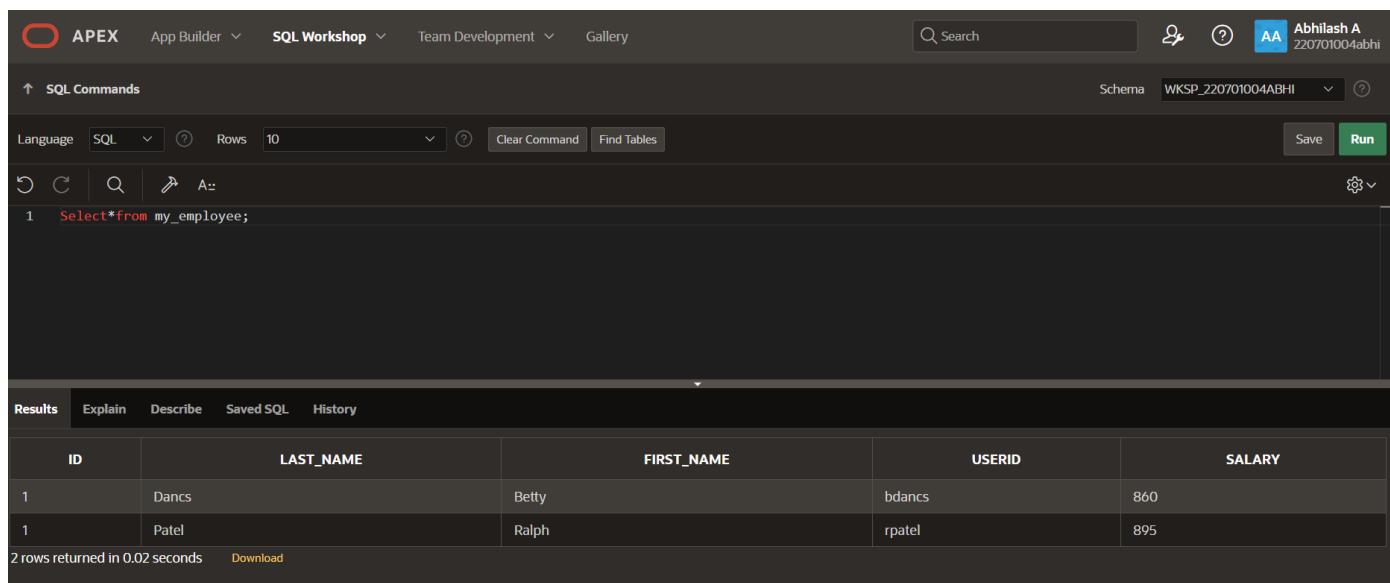
```
1 row(s) inserted.
```

3. Display the table with values.

QUERY:

```
Select*from my_employee;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL command:

```
1 Select*from my_employee;
```

The Results tab shows the output, displaying the data from the my_employee table:

| ID | LAST_NAME | FIRST_NAME | USERID | SALARY |
|----|-----------|------------|--------|--------|
| 1 | Dancs | Betty | bdancs | 860 |
| 1 | Patel | Ralph | rpatel | 895 |

At the bottom, it shows:

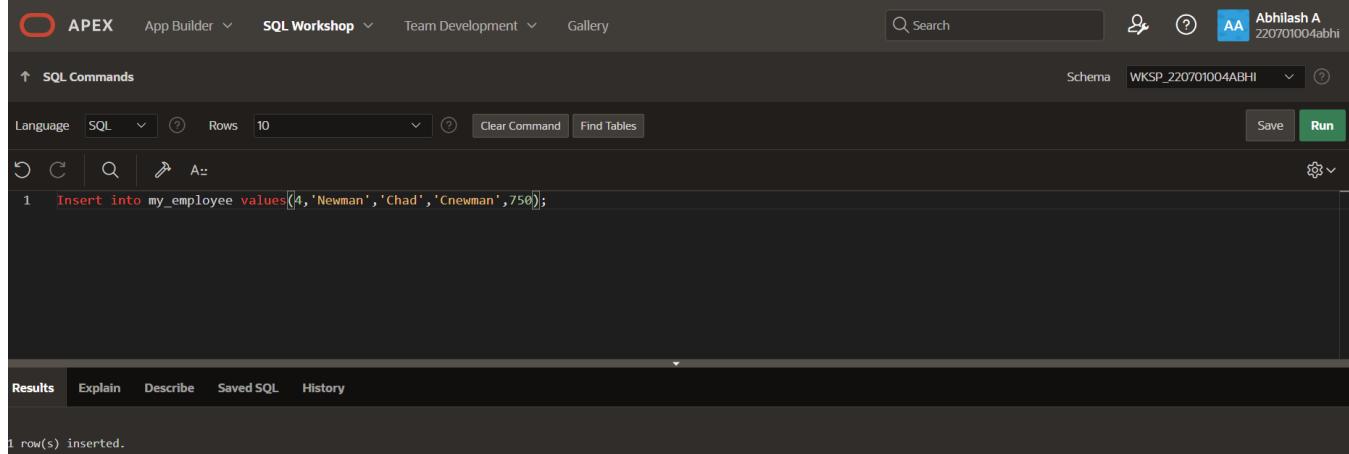
```
2 rows returned in 0.02 seconds Download
```

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

QUERY:

```
Insert into my_employee values(1,'Patel','Ralph','rpatel',1100);
Insert into my_employee values(1,'Dancs','Betty','bdancs',750);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a sub-section 'SQL'. The command entered is:

```
1 Insert into my_employee values(4,'Newman','Chad','Cnewman',750);
```

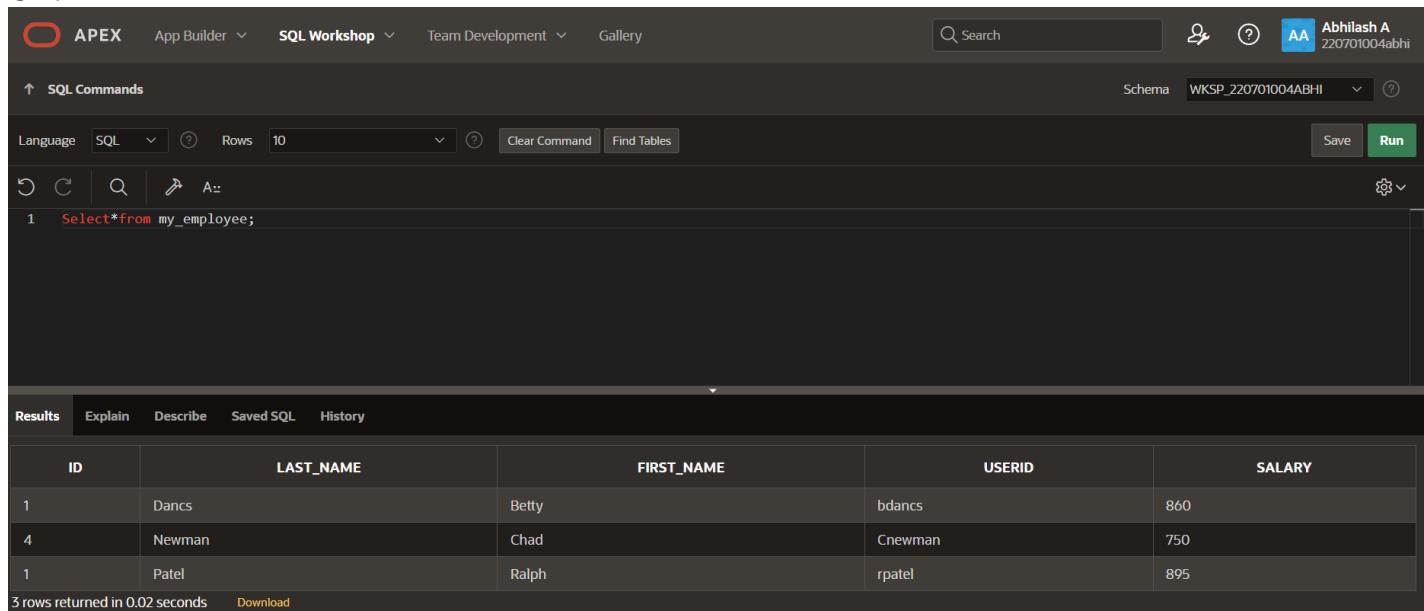
The results section shows the message '1 row(s) inserted.'

5. Make the data additions permanent.

QUERY:

```
Select*from my_employee;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a sub-section 'SQL'. The command entered is:

```
1 Select*from my_employee;
```

The results section displays a table with the following data:

| ID | LAST_NAME | FIRST_NAME | USERID | SALARY |
|----|-----------|------------|---------|--------|
| 1 | Dancs | Betty | bdancs | 860 |
| 4 | Newman | Chad | Cnewman | 750 |
| 1 | Patel | Ralph | rpatel | 895 |

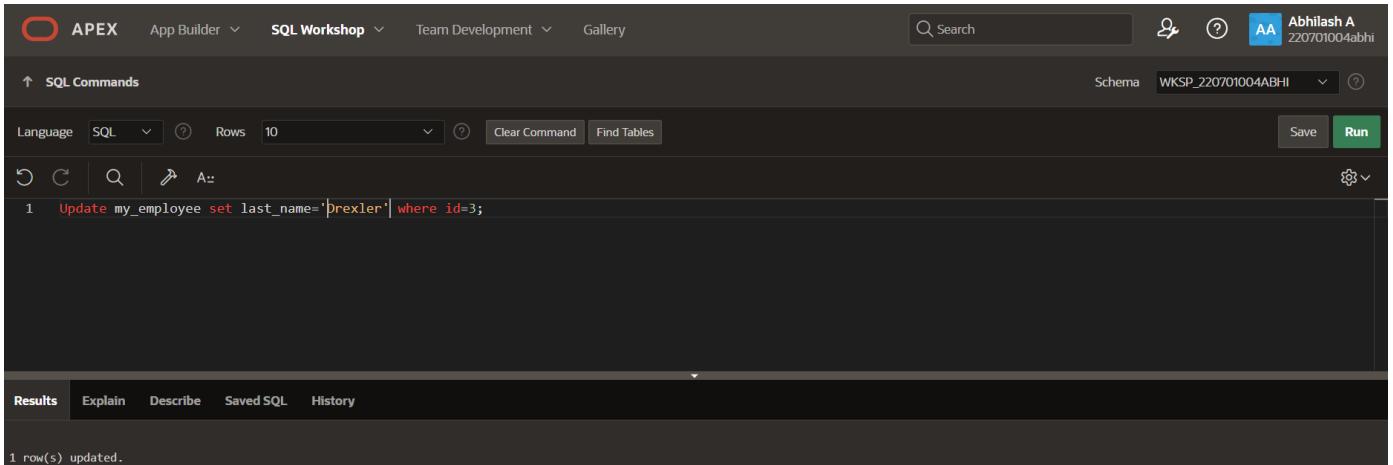
At the bottom, it says '3 rows returned in 0.02 seconds' and has a 'Download' link.

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

QUERY:

```
Update my_employee set last_name='Drexler' where id=3;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the user 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a sub-section '1 SQL Commands'. The SQL editor contains the following command:

```
1 Update my_employee set last_name='Drexler' where id=3;
```

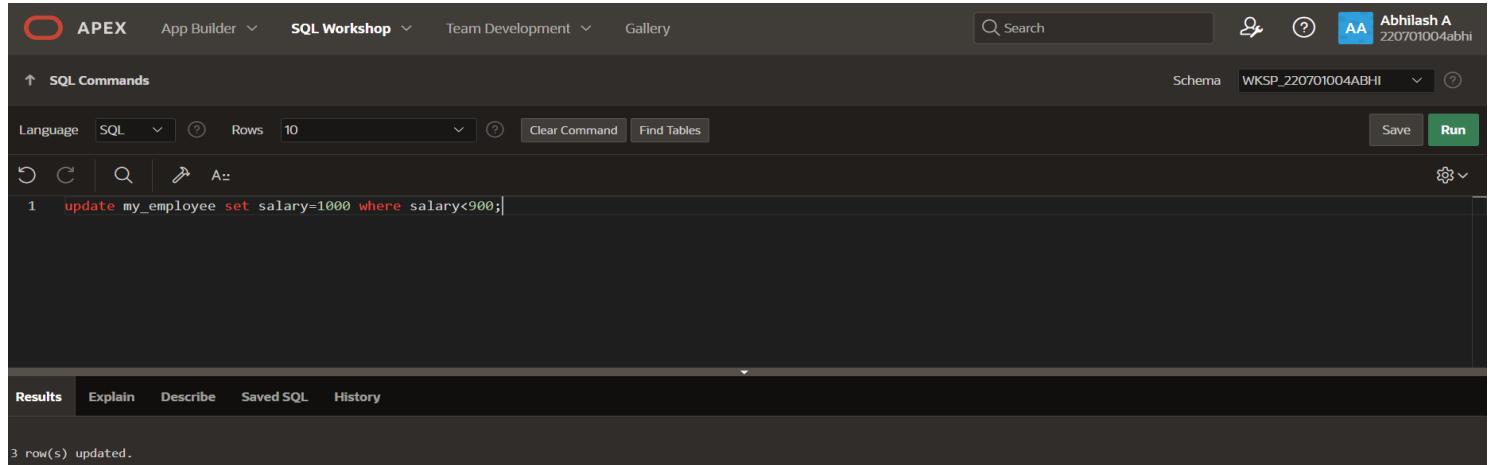
Below the editor, the results tab is selected, showing the message '1 row(s) updated.'

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

QUERY:

```
update my_employee set salary=1000 where salary<900;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the user 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a sub-section '1 SQL Commands'. The SQL editor contains the following command:

```
1 update my_employee set salary=1000 where salary<900;
```

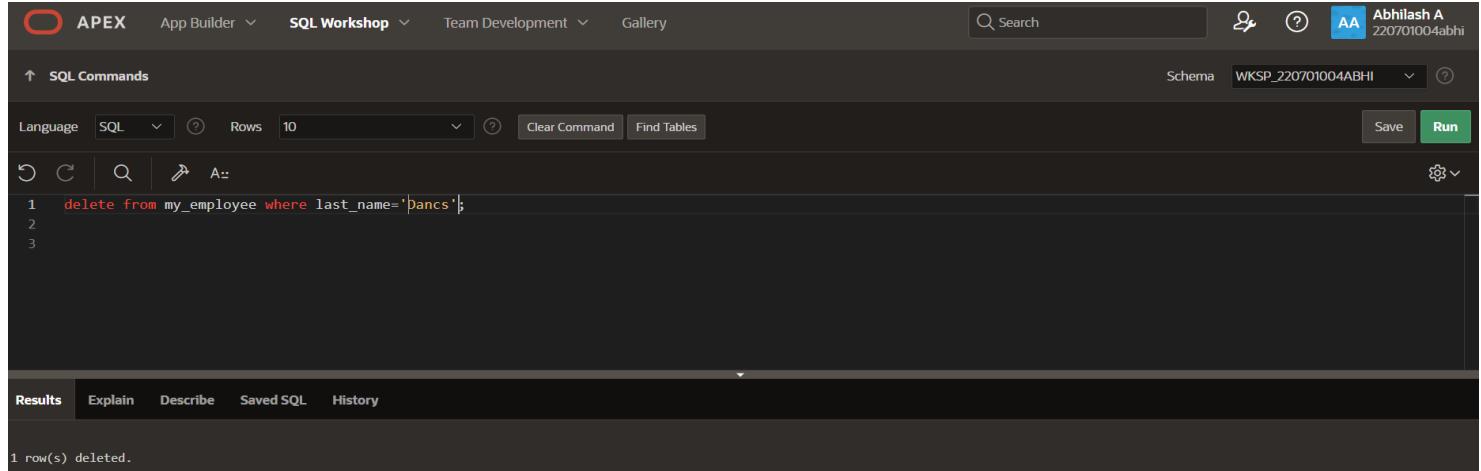
Below the editor, the results tab is selected, showing the message '3 row(s) updated.'

8.Delete Betty dancs from MY _EMPLOYEE table.

QUERY:

```
delete from my_employee where last_name='Dancs';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a user profile for 'Abhilash A' are on the right. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. It shows the following SQL command:

```
1 delete from my_employee where last_name='Dancs';
```

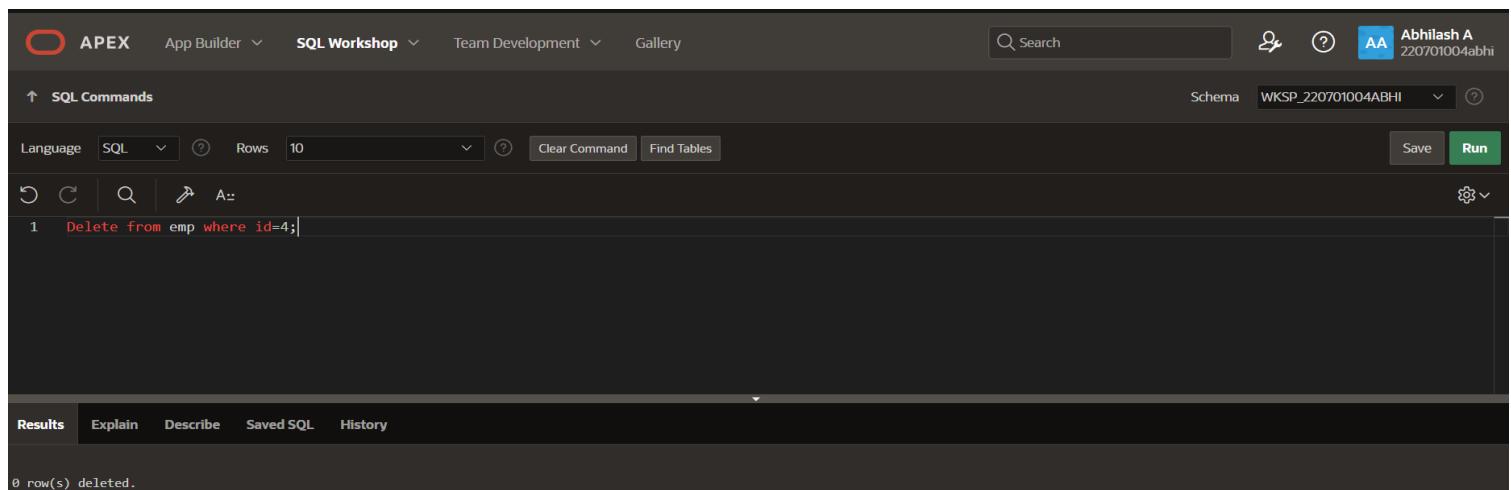
The 'Results' tab is selected, showing the output: '1 row(s) deleted.'

9.Empty the fourth row of the emp table.

QUERY:

```
Delete from emp where id=4;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a user profile for 'Abhilash A' are on the right. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. It shows the following SQL command:

```
1 Delete from emp where id=4;
```

The 'Results' tab is selected, showing the output: '0 row(s) deleted.'

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

RESULT:

INCLUDING CONSTRAINTS

EX_NO:3

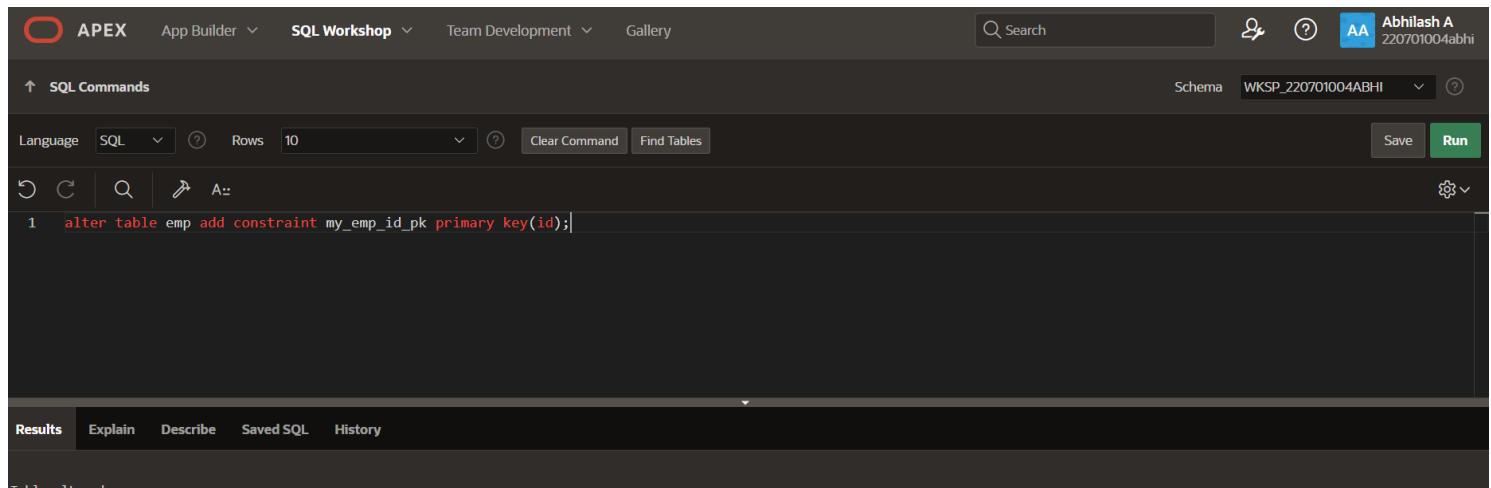
DATE:

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:

```
alter table emp add constraint my_emp_id_pk primary key(id);
```

OUTPUT:



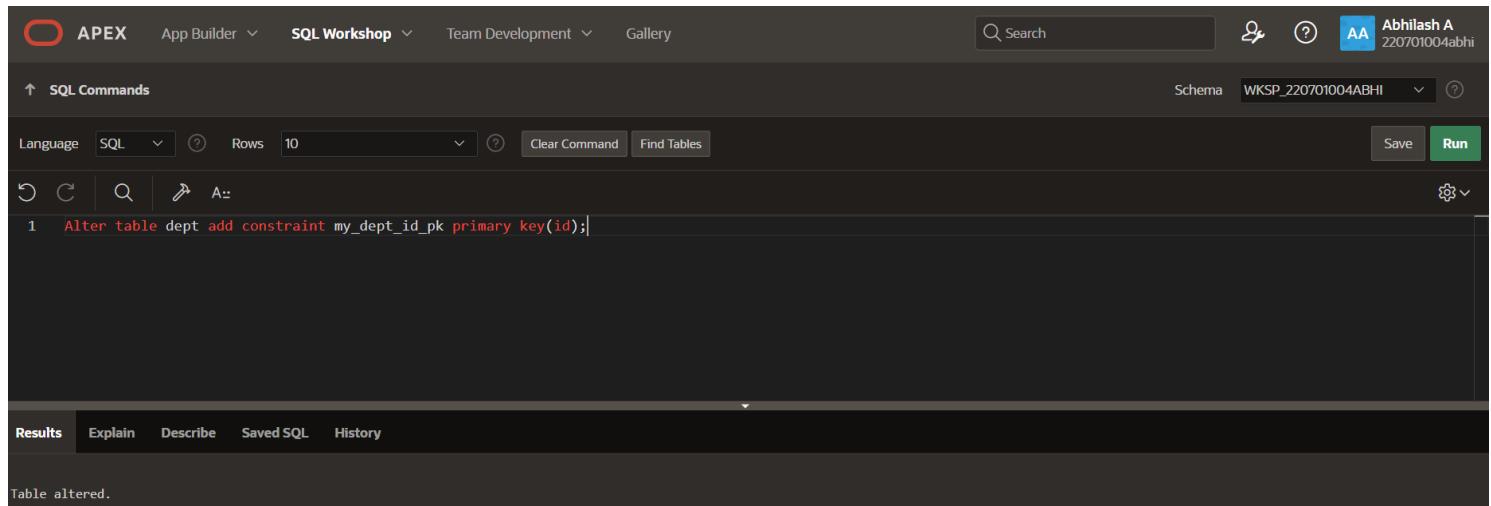
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side of the header shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The SQL editor contains the command: '1 alter table emp add constraint my_emp_id_pk primary key(id);'. The results tab at the bottom shows the output: 'Table altered.'

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

QUERY:

```
Alter table emp add constraint my_dept_id_pk primary key(id);
```

OUTPUT:



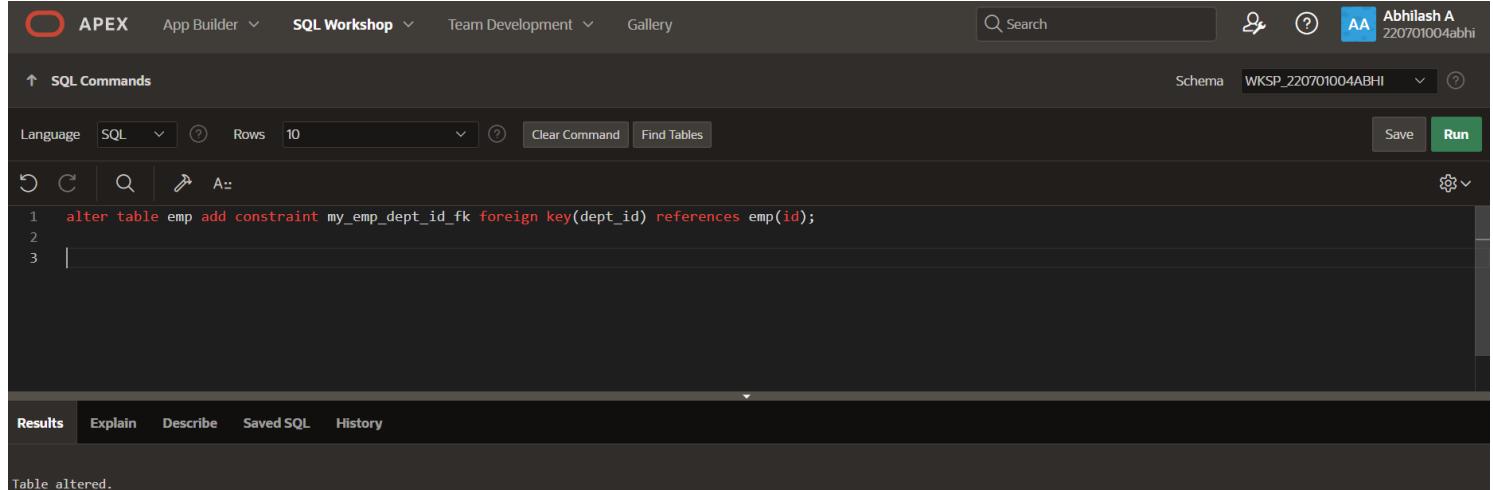
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side of the header shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The SQL editor contains the command: '1 Alter table dept add constraint my_dept_id_pk primary key(id);'. The results tab at the bottom shows the output: 'Table altered.'

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004abhi'. A search bar and a toolbar with various icons are also present. The main area is titled 'SQL Commands' and contains a text input field with the following SQL command:

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

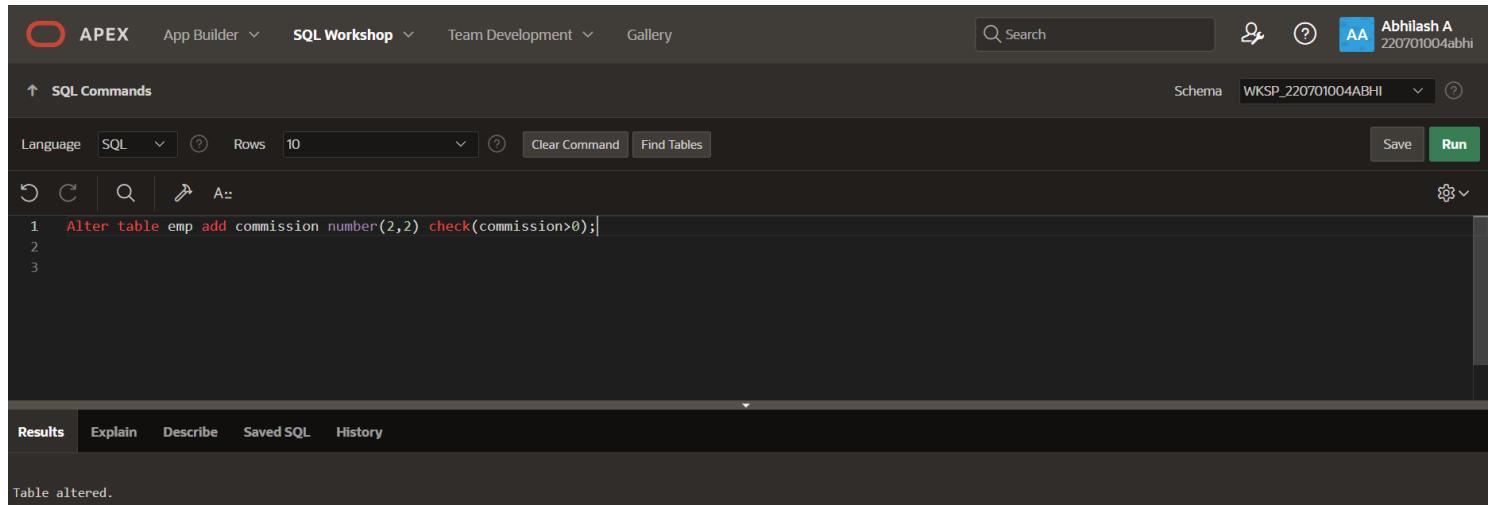
Below the command, the 'Results' tab is selected, showing the output: 'Table altered.'

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

QUERY:

```
Alter table emp add constraint number(2,2) check(commission);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004abhi'. A search bar and a toolbar with various icons are also present. The main area is titled 'SQL Commands' and contains a text input field with the following SQL command:

```
1 Alter table emp add commission number(2,2) check(commission>0);
```

Below the command, the 'Results' tab is selected, showing the output: 'Table altered.'

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

RESULT:

Writing Basic SQL SELECT Statements

EX_NO:4

DATE:

1. The following statement executes successfully.

Identify the Errors

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

QUERY:

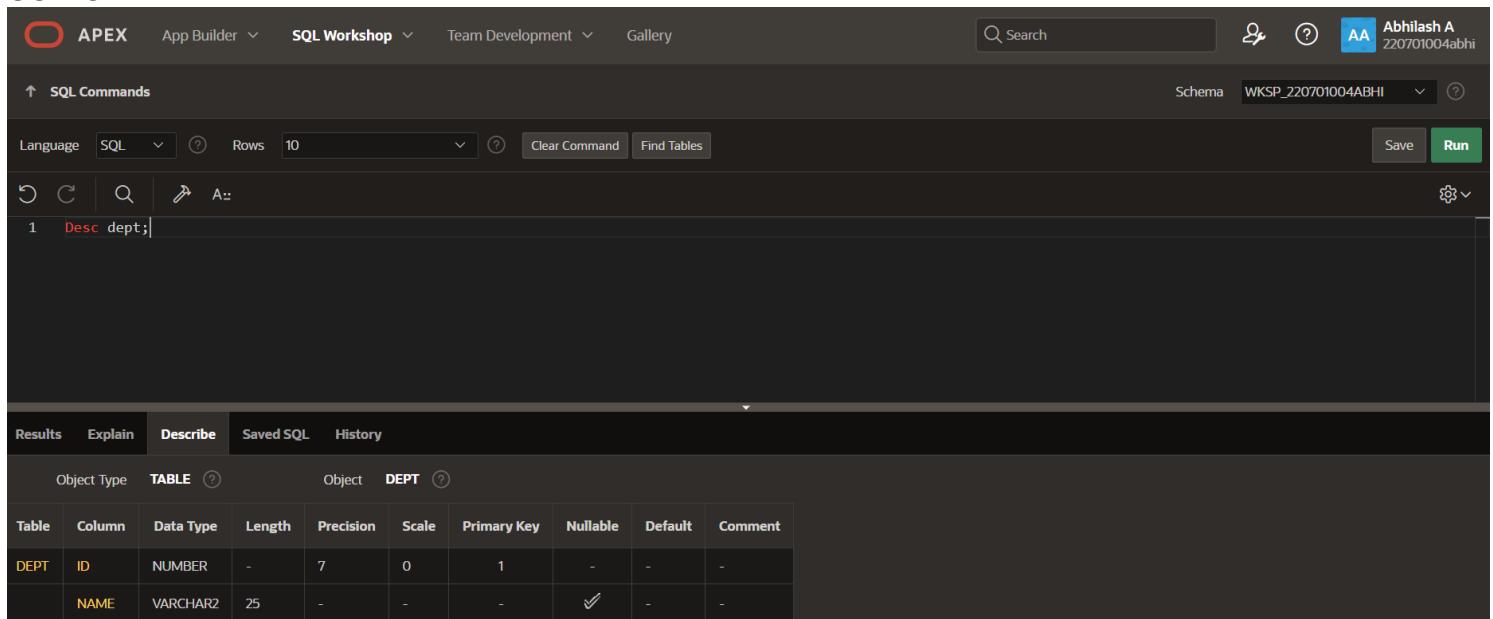
```
Select employee_id, last_name, sal*12 as "ANNUAL SALARY" from employees;
```

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

QUERY:

```
Desc dept;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' dropdown is open. The right side of the header shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. A search bar and a help icon are also present. The main workspace is titled 'SQL Commands'. It shows the command 'Desc dept;' entered in the text area. Below the command, the 'Describe' tab is selected in the results panel, which displays the structure of the 'DEPT' table. The table has two columns: 'ID' (NUMBER, 7, 0, Primary Key) and 'NAME' (VARCHAR2, 25, nullable). The 'Comment' column is empty.

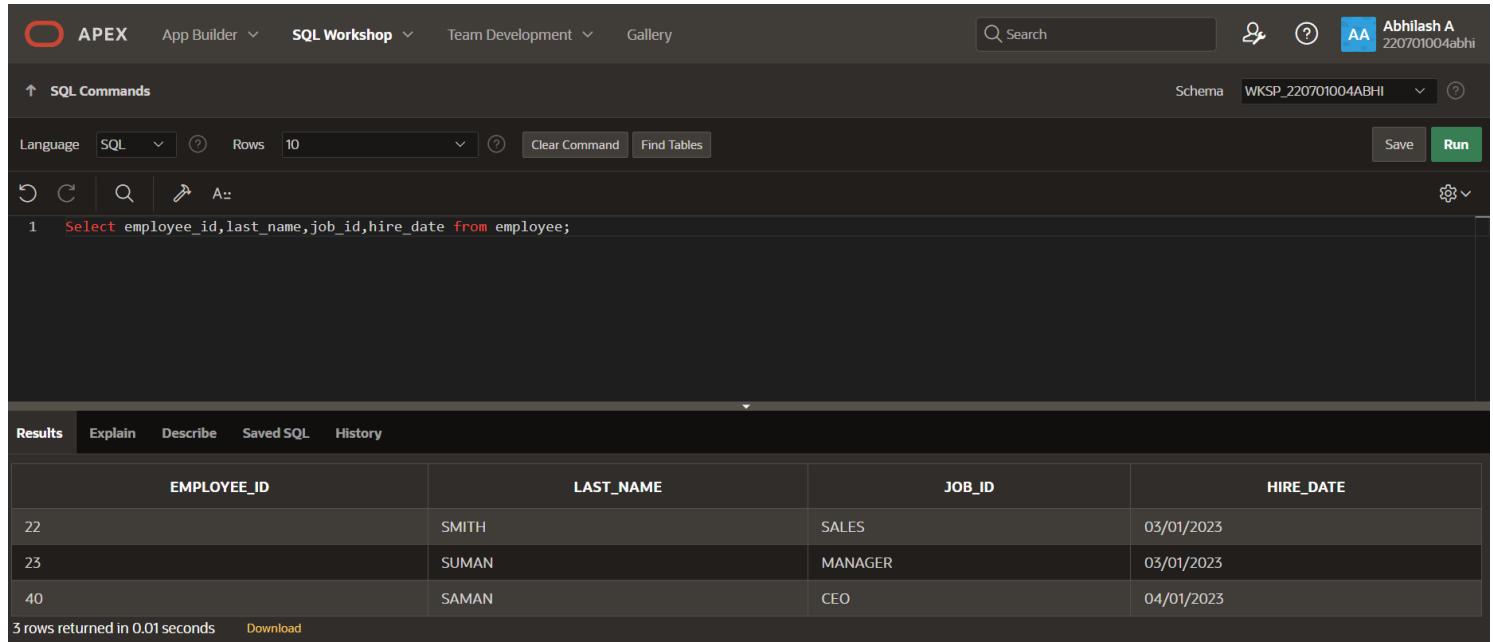
| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-------|--------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| DEPT | ID | NUMBER | - | 7 | 0 | 1 | - | - | - |
| | NAME | VARCHAR2 | 25 | - | - | - | ✓ | - | - |

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

QUERY:

```
Select employee_id, last_name, job_id, hire_date from employees;
```

OUTPUT:



| EMPLOYEE_ID | LAST_NAME | JOB_ID | HIRE_DATE |
|-------------|-----------|---------|------------|
| 22 | SMITH | SALES | 03/01/2023 |
| 23 | SUMAN | MANAGER | 03/01/2023 |
| 40 | SAMAN | CEO | 04/01/2023 |

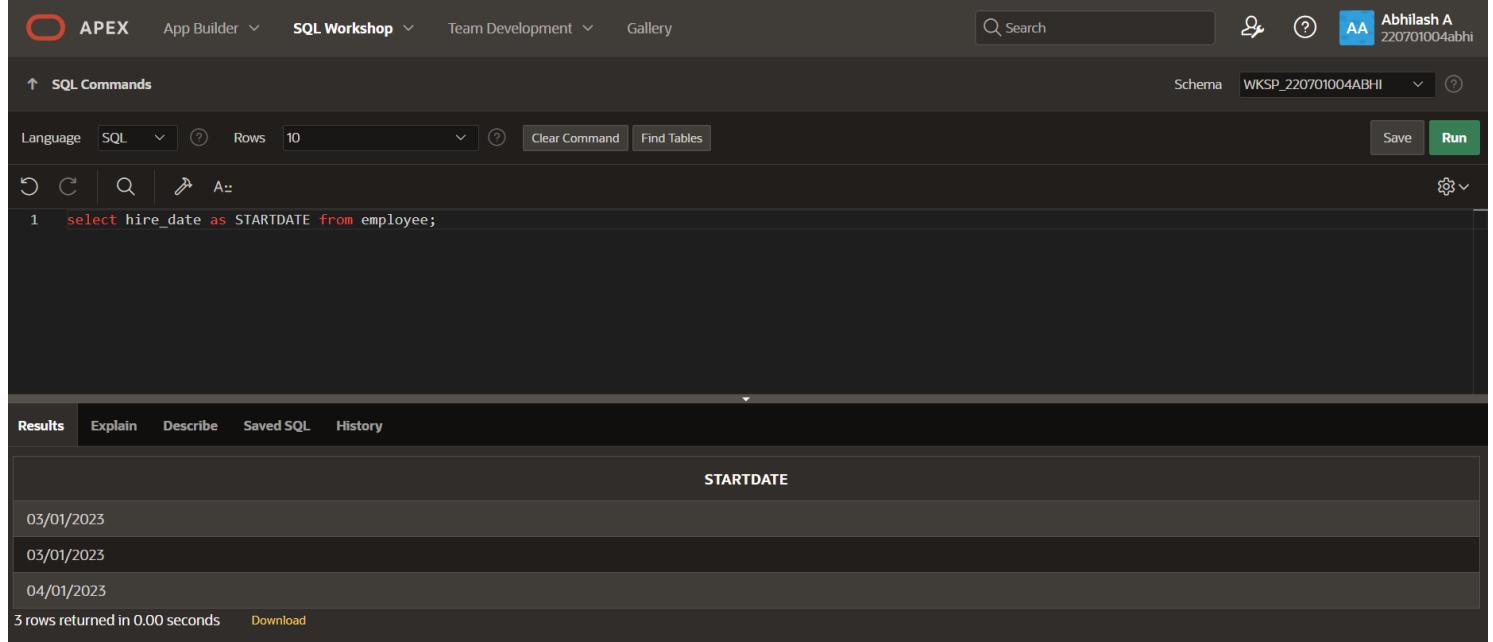
3 rows returned in 0.01 seconds [Download](#)

4. Provide an alias STARTDATE for the hire date.

QUERY:

```
select hire_date as "STARTDATE" from employees;
```

OUTPUT:



| STARTDATE |
|------------|
| 03/01/2023 |
| 03/01/2023 |
| 04/01/2023 |

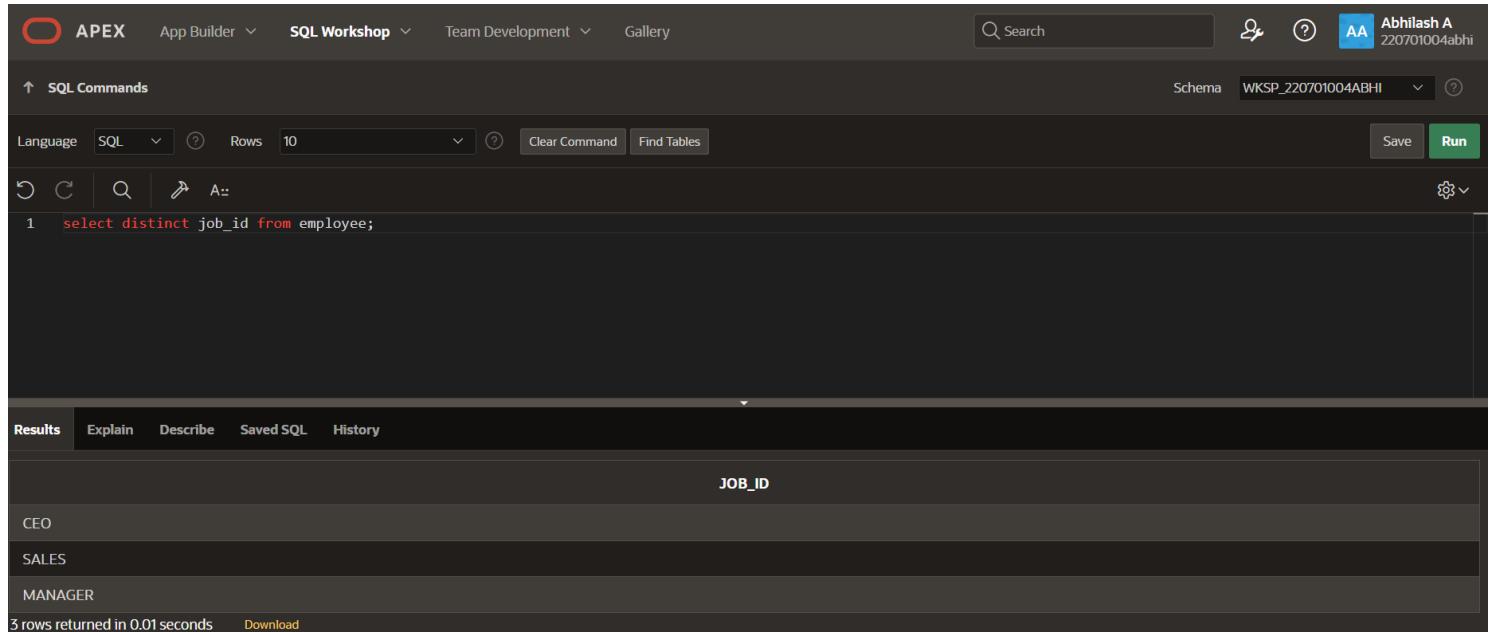
3 rows returned in 0.00 seconds [Download](#)

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

QUERY:

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

OUTPUT:



APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_220701004ABHI

Language SQL Rows 10 Clear Command Find Tables Save Run

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

Results Explain Describe Saved SQL History

| JOB_ID |
|---------|
| CEO |
| SALES |
| MANAGER |

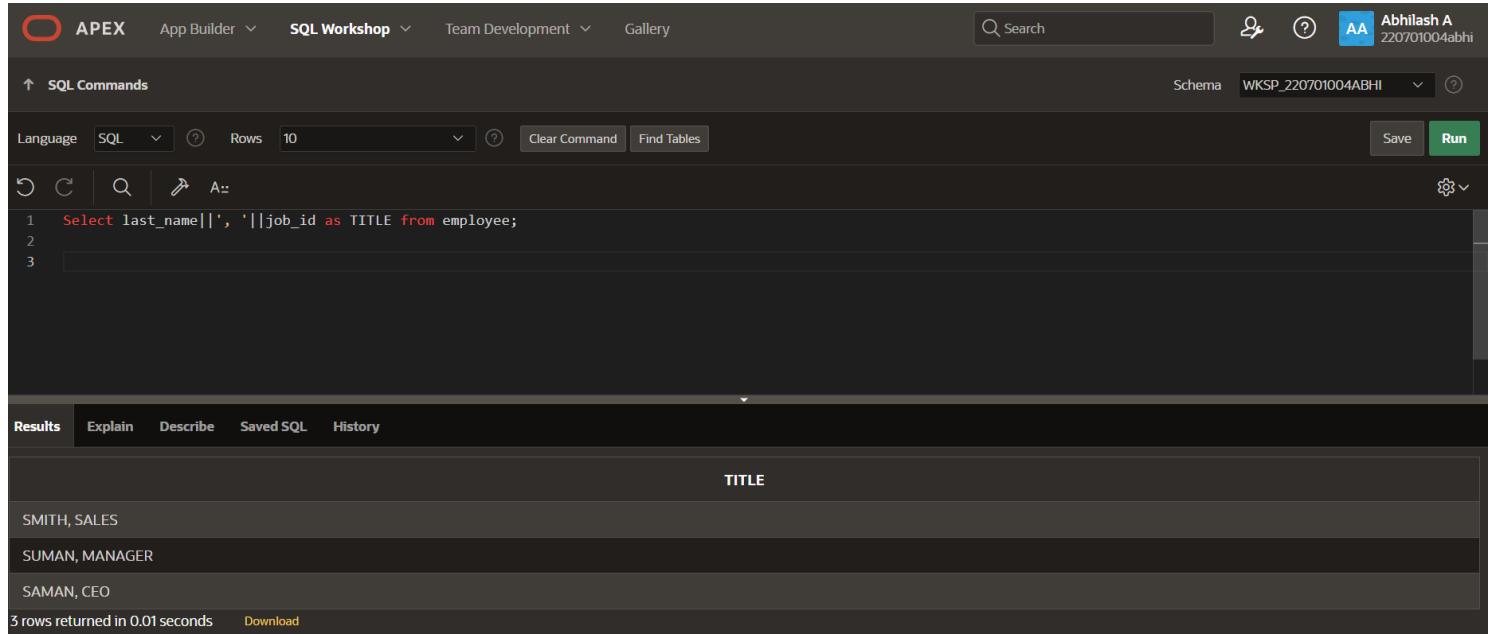
3 rows returned in 0.01 seconds Download

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

QUERY:

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

OUTPUT:



APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_220701004ABHI

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 Select last_name||', '||job_id as TITLE from employees;
```

Results Explain Describe Saved SQL History

| TITLE |
|----------------|
| SMITH, SALES |
| SUMAN, MANAGER |
| SAMAN, CEO |

3 rows returned in 0.01 seconds Download

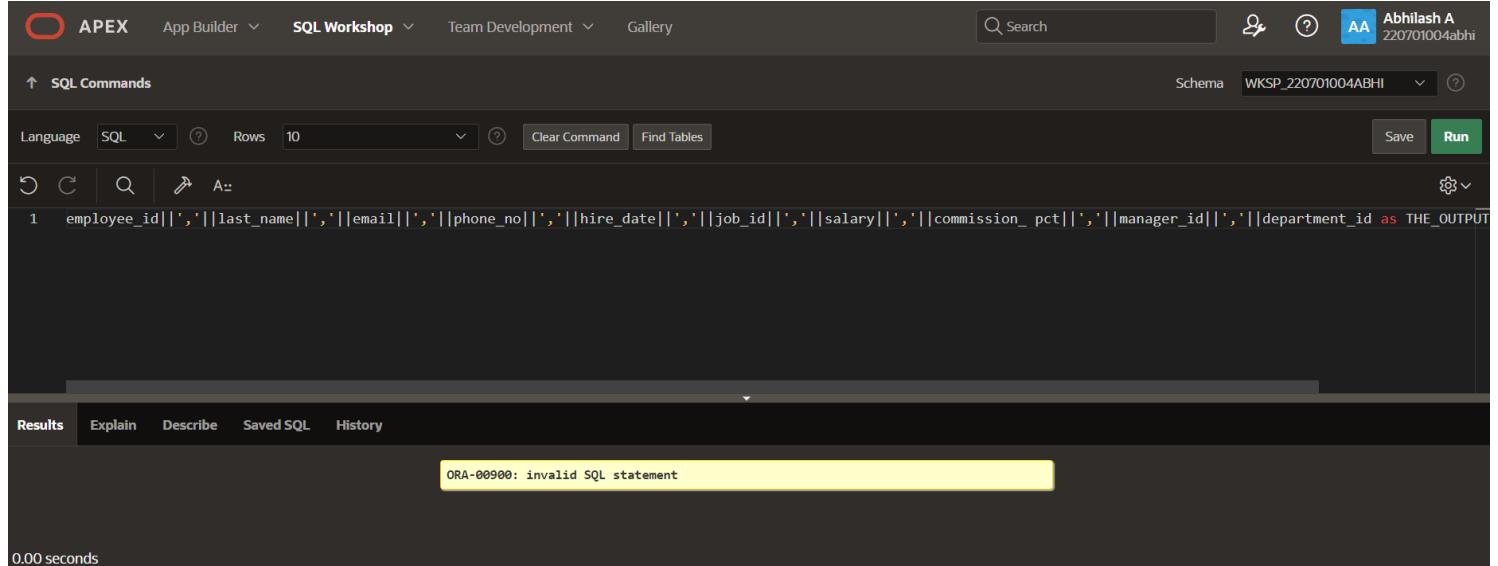
7.Create a query to display all the data from the employees table. Separate each column by a comma. Name the column THE_OUTPUT.

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:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' and contains a SQL editor with the following code:

```
1 employee_id||','||last_name||','||email||','||phone_no||','||hire_date||','||job_id||','||salary||','||commission_pct||','||manager_id||','||department_id as THE_OUTPUT
```

Below the editor, the 'Results' tab is selected. A yellow message box displays the error: 'ORA-00900: invalid SQL statement'. The execution time is listed as '0.00 seconds'.

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

RESULT:

RESTRICTING AND SORTING DATA

EX_NO:5

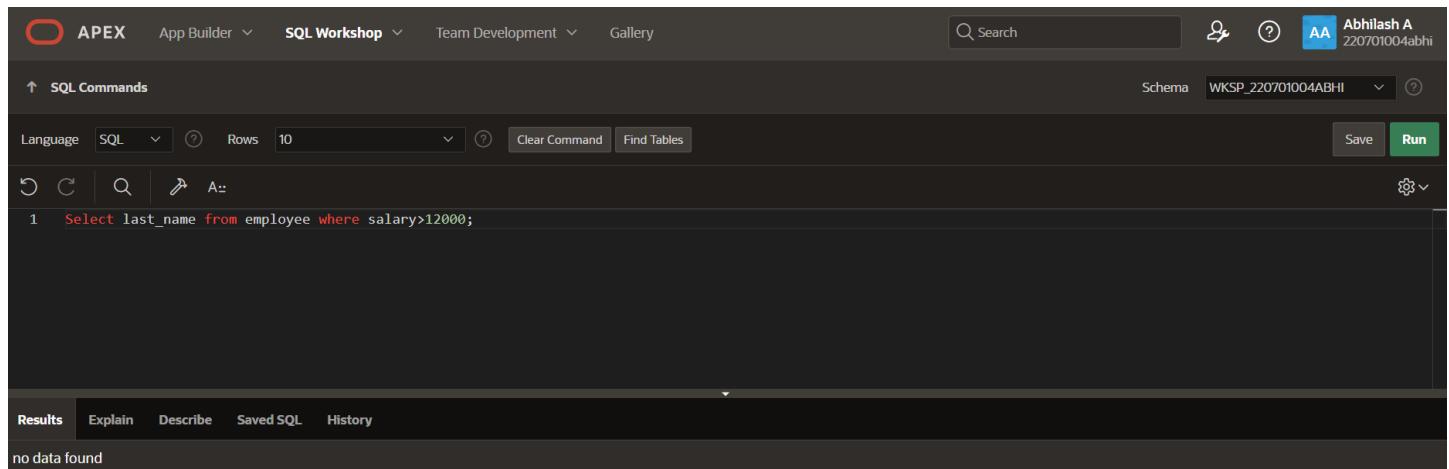
DATE: 07-03-2024

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

QUERY:

Select last_name from employees where salary>12000;

OUTPUT:



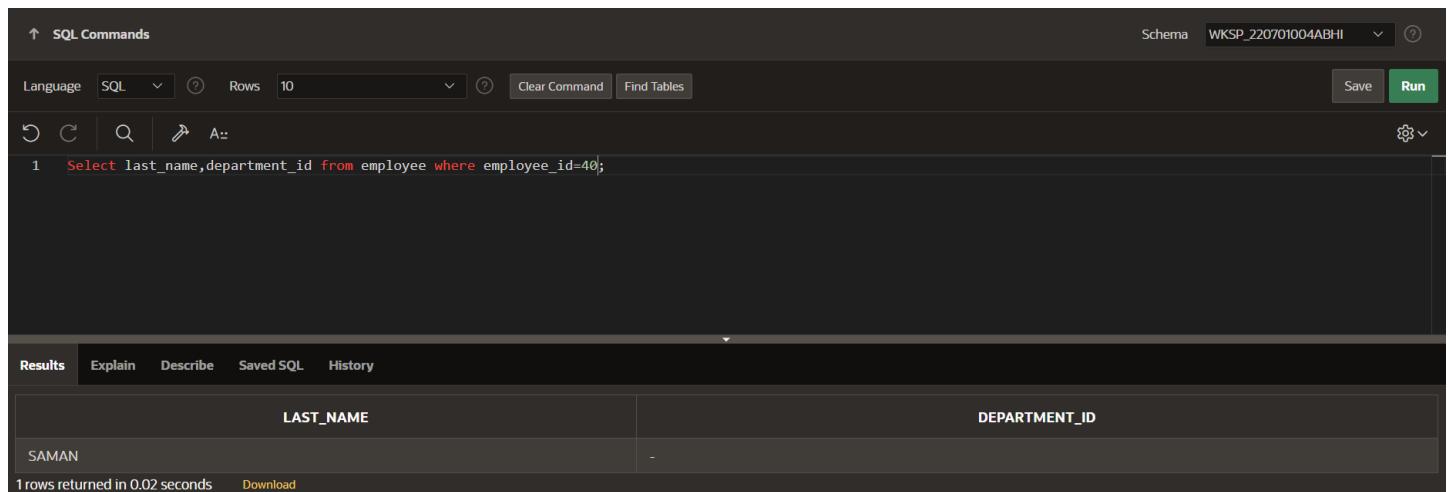
The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the query: `Select last_name from employees where salary>12000;`. The results tab shows the output: `no data found`.

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

QUERY:

Select last_name,department_id from employees where employee_id=176;

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the query: `Select last_name,department_id from employees where employee_id=176;`. The results tab shows the output:

| LAST_NAME | DEPARTMENT_ID |
|-----------|---------------|
| SAMAN | - |

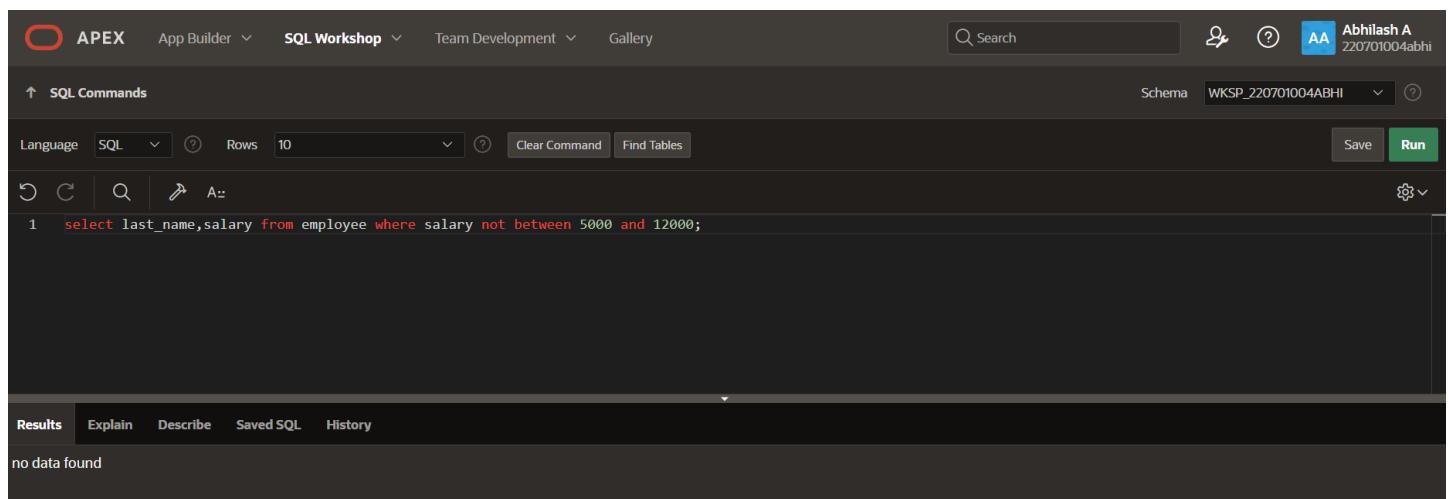
1 rows returned in 0.02 seconds [Download](#)

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

QUERY:

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

OUTPUT:



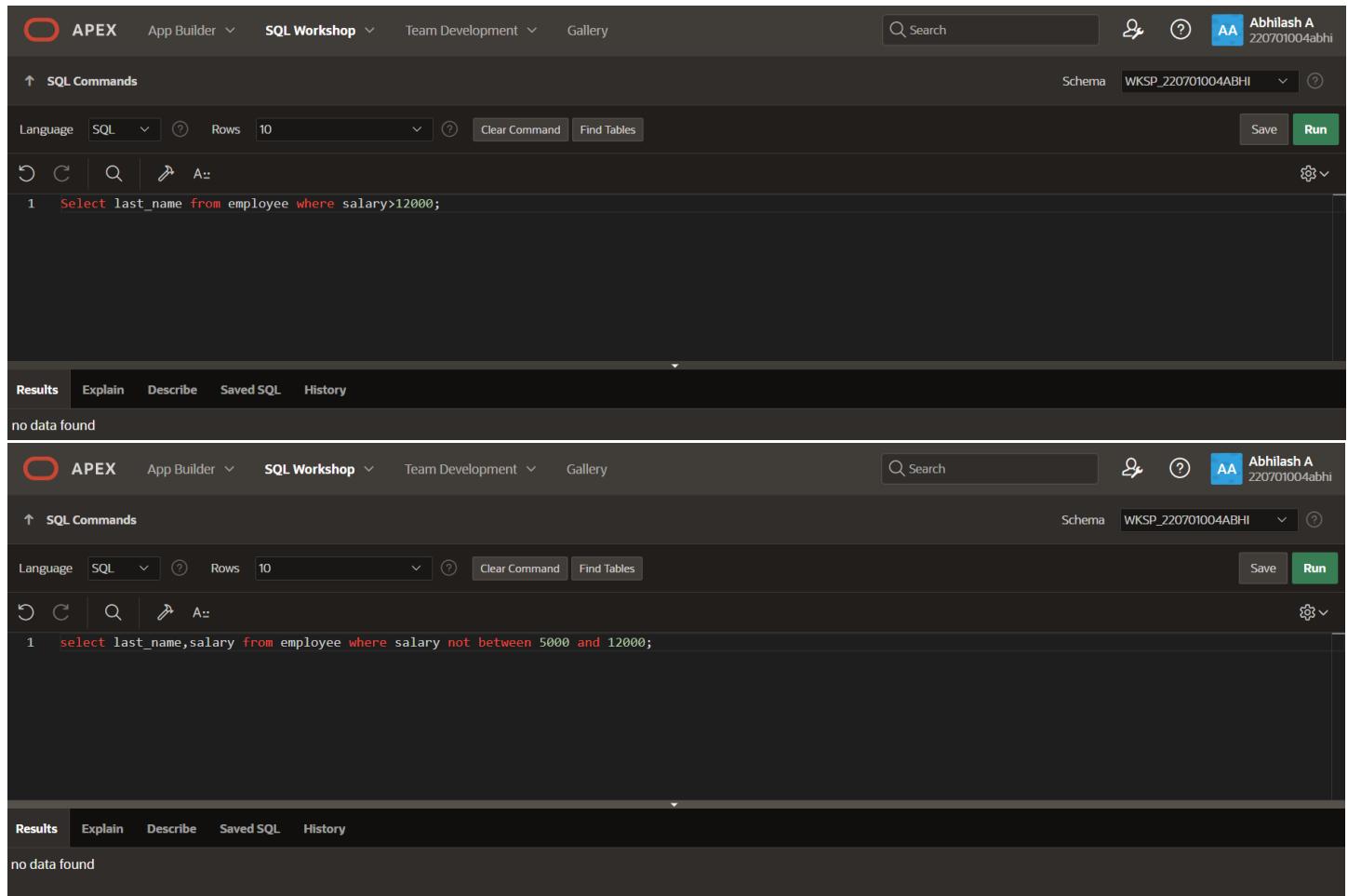
A screenshot of the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is selected. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. It shows a single line of SQL code: '1 select last_name,salary from employee where salary not between 5000 and 12000;'. Below the code, the 'Results' tab is selected, showing the message 'no data found'.

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

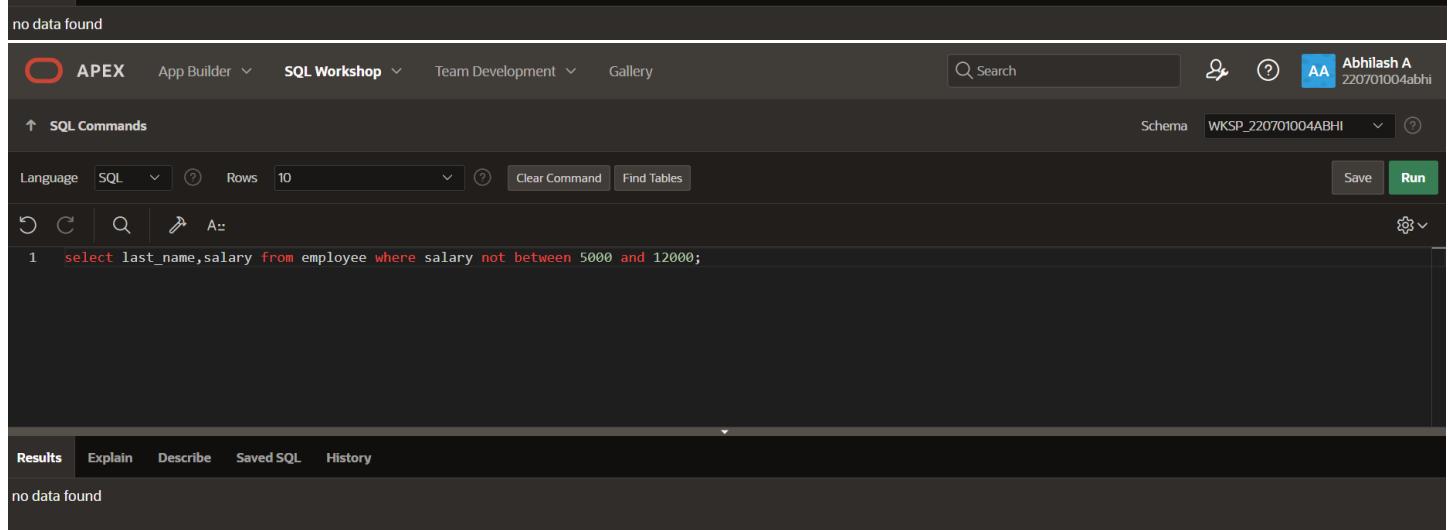
QUERY:

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

OUTPUT:



A screenshot of the Oracle SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar and user information are the same. The main area shows a single line of SQL code: '1 Select last_name from employee where salary>12000;'. Below the code, the 'Results' tab is selected, showing the message 'no data found'.



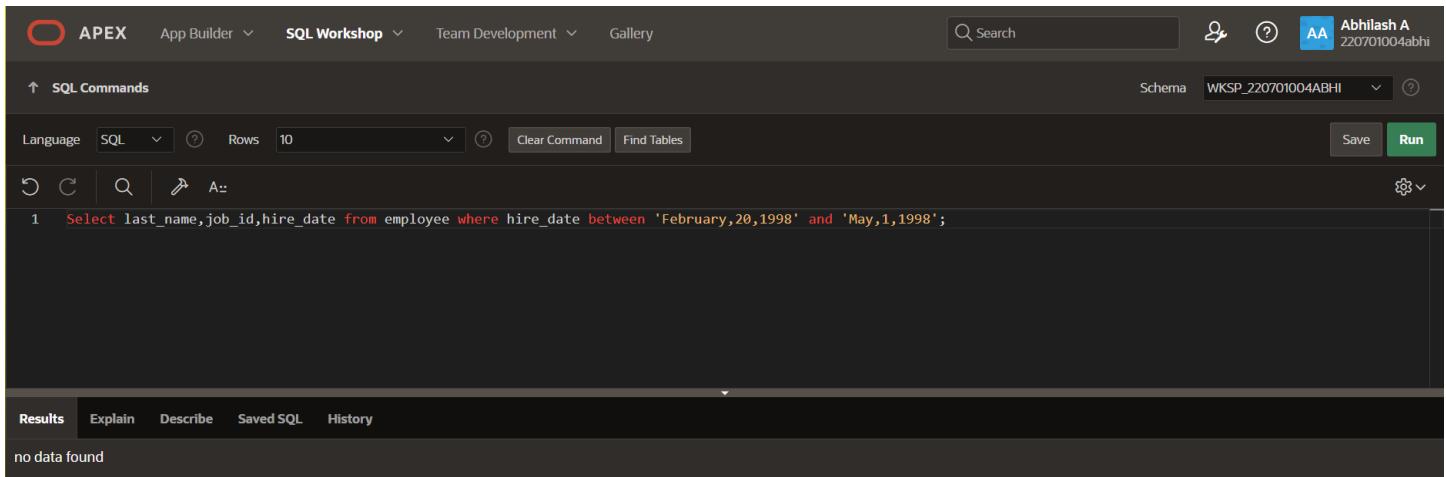
A second screenshot of the Oracle SQL Workshop interface, showing the same query as the first one: '1 select last_name,salary from employee where salary not between 5000 and 12000;'. The 'Results' tab is selected, showing the message 'no data found'.

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

QUERY:

```
select last_name,department_id from employees where department_id in(20,50) order by last_name;
```

OUTPUT:



Oracle SQL Workshop interface showing the query execution results. The query selected is:

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

The results tab shows:

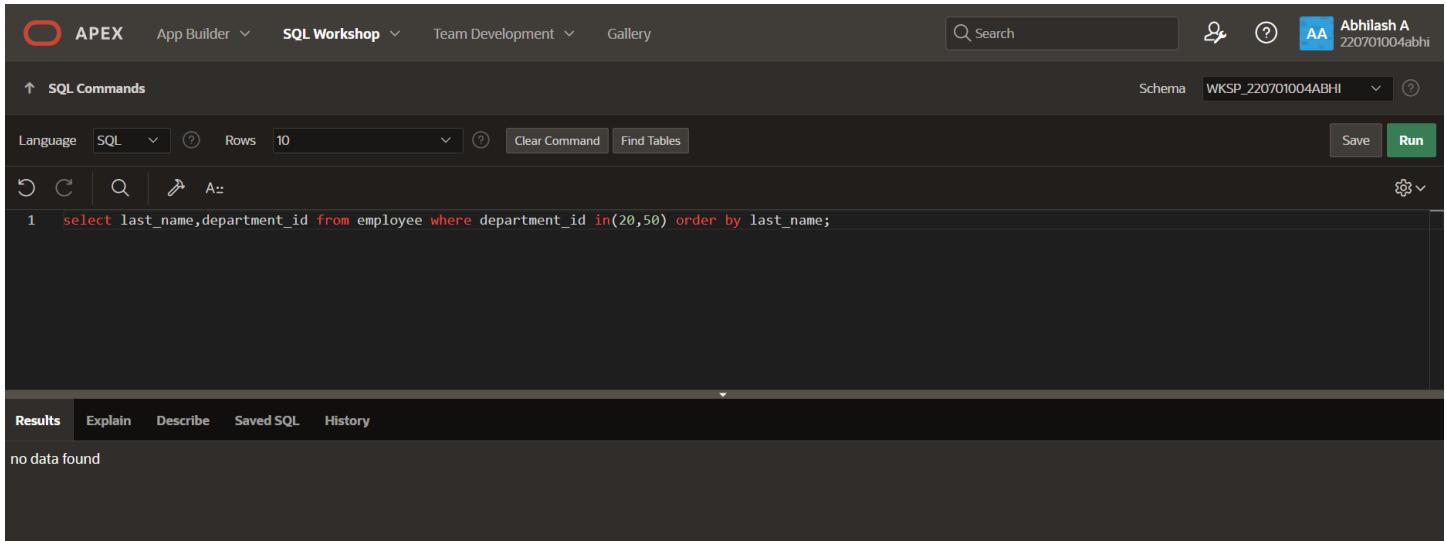
```
no data found
```

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

QUERY:

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

OUTPUT:



Oracle SQL Workshop interface showing the query execution results. The query selected is:

```
1  select last_name,department_id from employees where department_id in(20,50) order by last_name;
```

The results tab shows:

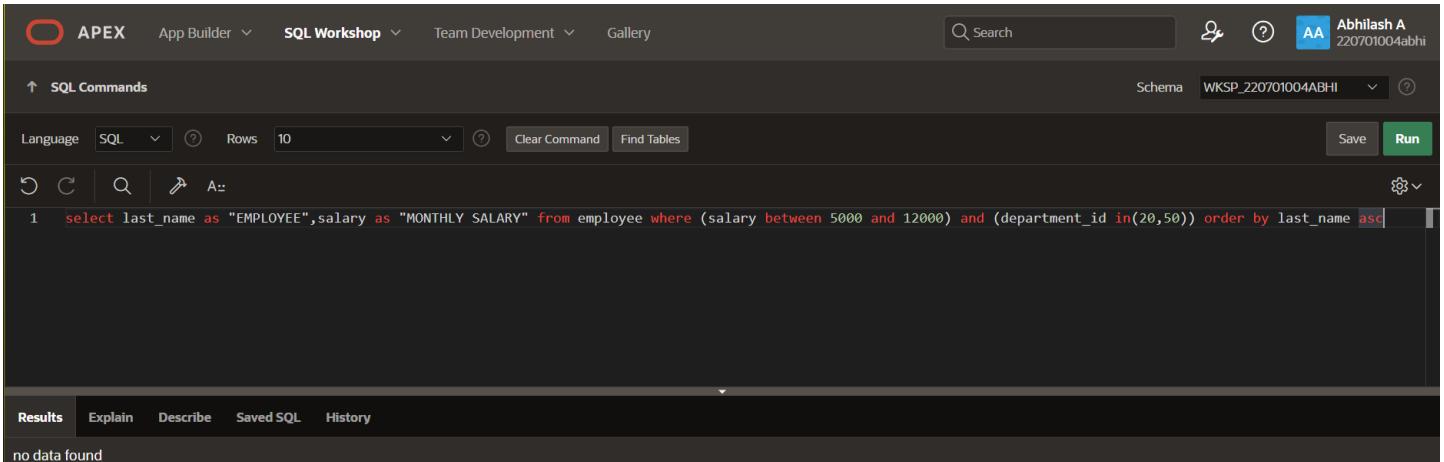
```
no data found
```

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

QUERY:

```
select last_name,hire_date from employees where hire_date like '1994';
```

OUTPUT:



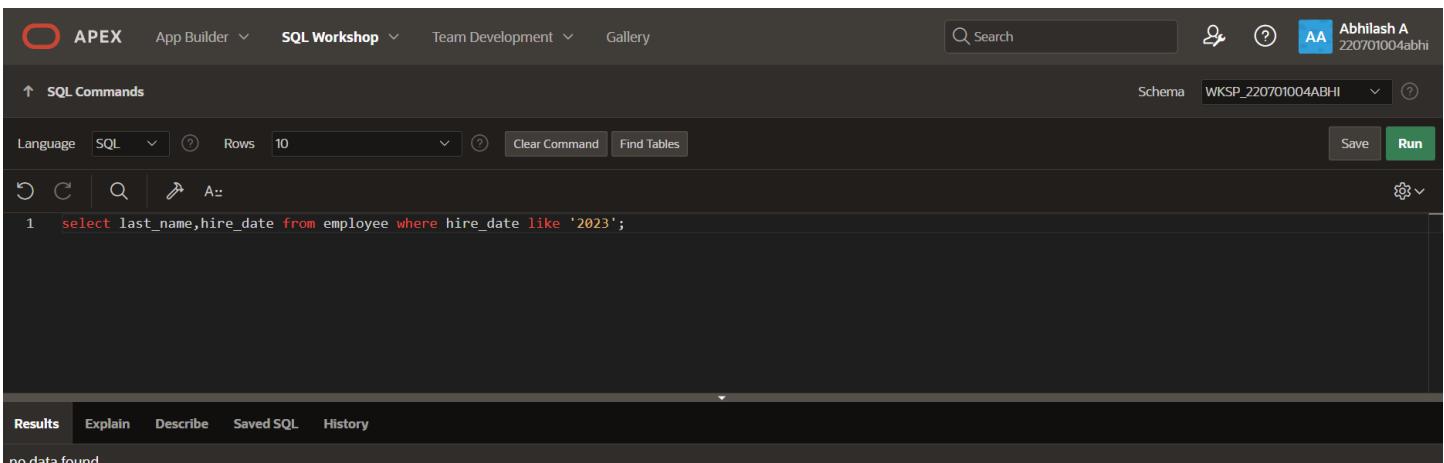
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows a user profile for 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. It shows a SQL command: 'select last_name as "EMPLOYEE",salary as "MONTHLY SALARY" from employee where (salary between 5000 and 12000) and (department_id in(20,50)) order by last_name asc'. Below the command is a results section with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, and the message 'no data found' is displayed.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows a user profile for 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. It shows a SQL command: 'select last_name,hire_date from employee where hire_date like '2023''. Below the command is a results section with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, and the message 'no data found' is displayed.

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Schema: WKSP_220701004ABHI

SQL Commands

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

```
1 select last_name, job_id from employee where manager_id is null;
```

Results Explain Describe Saved SQL History

| LAST_NAME | JOB_ID |
|-----------|---------|
| SAMAN | CEO |
| mohan | account |
| SMITH | SALES |
| SUMAN | MANAGER |

4 rows returned in 0.02 seconds Download

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Schema: WKSP_220701004ABHI

SQL Commands

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

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

Results Explain Describe Saved SQL History

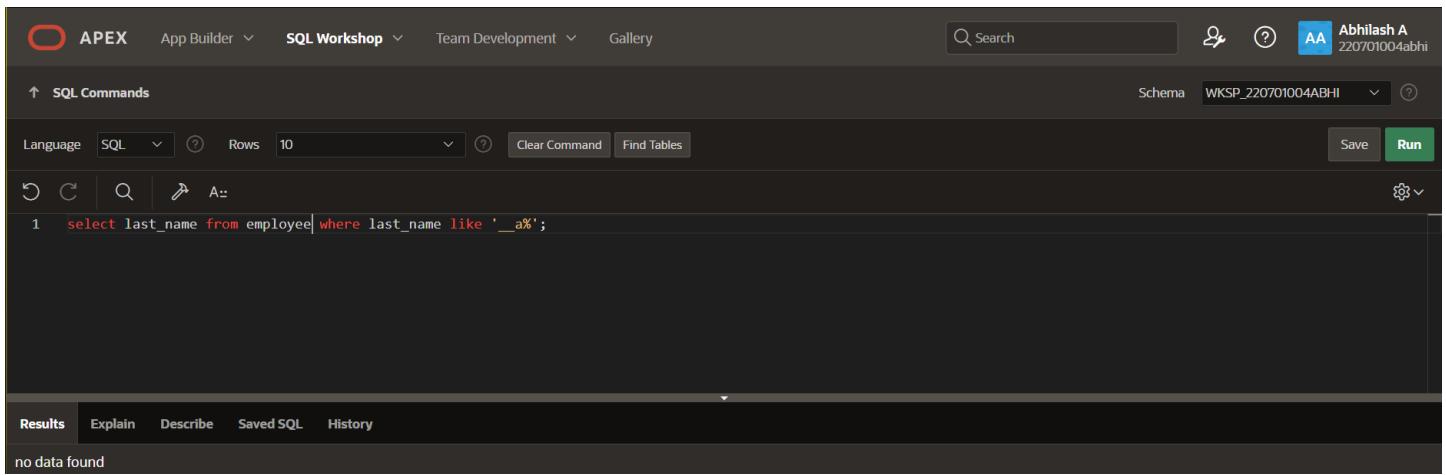
no data found

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

QUERY:

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

OUTPUT:



A screenshot of the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and a user profile for 'Abhilash A' are also present. The main workspace shows a SQL command line with the following query:

```
1 select last_name from employee| where last_name like '__a%';
```

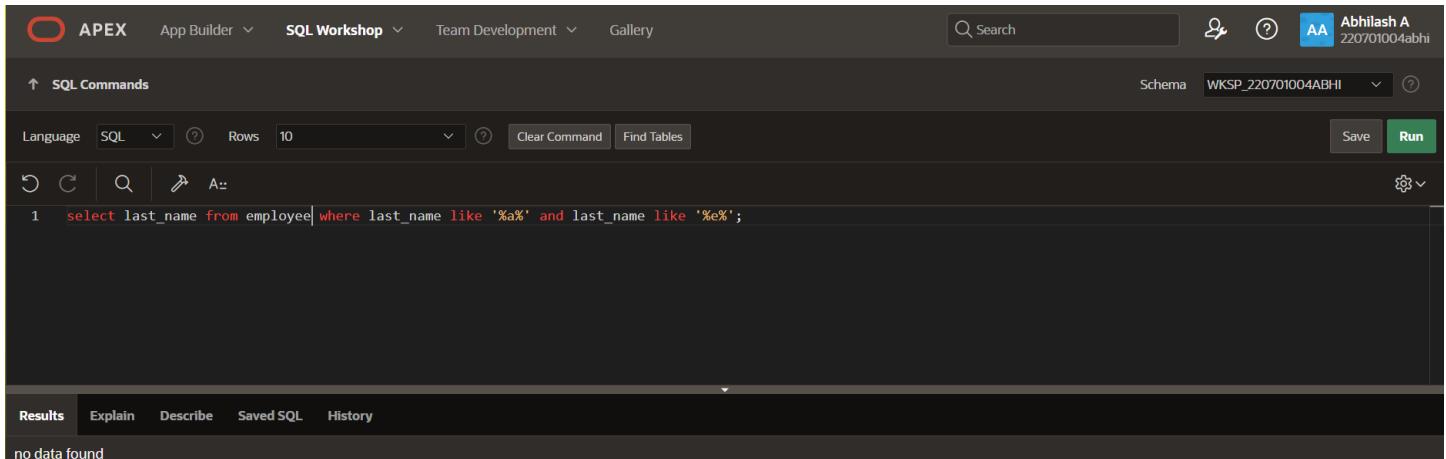
The results tab is selected, showing the message "no data found".

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

QUERY:

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

OUTPUT:



A screenshot of the Oracle SQL Workshop interface, similar to the previous one but with a different query. The main workspace shows the following query:

```
1 select last_name from employee| where last_name like '%a%' and last_name like '%e%';
```

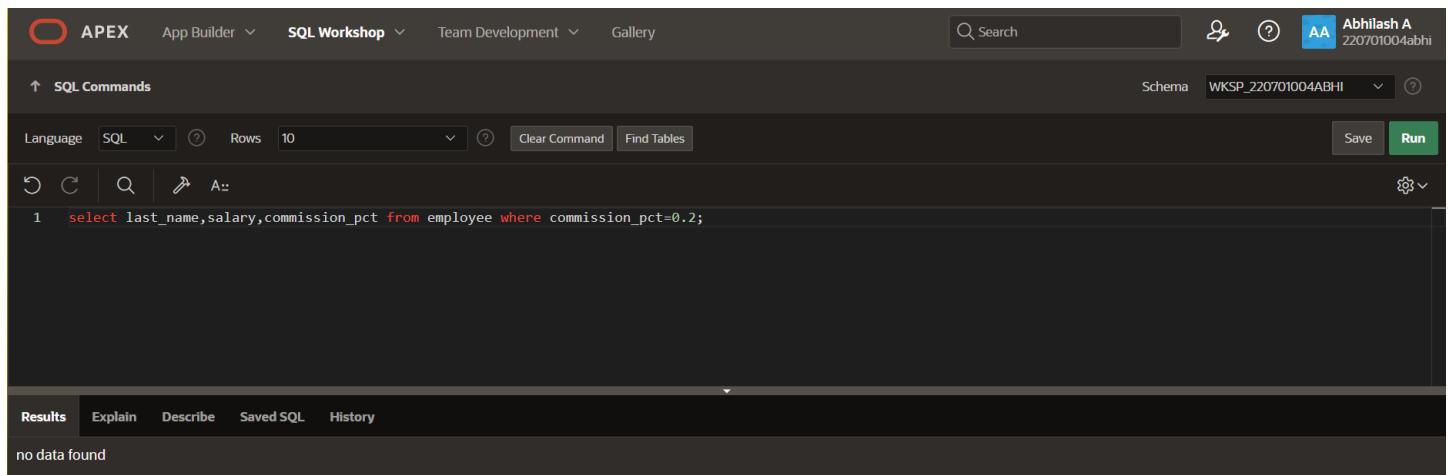
The results tab is selected, showing the message "no data found".

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

QUERY:

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

OUTPUT:



A screenshot of the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile for 'Abhilash A' are also present. The main workspace is titled 'SQL Commands' and contains a SQL editor with the following content:

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

The editor includes buttons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run. Below the editor, a toolbar has icons for Undo, Redo, Search, and Paste. The bottom navigation bar shows tabs for Results (selected), Explain, Describe, Saved SQL, and History. The results pane displays the message "no data found".

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

RESULT:

SINGLE ROW FUNCTIONS

EX_NO:6

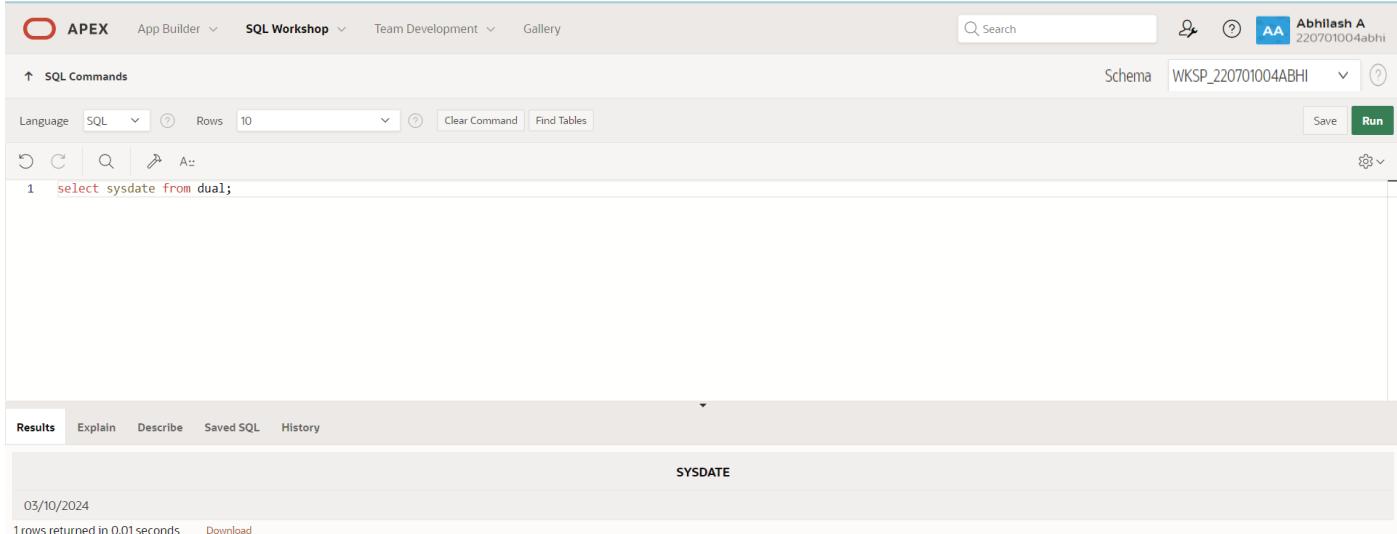
DATE:09-03-2024

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

QUERY:

```
select sysdate from dual;
```

OUTPUT:



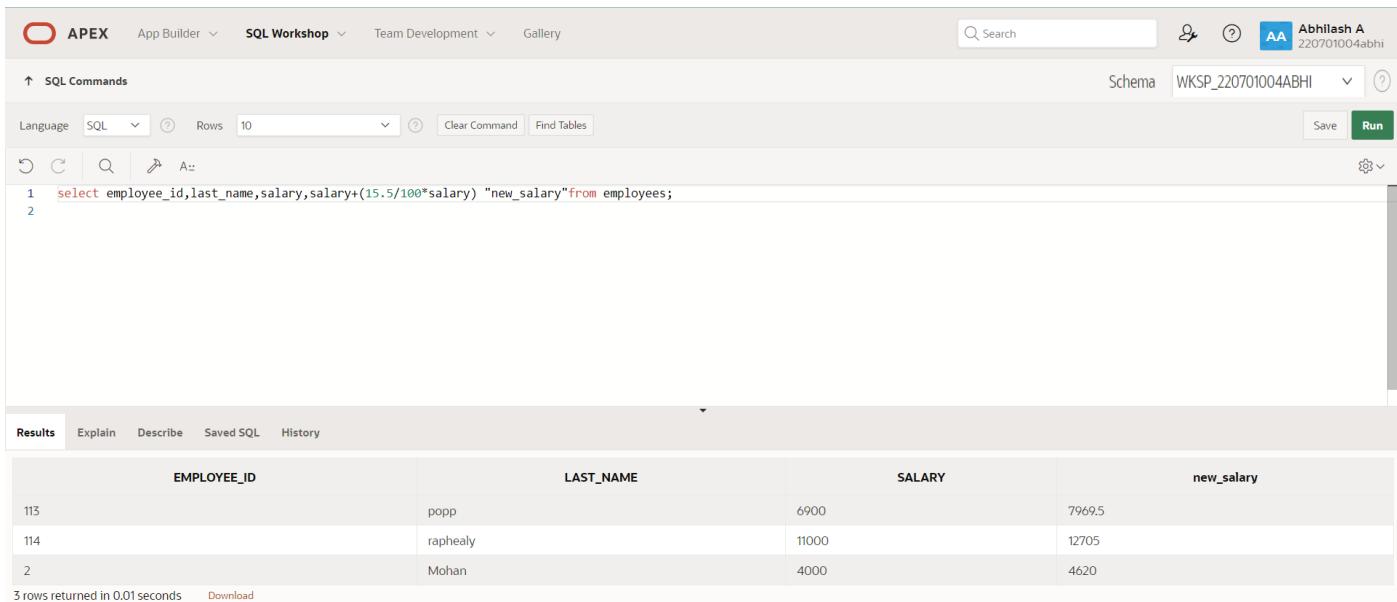
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a user profile for 'Abhilash A' (220701004abhi). The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_220701004ABHI'. The query editor contains the command: 'select sysdate from dual;'. The results tab is selected, showing the output: '03/10/2024'. Below the results, it says '1 rows returned in 0.01 seconds' and has a 'Download' link.

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:



APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_220701004ABHI

Language SQL Rows 10 Clear Command Find Tables Save Run

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

Results Explain Describe Saved SQL History

| EMPLOYEE_ID | LAST_NAME | SALARY | new_salary |
|-------------|-----------|--------|------------|
| 113 | popp | 6900 | 7969.5 |
| 114 | raphealy | 11000 | 12705 |
| 2 | Mohan | 4000 | 4620 |

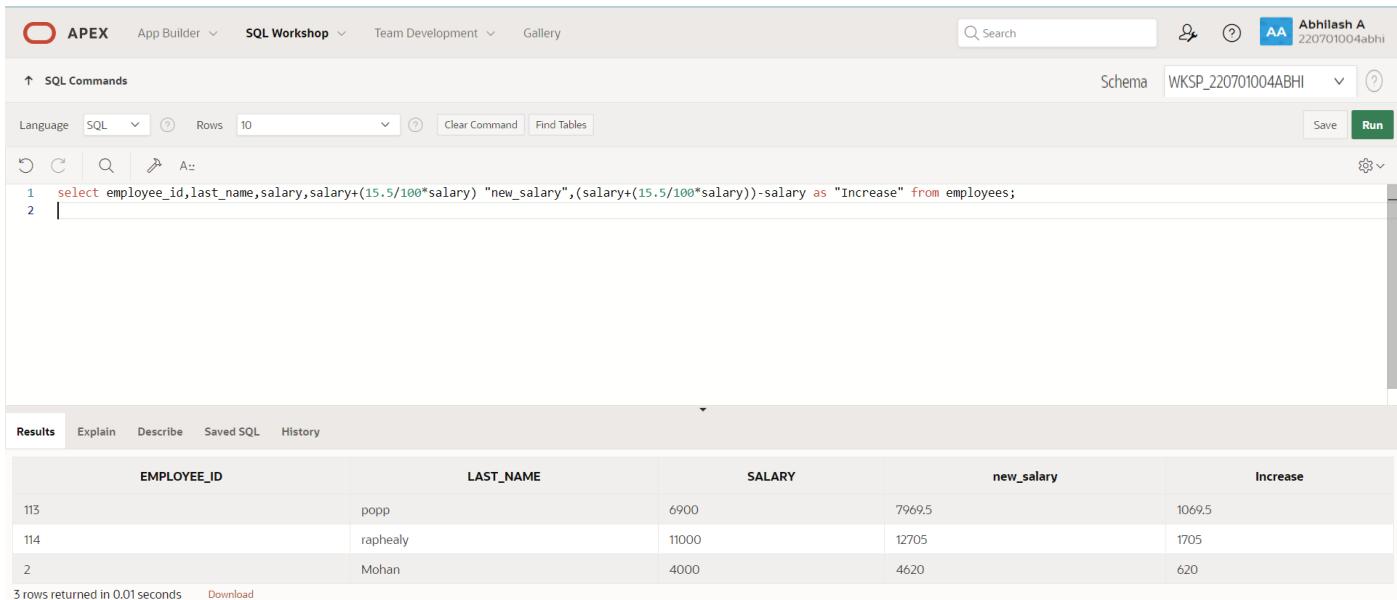
3 rows returned in 0.01 seconds [Download](#)

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:



APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_220701004ABHI

Language SQL Rows 10 Clear Command Find Tables Save Run

1 select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary", (salary+(15.5/100*salary))-salary as "Increase" from employees;
 2

Results Explain Describe Saved SQL History

| EMPLOYEE_ID | LAST_NAME | SALARY | new_salary | Increase |
|-------------|-----------|--------|------------|----------|
| 113 | popp | 6900 | 7969.5 | 1069.5 |
| 114 | raphealy | 11000 | 12705 | 1705 |
| 2 | Mohan | 4000 | 4620 | 620 |

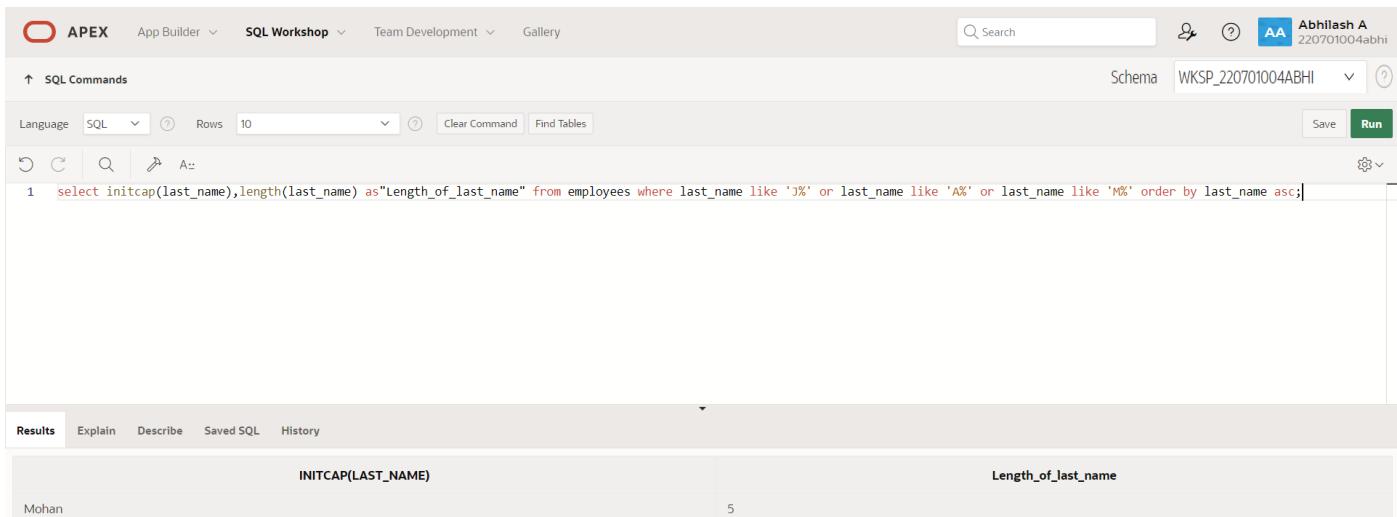
3 rows returned in 0.01 seconds [Download](#)

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

QUERY:

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

OUTPUT:



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

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

The results table has two columns: INITCAP(LAST_NAME) and Length_of_last_name. The data shows one row for Mohan, with a length of 5.

| INITCAP(LAST_NAME) | Length_of_last_name |
|--------------------|---------------------|
| Mohan | 5 |

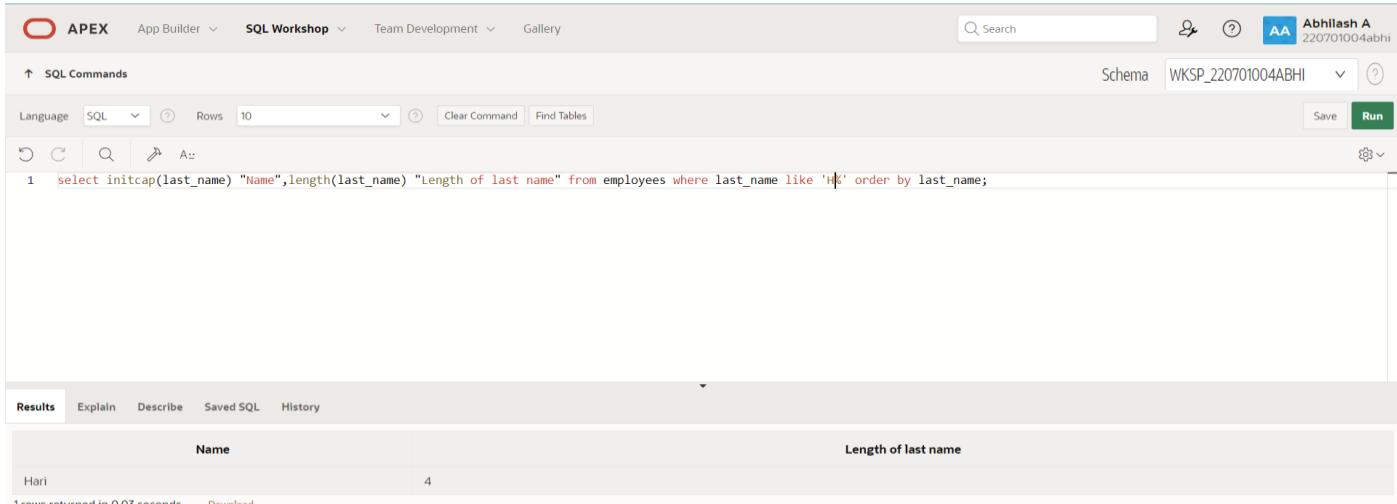
1 rows returned in 0.01 seconds [Download](#)

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

QUERY:

```
select initcap(last_name) "Name",length(last_name) "Length of last name" from employees where last_name like 'H%' order by last_name;
```

OUTPUT:



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

```
1 select initcap(last_name) "Name",length(last_name) "Length of last name" from employees where last_name like 'H%' order by last_name;
```

The results table has two columns: Name and Length of last name. The data shows one row for Hari, with a length of 4.

| Name | Length of last name |
|------|---------------------|
| Hari | 4 |

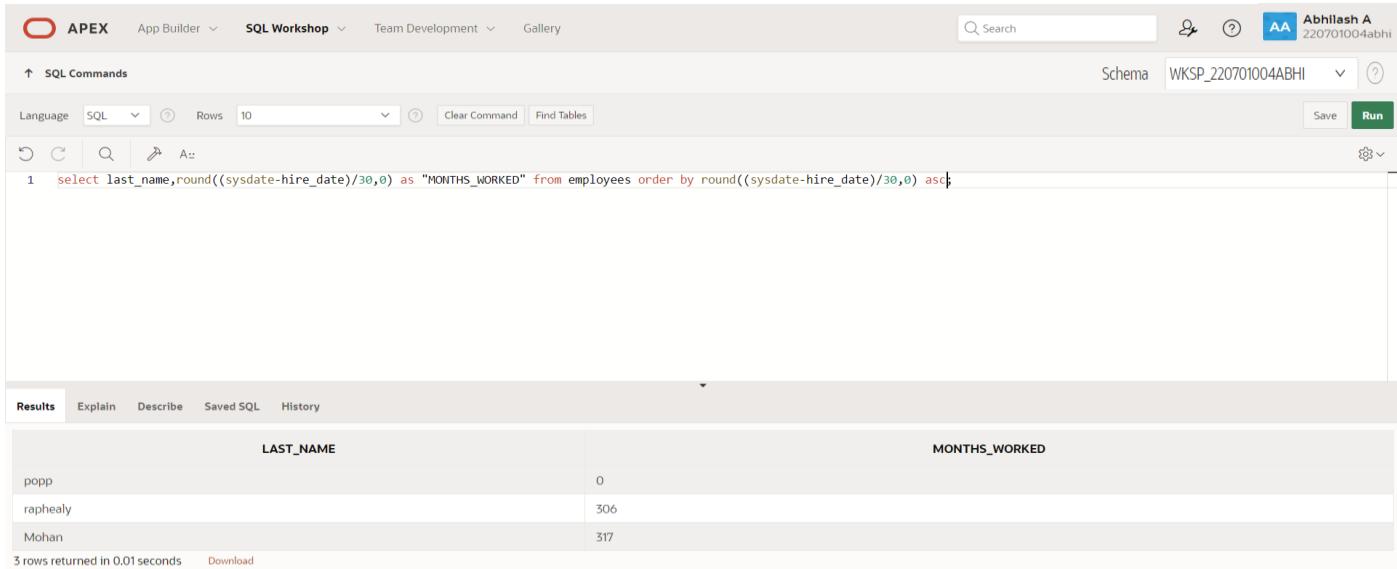
1 rows returned in 0.03 seconds [Download](#)

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 by round((sysdate-hire_date)/30,0) asc;
```

OUTPUT:



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

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

The results table has two columns: LAST_NAME and MONTHS_WORKED. The data is:

| LAST_NAME | MONTHS_WORKED |
|-----------|---------------|
| popp | 0 |
| raphealy | 306 |
| Mohan | 317 |

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

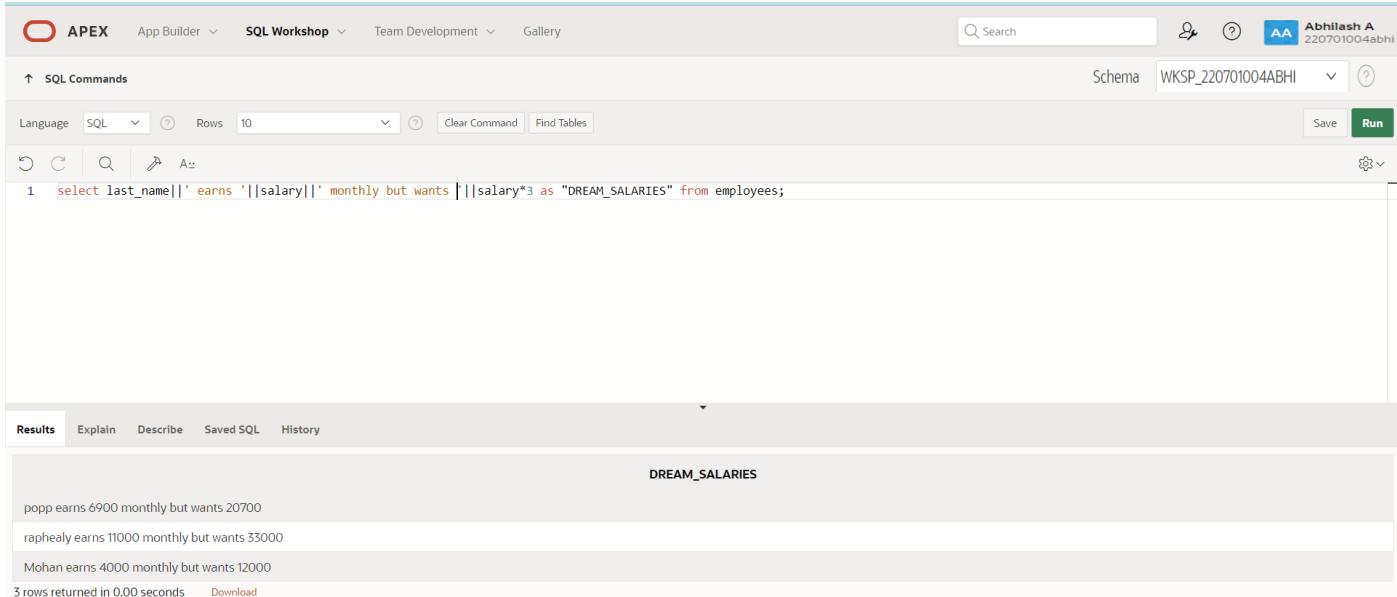
7. Create a report that produces the following for each employee:

<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:



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

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

The results table has one column: DREAM_SALARIES. The data is:

| DREAM_SALARIES |
|--|
| popp earns 6900 monthly but wants 20700 |
| raphealy earns 11000 monthly but wants 33000 |
| Mohan earns 4000 monthly but wants 12000 |

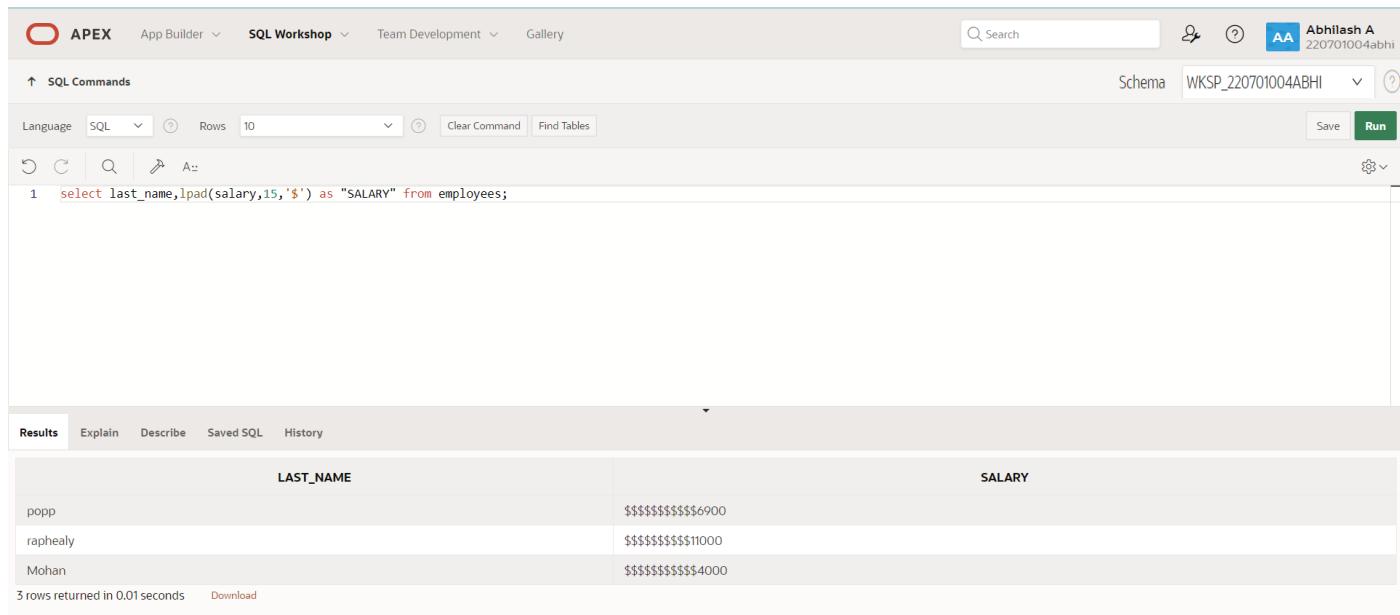
3 rows returned in 0.00 seconds [Download](#)

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004abhi'. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701004abhi'. The query entered is: `1 select last_name,lpad(salary,15,'$') as "SALARY" from employees;`. The results tab is selected, showing the output:

| LAST_NAME | SALARY |
|-----------|---------------------------|
| popp | \$\$\$\$\$\$\$\$\$\$6900 |
| raphealy | \$\$\$\$\$\$\$\$\$\$11000 |
| Mohan | \$\$\$\$\$\$\$\$\$\$4000 |

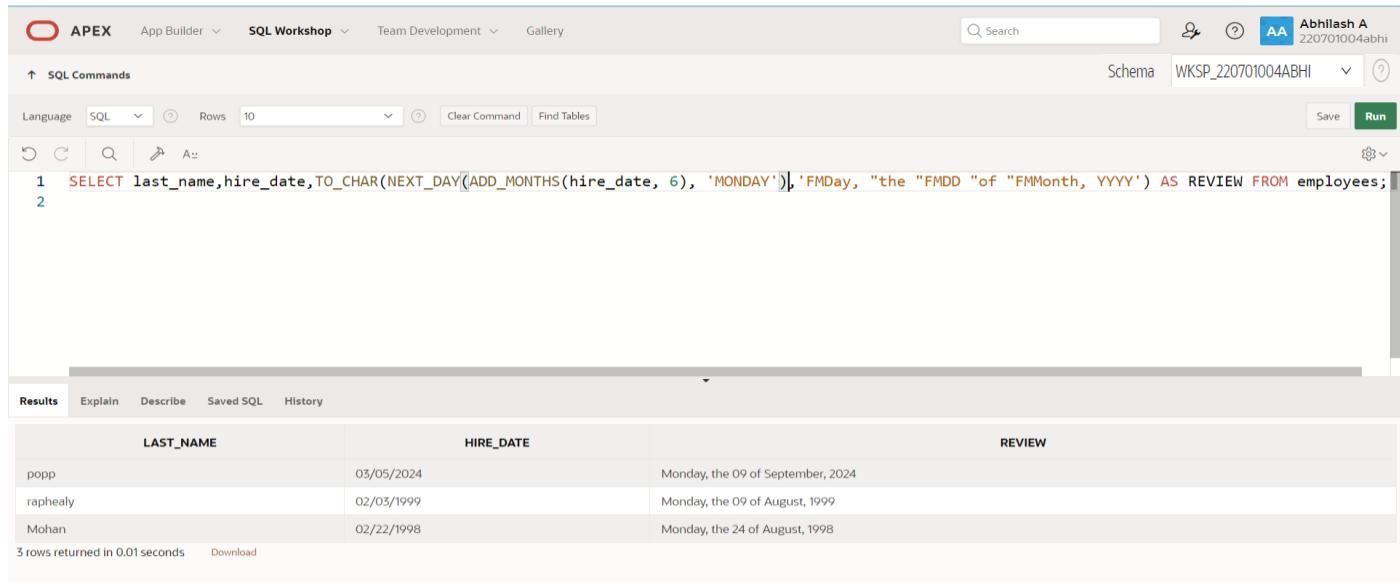
3 rows returned in 0.01 seconds [Download](#)

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004abhi'. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701004abhi'. The query entered is: `1 SELECT last_name,hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY'),'FMDay, "the "FMDD "of "FMMonth, YYYY') AS REVIEW FROM employees;`. The results tab is selected, showing the output:

| LAST_NAME | HIRE_DATE | REVIEW |
|-----------|------------|-----------------------------------|
| popp | 03/05/2024 | Monday, the 09 of September, 2024 |
| raphealy | 02/03/1999 | Monday, the 09 of August, 1999 |
| Mohan | 02/22/1998 | Monday, the 24 of August, 1998 |

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

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:

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

RESULT:

DISPLAYING DATA FROM MULTIPLE TABLES

EX_NO:7

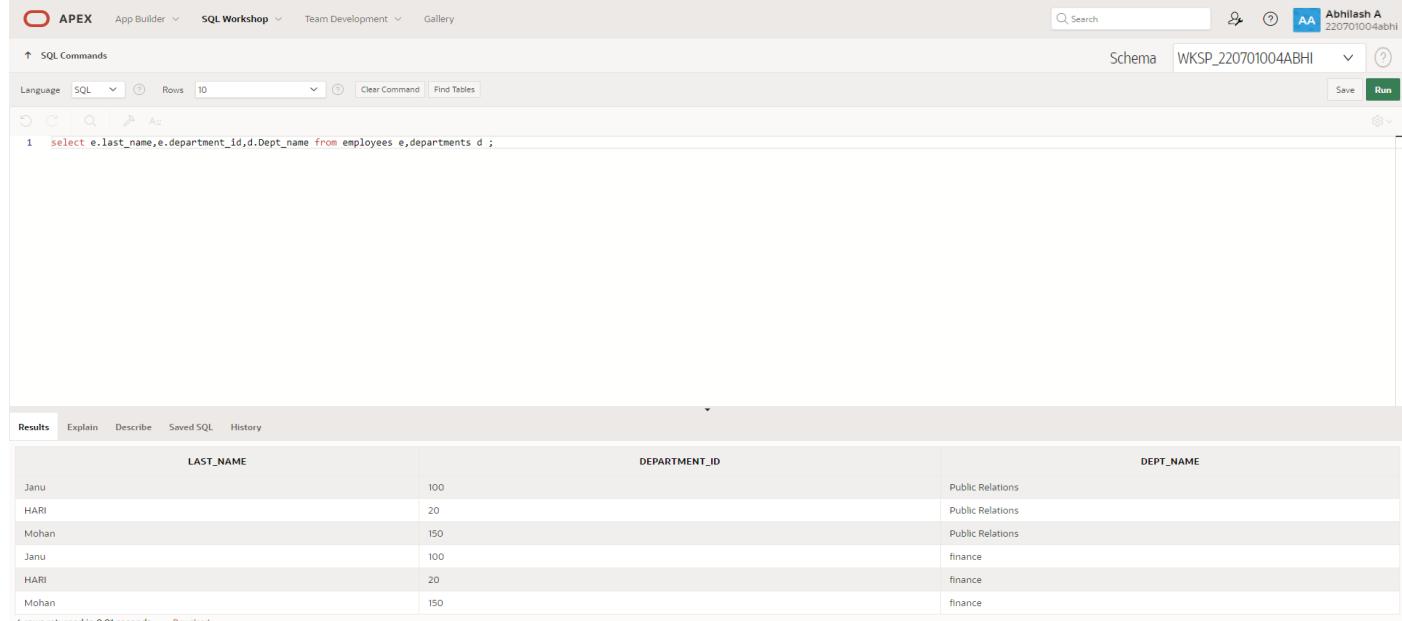
DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The top right shows the user's name, Abhilash A, and session ID. The main area has tabs for SQL Commands, SQL (selected), and Rows (10). The SQL editor contains the query: `1 select e.last_name,e.department_id,d.Dept_name from employees e,departments d ;`. The results tab shows the following data:

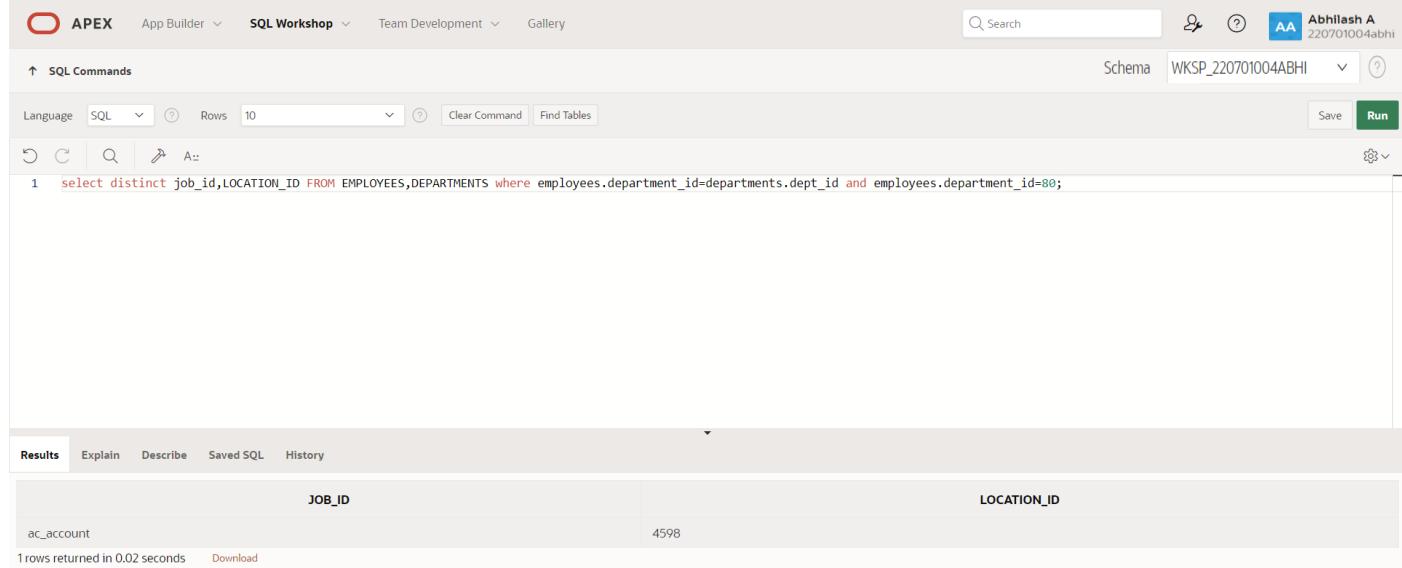
| LAST_NAME | DEPARTMENT_ID | DEPT_NAME |
|-----------|---------------|------------------|
| Janu | 100 | Public Relations |
| HARI | 20 | Public Relations |
| Mohan | 150 | Public Relations |
| Janu | 100 | finance |
| HARI | 20 | finance |
| Mohan | 150 | finance |

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

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_number=d.dept_id and e.department_number=80;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a 'Run' button. The query entered is:

```
1 select distinct job_id,LOCATION_ID FROM EMPLOYEES,DEPARTMENTS where employees.department_id=departments.dept_id and employees.department_id=80;
```

The results section shows a single row:

| JOB_ID | LOCATION_ID |
|------------|-------------|
| ac_account | 4598 |

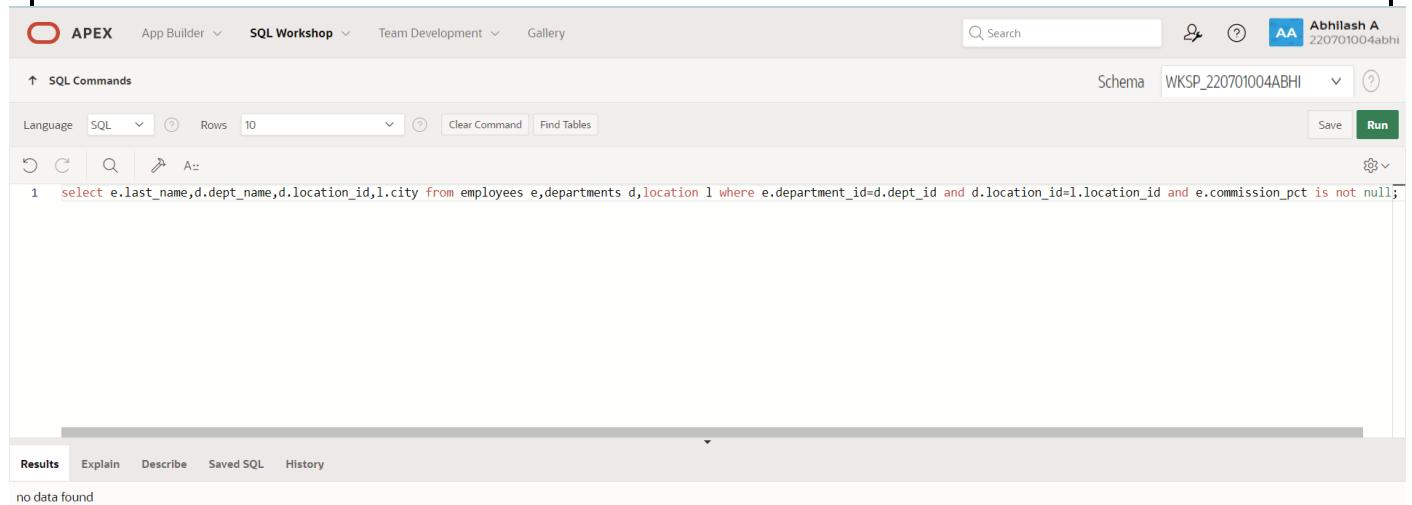
1 rows returned in 0.02 seconds [Download](#)

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile for 'Abhilash A' are also present. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_220701004ABHI'. The query editor contains the following SQL code:

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

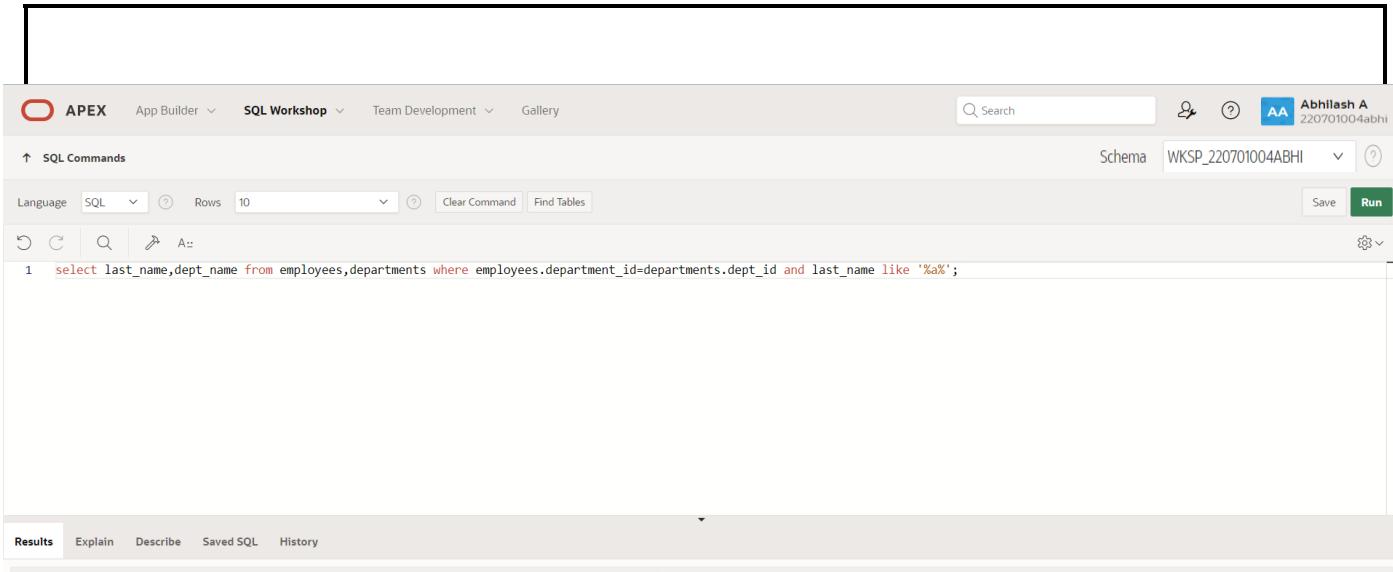
The results tab is selected, showing the message 'no data found'.

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,departments where employees.department_number=departments.dept_id and last_name like '%a%';
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there is a search bar, a user profile for 'Abhilash A' (220701004abhi), and a 'Schema' dropdown set to 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The query editor contains the following SQL code:

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

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

| LAST_NAME | DEPT_NAME |
|-----------|------------------|
| Janu | Public Relations |

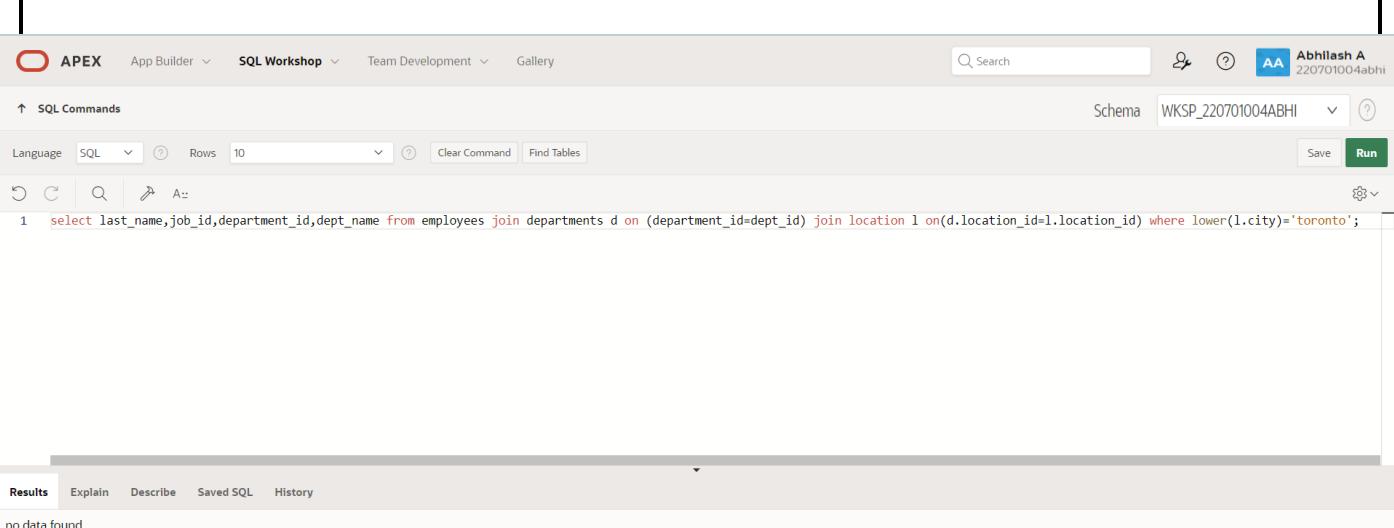
1 rows returned in 0.01 seconds [Download](#)

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a user profile for 'Abhilash A' are on the right. The main area is a query editor with the following content:

```
1 select last_name,job_id,department_id,dept_name from employees join departments d on (department_id=dept_id) join location l on(d.location_id=l.location_id) where lower(l.city)='toronto';
```

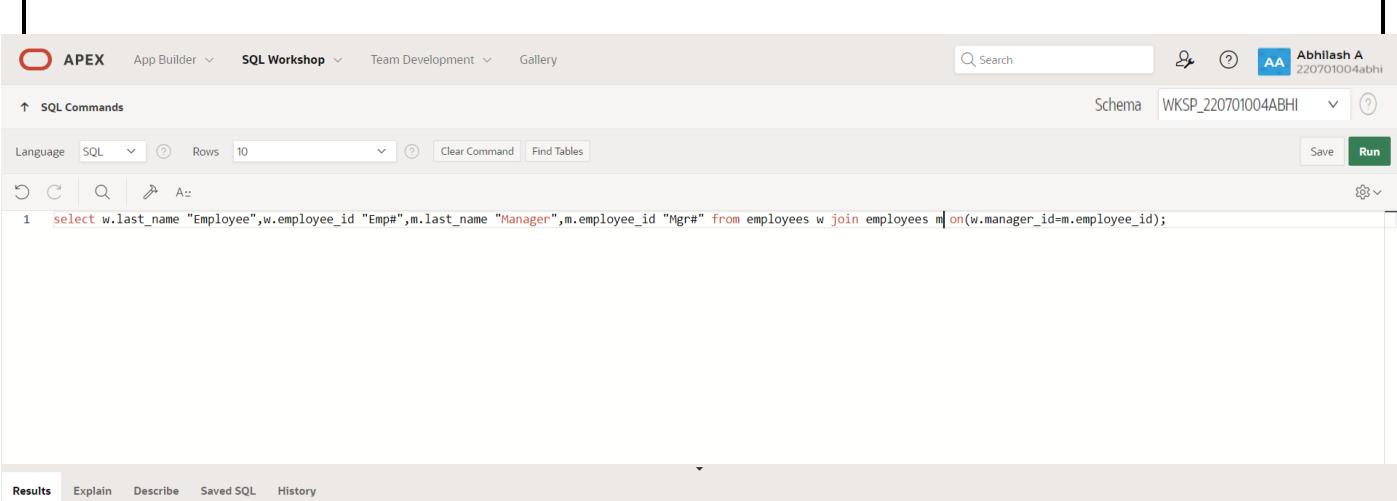
Below the editor, there are buttons for Language (SQL), Rows (10), Clear Command, and Find Tables. On the right, there are Save and Run buttons. The results panel at the bottom shows the following tabs: Results (selected), Explain, Describe, Saved SQL, and History. The results table is empty, displaying the message "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.emp_id "emp#",m.last_name 'manager',m.emp_id "Mgr#" from employees m on (w.manager_id=m.emp_id);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, a user profile for 'Abhilash A' (220701004abhi), and a schema dropdown set to 'WKSP_220701004ABHI'. Below the header, the 'SQL Commands' tab is selected. The SQL editor contains the following query:

```
1 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.employee_id);
```

Below the editor, there are buttons for Save and Run. The results section shows the following tabs: Results, Explain, Describe, Saved SQL, and History. The 'Results' tab is selected, and the message 'no data found' is displayed.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile for Abhilash A are also present. The main area is titled "SQL Commands" and contains a query editor with the following content:

```
1 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);
```

Below the query editor, there are several icons for navigation and search. The "Results" tab is selected, showing the following table output:

| Employee | Emp# | Manager | Mgr# |
|----------|------|---------|------|
| Janu | 113 | - | - |
| Mohan | 2 | - | - |
| HARI | 114 | - | - |

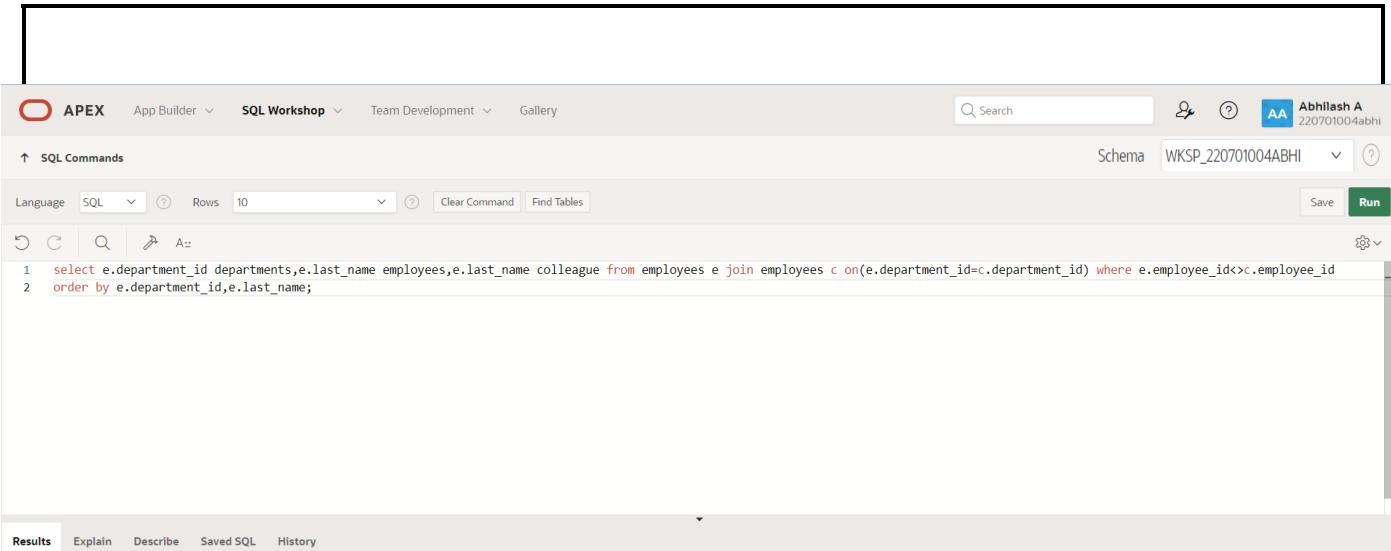
At the bottom, a message indicates "3 rows returned in 0.01 seconds" and a "Download" link is available.

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_number departments,e.last_name colleague from employees e join employees c on  
(e.department_number=c.department_number) where e.emp_id <> c.emp_id order by  
e.department_number,e.last_name,c.last_name;
```

QUTPUT:



The screenshot shows the Oracle SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there is a user profile for 'Abhilash A' with the ID '220701004abhi'. The main area is a 'SQL Commands' editor with the following query:

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

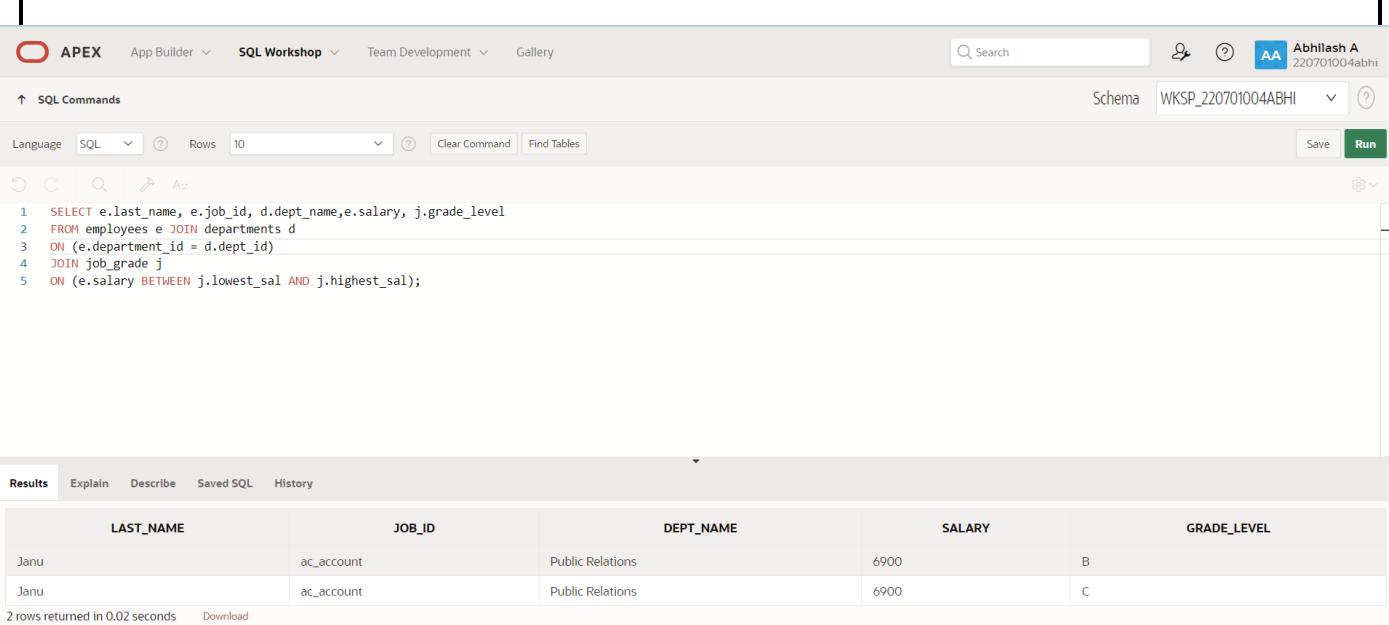
Below the editor, there are buttons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run. The 'Results' tab is selected in the bottom navigation bar. The results panel displays the message 'no data found'.

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

QUERY:

```
SELECT e.last_name, e.job_id, d.dept_name, e.salary, j.grade_level
FROM emp18 e JOIN dept18 d
ON (e.department_id = d.department_id)
JOIN job_grade j
ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a user profile for 'Abhilash A' are on the right. Below the tabs, a toolbar includes icons for Undo, Redo, Search, and Find Tables, along with buttons for Language (SQL), Rows (10), Clear Command, and Run. The main area contains a SQL command window with the following code:

```
1 SELECT e.last_name, e.job_id, d.dept_name, e.salary, j.grade_level
2 FROM employees e JOIN departments d
3 ON (e.department_id = d.department_id)
4 JOIN job_grade j
5 ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```

Below the code, the 'Results' tab is selected, showing a table with the following data:

| LAST_NAME | JOB_ID | DEPT_NAME | SALARY | GRADE_LEVEL |
|-----------|------------|------------------|--------|-------------|
| Janu | ac_account | Public Relations | 6900 | B |
| Janu | ac_account | Public Relations | 6900 | C |

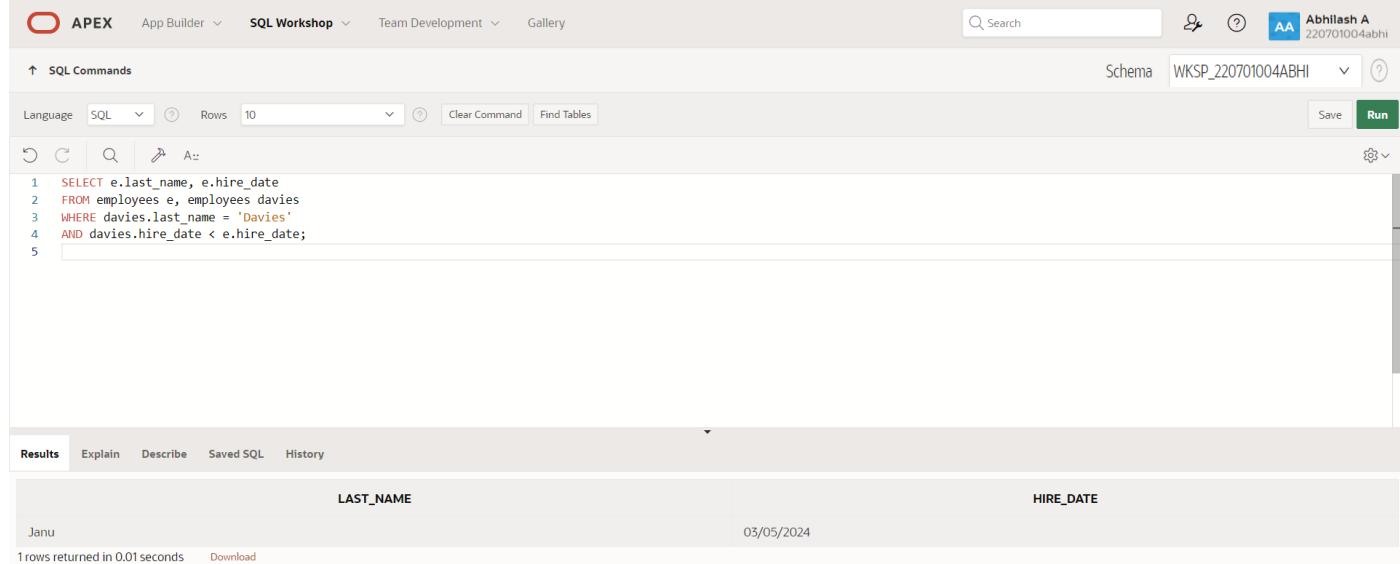
2 rows returned in 0.02 seconds [Download](#)

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 emp18 e, emp18 davies
WHERE davies.last_name = 'Davies'
AND davies.hire_date < e.hire_date;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there are user profile icons for 'Abhilash A' and a search bar. Below the tabs, there are buttons for Language (SQL), Rows (10), Clear Command, and Find Tables. The Schema dropdown is set to 'WKSP_220701004ABHI'. On the far right, there are 'Save' and 'Run' buttons. The main area contains a SQL command window with the following code:

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

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

| LAST_NAME | HIRE_DATE |
|-----------|------------|
| Janu | 03/05/2024 |

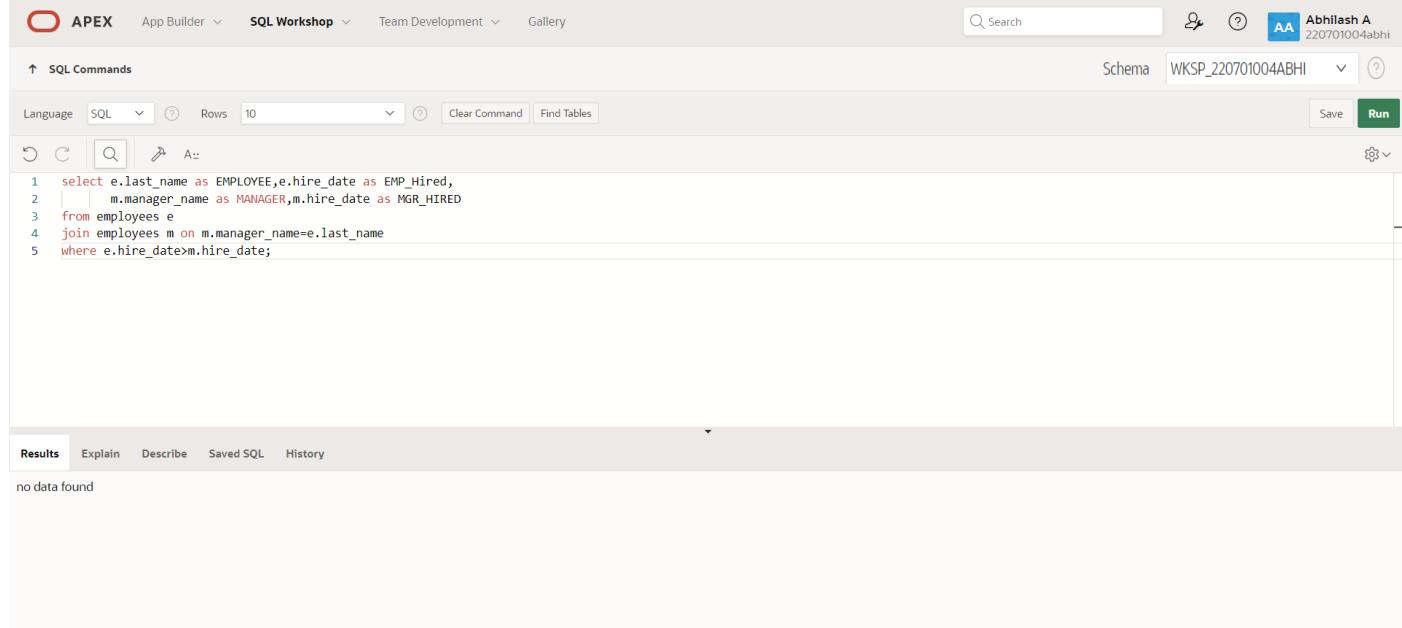
At the bottom left, it says '1 rows returned in 0.01 seconds'. There is also a 'Download' link.

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 emp18 e
JOIN emp18|m ON e.manager_name = m. last_name
WHERE e.hire_date < m.hire_date;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', 'Gallery', 'Search', and a user profile for 'Abhilash A'. The schema is set to 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The query editor contains the following SQL code:

```
1 select e.last_name as EMPLOYEE,e.hire_date as EMP_Hired,
2       m.manager_name as MANAGER,m.hire_date as MGR_HIRED
3  from employees e
4 join employees m on m.manager_name=e.last_name
5 where e.hire_date>m.hire_date;
```

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

| 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

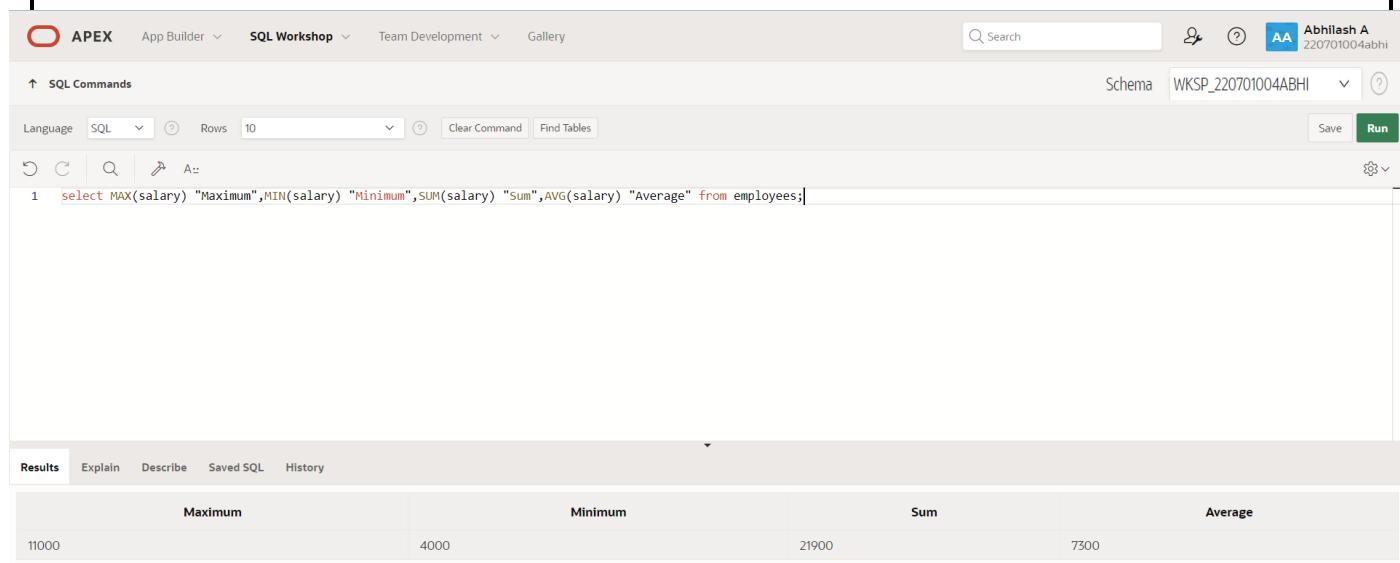
FALSE

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:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is active. The right side of the header shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a 'Run' button. The SQL editor contains the following query:

```
1 select MAX(salary) "Maximum",MIN(salary) "Minimum",SUM(salary) "Sum",AVG(salary) "Average" from employees;
```

The results are displayed in a table:

| Maximum | Minimum | Sum | Average |
|---------|---------|-------|---------|
| 11000 | 4000 | 21900 | 7300 |

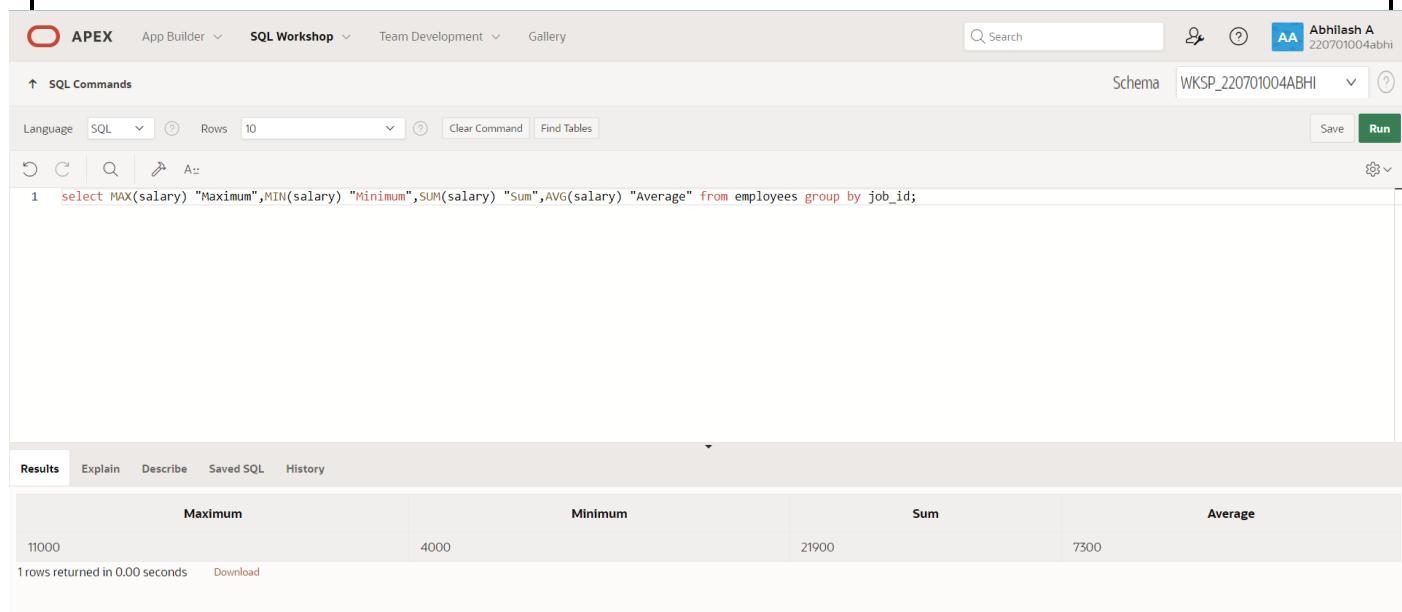
Below the table, it says '1 rows returned in 0.03 seconds' and has a 'Download' link.

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:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is active. The right side of the header shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. A search bar and various icons are also present. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The query entered is:

```
1 select MAX(salary) "Maximum",MIN(salary) "Minimum",SUM(salary) "Sum",AVG(salary) "Average" from employees group by job_id;
```

The results are displayed in a table with four columns: Maximum, Minimum, Sum, and Average. The data is:

| Maximum | Minimum | Sum | Average |
|---------|---------|-------|---------|
| 11000 | 4000 | 21900 | 7300 |

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

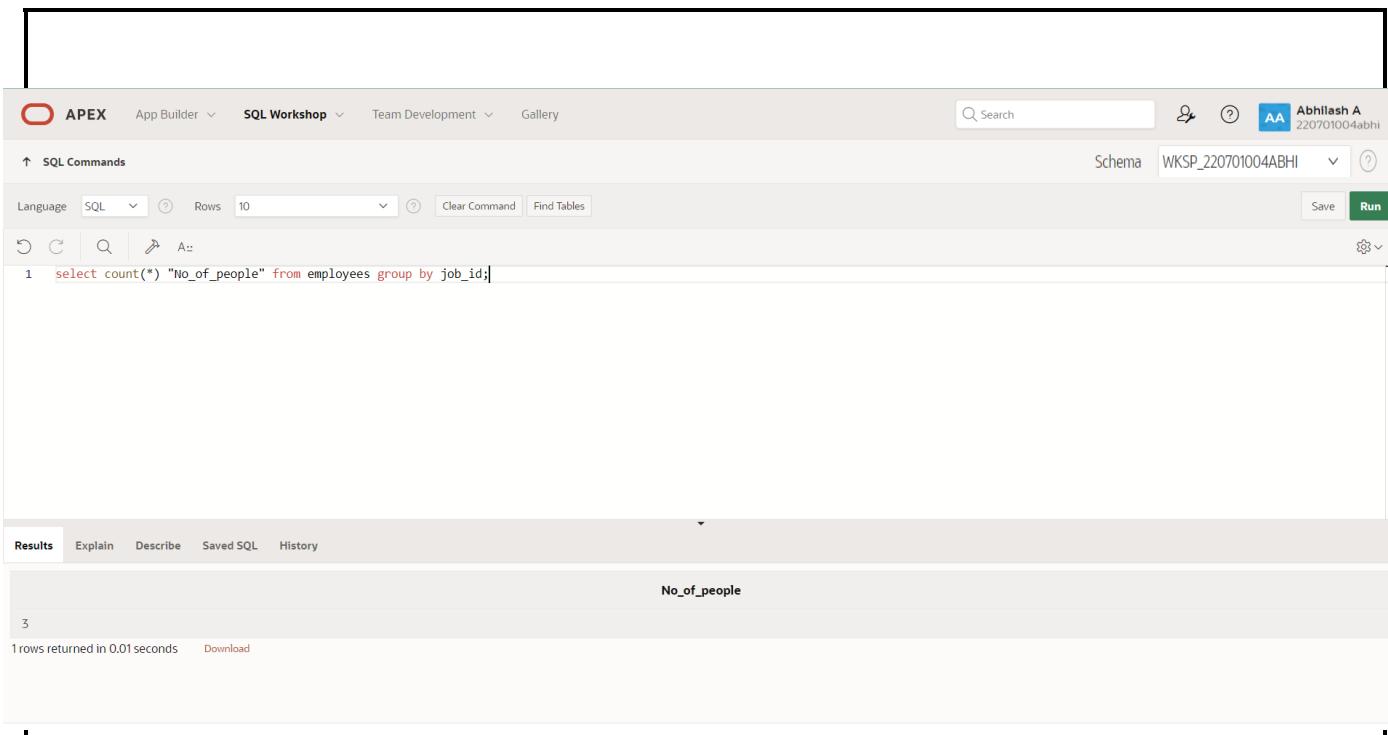
6. Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

QUERY:

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

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, a user profile for Abhilash A (220701004abhi), and a schema dropdown set to WKSP_220701004ABHI. Below the header, the SQL Commands tab is selected, showing a query editor with the following SQL code:

```
1 select count(*) "No_of_people" from employees group by job_id;
```

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

| No_of_people |
|--------------|
| 3 |

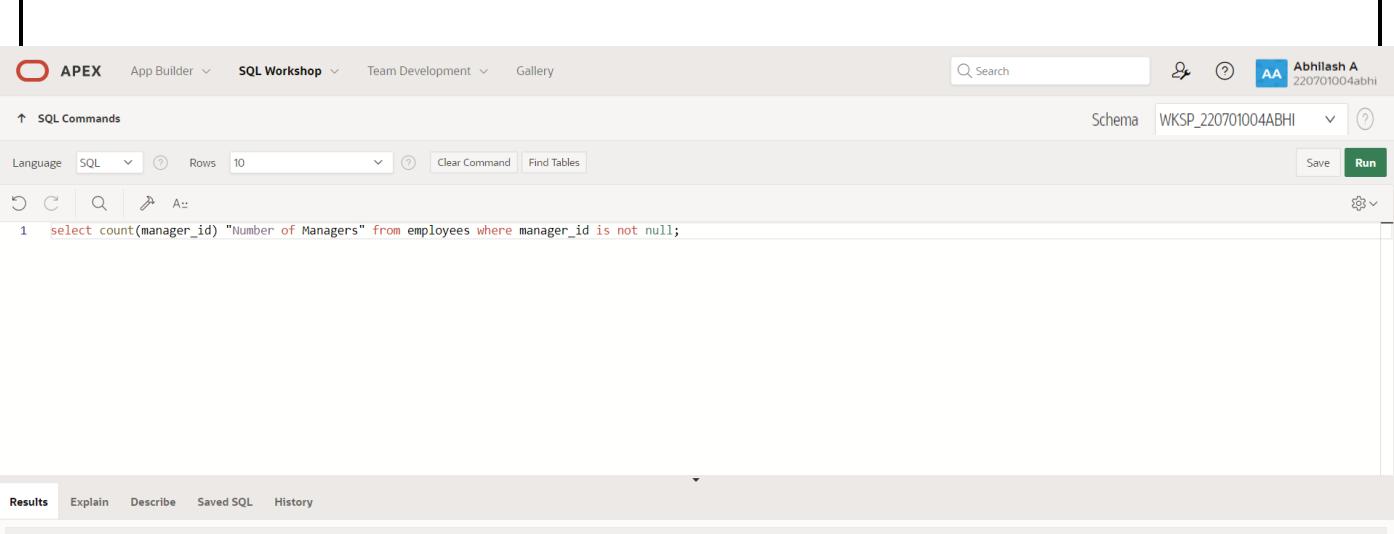
1 rows returned in 0.01 seconds [Download](#)

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

QUERY:

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

OUTPUT:



A screenshot of the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile for 'Abhilash A' are also present. The main area shows a SQL command window with the following content:

```
1 select count(manager_id) "Number of Managers" from employees where manager_id is not null;
```

The results tab is selected, showing the output:

| Number of Managers |
|--------------------|
| 3 |

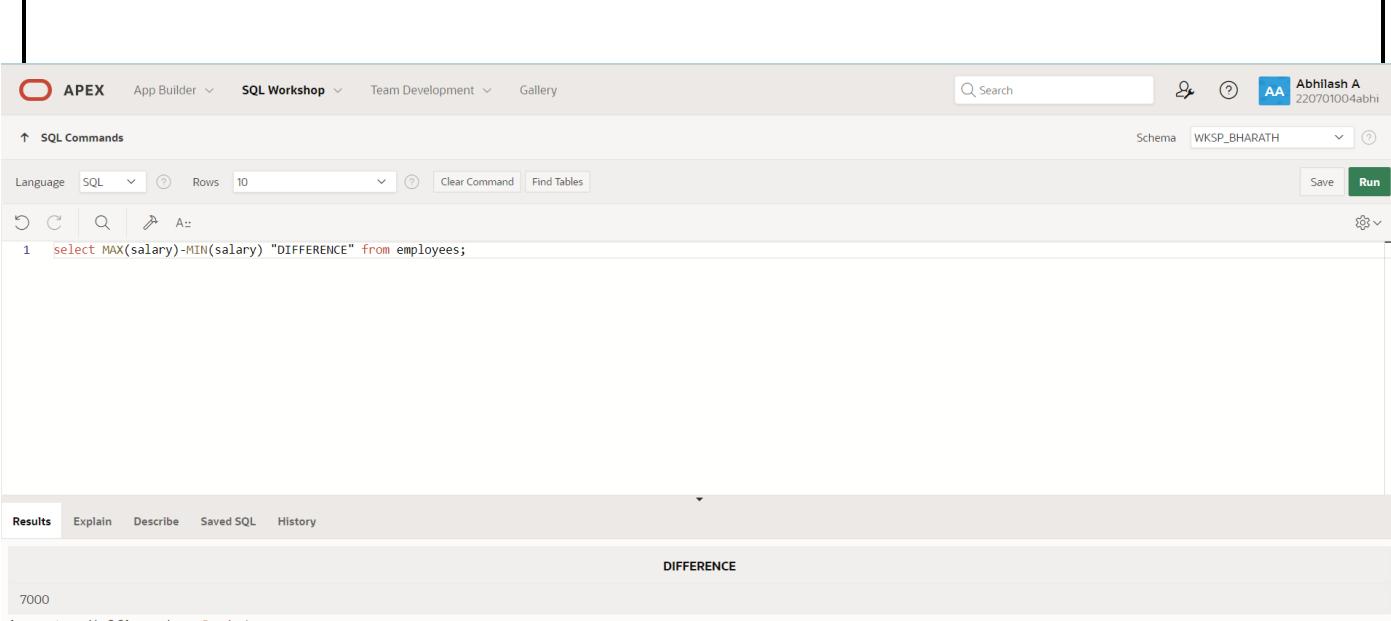
1 rows returned in 0.01 seconds [Download](#)

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

QUERY:

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

OUTPUT:



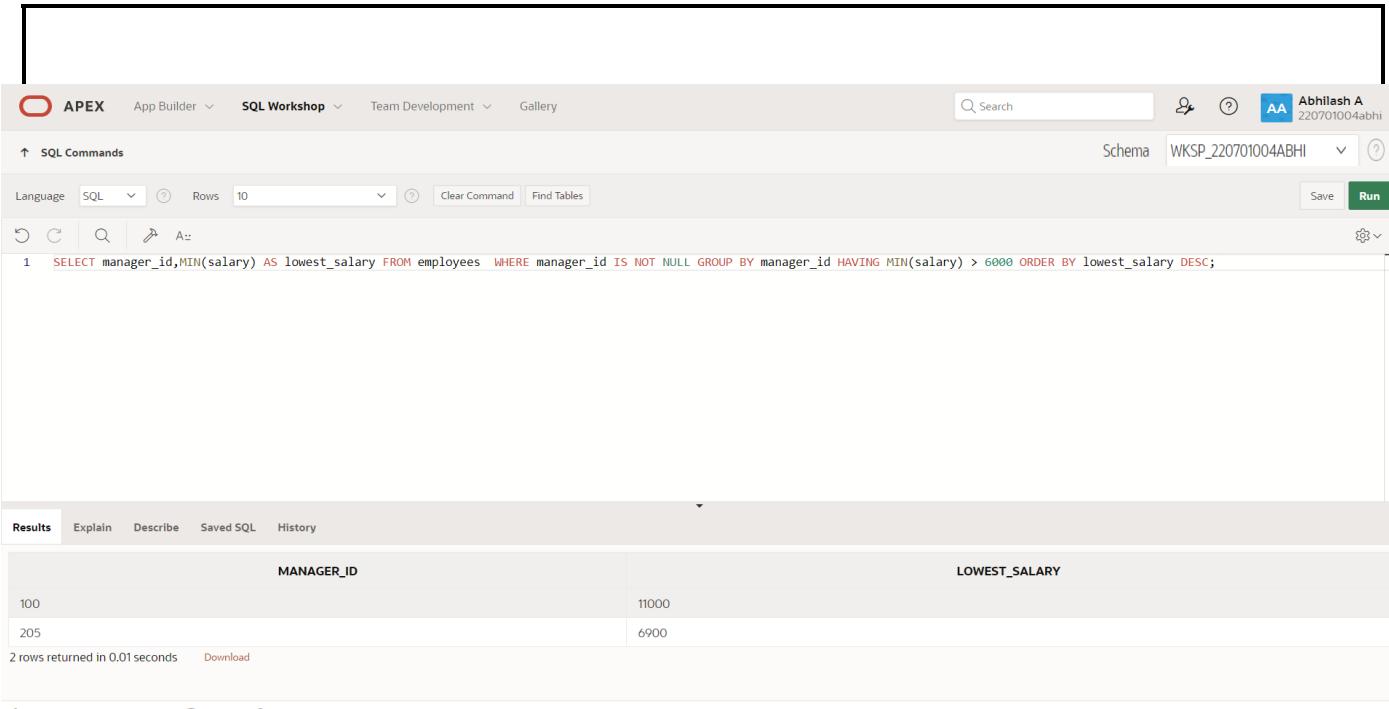
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name, Abhilash A, and a session ID. The main workspace is titled "SQL Commands" and contains a single line of SQL code: "select MAX(salary)-MIN(salary) "DIFFERENCE" from employees;". Below the code, the "Results" tab is selected, showing a single row with the value "7000" under the column "DIFFERENCE". The status bar at the bottom indicates "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:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows the user 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The query entered is:

```
1 SELECT manager_id,MIN(salary) AS lowest_salary FROM employees WHERE manager_id IS NOT NULL GROUP BY manager_id HAVING MIN(salary) > 6000 ORDER BY lowest_salary DESC;
```

The results section shows a table with two rows:

| MANAGER_ID | LOWEST_SALARY |
|------------|---------------|
| 100 | 11000 |
| 205 | 6900 |

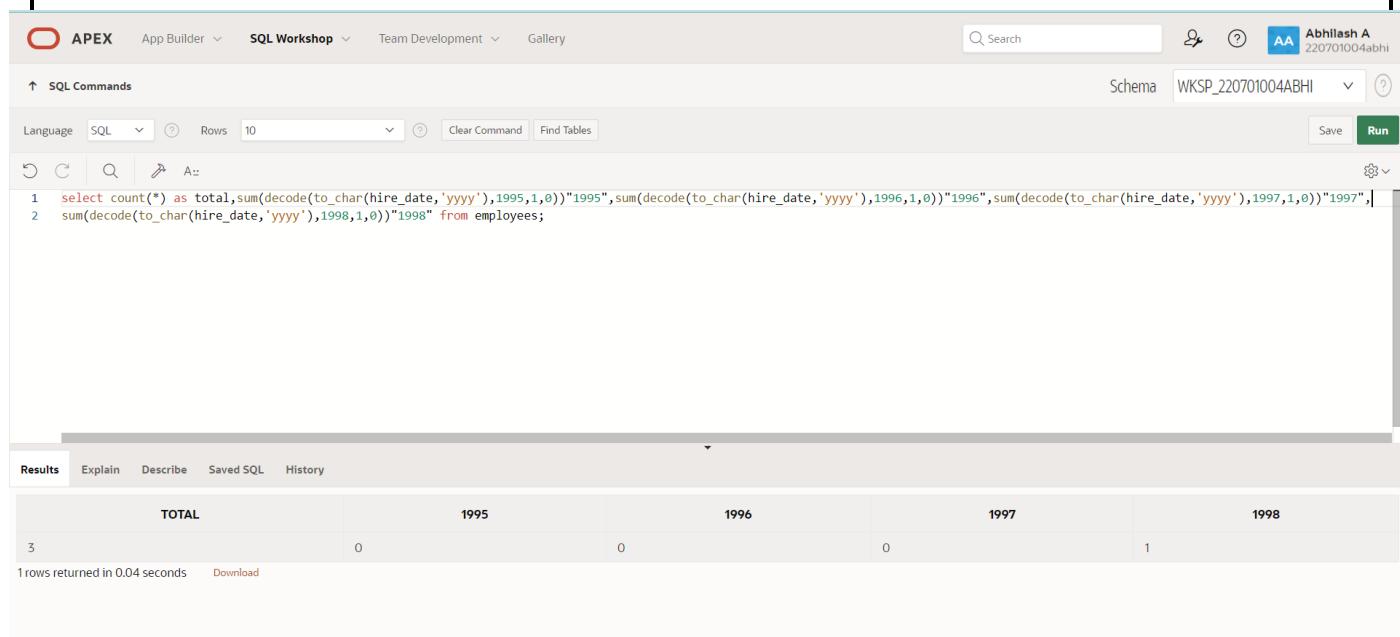
Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user's profile: Abhilash A, ID 220701004abhi. The main area is titled 'SQL Commands' with a schema of WKSP_220701004ABHI. The SQL editor contains the following query:

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

The results tab is selected, showing the following output:

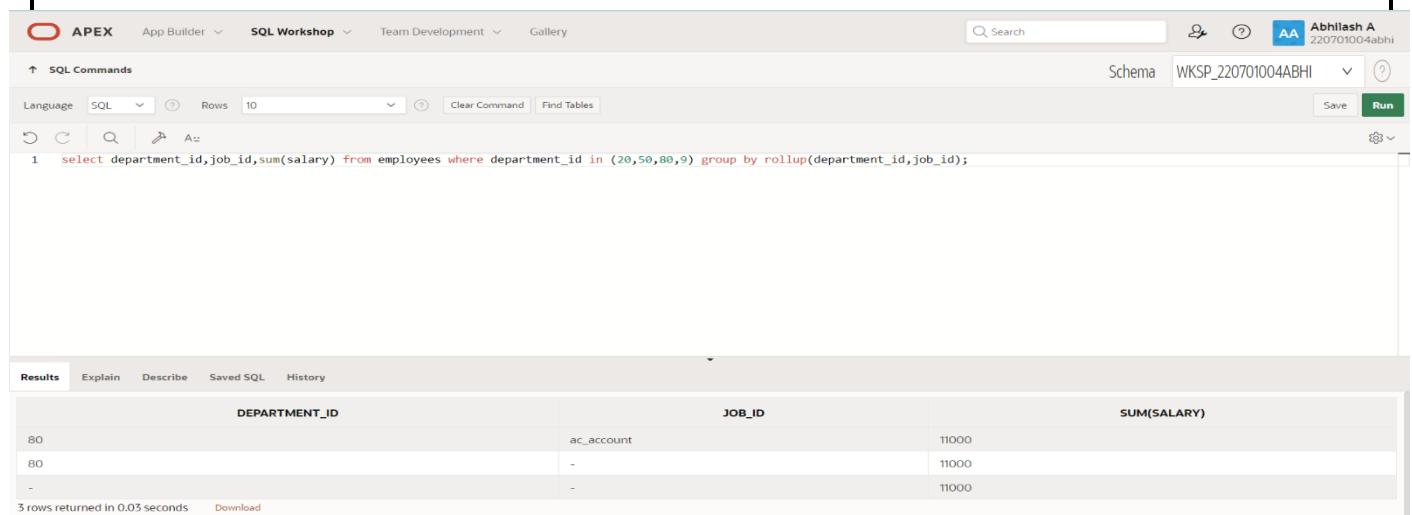
| TOTAL | 1995 | 1996 | 1997 | 1998 |
|-------|------|------|------|------|
| 3 | 0 | 0 | 0 | 1 |

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

1. 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:

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user 'Abhilash A' with the ID '220701004abhi'. The SQL Commands tab is active, with the schema set to 'WKSP_220701004ABHI'. The query entered is:

```
1 select department_id,job_id,sum(salary) from employees where department_id in (20,50,80,9) group by rollup(department_id,job_id);
```

The results tab shows the output of the query:

| DEPARTMENT_ID | JOB_ID | SUM(SALARY) |
|---------------|------------|-------------|
| 80 | ac_account | 11000 |
| 80 | - | 11000 |
| - | - | 11000 |

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

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.dpt_id = e.department_id ) group by d.dept_name ,d.loc;
```

OUTPUT:

APEx App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables

Schema WKSP_220701004ABHI Run

1 select d.dept_name as "Department name",l.location_id as "Location",count(e.department_id) as "Number of people",round(avg(e.salary),2) as "Salary"
2 from departments d,employees e,location l where d.dept_id=e.department_id group by d.dept_name,l.location_id,e.department_id;

Results Explain Describe Saved SQL History

| Department name | Location | Number of people | Salary |
|------------------|----------|------------------|--------|
| Public Relations | 4598 | 1 | 6900 |
| Public Relations | 1231 | 1 | 6900 |
| finance | 4598 | 1 | 11000 |
| finance | 1231 | 1 | 11000 |

4 rows returned in 0.02 seconds [Download](#)

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

RESULT:

SUB QUERIES

EX_NO:9

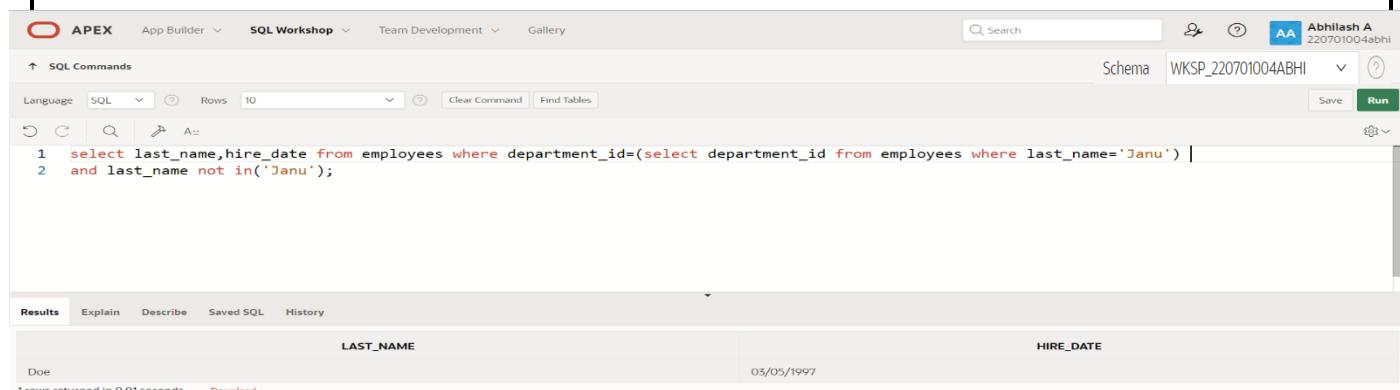
DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is selected. The right side of the interface shows the user 'Abhilash A' with the ID '220701004abhi'. The 'Schema' dropdown is set to 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The query entered is:

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

The results section shows a single row:

| LAST_NAME | HIRE_DATE |
|-----------|------------|
| Doe | 05/05/1997 |

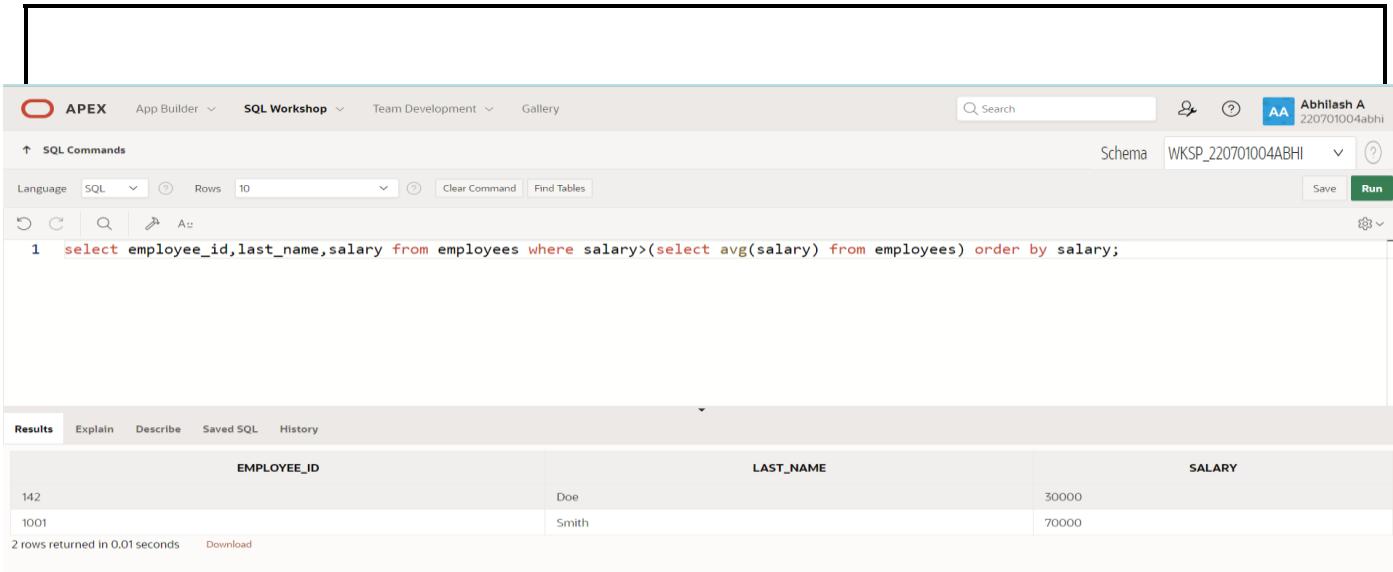
1 rows returned in 0.01 seconds. There are buttons for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query in the command window is:

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

The results window shows the following data:

| EMPLOYEE_ID | LAST_NAME | SALARY |
|-------------|-----------|--------|
| 142 | Doe | 30000 |
| 1001 | Smith | 70000 |

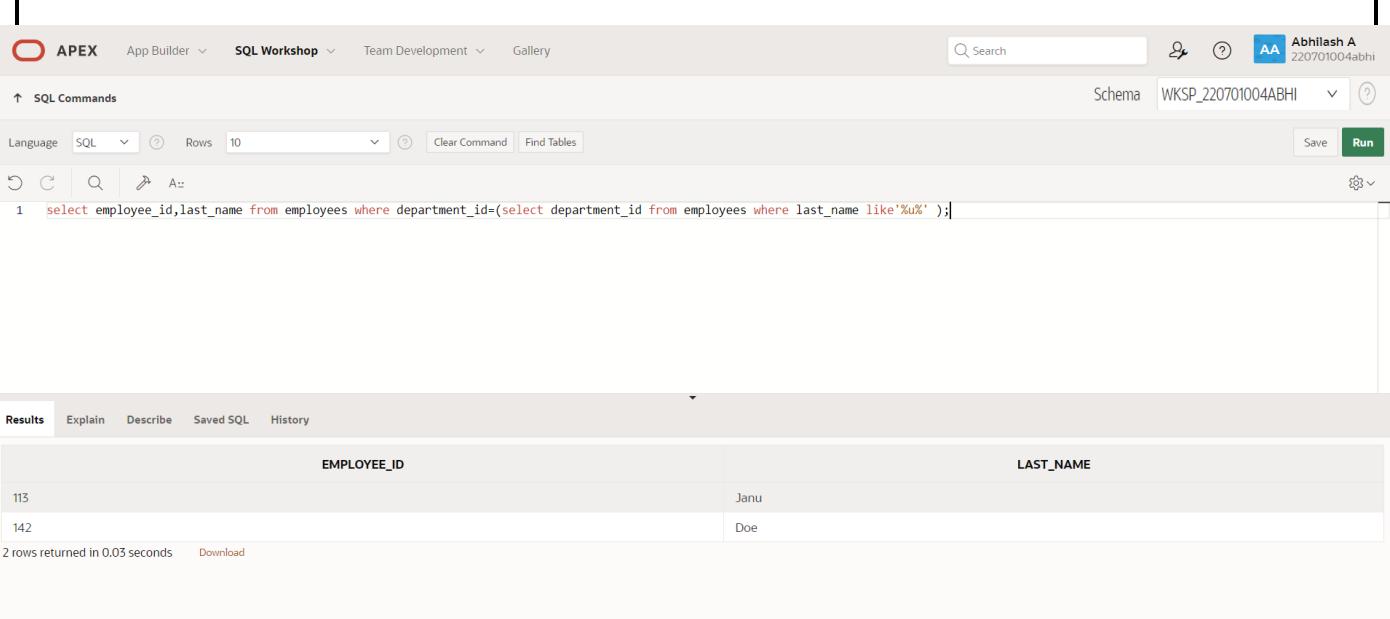
2 rows returned in 0.01 seconds [Download](#)

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there is a search bar, a user icon for 'Abhilash A 220701004abhi', and a 'Schema' dropdown set to 'WKSP_220701004ABHI'. The main area shows a SQL command in the editor:

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

The results table has two rows:

| EMPLOYEE_ID | LAST_NAME |
|-------------|-----------|
| 113 | Janu |
| 142 | Doe |

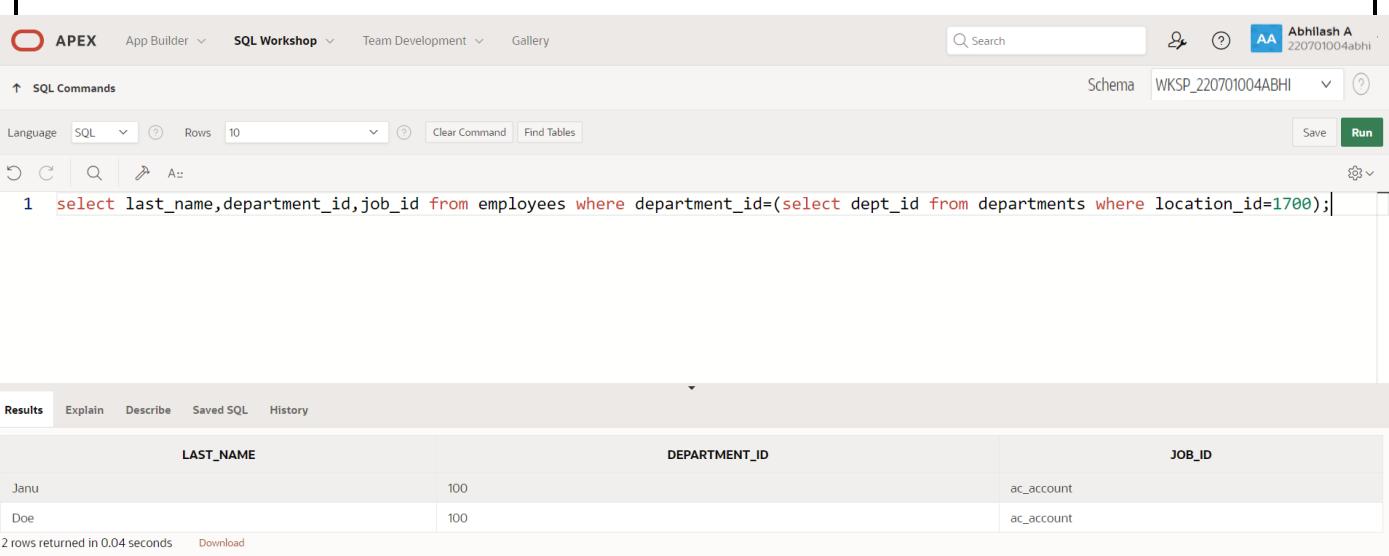
Below the table, it says '2 rows returned in 0.03 seconds' and there is a 'Download' link.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows the user 'Abhilash A' with the ID '220701004abhi'. The main area is a SQL editor with the following query:

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

The results tab is selected, displaying the following data:

| LAST_NAME | DEPARTMENT_ID | JOB_ID |
|-----------|---------------|------------|
| Janu | 100 | ac_account |
| Doe | 100 | ac_account |

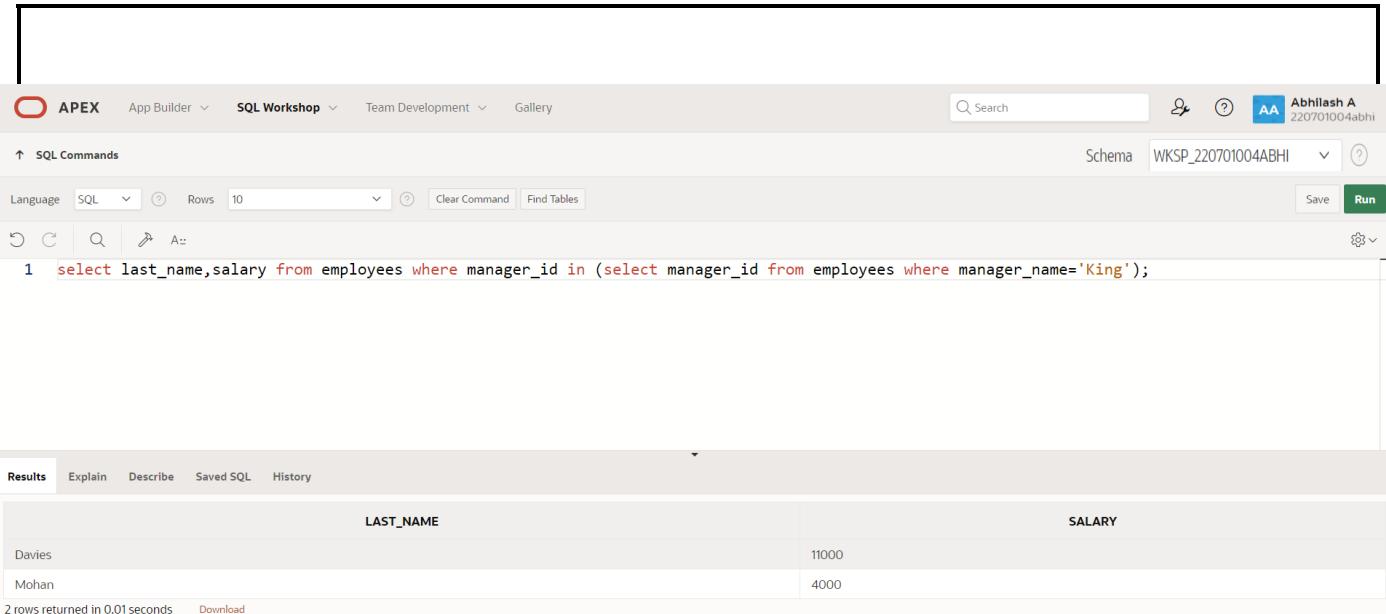
2 rows returned in 0.04 seconds [Download](#)

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:



A screenshot of the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows a user profile for 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701004ABHI'. The SQL editor contains the following query:

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

The results tab is selected, showing the output:

| LAST_NAME | SALARY |
|-----------|--------|
| Davies | 11000 |
| Mohan | 4000 |

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

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Search Schema WKSP_220701004ABHI

SQL Commands

Language SQL Rows 10 Clear Command Find Tables

Run

```

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

```

Results Explain Describe Saved SQL History

| DEPARTMENT_ID | LAST_NAME | JOB_ID |
|---------------|-----------|------------|
| 80 | Smith | sales_rep |
| 80 | Davies | ac_account |
| 20 | Doe | ac_account |

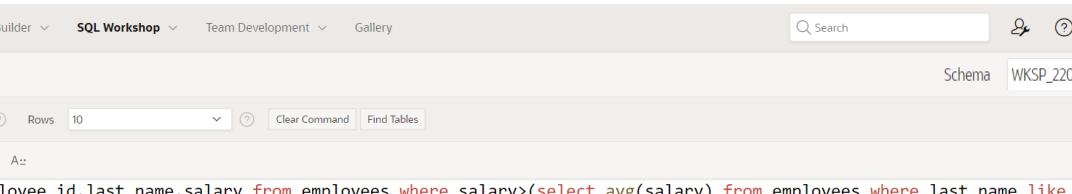
3 rows returned in 0.01 seconds [Download](#)

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:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'Abhilash A 220701004abhi' are on the right. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701004ABHI'. The SQL editor contains the following query:

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

The 'Results' tab is selected, showing the following output:

| EMPLOYEE_ID | LAST_NAME | SALARY |
|-------------|-----------|--------|
| 1001 | Smith | 70000 |
| 142 | Doeu | 30000 |

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

| Evaluation Procedure | Marks awarded |
|----------------------|---------------|
| Query(5) | |
| Execution (5) | |
| Viva(5) | |

| | |
|-------------------|--|
| Total (15) | |
| Faculty Signature | |

RESULT:

USING THE SET OPERATORS

EX_NO:10

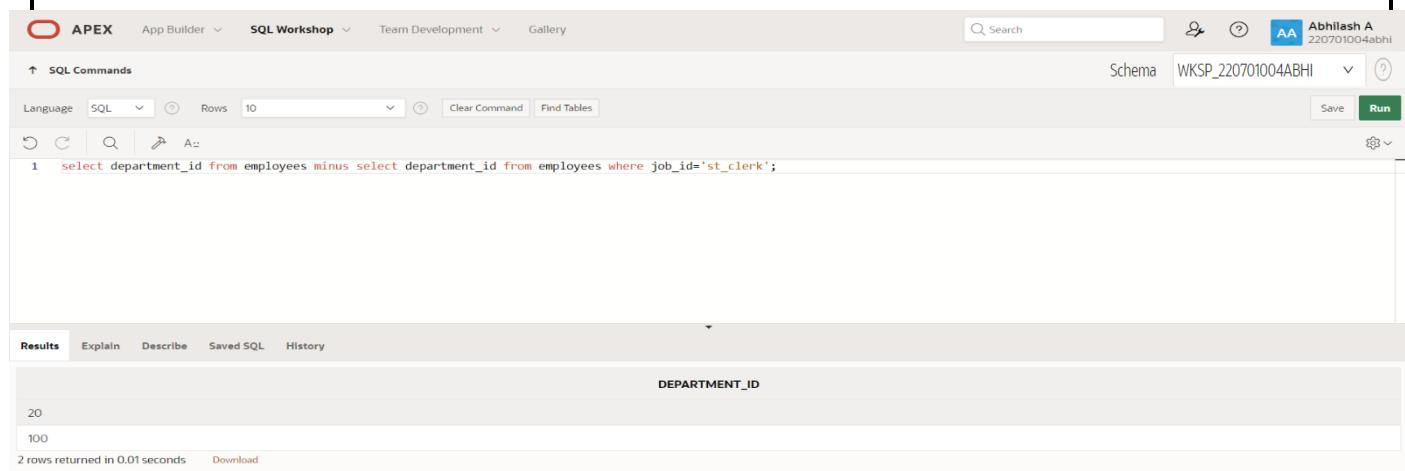
DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_220701004ABHI'. The SQL editor contains the following query:

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

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

| DEPARTMENT_ID |
|---------------|
| 20 |
| 100 |

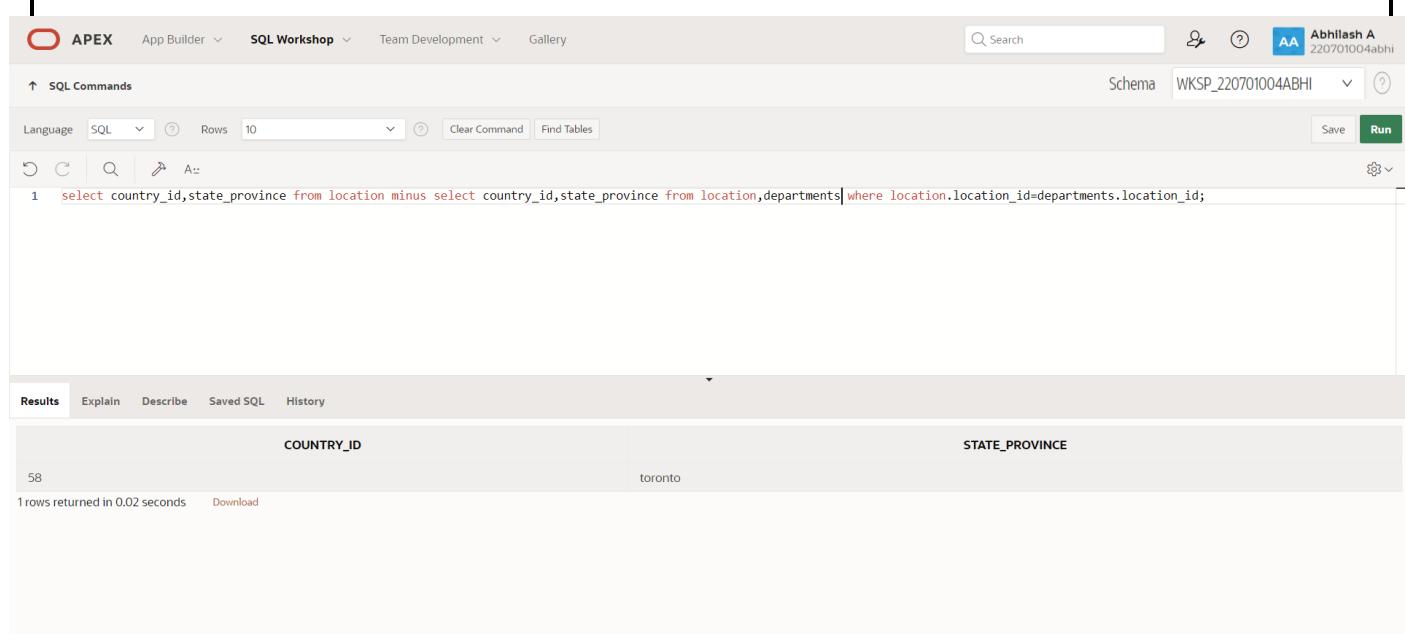
Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link.

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:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is a SQL editor with the following content:

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

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

| COUNTRY_ID | STATE_PROVINCE |
|------------|----------------|
| 58 | toronto |

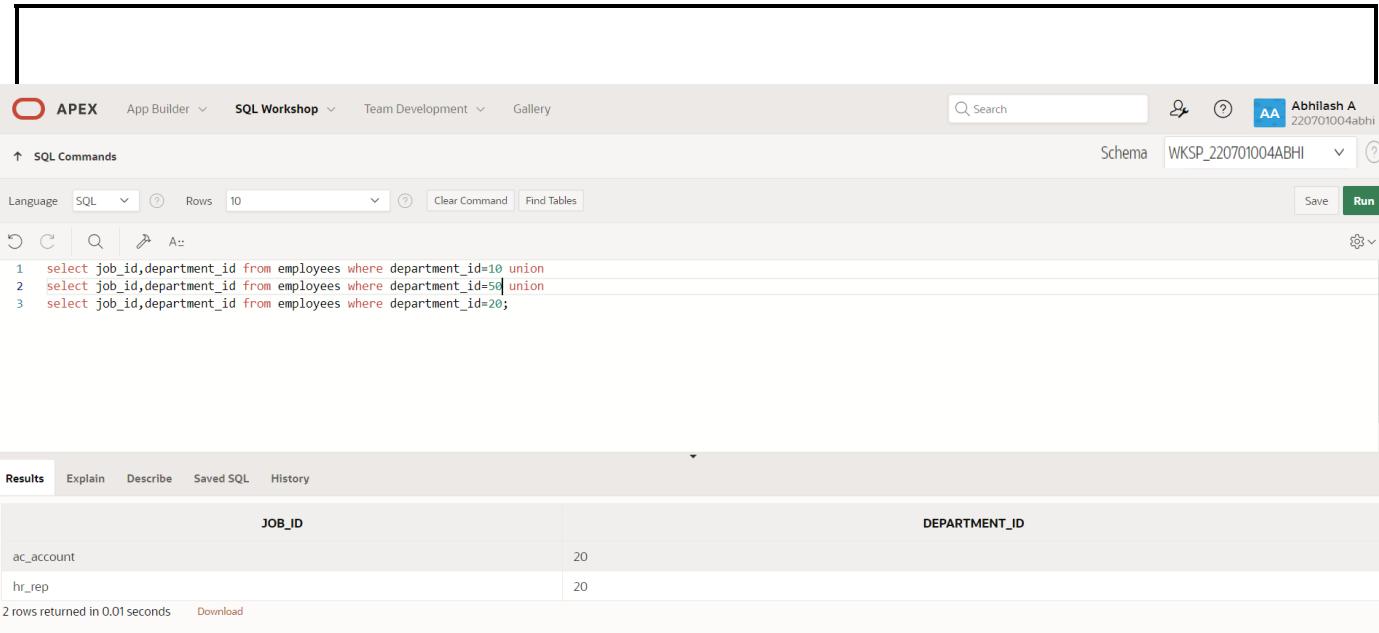
1 rows returned in 0.02 seconds [Download](#)

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:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', 'Gallery', a search bar, and a user profile for 'Abhilash A'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The query editor contains the following SQL code:

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

The results tab is selected, showing a table with two columns: 'JOB_ID' and 'DEPARTMENT_ID'. The data is as follows:

| JOB_ID | DEPARTMENT_ID |
|------------|---------------|
| ac_account | 20 |
| hr_rep | 20 |

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

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows a user profile for 'Abhilash A' with the ID '220701004abhi'. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701004ABHI'. The SQL editor contains the following query:

```
1 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;
```

The results tab is selected, showing the following data:

| JOB_ID | EMPLOYEE_ID |
|------------|-------------|
| ac_account | 113 |
| ac_account | 142 |
| sales_rep | 1001 |

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

- 5.) The HR department needs a report with the following specifications: - Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department. - Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them. Write a compound query to accomplish this.

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

Schema WKSP_220701004ABHI

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

Results Explain Describe Saved SQL History

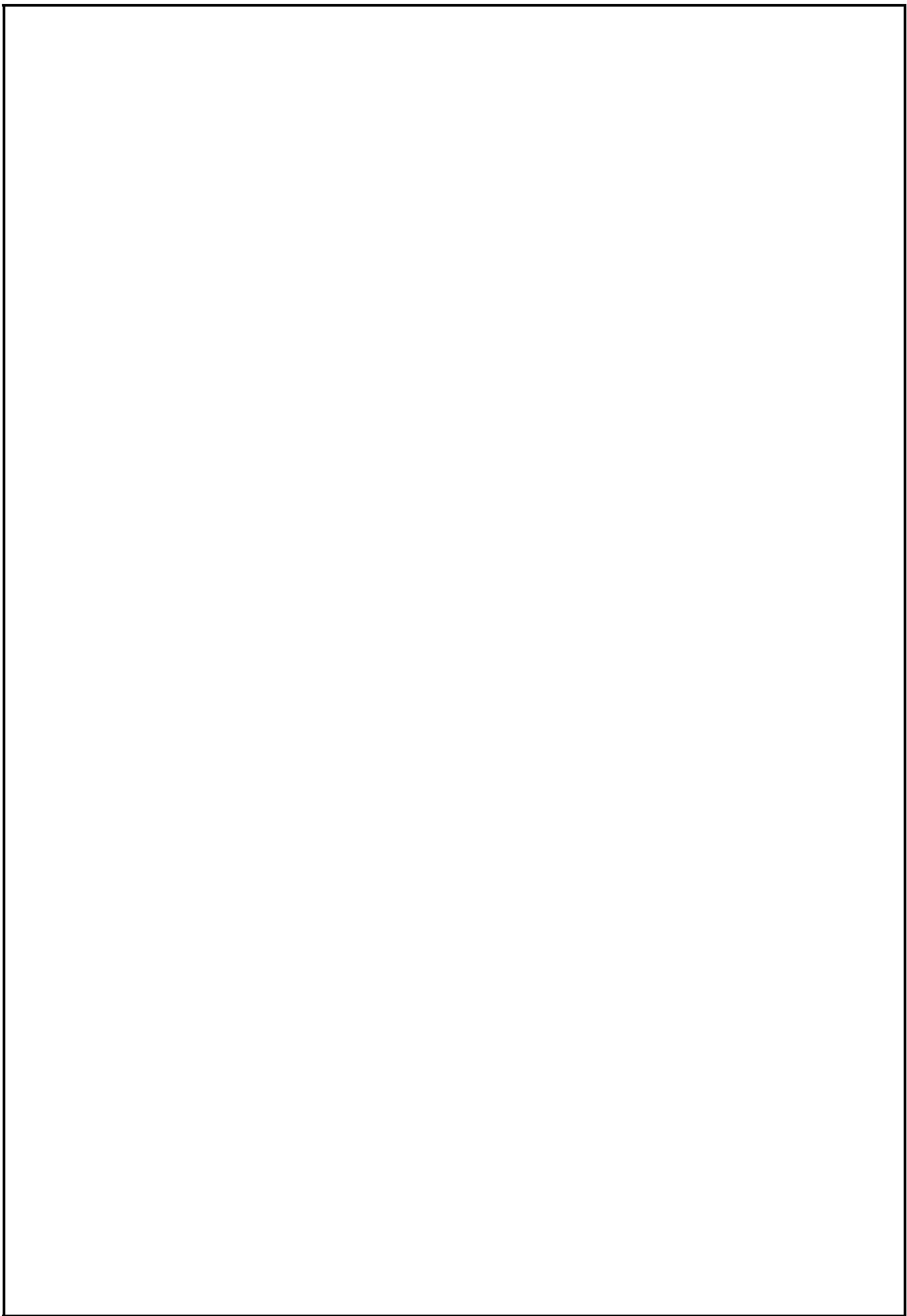
| Name | DEPARTMENT_ID |
|------------------|---------------|
| John Smith | 80 |
| Emily Johnson | 20 |
| Jaunty Janu | 100 |
| den Davies | 80 |
| Jane Doe | 20 |
| Vijaya Mohan | 150 |
| Public Relations | 100 |
| finance | 80 |
| Executive | 80 |
| Executive | 20 |

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

| Evaluation Procedure | Marks awarded |
|----------------------|---------------|
| Query(5) | |
| Execution (5) | |
| Viva(5) | |

| | |
|-------------------|--|
| Total (15) | |
| Faculty Signature | |

RESULT:



CREATING VIEWS

EX-NO : 11

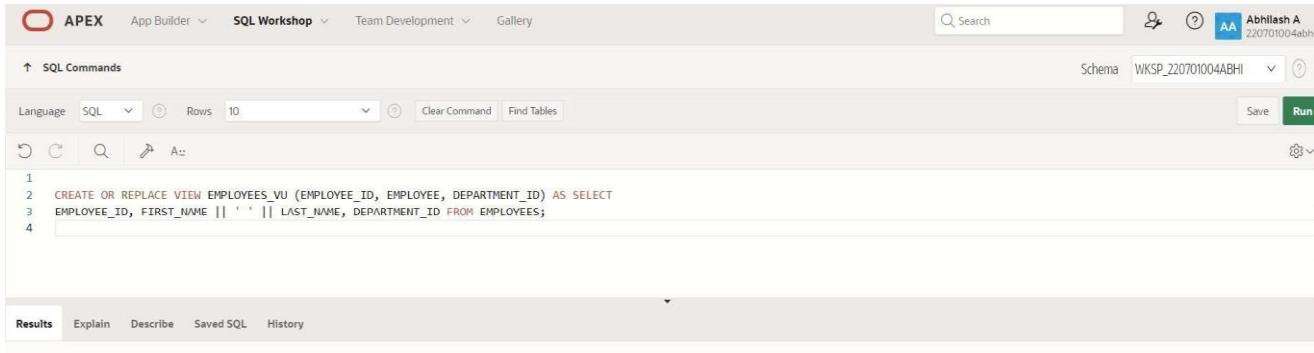
DATE:

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 (EMPLOYEE_ID, EMPLOYEE,  
DEPARTMENT_ID) AS SELECT  
EMPLOYEE_ID, FIRST_NAME || ' ' || LAST_NAME, DEPARTMENT_ID FROM EMPLOYEES;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The main area is titled "SQL Commands" and shows the following SQL code:

```
1 CREATE OR REPLACE VIEW EMPLOYEES_VU (EMPLOYEE_ID, EMPLOYEE, DEPARTMENT_ID) AS SELECT
2 EMPLOYEE_ID, FIRST_NAME || ' ' || LAST_NAME, DEPARTMENT_ID FROM EMPLOYEES;
3
4
```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected. A message "View created." is displayed, followed by "0.03 seconds".

2. Display the contents of the EMPLOYEES_VU view.

QUERY:

```
SELECT*FROM EMPLOYEES_VU;
```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

1 2 SELECT*FROM EMPLOYEES_VU;

Results Explain Describe Saved SQL History

| EMPLOYEE_ID | EMPLOYEE | DEPARTMENT_ID |
|-------------|--------------|---------------|
| 176 | SAM EMANUEL | 50 |
| 2 | ALEX JAY | 39 |
| 4 | MAX DAVIES | 5 |
| 172 | JENNIE JANE | 80 |
| 5 | MEENA PARTHI | 50 |
| 3 | JENNY UMA | 97 |

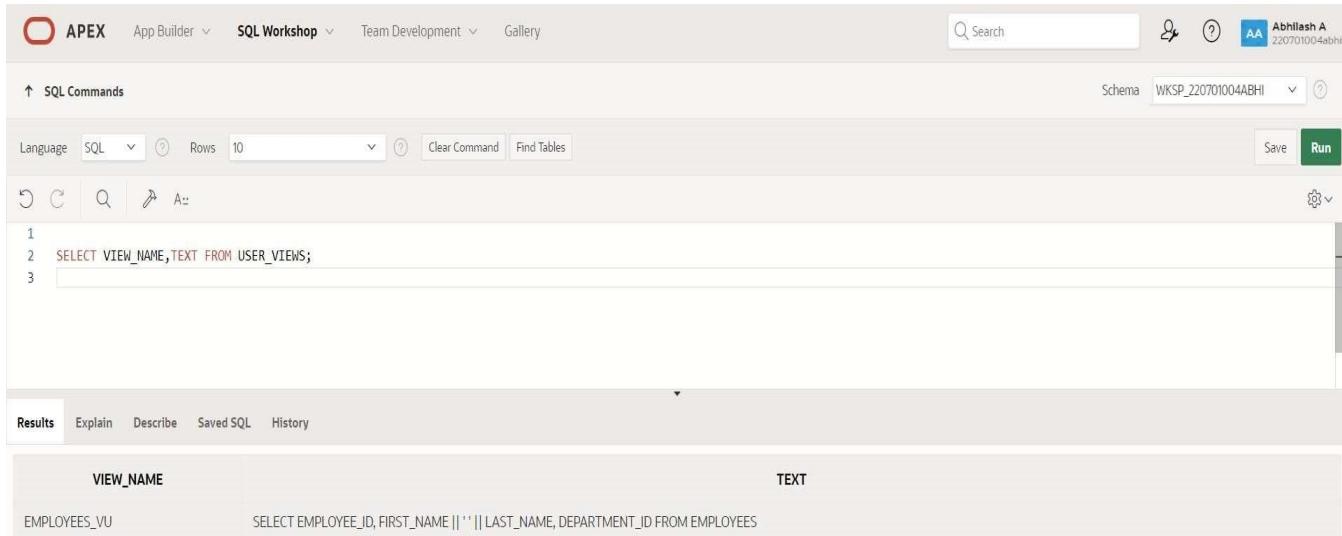
3.

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

QUERY:

```
SELECT VIEW_NAME,TEXT FROM USER_VIEWS;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The top right shows a search bar, user profile (Abhilash A), and session information (Schema: WKSP_220701004ABHI). The main area is titled 'SQL Commands' with a 'Run' button. The SQL editor contains the following code:

```
1 SELECT VIEW_NAME,TEXT FROM USER_VIEWS;
2
3
```

The results tab is selected, showing the output:

| VIEW_NAME | TEXT |
|--------------|--|
| EMPLOYEES_VU | SELECT EMPLOYEE_ID,FIRST_NAME ' ' LAST_NAME,DEPARTMENT_ID FROM EMPLOYEES |

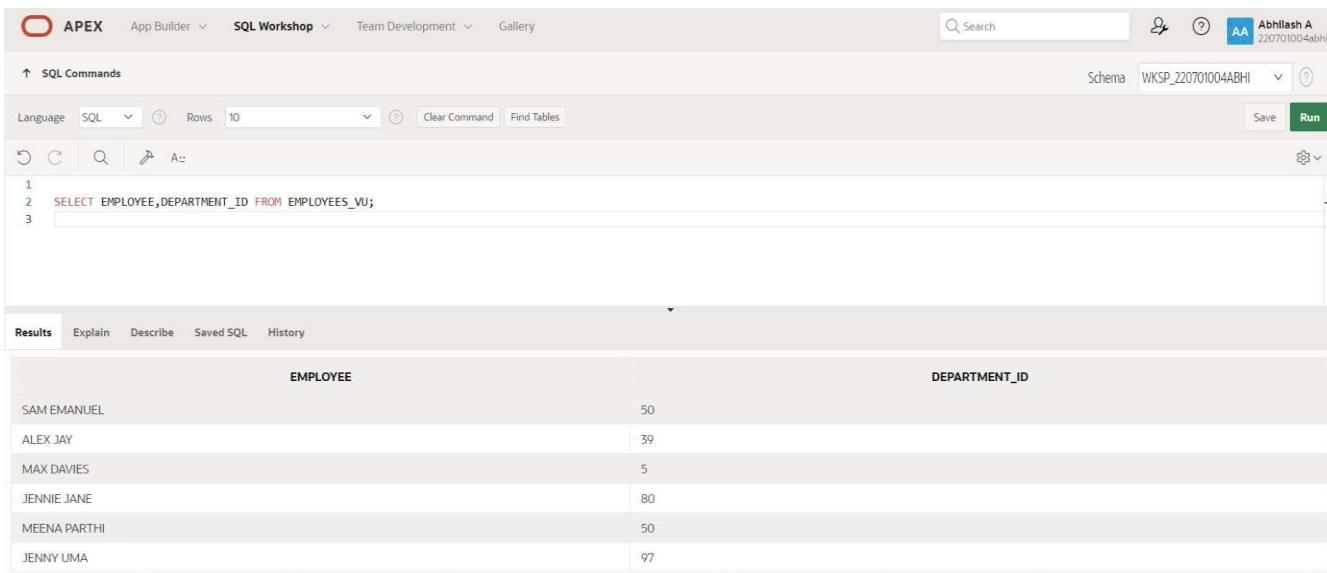
4.

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

QUERY:

```
SELECT EMPLOYEE,DEPARTMENT_ID FROM EMPLOYEES_VU;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and a schema dropdown 'WKSP_220701004abhi'. The main area has tabs for 'SQL Commands' and 'Results'. The 'SQL Commands' tab shows the query: '1 SELECT EMPLOYEE,DEPARTMENT_ID FROM EMPLOYEES_VU;'. The 'Results' tab displays the output as a table:

| EMPLOYEE | DEPARTMENT_ID |
|--------------|---------------|
| SAM EMANUEL | 50 |
| ALEX JAY | 39 |
| MAX DAVIES | 5 |
| JENNIE JANE | 80 |
| MEENA PARTHI | 50 |
| JENNY UMA | 97 |

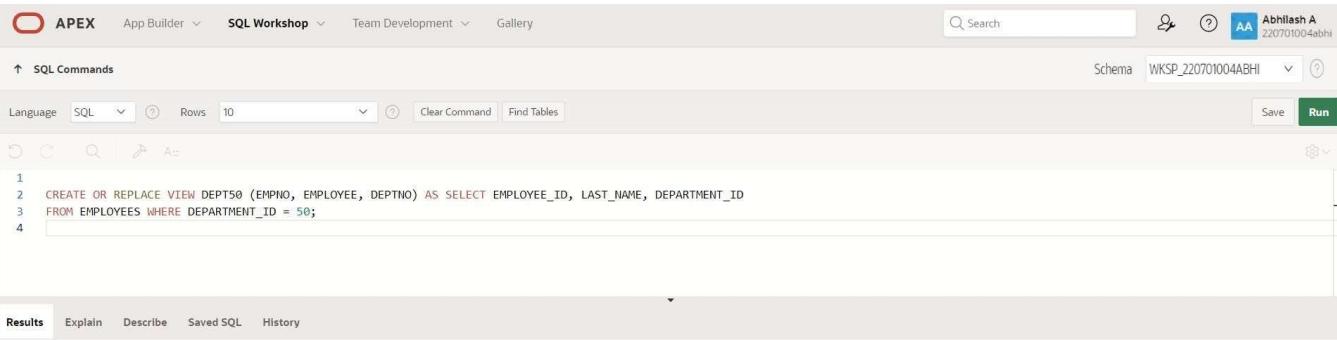
5.

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 OR REPLACE VIEW DEPT50 (EMPNO, EMPLOYEE, DEPTNO) AS
SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID
FROM EMPLOYEES WHERE DEPARTMENT_ID = 50;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows a user profile for 'Abhilash A' and a schema dropdown for 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The code area contains the following SQL command:

```
1 CREATE OR REPLACE VIEW DEPT50 (EMPNO, EMPLOYEE, DEPTNO) AS
2 SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID
3 FROM EMPLOYEES WHERE DEPARTMENT_ID = 50;
4
```

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

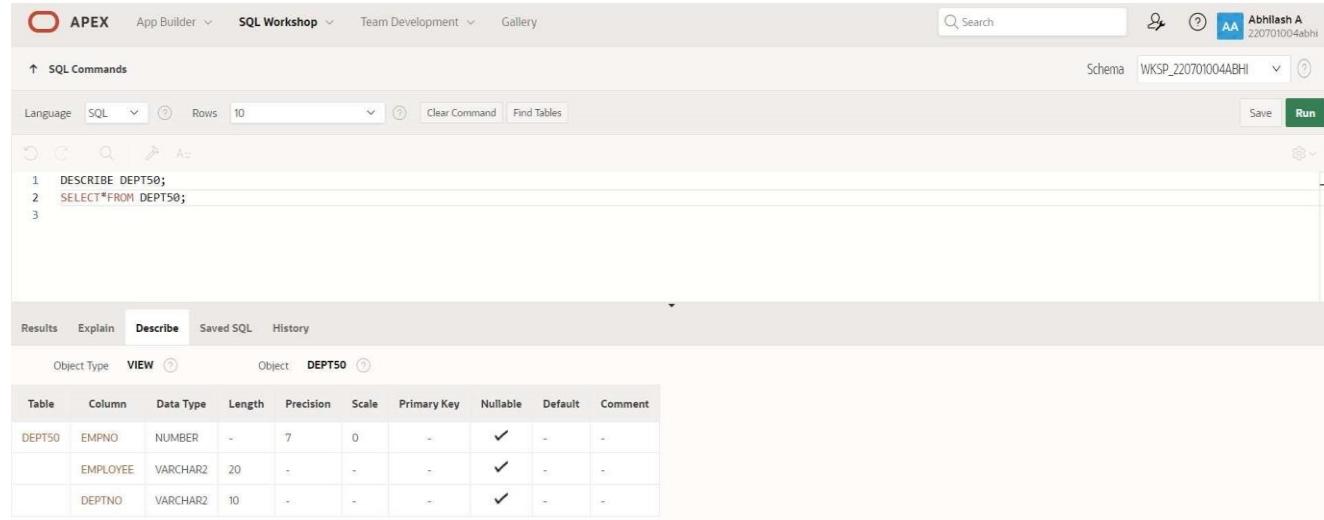
6.

Display the structure and contents of the DEPT50 view.

QUERY:

```
DESCRIBE DEPT50;  
SELECT*FROM DEPT50;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows a user profile for 'Abhilash A' (220701004abhi). The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_220701004ABHI'. The SQL editor contains the following commands:

```
1 DESCRIBE DEPT50;  
2 SELECT*FROM DEPT50;  
3
```

Below the editor, the 'Describe' tab is selected in the results pane. The results table shows the structure of the 'DEPT50' view:

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|--------|----------|-----------|--------|-----------|-------|-------------|----------|---------|---------|
| DEPT50 | EMPNO | NUMBER | - | 7 | 0 | - | ✓ | - | - |
| | EMPLOYEE | VARCHAR2 | 20 | - | - | - | ✓ | - | - |
| | DEPTNO | VARCHAR2 | 10 | - | - | - | ✓ | - | - |

7.

Screenshot of the Oracle APEX SQL Workshop interface showing the execution of a SQL query.

SQL Commands:

```
1 DESCRIBE DEPT50;
2 SELECT*FROM DEPT50;
3
```

Results:

| EMPNO | EMPLOYEE | DEPTNO |
|-------|----------|--------|
| 176 | EMANUEL | 50 |
| 5 | PARTHI | 50 |

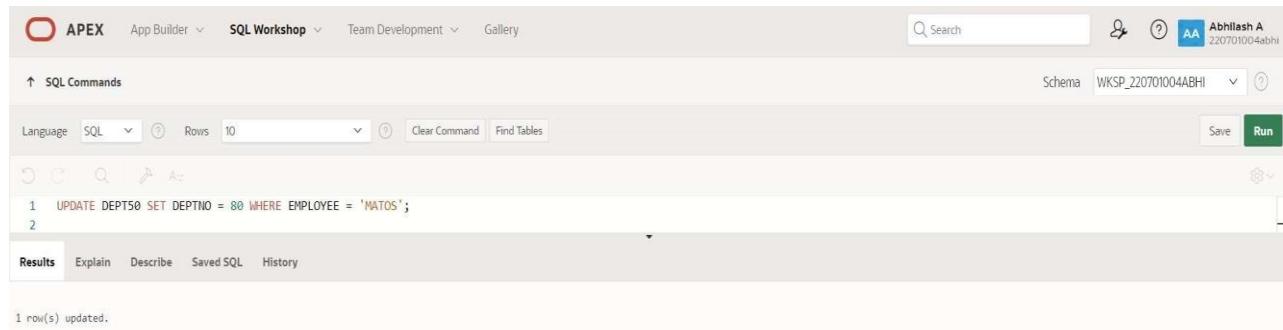
8.

7. Attempt to reassign Matos to department 80.

QUERY:

UPDATE DEPT50 SET DEPTNO = 80 WHERE EMPLOYEE = 'MATOS';

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The top right shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. The command entered is:

```
1 UPDATE DEPT50 SET DEPTNO = 80 WHERE EMPLOYEE = 'MATOS';
2
```

The 'Results' tab is selected, showing the message '1 row(s) updated.'

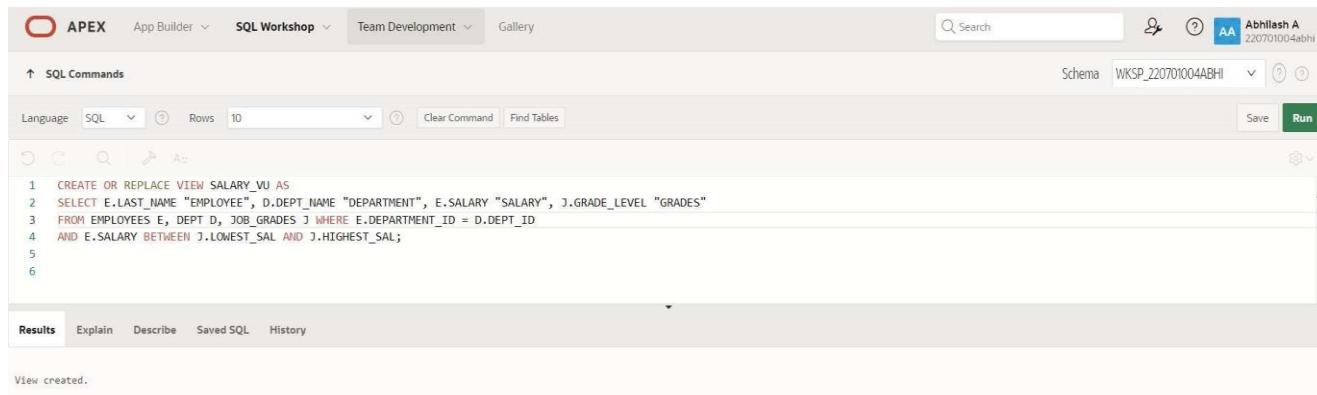
9.

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
E.LAST_NAME "EMPLOYEE", D.DEPARTMENT
"DEPARTMENT", E.SALARY "SALARY",
J.GRADE_LEVEL "GRADES" FROM EMPLOYEES E,
DEPARTMENT D, JOB_GRADES J WHERE
E.DEPARTMENT_ID = D.DEPARTMENT_ID AND E.SALARY
BETWEEN J.LOWEST_SAL AND J.HIGHEST_SAL;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side of the header shows the user 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 CREATE OR REPLACE VIEW SALARY_VU AS
2 SELECT E.LAST_NAME "EMPLOYEE", D.DEPARTMENT "DEPARTMENT", E.SALARY "SALARY", J.GRADE_LEVEL "GRADES"
3 FROM EMPLOYEES E, DEPT D, JOB_GRADES J WHERE E.DEPARTMENT_ID = D.DEPARTMENT_ID
4 AND E.SALARY BETWEEN J.LOWEST_SAL AND J.HIGHEST_SAL;
5
6
```

Below the code, the 'Results' tab is selected, showing the message 'View created.'

| Evaluation Procedure | Marks awarded |
|----------------------|---------------|
| Query(5) | |
| Execution (5) | |
| Viva(5) | |

10.

| | |
|-------------------|--|
| Total (15) | |
| Faculty Signature | |

RESULT :

11.

INTRODUCTION TO CONSTRAINTS: NOT NULL AND UNIQUE CONSTRAINTS

EX-NO : 12

DATE:

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

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

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

Ans:

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

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

Ans:

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

12.

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

Ans:

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

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

Ans:

| Global Fast Foods global_locations Table | | | | | | |
|--|------|----------|--------|-----------|-------|----------|
| NAME | TYPE | DataType | LENGTH | PRECISION | SCALE | NULLABLE |
| id | pk | NUMBER | 6 | 0 | | 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.

Ans:

| 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 |

13.

| | | | | | | |
|-------------------|----|----------|----|---|--|-----|
| 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.

Ans:

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

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

Ans:

Table Created.

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

Ans:

14.

```
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.

15.

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

Ans:

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

PRIMARY KEY, FOREIGN KEY, AND CHECK CONSTRAINTS

1. What is the purpose of a
• PRIMARY KEY

16.

- FOREIGN KEY
- CHECK CONSTRAINT

Ans:

a. PRIMARY KEY

Uniquely identify each row in table.

b. FOREIGN KEY

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

c. CHECK CONSTRAINT

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

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

Ans:

animal_id NUMBER(6) - **PRIMARY KEY**

name VARCHAR2(25)

license_tag_number NUMBER(10) - **UNIQUE**

admit_date DATE -**NOT NULL**

adoption_id NUMBER(5),

vaccination_date DATE -**NOT NULL**

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

Ans:

CREATE TABLE animals

(animal_id NUMBER(6,0) CONSTRAINT anl_anl_id_pk PRIMARY KEY ,

17.

```
name VARCHAR2(25),
license_tag_number NUMBER(10,0) CONSTRAINT anl_l_tag_num_uk UNIQUE,
admit_date DATE CONSTRAINT anl_adt_dat_nn NOT NULL ENABLE,
adoption_id NUMBER(5,0),
vaccination_date DATE CONSTRAINT anl_vcc_dat_nn NOT NULL ENABLE
);
```

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

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

Ans:

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

Ans:

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

a. ON DELETE CASCADE

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

18.

b. ON DELETE SET NULL

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

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

Ans:

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

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

RESULT :

19.

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?



The screenshot shows a database query results table with the following structure:

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

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

The table displays two rows of data from the view_d_songs. The first row has ID 47, Song Title "Hurrah for Today", and ARTIST "The Jubilant Trio". The second row has ID 49, Song Title "Lets Celebrate", and ARTIST "The Celebrants".

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
```

20.

```
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);
```

21.

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

22.

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
```

23.

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.

24.

```
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?

25.

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

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

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

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

3. When will an index be created automatically?

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

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

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

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

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

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

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

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

CREATE SYNONYM dj_tracks FOR d_track_listings;

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

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

26.

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 | |

27.

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:



28.

Schema WKSP_220701004ABHI  

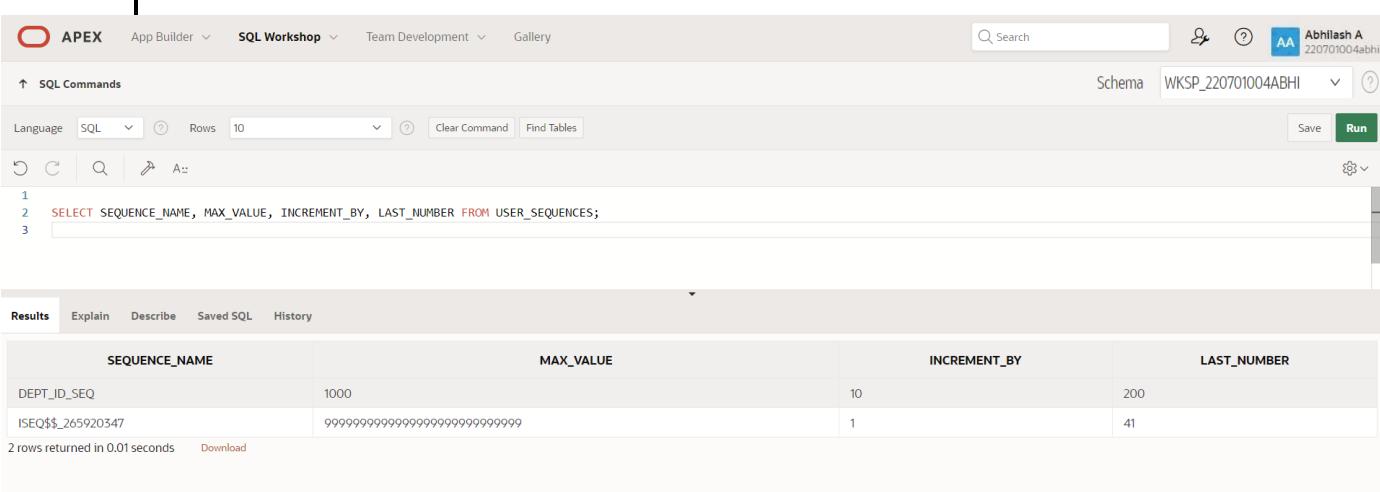
29.

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

QUERY:

```
SELECT SEQUENCE_NAME, MAX_VALUE, INCREMENT_BY, LAST_NUMBER FROM
USER_SEQUENCES;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile 'Abhilash A' and the schema 'WKSP_220701004ABHI'. The SQL Commands section contains the query:

```
1
2  SELECT SEQUENCE_NAME, MAX_VALUE, INCREMENT_BY, LAST_NUMBER FROM USER_SEQUENCES;
3
```

The Results tab is selected, displaying the query results in a table:

| SEQUENCE_NAME | MAX_VALUE | INCREMENT_BY | LAST_NUMBER |
|--------------------|----------------------------------|--------------|-------------|
| DEPT_ID_SEQ | 1000 | 10 | 200 |
| ISEQ\$\$_265920347 | 99999999999999999999999999999999 | 1 | 41 |

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

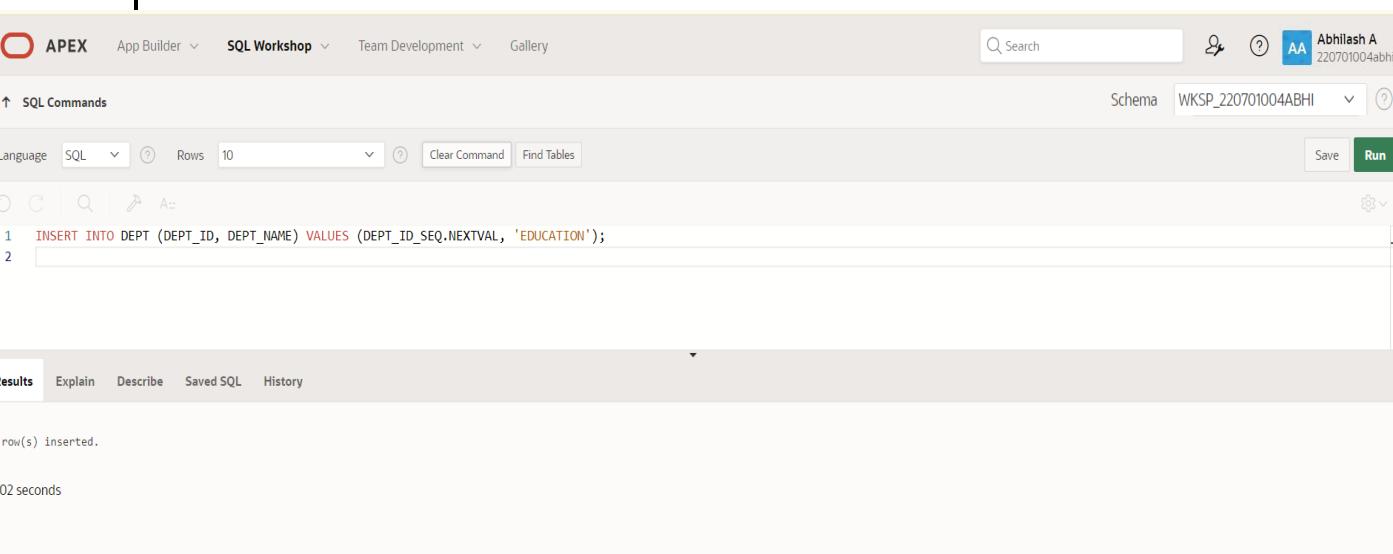
30.

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

QUERY:

```
INSERT INTO DEPT VALUES (DEPT_ID_SEQ.NEXTVAL, 'EDUCATION');
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile of Abhilash A (220701004abhi). The main area is titled 'SQL Commands' with a schema of WKSP_220701004ABHI. The command entered is:

```
1  INSERT INTO DEPT (DEPT_ID, DEPT_NAME) VALUES (DEPT_ID_SEQ.NEXTVAL, 'EDUCATION');
```

The command is run, and the results show:

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

Execution time: 02 seconds.

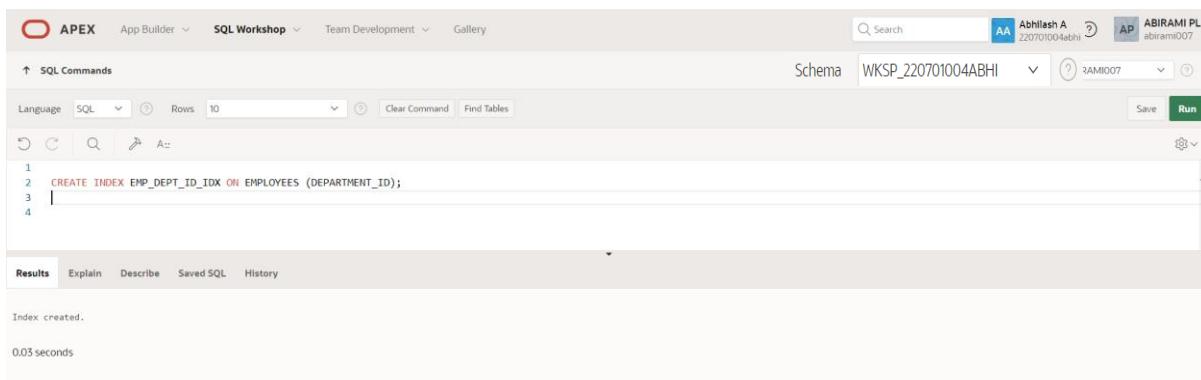
31.

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

QUERY:

```
CREATE INDEX EMP_DEPT_ID_IDX ON EMPLOYEES (DEPARTMENT_ID);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL code:

```
1 CREATE INDEX EMP_DEPT_ID_IDX ON EMPLOYEES (DEPARTMENT_ID);
```

The code is highlighted in red, indicating it is a SQL command. The schema dropdown shows 'WKSP_220701004ABHI'. The results pane below shows the output:

Index created.
0.03 seconds

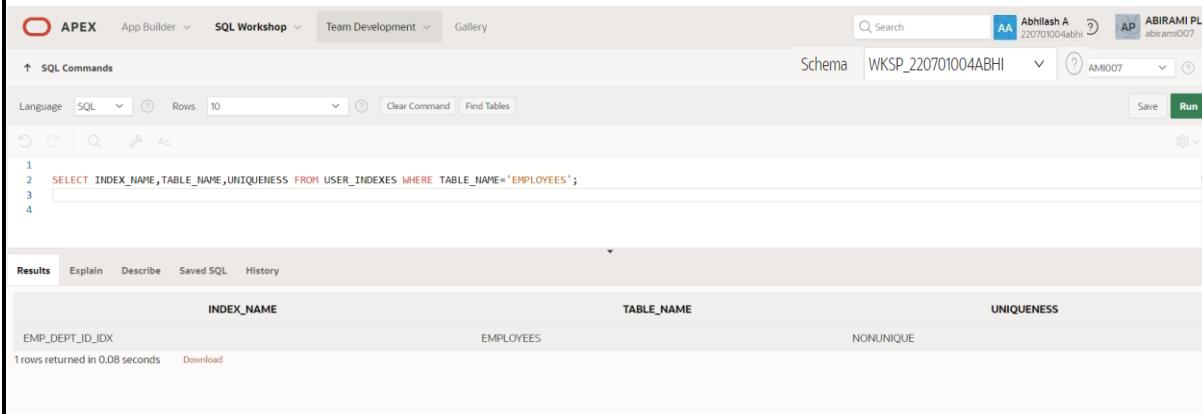
32.

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

QUERY:

```
SELECT INDEX_NAME, TABLE_NAME,UNIQUENESS FROM USER_INDEXES WHERE TABLE_NAME='EMPLOYEES';
```

OUTPUT:



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

```
1 SELECT INDEX_NAME, TABLE_NAME,UNIQUENESS FROM USER_INDEXES WHERE TABLE_NAME='EMPLOYEES';
```

The results table shows the following data:

| INDEX_NAME | TABLE_NAME | UNIQUENESS |
|-----------------|------------|------------|
| EMP_DEPT_ID_IDX | EMPLOYEES | NONUNIQUE |

1 rows returned in 0.08 seconds

33.

| 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?

34.

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.

35.

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

Team 2 revokes the privilege.

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

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

Team 1 executes this INSERT statement.

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

Team 2 executes this INSERT statement.

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

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

RESULT:

36.

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:

37.

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables

Schema WKSP_220701004ABHI

Save Run

1 2 3 4 5 6

```
1
2  DECLARE
3  |  incentive  NUMBER(8,2);
4  BEGIN
5  |  SELECT salary * 0.12 INTO incentive
6  |  FROM employees
```

Results Explain Describe Saved SQL History

incentive = 10800
Statement processed.

2 seconds

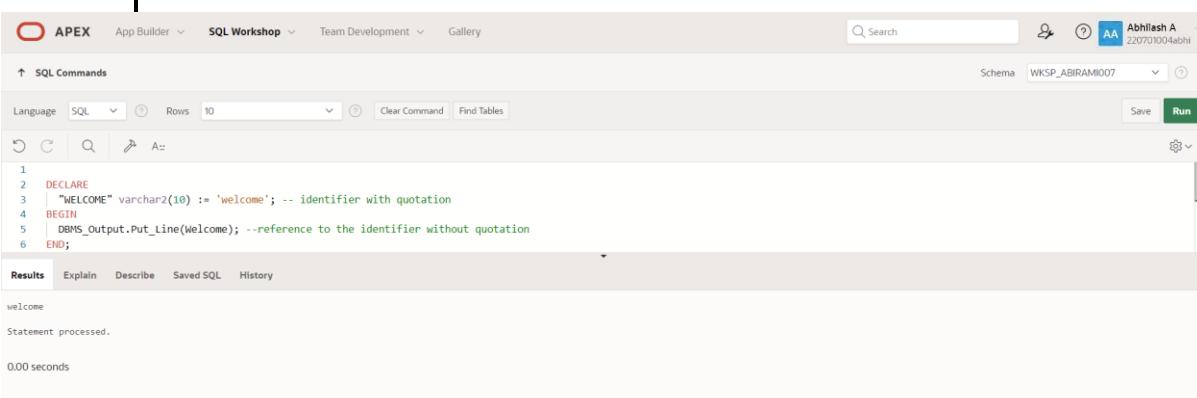
38.

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'; -- identifier with quotation
BEGIN
  DBMS_OUTPUT.Put_Line(Welcome); --reference to the identifier without quotation
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL editor contains the following PL/SQL block:

```
1 DECLARE
2   "WELCOME" varchar2(10) := 'welcome'; -- identifier with quotation
3 BEGIN
4   DBMS_OUTPUT.Put_Line(Welcome); --reference to the identifier without quotation
5 END;
```

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

```
welcome
Statement processed.
```

Execution time: 0.00 seconds

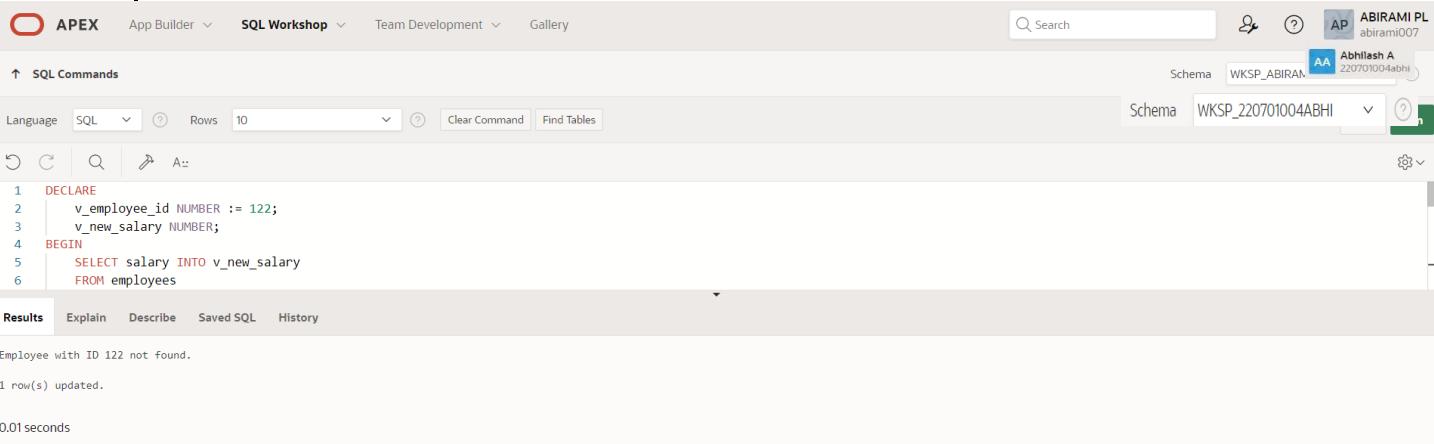
39.

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

QUERY:

```
DECLARE
  v_employee_id NUMBER := 122;
  v_new_salary NUMBER;
BEGIN
  SELECT salary INTO v_new_salary
  FROM employees
  WHERE employee_id = v_employee_id;
  v_new_salary := v_new_salary * 1.1;
  UPDATE employees
  SET salary = v_new_salary
  WHERE employee_id = v_employee_id;
  COMMIT;
  DBMS_OUTPUT.PUT_LINE('Salary of employee ' || v_employee_id || ' has been adjusted.');
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Employee with ID ' || v_employee_id || ' not found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
    ROLLBACK;
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is active. The top right shows the user 'Abhirami PL' and the schema 'WKSP_ABIRAN'. The main area is a code editor with the following content:

```
1  DECLARE
2    v_employee_id NUMBER := 122;
3    v_new_salary NUMBER;
4  BEGIN
5    SELECT salary INTO v_new_salary
6    FROM employees
```

The results pane at the bottom shows the output of the code execution:

```
Employee with ID 122 not found.
1 row(s) updated.
0.01seconds
```

40.

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.

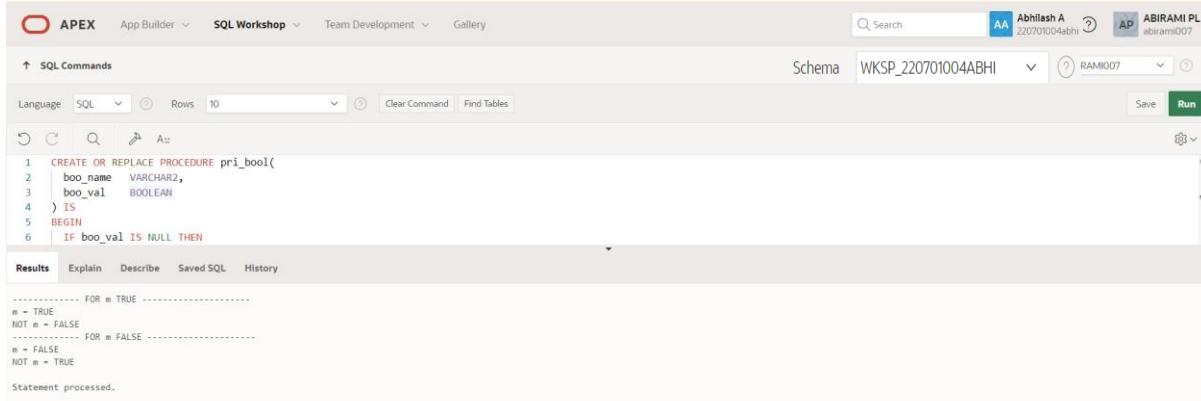
QUERY:

```
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;
/
DECLARE
  PROCEDURE pri_not_m (
    m BOOLEAN
  ) IS
  BEGIN
    pri_bool ('m', m);
    pri_bool ('NOT m', NOT m);
  END pri_not_m;
BEGIN
  DBMS_OUTPUT.PUT_LINE('----- FOR m TRUE -----');
  pri_not_m (TRUE);
  DBMS_OUTPUT.PUT_LINE('----- FOR m FALSE -----');
  pri_not_m (FALSE);
END;

```

OUTPUT:

41.



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The top right shows user profiles for 'Abhilash A' and 'ABIRAMI PL'. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701004ABHI'. The SQL editor contains the following PL/SQL code:

```
1 CREATE OR REPLACE PROCEDURE pri_bool(
2     boo_name  VARCHAR2,
3     boo_val   BOOLEAN
4 ) IS
5 BEGIN
6     IF boo_val IS NULL THEN

```

The 'Results' tab is selected, showing the output of the procedure execution:

```
----- FOR m TRUE -----
m = TRUE
NOT m = FALSE
----- FOR m FALSE -----
m = FALSE
NOT m = TRUE

Statement processed.
```

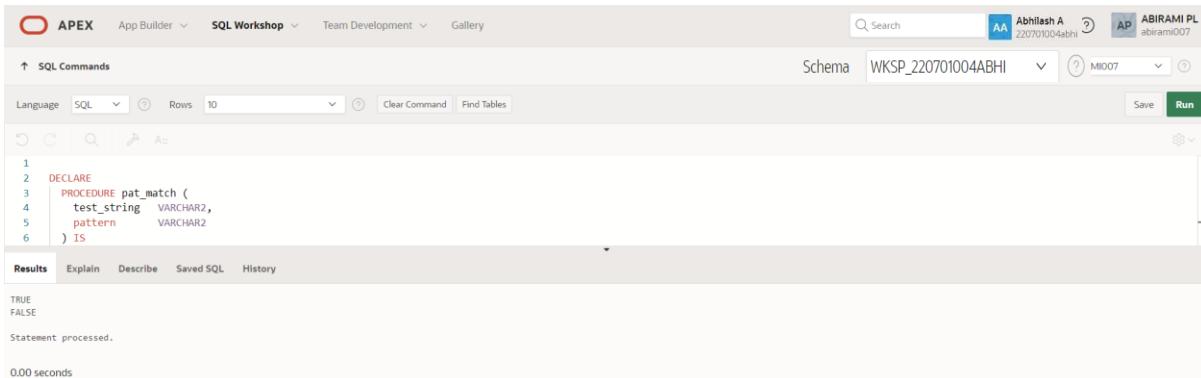
5. Write a PL/SQL block to describe the usage of LIKE operator including wildcard characters and escape character.

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:

42.



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The top right shows user profiles for 'Abhilash A' and 'ABIRAMI PL'. The main area is titled 'SQL Commands' with a 'Schema' dropdown set to 'WKSP_220701004ABHI'. The code editor contains the following PL/SQL code:

```
1
2  DECLARE
3      PROCEDURE pat_match (
4          test_string  VARCHAR2,
5          pattern      VARCHAR2
6      ) IS
```

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

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

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:

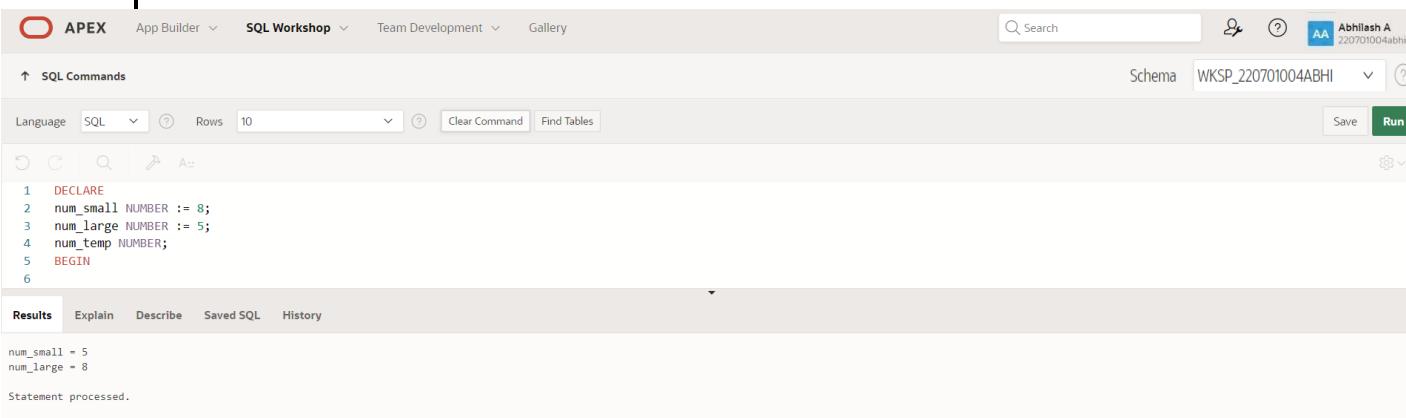
```
DECLARE
num_small NUMBER := 8;
num_large NUMBER := 5;
num_temp NUMBER;
BEGIN

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:

43.



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Workshop' tab is selected. The right side of the interface shows a user profile for 'Abhilash A' with the ID '220701004abhi'. The main area is a code editor with the following PL/SQL code:

```
1  DECLARE
2  num_small NUMBER := 8;
3  num_large NUMBER := 5;
4  num_temp NUMBER;
5  BEGIN
6
```

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

```
num_small = 5
num_large = 8

Statement processed.
```

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

QUERY:

```
DECLARE
  PROCEDURE test1 (sal_achieve NUMBER, target_qty NUMBER, emp_id NUMBER )
  IS
    incentive NUMBER := 0;
    updated VARCHAR2(3) := 'No';
  BEGIN
    IF sal_achieve > (target_qty + 200) THEN
      incentive := (sal_achieve - target_qty)/4;

      UPDATE employees
      SET salary = salary + incentive
      WHERE employee_id = emp_id;

      updated := 'Yes';
    END IF;
    DBMS_OUTPUT.PUT_LINE (
      'Table updated? ' || updated || ',' ||
      'incentive = ' || incentive || '!'
    )
  END;
```

```

44.

);
END test1;
BEGIN
  test1(2300, 2000, 144);
  test1(3600, 3000, 145);
END;
OUTPUT:

```



The screenshot shows the Oracle SQL Workshop interface. The SQL pane contains the following PL/SQL code:

```

1  DECLARE
2    PROCEDURE test1 (sal_achieve NUMBER, target_qty NUMBER, emp_id NUMBER )
3    IS
4      incentive  NUMBER := 0;
5      updated  VARCHAR2(3) := 'No';
6    BEGIN

```

The results pane shows the output of the procedure execution:

```

1  table updated? Yes, incentive = 75.
2  table updated? Yes, incentive = 150.
3  row(s) updated.

```

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

QUERY:

```

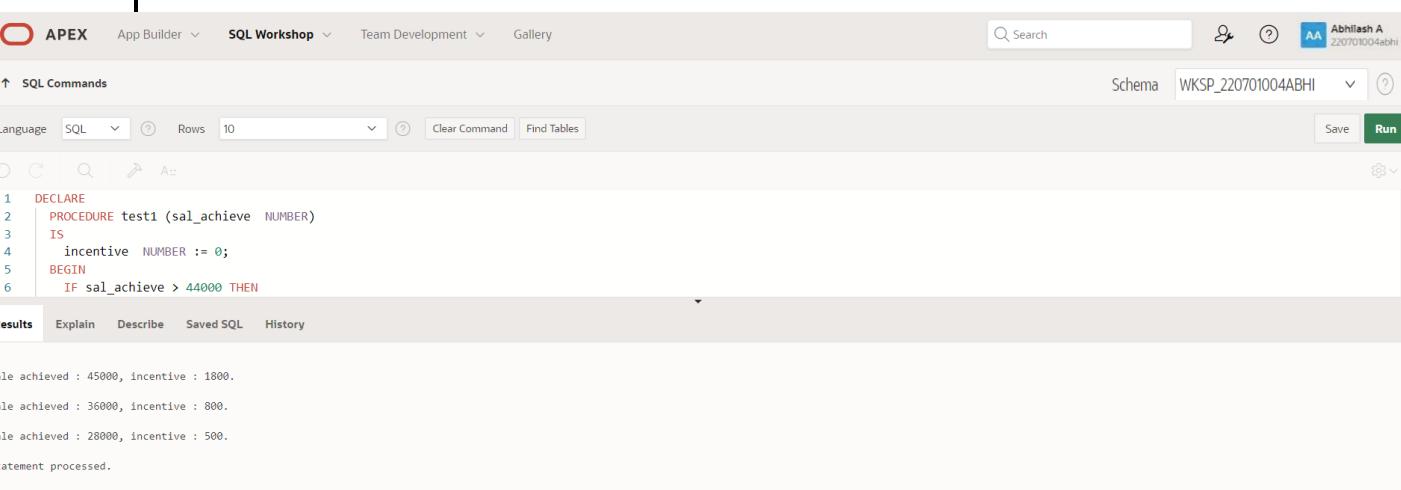
DECLARE
  PROCEDURE test1 (sal_achieve NUMBER)
  IS
    incentive NUMBER := 0;
  BEGIN
    IF sal_achieve > 44000 THEN
      incentive := 1800;
    ELSIF sal_achieve > 32000 THEN
      incentive := 800;
    ELSE
      incentive := 500;
    END IF;
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE (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

```

45.

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Abhilash A' (220701004abhi). The main area is titled 'SQL Commands' with a schema of 'WKSP_220701004ABHI'. The SQL editor contains the following PL/SQL code:

```
1 DECLARE
2   PROCEDURE test1 (sal_achieve  NUMBER)
3   IS
4     incentive  NUMBER := 0;
5   BEGIN
6     IF sal_achieve > 44000 THEN
```

The results tab shows the output of the code execution:

```
le achieved : 45000, incentive : 1800.
le achieved : 36000, incentive : 800.
le achieved : 28000, incentive : 500.
atement processed.
```

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

QUERY:

```
DECLARE
  v_emp_count NUMBER;
  v_vacancies NUMBER := 45;
BEGIN
  -- Count the number of employees in department 50
  SELECT COUNT(*)
  INTO v_emp_count
  FROM employees
  WHERE department_id = 50;

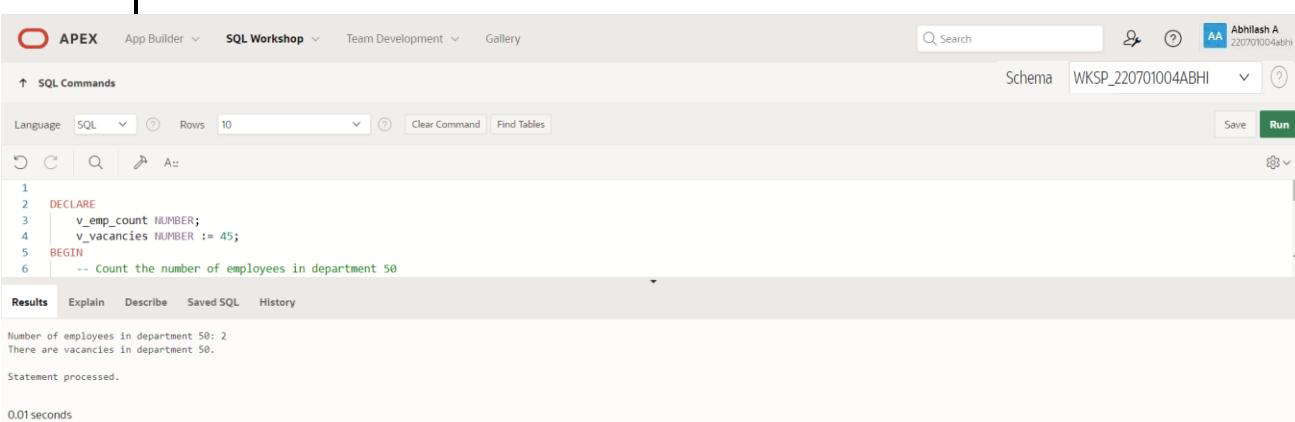
  -- Display the number of employees in department 50
  DBMS_OUTPUT.PUT_LINE('Number of employees in department 50: ' || v_emp_count);

  -- Check if there are any vacancies
  IF v_emp_count < v_vacancies THEN
```

46.

```
DBMS_OUTPUT.PUT_LINE('There are vacancies in department 50.');?>
ELSE
  DBMS_OUTPUT.PUT_LINE('There are no vacancies in department 50.');
END IF;
END;
```

OUTPUT:



APEX App Builder SQL Workshop Team Development Gallery

Search Schema: WKSP_220701004ABHI Run

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables

```
1 DECLARE
2   v_emp_count NUMBER;
3   v_vacancies NUMBER := 45;
4 BEGIN
5   -- Count the number of employees in department 50
6
```

Results Explain Describe Saved SQL History

Number of employees in department 50: 2
There are vacancies in department 50.

Statement processed.

0.01 seconds

47.

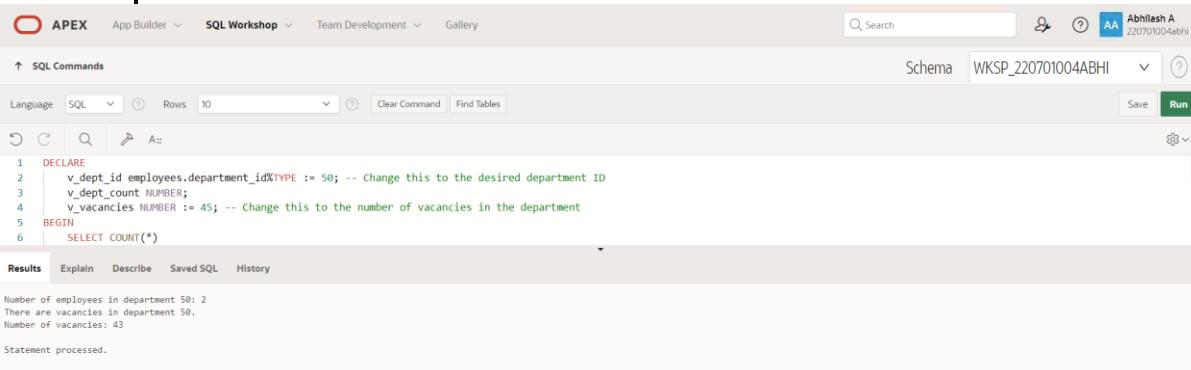
10. Write a PL/SQL program to count number of employees in a specific department and check whether this department have any vacancies or not. If any vacancies, how many vacancies are in that department.

QUERY:

DECLARE

```
v_dept_id employees.department_id%TYPE := 50; -- Change this to the desired department ID
v_dept_count NUMBER;
v_vacancies NUMBER := 45; -- Change this to the number of vacancies in the department
BEGIN
  SELECT COUNT(*)
  INTO v_dept_count
  FROM employees
  WHERE department_id = v_dept_id;
  DBMS_OUTPUT.PUT_LINE('Number of employees in department ' || v_dept_id || ':' ||
v_dept_count);
  IF v_dept_count < v_vacancies THEN
    DBMS_OUTPUT.PUT_LINE('There are vacancies in department ' || v_dept_id || '.');
    DBMS_OUTPUT.PUT_LINE('Number of vacancies: ' || (v_vacancies - v_dept_count));
  ELSE
    DBMS_OUTPUT.PUT_LINE('There are no vacancies in department ' || v_dept_id || '.');
  END IF;
END;
```

OUTPUT:



```
1  DECLARE
2    v_dept_id employees.department_id%TYPE := 50; -- Change this to the desired department ID
3    v_dept_count NUMBER;
4    v_vacancies NUMBER := 45; -- Change this to the number of vacancies in the department
5  BEGIN
6    SELECT COUNT(*)
```

Number of employees in department 50: 50
There are vacancies in department 50.
Number of vacancies: 43

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

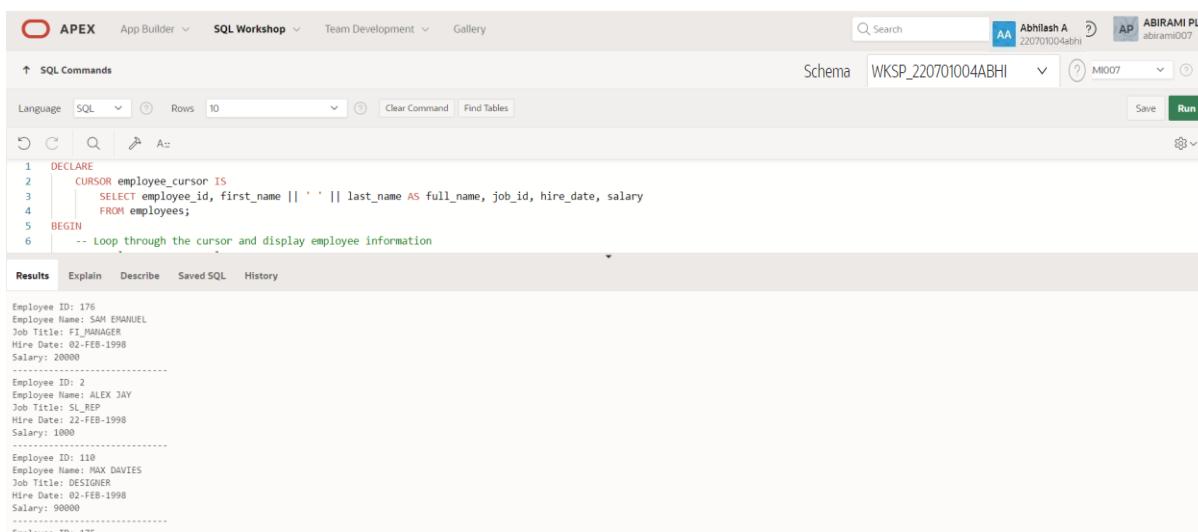
QUERY:

DECLARE

48.

```
CURSOR employee_cursor IS
  SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
  FROM employees;
BEGIN
  -- Loop through the cursor and display employee information
  FOR employee_rec IN employee_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
    DBMS_OUTPUT.PUT_LINE('Job Title: ' || employee_rec.job_id);
    DBMS_OUTPUT.PUT_LINE('Hire Date: ' || TO_CHAR(employee_rec.hire_date, 'DD-MON-YYYY'));
    DBMS_OUTPUT.PUT_LINE('Salary: ' || employee_rec.salary);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands pane contains the following PL/SQL code:

```
1  DECLARE
2    CURSOR employee_cursor IS
3      SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
4      FROM employees;
5  BEGIN
6    -- Loop through the cursor and display employee information
```

The Results pane displays the output of the code:

```
Employee ID: 176
Employee Name: SAM EMANUEL
Job Title: IT MANAGER
Hire Date: 02-FEB-1998
Salary: 20000
-----
Employee ID: 2
Employee Name: ALEX JAY
Job Title: SA REP
Hire Date: 02-FEB-1998
Salary: 10000
-----
Employee ID: 110
Employee Name: MAX DAVIES
Job Title: DESIGNER
Hire Date: 02-FEB-1998
Salary: 90000
-----
```

49.

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

QUERY:

DECLARE

```
  CURSOR employee_cursor IS
    SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, d.dept_name
    FROM employees e
    INNER JOIN department d ON e.department_id = d.dept_id;
```

BEGIN

```
  -- Loop through the cursor and display employee information
```

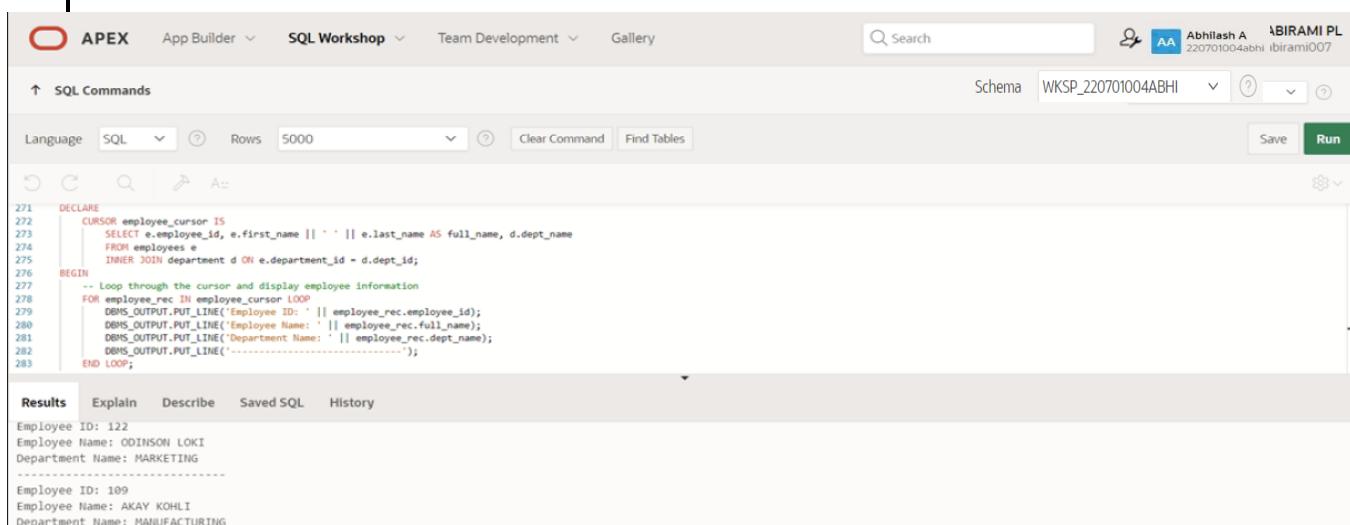
```
  FOR employee_rec IN employee_cursor LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
    DBMS_OUTPUT.PUT_LINE('Department Name: ' || employee_rec.dept_name);
    DBMS_OUTPUT.PUT_LINE('-----');
```

```
  END LOOP;
```

```
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown shows WKSP_220701004ABHI. The main area displays the following PL/SQL code:

```
271  DECLARE
272    CURSOR employee_cursor IS
273      SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, d.dept_name
274      FROM employees e
275      INNER JOIN department d ON e.department_id = d.dept_id;
276  BEGIN
277    -- Loop through the cursor and display employee information
278    FOR employee_rec IN employee_cursor LOOP
279      DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
280      DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
281      DBMS_OUTPUT.PUT_LINE('Department Name: ' || employee_rec.dept_name);
282      DBMS_OUTPUT.PUT_LINE('-----');
283    END LOOP;
```

The results pane shows the output for two employees:

| Employee ID | Employee Name | Department Name |
|-------------|---------------|-----------------|
| 122 | ODINSON LOKI | MARKETING |
| 109 | AKAY KOHLI | MANUFACTURING |

50.

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

QUERY:

DECLARE

```
1 CURSOR job_cursor IS
2     SELECT job_id, MIN(salary) AS min_salary
3     FROM employees
4     GROUP BY job_id;
```

BEGIN

-- Loop through the cursor and display job information

FOR job_rec IN job_cursor LOOP

```
    DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_rec.job_id);
```

```
    DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_rec.min_salary);
```

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

END LOOP;

END;

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Search Schema WKSP_220701004ABHI

Language SQL Rows 10 Save Run

SQL Commands

```
1 DECLARE
2     CURSOR job_cursor IS
3         SELECT job_id, MIN(salary) AS min_salary
4         FROM employees
5         GROUP BY job_id;
6 BEGIN
```

Results Explain Describe Saved SQL History

| Job ID | Minimum Salary |
|------------|----------------|
| FI_MANAGER | 20000 |
| ST_CLERK | 50000 |
| SL REP | 1000 |
| DESIGNER | 90000 |
| HR_MANAGER | 9000 |

Statement processed.

51.

52.

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

QUERY:

DECLARE

```
CURSOR employee_cursor IS
  SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.start_date
  FROM employees e
  JOIN job_history jh ON e.employee_id = jh.employee_id;
```

BEGIN

```
-- Loop through the cursor and display employee information
```

```
FOR employee_rec IN employee_cursor LOOP
```

```
  DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
```

```
  DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
```

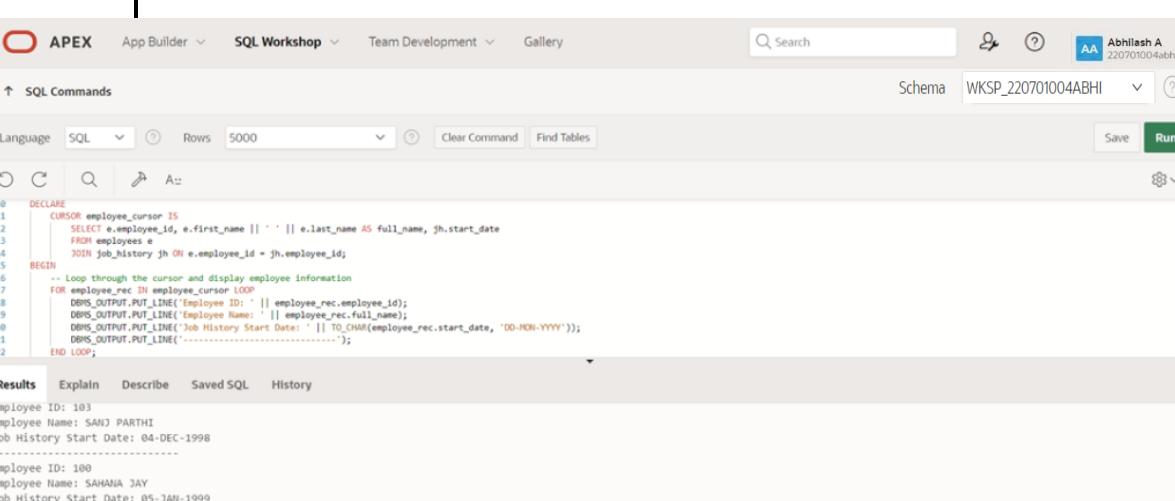
```
  DBMS_OUTPUT.PUT_LINE('Job History Start Date: ' || TO_CHAR(employee_rec.start_date
'DD-MON-YYYY'));
```

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

```
END LOOP;
```

END;

OUTPUT:



```
APEX App Builder SQL Workshop Team Development Gallery Search Schema WKSP_220701004ABHI
SQL Commands Rows 5000 Save Run
Language SQL Clear Command Find Tables
Results Explain Saved SQL History
Employee ID: 103
Employee Name: SANJU PARTHI
Job History Start Date: 04-DEC-1998
-----
Employee ID: 100
Employee Name: SAHANA JAY
Job History Start Date: 05-JAN-1999
```

53.

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

QUERY:

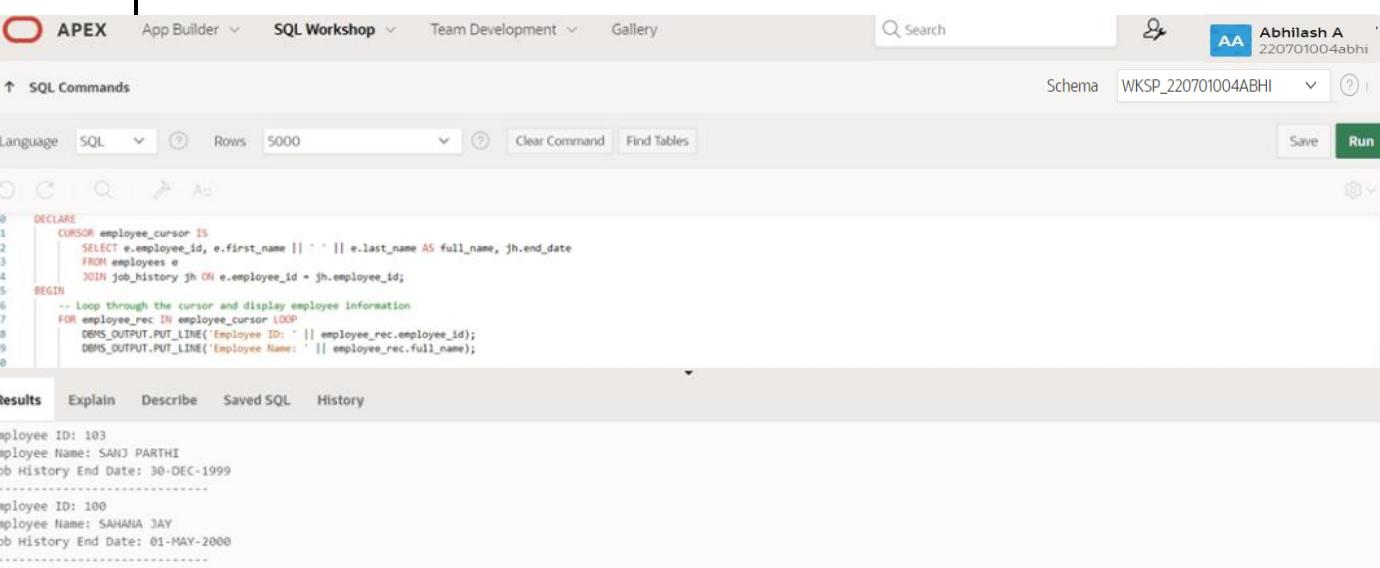
DECLARE

```
CURSOR employee_cursor IS
  SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.end_date
  FROM employees e
  JOIN job_history jh ON e.employee_id = jh.employee_id;
BEGIN
  -- Loop through the cursor and display employee information
  FOR employee_rec IN employee_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);

    -- Check if the end date is NULL (meaning the employee is currently in the job)
    IF employee_rec.end_date IS NULL THEN
      DBMS_OUTPUT.PUT_LINE('Job History End Date: (Still Employed)');
    ELSE
      DBMS_OUTPUT.PUT_LINE('Job History End Date: ' ||
TO_CHAR(employee_rec.end_date, 'DD-MON-YYYY'));
    END IF;

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

OUTPUT:



```
Employee ID: 100
Employee Name: SANJ PARTHI
Job History End Date: 30-DEC-1999
-----
Employee ID: 101
Employee Name: SAHANA JAY
Job History End Date: 01-MAY-2000
```

54.

| 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;

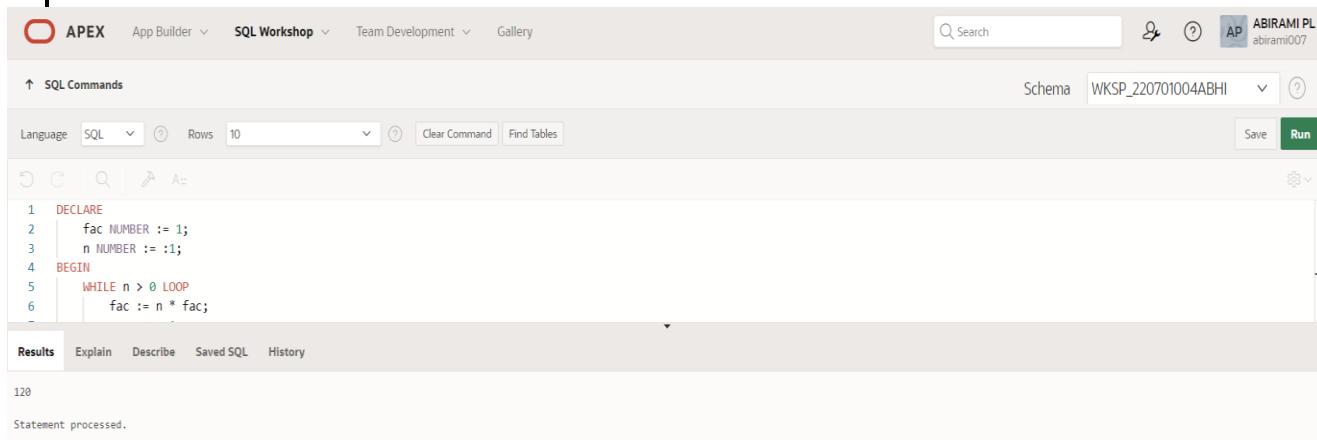
END LOOP;

 DBMS_OUTPUT.PUT_LINE(fac);

END;

55.

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a user profile for 'ABIRAMI PL' (abirami007). The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_220701004ABHI'. The code editor contains the following PL/SQL block:

```
1  DECLARE
2  |    fac NUMBER := 1;
3  |    n NUMBER := 1;
4  |    BEGIN
5  |      WHILE n > 0 LOOP
6  |        fac := n * fac;

```

The 'Results' tab is selected, showing the output '120' and the message 'Statement processed.'

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;
```

AA Abhilash A
220701004ab

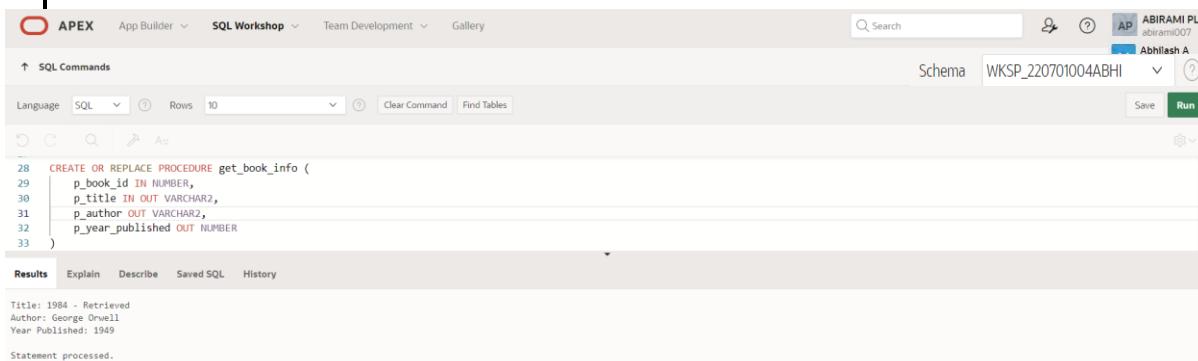
56.

```
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:



APEX App Builder SQL Workshop Team Development Gallery

Search Schema ABIRAMI PL abirami007 Abhilash A

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

```
28 CREATE OR REPLACE PROCEDURE get_book_info (
29   p_book_id IN NUMBER,
30   p_title IN OUT VARCHAR2,
31   p_author OUT VARCHAR2,
32   p_year_published OUT NUMBER
33 )
```

Results Explain Describe Saved SQL History

```
Title: 1984
Author: George Orwell
Year Published: 1949
```

Statement processed.

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

57.

| | |
|-------------------|--|
| 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:

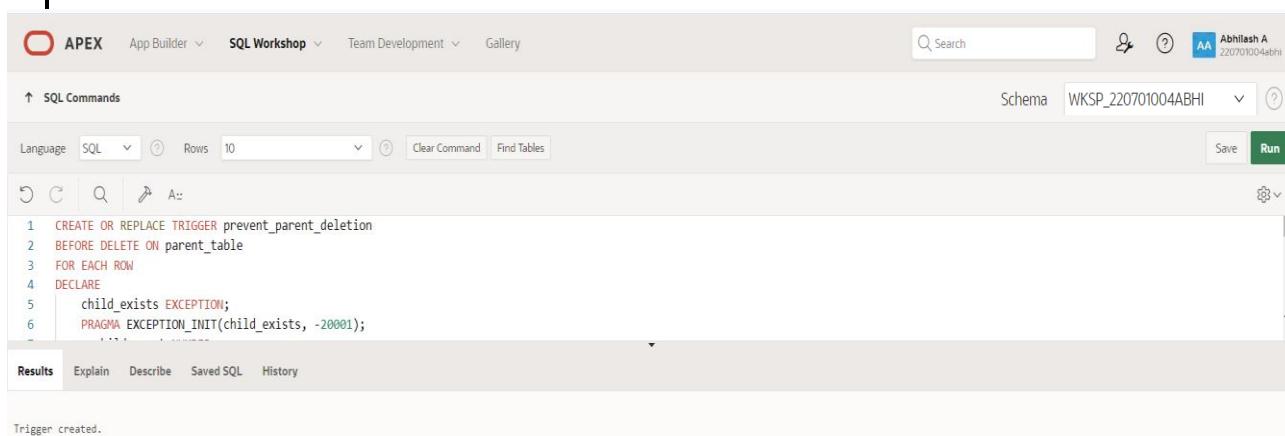
```
CREATE OR REPLACE TRIGGER prevent_parent_deletion
BEFORE DELETE ON parent_table
FOR EACH ROW
DECLARE
    child_exists EXCEPTION;
    PRAGMA EXCEPTION_INIT(child_exists, -20001);
    v_child_count NUMBER;
BEGIN
```

58.

```
SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id =  
:OLD.parent_id;  
IF v_child_count > 0 THEN  
    RAISE child_exists;  
END IF;  
EXCEPTION  
WHEN child_exists THEN  
    RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child  
records exist.');
```

END;

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side of the header shows a user profile for 'Abhilash A' and a schema dropdown set to 'WKSP_220701004ABHI'. The main workspace is titled 'SQL Commands' and contains the following PL/SQL code:

```
1 CREATE OR REPLACE TRIGGER prevent_parent_deletion  
2 BEFORE DELETE ON parent_table  
3 FOR EACH ROW  
4 DECLARE  
5     child_exists EXCEPTION;  
6     PRAGMA EXCEPTION_INIT(child_exists, -20001);
```

The 'Results' tab is selected at the bottom, and the message 'Trigger created.' is displayed.

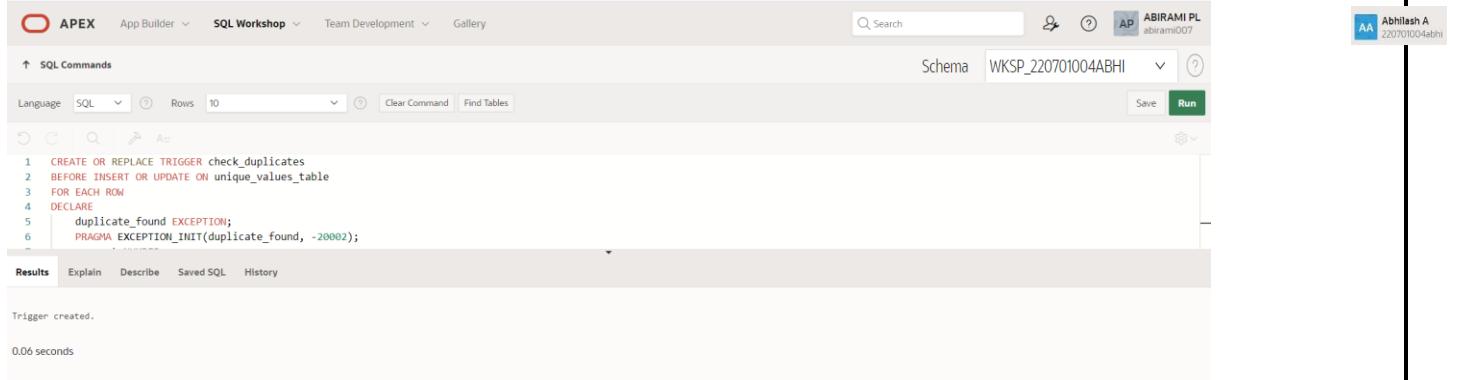
2.) Write a code in PL/SQL to create a trigger that checks for duplicate values in a specific column and raises an exception if found

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.');
```

59.

END;
OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The top right shows the user 'ABIRAMI PL' and 'abirami007'. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_220701004ABHI'. The code editor contains the following PL/SQL code:

```
1 CREATE OR REPLACE TRIGGER check_duplicates
2 BEFORE INSERT OR UPDATE ON unique_values_table
3 FOR EACH ROW
4 DECLARE
5     duplicate_found EXCEPTION;
6     PRAGMA EXCEPTION_INIT(duplicate_found, -20002);
```

The results pane shows the message 'Trigger created.' and a execution time of '0.06 seconds'.

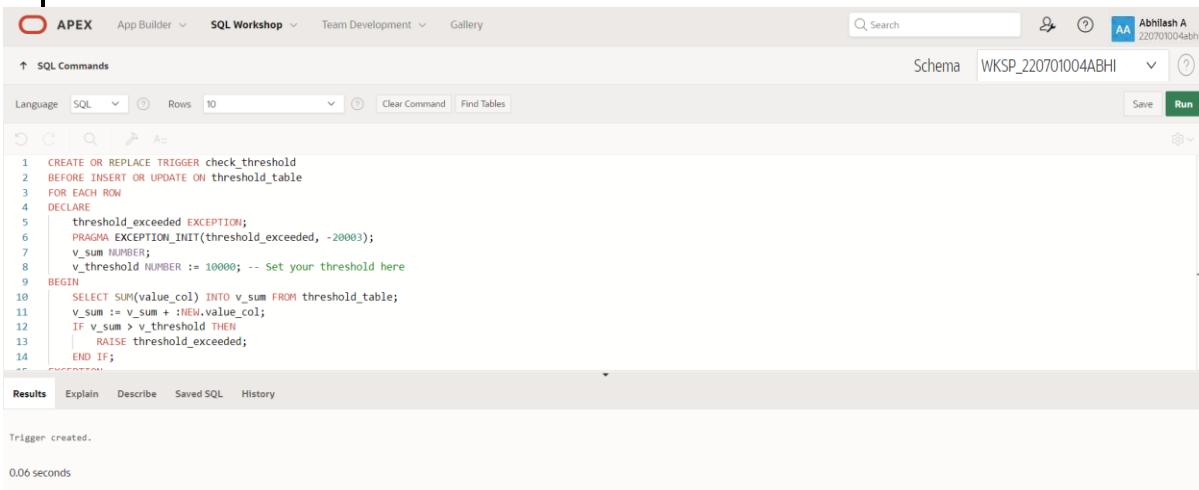
3.) Write a code in PL/SQL to create a trigger that restricts the insertion of new rows if the total of a column's values exceeds a certain threshold

QUERY:

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

OUTPUT:

60.



APEX App Builder SQL Workshop Team Development Gallery

Schema: WKSP_220701004ABHI

Language: SQL Rows: 10

SQL Commands

Clear Command Find Tables

Save Run

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

Trigger created.

0.06 seconds

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

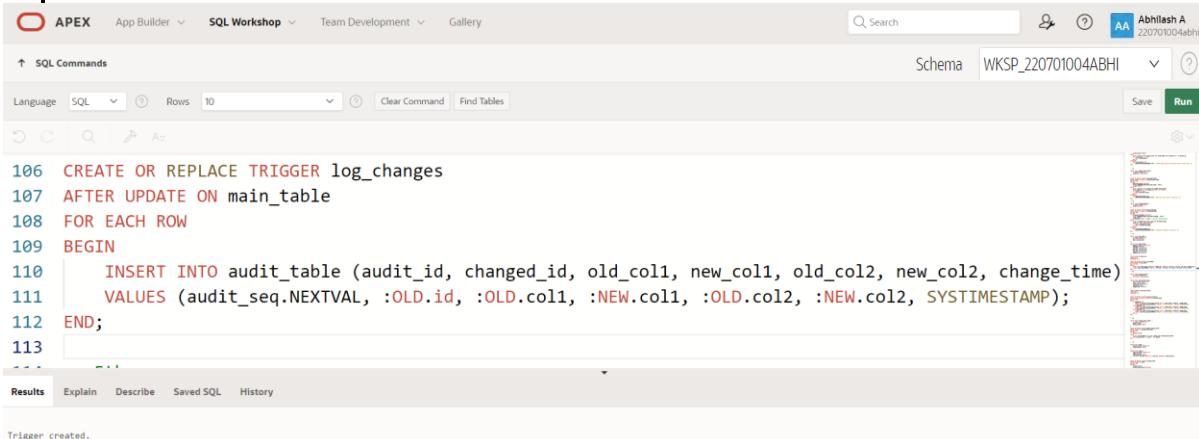
BEGIN

 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:



APEX App Builder SQL Workshop Team Development Gallery

Schema: WKSP_220701004ABHI

Language: SQL Rows: 10

SQL Commands

Clear Command Find Tables

Save Run

106 CREATE OR REPLACE TRIGGER log_changes
107 AFTER UPDATE ON main_table
108 FOR EACH ROW
109 BEGIN
110 INSERT INTO audit_table (audit_id, changed_id, old_col1, new_col1, old_col2, new_col2, change_time)
111 VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2, :NEW.col2, SYSTIMESTAMP);
112 END;
113

Trigger created.

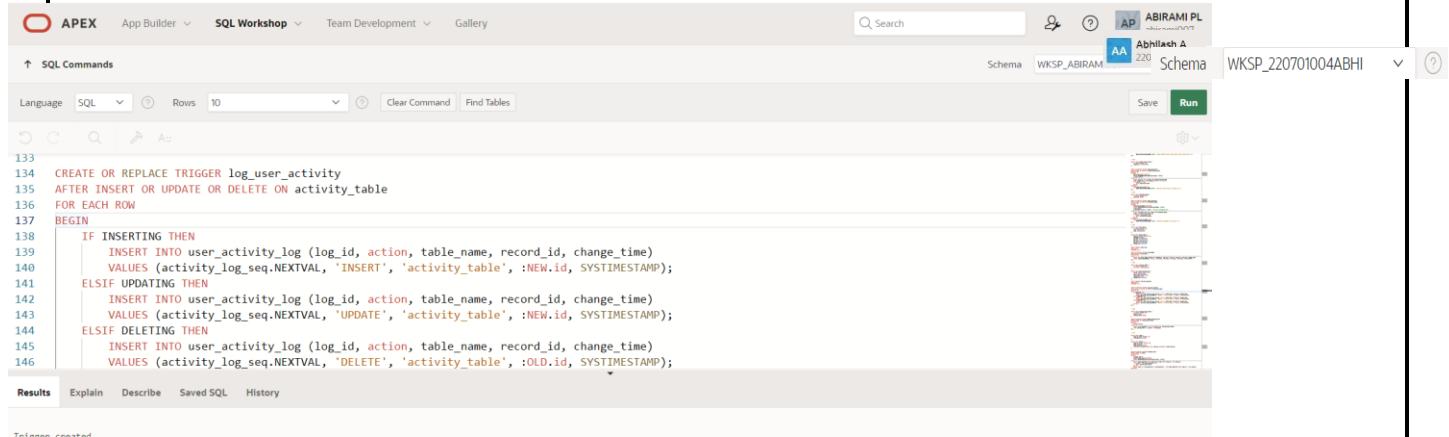
61.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side of the interface shows the schema 'WKSP_ABIRAM' and the user 'Abhilash A'. The main workspace displays the SQL code for the trigger creation, and the bottom status bar indicates 'Trigger created.'

```
133
134 CREATE OR REPLACE TRIGGER log_user_activity
135   AFTER INSERT OR UPDATE OR DELETE ON activity_table
136   FOR EACH ROW
137 BEGIN
138   IF INSERTING THEN
139     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
140       VALUES (activity_log_seq.NEXTVAL, 'INSERT', 'activity_table', :NEW.id, SYSTIMESTAMP);
141   ELSIF UPDATING THEN
142     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
143       VALUES (activity_log_seq.NEXTVAL, 'UPDATE', 'activity_table', :NEW.id, SYSTIMESTAMP);
144   ELSIF DELETING THEN
145     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
146       VALUES (activity_log_seq.NEXTVAL, 'DELETE', 'activity_table', :OLD.id, SYSTIMESTAMP);
```

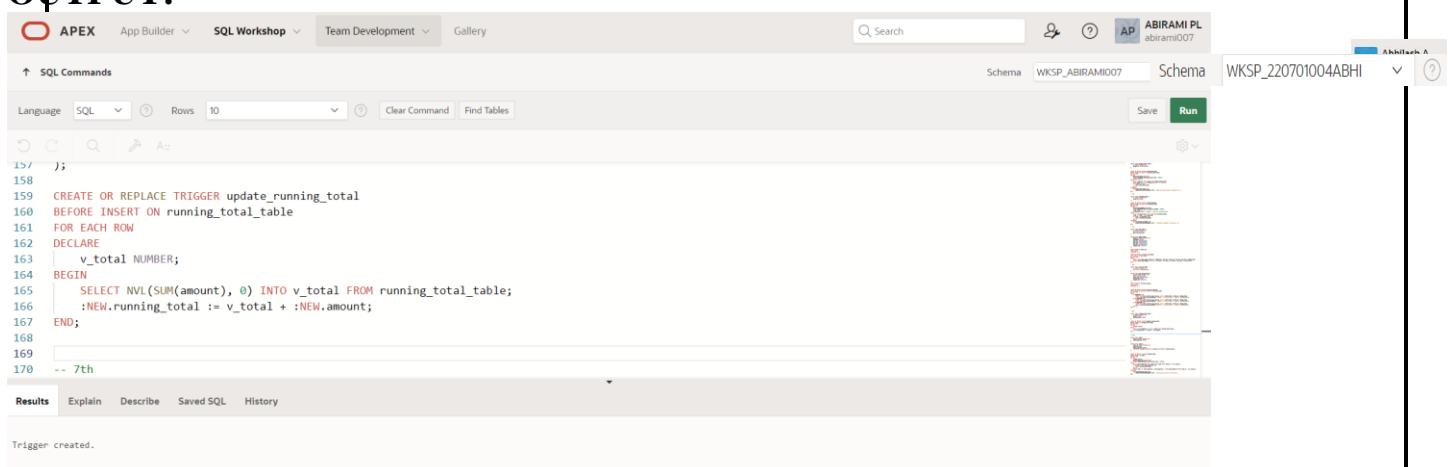
6.) Write a code in PL/SQL to implement a trigger that automatically calculates and updates a running total column for a table whenever new rows are inserted

62.

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the schema 'ABIRAMI PL' and 'abirami007'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The code area contains the PL/SQL code for the trigger, with line numbers 157 through 178. The code is as follows:

```
157  );
158
159 CREATE OR REPLACE TRIGGER update_running_total
160 BEFORE INSERT ON running_total_table
161 FOR EACH ROW
162 DECLARE
163   |  v_total NUMBER;
164   BEGIN
165     SELECT NVL(SUM(amount), 0) INTO v_total FROM running_total_table;
166     :NEW.running_total := v_total + :NEW.amount;
167   END;
168
169
170 -- 7th
```

The 'Results' tab at the bottom shows the message 'Trigger created.'.

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

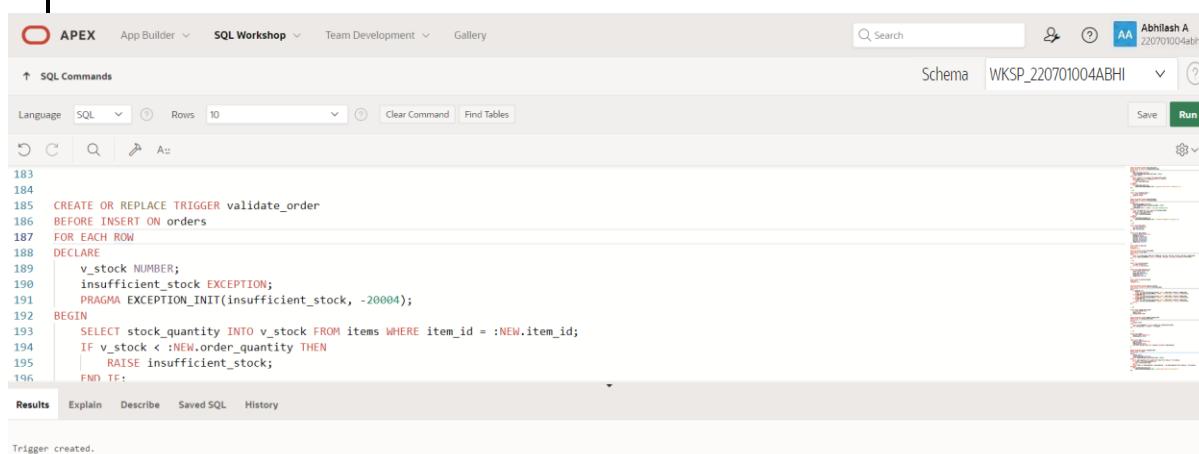
QUERY:

```
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
```

63.

```
SELECT stock_quantity INTO v_stock FROM items WHERE item_id = :NEW.item_id;
IF v_stock < :NEW.order_quantity THEN
    RAISE insufficient_stock;
END IF;
UPDATE items SET stock_quantity = stock_quantity - :NEW.order_quantity WHERE
item_id = :NEW.item_id;
EXCEPTION
WHEN insufficient_stock THEN
    RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');
END;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side of the interface shows a schema tree for 'WKSP_220701004ABHI'. The main workspace displays the SQL code for the trigger:

```
183
184 CREATE OR REPLACE TRIGGER validate_order
185 BEFORE INSERT ON orders
186 FOR EACH ROW
187 DECLARE
188     v_stock NUMBER;
189     insufficient_stock EXCEPTION;
190     PRAGMA EXCEPTION_INIT(insufficient_stock, -20004);
191 BEGIN
192     SELECT stock_quantity INTO v_stock FROM items WHERE item_id = :NEW.item_id;
193     IF v_stock < :NEW.order_quantity THEN
194         RAISE insufficient_stock;
195     END IF;
```

The status bar at the bottom indicates 'Trigger created.'

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

RESULT:

64.

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:

65.

```
:
  {
    $or: [
      { name: /wil/ },
      { cuisine: { $nin: ['American', 'Chinese'] } }
    ],
    {
      restaurant_id: 1,
      name: 1,
      borough: 1,
      cuisine: 1
    }
  );
}

[
  {
    _id: ObjectId('564c2d949eb21ad392f1d6de'),
    borough: 'Manhattan',
    cuisine: 'Other',
    name: '',
    restaurant_id: '50017887'
  },
  {
    _id: ObjectId('564c2d949eb21ad392f1d6ec'),
    borough: 'Brooklyn',
    cuisine: 'Other',
    name: '',
    restaurant_id: '50017910'
  },
  {
    _id: ObjectId('564c2d949eb21ad392f1d6ed'),
    borough: 'Manhattan',
    cuisine: 'Other',
    name: '',
    restaurant_id: '50017912'
  },
  {
    _id: ObjectId('564c2d949eb21ad392f1d6fb'),
    borough: 'Brooklyn',
    cuisine: 'Other',
    name: '',
    restaurant_id: '50017925'
  }
]
```

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:

```
abirami_07> db.restaurants.find( { grades: { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 } );
```

66.

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("2014-08-01T00:00:00Z")  
},  
{  
  "restaurant_id": 1,  
  "name": 1,  
  "grades": 1  

```

OUTPUT:

```
...  "grades.1.grade": "A",  
...  "grades.1.score": 9,  
...  "grades.1.date": ISODate("2014-08-01T00:00:00Z")  
},  
{  
  "restaurant_id": 1,  
  "name": 1,  
  "grades": 1  

```

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:

```
abirami_07> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, 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:

67.

```
[ {  
  address: {  
    building: '154',  
    coord: [ -73.9189064, 40.8654529 ],  
    street: 'Post Ave',  
    zipcode: '10034'  
  },  
  borough: 'Manhattan',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017887'  
},  
 {  
  address: {  
    building: '508',  
    coord: [ -73.999813, 40.683876 ],  
    street: 'Henry St',  
    zipcode: '11231'  
  },  
  borough: 'Brooklyn',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017910'  
},  
 {  
  address: {  
    building: '15',  
    coord: [ -73.9966882, 40.7139264 ],  
    street: 'Division St',  
    zipcode: '10002'  
  },  
  borough: 'Manhattan',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017912'  
},  
 {  
  address: {  
    building: '4704',  
    coord: [ -74.013391, 40.64943 ],  
    street: '3Rd Ave',  
    zipcode: '11220'  
  },  
  borough: 'Brooklyn',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017925'  
}]
```

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:

68.

```
[  
  {  
    address: {  
      building: '154',  
      coord: [ -73.9189064, 40.8654529 ],  
      street: 'Post Ave',  
      zipcode: '10034'  
    },  
    borough: 'Manhattan',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017887'  
  },  
  {  
    address: {  
      building: '508',  
      coord: [ -73.999813, 40.683876 ],  
      street: 'Henry St',  
      zipcode: '11231'  
    },  
    borough: 'Brooklyn',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017910'  
  },  
  {  
    address: {  
      building: '15',  
      coord: [ -73.9966882, 40.7139264 ],  
      street: 'Division St',  
      zipcode: '10002'  
    },  
    borough: 'Manhattan',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017912'  
  },  
  {  
    address: {  
      building: '4704',  
      coord: [ -74.013391, 40.64943 ],  
      street: '3Rd Ave',  
      zipcode: '11220'  
    },  
    borough: 'Brooklyn',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017925'  
},  
]
```

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:

69.

```
[  
  {  
    address: {  
      building: '154',  
      coord: [ -73.9189064, 40.8654529 ],  
      street: 'Post Ave',  
      zipcode: '10034'  
    },  
    borough: 'Manhattan',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017887'  
  },  
  {  
    address: {  
      building: '15',  
      coord: [ -73.9966882, 40.7139264 ],  
      street: 'Division St',  
      zipcode: '10002'  
    },  
    borough: 'Manhattan',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017912'  
  },  
  {  
    address: {  
      building: '508',  
      coord: [ -73.999813, 40.683876 ],  
      street: 'Henry St',  
      zipcode: '11231'  
    },  
    borough: 'Brooklyn',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017916'  
  },  
  {  
    address: {  
      building: '4704',  
      coord: [ -74.013391, 40.64943 ],  
      street: '3Rd Ave',  
      zipcode: '11220'  
    },  
    borough: 'Brooklyn',  
    cuisine: 'Other',  
    grades: [],  
    name: '',  
    restaurant_id: '50017925'  
},  
]
```

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:

70.

```
[ {  
  _id: ObjectId('564c2d949eb21ad392f1d6de'),  
  address: {  
    building: '154',  
    coord: [ -73.9189064, 40.8654529 ],  
    street: 'Post Ave',  
    zipcode: '10034'  
  },  
  borough: 'Manhattan',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017887'  
},  
{  
  _id: ObjectId('564c2d949eb21ad392f1d6ec'),  
  address: {  
    building: '508',  
    coord: [ -73.999813, 40.683876 ],  
    street: 'Henry St',  
    zipcode: '11231'  
  },  
  borough: 'Brooklyn',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017910'  
},  
{  
  _id: ObjectId('564c2d949eb21ad392f1d6ed'),  
  address: {  
    building: '15',  
    coord: [ -73.9966882, 40.7139264 ],  
    street: 'Division St',  
    zipcode: '10002'  
  },  
  borough: 'Manhattan',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017912'  
},  
{  
  _id: ObjectId('564c2d949eb21ad392f1d6f5'),  
  address: {  
    building: '4704',  
    coord: [ -74.013391, 40.64943 ],  
    street: '3Rd Ave',  
    zipcode: '11220'  
  },  
  borough: 'Brooklyn',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017925'  
}]
```

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:

71.

```
[ {  
  _id: ObjectId('564c2d949eb21ad392f1d6de'),  
  address: {  
    building: '154',  
    coord: [ -73.9189064, 40.8654529 ],  
    street: 'Post Ave',  
    zipcode: '10034'  
  },  
  borough: 'Manhattan',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017887'  
},  
{  
  _id: ObjectId('564c2d949eb21ad392f1d6ec'),  
  address: {  
    building: '508',  
    coord: [ -73.999813, 40.683876 ],  
    street: 'Henry St',  
    zipcode: '11231'  
  },  
  borough: 'Brooklyn',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017910'  
},  
{  
  _id: ObjectId('564c2d949eb21ad392f1d6ed'),  
  address: {  
    building: '15',  
    coord: [ -73.9966882, 40.7139264 ],  
    street: 'Division St',  
    zipcode: '10002'  
  },  
  borough: 'Manhattan',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017912'  
},  
{  
  _id: ObjectId('564c2d949eb21ad392f1d6f5'),  
  address: {  
    building: '4784',  
    coord: [ -74.013391, 40.64943 ],  
    street: '3Rd Ave',  
    zipcode: '11220'  
  },  
  borough: 'Brooklyn',  
  cuisine: 'Other',  
  grades: [],  
  name: '',  
  restaurant_id: '50017925'  
}]
```

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:

```
abirami_07> db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 });
```

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:

72.

```
abirami_07> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

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:

```
abirami_07> db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

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:

```
abirami_07> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
```

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:

```
abirami_07> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
```

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:

```
abirami_07> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

16. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

QUERY:

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

OUTPUT:

```
abirami_07> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

17. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

QUERY:

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

OUTPUT:

73.

```
abirami_07> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
```

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:

```
abirami_07> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
```

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:

```
abirami_07> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })
```

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:

```
abirami_07> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

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:

```
abirami_07> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

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:

```
abirami_07> 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"] } })
```

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:

```
abirami_07> db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
```

74.

RESULT:

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

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:

75.

```
[  
  {  
    _id: ObjectId('573a1390f29313caabcd4135'),  
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',  
    genres: [ 'Short' ],  
    runtime: 1,  
    cast: [ 'Charles Kayser', 'John Ott' ],  
    num_mflix_comments: 1,  
    title: 'Blacksmith Scene',  
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',  
    countries: [ 'USA' ],  
    released: ISODate('1893-05-09T00:00:00.000Z'),  
    directors: [ 'William K.L. Dickson' ],  
    rated: 'UNRATED',  
    awards: { wins: 1, nominations: 0, text: '1 win.' },  
    lastupdated: '2015-08-26 00:03:50.133000000',  
    year: 1893,  
    imdb: { rating: 6.2, votes: 1189, id: 5 },  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 3, numReviews: 184, meter: 32 },  
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')  
    }  
  }  
]
```

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:

76.

```
[ {  
  _id: ObjectId('665448ceab180f34b9cdcdf6'),  
  id: ObjectId('573a1390f22313caab9cd5967'),  
  plot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',  
  genres: [ 'Action', 'Adventure', 'Crime' ],  
  runtime: 399,  
  rated: 'NOT RATED',  
  cast: [ 'Musidora', 'éduard Mathé', 'Marcel Lévesque', 'Jean Aymé' ],  
  poster: 'https://m.media-amazon.com/images/M/MV5BMTC1NTY3NDIzNL5BMl5BanBnXkFzTgwNTIyODg5MTE@._V1_SY1000_SX677_AL.jpg',  
  title: 'Les vampires',  
  fullplot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',  
  languages: [ 'French' ],  
  released: ISODate('1916-11-23T00:00:00.000Z'),  
  directors: [ 'Louis Feuillade' ],  
  writers: [ 'Louis Feuillade' ],  
  awards: { wins: 0, nominations: 1, text: '1 nomination.' },  
  lastupdated: '2015-09-02 00:24:27.333000000',  
  year: 1915,  
  imdb: { rating: 6.8, votes: 2878, id: 6206 },  
  countries: [ 'France' ],  
  type: 'movie',  
  tomatoes: {  
    viewer: { rating: 3.8, numReviews: 2118, meter: 82 },  
    dvd: ISODate('2000-05-16T00:00:00.000Z'),  
    critic: { rating: 8.8, numReviews: 13, meter: 100 },  
    lastUpdated: ISODate('2015-09-15T17:02:33.000Z'),  
    rotten: 0,  
    fresh: 13  
  }  
}  
]
```

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

QUERY:

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

OUTPUT:

```
[ {  
  _id: ObjectId('573a1390f29313caabcd4135'),  
  plot: 'Three men hammer on an anvil and pass a bottle of beer around.',  
  genres: [ 'Short' ],  
  runtime: 1,  
  cast: [ 'Charles Kayser', 'John Ott' ],  
  num_mflix_comments: 1,  
  title: 'Blacksmith Scene',  
  fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',  
  countries: [ 'USA' ],  
  released: ISODate('1893-05-09T00:00:00.000Z'),  
  directors: [ 'William K.L. Dickson' ],  
  rated: 'UNRATED',  
  awards: { wins: 1, nominations: 0, text: '1 win.' },  
  lastupdated: '2015-08-26 00:03:50.133000000',  
  year: 1893,  
  imdb: { rating: 6.2, votes: 1189, id: 5 },  
  type: 'movie',  
  tomatoes: {  
    viewer: { rating: 3, numReviews: 184, meter: 32 },  
    lastUpdated: ISODate('2015-06-28T18:34:09.000Z')  
  }  
}
```

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:

77.

```
[ {  
  _id: ObjectId('573a1390f29313caabcd4135'),  
  plot: 'Three men hammer on an anvil and pass a bottle of beer around.',  
  genres: [ 'Short' ],  
  runtime: 1,  
  cast: [ 'Charles Kayser', 'John Ott' ],  
  num_mflix_comments: 1,  
  title: 'Blacksmith Scene',  
  fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',  
  countries: [ 'USA' ],  
  released: ISODate('1893-05-09T00:00:00.000Z'),  
  directors: [ 'William K.L. Dickson' ],  
  rated: 'UNRATED',  
  awards: { wins: 1, nominations: 0, text: '1 win.' },  
  lastupdated: '2015-08-26 00:03:50.133000000',  
  year: 1893,  
  imdb: { rating: 6.2, votes: 1189, id: 5 },  
  type: 'movie',  
  tomatoes: {  
    viewer: { rating: 3, numReviews: 184, meter: 32 },  
    lastUpdated: ISODate('2015-06-28T18:34:09.000Z')  
  }  
}  
]
```

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:

```
{
  _id: ObjectId('573a1990f29313caabcd4135'),
  plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
  genres: ['Short'],
  runtime: 1,
  cast: ['Charles Kaysen', 'John Ott'],
  num_mflix_comments: 1,
  title: 'Blacksmith Scene',
  fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
  countries: ['USA'],
  released: ISODate('1893-05-09T00:00:00.000Z'),
  directors: ['William K.L. Dickson'],
  rated: 'NOT RATED',
  awards: { wins: 1, nominations: 0, text: '1 win.' },
  lastupdated: '2015-05-26 00:01:58.133000000',
  year: 1893,
  imdb: { rating: 6.2, votes: 1189, id: 5 },
  type: 'movie',
  tomatoes: {
    viewer: { rating: 3, numReviews: 184, meter: 32 },
    lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
  }
},
{
  _id: ObjectId('665048eacab180f34b9cdccdf7'),
  id: ObjectId('573a1991f29313caabcd7a34'),
  plot: 'A kept woman runs into her one-time fiancé and finds herself torn between love and comfort.',
  genres: ['Drama', 'Romance'],
  runtime: 78,
  rated: 'TV-PG',
  cast: [
    'Edna Purviance',
    'Clarence Geldart',
    'Carl Miller',
    'Lydia Knott'
  ],
  num_mflix_comments: 3,
  poster: 'https://m.media-amazon.com/images/M/MV5BjJiMTU2NGQtNWRKni80ZjExLWEzMTUTMmNiNTU0NzRlMTAJXkEyXkFqcGdeQXVyNjUwNzk3NDc0._V1_SV1000_SX677_AL.jpg',
  title: 'A Woman of Paris: A Drama of Fate',
  fullplot: 'Marie St. Clair believes she has been jilted by her artist fiance Jean when he fails to meet her at the railway station. She goes off to Paris alone. A year later, mistress of wealthy Pierre Revel, she meets Jean again. Misinterpreting events she bounces back and forth between apparent security and true love.',
  countries: ['USA'],
  released: ISODate('1923-11-04T00:00:00.000Z'),
  directors: ['Charles Chaplin'],
  writers: ['Charles Chaplin'],
  awards: { wins: 1, nominations: 0, text: '1 win.' },
  lastupdated: '2015-05-02 00:22:09.303000000',
  year: 1923,
  imdb: { rating: 7.1, votes: 3179, id: 14624 },
  type: 'movie',
  tomatoes: {
    viewer: { rating: 3.7, numReviews: 886, meter: 78 },
    dvd: ISODate('2004-03-02T00:00:00.000Z'),
    critic: { rating: 7.4, numReviews: 11, meter: 91 },
    lastUpdated: ISODate('2015-05-23T18:34:44.000Z'),
    rotten: 1,
    production: 'Criterion Collection',
    fresh: 10
  }
},
{
  _id: ObjectId('66504905ab180f34b9cdccfb'),
  id: ObjectId('573a1991f29313caabcd8945'),
  plot: 'A married farmer falls under the spell of a slatternly woman from the city, who tries to convince him to drown his wife.',
  genres: ['Drama', 'Romance'],
  runtime: 94,
  rated: 'NOT RATED',
  cast: [
    'George O'Brien',
    'Janet Gaynor',
    'Margaret Livingston',
    'Basil Reising'
  ],
  num_mflix_comments: 1,
  poster: 'https://m.media-amazon.com/images/M/MV5BNDVkywM2lENzR1My00NWQhLTlNMjMtNDI2ODY0GVwRzJjXkEyXkFqcGdeQXVyNTgzNzU5MDI0._V1_SV1000_SX677_AL.jpg',
  title: 'Anas',
  fullplot: 'In this fable-morality subtitled "A Song of Two Humans", the "evil" temptress is a city woman who bewitches farmer Anses and tries to convince him to murder his neglect ed wife, Laure.',
  countries: ['USA'],
  released: ISODate('1927-11-04T00:00:00.000Z'),
  directors: ['F.W. Murnau'],
  writers: [
    'Carl Mayer (scenario)',
    'Hermann Sudermann (from an original theme by)',
    'Katherine Hilliker (titles)',
    'H. H. Caldwell (titles)'
  ]
}
```

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

QUERY:

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

OUTPUT:

79.

```
[  
  {  
    _id: ObjectId('573a1390f29313caabcd4135'),  
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',  
    genres: [ 'Short' ],  
    runtime: 1,  
    cast: [ 'Charles Kayser', 'John Ott' ],  
    num_mflix_comments: 1,  
    title: 'Blacksmith Scene',  
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',  
    countries: [ 'USA' ],  
    released: ISODate('1893-05-09T00:00:00.000Z'),  
    directors: [ 'William K.L. Dickson' ],  
    rated: 'UNRATED',  
    awards: { wins: 1, nominations: 0, text: '1 win.' },  
    lastupdated: '2015-08-26 00:03:50.133000000',  
    year: 1893,  
    imdb: { rating: 6.2, votes: 1189, id: 5 },  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 3, numReviews: 184, meter: 32 },  
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')  
    }  
  }  
]
```

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:

```
[  
  {  
    _id: ObjectId('665448ceab180f34b9cdcdf6'),  
    id: ObjectId('573a1390f22313caabcd5967'),  
    plot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',  
    genres: [ 'Action', 'Adventure', 'Crime' ],  
    runtime: 399,  
    rated: 'NOT RATED',  
    cast: [ 'Musidora', 'édouardMathè', 'Marcel Lèvesque', 'Jean Aymè' ],  
    poster: 'https://m.media-amazon.com/images/M/MV5BMTC1NTY3NDIzMl5BMl5BanBnXkFtZTgwNTIyODg5MTE@._V1_SY1000_SX677_AL.jpg',  
    title: 'Les vampires',  
    fullplot: 'An intrepid reporter and his loyal friend battle a bizarre secret society of criminals known as The Vampires.',  
    languages: [ 'French' ],  
    released: ISODate('1916-11-23T00:00:00.000Z'),  
    directors: [ 'Louis Feuillade' ],  
    writers: [ 'Louis Feuillade' ],  
    awards: { wins: 0, nominations: 1, text: '1 nomination.' },  
    lastupdated: '2015-09-02 00:24:27.333000000',  
    year: 1915,  
    imdb: { rating: 6.8, votes: 2878, id: 6206 },  
    countries: [ 'France' ],  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 3.8, numReviews: 2118, meter: 82 },  
      dvd: ISODate('2000-05-16T00:00:00.000Z'),  
      critic: { rating: 8.8, numReviews: 13, meter: 100 },  
      lastUpdated: ISODate('2015-09-15T17:02:33.000Z'),  
      rotten: 0,  
      fresh: 13  
    },  
  },  
  {  
    _id: ObjectId('573a1390f29313caabcd4135'),  
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',  
    genres: [ 'Short' ],  
    runtime: 1,  
    cast: [ 'Charles Kayser', 'John Ott' ],  
    num_mflix_comments: 1,  
    title: 'Blacksmith Scene',  
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',  
  },  
]
```

81.

```
countries: [ 'USA' ],
released: ISODate('1893-05-09T00:00:00.000Z'),
directors: [ 'William K.L. Dickson' ],
rated: 'UNRATED',
awards: { wins: 1, nominations: 0, text: '1 win.' },
lastupdated: '2015-08-26 00:03:50.133000000',
year: 1893,
imdb: { rating: 6.2, votes: 1189, id: 5 },
type: 'movie',
tomatoes: {
  viewer: { rating: 3, numReviews: 184, meter: 32 },
  lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
},
},
{
  _id: ObjectId('665448eeab180f34b9cdcdf7'),
  id: ObjectId('573a1391f29313caabcd7a34'),
  plot: 'A kept woman runs into her one-time fianc\u00e8 and finds herself torn between love and comfort.',
  genres: [ 'Drama', 'Romance' ],
  runtime: 78,
  rated: 'TV-PG',
  cast: [
    'Edna Purviance',
    'Clarence Geldart',
    'Carl Miller',
    'Lydia Knott'
  ],
  num_mflix_comments: 3,
  poster: 'https://m.media-amazon.com/images/M/MV5BZjJiMTU2NGQtNWRkNi00ZjExLWExMTUtMmNKTU0NzRlMTA3XkEyXkFqcGdeQXVyNjUwNzk3NDc@._V1_Sy1000_SX677_AL.jpg',
  title: 'A Woman of Paris: A Drama of Fate',
  fullplot: 'Marie St. Clair believes she has been jilted by her artist fiance Jean when he fails to meet her at the railway station. She goes off to Paris alone. A year later, mistress of wealthy Pierre Revel, she meets Jean again. Misinterpreting events she bounces back and forth between apparent security and true love.',
  countries: [ 'USA' ],
  released: ISODate('1923-11-04T00:00:00.000Z'),
  directors: [ 'Charles Chaplin' ],
  writers: [ 'Charles Chaplin' ],
  awards: { wins: 1, nominations: 0, text: '1 win.' },
  lastupdated: '2015-09-02 00:22:09.303000000',
  year: 1923,
  imdb: { rating: 7.1, votes: 3179, id: 14624 },
  type: 'movie',
  tomatoes: {
    viewer: { rating: 3.7, numReviews: 886, meter: 78 },
    dvd: ISODate('2004-03-02T00:00:00.000Z'),
    critic: { rating: 7.4, numReviews: 11, meter: 91 },
    lastUpdated: ISODate('2015-08-23T18:34:44.000Z'),
    rotten: 1,
    production: 'Criterion Collection',
    fresh: 10
  }
}
```

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:

82.

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:

83.

```
[  
  {  
    _id: ObjectId('66544905ab180f34b9cdcdf8'),  
    id: ObjectId('573a1391f29313caabcd8945'),  
    plot: 'A married farmer falls under the spell of a slatternly woman from the city, who tries to convince him to drown his wife.',  
    genres: [ 'Drama', 'Romance' ],  
    runtime: 94,  
    rated: 'NOT RATED',  
    cast: [  
      "George O'Brien",  
      'Janet Gaynor',  
      'Margaret Livingston',  
      'Bodil Rosing'  
    ],  
    num_mflix_comments: 1,  
    poster: 'https://m.media-amazon.com/images/M/MV5BNDVkYmYwM2ItNzRiMy00NWQ4LTlhMjMtNDI1ZDYyOGVmMzJjX  
KExXKFQcGdeQXVyNTgzMzU5MDI@._V1_SY1000_SX677_AL.jpg',  
    title: 'Sunrise',  
    fullplot: 'In this fable-morality subtitled "A Song of Two Humans", the "evil" temptress is a city  
woman who bewitches farmer Anses and tries to convince him to murder his neglected wife, Indre.',  
    countries: [ 'USA' ],  
    released: ISODate('1927-11-04T00:00:00.000Z'),  
    directors: [ 'F.W. Murnau' ],  
    writers: [  
      'Carl Mayer (scenario)',  
      'Hermann Sudermann (from an original theme by)',  
      'Katherine Hilliker (titles)',  
      'H.H. Caldwell (titles)'  
    ],  
    awards: {  
      wins: 5,  
      nominations: 1,  
      text: 'Won 3 Oscars. Another 2 wins & 1 nomination.'  
    },  
    lastupdated: '2015-09-12 00:26:13.493000000',  
    year: 1927,  
    imdb: { rating: 8.4, votes: 24480, id: 18455 },  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 4.4, numReviews: 9134, meter: 92 },  
      dvd: ISODate('2008-12-09T00:00:00.000Z'),  
      critic: { rating: 8.9, numReviews: 48, meter: 98 },  
      lastUpdated: ISODate('2015-09-10T19:15:02.000Z'),  
      consensus: 'Boasting masterful cinematography to match its well-acted, wonderfully romantic story line, Sunrise is perhaps the final -- and arguably definitive -- statement of the silent era.',  
      rotten: 1,  
      production: 'Fox Films',  
      fresh: 47  
    }  
  }  
]
```

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

QUERY:

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

OUTPUT:

84.

```
        {
            _id: ObjectId('573a1390f29313caabcd4135'),
            plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
            genres: [ 'Short' ],
            runtime: 1,
            cast: [ 'Charles Kayser', 'John Ott' ],
            num_mflix_comments: 1,
            title: 'Blacksmith Scene',
            fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
            countries: [ 'USA' ],
            released: ISODate('1893-05-09T00:00:00.000Z'),
            directors: [ 'William K.L. Dickson' ],
            rated: 'UNRATED',
            awards: { wins: 1, nominations: 0, text: '1 win.' },
            lastupdated: '2015-08-26 00:03:50.133000000',
            year: 1893,
            imdb: { rating: 6.2, votes: 1189, id: 5 },
            type: 'movie',
            tomatoes: {
                viewer: { rating: 3, numReviews: 184, meter: 32 },
                lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
            }
        },
        {
            _id: ObjectId('665448eeab180f34b9cdcdf7'),
            id: ObjectId('573a1391f29313caabcd7a34'),
            plot: 'A kept woman runs into her one-time fiance and finds herself torn between love and comfort.',
            genres: [ 'Drama', 'Romance' ],
            runtime: 78,
            rated: 'TV-PG',
            cast: [
                'Edna Purviance',
                'Clarence Geldart',
                'Carl Miller',
                'Lydia Knott'
            ],
            num_mflix_comments: 3,
            poster: 'https://m.media-amazon.com/images/M/MV5BZjJiMTU2NGQtNWRkNi00ZjExLWExMTUtMmNkNTU0NzRLMTA3XkEyXkFqcGdeQXVyNjUwNzk3NDc@._V1_SY1000_SX677_AL.jpg',
            title: 'A Woman of Paris: A Drama of Fate',
            fullplot: 'Marie St. Clair believes she has been jilted by her artist fiance Jean when he fails to meet her at the railway station. She goes off to Paris alone. A year later, mistress of wealthy Pierre Revel, she meets Jean again. Misinterpreting events she bounces back and forth between apparent security and true love.',
            countries: [ 'USA' ],
            released: ISODate('1923-11-04T00:00:00.000Z'),
            directors: [ 'Charles Chaplin' ],
            num_mflix_comments: 1
        }
    ]
}
```

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:

85.

```
[  
  {  
    _id: ObjectId('665448ceab180f34b9cdcdf6'),  
    genres: [ 'Action', 'Adventure', 'Crime' ],  
    runtime: 399,  
    cast: [ 'Musidora', 'éduardMathè', 'Marcel Lèvesque', 'Jean Aymè' ],  
    title: 'Les vampires',  
    languages: [ 'French' ],  
    released: ISODate('1916-11-23T00:00:00.000Z'),  
    directors: [ 'Louis Feuillade' ],  
    writers: [ 'Louis Feuillade' ],  
    awards: { wins: 0, nominations: 1, text: '1 nomination.' },  
    year: 1915,  
    countries: [ 'France' ]  
  },  
  {  
    _id: ObjectId('66544905ab180f34b9cdcdf8'),  
    genres: [ 'Drama', 'Romance' ],  
    runtime: 94,  
    cast: [  
      "George O'Brien",  
      'Janet Gaynor',  
      'Margaret Livingston',  
      'BodilRosing'  
    ],  
    title: 'Sunrise',  
    countries: [ 'USA' ],  
    released: ISODate('1927-11-04T00:00:00.000Z'),  
    directors: [ 'F.W. Murnau' ],  
    writers: [  
      'Carl Mayer (scenario)',  
      'Hermann Sudermann (from an original theme by)',  
      'Katherine Hilliker (titles)',  
      'H.H. Caldwell (titles)'  
    ],  
    awards: {  
      wins: 5,  
      nominations: 1,  
      text: 'Won 3 Oscars. Another 2 wins & 1 nomination.'  
    },  
    year: 1927  
  }  
]
```

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:

```
[  
  {  
    _id: ObjectId('573a1390f29313caabcd4135'),  
    genres: [ 'Short' ],  
    runtime: 1,  
    cast: [ 'Charles Kayser', 'John Ott' ],  
    title: 'Blacksmith Scene',  
    countries: [ 'USA' ],  
    released: ISODate('1893-05-09T00:00:00.000Z'),  
    directors: [ 'William K.L. Dickson' ],  
    writers: { wins: 1, nominations: 0, text: '1 win.' },  
    awards: { wins: 1, nominations: 0, text: '1 win.' },  
    year: 1893  
  }  
]
```

13.) Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that released on May 9, 1893.

86.

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:

```
[  
  {  
    _id: ObjectId('573a1390f29313caabcd4135'),  
    genres: [ 'Short' ],  
    runtime: 1,  
    cast: [ 'Charles Kayser', 'John Ott' ],  
    title: 'Blacksmith Scene',  
    countries: [ 'USA' ],  
    released: ISODate('1893-05-09T00:00:00.000Z'),  
    directors: [ 'William K.L. Dickson' ],  
    awards: { wins: 1, nominations: 0, text: '1 win.' },  
    year: 1893  
  }  
]  
abirami_07>  
abirami_07> db.movies.find( { released: ISODate("1893-05-09T00:00:00.000Z") }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )  
[  
  {  
    _id: ObjectId('573a1390f29313caabcd4135'),  
    title: 'Blacksmith Scene',  
    countries: [ 'USA' ],  
    released: ISODate('1893-05-09T00:00:00.000Z'),  
    directors: [ 'William K.L. Dickson' ]  
  }  
]
```

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:

```
[  
  {  
    _id: ObjectId('573a1390f29313caabcd4135'),  
    title: 'Blacksmith Scene',  
    countries: [ 'USA' ],  
    released: ISODate('1893-05-09T00:00:00.000Z'),  
    directors: [ 'William K.L. Dickson' ]  
  }  
]
```

87.

RESULT:

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