

Creating and Managing Tables

EX_NO:1

DATE:

1. Create the DEPARTMENTS 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 departments(id number(7),name varchar2(25));
```

OUTPUT:

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

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

In the Results pane, the output is:

```
Table created.  
0.02 seconds
```

At the bottom, the footer includes:

```
220701045@rajalakshmi.edu.in bharath en  
Copyright © 1999, 2023, Oracle and/or its affiliates.  
Oracle APEX 23.2.4
```

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

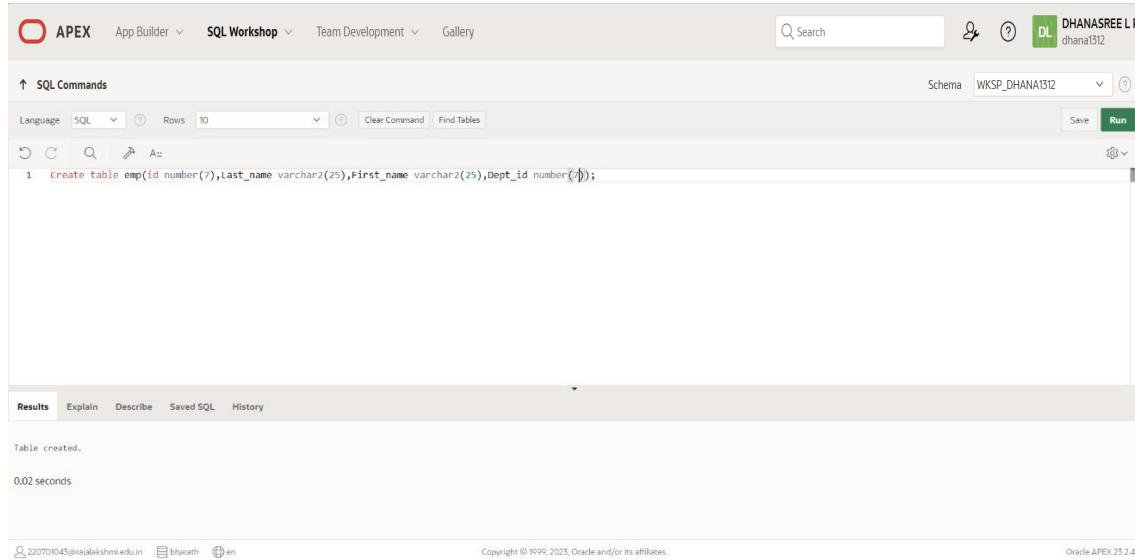
Column name	ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID
Key Type				
Nulls/Unique				
FK table				

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),Department_id number(7));

OUTPUT:



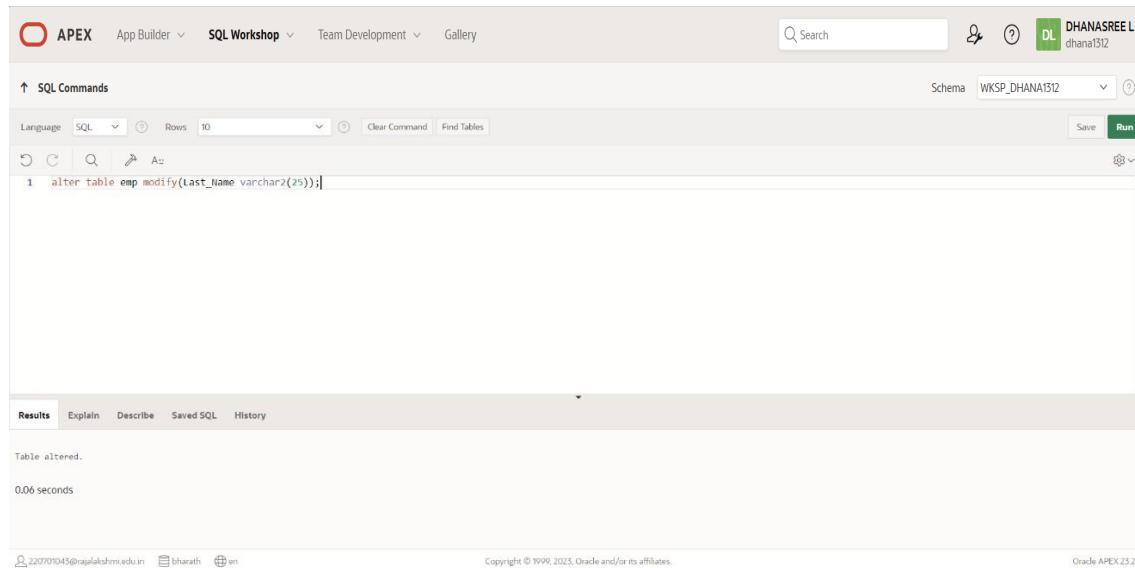
The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is chosen from a dropdown menu. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'create table emp(id number(7),Last_name varchar2(25),First_name varchar2(25),Dept_id number(7));'. The 'Run' button is visible at the top right of the command input field. Below the command, the results show 'Table created.' and a execution time of '0.02 seconds'. The bottom status bar indicates the user is 'bharath'.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is chosen from a dropdown menu. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'alter table emp modify(Last_Name varchar2(25));'. The 'Run' button is visible at the top right of the command input field. Below the command, the results show 'Table altered.' and a execution time of '0.06 seconds'. The bottom status bar indicates the user is 'bharath'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is chosen from a dropdown menu. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'create table employees2(id number(7),first_name varchar2(25),Last_name varchar2(25),Salary int,Department_id number(7));'. The 'Run' button at the bottom right of the command input field is highlighted in green. Below the command input, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active. The output shows the message 'Table created.' and a timestamp '0.03 seconds'.

This screenshot is identical to the one above, showing the creation of the 'employees2' table. The interface, SQL command, and resulting output ('Table created.' and '0.03 seconds') are all the same.

5. Drop the EMP table.

QUERY:

Drop table emp;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar has 'APEX' selected. 'SQL Workshop' is chosen from the dropdown. The 'SQL Commands' section contains the SQL command 'drop table emp;'. The 'Run' button is visible at the bottom right. The 'Results' tab is active, displaying the message 'Table dropped.' and a timestamp '0.06 seconds'.

6. Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename employees2 to emp;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon, and a schema dropdown set to 'WKSP_DHANA1312'. The main area is titled 'SQL Commands' with a sub-header '1 rename employees2 to emp;'. Below the command is a results section showing the output: 'Statement processed.' and '0.06 seconds'. The bottom of the screen displays copyright information for Oracle APEX 25.2.4.

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

QUERY:

comment on table departments is

'Department info'; comment on table emp is

'Employee info';

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon, and a schema dropdown set to 'WKSP_DHANA1312'. The main area is titled 'SQL Commands' with a sub-header '3 comment on table dept is 'Department info'' and '3 comment on table emp is 'Employee info''. Below the commands is a results section showing the output: 'Statement processed.' and '0.05 seconds'. The bottom of the screen displays copyright information for Oracle APEX 25.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name, Dhanasree LP, and session ID, dhana1312. The main workspace has a toolbar with various icons. The schema dropdown is set to WKSP_DHANA1312. The SQL command input field contains the query: `1 alter table emp drop column first_name;`. Below the input field, the results tab is selected, showing the output: `Table altered.`. The status bar at the bottom indicates the execution time was 0.05 seconds and provides copyright information for Oracle APEX 23.2.4.

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

RESULT:

MANIPULATING DATA

EX_NO:2

DATE:

1.Create MY_EMPLOYEE table with the following structure

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' and 'SQL Workshop' are selected. The schema is set to 'WKSP_DHANA1312'. The SQL command input field contains the following code:

```
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 shows the output of the query:

```
Table created.
```

Execution time: 0.03 seconds.

At the bottom, the footer includes user information (220701045@rajalakshmi.edu.in, bharath, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and version information (Oracle APEX 23.2.4).

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

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

QUERY:

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

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

OUTPUT:

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

```
1 insert into my_employee (id,Last_name,First_name,Userid,Salary)
2 values
3
4 (2,'Dancs','Betty','bdancs',860);
```

Below the code, the results are displayed under the 'Results' tab. It shows:

1 row(s) inserted.
0.01 seconds

At the bottom, the footer includes the URL 220707043@rajalakshmi.edu.in, the name bharath, and the language en. The copyright notice is Copyright © 1999, 2025, Oracle and/or its affiliates. The version is Oracle APEX 23.2.4.

3. Display the table with values.

QUERY:

```
Select*from my_employee;
```

OUTPUT:

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

```
1 select* from my_employee;
```

Below the code, the results are displayed under the 'Results' tab. It shows the following table:

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

At the bottom, the footer includes the URL 220707043@rajalakshmi.edu.in, the name bharath, and the language en. The copyright notice is Copyright © 1999, 2025, Oracle and/or its affiliates. The version is Oracle APEX 23.2.4.

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

QUERY:

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

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. On the right, a user profile for 'DHANASREE L P' is shown. The main area is titled 'SQL Commands' with a 'Run' button. The command entered is:

```
1 insert into my_employee values(4,'Newman','chad','Cnewman',750);
```

Below the command, the results section shows the output:

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

Execution time is listed as 0.01 seconds. The bottom of the page includes copyright information and a footer for Oracle APEX 25.2.4.

5. Make the data additions permanent.

QUERY:

```
Select*from my_employee;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar and user profile are identical. The SQL Workshop tab is selected. The command entered is:

```
1 select*from my_employee;
```

The results section displays the data from the 'my_employee' table:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	Cnewman	1000

3 rows returned in 0.04 seconds. The bottom of the page includes copyright information and a footer for Oracle APEX 25.2.4.

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 APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the schema 'WKSP_DHANA1312' and a user icon 'DL DHANASREE L P dhana1312'. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. It contains a text input field with the SQL command: 'update my_employee set last_name='Drexler' where id=3;'. Below the command, the results section shows '1 row(s) updated.' and '0.01 seconds'. The bottom footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

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

QUERY:

update my_employee set salary=1000 where salary<900;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the schema 'WKSP_DHANA1312' and a user icon 'DL DHANASREE L P dhana1312'. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. It contains a text input field with the SQL command: 'update my_employee set salary=1000 where salary<900;'. Below the command, the results section shows '3 row(s) updated.' and '0.01 seconds'. The bottom footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

8. Delete Betty dancs from MY_EMPLOYEE table.

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and a green button for 'DHNASREE LP' (dhanalakshmi1312). The main workspace is titled 'SQL Commands'. It contains a command line with the SQL statement: '1 delete from my_employee where last_name='Dancs';'. Below the command line, the results section shows the output: '1 row(s) deleted.' and '0.01 seconds'. At the bottom, it displays the URL '220701043@rajalakshmi.edu.in', session information 'bharath', and language 'en'. Copyright and version information are also present at the bottom right.

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

1 row(s) deleted.
0.01 seconds

9. Empty the fourth row of the emp table.

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar and user information are the same. The main workspace shows the command: '1 delete from emp where id=4;'. The results section shows '0 row(s) deleted.' and '0.03 seconds'. The bottom of the screen has the same footer information as the first screenshot.

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

0 row(s) deleted.
0.03 seconds

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

RESULT:

INCLUDING CONSTRAINTS

EX_NO:3

DATE:

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

QUERY:

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

OUTPUT:

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

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

Below the command, the results are displayed:

```
Table altered.
```

Execution time: 0.07 seconds

At the bottom, the copyright notice reads: Copyright © 1999, 2025, Oracle and/or its affiliates.

2. Create a PRIMARY KEY constraint to the DEPARTMENTS table using the ID colum. The constraint should be named at creation. Name the constraint my_department_id_pk.

QUERY:

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

OUTPUT:

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

```
1 alter table dept add constraint my_dept_id_pk primary key(id);
```

Below the command, the results are displayed:

```
Table altered.
```

Execution time: 0.07 seconds

At the bottom, the copyright notice reads: Copyright © 1999, 2025, Oracle and/or its affiliates.

3. Add a column DEPARTMENT_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_department_id_fk.

QUERY:

```
alter table emp add constraint my_emp_department_id_fk foreign key(department_id) references emp(id);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and a green button for 'DL DHANASREE L P dhanal1312'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

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

Below the code, the 'Results' tab is active, showing the output: 'Table altered.' and '0.06 seconds'. The bottom of the screen displays copyright information for Oracle and the APEX version.

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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and a green button for 'DL DHANASREE L P dhanal1312'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

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

Below the code, the 'Results' tab is active, showing the output: 'Table altered.' and '0.06 seconds'. The bottom of the screen displays copyright information for Oracle and the APEX version.

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

RESULT:

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 departments;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The right side shows a user profile 'DHANASREE L P' and a schema 'WKSP_DHANA1312'. The main area is titled 'SQL Commands' with tabs for 'Language', 'SQL', 'Rows', and 'Find Tables'. A command line at the bottom shows the query 'desc dept;'. Below the command line, the 'Describe' tab is selected, showing details for the 'DEPT' table. The table structure is displayed in a grid:

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

At the bottom of the page, there are footer links for 'Results', 'Explain', 'Saved SQL', and 'History'. The URL '220701043@rajalakshmi.edu.in' and session information 'bharath en' are also present.

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:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown is set to WKSP_DHANA1312. The main area displays the following SQL command:

```
1 select employee_id, last_name, job_id, hire_date from employees;
```

The results section shows a table with four columns: EMPLOYEE_ID, LAST_NAME, JOB_ID, and HIRE_DATE. The data returned is:

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
113	popp	ac_account	03/05/2024
114	raphealy	ac_account	02/03/1999

Below the table, it says "2 rows returned in 0.01 seconds" and there is a "Download" link. The bottom of the page includes copyright information and a link to Oracle APEX 23.2.4.

4. Provide an alias STARTDATE for the hire date.

QUERY:

select hire_date as "STARTDATE" from employees;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown is set to WKSP_DHANA1312. The main area displays the following SQL command:

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

The results section shows a table with one column named STARTDATE. The data returned is:

STARTDATE
03/05/2024
02/03/1999

Below the table, it says "2 rows returned in 0.01 seconds" and there is a "Download" link. The bottom of the page includes copyright information and a link to Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon (DL DHANASREE L P), and a schema dropdown set to WKSP_DHANA1512. The main workspace is titled "SQL Commands" and contains the following SQL code:

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

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

JOB_ID
ac_account

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

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon (DL DHANASREE L P), and a schema dropdown set to WKSP_DHANA1512. The main workspace is titled "SQL Commands" and contains the following SQL code:

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

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

EMPLOYEE_AND_TITLE
popp, ac_account
raphealy, ac_account

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

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

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

QUERY:

Select

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 APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is chosen from a dropdown menu. The main area contains the SQL command:

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

The results tab is selected, showing the output:

THE_OUTPUT
113,louis,popp,lpoppp,5151244567,03/05/2024,ac_account,6900,,205,100
114,den,rapheal,drapheal,5151274561,02/03/1999,ac_account,11000,,100,30

Below the table, it says "2 rows returned in 0.03 seconds" and there is a "Download" link. At the bottom of the page, there are footer links for "220701043@rajalakshmi.edu.in", "bharath", "en", "Copyright © 1999, 2023, Oracle and/or its affiliates.", and "Oracle APEX 23.2.4".

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

RESULT:

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 APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user icon for 'DHANASREE L P' and a schema dropdown set to 'WKSP_DHANA1312'. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. Below this is a toolbar with icons for Undo, Redo, Search, and Run. The command input field contains the SQL query: 'select last_name,salary from employees where salary>12000;'. The results tab is selected at the bottom, showing the message '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:

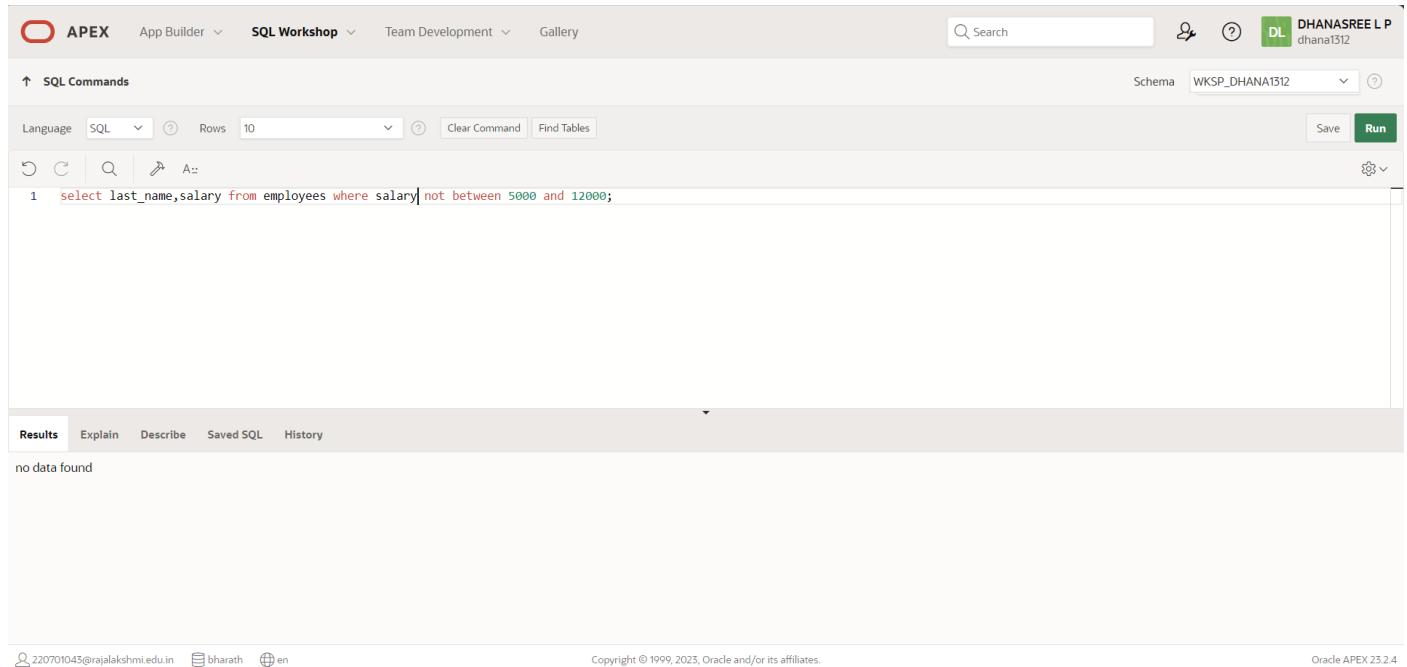
The screenshot shows the Oracle APEX SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar and schema selection are the same. The SQL command input field now contains: 'select last_name,department_id from employees where employee_id=176;'. The results tab is selected at the bottom, showing the message 'no data found'.

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

QUERY:

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

OUTPUT:



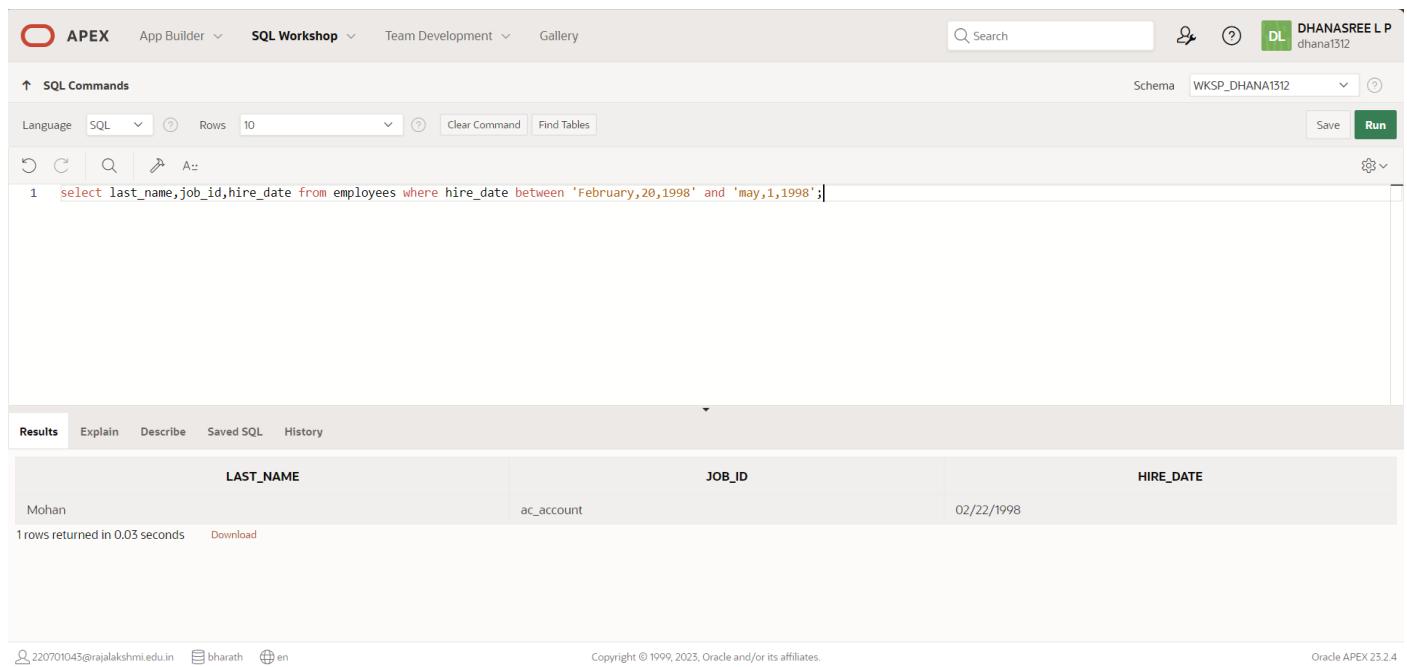
The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user icon for 'DHANASREE L P' and a schema dropdown set to 'WKSP_DHANA1312'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below it, a command line shows the query: '1 select last_name,salary from employees where salary not between 5000 and 12000;'. The 'Run' button is visible at the bottom right of the command line. Below the command line, tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History' are present, with 'Results' being the active tab. The results section displays 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:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user icon for 'DHANASREE L P' and a schema dropdown set to 'WKSP_DHANA1312'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below it, a command line shows the query: '1 select last_name,job_id,hire_date from employees where hire_date between 'February,20,1998' and 'May,1,1998';'. The 'Run' button is visible at the bottom right of the command line. Below the command line, tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History' are present, with 'Results' being the active tab. The results section displays a table with three columns: LAST_NAME, JOB_ID, and HIRE_DATE. One row is shown: Mohan, ac_account, 02/22/1998. At the bottom left, it says '1 rows returned in 0.03 seconds'. At the bottom right, it says 'Download'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'DHANASREE L P' (dhana1312). The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the query is displayed:

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

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

LAST_NAME	DEPARTMENT_ID
raphealy	20

Below the table, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes copyright information and a footer.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar and user profile are the same. The main area has tabs for 'SQL Commands' and 'Results'. Under 'SQL Commands', the query is displayed:

```
1 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;
```

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

EMPLOYEE	MONTHLY SALARY
raphealy	11000

Below the table, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes copyright information and a footer.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'DHANASREE L P dhana1312'. The main workspace has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the query: 'select last_name,hire_date from employees where hire_date like '1994';'. Below the query, the results tab is selected, showing the message 'no data found'. At the bottom, the footer includes user information (220701043@rajalakshmi.edu.in, bharath, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 23.2.4).

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar, session info, and footer are the same. The main workspace shows the query: 'select last_name,job_id from employees where manager_id is null;'. Below it, the results tab shows 'no data found'. The footer includes user information, copyright notice, and the version (Oracle APEX 23.2.4).

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'DHANASREE L P' (dhanal1312). The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. Below these are icons for Undo, Redo, Search, and Run. The SQL command entered is: `1 select last_name,salary,commission_pct from employees where commission_pct is not null order by salary desc;`. The results tab is selected, showing the message 'no data found'. At the bottom, the footer includes the user's email (220701043@rajalakshmi.edu.in), session information (bharath, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 23.2.4).

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, identical to the previous one but with a different query. The top navigation bar, user profile, and SQL command input area ('1 select last_name from employees where last_name like '__a%'') are the same. The results tab is selected, showing the message 'no data found'. The footer information at the bottom of the screen is also identical to the first screenshot.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'DHANASREE L P dhana1312'. The main workspace has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the query: 'select last_name from employees where last_name like '%a%' and last_name like '%e%';'. Below the query, the results section displays a single row: 'raphealy'. A note indicates '1 rows returned in 0.00 seconds'.

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

QUERY:

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

OUTPUT:

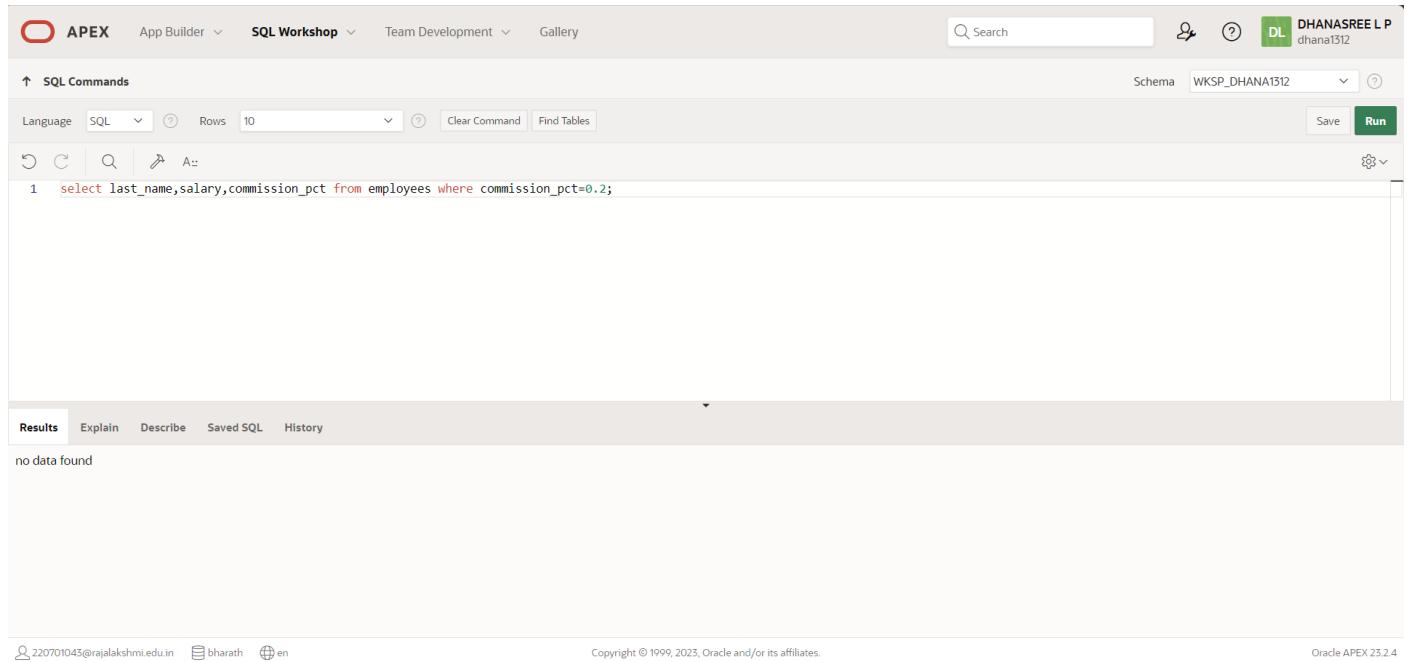
The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'DHANASREE L P dhana1312'. The main workspace has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab is active, showing the query: 'select last_name,job_id,salary from employees where job_id in ('sales representative','stock clerk') and salary not in(2500,3500,7000);'. Below the query, the results section displays the message 'no data found'. At the bottom, the footer includes copyright information for Oracle and the APEX version 'Oracle APEX 25.2.4'.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'DHNASREE LP dhana1312'. The main workspace is titled 'SQL Commands' and contains the following SQL statement:

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

Below the command, 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:

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 APEX SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The main area contains the SQL command: 'select sysdate from dual;'. Below the command, the results section displays the output: '03/10/2024'. The status bar at the bottom indicates '3 rows returned in 0.01 seconds'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The main area contains the SQL command: 'select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary" from employees;'. Below the command, the results section displays a table with four columns: EMPLOYEE_ID, LAST_NAME, SALARY, and new_salary. The data is as follows:

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

The status bar at the bottom indicates '3 rows returned in 0.01 seconds'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for DHANASREE L P (dhana1312). The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query: `select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary", new_salary-salary as "Increase" from employees;`. The Results tab displays the query results in a table:

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

Below the table, it says "3 rows returned in 0.01 seconds" and has a "Download" link. The bottom footer includes the URL 220701043@irajalakshmi.edu.in, session ID bharath, language en, copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for DHANASREE L P (dhana1312). The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the 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;`. The Results tab displays the query results in a table:

INITCAP(LAST_NAME)	Length_of_last_name
Mohan	5

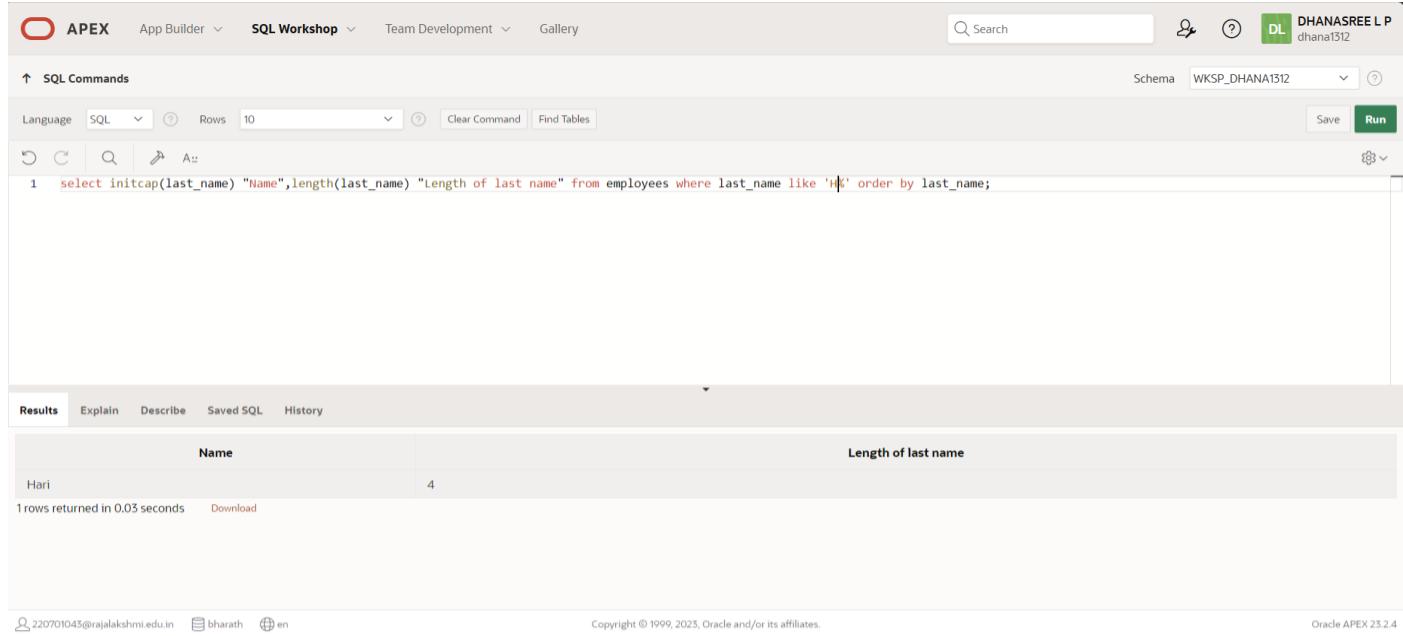
Below the table, it says "1 rows returned in 0.01 seconds" and has a "Download" link. The bottom footer includes the URL 220701043@irajalakshmi.edu.in, session ID bharath, language en, copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version Oracle APEX 23.2.4.

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

QUERY:

```
select initcap(last_name) "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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'DHANASREE L P dhana1312'. The main workspace shows the SQL command entered: 'select initcap(last_name) "Name",length(last_name) "Length of last name" from employees where last_name like 'H%' order by last_name;'. Below the command, the results section displays a single row: Hari, 4. The results table has two columns: 'Name' and 'Length of last name'. At the bottom left, it says '1 rows returned in 0.03 seconds' and 'Download'. The bottom of the page shows copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile, and schema information (WKSP_DHANA1312). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab shows a single query: "select last_name,round((sysdate-hire_date)/30,0) as \"MONTHS_WORKED\" from employees order by round((sysdate-hire_date)/30,0) asc". The Results tab displays the output in a table:

LAST_NAME	MONTHS_WORKED
popp	0
raphealy	306
Mohan	317

Below the table, it says "3 rows returned in 0.01 seconds" and has a "Download" link. The bottom of the screen shows copyright information and the Oracle APEX version (23.2.4).

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile, and schema information (WKSP_DHANA1312). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the query: "select last_name||' earns'||salary||' monthly but wants'||salary*3 as "DREAM_SALARIES" from employees;". The Results tab displays the output in a table:

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

Below the table, it says "3 rows returned in 0.00 seconds" and has a "Download" link. The bottom of the screen shows copyright information and the Oracle APEX version (23.2.4).

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile, and schema information (WKSP_DHANA1312). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains a code editor with the following query:

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

The Results tab displays the output:

LAST_NAME	SALARY
popp	\$\$\$\$\$\$\$\$\$\$6900
raphealy	\$\$\$\$\$\$\$\$\$\$1000
Mohan	\$\$\$\$\$\$\$\$\$\$4000

Below the table, it says "3 rows returned in 0.01 seconds" and has a "Download" link. At the bottom, there are copyright notices and a footer for Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile, and schema information (WKSP_DHANA1312). The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains a code editor with the following query:

```
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;
2
```

The Results tab displays 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

Below the table, it says "3 rows returned in 0.01 seconds" and has a "Download" link. At the bottom, there are copyright notices and a footer for Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'DHANASREE L P' (dhana1312), and buttons for Schema (WKSP_DHANA1312), Save, and Run.

In the main area, there's a toolbar with icons for Undo, Redo, Find, Replace, and others. Below the toolbar, the language is set to SQL, rows are set to 10, and there are buttons for Clear Command and Find Tables.

The SQL command entered is:

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

The results section shows the output of the query:

LAST_NAME	HIRE_DATE	DAY
Mohan	02/22/1998	Sunday
popp	03/05/2024	Tuesday
raphealy	02/03/1999	Wednesday

Below the table, it says "3 rows returned in 0.04 seconds" and has a "Download" link. At the bottom, it shows the URL 220701043@rajalakshmi.edu.in, the user bharath, and the session ID en. It also includes copyright information for Oracle and the APEX version 23.2.4.

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

RESULT:

DISPLAYING DATA FROM MULTIPLE TABLES

EX_NO:7

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for 'APEX', 'App Builder', 'SQL Workshop' (which is selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'DHANASREE L P' (dhana1312). Below the tabs, there's a search bar and some other navigation icons. The main area shows the SQL command entered:

```
1 select e.last_name,e.department_id,d.Dept_name from employees e,departments d ;
```

Below the command, the results are displayed in a grid format:

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

At the bottom of the results pane, it says '6 rows returned in 0.01 seconds'. The footer of the page includes copyright information for Oracle and the URL 'oracle.apex.oracle.com'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The schema dropdown is set to 'WKSP_DHANASREE1312'. The main area displays a SQL command window with the following content:

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

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

JOB_ID	LOCATION_ID
ac_account	4598

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

At the bottom of the page, there are footer links for 220701043@rajalakshmi.edu.in, bharath, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar and schema selection are identical. The SQL command window contains the same query as above:

```
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.department_id and d.location_id=l.location_id and e.commission_pct is not null;
```

The results tab shows the output:

JOB_ID	LOCATION_ID	CITY
no data found		

At the bottom of the page, there are footer links for 220701043@rajalakshmi.edu.in, bharath, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, a help icon, and a user profile for 'DHANASREE L P' with the identifier 'dhana1312'. Below the header, the 'SQL Commands' tab is selected. The SQL editor contains the following query:

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

The results section shows the output of the query:

LAST_NAME	DEPT_NAME
Janu	Public Relations

Below the table, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for 220701043@rajalakshmi.edu.in, bharath, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The 'SQL Commands' tab is selected, and the SQL editor contains the same query as above:

```
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’;
```

The results section shows 'no data found'.

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

QUERY:

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

OUTPUT:

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. On the right, a user profile for 'DHANASREE L P' (dhana1312) is shown with a 'Run' button. The main area displays a SQL command window with 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 command window, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing the message "no data found". At the bottom of the page, the footer includes the URL 220701045@rajalakshmi.edu.in, the name bharath, and the language en. It also states Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar and user profile are identical. The SQL command window contains the same query as the previous screenshot:

```
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);
```

The Results tab is selected, displaying the following data in a table:

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

Below the table, it says "3 rows returned in 0.01 seconds". At the bottom of the page, the footer includes the URL 220701045@rajalakshmi.edu.in, the name bharath, and the language en. It also states Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The schema is set to 'WKSP_DHANA1312'. The query entered is:

```
1 select e.department_id departments,e.last_name employees,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;
```

The results tab shows 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 employees e JOIN departments d
ON (e.department_id = d.dept_id)
JOIN job_grade j
ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The schema is set to 'WKSP_DHANA1312'. The query entered is:

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

The results tab displays 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

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

QUERY:

```
SELECT e.last_name, e.hire_date
FROM employees e, employees davies
WHERE davies.last_name = 'Davies'
AND davies.hire_date < e.hire_date;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon, and a schema dropdown set to 'WKSP_DHANA1512'. The main area is titled 'SQL Commands' with a 'Run' button. Below it, a code editor shows the following SQL query:

```
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 editor, a results table is displayed with two columns: 'LAST_NAME' and 'HIRE_DATE'. The single row returned is 'Janu' with a hire date of '03/05/2024'. The bottom of the page includes copyright information and a footer with user details.

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

QUERY:

```
SELECT e.last_name AS Employee, e.hire_date AS Emp_Hired,
e.manager_name AS Manager, m.hire_date AS Mgr_Hired
FROM employees e
JOIN employees|m ON e.manager_name = m.last_name
WHERE e.hire_date < m.hire_date;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon, and a schema dropdown set to 'WKSP_DHANA1512'. The main area is titled 'SQL Commands' with a 'Run' button. Below it, a code editor shows the following SQL query:

```
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;
```

Below the code editor, a results table is displayed with two columns: 'EMPLOYEE' and 'MGR_HIRED'. The output table is empty, indicating 'no data found'. The bottom of the page includes copyright information and a footer with user details.

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

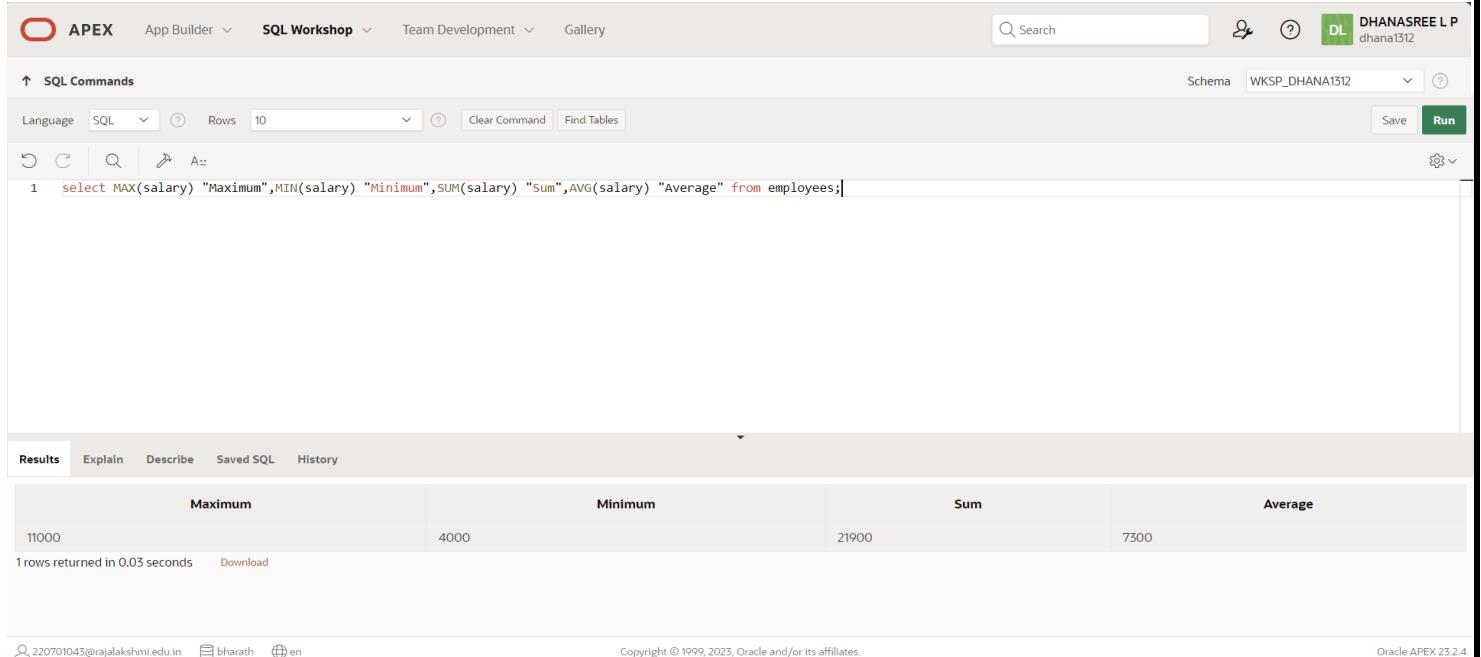
FALSE

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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and a green button labeled 'DL DHANASREE L P dhana1312'. The main workspace has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following query:

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

The Results tab displays the output of the query:

Maximum	Minimum	Sum	Average
11000	4000	21900	7300

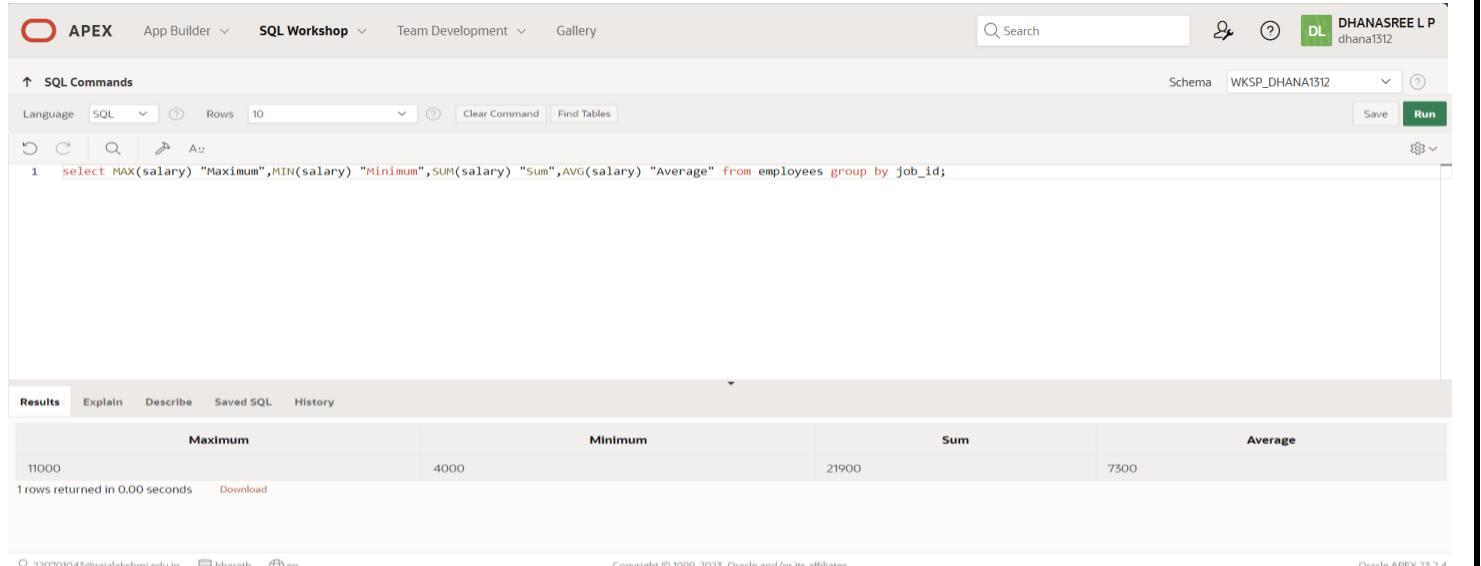
Below the table, it says "1 rows returned in 0.03 seconds" and "Download". The bottom of the page includes copyright information and a footer with user details.

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 APEX SQL Workshop interface, similar to the previous one. The top navigation bar and workspace are identical. The SQL Commands tab contains the modified query:

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

The Results tab displays the output of the query:

Maximum	Minimum	Sum	Average
11000	4000	21900	7300

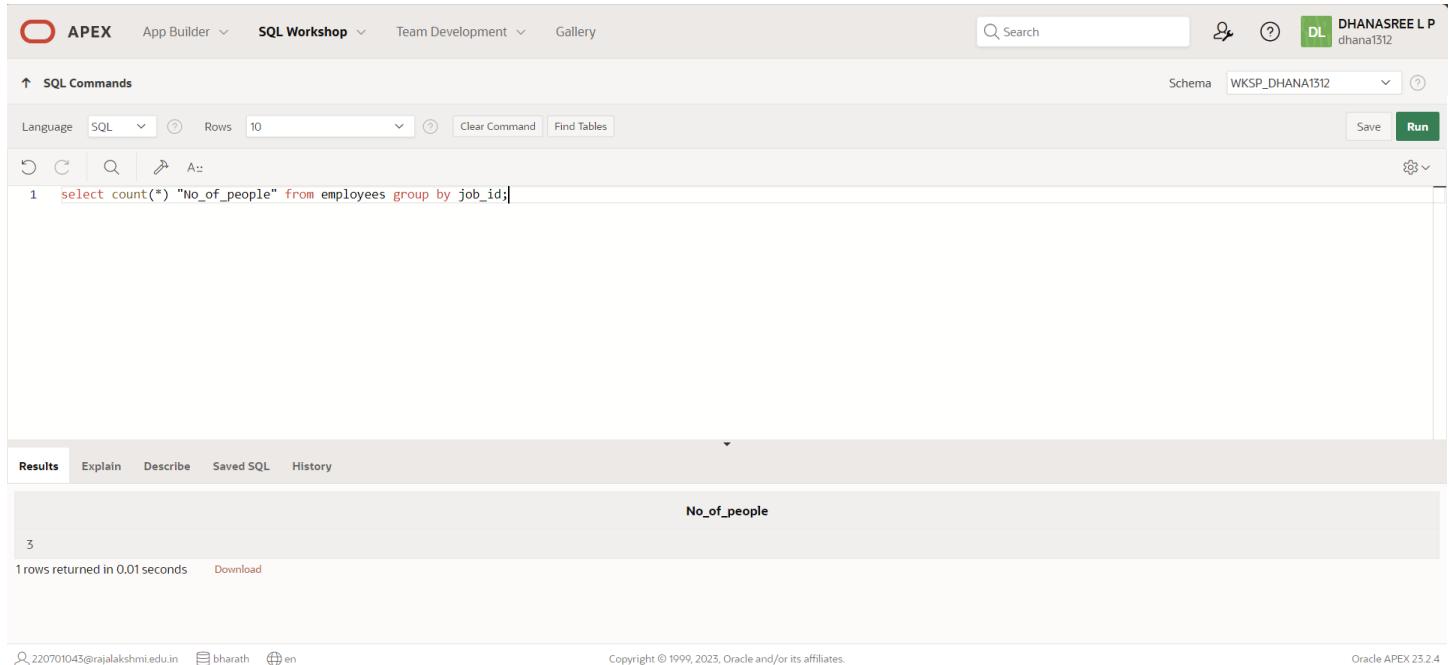
Below the table, it says "1 rows returned in 0.00 seconds" and "Download". The bottom of the page includes copyright information and a footer with user details.

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 APEX interface with the SQL Workshop tab selected. The schema is set to WKSP_DHANA1512. The SQL command window contains two queries:

```
1 select count(*) "No_of_people" from employees group by job_id;  
2 select count(*) "No_of_people" from employees where job_id='47' group by job_id;
```

The results pane shows the output for the first query:

No_of_people
3

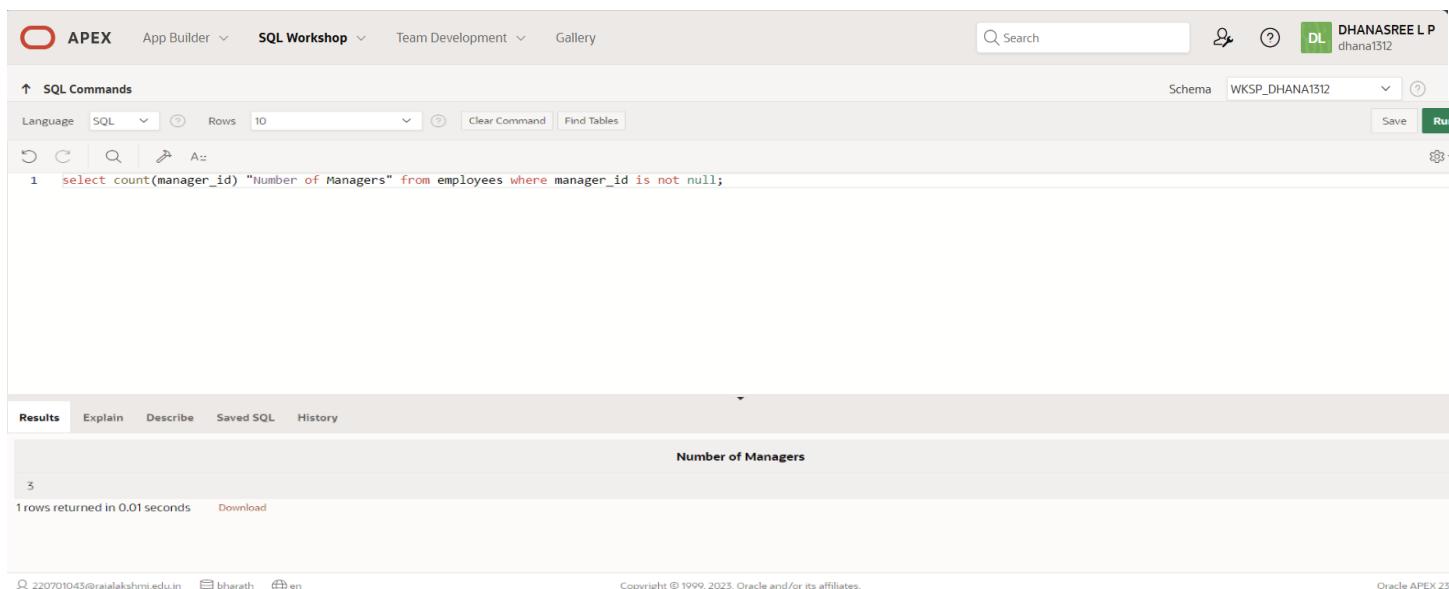
1 rows returned in 0.01 seconds [Download](#)

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:



The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. The schema is set to WKSP_DHANA1512. The SQL command window contains one query:

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

The results pane shows 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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'DHNASREE L P' (dhanalakshmi1312). The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query: 'select MAX(salary)-MIN(salary) "DIFFERENCE" from employees;'. The Results tab displays the output: 'DIFFERENCE' with a value of 7000. Below the results, it says '1 rows returned in 0.01 seconds' and provides a 'Download' link. The bottom footer includes copyright information for Oracle and the APEX version.

MANAGER_ID	LOWEST_SALARY
100	11000
205	6900

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 APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'DHNASREE L P' (dhanalakshmi1312). The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query: '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 tab displays the output: 'MANAGER_ID' and 'LOWEST_SALARY'. It shows two rows: one for manager_id 100 with a salary of 11000, and one for manager_id 205 with a salary of 6900. Below the results, it says '2 rows returned in 0.01 seconds' and provides a 'Download' link. The bottom footer includes copyright information for Oracle and the APEX version.

MANAGER_ID	LOWEST_SALARY
100	11000
205	6900

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon for 'DHANASREE L P dhana1312', and a 'Run' button. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. The code entered is:

```
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",sum(decode(to_char(hire_date,'yyyy'),1998,1,0)) "1998" from employees;
```

The results section shows a single row with the following data:

TOTAL	1995	1996	1997	1998
3	0	0	0	1

Below the results, it says '1 rows returned in 0.04 seconds' and has a 'Download' link. At the bottom, it shows the user information '220701043@rajalakshmi.edu.in bharath en' and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon for 'DHANASREE L P dhana1312', and a 'Run' button. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. The code entered is:

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

The results section shows a single row with the following data:

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

Below the results, it says '3 rows returned in 0.03 seconds' and has a 'Download' link. At the bottom, it shows the user information '220701043@rajalakshmi.edu.in bharath en' and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows a user profile for 'DHANASREE L P' (dhana1312). The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to 'WKSP_DHANA1312'. Below the command input field, there are buttons for Save and Run. The SQL command entered is:

```
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;
```

The results section displays the output of the query:

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

Below the table, it says '4 rows returned in 0.02 seconds' and has a 'Download' link. At the bottom, it shows the URL '220701045@rajalakshmi.edu.in', the user 'bharath', and the language 'en'. Copyright information for Oracle APEX 25.2.4 is also present.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	

Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

SUB QUERIES

EX_NO:9

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'DHANASREE L P dhanal1512', and a 'Run' button. 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	03/05/1997

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'DHANASREE L P dhanal1512', and a 'Run' button. The main area is titled 'SQL Commands' with a language dropdown set to 'SQL'. The query entered is:

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

The results section shows two rows:

EMPLOYEE_ID	LAST_NAME	SALARY
142	Doe	30000
1001	Smith	70000

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user information (DL DHANASREE L P dhana1312), and a schema dropdown set to WKSP_DHANA1312. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains a code editor with the following query:

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

The Results tab displays the output of the query:

EMPLOYEE_ID	LAST_NAME
113	Janu
142	Doe

Below the table, it says "2 rows returned in 0.03 seconds" and has 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 department_id
from departments where location_id=1700);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user information (DL DHANASREE L P dhana1312), and a schema dropdown set to WKSP_DHANA1312. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains 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 displays the output of the query:

LAST_NAME	DEPARTMENT_ID	JOB_ID
Janu	100	ac_account
Doe	100	ac_account

Below the table, it says "2 rows returned in 0.04 seconds" and has a "Download" link.

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

QUERY:

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

OUTPUT:

A screenshot of the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. On the right, a user profile for 'DHANASREE L P' (dhana1312) is shown. The main area shows a SQL command window with the following code:

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

The results section displays the output of the query:

LAST_NAME	SALARY
Davies	11000
Mohan	4000

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

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

QUERY:

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

OUTPUT:

A screenshot of the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. On the right, a user profile for 'DHANASREE L P' (dhana1312) is shown. The main area shows a SQL command window with the following code:

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

The results section displays the output of the query:

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

Below the table, it says "3 rows returned in 0.01 seconds" and has a "Download" link.

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 APEX 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 'DHANASREE L P' are also at the top. The main area is titled 'SQL Commands' with a 'Run' button. The SQL code entered is:

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

The results section shows a table with three columns: EMPLOYEE_ID, LAST_NAME, and SALARY. The data is:

EMPLOYEE_ID	LAST_NAME	SALARY
1001	Smith	70000
142	Doeu	30000

Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page includes copyright information and the text 'Oracle APEX 23.2.4'. To the right of the results table is a grid-based evaluation form:

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

RESULT:

USING THE SET OPERATORS

EX_NO:10

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.

DATE:

QUERY:

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

OUTPUT:

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

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

The results section displays the output:

DEPARTMENT_ID
20
100

2 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:

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

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

The results section displays the output:

COUNTRY_ID	STATE_PROVINCE
58	toronto

1 rows returned in 0.02 seconds

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

QUERY:

```
select job_id,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:

SQL Commands

```
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;
```

Results

JOB_ID	DEPARTMENT_ID
ac_account	20
hr_rep	20

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

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:

SQL Commands

```
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;
```

Results

JOB_ID	EMPLOYEE_ID
ac_account	113
ac_account	142
sales_rep	1001

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

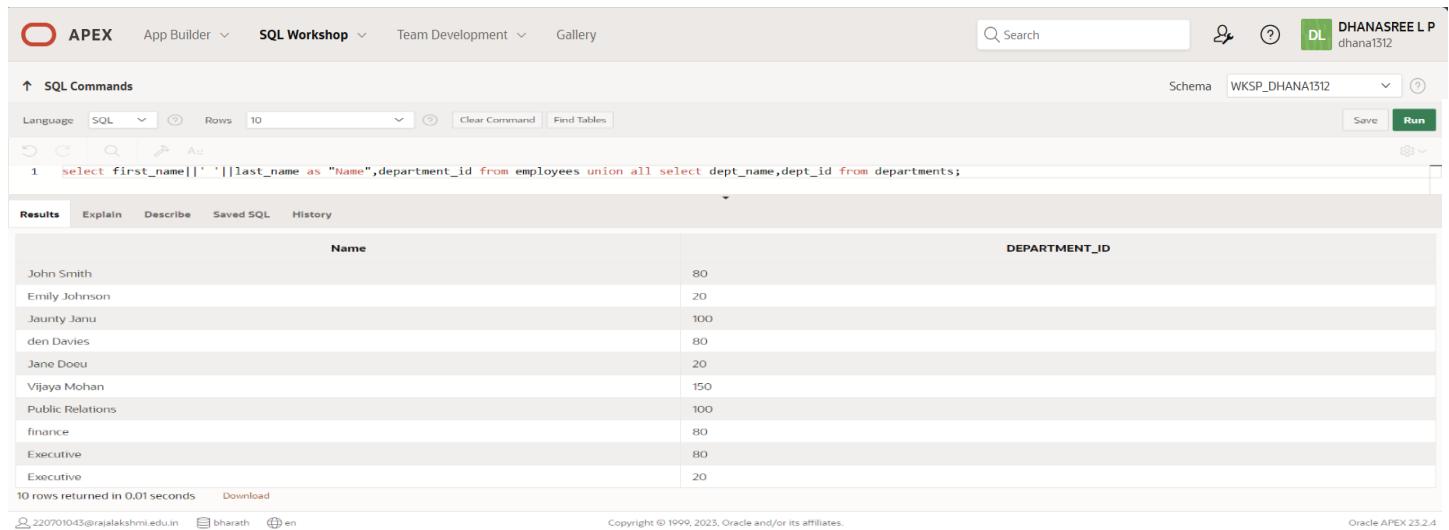
5.)The HR department needs a report with the following specifications: - Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department. -

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,department_id from departments;
```

OUTPUT:



The screenshot shows the Oracle APEX interface with the SQL Workshop module selected. A compound query is entered in the command line:

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

The results are displayed in a grid format:

Name	DEPARTMENT_ID
John Smith	80
Emily Johnson	20
Janity 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

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

RESULT:

CREATING VIEWS

EX_NO:11

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, 'SQL Workshop' is chosen from a dropdown menu. The main area is titled 'SQL Commands'. A code editor window contains the following SQL code:

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

Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active. The output section displays the message 'View created.' and '0.07 seconds'. At the bottom of the page, there are footer links for user information and copyright notice.

2.) Display the contents of the EMPLOYEES_VU view.

QUERY:

```
select * from employees_vu;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon for 'DHANASREE L P dhanal1312', and a 'Run' button. Below the tabs, the 'SQL Commands' section shows the query: 'SELECT * FROM employees_vu;'. The results section displays a table with columns 'EMPLOYEE_ID', 'EMPLOYEE', and 'DEPARTMENT_ID'. The data includes rows for Smith (80), Johnson (20), Janu (100), Davies (80), Doe (20), and Mohan (150). The bottom of the screen shows copyright information and the version 'Oracle APEX 23.2.4'.

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
1001	Smith	80
125	Johnson	20
113	Janu	100
114	Davies	80
142	Doe	20
115	Mohan	150

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The 'SQL Workshop' tab is selected. In the 'SQL Commands' section, the query 'SELECT view_name, text FROM user_views;' is entered. The results section shows a table with 'VIEW_NAME' and 'TEXT' columns. The single row has 'EMPLOYEES_VU' in the 'VIEW_NAME' column and the query 'SELECT employee_id, last_name employee, department_id FROM employees' in the 'TEXT' column. The bottom of the screen shows copyright information and the version 'Oracle APEX 23.2.4'.

VIEW_NAME	TEXT
EMPLOYEES_VU	SELECT employee_id, last_name employee, 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:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_DHANA1312

Language SQL Rows 10 Clear Command Find Tables Save Run

Results Explain Describe Saved SQL History

EMPLOYEE	DEPARTMENT_ID
Smith	80
Johnson	20
Janu	100
Davies	80
Doeu	20
Mohan	150

6 rows returned in 0.01 seconds Download

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

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_DHANA1312

Language SQL Rows 10 Clear Command Find Tables Save Run

Results Explain Describe Saved SQL History

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

View created.

0.06 seconds

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

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

QUERY:

Describe dept50;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a user icon for 'DHANASREE L P' are also present. The main workspace shows the SQL Commands tab selected, with the schema set to 'WKSP_DHANA1512'. The code editor contains the command: '1 DESCRIBE dept50 ;'. Below the code editor is a results grid titled 'DEPT50' showing three columns: EMPNO, EMPLOYEE, and DEPTNO, with their respective data types and constraints.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT50	EMPNO	NUMBER	-	6	0	-	-	-	-
	EMPLOYEE	VARCHAR2	20	-	-	-	-	-	-
	DEPTNO	NUMBER	-	4	0	-	✓	-	-

7.) Attempt to reassign Matos to department 80

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a user icon for 'DHANASREE L P' are also present. The main workspace shows the SQL Commands tab selected, with the schema set to 'WKSP_DHANA1512'. The code editor contains the UPDATE command: '1 UPDATE dept50 SET deptno=80 WHERE employee='Matos';'. The results section shows the output: '0 row(s) updated.' and '0.05 seconds'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile 'DHANASREE L P dhana1312' are also present. 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.dept_name "Department",e.salary "Salary",j.grade_level "Grades"
3 from employees e,departments d,job_grade j
4 where e.department_id=d.dept_id and e.salary between j.lowest_sal and j.highest_sal;
```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The 'Results' tab is selected, displaying the message 'View created.' and a execution time of '0.06 seconds'. At the bottom of the page, footer information includes the URL '220701043@rajalakshmi.edu.in', the name 'bharath', the language 'en', and the copyright notice 'Copyright © 1999, 2025, Oracle and/or its affiliates.' The page is identified as 'Oracle APEX 23.2.4'.

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

RESULT:

EXERCISE 12

PRACTICE QUESTIONS

Intro to Constraints; NOT NULL and UNIQUE Constraints

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

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

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

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

2. What are the limitations of constraints that may be applied at the column level and at the table level?
 - Constraints referring to more than one column are defined at Table Level
 - NOT NULL constraint must be defined at column level as per ANSI/ISO SQL standard.
3. Why is it important to give meaningful names to constraints?
 - If a constraint is violated in a SQL statement execution, it is easy to identify the cause with user-named constraints.
 - It is easy to alter names/drop constraint..

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

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

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

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

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

```

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

```

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

Table Created.

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

DESCRIBE f_global_locations;

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

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

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

PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

1. What is the purpose of a
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK CONSTRAINT

a. PRIMARY KEY

Uniquely identify each row in table.

b. FOREIGN KEY

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

c. CHECK CONSTRAINT

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

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

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

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

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

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

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

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

```
SELECT * FROM animals;
```

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

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

a. ON DELETE CASCADE

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

b. ON DELETE SET NULL

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

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

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

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

PRACTICE PROBLEM

Managing Constraints

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

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

1. What are four functions that an ALTER statement can perform on constraints?

- ADD
- DROP
- ENABLE
- DISABLE

2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy_d_clients table. Name the primary key copy_d_clients_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy_d_clients.table?

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

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

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

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

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

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

COPY_D_CLT_CLIENT_NUMBER_PK

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

```
ALTER TABLE copy_d_clients
```

```
DROP CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK CASCADE ;
```

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

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

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

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

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

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

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

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

1 row(s) inserted.

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

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

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

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

1 row(s) deleted.

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

Table altered.

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

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

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

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

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

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

RESULT:

EXERCISE 13

Creating Views

1. What are three uses for a view from a DBA's perspective?
 - **Restrict access and display selective columns**
 - **Reduce complexity of queries from other internal systems. So, providing a way to view same data in a different manner.**
 - **Let the app code rely on views and allow the internal implementation of tables to be modified later.**
2. Create a simple view called view_d_songs that contains the ID, title and artist from the DJs on Demand table for each "New Age" type code. In the subquery, use the alias "Song Title" for the title column.
3. SELECT * FROM view_d_songs. What was returned?

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

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

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

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

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

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

6. It is company policy that only upper-level management be allowed access to individual employee salaries. The department managers, however, need to know the minimum, maximum, and average salaries, grouped by department. Use the Oracle database to prepare a view that displays the needed information for department managers.

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

DML Operations and Views

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

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

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

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

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

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

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

```
SELECT * FROM view_copy_d_songs;
```

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

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

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

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

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

```
SELECT * FROM read_copy_d_cds;
```

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

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

6. Use REPLACE to modify read_copy_d_cds. Replace the READ ONLY option with WITH CHECK

OPTION CONSTRAINT ck_read_copy_d_cds. Execute a SELECT * statement to verify that the view exists.

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

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

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

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

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

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

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

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

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

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

DELETE,INSERT,MODIFY restricted if it contains:

Group functions
GROUP BY CLAUSE
DISTINCT
pseudocolumn ROWNUM Keyword

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

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

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

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

Managing Views

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

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

```
SELECT * FROM view_copy_d_songs;
```

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

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

ORA-00942: table or view does not exist

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

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

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

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

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

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

Indexes and Synonyms

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

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

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

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

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

3. When will an index be created automatically?

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

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

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

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

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

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

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

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

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

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

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

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

```
CREATE SYNONYM dj_tracks2 FOR d_track_listings;
```

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

10. Drop the synonym that you created in question

```
DROP SYNONYM dj_tracks2;
```

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

RESULT:

OTHER DATABASE OBJECTS

EX_NO:14

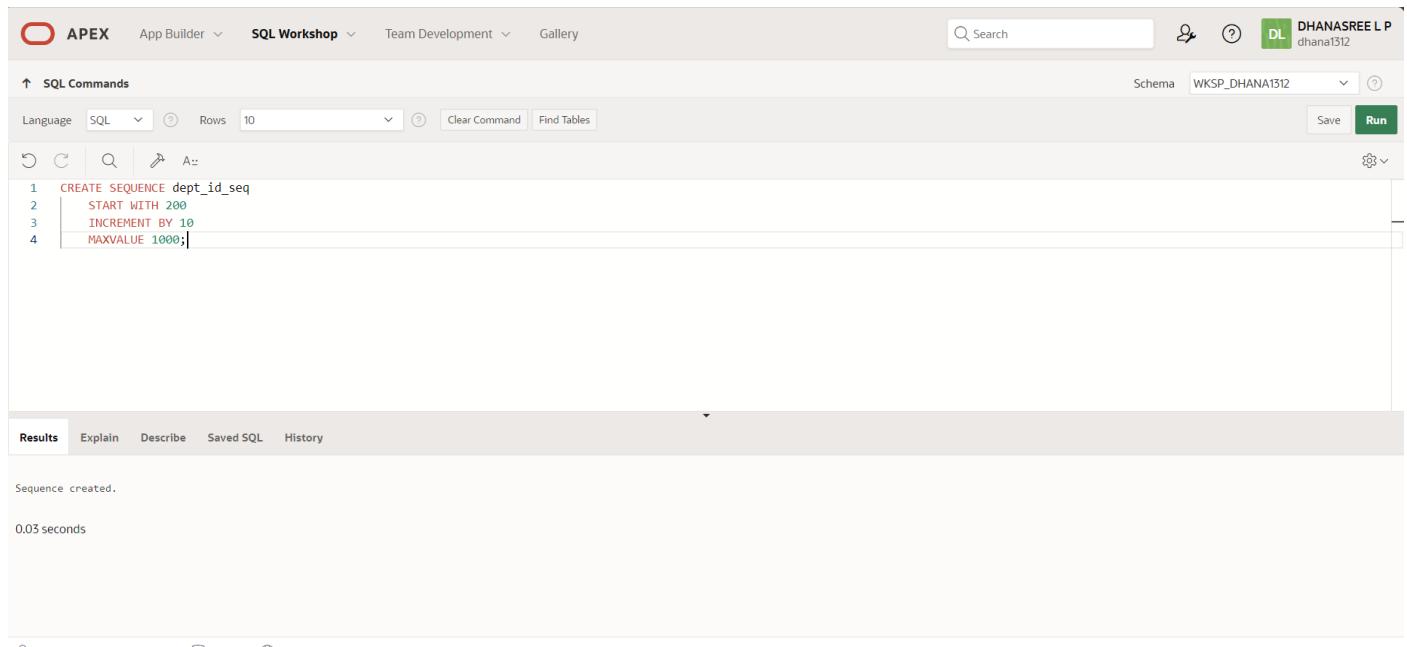
DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop' (which is highlighted in green), 'Team Development', and 'Gallery'. On the right side, the schema 'WKSP_DHANA1312' and user 'DHANASREE L P' are displayed. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. The code entered is:

```
1 CREATE SEQUENCE dept_id_seq
2   START WITH 200
3   INCREMENT BY 10
4   MAXVALUE 1000;|
```

Below the code, the results show:

Sequence created.
0.05 seconds

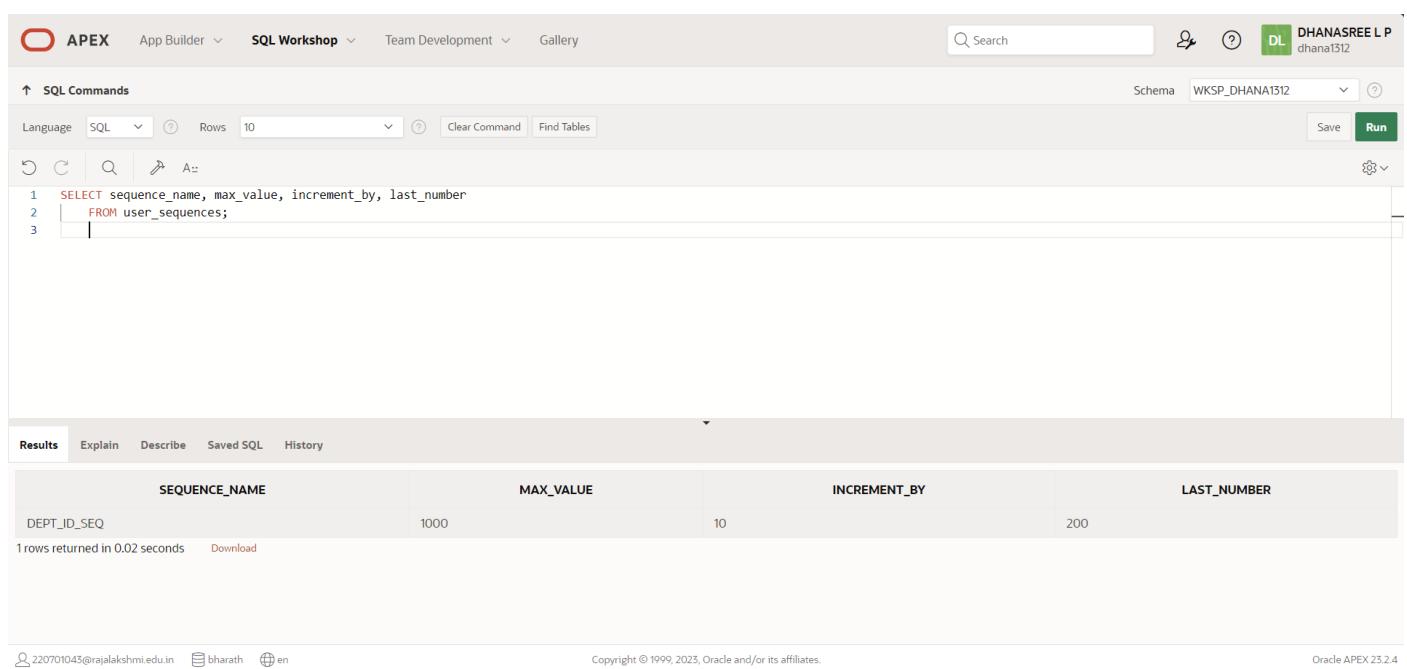
At the bottom, the footer includes the URL '220701043@rajalakshmi.edu.in', the name 'bharath', and the language 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

SELECT sequence_name, max_value, increment_by, last_number FROM user_sequences;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar and schema selection are identical to the previous screenshot. The main area is titled 'SQL Commands' with a sub-section 'Language: SQL'. The code entered is:

```
1 SELECT sequence_name, max_value, increment_by, last_number
2   FROM user_sequences;
3 |
```

Below the code, the results table shows:

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_ID_SEQ	1000	10	200

At the bottom, the footer includes the URL '220701043@rajalakshmi.edu.in', the name 'bharath', and the language 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name 'DHANASREE L P' and session ID 'dhana1312'. The main area is titled 'SQL Commands' and shows a summary of the executed script 'exercise14'. The summary table has columns for Number, Elapsed, Statement, Feedback, and Rows. One row is listed: '1' took '0.03' seconds and executed the statement 'INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education')'. The feedback indicates '1 row(s) inserted.' and the row count is '1'. Below the summary, there are three status indicators: 'Statements Processed' (1), 'Successful' (1), and 'With Errors' (0). At the bottom of the page, the URL '220701043@rajalakshmi.edu.in', the user 'bharath', and the language 'en' are shown, along with copyright information and the version 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name 'DHANASREE L P' and session ID 'dhana1312'. The main area is titled 'SQL Commands' and shows a SQL command being run. The command is 'CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);'. The results tab at the bottom shows the message 'Index created.' and a time of '0.03 seconds'. At the bottom of the page, the URL '220701043@rajalakshmi.edu.in', the user 'bharath', and the language 'en' are shown, along with copyright information and the version 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile, and a green button labeled 'DHANASREE L P' with the ID 'dhana1312'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), and buttons for Clear Command and Find Tables. Below this is a code editor window containing the following SQL query:

```
1 SELECT index_name, table_name, uniqueness
2   FROM user_indexes
3  WHERE table_name = 'EMPLOYEES';
4
```

Below the code editor is a results grid with three columns: INDEX_NAME, TABLE_NAME, and UNIQUENESS. The single row returned is:

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

At the bottom of the results grid, it says '1 rows returned in 0.03 seconds' and has a 'Download' link. The footer of the page includes the URL '220701043@rajalakshmi.edu.in', the name 'bharath', and the language 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

RESULT:

CONTROLLING USER ACCESS

EX_NO:15

DATE:

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

The CREATE SESSION system privilege

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

The CREATE TABLE privilege

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

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

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

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

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

The ALTER USER statement

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

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

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

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

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

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

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

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

SELECT table_name FROM user_tables;

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

Team 1 revokes the privilege.

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

Team 2 revokes the privilege.

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

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

Team 1 executes this INSERT statement.

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

Team 2 executes this INSERT statement.

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

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

RESULT:

PL/SQL

CONTROL STRUCTURES

EX_NO:

DATE:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a user profile for 'DHANASREE L P dhanat1312'. The main area is titled 'SQL Commands' with tabs for 'Language' (set to 'SQL'), 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. Below these are standard toolbar icons for copy, paste, search, and run. The code editor contains the provided PL/SQL block. The results panel at the bottom shows the output: 'Incentive = 8400', 'Statement processed.', and a execution time of '0.00 seconds'. The bottom footer includes copyright information for Oracle and the date '19 May 2024'.

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

Incentive = 8400
Statement processed.
0.00 seconds

19 May 2024 Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user profile for 'DHANASREE L P' (dhanal1312). The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. The code editor contains the following PL/SQL block:

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

Below the code, the 'Results' tab is selected. A yellow box highlights the error message from the database:

```
Error at line 4/25: ORA-06550: line 4, column 25:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.IMG$DBMS_SQL_APEK_230200", line 801
ORA-06550: line 4, column 3:
PL/SQL: Statement ignored
```

Following the error message, the rest of the code is shown:

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

This screenshot shows the same Oracle APEX SQL Workshop interface. The code editor contains the same PL/SQL block as the previous screenshot:

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

The 'Results' tab is selected, and a yellow box highlights the error message:

```
Error at line 4/25: ORA-06550: line 4, column 25:
PLS-00201: identifier 'Welcome' must be declared
ORA-06512: at "SYS.IMG$DBMS_SQL_APEK_230200", line 801
ORA-06550: line 4, column 3:
PL/SQL: Statement ignored
```

Following the error message, the rest of the code is shown:

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

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

QUERY:

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

BEGIN

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and a schema dropdown for 'WKSP_DHANA1312' are also present. The main workspace displays the PL/SQL code with line numbers 1 through 17. Lines 13 through 17 show the execution of the code, including the SELECT statement and the output of the DBMS_OUTPUT.PUT_LINE calls. The results tab at the bottom shows the output: 'Before invoking procedure, salary_of_emp: 6900' and 'After invoking procedure, salary_of_emp: 7900'. The status bar at the bottom indicates 'Statement processed.' and a execution time of '0.00 seconds'.

```
1  DECLARE
2      salary_of_emp NUMBER(8,2);
3
4      PROCEDURE approx_salary (
5          emp      NUMBER,
6          empsal IN OUT NUMBER,
7          addless  NUMBER
8      ) IS
9
10     BEGIN
11         empsal := empsal + addless;
12     END;
13
14     BEGIN
15         SELECT salary INTO salary_of_emp
16         FROM employees
17         WHERE employee_id = 122;
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
207
208
209
209
210
211
212
213
214
215
216
217
217
218
219
219
220
221
222
223
224
225
226
227
227
228
229
229
230
231
232
233
234
235
235
236
237
237
238
239
239
240
241
241
242
243
243
244
245
245
246
247
247
248
248
249
249
250
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
135
```

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. The code entered is the PL/SQL block provided above. The 'Run' button is visible in the top right corner of the editor area. Below the editor, the 'Results' tab is selected, showing the output: 'Procedure created.' and '0.03 seconds'. At the bottom, there are footer links for copyright information and the version 'Oracle APEX 23.2.4'.

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

Results Explain Describe Saved SQL History

Procedure created.
0.03 seconds

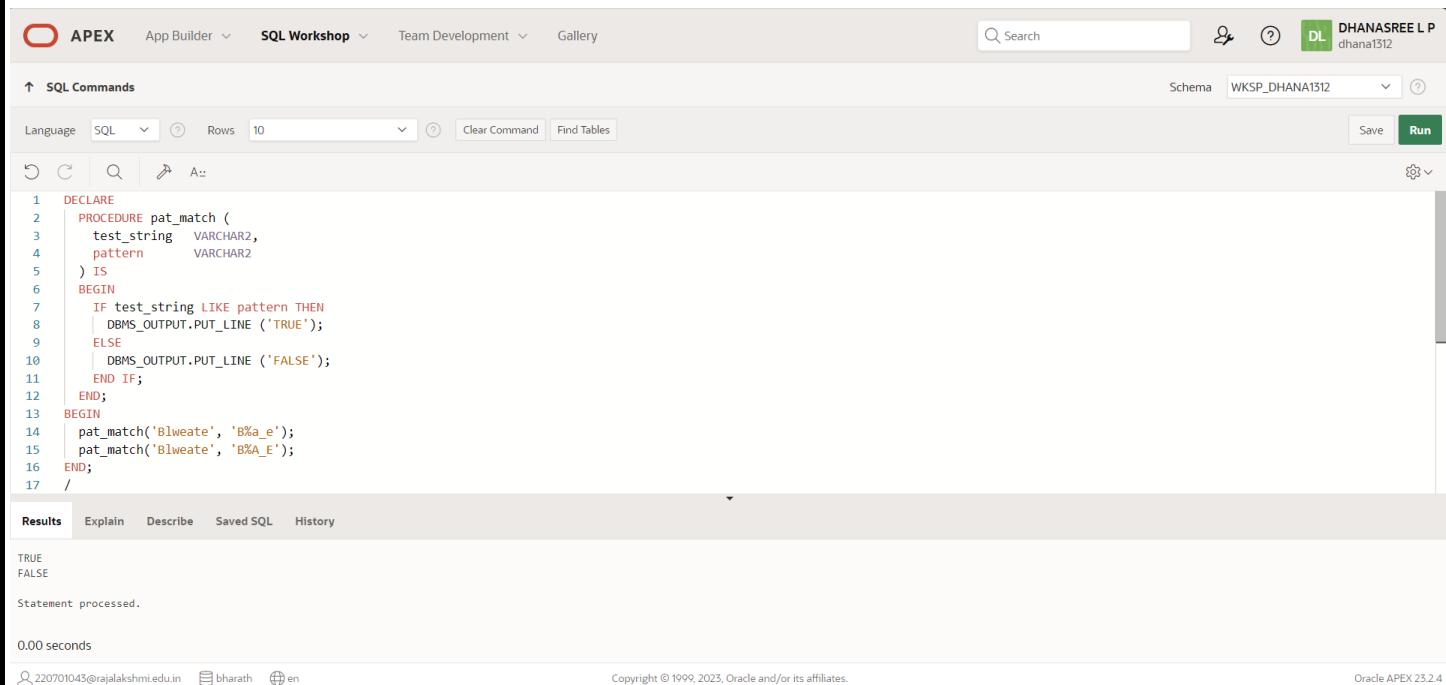
Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there is a search bar, a user icon, and a session identifier 'DHNASREE LP dhana1312'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code provided above. Below the code, the 'Results' tab is selected, showing the output of the execution. The output consists of two lines: 'TRUE' and 'FALSE', indicating the results of the LIKE comparisons. At the bottom, it says 'Statement processed.' and '0.00 seconds'.

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

Results Explain Describe Saved SQL History

TRUE
FALSE
Statement processed.
0.00 seconds

220701043@rajalakshmi.edu.in bharath en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

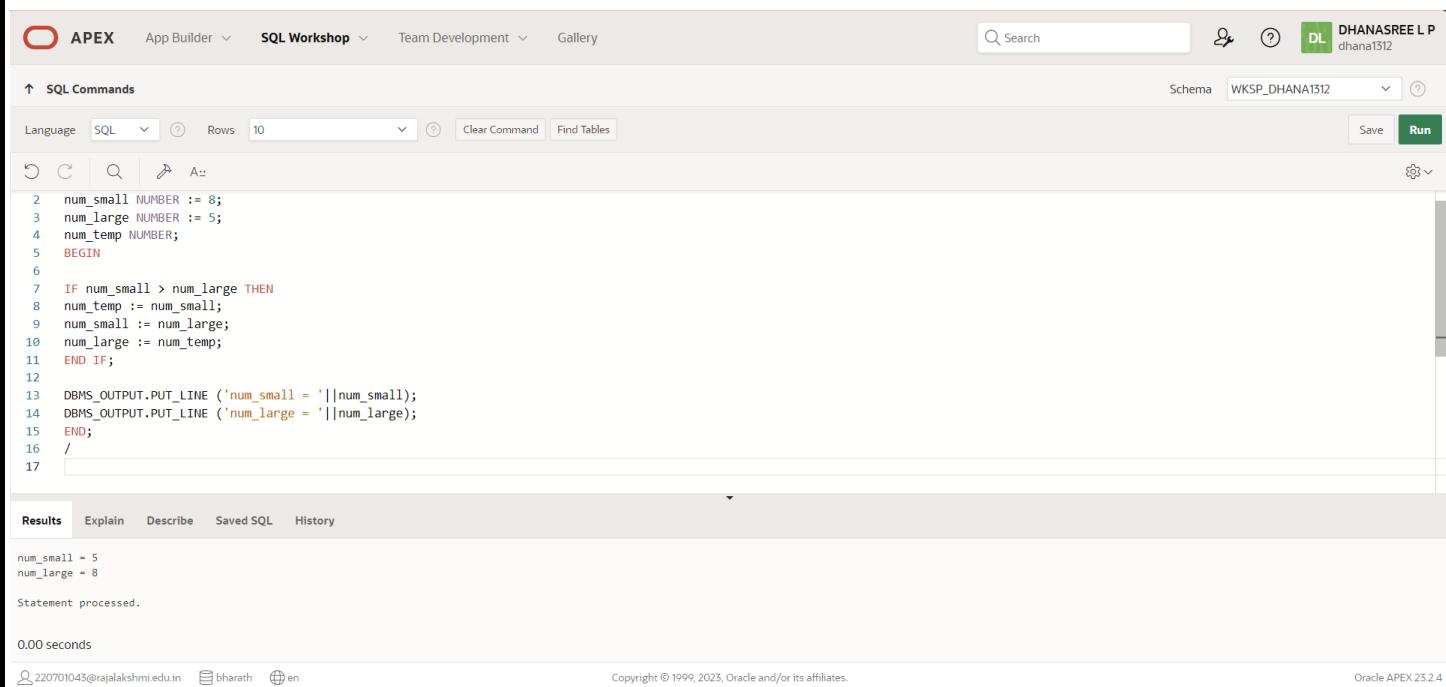
QUERY:

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

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

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, along with 'SQL Workshop'. The schema is set to 'WKSP_DHANA1312'. The main area displays the PL/SQL code. The code starts with declarations for three variables: num_small, num_large, and num_temp. It then checks if num_small is greater than num_large. If true, it swaps their values. Finally, it uses DBMS_OUTPUT.PUT_LINE to print the current values of num_small and num_large. The code ends with an END; statement and a slash (/) indicating a block terminator. Below the code, the 'Results' tab is active, showing the output of the executed code. The output indicates that num_small is now 5 and num_large is now 8, with a note that the statement was processed in 0.00 seconds.

```
2 num_small NUMBER := 8;
3 num_large NUMBER := 5;
4 num_temp NUMBER;
5 BEGIN
6
7 IF num_small > num_large THEN
8 num_temp := num_small;
9 num_small := num_large;
10 num_large := num_temp;
11 END IF;
12
13 DBMS_OUTPUT.PUT_LINE ('num_small = |||num_small);
14 DBMS_OUTPUT.PUT_LINE ('num_large = |||num_large);
15 END;
16 /
17
```

Results Explain Describe Saved SQL History

```
num_small = 5
num_large = 8

Statement processed.

0.00 seconds
```

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

QUERY:

DECLARE

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

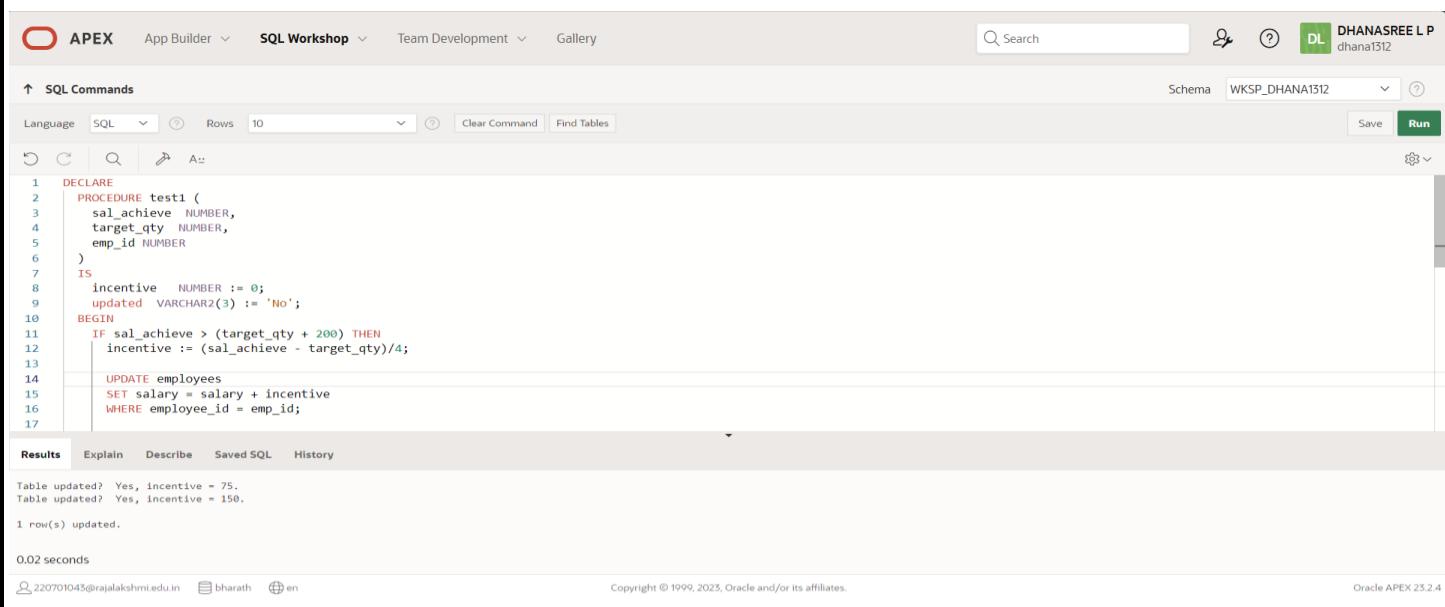
BEGIN

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

END;

/

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', 'Gallery', 'Search', and a user icon. The schema is set to 'WKSP_DHANA1312'. The main area displays the PL/SQL code for the 'test1' procedure. The code uses the DBMS_OUTPUT.PUT_LINE function to print the status of the update and the calculated incentive. The output pane shows the results of running the code twice with different parameters. The bottom status bar indicates the execution took 0.02 seconds.

```
1  DECLARE
2      PROCEDURE test1 (
3          sal_achieve NUMBER,
4          target_qty NUMBER,
5          emp_id NUMBER
6      )
7  IS
8      incentive NUMBER := 0;
9      updated VARCHAR2(3) := 'No';
10 BEGIN
11     IF sal_achieve > (target_qty + 200) THEN
12         incentive := (sal_achieve - target_qty)/4;
13         UPDATE employees
14         SET salary = salary + incentive
15         WHERE employee_id = emp_id;
16     updated := 'Yes';
17 END IF;
18 DBMS_OUTPUT.PUT_LINE (
19     'Table updated? ' || updated || ',' ||
20     'incentive = ' || incentive || '.';
21 );
22 END test1;
```

Results

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

0.02 seconds
```

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

Oracle APEX 23.2.4

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

QUERY:

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

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'Bharath Kumar L bharath'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code from the previous block. The code is syntax-highlighted with red for keywords like DECLARE, PROCEDURE, IF, ELSE, etc., and blue for identifiers like test1, sal_achieve, incentive. The output pane at the bottom displays the results of the executed code: 'Sale achieved : 45000, incentive : 1800.', 'Sale achieved : 36000, incentive : 800.', and 'Sale achieved : 28000, incentive : 500.'. Below the results, it says 'Statement processed.' and '0.00 seconds'. The bottom footer includes copyright information for Oracle and the APEX version.

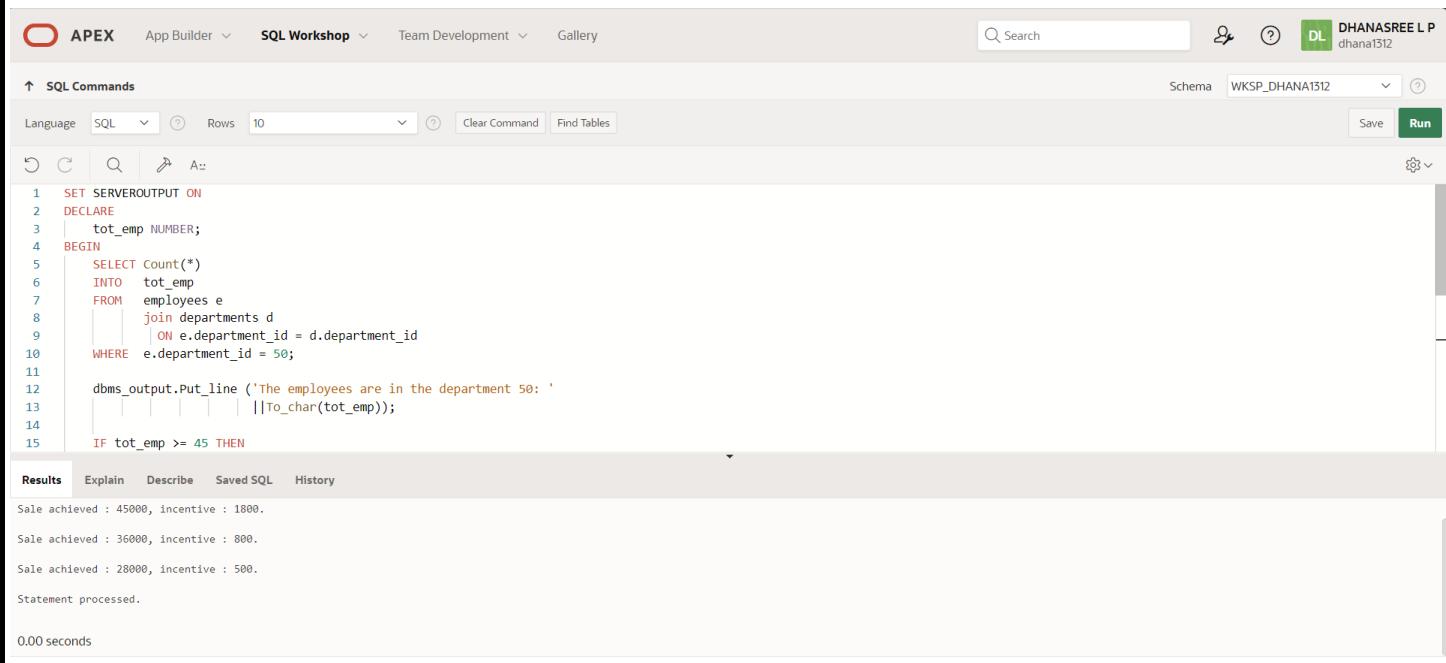
9.) Write a PL/SQL program to count number of employees in department 50 and check whether this department have any vacancies or not. There are 45 vacancies in this department.

QUERY:

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

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, there's a user icon for 'DHANASREE L P' and a schema dropdown set to 'WKSP_DHANA1512'. The main area is titled 'SQL Commands' with a 'Run' button. Below the title are filters for 'Language' (SQL), 'Rows' (10), and buttons for 'Clear Command' and 'Find Tables'. The code area contains the PL/SQL block from the previous step. The results section at the bottom displays the output of the executed code:

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

At the very bottom, it shows '0.00 seconds' and the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' followed by 'Oracle APEX 23.2.4'.

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

QUERY:

```
DECLARE
    tot_emp NUMBER;
    get_dep_id NUMBER;

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

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

    IF tot_emp >= 45 THEN
        dbms_output.Put_line ('There are no vacancies in the department '||get_dep_id);
    ELSE
        dbms_output.Put_line ('There are '||to_char(45-tot_emp)||' vacancies in department '||get_dep_id );
    END IF;
END;
/
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there are user profile icons for 'DHANASREE L P' (dhana1312) and a search bar. The main workspace displays a PL/SQL block:

```
1 DECLARE
2     tot_emp NUMBER;
3     get_dep_id NUMBER;
4
5 BEGIN
6     get_dep_id := 80;
7     SELECT Count(*)
8     INTO tot_emp
9     FROM employees e
10    join departments d
11    | ON e.department_id = d.dept_id;
12 WHERE e.department_id = get_dep_id;
13
14 dbms_output.Put_line ('The employees are in the department ||get_dep_id|| is: '
15                         ||To_char(tot_emp));
```

The code uses dynamic SQL to join the employees and departments tables based on department_id. It then counts the employees in department 80 and outputs the result using DBMS_OUTPUT.PUT_LINE. The results tab is selected, showing the output:

The employees are in the department 80 is: 4
There are 41 vacancies in department 80

Statement processed.

0.03 seconds

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

QUERY:

DECLARE

```
v_employee_id employees.employee_id%TYPE;
v_full_name employees.first_name%TYPE;
v_job_id employees.job_id%TYPE;
v_hire_date employees.hire_date%TYPE;
v_salary employees.salary%TYPE;
CURSOR c_employees IS
  SELECT employee_id, first_name || ' ' || last_
    FROM employees;
```

BEGIN

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

OUTPUT:

```

1  DECLARE
2    v_employee_id employees.employee_id%TYPE;
3    v_full_name employees.first_name%TYPE;
4    v_job_id employees.job_id%TYPE;
5    v_hire_date employees.hire_date%TYPE;
6    v_salary employees.salary%TYPE;
7    CURSOR c_employees IS
8      SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
9      FROM employees;
10   BEGIN
11     DBMS_OUTPUT.PUT_LINE('Employee ID | Full Name | Job Title | Hire Date | Salary');
12     DBMS_OUTPUT.PUT_LINE('-----');
13     OPEN c_employees;

```

Results

Employee ID	Full Name	Job Title	Hire Date	Salary
110	John Smith	sales_rep	02/28/1995	70000
125	Emily Johnson	hr_rep	04/26/1998	5000
122	Jaunty Janu	ac_account	03/05/2024	6900
114	den Davies	st_clerk	02/03/1999	11000
142	Jane Doe	ac_account	03/05/1997	30000
115	Vijaya Mohan	st_clerk	02/22/1998	4000

Statement processed.

220701045@rajalakshmi.edu.in bharath en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

DECLARE

```

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

```

OUTPUT:

```

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

```

Results Explain Describe Saved SQL History

```

Employee ID: 142
Employee Name: Jane
Manager Name:
-----
Employee ID: 110
Employee Name: John
Manager Name:
-----
Employee ID: 125
Employee Name: Emily
Manager Name:
-----
```

220701045@rajalakshmi.edu.in bharath en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

QUERY:

```

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

```

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```

1 DECLARE
2   CURSOR job_cursor IS
3     SELECT e.job_id, j.lowest_sal
4       FROM job_grade j,employees e;
5   job_record job_cursor%ROWTYPE;
6 BEGIN
7   OPEN job_cursor;
8   FETCH job_cursor INTO job_record;
9   WHILE job_cursor%FOUND LOOP
10    DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_record.job_id);
11    DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_record.lowest_sal);
12    DBMS_OUTPUT.PUT_LINE('-----');
13   FETCH job_cursor INTO job_record;

```

Results Explain Describe Saved SQL History

Job ID: sales_rep
Minimum Salary: 2500

Job ID: hr_rep
Minimum Salary: 2500

Job ID: ac_account
Minimum Salary: 2500

Job ID: st_clerk
Minimum Salary: 2500

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

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

QUERY:

DECLARE

```

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

BEGIN
  dbms_output.Put_line(Rpad('Employee ID', 15) || Rpad('Last Name', 25) || Rpad('Job Id', 35)
  || 'Start Date');
  dbms_output.Put_line('-----');

  FOR emp_sal_rec IN employees_cur LOOP
    -- find out most recent end_date in job_history
    SELECT Max(end_date) + 1
      INTO emp_start_date
      FROM job_history
     WHERE employee_id = emp_sal_rec.employee_id;
    IF emp_start_date IS NULL THEN
      emp_start_date := emp_sal_rec.start_date;
    END IF;
  
```

```

dbms_output.Put_line(Rpad(emp_sal_rec.employee_id, 15)
    || Rpad(emp_sal_rec.last_name, 25)
    || Rpad(emp_sal_rec.job_id, 35)
    || To_char(emp_start_date, 'dd-mon-yyyy'));

END LOOP;
END;
/

```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the schema 'WKSP_DHANA1312' and a user icon 'DL DHANASREE L P dhana1312'. The main area is titled 'SQL Commands' with a 'Run' button. The code block is as follows:

```

1  DECLARE
2      CURSOR employees_cur IS
3          SELECT employee_id,
4                  last_name,
5                  job_id,
6                  start_date
7              FROM employees
8          NATURAL JOIN job_history;
9          emp_start_date DATE;
10 BEGIN
11     dbms_output.Put_line(Rpad('Employee ID', 15)
12                           || Rpad('Last Name', 25)
13                           || Rpad('Job Id', 35)
14                           || 'Start Date');
15
16     dbms_output.Put_line('-----');
17

```

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

Employee ID	Last Name	Job Id	Start Date
125	Johnson	hr_rep	22-apr-1999
125	Johnson	hr_rep	22-apr-1999

Statement processed.

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

QUERY:

```

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

```

```
CLOSE c_employees;
END;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user information for 'DHANASREE L P' are also present. The main area displays a PL/SQL block:

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

```

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

```
Employee ID: 125
Employee Name: Emily
End Date: 04/21/1999
-----
Employee ID: 125
Employee Name: Emily
End Date: 03/21/1997
-----
```

Below the results, a message indicates the statement was processed.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	

Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

PROCEDURES AND FUNCTIONS

EX_NO: 17

DATE:

1.) Factorial of a number using function.

QUERY:

DECLARE

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

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

BEGIN

```
    WHILE n > 0 LOOP
```

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

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

END LOOP;

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information (Dhanasree L P, dhana1312), and a schema dropdown (WKSP_DHANA1312). The main area is titled 'SQL Commands' and contains a code editor with numbered lines. The code is a PL/SQL block for calculating factorial. Below the code editor is a toolbar with icons for Undo, Redo, Search, Find, and Run. The results tab is selected, showing the output of the executed command: 'Statement processed.' and '0.00 seconds'. Other tabs available are Explain, Describe, Saved SQL, and History.

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

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

QUERY:

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

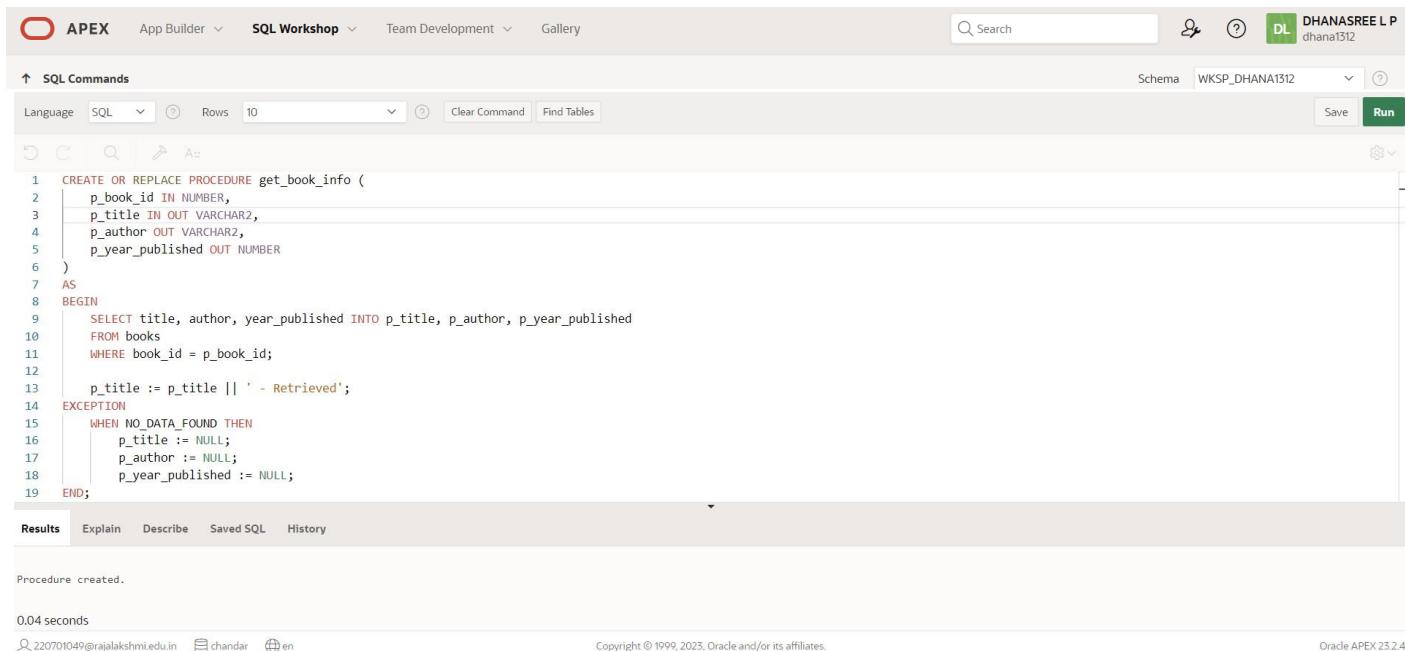
    p_title := p_title || ' - Retrieved';
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        p_title := NULL;
        p_author := NULL;
        p_year_published := NULL;
END;

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

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

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

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'DHANASREE L P' with ID 'dhana1312'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below it is a code editor containing the PL/SQL code for a procedure named 'get_book_info'. The code uses IN and OUT parameters to retrieve book details from the 'books' table. It includes a RETURN statement and an EXCEPTION block for handling NO_DATA_FOUND. The bottom of the screen shows the results of the execution, indicating 'Procedure created.' and a execution time of '0.04 seconds'. The footer includes copyright information for Oracle and the APEX version.

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

Results Explain Describe Saved SQL History

Procedure created.

0.04 seconds

220701049@rajalakshmi.edu.in chandar en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

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

RESULT:

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

TRIGGER

EX_NO: 18

DATE:

1.) Write a code in PL/SQL to develop a trigger that enforces referential integrity by preventing the deletion of a parent record if child records exist

QUERY:

```
CREATE OR REPLACE TRIGGER prevent_parent_deletion
BEFORE DELETE ON parent_table
FOR EACH ROW
DECLARE
    child_exists EXCEPTION;
    PRAGMA EXCEPTION_INIT(child_exists, -20001);
    v_child_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id =
:OLD.parent_id;
    IF v_child_count > 0 THEN
        RAISE child_exists;
    END IF;
EXCEPTION
    WHEN child_exists THEN
        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child records
exist.');
END;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'DHANASREE L P', and a schema dropdown set to 'WKSP_DHANA1312'. Below the tabs, there's a toolbar with icons for Save, Run, and other functions. The main area 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);
7     v_child_count NUMBER;
8 BEGIN
9     SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id = :OLD.parent_id;
10    IF v_child_count > 0 THEN
11        RAISE child_exists;
12    END IF;
13 EXCEPTION
14    WHEN child_exists THEN
15        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child records exist.');
16 END;
17 /

```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab shows the output: "Trigger created." and "0.04 seconds". At the bottom, it shows the user information (220701049@rajalakshmi.edu.in, chandar, en), copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates), and the Oracle APEX version (Oracle APEX 23.2.4).

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

QUERY:

```

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

```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' are visible. On the right, there's a search bar, user profile icon, and a session identifier 'DHANASREE L P dhana1312'. The main area is titled 'SQL Commands' with a sub-tab 'Language: SQL'. Below the tabs are buttons for 'Rows' (set to 10), 'Clear Command', and 'Find Tables'. To the right are 'Save' and 'Run' buttons. The code editor contains the following PL/SQL trigger definition:

```

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

```

Below the code editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the output: 'Trigger created.' and '0.04 seconds'. At the bottom, it shows the user information: '220701049@rajalakshmi.edu.in', 'chandar', and 'en', along with copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

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

QUERY:

```

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

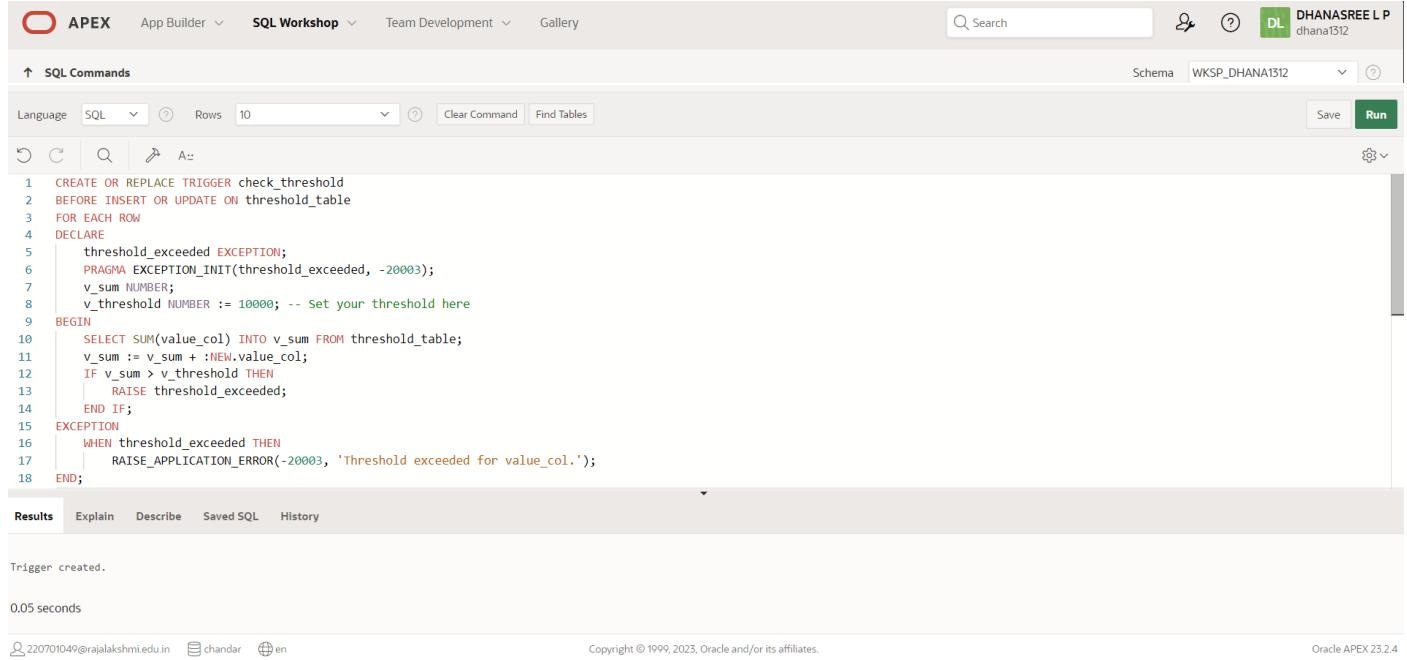
```

```

    RAISE threshold_exceeded;
END IF;
EXCEPTION
WHEN threshold_exceeded THEN
    RAISE_APPLICATION_ERROR(-20003, 'Threshold exceeded for value_col.');
END;

```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'DHANASREE L P' (dhana1312). The main workspace displays the PL/SQL code for creating a trigger named 'check_threshold'. The code uses a cursor to sum values from the 'threshold_table' and compares it against a threshold of 10000. If the sum exceeds the threshold, it raises an exception. An exception block handles this by raising the 'threshold_exceeded' exception with a specific error message. The code is numbered from 1 to 18. Below the code, the 'Results' tab is selected, showing the message 'Trigger created.' and a timestamp of '0.05 seconds'. The bottom of the screen shows the Oracle APEX footer with copyright information.

```

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

```

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'DHANASREE L P' (dhana1312). The main workspace is titled 'SQL Commands'. The SQL editor contains the trigger creation script. The 'Results' tab is selected at the bottom, showing the output: 'Trigger created.' and '0.04 seconds'. The bottom footer includes copyright information for Oracle and the APEX version.

```
1 CREATE OR REPLACE TRIGGER log_changes
2 AFTER UPDATE ON main_table
3 FOR EACH ROW
4 BEGIN
5     INSERT INTO audit_table (audit_id, changed_id, old_col1, new_col1, old_col2, new_col2,
6 change_time)
7     VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2, :NEW.col2,
8 SYSTIMESTAMP);
9 END;
10 /
```

Trigger created.
0.04 seconds

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, the user 'DHANASREE L P' (dhana1312) is logged in. The main area is titled 'SQL Commands' with a search bar and a schema dropdown set to 'WKSP_DHANA1312'. Below the search bar are buttons for 'Save' and 'Run'. The code editor contains the following PL/SQL trigger definition:

```

1 CREATE OR REPLACE TRIGGER log_user_activity
2 AFTER INSERT OR UPDATE OR DELETE ON activity_table
3 FOR EACH ROW
4 BEGIN
5   IF INSERTING THEN
6     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
7       VALUES (activity_log_seq.NEXTVAL, 'INSERT', 'activity_table', :NEW.id, SYSTIMESTAMP);
8   ELSIF UPDATING THEN
9     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
10    VALUES (activity_log_seq.NEXTVAL, 'UPDATE', 'activity_table', :NEW.id, SYSTIMESTAMP);
11   ELSIF DELETING THEN
12     INSERT INTO user_activity_log (log_id, action, table_name, record_id, change_time)
13       VALUES (activity_log_seq.NEXTVAL, 'DELETE', 'activity_table', :OLD.id, SYSTIMESTAMP);
14   END IF;
15 END;
16 /

```

Below the code editor, the 'Results' tab is selected, showing the output: 'Trigger created.' and '0.05 seconds'. At the bottom, the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are visible.

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

QUERY:

```

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

```

END;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'DHANASREE L P' (dhanalakshmi) and a search bar. The main workspace is titled 'SQL Commands'. The code area contains the following PL/SQL code:

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

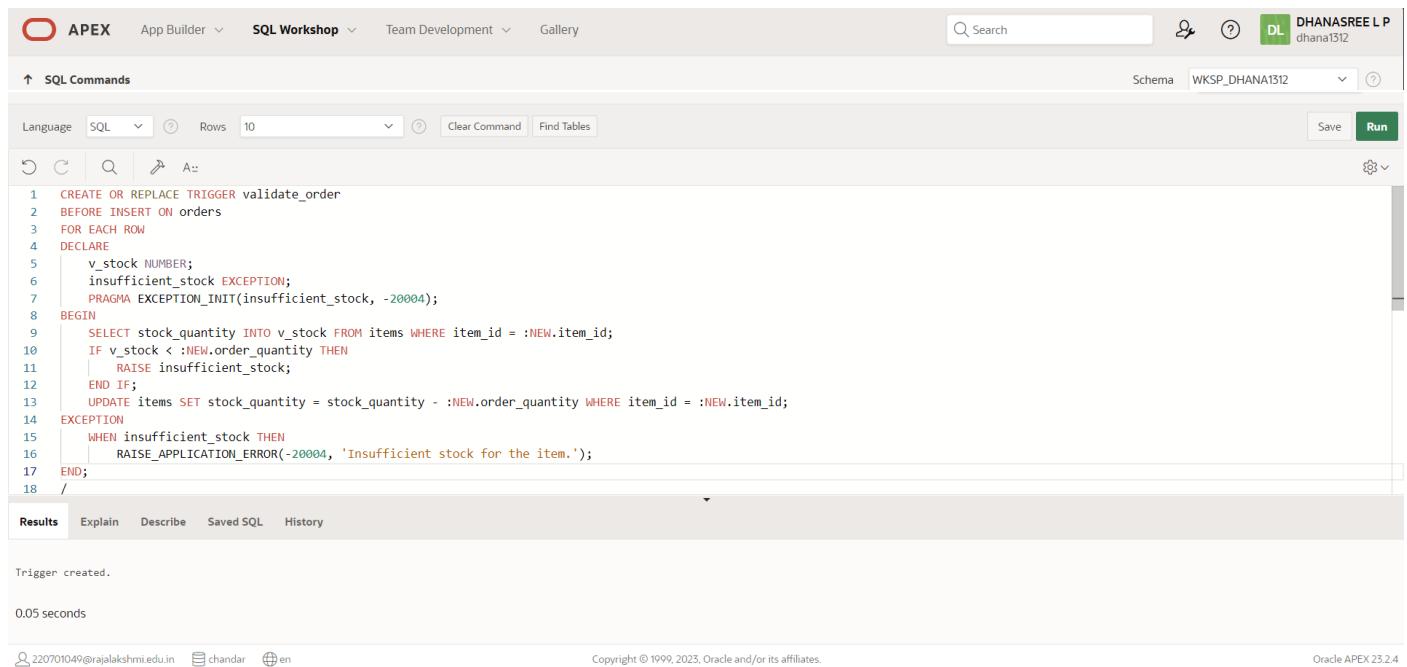
Below the code, the 'Results' tab is selected, showing the output: 'Trigger created.' and '0.04 seconds'. The bottom footer includes copyright information for Oracle and the APEX version.

7.) Write a code in PL/SQL to create a trigger that validates the availability of items before allowing an order to be placed, considering stock levels and pending orders

QUERY:

```
CREATE OR REPLACE TRIGGER validate_order
BEFORE INSERT ON orders
FOR EACH ROW
DECLARE
    v_stock NUMBER;
    insufficient_stock EXCEPTION;
    PRAGMA EXCEPTION_INIT(insufficient_stock, -20004);
BEGIN
    SELECT stock_quantity INTO v_stock FROM items WHERE item_id = :NEW.item_id;
    IF v_stock < :NEW.order_quantity THEN
        RAISE insufficient_stock;
    END IF;
    UPDATE items SET stock_quantity = stock_quantity - :NEW.order_quantity WHERE item_id
    = :NEW.item_id;
EXCEPTION
    WHEN insufficient_stock THEN
        RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');
END;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user 'DHANASREE L P' and the schema 'WKSP_DHANA1512'. The main workspace is titled 'SQL Commands' and contains the PL/SQL code for the 'validate_order' trigger. The code is syntax-highlighted, showing keywords in red and identifiers in blue. The bottom of the screen displays the 'Results' tab, which shows the message 'Trigger created.' and a execution time of '0.05 seconds'. The footer includes copyright information for Oracle and the APEX version 'Oracle APEX 23.2.4'.

```
1 CREATE OR REPLACE TRIGGER validate_order
2 BEFORE INSERT ON orders
3 FOR EACH ROW
4 DECLARE
5     v_stock NUMBER;
6     insufficient_stock EXCEPTION;
7     PRAGMA EXCEPTION_INIT(insufficient_stock, -20004);
8 BEGIN
9     SELECT stock_quantity INTO v_stock FROM items WHERE item_id = :NEW.item_id;
10    IF v_stock < :NEW.order_quantity THEN
11        RAISE insufficient_stock;
12    END IF;
13    UPDATE items SET stock_quantity = stock_quantity - :NEW.order_quantity WHERE item_id = :NEW.item_id;
14 EXCEPTION
15    WHEN insufficient_stock THEN
16        RAISE_APPLICATION_ERROR(-20004, 'Insufficient stock for the item.');
17 END;
18 /
```

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

RESULT:

MONGO DB

EX_NO: 19

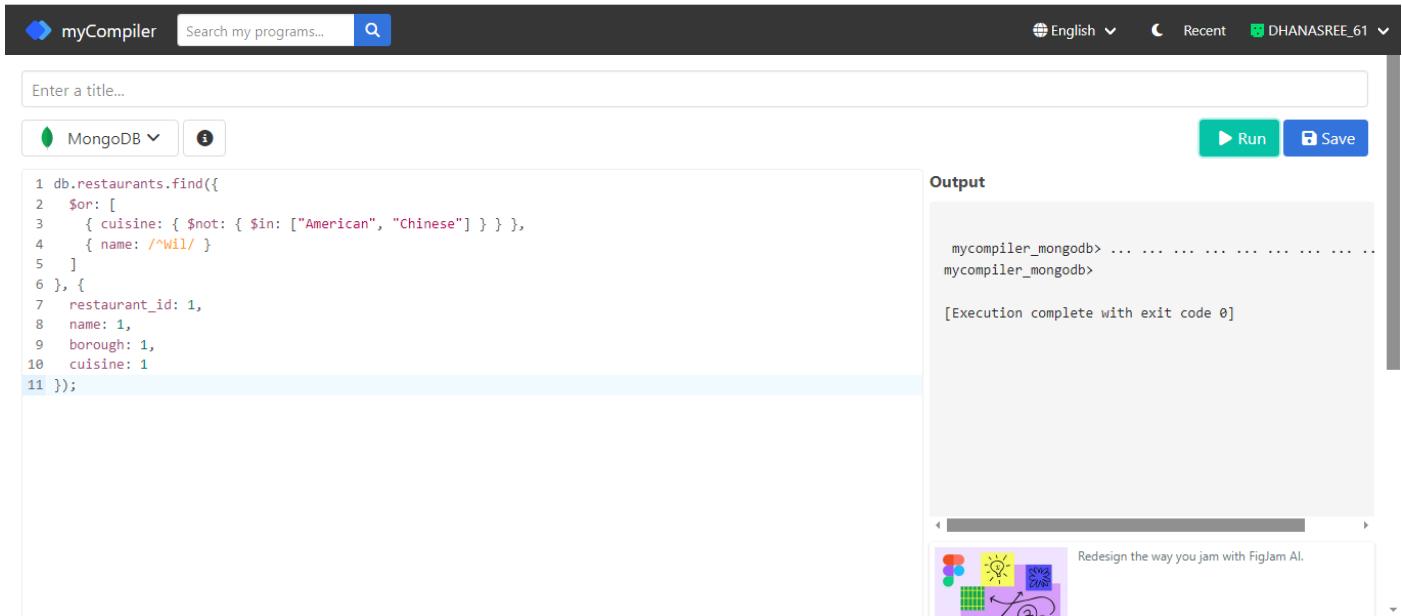
DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. In the top bar, there are icons for myCompiler, English, Recent, and a user profile (DHANASREE_61). Below the bar, there is a search bar labeled "Search my programs..." and a magnifying glass icon. A text input field is labeled "Enter a title...". To the right of the input field are buttons for "MongoDB" (with a dropdown arrow), "Run" (green button), and "Save" (blue button). The main area contains the following MongoDB code:

```
1 db.restaurants.find({  
2   $or: [  
3     { cuisine: { $not: { $in: ["American", "Chinese"] } } },  
4     { name: /^Wil/ }  
5   ]  
6 }, {  
7   restaurant_id: 1,  
8   name: 1,  
9   borough: 1,  
10  cuisine: 1  
11});
```

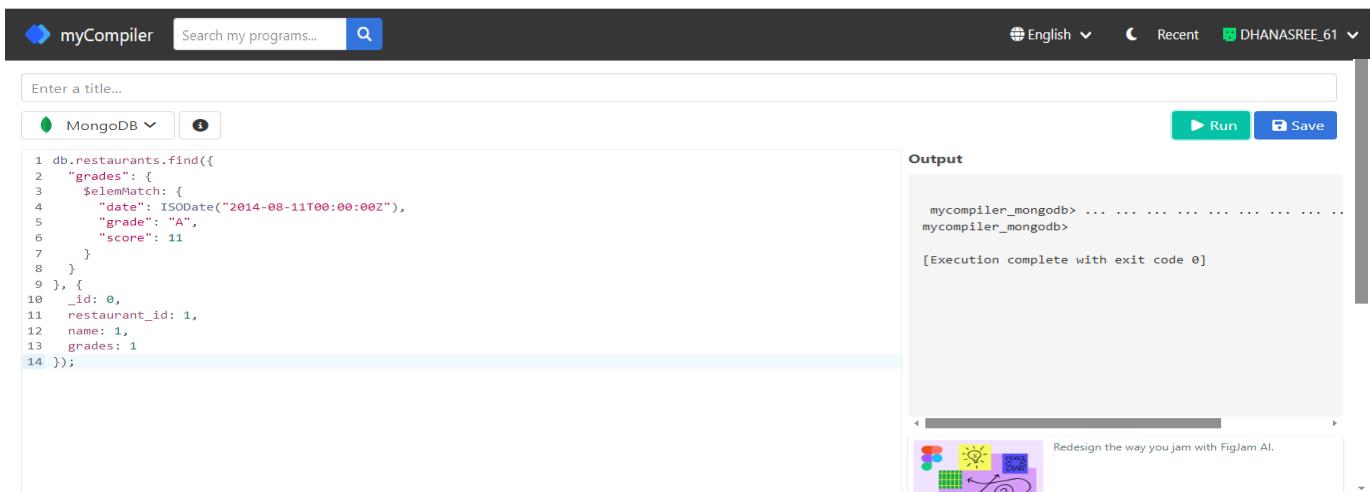
To the right of the code is an "Output" window. It shows the command "mycompiler_mongodb>" followed by "mycompiler_mongodb>". Below that is the message "[Execution complete with exit code 0]". At the bottom of the output window is a decorative footer with icons and the text "Redesign the way you jam with FigJam AI".

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:



The screenshot shows the myCompiler interface. In the top bar, there are icons for myCompiler, English, Recent, and a user profile (DHANASREE_61). Below the bar, there is a search bar labeled "Search my programs..." and a magnifying glass icon. A text input field is labeled "Enter a title...". To the right of the input field are buttons for "MongoDB" (with a dropdown arrow), "Run" (green button), and "Save" (blue button). The main area contains the following MongoDB code:

```
1 db.restaurants.find({  
2   "grades": {  
3     "$elemMatch": {  
4       "date": ISODate("2014-08-11T00:00:00Z"),  
5       "grade": "A",  
6       "score": 11  
7     }  
8   }  
9 }, {  
10   _id: 0,  
11   restaurant_id: 1,  
12   name: 1,  
13   grades: 1  
14 });
```

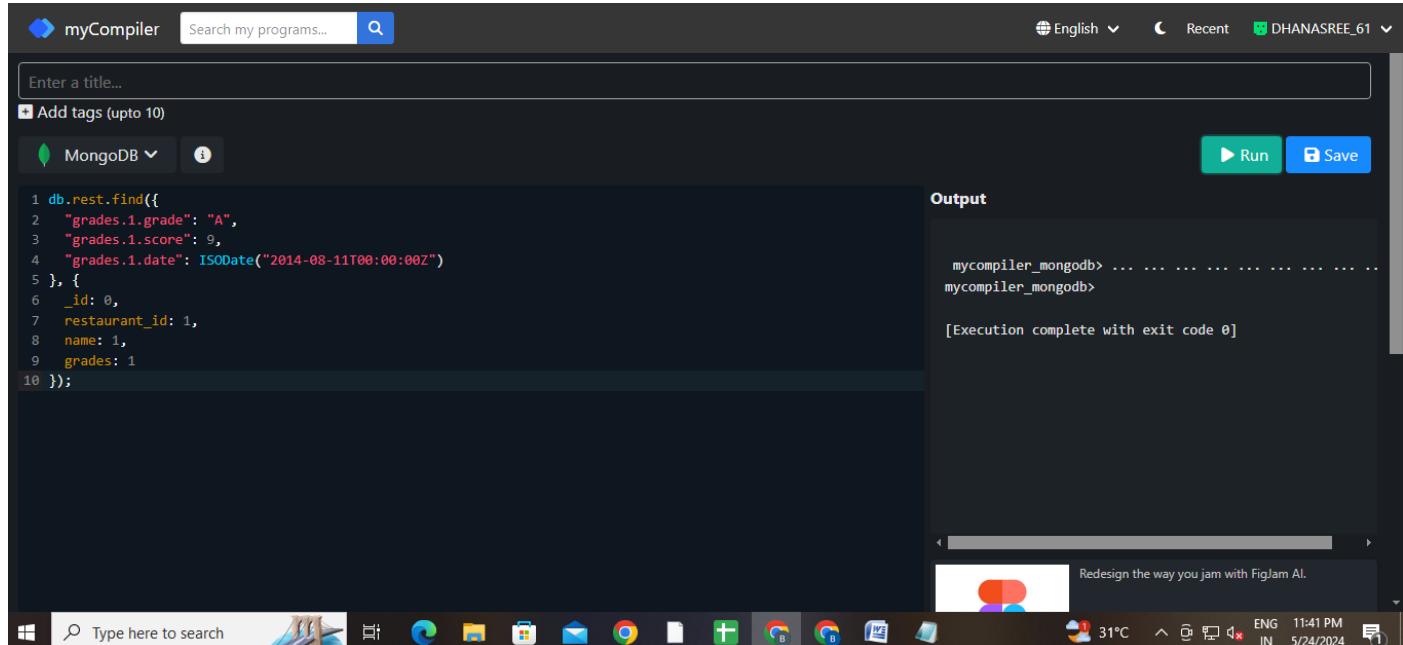
To the right of the code is an "Output" window. It shows the command "mycompiler_mongodb>" followed by "mycompiler_mongodb>". Below that is the message "[Execution complete with exit code 0]". At the bottom of the output window is a decorative footer with icons and the text "Redesign the way you jam with FigJam AI".

3.) Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z".

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. On the left, there's a code editor window with the following MongoDB query:

```
1 db.restaurants.find({  
2   "grades.1.grade": "A",  
3   "grades.1.score": 9,  
4   "grades.1.date": ISODate("2014-08-11T00:00:00Z")  
5 }, {  
6   _id: 0,  
7   restaurant_id: 1,  
8   name: 1,  
9   grades: 1  
10});
```

On the right, there's an "Output" window showing the command prompt:

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

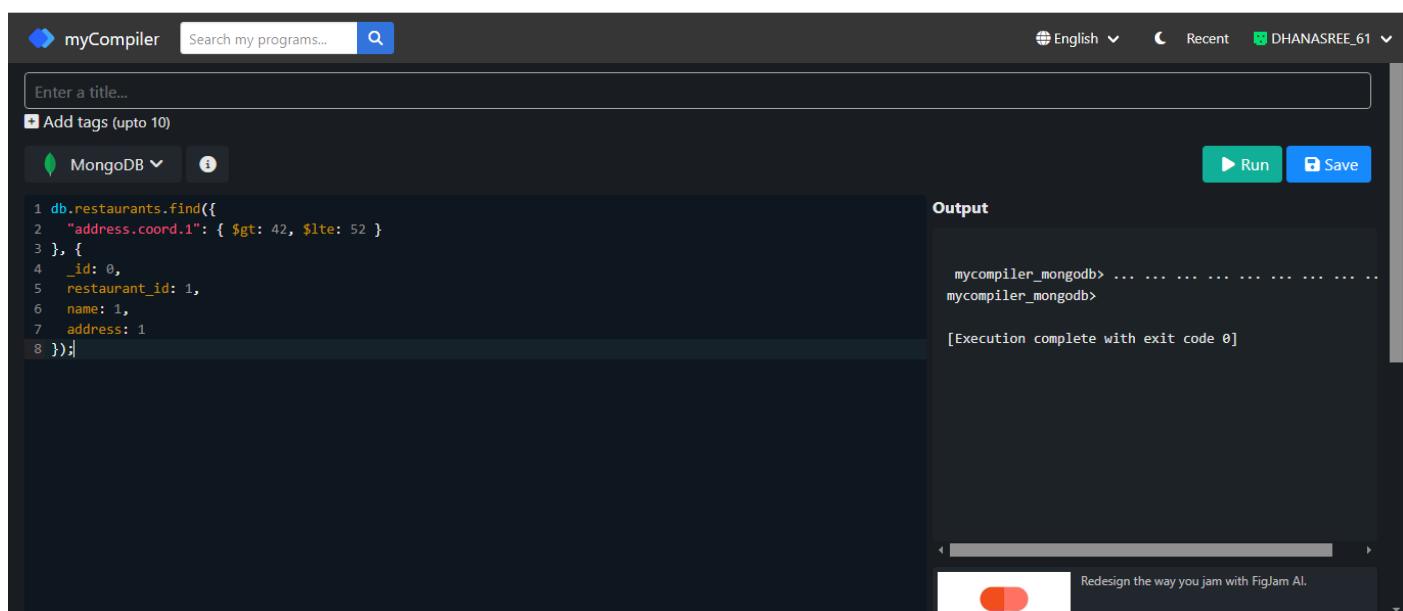
The bottom of the screen shows a Windows taskbar with various icons and system status.

4.) Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. On the left, there's a code editor window with the following MongoDB query:

```
1 db.restaurants.find({  
2   "address.coord.1": { $gt: 42, $lte: 52 }  
3 }, {  
4   _id: 0,  
5   restaurant_id: 1,  
6   name: 1,  
7   address: 1  
8});
```

On the right, there's an "Output" window showing the command prompt:

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

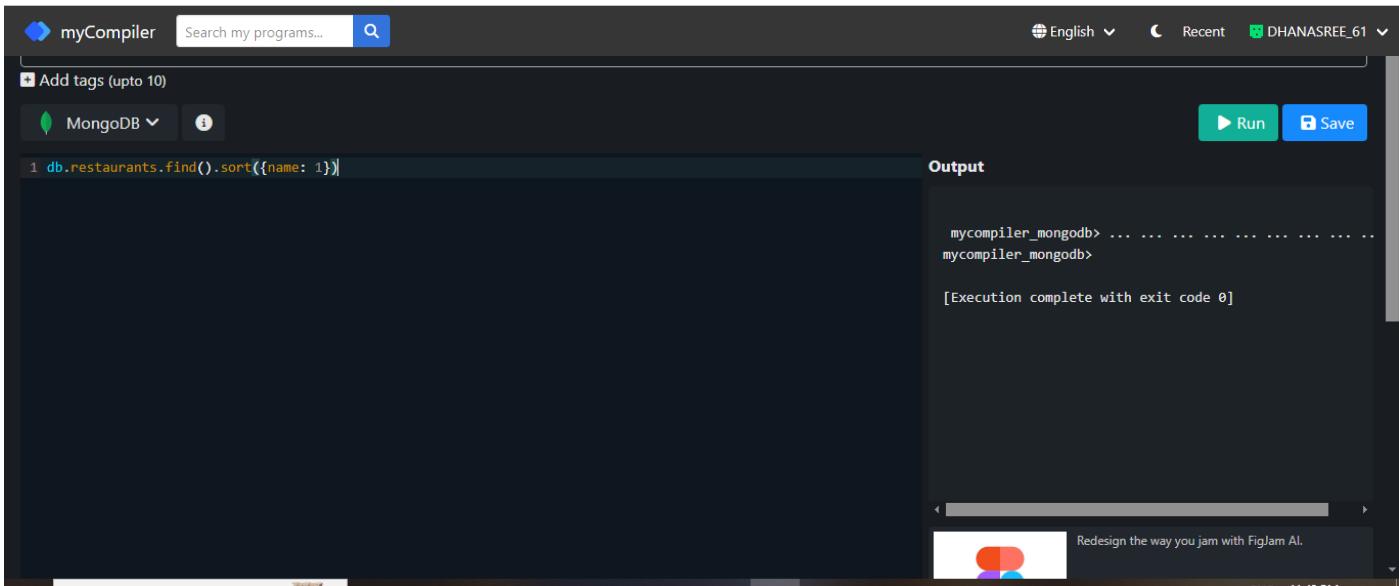
The bottom of the screen shows a Windows taskbar with various icons and system status.

5.) Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

QUERY:

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

OUTPUT:



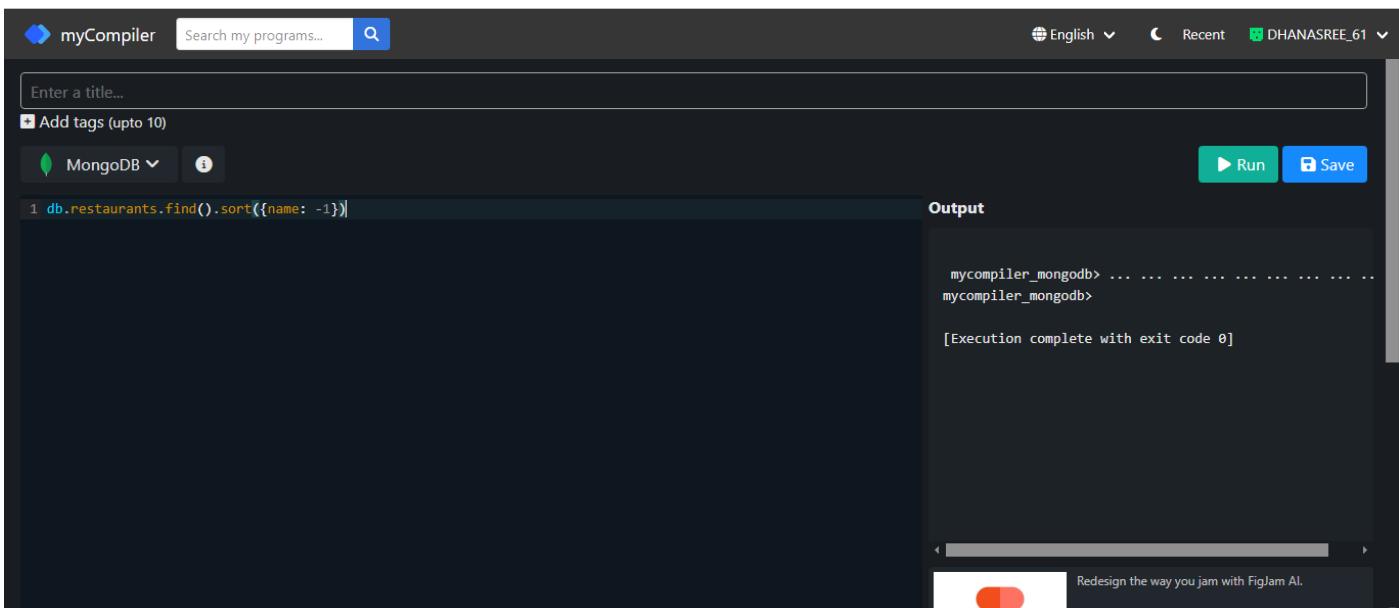
The screenshot shows the myCompiler interface. In the top bar, there are icons for myCompiler, search, language selection (English), recent projects (Recent), and user profile (DHANASREE_61). Below the bar, there's a toolbar with 'Add tags (upto 10)', a MongoDB dropdown, and 'Run' and 'Save' buttons. The main area has a dark background with white text. On the left, a code editor window displays the MongoDB query: `1 db.restaurants.find().sort({name: 1})`. On the right, an 'Output' window shows the command prompt: `mycompiler_mongodb> ...`, followed by several dots, and then `mycompiler_mongodb>`. Below this, it says '[Execution complete with exit code 0]'. At the bottom of the interface, there's a watermark for FigJam AI.

6.) Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. The layout is identical to the previous one, with the same top bar, toolbar, and watermark. The code editor window now contains the MongoDB query: `1 db.restaurants.find().sort({name: -1})`. The output window shows the command prompt: `mycompiler_mongodb> ...`, followed by several dots, and then `mycompiler_mongodb>`. Below this, it says '[Execution complete with exit code 0]'. The watermark for FigJam AI is at the bottom.

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:

The screenshot shows the myCompiler interface. In the top bar, there are icons for English, Recent, and a user account named DHANASREE_61. Below the bar, there's a search bar with placeholder text "Enter a title..." and a "Run" button. A "MongoDB" dropdown menu is open, showing a "Run" button and a "Save" button. The main area contains a code editor with the following MongoDB query:

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

To the right of the code editor is an "Output" panel. It displays the command "mycompiler_mongodb>" followed by "[Execution complete with exit code 0]".

8.) Write a MongoDB query to know whether all the addresses contains the street or not.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler interface. In the top bar, there are icons for English, Recent, and a user account named DHANASREE_61. Below the bar, there's a search bar with placeholder text "Enter a title..." and a "Run" button. A "MongoDB" dropdown menu is open, showing a "Run" button and a "Save" button. The main area contains a code editor with the following MongoDB query:

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

To the right of the code editor is an "Output" panel. It displays the command "mycompiler_mongodb>" followed by "[Execution complete with exit code 0]".

9.) Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. In the code editor, the following query is written:

```
1 db.restaurants.find({ "address.coord.0": { $type: "double" }, "address.coord.1": { $type: "double" } })
```

In the output panel, the results of the query are shown:

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

10. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. In the code editor, the following query is written:

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

In the output panel, the results of the query are shown:

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

11. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler web application interface. At the top, there's a navigation bar with the logo 'myCompiler', a search bar containing 'Search my programs...', and a magnifying glass icon. To the right are links for 'English' (with a dropdown arrow), 'Recent', and a user profile 'DHANASREE_61'. Below the header, a large input field prompts 'Enter a title...'. Underneath it, a button says '+ Add tags (upto 10)'. On the left, there's a MongoDB icon with a dropdown menu and a help icon. The main workspace contains a code editor with the following query:

```
1 db.restaurants.find({ name: { $regex: /mon/1 } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })
```

To the right of the code editor is a green 'Run' button with a play icon and a blue 'Save' button with a disk icon. Below the code editor is a section titled 'Output' which displays the results of the MongoDB query. The output shows the command 'mycompiler_mongodb> ...' followed by '[Execution complete with exit code 0]'. A small watermark for 'Redesign the way you jam with FigJam AI.' is visible at the bottom right.

12. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler interface. At the top, there's a search bar with "Search my programs..." and a magnifying glass icon. To the right are language selection ("English"), recent projects ("Recent"), and user information ("DHANASREE_61"). Below the header is a title input field ("Enter a title...") and a "Add tags (upto 10)" button. The main area has a "MongoDB" dropdown and a "Run" button. The code input field contains a MongoDB query: `db.restaurants.find({ name: { $regex: /^Mad/i } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })`. To the right, the "Output" panel shows the results of the query execution.

```
1 db.restaurants.find({ name: { $regex: /^Mad/i } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })
```

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

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:

The screenshot shows the myCompiler MongoDB interface. In the code editor, a query is written:

```
1 db.restaurants.find({ "grades.score": { $lt: 5 } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })
```

In the output panel, the results of the query are shown:

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

14. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. In the code editor, a query is written:

```
1 db.restaurants.find({ "grades.score": { $lt: 5 }, borough: "Manhattan" }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })
```

In the output panel, the results of the query are shown:

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

15. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. At the top, there's a search bar with "Search my programs..." and a magnifying glass icon. On the right, there are language selection ("English"), recent projects ("Recent"), and user information ("DHANASREE_61"). Below the header, there's a text input field for "Enter a title..." and a checkbox for "Add tags (upto 10)". A "MongoDB" dropdown menu is open, showing "MongoDB" and "Info". To the right of the dropdown are "Run" and "Save" buttons. The main area displays a query and its output. The query is:

```
1, {borough: "Brooklyn"}], "grades.score": { $lt: 5 } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 }) Output
```

The output shows the results of the query:

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

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

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. At the top, there's a search bar with "Search my programs..." and a magnifying glass icon. On the right, there are language selection ("English"), recent projects ("Recent"), and user information ("DHANASREE_61"). Below the header, there's a text input field for "Enter a title..." and a checkbox for "Add tags (upto 10)". A "MongoDB" dropdown menu is open, showing "MongoDB" and "Info". To the right of the dropdown are "Run" and "Save" buttons. The main area displays a query and its output. The query is:

```
1, {grades.score": { $lt: 5 }, cuisine: { $ne: "American" } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 }) Output
```

The output shows the results of the query:

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

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

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. In the top right corner, there are language and recent project dropdowns, and a user profile icon for DHANASREE_61. The main area has tabs for 'MongoDB' and 'Run/Save' buttons. A search bar says 'Enter a title...' and a tags section says 'Add tags (upto 10)'. Below is a code editor with the following query:

```
1$lt: 5 }, cuisine: { $nin: ["American", "Chinese"] } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })
```

The output window shows the command being run and the response:

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

18. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. In the top right corner, there are language and recent project dropdowns, and a user profile icon for DHANASREE_61. The main area has tabs for 'MongoDB' and 'Run/Save' buttons. A search bar says 'Enter a title...' and a tags section says 'Add tags (upto 10)'. Below is a code editor with the following query:

```
1b.restaurants.find({ "grades.score": { $in: [2, 6] } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })
```

The output window shows the command being run and the response:

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

19. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler interface. At the top, there's a search bar with "Search my programs..." and a magnifying glass icon. On the right, there are language selection ("English"), recent projects ("Recent"), and user information ("DHANASREE_61"). Below the search bar is a title input field with "Enter a title..." and a "Add tags (upto 10)" button. A dropdown menu shows "MongoDB" and an info icon. To the right are "Run" and "Save" buttons. The main area contains a code editor with the following MongoDB query:

```
1grades.score": { $in: [2, 6] }, borough: "Manhattan" }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 }) Output
```

Below the code editor is a terminal window showing the command "mycompiler_mongodb> ...". The output section shows the command again followed by "[Execution complete with exit code 0]". At the bottom right of the interface, there's an advertisement for FigJam AI.

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:

The screenshot shows the myCompiler interface. At the top, there's a search bar with "Search my programs..." and a magnifying glass icon. On the right, there are language selection ("English"), recent projects ("Recent"), and user information ("DHANASREE_61"). Below the search bar is a title input field with "Enter a title..." and a "Add tags (upto 10)" button. A dropdown menu shows "MongoDB" and an info icon. To the right are "Run" and "Save" buttons. The main area contains a code editor with the following MongoDB query:

```
1rough: "Brooklyn"}], "grades.score": { $in: [2, 6] } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 }) Output
```

Below the code editor is a terminal window showing the command "mycompiler_mongodb> ...". The output section shows the command again followed by "[Execution complete with exit code 0]". At the bottom right of the interface, there's an advertisement for FigJam AI.

21. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. In the top bar, there are tabs for English, Recent, and the user DHANASREE_61. The main area has a search bar 'Search my programs...' and a 'Run' button. A dropdown menu shows 'MongoDB'. Below the search bar is a text input field 'Enter a title...' and a 'Add tags (upto 10)' button. The code editor contains the following MongoDB query:

```
1{ $in: [2, 6] } }, { cuisine: { $ne: "American" } }], { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })| Output
```

The output window shows the command being run and the result:

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

At the bottom right of the interface, there is an advertisement for FigJam AI.

22. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler MongoDB interface. The top bar includes tabs for English, Recent, and the user DHANASREE_61. The main area features a search bar 'Search my programs...' and a 'Run' button. A 'MongoDB' dropdown is visible. The code editor contains the same MongoDB query as the previous screenshot:

```
1{ $in: [2, 6] } }, { cuisine: { $nin: ["American", "Chinese"] } }], { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })| Output
```

The output window displays the command and its execution results:

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

An advertisement for FigJam AI is present at the bottom right.

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:

myCompiler Search my programs... 

Enter a title...

Add tags (upto 10) 

MongoDB  

 Run  Save

```
1 db.restaurants.find({ "grades.score": { $in: [2, 6] } }, { _id: 0, name: 1, borough: 1, address: 1, cuisine: 1 })| Output
```

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

 Redesign the way you jam with FigJam AI.
ADS VIA CARBON

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

RESULT:

MONGO DB

EX_NO: 20

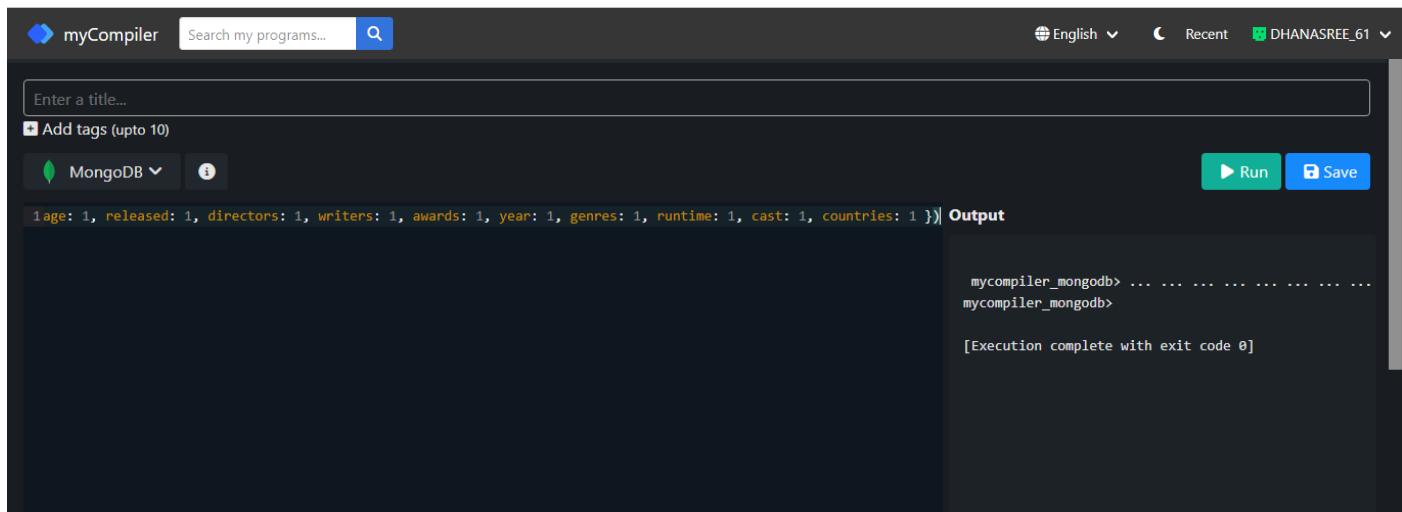
DATE:

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

QUERY:

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

OUTPUT:



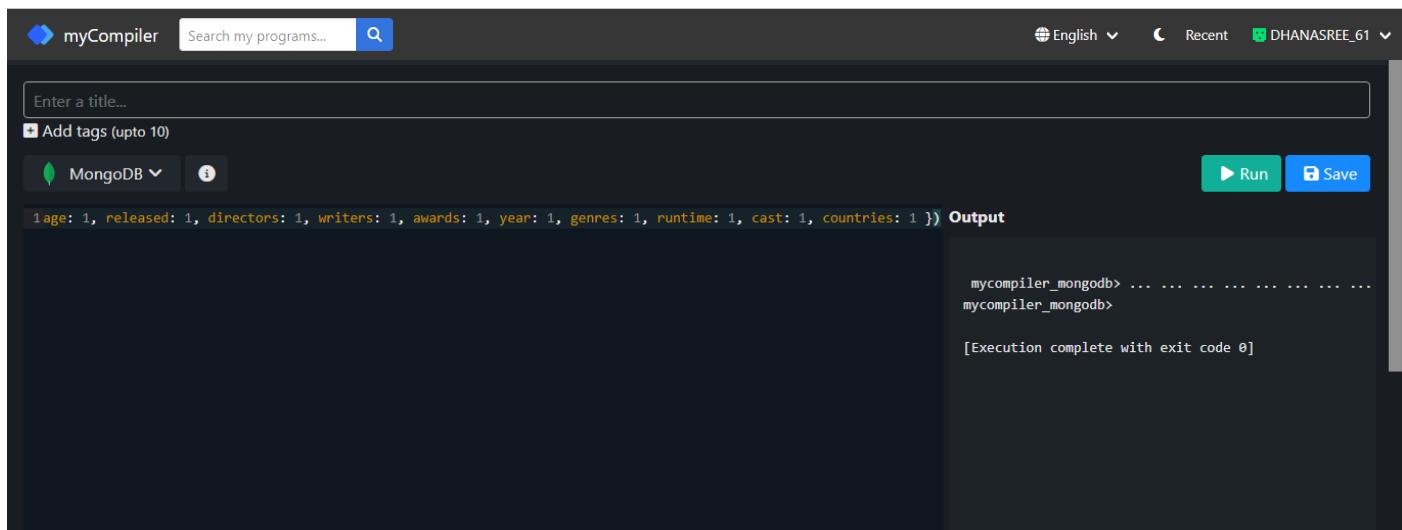
The screenshot shows the 'myCompiler' interface with a dark theme. At the top, there's a search bar labeled 'Search my programs...' and a 'Run' button. Below the search bar is a text input field with placeholder text 'Enter a title...'. Underneath the input field is a button labeled 'Add tags (upto 10)'. On the left side, there's a dropdown menu set to 'MongoDB' and a help icon. On the right side, there are 'Run' and 'Save' buttons. The main area is titled 'Output' and contains the following text:
`1 age: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }`
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

2.)Find all movies with full information from the 'movies' collection that have a runtime greater than 120 minutes.

QUERY:

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

OUTPUT:



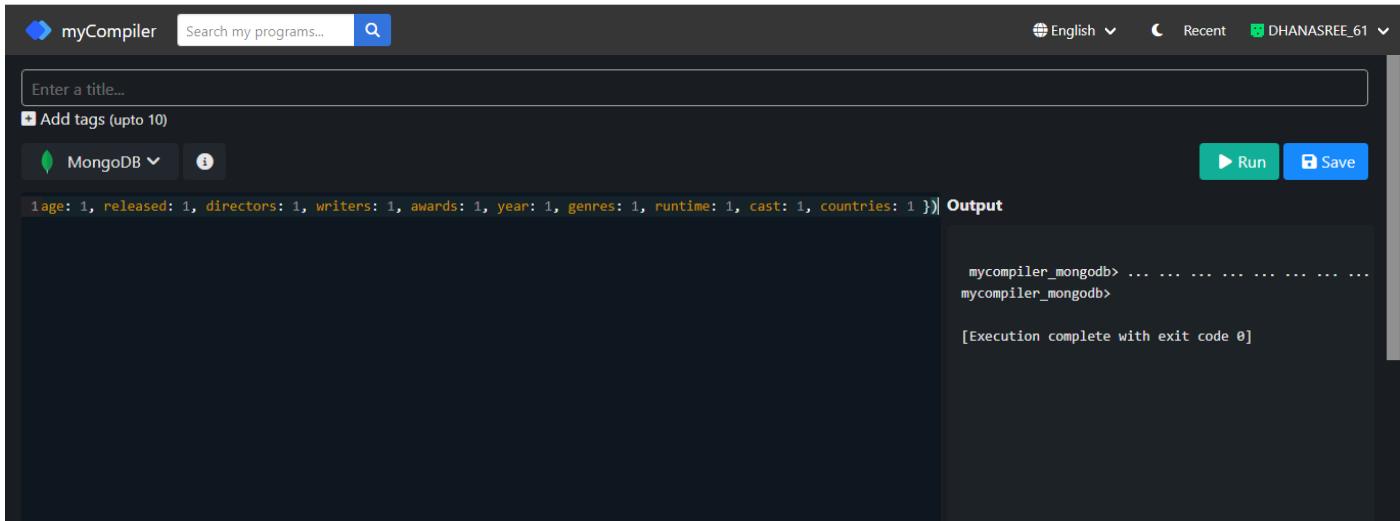
The screenshot shows the 'myCompiler' interface with a dark theme. At the top, there's a search bar labeled 'Search my programs...' and a 'Run' button. Below the search bar is a text input field with placeholder text 'Enter a title...'. Underneath the input field is a button labeled 'Add tags (upto 10)'. On the left side, there's a dropdown menu set to 'MongoDB' and a help icon. On the right side, there are 'Run' and 'Save' buttons. The main area is titled 'Output' and contains the following text:
`1 age: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }`
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

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

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. At the top, there's a search bar for programs and a language selection dropdown set to English. On the left, there's a text input field for titles and a checkbox for adding tags. Below that is a MongoDB dropdown menu. The main area contains a command line interface with the following text:

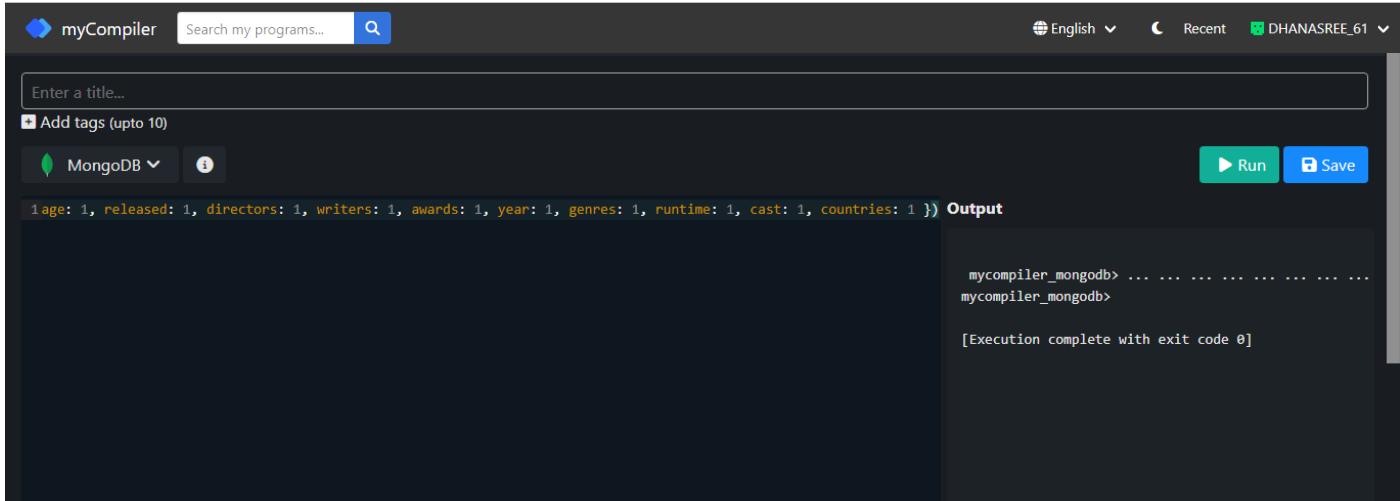
```
1 age: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }]} Output  
mycompiler_mongodb> ... . . . . .  
mycompiler_mongodb>  
[Execution complete with exit code 0]
```

4.)Retrieve all movies from the 'movies' collection that were directed by "William K.L. Dickson" and include complete information for each movie.

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. At the top, there's a search bar for programs and a language selection dropdown set to English. On the left, there's a text input field for titles and a checkbox for adding tags. Below that is a MongoDB dropdown menu. The main area contains a command line interface with the following text:

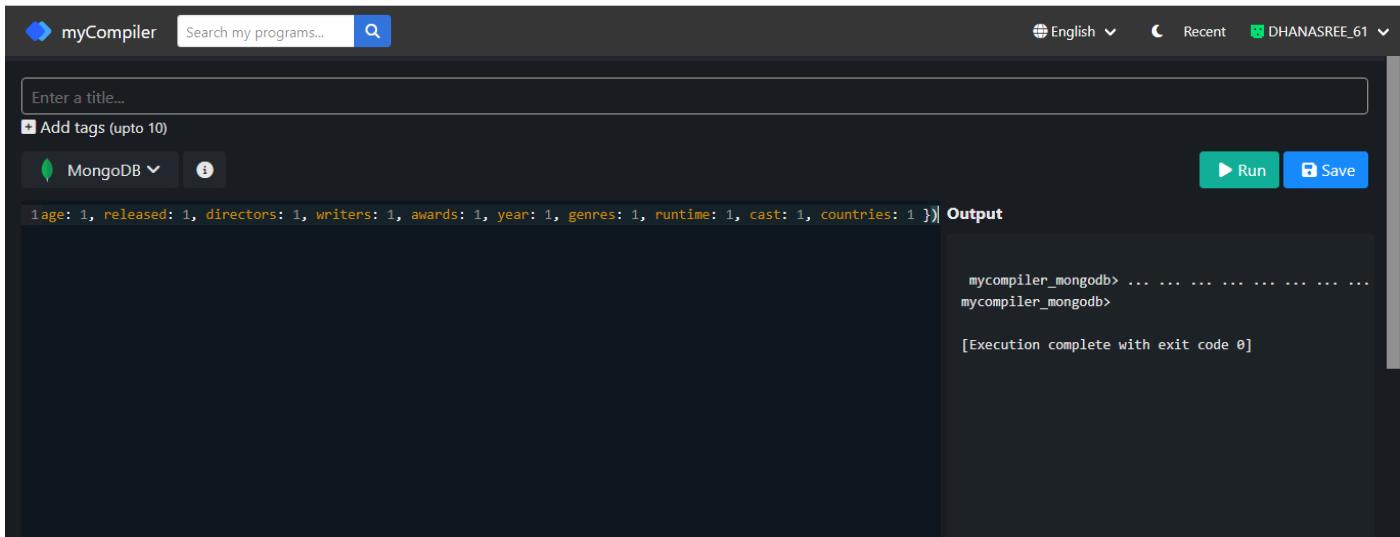
```
1 age: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }]} Output  
mycompiler_mongodb> ... . . . . .  
mycompiler_mongodb>  
[Execution complete with exit code 0]
```

5.) Retrieve all movies from the 'movies' collection that were released in the USA and include complete information for each movie.

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. At the top, there's a search bar for programs and a language dropdown set to English. Below the search bar, there's a text input field labeled "Enter a title..." and a checkbox for "Add tags (upto 10)". On the left, there's a MongoDB dropdown and a help icon. On the right, there are "Run" and "Save" buttons. The main area is titled "Output" and contains the following text:
`age: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }`| Output

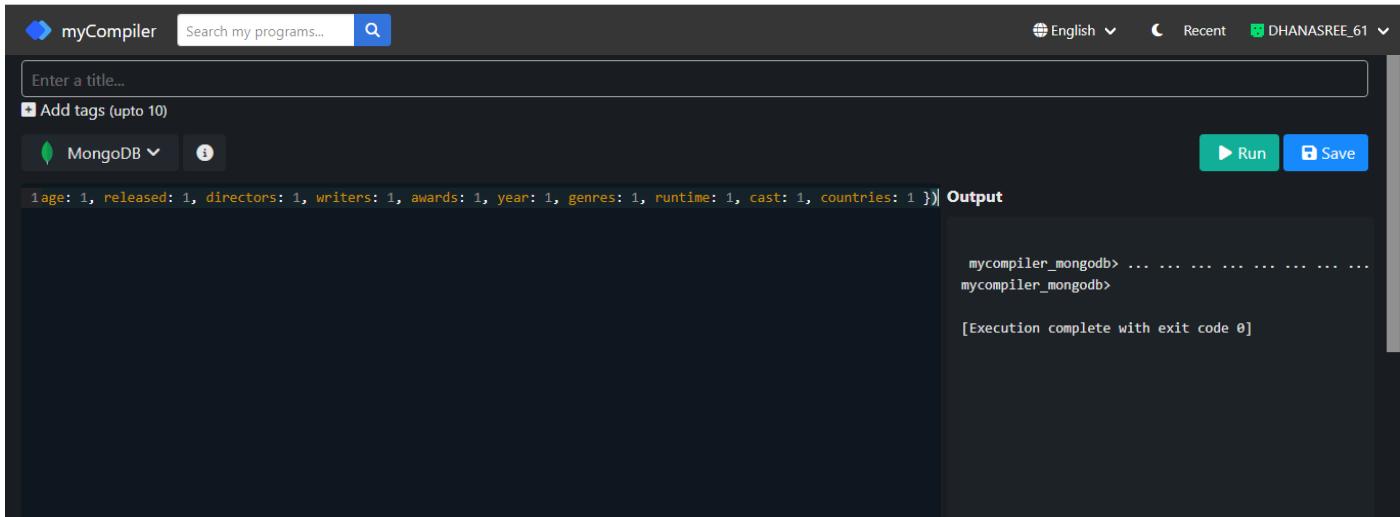
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

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

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. At the top, there's a search bar for programs and a language dropdown set to English. Below the search bar, there's a text input field labeled "Enter a title..." and a checkbox for "Add tags (upto 10)". On the left, there's a MongoDB dropdown and a help icon. On the right, there are "Run" and "Save" buttons. The main area is titled "Output" and contains the following text:
`age: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 }`| Output

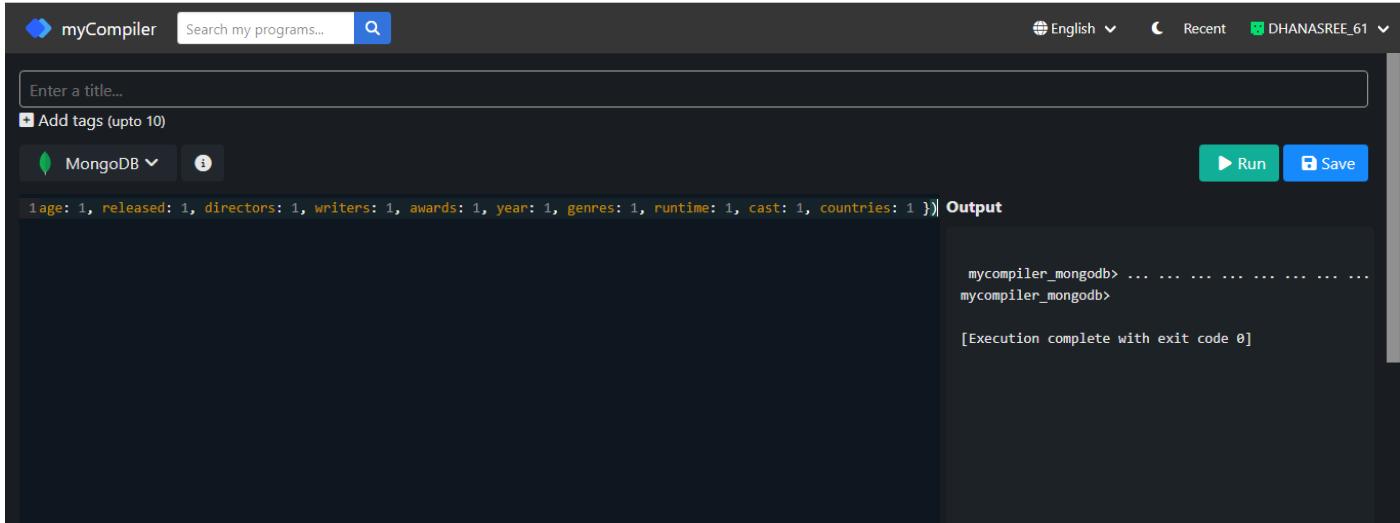
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

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:



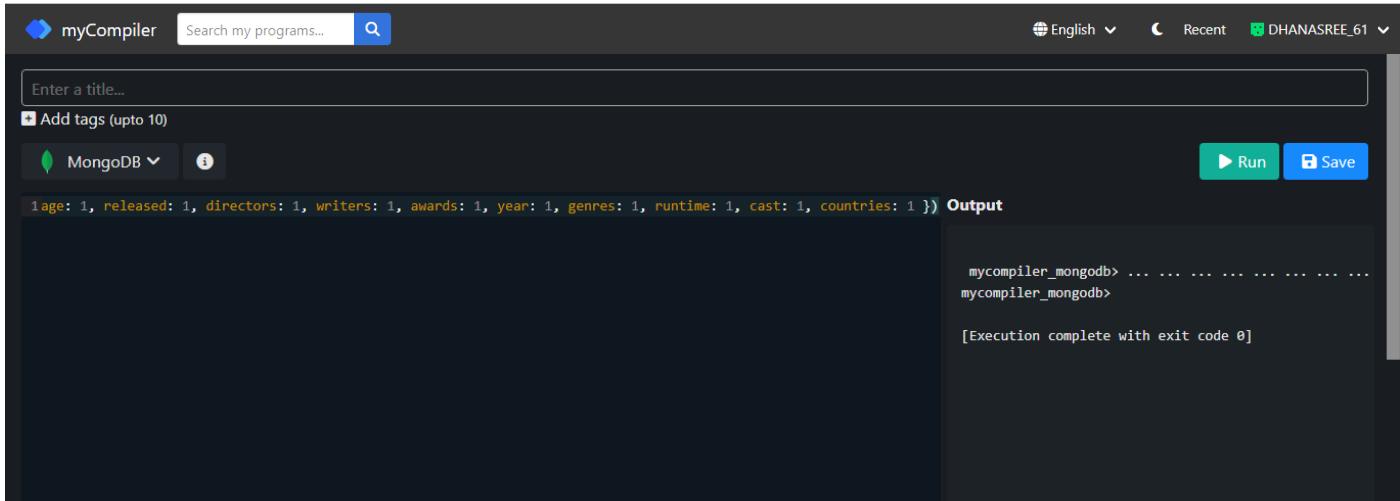
The screenshot shows the myCompiler interface. In the top bar, there's a logo, a search bar with 'Search my programs...', and a user profile 'DHANASREE_61'. Below the search bar is a text input field labeled 'Enter a title...' and a button '+ Add tags (upto 10)'. On the left, there's a MongoDB dropdown and a help icon. On the right, there are 'Run' and 'Save' buttons. The main area has a title 'Output' and contains the command: 'db.movies.find({ "imdb.votes": { "\$gt": 1000 } })'. The output window shows the command being run: 'mycompiler_mongodb> db.movies.find({ "imdb.votes": { "\$gt": 1000 } })'. It then displays the results of the query, which are empty. The message '[Execution complete with exit code 0]' is at the bottom of the output window.

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:



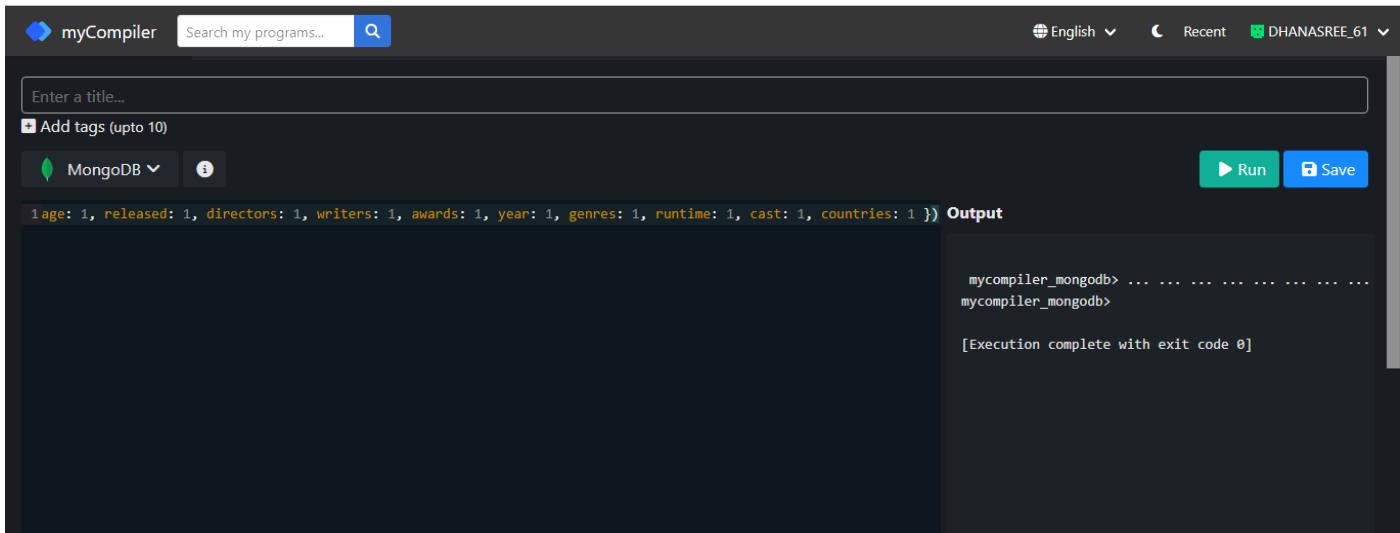
The screenshot shows the myCompiler interface. The setup is identical to the previous one: logo, search bar, user profile 'DHANASREE_61'. The text input field 'Enter a title...' and '+ Add tags (upto 10)' button are present. On the left, there's a MongoDB dropdown and a help icon. On the right, there are 'Run' and 'Save' buttons. The main area has a title 'Output' and contains the command: 'db.movies.find({ "imdb.rating": { "\$gt": 7 } })'. The output window shows the command being run: 'mycompiler_mongodb> db.movies.find({ "imdb.rating": { "\$gt": 7 } })'. It then displays the results of the query, which are empty. The message '[Execution complete with exit code 0]' is at the bottom of the output window.

9.) Retrieve all movies from the 'movies' collection that have complete information and have a viewer rating higher than 4 on Tomatoes.

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. At the top, there's a search bar for programs and a language dropdown set to English. Below the search bar is a text input field labeled "Enter a title...". Underneath it is a button for "Add tags (upto 10)". On the left, there's a MongoDB dropdown and a help icon. On the right, there are "Run" and "Save" buttons. The main area contains a code editor with the following MongoDB query:

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

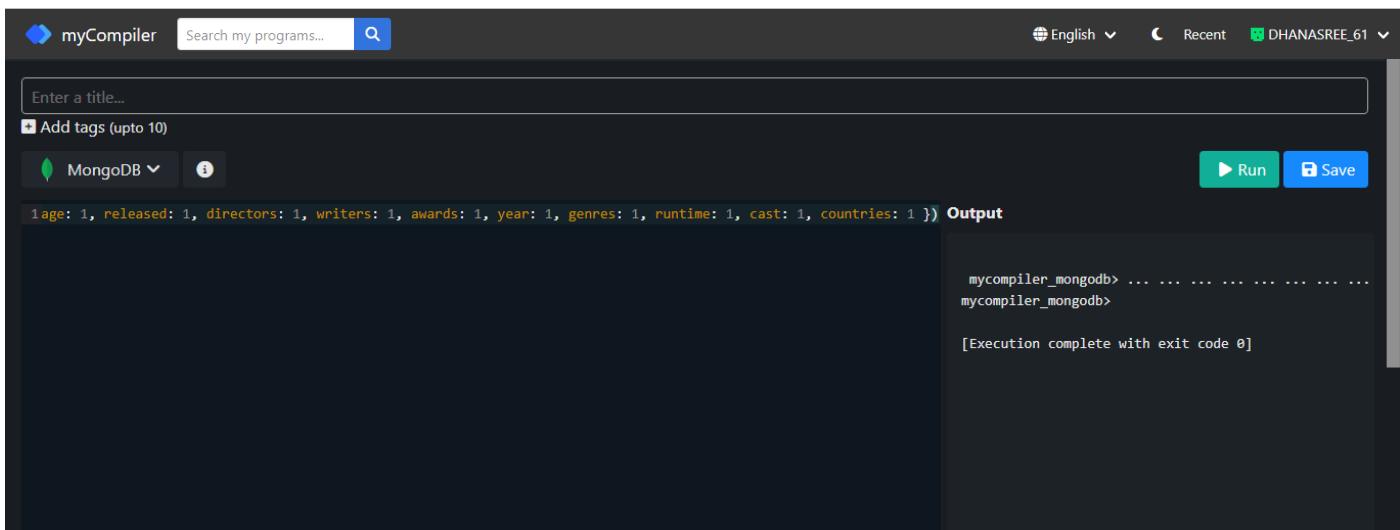
Below the code editor is a "Output" section. It shows the command "mycompiler_mongodb>" followed by the response "mycompiler_mongodb>". At the bottom of the output section, it says "[Execution complete with exit code 0]".

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

QUERY:

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

OUTPUT:



The screenshot shows the myCompiler interface. The layout is identical to the previous one, with a search bar, English language selection, and a code editor. The code editor contains the same MongoDB query as the previous screenshot:

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

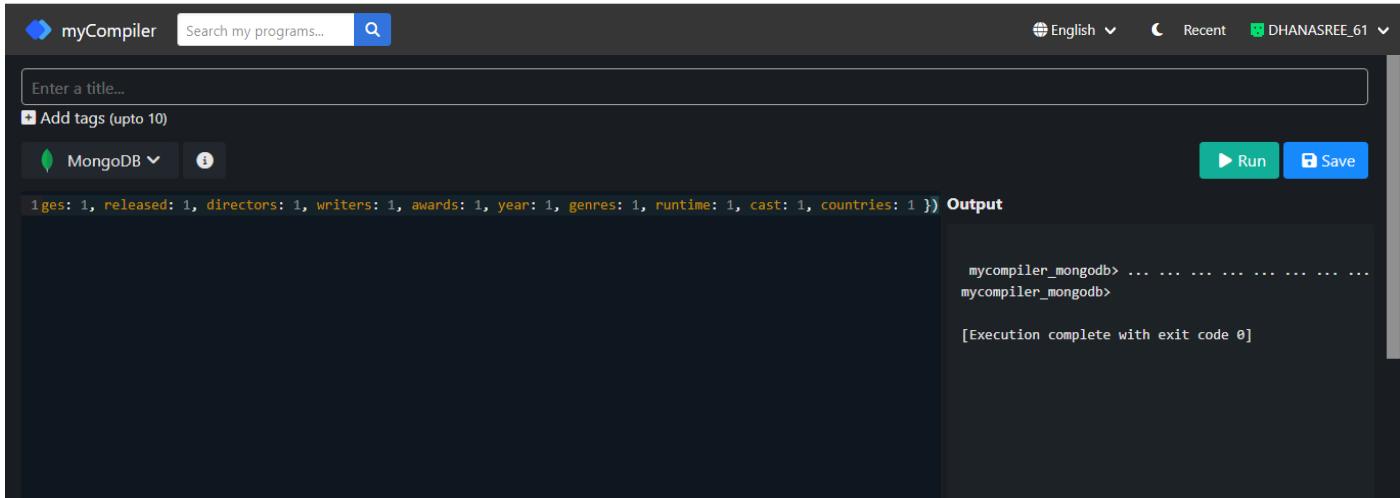
The "Output" section shows the command "mycompiler_mongodb>" followed by "mycompiler_mongodb>". At the bottom, it says "[Execution complete with exit code 0]".

11.)Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB that have at least one nomination.

QUERY:

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

OUTPUT:



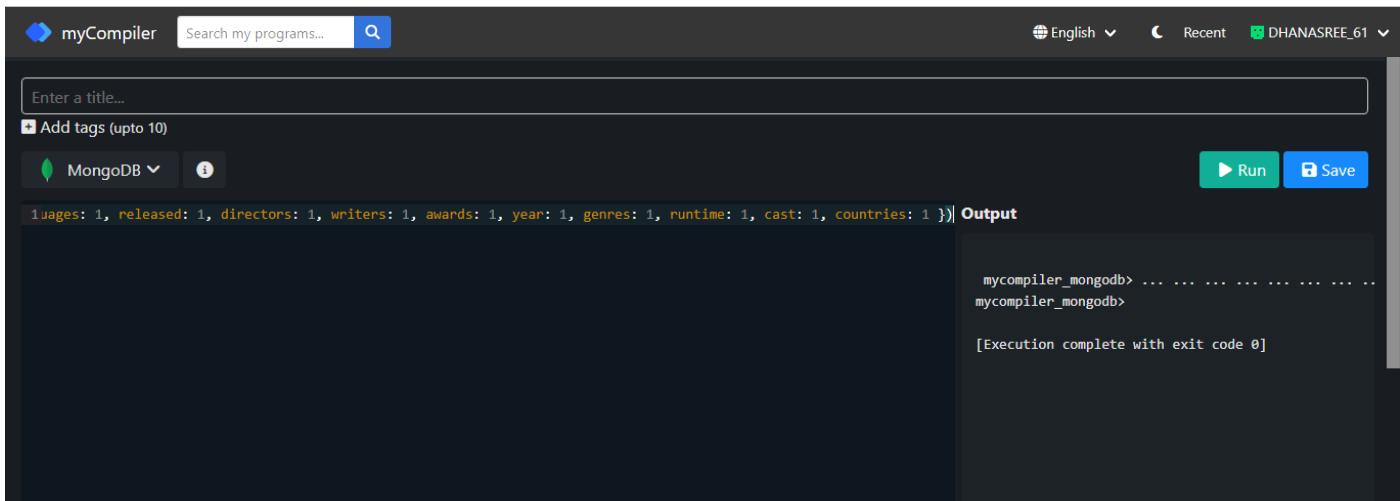
The screenshot shows the 'myCompiler' interface. In the top bar, there are tabs for 'myCompiler', 'Search my programs...', and a magnifying glass icon. On the right, there are language selection dropdowns ('English'), a 'Recent' section, and a user profile ('DHANASREE_61'). Below the header is a search bar with placeholder text 'Enter a title...' and a 'Add tags (upto 10)' button. Underneath the search bar are two buttons: 'MongoDB' with a dropdown arrow and an info icon. To the right are 'Run' and 'Save' buttons. The main area is titled 'Output' and contains the MongoDB command: '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 })'. The output window shows the command being run and the response: 'mycompiler_mongodb> ...'. At the bottom of the output window, it says '[Execution complete with exit code 0]'. The background of the interface is dark grey.

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:



The screenshot shows the 'myCompiler' interface. The layout is identical to the previous screenshot, with tabs for 'myCompiler', 'Search my programs...', and a magnifying glass icon. The header also includes language selection ('English'), a 'Recent' section, and a user profile ('DHANASREE_61'). The search bar and 'Add tags' button are present. The 'MongoDB' button has a dropdown arrow. The 'Run' and 'Save' buttons are on the right. The 'Output' window contains the MongoDB command: '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 })'. The output window shows the command being run and the response: 'mycompiler_mongodb> ...'. At the bottom of the output window, it says '[Execution complete with exit code 0]'. The background is dark grey.

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

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler interface with a dark theme. At the top, there's a search bar with placeholder text "Search my programs..." and a magnifying glass icon. On the right, there are language selection ("English"), recent projects ("Recent"), and user profile ("DHANASREE_61") buttons.

The main workspace has a title input field ("Enter a title...") and a "Add tags (upto 10)" button. Below this, there are two dropdown menus: "MongoDB" and "i". To the right are "Run" and "Save" buttons.

The code editor contains a single line of MongoDB query:

```
1sed.date": "May 9, 1893"}, { _id: 0, title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 }]
```

The "Output" panel shows the command prompt "mycompiler_mongodb> ..." followed by "[Execution complete with exit code 0]".

14.) Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that have a word "scene" in the title.

QUERY:

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

OUTPUT:

The screenshot shows the myCompiler interface. At the top, there's a search bar with the placeholder "Search my programs..." and a magnifying glass icon. To the right are language selection ("English"), recent projects ("Recent"), and user profile ("DHANASREE_61"). Below the search bar is a text input field with the placeholder "Enter a title...". Underneath it is a button labeled "Add tags (upto 10)" with a checkmark icon. On the left, there's a dropdown menu for "MongoDB" with a green leaf icon and a dropdown arrow, and a help icon (info symbol). On the right, there are two buttons: "Run" (green with white play icon) and "Save" (blue with white save icon). The main area contains a code editor with the following MongoDB query:

```
1s.find({ title: /scene/i }, { _id: 0, title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })
```

To the right of the code editor is a "Output" section. It shows the command prompt "mycompiler_mongodb> ..." followed by a series of dots, indicating a long list of results. Below the prompt is the message "[Execution complete with exit code 0]".

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

RESULT: