

Creating and Managing Tables

EX_NO:1

DATE:

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Commands interface. The SQL tab is selected, displaying the command: `create table dept(id int,name varchar(25));`. The Run button is visible at the top right. Below the command, the results show the message: `table created.` and `0.05 seconds`. The bottom status bar indicates the user is connected to `aadhitya02@rajalakshmi.edu.in` and the session ID is `aadhitya02`.

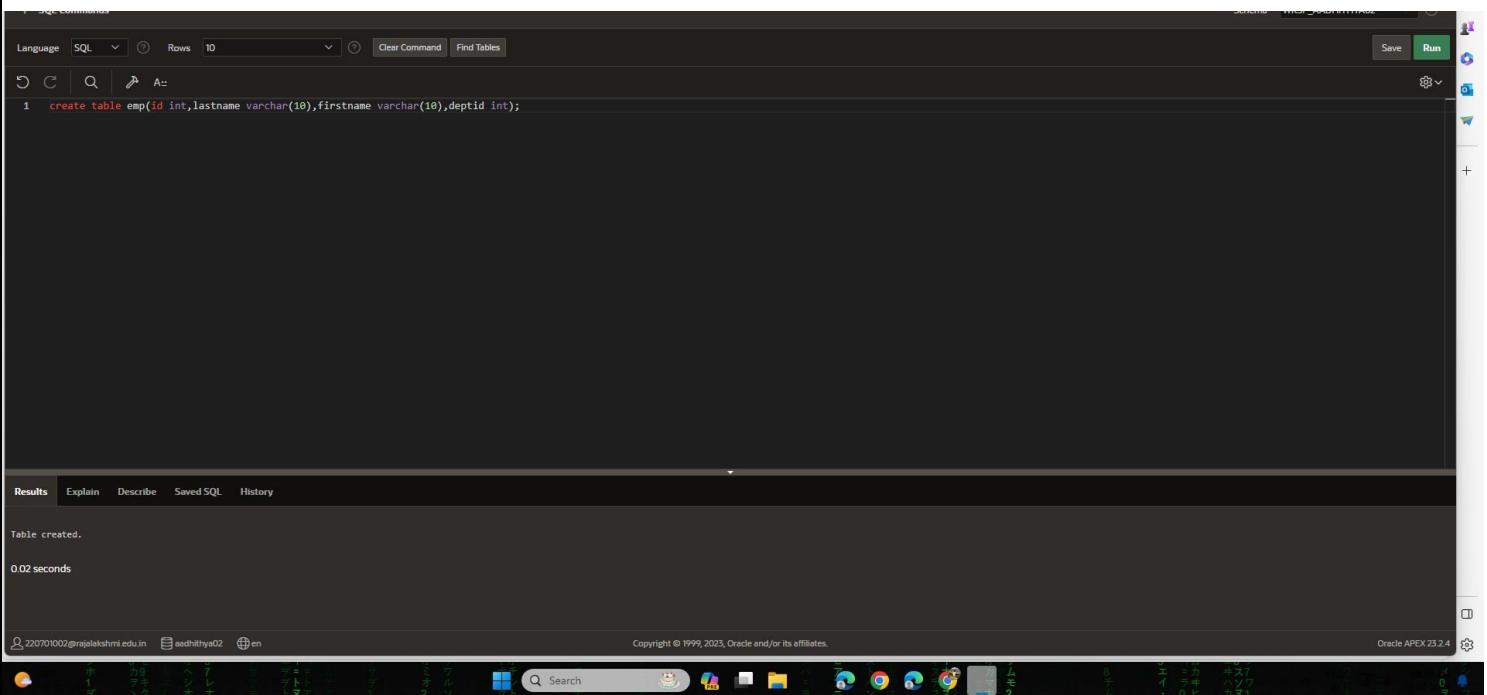
2. Create the EMP table based on the following instance chart. Confirm that the table is created.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a toolbar with various icons. Below it is a search bar and a dropdown menu. The main area is a large text editor where the SQL command to create the table has been entered. The command is:

```
create table emp(id int,lastname varchar(10),firstname varchar(10),deptid int);
```

After running the command, the results pane at the bottom shows the message "Table created." and a timestamp of "0.02 seconds". The status bar at the bottom right indicates the session is connected to "Oracle APEX 25.2.4".

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a toolbar with icons for Undo, Redo, Find, and Replace. The main area is a dark-themed SQL editor window. At the top of the editor, it says "SQL Commands". Below that, there are dropdown menus for "Language" (set to "SQL"), "Rows" (set to 10), and "Schema" (set to "WKSP_AADHITHYA02"). On the right side of the editor, there are buttons for "Save" and "Run". The SQL command entered is:

```
1 | alter table emp modify(lastname varchar(50));
2 |
```

Below the editor, there's a "Results" tab which is currently selected. It displays the output of the command:

```
Table altered.
```

At the bottom of the screen, there's a taskbar with various application icons. The system tray on the right shows the date and time as "Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4".

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

QUERY:

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

OUTPUT:

5. Drop the EMP table.

QUERY:

```
Drop table 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. The SQL Workshop tab is selected. The schema dropdown is set to 'WKSP_AADHITHYA02'. The main area displays the following SQL command:

```
drop table emp;
```

The results section at the bottom shows the output of the command:

Owner	Name	Description	SQL	Updated By	Updated
220701002@RAJALAKSHMI.EDU.IN	dbms ex15	-	drop table emp;	220701002@RAJALAKSHMI.EDU.IN	1 seconds ago

At the bottom of the screen, the Windows taskbar is visible with various icons for applications like File Explorer, Microsoft Edge, and others.

6. Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename employees2 to emp;

OUTPUT:

The screenshot shows the Oracle APEX SQL Commands interface. At the top, there's a toolbar with various icons and buttons like 'Save' and 'Run'. The main area contains a SQL command: '1 rename employees2 to emp;'. Below the command, the results section shows the output: 'Statement processed.' and '0.05 seconds'. The bottom part of the interface shows a standard Windows taskbar with icons for various applications like File Explorer, Edge, and Google Chrome.

```
↑ SQL Commands
Schema: WKSP_AADHITHYA02
Language: SQL Rows: 10 Clear Command Find Tables Save Run
1 rename employees2 to emp;

Statement processed.
0.05 seconds
```

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

QUERY:

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

OUTPUT:

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_AADHITHYA02'. The SQL Commands tab is active, showing the following command:

```
1 alter table emp drop column firstname;
```

Below the command, the results table shows one row of data:

Owner	Name	Description	SQL	Updated By	Updated
220701002@RAJALAKSHMI.EDU.IN	dbms_ext.8	-	alter table emp drop column firstname;	220701002@RAJALAKSHMI.EDU.IN	1 seconds ago

The status bar at the bottom indicates the session is connected to '220701002@rajalakshmi.edu.in' and the user is 'aadhithya02'. The system language is set to Japanese ('日本語').

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

QUERY:

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

OUTPUT:

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

RESULT:

MANIPULATING DATA

EX_NO:2

DATE:

1. Create MY_EMPLOYEE table with the following structure

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

QUERY:

```
Create table my_emp(id int,last_name varchar(25),first_name varchar(25),userid  
varchar(25),salary int);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the user 'aadhittha rk' and the schema 'WKS_AADHITHYA02'. The main area has tabs for 'SQL Commands' (selected), 'Language' (SQL), 'Rows' (10), 'Clear Command', and 'Find Tables'. Below this is a code editor with two lines of SQL: 'create table my_emp(id int,last_name varchar(25),first_name varchar(25),userid varchar(25),salary int);'. At the bottom, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The results tab shows a table with columns 'Owner', 'Name', 'Description', 'SQL', 'Updated By', and 'Updated'. One row is listed: 'Owner: 220701002@RAJALAKSHMLEDUIN, Name: exp21, Description: -, SQL: create table my_emp(id int,last_name varchar(25),first_name varchar(25),userid varchar(25),salary int);, Updated By: 220701002@RAJALAKSHMLEDUIN, Updated: Now'. The status bar at the bottom indicates the session is connected to '220701002@rajalakshmi.edu.in' and the schema is 'aadhittha02'. The system also shows 'en' and 'Copyright © 1999, 2023, Oracle and/or its affiliates.'

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

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

QUERY:

```
Insert into my_emp values(1,'patel','ralph','rpatel',895);
```

```
Insert into my_emp values(2,'dancs','betty','bdancs',860);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, two INSERT statements are executed:

```
1  insert into my_emp values(1,'patel','ralph','rpatel',895);
2  insert into my_emp values(2,'dancs','betty','bdancs',860);
```

In the Results tab, the output shows the inserted data:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	patel	ralph	rpatel	1000
2	dancs	betty	bdancs	860

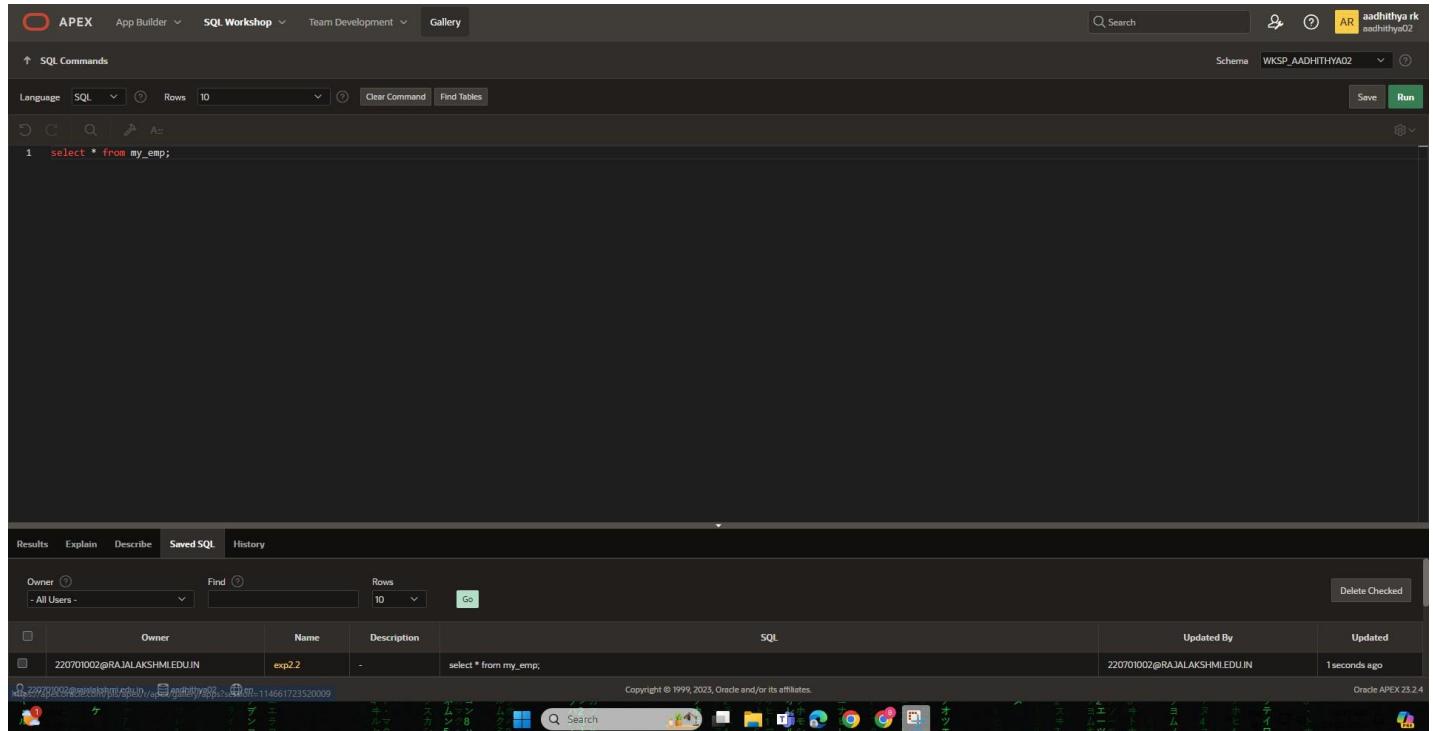
Below the results, it says "2 rows returned in 0.00 seconds". The status bar at the bottom shows system information including weather (Partly sunny, 32°C), date (04-03-2024), and time (12:18).

3. Display the table with values.

QUERY:

```
Select * from my_emp;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. On the right side, there are search, help, and user profile icons. The main area has a toolbar with various icons. Below the toolbar, the SQL command is displayed: "select * from my_emp;". The results section shows a single row of data with the following details:

Owner	Name	Description	SQL	Updated By	Updated
220701002@RAJALAKSHMILEDU.IN	exp2.2	-	select * from my_emp;	220701002@RAJALAKSHMILEDU.IN	1 seconds ago

At the bottom of the screen, the Windows taskbar is visible with various pinned icons like File Explorer, Edge, and File Manager.

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

QUERY:

```
Insert into my_emp values(3,'biri','ben','bbiri',1100);
```

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 schema is set to 'WKSP_AADHITHYA02'. The SQL command entered is:

```
1 insert into my_emp values(3,'biri','ben','bbiri',1100);
2
```

The results section shows the output:

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

Execution time is listed as 0.01 seconds. The bottom status bar indicates the session is connected to '220701002@rajalakshmi.edu.in' and the schema is 'aadhithya02'. The status bar also shows the operating system taskbar with various icons.

5. Make the data additions permanent.

QUERY:

Select * from my_emp;

OUTPUT:

```
1 select * from my_emp;
```

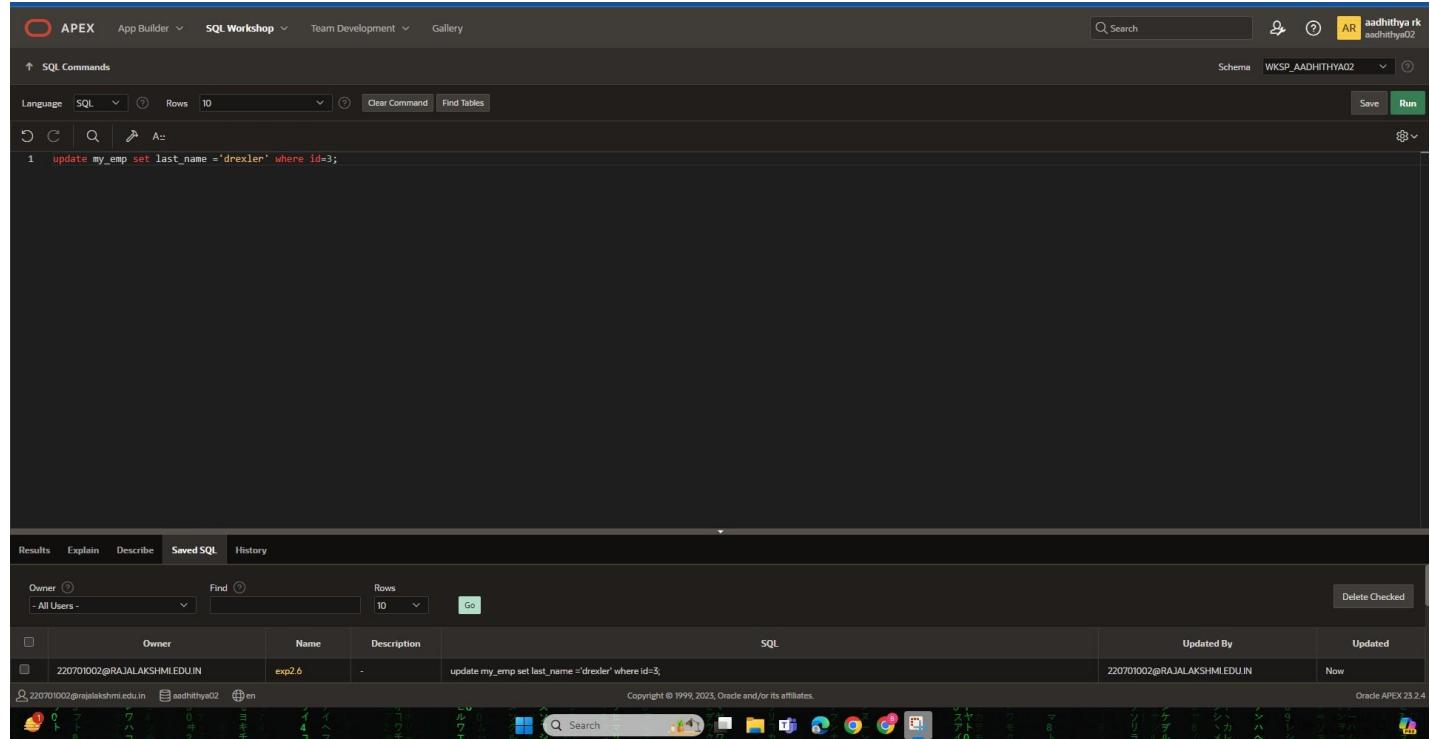
ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	patel	ralph	rpatel	895
2	dancs	betty	bdancs	860
3	biri	ben	bbiri	1100

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

QUERY:

```
Update my_emp set last_name='drexler' where id=3;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. At the top, there are tabs for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'aadhihya rk', and a schema dropdown for 'WKSP_AADHITHYA02'. Below the tabs, there's a toolbar with icons for Undo, Redo, Find, and Save/Run. The main area is titled 'SQL Commands' and contains a text input field with the SQL command: 'update my_emp set last_name = 'drexler' where id=3;'. The results section shows a table with one row affected. The table has columns: Owner, Name, Description, SQL, Updated By, and Updated. The row details are: Owner '220701002@RAJALAKSHMI.EDU.IN', Name 'exp2.6', Description '...', SQL 'update my_emp set last_name = 'drexler' where id=3;', Updated By '220701002@RAJALAKSHMI.EDU.IN', and Updated 'Now'. The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Edge, and Google Chrome.

Owner	Name	Description	SQL	Updated By	Updated
220701002@RAJALAKSHMI.EDU.IN	exp2.6	-	update my_emp set last_name = 'drexler' where id=3;	220701002@RAJALAKSHMI.EDU.IN	Now

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

QUERY:

```
Update my_emp set salary=1000 where salary<900;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Commands interface. In the top navigation bar, 'SQL Commands' is selected. The schema is set to 'WKSP_AADITYA03'. The SQL command entered is:

```
1 update myemp set Salary=1000 where Salary<900
```

In the results section, it shows:

3 row(s) updated.
0.01 seconds

At the bottom, the footer includes:

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

8. Delete Betty dancs from MY_EMPLOYEE table.

QUERY:

Delete from my_emp where first_name='betty';

OUTPUT:

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

```
1 delete from my_emp where first_name='betty';
```

Below the command, the results table displays the executed query:

Owner	Name	Description	SQL	Updated By	Updated
220701002@RAJALAKSHMI.EDU.IN	exp2.8	-	delete from my_emp where first_name='betty';	220701002@RAJALAKSHMI.EDU.IN	1 seconds ago

The status bar at the bottom indicates the session ID is 220701002@rajalakshmi.edu.in and the schema is exp2.8. The system also shows the Oracle APEX version 23.2.4.

9. Empty the fourth row of the emp table.

QUERY:

Delete from my_emp where first_name='chad';

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, it shows the schema 'WKS_AADHITHYA02' and a user icon 'aadhithya rk'. Below the navigation is a toolbar with 'SQL Commands', 'Language' dropdown (set to 'SQL'), 'Rows' dropdown (set to '10'), 'Clear Command', 'Find Tables', 'Save' button, and a 'Run' button.

In the main area, there's a code editor with the following SQL command:

```
1 delete from my_emp where first_name='chad';
```

Below the code editor is a results grid. The grid has columns: 'Owner', 'Name', 'Description', 'SQL', 'Updated By', and 'Updated'. There is one row in the grid:

Owner	Name	Description	SQL	Updated By	Updated
220701002@RAJALAKSHMI.EDU.IN	exp29	-	delete from my_emp where first_name='chad';	220701002@RAJALAKSHMI.EDU.IN	1 seconds ago

At the bottom of the window, there's a status bar with the text 'Copyright © 1999, 2023, Oracle and/or its affiliates' and 'Oracle APEX 25.2.4'. The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Microsoft Edge, and File Manager.

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

RESULT:

INCLUDING CONSTRAINTS

EX_NO:3

DATE:

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

QUERY:

```
Create table emp2(id int);
alter table emp2 add constraint my_emp_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 is titled 'SQL Commands'. The code entered is:

```
1 create table emp2(id int);
2 alter table emp2 add constraint my_emp_id_pk primary key(id);
3
```

The 'Run' button is highlighted in green. Below the code, the results show:

Table altered.
0.06 seconds

At the bottom, the status bar indicates the user is 220701002@rajalakshmi.edu.in, connected to schema aadhistya02, and the system time is 09:18 AM on 27-02-2024. The Oracle APEX version shown is 23.2.4.

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

QUERY:

```
Create table dept02(my_dept_id_pk int primary key);
```

OUTPUT:

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

```
1 create table dept02(my_dept_id_pk int primary key);
```

Below the command, the results are displayed in the 'Results' tab:

```
Table created.  
0.04 seconds
```

The status bar at the bottom shows the user's email (220701002@rajalakshmi.edu.in), session name (aadithya02), and language (en). The copyright notice reads "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version is "Oracle APEX 23.2.4".

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

QUERY:

```
Alter table employee add constraint my_emp_dept_id_fk foreign key(dept_id)references employee(id);
```

OUTPUT:

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

```
1 alter table employee add constraint my_emp_dept1_id_fk foreign key(dept1_id)references employee(id);
```

Below the command, the results section displays the output:

```
Table altered.  
0.05 seconds
```

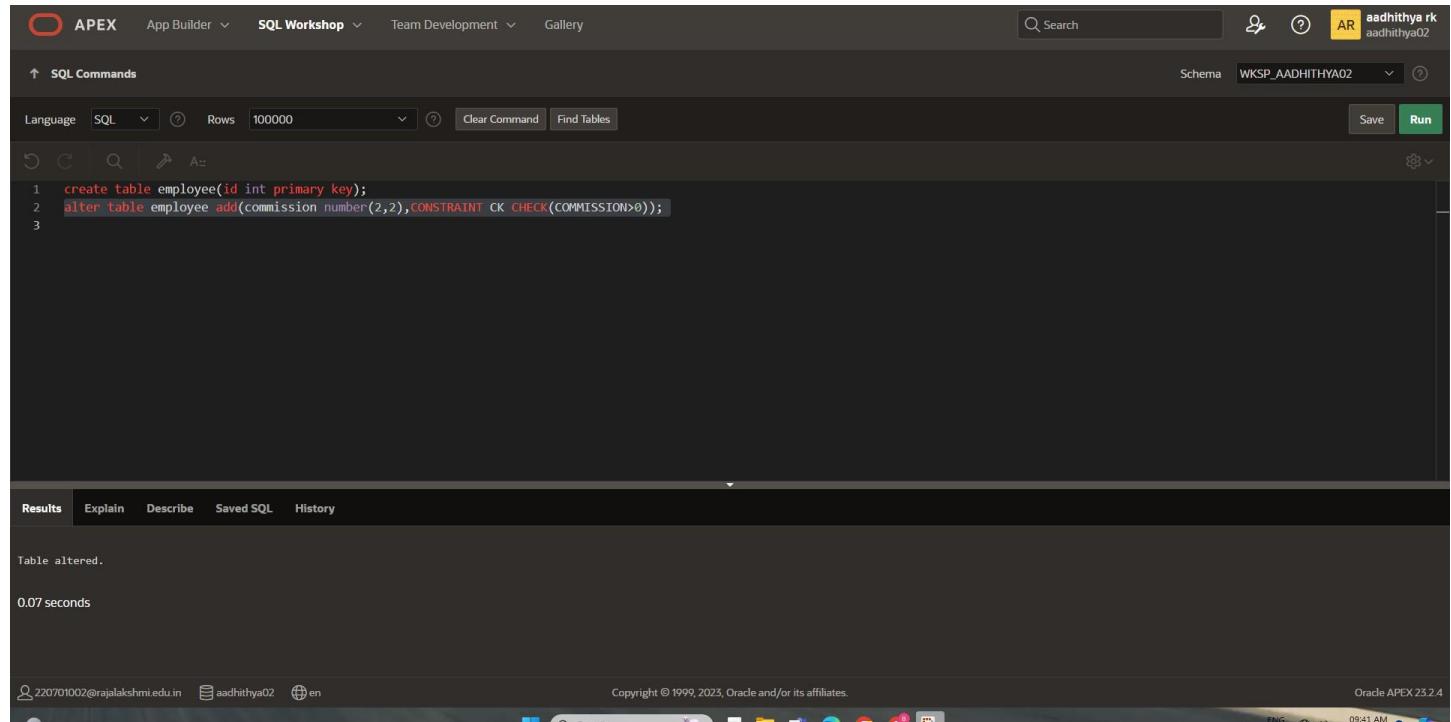
The status bar at the bottom indicates the session information: 220701002@rajalakshmi.edu.in, aadhihya02, and the system date/time: 01-05-2024 11:33 AM. The Oracle APEX version shown is 23.2.4.

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

QUERY:

```
Create table employee(id int primary key);
Alter table employee add(commission number(2,2) constraint ck check(commission>0));
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, tabs for 'App Builder', 'SQL Workshop' (which is active), 'Team Development', and 'Gallery' are visible. On the right side, there's a search bar, a help icon, and a user profile 'aadhiytha rk aadhiytha02'. The main area is titled 'SQL Commands' with a sub-section '↑ SQL Commands'. It includes dropdowns for 'Language' (set to SQL), 'Rows' (set to 100000), and buttons for 'Clear Command' and 'Find Tables'. On the far right are 'Save' and 'Run' buttons. The SQL command entered is:

```
1 create table employee(id int primary key);
2 alter table employee add(commission number(2,2),CONSTRAINT CK CHECK(COMMISSION>0));
3
```

Below the command area, a 'Results' tab is selected. The output shows the message 'Table altered.' and a execution time of '0.07 seconds'. At the bottom of the page, there are footer links for 'Copyright © 1999, 2023, Oracle and/or its affiliates.', the URL '220701002@rajalakshmi.edu.in', the session ID 'aadhiytha02', and the language 'en'. The bottom right corner shows the system status '09:41 AM'.

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, "salary" * 12 "annual_salary" from employee;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'aadhittha rk', and a schema dropdown set to 'WKSP_AADHITHYA02'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select employee_id, last_name, "SALARY" *12 "ANNUAL_SALARY" from employee;
```

Below the code, the results section displays a single row of data:

EMPLOYEE_ID	LAST_NAME	ANNUAL_SALARY
1	kumar	180000

Text at the bottom left indicates '1 rows returned in 0.02 seconds' and a 'Download' link. The bottom right corner shows the Oracle APEX version '23.2.4'.

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

QUERY:

Select * from employee;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, the command `SELECT * FROM EMPLOYEE;` is entered. The Results tab displays the following data:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID	ANNUAL_SALARY
1	raj	kumar	abc@gmail.com	9171693361	12/10/2004	35	15000	12	1	100000

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

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

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 as employee_number, last_name, job_id, hire_date from employee;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user 'aadhisthya rk' and the schema 'WKSP_AADHITHYA02'. The main workspace has tabs for SQL Commands, SQL, Rows (set to 10), Clear Command, and Find Tables. Below these are icons for Undo, Redo, Search, and Copy. The SQL command entered is:

```
1  SELECT EMPLOYEE_ID AS EMPLOYEE_NUMBER, LAST_NAME, JOB_ID, HIRE_DATE FROM EMPLOYEE;
```

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

EMPLOYEE_NUMBER	LAST_NAME	JOB_ID	HIRE_DATE
1	kumar	35	12/10/2004

Below the table, it says '1 rows returned in 0.00 seconds' and there is a 'Download' link. The bottom status bar shows the user's email (220701002@rajalakshmi.edu.in), session ID (aadhisthya02), and language (en). It also displays the operating system's taskbar with various pinned icons like File Explorer, Edge, and File History. The system tray shows the date (04-03-2024), time (17:43), battery level, and network connection.

4. Provide an alias STARTDATE for the hire date.

QUERY:

Select hire_date as startdate from employee;

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 'aadithya rk aadithya02', and a schema dropdown set to 'WKSP_AADHITHYA02'. Below the tabs, there's a toolbar with Language (SQL selected), Rows (10), Clear Command, Find Tables, Save, and Run buttons. The main area contains the SQL command: 'SELECT HIRE_DATE AS START_DATE FROM EMPLOYEE;'. The results section shows a single row with the value '12/10/2004' under the column 'START_DATE'. Below the results, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Edge, and Google Chrome, along with system status icons for battery, signal, and volume.

START_DATE
12/10/2004

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

QUERY:

Select distinct job_id from employee;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side of the header shows the schema (WKSP_AADHITHYA02) and a user icon (aadhithya02). The main workspace is titled "SQL Commands" and contains the following command:

```
1 SELECT DISTINCT JOB_ID FROM EMPLOYEE;
```

Below the command, the results tab is selected, showing a single row of data:

JOB_ID
35

Text below the table indicates "1 rows returned in 0.00 seconds" and a "Download" link. The bottom of the screen shows the Windows taskbar with various pinned icons and system status information.

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. 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 'aadhisthya02', and a schema dropdown set to 'WKSP_AADHITHYA02'. Below the tabs, there are buttons for Language (SQL selected), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The main area contains the SQL command:

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

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

EMPLOYEE_AND_TITLE
kumar,35

Below the table, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The bottom of the page shows browser status bars with weather information (30°C, Partly sunny), system icons (Search, File Explorer, etc.), and system status (ENG IN, 17:46, 04-03-2024).

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_number||','||hire_date||',
'||job_id||'||salary||','||manager_id||','||','||department_id as "the_output" 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 for 'aadithya rk', and a schema dropdown set to 'WKSP_AADITHYA02'. Below the navigation is a toolbar with Language (SQL selected), Rows (set to 10), Clear Command, Find Tables, Save, and Run buttons. The main area contains the SQL command entered in the previous step. The results section shows the output of the query, which is a single row with the column name 'THE_OUTPUT' and the value '1,raj,kumar,abc@gmail.com,9171693361,12/10/2004,35,15000,12,1'. Below the results, it says '1 rows returned in 0.02 seconds' and provides a 'Download' link. The bottom of the page includes standard footer links for Oracle APEX 25.2.4.

```
1   SELECT EMPLOYEE_ID||','||FIRST_NAME||','||LAST_NAME||','||EMAIL||','||PHONE_NUMBER||','||HIRE_DATE||','||JOB_ID||'||SALARY||','||MANAGER_ID||','||DEPARTMENT_ID AS "THE_OUTPUT" FROM EMPLOYEES
```

THE_OUTPUT
1,raj,kumar,abc@gmail.com,9171693361,12/10/2004,35,15000,12,1

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

RESULT:

RESTRICTING AND SORTING DATA

EX_NO:5

DATE:

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

QUERY:

```
select last_name,salary from EMPLOYEE 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'. The right side shows a user profile for 'aadithya rk' with schema 'WKSP_AADITHYA02'. The main area has tabs for 'SQL Commands' and 'Results'. In the 'SQL Commands' tab, the query 'select last_name,salary from employee where salary >12000;' is entered. The 'Results' tab displays the output in a table:

LAST_NAME	SALARY
raj	17000
kumar	15000
raj	18000
mohan	16000

Below the table, it says '4 rows returned in 0.00 seconds'. The bottom status bar shows the URL '220701002@rajalakshmi.edu.in', session 'aadithya02', and browser icons. The system status bar indicates '28°C Haze' and the date '08-03-2024'.

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

QUERY:

```
select last_name,department_number from EMPLOYEE where emp_id=176;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface, similar to the previous one. The top navigation bar and user profile are identical. The main area has tabs for 'SQL Commands' and 'Results'. In the 'SQL Commands' tab, the query 'select last_name,department_id from employee where employee_id=176;' is entered. The 'Results' tab displays the output in a table:

LAST_NAME	DEPARTMENT_ID
raj	34

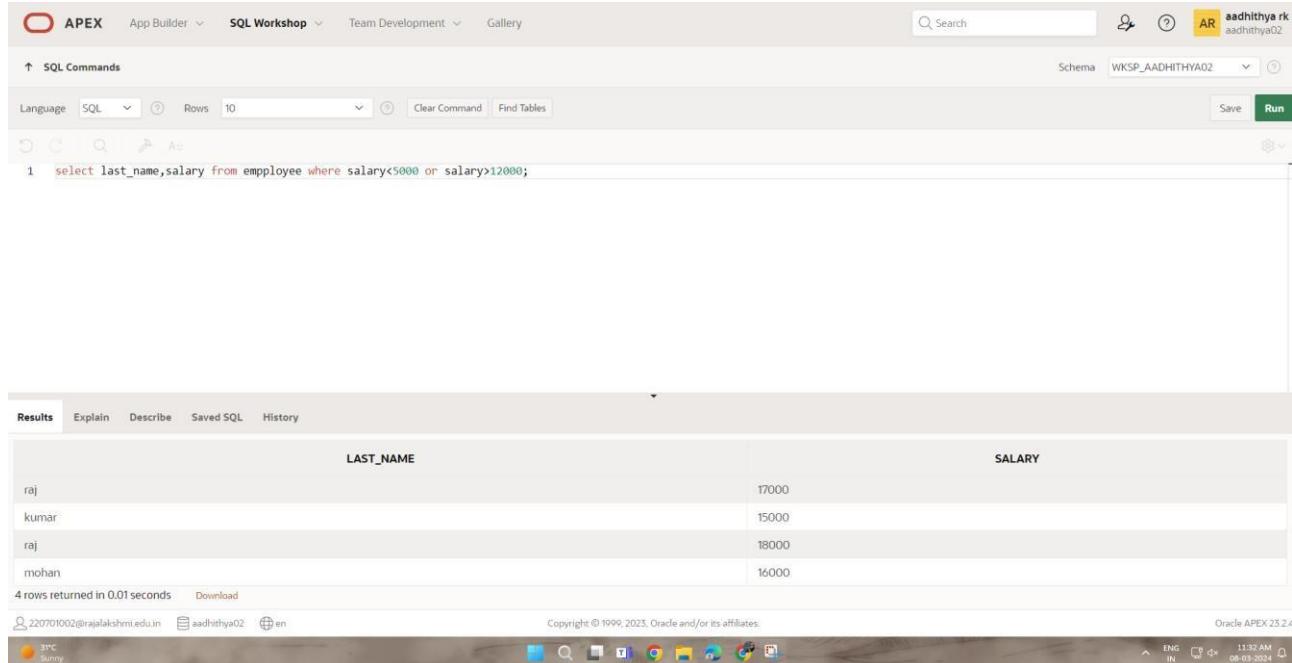
Below the table, it says '1 rows returned in 0.00 seconds'. The bottom status bar shows the URL '220701002@rajalakshmi.edu.in', session 'aadithya02', and browser icons. The system status bar indicates '28°C High UV' and the date '08-03-2024'.

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 EMPLOYEE where salary not between 5000 and 12000;
```

OUTPUT:



LAST_NAME	SALARY
raj	17000
kumar	15000
raj	18000
mohan	16000

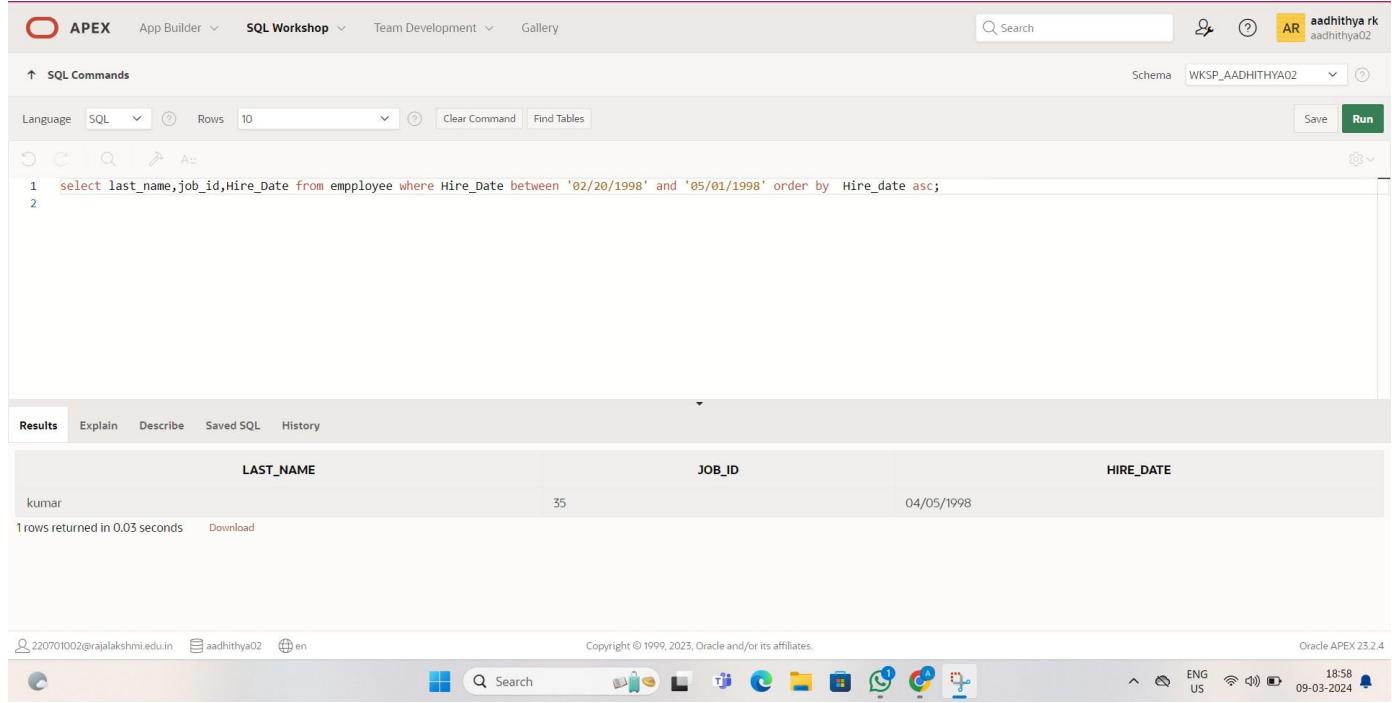
4 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:



LAST_NAME	JOB_ID	HIRE_DATE
kumar	35	04/05/1998

1 rows returned in 0.03 seconds

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

QUERY:

```
select last_name, department_number from EMPLOYEE where department in (20,50)
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', 'Team Development', and 'Gallery'. On the right, it shows the schema 'WKS_AADHITHYA02' and a user icon 'aadhithya rk'. The main area has tabs for 'SQL Commands', 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected. The SQL command entered is: 'select last_name, department_id from employee where department_id between 20 and 50 order by first_name;'. The results table has columns 'LAST_NAME' and 'DEPARTMENT_ID'. The data returned is:

LAST_NAME	DEPARTMENT_ID
raj	34
raj	50
kumar	20

Below the table, it says '3 rows returned in 0.01 seconds' and has a 'Download' link. The bottom status bar shows the user '220701002@rajalakshmi.edu.in', the schema 'aadhithya02', and the date/time '08-03-2024 11:40 AM'.

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 employee where (salary between 5000 and 12000) and (department_id in(20,50)) order by last_name asc;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. On the right, it shows the schema 'WKS_AADHITHYA02' and a user icon 'aadhithya rk'. The main area has tabs for 'SQL Commands', 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected. The SQL command entered is: 'select last_name as "employee", salary as "monthly salary" from employee where department_id between 20 and 50 and salary between 5000 and 12000 order by last_name;'. The results table has columns 'employee' and 'monthly salary'. The data returned is:

employee	monthly salary
kumar	11000
raj	6000

Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link. The bottom status bar shows the user '220701002@rajalakshmi.edu.in', the schema 'aadhithya02', and the date/time '08-03-2024 11:46 AM'.

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 employee where hire_date like '%94';
```

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 'aadithya02'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is: `select last_name,hire_date from employee where hire_date like '%94';`. The results table shows one row: mohan with hire date 07/09/1994. Below the table, it says '1 rows returned in 0.00 seconds'. The bottom status bar shows the URL '220701002@rajalakshmi.edu.in', the schema 'aadithya02', and the environment 'en'. It also displays system information like 'Copyright © 1999, 2023, Oracle and/or its affiliates.', the date '09-03-2024', and the time '19:01'.

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 EMPLOYEE WHERE manager_id IS NULL;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows a user profile for 'aadithya02'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is: `select last_name, job_id from employee where manager_id is null;`. The results table shows one row: raj with job_id 40. Below the table, it says '1 rows returned in 0.01 seconds'. The bottom status bar shows the URL '220701002@rajalakshmi.edu.in', the schema 'aadithya02', and the environment 'en'. It also displays system information like 'Copyright © 1999, 2023, Oracle and/or its affiliates.', the date '08-03-2024', and the time '11:51 AM'.

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

QUERY:

```
SELECT last_name, salary, commission_pct FROM EMPLOYEE WHERE commission_pct IS NOT NULL ORDER BY salary DESC, commission_pct 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'. The right side shows user information: 'aadithya rk' and 'aadithya02'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is:

```
1 SELECT last_name, salary, commission_pct
2 FROM EMPLOYEE
3 WHERE commission_pct IS NOT NULL
4 ORDER BY salary DESC, commission_pct DESC;
```

The results section displays a table with three columns: LAST_NAME, SALARY, and COMMISSION_PCT. The data is as follows:

LAST_NAME	SALARY	COMMISSION_PCT
raj	18000	55
mohan	16000	78
kumar	11000	24
raj	6000	33
eshawar	2500	20

Below the table, it says '5 rows returned in 0.01 seconds'. The bottom status bar shows the user's email ('220701002@rajalakshmi.edu.in'), session ID ('aadithya02'), and the system date/time ('09-03-2024 19:10').

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

QUERY:

```
SELECT last_name FROM EMPLOYEE WHERE last_name LIKE '__a%';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows user information: 'aadithya rk' and 'aadithya02'. The main area has tabs for 'SQL Commands' and 'Results'. The SQL command entered is:

```
1 SELECT last_name
2 FROM EMPLOYEE
3 WHERE last_name LIKE '__a%';
```

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

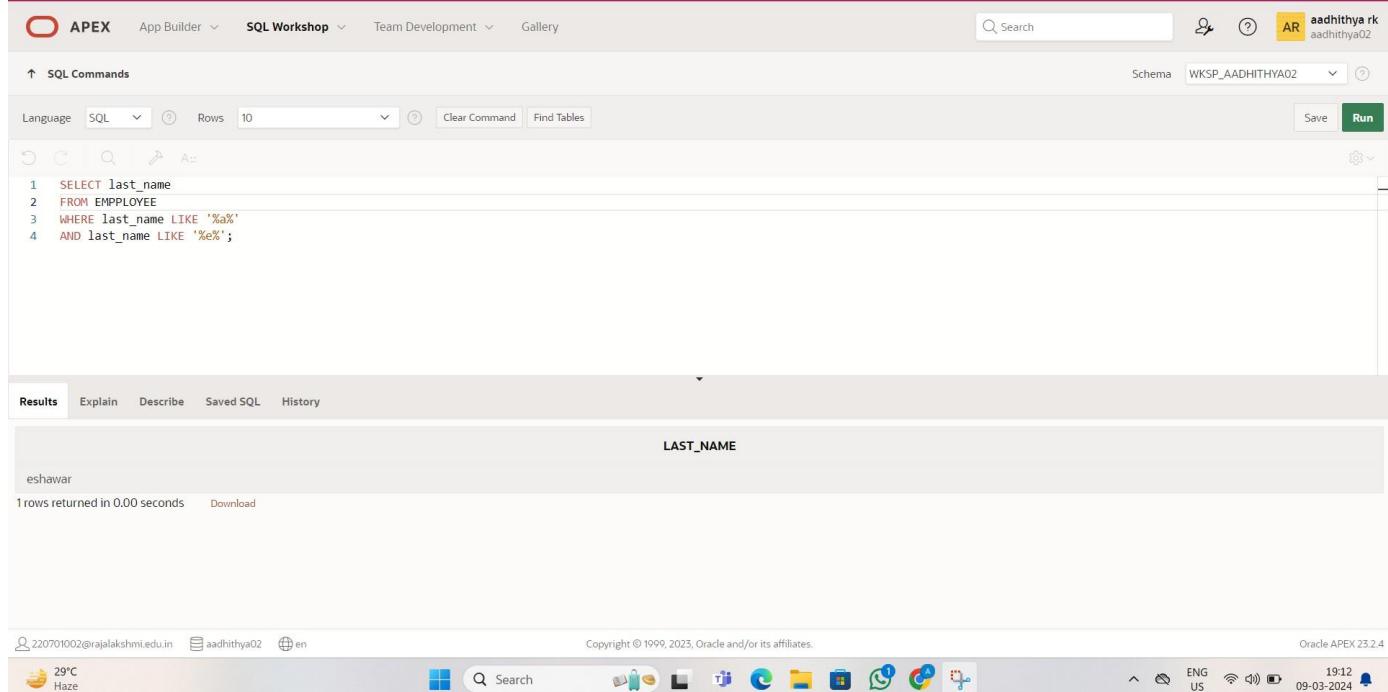
Below the message, the bottom status bar shows the user's email ('220701002@rajalakshmi.edu.in'), session ID ('aadithya02'), and the system date/time ('09-03-2024 19:11').

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 EMPLOYEE WHERE last_name LIKE '%a%' AND last_name LIKE '%e%';
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user's profile: AR aadhisthyaa02. The main area is titled "SQL Commands" with a search bar and a "Run" button. The code entered is:

```
1 SELECT last_name
2 FROM EMPLOYEE
3 WHERE last_name LIKE '%a%'
4 AND last_name LIKE '%e%';
```

Below the code, the results tab is selected, showing a single row:

LAST_NAME
eshawar

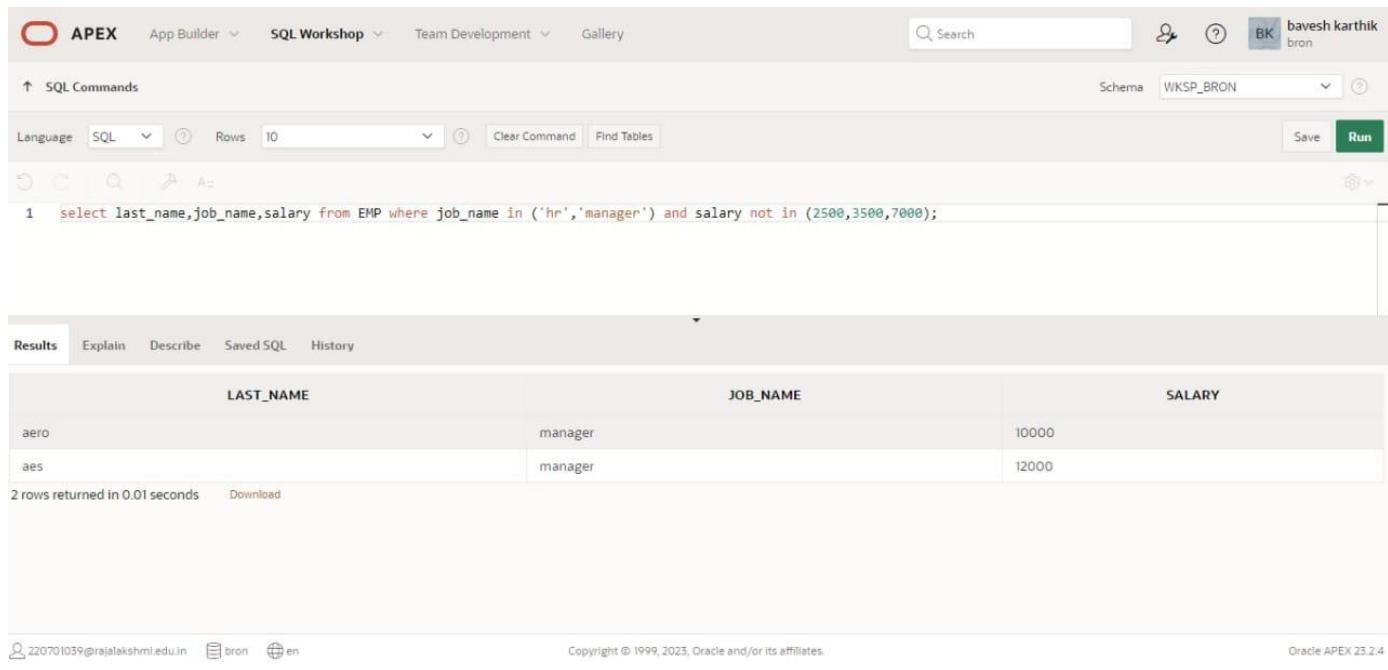
1 rows returned in 0.00 seconds. The bottom status bar shows the user's email (220701002@rajalakshmi.edu.in) and session ID (aadhisthyaa02).

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_name,salary from EMPLOYEE where job_name in ('stock clerk', 'sales representative') 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. The right side shows the user's profile: BK bavesh.karthik.bron. The main area is titled "SQL Commands" with a search bar and a "Run" button. The code entered is:

```
1 select last_name,job_name,salary from EMP where job_name in ('hr','manager') and salary not in (2500,3500,7000);
```

Below the code, the results tab is selected, showing two rows:

LAST_NAME	JOB_NAME	SALARY
aero	manager	10000
aes	manager	12000

2 rows returned in 0.01 seconds. The bottom status bar shows the user's email (220701039@rajalakshmi.edu.in) and session ID (bron).

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 employee where commission_pct=0.2;
```

OUTPUT:

The screenshot shows the Oracle 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 search bar, a user icon for 'aadithya', and a schema dropdown set to 'WKSP_AADHITHYA02'. Below the tabs, there are buttons for Language (SQL), Rows (10), Clear Command, and Find Tables. The main area contains the SQL command: '1 select last_name,salary,commission_pct from employee where commission_pct=0.2;'. Underneath this, there's a results section with tabs for Results, Explain, Describe, Saved SQL, and History. The results table is currently empty, displaying '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:

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

QUERY:

```
select sysdate from dual;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'aadhihya rk' and the schema 'WKSP_AADHIHYA02'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The code entered is 'select sysdate from dual;'. The results section shows the output: 'SYSDATE' followed by the date '04/11/2024'. Below the results, it says '1 rows returned in 0.03 seconds' and has a 'Download' link.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'aadhihya rk' and the schema 'WKSP_AADHIHYA02'. The main area is titled 'SQL Commands' with a search bar and a 'Run' button. The code entered is 'select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary" from employees;'. 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
12	raj	6000	6930
4	davies	2500	2887.5
1	kumar	11000	12705
2	arun	18000	20790
1	Mohan	16000	18480

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 SQL Workshop interface. The query `select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary", new_salary-salary as "Increase" from employees;` is run, and the results are displayed in a table. The table has columns: EMPLOYEE_ID, LAST_NAME, SALARY, new_salary, and Increase. The data shows five rows of employee information with their calculated new salaries and increases.

EMPLOYEE_ID	LAST_NAME	SALARY	new_salary	Increase
12	raj	6000	6930	-
4	davies	2500	2887.5	-
1	kumar	11000	12705	-
2	arun	18000	20790	-
1	Mohan	16000	18480	-

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The 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;` is run, and the results are displayed in a table. The table has columns: INITCAP(LAST_NAME) and Length_of_last_name. The data shows one row for employee 'Mohan'.

INITCAP(LAST_NAME)	Length_of_last_name
Mohan	5

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 last_name from employee where last_name like 'a%';
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'aadithya rk' with the schema 'WKSP_AADITHYA02'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

```
1 select last_name from employee where last_name like 'a%';
2
3
```

The Results tab displays the output:

LAST_NAME
arun

1 rows returned in 0.05 seconds [Download](#)

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'aadithya rk' with the schema 'WKSP_AADITHYA02'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

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

The Results tab displays the output:

LAST_NAME	MONTHS_WORKED
arun	286
davies	295
kumar	317
raj	319
Mohan	362

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

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for 'aadithya rk' with the schema 'WKSP_AADITHYA02'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following code:

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

The Results tab displays the output:

DREAM_SALARIES
raj earns 6000 monthly but wants 18000
davies earns 2500 monthly but wants 7500
kumar earns 11000 monthly but wants 33000
arun earns 18000 monthly but wants 54000
Mohan earns 16000 monthly but wants 48000

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information (aadhihya rk), and a schema dropdown set to WKSP_AADHIHYA02. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query: `select last_name,lpad(salary,15,'$') as "SALARY" from employees;`. The Results tab displays the output in a table:

LAST_NAME	SALARY
raj	\$\$\$\$\$\$\$\$\$\$6000
davies	\$\$\$\$\$\$\$\$\$\$2500
kumar	\$\$\$\$\$\$\$\$\$\$11000
arun	\$\$\$\$\$\$\$\$\$\$18000
Mohan	\$\$\$\$\$\$\$\$\$\$16000

Below the table, it says "5 rows returned in 0.01 seconds" and has a "Download" link.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface, identical to the previous one but with a different query. 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 (aadhihya rk), and a schema dropdown set to WKSP_AADHIHYA02. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the query: `SELECT last_name,hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY'),'FMDay, "the
"FMDD "of "FMMonth, YYYY') AS REVIEW FROM employees;`. The Results tab displays the output in a table:

LAST_NAME	HIRE_DATE	REVIEW
raj	02/10/1998	Monday, the 17 of August, 1998
davies	01/08/2000	Monday, the 10 of July, 2000
kumar	04/05/1998	Monday, the 12 of October, 1998
arun	10/07/2000	Monday, the 09 of April, 2001
Mohan	07/09/1994	Monday, the 16 of January, 1995

Below the table, it says "5 rows returned in 0.01 seconds" and has a "Download" link.

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 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 (aadhihya rk aadhihya02), and a schema dropdown set to WKSP_AADHIHYA02. Below the toolbar, the SQL Commands section shows the query being run: "SELECT last_name,hire_date,TO_CHAR(hire_date,'Day') as Day from employees order by TO_CHAR(hire_date,'Day');". The results tab is selected, displaying the output in a grid format:

LAST_NAME	HIRE_DATE	DAY
arun	10/07/2000	Saturday
davies	01/08/2000	Saturday
Mohan	07/09/1994	Saturday
kumar	04/05/1998	Sunday
raj	02/10/1998	Tuesday

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

RESULT:

DISPLAYING DATA FROM MULTIPLE TABLES

EX_NO:7

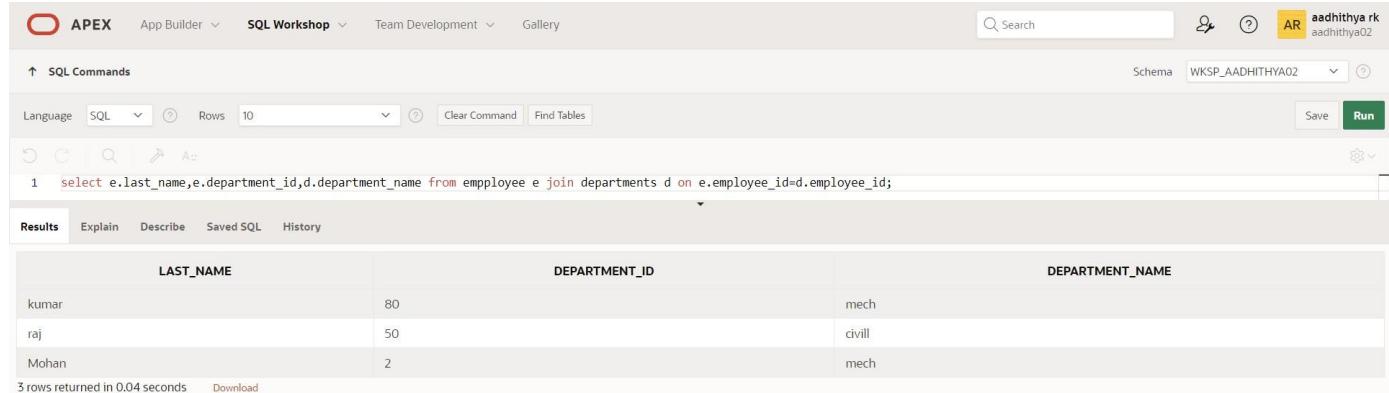
DATE:

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

QUERY:

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

OUTPUT:



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

```
1 select e.last_name,e.department_id,d.department_name from employee e join departments d on e.employee_id=d.employee_id;
```

The results tab displays the output:

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
kumar	80	mech
raj	50	civil
Mohan	2	mech

3 rows returned in 0.04 seconds

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

QUERY:

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

OUTPUT:



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

```
1 select e.job_name,d.location_id from employee e join departments d on e.employee_id=d.employee_id where e.department_id=80
```

The results tab displays the output:

JOB_NAME	LOCATION_ID
manager	67

1 rows returned in 0.01 seconds

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_AADHITHYA02

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 select e.last_name,d.department_name,d.location_id,d.location_city from employee e join departments d on e.employee_id=d.employee_id where e.commission_pct is not null;
```

Results Explain Describe Saved SQL History

LAST_NAME	DEPARTMENT_NAME	LOCATION_ID	LOCATION_CITY
kumar	mech	67	nagpur
raj	civil	33	chennai
Mohan	mech	67	nagpur

3 rows returned in 0.05 seconds Download

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

QUERY:

Select e.last_name,d.dept_name from empployee,departments where e.department_number=d.dept_id and last_name like '%a%';

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_AADHITHYA02

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 select e.last_name,d.department_name from empployee e join departments d on e.employee_id=d.employee_id where e.last_name like 'a%'
```

Results Explain Describe Saved SQL History

LAST_NAME	DEPARTMENT_NAME
arun	civil

1 rows returned in 0.01 seconds Download

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

QUERY:

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

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands Schema WKSP_AADHITHYA02

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 select last_name,job_name,department_id,department_name from empployee where location_city='toronto'
```

Results Explain Describe Saved SQL History

LAST_NAME	JOB_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
raj	sales executive	34	cse

1 rows returned in 0.02 seconds Download

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

QUERY:

```
Select w.last_name "Employee",w.emp_id "emp#",m.last_name 'manager',m.emp_id "Mgr#" from empemployee m on (w.manager_id=m.emp_id);
```

OUTPUT:

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

```
1 select last_name as "Employee",emp_id as "emp#",manager_name as "manager",manager_id as "mgr" from empemployee;
```

The results are displayed in a grid:

employee	emp#	manager	mgr
raj	12	kumar	-
eshawar	4	vinoth	13
kumar	1	thosh	12
arun	2	kamaraj	9
Mohan	1	akilan	19

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

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

QUERY:

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

OUTPUT:

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

```
1 e.last_name,e.employee_id,e.manager_name,e.department_name,d.location_id,d.location_city,e.salary from empemployee e join departments d on e.employee_id=d.employee_id order by e.employee_id;
```

The results are displayed in a grid:

LAST_NAME	EMPLOYEE_ID	MANAGER_NAME	DEPARTMENT_NAME	LOCATION_ID	LOCATION_CITY	SALARY
kumar	1	thosh	it	67	TORONTO	11000
Mohan	1	akilan	cse	67	TORONTO	16000
arun	2	kamaraj	aero	33	chennai	18000

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

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

QUERY:

```
select e.department_number dept23,e.last_name colleague from empemployee e join empemployee c on (e.department_number=c.department_number) where e.emp_id <> c.emp_id order by e.department_number,e.last_name,c.last_name;
```

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's a search bar, user profile information for 'aadithya rk' (aadithya02), and a schema dropdown set to 'WKSP_AADHITHYA02'. Below the header, the SQL Commands tab is selected, showing a query: 'select e.last_name, e.department_id, d.department_name from employee e join departments d on e.employee_id=d.employee_id order by e.last_name;'. The Results tab is active, displaying the output:

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Mohan	2	mech
arun	50	civil
kumar	80	mech

3 rows returned in 0.01 seconds. There are 'Download' and 'Run' buttons at the bottom.

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 employee e
JOIN departments d ON (e.dept_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. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile information for 'BHANU PRIYA' (bhunu25), and a schema dropdown set to 'WKSP_BHANU25'. Below the header, the SQL Commands tab is selected, showing the same query as above. The Results tab is active, displaying the output:

LAST_NAME	JOB_ID	DEPT_NAME	SALARY	GRADE_LEVEL

no data found

At the bottom, there are footer links for '220701040@rajalakshmi.edu.in', 'bhunu25', and 'en'. Copyright information 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also present.

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

OUTPUT:

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

```
1 select e.last_name,e.hire_date from employee e join employee davies on(davies.last_name='davies')where davies.hire_date < e.hire_date;
```

The results table has two columns: LAST_NAME and HIRE_DATE. One row is returned:

LAST_NAME	HIRE_DATE
arun	10/07/2000

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

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

OUTPUT:

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

```
1 select e.last_name as "employee",e.hire_date as "emp_hire",e.manager_name as "manager",d.manager_hiredate as "mgr_hire"from employee e join departments d on (e.employee_id=d.employee_id)
2 where e.hire_date < d.manager_hiredate;
```

The results table has four columns: employee, emp_hire, manager, and mgr_hire. Three rows are returned:

employee	emp_hire	manager	mgr_hire
kumar	04/05/1998	thosh	08/07/2000
arun	10/07/2000	kamaraj	04/24/2001
Mohan	07/09/1994	akilan	08/07/2000

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

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 EMPLOYEE;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a user profile 'aadhistya rk' and a schema dropdown 'WKSP_AADHITHYA02'. The main workspace has a SQL command editor with the following content:

```
1 select round(max(salary),0)"maximum",round(min(salary),0)"minimum",round(sum(salary),0)"sum",round(avg(salary),0)"average" from employee;
```

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

maximum	minimum	sum	average
18000	2500	53500	10700

At the bottom left, it says '1 rows returned in 0.05 seconds'.

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 EMPLOYEE group by job_id;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'aadithya rk' and a schema dropdown set to 'WKSP_AADITHYA02'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

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

The Results tab displays a table with the following data:

JOB_ID	MAXIMUM	MINIMUM	sum	average
40	6000	6000	6000	6000
56	2500	2500	2500	2500
17	18000	18000	18000	18000
35	11000	11000	11000	11000
38	16000	16000	16000	16000

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 EMPLOYEE where job_id='47' group by job_id ;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'aadithya rk' and a schema dropdown set to 'WKSP_AADITHYA02'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

```
1  
2 select job_id, count(*) from EMPLOYEE where job_id='47' group by job_id ;  
3  
4
```

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

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 employee;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a user profile for 'aadithya rk' and a schema dropdown set to 'WKSP_AADITHYA02'. The main area has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

```
1  
2 select count(distinct manager_id )"Number of managers" from employee;  
3  
4
```

The Results tab displays a table with one row:

Number of managers
4

Below the table, it says '1 rows returned in 0.01 seconds'.

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

QUERY:

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

OUTPUT:

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

```
1 select max(salary)-min(salary) difference from employee
```

The results show a single row with the value 15500, labeled 'DIFFERENCE'.

DIFFERENCE
15500

3 rows returned in 0.00 seconds

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

QUERY:

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

OUTPUT:

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

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

The results show three rows with manager IDs 9, 19, and 12, and their corresponding minimum salaries: 18000, 16000, and 11000, respectively.

MANAGER_ID	MIN(SALARY)
9	18000
19	16000
12	11000

3 rows returned in 0.01 seconds

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

QUERY:

```
select count(*) 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 employee;
```

OUTPUT:

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

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

The results show a single row with columns TOTAL, 1995, 1996, 1997, and 1998. The values are 5, 0, 0, 0, and 2 respectively.

TOTAL	1995	1996	1997	1998
5	0	0	0	2

1 rows returned in 0.01 seconds

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

QUERY:

```
select job_id "job", sum(decode(dept_id,20,salary))"Dept20",sum (decode(dept_id ,50, salary))  
"dept50",sum (decode(dept_id ,80, salary)) "dept80",sum (decode(dept_id ,90, salary)) "dept90",sum(salary)  
"TOTAL" from employee group by job_id
```

OUTPUT:

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

```
1 select job_id "job", sum(decode(department_id,20,salary))"Dept20",sum (decode(department_id ,50, salary))  
2 "dept50",sum (decode(department_id ,80, salary)) "dept80",sum (decode(department_id ,90, salary)) "dept90",sum(salary)  
3 "TOTAL" from employee group by job_id
```

The results are displayed in a matrix format:

job	Dept20	dept50	dept80	dept90	TOTAL
40	-	-	-	-	6000
56	-	-	2500	-	2500
17	-	18000	-	-	18000
35	-	-	11000	-	11000
38	-	-	-	-	16000

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

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

QUERY:

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

OUTPUT:

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

```
1 t(*) "Number of people",round(avg(salary),2) "salary" from mydept d inner join employee e on(d.deptid =e.department_id ) group by d.dept_name ,d.loc;
```

The results are displayed in a table:

dept_name	department location	Number of people	salary
CSBS	Bangaluru	1	15000
CSE	Chennai	3	16333.33

2 rows returned in 0.05 seconds [Download](#)

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

RESULT:

SUB QUERIES

EX_NO:9

DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon, and a session identifier 'aadhihya02'. The main workspace shows a SQL command line with the following code:

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

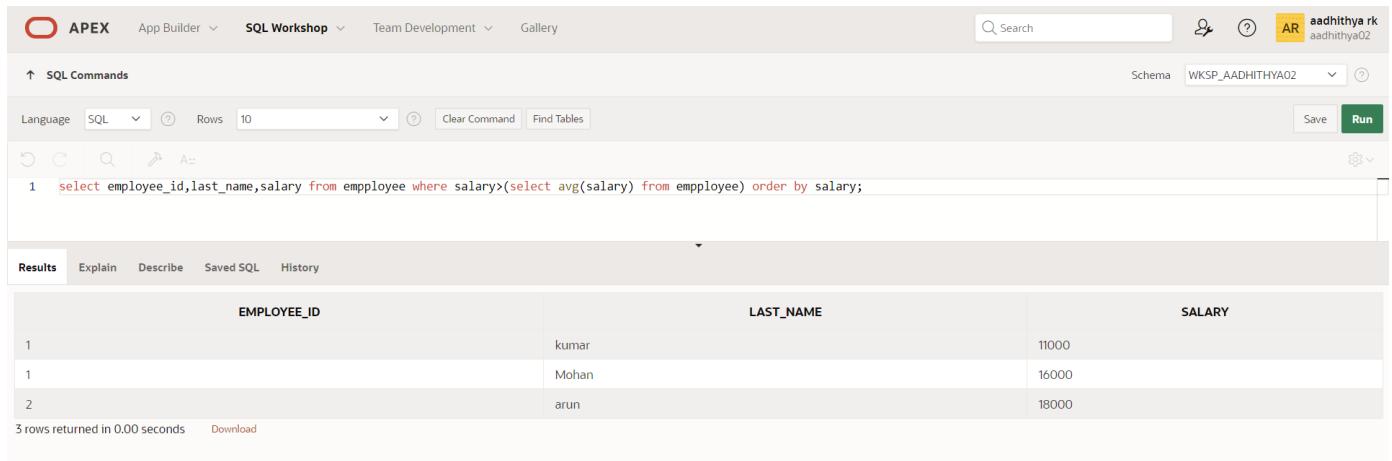
The results tab is selected, showing the message "no data found".

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface with the same navigation and session information as the previous screenshot. The SQL command line contains the same query as above. The results tab displays the following data:

EMPLOYEE_ID	LAST_NAME	SALARY
1	kumar	11000
1	Mohan	16000
2	arun	18000

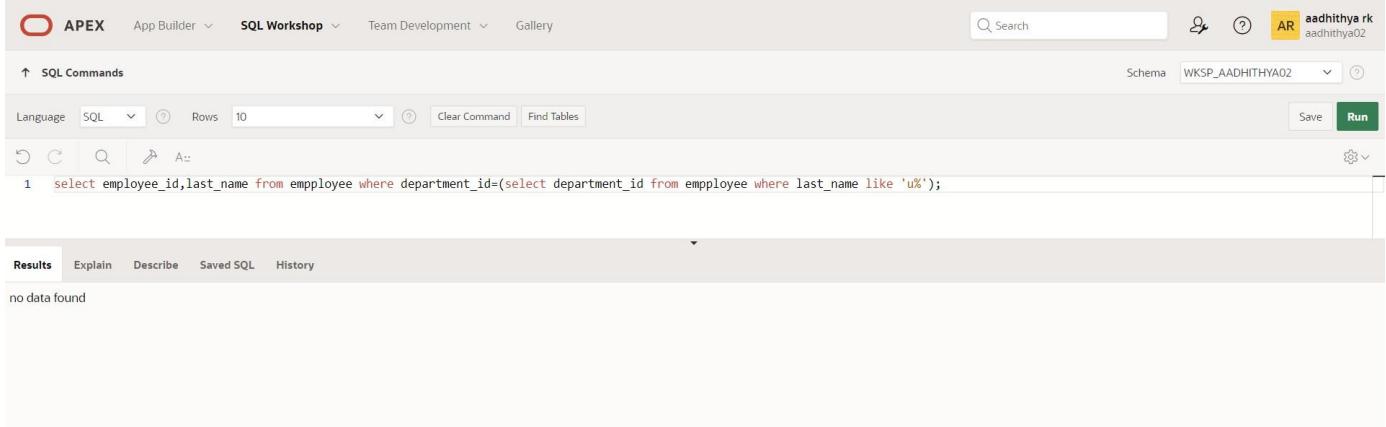
At the bottom, it says "3 rows returned in 0.00 seconds" and has a "Download" link.

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



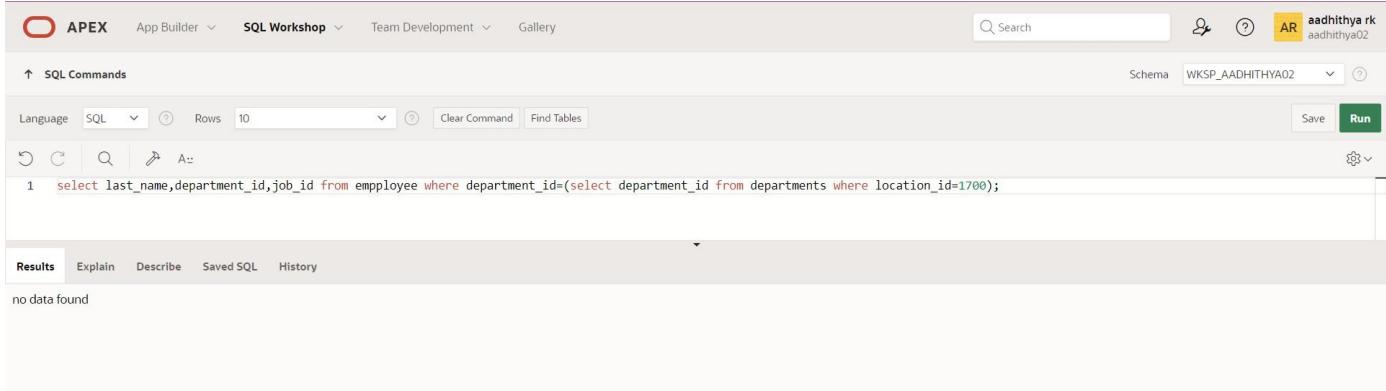
The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'aadithya rk', and a schema dropdown set to 'WKSP_AADHITHYA02'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The command entered is: `1 select employee_id,last_name from employees where department_id=(select department_id from employees where last_name like 'u%');`. Below the command, the 'Results' tab is selected, showing the message 'no data found'.

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'aadithya rk', and a schema dropdown set to 'WKSP_AADHITHYA02'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The command entered is: `1 select last_name,department_id,job_id from employees where department_id=(select department_id from departments where location_id=1700);`. Below the command, the 'Results' tab is selected, showing the message 'no data found'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle 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 (aadithya rk), and a schema dropdown set to WKSP_AADHITHYA02. The main area is titled 'SQL Commands' with a 'Run' button. The code input field contains the following SQL query:

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

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user information (aadithya rk), and a schema dropdown set to WKSP_AADHITHYA02. The main area is titled 'SQL Commands' with a 'Run' button. The code input field contains the following SQL query:

```
1 select department_id,last_name,job_id from employee where department_id in (select department_id from departments where department_name='Executive');
```

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

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user information (aadithya rk), and a schema dropdown set to WKSP_AADHITHYA02. The main area is titled 'SQL Commands' with a 'Run' button. The code input field contains the following SQL query:

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

The results tab is selected, displaying the output:

EMPLOYEE_ID	LAST_NAME	SALARY
2	arun	18000
1	Mohan	16000

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

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

RESULT:

USING THE SET OPERATORS

EX_NO:10

DATE:

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

QUERY:

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

OUTPUT:



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

```
1 SELECT department_id FROM departments MINUS SELECT department_id FROM employee WHERE
2 job_name = 'st_clerk';
```

The results table has one column labeled "DEPARTMENT_ID" with two rows: 56 and 90. A note at the bottom says "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 SQL Workshop interface. The SQL command entered is:

```
1 SELECT country_id,country_name FROM country MINUS SELECT
2 1.country_id,c.country_name FROM location 1, country c WHERE 1.country_id
3 =>c.country_id;
```

The results table has two columns: "COUNTRY_ID" and "COUNTRY_NAME". There is one row with "1" in the COUNTRY_ID column and "new zealand" in the COUNTRY_NAME column. A note at the bottom says "1 rows returned in 0.01 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:

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

```
1 SELECT DISTINCT job_id, department_id FROM employee WHERE
2 department_id = 10 UNION ALL
3 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 50
4 UNION ALL
5 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 20
```

The results table has two columns: JOB_ID and DEPARTMENT_ID. The data returned is:

JOB_ID	DEPARTMENT_ID
17	50

1 rows returned in 0.03 seconds

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

QUERY:

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

OUTPUT:

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

```
1 SELECT employee_id,job_id FROM employee
2 INTERSECT
3 SELECT employee_id,job_id FROM job_history;
```

The results table has two columns: EMPLOYEE_ID and JOB_ID. The data returned is:

EMPLOYEE_ID	JOB_ID
3	56

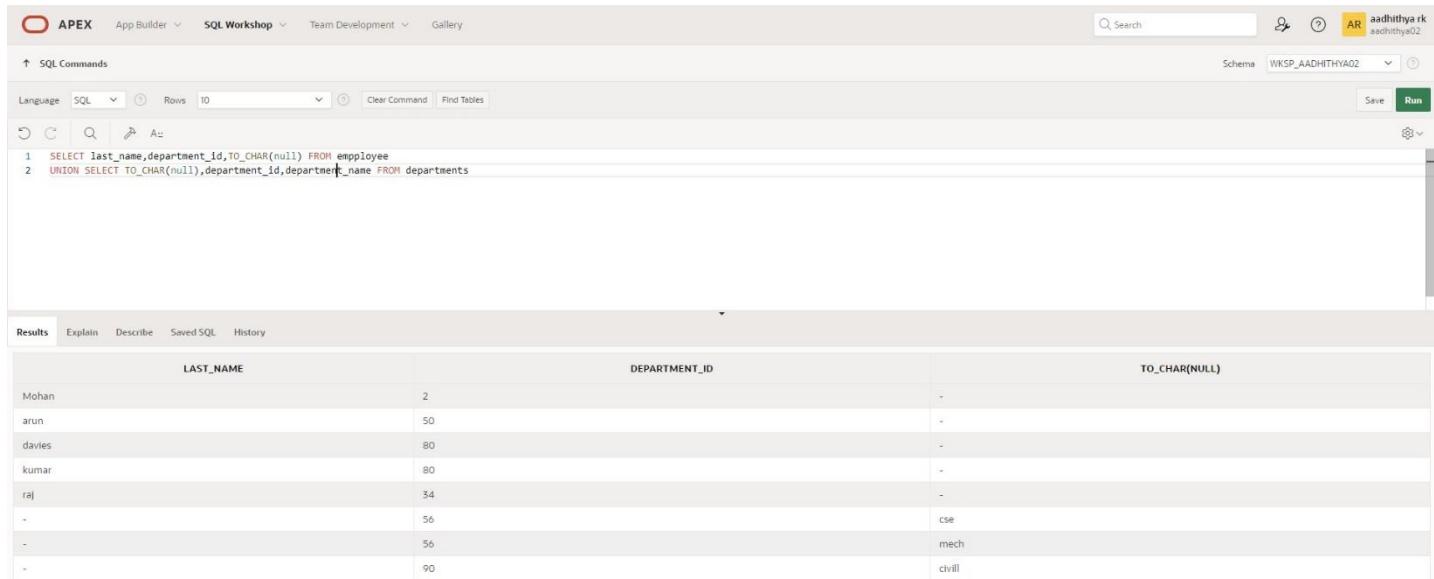
1 rows returned in 0.00 seconds

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle 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 'aadithya rk' with session information. The main area displays a SQL command window with the following content:

```
1 SELECT last_name,department_id,TO_CHAR(null) FROM employee
2 UNION SELECT TO_CHAR(null),department_id,department_name FROM departments
```

Below the command window is a results grid. The columns are labeled LAST_NAME, DEPARTMENT_ID, and TO_CHAR(NULL). The data is as follows:

LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
Mohan	2	-
arun	50	-
davies	80	-
kumar	80	-
raj	54	-
-	56	cse
-	56	mech
-	90	civil

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

RESULT:

CREATING VIEWS

EX NO:11

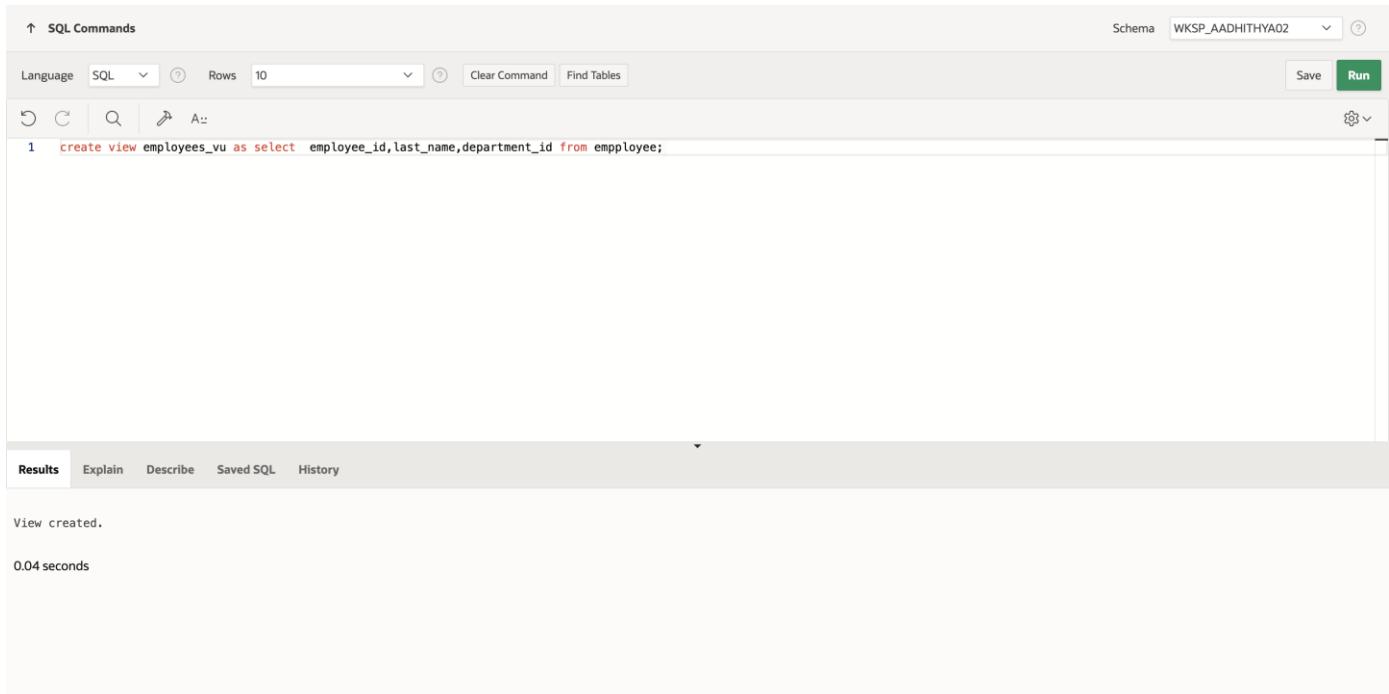
DATE:

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 a SQL command window with the following details:

- Header:** SQL Commands, Schema: WKSP_AADHITHYA02
- Toolbar:** Language (SQL), Rows (10), Clear Command, Find Tables, Save, Run
- Query Editor:** A single line of SQL code: `1 create view employees_vu as select employee_id, last_name, department_id from employee;`
- Results Tab:** Results, Explain, Describe, Saved SQL, History
- Output:** View created.
0.04 seconds

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's a toolbar with various icons and a schema dropdown set to 'WKSP_AADHITHYA02'. Below the toolbar is a command input area where the query 'select * from employees_vu;' has been entered. The 'Run' button is highlighted in green. Below the command area is a results grid. The first row of the grid is labeled with column names: 'EMPLOYEE_ID', 'LAST_NAME', and 'DEPARTMENT_ID'. The subsequent rows contain data: (3, raj, 34), (4, davies, 80), (1, kumar, 80), (2, arun, 50), and (1, Mohan, 2). The bottom of the screen displays user information and copyright notices.

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
3	raj	34
4	davies	80
1	kumar	80
2	arun	50
1	Mohan	2

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

QUERY:

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

OUTPUT:

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL workspace interface. At the top, there's a toolbar with various icons and a dropdown for 'Schema' set to 'WKSP_AADHITHYA02'. Below the toolbar is a search bar and a 'Run' button. The main area contains a SQL command line with the following code:

```
1 select last_name,department_id from employee;
```

Below the command line, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, displaying a table with two columns: 'LAST_NAME' and 'DEPARTMENT_ID'. The data in the table is:

LAST_NAME	DEPARTMENT_ID
raj	34
davies	80
kumar	80
arun	50
Mohan	2

At the bottom of the page, there are footer links for '220701002@rajalakshmi.edu.in', 'aadhitanya02', 'en', 'Copyright © 1999, 2023, Oracle and/or its affiliates.', and 'Oracle APEX 23.2.4'.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle APEX SQL Commands interface. The top navigation bar includes 'SQL Commands', 'Schema' set to 'WKSP_AADHITHYA02', and a 'Run' button. Below the toolbar are buttons for Language (SQL), Rows (10), Clear Command, and Find Tables. The main area contains a single SQL command:

```
1 create view dept_50 as select employee_id EMPNO, last_name EMPLOYEE, department_id DEPTNO from employee where department_id=50;
```

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

```
View created.  
0.03 seconds
```

At the bottom, footer information includes the user's email (220701002@rajalakshmi.edu.in), the schema name (aadhittha02), and the language (en). The copyright notice reads "Copyright © 1999, 2023, Oracle and/or its affiliates." and the version is "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 'SQL Commands', 'Schema' set to 'WKSP_AADHITHYA02', and a 'Run' button. The main area contains a SQL editor with the command 'select * from dept_50;'. Below the editor is a results grid with three columns: 'EMPNO', 'EMPLOYEE', and 'DEPTNO'. One row is displayed, showing values 2, arun, and 50 respectively. The bottom status bar indicates '1 rows returned in 0.03 seconds' and provides download options.

EMPNO	EMPLOYEE	DEPTNO
2	arun	50

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

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 workspace interface. The top navigation bar includes 'SQL Commands', 'Schema' set to 'WKSP_AADHITHYA02', and a 'Run' button. The main area displays the SQL command:

```
1 update dept_50 set deptno=80 where employee='matos';
```

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

```
0 row(s) updated.
```

Execution time is listed as 0.05 seconds. The bottom footer includes user information, copyright notice, and version details.

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 'SQL Commands', 'Schema' set to 'WKSP_AADHITHYA02', and a 'Run' button. The main area displays the SQL command for creating the 'salary_vu' view:

```
1 create view salary_vu as select last_name EMPLOYEE,department_name DEPARTMENT,salary SALARY from empployee;
```

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

```
View created.
```

Execution time: 0.05 seconds

At the bottom, the footer includes user information (220701002@rajalakshmi.edu.in, aadhithya02, en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the version (Oracle APEX 23.2.4).

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

RESULT:

EXERCISE 12

PRACTICE QUESTIONS

Intro to Constraints; NOT NULL and UNIQUE Constraints

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Table Created.

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

```
DESCRIBE f_global_locations;
```

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

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

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

PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

1. What is the purpose of a

- PRIMARY KEY
- FOREIGN KEY
- CHECK CONSTRAINT

a. PRIMARY KEY

Uniquely identify each row in table.

b. FOREIGN KEY

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

c. CHECK CONSTRAINT

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

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

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

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

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

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

```
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

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

added to the animals table.

COLUMN LEVEL STATEMENT:

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

TABLE LEVEL STATEMENT:

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

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

- a. ON DELETE CASCADE

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

- b. ON DELETE SET NULL

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

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

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

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

PRACTICE PROBLEM

Managing Constraints

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

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

1. What are four functions that an ALTER statement can perform on constraints?
 - ADD
 - DROP
 - ENABLE
 - DISABLE
2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy_d_clients table. Name the primary key copy_d_clients_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy_d_clients.table?

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

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

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

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

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

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

COPY_D_CLT_CLIENT_NUMBER_PK

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

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

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

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

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

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

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

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

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

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

1 row(s) inserted.

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

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

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

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

1 row(s) deleted.

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

Table altered.

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

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

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

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

- C - Check constraint

Sub-case - if I see SEARCH_CONDITION something like "FIRST_NAME" IS NOT NULL , its a NOT NULL constraint.

- P - Primary key
- R - Referential integrity (fk)
- U - Unique key

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

EXERCISE 13

Creating Views

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

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

3. SELECT * FROM view_d_songs. What was returned?

Results	Explain	Describe	Saved SQL	History
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;
```

- 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

- 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;
```

- 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';

- 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;

- Use the read_copy_d_cds view to delete year 2001 records.

DELETE FROM read_copy_d_cds WHERE year = '2001';

- 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

- 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**

- 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.

- 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, artistFROM  
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_idFROM(SELECT dpt.department_id,  
MAX(NVL(emp.salary,0)) max_dpt_salFROM departments dpt LEFT OUTER JOIN employees emp ON  
dpt.department_id = emp.department_idGROUP BY dpt.department_id) dptmx LEFT OUTER JOIN  
employees empm ON dptmx.department_id = empm.department_idWHERE 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, salaryFROM(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_idxON 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;
```

OTHER DATABASE OBJECTS

EX_NO:14

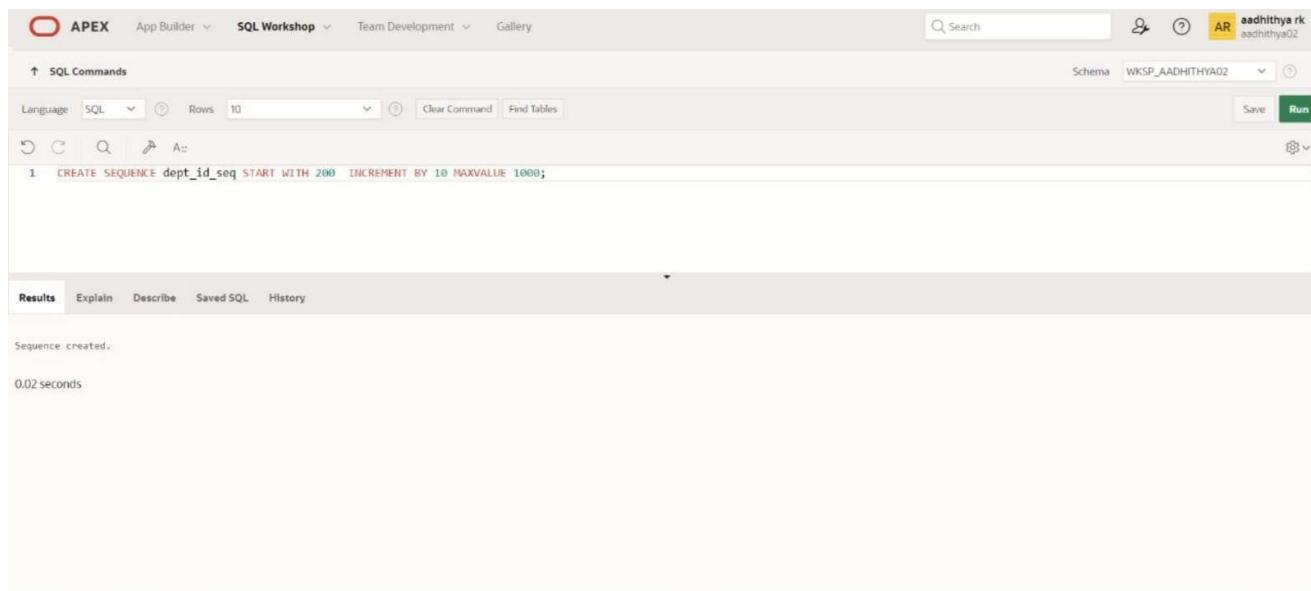
DATE:

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 SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'CREATE SEQUENCE dept_id_seq START WITH 200 INCREMENT BY 10 MAXVALUE 1000;'. Below the command, the results section displays the message 'Sequence created.' and a execution time of '0.02 seconds'.

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

QUERY:

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

OUTPUT:

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

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

The results pane shows the output of the query:

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPT_ID_SEQ	1000	10	200

Below the table, it says "1 rows returned in 0.02 seconds" and there is a "Download" link.

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

QUERY:

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface with the execution summary for script 'ex14'. The summary table is as follows:

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

Below the table, it says "row(s) 1 - 1 of 1".

At the bottom, there are three status indicators: "Statements Processed" (1), "Successful" (1), and "With Errors" (0).

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. In the top navigation bar, 'APEX' is selected. The main area is titled 'SQL Commands'. A single line of SQL code is entered: 'CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);'. Below the command, the results tab is active, showing the output: 'Index created.' and '0.03 seconds'. The schema dropdown at the top right is set to 'WKSP_AADHITHYA02'.

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 SQL command entered is 'SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';'. The results tab displays a table with three columns: 'INDEX_NAME', 'TABLE_NAME', and 'UNIQUENESS'. The data row is 'EMP_DEPT_ID_IDX', 'EMPLOYEES', and 'NONUNIQUE'. The bottom left of the results panel indicates '1 rows returned in 0.04 seconds'.

INDEX_NAME	TABLE_NAME	UNIQUENESS
EMP_DEPT_ID_IDX	EMPLOYEES	NONUNIQUE

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

RESULT:

CONTROLLING USER ACCESS

EX_NO:15

DATE:

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

The CREATE SESSION system privilege

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

The CREATE TABLE privilege

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

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

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

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

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

The ALTER USER statement

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

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

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

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

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

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

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

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

SELECT table_name FROM user_tables;

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

Team 1 revokes the privilege.

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

Team 2 revokes the privilege.

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

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

Team 1 executes this INSERT statement.

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

Team 2 executes this INSERT statement.

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

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

RESULT:

PL/SQL CONTROL STRUCTURES

EX_NO:16

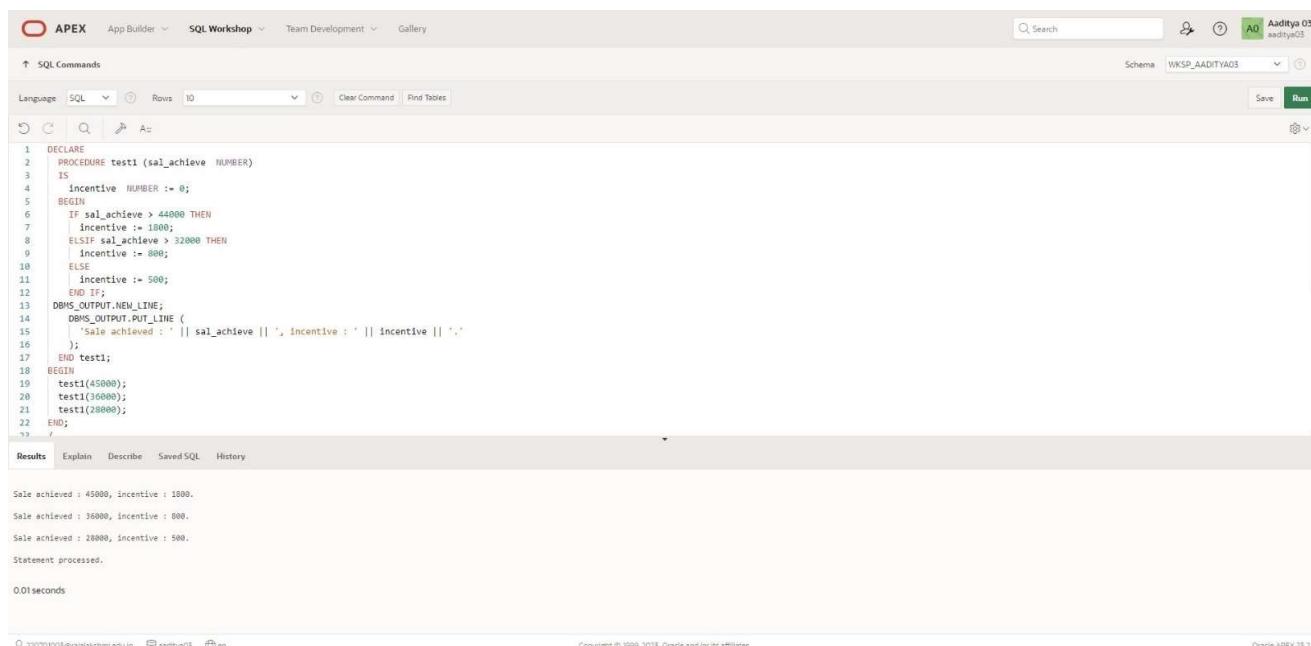
DATE:

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

QUERY:

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

OUTPUT:



```
1  DECLARE
2      PROCEDURE test1 (sal_achieve NUMBER)
3  IS
4      incentive NUMBER := 0;
5  BEGIN
6      IF sal_achieve > 44000 THEN
7          incentive := 1800;
8      ELSIF sal_achieve > 32000 THEN
9          incentive := 800;
10     ELSE
11         incentive := 500;
12     END IF;
13     DBMS_OUTPUT.NEW_LINE;
14     DBMS_OUTPUT.PUT_LINE (
15         'Sale achieved : ' || sal_achieve || ', Incentive : ' || incentive || ','
16     );
17  END test1;
18  BEGIN
19      test1(45000);
20      test1(36000);
21      test1(28000);
22  END;
23 /
```

Sale achieved : 45000, incentive : 1800.
Sale achieved : 36000, incentive : 800.
Sale achieved : 28000, incentive : 500.
Statement processed.
0.01seconds

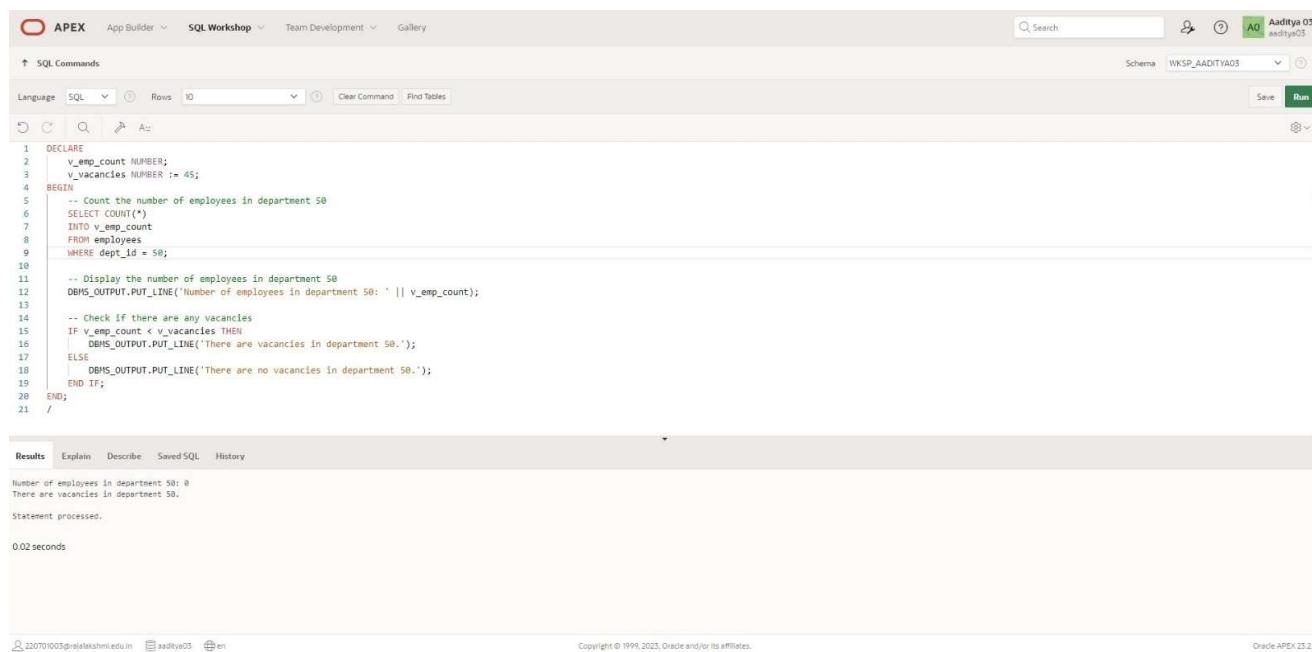
2.) Write a PL/SQL block to show an invalid case-insensitive reference to a quoted and without quoted user-defined identifier

QUERY:

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

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

OUTPUT:



```
1  DECLARE
2      v_emp_count NUMBER;
3      v_vacancies NUMBER := 45;
4  BEGIN
5      -- Count the number of employees in department 50
6      SELECT COUNT(*)
7          INTO v_emp_count
8      FROM employees
9      WHERE dept_id = 50;
10
11     -- Display the number of employees in department 50
12     DBMS_OUTPUT.PUT_LINE('Number of employees in department 50: ' || v_emp_count);
13
14     -- Check if there are any vacancies
15     IF v_emp_count < v_vacancies THEN
16         DBMS_OUTPUT.PUT_LINE('There are vacancies in department 50.');
17     ELSE
18         DBMS_OUTPUT.PUT_LINE('There are no vacancies in department 50.');
19     END IF;
20
21 END;
```

Results Explain Describe Saved SQL History

Number of employees in department 50: 8
There are vacancies in department 50.

Statement processed.

0.02 seconds

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

Oracle APEX 23.2.4

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,
    address  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 code in the editor is:

```

1  DECLARE
2    v_dept_id employees.dept_id%TYPE := 50; -- Change this to the desired department ID
3    v_dept_count NUMBER;
4    v_vacancies NUMBER := 45; -- Change this to the number of vacancies in the department
5  BEGIN
6    -- Count the number of employees in the specified department
7    SELECT COUNT(*)
8    INTO v_dept_count
9    FROM employees
10   WHERE dept_id = v_dept_id;
11
12  -- Display the count
13  DBMS_OUTPUT.PUT_LINE('Number of employees in department ' || v_dept_id || ': ' || v_dept_count);
14
15  -- Check for vacancies
16  IF v_dept_count < v_vacancies THEN
17    DBMS_OUTPUT.PUT_LINE('There are vacancies in department ' || v_dept_id || '.');
18    DBMS_OUTPUT.PUT_LINE('Number of vacancies: ' || (v_vacancies - v_dept_count));
19  ELSE
20    DBMS_OUTPUT.PUT_LINE('There are no vacancies in department ' || v_dept_id || '.');
21  END IF;
22 END;
23 /

```

The results pane shows the output of the code execution:

```

Number of employees in department 50: 2
There are vacancies in department 50.
Number of vacancies: 43
Statement processed.

0.01 seconds

```

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

QUERY:

```

CREATE OR REPLACE PROCEDURE pri_bool(
  boo_name  VARCHAR2,
  boo_val   BOOLEAN
) IS
BEGIN
  IF boo_val IS NULL THEN

```

```

DBMS_OUTPUT.PUT_LINE( boo_name || '=' NULL');
ELSIF boo_val = TRUE THEN
  DBMS_OUTPUT.PUT_LINE( boo_name || '=' TRUE');
ELSE
  DBMS_OUTPUT.PUT_LINE( boo_name || '=' FALSE');
END IF;
END;
/

```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The code in the editor is:

```

1  DECLARE
2    CURSOR employee_cursor IS
3      SELECT employee_id, first_name || ' ' || last_name AS full_name, job_id, hire_date, salary
4      FROM employee;
5  BEGIN
6    -- Loop through the cursor and display employee information
7    FOR employee_rec IN employee_cursor LOOP
8      DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
9      DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
10     DBMS_OUTPUT.PUT_LINE('Job Title: ' || employee_rec.job_id);
11     DBMS_OUTPUT.PUT_LINE('Hire Date: ' || TO_CHAR(employee_rec.hire_date, 'DD-MON-YYYY'));
12     DBMS_OUTPUT.PUT_LINE('Salary: ' || employee_rec.salary);
13     DBMS_OUTPUT.PUT_LINE('-----');
14   END LOOP;
15 END;

```

The results pane displays the output for three employees:

```

Employee ID: 101
Employee Name: Rohit Davies
Job Title: 58
Hire Date: 03-MAY-1997
Salary: 22000
-----
Employee ID: 4
Employee Name: Surya lucifer
Job Title: 55
Hire Date: 01-APR-1996
Salary: 11000
-----
Employee ID: 176
Employee Name: Virat Kohli
Job Title: 36
Hire Date: 03-APR-2000
Salary: 21000
-----
```

Statement processed.
0.02 seconds

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. In the top navigation bar, 'APEX' is selected. Below it, tabs for 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' are visible. The main area is titled 'SQL Commands'. A search bar and a schema dropdown set to 'WKSP_AADITYA03' are at the top right. The code editor contains a PL/SQL block:

```
1  DECLARE
2      CURSOR employee_cursor IS
3          SELECT a.employee_id, e.first_name || ' ' || e.last_name AS full_name, d.dept_name
4          FROM employee e
5              INNER JOIN mydept d ON e.department_id = d.deptid;
6  BEGIN
7      -- Loop through the cursor and display employee information
8      FOR employee_rec IN employee_cursor LOOP
9          DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
10         DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
11         DBMS_OUTPUT.PUT_LINE('Department Name: ' || employee_rec.dept_name);
12         DBMS_OUTPUT.PUT_LINE('-----');
13     END LOOP;
14 END;
15 /
```

The results tab shows the output of the query:

```
Employee ID: 3
Employee Name: Ajith Zicotkey
Department Name: Executive
-----
Employee ID: 50
Employee Name: Rohit Davies
Department Name: Executive
-----
Employee ID: 176
Employee Name: Vlrat Konli
Department Name: Executive
-----
Statement processed.
```

At the bottom left, the session identifier is 220701003@relakshmi.edu.in, and the workspace is Aaditya03. The bottom right corner indicates 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 interface with the SQL Workshop tab selected. The code entered is a PL/SQL block that retrieves minimum salaries for each job ID and prints them to the DBMS_OUTPUT. The results show the job ID and its corresponding minimum salary for various jobs.

```
2  CURSOR job_cursor IS
3      SELECT job_id, MIN(salary) AS min_salary
4          FROM employee
5      GROUP BY job_id;
6
7  BEGIN
8      -- Loop through the cursor and display job information
9      FOR job_rec IN job_cursor LOOP
10          DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_rec.job_id);
11          DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_rec.min_salary);
12          DBMS_OUTPUT.PUT_LINE('-----');
13      END LOOP;
14  END;
15 /
```

Results

Job ID	MinSalary
58	22000
56	21000
56	15000
55	11000
24	6000

Statement processed.
0.02 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;
END;
```

```

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 interface with the SQL Workshop tab selected. The code in the editor is:

```

1 DECLARE
2     CURSOR employee_cursor IS
3         SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.start_date
4         FROM employee e
5             JOIN job_history jh ON e.employee_id = jh.employee_id;
6 BEGIN
7     -- Loop through the cursor and display employee information
8     FOR employee_rec IN employee_cursor LOOP
9         DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
10        DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
11        DBMS_OUTPUT.PUT_LINE('Job History Start Date: ' || TO_CHAR(employee_rec.start_date, 'DD-MON-YYYY'));
12        DBMS_OUTPUT.PUT_LINE('-----');
13    END LOOP;
14 END;
15 /

```

The results pane shows the output:

```

Employee ID: 3600
Employee Name: Ajith Zilekay
Job History Start Date: 25-FEB-2006
-----
```

Statement processed.

0.02 seconds

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

QUERY:

```

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

```

```

DBMS_OUTPUT.PUT_LINE (
    'Sale achieved : ' || sal_achieve || ', incentive : ' || incentive || ')';
END test1;
BEGIN
    test1(45000);
    test1(36000);
    test1(28000);
END;
/

```

OUTPUT:

The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. The code in the editor is:

```

1 DECLARE
2     CURSOR employee_cursor IS
3         SELECT e.employee_id, e.first_name || ' ' || e.last_name AS full_name, jh.end_date
4             FROM employee e
5                 JOIN job_history jh ON e.employee_id = jh.employee_id;
6 BEGIN
7     -- Loop through the cursor and display employee information
8     FOR employee_rec IN employee_cursor LOOP
9         DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_rec.employee_id);
10        DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_rec.full_name);
11
12        -- Check if the end date is NULL (meaning the employee is currently in the job)
13        IF employee_rec.end_date IS NULL THEN
14            DBMS_OUTPUT.PUT_LINE('Job History End Date: (Still Employed)');
15        ELSE
16        END IF;
17    END LOOP;
18 END;
19 /

```

The results pane shows the output:

```

Employee ID: 3
Employee Name: Ajitn Zlotkey
Job History End Date: 23-FEB-2014
-----
Statement processed.

0.01 seconds

```

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

QUERY:

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 SQL Workshop interface. In the code editor, a PL/SQL block is written to count employees in department 50 and check for vacancies. The results panel shows the output: "The employees are in the department 50: 0. There are some vacancies in department 50." The statement was processed in 0.03 seconds.

```

1  DECLARE
2    tot_emp NUMBER;
3  BEGIN SELECT Count(*) INTO  tot_emp FROM  emp1 e join department d ON e.dept_id = d.dept_id WHERE  e.dept_id = 50;
4    dbms_output.Put_line ('The employees are in the department 50: '||To_char(tot_emp));
5    IF tot_emp >= 45 THEN
6      dbms_output.Put_line ('There are no vacancies in the department 50.');
7    ELSE
8      dbms_output.Put_line ('There are some vacancies in department 50.');
9    END IF;
10   END;

```

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 SQL Workshop interface. In the SQL Commands tab, a PL/SQL block is run. The code prints the department name and calculates the number of vacancies. The Results tab shows the output: "The employees are in the department 80 is: 6" and "There are 39 vacancies in department 80". The total execution time is 0.01 seconds.

```

10  -- cursor c_employee cursor := dbmsopen.cursor;
11  WHERE e.department_id = get_dep_id;
12  dbms_output.Put_line ('The employees are in the department'||get_dep_id|| is: '
13  ||To_char(tot_emp));
14  IF tot_emp >= 45 THEN
15  dbms_output.Put_line ('There are no vacancies in the department'||get_dep_id);
16  ELSE
17  dbms_output.Put_line ('There are'||to_char(45-tot_emp)||' vacancies in department'|| get_dep_id
18  );
19  END IF;
20  END;
21
22

```

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

QUERY:

DECLARE

```

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

```

CURSOR c_employees IS

```

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

```

BEGIN

```

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

```

```

OPEN c_employees;
FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
WHILE c_employees%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(v_employee_id || ' ' || v_full_name || ' ' || v_job_id || ' ' ||
v_hire_date || ' ' || v_salary);
    FETCH c_employees INTO v_employee_id, v_full_name, v_job_id, v_hire_date, v_salary;
END LOOP;
CLOSE c_employees;
END;
/

```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The code in the SQL Commands pane is:

```

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;
14   LOOP
15     DBMS_OUTPUT.PUT_LINE(v_employee_id || ' ' || v_full_name || ' ' || v_job_id || ' ' ||
v_hire_date || ' ' || v_salary);
16   END LOOP;
17   CLOSE c_employees;
18 END;

```

The Results pane displays the output:

Employee ID	Full Name	Job Title	Hire Date	Salary
2	Fleming Janu	ac_account	03/01/1998	60000
1	Stephen Davies	sales_rep	02/23/1996	50000
4	Chris Jones	ac_account	05/01/1998	40000
5	Cameron Brown	st_clerk	04/22/1997	80000
3	John Williams	hr_rep	02/19/1994	20000

Statement processed.
0.01 seconds

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);
  END LOOP;
END;

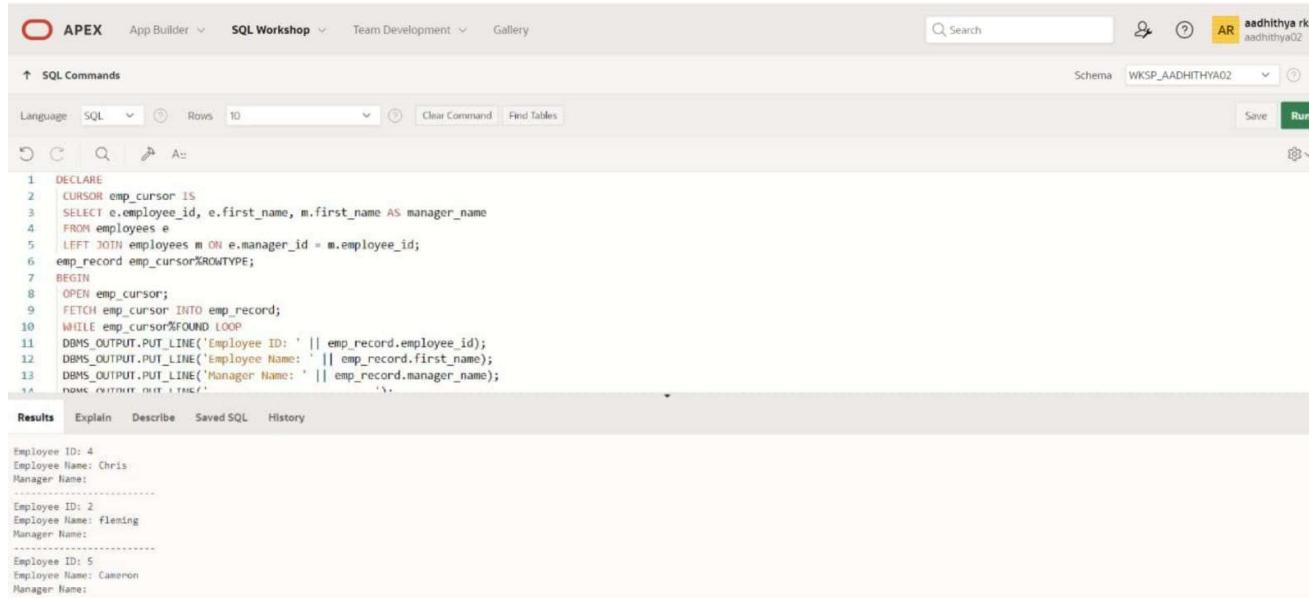
```

```

DBMS_OUTPUT.PUT_LINE('-----');
FETCH emp_cursor INTO emp_record;
END LOOP;
CLOSE emp_cursor;
END;
/

```

OUTPUT:



```

APEX App Builder SQL Workshop Team Development Gallery
Search Schema WKSP_AADHITHYA02 AR aadithya02
SQL Commands Language SQL Rows 10 Clear Command Find Tables Save Run
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);
14 DBMS_OUTPUT.PUT_LINE('');
15 END LOOP;
16 CLOSE emp_cursor;
17 END;
18 /

```

Results

```

Employee ID: 4
Employee Name: Chris
Manager Name:
-----
Employee ID: 2
Employee Name: Fleming
Manager Name:
-----
Employee ID: 5
Employee Name: Cameron
Manager Name:

```

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

QUERY:

```

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

```

OUTPUT:

```

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;
14         END LOOP;

```

Results

```

Job ID: ac_account
Minimum Salary: 40000
-----
Job ID: sales_rep
Minimum Salary: 40000
-----
Job ID: ac_account
Minimum Salary: 40000
-----
Job ID: st_clerk
Minimum Salary: 40000

```

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)

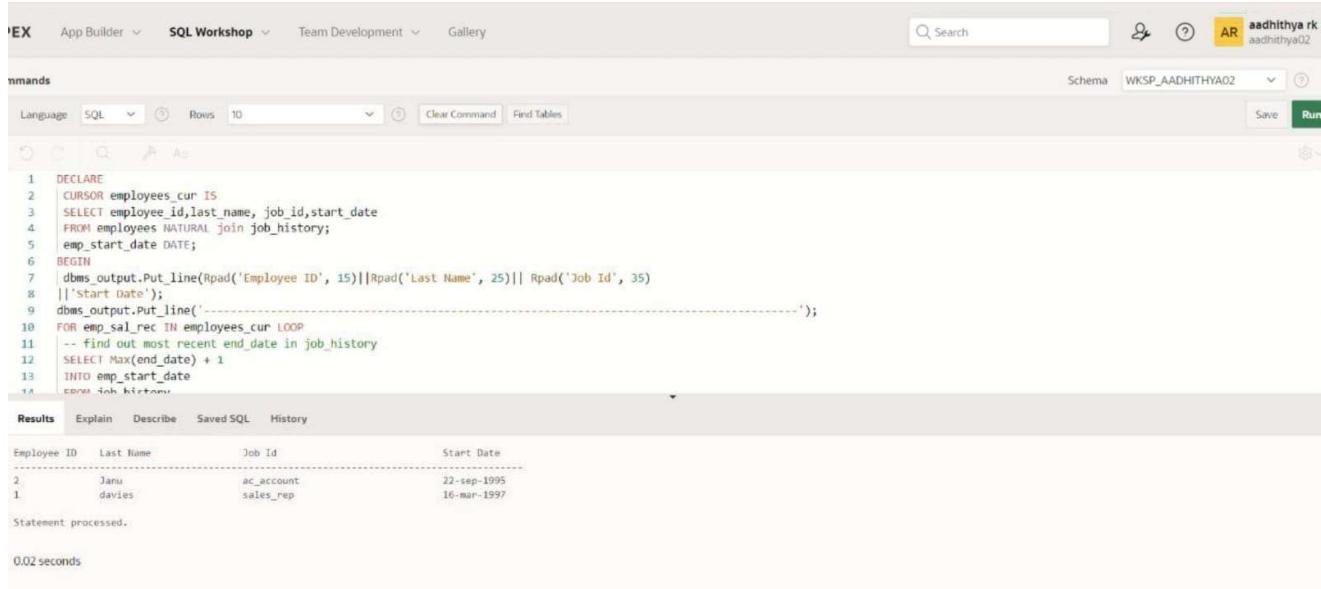
|| To_char(emp_start_date, 'dd-mon-yyyy'));

END LOOP;

END;

/

OUTPUT:



```
1  DECLARE
2  CURSOR employees_cur IS
3  SELECT employee_id, last_name, job_id, start_date
4  FROM employees NATURAL JOIN job_history;
5  emp_start_date DATE;
6  BEGIN
7  dbms_output.Put_line(Rpad('Employee ID', 15) || Rpad('Last Name', 25) || Rpad('Job Id', 35)
8  || 'Start Date');
9  dbms_output.Put_line('-----');
10 FOR emp_sal_rec IN employees_cur LOOP
11   -- find out most recent end_date in job_history
12   SELECT Max(end_date) + 1
13   INTO emp_start_date
14   FROM job_history;
15 END LOOP;
16
17 END;
```

Employee ID	Last Name	Job Id	Start Date
2	Janu	ac_account	22-sep-1995
1	davies	sales_rep	16-mar-1997

Statement processed.
0.02 seconds

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

QUERY:

DECLARE

```
v_employee_id employees.employee_id%TYPE;
v_first_name employees.last_name%TYPE;
v_end_date job_history.end_date%TYPE;
CURSOR c_employees IS
  SELECT e.employee_id, e.first_name, jh.end_date
  FROM employees e
  JOIN job_history jh ON e.employee_id = jh.employee_id;
```

BEGIN

OPEN c_employees;

FETCH c_employees INTO v_employee_id, v_first_name, v_end_date;

WHILE c_employees%FOUND LOOP

```
DBMS_OUTPUT.PUT_LINE('Employee ID: ' || v_employee_id);
DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_first_name);
DBMS_OUTPUT.PUT_LINE('End Date: ' || v_end_date);
DBMS_OUTPUT.PUT_LINE('-----');
```

```
FETCH c_employees INTO v_employee_id, v_first_name, v_end_date;
```

END LOOP;

CLOSE c_employees;

END;

OUTPUT:

APEX App Builder SQL Workshop Team Development Gallery

Search Schema WKSP_AADHITHYA02 Run

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

```
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);
15   END LOOP;
16   CLOSE c_employees;
17 END;
```

Results Explain Describe Saved SQL History

```
Employee ID: 2
Employee Name: Fleming
End Date: 09/21/1995
-----
Employee ID: 1
Employee Name: Stephen
End Date: 03/15/1997
-----
Statement 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:

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

QUERY:

```
CREATE OR REPLACE PROCEDURE get_book_info (
    p_book_id IN NUMBER,
    p_title IN OUT VARCHAR2,
```

```
    p_author OUT VARCHAR2,
    p_year_published OUT NUMBER
)
AS
BEGIN
  SELECT title, author, year_published INTO p_title, p_author, p_year_published
  FROM books
  WHERE book_id = p_book_id;

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

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

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

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

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. 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 'aadhitya rk', and a schema dropdown set to 'WKSP_AADHITHYA02'. The main area has a toolbar with Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run buttons. Below the toolbar is a code editor with the following PL/SQL script:

```
1  DECLARE
2      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;
```

Below the code editor, there are tabs for Results (selected), Explain, Describe, Saved SQL, and History. The Results tab shows the output of the executed statement:

Statement processed.
0.00 seconds

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

RESULT:

TRIGGER

EX_NO: 18

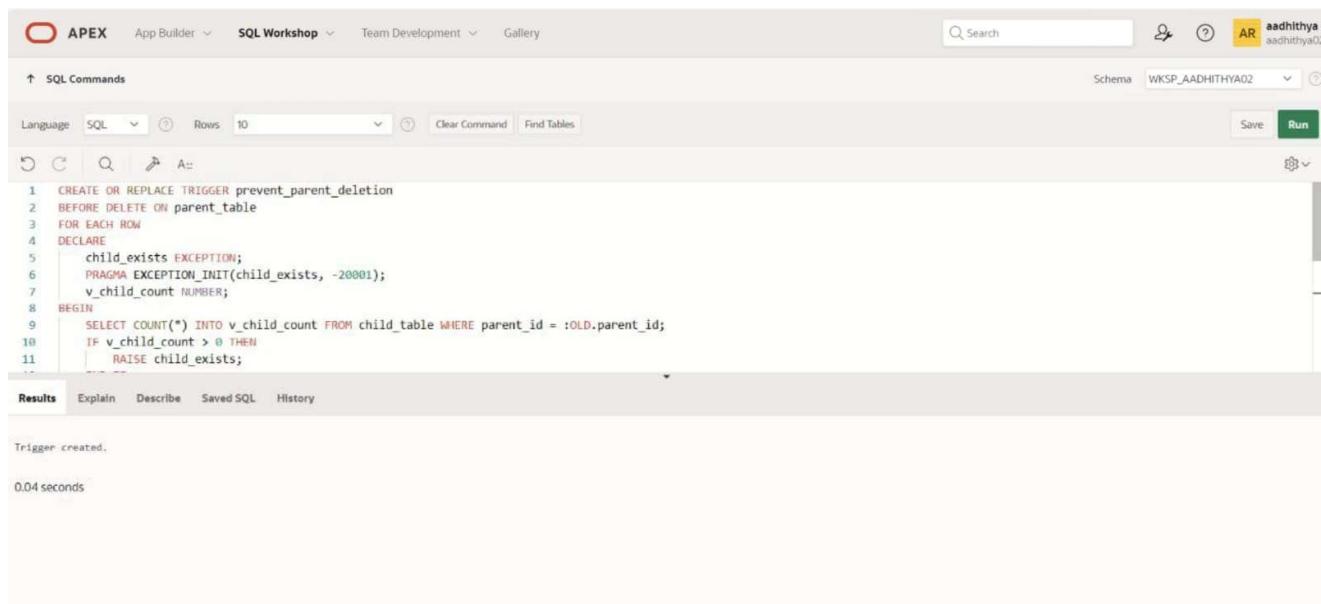
DATE:

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below the toolbar, the 'SQL Commands' tab is active. The main area contains the PL/SQL code for the trigger. At the bottom, the 'Results' tab is selected, displaying the message 'Trigger created.' and a execution time of '0.04 seconds'.

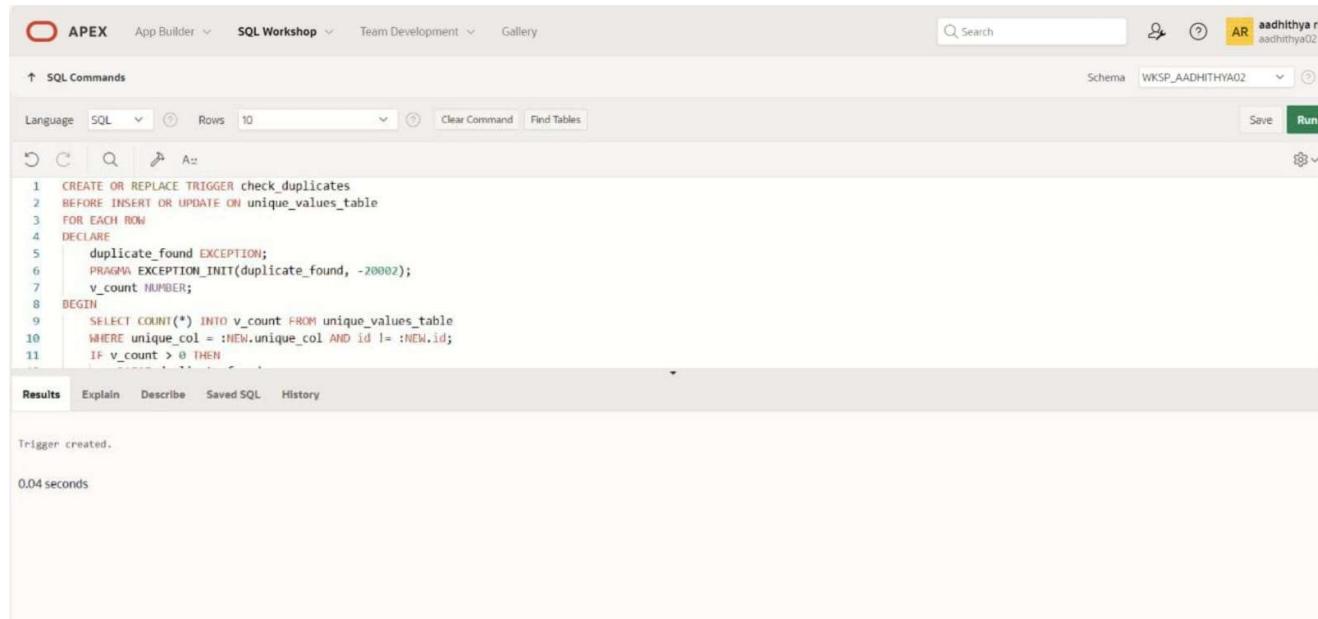
```
1 CREATE OR REPLACE TRIGGER prevent_parent_deletion
2 BEFORE DELETE ON parent_table
3 FOR EACH ROW
4 DECLARE
5     child_exists EXCEPTION;
6     PRAGMA EXCEPTION_INIT(child_exists, -20001);
7     v_child_count NUMBER;
8 BEGIN
9     SELECT COUNT(*) INTO v_child_count FROM child_table WHERE parent_id = :OLD.parent_id;
10    IF v_child_count > 0 THEN
11        RAISE child_exists;
12    END IF;
13EXCEPTION
14    WHEN child_exists THEN
15        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record while child
16records exist.');
17END;
```

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

QUERY:

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

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays the SQL command for creating the trigger 'check_duplicates'. The command is as follows:

```
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

```

Below the SQL editor, the 'Results' tab is active, showing the message "Trigger created." and a execution time of "0.04 seconds".

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

QUERY:

```

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

```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected. The main area displays the SQL command for creating the 'check_threshold' trigger. The command is as follows:

```

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 BEGIN
9     SELECT SUM(value_col) INTO v_sum FROM threshold_table;
10    v_sum := v_sum + :NEW.value_col;
11    IF v_sum > v_threshold THEN
12        RAISE threshold_exceeded;
13    END IF;
14EXCEPTION
15    WHEN threshold_exceeded THEN
16        RAISE_APPLICATION_ERROR(-20003, 'Threshold exceeded for value_col.');
17END;

```

The 'Results' tab at the bottom shows the output: "Trigger created." and "0.05 seconds".

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

QUERY:

```

CREATE OR REPLACE TRIGGER log_changes
AFTER UPDATE ON main_table
FOR EACH ROW
BEGIN

```

```

INSERT INTO audit_table (audit_id, changed_id, old_col1, new_col1, old_col2,
new_col2, change_time)
VALUES (audit_seq.NEXTVAL, :OLD.id, :OLD.col1, :NEW.col1, :OLD.col2,
:NEW.col2, SYSTIMESTAMP);
END;

```

OUTPUT:

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

```

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

```

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

```

Trigger created.

0.05 seconds

```

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

QUERY:

```

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

```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'SQL Workshop'. The main area contains the following PL/SQL code:

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

Below the code, the 'Results' tab is selected, showing the output: "Trigger created." and "0.05 seconds".

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 SQL Workshop interface. In the top navigation bar, 'APEX' is selected. Below it, tabs for 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' are visible. On the right side, a user profile for 'aadithya' is shown with the identifier 'AR_aadithya02'. The main workspace is titled 'SQL Commands' and 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, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing the output: 'Trigger created.' and '0.02 seconds'. There is also a large, empty text area below the results.

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:

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 10 Clear Command Find Tables

Save Run

```
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;
```

Results Explain Describe Saved SQL History

Trigger created.

0.04 seconds

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

RESULT:

MONGO DB

EX_NO: 19

DATE:

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

QUERY:

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

OUTPUT:

```
Aadhithya_02> db.restaurants.find({$and:[{ $or: [{cuisine: { $nin: ["American", "Chinese"] }}, {name: /Wil/}]}]}, {restaurant_id: 1, name: 1, borough: 1, cuisine: 1})
[
  {
    _id: ObjectId('6650a544c5dcab95e3cdcdf6'),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
Aadhithya_02> |
```

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:

```
Aadhithya_02> db.restaurants.find( { "grades": { $elemMatch: { "grade": "A", "score": 11, "date": ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 })
Aadhithya_02>
```

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:

```
Aadhithya_02> 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 } )
|Aadhithya_02>
```

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:

```
Bhanupriya_40> db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, { _id:0, restaurant_id:1, name:1, address:1})
Bhanupriya_40>
```

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:

```
|Aadhithya_02> db.restaurants.find({}, { _id: 0 }).sort({ name: 1 })
[
  {
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '38875445'
  }
]
|Aadhithya_02>
```

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:

```
|Aadhithya_02> db.restaurants.find({}, { _id: 0 }).sort({ name: -1 })
[
  {
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '38875445'
  }
]
|Aadhithya_02>
```

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:

```
|Aadhithya_02> db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })  
[  
  {  
    _id: ObjectId('6650a544c5dcab95e3cdcf6'),  
    address: {  
      building: '1007',  
      coord: [-73.856077, 40.848447],  
      street: 'Morris Park Ave',  
      zipcode: '10462'  
    },  
    borough: 'Bronx',  
    cuisine: 'Bakery',  
    grades: [  
      {  
        date: ISODate('2014-03-03T00:00:00.000Z'),  
        grade: 'A',  
        score: 2  
      },  
      {  
        date: ISODate('2013-09-11T00:00:00.000Z'),  
        grade: 'A',  
        score: 6  
      },  
      {  
        date: ISODate('2013-01-24T00:00:00.000Z'),  
        grade: 'A',  
        score: 10  
      },  
      {  
        date: ISODate('2011-11-23T00:00:00.000Z'),  
        grade: 'A',  
        score: 9  
      },  
      {  
        date: ISODate('2011-03-10T00:00:00.000Z'),  
        grade: 'B',  
        score: 14  
      }  
    ],  
    name: 'Morris Park Bake Shop',  
    restaurant_id: '38075445'  
  }  
]  
Aadhithya_02>
```

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:

```
|Aadhithya_02> db.restaurants.find({ "address.street": { $exists: true, $ne: "" } })  
[  
  {  
    _id: ObjectId('6650a544c5dcab95e3cdcf6'),  
    address: {  
      building: '1007',  
      coord: [-73.856077, 40.848447],  
      street: 'Morris Park Ave',  
      zipcode: '10462'  
    },  
    borough: 'Bronx',  
    cuisine: 'Bakery',  
    grades: [  
      {  
        date: ISODate('2014-03-03T00:00:00.000Z'),  
        grade: 'A',  
        score: 2  
      },  
      {  
        date: ISODate('2013-09-11T00:00:00.000Z'),  
        grade: 'A',  
        score: 6  
      },  
      {  
        date: ISODate('2013-01-24T00:00:00.000Z'),  
        grade: 'A',  
        score: 10  
      },  
      {  
        date: ISODate('2011-11-23T00:00:00.000Z'),  
        grade: 'A',  
        score: 9  
      },  
      {  
        date: ISODate('2011-03-10T00:00:00.000Z'),  
        grade: 'B',  
        score: 14  
      }  
    ],  
    name: 'Morris Park Bake Shop',  
    restaurant_id: '38075445'  
  }  
]  
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
[
  {
    _id: ObjectId('6650a544c5dcab95e3cdcdf6'),
    address: {
      building: '1087',
      coord: [ -73.856877, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
```

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:

```
Aadhithya_02> db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 })
[
  {
    _id: ObjectId('6650a544c5dcab95e3cdcdf6'),
    grades: [
      {
        date: ISODate('2014-03-03T00:00:00.000Z'),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate('2013-09-11T00:00:00.000Z'),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate('2013-01-24T00:00:00.000Z'),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate('2011-11-23T00:00:00.000Z'),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate('2011-03-10T00:00:00.000Z'),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]
```

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:

```
Aadhithya_02> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })  
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
{
  "_id": ObjectId("6650a544c5dcab95e3cdcdf6"),
  "address": {
    "building": "1087",
    "coord": [-73.856877, 40.848447],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    {
      "date": ISODate("2014-03-03T00:00:00.000Z"),
      "grade": "A",
      "score": 2
    },
    {
      "date": ISODate("2013-09-11T00:00:00.000Z"),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate("2013-01-24T00:00:00.000Z"),
      "grade": "A",
      "score": 10
    },
    {
      "date": ISODate("2011-11-23T00:00:00.000Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2011-03-10T00:00:00.000Z"),
      "grade": "B",
      "score": 14
    }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })  
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })  
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
{
  "_id": ObjectId('6650a544c5dcab95e3cdcf6'),
  "address": {
    "building": '1007',
    "coord": [-73.856977, 40.848447],
    "street": 'Morris Park Ave',
    "zipcode": '10462'
  },
  "borough": 'Bronx',
  "cuisine": 'Bakery',
  "grades": [
    {
      "date": ISODate('2014-03-03T00:00:00.000Z'),
      "grade": 'A',
      "score": 2
    },
    {
      "date": ISODate('2013-09-11T00:00:00.000Z'),
      "grade": 'A',
      "score": 6
    },
    {
      "date": ISODate('2013-01-24T00:00:00.000Z'),
      "grade": 'A',
      "score": 10
    },
    {
      "date": ISODate('2011-11-23T00:00:00.000Z'),
      "grade": 'A',
      "score": 9
    },
    {
      "date": ISODate('2011-03-10T00:00:00.000Z'),
      "grade": 'B',
      "score": 14
    }
  ],
  "name": 'Morris Park Bake Shop',
  "restaurant_id": '38075445'
}
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

Aadhithya_02>

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:

```
Aadhithya_02> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

Aadhithya_02>

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:

```
Aadhithya_02> db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

```
Aadhithya_02>
```

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:

```
Aadhithya_02> 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"] } })
```

```
Aadhithya_02>
```

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:

```
Aadhithya_02> db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
{
  "_id": ObjectId("6658a544c5dcab95e3cdcf6"),
  "address": {
    "building": "1807",
    "coord": [-73.856677, 40.848447],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    {
      "date": ISODate('2014-03-03T00:00:00.000Z'),
      "grade": "A",
      "score": 2
    },
    {
      "date": ISODate('2013-09-11T00:00:00.000Z'),
      "grade": "A",
      "score": 6
    },
    {
      "date": ISODate('2013-01-24T00:00:00.000Z'),
      "grade": "A",
      "score": 10
    },
    {
      "date": ISODate('2011-11-23T00:00:00.000Z'),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate('2011-03-10T00:00:00.000Z'),
      "grade": "B",
      "score": 14
    }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
Aadhithya_02>
```

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 year.

```
Aadhithya_02>db.movies.find({ year: 1893 })  
Aadhithya_02>
```

- 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:

```
Aadhithya_02> db.movies.find({ runtime: { $gt: 120 } })  
Aadhithya_02>
```

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

QUERY:

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

OUTPUT:

```
Aadhithya_02> db.movies.find({ genres: 'Short' })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      'Gilbert M. 'Broncho Billy' Anderson',
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYYNzQ0MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQzI@._V1_SV1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Aadhithya_02>
```

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:

```
Aadhithya_02> db.movies.find({ genres: 'Short' })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      'Gilbert M. 'Broncho Billy' Anderson',
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYYNzQ0MDVmLWIwYjgtMmYwYWIxZDYyNzU2XkEyXkFqcGdeQXVyNzQzNzQzI@._V1_SV1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  }
]
Aadhithya_02>
```

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:

```
Aadhithya_02> db.movies.find({ directors: 'William K.L. Dickson' })  
Aadhithya_02>
```

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

QUERY:

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

OUTPUT:

```
Aadhithya_02> db.movies.find({ directors: 'William K.L. Dickson' })  
Aadhithya_02>
```

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:

```
Aadhithya_02> db.movies.find({ directors: 'William K.L. Dickson' })  
Aadhithya_02>
```

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

```
Aadhithya_02> db.movies.find({ 'imdb.rating': { $gt: 7 } })  
[  
  {  
    _id: ObjectId('573a1390f29313caabcd42e6'),  
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',  
    genres: [ 'Short', 'Western' ],  
    runtime: 11,  
    cast: [  
      'A.C. Abadie',  
      'Gilbert M. 'Broncho Billy' Anderson',  
      'George Barnes',  
      'Justus D. Barnes',  
    ],  
    poster: 'https://m.media-amazon.com/images/M/MVSBNTU3NjE5NzYtYTYYN500MDVmLWIwYjgtNmYwYNIxZDYyNzU2XhEyXkFqcGdeQXVyNzQzNzQzI@._V1_SY1000_SX677_AL_.jpg',  
    title: 'The Great Train Robbery',  
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",  
    languages: [ 'English' ],  
    released: ISODate('1903-12-01T00:00:00.000Z'),  
    directors: [ 'Edwin S. Porter' ],  
    rated: 'TV-G',  
    awards: { wins: 1, nominations: 0, text: '1 win.' },  
    lastupdated: '2015-08-13 00:27:59.177000000',  
    year: 1903,  
    imdb: { rating: 7.4, votes: 9847, id: 439 },  
    countries: [ 'USA' ],  
    type: 'movie',  
    tomatoes:  
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },  
      fresh: 6,  
      critic: { rating: 7.6, numReviews: 6, meter: 100 },  
      rotten: 0,  
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')  
    }  
  ]  
Aadhithya_02>
```

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:

```
Aadhithya_02> db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })  
Aadhithya_02>
```

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

QUERY:

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

OUTPUT:

```
Aadhithya_02> db.movies.find({ 'awards.wins': { $gt: 0 } })  
[  
  {  
    _id: ObjectId('573a1390f20313caabcd42e8'),  
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',  
    genres: ['Short', 'Western'],  
    runtime: 11,  
    cast: [  
      'A.C. Abadie',  
      'Gilbert M. 'Broncho Billy' Anderson',  
      'George Barnes',  
      'Justus D. Barnes'  
    ],  
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTYYN500MDVmLWIwYjgtMmYwYIxZDYyNzU2XxEyXiFqcGdeQXVyNzQzNzI@._V1_SV1000_SX677_AL_.jpg',  
    fullplot: "Among the earliest existing films in American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hold up a train and rob the passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",  
    languages: ['English'],  
    released: ISODate('1903-12-01T00:00:00.000Z'),  
    directors: [ 'Edwin S. Porter' ],  
    rated: 'TV-G',  
    awards: { wins: 1, nominations: 0, text: '1 win.' },  
    lastupdated: '2015-08-13 08:27:59.177000000',  
    year: 1903,  
    imbd: { rating: 7.4, votes: 9847, id: 439 },  
    countries: [ 'USA' ],  
    type: 'movie',  
    tomatoes: {  
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },  
      fresh: 6,  
      critic: { rating: 7.6, numReviews: 6, meter: 100 },  
      rotten: 0,  
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')  
    }  
  }  
]  
Aadhithya_02> |
```

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:

```
Aadhithya_02> 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 }
... )
```

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:

```
Aadhithya_02> 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 }
... )
```

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:

```
Aadhithya_02> db.movies.find(
...   { released: ISODate("1893-05-09T00:00:00.000Z") },
...   { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 }
... )
Aadhithya_02>
```

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:

```
Aadhithya_02> db.movies.find(
...   { title: /scene/i },
...   { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 }
... )
Aadhithya_02>
```

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

RESULT: