Hindi Vidya Prachar Samiti's

# RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE & COMMERCE (AUTONOMUS)

## Advanced Database Management System


**Name:** Richard Caleb

**Roll No.:** 735

**Class:** MSc Data Science and Artificial Intelligence Part-I

**Ramniranjan Jhunjhunwala College of Arts, Science and Commerce**

## Department of Data Science and Artificial Intelligence

# CERTIFICATE

This is to certify Mast. Richard Caleb of MSc. Data Science and Artificial Intelligence, Roll no. 735 has successfully completed the practical of Advanced Database Management System during the Academic Year 2023-2024.

Date:

Prof. Kajal Shirke
(Prof-In-Charge)

External Examiner

| INDEX | | | |
|---|---|---|---|
| Sr. No | Practical Name | Date | Signature |
| 1. | SQL Statements-I | | |
| 2. | SQL Statements-II | | |
| 3. | Manipulating Data | | |
| 4. | Creating and Managing Tables | | |
| 5. | Creating and Managing other Databases objects | | |
| 6. | Using SET operators, Date/Time Fn. GROUP BY clause (advanced features) and advanced subqueries | | |
| 7. | PL/SQL Basic | | |
| 8. | Composite data types, cursors and exceptions | | |
| 9. | Procedures and Functions | | |
| 10. | MongoDB | | |
| 11. | XML (Displaying, Update, Deleting record from table) | | |
| 12. | XML (Retrieve, Updating, Deleting record from table) | | |

# Practical-I Date:

## AIM: SQL Statements-I

### a) Writing Basic SQL SELECT Statements

Before Starting with the SELECT Statements

**PRE-REQUISTIC :**

Create database using CREATE <db name> command and use it with USE <db_name> command.

Then Create Table using CREATE TABLE <table_name> (attribute name data type,range,constraint….)

```
mysql> create database org;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| org                |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql> use org;
Database changed
mysql> use org;
Database changed
mysql> create table Worker(W_Id int not null primary key auto_increment, First_Name char(25), Last_Name char(25), Salary int(15),Join
ing_Date DateTime, Dept char(25));
Query OK, 0 rows affected, 1 warning (0.03 sec)
```

After initializing the table, insert the values into it

```
mysql> Insert into Worker(W_Id, First_Name, Last_Name, Salary, Joining_Date, Dept) Values (001,'Monica','Arora',100000,'21-02-20 09.0
0.00','HR'),(002,'Niharika','Verma',80000,'21-06-11 09.00.00','Admin'),(003,'Vishal','Singh',300000,'21-02-20 09.00.00','HR'),(004,'V
ivek','Bhati',5000000,'21-02-20 09.00.00','Admin'),(005,'Satish','Kumar',750000,'21-01-20 09.00.00','Account');
Query OK, 5 rows affected, 5 warnings (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 5
```

```
mysql> Insert into Bonus(W_Ref_Id,Bonus_Amt,Bonus_Date) values (001,5000,'23-02-20'),(002,3000,'23-06-11'),(003,4000,'23-02-20'),(001
,4500,'23-02-20'),(002,3500,'23-06-11');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

**BASIC SELECT STATEMENT**

Retriving all the rows with the help of SELECT STATEMENT

```
mysql> select * from Bonus;
+----------+-----------+---------------------+
| W_Ref_Id | Bonus_Amt | Bonus_Date          |
+----------+-----------+---------------------+
|        1 |      5000 | 2023-02-20 00:00:00 |
|        2 |      3000 | 2023-06-11 00:00:00 |
|        3 |      4000 | 2023-02-20 00:00:00 |
|        1 |      4500 | 2023-02-20 00:00:00 |
|        2 |      3500 | 2023-06-11 00:00:00 |
+----------+-----------+---------------------+
5 rows in set (0.00 sec)
```

DISTINCT keyword with SELECT STATEMENT

```
mysql> select distinct Dept from worker;
+---------+
| Dept    |
+---------+
| HR      |
| Admin   |
| Account |
+---------+
3 rows in set (0.00 sec)
```

Retriving data of particular column with SELECT STATEMENT

```
mysql> select First_Name,Last_Name from worker;
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| Monica     | Arora     |
| Niharika   | Verma     |
| Vishal     | Singh     |
| Vivek      | Bhati     |
| Satish     | Kumar     |
+------------+-----------+
5 rows in set (0.00 sec)
```

## b) Restricting and Sorting data

Retrieving only desired data using WHERE CLAUSE and LIKE operator and SORTING DATA

```
mysql> select * from worker where First_Name like '%a%' order by First_Name;
+------+------------+-----------+--------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary | Joining_Date        | Dept    |
+------+------------+-----------+--------+---------------------+---------+
|    1 | Monica     | Arora     | 100000 | 2021-02-20 09:00:00 | HR      |
|    2 | Niharika   | Verma     |  80000 | 2021-06-11 09:00:00 | Admin   |
|    5 | Satish     | Kumar     | 750000 | 2021-01-20 09:00:00 | Account |
|    3 | Vishal     | Singh     | 300000 | 2021-02-20 09:00:00 | HR      |
+------+------------+-----------+--------+---------------------+---------+
4 rows in set (0.00 sec)
```

Retrieving only desired data using WHERE CLAUSE and RELATIONAL operator and getting the result in SORTED manner

```
mysql> select * from worker where salary > 100000 order by Dept Desc;
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    3 | Vishal     | Singh     |  300000 | 2021-02-20 09:00:00 | HR      |
|    4 | Vivek      | Bhati     | 5000000 | 2021-02-20 09:00:00 | Admin   |
|    5 | Satish     | Kumar     |  750000 | 2021-01-20 09:00:00 | Account |
+------+------------+-----------+---------+---------------------+---------+
3 rows in set (0.01 sec)
```

Retrieving data in sorted manner

```
mysql> select * from worker order by Salary;
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    2 | Niharika   | Verma     |   80000 | 2021-06-11 09:00:00 | Admin   |
|    1 | Monica     | Arora     |  100000 | 2021-02-20 09:00:00 | HR      |
|    3 | Vishal     | Singh     |  300000 | 2021-02-20 09:00:00 | HR      |
|    5 | Satish     | Kumar     |  750000 | 2021-01-20 09:00:00 | Account |
|    4 | Vivek      | Bhati     | 5000000 | 2021-02-20 09:00:00 | Admin   |
+------+------------+-----------+---------+---------------------+---------+
5 rows in set (0.00 sec)
```

**c) Single Row Function**

Displaying the required data in UPPER_CASE

```
mysql> select upper(First_Name) from worker;
+--------------------+
| upper(First_Name) |
+--------------------+
| MONICA             |
| NIHARIKA           |
| VISHAL             |
| VIVEK              |
| SATISH             |
+--------------------+
5 rows in set (0.01 sec)
```

Displaying the SUBSTRING of required length

```
mysql> select substr(last_name,1,3) as 'sliced' from worker;
+--------+
| sliced |
+--------+
| Aro    |
| Ver    |
| Sin    |
| Bha    |
| Kum    |
+--------+
5 rows in set (0.00 sec)
```

Concating two different column values as one and setting in a table with user given alias name

```
mysql> select concat(first_name,last_name) 'Full Name' from worker;
+---------------+
| Full Name     |
+---------------+
| MonicaArora   |
| NiharikaVerma |
| VishalSingh   |
| VivekBhati    |
| SatishKumar   |
+---------------+
5 rows in set (0.00 sec)
```

# Practical-II

Date:

## AIM: SQL Statements-II

### a) Displaying Data from Multiple Table

Displaying common data from 2 different table using inner join

```
mysql> SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE FROM Worker W INNER JOIN Title T ON W.W_ID = T.WORKER_REF_ID
 SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE FROM Worker W INNER JOIN Title T ON W.W_ID = T.WORKER_REF_ID AND T.WORKER_TITLE in ('Ma
nager');
+------------+--------------+
| FIRST_NAME | WORKER_TITLE |
+------------+--------------+
| Monica     | Manager      |
| Satish     | Manager      |
+------------+--------------+
2 rows in set (0.00 sec)
```

Displaying data Outer Join

```
mysql> SELECT worker.First_Name, Title.Worker_Title FROM Worker LEFT OUTER JOIN Title ON Worker.W_Id=Title.Worker_Ref_Id;
+------------+---------------+
| First_Name | Worker_Title  |
+------------+---------------+
| Monica     | Manager       |
| Monica     | Manager       |
| Niharika   | Executive     |
| Niharika   | Executive     |
| Vishal     | Lead          |
| Vishal     | Lead          |
| Vivek      | Asst. Manager |
| Vivek      | Asst. Manager |
| Satish     | Manager       |
| Satish     | Manager       |
+------------+---------------+
10 rows in set (0.00 sec)
```

### b) Aggregating Data Using Group Functions

Using COUNT function along with GROUP BY

```
mysql> SELECT DEPT, count(W_ID) No_Of_Workers FROM worker GROUP BY DEPT;
+---------+---------------+
| DEPT    | No_Of_Workers |
+---------+---------------+
| HR      |             2 |
| Admin   |             2 |
| Account |             1 |
+---------+---------------+
3 rows in set (0.00 sec)
```

Using MAX function to get maximize from the table, along with GROUP BY

```
mysql> SELECT dept, MAX(salary) FROM worker GROUP BY dept;
+---------+-------------+
| dept    | MAX(salary) |
+---------+-------------+
| HR      |      300000 |
| Admin   |     5000000 |
| Account |      750000 |
+---------+-------------+
3 rows in set (0.00 sec)
```

Using AVG function to get average from the int datatype column along with GROUP BY

```
mysql> SELECT dept, avg(salary) FROM worker GROUP BY dept;
+---------+--------------+
| dept    | avg(salary)  |
+---------+--------------+
| HR      |  200000.0000 |
| Admin   | 2540000.0000 |
| Account |  750000.0000 |
+---------+--------------+
3 rows in set (0.01 sec)
```

## c) Subqueries

Query within in a Query to get desired data

```
mysql> SELECT * FROM worker WHERE W_ID IN (SELECT W_ID FROM worker WHERE SALARY > 450000);
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    4 | Vivek      | Bhati     | 5000000 | 2021-02-20 09:00:00 | Admin   |
|    5 | Satish     | Kumar     |  750000 | 2021-01-20 09:00:00 | Account |
+------+------------+-----------+---------+---------------------+---------+
2 rows in set (0.01 sec)
```

# Practical-III

Date:

## AIM: Manipulating Data

### a) Using INSERT statement

INSERT values into newly created TABLE

```
mysql> INSERT INTO Worker (W_ID, FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE, DEPT) VALUES (006, 'Vipul', 'Diwan', 200000, '21-06-11
09.00.00', 'Account'),(007, 'Satish', 'Kumar', 75000, '21-01-20 09.00.00', 'Account'),(008, 'Geetika', 'Chauhan', 90000, '21-04-11 09
.00.00', 'Admin');
Query OK, 3 rows affected, 3 warnings (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 3

mysql> select * from worker;
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    1 | Monica     | Arora     |  100000 | 2021-02-20 09:00:00 | HR      |
|    2 | Niharika   | Verma     |   80000 | 2021-06-11 09:00:00 | Admin   |
|    3 | Vishal     | Singh     |  300000 | 2021-02-20 09:00:00 | HR      |
|    4 | Vivek      | Bhati     | 5000000 | 2021-02-20 09:00:00 | Admin   |
|    5 | Satish     | Kumar     |  750000 | 2021-01-20 09:00:00 | Account |
|    6 | Vipul      | Diwan     |  200000 | 2021-06-11 09:00:00 | Account |
|    7 | Satish     | Kumar     |   75000 | 2021-01-20 09:00:00 | Account |
|    8 | Geetika    | Chauhan   |   90000 | 2021-04-11 09:00:00 | Admin   |
+------+------------+-----------+---------+---------------------+---------+
8 rows in set (0.00 sec)
```

### b) Using DELETE statement

DELETE the whole table from the database

```
mysql> delete from title;
Query OK, 10 rows affected (0.01 sec)

mysql> select * from title;
Empty set (0.00 sec)
```

DELETING the particular from the table with the help of DELETE AND WHERE

```
mysql> DELETE FROM worker WHERE first_name='Geetika';
Query OK, 1 row affected (0.01 sec)

mysql> select * from worker;
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    1 | Monica     | Arora     |  100000 | 2021-02-20 09:00:00 | HR      |
|    2 | Niharika   | Verma     |   80000 | 2021-06-11 09:00:00 | Admin   |
|    3 | Vishal     | Singh     |  300000 | 2021-02-20 09:00:00 | HR      |
|    4 | Vivek      | Bhati     | 5000000 | 2021-02-20 09:00:00 | Admin   |
|    5 | Satish     | Kumar     |  750000 | 2021-01-20 09:00:00 | Account |
|    6 | Vipul      | Diwan     |  200000 | 2021-06-11 09:00:00 | Account |
|    7 | Satish     | Kumar     |   75000 | 2021-01-20 09:00:00 | Account |
+------+------------+-----------+---------+---------------------+---------+
7 rows in set (0.00 sec)
```

## c) Using UPDATE statement

Incrementing Salary of each individual with the help of UPDATE

```
mysql> UPDATE worker SET SALARY = SALARY+3000;
Query OK, 7 rows affected (0.01 sec)
Rows matched: 7  Changed: 7  Warnings: 0

mysql> select * from worker;
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    1 | Monica     | Arora     |  103000 | 2021-02-20 09:00:00 | HR      |
|    2 | Niharika   | Verma     |   83000 | 2021-06-11 09:00:00 | Admin   |
|    3 | Vishal     | Singh     |  303000 | 2021-02-20 09:00:00 | HR      |
|    4 | Vivek      | Bhati     | 5003000 | 2021-02-20 09:00:00 | Admin   |
|    5 | Satish     | Kumar     |  753000 | 2021-01-20 09:00:00 | Account |
|    6 | Vipul      | Diwan     |  203000 | 2021-06-11 09:00:00 | Account |
|    7 | Satish     | Kumar     |   78000 | 2021-01-20 09:00:00 | HR      |
+------+------------+-----------+---------+---------------------+---------+
7 rows in set (0.00 sec)
```

UPDATING particular's position

```
mysql> UPDATE worker SET Dept = 'HR' WHERE W_ID = 7;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from worker;
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    1 | Monica     | Arora     |  100000 | 2021-02-20 09:00:00 | HR      |
|    2 | Niharika   | Verma     |   80000 | 2021-06-11 09:00:00 | Admin   |
|    3 | Vishal     | Singh     |  300000 | 2021-02-20 09:00:00 | HR      |
|    4 | Vivek      | Bhati     | 5000000 | 2021-02-20 09:00:00 | Admin   |
|    5 | Satish     | Kumar     |  750000 | 2021-01-20 09:00:00 | Account |
|    6 | Vipul      | Diwan     |  200000 | 2021-06-11 09:00:00 | Account |
|    7 | Satish     | Kumar     |   75000 | 2021-01-20 09:00:00 | HR      |
+------+------------+-----------+---------+---------------------+---------+
7 rows in set (0.00 sec)
```

# Practical-IV

Date:

## AIM: Creating and Managing Tables

### a) Creating and Managing Tables

Creating a TABLE with one name and then after ALTERING with another name.

```
mysql> CREATE TABLE Title (WORKER_REF_ID INT,WORKER_TITLE CHAR(25),AFFECTED_FROM DATETIME,FOREIGN KEY (WORKER_REF_ID)REFERENCES Worke
r(W_ID) ON DELETE CASCADE);
Query OK, 0 rows affected (0.02 sec)

mysql> ALTER TABLE 'title' RENAME TO 'Desgn';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the righ
t syntax to use near ''title' RENAME TO 'Desgn'' at line 1
mysql> ALTER TABLE title RENAME TO Desgn;
Query OK, 0 rows affected (0.02 sec)
```

Altering the TABLE by adding new COLUMN and then deleting the same new COLUMN from the TABLE

```
mysql> ALTER TABLE desgn ADD COLUMN borough varchar(20);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE desgn DROP COLUMN borough;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

### b) Including Constraints

Creating Table including Primary Constraint

```
mysql> desc worker;
+--------------+----------+------+-----+---------+----------------+
| Field        | Type     | Null | Key | Default | Extra          |
+--------------+----------+------+-----+---------+----------------+
| W_Id         | int      | NO   | PRI | NULL    | auto_increment |
| First_Name   | char(25) | YES  |     | NULL    |                |
| Last_Name    | char(25) | YES  |     | NULL    |                |
| Salary       | int      | YES  |     | NULL    |                |
| Joining_Date | datetime | YES  |     | NULL    |                |
| Dept         | char(25) | YES  |     | NULL    |                |
+--------------+----------+------+-----+---------+----------------+
6 rows in set (0.02 sec)
```

```
mysql> insert into (W_Id,First_Name,Last_Name,Salary,Joining_Date,Dept) values ('James','Harden',1200000,'22-03-11 09.00.00','Tech');
                                                                                                                              E
RROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the righh
t syntax to use near '(W_Id,First_Name,Last_Name,Salary,Joining_Date,Dept) values ('James','Harden',12' at line 1
mysql> insert into (W_Id,First_Name,Last_Name,Salary,Joining_Date,Dept) values ('James','Harden',1200000,'22-03-11 09.00.00','Tech');
```

Foreign Key Constraint

```
mysql> desc desgn;
+---------------+----------+------+-----+---------+-------+
| Field         | Type     | Null | Key | Default | Extra |
+---------------+----------+------+-----+---------+-------+
| WORKER_REF_ID | int      | YES  | MUL | NULL    |       |
| WORKER_TITLE  | char(25) | YES  |     | NULL    |       |
| AFFECTED_FROM | datetime | YES  |     | NULL    |       |
+---------------+----------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> delete from worker where W_Id=3;
Query OK, 1 row affected (0.01 sec)

mysql> select * from bonus;
+----------+-----------+---------------------+
| W_Ref_Id | Bonus_Amt | Bonus_Date          |
+----------+-----------+---------------------+
|        1 |      5000 | 2023-02-20 00:00:00 |
|        2 |      3000 | 2023-06-11 00:00:00 |
|        1 |      4500 | 2023-02-20 00:00:00 |
|        2 |      3500 | 2023-06-11 00:00:00 |
+----------+-----------+---------------------+
4 rows in set (0.00 sec)
```

# Practical-V

Date:

## AIM: Creating and Managing other database objects

### a) Creating Views

Creating View from existing table based on come condition

```
mysql> create view new_worker as select First_Name,Dept,Joining_Date from worker where W_Id>1;
Query OK, 0 rows affected (0.01 sec)
```

Displaying Values from View

```
mysql> select * from new_worker;
+------------+---------+---------------------+
| First_Name | Dept    | Joining_Date        |
+------------+---------+---------------------+
| Niharika   | Admin   | 2021-06-11 09:00:00 |
| Vivek      | Admin   | 2021-02-20 09:00:00 |
| Satish     | Account | 2021-01-20 09:00:00 |
| Vipul      | Account | 2021-06-11 09:00:00 |
| Satish     | HR      | 2021-01-20 09:00:00 |
+------------+---------+---------------------+
5 rows in set (0.00 sec)
```

Dropping the created View

```
mysql> drop view new_worker;
Query OK, 0 rows affected (0.01 sec)
```

### b) Other Database Objects

Creating index on particular column

```
mysql> create index First_Name on worker(First_Name);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Displaying Index

```
mysql> show indexes from worker;
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+--
----------+---------+---------------+---------+------------+
| Table  | Non_unique | Key_name   | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | I
ndex_type | Comment | Index_comment | Visible | Expression |
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+--
----------+---------+---------------+---------+------------+
| worker |          0 | PRIMARY    |            1 | W_Id        | A         |           7 |     NULL |   NULL |      | B
TREE      |         |               | YES     | NULL       |
| worker |          1 | First_Name |            1 | First_Name  | A         |           5 |     NULL |   NULL | YES  | B
TREE      |         |               | YES     | NULL       |
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+--
----------+---------+---------------+---------+------------+
2 rows in set (0.01 sec)
```

### c) Controlling User Access

Log-in as other user

Creating new user

```
mysql> create user 'mytester'@'localhost' identified by 'root';
Query OK, 0 rows affected (0.02 sec)
```

Granting only required permission to the user

```
mysql> grant create, insert on org.worker to 'mytester'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Displaying the permission made accessed to the user

```
mysql> show grants for 'mytester'@'localhost';
+-----------------------------------------------------------------+
| Grants for mytester@localhost                                   |
+-----------------------------------------------------------------+
| GRANT USAGE ON *.* TO `mytester`@`localhost`                    |
| GRANT INSERT, CREATE ON `org`.`worker` TO `mytester`@`localhost` |
+-----------------------------------------------------------------+
2 rows in set (0.00 sec)
```

Entering into database as new user

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u mytester -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Making use of the permission accessed INSERT by Inserting data into the table

```
mysql> insert into worker (w_id,First_Name,Last_Name,Salary,Joining_Date,Dept) values (8,'Leo','Das',145000,'2021-03-20 09:00:00','Te
ch');
Query OK, 1 row affected (0.01 sec)
```

Data reflected into the original table

```
mysql> select * from worker;
+------+------------+-----------+---------+---------------------+---------+
| W_Id | First_Name | Last_Name | Salary  | Joining_Date        | Dept    |
+------+------------+-----------+---------+---------------------+---------+
|    1 | Monica     | Arora     |  103000 | 2021-02-20 09:00:00 | HR      |
|    2 | Niharika   | Verma     |   83000 | 2021-06-11 09:00:00 | Admin   |
|    4 | Vivek      | Bhati     | 5003000 | 2021-02-20 09:00:00 | Admin   |
|    5 | Satish     | Kumar     |  753000 | 2021-01-20 09:00:00 | Account |
|    6 | Vipul      | Diwan     |  203000 | 2021-06-11 09:00:00 | Account |
|    7 | Satish     | Kumar     |   78000 | 2021-01-20 09:00:00 | HR      |
|    8 | Leo        | Das       |  145000 | 2021-03-20 09:00:00 | Tech    |
+------+------------+-----------+---------+---------------------+---------+
7 rows in set (0.00 sec)
```

# Practical-VI

Date:

## AIM: Using SET operators, Date/Time Functions, GROUP BY Clause and advanced subqueries

### a) Using SET Operators

Performing UNION and INTERSECT on the created tables

```
mysql> SELECT * FROM bonus UNION SELECT * FROM desgn;
+----------+---------------+---------------------+
| W_Ref_Id | Bonus_Amt     | Bonus_Date          |
+----------+---------------+---------------------+
|        1 | 5000          | 2023-02-20 00:00:00 |
|        2 | 3000          | 2023-06-11 00:00:00 |
|        1 | 4500          | 2023-02-20 00:00:00 |
|        2 | 3500          | 2023-06-11 00:00:00 |
|        1 | Manager       | 2023-02-20 00:00:00 |
|        2 | Executive     | 2023-06-11 00:00:00 |
|        4 | Asst. Manager | 2023-06-11 00:00:00 |
|        5 | Manager       | 2023-06-11 00:00:00 |
+----------+---------------+---------------------+
8 rows in set (0.00 sec)

mysql> SELECT *FROM bonus INTERSECT SELECT *FROM desgn;
Empty set (0.00 sec)
```

### b) Datetime Functions

```
mysql> select CURDATE();
+------------+
| CURDATE()  |
+------------+
| 2023-10-19 |
+------------+
1 row in set (0.00 sec)

mysql> select now();
+---------------------+
| now()               |
+---------------------+
| 2023-10-19 00:28:52 |
+---------------------+
1 row in set (0.00 sec)

mysql> SELECT EXTRACT(YEAR FROM Curdate());
+------------------------------+
| EXTRACT(YEAR FROM Curdate()) |
+------------------------------+
|                         2023 |
+------------------------------+
1 row in set (0.00 sec)
```

### c) Enhancement to the GROUP BY Clause

```
mysql> SELECT dept,COUNT(*) FROM worker GROUP BY dept HAVING COUNT(*) > 1;
+---------+----------+
| dept    | COUNT(*) |
+---------+----------+
| HR      |        2 |
| Admin   |        2 |
| Account |        2 |
+---------+----------+
3 rows in set (0.00 sec)
```

## d) Advanced Subqueries

Making use of aggregation function inside a query within query

```
mysql> SELECT first_name, last_name, salary FROM worker WHERE salary > (SELECT AVG(salary) FROM worker);
+------------+-----------+---------+
| first_name | last_name | salary  |
+------------+-----------+---------+
| Vivek      | Bhati     | 5003000 |
+------------+-----------+---------+
1 row in set (0.00 sec)
```

```
mysql> SELECT first_name, last_name, salary FROM worker WHERE (first_name, last_name) IN (SELECT first_name, last_name FROM worker  W
HERE salary = (SELECT MAX(salary) FROM worker));
+------------+-----------+---------+
| first_name | last_name | salary  |
+------------+-----------+---------+
| Vivek      | Bhati     | 5003000 |
+------------+-----------+---------+
1 row in set (0.01 sec)
```

# Practical-VII

Date:

## AIM: PL/SQL BASIC

Establishing connection and displaying output through print statement

```
SQL*Plus: Release 11.2.0.2.0 Production on Thu Oct 19 09:13:02 2023

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect sys as sysdba
Enter password:
Connected.
SQL> set serveroutput on;
SQL> begin
  2  dbms_output.put_line('hello world');
  3  end;
  4  /
hello world
```

Declaring a variable

```
SQL> declare
  2  x int;
  3  begin
  4  x := &x;
  5  dbms_output.put_line('your number is -'||x);
  6  end;
  7  /
Enter value for x: 12
old   4: x := &x;
new   4: x := 12;
your number is -12

PL/SQL procedure successfully completed.
```

Addition of two numbers by getting input from the user

```
SQL>  declare
  2  num1 int;
  3  num2 int;
  4  result int;
  5  begin
  6  num1 := &num1;
  7  num2 := &num2;
  8  result := num1+num2;
  9  dbms_output.put_line('addition of'||num1||' and '||num2||' is ='||result);
 10  end;
 11  /
Enter value for num1: 3
old   6: num1 := &num1;
new   6: num1 := 3;
Enter value for num2: 2
old   7: num2 := &num2;
new   7: num2 := 2;
addition of3 and 2 is =5

PL/SQL procedure successfully completed.
```

Calculating the area of rectangle by getting required parameter from the user as input and displaying the rest

```
SQL> declare
  2  length int;
  3  breadth int;
  4  begin
  5  length := &length;
  6  breadth := &breadth;
  7  dbms_output.put_line('area of rectangle-'||length*breadth);
  8  end;
  9  /
Enter value for length: 5
old   5: length := &length;
new   5: length := 5;
Enter value for breadth: 4
old   6: breadth := &breadth;
new   6: breadth := 4;
area of rectangle-20

PL/SQL procedure successfully completed.
```

Creating and inserting values into the table in PL/SQL

```
SQL> create table people (
  2  p_id integer not null primary key,
  3  name varchar(25) not null,
  4  fam_name varchar(30) not null,
  5  title varchar(30),
  6  birth_date date
  7  )
  8  ;

Table created.
```

```
SQL> insert into people
  2  values(3,'wex','smith','mr',date'1998-06-01');

1 row created.

SQL> insert into people
  2  values(4,'jeni','lopez','mrs',date'1994-05-11);
ERROR:
ORA-01756: quoted string not properly terminated


SQL> insert into people
  2  values(4,'jeni','lopez','mrs',date'1994-05-11);
ERROR:
ORA-01756: quoted string not properly terminated


SQL> insert into people
  2  values(4,'jeni','lopez','mrs',date'1994-05-11');

1 row created.

SQL> insert into people
  2  values(5,'taylor','smith','mrs',date'1991-05-25');

1 row created.

SQL> select * from people;

      P_ID NAME                      FAM_NAME
---------- ------------------------ ------------------------------
TITLE                          BIRTH_DAT
------------------------------ ---------
         1 dave                        badger
mr                             01-MAY-90

         2 simon                       fox
mr                             11-AUG-99

         3 wex                         smith
mr                             01-JUN-98
```

Creating and storing the area of circle by calculating it, by getting the input from the user.

```
SQL> create table aoc(r int, a float);

Table created.
```

```
SQL> declare
  2  r int :=9;
  3  a float;
  4  begin
  5  a:=3.14*r*r;
  6  insert into aoc values(r,a);
  7  end;
  8  /

PL/SQL procedure successfully completed.
```

```
SQL> select * from aoc;

         R          A
---------- ----------
         9     254.34
```

Swapping of two numbers

```
SQL> declare
  2  a int := 100;
  3  b int := 200;
  4  c int := a;
  5  begin
  6  dbms_output.put_line('Before swapping');
  7  dbms_output.put_line('first number :'||a);
  8  dbms_output.put_line('second number :'||b);
  9  a:=b;
 10  b:=c;
 11  dbms_output.put_line('first number :'||a);
 12  dbms_output.put_line('second number :'||b);
 13  end;
 14  /
Before swapping
first number :100
second number :200
first number :200
second number :100

PL/SQL procedure successfully completed.
```

# Practical-VIII

Date:

## AIM: Composite data types, cursor and exceptions.

### Handling Exceptions

Displaying our own Exception

```
SQL> declare
  2  a int := &a;
  3  b int := &b;
  4  c int;
  5  begin
  6  c := a/b;
  7  dbms_output.put_line(c);
  8  exception
  9  when zero_divide then
 10  dbms_output.put_line('value of b must be greater than 0');
 11  end;
 12  /
Enter value for a: 12
old   2: a int := &a;
new   2: a int := 12;
Enter value for b: 0
old   3: b int := &b;
new   3: b int := 0;
value of b must be greater than 0

PL/SQL procedure successfully completed.
```

Displaying Exception in table

```
SQL> create table deno(sname varchar(5));

Table created.
```

```
SQL> begin
  2  insert into deno values('&sname');
  3  exception
  4  when others then
  5  dbms_output.put_line('too small');
  6  end;
  7  /
Enter value for sname: richard
old   2: insert into deno values('&sname');
new   2: insert into deno values('richard');
too small

PL/SQL procedure successfully completed.
```

## Creating Cursor

Create Table

```
SQL> connect sys as sysdba
Enter password:
Connected.
SQL> set serveroutput on;
SQL> CREATE TABLE employees (
  2     employee_id NUMBER,
  3     first_name VARCHAR2(50),
  4     last_name VARCHAR2(50),
  5     salary NUMBER,
  6     department_id NUMBER
  7  );
```

Writing Cursor explicitly

```
SQL>    DECLARE
  2     CURSOR emp_cursor IS
  3        SELECT employee_id, first_name, last_name, salary
  4        FROM employees
  5        WHERE department_id = 100;
  6  emp_id employees.employee_id%TYPE;
  7     emp_first_name employees.first_name%TYPE;
  8     emp_last_name employees.last_name%TYPE;
  9     emp_salary employees.salary%TYPE;
 10  BEGIN
 11  OPEN emp_cursor;
 12   LOOP
 13        FETCH emp_cursor INTO emp_id, emp_first_name, emp_last_name, emp_salary;
 14        EXIT WHEN emp_cursor%NOTFOUND;
 15   DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_id);
 16        DBMS_OUTPUT.PUT_LINE('Name: ' || emp_first_name || ' ' || emp_last_name);
 17        DBMS_OUTPUT.PUT_LINE('Salary: ' || emp_salary);
 18        DBMS_OUTPUT.PUT_LINE('-------------------------');
 19     END LOOP;
 20
 21     -- Close the cursor
 22     CLOSE emp_cursor;
 23  END;
 24  /

PL/SQL procedure successfully completed.
```

# Practical-IX

Date:

## AIM: Procedure and Functions

### Creating Procedures

Creating procedure which will do no action

```
SQL> create procedure proc1
  2  as
  3  begin
  4  null;
  5  end;
  6  /

Procedure created.
```

Procedure for square of a number

```
SQL> create procedure squ(a in out int)
  2  as
  3  begin
  4  dbms_output.put_line(a*a);
  5  end;
  6  /

Procedure created.

SQL> declare
  2  a int := 12;
  3  begin
  4  squ(a);
  5  end;
  6  /
144

PL/SQL procedure successfully completed.
```

### Functions

Function for squaring of number

```
SQL> create function func1
  2  return int
  3  as
  4  a int := 24;
  5  begin
  6  return a*a;
  7  end;
  8  /

Function created.

SQL> select func1 from dual;

     FUNC1
----------
       576
```

Creating a function for a swapping a number

```
SQL> create or replace function func2
  2  return varchar
  3  as
  4  a int := 24;
  5  b int := 32;
  6  c int;
  7  begin
  8  a := a;
  9  c := a;
 10  a := b;
 11  b := c;
 12  return('a is now: '||a||' '||' and b is now: '||b);
 13  end;
 14  /

Function created.

SQL> select func2 from dual;

FUNC2
--------------------------------------------------------------------------------
a is now: 32  and b is now: 24

SQL>
```

**Creating Packages**

```
SQL> create or replace package natural_nos
  2  as
  3  procedure sumno(a in int);
  4  function sumno(a int) return int;
  5  end natural_nos;
  6  /

Package created.
```