DBMS Assignment A4

```
mysql> create table cust_mstr ( cust_no varchar(25) primary key , fname varchar(25) ,
lname varchar(25));
Query OK, 0 rows affected (0.09 sec)

mysql> insert into cust_mstr values ( '1000' , 'Dhruvil' , 'Shah');
Query OK, 1 row affected (0.02 sec)

mysql> insert into cust_mstr values ( '1001' , 'Soham' , 'Khade');
Query OK, 1 row affected (0.01 sec)

mysql> insert into cust_mstr values ( '1002' , 'Gaurav' , 'Verma');
Query OK, 1 row affected (0.01 sec)
mysql> select * from cust_mstr;
+---------+---------+-------+
| cust_no | fname   | lname |
+---------+---------+-------+
| 1000    | Dhruvil | Shah  |
| 1001    | Soham   | Khade |
| 1002    | Gaurav  | Verma |
+---------+---------+-------+
3 rows in set (0.01 sec)

mysql> create table add_dets ( code_no varchar(25) primary key , add1 varchar(25) , add2
varchar(25) , state varchar(25) , city varchar(25) , pincode int, foreign key (code_no)
references cust_mstr(cust_no));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into add_dets values ( '1000' , 'Bhawani Peth' , '' , 'Maharashtra' , 'Pune' ,
411002);
Query OK, 1 row affected (0.01 sec)
```

mysql> insert into add_dets values ('1001' , 'Amanora' , '' , 'Maharashtra' , 'Pune' , 411037);
Query OK, 1 row affected (0.01 sec)

mysql> insert into add_dets values ('1002' , 'Viman Nagar' , '' , 'Maharashtra' , 'Pune' , 411037);
Query OK, 1 row affected (0.01 sec)

mysql> select * from add_dets;

code_no	add1	add2	state	city	pincode
1000	Bhawani Peth		Maharashtra	Pune	411002
1001	Amanora		Maharashtra	Pune	411037
1002	Viman Nagar		Maharashtra	Pune	411037

3 rows in set (0.00 sec)

Retrieve the address of customer Fname as 'Dhruvil' and Lname as 'Shah'
mysql> select add1,add2,state,city,pincode from add_dets where code_no in(select cust_no from cust_mstr where fname="Dhruvil" and lname="Shah");

add1	add2	state	city	pincode
Bhawani Peth		Maharashtra	Pune	411002

1 row in set (0.00 sec)

mysql> create table acc_fd_cust_dets(codeno varchar(25),acc_fd_no varchar(25) primary key);
Query OK, 0 rows affected (0.04 sec)

```
mysql> insert into acc_fd_cust_dets values( '1000' , 'F10001');
Query OK, 1 row affected (0.01 sec)

mysql> insert into acc_fd_cust_dets values( '1001' , 'F10002');
Query OK, 1 row affected (0.01 sec)

mysql> select * from acc_fd_cust_dets;
+--------+-----------+
| codeno | acc_fd_no |
+--------+-----------+
| 1000   | F10001    |
| 1001   | F10002    |
+--------+-----------+
2 rows in set (0.00 sec)
mysql> create table fd_dets(fd_sr_no varchar(25) primary key ,amt int , foreign key
(fd_sr_no) references acc_fd_cust_dets(acc_fd_no));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into fd_dets values ( 'F10001' , 500000 );
Query OK, 1 row affected (0.02 sec)

mysql> insert into fd_dets values ( 'F10002' , 7000 );
Query OK, 1 row affected (0.01 sec)

mysql> select * from fd_dets;
+----------+--------+
| fd_sr_no | amt    |
+----------+--------+
| F10001   | 500000 |
| F10002   |   7000 |
+----------+--------+
2 rows in set (0.00 sec)
```

List the customer holding fixed deposit of amount more than 5000

```
mysql> select fname,lname from cust_mstr where cust_no in (select codeno from
acc_fd_cust_dets where acc_fd_no in (select fd_sr_no from fd_dets where amt>5000));
+---------+-------+
| fname   | lname |
+---------+-------+
| Dhruvil | Shah  |
| Soham   | Khade |
+---------+-------+
2 rows in set (0.01 sec)

mysql> create table emp_mstr(e_mpno varchar(25) primary key ,f_name
varchar(25),l_name varchar(25) , m_name varchar(25) ,dept varchar(25) ,desg varchar(25)
,branch_no varchar(25));
Query OK, 0 rows affected (0.05 sec)

mysql> insert into emp_mstr values( 'E100' , 'Manish' , 'Shah' ,'','W' , 'Manager' , 'B123');
Query OK, 1 row affected (0.02 sec)

mysql> insert into emp_mstr values( 'E101' , 'Virat' , 'Kohli' ,'','E' , 'Head' , 'B125');
Query OK, 1 row affected (0.01 sec)

mysql> insert into emp_mstr values( 'E102' , 'Yuzi' , 'Chahal' ,'','N' , 'GM' , 'B135');
Query OK, 1 row affected (0.01 sec)

mysql> select * from emp_mstr;
+--------+--------+--------+--------+------+---------+-----------+
| e_mpno | f_name | l_name | m_name | dept | desg    | branch_no |
+--------+--------+--------+--------+------+---------+-----------+
| E100   | Manish | Shah   |        | W    | Manager | B123      |
| E101   | Virat  | Kohli  |        | E    | Head    | B125      |
| E102   | Yuzi   | Chahal |        | N    | GM      | B135      |
+--------+--------+--------+--------+------+---------+-----------+
```

3 rows in set (0.00 sec)

mysql> create table branch_mstr(name varchar(25),b_no varchar (25) primary key);
Query OK, 0 rows affected (0.04 sec)

mysql> insert into branch_mstr values('NIBM' , 'B123');
Query OK, 1 row affected (0.01 sec)

mysql> insert into branch_mstr values('Katraj' , 'B125');
Query OK, 1 row affected (0.01 sec)

mysql> insert into branch_mstr values('Shivajinagar' , 'B135');
Query OK, 1 row affected (0.01 sec)

mysql> select * from branch_mstr;

```
+--------------+------+
| name         | b_no |
+--------------+------+
| NIBM         | B123 |
| Katraj       | B125 |
| Shivajinagar | B135 |
+--------------+------+
```

3 rows in set (0.00 sec)

List the employee details along with branch names to which they belong

mysql> select E.e_mpno,E.f_name,E.l_name,E.m_name,E.dept,E.desg,B.name from emp_mstr E,branch_mstr B where E.branch_no = B.b_no;

```
+--------+--------+--------+--------+------+---------+-------------+
| e_mpno | f_name | l_name | m_name | dept | desg    | name        |
+--------+--------+--------+--------+------+---------+-------------+
| E100   | Manish | Shah   |        | W    | Manager | NIBM        |
| E101   | Virat  | Kohli  |        | E    | Head    | Katraj      |
```

```
| E102   | Yuzi   | Chahal |        | N    | GM      | Shivajinaga |
+--------+--------+--------+--------+------+---------+-------------+
```

3 rows in set (0.00 sec)

```
mysql> create table cntc_dets(code_no varchar(25) primary key ,cntc_type varchar(25)
,cntc_data int);
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> insert into cntc_dets values ( 'E100' , 'Home' , 12345678 );
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into cntc_dets values ( 'E101' , 'Temp' , 1234567890 );
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into cntc_dets values ( 'E102' , 'Work' , 1234567891 );
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from cntc_dets;
+---------+-----------+------------+
| code_no | cntc_type | cntc_data  |
+---------+-----------+------------+
| E100    | Home      |   12345678 |
| E101    | Temp      | 1234567890 |
| E102    | Work      | 1234567891 |
+---------+-----------+------------+
```

3 rows in set (0.00 sec)

List the employee details along with contact details using left outer join & right join

```
mysql> select e_mpno,f_name,l_name,m_name,dept,cntc_type,cntc_data from emp_mstr
left outer join cntc_dets on emp_mstr.e_mpno=cntc_dets.code_no;
```

```
+--------+--------+--------+--------+------+-----------+-----------+
| e_mpno | f_name | l_name | m_name | dept | cntc_type | cntc_data |
+--------+--------+--------+--------+------+-----------+-----------+
| E100   | Manish | Shah   |        | W    | Home      |  12345678 |
| E101   | Virat  | Kohli  |        | E    | Temp      |1234567890 |
| E102   | Yuzi   | Chahal |        | N    | Work      |1234567891 |
+--------+--------+--------+--------+------+-----------+-----------+
3 rows in set (0.01 sec)

mysql> select e_mpno,f_name,l_name,m_name,dept,cntc_type,cntc_data from emp_mstr
right join cntc_dets on emp_mstr.e_mpno=cntc_dets.code_no;
+--------+--------+--------+--------+------+-----------+------------+
| e_mpno | f_name | l_name | m_name | dept | cntc_type | cntc_data  |
+--------+--------+--------+--------+------+-----------+------------+
| E100   | Manish | Shah   |        | W    | Home      |   12345678 |
| E101   | Virat  | Kohli  |        | E    | Temp      | 1234567890 |
| E102   | Yuzi   | Chahal |        | N    | Work      | 1234567891 |
+--------+--------+--------+--------+------+-----------+------------+
3 rows in set (0.00 sec)
```

List the customer who do not have bank branches in their vicinity.

```
mysql> create table Borrower ( Cust_no varchar(25) , loan_no varchar(25) primary key );
Query OK, 0 rows affected (0.04 sec)

mysql> create view borrow as select * from Borrower;
Query OK, 0 rows affected (0.01 sec)

mysql> insert into borrow values("Tommy Shelby","K67834");
```

```
Query OK, 1 row affected (0.01 sec)

mysql> insert into borrow values("Tom Sheldon","K67835");
Query OK, 1 row affected (0.01 sec)

mysql> insert into borrow values("Tom Shel","K67836");
Query OK, 1 row affected (0.01 sec)

mysql> update borrow set cust_no ="Dhruvil" where cust_no ="Tom Sheldon";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> delete from borrow where loan_no="K67834";
Query OK, 1 row affected (0.01 sec)

mysql> select * from borrow;
+----------+---------+
| Cust_no  | loan_no |
+----------+---------+
| Dhruvil  | K67835  |
| Tom Shel | K67836  |
+----------+---------+
2 rows in set (0.00 sec)
```

MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student: Dhruvil Shah	**Class: TE Comp 1**
Semester/Year: 5th/2020	**Roll No: 047**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: Part A-04**

GROUP: B ASSIGNMENT NO: 04

AIM: Design at least 10 SQL queries for suitable database application using SQL DML statements:all types of Join, Sub-Query and View..

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.
- To develop basic Database administration skill.

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End: Java/PHP/Python
- Backend: MySql 5.5

IMPLEMENTATION:

1. Create following Tables

cust_mstr(cust_no,fname,lname)

add_dets(code_no,add1,add2,state,city,pincode)

Retrieve the address of customer Fname as 'xyz' and Lname as 'pqr'

2. Create following Tables

cust_mstr(custno,fname,lname)

acc_fd_cust_dets(codeno,acc_fd_no)

fd_dets(fd_sr_no,amt)

List the customer holding fixed deposit of amount more than 5000

3. Create following Tables
emp_mstr(e_mpno,f_name,l_name,m_name,dept,desg,branch_no)
branch_mstr(name,b_no)

List the employee details along with branch names to which they belong

4. Create following Tables
emp_mstr(emp_no,f_name,l_name,m_name,dept)
cntc_dets(code_no,cntc_type,cntc_data)

List the employee details along with contact details using left outer join & right join

5. Create following Tables
cust_mstr(cust_no,fname,lname)
add_dets(code_no,pincode)

List the customer who do not have bank branches in their vicinity.

6. a) Create View on borrower table by selecting any two columns and perform insert update delete operations

b) Create view on borrower and depositor table by selecting any one column from each table

perform insert update delete operations

c) create updateable view on borrower table by selecting any two columns and perform insert update
delete operations.

CONCLUSION:

QUESTIONS:

1. What is Join Clause?
2. What are the different kinds of Joins ,explain in details?
3. What is cross Join?
4. What is Null value ? How it is different from zero value?
5. What are different MySql constraints?
6. What is purpose of Normalization ? How does it work?
7. What is difference between Join and Union?
8. What are the different aggregation function in MySql?
9. Explain Where and Having clause in detail.
10. What is difference between Unique Key and Primary Key?

Q1 What is Join clause?

Ans A join clause is used to combine rows from two or more tables based on the related column between them.
There are different join clause in SQL
1] INNER JOIN
2] LEFT OUTER JOIN
3] RIGHT OUTER JOIN
4] FULL OUTER JOIN

eg: SELECT orders.orderid, customers.customername from orders INNERJOIN customers on Oders.customer id = customers.customerid

Q2 What are different kind of JOINS, explain in details

Ans. The different kind of joins are

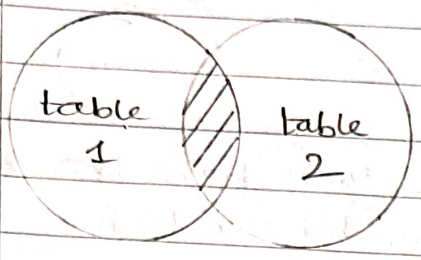
1] INNER JOIN:
Return records that have matching values in both tables.

2] LEFT OUTER JOIN : Returns all records from left table and the matched records from the right table.

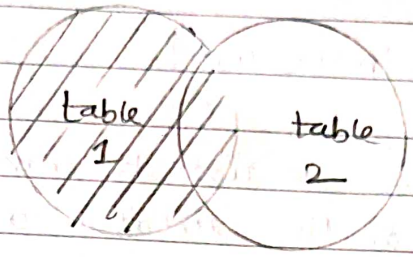
3] RIGHT OUTER JOIN: Returns all records from the right table and the matched records from the left table.

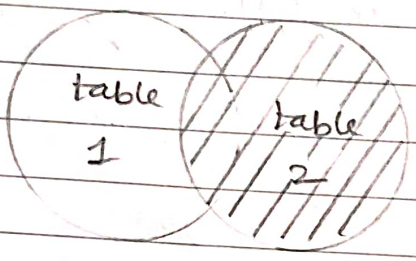
4] FULL OUTER JOIN : Return records when there is a match in either left or right table.

INNER JOIN

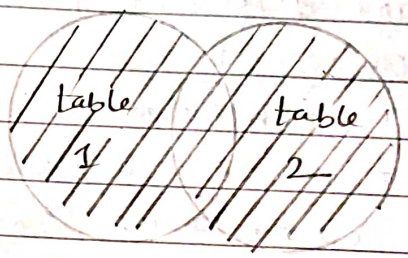


LEFT JOIN





RIGHT JOIN



FULL JOIN

Q3 What is a cross join?

Ans The cross join is used to generate a paired combination of each row of the first table with each row of the second table. This is also called as cartesian join. SQL CROSS JOIN ~~sssss~~ creates all paired combinations of the rows of the tables that will be joined.

Syntax:

```
SELECT column_name 1, column_name 2 .... column_nameN
FROM table 1, table 2..
```

Q4 What is NULL value? How is it different from zero value?

Ans A NULL value is a special marker used in SQL to indicate that a data value does not exist in the database. It is just a placeholder to denote values that are missing or that we do. A NULL value is an unassigned value whereas zero value

is a number and blank space is character.

Q5 What are different mySQL constraints?

Ans MySQL constraints are:
1] NOT NULL
2] UNIQUE
3] PRIMARY KEY
4] FOREIGN KEY
5] CHECK
6] DEFAULT

Q6 What is purpose of Normalization? How does it work?

Ans The main purpose of normalization is to minimize the redundancy and remove insert, update and delete Anomaly. It divides larger tables to smaller tables and links them using relationships. Data redundancy happens when the same piece of data is held in two separate place Insert anomaly occurs when certain attributes cannot be inserted into the database without the presence of other attributes.

Q7 What is the difference between Join and Union?

Ans

	Join	Union
1	Join combines data from many tables based on a matched condition between them	SQL combines the result-set of two or more SELECT statements
2	It combines data into new columns	It combines data into new rows
3	Number of columns selected from each table may not be same	Number of columns selected from each other should be same.
4	It may not return distinct columns	It returns distinct rows

Q8 What are the different aggregation functions in MySQL

Ans The different functions are:

1] COUNT
2] SUM
3] AVG
4] MIN
5] MAX.

Q9 Explain WHERE and HAVING clause in detail.

Ans The HAVING clause enables you to specify conditions that filter which group results appear in results.
The WHERE clause places conditions on the selected columns, whereas the HAVING clause places condition on groups created by the GROUP BY clause.
SYNTAX:
SELECT FROM WHERE GROUP BY HAVING ORDER BY
The having clause must follow the GROUP BY clause in a query and must also precede the ORDER BY clause if used.

Q10 What is the difference between UNIQUE key and PRIMARY key.

Ans

	UNIQUE key	PRIMARY key
1	Can~~not~~ be NULL	Cannot be NULL
2	Unique identifier for rows of a table when primary key is not present	Unique identifier for rows of a table
3	Multiple unique keys can be present in the table	Only one primary key can be there in a table
4	Selection using unique key creates non-clustered index	Selection using primary key creates clustered index