DBMS – Assignment A3

Name:
PRN:
Roll no.:
Class:

Setting up the database

```
mysql> CREATE DATABASE Database_A3;
Query OK, 1 row affected (0.00 sec)

mysql> USE Database_A3;
Database changed
mysql> CREATE TABLE Account(acc_no INT, branch_name VARCHAR(50), balance
INT, PRIMARY KEY (acc_no));
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Branch(branch_name VARCHAR(50), branch_city
VARCHAR(50), assets INT, PRIMARY KEY (branch_name));
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Customer (cust_name VARCHAR(50), cust_street
VARCHAR(50), cust_city VARCHAR(50), PRIMARY KEY (cust_name));
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Depositor (cust_name VARCHAR(50), acc_no INT);
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Loan (loan_no INT, branch_name VARCHAR(50), amount
INT, PRIMARY KEY (loan_no));
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Borrower (cust_name VARCHAR(50), loan_no INT);
Query OK, 0 rows affected (0.02 sec)

mysql> ALTER TABLE Account ADD FOREIGN KEY (branch_name) REFERENCES
Branch(branch_name);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Depositor ADD FOREIGN KEY (cust_name) REFERENCES
Customer (cust_name);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Depositor ADD FOREIGN KEY (acc_no) REFERENCES Account
(acc_no);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Loan ADD FOREIGN KEY (branch_name) REFERENCES Branch
(branch_name);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Borrower ADD FOREIGN KEY (cust_name) REFERENCES
Customer (cust_name);
```

```
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Borrower ADD FOREIGN KEY (loan_no) REFERENCES Loan
(loan_no);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Adding data

```
mysql> INSERT INTO Branch (branch_name, branch_city, assets) VALUES
    -> ("Pune_Station", "Pune", 5000),
    -> ("Hadapsar", "Pune", 20000),
    -> ("Dhole_Patil", "Mumbai", 7500),
    -> ("Nagarwala", "Nandurbar", 3200);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
mysql> INSERT INTO Customer (cust_name, cust_street, cust_city) VALUES
    -> ("Kalas", "Airport Road", "Pune"),
    -> ("Mehul", "Shahdha", "Nandurbar"),
    -> ("Tanmay", "Porwal Road", "Pune"),
    -> ("Kshitij", "Hadapasar", "Pune"),
    -> ("Aditya", "Mira RD", "Mumbai"),
    -> ("Himanshu", "Smart City", "Nandurbar");
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0
mysql> INSERT INTO Account (acc_no, branch_name, balance) VALUES
    -> (2501, "Dhole_Patil", 5000),
    -> (2511, "Pune_Station", 1500),
    -> (2521, "Hadapsar", 2000),
    -> (2512, "Nagarwala", 5000),
    -> (2531, "Pune_Station", 1000);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
mysql> INSERT INTO Loan (loan_no, branch_name, amount) VALUES
    -> (155, "Dhole_Patil", 500),
    -> (156, "Pune_Station", 250),
    -> (157, "Hadapsar", 600),
    -> (158, "Nagarwala", 1400),
    -> (159, "Pune_Station", 25000);
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Borrower VALUES
    -> ("Kalas", 156),
    -> ("Mehul", 158),
    -> ("Tanmay", 155),
    -> ("Kshitij", 157),
    -> ("Aditya", 159),
    -> ("Himanshu", 158);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Depositor VALUES
    -> ("Kalas", 2511),
    -> ("Mehul", 2512),
    -> ("Tanmay", 2501),
    -> ("Kshitij", 2521),
    -> ("Aditya", 2531),
    -> ("Himanshu", 2512);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

Questions

A) Consider following relation and solve the queries: Create different tables given below with appropriate constraints like primary key, foreign key, check constrains, not null etc.

```
Account (Acc_no, branch_name,balance)
Branch (branch_name,branch_city,assets)
Customer (cust_name,cust_street,cust_city)
Depositor (cust_name,acc_no)
Loan (loan_no,branch_name,amount)
Borrower (cust_name,loan_no)
```

1. Create a View1 to display List all customers in alphabetical order who have loan from Pune_Station branch.

```
mysql> CREATE VIEW View1 AS
    -> SELECT cust_name
    -> FROM Borrower
    -> INNER JOIN Loan ON Borrower.loan_no = Loan.loan_no
    -> WHERE branch_name = "Pune_Station"
    -> ORDER BY cust_name;
Query OK, 0 rows affected (0.02 sec)


mysql> SELECT * FROM View1;
```

```
+-----------+
| cust_name |
+-----------+
| Aditya    |
| Kalas     |
+-----------+
```

2. Create View2 on branch table by selecting any two columns and perform insert update delete operations.

```
mysql> CREATE VIEW View2 AS
    -> SELECT branch_name, branch_city
    -> FROM Branch;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM View2;
+--------------+-------------+
| branch_name  | branch_city |
+--------------+-------------+
| Dhole_Patil  | Mumbai      |
| Hadapsar     | Pune        |
| Nagarwala    | Nandurbar   |
| Pune_Station | Pune        |
+--------------+-------------+

mysql> INSERT INTO View2 (branch_name, branch_city) VALUES ('Yerwada',
'Pune');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM View2;
+--------------+-------------+
| branch_name  | branch_city |
+--------------+-------------+
| Dhole_Patil  | Mumbai      |
| Hadapsar     | Pune        |
| Nagarwala    | Nandurbar   |
| Pune_Station | Pune        |
| Yerwada      | Pune        |
+--------------+-------------+

mysql> UPDATE View2 SET branch_name = 'Peachtree' WHERE branch_name =
'Yerwada';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT * FROM View2;
+--------------+-------------+
| branch_name  | branch_city |
+--------------+-------------+
| Dhole_Patil  | Mumbai      |
| Hadapsar     | Pune        |
| Nagarwala    | Nandurbar   |
| Peachtree    | Pune        |
| Pune_Station | Pune        |
+--------------+-------------+

mysql> DELETE FROM View2 WHERE branch_name = 'Peachtree';
Query OK, 1 row affected (0.00 sec)
mysql> SELECT * FROM View2;
+--------------+-------------+
| branch_name  | branch_city |
+--------------+-------------+
| Dhole_Patil  | Mumbai      |
| Hadapsar     | Pune        |
| Nagarwala    | Nandurbar   |
| Pune_Station | Pune        |
+--------------+-------------+
```

3. Create View3 on borrower and depositor table by selecting any one column from each table perform insert update delete operations.

```
mysql> CREATE VIEW View3 AS
    -> SELECT Borrower.cust_name, Depositor.acc_no
    -> FROM Borrower JOIN Depositor ON Borrower.cust_name =
Depositor.cust_name;
Query OK, 0 rows affected (0.01 sec)
mysql> SELECT * FROM View3;
+-----------+--------+
| cust_name | acc_no |
+-----------+--------+
| Aditya    |   2531 |
| Himanshu  |   2512 |
| Kalas     |   2511 |
| Kshitij   |   2521 |
| Mehul     |   2512 |
| Tanmay    |   2501 |
+-----------+--------+
```

```
mysql> INSERT INTO Customer (cust_name, cust_street, cust_city) VALUES
("Macho", "Pedgaon", "Ahemadnagar");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Account (acc_no, branch_name, balance) VALUES (2502,
"Hadapsar", 3000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Loan (loan_no, branch_name, amount) VALUES (160,
"Hadapsar", 500);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Borrower (cust_name, loan_no) VALUES ("Macho", 160);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Depositor(cust_name, Acc_no) VALUES("Macho", 2502);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM View3;
+-----------+--------+
| cust_name | acc_no |
+-----------+--------+
| Aditya    |   2531 |
| Himanshu  |   2512 |
| Kalas     |   2511 |
| Kshitij   |   2521 |
| Macho     |   2502 |
| Mehul     |   2512 |
| Tanmay    |   2501 |
+-----------+--------+

mysql> INSERT INTO Account (acc_no, branch_name, balance) VALUES (2566,
'Hadapsar', 3000);
Query OK, 1 row affected (0.00 sec)

mysql> UPDATE Depositor SET acc_no = 2566 WHERE cust_name = 'Macho';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM View3;
+-----------+--------+
| cust_name | acc_no |
+-----------+--------+
```

```
| Aditya    |   2531 |
| Himanshu  |   2512 |
| Kalas     |   2511 |
| Kshitij   |   2521 |
| Macho     |   2566 |
| Mehul     |   2512 |
| Tanmay    |   2501 |
+-----------+--------+

mysql> DELETE FROM Borrower WHERE cust_name = 'Macho';
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM Depositor WHERE cust_name = 'Macho';
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM View3;
+-----------+--------+
| cust_name | acc_no |
+-----------+--------+
| Aditya    |   2531 |
| Himanshu  |   2512 |
| Kalas     |   2511 |
| Kshitij   |   2521 |
| Mehul     |   2512 |
| Tanmay    |   2501 |
+-----------+--------+
```

4. Create Union of left and right joint for all customers who have an account or loan or both at bank

```
mysql> SELECT DISTINCT Customer.cust_name
    -> FROM Customer
    -> LEFT JOIN Depositor ON Customer.cust_name = Depositor.cust_name
    -> LEFT JOIN Borrower ON Customer.cust_name = Borrower.cust_name
    -> WHERE Depositor.acc_no IS NOT NULL OR Borrower.loan_no IS NOT
NULL;
+-----------+
| cust_name |
+-----------+
| Aditya    |
| Himanshu  |
| Kalas     |
| Kshitij   |
```

```
| Mehul     |
| Tanmay    |
+-----------+
```

5. Display content of View1,View2,View3

```
mysql> SELECT * FROM View1;
+-----------+
| cust_name |
+-----------+
| Aditya    |
| Kalas     |
+-----------+

mysql> SELECT * FROM View2;
+--------------+-------------+
| branch_name  | branch_city |
+--------------+-------------+
| Dhole_Patil  | Mumbai      |
| Hadapsar     | Pune        |
| Nagarwala    | Nandurbar   |
| Pune_Station | Pune        |
+--------------+-------------+

mysql> SELECT * FROM View3;
+-----------+--------+
| cust_name | acc_no |
+-----------+--------+
| Aditya    |   2531 |
| Himanshu  |   2512 |
| Kalas     |   2511 |
| Kshitij   |   2521 |
| Mehul     |   2512 |
| Tanmay    |   2501 |
+-----------+--------+
```

6. Create Simple and Unique index.

```
Simple Index
mysql> CREATE INDEX cust_ind ON Customer (cust_city);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
Unique Index
```

```
mysql> CREATE UNIQUE INDEX branch_ind ON Branch (branch_name);
```

7. Display index Information

```
mysql> SHOW INDEX FROM Customer;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Customer |          0 | PRIMARY  |            1 | cust_name   | A         |           6 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| Customer |          1 | cust_ind |            1 | cust_city   | A         |           4 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.00 sec)

mysql> SHOW INDEX FROM Branch;
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table  | Non_unique | Key_name   | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Branch |          0 | PRIMARY    |            1 | branch_name | A         |           4 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| Branch |          0 | branch_ind |            1 | branch_name | A         |           4 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+--------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.01 sec)
```

8. Truncate table Customer.

```
mysql> DROP TABLE Depositor;
Query OK, 0 rows affected (0.02 sec)
mysql> DROP TABLE Borrower;
Query OK, 0 rows affected (0.02 sec)
mysql> TRUNCATE TABLE Customer;
Query OK, 0 rows affected (0.04 sec)
```

B) Consider following Relation:

```
Companies (comp_id, name, cost, year)
      C001  ONGC  2000  2010
      C002  HPCL  2500  2012
      C005  IOCL 1000 2014
      C006  BHEL  3000  2015
Orders (comp_id, domain, quantity)
      C001  Oil1       09
      C002  Gas        121
      C005  Telecom    115
```

Create above tables with appropriate constraints execute the following query:

Setting up the database

```
mysql> CREATE TABLE Companies(comp_id varchar(50), name varchar(50), cost
int, year int);
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TABLE Orders(comp_id varchar(50), domain varchar(50),
quantity int);
Query OK, 0 rows affected (0.03 sec)
```

Adding values

```
mysql> INSERT INTO Companies (comp_id, name, cost, year) VALUES
    -> ("C001", "ONGC", 2000, 2010),
    -> ("C002", "HPCL", 2500, 2012),
    -> ("C005", "IOCL", 1000, 2014),
    -> ("C006", "BHEL", 3000, 2015);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0


mysql> INSERT INTO Orders (comp_id, domain, quantity) VALUES
    -> ("C001", "Oil", 109),
    -> ("C002", "Gas", 121),
    -> ("C005", "Telecom", 115);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

1. Find names, costs, domains and quantities for companies using inner join.

```
mysql> SELECT name, cost, domain, quantity FROM Companies INNER JOIN
Orders on Companies.comp_id = Orders.comp_id;
+------+------+---------+----------+
| name | cost | domain  | quantity |
+------+------+---------+----------+
| ONGC | 2000 | Oil     |      109 |
| HPCL | 2500 | Gas     |      121 |
| IOCL | 1000 | Telecom |      115 |
+------+------+---------+----------+
```

2. Find names, costs, domains and quantities for companies using left outer join.

```
mysql> SELECT name, cost, domain, quantity FROM Companies lEFT OUTER JOIN
Orders ON Companies.comp_id = Orders.comp_id;
+------+------+---------+----------+
| name | cost | domain  | quantity |
+------+------+---------+----------+
| ONGC | 2000 | Oil     |      109 |
| HPCL | 2500 | Gas     |      121 |
| IOCL | 1000 | Telecom |      115 |
| BHEL | 3000 | NULL    |     NULL |
+------+------+---------+----------+
```

3. Find names, costs, domains and quantities for companies using right outer join.

```
mysql> SELECT name, cost, domain, quantity FROM Companies RIGHT OUTER
JOIN Orders ON Companies.comp_id = Orders.comp_id;
+------+------+---------+----------+
| name | cost | domain  | quantity |
+------+------+---------+----------+
| ONGC | 2000 | Oil     |      109 |
| HPCL | 2500 | Gas     |      121 |
| IOCL | 1000 | Telecom |      115 |
+------+------+---------+----------+
```

4. Find names, costs, domains and quantities for companies using Union operator.

```
mysql> SELECT name, cost FROM Companies UNION SELECT domain, quantity
FROM Orders;
+---------+------+
| name    | cost |
+---------+------+
| ONGC    | 2000 |
| HPCL    | 2500 |
| IOCL    | 1000 |
| BHEL    | 3000 |
| Oil     |  109 |
| Gas     |  121 |
| Telecom |  115 |
+---------+------+
```

5. Create View View1 by selecting both tables to show company name and quantities.

```
mysql> CREATE VIEW view1 AS SELECT name, quantity FROM Companies JOIN
Orders ON Companies.comp_id = Orders.comp_id;
Query OK, 0 rows affected (0.01 sec)


mysql> SELECT * FROM view1;
+------+----------+
| name | quantity |
+------+----------+
| ONGC |      109 |
| HPCL |      121 |
| IOCL |      115 |
+------+----------+
```

6. Create View2 on branch table by selecting any two columns and perform insert update delete operations.

```
mysql> CREATE VIEW view2 AS SELECT name, cost FROM Companies;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM view2;
+------+------+
| name | cost |
+------+------+
| ONGC | 2000 |
| HPCL | 2500 |
| IOCL | 1000 |
| BHEL | 3000 |
+------+------+

mysql> INSERT INTO view2 (name, cost) VALUES ("BCCC", 3100);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM view2;
+------+------+
| name | cost |
+------+------+
| ONGC | 2000 |
| HPCL | 2500 |
| IOCL | 1000 |
| BHEL | 3000 |
| BCCC | 3100 |
+------+------+

mysql> UPDATE view2 SET cost = 3500 WHERE name = "BCCC";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM view2;
+------+------+
| name | cost |
+------+------+
| ONGC | 2000 |
| HPCL | 2500 |
| IOCL | 1000 |
| BHEL | 3000 |
| BCCC | 3500 |
+------+------+

mysql> DELETE FROM view2 WHERE name = "BCCC";
```

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

mysql> SELECT * FROM view2;
+------+------+
| name | cost |
+------+------+
| ONGC | 2000 |
| HPCL | 2500 |
| IOCL | 1000 |
| BHEL | 3000 |
+------+------+
```

7. Display content of View1, View2.

```
mysql> SELECT * from view1;
+------+----------+
| name | quantity |
+------+----------+
| ONGC |      109 |
| HPCL |      121 |
| IOCL |      115 |
+------+----------+

mysql> SELECT * FROM view2;
+------+------+
| name | cost |
+------+------+
| ONGC | 2000 |
| HPCL | 2500 |
| IOCL | 1000 |
| BHEL | 3000 |
+------+------+
4 rows in set (0.00 sec)
```

--- END OF DOCUMENT ---