

MYSQL Assignment No:3

Consider the following table structure for this assignment:

- `mysql> CREATE DATABASE Saranya;`
Query OK, 1 row affected (0.02 sec)
- `mysql> USE Saranya;`
Database changed
- ◆ `mysql> CREATE TABLE CUSTOMER(Cust_id INT PRIMARY KEY,C_name VARCHAR(15),City VARCHAR(20));`
Query OK, 0 rows affected (0.07 sec)
- `mysql> INSERT INTO CUSTOMER(Cust_id, C_name, City) VALUES(1,"Saranya","Idukki"),`
(2,"Anjali","Palakkad"),(3,"Gopika","Kozhikode"),(4,"Parvathi","Palakkad"),
(5,"Renu","Malappuram"),(6,"Sethu","Pathanamthitta"),(7,"Aparna","Kozhikode"),
(8,"Anil","Idukki"),(9,"Revathi","Eranakulam"),(10,"Amala","Eranakulam");
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0
- ◆ `mysql> CREATE TABLE BRANCH(Branch_id INT PRIMARY KEY,B_name VARCHAR(25),City VARCHAR(20));`
Query OK, 0 rows affected (0.07 sec)
- `mysql> INSERT INTO BRANCH(Branch_id, B_name, City) VALUES(1,"Aroor","Kochi"),`
(2,"Vaikom","Kottayam"),(4,"Kaloore","Eranakulam"),(7,"Alappuzha","Kuttanad"),
(6,"Cherthala","Pallipuram"),(5,"Nedumkandam","Idukki"),
(14,"Mundakkayam","Kottayam"),(12,"Edappally","Kollam"),(8,"Kaladi","Thrissur"),
(9,"Kattapana","Idukki");
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0
- ◆ `mysql> CREATE TABLE DEPOSIT(Acc_no VARCHAR(15), Cust_id INT, Amount INT, Branch_id INT, Open_date DATE, FOREIGN KEY (Cust_id) REFERENCES CUSTOMER (Cust_id), FOREIGN KEY (Branch_id) REFERENCES BRANCH (Branch_id));`
Query OK, 0 rows affected (0.10 sec)
- `mysql> INSERT INTO DEPOSIT (Acc_no, Cust_id, Amount, Branch_id, Open_date)`
-> VALUES
-> ('0732108020299', 1, 40500, 1, '2018-01-21'),
-> ('0732108020292', 5, 31000, 6, '2009-07-12'),
-> ('0732108020293', 4, 50000, 5, '2010-03-23'),
-> ('0732108020294', 2, 60000, 2, '2005-08-20'),
-> ('0732108020295', 9, 8000, 12, '2000-05-25'),
-> ('0732108020296', 7, 6300, 8, '2007-06-26'),
-> ('0732108020297', 10, 900, 14, '2008-07-27'),
-> ('0732108020298', 8, 56600, 9, '2009-08-28'),
-> ('0732108020291', 3, 10000, 4, '2010-09-29'),
-> ('0732108020210', 6, 400, 7, '2011-10-30');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
- ◆ `mysql> CREATE TABLE BORROW(Loan_no INT, Cust_id INT, Branch_id INT, Amount INT, FOREIGN KEY (Cust_id) REFERENCES CUSTOMER (Cust_id), FOREIGN KEY (Branch_id) REFERENCES BRANCH (Branch_id));`
Query OK, 0 rows affected (0.10 sec)
- `mysql> INSERT INTO BORROW(Loan_no, Cust_id, Branch_id, Amount)`
VALUES(109,10,14,900),(205,1,1,40500),(34,5,6,31000),(22,6,7,400),(12,3,4,10000),
(2,8,9,56600),(67,7,8,6300),(5,9,12,8000),(78,2,2,60000),(44,8,9,56600);
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0

- mysql> SELECT * FROM CUSTOMER;

Cust_id	C_name	City
1	Saranya	Idukki
2	Anjali	Palakkad
3	Gopika	Kozhikode
4	Parvathi	Palakkad
5	Renu	Malappuram
6	Sethu	Pathanamthitta
7	Aparna	Kozhikode
8	Anil	Idukki
9	Revathi	Eranakulam
10	Amala	Eranakulam

10 rows in set (0.01 sec)

- mysql> SELECT * FROM BRANCH;

Branch_id	B_name	City
1	Aroor	Kochi
2	Vaikom	Kottayam
4	Kaloor	Eranakulam
5	Nedumkandam	Idukki
6	Cherthala	Pallipuram
7	Alappuzha	Kuttanad
8	Kaladi	Thrissur
9	Kattapana	Idukki
12	Edappaly	Kollam
14	Mundakkayam	Kottayam

10 rows in set (0.01 sec)

- mysql> SELECT * FROM DEPOSIT;

Acc_no	Cust_id	Amount	Branch_id	Open_date
0732108020299	1	40500	1	2018-01-21
0732108020292	5	31000	6	2009-07-12
0732108020293	4	50000	5	2010-03-23
0732108020294	2	60000	2	2005-08-20
0732108020295	9	8000	12	2000-05-25
0732108020296	7	6300	8	2007-06-26
0732108020297	10	900	14	2008-07-27
0732108020298	8	56600	9	2009-08-28
0732108020291	3	10000	4	2010-09-29
0732108020210	6	400	7	2011-10-30

10 rows in set (0.00 sec)

- mysql> SELECT * FROM BORROW;

Loan_no	Cust_id	Branch_id	Amount
109	10	14	900
205	1	1	40500
34	5	6	31000
22	6	7	400
12	3	4	10000
2	8	9	56600
67	7	8	6300
5	9	12	8000
78	2	2	60000
44	8	9	56600

10 rows in set (0.00 sec)

1) **List total loan**

```
mysql> SELECT SUM(Amount) AS TotalLoan FROM BORROW;
+-----+
| TotalLoan |
+-----+
|      270300 |
+-----+
1 row in set (0.01 sec)
```

2) **List total deposit**

```
◆ mysql> SELECT SUM(Amount) AS TotalDeposit FROM DEPOSIT;
+-----+
| TotalDeposit |
+-----+
|      263700 |
+-----+
1 row in set (0.00 sec)
```

3) **List maximum deposit of customers living in Ernakulam**

```
◆ mysql> SELECT MAX(Amount) AS MaxDeposit FROM DEPOSIT WHERE Cust_id IN (SELECT Cust_id FROM CUSTOMER WHERE City = 'Ernakulam');
+-----+
| MaxDeposit |
+-----+
|        8000 |
+-----+
1 row in set (0.01 sec)
```

4) **Count total number of branch cities**

```
◆ mysql> SELECT COUNT(DISTINCT City) AS TotalBranchCities FROM BRANCH;
+-----+
| TotalBranchCities |
+-----+
|                8 |
+-----+
1 row in set (0.00 sec)
```

5) **List branch_id and branch wise deposit**

```
◆ mysql> SELECT Branch_id, SUM(Amount) AS TotalDeposit FROM DEPOSIT GROUP BY Branch_id;
+-----+-----+
| Branch_id | TotalDeposit |
+-----+-----+
|          1 |          40500 |
|          2 |          60000 |
|          4 |          10000 |
|          5 |          50000 |
|          6 |          31000 |
|          7 |           400 |
|          8 |          6300 |
|          9 |          56600 |
|         12 |          8000 |
|         14 |           900 |
+-----+-----+
10 rows in set (0.00 sec)
```

6) **How many customers have opened deposit after '01-01-2016'**

```
◆ mysql> SELECT COUNT(DISTINCT Cust_id) AS CustomersCount FROM DEPOSIT WHERE Open_date > '2016-01-01';
+-----+
| CustomersCount |
+-----+
|                1 |
+-----+
1 row in set (0.00 sec)
```

7) List the branches having sum of deposit more than 4000

◆ mysql> SELECT Branch_id, SUM(Amount) AS TotalDeposit FROM DEPOSIT GROUP BY Branch_id HAVING SUM(Amount) > 4000;

Branch_id	TotalDeposit
1	40500
2	60000
4	10000
5	50000
6	31000
8	6300
9	56600
12	8000

8 rows in set (0.00 sec)

8) List the names of customers having minimum deposit

◆ mysql> SELECT C_name FROM CUSTOMER WHERE Cust_id IN (SELECT Cust_id FROM DEPOSIT GROUP BY Cust_id HAVING MIN(Amount) = (SELECT MIN(Amount) FROM DEPOSIT));

C_name
Sethu

1 row in set (0.01 sec)

9) Count the number of depositors living in 'Palakkad'

◆ mysql> SELECT COUNT(DISTINCT Cust_id) AS DepositorsCount FROM DEPOSIT WHERE Cust_id IN (SELECT Cust_id FROM CUSTOMER WHERE City = 'Palakkad');

DepositorsCount
2

1 row in set (0.00 sec)

10) Find the maximum deposit of the Idukki branch

◆ mysql> SELECT MAX(Amount) AS MaxDeposit FROM DEPOSIT WHERE Branch_id IN (SELECT Branch_id FROM BRANCH WHERE City = 'Idukki');

MaxDeposit
56600

1 row in set (0.00 sec)

11) Find out number of customers living in Ernakulam

◆ mysql> SELECT COUNT(*) AS CustomerCount FROM CUSTOMER WHERE City = 'Ernakulam';

CustomerCount
2

1 row in set (0.00 sec)

12) Find out the customers who are not living in Ernakulam or Alappuzha

◆ mysql> SELECT * FROM CUSTOMER WHERE City NOT IN ('Eranakulam', 'Alappuza');

Cust_id	C_name	City
1	Saranya	Idukki
2	Anjali	Palakkad
3	Gopika	Kozhikode
4	Parvathi	Palakkad
5	Renu	Malappuram
6	Sethu	Pathanamthitta
7	Aparna	Kozhikode
8	Anil	Idukki

8 rows in set (0.00 sec)

13) List out Cust_id and C_name in descending order of their C_name

◆ mysql> SELECT Cust_id, C_name FROM CUSTOMER ORDER BY C_name DESC;

Cust_id	C_name
6	Sethu
1	Saranya
9	Revathi
5	Renu
4	Parvathi
3	Gopika
7	Aparna
2	Anjali
8	Anil
10	Amala

10 rows in set (0.00 sec)

14) Display the number of depositors in branch wise

◆ mysql> SELECT Branch_id, COUNT(DISTINCT Cust_id) AS DepositorsCount FROM DEPOSIT GROUP BY Branch_id;

Branch_id	DepositorsCount
1	1
2	1
4	1
5	1
6	1
7	1
8	1
9	1
12	1
14	1

10 rows in set (0.00 sec)

15) Find out the branch which has not borrowers

◆ mysql> SELECT * FROM BRANCH WHERE Branch_id NOT IN (SELECT DISTINCT Branch_id FROM BORROW);

Branch_id	B_name	City
5	Nedumkandam	Idukki

1 row in set (0.00 sec)