Lab Assignment No: 04

NAME: Vivek Pundkar ROLLNO: 77

CLASS: C BRANCH: ENTC BATCH: 3

--------------------------------------------------------------------------------------------------------------------------------

**Problem statement: -**

Create a bank databse with following tables:

Branch master(branch\_id,branch\_name)

Employee master(emp\_no,emp\_name,branch\_id,salary,Dept,manager\_id) (manager\_id & branch\_id is foreign key)

Conatct details(emp\_id,email\_id,phone\_no) (Apply on delete set null constraint & foreign key on emp\_id)

EmpAddress details(emp\_id,street,city,state) (Apply on delete set cascade constraint & foreign key on emp\_id)

Branch address(branch\_id,city,state)(branch\_id is foreign key)

**Objective: -**

The main objective is to learn about different keys and their uses in MySQL.

**Theory/Algorithm/Methodology: -**

**Natural Join:**   
Natural Join joins two tables based on the same attribute name and data types. The resulting table will contain all the attributes of both the table but keep only one copy of each common column.

**Inner Join:**   
Inner Join joins two tables based on the column which is explicitly specified in the ON clause. The resulting table will contain all the attributes from both the tables including the common column also.

In MySQL, a key is a data item that identifies a record exclusively. In other terms, a key is a group of columns used to uniquely define a record in a table. It is used to retrieve or extract rows from a table as needed. Keys have many forms of constraint, such as columns, which cannot hold repeated values or null values.

Types of Keys

MySQL supports the following types of keys. Let’s see.

1. Super Key
2. Candidate Key
3. Primary Key
4. Unique Key
5. Foreign Key

**Primary Key**

A primary key is a column or a group of columns that represent each row in the table in a unique manner. As per Wikipedia, “A primary key is a specific choice of a minimal set of attributes that uniquely identify a tuple (row) in a relation.

## UNIQUE KEY

The altered unique key is to be used to change the structure of the table and to add a unique key of the specified column in the table. The unique key allows one of the null values to be inserted in the column and the unique key table can contain more than one unique key. It is a column constraint.

**FOREIGN KEY**

A Foreign key relationship between two database tables. These tables follow three conditions which are given below.

1. Both tables must be of the same database table types.
2. These fields which have to use in the foreign key relationship must be indexed.
3. The foreign key field relationship must be similar in datatype

**Create a bank databse with following tables:**

**Branch master(branch\_id,branch\_name)**

**Employee master(emp\_no,emp\_name,branch\_id,salary,Dept,manager\_id) (manager\_id & branch\_id**

**is foreign key)**

**Conatct details(emp\_id,email\_id,phone\_no) (Apply on delete set null constraint & foreign key on**

**emp\_id)**

**EmpAddress details(emp\_id,street,city,state) (Apply on delete set cascade constraint & foreign key**

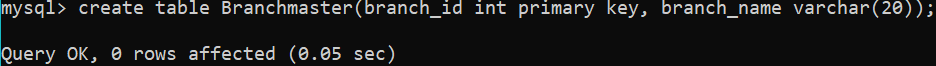
**on emp\_id)**

**Branch address(branch\_id,city,state)(branch\_id is foreign key)**

**1. Insert 5 records in each table.**

**Text

Description automatically generated**

****

**Text

Description automatically generated**

**A screenshot of a computer

Description automatically generated with medium confidence**

Graphical user interface, text

Description automatically generated

A picture containing text, plaque, screenshot

Description automatically generated

**2. List the employee details along with branch name using the inner join and in the order**

**of emp\_no.**

Text

Description automatically generated

**3. List the details of employees who belong to the admin department along with the branch**

**name to which they belong.**

**Text

Description automatically generated**

1. **List the employee’s name along with the phone no and city using an inner join.**

**Text

Description automatically generated**

1. **List the employee’s name with the contact details (if any).**

**Text

Description automatically generated**

1. **List the employee contact details irrespective of whether they are working or have left.**

**Text

Description automatically generated**

1. **Retrieve the employee’s name and their respective manager’s name.**

**Text

Description automatically generated**

1. **List the employee details along with branch name using natural join.**

**Text

Description automatically generated with low confidence**

**9. List the employee names who work at the vegan branch along with the city of that**

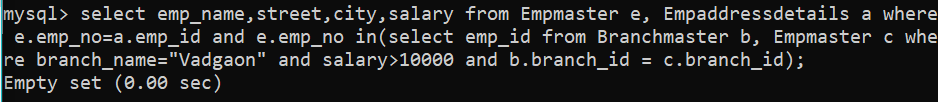
**employee.**

**Text

Description automatically generated**

**10. Find the employee who works at the vegan branch with salary>10000 and list the**

**employee names with streets and city they live in.**

****

**11. Find the employees who live and work in the same city.**

**Text

Description automatically generated**

**12. Find the employees whose salaries are more than everybody who works at the branch**

**vegan.**

**Text

Description automatically generated**

**13. Create a view that will contain the total employees at each branch.**

**Text

Description automatically generated**

**14. List the branch names where the employee has a salary>100000.**

**Text

Description automatically generated**

**15. Create a view which will show the avg salary and the total salary at each branch.**

**Text

Description automatically generated**

**16. Find the employee who does not have a job a the t vegan branch.**

**Text

Description automatically generated**

**CONCLUSION: -** By using different keys we performed many operations and learned about them. and we also learned about inner join and natural join.