LAB ASSIGNMENT - 1

Q1)
Create a table: Employee_Info(emp_id, emp_name, dept, salary)

```
string create_table;
create_table = "CREATE TABLE Employee_Info(emp_id int AUTO_INCREMENT,emp_name
varchar(30),department varchar(60),salary int,PRIMARY KEY (emp_id))";
const char* q = create_table.c_str();
mysql_query(conn,q);
```

a. Find the third_highest salary from the Employee_Info table

```
string third_hi_salary = "SELECT * FROM `employee_info` ORDER BY `salary` DESC LIMIT 1
OFFSET 2";
const char* q = third_hi_salary.c_str();
mysql_query(conn,q);
MYSQL_RES* res=mysql_store_result(conn);
MYSQL_ROW row = mysql_fetch_row(res);
cout<<"Third highest salary is: "<<row[3]<<"\n";
```

Query explanation: ORDER BY salary means it is ordered on the basis of salaries then
DESC key word describes descending order.
LIMIT 1 means we want one record only.
OFFSET 2 implies 1 record after leaving the first 2 in descending order.

b. Display the first and last record from the Employee_Info table

```
string display_fir_las = "(select *from employee_info order by emp_id ASC LIMIT 1)UNION(select *from
employee_info order by emp_id DESC LIMIT 1)";
const char* q1 = display_fir_las.c_str();
mysql_query(conn,q1);
res=mysql_store_result(conn);
cout<<"1st and last entry of employee_info table : \n";
display(res);
```

Query explanation: ORDER BY emp_id(as emp_id is auto incremented whenever new record is added) means it is ordered on the basis of emp_id then
ASC key word describes ascending order.
DESC key word describes descending order.
LIMIT 1 means we want one record only.

c) Copy all rows of the Employee_Info table into another new table Emp_Details

- First created new table with same structure

```
create_table = "CREATE TABLE Emp_Details(emp_id int AUTO_INCREMENT,emp_name
varchar(30),department varchar(60),salary int,PRIMARY KEY (emp_id))";
const char* q2 = create_table.c_str();
mysql_query(conn,q2);
```

- Then copied all the data using query :

```
string q_copy_table = "INSERT INTO `Emp_Details` SELECT * FROM `employee_info`";
const char* q3 = q_copy_table.c_str();
mysql_query(conn,q3);
```

Q2)

Create the following four tables: (10 points)

a. Identify the Primary Key-Foreign Key relationships between the tables. You may design your own Primary Keys, if required. Clearly state assumptions, if any.

Employee(emp_name, street, city)

```
string create_table;
create_table = "CREATE TABLE Employee(emp_id int AUTO_INCREMENT,emp_name  varchar(30)
,street varchar(100),city varchar(30),PRIMARY KEY (emp_id))";
const char* q = create_table.c_str();
mysql_query(conn,q);
```

The primary key for the Employee table is emp_id.(Since name of many employees could be same have used emp_id inorder to get unique value for primary keys)
Have used the auto_increment function inorder to assign int to emp_id from 1 on inserting the data.

Company(company_name, city)

```
create_table = "CREATE TABLE Company(comp_id int AUTO_INCREMENT,comp_name varchar(60),
city varchar(30),PRIMARY KEY (comp_id))";
const char* q1 = create_table.c_str();
mysql_query(conn,q1);
```

The primary key for the Company table is comp_id.(Since company name of many companies could be same have used comp_id inorder to get unique value for primary keys)
Have used the auto_increment function inorder to assign int to comp_id from 1 on inserting the data.

Works(emp_name, company_name, salary)

```
create_table = "CREATE TABLE Works(emp_id int,comp_id int,emp_name varchar(30),comp_name
varchar(60),salary int,FOREIGN KEY (emp_id) REFERENCES Employee(emp_id),FOREIGN KEY
(comp_id) REFERENCES Company(comp_id))";
const char* q2 = create_table.c_str();
```

mysql_query(conn,q2);

There are two foreign keys :
Emp_id from Employee table inorder to recognize employees uniquely.
Comp_id from Company table inorder to recognize employees uniquely.

Managers(emp_name, manager_name)

```
create_table = "CREATE TABLE Managers(emp_id int,emp_name varchar(30),manager_emp_id
int,manager_name varchar(30),FOREIGN KEY (emp_id) REFERENCES Employee(emp_id),FOREIGN
KEY (manager_emp_id) REFERENCES Employee(emp_id))";
const char* q3 = create_table.c_str();
mysql_query(conn,q3)
```

There are 2 foreign keys:
Emp_id from Employee table inorder to recognize employees uniquely.
manager_emp_id from Employee table as managers are employees as well so they also need to be recognized uniquely.

b. Write SQL queries for the following:

1. Find names of all employees who work for SBI

```
string q_SBI_emp = "SELECT emp_id,emp_name from Works WHERE comp_name = 'SBI'";
const char* q4 = q_SBI_emp.c_str();
mysql_query(conn,q4);
MYSQL_RES* res = mysql_store_result(conn);
cout<<"Names of employees working in SBI : \n";
display(res,2);
```

2. Find cities of residence of all employees who work for SBI

```
string q_city = "SELECT emp_name,city from Employee WHERE emp_id IN(SELECT emp_id FROM
Works WHERE comp_name='SBI')";
const char* q5 = q_city.c_str();
mysql_query(conn,q5);
res = mysql_store_result(conn);
cout<<"Names of employees and cities working in SBI : \n";
display(res,2);
```

Explanation : WHERE emp_id IN(SELECT emp_id FROM Works WHERE comp_name='SBI'
First get the emp_id from the Work table.
SELECT emp_name,city from Employee : Used it to get req info from Employee table.

3.Find names of all employees who don’t work for SBI

```
string q_not_SBI_emp = "SELECT emp_id,emp_name from Works WHERE comp_name != 'SBI'";
const char* q6 = q_not_SBI_emp.c_str();
```

```
mysql_query(conn,q6);
res = mysql_store_result(conn);
cout<<"Names of employees not working in SBI : \n";
display(res,2);
```

4.Find names of all employees who have worked for all branches of SBI

c)Simulate examples of various anomalies like Insertion, Deletion, and Update on referenced as well as referencing relations in the aforementioned database. Specify what anomalies would violate the Referential integrity constraint and what could be a potential solution for the same.

Employee table & Company table(REFERENCED TABLE):

Both table are referencing table for Work table and Managers table:

- Inserting in both the tables will not violate any referential integrity as it is a base table.

```
void insert_employee(string name,string street,string city,MYSQL* conn){
   string qti = "INSERT INTO Employee(emp_name,street,city)    VALUES  (
'"+name+"','"+street+"','"+city+"')";
   const char* q = qti.c_str();
   mysql_query(conn,q);
   qti = "SELECT LAST_INSERT_ID()";
   const char* q1 = qti.c_str();
   mysql_query(conn,q1);
   MYSQL_RES* res = mysql_store_result(conn);
   MYSQL_ROW row = mysql_fetch_row(res);
   cout<<"Your employee id is "<<row[0]<<"\n";
}
void insert_company(string name,string city,MYSQL* conn){
   string qti = "INSERT INTO Company(comp_name,city) VALUES('"+name+"','"+city+"')";
   const char* q = qti.c_str();
   mysql_query(conn,q);
   qti = "SELECT LAST_INSERT_ID()";
   const char* q1 = qti.c_str();
   mysql_query(conn,q1);
   MYSQL_RES* res = mysql_store_result(conn);
   MYSQL_ROW row = mysql_fetch_row(res);
   cout<<"Your company id is "<<row[0]<<"\n";
}
```

- When we delete from the employee table all the tables which use it as reference (Works and Managers table) need to delete that record for maintaining referential integrity as when there is no employee so they can't exist in the work table or Managers table.

If the deleted key is manager. In the manager table then we need to set it NULL as the employee will not have any manager now.
When we delete a record in the company table. We need to set NULL for company id in the Work table.

Direct delete gives error
Solution:

- Use delete cascade while creating a reference table: It will get all the records which gave emp_id as foreign key deleted.
- Use on delete set NULL: It will get all the records as emp_id (when manager) and comp_id as foreign key set NULL.

"CREATE TABLE Works(emp_id int,comp_id int,emp_name varchar(30),comp_name varchar(60),salary int,**FOREIGN KEY (emp_id) REFERENCES Employee(emp_id) ON DELETE CASCADE**,**FOREIGN KEY (comp_id) REFERENCES Company(comp_id) ON DELETE SET NULL**)"

"CREATE TABLE Managers(emp_id int,emp_name varchar(30),manager_emp_id int,manager_name varchar(30),**FOREIGN KEY (emp_id) REFERENCES Employee(emp_id) ON DELETE CASCADE,FOREIGN KEY (manager_emp_id) REFERENCES Employee(emp_id) ON DELETE SET NULL)"**

- Similarly for updating ON UPDATE CASCADE can be written.

Work table & Managers table(REFERENCING TABLE):

- Insertion Violation if foreign key is not present in ref table
- Updation may cause violation
- Deletion No violation

Q3)Create the following four tables: (10 points)
a. Identify Primary Key-Foreign Key relationships between the tables. You may design your own Primary Keys, if required. Clearly state assumptions, if any.

Employee(emp_name, email, contact_no., department)

Query : "CREATE TABLE Employee(emp_id int AUTO_INCREMENT,emp_name varchar(30),email varchar(50),contact_no BIGINT,department varchar(60),PRIMARY KEY (emp_id))"

C++ code will be the same as before.

The primary key for the Employee table is emp_id.(Since name of many employees could be same have used emp_id inorder to get unique value for primary keys)
Have used the auto_increment function inorder to assign int to emp_id from 1 on inserting the data.

Department(emp_name, salary, emp_designation)

Query: “CREATE TABLE Department (emp_id int,emp_name varchar(30),salary int,emp_designation varchar(70), FOREIGN KEY (emp_id) REFERENCES Employee(emp_id))"

There was one foreign keys :
Emp_id from Employee table inorder to recognize employees uniquely.

Awardee(emp_name, email, department, experience)

Query: “CREATE TABLE Awardee(emp_id int,emp_name varchar(30),email varchar(70),department varchar(60),exp varchar(1000), FOREIGN KEY (emp_id) REFERENCES Employee(emp_id))"

There was one foreign keys :
Emp_id from Employee table inorder to recognize employees uniquely.

b) 1)
If an update is made on an entry for a particular employee (X) in one table, then it will automatically update corresponding values for X in all other associated tables

Solution : ON UPDATE CASCADE
When a referenced foreign key is deleted or updated, all rows referencing that key are deleted or updated, respectively
So
Query : “CREATE TABLE Department (emp_id int,emp_name varchar(30),salary int,emp_designation varchar(70), FOREIGN KEY (emp_id) REFERENCES Employee(emp_id)) ON UPDATE CASCADE"

Query : “CREATE TABLE Awardee(emp_id int,emp_name varchar(30),email varchar(70),department varchar(60),exp varchar(1000), FOREIGN KEY (emp_id) REFERENCES Employee(emp_id)) ON UPDATE CASCADE"

2. Find no. of awardees from each department

Query : “SELECT department,COUNT(*) FROM awardee GROUP BY department”

C++ code:

```
res= mysql_store_result (conn);
int count = mysql_num_fields (res);
while (row=mysql_fetch_row(res)){
        for(int i=0; i<count; i++){
                cout<<"\t"<<row[i];
            }
            cout<<endl;
          }
}
```

Explanation: From awardee table all are grouped by departments.Then count is used to get no.of awardee from each department.