SQL Assignment MODULE 4

1.Statement to create the Contact table.

```
Create TABLE Contact (
ContactId int primary key
CompanyId int,
FirstName varchar(45),
LastName varchar(45),
Street varchar(45),
City varchar (45),
State varchar(2),
Zip varchar(10),
IsMain Boolean,
Email varchar(45),
Phone varchar(12)
);
```

2.Statement to create the Employee table

```
Create table Employee (
Employeeid int,
FirstName varchar(45),
LastName varchar(45),
Salary decimal(10,2),
Hiredate date,
Jobtitle varchar(45),
Email varchar(50),
Phone varchar(20)
);
```

3.Statement to create the ContactEmployee table

```
Create Table ConatctEmployee (
ContactEmployeeid int primary key,
ContactId int,
Employeeid int ,
Contactdate date,
Description varchar(100)
);
```

4.In the Employee table, the statement that changes Lesley Bland's phone number to 215-555-8800

→ Before

```
80 •   select * from employee;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Employeeid	FirstName	LastName	Salary	Hiredate	Jobtitle	Email	Phone
1	Lesley	Bland	5000.00	2013-01-01	Executive	lasley.bland@helpdesk.com	222-555-8800
2	Alan	border	8000.00	2013-02-01	salseman	alan.border@selsehelp.com	222-656-8800
3	mark	wough	10000.00	2013-01-05	Manager	mark.wough@managerhelp.com	222-585-8970
4	ricky	lee	9000.00	2013-03-01	supervisor	riky.lee@helpdesk.com	222-698-8700

→ After

```
82 •   set sql_safe_updates=0;
83
84 •   update employee
85     set phone = '215-555-8800'
86     where Firstname = 'lesley';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Employeeid	FirstName	LastName	Salary	Hiredate	Jobtitle	Email	Phone
1	Lesley	Bland	5000.00	2013-01-01	Executive	lasley.bland@helpdesk.com	215-555-8800
2	Alan	border	8000.00	2013-02-01	salseman	alan.border@selsehelp.com	222-656-8800
3	mark	wough	10000.00	2013-01-05	Manager	mark.wough@managerhelp.com	222-585-8970
4	ricky	lee	9000.00	2013-03-01	supervisor	riky.lee@helpdesk.com	222-698-8700

5.In the Company table, the statement that changes the name of “Urban Outfitters, Inc.” to “Urban Outfitters” .

→ Before

```
88 •  select * from company;
```

Result Grid | Filter Rows: | Edit: | Export/Impo

Companyid	CompanyName	Street	City	State	Zip
1	VISA	BAKERst	LONDON	EG	250221
2	Urban Outfitters,Inc	C.T.Square	LAS Vegas	US	250224
3	SUZUKI	B.B.st	TOKYO	JP	250225
4	AMUL	A.D.st	ANAND	ID	250227
NULL	NULL	NULL	NULL	NULL	NULL

→ After

```
90 •  update company
91    set companyname = 'Urban Outfitters'
92    where companyid = 2;
```

Result Grid | Filter Rows: | Edit: | Export/Imp

Companyid	CompanyName	Street	City	State	Zip
1	VIsa	BAKERst	LONDON	EG	250221
2	Urban Outfitters	C.T.Square	LAS Vegas	US	250224
3	SUZUKI	B.B.st	TOKYO	JP	250225
4	AMUL	A.D.st	ANAND	ID	250227
NULL	NULL	NULL	NULL	NULL	NULL

6.In ContactEmployee table, the statement that removes Dianne Connor’s contact event with Jack Lee (one statement).

→ Before

```
104 •  select * from contact;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ContactId	CompanyId	FirstName	LastName	Street	City	State	Zip	IsMain	Email	Phone
101	1	Toll	Brothers	B.Bst	LONDON	EG	250222	1	toll.brothers@example.com	222-555-8585
102	2	dianne	Connor	N.Bst	Tokyo	JP	250225	0	dianne.connor@example.com	222-666-8085
103	3	Lesley	Bland	Baker	LAS vegas	US	250210	1	Lesley.bland@example.com	215-555-8800
104	4	Mach	brown	United	Swiss	IL	260250	0	mach.brown@example.com	210-556-8560
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

contact 1 x

→ **After**

```
108 •  select c.contactid,c.firstname,c.lastname
109    from contact c
110    join ConatctEmployee ce
111    on c.contactid = ce.contactid;
112
113 •  delete from ConatctEmployee
114    where contactid = 102;
115
116
117
118
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

contactid	firstname	lastname
101	Toll	Brothers
103	Lesley	Bland
104	Mach	brown

7. Write the SQL SELECT query that displays the names of the employees that have contacted Toll Brothers (one statement). Run the SQL SELECT query in MySQL Workbench. Copy the results below as well.

```
96 •  select e.firstname,e.lastname
97    from employee e
98    join  ConatctEmployee ce
99    on e.employeeid = ce.employeeid
100   join contact c
101   on ce.contactid = c.contactid
102   where c.firstname = 'toll' and c.lastname = 'brothers';
103
104
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

firstname	lastname
Lesley	Bland

8. What is the significance of “%” and “_” operators in the LIKE statement?

→ In SQL LIKE->

- ‘%’ Matches zero or more characters (e.g. ‘A%’ matches any string that starts with ‘A’,(e.g. “Apple” , ”Anna”)
- ‘_’ Matches exactly one character (e.g. ‘J_n’ matches any three letter string starting with ‘J’ and ending ‘N’ .

9. Explain normalization in the context of databases.

→ Normalization in database is the process of organizing data in database to reduce redundancy and improve data integrity by dividing database into multiple related tables. It reduces redundancy, improves Consistency and Prevents Anomalies.

10.What does a join in MySQL mean?

→ In MySQL a Join combines rows from multiple tables based on a related column. Join enabling you to combine and analyze related data efficiently.

11. What do you understand about DDL, DCL, and DML in MySQL?

→ These are the categories of SQL statements.

- DDL (**Data Definition Language**) :- It defines and modifies the structure of database objects. (e.g. CREATE, ALTER , DROP)

  →Changes are permanent and affects the database schema

- DML (**Data Manipulation Language**) :- It manages the data within tables. (e.g. SELECT , INSERT , UPDATE, DELETE)

→Operates on table contents, not structure

- DCL (**Data Control Language**) :- It controls access and permissions for database users. (e.g. GRANT , REVOKE)

  → Manages security.

12) What is the role of the MySQL JOIN clause in a query, and what are some common types of joins?

→ Role of MySQL JOIN clause in a query is it joins tables based on common column, and help to retrieve data from multiple table in a single query by linking them through specific conditions.

→ These are the common types of joins.

- INNER JOIN()
- LEFT JOIN()
- RIGHT JOIN
- FULL JOIN
- CROSS JOIN