

## EXPERIMENT 2

**Aim:** To perform the primary and foreign key in the table and perform various Joins.

**Software Used:** MySQL

**Theory:**

### Primary Key:

The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

### Foreign Key:

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table. The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

```
mysql> create table Customers(  
    -> cid int not null primary key,  
    -> cname varchar(255),  
    -> cemail varchar(255) );  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> desc Customers;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| cid   | int           | NO   | PRI | NULL    |       |  
| cname | varchar(255)  | YES  |     | NULL    |       |  
| cemail| varchar(255)  | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.01 sec)
```

```
mysql> create table Orders(  
    -> oid int not null primary key,  
    -> orderdate date,  
    -> oamount int,  
    -> cid int, foreign key(cid) references Customers(cid) );  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> desc Orders;
```

Field	Type	Null	Key	Default	Extra
oid	int	NO	PRI	NULL	
orderdate	date	YES		NULL	
oamount	int	YES		NULL	
cid	int	YES	MUL	NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> insert into Customers values(1,'vinod','vinod@thapa.com'),  
    -> (2,'bahadur','bhahadur@thapa.com'),  
    -> (3,'thapa','thapa@thapa.com');
```

```
Query OK, 3 rows affected (0.01 sec)
```

```
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> insert into Orders value(1,'2019/05/05',55,1),
-> (2,'2019/08/06',85,2),
-> (3,'2019/08/05',155,1),
-> (4,'2019/05/12',95,3);
```

Query OK, 4 rows affected, 4 warnings (0.00 sec)

Records: 4 Duplicates: 0 Warnings: 4

```
mysql> select * from Customers;
```

cid	cname	cemail
1	vinod	vinod@thapa.com
2	bahadur	bahadur@thapa.com
3	thapa	thapa@thapa.com

3 rows in set (0.00 sec)

```
mysql> select * from Orders;
```

oid	orderdate	oamount	cid
1	2019-05-05	55	1
2	2019-08-06	85	2
3	2019-08-05	155	1
4	2019-05-12	95	3

4 rows in set (0.00 sec)

```
mysql> select * from Orders,Customers;
```

oid	orderdate	oamount	cid	cid	cname	cemail
1	2019-05-05	55	1	3	thapa	thapa@thapa.com
1	2019-05-05	55	1	2	bahadur	bhahadur@thapa.com
1	2019-05-05	55	1	1	vinod	vinod@thapa.com
2	2019-08-06	85	2	3	thapa	thapa@thapa.com
2	2019-08-06	85	2	2	bahadur	bhahadur@thapa.com
2	2019-08-06	85	2	1	vinod	vinod@thapa.com
3	2019-08-05	155	1	3	thapa	thapa@thapa.com
3	2019-08-05	155	1	2	bahadur	bhahadur@thapa.com
3	2019-08-05	155	1	1	vinod	vinod@thapa.com
4	2019-05-12	95	3	3	thapa	thapa@thapa.com
4	2019-05-12	95	3	2	bahadur	bhahadur@thapa.com
4	2019-05-12	95	3	1	vinod	vinod@thapa.com

```
12 rows in set (0.00 sec)
```

```
mysql> select * from Customers,Orders;
```

cid	cname	cemail	oid	orderdate	oamount	cid
3	thapa	thapa@thapa.com	1	2019-05-05	55	1
2	bahadur	bhahadur@thapa.com	1	2019-05-05	55	1
1	vinod	vinod@thapa.com	1	2019-05-05	55	1
3	thapa	thapa@thapa.com	2	2019-08-06	85	2
2	bahadur	bhahadur@thapa.com	2	2019-08-06	85	2
1	vinod	vinod@thapa.com	2	2019-08-06	85	2
3	thapa	thapa@thapa.com	3	2019-08-05	155	1
2	bahadur	bhahadur@thapa.com	3	2019-08-05	155	1
1	vinod	vinod@thapa.com	3	2019-08-05	155	1
3	thapa	thapa@thapa.com	4	2019-05-12	95	3
2	bahadur	bhahadur@thapa.com	4	2019-05-12	95	3
1	vinod	vinod@thapa.com	4	2019-05-12	95	3

```
12 rows in set (0.00 sec)
```

## Joints

### A. INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

```
mysql> select * from Customers,Orders
-> where Customers.cid=Orders.cid;
+-----+-----+-----+-----+-----+-----+-----+
| cid | cname | cemail | oid | orderdate | oamount | cid |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | vinod | vinod@thapa.com | 1 | 2019-05-05 | 55 | 1 |
| 1 | vinod | vinod@thapa.com | 3 | 2019-08-05 | 155 | 1 |
| 2 | bahadur | bhahadur@thapa.com | 2 | 2019-08-06 | 85 | 2 |
| 3 | thapa | thapa@thapa.com | 4 | 2019-05-12 | 95 | 3 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from Customers
-> join Orders
-> on Customers.cid=Orders.cid;
+-----+-----+-----+-----+-----+-----+-----+
| cid | cname | cemail | oid | orderdate | oamount | cid |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | vinod | vinod@thapa.com | 1 | 2019-05-05 | 55 | 1 |
| 1 | vinod | vinod@thapa.com | 3 | 2019-08-05 | 155 | 1 |
| 2 | bahadur | bhahadur@thapa.com | 2 | 2019-08-06 | 85 | 2 |
| 3 | thapa | thapa@thapa.com | 4 | 2019-05-12 | 95 | 3 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

### B. LEFT JOIN

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain *null*. LEFT JOIN is also known as LEFT OUTER JOIN.

```
mysql> select * from Customers
-> left join Orders
-> on Customers.cid=Orders.cid;
+-----+-----+-----+-----+-----+-----+-----+
| cid | cname | cemail | oid | orderdate | oamount | cid |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | vinod | vinod@thapa.com | 1 | 2019-05-05 | 55 | 1 |
| 1 | vinod | vinod@thapa.com | 3 | 2019-08-05 | 155 | 1 |
| 2 | bahadur | bhahadur@thapa.com | 2 | 2019-08-06 | 85 | 2 |
| 3 | thapa | thapa@thapa.com | 4 | 2019-05-12 | 95 | 3 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select Customers.cid, cname, oamount
-> from Customers
-> left join Orders
-> on Customers.cid=Orders.cid;
```

cid	cname	oamount
1	vinod	55
1	vinod	155
2	bahadur	85
3	thapa	95

4 rows in set (0.00 sec)

### C. RIGHT JOIN

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain *null*. RIGHT JOIN is also known as RIGHT OUTER JOIN.

```
mysql> select * from Customers
-> right join Orders
-> on Customers.cid=Orders.cid;
```

cid	cname	cemail	oid	orderdate	oamount	cid
1	vinod	vinod@thapa.com	1	2019-05-05	55	1
2	bahadur	bahadur@thapa.com	2	2019-08-06	85	2
1	vinod	vinod@thapa.com	3	2019-08-05	155	1
3	thapa	thapa@thapa.com	4	2019-05-12	95	3

4 rows in set (0.00 sec)

```
mysql> select Customers.cid, cname, oamount
-> from Customers
-> right join Orders
-> on Customers.cid=Orders.cid;
```

cid	cname	oamount
1	vinod	55
2	bahadur	85
1	vinod	155
3	thapa	95

4 rows in set (0.00 sec)

#### D. FULL JOIN

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain *NULL* values.

```
mysql> select Customers.cid, cname, oamount
-> from Customers
-> full join Orders
-> on Customers.cid=Orders.cid;
```

**Conclusion:** Performed the primary and foreign key in the table and perform various Joins.