

### LAB Assignment-4

#### 1. Create the following tables with the constraints mentioned:

**Note:** the data type and size should be give in relevance with the data to be inserted. Constraint name are not required to be given for this assignment.

##### Customer

Cust_id	Fname	Lname	Area	Phone
Primary Key	Not Null		Not Null	
A01	Ivan	Ross	SA	6125467
A02	Vandana	Ray	MU	5560379
A03	Pramada	Jauguste	DA	4560389
A04	Basu	Navindi	BA	6125401
A05	Ravi	Shridhar	NA	null
A06	Rukmini	Aiyer	GH	5125274

##### Movie

Mv_no	Cust_id	Title	Star	Price
Primary Key	Foreign Key	Not Null	Not Null	Should be between 100 and 250
1	A02	Bloody	JC	181
2	A04	The Firm	TC	200
3	A01	Pretty Woman	RG	151
4	A06	Home Alone	MC	150
5	A05	The Fugitive	MF	200
6	A03	Coma	MD	100
7	A02	Dracula	GO	150
8	A06	Quick Change	BM	100
9	A03	Gone with the Wind	CB	200
10	A05	Carry on Doctor	LP	100

```
mysql> CREATE TABLE Customer (  
-> Cust_id VARCHAR(3) PRIMARY KEY,  
-> Fname VARCHAR(20) NOT NULL,  
-> Lname VARCHAR(20),  
-> Area VARCHAR(5) NOT NULL,  
-> Phone VARCHAR(10)  
-> );
```

Query OK, 0 rows affected (0.11 sec)

```
mysql>  
mysql> CREATE TABLE Movie (  
-> Mv_no INT PRIMARY KEY,  
-> Cust_id VARCHAR(3),  
-> Title VARCHAR(30) NOT NULL,  
-> Star VARCHAR(5) NOT NULL,  
-> Price INT CHECK (Price BETWEEN 100 AND 250),  
-> FOREIGN KEY (Cust_id) REFERENCES Customer(Cust_id)  
-> );
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> INSERT INTO Customer VALUES  
-> ('A01', 'Ivan', 'Ross', 'SA', '6125467'),  
-> ('A02', 'Vandana', 'Ray', 'MU', '5560379'),  
-> ('A03', 'Pramada', 'Jauguste', 'DA', '4560389'),  
-> ('A04', 'Basu', 'Navindi', 'BA', '6125401'),  
-> ('A05', 'Ravi', 'Shridhar', 'NA', NULL),  
-> ('A06', 'Rukmini', 'Aiyer', 'GH', '5125274');
```

Query OK, 6 rows affected (0.02 sec)

Records: 6 Duplicates: 0 Warnings: 0

```
mysql>  
mysql> INSERT INTO Movie VALUES  
-> (1, 'A02', 'Bloody', 'JC', 181),  
-> (2, 'A04', 'The Firm', 'TC', 200),  
-> (3, 'A01', 'Pretty Woman', 'RG', 151),  
-> (4, 'A06', 'Home Alone', 'MC', 150),  
-> (5, 'A05', 'The Fugitive', 'MF', 200),  
-> (6, 'A03', 'Coma', 'MD', 100),  
-> (7, 'A02', 'Dracula', 'GO', 150),  
-> (8, 'A06', 'Quick Change', 'BM', 100),  
-> (9, 'A03', 'Gone with the Wind', 'CB', 200),  
-> (10, 'A05', 'Carry on Doctor', 'LP', 100);
```

Query OK, 10 rows affected (0.01 sec)

Records: 10 Duplicates: 0 Warnings: 0

2. Prove that entity integrity constraint is ensured by both the tables. (2 conditions to be checked).

Primary keys (Cust\_id, Mv\_no) are defined and not null.

No duplicate values in primary key columns.

Both tables satisfy entity integrity.

3. Prove that referential integrity constraint is ensured by both the tables.

4. Prove that domain integrity constraint is ensured by the Movie table.

5. Display the movie titles, whose price is greater than 100 but less than 200.

```
mysql> SELECT Title FROM Movie WHERE Price > 100 AND Price < 200;
+-----+
| Title |
+-----+
| Bloody |
| Pretty Woman |
| Home Alone |
| Dracula |
+-----+
```

6. Display the cust\_id who have seen movies having stars as either JC or TC or MC.

```
mysql> SELECT Cust_id FROM Movie WHERE Star IN ('JC', 'TC', 'MC');
+-----+
| Cust_id |
+-----+
| A02     |
| A04     |
| A06     |
+-----+
```

7. Display the details of those customers who have an A in their area name.

```
mysql> SELECT * FROM Customer WHERE Area LIKE '%A%';
+-----+-----+-----+-----+-----+
| Cust_id | Fname | Lname | Area | Phone |
+-----+-----+-----+-----+-----+
| A01     | Ivan  | Ross  | SA   | 6125467 |
| A03     | Pramada | Jauguste | DA   | 4560389 |
| A04     | Basu  | Navindi | BA   | 6125401 |
| A05     | Ravi  | Shridhar | NA   | NULL    |
+-----+-----+-----+-----+-----+
```

8. Display the movie titles, whose price is within 180 and the movie titles are of exactly 6 characters.

```
mysql> SELECT Title FROM Movie
-> WHERE Price <= 180 AND LENGTH(Title) = 6;
Empty set (0.00 sec)
```

9. Display the movie name, their original prices and the prices after 10% increment. Give alias name to the incremented price column.

```
mysql> SELECT Title, Price, Price * 1.10 AS "Incremented Price" FROM Movie;
```

Title	Price	Incremented Price
Bloody	181	199.10
The Firm	200	220.00
Pretty Woman	151	166.10
Home Alone	150	165.00
The Fugitive	200	220.00
Coma	100	110.00
Dracula	150	165.00
Quick Change	100	110.00
Gone with the Wind	200	220.00
Carry on Doctor	100	110.00

10. Display all the customer details in the following way:

‘Ivan Ross stays in SA and his phone number is 6125467.’

```
mysql> SELECT CONCAT(Fname, ' ', Lname, ' stays in ', Area, ' and his phone number is ', Phone, '.') AS Details
-> FROM Customer;
```

Details
Ivan Ross stays in SA and his phone number is 6125467.
Vandana Ray stays in MU and his phone number is 5560379.
Pramada Jauguste stays in DA and his phone number is 4560389.
Basu Navindi stays in BA and his phone number is 6125401.
NULL
Rukmini Aiyer stays in GH and his phone number is 5125274.

11. Add a not null constraint to the Lname field in Customer.

```
mysql> ALTER TABLE Customer MODIFY Lname VARCHAR(20) NOT NULL;
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

12. Display the customer name whose phone number is not recorded.

```
mysql> SELECT Fname, Lname FROM Customer WHERE Phone IS NULL;
+-----+-----+
| Fname | Lname   |
+-----+-----+
| Ravi  | Shridhar |
+-----+-----+
```

13. Add the phone number according to your own wish for the person mentioned in problem no 7.

```
mysql> UPDATE Customer SET Phone = '9999999999' WHERE Cust_id = 'A05';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

14. Display the unique customer id's from movie table.

```
mysql> SELECT DISTINCT Cust_id FROM Movie;
+-----+
| Cust_id |
+-----+
| A01     |
| A02     |
| A03     |
| A04     |
| A05     |
| A06     |
+-----+
```

15. Remove the not null constraint from Star column in movie table.

```
mysql> ALTER TABLE Movie MODIFY Star VARCHAR(5);
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

16. Delete any row from the Customer table. If you cannot delete, then note the error message displayed.

```
mysql> DELETE FROM Customer WHERE Cust_id = 'A06';  
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails ('sem3`.`movie`, CONSTRAINT `movie_ibfk_1` FOREIGN KEY (`Cust_id`) REFERENCES `customer` (`Cust_id`))
```

**17. Delete any row from the Movie table. If you cannot delete, then note the error message displayed.**

```
mysql> DELETE FROM Movie WHERE Mv_no = 10;  
Query OK, 1 row affected (0.00 sec)
```

**18. Drop the Customer table. If you cannot drop, then note the error message displayed.**

```
mysql> DROP TABLE Customer;  
ERROR 3730 (HY000): Cannot drop table 'customer' referenced by a foreign key constraint 'movie_ibfk_1' on table 'movie'.
```

**19. Drop the Movie table. If you cannot drop, then note the error message displayed.**

```
mysql> DROP TABLE Movie;  
Query OK, 0 rows affected (0.05 sec)
```

**20. Drop the foreign key from Movie table.**

```
mysql> ALTER TABLE Movie DROP FOREIGN KEY Cust_id;  
ERROR 1146 (42S02): Table 'sem3.movie' doesn't exist
```