Exercise
Databases
Name: Mahesh Inder

Problem Statement: There can be multiple customers, who can place multiple orders on the site. Now a sales person can handle these orders will distribute into multiple sales persons (One order will be assign to one salesperson only). So a sales person can have multiple orders of multiple customers.

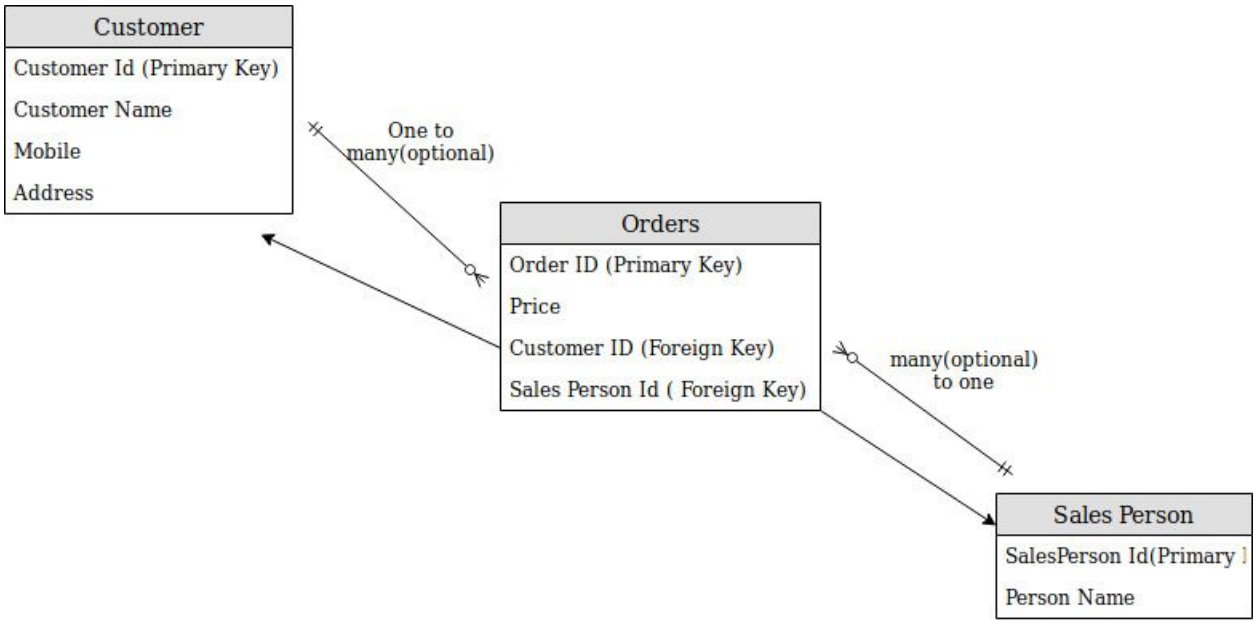
1. **Create Database.**

A. Databases can be created by using “create database db_name” command.

```
mysql> create database assignment;
Query OK, 1 row affected (0.00 sec)

mysql> use assignment;
Database changed
mysql> select database();
+------------+
| database() |
+------------+
| assignment |
+------------+
1 row in set (0.00 sec)
```

2. Design Schema



ER Diagram

(Schema for the given Problem)

3. Create tables.

A. Using create table command to create table:

- Customer table :

```
mysql> create table customer (cust_id int PRIMARY KEY, name char(50), mobile bigint, adddress varchar(50));
Query OK, 0 rows affected (0.30 sec)
```

- Seller and Orders tables:

```
mysql> create table seller ( seller_id int PRIMARY KEY , name char(50) );
Query OK, 0 rows affected (0.32 sec)

mysql> create table orders(order_id int PRIMARY KEY, price int, cust_id int, seller_id int,
FOREIGN KEY (cust_id) REFERENCES customer(cust_id), FOREIGN KEY (seller_id) REFERENCES s
eller(seller_id));
Query OK, 0 rows affected (0.44 sec)
```

Let's have a look at created tables.

```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_assignment |
+----------------------+
| customer             |
| orders               |
| seller               |
+----------------------+
3 rows in set (0.00 sec)

mysql> DESC CUSTOMER;
ERROR 1146 (42S02): Table 'assignment.CUSTOMER' doesn't exist
mysql> DESC customer;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| cust_id  | int(11)     | NO   | PRI | NULL    |       |
| name     | char(50)    | YES  |     | NULL    |       |
| mobile   | bigint(20)  | YES  |     | NULL    |       |
| adddress | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> DESC orders;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| order_id  | int(11) | NO   | PRI | NULL    |       |
| price     | int(11) | YES  |     | NULL    |       |
| cust_id   | int(11) | YES  | MUL | NULL    |       |
| seller_id | int(11) | YES  | MUL | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> DESC seller;
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| seller_id | int(11)  | NO   | PRI | NULL    |       |
| name      | char(50) | YES  |     | NULL    |       |
+-----------+----------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Seller table

4. Insert sample data.

A. Using insert command, data can be inserted into the table.

```
mysql> insert into customer values (1, "Alex", 7898585, "Kingspark, London");
Query OK, 1 row affected (0.09 sec)

mysql> insert into customer values (2, "Harry", 2489745, "Silicon Valley");
Query OK, 1 row affected (0.09 sec)

mysql> insert into customer values (3, "Yoyo", 7895652, "Townhall, Massachussets");
Query OK, 1 row affected (0.10 sec)

mysql> select * from customer;
+---------+-------+---------+-------------------------+
| cust_id | name  | mobile  | adddress                |
+---------+-------+---------+-------------------------+
|       1 | Alex  | 7898585 | Kingspark, London       |
|       2 | Harry | 2489745 | Silicon Valley          |
|       3 | Yoyo  | 7895652 | Townhall, Massachussets |
+---------+-------+---------+-------------------------+
3 rows in set (0.00 sec)
```

Insertion in Customer table.

```
mysql> insert into seller values (21, "Yoyo");
Query OK, 1 row affected (0.10 sec)

mysql> insert into seller values (22, "Bingo");
Query OK, 1 row affected (0.10 sec)

mysql> select * from seller;
+-----------+-------+
| seller_id | name  |
+-----------+-------+
|        21 | Yoyo  |
|        22 | Bingo |
+-----------+-------+
2 rows in set (0.00 sec)
```

Insertion in Seller table.

```
mysql> insert into orders values (11, 25, 1, 22);
Query OK, 1 row affected (0.10 sec)

mysql> insert into orders values (12, 50, 2, 21);
Query OK, 1 row affected (0.10 sec)

mysql> select * from orders
    -> ;
+----------+-------+---------+-----------+
| order_id | price | cust_id | seller_id |
+----------+-------+---------+-----------+
|       11 |    25 |       1 |        22 |
|       12 |    50 |       2 |        21 |
+----------+-------+---------+-----------+
2 rows in set (0.00 sec)
```

Insertion in Orders table.

4. **Find the sales person have multiple orders.**

A. To display the details of the seller who has multiple orders, we will group the data by seller_id and display the results where the count is more than one.

Command : SELECT *o.seller_id, s.name*, COUNT(*) FROM *orders o, seller* s WHERE *o.seller_id = s.seller_id* GROUP BY *o.seller_id* HAVING COUNT(*) > 1;

```
mysql> select o.seller_id, s.name, count(*)  from  orders o, seller s where o.seller_id
=s.seller_id group by o.seller_id having count(*)>1;
+-----------+------+----------+
| seller_id | name | count(*) |
+-----------+------+----------+
|        21 | Yoyo |        2 |
+-----------+------+----------+
1 row in set (0.00 sec)
```

5. **Find the all sales person details along with order details.**

A. By using where clause we can filter the data before projecting it.

```
mysql> select s.seller_id, s.name, o.order_id, o.price, o.cust_id from seller s, orders
 o  where s.seller_id=o.seller_id;
+-----------+-------+----------+-------+---------+
| seller_id | name  | order_id | price | cust_id |
+-----------+-------+----------+-------+---------+
|        22 | Bingo |       11 |    25 |       1 |
|        21 | Yoyo  |       12 |    50 |       2 |
|        21 | Yoyo  |       13 |    75 |       2 |
+-----------+-------+----------+-------+---------+
3 rows in set (0.00 sec)

mysql>
```

6. **Create index.**

A. Command used: create index index_name on tableName(columnName)

```
mysql> create index price_index on orders (price);
Query OK, 0 rows affected (0.40 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

7. **How to show index on a table**
    A. Command used: show index from tableName

```
mysql> show index  from  orders;
+--------+------------+-------------+--------------+-------------+-----------+---------
----+----------+--------+------+------------+---------+---------------+
| Table  | Non_unique | Key_name    | Seq_in_index | Column_name | Collation | Cardinal
ity | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+--------+------------+-------------+--------------+-------------+-----------+---------
----+----------+--------+------+------------+---------+---------------+
| orders |          0 | PRIMARY     |            1 | order_id    | A         |
  2 |     NULL | NULL   |      | BTREE      |         |               |
| orders |          1 | cust_id     |            1 | cust_id     | A         |
  2 |     NULL | NULL   | YES  | BTREE      |         |               |
| orders |          1 | seller_id   |            1 | seller_id   | A         |
  2 |     NULL | NULL   | YES  | BTREE      |         |               |
| orders |          1 | price_index |            1 | price       | A         |
  3 |     NULL | NULL   | YES  | BTREE      |         |               |
+--------+------------+-------------+--------------+-------------+-----------+---------
----+----------+--------+------+------------+---------+---------------+
4 rows in set (0.00 sec)
```

8. **Find the order number, sales person name, along with the customer to whom that order belongs to.**

A. Command used: SELECT *o.order_id* AS *order_number, s.name* AS *seller_name, c.name* AS *customer* FROM *orders o, seller s, customer c* WHERE *s.seller_id = o.seller_id && o.cust_id = c.cust_id;*

```
mysql> select o.order_id as order_number, s.name as seller_name, c.name as customer fro
m orders o, seller s, customer c where s.seller_id=o.seller_id && o.cust_id=c.cust_id;
+--------------+-------------+----------+
| order_number | seller_name | customer |
+--------------+-------------+----------+
|           11 | Bingo       | Alex     |
|           12 | Yoyo        | Harry    |
|           13 | Yoyo        | Harry    |
+--------------+-------------+----------+
3 rows in set (0.00 sec)
```

Usage of where clause with &&.