Submitted by Mohit Dhande

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

```
MySQL  localhost:33060+ ssl  SQL > create database inventory;
Query OK, 1 row affected (0.0372 sec)
MySQL  localhost:33060+ ssl  SQL > use inventory;
Default schema set to `inventory`.
```

2. Design Schema

```
+----------------------+
| Tables_in_inventory  |
+----------------------+
| customer             |
| orders               |
| salesperson          |
+----------------------+
```

```
MySQL  localhost:33060+ ssl  inventory  SQL > select * from orders;
+----------+--------+---------+----------+
| order_id | amount | cust_id | sales_id |
+----------+--------+---------+----------+
|      101 |    100 |       1 |        2 |
|      102 |    450 |       1 |        1 |
|      103 |    450 |       2 |        1 |
+----------+--------+---------+----------+
3 rows in set (0.0012 sec)
MySQL  localhost:33060+ ssl  inventory  SQL > select * from salesperson;
+----------------+------------------+
| salesperson_id | salesperson_name |
+----------------+------------------+
|              1 | Bob              |
|              2 | Andy             |
+----------------+------------------+
2 rows in set (0.0012 sec)
MySQL  localhost:33060+ ssl  inventory  SQL > select * from customer;
+-------------+------------+
| customer_id | first_name |
+-------------+------------+
|           1 | Tom        |
|           2 | Jerry      |
+-------------+------------+
```

3. Create tables

```
MySQL localhost:33060+ ssl inventory SQL > create table customer(
                                         -> customer_id int not null,
                                         -> first_name varchar(30),
                                         -> primary key (customer_id));
Query OK, 0 rows affected (0.0644 sec)
```

```
MySQL localhost:33060+ ssl inventory SQL > create table salesperson(
                                        -> salesperson_id int not null,
                                        -> salesperson_name varchar(30),
                                        -> primary key (salesperson_id));
Query OK, 0 rows affected (0.0676 sec)
```

```
MySQL localhost:33060+ ssl inventory SQL > create table orders(
                                       -> order_id int not null,
                                       ->   amount int,
                                       ->   cust_id int,
                                       ->   sales_id int,
                                       ->   PRIMARY KEY (order_id),
                                       ->     FOREIGN KEY (cust_id) REFERENCES customer(customer_id),
                                       ->   FOREIGN KEY (sales_id) REFERENCES salesperson(salesperson_id)
                                       -> );
```

4. Insert sample data

```
MySQL localhost:33060+ ssl inventory SQL > INSERT INTO customer (customer_id, first_name) VALUES ('1', 'Tom');
Query OK, 1 row affected (0.0378 sec)
MySQL localhost:33060+ ssl inventory SQL > INSERT INTO customer (customer_id, first_name) VALUES ('2', 'Jerry');
Query OK, 1 row affected (0.0043 sec)
MySQL localhost:33060+ ssl inventory SQL >
MySQL localhost:33060+ ssl inventory SQL > INSERT INTO salesperson (salesperson_id, salesperson_name) VALUES ('1', 'Bob');
Query OK, 1 row affected (0.0076 sec)
MySQL localhost:33060+ ssl inventory SQL > INSERT INTO salesperson (salesperson_id, salesperson_name) VALUES ('2', 'Andy');
Query OK, 1 row affected (0.0040 sec)
MySQL localhost:33060+ ssl inventory SQL >
MySQL localhost:33060+ ssl inventory SQL > INSERT INTO orders (order_id, amount, cust_id, sales_id) VALUES ('101', '100', '1', '2');
Query OK, 1 row affected (0.0085 sec)
MySQL localhost:33060+ ssl inventory SQL > INSERT INTO orders (order_id, amount, cust_id, sales_id) VALUES ('102', '450', '1', '1');
Query OK, 1 row affected (0.0045 sec)
MySQL localhost:33060+ ssl inventory SQL > INSERT INTO orders (order_id, amount, cust_id, sales_id) VALUES ('103', '450', '2', '1');
Query OK, 1 row affected (0.0058 sec)
```

5. Find the sales person have multiple orders.

```
MySQL localhost:33060+ ssl inventory SQL > select sp.salesperson_id, sp.salesperson_name
                                         -> from salesperson sp , orders o
                                         -> where sp.salesperson_id = o.sales_id
                                         -> group by sp.salesperson_id
                                         -> having count(sp.salesperson_id) > 1;
+----------------+------------------+
| salesperson_id | salesperson_name |
+----------------+------------------+
|              1 | Bob              |
+----------------+------------------+
```

6. Find the all sales person details along with order details

```
MySQL  localhost:33060+ ssl  inventory  SQL > select sp.salesperson_id, sp.salesperson_name, o.order_id, o.amount, o.cust_id
                                           -> from salesperson sp , orders o
                                           -> where sp.salesperson_id = o.sales_id;
+----------------+------------------+----------+--------+---------+
| salesperson_id | salesperson_name | order_id | amount | cust_id |
+----------------+------------------+----------+--------+---------+
|              2 | Andy             |      101 |    100 |       1 |
|              1 | Bob              |      102 |    450 |       1 |
|              1 | Bob              |      103 |    450 |       2 |
+----------------+------------------+----------+--------+---------+
```

7. Create index

```
MySQL  localhost:33060+ ssl  inventory  SQL > CREATE INDEX index_on_cust_name ON customer (first_name);
Query OK, 0 rows affected (0.0923 sec)
```

8. How to show index on a table

```
MySQL  localhost:33060+ ssl  inventory  SQL > SHOW INDEX FROM customer;
+----------+------------+--------------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------
-------+------------+
| Table    | Non_unique | Key_name           | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment
Visible | Expression |
+----------+------------+--------------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------
-------+------------+
| customer |          0 | PRIMARY            |            1 | customer_id | A         |           2 |     NULL | NULL   |      | BTREE      |
YES     | NULL       |
| customer |          1 | index_on_cust_name |            1 | first_name  | A         |           2 |     NULL | NULL   | YES  | BTREE      |
YES     | NULL       |
+----------+------------+--------------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------
-------+------------+
2 rows in set (0.0089 sec)
```

9. Find the order number, sale person name, along with the cu stomer to whom that order belongs to

```
MySQL  localhost:33060+ ssl  inventory  SQL > select o.order_id, sp.salesperson_name, c.first_name
                                           -> from orders o, customer c, salesperson sp
                                           -> where c.customer_id = o.cust_id and o.sales_id = sp.salesperson_id;
+----------+------------------+------------+
| order_id | salesperson_name | first_name |
+----------+------------------+------------+
|      101 | Andy             | Tom        |
|      102 | Bob              | Tom        |
|      103 | Bob              | Jerry      |
+----------+------------------+------------+
3 rows in set (0.0058 sec)
```