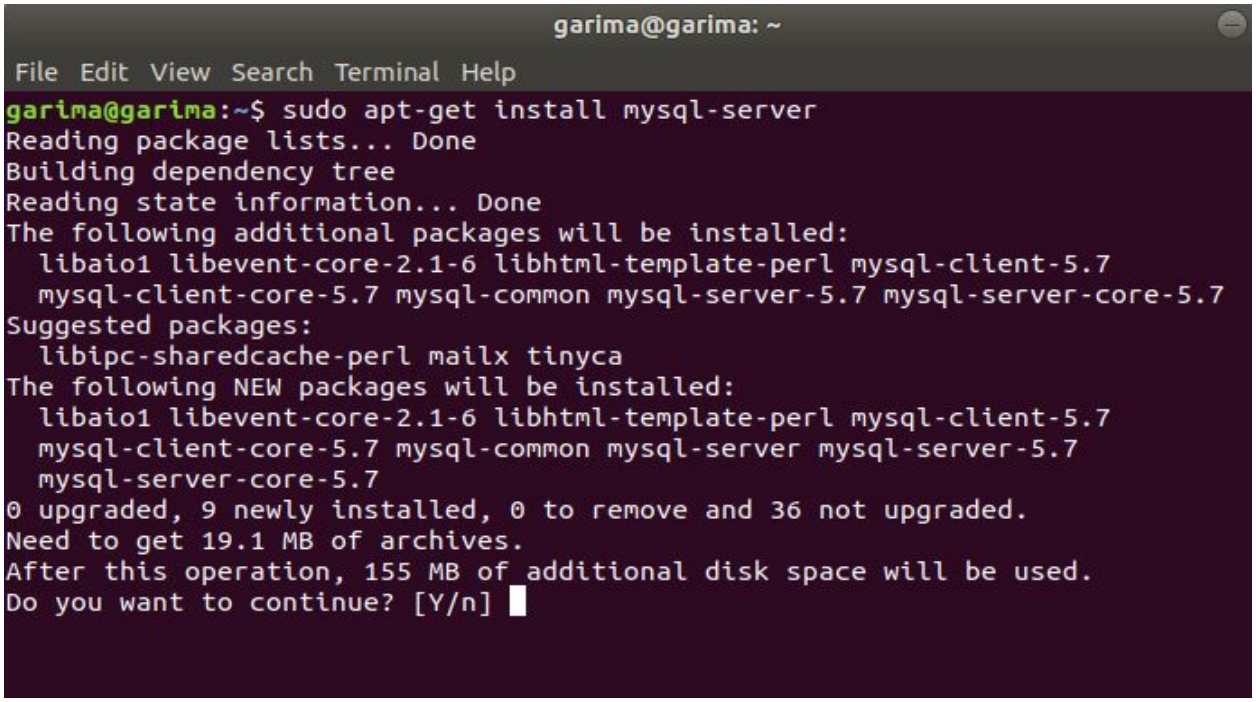
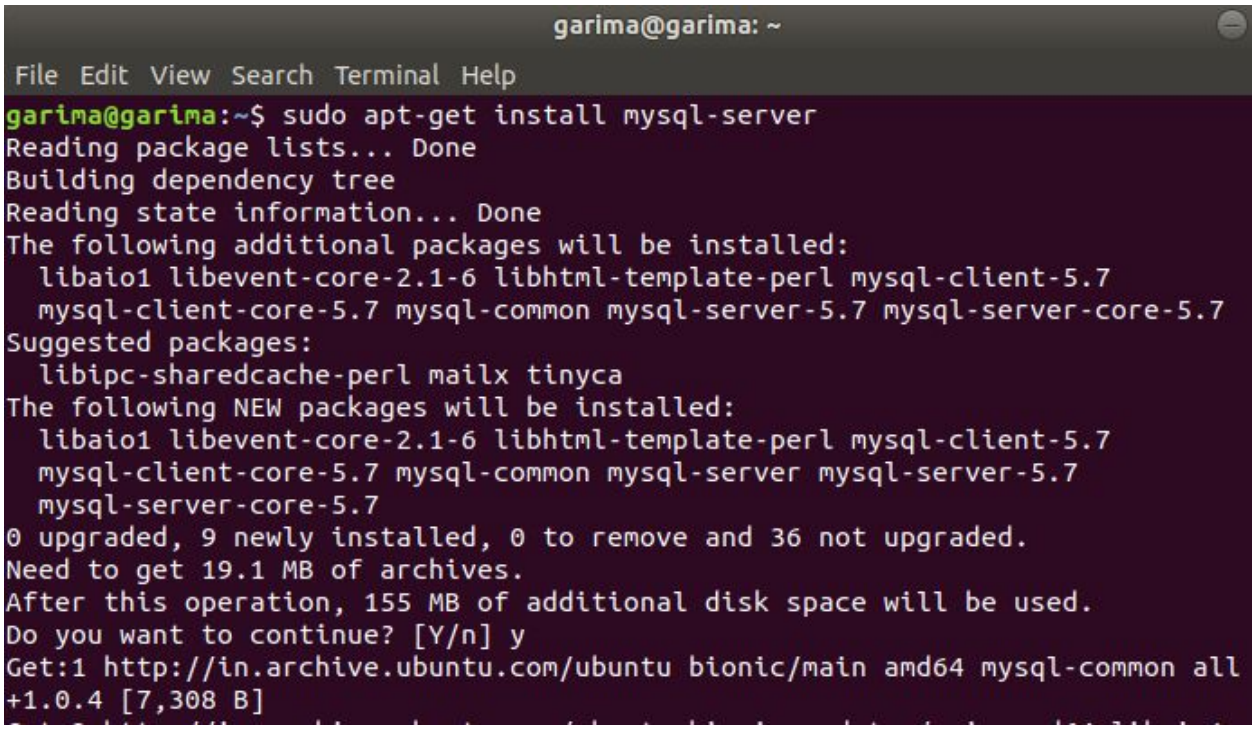
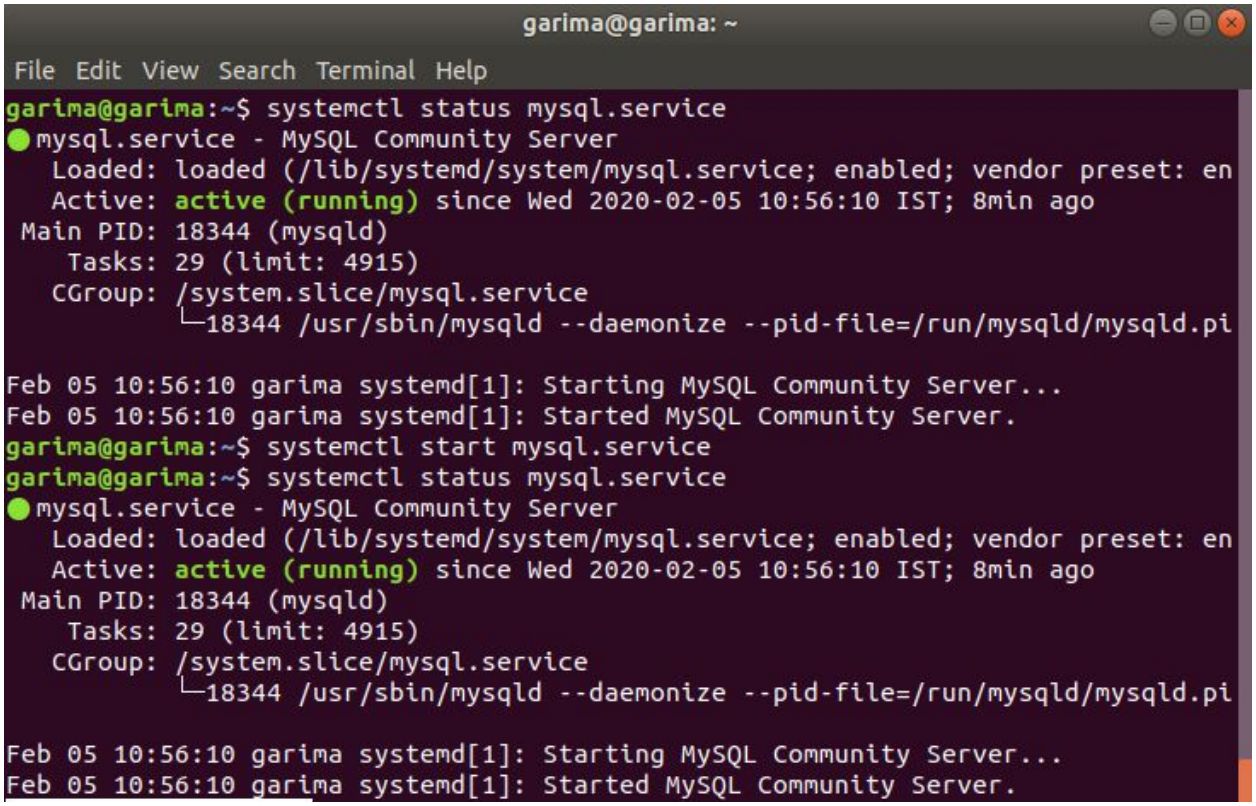
ASSESSMENT - 3

Introduction to Databases

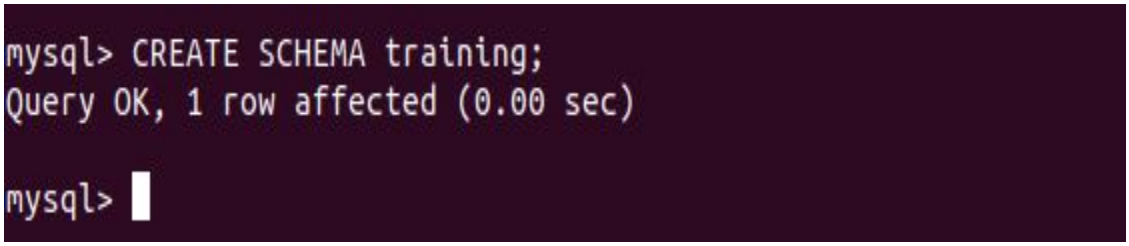
Problem Statement: There can be multiple customers, who can place multiple orders on the site. Now a sales person can handle these orders will distribute into multiple sales persons (One order will be assign to one salesperson only). So a sales person can have multiple orders of multiple customers



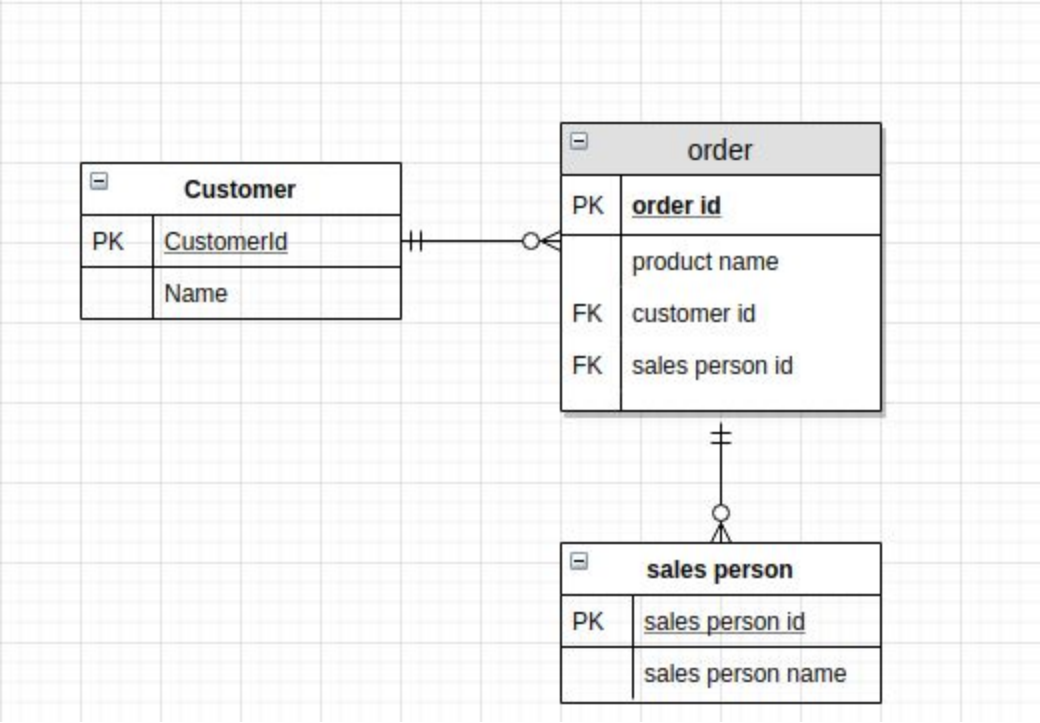




1. **Create Database**



2. Design Schema



3. Create tables

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| garima             |
| mysql              |
| performance_schema |
| sys                |
| training           |
+--------------------+
6 rows in set (0.00 sec)

mysql> USE garima;
Database changed
mysql> CREATE TABLE CUSTOMER_TAB(cust_id INT PRIMARY KEY, cust_name VARCHAR(100));
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE SALES_PERSON(salesp_id INT PRIMARY KEY, salesp_name VARCHAR(100));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE CUST_ORDERS(order_id INT PRIMARY KEY, pro_name VARCHAR(100),
 f_custid INT, f_salespid INT, FOREIGN KEY(f_custid) REFERENCES CUSTOMER_TAB(cus
t_id),FOREIGN KEY(f_salespid) REFERENCES SALES_PERSON(salesp_id));
Query OK, 0 rows affected (0.05 sec)

mysql>
```

```
mysql> SHOW TABLES;
+-------------------+
| Tables_in_garima  |
+-------------------+
| CUSTOMER_TAB      |
| CUST_ORDERS       |
| ORDERS            |
| SALES_PERSON      |
+-------------------+
4 rows in set (0.00 sec)
```

4. **Insert sample data**

```
mysql> INSERT INTO CUSTOMER_TAB VALUES(1,"Joe"),(2,"Daniel"),(3,"Ajay"),(4,"Joey
");
Query OK, 4 rows affected (0.03 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> INSERT INTO SALES_PERSON VALUES(1,"SP1"),(2,"SP2"),(3,"SP3"),(4,"SP4");
Query OK, 4 rows affected (0.04 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> INSERT INTO CUST_ORDERS VALUES(1,"Blanket",2,4),(2,"Table",1,1),(3,"Book"
,3,2),(4,"Bottle",4,3);
Query OK, 4 rows affected (0.04 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql>
```

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

```
mysql> SELECT salesp_name
    -> FROM SALES_PERSON
    -> INNER JOIN CUST_ORDERS
    -> ON SALES_PERSON.salesp_id=CUST_ORDERS.f_salespid
    -> GROUP BY f_salespid
    -> HAVING COUNT(order_id) > 1;
+-------------+
| salesp_name |
+-------------+
| SP4         |
+-------------+
1 row in set (0.00 sec)

mysql>
```

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

```
mysql> SELECT salesp_name, salesp_id, order_id, pro_name
    -> FROM SALES_PERSON
    -> INNER JOIN CUST_ORDERS
    -> WHERE SALES_PERSON.salesp_id=CUST_ORDERS.f_salespid;
+-------------+-----------+----------+----------+
| salesp_name | salesp_id | order_id | pro_name |
+-------------+-----------+----------+----------+
| SP1         |         1 |        2 | Table    |
| SP2         |         2 |        3 | Book     |
| SP3         |         3 |        4 | Bottle   |
| SP4         |         4 |        1 | Blanket  |
| SP4         |         4 |        5 | Bag      |
| SP4         |         4 |        6 | Bottle   |
+-------------+-----------+----------+----------+
6 rows in set (0.00 sec)
```

7. **Create index**

```
mysql> CREATE INDEX Index_Key ON SALES_PERSON(salesp_id);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

8. **How to show index on a table**

```
mysql> desc SALES_PERSON
    -> ;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| salesp_id   | int(11)      | NO   | PRI | NULL    |       |
| salesp_name | varchar(100) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

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

```
mysql> SELECT order_id, salesp_name, cust_name
    -> FROM CUST_ORDERS
    -> INNER JOIN SALES_PERSON
    -> ON SALES_PERSON.salesp_id=CUST_ORDERS.f_salespid
    -> INNER JOIN CUSTOMER_TAB
    -> ON CUSTOMER_TAB.cust_id=CUST_ORDERS.f_custid;
+----------+-------------+-----------+
| order_id | salesp_name | cust_name |
+----------+-------------+-----------+
|        2 | SP1         | Joe       |
|        1 | SP4         | Daniel    |
|        5 | SP4         | Daniel    |
|        3 | SP2         | Ajay      |
|        6 | SP4         | Ajay      |
|        4 | SP3         | Joey      |
+----------+-------------+-----------+
6 rows in set (0.00 sec)
```