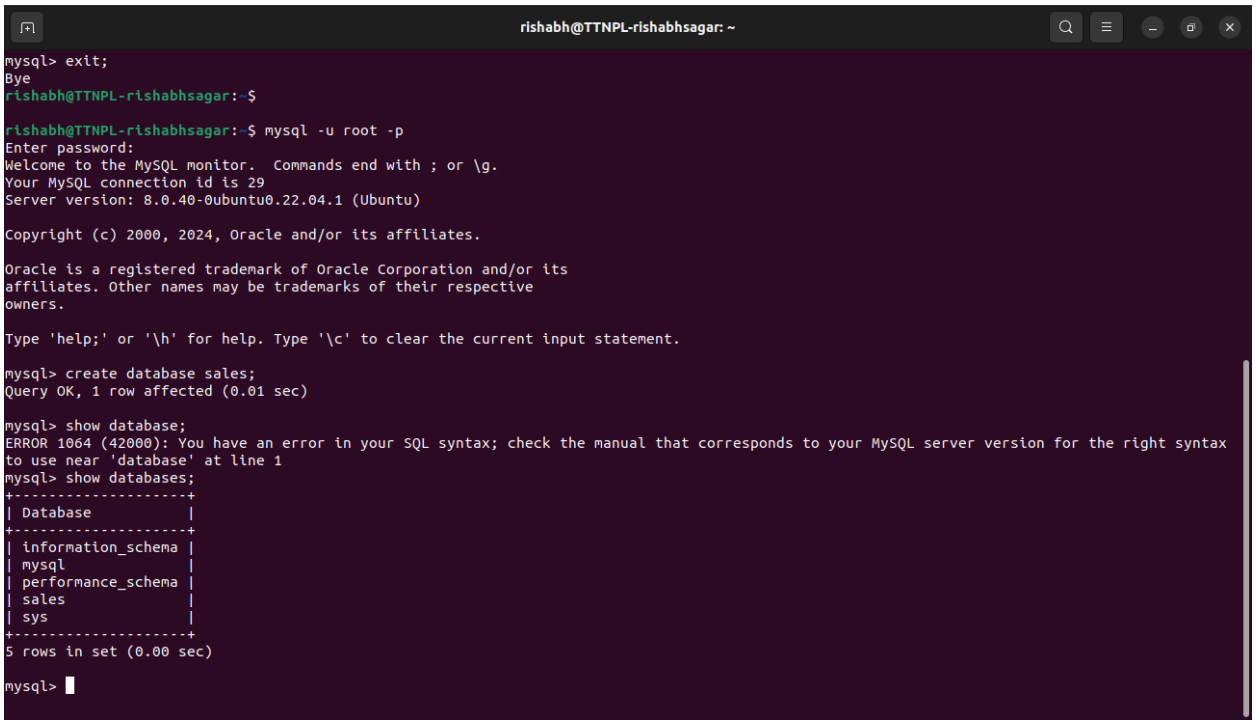
Introduction to Databases

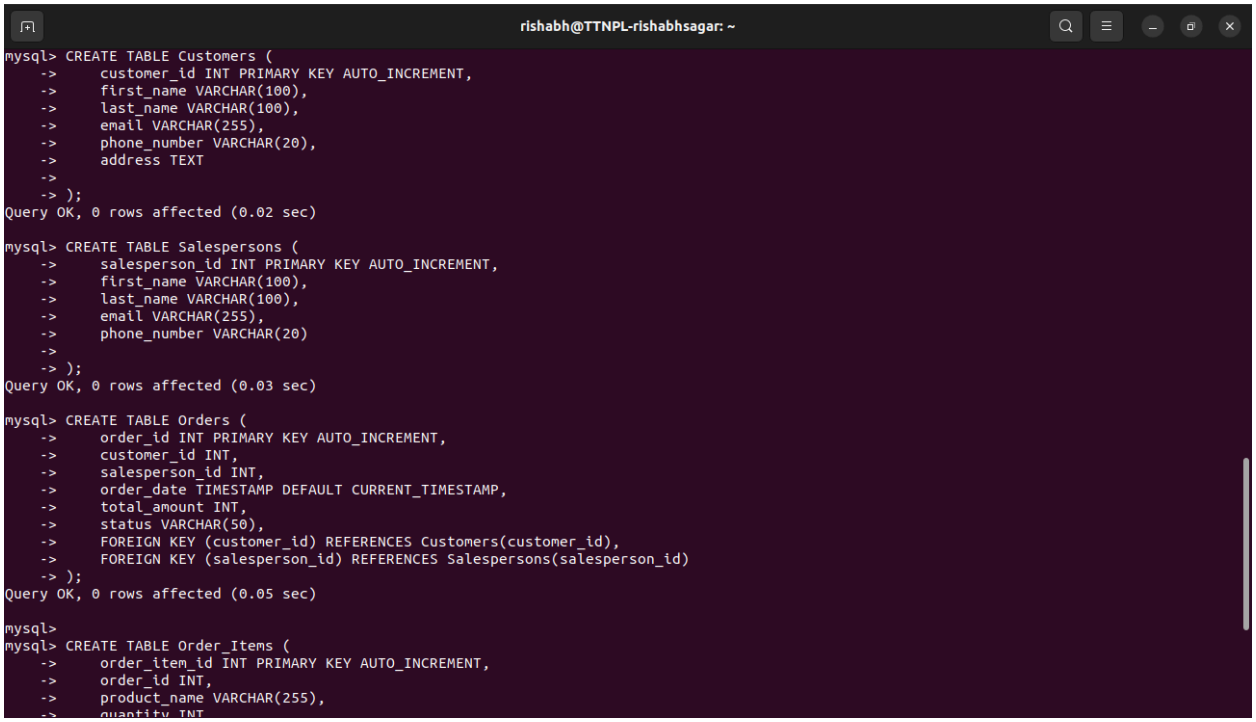
Q1)Problem Statement: There can be multiple customers, who can place multiple orders on the site. Now a sales person can handle these orders will distribute into multiple sales persons (One order will be assign to one salesperson only). So a sales person can have multiple orders of multiple customers Create Database

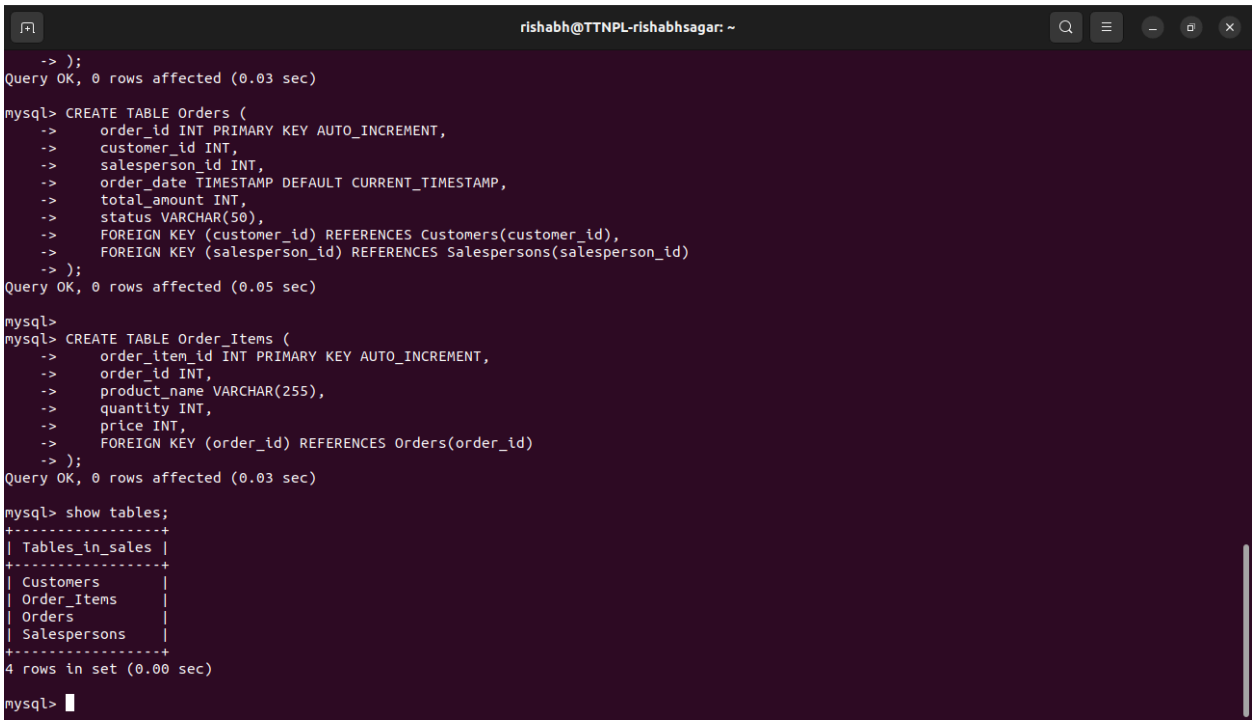
Ans. Cmd — create database database_name
To create the database you have to first login the the dbms and then run the cmd and to verify whether the database is created or not run the 'show databases' if the name is present that means database is created to use that database run 'use database_name'



Q2)Design Schema
Ans. Customers: stores customer data.
Salespersons: stores salesperson data.
Orders: stores order data and has foreign keys linking to both customers and salespersons.

Customers: Customers can place multiple orders.
Orders: Each order is linked to a specific customer and a specific salesperson.
Salespersons: Salespersons handle orders and may handle orders from multiple customers.

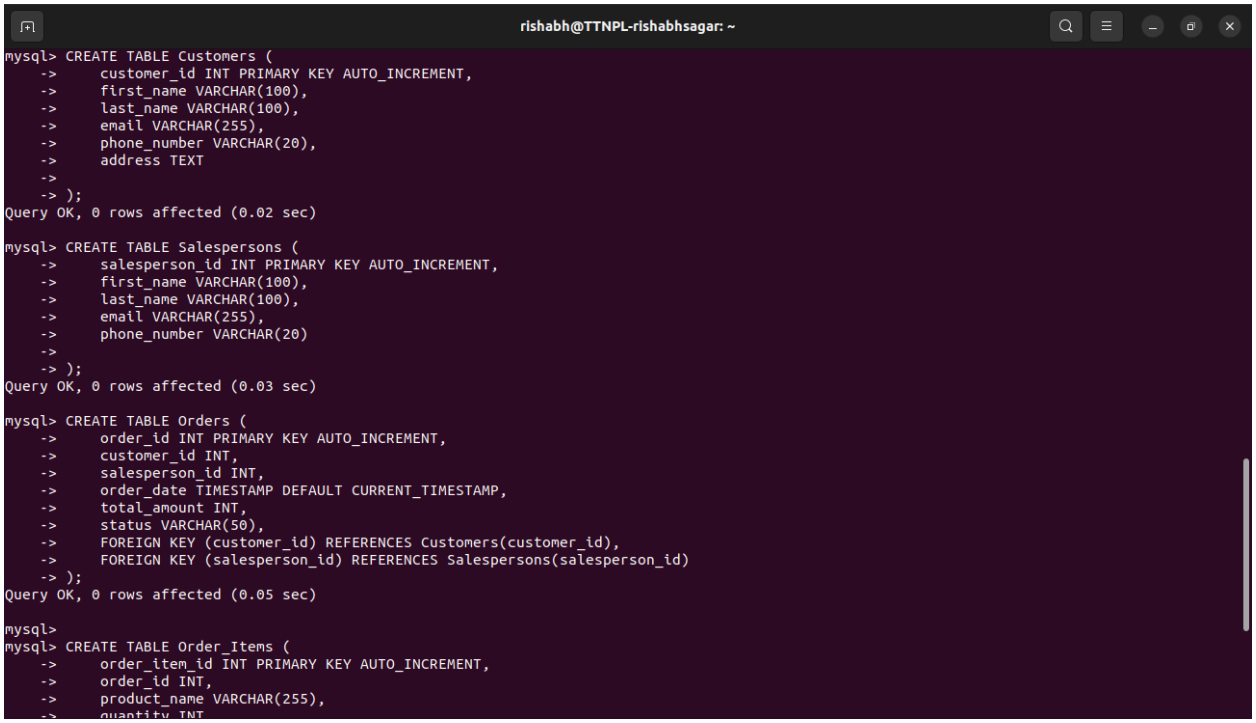


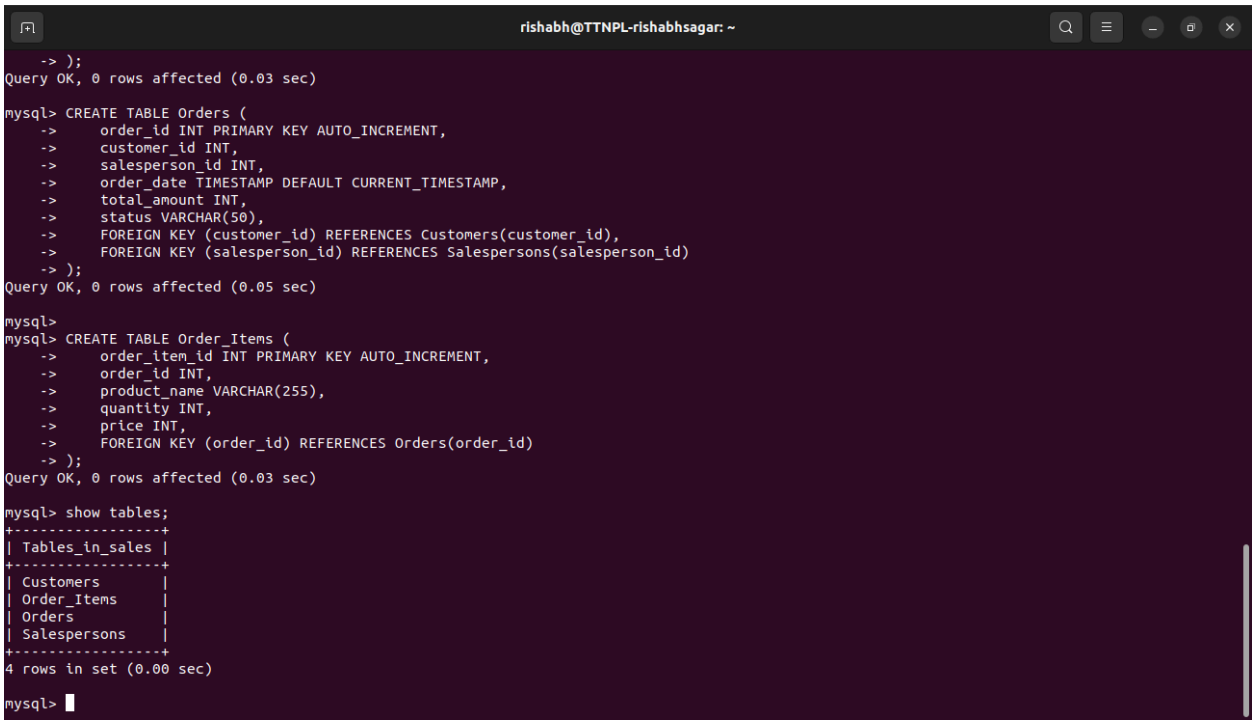


Q3)Create tables

Ans. Cmd — CREATE TABLE table_name (column1 datatype constraint, column2 datatyp e constraint, column3 datatype constraint, ...);

This will create the table in the particular database(you have to create the database and select the database first Q1 to make the table). In a database we can make ‘n’ number of tables.

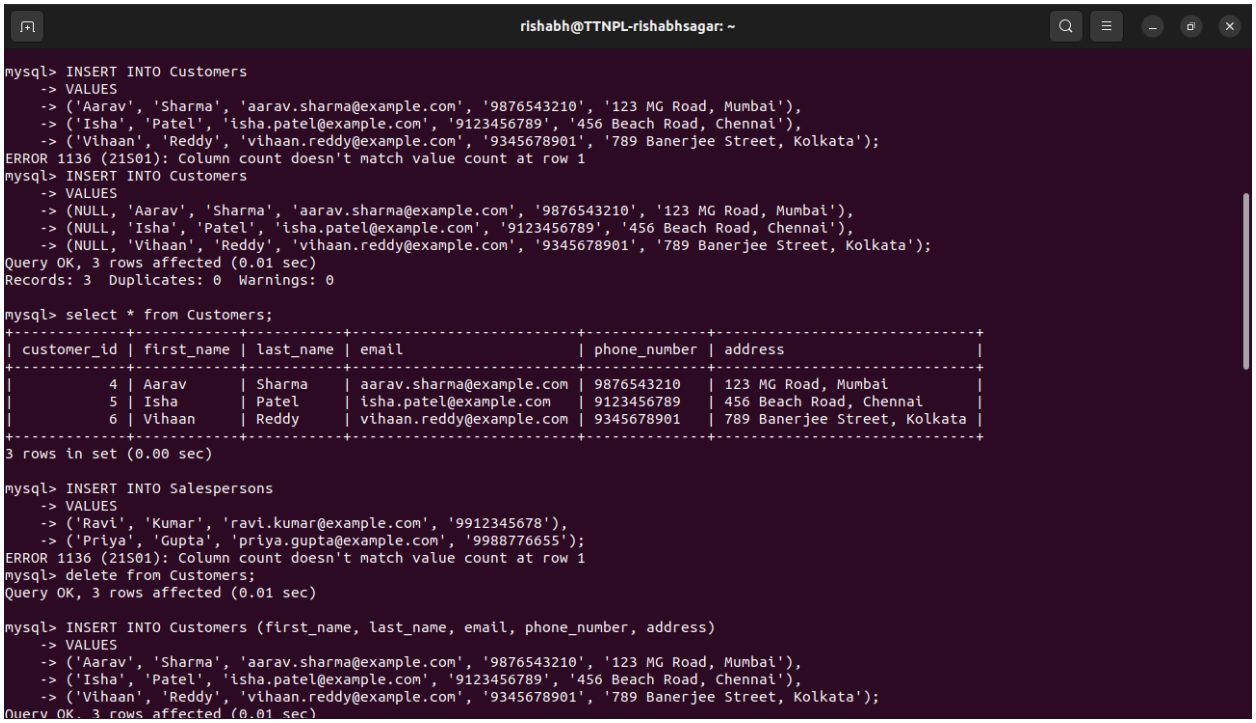




Q4)Insert sample data

Ans. Cmd —INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...)

We can omit (column1, column2, column3, ...) if we know the sequence of filling the data in the table, also we can insert multiple rows at the same time just putting comma after each tuple data as we see in the screenshot.



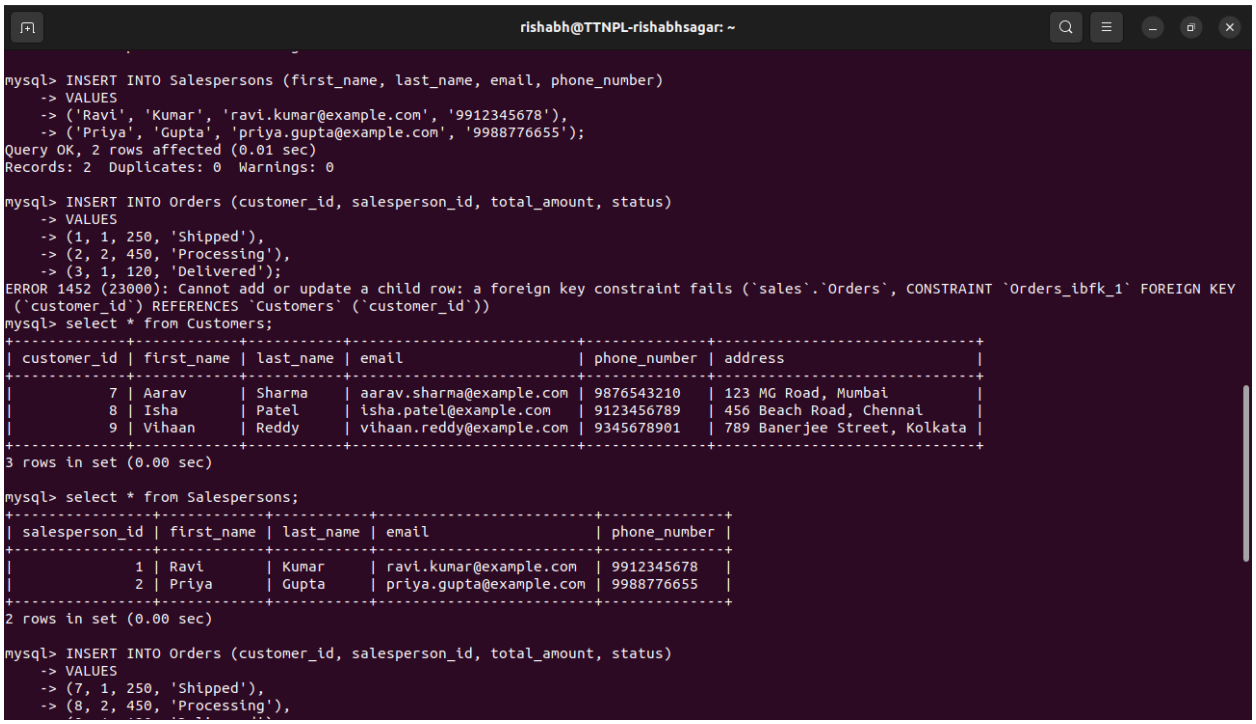
```
rishabh@TTNPL-rishabhsagar: ~

mysql> INSERT INTO Customers
    -> VALUES
    -> ('Aarav', 'Sharma', 'aarav.sharma@example.com', '9876543210', '123 MG Road, Mumbai'),
    -> ('Isha', 'Patel', 'isha.patel@example.com', '9123456789', '456 Beach Road, Chennai'),
    -> ('Vihaan', 'Reddy', 'vihaan.reddy@example.com', '9345678901', '789 Banerjee Street, Kolkata');
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> INSERT INTO Customers
    -> VALUES
    -> (NULL, 'Aarav', 'Sharma', 'aarav.sharma@example.com', '9876543210', '123 MG Road, Mumbai'),
    -> (NULL, 'Isha', 'Patel', 'isha.patel@example.com', '9123456789', '456 Beach Road, Chennai'),
    -> (NULL, 'Vihaan', 'Reddy', 'vihaan.reddy@example.com', '9345678901', '789 Banerjee Street, Kolkata');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from Customers;
+-------------+------------+-----------+-------------------------+--------------+-----------------------------+
| customer_id | first_name | last_name | email                   | phone_number | address                     |
+-------------+------------+-----------+-------------------------+--------------+-----------------------------+
|           4 | Aarav      | Sharma    | aarav.sharma@example.com | 9876543210   | 123 MG Road, Mumbai         |
|           5 | Isha       | Patel     | isha.patel@example.com   | 9123456789   | 456 Beach Road, Chennai     |
|           6 | Vihaan     | Reddy     | vihaan.reddy@example.com | 9345678901   | 789 Banerjee Street, Kolkata |
+-------------+------------+-----------+-------------------------+--------------+-----------------------------+
3 rows in set (0.00 sec)

mysql> INSERT INTO Salespersons
    -> VALUES
    -> ('Ravi', 'Kumar', 'ravi.kumar@example.com', '9912345678'),
    -> ('Priya', 'Gupta', 'priya.gupta@example.com', '9988776655');
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> delete from Customers;
Query OK, 3 rows affected (0.01 sec)

mysql> INSERT INTO Customers (first_name, last_name, email, phone_number, address)
    -> VALUES
    -> ('Aarav', 'Sharma', 'aarav.sharma@example.com', '9876543210', '123 MG Road, Mumbai'),
    -> ('Isha', 'Patel', 'isha.patel@example.com', '9123456789', '456 Beach Road, Chennai'),
    -> ('Vihaan', 'Reddy', 'vihaan.reddy@example.com', '9345678901', '789 Banerjee Street, Kolkata');
Query OK, 3 rows affected (0.01 sec)
```



```
rishabh@TTNPL-rishabhsagar: ~

mysql> INSERT INTO Salespersons (first_name, last_name, email, phone_number)
    -> VALUES
    -> ('Ravi', 'Kumar', 'ravi.kumar@example.com', '9912345678'),
    -> ('Priya', 'Gupta', 'priya.gupta@example.com', '9988776655');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Orders (customer_id, salesperson_id, total_amount, status)
    -> VALUES
    -> (1, 1, 250, 'Shipped'),
    -> (2, 2, 450, 'Processing'),
    -> (3, 1, 120, 'Delivered');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`sales`.`Orders`, CONSTRAINT `Orders_ibfk_1` FOREIGN KEY
 (`customer_id`) REFERENCES `Customers` (`customer_id`))
mysql> select * from Customers;
+-------------+------------+-----------+-------------------------+--------------+-----------------------------+
| customer_id | first_name | last_name | email                   | phone_number | address                     |
+-------------+------------+-----------+-------------------------+--------------+-----------------------------+
|           7 | Aarav      | Sharma    | aarav.sharma@example.com | 9876543210   | 123 MG Road, Mumbai         |
|           8 | Isha       | Patel     | isha.patel@example.com   | 9123456789   | 456 Beach Road, Chennai     |
|           9 | Vihaan     | Reddy     | vihaan.reddy@example.com | 9345678901   | 789 Banerjee Street, Kolkata |
+-------------+------------+-----------+-------------------------+--------------+-----------------------------+
3 rows in set (0.00 sec)

mysql> select * from Salespersons;
+---------------+------------+-----------+-------------------------+--------------+
| salesperson_id | first_name | last_name | email                   | phone_number |
+---------------+------------+-----------+-------------------------+--------------+
|             1 | Ravi       | Kumar     | ravi.kumar@example.com  | 9912345678   |
|             2 | Priya      | Gupta     | priya.gupta@example.com | 9988776655   |
+---------------+------------+-----------+-------------------------+--------------+
2 rows in set (0.00 sec)

mysql> INSERT INTO Orders (customer_id, salesperson_id, total_amount, status)
    -> VALUES
    -> (7, 1, 250, 'Shipped'),
    -> (8, 2, 450, 'Processing'),
```

```
rishabh@TTNPL-rishabhsagar: ~
2 rows in set (0.00 sec)

mysql> INSERT INTO Orders (customer_id, salesperson_id, total_amount, status)
    -> VALUES
    -> (7, 1, 250, 'Shipped'),
    -> (8, 2, 450, 'Processing'),
    -> (9, 1, 120, 'Delivered');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO Order_Items (order_id, product_name, quantity, price)
    -> VALUES
    -> (1, 'Laptop', 1, 250),
    -> (2, 'Smartphone', 2, 225),
    -> (3, 'Headphones', 1, 120);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`sales`.`Order_Items`, CONSTRAINT `Order_Items_ibfk_1` F
OREIGN KEY (`order_id`) REFERENCES `Orders` (`order_id`))
mysql> select * from Orders;
+----------+-------------+---------------+---------------------+--------------+------------+
| order_id | customer_id | salesperson_id | order_date          | total_amount | status     |
+----------+-------------+---------------+---------------------+--------------+------------+
|        4 |           7 |             1 | 2025-01-27 14:32:05 |          250 | Shipped    |
|        5 |           8 |             2 | 2025-01-27 14:32:05 |          450 | Processing |
|        6 |           9 |             1 | 2025-01-27 14:32:05 |          120 | Delivered  |
+----------+-------------+---------------+---------------------+--------------+------------+
3 rows in set (0.00 sec)

mysql> INSERT INTO Order_Items (order_id, product_name, quantity, price)
    -> VALUES
    -> (4, 'Laptop', 1, 250),
    -> (5, 'Smartphone', 2, 225),
    -> (6, 'Headphones', 1, 120);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql>
```

Q5)Find the sales person have multiple orders.

Ans. To find the salesperson having multiple order we have to first join the salesperson and order table so that we get the mapping for the salesperson and their respective order, then we can just use the group by sales id not by salesperson name as name can be same, this will accumulate the order for the particular salesperson that we can just put the condition of count(o.order_id)>1 using HAVING(used as where but with group by).

```
mysql> SELECT sp.salesperson_id, sp.first_name, sp.last_name, COUNT(o.order_id) AS num_orders
    -> FROM Salespersons sp
    -> JOIN Orders o ON sp.salesperson_id = o.salesperson_id
    -> GROUP BY sp.salesperson_id
    -> HAVING COUNT(o.order_id) > 1;
+----------------+------------+-----------+------------+
| salesperson_id | first_name | last_name | num_orders |
+----------------+------------+-----------+------------+
|              1 | Ravi       | Kumar     |          2 |
+----------------+------------+-----------+------------+
1 row in set (0.01 sec)
```

Q6)Find the all sales person details along with order details

Ans. To find the sales person details with order details we have to join the table as it will map the table with another table after that we can select the column that we need.

Cmd– select name1,name2 …. From table1 left join table2 on table1.column=table2.column

LEFT JOIN takes all the field & column from the table1 and map it to table2, if mapping is not found in table2 for the particular field, the columns field is filled with NULL of the table2 with respect to the given field.

```
mysql> SELECT
    ->     sp.salesperson_id,
    ->     sp.first_name AS salesperson_first_name,
    ->     sp.last_name AS salesperson_last_name,
    ->     sp.email AS salesperson_email,
    ->     sp.phone_number AS salesperson_phone_number,
    ->     o.order_id,
    ->     o.order_date,
    ->     o.total_amount,
    ->     o.status
    -> FROM
    ->     Salespersons sp
    -> LEFT JOIN
    ->     Orders o ON sp.salesperson_id = o.salesperson_id;
+----------------+----------------------+----------------------+------------------------+--------------------------+----------+-----------
----------+-------------+-----------+
| salesperson_id | salesperson_first_name | salesperson_last_name | salesperson_email      | salesperson_phone_number | order_id | order_date
          | total_amount | status    |
+----------------+----------------------+----------------------+------------------------+--------------------------+----------+-----------
----------+-------------+-----------+
|              1 | Ravi                 | Kumar                | ravi.kumar@example.com | 9912345678               |        4 | 2025-01-27
 14:32:05 |          250 | Shipped   |
|              1 | Ravi                 | Kumar                | ravi.kumar@example.com | 9912345678               |        6 | 2025-01-27
 14:32:05 |          120 | Delivered |
|              2 | Priya                | Gupta                | priya.gupta@example.com | 9988776655              |        5 | 2025-01-27
 14:32:05 |          450 | Processing |
+----------------+----------------------+----------------------+------------------------+--------------------------+----------+-----------
----------+-------------+-----------+
3 rows in set (0.00 sec)

mysql>
```

Q7)Create index

Ans. index is used for the fast retrieval of the data from the table, it is created in mysql with the cmd-CREATE INDEX index_name on table_name(column name).

Advantage:

1. **Faster search query**
2. **Enhanced join operation**
3. **Uniqueness enforcement**

Disadvantage:

1. **Slower write operation**
2. **Increased storage requirement**
3. **Complex Maintenance**

```
mysql> CREATE INDEX idx_salesperson_id ON Orders(salesperson_id);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

Q8)How to show index on a table

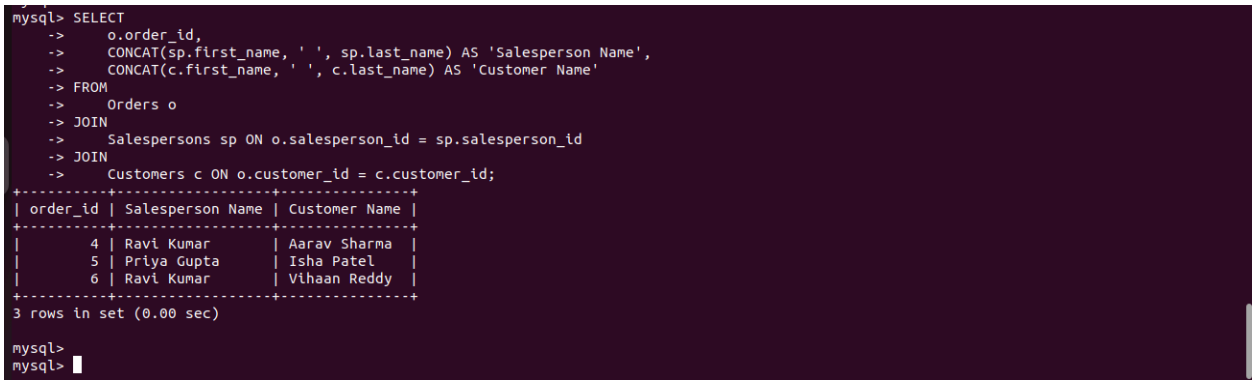
Ans. To know all the index that is present on a particular table we can use the cmd– SHOW INDEXES FROM table_name

```
mysql> SHOW INDEXES FROM Orders;
+--------+------------+--------------------+--------------+---------------+-----------+-------------+----------+--------+------+------------
---------+---------------+---------+------------+
| Table  | Non_unique | Key_name           | Seq_in_index | Column_name   | Collation | Cardinality | Sub_part | Packed | Null | Index_type |
 Comment | Index_comment | Visible | Expression |
+--------+------------+--------------------+--------------+---------------+-----------+-------------+----------+--------+------+------------
---------+---------------+---------+------------+
| Orders |          0 | PRIMARY            |            1 | order_id      | A         |           3 |     NULL |   NULL |      | BTREE      |
         |               | YES     | NULL       |
| Orders |          1 | customer_id        |            1 | customer_id   | A         |           3 |     NULL |   NULL | YES  | BTREE      |
         |               | YES     | NULL       |
| Orders |          1 | idx_salesperson_id |            1 | salesperson_id | A         |           2 |     NULL |   NULL | YES  | BTREE      |
         |               | YES     | NULL       |
+--------+------------+--------------------+--------------+---------------+-----------+-------------+----------+--------+------+------------
---------+---------------+---------+------------+
3 rows in set (0.00 sec)

mysql>
```

Q9)Find the order number, sale person name, along with the customer to whom that order belongs to

Ans. To find the given fields we have to first join the table so that we are able to know the order with respect to its salesperson and customer. Hence we join the table and from that result we select the column that we need. 'AS' is just the alias to make the name of refer the column properly while concat is used for concatenating the column result into one.



```
mysql> SELECT
    ->     o.order_id,
    ->     CONCAT(sp.first_name, ' ', sp.last_name) AS 'Salesperson Name',
    ->     CONCAT(c.first_name, ' ', c.last_name) AS 'Customer Name'
    -> FROM
    ->     Orders o
    -> JOIN
    ->     Salespersons sp ON o.salesperson_id = sp.salesperson_id
    -> JOIN
    ->     Customers c ON o.customer_id = c.customer_id;
+----------+------------------+---------------+
| order_id | Salesperson Name | Customer Name |
+----------+------------------+---------------+
|        4 | Ravi Kumar       | Aarav Sharma  |
|        5 | Priya Gupta      | Isha Patel    |
|        6 | Ravi Kumar       | Vihaan Reddy  |
+----------+------------------+---------------+
3 rows in set (0.00 sec)

mysql>
mysql>
```