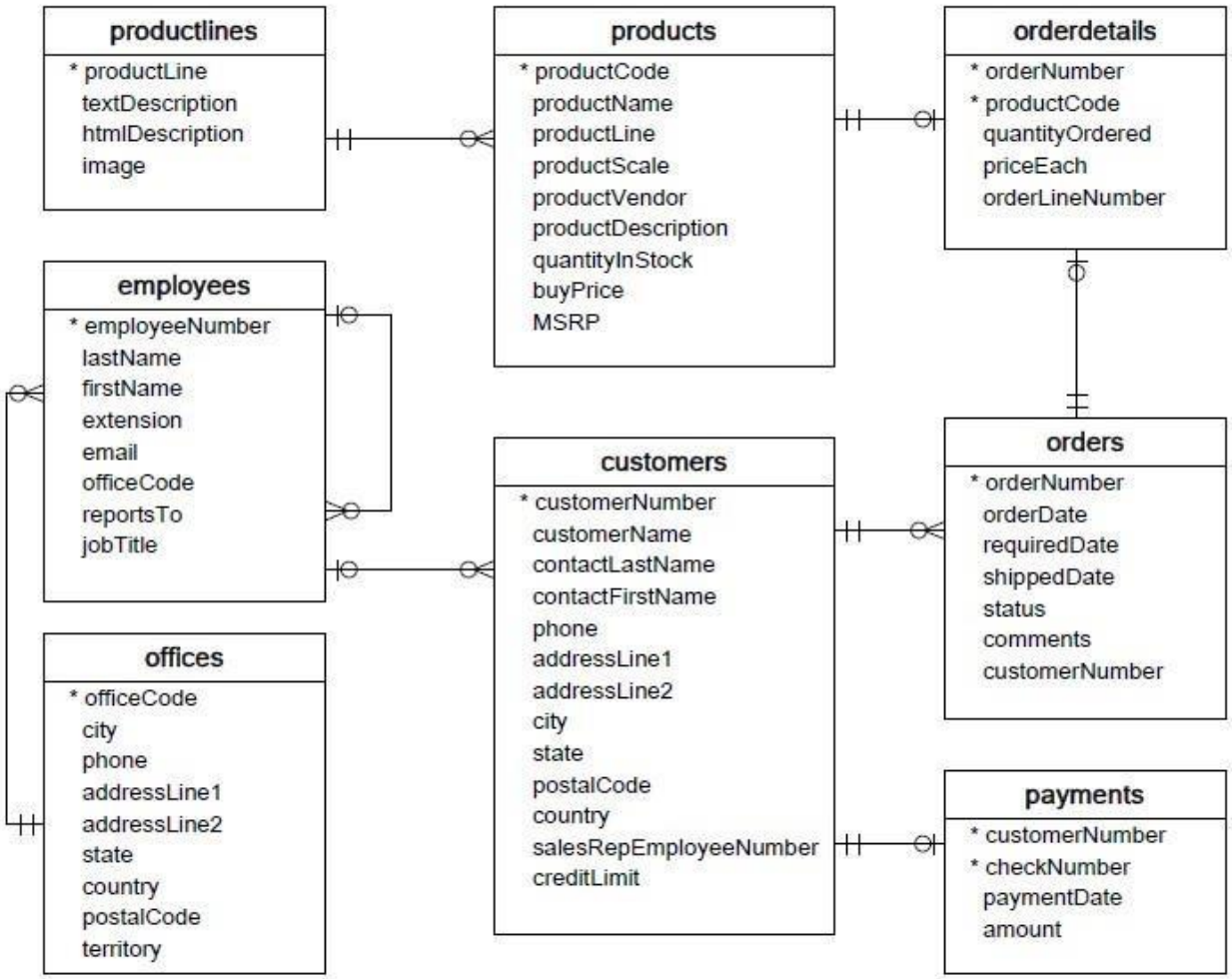
WORKSHEET 3 SQL

Refer the following ERD and answer all the questions in this worksheet. You have to write the queries using mysql for the required Operation.



- **Customers**: stores customer's data.
- **Products**: stores a list of scale model cars.
- **ProductLines**: stores a list of product line categories.
- **Orders**: stores sales orders placed by customers.
- **OrderDetails**: stores sales order line items for each sales order.
- **Payments**: stores payments made by customers based on their accounts.
- **Employees**: stores all employee information as well as the organization structure such as who reports to whom.
- **Offices**: stores sales office data.

1. Write SQL query to create table **Customers.**

Answer:-

```
DROP TABLE IF EXISTS  customers

CREATE TABLE  customers (
 customerNumber  int(11) NOT NULL,
 customerName  varchar(50) NOT NULL,
 contactLastName  varchar(50) NOT NULL,
 contactFirstName  varchar(50) NOT NULL,
 phone varchar(50) NOT NULL,
 addressLine1  varchar(50) NOT NULL,
 addressLine2  varchar(50) DEFAULT NULL,
 city  varchar(50) NOT NULL,
 state  varchar(50) DEFAULT NULL,
 postalCode  varchar(15) DEFAULT NULL,
 country  varchar(50) NOT NULL,
 salesRepEmployeeNumber  int(11) DEFAULT NULL,
 creditLimit  decimal(10,2) DEFAULT NULL,
 PRIMARY KEY (customerNumber),
 KEY  salesRepEmployeeNumber  (salesRepEmployeeNumber),
 CONSTRAINT  customers_ibfk_1 FOREIGN KEY (salesRepEmployeeNumber) REFERENCES
    employees (employeeNumber)
    )
```

2. Write SQL query to create table **Orders.**

Answer:-

```
DROP TABLE IF EXISTS  orders;

CREATE TABLE  orders (
  orderNumber  int(11) NOT NULL,
  orderDate  date NOT NULL,
  requiredDate date NOT NULL,
  shippedDate  date DEFAULT NULL,
  status varchar(15) NOT NULL,
  comments  text,
  customerNumber  int(11) NOT NULL,
  PRIMARY KEY (orderNumber),
  KEY  customerNumber  (customerNumber),
  CONSTRAINT  orders_ibfk_1  FOREIGN KEY (customerNumber) REFERENCES  customers
     (customerNumber)
   )
```

3. Write SQL query to show all the columns data from the **Orders** Table.

Answer:-

```
cur = db.cursor()
data = cur.execute(SELECT * FROM orders)
data.fetchall()
```

4. Write SQL query to show all the comments from the **Orders** Table.

Answer:-

```
1. Single line comment
SELECT * FROM orders -- WHERE City='Berlin';

2. Multi line comment
/*Select all the columns
of all the records
in the orders table:*/
SELECT * FROM orders;
```

5. Write a SQL query to show orderDate and Total number of orders placed on that date, from **Orders** table.

Answer:-

SELECT date(`orderDate`), COUNT(`orderNumber`) AS num_orders, SUM(order_total) AS daily_total
FROM orders
WHERE `orderDate` >=date_sub(current_date, INTERVAL 31 DAY)
GROUP BY date(`orderDate`)

6. Write a SQL query to show employeNumber, lastName, firstName of all the employees from **employees** table.

Answer:-

Data=cur.execute(SELECT show employeNumber, lastName, firstName FROM employees
Data.fetchall()

7. Write a SQL query to show all orderNumber, customerName of the person who placed the respective order.

Answer:-

```
Data=cur.execute(SELECT a. customerName, b. orderNumber
FROM customers a
LEFT OUTER JOIN orders b
ON a. customerNumber =b. customerNumber
order by b.ord_date)

Data.fetchall()
```

8. Write a SQL query to show name of all the customers in one column and salerepemployee name inanother column.

Answer:-

```
Data=cur.execute(SELECT a. customerName, b. firstName, b.  lastName
FROM customers a
LEFT OUTER JOIN employees b
ON a. salesRepEmployeeNumber =b. employeeNumber
)

Data.fetchall()
```

9. Write a SQL query to show Date in one column and total payment amount of the payments made on that date from the **payments** table.

Answer:-

```
SELECT date(paymentDate), COUNT(amount) AS amount, SUM(amount) AS daily_total
FROM orders
WHERE paymentDate >=date_sub(current_date, INTERVAL 31 DAY)
GROUP BY date(paymentDate)
```

10. Write a SQL query to show all the products productName, MSRP, productDescription from the **products** table.

Answer:-

```
SELECT  productName, MSRP, productDescription
FROM products
```