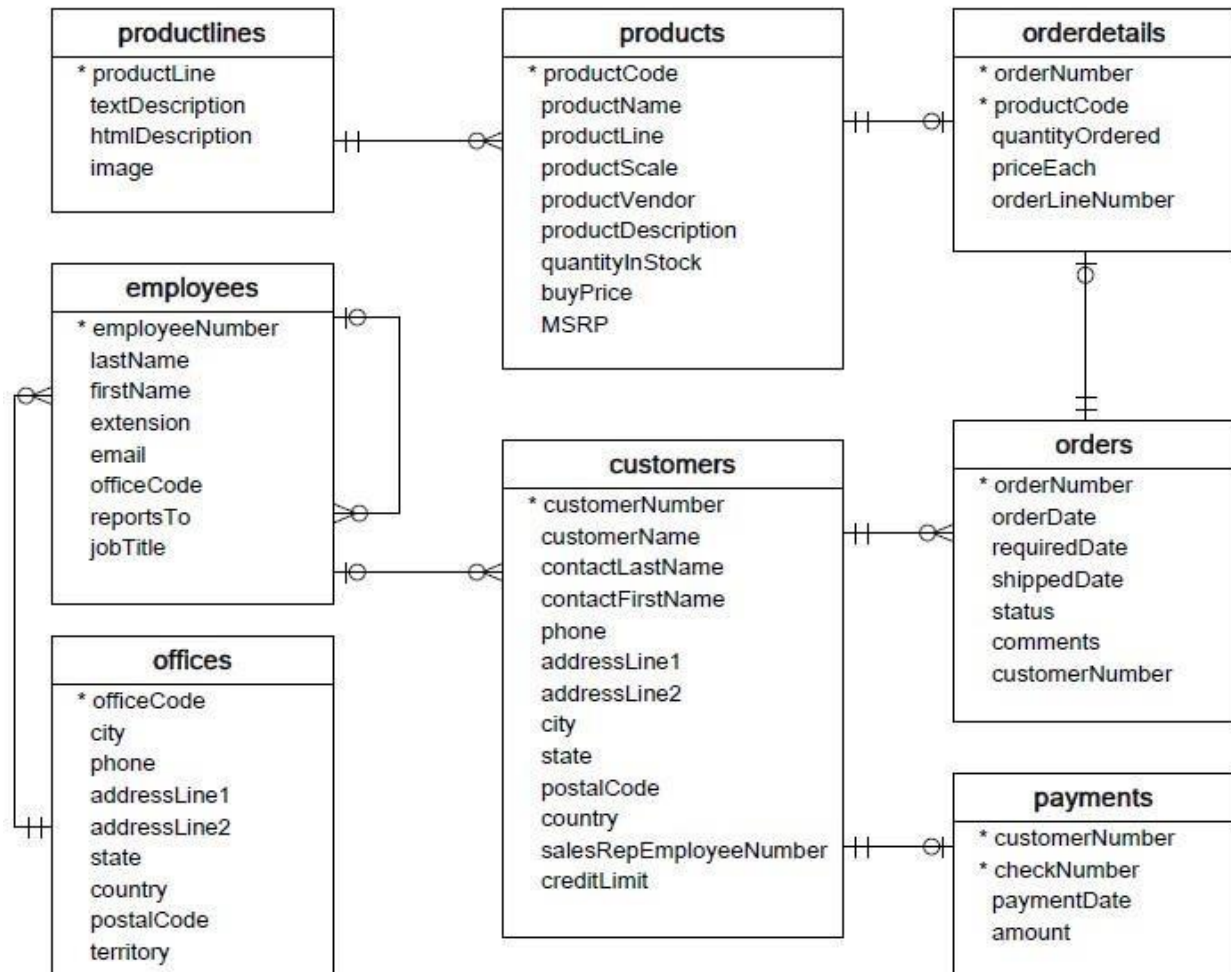


WORKSHEET 3 SQL

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**.

Ans : he first thing to do is you need to make a database new.

In order to display Vietnamese language, you can choose character_set is utf8, and the value of collate is utf8_general_ci. Normally, you can choose utf8 - utf8_general_ci.

```
create database if not exists db_test

character set utf8

collate utf8_unicode_ci;
```

To delete this database, you can type: **drop database if exists db_test.**

To appear all of the tables in your database, use **show database;**.

When you have so many database in your MySQL workbench, you want to utilize specific database - db_test in set of them. Use **use db_test;**.

Relationship between **productlines** table and **products** table.

To see the above image, you can be aware of their relation is one-to-many relationship. Each productlines has zero or more products and each products belongs to only one productlines. Of course, there is a foreign key - id of productlines in the products table.

The productlines table is called *parent table* or *referenced table*, and the products table is known as *child table* or *referencing table**.

A foreign key can be a column or a set of columns. The columns in the child table often refer to the primary key columns in the parent table.

A table may have more than one foreign key, and each foreign key in the child table may refer to a different parent table.

So, you can write sql statement to make these two tables:

```
create table if not exists customers(

    customerNumber int not null auto_increment primary key,

    customerName varchar(255),

    contactLastName varchar(255),

    contactFirstName varchar(255),

    phone varchar(15),

    addressLine1 varchar(255),
```

```
addressLine2 varchar(255),  
city varchar(255),  
state varchar(255),  
postalCode varchar(255),  
country varchar(100),  
employeeNumber int not null,  
creditLimit decimal(15, 2),  
foreign key fk_employees(employeeNumber)  
references employees(employeeNumber)  
on update cascade  
on delete restrict  
) engine = innodb;
```

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

ANS: create table if not exists orders(
orderNumber int auto_increment not null primary key,
orderDate date,
requiredDate date,
shippedDate date,

```
statuses text,  
comments text,  
customerNumber int not null,  
foreign key fk_customers(customerNumber)  
references customers(customerNumber)  
on update cascade  
on delete restrict  
)  
engine = innodb;
```

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

ANS: : create table if not exists orders(

orderNumber int auto_increment not null primary key,
orderDate date,
requiredDate date,
shippedDate date,
statuses text,
comments text,
customerNumber int not null,
foreign key fk_customers(customerNumber)
references customers(customerNumber)
on update cascade
on delete restrict
)
engine = innodb;

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

ANS:

```
Create table if not exists orders(  
  
comments text,
```

```
) engine = innodb;
```

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

ANS:

```
create table if not exists orders(  
    orderNumber int auto_increment not null primary key,  
    orderDate date,  
    requiredDate date,  
)  
engine = innodb;
```

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

ANS:

```
create table if not exists employees(  
    employeeNumber int auto_increment not null primary key,  
    lastName varchar(255),  
    firstName varchar(255),  
)  
engine = innodb;
```

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

ANS:

```
create table orders(  
    orderNumber int
```

```
customerName varchar(255),  
) engine = innodb;
```

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

Ans: **create table** (

```
customerName varchar(255),  
salerepemployee varchar(255)  
) engine = innodb;
```

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.

Ans :

```
create table (  
Date int,  
Total payment int,  
Payment Date int  
  
) engine = innodb;
```

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

Ans: **create table if not exists** products(

```
productName varchar(255) not null,  
productDescription text,  
MSRP varchar(255),  
) engine = innodb;
```

11. Write a SQL query to print the productName, productDescription of the most ordered product.

```
Ans:      create table if not exists products(

        productName varchar(255) not null,

        productDescription text,

        ) engine = innodb;
```

12. Write a SQL query to print the city name where maximum number of orders were placed

Ans :

```
select * (city name)
WHERE maximum quantity ordered
print("city_name")
```

13. Write a SQL query to get the name of the state having maximum number of customers.

```
ANS: Select *(state_name)
WHERE max_no_of_customers;
Print("state_name")
```

14 .Write a SQL query to print the employee number in one column and Full name of the employee in the second column for all the employees.

```
ANS: create table employee (employeeNumber int

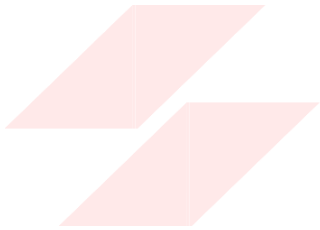
        lastName varchar(255),

        firstName varchar(255),

        ) engine = innodb;
```

15 .Write a SQL query to print the orderNumber, customer Name and total amount paid by the customer for that order (quantityOrdered × priceEach).

```
ANS : SELECT * FROM orders a WHERE purch_amt > (SELECT AVG(purch_amt) FROM orders b WHERE b.customer_id = a.customer_id);
```



FLIP ROBO