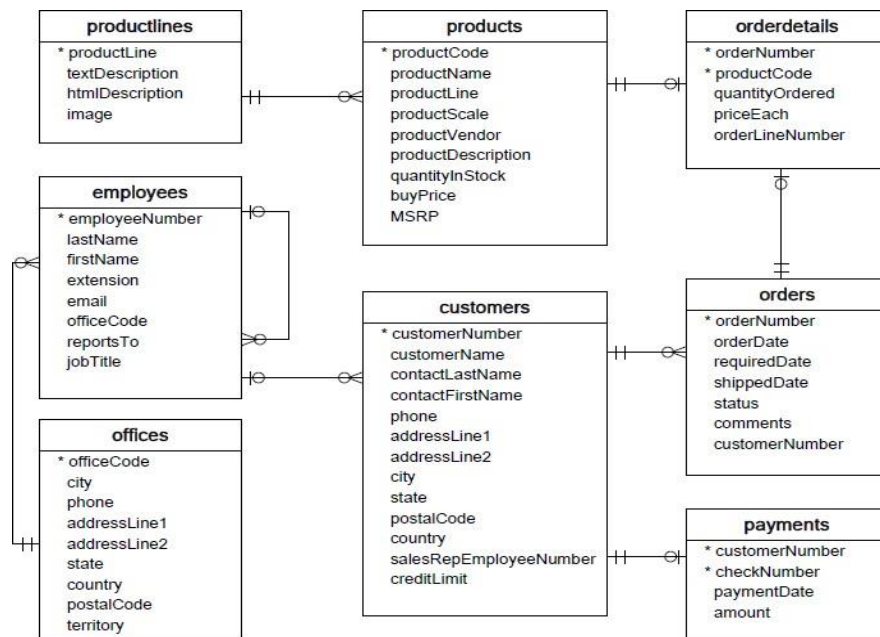


## WORKSHEET 3 ANSWERKEYS SQL

Please go through the below ERD before referring the answers.



- **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.

### Answers:

```

1. 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` ),
    FOREIGN KEY ( `salesRepEmployeeNumber` )
    REFERENCES `employees` ( `employeeNumber` )
);
    
```

2. 

```
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`),  
    FOREIGN KEY (`customerNumber`)  
    REFERENCES `customers` (`customerNumber`)  
    );
```
  3. 

```
SELECT * FROM Orders;
```
  4. 

```
SELECT `comments` FROM Orders;
```
  5. 

```
SELECT `orderDate`, COUNT(`orderNumber`) as `Total number of orders`  
FROM Orders  
GROUP BY `orderDate`;
```
  6. 

```
SELECT `employeeNumber`, `lastName`, `firstName` FROM Employees;
```
  7. 

```
SELECT `orderNumber`, `customerName`  
FROM Orders INNER JOIN customers  
ON orders.`customerNumber` = customers.`customerNumber`;
```
  8. 

```
SELECT `customerName`, CONCAT(`firstName`, `lastName`)  
FROM employees INNER JOIN customers  
ON Employees.`employeeNumber` = customers.`salesRepEmployeeNumber`;
```
  9. 

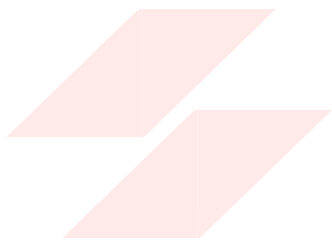
```
SELECT `paymentDate`, SUM(amount)  
FROM payments  
GROUP BY `paymentDate`;
```
  10. 

```
SELECT `productName`, `MSRP`, `productDescription` FROM products;
```
  11. 

```
SELECT `productName`, `productDescription` FROM Products  
INNER JOIN Orderdetails  
ON Products.`productCode` = Orderdetails.`productCode`  
GROUP BY Products.`productCode`  
ORDER BY SUM(`quantityOrdered`) DESC  
LIMIT 1;
```
  12. 

```
SELECT `city` FROM Orders as a  
INNER JOIN Customers as b  
ON a.`customerNumber` = b.`customerNumber`  
GROUP BY `city`  
ORDER BY COUNT(`orderNumber`) DESC  
LIMIT 1;
```
-

13. `SELECT `state` FROM Customers  
GROUP BY `state`  
ORDER BY COUNT(`customerNumber`) DESC  
LIMIT 1;`
14. `SELECT `employeeNumber`, concat(`firstName`, `lastName`) as `Full name` FROM Employees;`
15. `SELECT `orderNumber`, `customerName`, `quantityOrdered` * `priceEach` as `total amount paid`  
FROM OrderDetails as a INNER JOIN Orders as b  
ON a.`orderNumber` = b.`orderNumber`  
INNER JOIN Customers as c  
ON b.`customerNumber` = c.`customerNumber`;`



# FLIP ROBO