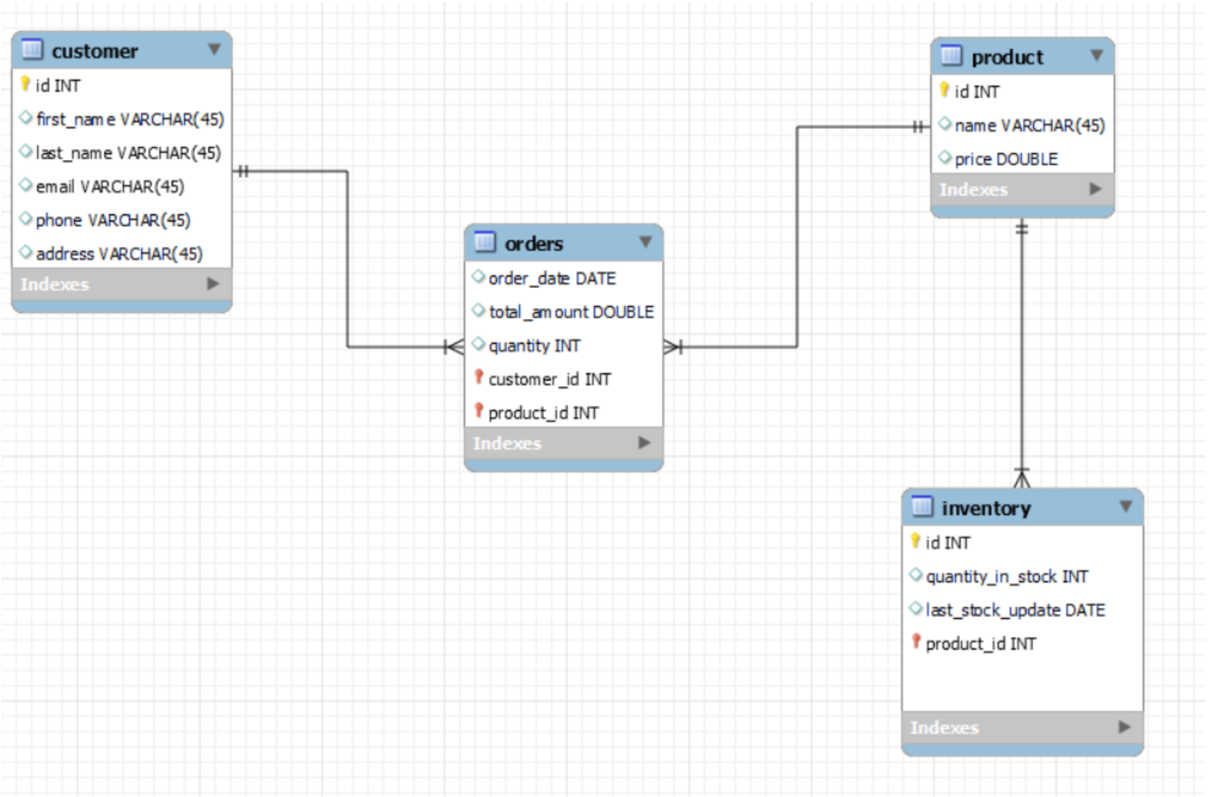
TECHSHOP



CODE:

```
use techshop_hex_feb_24;

insert into customer (first_name, last_name, email, phone, address) values
('diya','sara', 'diya@gmail.com', 4545545455, 'chennai'),
('dev', 'suriya','dev@gmail.com', 3434434344, 'mumbai'),
('tara','krish', 'tara@gmail.com', 2323323233, 'kolkata'),
('atul','bose' ,'atul@gmail.com', 1212212122,'pune'),
('ajay','josh', 'ajay@gmail.com', 4545455454, 'pondy');

insert into product (name, price) values
('laptop', 75000.0),
('iphone', 100000.0),
('macbook', 125000.0),
('earphone', 500.0),
```

```sql
('charge', 2000.0);

insert into orders (order_date, total_amount, quantity, customer_id, product_id) values
('2024-03-06', 810000.0, 3, 1,5),
('2024-02-21', 650000.0, 2, 2,4),
('2024-01-04', 230000.0, 4, 3,3),
('2024-02-17', 150000.0, 2, 4,2),
('2024-01-30', 50000.0, 1, 5,1);

insert into inventory (quantity_in_stock, last_stock_update, product_id) values
(23,'2024-02-27', 1),
(18,'2024-01-03', 2),
(6,'2024-03-07', 3),
(37,'2024-01-13', 4);
```

# Tasks 2

```sql
-- 1. Write an SQL query to retrieve the names and emails of all customers.
select first_name, last_name, email
from customer;

-- 2. Write an SQL query to list all orders with their order dates and corresponding customer names.
select o.order_date, c.first_name, c.last_name
FROM orders o, customer c
where o.customer_id = c.id;

-- 3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.
insert into customer (first_name, last_name, email, phone, address) values
('john', 'lee', 'john@gmail.com', '1231223233', 'pondy');
```

```sql
-- 4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by

update product

SET price = price * 1.10

where name IN ('laptop', 'iphone', 'macbook', 'earphone', 'charge');
```

```sql
-- 5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.

delete from orders

where order_id = 1;
```

```sql
-- 6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.

insert into orders (order_date, total_amount, quantity, customer_id, product_id)

values ('2024-03-08', 100000, 1, 1, 2);
```

```sql
-- 7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.

update customer

SET address = 'bangalore'

where customer_id = 1;
```

```sql
-- 9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.

delete from orders

where customer_id = 1;
```

```sql
-- 10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.

insert into product (name, price)

values ('mouse', 1200);
```

-- 11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.

```
update orders
SET status = 'Shipped'
WHERE order_id = 2;
```

# Task 3

-- 1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.

```
select o.order_date, c.first_name, c.last_name, c.email
from orders o
JOIN customer c ON o.customer_id = c.id;
```

-- 2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.

```
select p.name, sum(o.total_amount)
from orders o
JOIN product p ON o.product_id = p.id
group by p.name;
```

-- 3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.

```
select c.first_name, c.last_name, c.email, c.phone
from customer c
JOIN orders o ON c.id = o.customer_id
group by c.id;
```

-- 4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest

```
select p.name, sum(o.quantity)
from orders o
JOIN product p ON o.product_id = p.id
```

```sql
group by p.name
order by total_quantity_ordered desc
limit 0,1;

-- 5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
select p.name , p.category
from product ;

-- 6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.
select c.first_name, c.last_name, avg(o.total_amount)
from orders o
JOIN customer c ON o.customer_id = c.id
group by c.id;

-- 7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.
select o.id, o.order_date, c.first_name, c.last_name, c.email, sum(o.total_amount) as revenue
from orders o
JOIN customer c ON o.customer_id = c.id
group by o.id
order by revenue desc
limit 0,1;   -- order_id is not present in the database

-- 8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.
select p.name, count(o.order_id)
from orders o
JOIN product p ON o.product_id = p.id
group by p.name;
```

-- 9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.

```
select c.first_name

from customer c

JOIN orders o ON c.id = o.customer_id

JOIN product p ON o.product_id = p.id

where p.name = 'macbook';
```

-- 10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.

```
select sum(total_amount)

from orders

where order_date between '2024-03-01' and '2024-01-01';
```

# Task 4

-- 1. Write an SQL query to find out which customers have not placed any orders.

```
select id, first_name

from customer

where id NOT IN (select distinct customer_id

                                        from orders);
```

-- 2. Write an SQL query to find the total number of products available for sale.

```
select count(*) as total_products

from product;
```

-- 3. Write an SQL query to calculate the total revenue generated by TechShop.

```
select sum(total_amount) as total_revenue

from orders;
```

-- 4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.

```
select avg(quantity)

from orders

where product_id in (select id

                                         from product

            where category = 'iphone');
```

-- 5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

```
select sum(total_amount)

from orders

where customer_id = 1;
```