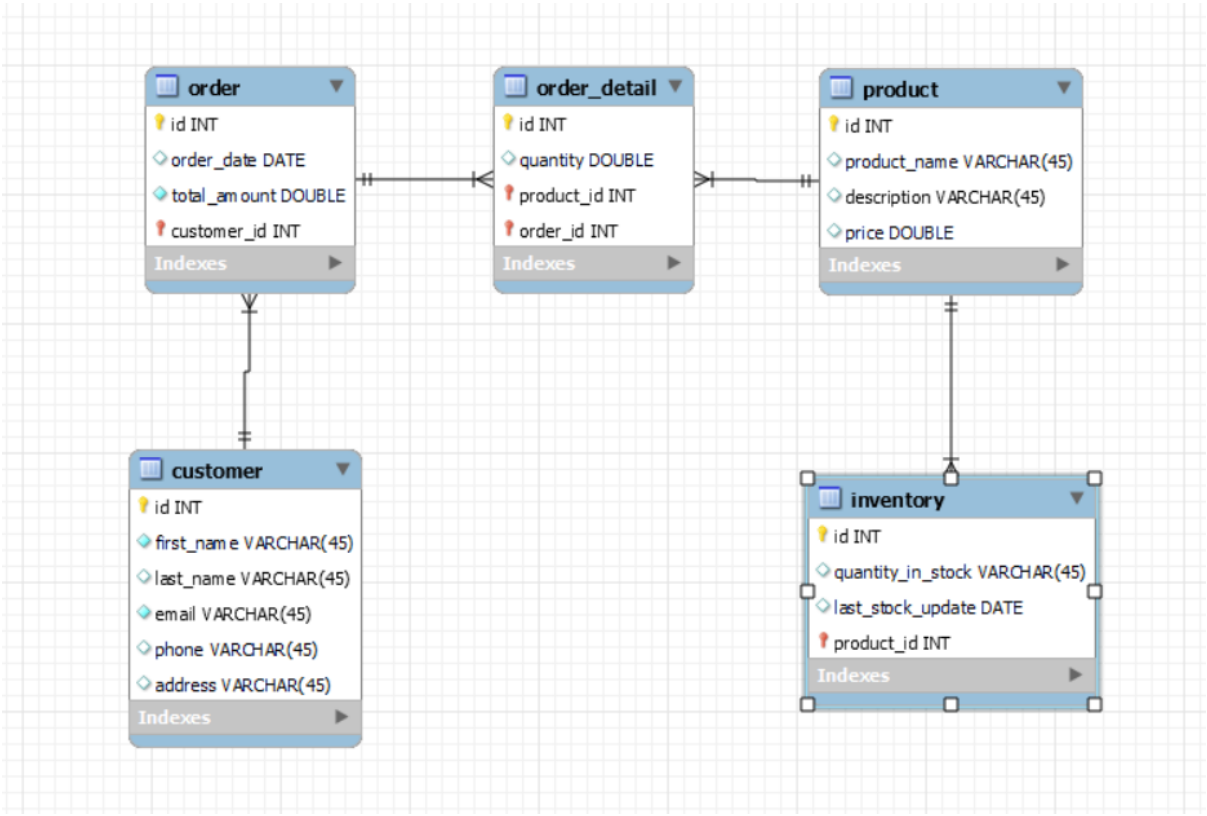
Electronic Gadget Assignment

Task-1:



Task-2:

-- 1. Write an SQL query to retrieve the names and emails of all customers.

```
select first_name,email from customer;
```

-- 2. Write an SQL query to list all orders with their order dates and corresponding customer names.

```
select c.first_name,o.order_date
from customer c,orders o
where c.id=o.customer_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
('usopp','M','usopp@gmail.com','1234','gujarat');
```

-- 4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.

```
update product
set price=price+(price*0.1)
where id=1;
```

-- 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 order_detail
where order_id=1;

delete from orders
where id=1;
```

-- 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,customer_id)
values
('2024.2.20',2500,4);
```

-- 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 email='lufffy@gmail.com' , address='pune'
where id=1;
```

-- 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 order_details
where order_id in (select id from orders where customer_id =3);

delete from orders
where customer_id=3;
```

-- 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(product_name,description,price)
values
('Acer','desktop',50000);
```

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 c.first_name,o.order_date,o.total_amount
from customer c join orders o on c.id=o.customer_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.product_name,sum(o.total_amount) as Total_revenue
from orders o join order_detail od on o.id=od.order_id join product p on p.id=od.product_id
group by p.id;
```

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

```
select distinct c.first_name,c.email
from customer c join orders o on c.id=o.customer_id ;
```

-- 4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.

```
select p.product_name,od.quantity
from product p join order_detail od on p.id=od.product_id
order by quantity desc
limit 1;
```

-- 5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.

```
select product_name,description
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,avg(total_amount)
from customer c join orders o on c.id=o.customer_id
group by o.customer_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 od.order_id,sum(total_amount) as total_revenue
from customer c join orders o on c.id=o.customer_id join order_detail od on od.order_id=o.id
group by od.order_id
order by total_revenue desc
limit 1
;
```

-- 8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.

```
select p.product_name,count(od.id) as Order_count
from order_detail od join product p on od.product_id=p.id
group by od.product_id;
```

-- 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 ,c.email

from customer c join orders o on c.id=o.customer_id join order_detail od on od.order_id=o.id join product p on od.product_id=p.id

where p.id=2;
```

-- 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 od.order_id,sum(total_amount) as total_revenue

from customer c join orders o on c.id=o.customer_id join order_detail od on od.order_id=o.id

where o.order_date between '2023-12-31' and '2024-1-31'

group by od.order_id;
```

Task-4:

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

```
select * from customer

where id not in (select customer_id from orders);
```

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

```
select * from product

where id in (select product_id from inventory where quantity_in_stock is not null);
```

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

```
select sum(total_amount) as 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 (sum(quantity)/(select count(id) from product where
description="mobile")) as Average_quantity_mobile from order_detail

where product_id in (select id from product p where description="mobile");

select * from product;

select * from order_detail;
```

-- 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) as Revenue_From_1 from orders

where customer_id in (select id from customer where id=1);
```

-- 7. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.

```
select c.first_name from customer c

where id in

(select customer_id from orders

 where total_amount=(select max(total_amount) from orders));
```

-- 8. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.

```
select first_name,t.* from customer c

join(select o.customer_id ,count(o.id) from orders o group by o.customer_id)
as t on t.customer_id=c.id;
```