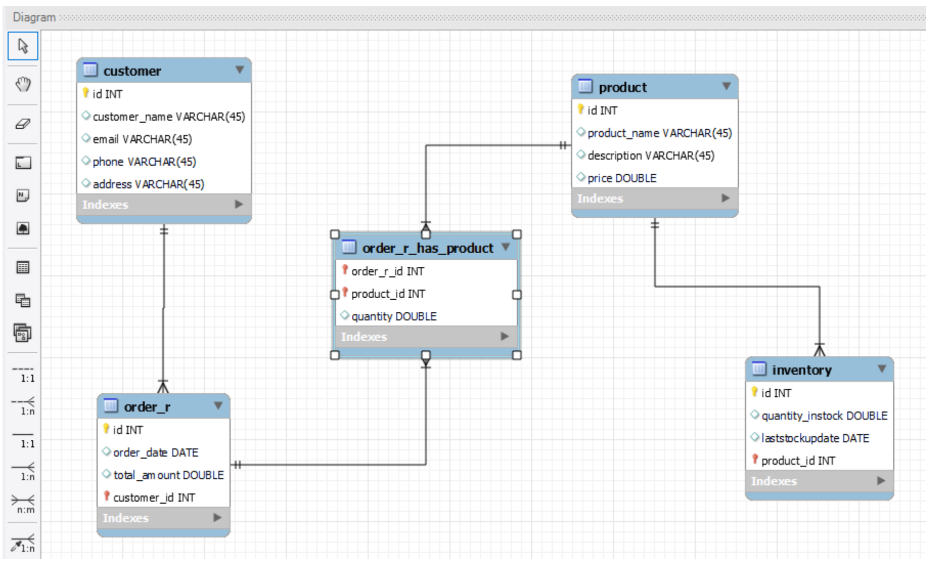
ELECTRONIC GADGET ASSIGNMENT

ER DIAGRAM:



INSERTIONS:

insert into customer (customer_name,email,phone,address) values

('Harsha','harshha76@gmail.com','9876761243','Chennai'),

('Ishaan','ishann98@gmail.com','9234156781','Rajasthan'),

('Anika','anika982@gmail.com','91276486321','Rajasthan'),

('Arohi','arohi23h@gmail.com','9876761243','Chennai'),

('Rudhran','rudhran673@gmail.com','9876761243','Bangalore'),

('Annaya','annanyar45@gmail.com','8976543210','Bangalore'),

('Vishwa','vishwag76@gmail.com','8723410762','Hyderabad'),

('Vinothini','vinovisha876@gmail.com','7654321890','Hyderabad');

----

insert into product (product_name,description,price) values

('realme c53','Mobile','10000'),

('vivo i8','Mobile','21000'),

('Intel i4','Laptop','50000'),

('Dell D4','Laptop','65000'),

```
('Sony','TV','28000'),

('LG','TV','25000'),

('samsung s20','Mobile','21000'),

('Bajaj','Fans','1949'),

('Orient','Fans','2599');

---------------------------------------------------------------------------------------------------------------------------

insert into order_r(order_date,total_amount,customer_id) values

('2022-03-31','21000',8),

('2021-07-05','2599',1),

('2020-09-23','28000',7),

('2024-04-23','25000',5),

('2023-10-23','50000',4),

('2024-01-23','1949',5),

('2023-07-09','10000',7),

('2022-04-23','65000',1);

---------------------------------------------------------------------------------------------------------------------------

insert into inventory (quantity_instock,laststockupdate,product_id) values

('20','2023-07-30',1),

('9','2022-11-28',2),

('30','2021-09-02',3),

('6','2023-10-03',4),

('10','2024-01-21',5),

('11','2023-10-30',6),

('0','2020-07-22',7),

('44','2023-06-03',8);

---------------------------------------------------------------------------------------------------------------------------

insert into order_r_has_product (order_r_id,product_id,quantity) values

(33,6,'1'),

(27,5,'4'),

(29,3,'2'),
```

(30,1,'3');

Tasks 2: Select, Where, Between, AND, LIKE:

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

select customer_name,email

from customer;

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

select o.id,o.order_date,c.customer_name

from order_r o

join customer c on 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 (customer_name,email,phone,address) values

('Aryan','aryanp45@gmail.com','8976543210','Bangalore');

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*(10/100));

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_r

where id= '&userid';

delete from order_r_has_product

where order_r_id = '&userorder_r_id';

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 order_r(order_date,total_amount,customer_id) values

('2024-02-18','10000',5);

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='harshakrishna@gmail.com',address='Hyderabad'

where id= '&userid';
```

8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.

```
update order_r

set total_amount = (

   select sum(price * quantity) as ordertotal

   from order_r_has_product

   where order_r_has_product.order_r_id = order_r.id

   group by order_r_has_product.order_r_id  );
```

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_r_has_product

where order_r_id in (select id from order_r where customer_id = '&customerid');
```

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

('Sony','AC','30000');
```

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 = '&newstatus'

where orderid = '&orderid';
```

12. Write an SQL query to calculate and update the number of orders placed by each customer

in the "Customers" table based on the data in the "Orders" table.

```
update customer

set no_of_orders = (

   select count(*)

   from order_r
```

where order_r.customer_id = customer.id);

NOTE: As for the few questions in the above tasks they have asked for user input. I have no idea about to solve that. So I have searched through internet and have written if something over that is right or wrong please correct me.

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

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.id,o.order_date,c.id,c.customer_name,c.email

from order_r 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.product_name,SUM(p.price*o.quantity) as total_revenue

from product p

JOIN order_r_has_product o ON p.id=o.product_id

group by product_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.customer_name,c.email,c.phone,c.address

from customer c

JOIN order_r r ON c.id=r.customer_id

group by c.customer_name;

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 product_name

from product

where id IN (select max(quantity) from order_r_has_product);

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

select description , product_name

from product

group by description;

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.customer_name, avg(o.total_amount)

from customer c

JOIN order_r o ON c.id=o.customer_id

group by c.customer_name;
```

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,c.id,c.customer_name,c.customer_email,sum(r.price * r.quantity) as totalrevenue

from order_r o

join customer c on o.customer_id = c.id

join order_r_has_product r on o.id = r.order_r_id

group by o.id,c.id,c.customer_name,c.customer_email

order by totalrevenue desc;
```

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

```
select p.product_name,count(p.id)

from product p

JOIN order_r_has_product o ON p.id=o.product_id

group by p.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 distinct c.id,c.customer_name,c.email

from customer c

join order_r o on c.id= o.customer_id

join order_r_has_product od on o.id = od.order_r_id

join product p on od.product_id = p.id

where p.description = 'mobile';
```

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 o.total_amount as total_revenue

from order_r o
```

```
join order_r_has_product od on o.id = od.order_r_id

where o.order_date >= '2023-01-01' and o.order_date <= '2023-12-31';
```

Task 4. Subquery and its type:

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

```
select c.id, c.customer_name

from customer c

join order_r r on c.id = r.customer_id

where r.customer_id is null;
```

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

```
select id,product_name

from product

where id IN (select product_id from inventory

             where quantity_instock = '0');
```

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

```
select id,sum(total_amount) as total_revenue

from order_r

where id in (select order_r_id from order_r_has_product

             where shop_name='techshop');
```

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 od.product_id,AVG(od.quantity)

from order_r_has_product od

JOIN product p ON od.product_id=p.id

where p.id IN (select quantity_instock from inventory )

group by od.product_id;
```

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 total_revenue

from order_r

where customer_id = '3';
```

6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.

```
select c.id, c.customer_name, count(o.id) as num_orders
from customer c
join order_r o on c.id = o.customer_id
group by c.id, c.customer_name
order by num_orders desc;
```

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.

```
select p.description, sum(od.quantity) as total_quantity_ordered
from order_r_has_product od
join product p on od.product_id = p.id
group by p.description
order by total_quantity_ordered desc;
```

8. 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.customer_name, sum(o.total_amount) as total_revenue
from customer c
join order_r o on c.id = o.customer_id
join product p on o.product_id = p.id
WHERE p.description = 'Mobile'
GROUP BY c.id, c.customer_name
ORDER BY total_revenue DESC;
```

9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.

10. 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 c.customer_name, count(o.id) as order_count
from customer c
join order_r o on c.id = o.customer_id
group by c.id, c.customer_name;
```