

Лабораторна робота №4

Join запити:

```
-- INNER JOIN  
select * from  
customers c inner join orders o using (customerid);
```

```
-- LEFT JOIN  
select * from  
orders o left join order_items oi using (orderid);
```

```
-- LEFT JOIN 2  
select * from  
orders o left join payments p using (orderid);
```

Aggregation запити:

```
-- COUNT()  
select orderid, count(*)  
from order_items  
group by orderid;
```

```
-- AVG()  
select avg(sum)  
from payments;
```

```
-- MIN()  
select min(cost_for_one)  
from products;
```

```
-- MAX()  
select max(cost_for_one)  
from products;
```

```
-- SUM()  
select sum(cost)  
from orders;
```

Group запити:

```
-- GROUP  
select categoryid, count(*)  
from product_categories  
group by categoryid;
```

```
-- GROUP HAVING  
select orderid, count(*)  
from order_items  
group by orderid  
having count(*) > 1;
```

-- MULTIPLE TABLES AGGREGATION

```
select o.orderid, sum(oi.cost)  
from orders o left join order_items oi using(orderid)  
group by orderid;
```

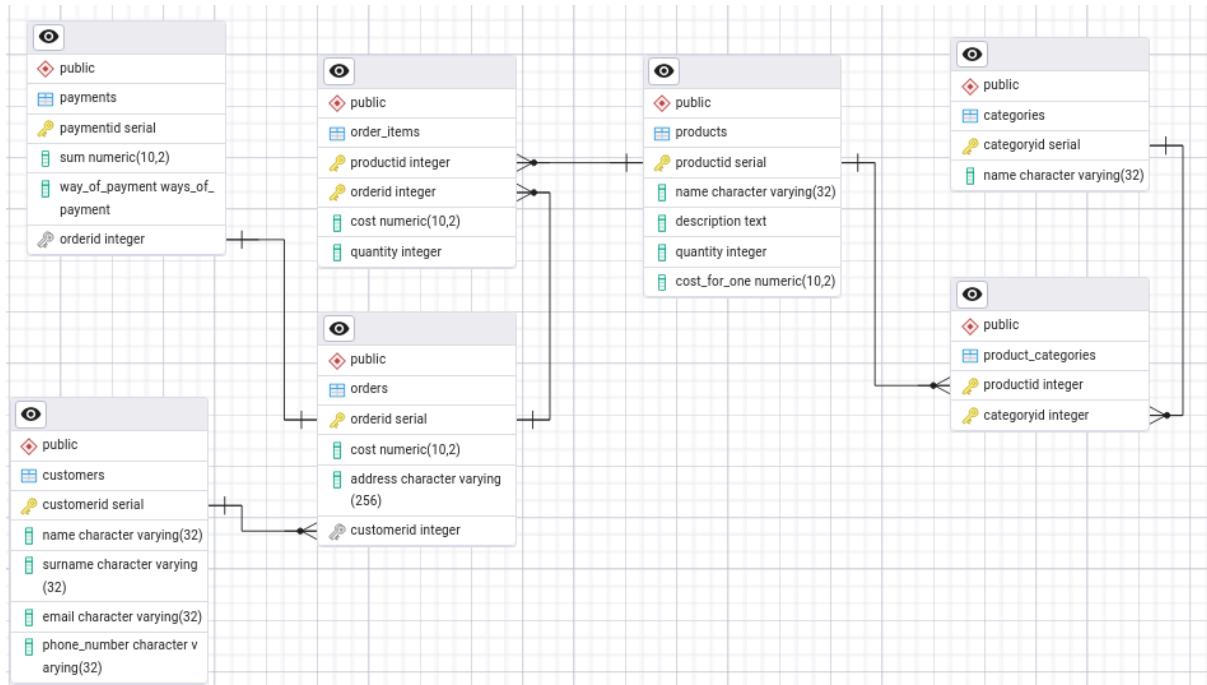
Запити, які використовують підзапити:

```
select *  
from orders  
where customerid = (  
    select customerid  
    from customers  
    where name = 'Ivan' and surname = 'Petrenko');
```

```
select *  
from product_categories  
where categoryid = (  
    select categoryid  
    from categories  
    where name = 'Audio');
```

```
select *  
from order_items  
where orderid = (  
    select orderid  
    from orders  
    where orderid = 1);
```

ER діаграма бази даних:



Письмовий звіт:

На схемі реалізовані наступні таблиці:

- **customers** - таблиця для покупців.
 - customerID - primary key.
 - name, surname, address, phone_number - дані користувача, текстові дані, тому varchar(32).
- **orders** - таблиця для замовлень.
 - orderID - primary key.
 - cost - ціна, decimal(10, 2) - для точності. Обмеження: ціна не може бути від'ємною, тому робиться перевірка check (cost >= 0)
 - address - адреса замовлення - текст, тому varchar(256).
 - customerID - реалізує зв'язок між customer і orders, посилаючись на customers(customerID).
- **payments** - таблиця для платіжок.
 - paymentID - primary key.
 - orderID - реалізує зв'язок між payments і orders, посилаючись на orders(orderID).
 - sum - сума платежу, для більшої точності використовується decimal(10, 2), також обмежена як і cost в orders.
 - way_of_payment є типом ways_of_payment, який є в свою чергу enum, з переліком способів оплати.
- **order_items** - таблиця для елементів замовлення.
 - productID, orderID - foreign keys, які формують primary key. Це реалізує зв'язок між orders - order_items - product.

- cost - ціна, decimal(10, 2) - для точності, також обмежена як і cost в orders.
 - quantity - кількість продукту, тому int. Обмеження: кількість товарів в замовленні не може бути від'ємною або нульовою, тому ($quantity > 0$).
- products - таблиця для продуктів.
 - productID - primary key.
 - name - назва продукту, тому varchar(32).
 - cost_for_one - ціна одиниці, decimal(10, 2) - для точності, також обмежена як і cost в orders.
 - quantity - кількість продукту, тому int. Кількість товарів на складі не може бути від'ємною, тому ($quantity \geq 0$).
 - description - опис продукту, може бути досить великим, тому тип text. Опис не обов'язковий, тому не not null.
- categories - таблиця для категорій.
 - categoryID - primary key.
 - name - назва категорії, тому varchar(32).
- category_product - таблиця для зв'язку “багато до багатьох” категорій та продуктів.
 - categoryID та productID - foreign keys, які формують primary key, для реалізації зв'язку “багато до багатьох” між таблицями categories та products.