

## Лабораторна робота №3

### Create tables:

```
create table if not exists customers
```

```
(  
    customerID      serial primary key,  
    name            varchar(32) not null,  
    surname         varchar(32) not null,  
    email           varchar(32) not null,  
    phone_number    varchar(32)  
)
```

```
create table if not exists orders
```

```
(  
    orderID        serial primary key,  
    cost            decimal(10,2) not null      check (cost >= 0),  
    address         varchar(256) not null,  
    customerID     int not null references customers(customerID)  
)
```

```
create table if not exists products
```

```
(  
    productID      serial primary key,  
    name            varchar(32) not null,  
    description     text,  
    quantity        int not null              check (quantity >= 0),  
    cost_for_one    decimal(10,2) not null      check (cost_for_one >= 0)  
)
```

```
create table if not exists order_items
```

```
(  
    productID      int not null references products(productID),  
    orderID        int not null references orders(orderID),  
    cost            decimal(10,2) not null      check (cost >= 0),  
    quantity        int not null              check (quantity > 0),  
  
    primary key (productID, orderID)  
)
```

```
create table if not exists categories
```

```
(  
    categoryID     serial primary key,  
    name            varchar(32) not null  
)
```

```

create table if not exists product_categories
(
    productID      int not null references products(productID),
    categoryID     int not null references categories(categoryID),
    primary key (productID, categoryID)
);

create type ways_of_payment as enum('cash', 'card');

create table if not exists payments
(
    paymentID        serial primary key,
    sum              decimal(10,2) not null          check (sum >= 0),
    way_of_payment   ways_of_payment,
    orderID          int unique not null references orders(orderID)
);

```

## Insert some data:

```

insert into customers (name, surname, email, phone_number) values
('Ivan', 'Petrenko', 'ivan.p@gmail.com', '380991112233'),
('Oksana', 'Shevchenko', 'oks.shev@gmail.com', '380932224455'),
('Dmytro', 'Bondar', 'd.bondar@ukr.net', '380673335577'),
('Anna', 'Melnyk', 'anna.melnyk@gmail.com', '380503331122');

```

```

insert into products (name, description, quantity, cost_for_one) values
('Laptop', '15-inch lightweight laptop', 20, 29999.00),
('Headphones', 'Wireless noise-cancelling', 50, 2999.00),
('Keyboard', 'Mechanical RGB keyboard', 35, 1999.00),
('Smartphone', '6.5-inch OLED, 128GB', 15, 19999.00);

```

```

insert into categories (name) VALUES
('Electronics'),
('Computers'),
('Accessories'),
('Audio');

```

```

insert into product_categories (productID, categoryID) values
(1, 2),
(2, 4),
(3, 3),
(4, 1);

```

```

insert into orders (cost, address, customerID) values
(32998.00, 'Kyiv, Khreshchatyk 10', 1),
(22998.00, 'Lviv, Svobody Ave 5', 2),

```

(4998.00, 'Odesa, Deribasivska 7', 3),  
(19999.00, 'Kharkiv, Sumska 15', 4);

```
insert into order_items (productID, orderID, cost, quantity) values  
(1, 1, 29999.00, 1),  
(2, 1, 2999.00, 1),  
(4, 2, 19999.00, 1),  
(3, 3, 1999.00, 1),  
(2, 4, 2999.00, 1);
```

```
insert into payments (sum, way_of_payment, orderID) values  
(32998.00, 'card', 1),  
(22998.00, 'cash', 2),  
(4998.00, 'card', 3),  
(19999.00, 'card', 4);
```

## Create a new order. (example of using INSERT, UPDATE, SELECT, DELETE)

-- process of creating a new order, add some products, then create a payment and delete the order.

```
-- create a new order  
insert into orders(cost, address, customerID) values (0.00, 'Kyiv, Pivnichna 74', 0);  
-- create an order_item  
insert into order_items(productID, orderID, cost, quantity) values (1, 5, 29999.00, 1);
```

```
-- check before update  
select * from orders  
where orderID = 5;
```

```
-- update the order's cost with a new order_item  
update orders  
set cost = (select sum(cost) from order_items where orderID = 5)  
where orderID = 5;
```

```
-- create a new order_item  
insert into order_items(productID, orderID, cost, quantity) values (2, 5, 2999.00, 1);
```

```
-- update the order's cost again  
update orders  
set cost = (select sum(cost) from order_items where orderID = 5)
```

```
where orderID = 5;

-- check before delete
select * from order_items
where orderID = 5 and productID = 1;

-- delete an order_item that we don't need anymore
delete from order_items
where orderID = 5 and productID = 1;

-- update the order's cost
update orders
set cost = (select sum(cost) from order_items where orderID = 5)
where orderID = 5;

-- create a new payment
insert into payments(sum, way_of_payment, orderID)
values ((select cost from orders where orderID = 5), 'card', 5);

-- check before delete
select * from order_items
where orderID = 5;

-- delete an order_item
delete from order_items
where orderID = 5;

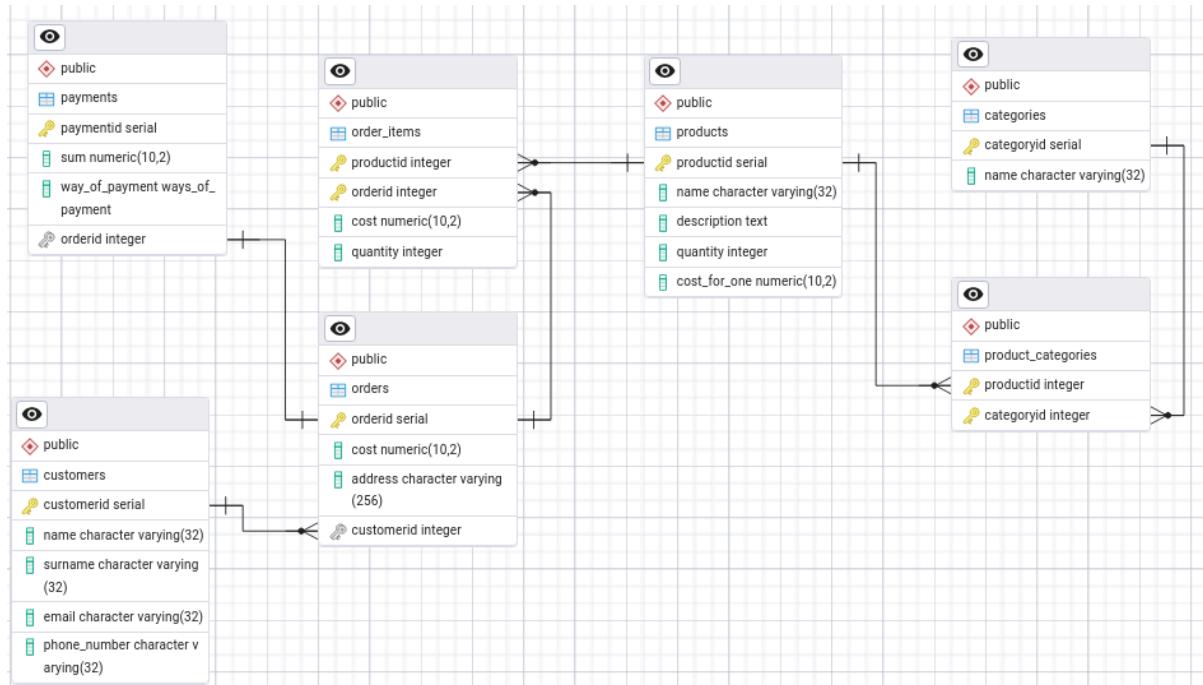
-- check before delete
select * from payments
where orderID = 5;

-- delete payment (if the order is paid)
delete from payments
where orderID = 5;

-- check before delete
select * from orders
where orderID = 5;

-- delete the order
delete from orders
where orderID = 5;
```

## ER діаграма:



## 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.

## Скріншоти результатів початкового заповнення таблиць (не зважайте на індекси):

### Customers:

	customerid [PK] integer	name character varying (32)	surname character varying (32)	email character varying (32)	phone_number character varying (32)
1	9	Ivan	Petrenko	ivan.p@gmail.com	380991112233
2	10	Oksana	Shevchenko	oks.shev@gmail.com	380932224455
3	11	Dmytro	Bondar	d.bondar@ukr.net	380673335577
4	12	Anna	Melnyk	anna.melnyk@gmail.c...	380503331122

### Products

	productid [PK] integer	name character varying (32)	description text	quantity integer	cost_for_one numeric (10,2)
1	9	Laptop	15-inch lightweight lapt...	20	29999.00
2	10	Headphones	Wireless noise-cancelli...	50	2999.00
3	11	Keyboard	Mechanical RGB keybo...	35	1999.00
4	12	Smartphone	6.5-inch OLED, 128GB	15	19999.00

### Categories:

	categoryid [PK] integer	name character varying (32)
1	9	Electronics
2	10	Computers
3	11	Accessories
4	12	Audio

### product\_categories:

	productid [PK] integer	categoryid [PK] integer
1	9	10
2	10	12
3	11	11
4	12	9

### Orders:

	orderid [PK] integer	cost numeric (10,2)	address character varying (256)	customerid integer
1	1	32998.00	Kyiv, Khreshchatyk 10	9
2	2	22998.00	Lviv, Svobody Ave 5	10
3	3	4998.00	Odesa, Deribasivska 7	11
4	4	19999.00	Kharkiv, Sumska 15	12

**order\_items:**

	productid [PK] integer	orderid [PK] integer	cost numeric (10,2)	quantity integer
1	9	1	29999.00	1
2	10	1	2999.00	1
3	12	2	19999.00	2
4	11	3	1999.00	1
5	10	4	2999.00	3

**Payments:**

	paymentid [PK] integer	sum numeric (10,2)	way_of_payment ways_of_payment	orderid integer
1	1	32998.00	card	1
2	2	22998.00	cash	2
3	3	4998.00	card	3
4	4	19999.00	card	4

## Скріншоти протягом створення замовлення (не зважайте на індекси):

### додавання замовлення (id = 7):

	orderid [PK] integer	cost numeric (10,2)	address character varying (256)	customerid integer
1	1	32998.00	Kyiv, Khreshchatyk 10	1
2	2	22998.00	Lviv, Svobody Ave 5	2
3	3	4998.00	Odesa, Deribasivska 7	3
4	4	19999.00	Kharkiv, Sumska 15	4
5	7	0.00	Kyiv, Pivnichna 74	1

### додавання нового елементу замовлення (orderid = 7):

	productid [PK] integer	orderid [PK] integer	cost numeric (10,2)	quantity integer
1	1	1	29999.00	1
2	2	1	2999.00	1
3	4	2	19999.00	1
4	3	3	1999.00	1
5	2	4	2999.00	1
6	1	7	29999.00	1

### замовлення після оновлення суми:

	orderid [PK] integer	cost numeric (10,2)	address character varying (256)	customerid integer
1	7	29999.00	Kyiv, Pivnichna 74	1

### елементи замовлення після додавання другого елементу:

	orderid integer	cost numeric (10,2)	address character varying (256)	customerid integer	productid integer	cost numeric (10,2)	quantity integer
1	7	29999.00	Kyiv, Pivnichna 74	1	1	29999.00	1
2	7	29999.00	Kyiv, Pivnichna 74	1	2	2999.00	1

### оновлення суми замовлення:

	orderid [PK] integer	cost numeric (10,2)	address character varying (256)	customerid integer
1	7	32998.00	Kyiv, Pivnichna 74	1

### елементи замовлення після видалення першого елементу та оновлення суми замовлення:

	orderid integer	cost numeric (10,2)	address character varying (256)	customerid integer	productid integer	cost numeric (10,2)	quantity integer
1	7	2999.00	Kyiv, Pivnichna 74	1	2	2999.00	1

**створення нового платежу (payment id = 6):**

	paymentid [PK] integer	sum numeric (10,2)	way_of_payment ways_of_payment	orderid integer
1	1	32998.00	card	1
2	2	22998.00	cash	2
3	3	4998.00	card	3
4	4	19999.00	card	4
5	6	2999.00	card	7

**елементи замовлення після видалення всіх елементів даного замовлення:**

	orderid integer	cost numeric (10,2)	address character varying (256)	customerid integer	productid integer	cost numeric (10,2)	quantity integer

**таблиця платіжок після видалення платіжки, пов'язаної з даним замовленням:**

	paymentid [PK] integer	sum numeric (10,2)	way_of_payment ways_of_payment	orderid integer
1	1	32998.00	card	1
2	2	22998.00	cash	2
3	3	4998.00	card	3
4	4	19999.00	card	4

**таблиця замовлень після видалення замовлення:**

	orderid [PK] integer	cost numeric (10,2)	address character varying (256)	customerid integer
1	1	32998.00	Kyiv, Khreshchatyk 10	1
2	2	22998.00	Lviv, Svobody Ave 5	2
3	3	4998.00	Odesa, Deribasivska 7	3
4	4	19999.00	Kharkiv, Sumska 15	4