

School of Digital Technologies

**Documentation of the project**

Discipline**:** Database management system. SUBD 2216-27--Ch

Teacher: Batyrkhan Omarovа

Student: Miramuly Zharaskhan

Specialize: Digital Engineering 2nd Course

Project tittle : Online internet market

**Introduction about project**

While studying at university, I worked as an operator specialist in the field of online stores. Inspired by this experience, I created a database to deepen my understanding of the subject. This discipline has taught me how to design and structure databases effectively.

The purpose of this project is to enhance my understanding and skills in database design and management by creating a comprehensive database schema for an online market.

**Including tables in database with descriptions:**

**CATEGORIES**: Stores information about product categories (categoryid: serial, categoryname: varchar(100), description: text).

**USERS**: Contains user details such as name and email (userid: serial, firstname: varchar(50), lastname: varchar(50), email: varchar(100), passwordhash: varchar(255), createdat: timestamp).

**ADDRESSES**: Records addresses linked to users (addressid: serial, userid: integer, street: varchar(100), city: varchar(50), state: varchar(50), postalcode: varchar(10), country: varchar(50)).

**PRODUCTS**: Includes product details such as name, price, and stock (productid: serial, productname: varchar(100), description: text, price: numeric(10,2), stock: integer, categoryid: integer).

**DISCOUNTS**: Contains discount details (discountid: serial, discountcode: varchar(20), discountamount: numeric(10,2), expirydate: date).

**ORDERS**: Stores information about user orders (orderid: serial, userid: integer, orderdate: timestamp, status: varchar(50), totalamount: numeric(10,2)).

**ORDERDETAILS**: Records products included in orders (orderdetailid: serial, orderid: integer, productid: integer, quantity: integer, price: numeric(10,2)).

**ORDERDISCOUNTS**: Links discounts to specific orders (orderdiscountid: serial, orderid: integer, discountid: integer).

**REVIEWS**: Stores user reviews for products (reviewid: serial, userid: integer, productid: integer, rating: integer, reviewtext: text, createdat: timestamp).

**PAYMENTS**: Tracks payments for orders (paymentid: serial, orderid: integer, paymentdate: timestamp, amount: numeric(10,2), paymentmethod: varchar(50)).

**Изображение выглядит как текст, снимок экрана, Шрифт, Параллельный

Автоматически созданное описание**

* **Primary Keys**: Each table has a unique identifier such as userid, productid, or orderid.
* **Foreign Keys**: Relationships between tables are established using foreign keys (e.g., userid in addresses links to users).
* **Unique Constraints**: Enforce uniqueness on fields like email in users and discountcode in discounts.
* **NOT NULL Constraints**: Ensure critical fields like productname, price, and orderid cannot be null.
* **CHECK Constraints**: Validate data (e.g., rating in reviews must be between 1 and 5).

**Table Relationships with description:**

**One-to-One**: Each orderdiscountid in orderdiscounts links to a unique orderid and discountid.

**One-to-Many**: A single userid in users can have multiple entries in addresses and orders.

**Many-to-Many**: Products and orders are linked via orderdetails, and orders and discounts are linked via orderdiscounts.

**SQL Queries with outputs:**

**ORDER BY and HAVING**:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описаниеSELECT categoryid, COUNT(productid) AS product\_count

FROM products

GROUP BY categoryid

HAVING COUNT(productid) > 5

ORDER BY product\_count DESC;

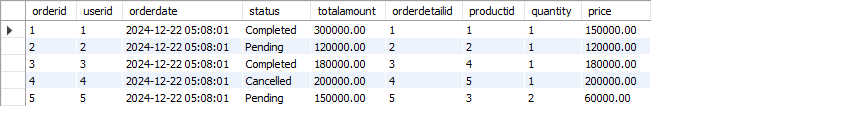
**NATURAL JOIN**:

SELECT \*

FROM orders

NATURAL JOIN orderdetails

WHERE totalamount > 100;



Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**USING Clause**:

SELECT orders.orderid, payments.amount

FROM orders

JOIN payments USING (orderid)

WHERE payments.amount > 50;

**ON Clause**:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описаниеSELECT users.firstname, orders.totalamount

FROM users

JOIN orders ON users.userid = orders.userid

WHERE orders.status = 'Completed';

**LEFT OUTER JOIN**:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеSELECT products.productname, orderdetails.quantity

FROM products

LEFT OUTER JOIN orderdetails ON products.productid = orderdetails.productid

WHERE orderdetails.quantity IS NULL;

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**RIGHT OUTER JOIN**:

SELECT orders.orderid, discounts.discountcode

FROM orders

RIGHT OUTER JOIN orderdiscounts ON orders.orderid = orderdiscounts.orderid

WHERE discounts.discountcode IS NOT NULL;

**FULL OUTER JOIN**:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описаниеSELECT users.firstname, payments.amount

FROM users

FULL OUTER JOIN payments ON users.userid = payments.userid

WHERE payments.amount IS NULL OR users.firstname IS NULL;

**Conclusion**

This project helped me understand how to design database schemas, establish relationships, and execute complex SQL queries. It strengthened my skills in database management and laid the foundation for more advanced studies.