**Фамилия и номер группы**: Нуштаев Никита Петрович ПРИ-О-21/1

1. **Код**:  
   **Запросы с условием на числовые данные:**

a. Получить все продукты с ценой больше 20.

SELECT \* FROM Products WHERE Price > 20;

b. Получить все заказы с суммой больше 100.

SELECT \* FROM Orders WHERE OrderID IN (SELECT OrderID FROM OrderItems GROUP BY OrderID HAVING SUM(Quantity \* Price) > 100);

1. **Запросы с условием на текстовые данные:**

a. Найти клиентов, чьи имена начинаются с "J".

SELECT \* FROM Customers WHERE Name LIKE 'J%';

b. Найти продукты с именами "Product A" и "Product B".

SELECT \* FROM Products WHERE Name IN ('Product A', 'Product B');

1. **Запрос с вычисляемым полем:**

Получить стоимость каждого заказа.

SELECT OrderID, SUM(Quantity \* Price) AS TotalCost FROM OrderItems GROUP BY OrderID;

1. **Запрос к нескольким таблицам (без явного указания JOIN):**

Получить информацию о заказчиках и заказах.

SELECT Customers.Name, Orders.OrderDate FROM Customers, Orders WHERE Customers.CustomerID = Orders.CustomerID;

1. **Запрос с агрегирующей функцией:**

Получить среднюю цену продуктов.

SELECT AVG(Price) AS AveragePrice FROM Products;

1. **Запрос с группировкой:**

Получить общее количество продуктов для каждого поставщика.

SELECT Suppliers.Name, COUNT(Products.ProductID) AS ProductCount FROM Suppliers, Products WHERE Suppliers.SupplierID = Products.SupplierID GROUP BY Suppliers.Name;

Задание по варианту:

Определить покупателя, который купил максимальное количество товаров:

CREATE VIEW max\_purchase\_customer AS

SELECT c.customerid, c.name, COUNT(oi.productid) AS purchased\_count

FROM customers c

JOIN orders o ON c.customerid = o.customerid

JOIN orderitems oi ON o.orderid = oi.orderid

GROUP BY c.customerid, c.name

ORDER BY purchased\_count DESC

LIMIT 1;

Для каждой покупки рассчитать общую стоимость:

CREATE VIEW order\_total\_cost AS

SELECT o.orderid, SUM(oi.quantity \* p.price) AS total\_cost

FROM orders o

JOIN orderitems oi ON o.orderid = oi.orderid

JOIN products p ON oi.productid = p.productid

GROUP BY o.orderid;

Определить сумму продажи для каждого месяца:

CREATE VIEW monthly\_sales AS

SELECT DATE\_TRUNC('month', o.orderdate) AS month, SUM(oi.quantity \* p.price) AS total\_sales

FROM orders o

JOIN orderitems oi ON o.orderid = oi.orderid

JOIN products p ON oi.productid = p.productid

GROUP BY month

ORDER BY month;

Определить покупателей, купивших товаров на сумму, превышающую среднюю сумму покупок всех покупателей:

CREATE VIEW high\_value\_customers AS

SELECT c.customerid, c.name, SUM(oi.quantity \* p.price) AS total\_purchase

FROM customers c

JOIN orders o ON c.customerid = o.customerid

JOIN orderitems oi ON o.orderid = oi.orderid

JOIN products p ON oi.productid = p.productid

GROUP BY c.customerid, c.name

HAVING SUM(oi.quantity \* p.price) > (SELECT AVG(SUM(oi.quantity \* p.price)) FROM orders o JOIN orderitems oi ON o.orderid = oi.orderid JOIN products p ON oi.productid = p.productid GROUP BY o.customerid);

Определить тип товаров, которого куплено больше всего:

CREATE VIEW most\_purchased\_type AS

SELECT p.type, COUNT(oi.productid) AS total\_purchased

FROM products p

JOIN orderitems oi ON p.productid = oi.productid

GROUP BY p.type

ORDER BY total\_purchased DESC

LIMIT 1;

**Принтскрины**:  
 















