Часть 1: WHERE
Задача 1. Найдите всех клиентов из страны 'USA', которым больше 25 лет.

```
SELECT *
FROM Customers
WHERE country = 'USA' AND age > 25;
```

ГДЕ:
SELECT *: Выбирает все столбцы (*) из таблицы.
FROM Customers: Указывает, из какой таблицы выбирать данные.
WHERE country = 'USA' AND age > 25
- country = 'USA': Выбирает только тех клиентов, у которых значение столбца country равно 'USA'.
- AND age > 25: Дополнительно фильтрует результаты, выбирая только тех клиентов, у которых значение столбца age больше 25.

Задача 2. Выведите все заказы, у которых сумма (amount) больше 1000.

```
SELECT *
FROM Orders
WHERE amount > 1000;
```

ГДЕ:
SELECT *: Выбирает все столбцы из таблицы.
FROM Orders: Указывает таблицу, из которой выбираются данные.
WHERE amount > 1000: Фильтрует результаты, выбирая только те строки, где значение столбца amount больше 1000.

Часть 2: JOIN
Задача 1. Получите список заказов вместе с именем клиента, который сделал заказ.

```
SELECT
    o.order_id,
    o.item,
    o.amount,
    c.first_name,
    c.last_name
FROM
    Orders o
INNER JOIN
    Customers c ON o.customer_id = c.customer_id;
```

ГДЕ:
SELECT o.order_id, o.item, o.amount, c.first_name, c.last_name
Эта часть запроса определяет, какие столбцы будут включены в набор данных.
FROM Orders o
Эта часть запроса указывает основную таблицу, из которой начинается выборка данных.
INNER JOIN Customers c ON o.customer_id = c.customer_id
Эта часть запроса выполняет соединение таблицы Orders с таблицей Customers.

Задача 2. Выведите список доставок со статусом и именем клиента.

```
SELECT
    s.status,
    c.first_name,
    c.last_name
FROM
```

```
    Shippings s
INNER JOIN
    Customers c ON s.customer = c.customer_id;
```

ГДЕ:
SELECT s.status, c.first_name, c.last_name:
Определяет столбцы, которые будут включены в результат запроса.
FROM Shippings s:
Указывает таблицу Shippings как основную таблицу для запроса и присваивает ей псевдоним s.
INNER JOIN Customers c ON s.customer = c.customer_id:
Выполняет внутреннее соединение (INNER JOIN) между таблицами Shippings и Customers.
Customers: Указывает таблицу Customers, с которой соединяется таблица Shippings.
c: Присваивает таблице Customers псевдоним c.
ON s.customer = c.customer_id: Записи из таблицы Shippings соединяются с записями из таблицы Customers, когда значение столбца customer в таблице Shippings совпадает со значением столбца customer_id в таблице Customers.

Часть 3: GROUP BY
Задача 1. Подсчитайте количество клиентов в каждой стране.

```
SELECT
    country,
    COUNT(*) AS customer_count
FROM
    Customers
GROUP BY
    country
ORDER BY
    customer_count DESC;
```

ГДЕ:
SELECT country, COUNT(*) AS customer_count
country Выбирает значение столбца country (страна).
COUNT(*)Подсчитывает количество строк в каждой группе. * означает все столбцы.
AS customer_count Присваивает псевдоним customer_count количеству клиентов.
FROM Customers Указывает таблицу из которой выбираются данные.
GROUP BY country Группирует строки таблицы Customers по значениям столбца country.
ORDER BY customer_count DESC Сортирует результаты по столбцу customer_count в убывающем порядке (DESC).

Задача 2. Посчитайте общее количество заказов и среднюю сумму по каждому товару

```
SELECT
    item,
    COUNT(*) total_orders,
    AVG(amount) average_amount
FROM
    Orders
GROUP BY
    item
ORDER BY
    total_orders DESC;
```

ГДЕ:
SELECT item, COUNT(*) total_orders, AVG(amount) average_amount
Эта часть запроса определяет столбцы, которые будут включены в результат запроса.
FROM Orders
Указывает таблицу из которой выбираются данные.
GROUP BY item
Группирует строки таблицы Orders по значениям столбца item.
ORDER BY total_orders DESC
Сортирует данные по столбцу total_orders в убывающем порядке (DESC).

Часть 4: ORDER BY
Задача 1. Выведите список клиентов, отсортированный по возрасту по убыванию.

```
SELECT
    *
FROM
    Customers
ORDER BY
    age DESC;
```

ГДЕ:
SELECT * Выбирает все столбцы из таблицы Customers.
FROM Customers Указывает таблицу Customers, из которой выбираются данные.
ORDER BY age DESC Сортирует данные по столбцу age в убывающем порядке (DESC).

Часть 5: SUBQUERIES
Задача 1. Найдите всех клиентов, которые сделали заказ с максимальной суммой.

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    o.amount AS order_amount
FROM
    Customers c
INNER JOIN
    Orders o ON c.customer_id = o.customer_id
WHERE
    o.amount = (SELECT MAX(amount) FROM Orders);
```

ГДЕ:
SELECT c.customer_id, c.first_name, c.last_name, o.amount AS order_amount
Эта часть запроса определяет столбцы, которые будут включены в результат запроса.
FROM Customers c INNER JOIN Orders o ON c.customer_id = o.customer_id
Указывает таблицы, которые будут использоваться в запросе, и как они связаны между собой.
WHERE o.amount = (SELECT MAX(amount) FROM Orders)
Фильтрует данные, чтобы включить только тех клиентов, которые сделали заказ с максимальной суммой.

Часть 6: WINDOW FUNCTIONS
Задача 1. Для каждого заказа добавьте колонку с суммой всех заказов этого клиента (используя оконную функцию).

```
SELECT
    order_id,
    customer_id,
    item,
    amount,
    SUM(amount) OVER (PARTITION BY customer_id) AS total_by_customer
FROM
    Orders;
```

ГДЕ:
SELECT order_id, customer_id, item, amount
Определяет, какие столбцы будут выбраны из таблицы Orders и включены в результат запроса.
SUM(amount) OVER (PARTITION BY customer_id) AS total_by_customer:
Этот запрос добавляет вычисляемый столбец total_by_customer с использованием оконной функции.
SUM(amount) Вычисляет сумму значений столбца amount (сумма заказа).
OVER (PARTITION BY customer_id) указывает, что сумма должна быть вычислена внутри каждого окна, то есть для каждой группы строк, имеющих одинаковый customer_id.
AS total_by_customer Присваивает псевдоним total_by_customer.
FROM Orders Указывает таблицу, из которой выбираются данные.

Часть 7 (Опционально)
Найдите клиентов, которые:
1. Сделали хотя бы 2 заказа (любых),
2. Имеют хотя бы одну доставку со статусом 'Delivered'. Для каждого такого клиента выведите:
- full_name (имя + фамилия),
- общее количество заказов,
- общую сумму заказов,
- страну проживания.

```
SELECT
    c.first_name || ' ' || c.last_name AS full_name,
    COUNT(DISTINCT o.order_id) AS total_orders,
    SUM(o.amount) AS total_amount,
    c.country
FROM
    Customers c
JOIN
    Orders o ON c.customer_id = o.customer_id
JOIN
    Shippings s ON c.customer_id = s.customer
WHERE
    s.status = 'Delivered'
GROUP BY
```

```
    c.customer_id, c.first_name, c.last_name, c.country
HAVING
    COUNT(DISTINCT o.order_id) >= 2;
```

ГДЕ:
c.first_name || ' ' || c.last_name AS full_name Выводит полное имя клиента, объединяя имя и фамилию.
COUNT(DISTINCT o.order_id) AS total_orders Выводит общее количество заказов для каждого клиента.
SUM(o.amount) AS total_amount Выводит общую сумму всех заказов для каждого клиента.
c.country Выводит страну проживания клиента.
FROM Customers c JOIN Orders o ON c.customer_id = o.customer_id JOIN Shippings s ON c.customer_id = s.customer Этот раздел определяет таблицы, из которых будут извлекаться данные, и как они связаны между собой.
WHERE s.status = 'Delivered'
s.status = 'Delivered' Оставляет только те строки, в которых статус отгрузки (s.status) равен "Delivered".
GROUP BY c.customer_id, c.first_name, c.last_name, c.country:
Группируем по идентификатору клиента, имени, фамилии и стране.
HAVING COUNT(DISTINCT o.order_id) >= 2
WHERE фильтрует до группировки, а HAVING - после.
COUNT(DISTINCT o.order_id) >= 2 Оставляет только тех клиентов, у которых общее количество уникальных заказов >= 2.