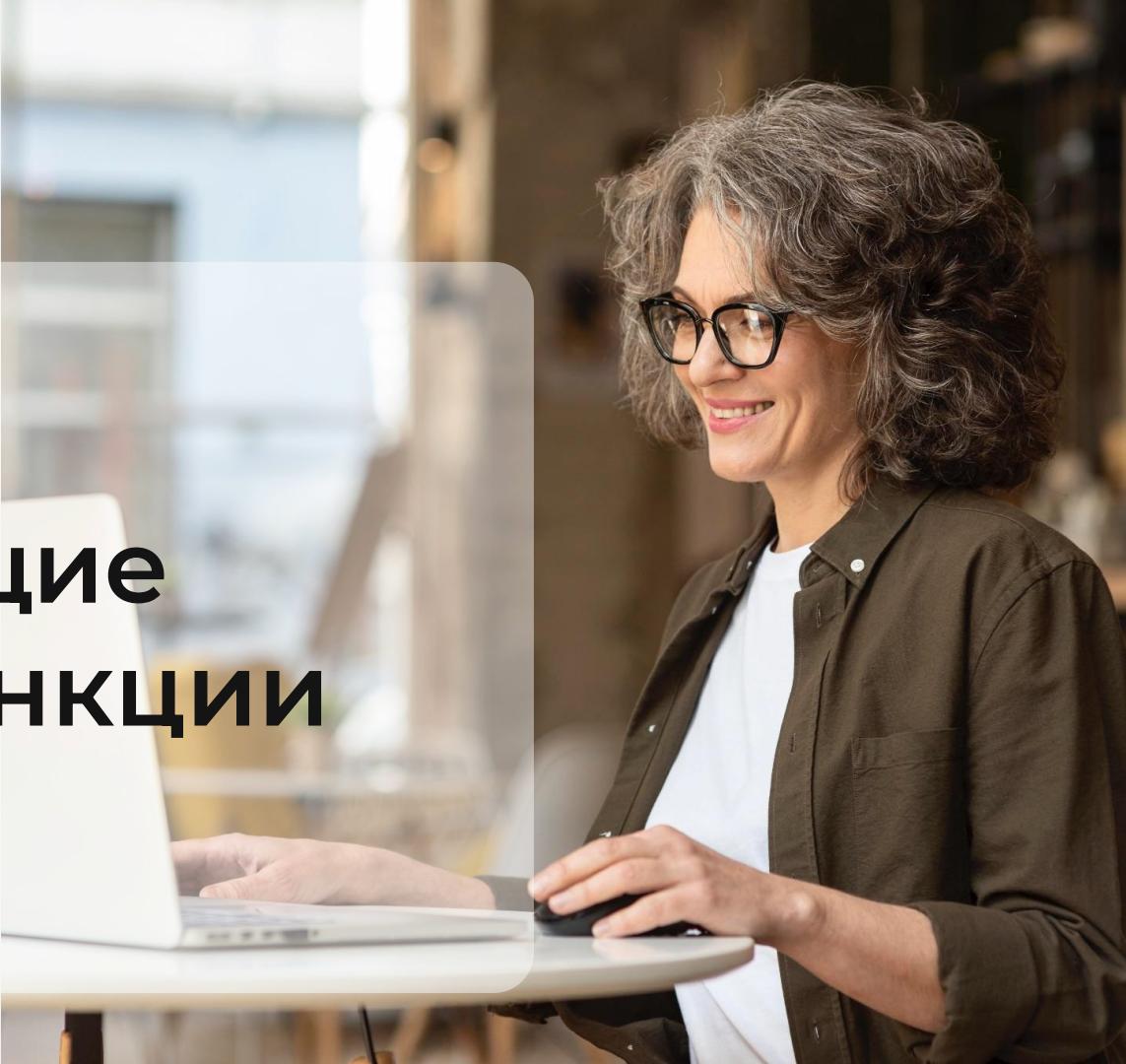


Database

Агрегирующие оконные функции



Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)

Важно



Камера должна быть включена на протяжении всего занятия



В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы



Вести себя уважительно и этично по отношению к остальным участникам занятия



Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях



Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение



Основные различия GROUP BY и оконных функций



Синтаксис оконных функций



Основные концепции оконных функций

План занятия

- Основные агрегирующие оконные функции
- Примеры функций
- Подсчет кумулятивных значений

ОСНОВНОЙ БЛОК



Основные агрегирующие оконные функции



Агрегирующие оконные функции

Это функции, позволяющие выполнять вычисления, которые агрегируют данные в рамках определенного окна (группы строк), сохраняя при этом детализированные строки.

Основные агрегирующие оконные функции

SUM()

AVG()

MIN()

MAX()

COUNT()

ВОПРОСЫ



Примеры функций

ЗАДАНИЕ

Задача

- Для каждого клиента посчитать общую сумму всех его заказов.

OrderID	CustomerID	OrderAmount
1	101	300
2	101	500
3	102	150
4	101	700
5	102	200

Решение

```
SELECT OrderID, CustomerID, OrderAmount,  
       SUM(OrderAmount) OVER (PARTITION BY CustomerID) AS TotalCustomerOrders  
FROM Orders;
```

OrderID	CustomerID	OrderAmount	TotalCustomerOrders
1	101	300	1500
2	101	500	1500
4	101	700	1500
3	102	150	350
5	102	200	350

ВОПРОСЫ

ЗАДАНИЕ

Задача

1. Подсчитать количество заказов для каждого клиента.

Решение

```
SELECT OrderID, CustomerID, OrderAmount,  
       COUNT(*) OVER (PARTITION BY CustomerID) AS OrdersPerCustomer  
FROM Orders;
```

OrderID	CustomerID	OrderAmount	OrdersPerCustomer
1	101	300	3
2	101	500	3
4	101	700	3
3	102	150	2
5	102	200	2

ВОПРОСЫ

ЗАДАНИЕ

Задача

1. Из таблицы products выведите максимальный list_price для каждой строки, имя продукта и его list_price.
2. Используя предыдущий запрос, посчитайте разницу в процентах между ценой продукта и максимальной ценой.
3. Посчитайте количество продуктов в каждой категории с помощью оконной функции. Оптимально ли использование оконной функции для выполнения этого задания.
4. Найдите разницу между standard_cost продукта и средним list_price по всей таблицы для каждой строки.
5. Можно ли решить предыдущее задание без оконных функций.

Задачи



```
SELECT product_name, list_price, MAX(list_price) OVER ()  
FROM products
```

```
SELECT product_name, (MAX(list_price) OVER () - list_price)/(MAX(list_price) OVER ())*100  
FROM products;
```

```
SELECT category, COUNT(id) OVER (partition by category)  
FROM products
```

Задачи



```
SELECT product_name, standard_cost - AVG(list_price) OVER ()  
FROM products
```

```
with avg_price as (SELECT AVG(list_price) as ap FROM products)  
SELECT product_name, standard_cost - avg_price.ap  
FROM products JOIN avg_price ON 1=1
```

ВОПРОСЫ



Подсчет кумулятивных значений



Кумулятивная сумма

Это сумма значений от начала до текущей строки.

Подсчет кумулятивных значений



При подсчете кумулятивных значений с помощью оконных функций в оператор OVER дополнительно вписывается порядок сортировки ORDER BY.



Кумулятивные значения рассчитываются только при использовании столбца с датами в ORDER BY.

ЗАДАНИЕ

Задача

1. На основе таблицы ORDERS в базе данных Shop рассчитать кумулятивную сумму заказов

SaleID	SaleDate	SaleAmount
1	2024-01-01	100
2	2024-01-02	150
3	2024-01-03	200

Решение

```
SELECT ORDER_ID, CUST_ID, ODATE, AMT,  
       SUM(AMT) OVER (ORDER BY ODATE) AS CumulativeSales  
FROM ORDERS;
```

SaleID	SaleDate	SaleAmount	CumulativeSales
1	2024-01-01	100	100
2	2024-01-02	150	250
3	2024-01-03	200	450

ВОПРОСЫ

ЗАДАНИЕ

Задача

Задача: На основе таблицы ORDERS в базе данных Shop рассчитать среднюю сумму заказов для каждого игрока

OrderID	CustomerID	OrderDate	OrderAmount
1	101	2024-01-01	300
2	101	2024-01-05	500
3	102	2024-01-02	150
4	101	2024-01-10	700
5	102	2024-01-07	200

Решение

```
SELECT ORDER_ID, CUST_ID, ODATE, AMT,  
       AVG(AMT) OVER (PARTITION BY CUST_ID ORDER BY ODATE) AS RunningAvg  
FROM ORDERS;
```

OrderID	CustomerID	OrderDate	OrderAmount	RunningAvg
1	101	2024-01-01	300	300
2	101	2024-01-05	500	400
4	101	2024-01-10	700	500
3	102	2024-01-02	150	150
5	102	2024-01-07	200	175

ВОПРОСЫ

Домашнее задание

1. Для каждого заказа order_id выведите минимальный, максимальный и средний unit_cost.
2. Оставьте только уникальные строки из предыдущего запроса.
3. Посчитайте стоимость продукта в заказе как quantity*unit_cost.
 - Выведите суммарную стоимость продуктов с помощью оконной функции.
 - Сделайте то же самое с помощью GROUP BY.
4. Посчитайте количество заказов по дате получения и posted_to_inventory.
 - Если оно превышает 1 то выведите '>1' в противном случае '=1'.
5. Выведите purchase_order_id, date_received и вычисленный столбец.

Заключение

