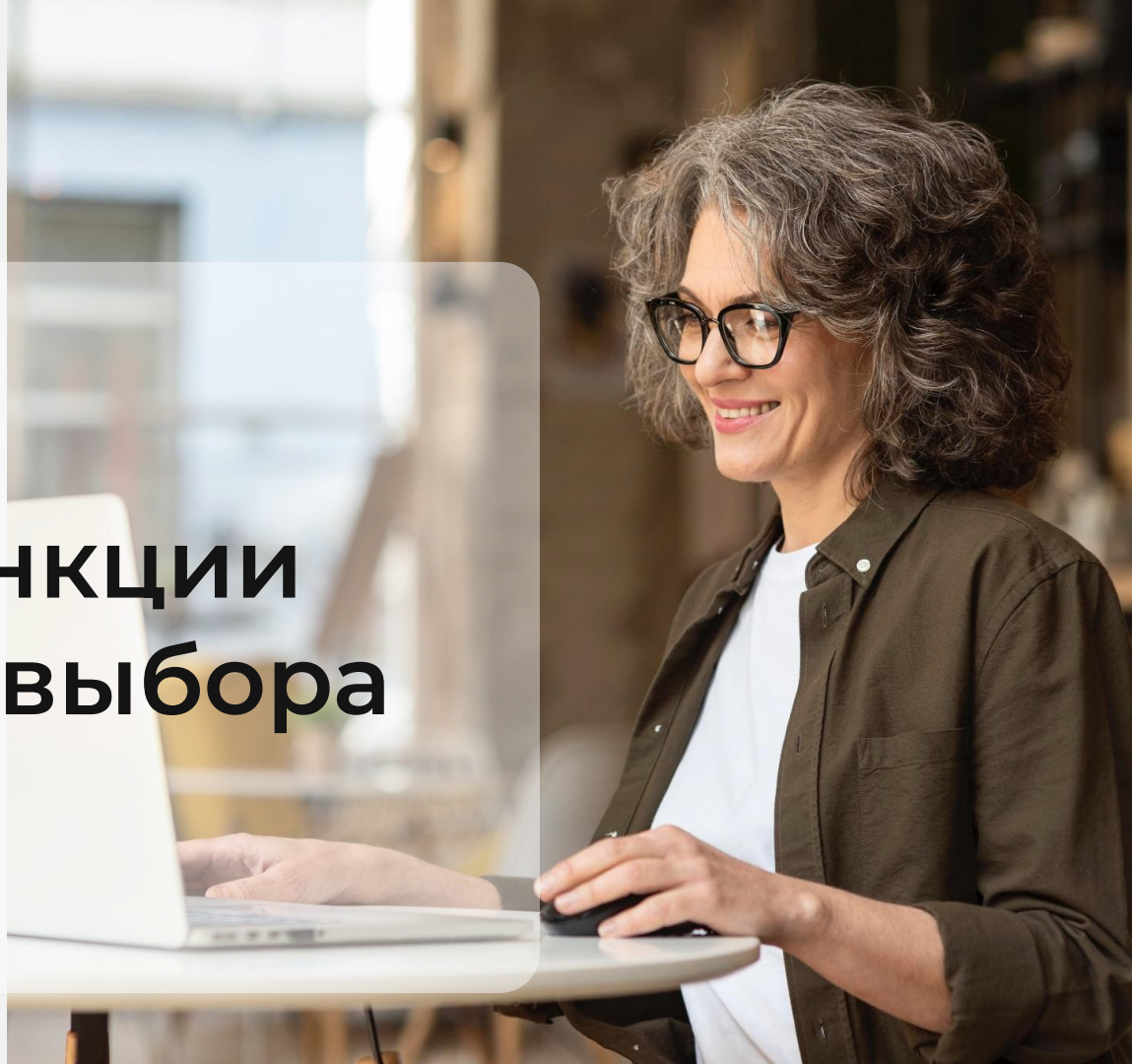


Node.JS

# Оконные функции смещения и выбора



# 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 ранжирующие функции



Применение на практике



Общий синтаксис



Примеры оконных функций ранжирования

# План занятия

- Оконные функции смещения и выбора
- Общий синтаксис
- Основные оконные функции смещения и выбора

# База данных с доступом на чтение

**hostname:** ich-db.edu.itcareerhub.de

**username:** ich1

**password:** password



# ОСНОВНОЙ БЛОК





Оконные функции  
смещения и выбора



## Оконные функции смещения и выбора

Эти функции позволяют выполнять вычисления, которые зависят от значений в других строках в пределах определенного окна.

# Основные оконные функции смещения и выбора

LEAD()

LAG()

FIRST\_VALUE()

LAST\_VALUE()

NTH\_VALUE()



# ВОПРОСЫ





# Общий синтаксис

# Оконные функции смещения и выбора



## Общий синтаксис

```
SELECT column1,  
       window_function(column2) OVER (  
           PARTITION BY column3  
           ORDER BY column4 ) AS  
result_column  
FROM table_name;
```

## Пояснение

- `window_function(column2)`: Оконная функция смещения или выбора, которая применяется
- `PARTITION BY column3`: Разделяет данные на группы, если требуется
- `ORDER BY column4`: Устанавливает порядок строк в рамках окна



# ВОПРОСЫ





Основные оконные  
функции смещения и  
выбора



## LEAD()

Это функция, которая возвращает значение из следующей строки в пределах окна, которое определяется ORDER BY.

# Пример использования функции LEAD()

## Исходные данные

1. Найти значение SaleAmount следующей продажи по дате.

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

# Пример использования функции LEAD()

## Решение и результат

```
SELECT SaleID, SaleDate, SaleAmount,
       LEAD(SaleAmount, 1) OVER (ORDER BY SaleDate) AS NextSaleAmount
FROM Sales;
```

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



## LAG()

Это функция, которая возвращает значение из предыдущей строки в пределах окна.

# Пример использования функции LAG()

## Исходные данные

1. Найти значение SaleAmount предыдущей продажи по дате.

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

# Пример использования функции LAG()

## Решение и результат

```
SELECT SaleID, SaleDate, SaleAmount,
       LAG(SaleAmount, 1) OVER (ORDER BY SaleDate) AS PrevSaleAmount
FROM Sales;
```

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



## FIRST\_VALUE()

Это функция, которая возвращает первое значение в рамках окна.

# Пример использования функции FIRST\_VALUE()

## Исходные данные

1. Найти первую цену продажи для каждого месяца.

OrderID	OrderDate	UnitPrice
1	2024-01-01	10
2	2024-01-05	20
3	2024-02-01	15
4	2024-02-15	25

# Пример использования функции FIRST\_VALUE()

## Решение и результат

```
SELECT OrderID, OrderDate, UnitPrice,
       FIRST_VALUE(UnitPrice) OVER (PARTITION BY EXTRACT(YEAR FROM OrderDate), EXTRACT(MONTH
FROM OrderDate) ORDER BY OrderDate) AS FirstPriceOfMonth
FROM OrderDetails;
```

OrderID	OrderDate	UnitPrice	FirstPriceOfMonth
1	2024-01-01	10	10
2	2024-01-05	20	10
3	2024-02-01	15	15
4	2024-02-15	25	15



## LAST\_VALUE()

Это функция, которая возвращает последнее значение в рамках окна.

# Пример использования функции LAST\_VALUE()

## Исходные данные

1. Найти последнюю цену продажи для каждого месяца.

OrderID	OrderDate	UnitPrice
1	2024-01-01	10
2	2024-01-05	20
3	2024-02-01	15
4	2024-02-15	25

# Пример использования функции LAST\_VALUE()

## Решение и результат

```
SELECT OrderID, OrderDate, UnitPrice,
       LAST_VALUE(UnitPrice) OVER (PARTITION BY EXTRACT(YEAR FROM OrderDate), EXTRACT(MONTH
FROM OrderDate) ORDER BY OrderDate) AS LastPriceOfMonth
FROM OrderDetails;
```

OrderID	OrderDate	UnitPrice	LastPriceOfMonth
1	2024-01-01	10	20
2	2024-01-05	20	20
3	2024-02-01	15	25
4	2024-02-15	25	25



## NTH\_VALUE()

Это функция, которая возвращает Возвращает  
n-е значение в рамках окна.

# Пример использования функции NTH\_VALUE()

## Исходные данные

1. Найти третью цену продажи для каждого месяца.

OrderID	OrderDate	UnitPrice
1	2024-01-01	10
2	2024-01-05	20
3	2024-01-10	30
4	2024-02-01	15
5	2024-02-15	25
6	2024-02-20	35

# Пример использования функции NTH\_VALUE()

## Решение и результат

```
SELECT OrderID, OrderDate, UnitPrice,
       NTH_VALUE(UnitPrice, 3) OVER (PARTITION BY EXTRACT(YEAR FROM OrderDate),
EXTRACT(MONTH FROM OrderDate) ORDER BY OrderDate) AS ThirdPriceOfMonth
FROM OrderDetails;
```

OrderID	OrderDate	UnitPrice	ThirdPriceOfMonth
1	2024-01-01	10	30
2	2024-01-05	20	30
3	2024-01-10	30	30
4	2024-02-01	15	35
5	2024-02-15	25	35
6	2024-02-20	35	35



# ВОПРОСЫ





# ЗАДАНИЕ



# Работа с таблицей `purchase_orders`

1. Из таблицы `purchase_orders` для каждого поставщика `supplier_id` выведите дату создания заказа, а также дату создания предыдущего заказа. Посчитайте разницу между этими датами.
2. Измените предыдущий запрос таким образом, чтобы узнать среднее время между двумя заказами.
3. Напишите аналогичный второму заданию запрос, но с использованием функции `LEAD`. Сравните результаты.
4. Найдите самую раннюю дату `submitted_date` для каждого менеджера `created_by`. Решите данное задание используя оконные функции `MIN` и `FIRST VALUE`. Сравните результаты.



# ВОПРОСЫ



# Домашнее задание

Таблица `purchase_orders`

1. Для каждого `product_id` выведите `inventory_id` а также предыдущий и последующей `inventory_id` по убыванию `quantity`.
2. Выведите максимальный и минимальный `unit_price` для каждого `order_id` с помощью функции `FIRST VALUE`. Выведите `order_id` и полученные значения.
3. Выведите `order_id` и столбец с разнице между `unit_price` для каждой заказа и минимальным `unit_price` в рамках одного заказа.  
! Задачу решить двумя способами - с помощью `First Value` и `MIN`.
4. Присвойте ранг каждой строке, используя `RANK` по убыванию `quantity`.
5. Из предыдущего запроса выберите только строки с рангом до 10 включительно.

## Заключение

