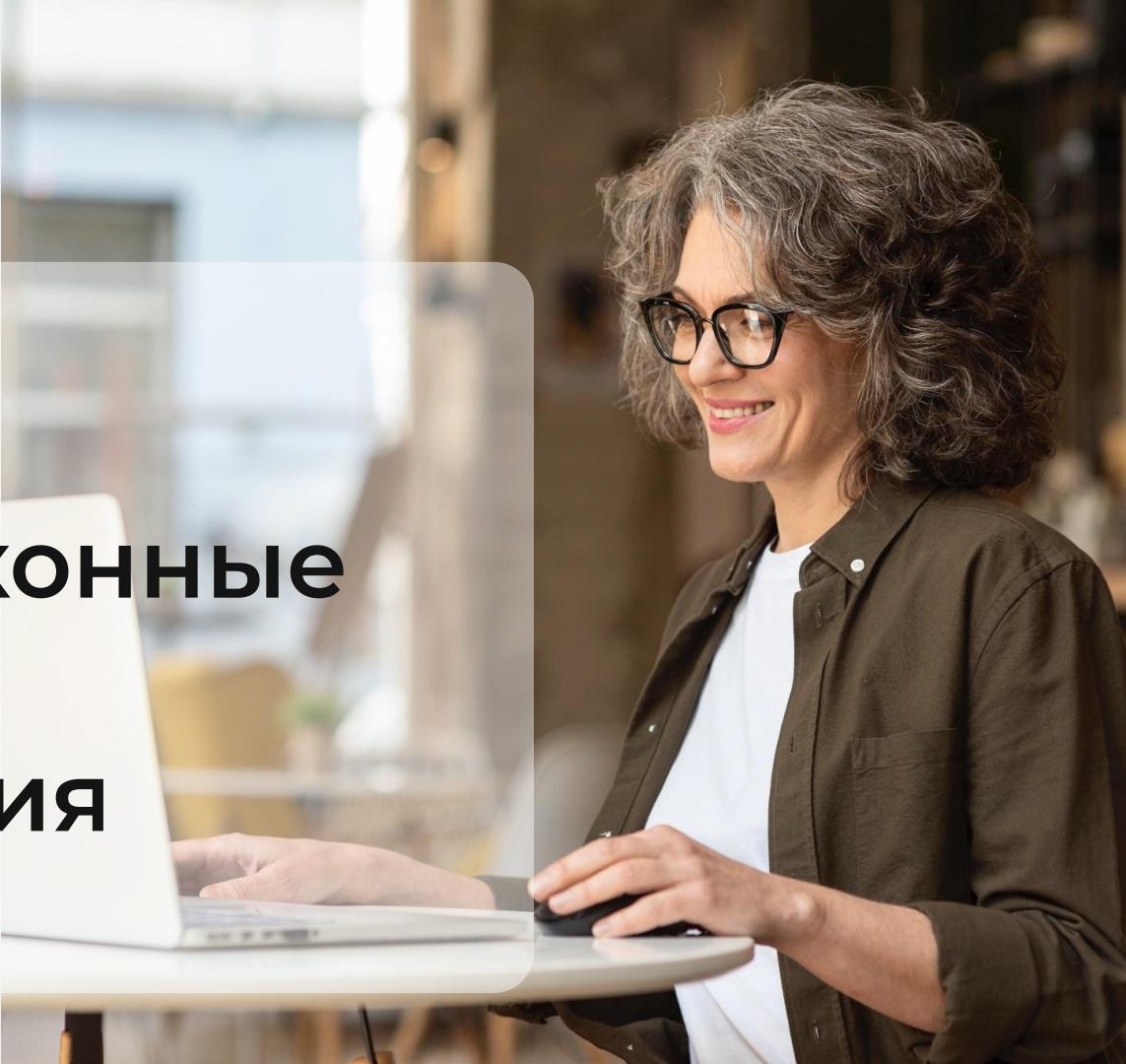


Node.JS

# Основные оконные функции ранжирования



# 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 Агрегирующие Оконные Функции

 Основные агрегирующие оконные функции

 Подсчет кумулятивных значений

# План занятия

- Основные ранжирующие функции
- Применение на практике
- Общий синтаксис
- Примеры Оконных Функций Ранжирования

База данных для работы на занятии

# База данных с доступом на чтение

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

**username:** ich1

**password:** password

# ОСНОВНОЙ БЛОК



# Основные ранжирующие функции

# Основные ранжирующие функции

---

ROW\_NUMBER()

RANK()

DENSE\_RANK()

NTILE()



## ROW\_NUMBER()

Это функция, которая присваивает уникальный номер каждой строке в рамках окна.



## RANK()

Это функция, которая присваивает ранги строкам, пропуская ранги при одинаковых значениях.



## DENSE\_RANK()

Это функция, которая присваивает ранги строкам, без пропуска рангов при одинаковых значениях.



## NTILE()

Это функция, которая делит строки на равные части (квантили) и присваивает номер группе.

# ВОПРОСЫ



**Применение на  
практике**

# Применение на практике

Анализ позиций и рангов

Упрощение сложных запросов

Упорядочение и группировка данных

Рейтинг товаров или сотрудников

Замена подзапросов

Группировка данных

Оценка результатов конкурсов

Избежание дублирования данных

Создание отчетов

# Применение на практике

Анализ позиций и рангов

Упрощение сложных запросов

Упорядочение и группировка данных

Рейтинг товаров или сотрудников

Замена подзапросов

Группировка данных

Оценка результатов конкурсов

Избежание дублирования данных

Создание отчетов

# Применение на практике

Анализ позиций и рангов

Упрощение сложных запросов

Упорядочение и группировка данных

Рейтинг товаров или сотрудников

Замена подзапросов

Группировка данных

Оценка результатов конкурсов

Избежание дублирования данных

Создание отчетов

# ВОПРОСЫ



# Общий синтаксис

# Ранжирующие функции



## Общий синтаксис

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

## Пояснения

- ranking\_function(column2) - Оконная функция ранжирования, которая применяется.
- PARTITION BY column3 - Разделяет данные на группы, если требуется.
- ORDER BY column4 - Устанавливает порядок строк в рамках окна для ранжирования.

# ВОПРОСЫ



# Примеры оконных функций ранжирования

# Пример 1. ROW\_NUMBER()

## Исходные данные

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

# Пример 1. ROW\_NUMBER()

## Решение и результат

```
SELECT SaleID, SaleDate, SaleAmount,  
       ROW_NUMBER() OVER (ORDER BY SaleDate) AS RowNum  
FROM Sales;
```

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

## Пример 2. RANK()

### Исходные данные

ProductID	DepartmentID	SaleAmount
1	10	300
2	10	300
3	10	200
4	20	500
5	20	400

## Пример 2. RANK()

### Решение и результат

```
SELECT ProductID, DepartmentID, SaleAmount,  
       RANK() OVER (PARTITION BY DepartmentID ORDER BY SaleAmount DESC) AS Rank  
FROM ProductSales;
```

ProductID	DepartmentID	SaleAmount	Rank
1	10	300	1
2	10	300	1
3	10	200	3
4	20	500	1
5	20	400	2

## Пример 3. DENSE\_RANK()

### Исходные данные

ProductID	SaleAmount
1	300
2	300
3	200
4	500
5	400

## Пример 3. DENSE\_RANK()

### Решение и результат

```
SELECT ProductID, SaleAmount,  
       DENSE_RANK() OVER (ORDER BY SaleAmount DESC) AS DenseRank  
FROM ProductSales;
```

ProductID	SaleAmount	DenseRank
1	300	1
2	300	1
3	200	3
4	500	1
5	400	2

# Пример 4. NTILE()

## Исходные данные

SaleID	SaleAmount
1	100
2	200
3	300
4	400
5	500

# Пример 4. NTILE()

## Решение и результат

```
SELECT SaleID, SaleAmount,  
       NTILE(3) OVER (ORDER BY SaleAmount) AS Quartile  
FROM Sales;
```

SaleID	SaleAmount	Quartile
1	100	1
2	200	1
3	300	2
4	400	2
5	500	3

# ВОПРОСЫ

# ПРАКТИЧЕСКАЯ РАБОТА

# Применение функции ранжирования

1. Присвоить ранг продукту без пропусков значений в ранге от больше себестоимости к меньшей. Вывести ТОП 10 продуктов product\_name.
2. Пронумеровать строки в таблице в зависимости от названия товара от А до Z.
3. Разделить все продукты на 4 равных группы в зависимости от list\_price.  
Вывести имя продукта, list\_price и номер группы.

# ВОПРОСЫ

## Заключение

