

Database

# Операторы JOIN и UNION



# 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 типы связей между таблицами



Ключи для связи таблиц



Основные характеристики внешнего ключа



Графическое отображение связей. ER-диаграммы



Основные компоненты ER-диаграммы

# План занятия

- Операторы JOIN и UNION
- Операторы JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN



# ОСНОВНОЙ БЛОК





# Операторы JOIN и UNION



## Операторы JOIN и UNION

Это операторы, которые используются для горизонтального объединения результатов двух или более SELECT запросов в один результирующий набор.

# UNION



## Определение

- При объединении строк удаляет полные дубликаты
- Требуется очень много памяти

## Код

```
SELECT * FROM table1
UNION
SELECT * FROM table 2
```

# UNION ALL



## Определение

- объединяет результаты и сохраняет все строки, включая дубликаты.

## Код

```
SELECT * FROM table1
UNION ALL
SELECT * FROM table 2;
SELECT name, email FROM table1
UNION ALL
SELECT name, email FROM table2
```

# Основные особенности

UNION автоматически удаляет дубликаты, если нужно сохранить все дубликаты, используется UNION ALL.

Все запросы, объединяемые с помощью UNION, должны иметь одинаковое количество столбцов, и соответствующие столбцы должны иметь совместимые типы данных.



# ВОПРОСЫ





# ЗАДАНИЕ



# Задание

Выведите в одну общую выборку из таблиц customers и employees имена и фамилии клиентов и сотрудников.



# Решение

```
USE northwind;
```

```
SELECT first_name, last_name from  
employees
```

```
UNION ALL
```

```
SELECT first_name, last_name from  
customers
```



# Задание

Добавьте дополнительный столбец в котором будет значение employee для сотрудника и customer для клиента.



# Решение

```
USE northwind;
```

```
SELECT first_name, last_name, 'employee'  
as status
```

```
from employees
```

```
UNION ALL
```

```
SELECT first_name, last_name, 'customer'  
as status from customers
```





# ВОПРОСЫ





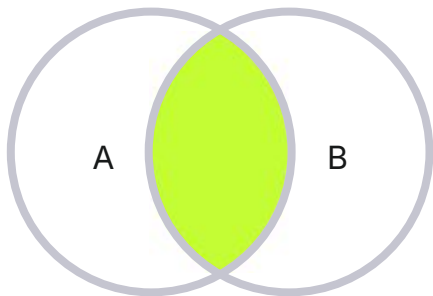
**Операторы JOIN,  
LEFT JOIN , RIGHT  
JOIN, CROSS JOIN**



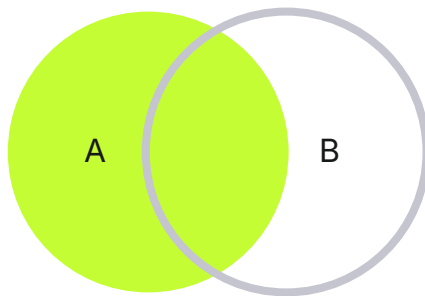
## Операторы JOIN

Это операторы, которые используются для объединения строк из двух или более таблиц на основе логической связи между ними, обычно с помощью общего поля.

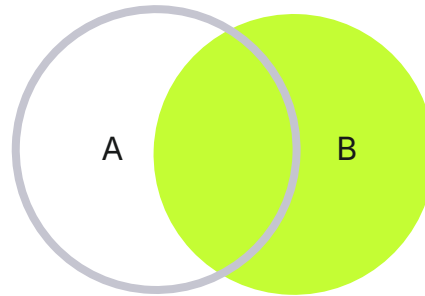
# Операторы JOIN



Внутреннее  
соединение

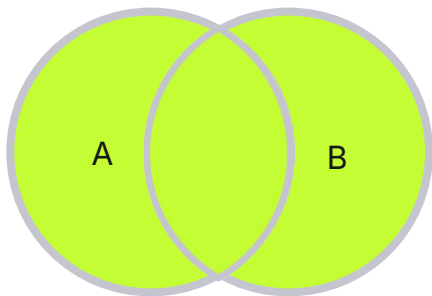


Левое внешнее  
соединение

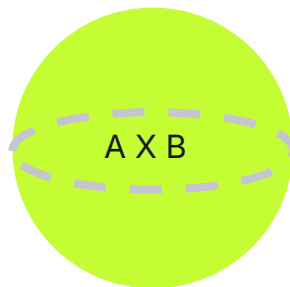


Правое внешнее  
соединение

# Операторы JOIN



Полное внешнее  
соединение



Декартово (перекрёстное)  
соединение

# Операторы JOIN



- **таблица1** и **таблица2** — это таблицы, которые вы хотите соединить.
- **условие\_совпадения** — это логическое условие, обычно выражение **таблица1.колонка = таблица2.колонка**, которое определяет, как строки из одной таблицы соединяются со строками из другой.

SELECT столбцы

FROM таблица1

INNER/LEFT/RIGHT JOIN

таблица2

ON таблица1.колонка = таблица2. колонка;

Общий синтаксис JOINов



# ЗАДАНИЕ



# Использование операторов JOIN

1. Выведите все строки из объединенных таблиц `employees` и `employee_privileges` с помощью `INNER/RIGHT` и `LEFT JOIN`. Объясните полученные результаты.
2. Выведите идентификаторы заказов из таблицы `order_details`. Дополнительно выведите вместо `product_id` столбец с именем продукта `product_name` из `products`.
3. Используя предыдущий запрос, посчитайте количество заказов для каждого наименования продукта.
4. Выведите идентификаторы заказов из таблицы `order_details`. Дополнительно выведите вместо `product_id` столбец с именем продукта `product_name` из `products` и столбец `payment_amount` из таблицы `purchase_orders`.
5. Оставить все строки из таблицы `order_details`.

# Использование операторов JOIN

```
SELECT *
from employees as e
JOIN employee_privileges as ep;
SELECT *
from employees as e
LEFT JOIN employee_privileges as ep;
SELECT *
from employees as e
RIGHT JOIN employee_privileges as ep;
```

```
SELECT order_id, product_name
from order_details as od
JOIN products as p
ON od.product_id = p.id
```



# Использование операторов JOIN



```
SELECT product_name, COUNT(order_id)
from order_details as od
JOIN products as p
ON od.product_id = p.id
GROUP BY product_name
```

```
SELECT product_name, order_id, po.payment_amount
from order_details as od
LEFT JOIN products as p
ON od.product_id = p.id
LEFT JOIN purchase_orders po
ON od.purchase_order_id = po.id
```



# ВОПРОСЫ



# Домашнее задание

1. Выведите одним запросом с использованием UNION столбцы id, employee\_id из таблицы orders и соответствующие им столбцы из таблицы purchase\_orders. В таблице purchase\_orders created\_by соответствует employee\_id.
2. Из предыдущего запроса удалите записи там где employee\_id не имеет значения. Добавьте дополнительный столбец со сведениями из какой таблицы была взята запись.
3. Выведите все столбцы таблицы order\_details а также дополнительный столбец payment\_method из таблицы purchase\_orders. Оставьте только заказы для которых известен payment\_method.
4. Выведите заказы orders и фамилии клиентов customers для тех заказов по которым были инвойсы  
таблица invoices.
5. Подсчитайте количество инвойсов для каждого клиента из предыдущего запроса.

## Заключение

