

Урок 6.2. Операторы JOIN и UNION

Операторы JOIN и UNION	2
Задание для закрепления	4
Операторы JOIN, LEFT JOIN , RIGHT JOIN, CROSS JOIN	5
Общий синтаксис JOINов	7
Задание для закрепления	8

Операторы JOIN и UNION



Важно!

База данных с доступом на чтение:

hostname: ich-db.edu.itcareerhub.de

username: ich1

password: password



Операторы **UNION** и **UNION ALL** — это операторы, которые используются для горизонтального объединения результатов двух или более **SELECT** запросов в один результирующий набор.

UNION

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

Unset

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

UNION ALL

- объединяет результаты и сохраняет все строки, включая дубликаты.

Unset

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

Основные особенности:

- 1) UNION автоматически удаляет дубликаты, если нужно сохранить все дубликаты, используется UNION ALL.
- 2) Все запросы, объединяемые с помощью UNION, должны иметь одинаковое количество столбцов, и соответствующие столбцы должны иметь совместимые типы данных.



Задание для закрепления

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

Unset

```
USE northwind;
```

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

```
UNION ALL
```

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

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

Unset

```
USE northwind;
```

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

```
from employees
```

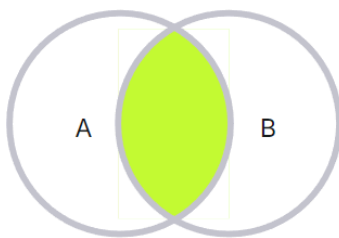
```
UNION ALL
```

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

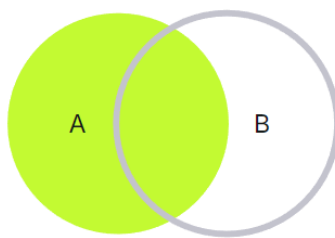
Операторы JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN



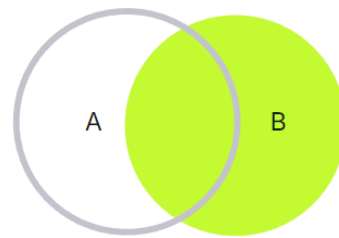
Операторы JOIN — это операторы, которые используются для объединения строк из двух или более таблиц на основе логической связи между ними, обычно с помощью общего поля.



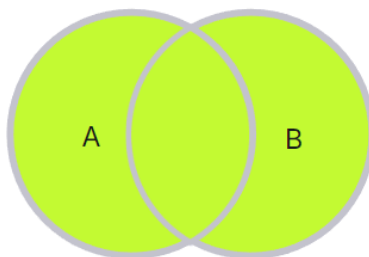
Внутреннее
соединение



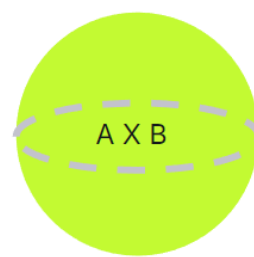
Левое внешнее
соединение



Правое внешнее
соединение



Полное внешнее
соединение



Декартово (перекрёстное)
соединение

1. INNER JOIN



INNER JOIN — это оператор, который возвращает только те строки, которые имеют совпадения в обеих таблицах.

Если в одной из таблиц нет соответствующего значения, строка не включается в результирующий набор.

2. LEFT JOIN (или LEFT OUTER JOIN)



LEFT JOIN — это оператор, который возвращает все строки из левой таблицы и совпадающие строки из правой таблицы.

Если совпадений нет, возвращаются строки из левой таблицы с **NULL** значениями для полей правой таблицы.

3. RIGHT JOIN



RIGHT JOIN — это оператор, который возвращает все строки из правой таблицы и совпадающие строки из левой таблицы.

Если совпадений нет, возвращаются строки из правой таблицы с **NULL** значениями для полей левой таблицы. Обычно не используется поскольку такое соединения мы можем реализовать с помощью **LEFT JOIN** просто поменяв таблички местами

4. CROSS JOIN



CROSS JOIN — это оператор, который возвращает декартово произведение строк обеих таблиц.

Это означает, что каждая строка из первой таблицы соединяется с каждой строкой из второй таблицы. Редко используется из-за большого объема потребления памяти

5. FULL JOIN (Не реализован в My SQL)



FULL JOIN — это оператор, который возвращает все строки, когда есть совпадения в одной из таблиц.

Если в одной из таблиц нет соответствующего значения, строки будут заполнены **NULL** значениями.

Общий синтаксис JOINов

Unset

SELECT столбцы

FROM таблица1

INNER/LEFT/RIGHT JOIN

таблица2

ON таблица1.колонка = таблица2.колонка;

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.

Общие строки для таблиц очень удобно смотреть на ER диаграмме.



Задание для закрепления

Выведите все строки из объединенных таблиц `employees` и `employee_privileges` с помощью `INNER/RIGHT` и `LEFT JOIN`. Объясните полученные результаты.

Unset

```
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;
```

Выведите идентификаторы заказов из таблицы `order_details`. Дополнительно выведите вместо `product_id` столбец с именем продукта `product_name` из `products`.

Unset

```
SELECT order_id, product_name

from order_details as od

JOIN products as p

ON od.product_id = p.id
```

Используя предыдущий запрос, посчитайте количество заказов для каждого наименования продукта.

Unset

```
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
```

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`.

Оставить все строки из таблицы `order_details`.

Unset

```
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
```