# Тема №4 - Стандартные соединения и объединения таблиц.

PostgreSQL оператор UNION используется для объединения результирующих наборов из 2 или более операторов SELECT. Он удаляет повторяющиеся строки между различными операторами SELECT.

Каждый оператор SELECT в операторе UNION должен иметь одинаковое количество полей в наборах результатов с одинаковыми типами данных.

Общий синтаксис для оператора UNION в PostgreSQL выглядит следующим образом:

SELECT expression1, expression2,… expression\_n

FROM tables

[WHERE conditions]

UNION

SELECT expression1, expression2,… expression\_n

FROM tables

[WHERE conditions];

Где:

**expression1, expression2,… expression\_n –** это столбцы или вычисления, которые вы хотите получить.

**tables –** это таблицы, из которых вы хотите получить записи. В операторе FROM должна быть указана хотя бы одна таблица.

**WHERE conditions –** это условия, которые должны быть выполнены для записей, которые будут выбраны (они являются необязательными).

**Например,** для того чтобы получить в одной таблице ширину и высоту предметов на складе, цена которых не является нулевой, можно использовать следующий запрос:

SELECT "WIDTH","LENGTH"

FROM cloth

WHERE "PRICE" IS NOT NULL

UNION

SELECT "WIDTH","LENGTH"

FROM product

WHERE "PRICE" IS NOT NULL

При работе с оператором UNION имеются примечания:

1. В обоих операторах SELECT должно быть одинаковое количество выражений.
2. Поскольку оператор UNION по умолчанию удаляет все повторяющиеся строки из набора результатов, предоставление модификатора UNION DISTINCT не влияет на результаты.
3. Имена столбцов из первого оператора SELECT в операторе UNION используются в качестве имен столбцов для набора результатов.

Через оператор UNION можно вернуть данные из полей с одним и тем же названием, но из разных таблиц (если оба поля имеют одинаковый тип данных).

**Например**, выведем все цены из двух разных таблиц в одном запросе:

SELECT "PRICE"

FROM cloth

UNION

SELECT "PRICE"

FROM product

В рамках данных таблиц, мы получили в ответ 222 строки с данными, выглядит неплохо. В этом примере оператора PostgreSQL UNION, если “PRICE” появилось как в таблице products, так и в categories, оно будет отображаться в вашем результирующем наборе один раз. Оператор PostgreSQL UNION удаляет дубликаты. Если вы не хотите удалить дубликаты, попробуйте использовать PostgreSQL оператор **UNION ALL**. Вот так:

SELECT "PRICE"

FROM cloth

UNION **ALL**

SELECT "PRICE"

FROM product

Теперь мы получаем на выходе все строки, в рамках данных таблиц их количество равно 1439. UNION ALL работает быстрее, чем UNION, т.к. отсутствует принудительная сортировка для устранения дубликатов.

Оператор PostgreSQL UNION может использовать оператор **ORDER BY** для упорядочивания результатов запроса. Например, если нам нужно отсортировать данные второго столбца (в нашем случае это столбец “LENGTH”), то в ORDER BY запишем цифру 2:

SELECT "WIDTH","LENGTH"

FROM cloth

WHERE "PRICE" IS NOT NULL

UNION

SELECT "WIDTH","LENGTH"

FROM product

WHERE "PRICE" IS NOT NULL

**ORDER BY** 2

Если нужно отсортировать по первому столбцу, запишем:

SELECT "WIDTH","LENGTH"

FROM cloth

WHERE "PRICE" IS NOT NULL

UNION

SELECT "WIDTH","LENGTH"

FROM product

WHERE "PRICE" IS NOT NULL

**ORDER BY** 1

Поскольку в SELECT мы взяли 2 столбца, если запишем после ORDER BY 0 или 3 и более, то в ответ мы получим ошибку.

Через UNION можно сочетать между собой бесконечное количество запросов. Операции над множествами тоже можно вкладывать и соединять, например:

*запрос1* UNION *запрос2* UNION *запрос3*

Такие сложные запросы выполняются следующим образом:

(*запрос1* UNION *запрос2*) UNION *запрос3*