

# Функции и операторы

- [Оператор DISTINCT](#)

# Оператор DISTINCT

PostgreSQL оператор `DISTINCT` используется для удаления дубликатов из набора результатов. `DISTINCT` может использоваться только с операторами `SELECT`.

## Синтаксис

Синтаксис для оператора DISTINCT в PostgreSQL:

```
SELECT DISTINCT | DISTINCT ON (distinct_expressions) expressions FROM tables [WHERE conditions];
```

### Параметры и аргументы

- `distinct_expressions` - выражения, используемые для удаления дубликатов.
- `expressions` - столбцы или вычисления, которые вы хотите получить.
- `tables` - таблицы, из которых вы хотите получить записи. В операторе `FROM` должна быть указана хотя бы одна таблица.
- `WHERE conditions` - необязательный. Условия, которые должны быть выполнены для записей, которые будут выбраны.

## Примечание

- Если в `DISTINCT` указано только одно выражение, запрос возвратит уникальные значения для этого выражения.
- Если в `DISTINCT` указано несколько выражений, запрос извлекает уникальные комбинации для перечисленных выражений.

- Если заданы ключевые слова `DISTINCT ON` , запрос возвратит уникальные значения для `distinct_expressions` и вернет другие поля для выбранных записей на основе предложения `ORDER BY (limit 1)` .
- В PostgreSQL `DISTINCT` не игнорирует значения `NULL` . Поэтому при использовании `DISTINCT` в вашем операторе `SQL` ваш результирующий набор будет содержать значение `NULL` как отдельное значение.

## Пример с одним выражением

Рассмотрим на простейший пример **`DISTINCT`** в PostgreSQL. Мы можем использовать оператор `DISTINCT` , чтобы вернуть одно поле, которое удаляет дубликаты из набора результатов.

Например:

```
SELECT DISTINCT last_name  
FROM contacts  
ORDER BY last_name;
```

В этом PostgreSQL примере **`DISTINCT`** будут возвращены все уникальные значения `last_name` из таблицы `contacts`.

## Пример с несколькими выражениями

Давайте посмотрим, как вы можете использовать оператор PostgreSQL `DISTINCT` для удаления дубликатов из более чем одного поля в вашем операторе `SELECT` .

Например:

```
SELECT DISTINCT last_name, city, state
FROM contacts
ORDER BY last_name, city, state;
```

Этот пример будет возвращать каждую уникальную комбинацию `last_name`, `city` и `state` из таблицы `contacts`. В этом случае `DISTINCT` применяется к каждому полю, указанному после ключевого слова `DISTINCT`, и, следовательно, возвращает различные комбинации.

## Пример DISTINCT ON

Одна вещь, которая уникальна в PostgreSQL, по сравнению с другими базами данных, заключается в том, что у вас есть еще одна опция при использовании оператора `DISTINCT`, которая называется `DISTINCT ON`. `DISTINCT ON` вернет только первую строку для `DISTINCT ON (diver_expressions)` на основе оператора `ORDER BY`, предоставленного в запросе. Любые другие поля, перечисленные в операторе `SELECT`, будут возвращены для этой первой строки. Это похоже на выполнение `LIMIT` в 1 для каждой комбинации `DISTINCT ON (different_expressions)`.

Давайте подробнее рассмотрим, как использовать `DISTINCT ON` в операторе `DISTINCT` и что он возвращает.

Таким образом, мы могли бы изменить пример выше:

```
SELECT DISTINCT ON (last_name) last_name, city, state
FROM contacts
ORDER BY last_name, city, state;
```

Этот пример `DISTINCT`, в котором используются ключевые слова `DISTINCT ON`, вернет все уникальные значения `last_name`. Но в этом случае для каждого уникального значения `last_name` он будет возвращать только первую уникальную запись `last_name`, с которой он сталкивается, на основе оператора `ORDER BY` вместе с `city` и `state` значениями из этой записи. Он не возвращает уникальные комбинации `last_name`, `city` и `state`. По сути, он выполняет `LIMIT`, равный 1, для каждого `DISTINCT ON (last_name)` и возвращает соответствующие значения `city` и `state` после того, как он выбрал

возвращенные записи.

# Список используемых ИСТОЧНИКОВ

- [PostgreSQL оператор DISTINCT](#)