**Тема занятия №5: Агрегирующие функции. Сортировка данных. Приведение типов**

**Основные агрегирующие функции:**

**- SUM**: находит сумму значений

**- MIN**: находит наименьшее значение

**- MAX**: находит наибольшее значение

**- AVG**: находит среднее значение

**- COUNT(\*)**: находит количество строк в запросе

**SELECT -** агрегирующая функция (*column\_name*)

**FROM** - *table\_name*

**WHERE** - condition

**Операторы для получения диапазона строк. LIMIT и OFFSET**

Оператор **LIMIT** позволяет извлечь определенное количество строк.

Оператор **OFFSET** позволяет указать, с какой строки надо начинать выборку.

SELECT \*

FROM *table\_name*

ORDER BY *table\_field*

LIMIT количество строк для ограничения;

OFFSET c какой строки начать выборку

Оператор **DISTINCT** позволяет выбрать уникальные данные по определенным столбцам.

Оператор **DISTINCT** нашел широкое применение в операторе SQL SELECT, для выборки уникальных значений и так же используется в агрегатных функциях.

SELECT **DISTINCT** *column\_name*

FROM *table\_name*

**Основные операторы фильтрации –** **IN, BETWEEN, LIKE**

Оператор **IN** позволяет определить набор значений, которые должны иметь столбцы.

Оператор **BETWEEN** определяет диапазон значений с помощью начального и конечного значения, которому должно соответствовать выражение.

Оператор **LIKE** принимает шаблон строки, которому должно соответствовать выражение.

Оператор **ORDER BY** позволяет отсортировать значения по определенному столбцу.

Выражение CASE в SQL представляет собой общее условное выражение, напоминающее операторы if/else в других языках программирования:

В первом подходе скобки функции CAST PostgreSQL используются для определения значения и желаемого типа данных

**SELECT CAST ( VALUE AS TYPE )**

Добавление нотации :: между значением и желаемым типом данных — еще один подход к приведению значения. Синтаксис показан ниже:

**SELECT VALUE::TYPE**