Урок 5.1. Агрегирующие функции Оператор Group BY

Агрегирующие функции Оператор Group BY

Важно!

База данных с доступом на чтение:

hostname: ich-db.edu.itcareerhub.de

username: ich1

password: password

Агрегация — это процесс объединения данных, чтобы получить общее представление о наборе данных.

В контексте баз данных агрегация означает выполнение таких операций, как подсчет количества записей, вычисление суммы, среднего значения, нахождение максимума или минимума для группы данных.

Зачем нужна агрегация

Агрегация помогает нам обобщить большие объемы данных и извлечь из них полезную информацию.

Например, если у вас есть таблица с продажами за год, вы можете с помощью агрегации узнать:

- Общее количество продаж за год
- Среднюю сумму продажи
- Максимальную и минимальную цену проданного товара

Эти данные позволяют лучше понять тенденции, принять обоснованные решения и сделать выводы, не погружаясь в детали каждой отдельной записи.

Что такое агрегирующие функции

Агрегирующие функции — это функции, которые позволяют производить вычисления над набором строк, возвращая одно итоговое значение.

Эти функции часто используются для подведения итогов, создания отчетов и аналитики данных в базе данных.

IT CAREER HUB

Основные агрегирующие функции

- **COUNT()** — Подсчитывает количество строк в наборе данных.
  - **Пример:** COUNT(*) возвращает общее количество строк в таблице.
  - **Особенности:** Можно использовать COUNT(column_name) для подсчета только тех строк, где column_name не NULL.
- **SUM()** — Вычисляет сумму значений в столбце.
  - **Пример:** SUM(price) возвращает общую сумму значений столбца price.
  - **Особенности:** Работает только с числовыми типами данных.
- **AVG()** — Вычисляет среднее значение по столбцу.
  - **Пример:** AVG(price) возвращает среднее значение в столбце price.
  - **Особенности:** Игнорирует NULL значения при вычислении среднего.
- **MIN()** — Определяет минимальное значение в столбце.
  - **Пример:** MIN(price) возвращает минимальное значение в столбце price.
  - **Особенности:** Работает с числовыми, строковыми и датами.
- **MAX()** — Определяет максимальное значение в столбце.
  - **Пример:** MAX(price) возвращает максимальное значение в столбце price.
  - **Особенности:** Работает с числовыми, строковыми и датами.

GROUP_CONCAT — это специфическая агрегатная функция в MySQL, которая объединяет значения из нескольких строк в одну строку с разделителем.

Эта функция особенно полезна, когда нужно вывести данные в виде строки, например, список значений для определенной группы.

- Агрегатные функции игнорируют NULL значения, кроме COUNT(column_name), которая учитывает только те строки, где column_name не NULL.
- Можно использовать **DISTINCT** внутри агрегатных функций, чтобы учитывать только уникальные значения.

```
Unset
SELECT COUNT(DISTINCT customer_id) AS unique_customers

FROM orders;
```

В этом примере COUNT(DISTINCT customer_id) подсчитывает количество уникальных клиентов в таблице orders.

IT CAREER HUB

Задание для закрепления

1. Найдите общее количество товаров quantity в таблице order_details.

```
Unset

SELECT SUM(quantity) FROM order_details
```

2. Посчитайте количество уникальных order_id в таблице order_details.

```
Unset

SELECT COUNT(DISTINCT order_id) FROM order_details
```

3. Перечислите через запятую имена всех сотрудников из таблицы employees.

```
Unset
SELECT GROUP_CONCAT(first_name) FROM employees
```

4. Выведите среднее, минимум и макисиму столбца unit_price таблицы order_details.

```
Unset
SELECT AVG(unit_price), MIN(unit_price), MAX(unit_price)

FROM order_details
```

IT CAREER HUB

Оператор GROUP BY

GROUP BY — это оператор SQL, который используется для группировки строк, имеющих одинаковые значения в одном или нескольких столбцах.

В каждой группе данные можно агрегировать с помощью агрегирующих функций.

Когда используется GROUP BY:

- SQL Server группирует строки по указанным столбцам.
- Все строки с одинаковыми значениями в этих столбцах объединяются в одну группу.
- Агрегирующая функция применяется к каждой группе отдельно.

```
Unset

SELECT column_name, AGGREGATE_FUNCTION(column_name)

FROM table_name

GROUP BY column_name;
```

- **column_name** — столбец, по которому производится группировка.
- **AGGREGATE_FUNCTION(column_name)** — агрегирующая функция, применяемая к каждой группе.

Все столбцы в SELECT, которые не используются в агрегирующих функциях, должны быть перечислены в GROUP BY.

Если вы выбираете столбец, который не является частью агрегирующей функции, этот столбец должен быть указан в GROUP BY. В противном случае запрос не выполнится или вернет непредсказуемые результаты.

IT CAREER HUB

Задание для закрепления

Из таблицы employees посчитать количество сотрудников в каждом городе city.

Отсортировать результаты по убыванию.

```
Unset

SELECT city, COUNT(id) as number_employees

FROM employees

GROUP BY 1

ORDER BY 2 DESC
```

Посчитать общее количество продуктов из таблицы order_details для каждого заказа.

Отсортировать по убыванию общего количества продуктов. Для краткости записи в GROUP BY можно не указывать конкретное имя колонки, а указать ее порядковый номер в SELECT.

Сделать то же самое в ORDER BY.

```
Unset

SELECT order_id, SUM(quantity) total_quantity

FROM order_details

GROUP BY 1

ORDER BY 2 DESC
```

Посчитать сколько сотрудников работает в каждой компании из таблицы customers. Учитывать только тех сотрудников, у которых job_id равен 'Purchasing Manager'.

```
Unset

SELECT company, COUNT(id) as number_customers

FROM customers

WHERE job_title = 'Purchasing Manager'
```

Если столбец, который вы хотите использовать для группировки содержит только уникальные неповторяющиеся значения, то в группировке нет смысла - любая агрегатная функция даст один и тот же результат. Попробуйте сгруппировать любую таблицу по первичному ключу и применить агрегатные функции к столбцам.

Посчитать количество сотрудников в разрезе компании и занимаемой должности из таблицы employees.

```
Unset

SELECT Company, job_title, COUNT(id)

FROM employees

GROUP BY company, job_title;
```