**SQL: агрегирующие функции.**

Некоторые запросы в SQL требуют подсчитать количество элементов или максимальное значение среди элементов и т. д. Для этого существуют агрегирующие функции. Перечислим основные из них:

* **AVG**: среднее значение;
* **MIN**: наименьшее значение;
* **MAX**: наибольшее значение;
* **COUNT**: количество строк;
* **SUM**: сумма значений.

|  |  |
| --- | --- |
| 1 | **SELECT** **AVG**(столбец) **FROM** название\_таблицы; |
| 1 | **SELECT MIN(**столбец**) FROM** название\_таблицы**;** |

**Примеры:**

1. Количество товаров, цена которых выше 150:

|  |  |
| --- | --- |
| 1  2  3 | **SELECT** **COUNT**(\*)  **FROM** product  **WHERE** quantity > 50; |



1. Вывести товар с наименьшей ценой и его название:

|  |  |
| --- | --- |
| 1  2 | **SELECT** **MIN**(price), name  **FROM** product; |



1. Вывести максимальное количество и название товара, цена которого больше 70:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3 | **SELECT** **MAX**(quantity), name  **FROM** product  **WHERE** price > 270; | |



1. Вывести среднюю цену товара «Молочный шоколад»:

|  |  |
| --- | --- |
| 1  2  3 | **SELECT** **AVG**(price)  **FROM** product  **WHERE** name = 'Молочный шоколад'; |



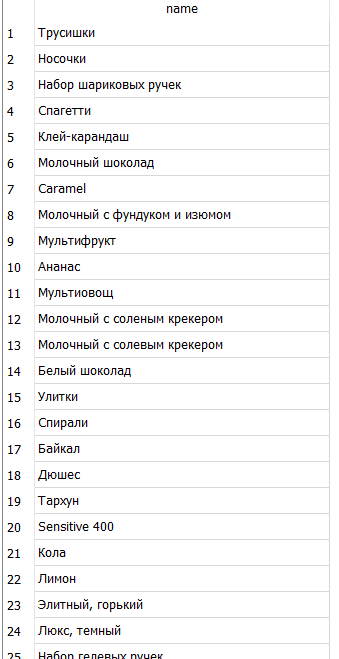
**SQL: сортировка.**

Для упорядочения данных следует использовать сортировку. Для этого в конце запроса используется оператор **ORDER BY**. Сортировать можно как по возрастанию, так и по убыванию. Сортировать можно по нескольким столбцам.

**Примеры:**

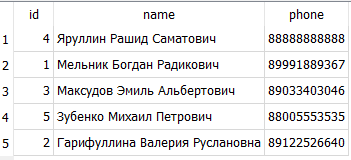
1. Вывести без повторений названия товаров, сортируя по цене (по возрастанию):

|  |  |
| --- | --- |
| 1  2  3 | **SELECT** **DISTINCT**(Name)  **FROM** Product  **ORDER** **BY** Price; |



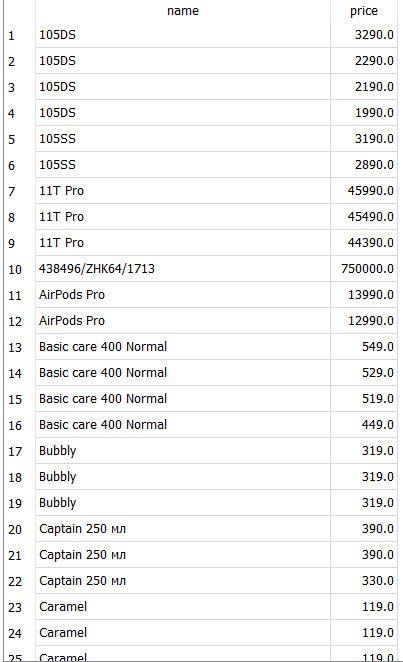
1. Вывести таблицу "Покупатели", сортируя имена по убыванию:

|  |  |
| --- | --- |
| 1  2  3 | **SELECT** \*  **FROM** buyer  **ORDER** **BY** name **DESC**; |



1. Вывести список товаров, отсортированных сначала по названию (от А до Я), а потом по цене (по убыванию):

|  |  |
| --- | --- |
| 1  2  3 | **SELECT** name, price  **FROM** product  **ORDER** **BY** name **ASC**, price **DESC**; |



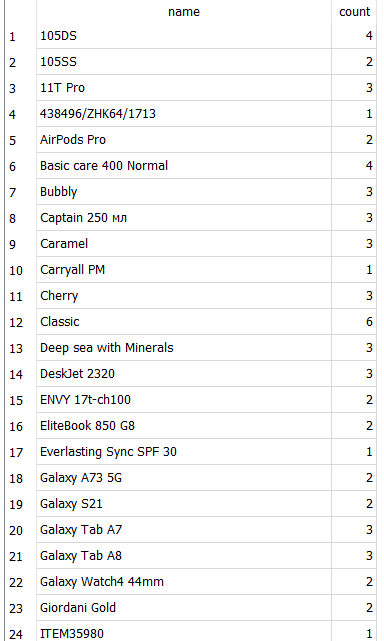
**SQL: группировка.**

Оператор **GROUP BY** используется для группировки строк. Этот оператор возвращает одну строку для каждой группы, для которых потом можно выполнить агрегирующиефункции.

**Пример:**

1. Вывести количество товара по наименованию:

|  |  |
| --- | --- |
| 1  2  3 | **SELECT** Name, **COUNT**(\*) **AS** **count**  **FROM** Product  **GROUP** **BY** Name; |

****

### Для фильтрации групп используют оператор HAVING.

### Пример:

1. Вывести наименования товаров, количество которых превышает 3 по убыванию:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **SELECT** name, **COUNT**(\*) **AS** **count**  **FROM** product  **GROUP** **BY** name  **HAVING** **COUNT**(\*) > 3  **ORDER** **BY** **count** **DESC**; |

****

**SQL: подзапросы.**

Подзапросы в SQL - это запросы, которые вложены в другие запросы. Они используются для выполнения более сложных операций, комбинирования данных из разных таблиц или фильтрации результатов. Подзапросы могут быть использованы в различных частях запроса (SELECT, FROM, WHERE, HAVING и других). Они позволяют выполнять операции с меньшим набором данных, а затем использовать эти результаты в основном запросе.

Подзапросы бывают **коррелирующими** и **некоррелирующими**. Результат коррелирующего запроса зависит от строк, которые выбираются в основном запросе, в то время как у некоррелирующего – не зависит.

**Примеры:**

1. Некоррелирующий подзапрос:

Вывести товары, цена которых превышает среднюю цену, но не превышают 150000.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **SELECT** name, price  **FROM** Product  **WHERE** price > (  **SELECT** **AVG**(price)  **FROM** Product)  **AND** price < 150000; |



1. Коррелирующий подзапрос:

Вывести наименование товара, его производителя и цену, склад, в котором он находится и среднюю цену по этому складу. Условие: цена товара должна быть больше средней цены по складу, в котором он находится. Результаты запроса отсортировать сначала по средней цене склада (в порядке убывания), а потом по цене товара (в порядке возрастания).

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **SELECT** name,  (**SELECT** name  **FROM** producer  **WHERE** producer.id = product.producer\_id) **AS** producer\_name,  price,  (**SELECT** (city || ', ' || adress) *-- слияние столбцов в один*  **FROM** store  **WHERE** store.id = product.store\_id) **AS** store\_location,  (**SELECT** ROUND(**AVG**(price),2) *-- округление до сотых*  **FROM** product **AS** s\_product  **WHERE** s\_product.store\_id = product.store\_id) **AS** avg\_price  **FROM** product  **WHERE** price > (  **SELECT** **AVG**(price)  **FROM** product **AS** s\_product  **WHERE** s\_product.store\_id = product.store\_id)  **AND** category\_id = (  **SELECT** id  **FROM** category  **WHERE** name = 'Декор и интерьер')  **ORDER** **BY** avg\_price **DESC**, price **ASC**; |



1. Подзапросы в INSERT:

Добавить товар на склад по адресу г. Казань, ул. Пушкина, д. Колотушкина.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | **INSERT** **INTO** [product] (store\_id, producer\_id, category\_id,  name, availability, date, quantity, price)  **VALUES** ((**SELECT** id  **FROM** store  **WHERE** city = 'Казань'  **AND** adress = 'ул. Пушкина, д. Колотушкина'),  (**SELECT** id  **FROM** producer  **WHERE** name = 'Жумайсынба'),  (**SELECT** id  **FROM** category  **WHERE** name = 'Продукты питания'),  'Вода Бомжоми', 1, '22.08.2023', 54, 333); |

1. Подзапросы в UPDATE:

Повысить цены на все товары на складах Ижевска на 10.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **UPDATE** [product]  **SET** price = price + 10  **WHERE** store\_id = (  **SELECT** id  **FROM** store  **WHERE** city = 'Ижевск'); |

1. Подзапросы в DELETE:

Удалить все товары из магазина по адресу г. Казань, ул. Кремлевская, д. 35.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **DELETE** **FROM** product  **WHERE** store\_id = (  **SELECT** id  **FROM** store  **WHERE** city = 'Казань'  **AND** adress = 'ул. Кремлевская, д. 35'); |