1. Какая команда позволяет сделать выборку данных из таблицы?   
   Что нужно указать для этой команды для выборки значений конкретных столбцов из конкретной таблицы?

Команда SELECT позволяет сделать выборку данных из таблицы.

Для выборки значений конкретных столбцов из конкретной таблицы, необходимо указать название столбца или столбцов после ключевого слова SELECT, разделенных запятыми. Также нужно указать имя таблицы после ключевого слова FROM. Пример:

SELECT column1, column2 FROM table1;

1. Как выбрать из таблицы только строки, соответствующие определённому условию?

Для выбора только строк, которые соответствуют определенному условию, можно использовать оператор WHERE в запросе на выборку данных из таблицы.

Синтаксис оператора WHERE: SELECT \* FROM таблица WHERE условие;

На месте "таблица" нужно указать имя нужной таблицы, а на месте "условие" нужно указать логическое условие, которое должно быть выполнено для отбора строк. Условие может содержать сравнения, логические операторы и функции.

1. Как отсортировать результат выборки?

Результат выборки можно отсортировать по одному или нескольким столбцам с помощью команды ORDER BY в SQL.

Пример сортировки по одному столбцу:

SELECT column1, column2

FROM table

ORDER BY column1 ASC;

В данном примере, результат выборки будут отсортированы по возрастанию значения в столбце column1. Для сортировки по убыванию используется ключевое слово DESC.

Пример сортировки по нескольким столбцам:

SELECT column1, column2

FROM table

ORDER BY column1 ASC, column2 DESC;

В данном примере, результат выборки будет отсортирован по возрастанию столбца column1. При равных значениях в столбце column1, результат будет сортироваться по убыванию столбца column2.

Также можно использовать номера столбцов вместо имен при указании порядка сортировки, например: SELECT column1, column2

FROM table

ORDER BY 1 ASC, 2 DESC;

В данном примере, результат выборки будет отсортирован по возрастанию первого столбца (column1) и по убыванию второго столбца (column2).

4 Что такое агрегирующая функция?

Агрегирующая функция в SQL – это функция, которая применяется для выполнения вычислений над множеством значений столбца и возвращает одно суммарное значение. Эти функции позволяют агрегировать данные и выполнять операции, такие как подсчет количества строк, нахождение среднего значения, определение минимального или максимального значения и т.д. Некоторые примеры таких функций в SQL включают функции SUM(), COUNT(), AVG(), MIN() и MAX().

5 Как распределить результаты выборки по группам?

В SQL есть несколько способов распределить результаты выборки по группам. Вот несколько примеров:

1. С использованием оператора GROUP BY:Этот запрос позволяет сгруппировать строки из таблицы по значению в колонке column1 и посчитать количество строк в каждой группе.
2. С использованием оператора CASE:В этом примере мы создаем различные столбцы (group1\_count, group2\_count, group3\_count) для каждой группы результатов, определяемой значением в колонке column2.
3. С использованием функций агрегации:

В этом случае мы используем функции агрегации (SUM, AVG, MAX) для вычисления суммы, среднего значения и максимального значения определенной колонки для каждой группы. Конкретный способ распределения результатов выборки по группам зависит от требований и целей вашей задачи. Выберите подходящий метод в зависимости от того, какие данные и результаты вам нужны.

1. Как подсчитать количество объектов в каждой группе?

Подсчитать количество объектов в группе можно при помощи агрегирующей функции COUNT, которая подсчитывает количество значений в столбце.

1. Как выполнить фильтрацию по группам объектов?

Для фильтрации и сортировки групп можно применять команды WHERE, HAVING.

SELECTстолбцы  
FROM название таблицы  
GROUPBYназвание столбца  
HAVING условие /WHERE условие

1. В каком порядке выполняются выражения оператора SELECT?

Очередность выполнения выражений оператора SELECT:

* 1. FROM (выбор таблицы)
  2. JOIN (комбинация с подходящими по условию данными из других таблиц)
  3. WHERE (фильтрация строк)
  4. GROUP BY (группировка данных)
  5. HAVING (фильтрация данных)
  6. SELECT (возврат результата и сокрытие ненужных столбцов)
  7. ORDER BY (сортировка).