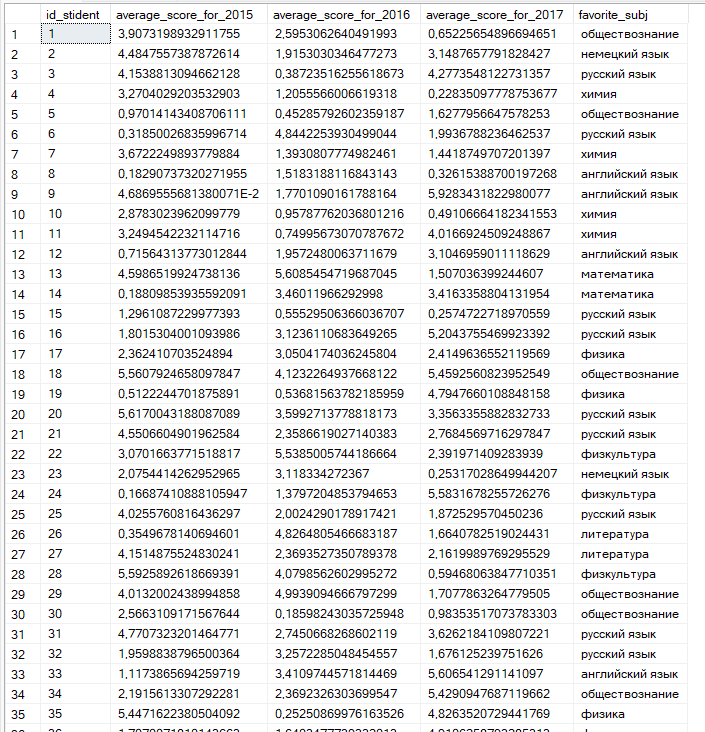
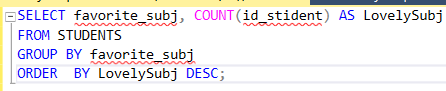
**Лабораторная работа № 8**

**Выборка данных. Группировка и сортировка данных**

2 вариант:



1. Запрос, который покажет, сколько человек выбрало тот или иной предмет для первого экзамена, сделайте сортировку по убыванию.



SELECT favorite\_subj, COUNT(id\_stident) AS LovelySubj

FROM STUDENTS

GROUP BY favorite\_subj

ORDER BY LovelySubj DESC;



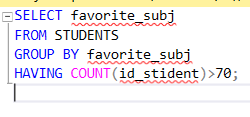
1. Запрос, который покажет, какие предметы в качестве любимого выбрало более 70 человек.

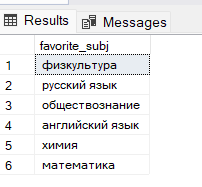
SELECT favorite\_subj

FROM STUDENTS

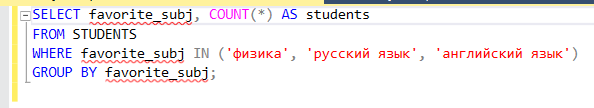
GROUP BY favorite\_subj

HAVING COUNT(id\_stident)>70;





1. Запрос, который покажет сколько студентов выбрало в качестве любимого физику, русский или английский. Создайте два запроса один с помощью WHERE, другой используя HAVING.

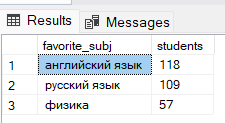


SELECT favorite\_subj, COUNT(\*) AS students

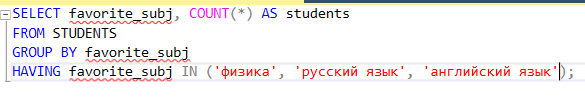
FROM STUDENTS

WHERE favorite\_subj IN ('физика', 'русский язык', 'английский язык')

GROUP BY favorite\_subj;



Having:

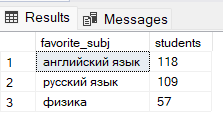


SELECT favorite\_subj, COUNT(\*) AS students

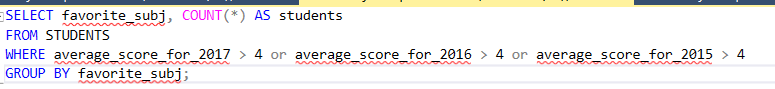
FROM STUDENTS

GROUP BY favorite\_subj

HAVING favorite\_subj IN ('физика', 'русский язык', 'английский язык');



1. Запрос, который покажет, сколько студентов выбрало тот или иной предмет в качестве любимого и имеющих средний балл хотя бы за один год более 4.

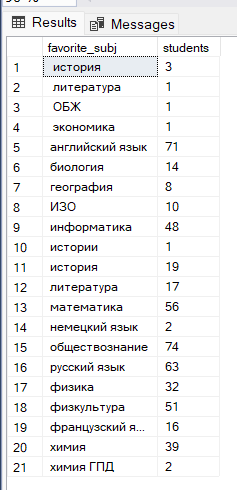


SELECT favorite\_subj, COUNT(\*) AS students

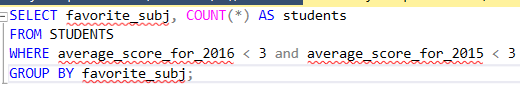
FROM STUDENTS

WHERE average\_score\_for\_2017 > 4 or average\_score\_for\_2016 > 4 or average\_score\_for\_2015 > 4

GROUP BY favorite\_subj;



1. Запрос, который покажет сколько студентов имею средний бал за 2015 или 2016 год менее 3.



SELECT favorite\_subj, COUNT(\*) AS students

FROM STUDENTS

WHERE average\_score\_for\_2016 < 3 and average\_score\_for\_2015 < 3

GROUP BY favorite\_subj;



Контрольные вопросы:

1. Что такое агрегатная функция?

Агрегатная функция выполняет вычисление на наборе значений и возвращает одиночное значение. Агрегатные функции, за исключением COUNT(\*) , не учитывают значения NULL. Агрегатные функции часто используются в выражении GROUP BY инструкции SELECT. Все агрегатные функции являются детерминированными.

1. Зачем нужно самосоединение?

Самосоединение используется, когда необходимо объединить таблицу с самой собой. Оно может быть полезно, когда требуется связать строки в таблице на основе условия, но информация, необходимая для связи, содержится в той же таблице.

1. Какие типы соединений вы знаете?

INNER JOIN: Возвращает только совпадающие строки из обоих таблиц.

Левое внешнее соединение.

LEFT JOIN (или LEFT OUTER JOIN): Возвращает все строки из левой таблицы и совпадающие строки из правой таблицы.

Правое внешнее соединение

RIGHT JOIN (или RIGHT OUTER JOIN): Возвращает все строки из правой таблицы и совпадающие строки из левой таблицы.

Полное внешнее соединение

FULL JOIN (или FULL OUTER JOIN): Возвращает все строки из обеих таблиц и совпадающие и непересекающиеся строки.

Полное внешнее соединение

CROSS JOIN: Возвращает декартово произведение строк из обеих таблиц.

1. Что такое группировка? Приведите пример использования.

Группировка - это процесс объединения рядов данных на основе значения одного или нескольких атрибутов, чтобы сделать сводные вычисления по этим группам. Пример использования группировки:

SELECT поставщик, COUNT(\*) AS количество\_товаров

FROM ТОВАРЫ

GROUP BY поставщик;

1. Как вывести всю строку, содержащую максимальное значение по тому или иному атрибуту?

Для вывода всей строки, содержащей максимальное значение по определенному атрибуту можно использовать подзапрос и оператор IN. Например:

SELECT \*

FROM ТОВАРЫ

WHERE артикул\_товара IN (SELECT артикул\_товара FROM ТОВАРЫ ORDER BY цена DESC LIMIT 1);

1. В чем разница между операторами WHERE и HAVING?

Оператор WHERE используется для фильтрации строк до их группировки, тогда как оператор HAVING применяется к группированным результатам. Оператор HAVING позволяет фильтровать результаты на основе агрегатных функций или групповых атрибутов. Таким образом, оператор HAVING может использоваться только после оператора GROUP BY.

1. В чем разница между функциями COUNT и SUM?

Разница между функциями COUNT и SUM:

COUNT возвращает количество строк в результате запроса или количество значений в заданной колонке (если указано имя колонки).

SUM возвращает сумму значений в заданной колонке (если указано имя колонки).

1. Можно ли отсортировать данные таблицы по нескольким атрибутам? Если нет, то почему, если да, то какой будет результат?

Да, можно отсортировать данные таблицы по нескольким атрибутам. Порядок сортировки атрибутов определяется последовательностью указания этих атрибутов в операторе ORDER BY. Результат будет отсортирован сначала по первому указанному атрибуту, а затем по следующим атрибутам в порядке указания.

1. Можно ли одновременно использовать в одном запросе операторы WHERE и HAVING?

Да, можно одновременно использовать в одном запросе операторы WHERE и HAVING. Оператор WHERE используется для фильтрации строк до группировки, а оператор HAVING применяется к группированным результатам. Оба оператора могут использоваться в одном запросе для уточнения условий поиска и фильтрации.

1. В какой последовательности происходит сортировка по умолчанию?

Порядок сортировки по умолчанию отсутствует. Если не указан оператор ORDER BY, то порядок записей в результирующем наборе может быть неопределенным и зависит от реализации СУБД.

1. Как можно отсортировать уже сгруппированные записи?

Сгруппированные записи можно отсортировать с помощью оператора ORDER BY, в котором указываются атрибуты для сортировки. Например:

SELECT поставщик, COUNT(\*) AS количество\_товаров

FROM ТОВАРЫ

GROUP BY поставщик

ORDER BY количество\_товаров DESC;