

# MySQL

## Занятие 3

# ORDER BY

Выбрать все данные из таблицы **SUBJECT**, упорядочив выдачу по столбцу **SUBJ\_NAME**:

```
SELECT *  
    FROM SUBJECT  
    ORDER BY SUBJ_NAME; - по умолчанию по возрастанию (ASC)
```

То же самое, но в обратном порядке:

```
SELECT *  
    FROM SUBJECT  
    ORDER BY SUBJ_NAME DESC;
```

# ORDER BY


Выбрать все данные из таблицы **SUBJECT**, упорядочив выдачу по значениям семестров (**SEMESTER**), а внутри семестров по наименованиям предметов (**SUBJ\_NAME**):

```
SELECT *  
FROM SUBJECT  
ORDER BY SEMESTER, SUBJ_NAME;
```

Указывать столбец можно по номеру – определяются выходными данными запроса:

```
SELECT SUBJ_ID, SEMESTER  
FROM SUBJECT  
ORDER BY 2 DESC;
```

Теперь это 2ой столбец (!)



NULL значения либо все в конце, либо все в начале выходной таблицы

# ORDER BY + GROUP BY

Упорядочиваем группы, ORDER BY ставим в конец (!) :

```
SELECT SUBJ_NAME, SEMESTER, MAX(HOUR)
FROM SUBJECT
GROUP BY SEMESTER, SUBJ_NAME
ORDER BY SEMESTER;
```

# Задание 1

Напишите запрос, который выполняет вывод суммы баллов для каждой даты сдачи экзаменов и представляет результаты в порядке убывания этих сумм (таблица EXAM\_MARKS)

## Задание 2

Напишите запрос, который выполняет вывод среднего, минимального и максимального значений баллов всех студентов для каждой даты сдачи экзаменов

# Вложенные подзапросы

В таблице STUDENT есть информация о студенте Акимове.

Нужно узнать оценки этого студента из таблицы EXAM\_MARKS по STUDENT\_ID:

```
SELECT *  
  FROM EXAM_MARKS  
 WHERE STUDENT_ID =  
      (SELECT STUDENT_ID FROM STUDENT WHERE SURNAME = 'Акимов');
```

1. Читаем первую строку таблицы из внешнего запроса
2. Выполняем подзапрос.
3. Значение, полученное в результате выполнения подзапроса, применяем для анализа строки из п.1 в условии WHERE внешнего запроса
4. Выполняем п.1 для следующей строки таблицы из внешнего запроса
5. Подзапрос должен выдавать единственное значение (!) (вспомним DISTINCT)

# Вложенные подзапросы + IN

Показать оценки для студентов из Воронежа:

```
SELECT *  
FROM EXAM_MARKS  
WHERE STUDENT_ID IN  
    (SELECT STUDENT_ID  
     FROM STUDENT  
     WHERE CITY = 'Воронеж');
```



# Вложенные подзапросы + HAVING

Определить количество предметов обучения с оценкой, превышающей среднее значение оценки студента с идентификатором 301:

```
SELECT COUNT(DISTINCT SUBJ_ID), MARK  
FROM EXAM_MARKS  
GROUP BY MARK  
HAVING MARK >  
      (SELECT AVG(MARK)  
      FROM EXAM_MARKS  
      WHERE STUDENT_ID = 301);
```

# Вложенные подзапросы + HAVING

Разберем по частям:

Группируем количества предметов по оценкам:

```
SELECT COUNT(DISTINCT SUBJ_ID), MARK  
FROM exam_marks  
GROUP BY MARK;
```

Вычисляем порог, по которому будем отбирать группы:

```
SELECT AVG(MARK)  
FROM EXAM_MARKS  
WHERE STUDENT_ID = 301;
```

## Задание 3

Напишите запрос, который выполняет выборку всех имен студентов, имеющих по предмету с идентификатором 102 балл ниже общего среднего балла.

## Задание 4

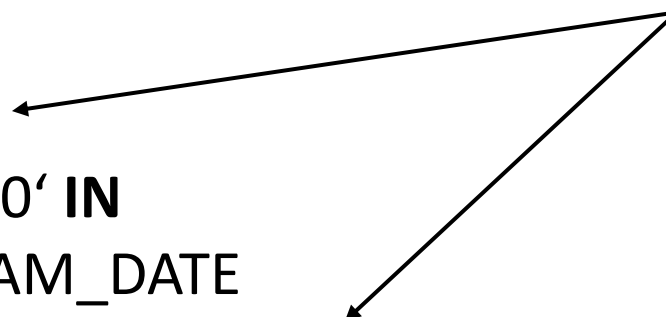
Напишите запрос, который выполняет вывод количества предметов, по которым экзаменовался каждый студент, сдававший более 20 предметов.

# Связные подзапросы

Выбрать сведения обо всех предметах обучения, по которым проводился экзамен 20 марта 2000 года:

```
SELECT *  
FROM SUBJECT SU  
WHERE '2000-03-20' IN  
  (SELECT EXAM_DATE  
   FROM EXAM_MARKS EX  
   WHERE SU.SUBJ_ID = EX.SUBJ_ID);
```

Псевдонимы / алиасы



# Связные подзапросы

Найти идентификаторы, фамилии и стипендии студентов, получающих стипендию выше средней на курсе, на котором они учатся

```
SELECT DISTINCT STUDENT_ID, SURNAME, STIPEND  
FROM STUDENT E1  
WHERE STIPEND >  
      (SELECT AVG(STIPEND)  
       FROM STUDENT E2  
       WHERE E1.KURS = E2.KURS);
```

AVG выполняется над таблицей из подзапроса для каждой строки из таблицы внешнего запроса.

# Связные подзапросы

Второй способ:

```
SELECT DISTINCT STUDENT_ID, SURNAME, STIPEND  
  FROM STUDENT E1,  
        (SELECT KURS, AVG(STIPEND) AS AVG_STIPEND  
         FROM STUDENT E2  
        GROUP BY E2.KURS) E3  
 WHERE E1.STIPEND > AVG_STIPEND  
 AND E1.KURS=E3.KURS;
```

Считаем AVG один раз для таблицы E2, формируя вспомогательную таблицу E3, строками которой соединяются строки первой таблицы E1.

## Задание 5

Напишите запрос, использующий связанные подзапросы и выполняющий вывод имен и идентификаторов студентов, у которых стипендия совпадает с максимальным значением стипендии для города, в котором живет студент



# Связные подзапросы + HAVING

По данным из таблицы EXAM\_MARKS определить сумму полученных студентами оценок, сгруппировав значения оценок по датам экзаменов и исключив те дни, когда число студентов, сдававших экзамен, было меньше 10

```
SELECT EXAM_DATE, SUM(MARK)
FROM EXAM_MARKS A
GROUP BY EXAM_DATE
HAVING 10 <
    (SELECT COUNT(MARK)
     FROM EXAM_MARKS B
     WHERE A.EXAM_DATE = B.EXAM_DATE);
```

## Задание 6

Напишите 2 запроса, которые позволяют вывести имена и идентификаторы всех студентов, для которых точно известно, что они проживают не в том городе, где расположен их университет. Один запрос с использованием соединения, а другой – с использованием связного подзапроса.