

SQL. Вводный курс

Лекция №5: Группировка, вложенные запросы

Упорядочивание, сортировка ORDER BY

Чтобы отсортировать выборку по одному или нескольким полям используется предложение **ORDER BY**. При этом можно задать либо возрастающую последовательность сортировки с использованием ключевого слова **ASC** (по умолчанию), либо убывающую - **DESC**.

```
SELECT <список полей 1>  
  FROM <список таблиц>  
  WHERE <список критериев 1>  
    ORDER BY <список полей 2> [ASC | DESC]
```

Упорядочивание, сортировка ORDER BY

Выбрать всех лекторов упорядоченных по алфавиту по городам в убывающем порядке, а также по фамилиям и именам в возрастающем порядке?

```
SELECT SURNAME, NAME, CITY  
FROM LECTURERS  
ORDER BY CITY DESC, SURNAME DESC, NAME
```

SURNAME	NAME	CITY
Михно	СП	Херсон
Мельник	СФ	Херсон
Телегин	ВИ	Харьков
Сулимова	ИИ	Харьков
Синозацкий	АЮ	Львов
Осипова	ОВ	Львов
Цыганок	ОЮ	Киев

Агрегирование

Группировка

Агрегирующие функции позволяют получать из таблицы сводную (*агрегированную*) информацию, выполняя операции над группой строк таблицы.

Для задания в SELECT-запросе агрегирующих операций используются следующие ключевые слова:

- **COUNT** определяет количество строк или значений поля, выбранных посредством запроса и не являющихся NULL-значениями;
- **SUM** вычисляет арифметическую сумму всех выбранных значений данного поля;
- **AVG** вычисляет среднее значение для всех выбранных значений данного поля;
- **MAX** вычисляет наибольшее из всех выбранных значений данного поля;
- **MIN** вычисляет наименьшее из всех выбранных значений данного поля.

Агрегирование

Группировка

Какое количество студентов учится на 2ом курсе?

```
SELECT COUNT(*) QTY FROM STUDENTS WHERE COURSE=2
```

QTY
5

Какова общая сумма выплат и средняя стипендия в месяц в Киеве?

```
SELECT  
    SUM(STIPEND) AMOUNT,  
    AVG(STIPEND) AVG_AMOUNT  
FROM STUDENTS WHERE CITY='Киев'
```

AMOUNT	AVG_AMOUNT
7100.00	645.454545

Агрегирование

Группировка

Самая ранняя и самая поздняя даты рождения студентов на 1 и 2 курсах?

SELECT

MIN(*BIRTHDAY*) *MIN_BDAY*,

MAX(*BIRTHDAY*) *MAX_BDAY*

FROM STUDENTS WHERE COURSE IN (1, 2)

MIN_BDAY	MAX_BDAY
1990-08-30 00:00:00	1994-04-26 00:00:00

Агрегирование

Группировка

Для разбиения выборки по группам, необходимо указать поля группировки в предложении **GROUP BY**

```
SELECT <список полей 1>  
  FROM <список таблиц>  
  WHERE <список критериев 1>  
    GROUP BY <список полей 2>  
    [HAVING <список критериев 2>]
```

Агрегирование Группировка

Минимальная, максимальная, общая, средняя стипендии на каждом курсе?

```
SELECT COURSE,  
       MIN(STIPEND) MIN_STIPEND, MAX(STIPEND) MAX_STIPEND,  
       SUM(STIPEND) SUM_STIPEND, AVG(STIPEND) AVG_STIPEND  
FROM STUDENTS  
GROUP BY COURSE
```

COURSE	MIN_STIPEND	MAX_STIPEND	SUM_STIPEND	AVG_STIPEND
1	500.00	800.00	3150.00	630.00000
2	500.00	750.00	3300.00	660.00000
3	500.00	650.00	1800.00	600.00000
4	450.00	900.00	2500.00	625.00000
5	500.00	750.00	2600.00	650.00000

Агрегирование

Группировка

В каком городе средняя стипендия превышает 650 грн?

```
SELECT CITY  
FROM STUDENTS  
GROUP BY CITY  
HAVING AVG(STIPEND) > 650
```

CITY
Харьков
Херсон

Вложенные подзапросы

SQL позволяет использовать одни запросы внутри других запросов, то есть вкладывать запросы друг в друга.

Предложение **SELECT** может быть использовано повторно внутри предложений **FROM**, **WHERE** и **HAVING**.

Получить все оценки студента по его фамилии:

```
SELECT * FROM EXAM_MARKS  
WHERE STUDENT_ID = (  
    SELECT ID FROM STUDENTS WHERE SURNAME='Кабанов')
```

ID	STUDENT_ID	SUBJ_ID	MARK	EXAM_DATE
1	1	1	5	2012-06-07
2	1	2	3	2012-06-07

Примечание: результатом подзапроса должно быть единственное

Вложенные подзапросы

В случае возврата множества значений подзапросом используйте оператор **IN**.

Получить оценки всех студентов из Киева:

```
SELECT * FROM EXAM_MARKS  
WHERE STUDENT_ID IN (  
    SELECT ID FROM STUDENTS WHERE CITY='Kиев')
```

ID	STUDENT_ID	SUBJ_ID	MARK	EXAM_DATE
3	2	2	5	2012-06-10
5	4	5	4	2012-06-06
6	4	4	3	2012-06-07
7	4	1	3	2012-06-05

Вложенные запросы

Определить количество предметов обучения с оценкой, превышающей среднее значение оценки студента с ID = 1?

```
SELECT COUNT(DISTINCT SUBJ_ID), MARK FROM EXAM_MARKS
WHERE MARK > (
    SELECT AVG(MARK) FROM EXAM_MARKS
    WHERE STUDENT_ID=1)
GROUP BY MARK
```

COUNT(DISTINCT SUBJ_ID)	MARK
4	5

Коррелирующий подзапрос

При использовании подзапросов во внутреннем запросе можно ссылаться на таблицу, имя которой указано в предложении **FROM** внешнего запроса.

Выбрать преподавателей, которые преподают в том же городе, в котором живут.

```
SELECT * FROM LECTURERS LECT WHERE UNIV_ID IN  
(SELECT U.ID FROM UNIVERSITIES U WHERE U.CITY = LECT.CITY)
```

ID	SURNAME	NAME	CITY	UNIV_ID
1	Кравчук	ЛИ	Днепропетровск	7
2	Совва	СС	Днепропетровск	7
9	Хижна	ЛЛ	Киев	1
11	Вовчок	ДВ	Киев	1