

Расширенные возможности SELECT

Классификация существ. данных

Если 12 в столбце **month** в столбец **name_month** заносится **‘декабрь’**

Если 11 в столбце **month** в столбец **name_month** заносится **‘ноябрь’**

Если 10 в столбце **month** в столбец **name_month** заносится **‘октябрь’**

Выражение **CASE** объединяет множество команд **UPDATE**, проверяя значение существующего столбца по условию. Если условие выполняется, то новый столбец заполняется заданным значением.

UPDATE таблица

SET

CASE

WHEN столбец1=значение1

THEN новое_значение1

WHEN столбец2=значение2

THEN новое_значение2

ELSE значение3

END;

*Если выполняется
это условие ...*

*Этому столбцу
присваивается одно
из перечисленных
ниже значений*

*То столбцу
присваиваетс
я ЭТО
значение*

*Завершает всю
команду*

*Если ни одно из
условий не
выполнится*

```
mysql> alter table student
-> add column name_month varchar(10);
Query OK, 4 rows affected (0.24 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> select * from student;
```

id_stud	last_name	first_name	otch	birth	name_month
1	Иванов	Иван		1999	
2	Соколова	Инга		1999	
3	Иванов	Иван		1999	
4	Волков	Валерий		1999	
5					
6					

```
4 rows in set (0.00 sec)
```

```
mysql> update student
-> set name_month=
-> case
-> when month=1 then 'январь'
-> when month=2 then 'февраль'
-> when month=3 then 'март'
-> when month=4 then 'апрель'
-> when month=5 then 'май'
-> when month=6 then 'июнь'
-> when month=7 then 'июль'
-> when month=8 then 'август'
-> when month=9 then 'сентябрь'
-> when month=10 then 'октябрь'
-> when month=11 then 'ноябрь'
-> when month=12 then 'декабрь'
-> else null
-> end;
Query OK, 4 rows affected (0.09 sec)
Rows matched: 4 Changed: 4 Warnings: 0
```

```
mysql> select * from student;
```

id_stud	last_name	first_name	otch	birth_date	stip	gender	klas
s	month	day	year	name_month			
1	Иванов	Иван	NULL	1999-12-12	1520.20	м	
2	Соколова	Инга	NULL	1995-03-25	1200.15	ж	
3	Иванов	Иван	NULL	1997-05-17	1100.00	м	
4	Волков	Валерий	NULL	1995-06-22	1500.00	м	
5							
6							

```
4 rows in set (0.00 sec)
```

ORDER BY позволяет
отсортировать данные
любого столбца.

Упорядочение результатов выборки

```
SELECT last_name, first_name  
FROM student  
WHERE gender = 'М'  
ORDER BY first_name;
```

```
mysql> select last_name, first_name from student  
-> where gender='М'  
-> order by first_name;  
+-----+-----+  
| last_name | first_name |  
+-----+-----+  
| Волков   | Валерий   |  
| Иванов   | Иван      |  
| Иванов   | Иван      |  
+-----+-----+  
3 rows in set (0.02 sec)
```

Сортировать поле first_name
по возрастанию.

```
SELECT last_name, first_name, stip
```

```
FROM student
```

```
ORDER BY first_name, stip;
```

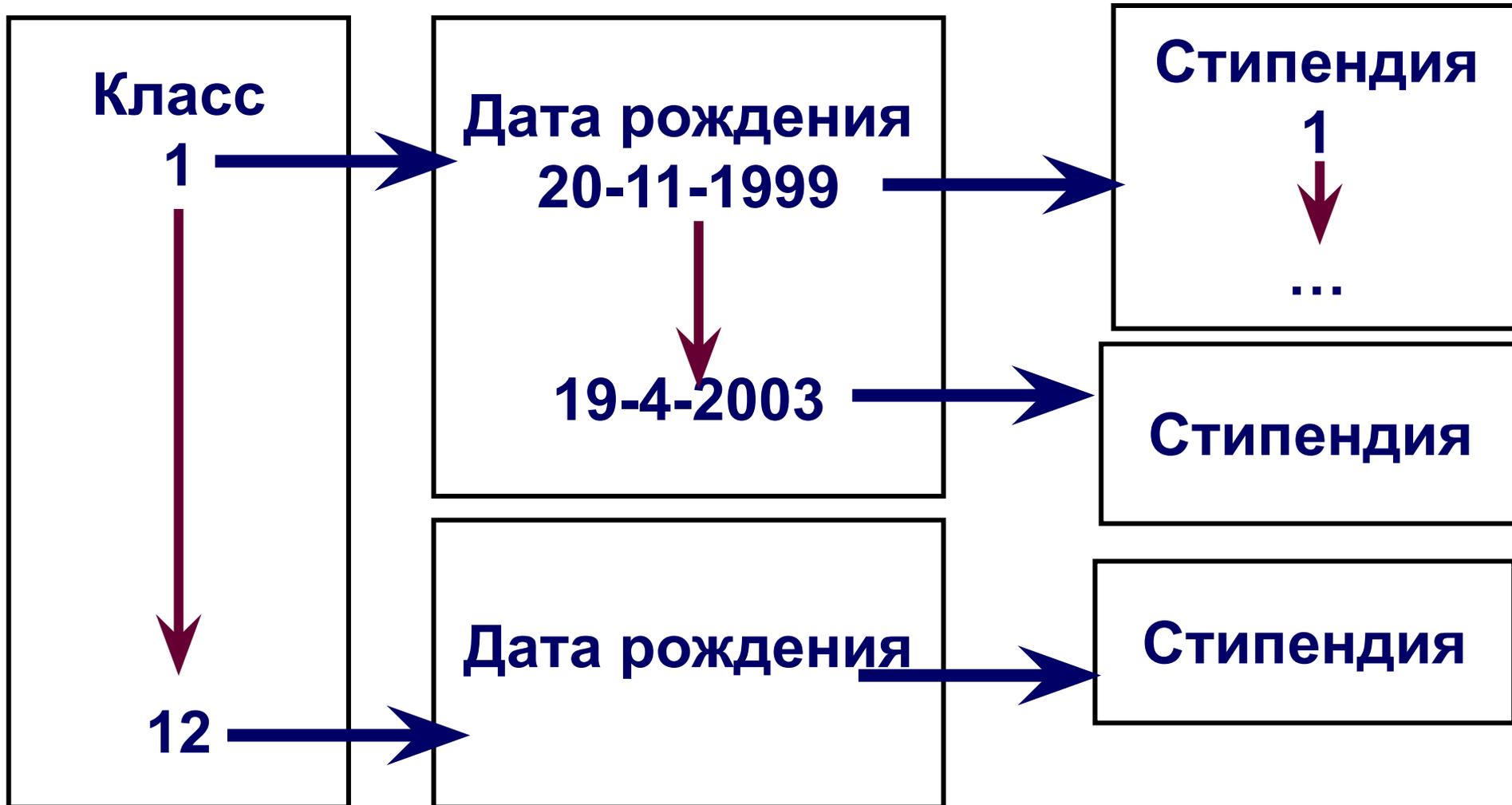
```
mysql> select last_name, first_name, s
-> order by first_name, stip;
+-----+-----+-----+
| last_name | first_name | stip |
+-----+-----+-----+
| Волков   | Валерий   | 1500.00 |
| Иванов  | Иван      | 1100.00 |
| Иванов  | Иван      | 1520.20 |
| Соколова | Инга      | 1200.15 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Столбец первичной сортировки. Мы получим список всех студентов, упорядоченный по столбцу «first_name».

А по этому столбцу будет выполняться вторичная сортировка ПОСЛЕ сортировки по столбцу «first_name».

SELECT * FROM student

ORDER BY class, birth_date, stip;



Ключевое слово **DESC**
после имени столбца в
условии **ORDER BY**
упорядочивает результаты
по убыванию.

```
mysql> select last_name, first_name, stip from student
-> order by stip;
```

last_name	first_name	stip
Иванов	Иван	1100.00
Соколова	Инга	1200.15
Волков	Валерий	1500.00
Иванов	Иван	1520.20

```
4 rows in set (0.00 sec)
```

```
mysql> select last_name, first_name, stip from student
-> order by stip desc;
```

last_name	first_name	stip
Иванов	Иван	1520.20
Волков	Валерий	1500.00
Соколова	Инга	1200.15
Иванов	Иван	1100.00

```
4 rows in set (0.00 sec)
```

Функция SUM суммирует значения столбца «stip».

```
SELECT SUM(stip)
```

```
FROM student
```

```
WHERE gender= 'м';
```

Это условие ограничивает запрос, чтобы он складывал только стипендии мужского пола. Без него запрос просуммирует все содержимое столбца «stip».

Результат запроса

```
mysql> select sum(stip)
-> from student
-> where gender='M';
+-----+
| sum(stip) |
+-----+
|    4120.20 |
+-----+
1 row in set (0.00 sec)
```

SUM

**Суммирует числовые
значения в столбце**

GROUP BY

**группирует записи по
одинаковым значениям
столбца**

```
mysql> select gender, sum(stip) from student
-> group by gender;
```

gender	sum(stip)
Ж	1200.15
М	4120.20

```
2 rows in set (0.02 sec)
```

```
mysql> select gender, stip from student;
```

gender	stip
М	1520.20
Ж	1200.15
М	1100.00
М	1500.00

```
4 rows in set (0.00 sec)
```

AVG

**Возвращает среднее
значение для числового
столбца**

```
mysql> select gender, avg(stip) from student
-> group by gender;
```

gender	avg(stip)
W	1200.150000
M	1373.400000

```
2 rows in set (0.00 sec)
```

MAX (MIN)

**Возвращает наибольшее
(наименьшее) значение
столбца/группы**

```
mysql> select Last_name, first_name, stip
-> from student;
```

Last_name	first_name	stip
Иванов	Иван	1520.20
Соколова	Инга	1200.15
Иванов	Иван	1100.00
Волков	Валерий	1500.00

```
4 rows in set (0.00 sec)
```

```
mysql> select Last_name, first_name, max(stip)
-> from student;
```

Last_name	first_name	max(stip)
Иванов	Иван	1520.20

```
1 row in set (0.04 sec)
```

```
mysql> select Last_name, first_name, min(stip)
-> from student;
```

Last_name	first_name	min(stip)
Иванов	Иван	1100.00

```
1 row in set (0.00 sec)
```

Функция **COUNT** возвращает количество записей в столбце. Если запись содержит NULL, она не включается в подсчет.

```
mysql> select count(stip)
-> from student;
+-----+
| count(stip) |
+-----+
|          4 |
+-----+
1 row in set (0.01 sec)
```

DISTINCT

**возвращает только
уникальные значения,
без дубликатов**

```
mysql> select class  
-> from student;
```

class
9
11
11
11

```
4 rows in set (0.00 sec)
```

```
mysql> select distinct class  
-> from student;
```

class
9
11

```
2 rows in set (0.00 sec)
```

```
mysql> select count(distinct class)
-> from student;
```

```
+-----+
| count(distinct class) |
+-----+
|                2     |
+-----+
```

```
1 row in set (0.02 sec)
```

LIMIT

Определяет, сколько именно записей должен вернуть запрос, и с какой записи следует начинать отсчет

```
mysql> select Last_name, first_name, stip
-> from student
-> order by 3 desc
-> limit 2;
```

Last_name	first_name	stip
Иванов	Иван	1520.20
Волков	Валерий	1500.00

```
2 rows in set (0.01 sec)
```

Tovar	Cost	Count
Молоко	39	10
Кефир	26	9
Масло	99	4

SELECT Tovar **AS** 'Товар', cost*count **AS**
'К оплате'
FROM Zakaz;



Товар	К оплате
Молоко	390
Кефир	234
Масло	396