

Язык запросов SQL - Structured Query Language

Типы запросов данных:

- **SELECT** – выбрать строки из таблиц;
- **INSERT** – добавить строки в таблицу;
- **UPDATE** – изменить строки в таблице;
- **DELETE** – удалить строки в таблице.

Примеры на выборку данных:

```
SELECT column1, column2 FROM table_name;
SELECT * FROM table_name;
SELECT * FROM table_name WHERE column1=3;
```

Условные операторы: = равно, <> не равно, > больше,
 < меньше, >= больше или равно, <= меньше или равно

Использование условий BETWEEN и LIKE для сравнения с условием WHERE, а так же комбинации операторов AND и OR:

```
SELECT * FROM table_name WHERE ((Age >= 18) AND (LastName BETWEEN
'Иванов' AND 'Сидоров')) OR Company LIKE '%Motorola%';
```

Пример на добавление данных в таблицу

```
INSERT INTO table_name (column1, column2, column3) VALUES ('data1',
'data2', 'data3');
```

Примеры на изменение данных в таблице

```
UPDATE table_name SET column1 = 'data1', column2 = 'data2' WHERE
column3 = 'data3';
UPDATE table_name SET FirstName = 'Василий' WHERE FirstName = 'Вася'
AND LastName = 'Пупкин';
```

Пример на удаление строк из таблицы

```
DELETE FROM table_name WHERE column1 = 'data1';
```

Типы данных языка SQL

INT[(len)] - целое число длиной 4 байта, представляемое при выводе максимально len цифрами;

SMALLINT[(len)] - целое число длиной 2 байта, представляемое при выводе максимально len цифрами;

FLOAT[(len,dec)] - действительное число, представляемое при выводе максимально len символами с dec цифрами после десятичной точки;

CHAR(size) - строка символов фиксированной длины размером size символов;

VARCHAR(size) - строка символов переменной длины максимальным размером до size символов;

BLOB (Binary Large Object) - массив произвольных (двоичных) байтов (максимальный размер зависит от реализации, обычно это 65535 байт); этот тип данных может использоваться, например, для хранения изображений;

DATE - астрономическая дата;

TIME - астрономическое время.

Создание таблицы

CREATE TABLE имя_табл (спецификация, ...);

Примеры спецификаций:

имя_столбца тип_данных [NULL]

имя_столбца тип_данных NOT NULL [DEFAULT по_умолч] [PRIMARY KEY]

Описание первичного ключа

PRIMARY KEY имя_ключа (имя_столбца, ...)

Пример создания таблицы

```
CREATE TABLE nodes (
    id SMALLINT NOT NULL PRIMARY KEY,      # номер узла
    x FLOAT NOT NULL                       # x-координата
    y FLOAT NOT NULL                       # y-координата
```

Удаление таблицы

DROP TABLE имя_табл, ...

Арифметические функции

ABS(x) абсолютное значение x

SQRT(x) квадратный корень от x

MAX(x, y, ...) значение наибольшего элемента из списка x, y, ...

MIN(x, y, ...) значение наименьшего элемента из списка x, y, ...

Строковые функции

LEFT(s,n) первые n символов строки s

RIGHT(s,n) последние n символов строки s

SUBSTRING(s, m, n) строка, получаемая копированием n символов из строки s, начиная с m-ого символа строки s

LCASE(s) строка, полученная из s преобразованием всех букв в строчные

UCASE(s) строка, полученная из s преобразованием всех букв в прописные

CONCAT(s1, s2, ...) строка, полученная конкатенацией (слиянием) строк s1, s2, ...

LENGTH(s) lkbyf cnhjr b s

Агрегативные функции

SUM(x) сумма значений столбца x результирующей таблицы

MAX(x) наибольшее значение из всех значений ячеек столбца x

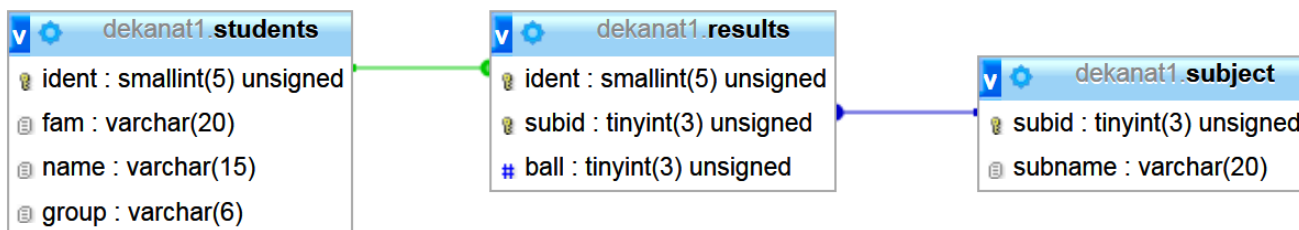
MIN(x) наименьшее значение из всех значений ячеек столбца x

AVG(x) среднее значение для всех значений ячеек столбца x

COUNT(x) общее количество ячеек в столбце x

Многотабличные запросы

Структура учебной БД «dekanat»



Целостность БД: связь ссылочного ключа (REFERENCE KEY) и внешнего (FOREIGN KEY)

ON DELETE } CASCADE, SET NULL, NO ACTION, RESTRICT
ON UPDATE }

Запрос на выборку результатов сдачи всех экзаменов:

```
SELECT `fam` AS 'Фамилия', `name` AS 'Имя', `group` AS 'Группа',
`subname` AS 'Предмет', `ball` AS 'Оценка' FROM `students`,
`subject`, `results` WHERE `students`.`ident` = `results`.`ident` AND
`subject`.`subid` = `results`.`subid` ORDER BY `fam`;
```

Таблица `results`

ident	subid	ball
1	1	3
1	2	4
2	1	5
2	2	4
3	1	2
3	2	3
4	1	5
4	2	5

Результат выполнения запроса

Фамилия	Имя	Группа	Предмет	Оценка
Васильев	Сергей	40-302	ПОВС	2
Васильев	Сергей	40-302	ПУ	3
Иванов	Петр	40-301	ПОВС	3
Иванов	Петр	40-301	ПУ	4
Николаев	Василий	40-302	ПОВС	5
Николаев	Василий	40-302	ПУ	5
Сергеев	Николай	40-301	ПОВС	5
Сергеев	Николай	40-301	ПУ	4

```
SELECT fam, name, students.group, subname, ball FROM students JOIN
(subject, results) ON students.ident = results.ident AND
subject.subid = results.subid ORDER BY fam;
```

fam	name	group	subname	ball
Васильев	Сергей	40-302	ПОВС	2
Васильев	Сергей	40-302	ПУ	3
Иванов	Петр	40-301	ПОВС	3
Иванов	Петр	40-301	ПУ	4
Николаев	Василий	40-302	ПОВС	5
Николаев	Василий	40-302	ПУ	5
Сергеев	Николай	40-301	ПОВС	5
Сергеев	Николай	40-301	ПУ	4

```
INSERT INTO students (fam, name, `group`) VALUES ('Новиков',
'Семен', '40-301');
INSERT INTO results (ident, subid, ball) VALUES (6,1,4),(6,2,3);
```

Таблица «students»

ident	fam	name	group
1	Иванов	Петр	40-301
2	Сергеев	Николай	40-301
3	Васильев	Сергей	40-302
4	Николаев	Василий	40-302
5	Новиков	Семен	30-301

Таблица «results»

ident	subid	ball
1	1	3
1	2	4
2	1	5
2	2	4
3	1	2
3	2	3
4	1	5
4	2	5
6	1	4
6	2	3

```
SELECT fam, name, students.group, subname, ball FROM students LEFT
JOIN (subject, results) ON students.ident = results.ident AND
subject.subid = results.subid ORDER BY fam;
```

fam	name	group	subname	ball
Васильев	Сергей	40-302	ПОВС	2
Васильев	Сергей	40-302	ПУ	3
Иванов	Петр	40-301	ПОВС	3
Иванов	Петр	40-301	ПУ	4
Николаев	Василий	40-302	ПОВС	5
Николаев	Василий	40-302	ПУ	5
Новиков	Семен	30-301	NULL	NULL
Сергеев	Николай	40-301	ПОВС	5
Сергеев	Николай	40-301	ПУ	4

```
SELECT fam, name, students.group, subname, ball FROM students RIGHT
JOIN (subject, results) ON students.ident = results.ident AND
subject.subid = results.subid ORDER BY subname;
```

fam	name	group	subname	ball
NULL	NULL	NULL	ПОВС	4
NULL	NULL	NULL	ПОВС	4
NULL	NULL	NULL	ПОВС	4
NULL	NULL	NULL	ПОВС	3
NULL	NULL	NULL	ПОВС	5
NULL	NULL	NULL	ПОВС	3
Иванов	Петр	40-301	ПОВС	3
Сергеев	Николай	40-301	ПОВС	5
Васильев	Сергей	40-302	ПОВС	2
Николаев	Василий	40-302	ПОВС	5
NULL	NULL	NULL	ПУ	3
NULL	NULL	NULL	ПУ	5
NULL	NULL	NULL	ПУ	2
NULL	NULL	NULL	ПУ	5
NULL	NULL	NULL	ПУ	4
Иванов	Петр	40-301	ПУ	4
Сергеев	Николай	40-301	ПУ	4
Васильев	Сергей	40-302	ПУ	3
Николаев	Василий	40-302	ПУ	5
NULL	NULL	NULL	ПУ	3

Запрос средних баллов студентов

```
SELECT fam, name, students.group, AVG(results.ball) AS 'ср.балл'
FROM results JOIN subject ON subject.subid = results.subid JOIN
students ON results.ident = students.ident GROUP BY students.ident
ORDER BY fam;
```

fam	name	group	ср.балл
Васильев	Сергей	40-302	2.5000
Иванов	Петр	40-301	3.5000
Николаев	Василий	40-302	5.0000
Сергеев	Николай	40-301	4.5000

Запрос средних баллов по группам и предметам

```
SELECT students.group, subname, AVG(results.ball) AS 'ср.балл' FROM
results JOIN subject ON subject.subid = results.subid JOIN students
ON results.ident = students.ident GROUP BY students.group, subname
ORDER BY students.group;
```

group	subname	ср.балл
40-301	ПОВС	4.0000
40-301	ПУ	4.0000
40-302	ПОВС	3.5000
40-302	ПУ	4.0000

Поиск и регулярные выражения

```
SELECT fam, name, students.group FROM students WHERE fam LIKE '%с%'
ORDER BY fam;
```

fam	name	group
Васильев	Сергей	40-302
Сергеев	Николай	40-301

```
SELECT fam, name, students.group FROM students WHERE name NOT LIKE
'%с%' ORDER BY fam;
```

fam	name	group
Иванов	Петр	40-301
Сергеев	Николай	40-301

Поиск соответствия шаблону *pat* :

expr REGEXP *pat* или *expr* RLIKE *pat*

```
SELECT fam, name, students.group FROM students WHERE fam RLIKE 'c'
ORDER BY fam;
```

```
+-----+-----+-----+
| fam      | name      | group    |
+-----+-----+-----+
| Васильев | Сергей    | 40-302   |
| Сергеев  | Николай  | 40-301   |
+-----+-----+-----+
```

```
SELECT fam, name, students.group FROM students WHERE fam RLIKE '^c'
ORDER BY fam;
```

```
+-----+-----+-----+
| fam      | name      | group    |
+-----+-----+-----+
| Сергеев  | Николай  | 40-301   |
+-----+-----+-----+
```

```
SELECT fam, name, students.group FROM students WHERE fam RLIKE 'ов$'
ORDER BY fam;
```

```
+-----+-----+-----+
| fam      | name      | group    |
+-----+-----+-----+
| Иванов   | Петр     | 40-301   |
+-----+-----+-----+
```

RLIKE '[авс]' соответствует подстроке содержащей 'а' или 'в' или 'с'.

RLIKE '[a-c]' соответствует подстроке содержащей любую букву в интервале от 'а' до 'с'.

Классы символов POSIX регулярных выражений

[**:alnum:**] – алфавитно-цифровые символы;

[**:alpha:**] – алфавитные символы;

[**:cntrl:**] – управляющие символы;

[**:blank:**] – символы пробела и табуляции;

[**:digits:**] – десятичные цифры;

[**:punct:**] – знаки препинания;

[**:lower:**] – символы алфавита в нижнем регистре;

[**:upper:**] – символы алфавита в верхнем регистре.

Пример **RLIKE** '[[:digit:]]' – соответствует строке, содержащей хотя бы одну цифру

Квантофикаторы

? - символ входит в строку один раз или вообще не входит;

* - любое число вхождений, в том числе и 0;

+ - одно или более число вхождений.