

# Язык SQL: манипулирование структурой данных

**Курс: MySQL 5.1**

Автор курса: Тарасов Алексей Владимирович

E-mail: [atarasov@specialist.ru](mailto:atarasov@specialist.ru)

# Темы модуля: Язык SQL

- Язык SQL
- Операторы определения данных
  - *Лабораторная работа: создание БД на сервере MySQL*
- Схемы данных
- Обзор INFORMATION\_SCHEMA
- Индексы и ограничения
- Добавление и удаление индексов
  - *Лабораторная работа: создание отношений на сервере*
- Операторы манипулирования данными
  - *Лабораторная работа: наполнение БД из внешних файлов*

# SQL – язык манипулирования данными

- **SQL** (*англ.* Structured Query Language — язык структурированных запросов) — универсальный язык, применяемый для создания, модификации и управления данными в реляционных базах данных
- Язык SQL делится на три части:
  - Операторы определения данных ([Data Definition Statements](#) , DDS)
  - Операторы манипуляции данными ([Data Manipulation Statements](#) , DMS)
  - Операторы определения доступа к данным ([Database Administration Statements](#), DAS)

# Операторы определения данных (DDL)

- Создание баз данных
- Удаление базы данных
- Создание таблиц
- Удаление таблиц
- Модификация таблиц

# Создание и удаление баз данных

- `CREATE DATABASE [IF NOT EXISTS] db_name`
- `DROP DATABASE [IF EXISTS] db_name`

## Пример

```
CREATE DATABASE IF NOT EXISTS courses
```

```
DROP DATABASE courses
```

# Создание таблиц

- `CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name (create_definition,...) [table_option ...]`
- `CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name [(create_definition,...)] [table_option ...] select_statement`
- `CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name { LIKE old_tbl_name | (LIKE old_tbl_name) }`

## Пример создания таблиц

```
CREATE TABLE IF NOT EXISTS `mydb`.`courses`  
(  
  `idcourses` INT NOT NULL ,  
  `title` VARCHAR(245) NULL ,  
  `hours` TINYINT UNSIGNED NULL ,  
  PRIMARY KEY (`idcourses`) )  
ENGINE = InnoDB
```

```
CREATE TABLE test (a INT NOT NULL  
  AUTO_INCREMENT,  
  PRIMARY KEY (a), KEY (b))  
ENGINE=MyISAM SELECT b,c FROM test2;
```

## Удаление таблиц

- DROP [TEMPORARY] TABLE  
[IF EXISTS] *tbl\_name* [, *tbl\_name*]  
...  
[RESTRICT | CASCADE]



# Модификация таблиц

```
ALTER [IGNORE] TABLE tbl_name  
    alter_specification [,  
    alter_specification] ...
```

alter\_specification:

```
    ADD [COLUMN] column_definition  
    [FIRST | AFTER col_name ]  
    | ADD {INDEX|KEY} [index_name]  
    [index_type] (index_col_name,...)  
    | ADD [CONSTRAINT [symbol]]  
        PRIMARY KEY [index_type]  
    (index_col_name,...)  
    | ADD [CONSTRAINT [symbol]]  
        UNIQUE [INDEX|KEY]  
    [index_name] [index_type]  
    (index_col_name,...)  
    | ADD [FULLTEXT|SPATIAL]  
    [INDEX|KEY] [index_name]  
    (index_col_name,...)  
    | ADD [CONSTRAINT [symbol]]  
        FOREIGN KEY [index_name]  
    (index_col_name,...)  
        [reference_definition]  
    | ALTER [COLUMN] col_name {SET  
    DEFAULT literal | DROP DEFAULT}
```

```
| CHANGE [COLUMN] old_col_name  
column_definition  
    [FIRST|AFTER col_name]  
    | MODIFY [COLUMN]  
column_definition [FIRST | AFTER  
col_name]  
    | DROP [COLUMN] col_name  
    | DROP PRIMARY KEY  
    | DROP {INDEX|KEY} index_name  
    | DROP FOREIGN KEY fk_symbol  
    | DISABLE KEYS  
    | ENABLE KEYS  
    | RENAME [TO] new_tbl_name  
    | ORDER BY col_name  
    | CONVERT TO CHARACTER SET  
charset_name [COLLATE  
collation_name]  
    | [DEFAULT] CHARACTER SET  
charset_name [COLLATE  
collation_name]  
    | DISCARD TABLESPACE  
    | IMPORT TABLESPACE
```

# Примеры применения ALTER

- `ALTER TABLE t1 RENAME t2;`
- `ALTER TABLE t2 MODIFY a TINYINT NOT NULL, CHANGE b c CHAR(20);`
- `ALTER TABLE t2 ADD d TIMESTAMP;`
- `ALTER TABLE t2 DROP COLUMN c;`
- `ALTER TABLE t2 ADD c INT UNSIGNED NOT NULL AUTO_INCREMENT, ADD PRIMARY KEY (c);`
- [Документация](#)

# Оператор DESCRIBE

**{DESCRIBE | DESC} *tbl\_name* [*col\_name* | *wild*]**

```
mysql> DESCRIBE city;
```

| Field      | Type     | Null | Key | Default | Extra          |
|------------|----------|------|-----|---------|----------------|
| Id         | int(11)  | NO   | PRI | NULL    | auto_increment |
| Name       | char(35) | NO   |     |         |                |
| Country    | char(3)  | NO   | UNI |         |                |
| District   | char(20) | YES  | MUL |         |                |
| Population | int(11)  | NO   |     | 0       |                |

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

# Лабораторная работа



- Создайте базу данных courses
- Создайте таблицы в этой базе данных
- Проверьте описания созданных таблиц

# Схема данных

- Use INFORMATION\_SCHEMA;
- SHOW TABLES;

```

+-----+
| Tables_in_information_schema |
+-----+
| CHARACTER_SETS |
| COLLATIONS |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS |
| COLUMN_PRIVILEGES |
| ENGINES |
| EVENTS |
| FILES |
| GLOBAL_STATUS |
| GLOBAL_VARIABLES |
| KEY_COLUMN_USAGE |
| PARTITIONS |
| PLUGINS |
| PROCESSLIST |
| PROFILING |
| REFERENTIAL_CONSTRAINTS |
| ROUTINES |
| SCHEMATA |
| SCHEMA_PRIVILEGES |
| SESSION_STATUS |
| SESSION_VARIABLES |
| STATISTICS |
| TABLES |
| TABLE_CONSTRAINTS |
| TABLE_PRIVILEGES |
| TRIGGERS |
| USER_PRIVILEGES |
| VIEWS |
+-----+
28 rows in set (0.00 sec)
```

# Получение данных о базе данных сервера

```
mysql> SELECT * FROM SCHEMATA\G
***** 1. row *****
      CATALOG_NAME: NULL
      SCHEMA_NAME: information_schema
DEFAULT_CHARACTER_SET_NAME: utf8
DEFAULT_COLLATION_NAME: utf8_general_ci
      SQL_PATH: NULL
***** 2. row *****
      CATALOG_NAME: NULL
      SCHEMA_NAME: mysql
DEFAULT_CHARACTER_SET_NAME: utf8
DEFAULT_COLLATION_NAME: utf8_general_ci
      SQL_PATH: NULL
***** 3. row *****
      CATALOG_NAME: NULL
      SCHEMA_NAME: test
DEFAULT_CHARACTER_SET_NAME: utf8
DEFAULT_COLLATION_NAME: utf8_general_ci
      SQL_PATH: NULL
3 rows in set (0.02 sec)
```

## Получение данных о таблице

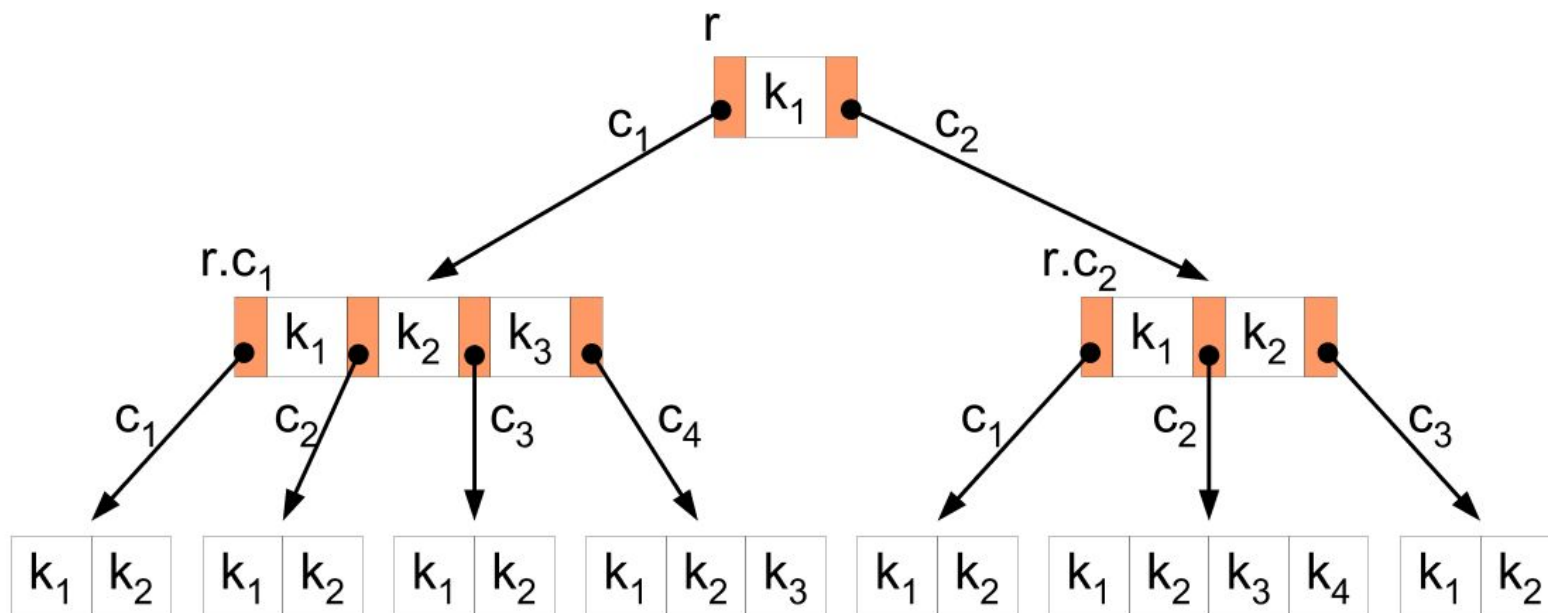
- `SELECT table_name  
FROM INFORMATION_SCHEMA.TABLES  
[WHERE table_schema = 'db_name']`

# Демонстрация работы с INFORMATION\_SCHEMA

- База данных INFORMATION\_SCHEMA
- Просмотр списка БД
- Просмотр информации о таблицах



# Индексы и ограничения



Взято из [Википедия](#)

# Создание индексов

```
CREATE [UNIQUE|FULLTEXT|SPATIAL]  
INDEX index_name  
    [index_type]  
ON tbl_name (index_col_name,...)
```

index\_col\_name:

```
col_name [(length)] [ASC | DESC]
```

index\_type:

```
USING {BTREE | HASH}
```

## Добавление и удаление индексов

```
ALTER TABLE t2  
  ADD INDEX (d) ,  
  ADD INDEX (a) ;
```

```
DROP INDEX index_name ON  
  tbl_name
```

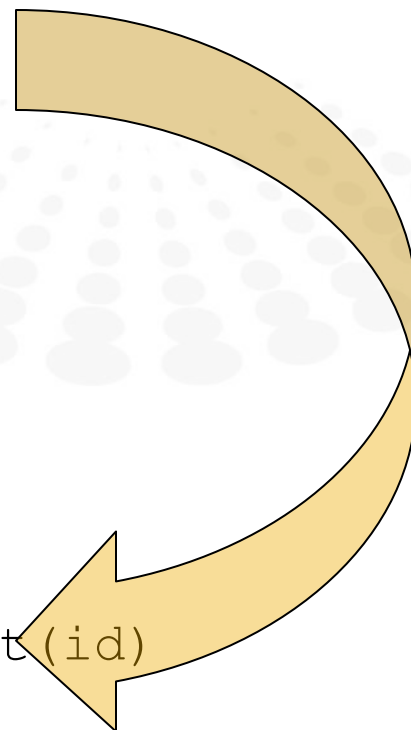
# Ограничения

- PRIMARY KEY
- UNIQUE
- FOREIGN KEY
- ENUM
- SET

# Пример связи таблиц ограничениями

```
CREATE TABLE parent (  
  id INT NOT NULL,  
  PRIMARY KEY (id)  
) ENGINE=INNODB;
```

```
CREATE TABLE child (  
  id INT,  
  parent_id INT,  
  INDEX par_ind (parent_id),  
  FOREIGN KEY (parent_id) REFERENCES parent(id)  
  ON DELETE CASCADE  
) ENGINE=INNODB;
```



# Операторы манипулирования данными

- INSERT
- LOAD DATA INFILE
- DELETE
- UPDATE

# Оператор INSERT

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name [(col_name,...)]
    VALUES ({expr | DEFAULT},...), (...), ...
    [ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name
    SET col_name={expr | DEFAULT}, ...
    [ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name [(col_name,...)]
    SELECT ...
    [ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

# Примеры оператора INSERT

```
INSERT INTO tbl_name (col1,col2) VALUES (15,col1*2);
```

```
INSERT INTO table (a,b,c)  
VALUES (1,2,3)  
ON DUPLICATE KEY UPDATE c=c+1;
```

```
INSERT INTO tbl_temp2 (fld_id)  
SELECT tbl_temp1.fld_order_id  
FROM tbl_temp1  
WHERE tbl_temp1.fld_order_id > 100;
```



# Оператор LOAD DATA INFILE

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE
'file_name'
[REPLACE | IGNORE]
INTO TABLE tbl_name
[FIELDS
    [TERMINATED BY 'string']
    [[OPTIONALLY] ENCLOSED BY 'char']
    [ESCAPED BY 'char']
]
[LINES
    [STARTING BY 'string']
    [TERMINATED BY 'string']
]
[IGNORE number LINES]
[(col_name_or_user_var,...)]
[SET col_name = expr,...)]
```

## Пример LOAD DATA INFILE

```
LOAD DATA INFILE 'data.txt'  
  INTO TABLE db2.my_table;
```

```
LOAD DATA INFILE 'data.txt'  
  INTO TABLE db2.my_table  
  FIELDS TERMINATED BY '\t'  
  ENCLOSED BY ''  
  ESCAPED BY '\\'  
  LINES TERMINATED BY '\n'  
  STARTING BY '';
```

```
mysql -e "LOAD DATA INFILE 'x' INTO TABLE x" x
```

# Оператор DELETE

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name  
    [WHERE where_condition]  
    [ORDER BY ...]  
    [LIMIT row_count]
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]  
    tbl_name[.*] [, tbl_name[.*]] ...  
FROM table_references  
    [WHERE where_condition]
```

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]  
    FROM tbl_name[.*] [, tbl_name[.*]] ...  
USING table_references  
    [WHERE where_condition]
```

## Примеры DELETE

```
DELETE FROM somelog WHERE user = 'jcole';
```

```
DELETE FROM somelog WHERE user = 'jcole'  
ORDER BY timestamp_column LIMIT 1;
```

```
DELETE t1, t2 FROM t1, t2, t3  
WHERE t1.id=t2.id AND t2.id=t3.id;
```

```
DELETE FROM t1, t2  
USING t1, t2, t3  
WHERE t1.id=t2.id AND t2.id=t3.id
```

# Оператор UPDATE

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name  
    SET col_name1=expr1  
    [, col_name2=expr2 ...]  
    [WHERE where_condition]  
    [ORDER BY ...]  
    [LIMIT row_count]
```

```
UPDATE [LOW_PRIORITY] [IGNORE] table_references  
    SET col_name1=expr1  
    [, col_name2=expr2 ...]  
    [WHERE where_condition]
```

# Примеры UPDATE

```
UPDATE t SET id = id + 1;
```

```
UPDATE t  
  SET id = id + 1  
  ORDER BY id DESC;
```

```
UPDATE items, month  
  SET items.price=month.price  
  WHERE items.id=month.id;
```

# Лабораторная работа



- **Наполнение БД из внешних файлов**
  - Найдите файл в раздаче с данными
  - Импортируйте данные в Вашу базу данных
  - Проверьте заполненные таблицы

# Создание и удаление событий

- `CREATE EVENT [IF NOT EXISTS] event_name`  
`ON SCHEDULE schedule`  
`AT timestamp [+ INTERVAL interval] ... |`  
`EVERY interval [STARTS timestamp [+ INTERVAL`  
`interval] ...] [ENDS timestamp [+ INTERVAL`  
`interval] ...]`

Interval: quantity {YEAR | QUARTER | MONTH |  
DAY | HOUR | MINUTE | WEEK | SECOND |  
YEAR\_MONTH | DAY\_HOUR | DAY\_MINUTE |  
DAY\_SECOND | HOUR\_MINUTE | HOUR\_SECOND |  
MINUTE\_SECOND}

- `DROP EVENT [IF EXISTS] event_name`



## Пример создания и удаления события

- `CREATE EVENT myevent`

```
ON SCHEDULE AT CURRENT_TIMESTAMP +  
INTERVAL 5 SECOND DO UPDATE  
shop.product SET price = price + 5 WHERE  
id = 4;
```

```
CREATE EVENT myevent2 ON SCHEDULE EVERY  
5 SECOND
```

```
DO UPDATE module6.product SET price =  
price + 5 WHERE id = 3;
```

```
DROP EVENT IF EXISTS myevent2
```

## ВЫВОДЫ:

- Операторы определения данных
- Схемы данных
- Индексы и ограничения
- Добавление и удаление индексов
- Добавление и удаление событий
- Операторы манипулирования данными

компьютерного  
**Центр**  
(ОБУЧЕНИЯ)  
**«СПЕЦИАЛИСТ»**  
при МГТУ им. Н.Э.Баумана

## КУРС ВЕДЕТ

Тарасов Алексей Владимирович

## КОНТАКТНЫЕ ДАННЫЕ:

email: [atarasov@specialist.ru](mailto:atarasov@specialist.ru)

сайты: <http://imysql.ru>, <http://jdrupal.ru>

twitter: altarasov

[Расписание занятий](#)

