

Введение в SQL

Часть 1

Стандарты SQL

Год	Название	Др. название	Изменения
1986	SQL-86	SQL-87	Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987 году.
1989	SQL-89	FIPS 127-1	Немного доработанный вариант предыдущего стандарта
1992	SQL-92	SQL2, FIPS-127-2	Значительные изменения (ISO 9075); уровень Entry Level стандарта SQL-92 был принят как стандарт FIPS-127-2
1999	SQL: 1999	SQL3	Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, не скалярные типы данных и некоторые объектно-ориентированные возможности.
2003	SQL: 2003		Введены расширения для работы с XML данными, оконные функции(применяемые для работы с OLAP-базами данных), генераторы последовательностей и основанные на них типы данных, появление команды MERGE.
2006	SQL: 2006		Функциональность работы с XML-данными значительно расширена. возможность совместно использовать в запросах SQL и XQuery
2008	SQL: 2008		Улучшены возм-ти оконных фун-й, устранены некот. неоднозначности стандарта SQL:2003
2011	SQL: 2011		Delete в MERGE, изменение данных в операторе Select, изменение синтаксиса вызова процедур, стандартизация возврата запросом процента от всех строк (не количества), возможность отключать ограничения на данные в таблице(CHECK, UNIQUE, REFERENCE) ,улучшение работы оконных ф-й
2016	SQL: 2016		Поддержка JSON, функция LISTAGG (перечисление через запятую значений поля),распознавание шаблонов строк, (регулярные выражения применительно к строкам) , в т.ч работа с оконными функциями
2019	SQL: 2019		Многомерные массивы

Состав SQL

- SQL (*structured query language*)

Data Definition Language (DDL) *язык*

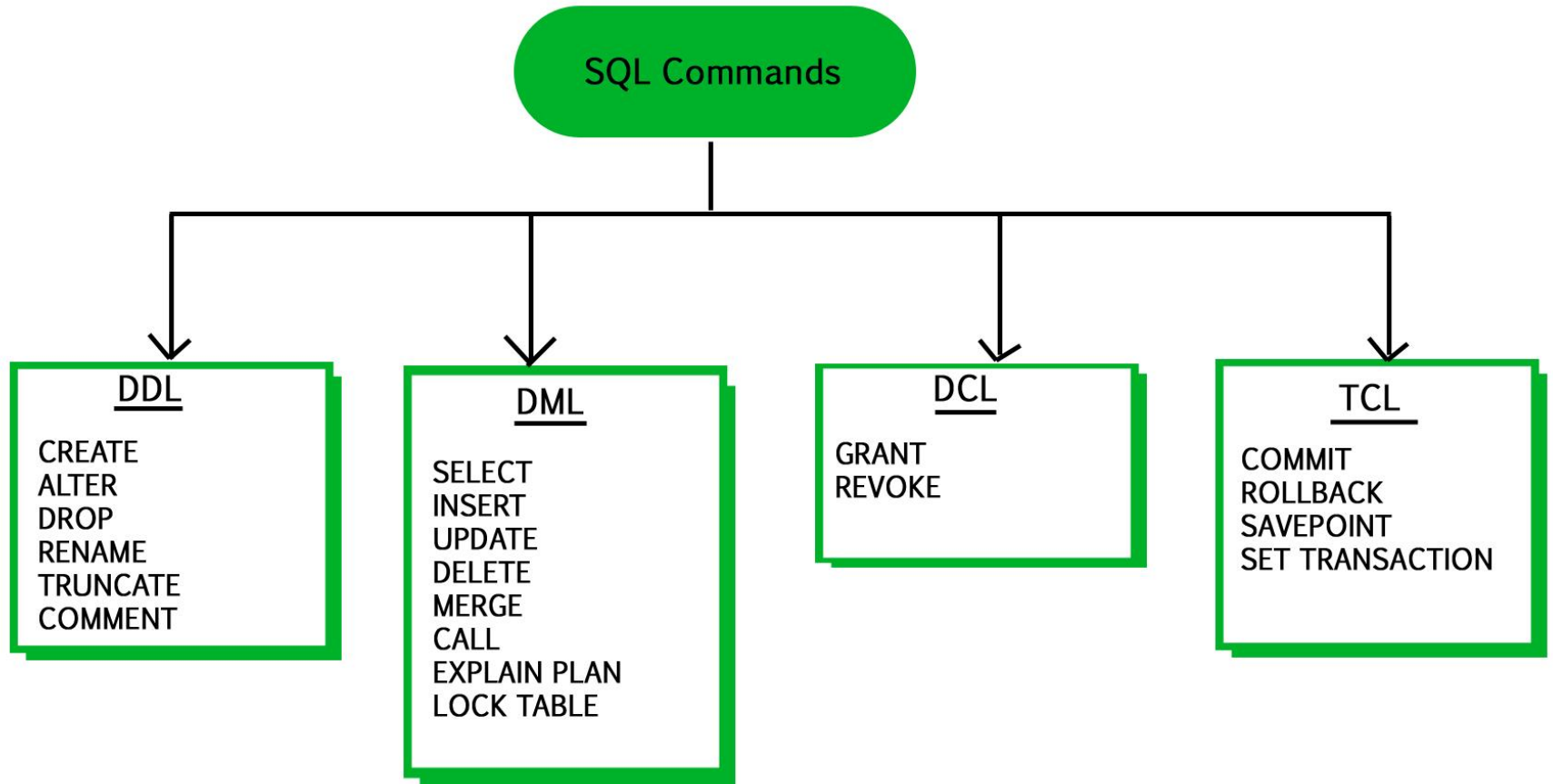
определения данных

• **Data Manipulation Language (DML)** *язык манипулирования данными*

• Transaction Control Language (TCL)

• Data Control Language (DCL)

Состав SQL



DDL vs DML

	DDL	DML
+	CREATE	INSERT
-	DROP	DELETE
Δ	ALTER	UPDATE

Типы данных SQL

Тип данных	Объявления	Реализация Microsoft SQL Server	Реализация MySQL
boolean (Логический)	BOOLEAN		BOOL, BOOLEAN, TINYINT(1).
character (Символьный)	CHAR	CHAR[(M)], NCHAR[(M)](unicode)	CHAR [(M)]
	VARCHAR	VARCHAR[(M)], NVARCHAR[(M)](unicode)	VARCHAR[(M)]
bit (Битовый)	BIT, BIT VARYING	BIT	BIT BIT[(M)]

Символьные типы данных

- Char(7)

- Лесоповал →
“Лесопов”

- Лес →
“Лес_ _ _ _”

- Varchar(7)

- Лесоповал →
“Лесопов”

- Лес →
“Лес”

Character Large object

- до 2 ГБ
- Нет значения по умолчанию
- Поиск по шаблону (like) не возможен
- Ограниченная сортировка и индексы

CHARSET и COLLATION

- CHARACTER SET — кодировка

SHOW CHARACTER SET;

- Collation — правила сравнения и хранения данных

- Collation - это набор правил (например, одно правило): «сравни коды»

SHOW CHARACTER SET LIKE 'utf%';

CHARSET кодировка

кодировка	описание	collation по умолчанию	Макс длина
armscii8	ARMSCII-8 Armenian	armscii8_general_ci	1
ascii	US ASCII	ascii_general_ci	1
big5	Big5 Traditional Chinese	big5_chinese_ci	2
binary	Binary pseudo charset	binary	1
cp1250	Windows Central European	cp1250_general_ci	1
cp1251	Windows Cyrillic	cp1251_general_ci	1
cp1256	Windows Arabic	cp1256_general_ci	1
	...		
ucs2	UCS-2 Unicode	ucs2_general_ci	2
ujis	EUC-JP Japanese	ujis_japanese_ci	3
utf16	UTF-16 Unicode	utf16_general_ci	4
utf16le	UTF-16LE Unicode	utf16le_general_ci	4
utf32	UTF-32 Unicode	utf32_general_ci	4
utf8	UTF-8 Unicode	utf8_general_ci	3
utf8mb4	UTF-8 Unicode	utf8mb4_0900_ai_ci	4

Типы данных SQL

Тип данных	Объявление	Реализация Microsoft SQL Server	Реализация MySQL
exact numeric (Точные числа)	NUMERIC	NUMERIC[(M[,D])]	DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]
	DECIMAL	DECIMAL[(M[,D])]	DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]
	SMALLINT	SMALLINT	SMALLINT[(M)] [UNSIGNED] [ZEROFILL]
	INTEGER	INT	INT[(M)] [UNSIGNED] [ZEROFILL] INTEGER[(M)] [UNSIGNED] [ZEROFILL]
	TINYINT	TINYINT	TINYINT[(M)] [UNSIGNED] [ZEROFILL] -128 .. 127 или 0..255
	BIGINT	BIGINT	BIGINT (8 байт)

Типы данных SQL

Тип данных	Объявления	Реализация Microsoft SQL Server	Реализация MySQL
approximate numeric (Округленные числа)	FLOAT	FLOAT[(N)]	FLOAT[(M,D)] [UNSIGNED] [ZEROFILL] REAL[(M,D)] [UNSIGNED] [ZEROFILL](если задан режим REAL_AS_FLOAT)
	REAL	REAL (эквивалентно Float(24))	
	DOUBLE PRECISION	DOUBLE PRECISION	DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL], REAL[(M,D)] [UNSIGNED] [ZEROFILL](если не задан режим REAL_AS_FLOAT)

Типы данных SQL

Тип данных	Объявления	Реализация Microsoft SQL Server	Реализация MySQL
datetime (Дата/время)	DATE	DATE	DATE
	TIME	TIME	TIME
	TIMESTAMP	DATETIME TIMESTAMP	DATETIME TIMESTAMP
interval (Интервал)	INTERVAL		С другим синтаксисом , но используются TIME и TIMESTAMP
LOB (Большой объект)	CHARACTER LARGE OBJECT,	TEXT, VARCHAR(MAX)	TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
	BINARY LARGE OBJECT	VARBINARY(MAX) VARBINARY(n)	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
XML (Правильный XML документ)		XML	XML

Создание базы данных MySQL

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS]  
  db_name [create_specification] ...
```

create_specification:

```
[DEFAULT] CHARACTER SET [=] charset_name  
| [DEFAULT] COLLATE [=] collation_name
```

Пример

- CREATE DATABASE IF NOT EXISTS my_db
CHARACTER SET ='cp1251'

Создание таблиц

```
CREATE TABLE TableName  
(  
  Описание 1,  
  Описание 2,  
  ...  
  Ограничение 1,  
  ...  
);
```

Описания столбцов таблицы

- имя столбца, `columnName`
- тип данных столбца, `dataType`
- обязательность столбца, `[NOT NULL] | [NULL]`
- ограничения на данные столбца, `[UNIQUE] [CHECK (searchCondition)]]`
- значение столбца по умолчанию `[DEFAULT defaultoption]`
- признак первичного ключа, если первичный ключ не является составным `[PRIMARY KEY]`

Первичный ключ

- *состав первичного ключа* [PRIMARY KEY(
ListOf Columns)]

Ограничения

Перед любым ограничением ВОЗМОЖНО

CONSTRAINT name_of_Constraint

- Уникальность **[UNIQUE(ListOfColumns)]**
- Проверка значений **[CHECK (searchCondition)]**
- Внешний ключ **[FOREIGN KEY
(listOfForeignKeyColumns)
REFERENCES ParentTableName
[(listOfCandidateKeyColumns)] ,
[MATCH [PARTIAL | FULL]]
[ON UPDATE. referentialAction]
[ON DELETE referentialAction] [, ...]]**

Пример стандарт

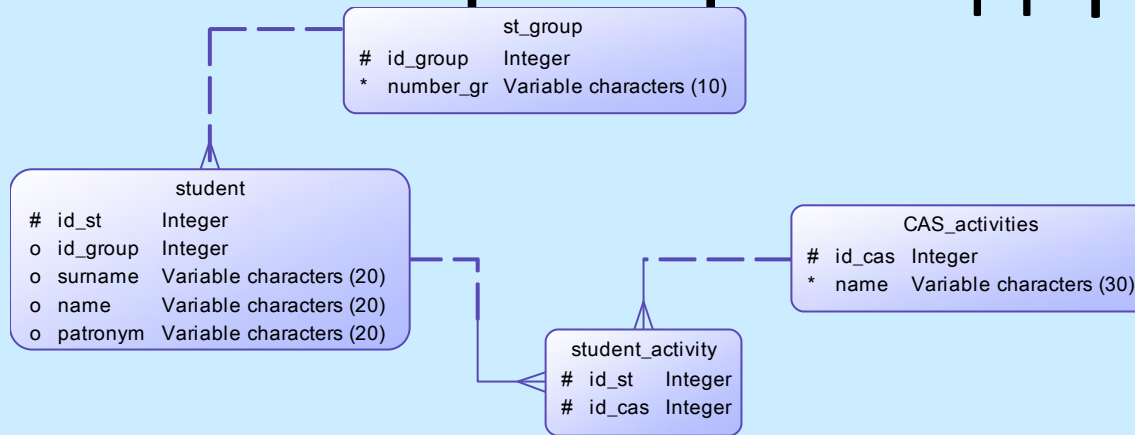


Таблица студент (student)

```
CREATE TABLE student (  
  id_st int NOT NULL AUTO_INCREMENT  
  PRIMARY KEY ,  
  surname varchar(20) NULL,  
  name varchar(20) NULL,  
  patronym varchar(20) DEFAULT NULL,  
  id_gr int DEFAULT NULL,  
  CONSTRAINT fk_gr_in_st FOREIGN KEY (id_gr)  
  REFERENCES st_group (id_gr)  
  ON DELETE RESTRICT  
  ON UPDATE RESTRICT  
)
```

Таб. кружки_студента (student_activity)

```
CREATE TABLE student_activity (  
  id_st int NOT NULL,  
  id_cas int NOT NULL,  
  PRIMARY KEY (id_st,id_cas),  
  CONSTRAINT st_act_cas FOREIGN KEY  
  (id_cas) REFERENCES cas_activities  
  (id_cas) ON DELETE RESTRICT  
  ON UPDATE RESTRICT,  
  CONSTRAINT st_act_stud FOREIGN KEY  
  (id_st) REFERENCES student (id_st)  
  ON DELETE RESTRICT  
  ON UPDATE RESTRICT  
)
```

Создание таблиц MySQL

- CREATE [TEMPORARY] TABLE [IF NOT EXISTS]
tbl_name (*create_definition*,...)
[table_options] *[partition_options]*

table_options

- AUTO_INCREMENT [=] value
- | AVG_ROW_LENGTH [=] value
- | [DEFAULT] CHARACTER SET [=] charset_name
- | CHECKSUM [=] {0 | 1}
- | [DEFAULT] COLLATE [=] collation_name
- | COMMENT [=] 'string'
- | COMPRESSION [=] {'ZLIB'|'LZ4'|'NONE'}
- | CONNECTION [=] 'connect_string'
- | {DATA|INDEX} DIRECTORY [=] 'absolute path to directory'
- | DELAY_KEY_WRITE [=] {0 | 1}
- | ENCRYPTION [=] {'Y' | 'N'}
- | ENGINE [=] engine_name
- | INSERT_METHOD [=] { NO | FIRST | LAST }
- | KEY_BLOCK_SIZE [=] value
- | MAX_ROWS [=] value
- | MIN_ROWS [=] value
- | PACK_KEYS [=] {0 | 1 | DEFAULT}
- | PASSWORD [=] 'string'
- | ROW_FORMAT [=] {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}
- | STATS_AUTO_RECALC [=] {DEFAULT|0|1}
- | STATS_PERSISTENT [=] {DEFAULT|0|1}
- | STATS_SAMPLE_PAGES [=] value
- | TABLESPACE tablespace_name
- | UNION [=] (tbl_name[,tbl_name]...)

Storage engine

механизм хранения	Описание
InnoDB	Транзакционно-безопасные таблицы с блокировкой строк и внешними ключами. Механизм хранения по умолчанию.
MyISAM	Бинарный переносимый механизм хранения, в основном используемый для систем с только чтением или в основном чтением
MEMORY	Таблицы, хранящиеся только в оперативной памяти
CSV	Таблицы, хранящие строки с данными, разделенными запятыми
ARCHIVE	Механизм хранения ARCHIVE создает специальные таблицы, в которых хранятся большие объемы неиндексированных данных на очень небольшом месте.
EXAMPLE	Движок EXAMPLE - это заглушка, которая ничего не делает. Его цель - служить примером в исходном коде MySQL, который иллюстрирует, как начать писать новые механизмы хранения.
FEDERATED	механизм хранения для работы с удаленными таблицами (remote)
HEAP	Синоним таблиц MEMORY
MERGE	Коллекция таблиц MyISAM, используемые как одна таблица
NDB	Кластерные, отказоустойчивые таблицы на основе памяти, поддерживающие транзакции и внешние ключи.

Пример MySQL

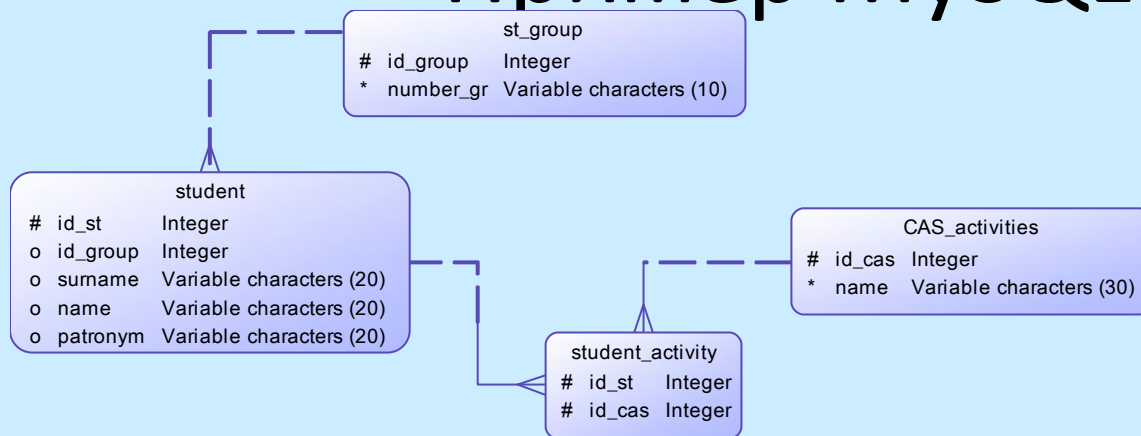


Таблица студент (student)

```
CREATE TABLE `student` (  
  `id_st` int NOT NULL AUTO_INCREMENT  
  PRIMARY KEY ,  
  `surname` varchar(20) DEFAULT NULL,  
  `name` varchar(20) DEFAULT NULL,  
  `patronym` varchar(20) DEFAULT NULL,  
  `id_gr` int DEFAULT NULL,  
  CONSTRAINT `fk_gr_in_st` FOREIGN KEY (`id_gr`)  
  REFERENCES `st_group` (`id_gr`)  
  ON DELETE RESTRICT ON UPDATE RESTRICT  
) ENGINE=InnoDB  
  AUTO_INCREMENT=1000  
  DEFAULT CHARSET=utf8mb4  
  COLLATE=utf8mb4_0900_ai_ci;
```

Таб. кружки_студента (student_activity)

```
CREATE TABLE `student_activity` (  
  `id_st` int NOT NULL,  
  `id_cas` int NOT NULL,  
  PRIMARY KEY (`id_st`,`id_cas`),  
  CONSTRAINT `st_act_cas` FOREIGN KEY  
  (`id_cas`) REFERENCES `cas_activities` (`id_cas`)  
  ON DELETE RESTRICT  
  ON UPDATE RESTRICT,  
  CONSTRAINT `st_act_stud` FOREIGN KEY  
  (`id_st`) REFERENCES `student` (`id_st`)  
  ON DELETE RESTRICT  
  ON UPDATE RESTRICT  
) ENGINE=InnoDB  
  DEFAULT CHARSET=utf8mb4  
  COLLATE=utf8mb4_0900_ai_ci;
```

Partition (разделы)

Способ разделения	
1. RANGE По диапазону значений	<pre>PARTITION BY RANGE (store_id) (PARTITION p0 VALUES LESS THAN (10), PARTITION p1 VALUES LESS THAN (20), PARTITION p3 VALUES LESS THAN (30));</pre>
2. LIST По точному списку значений	<pre>PARTITION BY LIST(store_id) (PARTITION pNorth VALUES IN (3,5,6,9,17), PARTITION pEast VALUES IN (1,2,10,11,19,20))</pre>
3. HASH По хэшу указанного поля	<pre>PARTITION BY HASH(store_id) PARTITIONS 4;</pre>
4. KEY по ключу.	<pre>PARTITION BY KEY(s1) PARTITIONS 10;</pre>

Partition (разделы)

partition_options:

```
PARTITION BY    {  
    [LINEAR] HASH(expr)      |  
    [LINEAR] KEY [ALGORITHM={1|2}] (column_list)  
    |  
    RANGE{(expr) | COLUMNS(column_list)}      |  
    LIST{(expr) | COLUMNS(column_list)}  
    }  
[PARTITIONS num]  
    [SUBPARTITION BY    {  
    [LINEAR] HASH(expr)      |  
    [LINEAR] KEY [ALGORITHM={1|2}] (column_list) }  
    [SUBPARTITIONS num]  ]  
[(partition_definition [,  
    partition_definition] ...)]
```

partition_definition

partition_definition:

PARTITION partition_name

[VALUES

{LESS THAN {(expr | value_list) | MAXVALUE}

|

IN (value_list))]

[[STORAGE] ENGINE [=] engine_name]

[COMMENT [=] 'string']

[DATA DIRECTORY [=] 'data_dir']

[INDEX DIRECTORY [=] 'index_dir']

[MAX_ROWS [=] max_number_of_rows]

[MIN_ROWS [=] min_number_of_rows]

[TABLESPACE [=] tablespace_name]

[(subpartition_definition [, subpartition_definition] ...)]

Partition (пример)

- ```
CREATE TABLE Timetable_exam (
 Discipline_name VARCHAR(30),
 group_num VARCHAR(8),
 lecturer_id INT,
 exam_date DATE,
 note VARCHAR(500)
) ENGINE = MYISAM
PARTITION BY RANGE(YEAR(exam_date)) (
 PARTITION p_old VALUES LESS THAN(2016),
 PARTITION p_2019 VALUES LESS THAN(2020),
 PARTITION p_2020 VALUES LESS THAN(MAXVALUE)
);
```

# Удаление таблиц

- DROP TABLE **TableName** [RESTRICT | CASCADE]

# Изменение структуры таблиц

- ввести новый столбец в таблицу;
- удалить столбец из таблицы;
- ввести новое ограничение таблицы;
- удалить ограничение таблицы;
- задать для столбца значение, применяемое по умолчанию;
- удалить опцию, предусматривающую применение для столбца значения, заданного по умолчанию.

# Изменение структуры таблиц

ALTER [IGNORE] TABLE tbl\_name alter\_spec [, alter\_spec ...]

alter\_specification:     ADD [COLUMN] create\_definition [FIRST | AFTER column\_name  
                          ]

или ADD [COLUMN] (create\_definition, create\_definition,...)

или ADD INDEX [index\_name] (index\_col\_name,...)

или ADD PRIMARY KEY (index\_col\_name,...)

или ADD UNIQUE [index\_name] (index\_col\_name,...)

или ADD FULLTEXT [index\_name] (index\_col\_name,...)

или ADD [CONSTRAINT symbol] FOREIGN KEY index\_name (index\_col\_name,...)  
      [reference\_definition]

или ALTER [COLUMN] col\_name {SET DEFAULT literal | DROP DEFAULT}

или CHANGE [COLUMN] old\_col\_name create\_definition               [FIRST | AFTER  
      column\_name]

или MODIFY [COLUMN] create\_definition [FIRST | AFTER column\_name]

или DROP [COLUMN] col\_name

или DROP PRIMARY KEY

или DROP INDEX index\_name

или DISABLE KEYS или ENABLE KEYS

или RENAME [TO] new\_tbl\_name

или ORDER BY col или table\_options

# Пример



- Alter table st\_group add column id\_gr integer auto\_increment;
- Alter table st\_group drop primary key ;
- Alter table st\_group add primary key(id\_gr);
- Alter table st\_group rename to st\_group2;