ТЕХНОТРЕК

Занятие №2

Работа с данными. Введение.

Лысак Павел, Дина Сафина

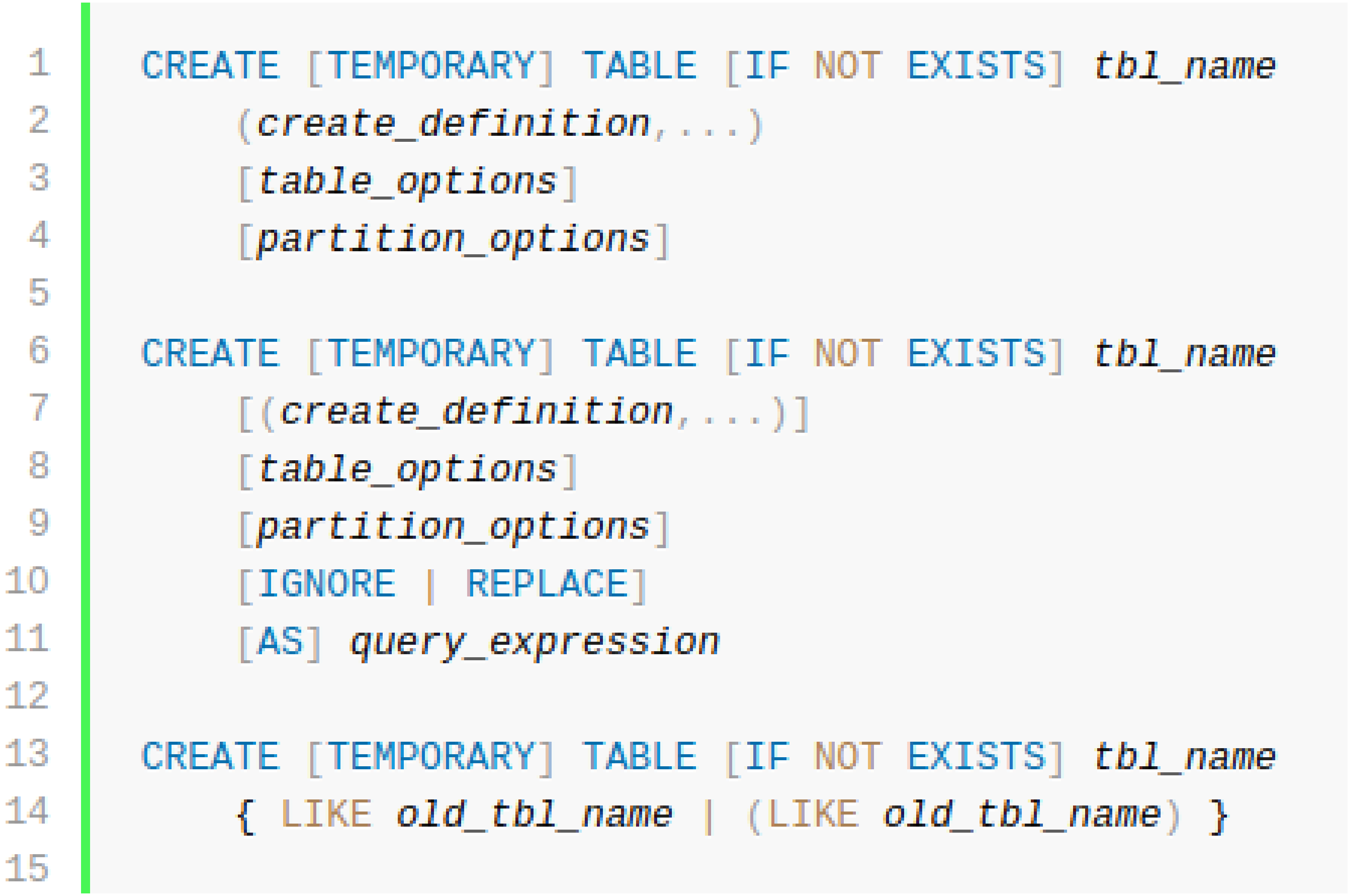
Работа с данными

1. DDL
2. DML

DDL - Data Definition Statements

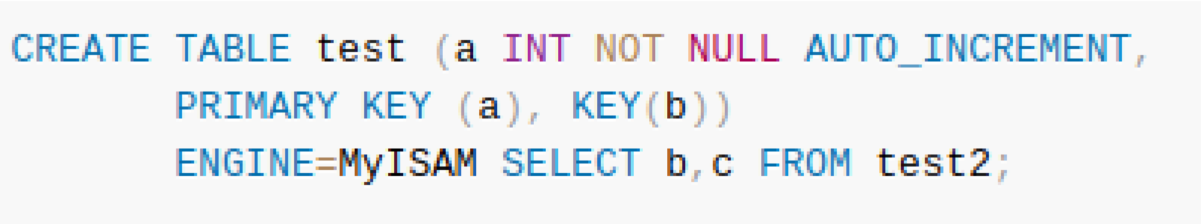
- CREATE - to create a database and its objects like (table, index, views, store procedure, function, and triggers)
- ALTER - alters the structure of the existing database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed

CREATE TABLE

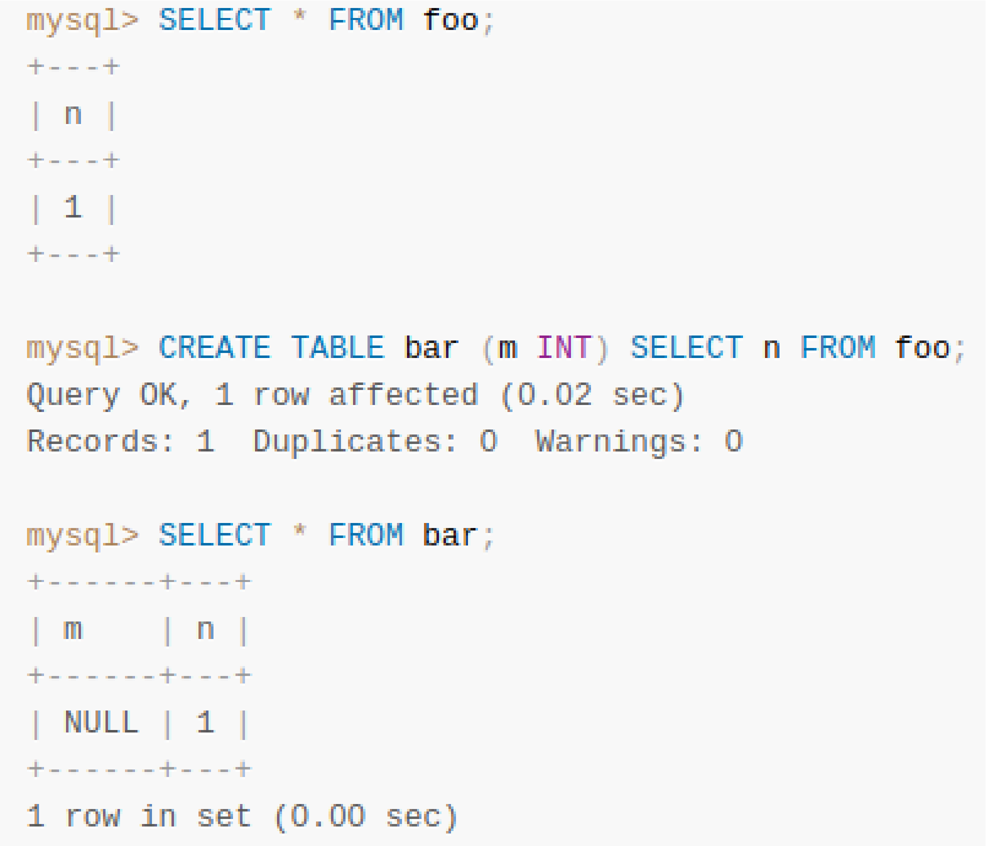


```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_options]
    [partition_options]

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_options]
    [partition_options]
    [IGNORE | REPLACE]
    [AS] query_expression

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    { LIKE old_tbl_name | (LIKE old_tbl_name) }
```

CREATE TABLE AS

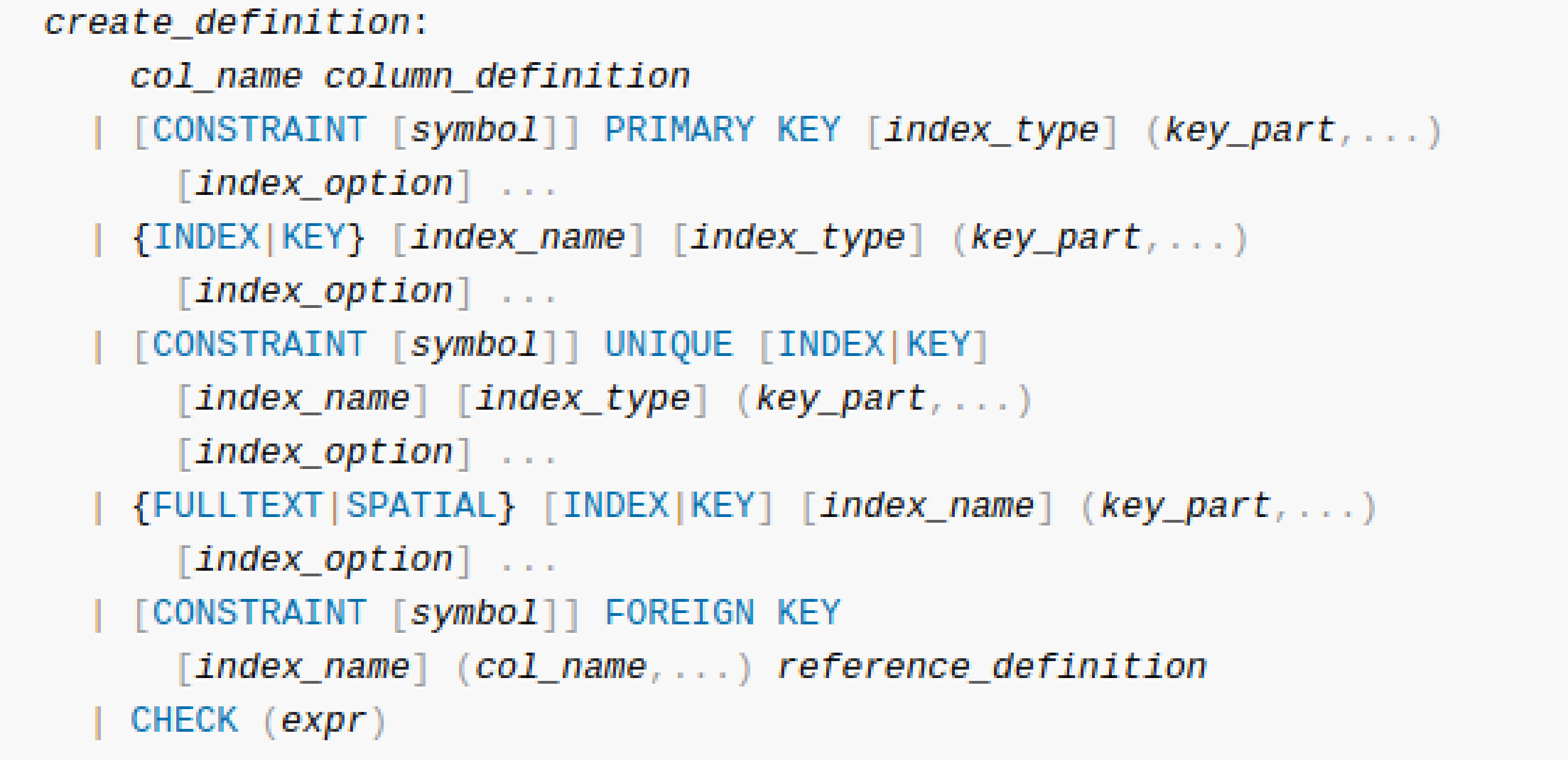


```
CREATE TABLE test (a INT NOT NULL AUTO_INCREMENT,
        PRIMARY KEY (a), KEY(b))
        ENGINE=MyISAM SELECT b,c FROM test2;
```



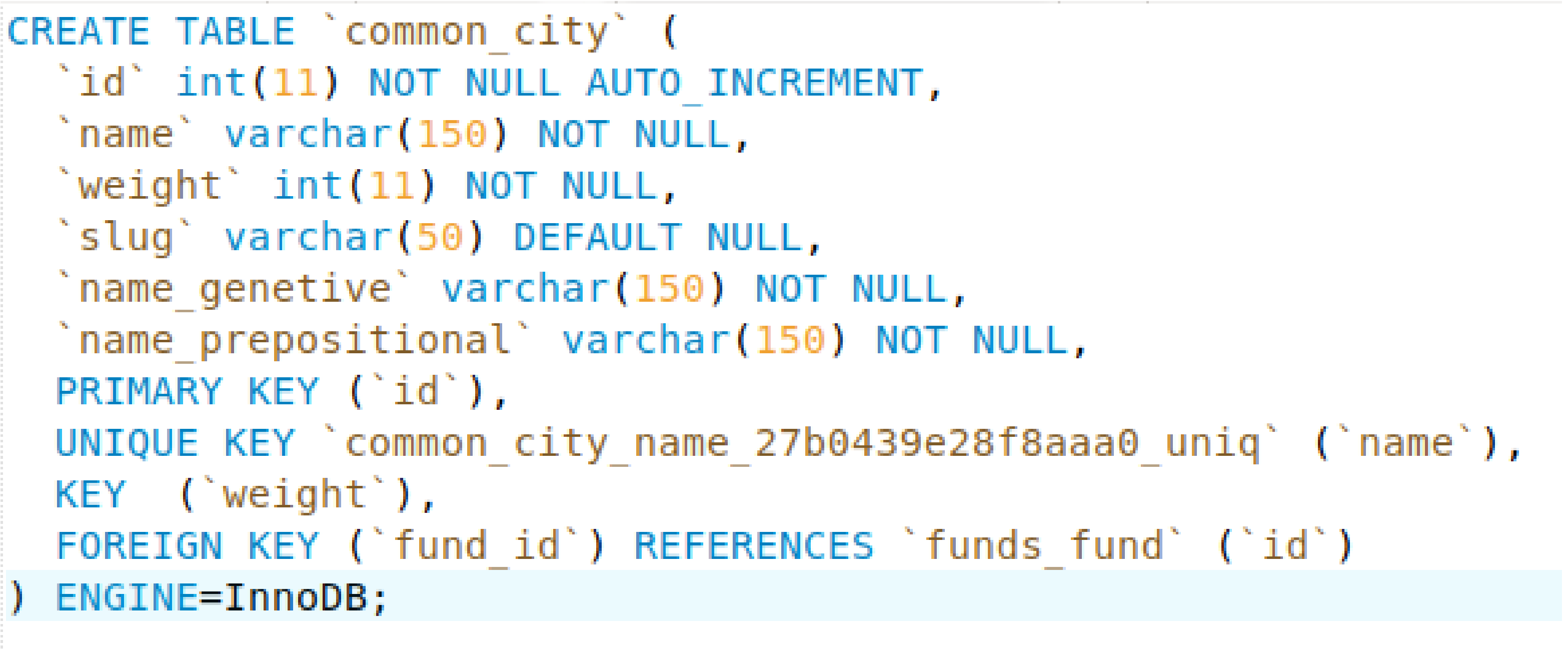
```
mysql> SELECT * FROM foo;
+---+
| n |
+---+
| 1 |
+---+

mysql> CREATE TABLE bar (m INT) SELECT n FROM foo;
Query OK, 1 row affected (0.02 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM bar;
+------+---+
| m    | n |
+------+---+
| NULL | 1 |
+------+---+
1 row in set (0.00 sec)
```

CREATE DEFENITION



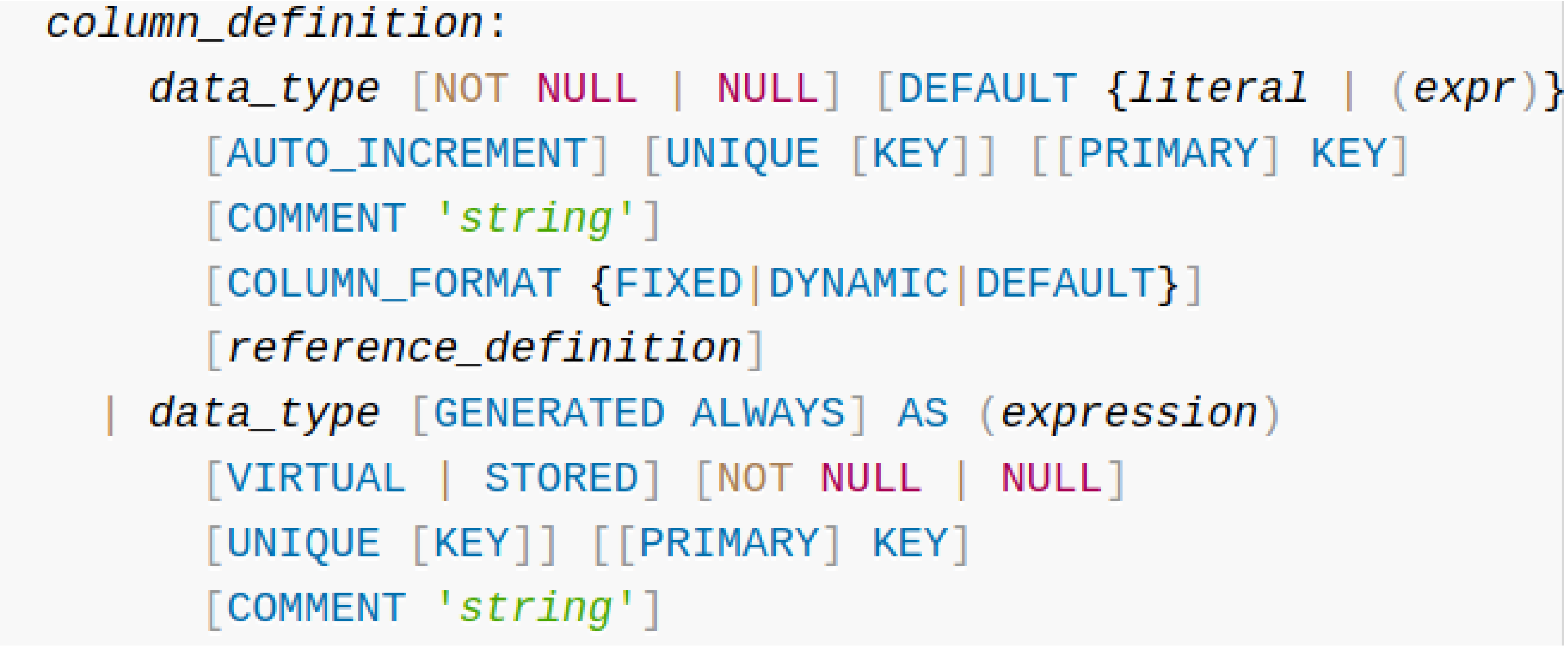
```
create_definition:
    col_name column_definition
  | [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (key_part,...)
      [index_option] ...
  | {INDEX|KEY} [index_name] [index_type] (key_part,...)
      [index_option] ...
  | [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]
      [index_name] [index_type] (key_part,...)
      [index_option] ...
  | {FULLTEXT|SPATIAL} [INDEX|KEY] [index_name] (key_part,...)
      [index_option] ...
  | [CONSTRAINT [symbol]] FOREIGN KEY
      [index_name] (col_name,...) reference_definition
  | CHECK (expr)
```

CREATE DEFENITION



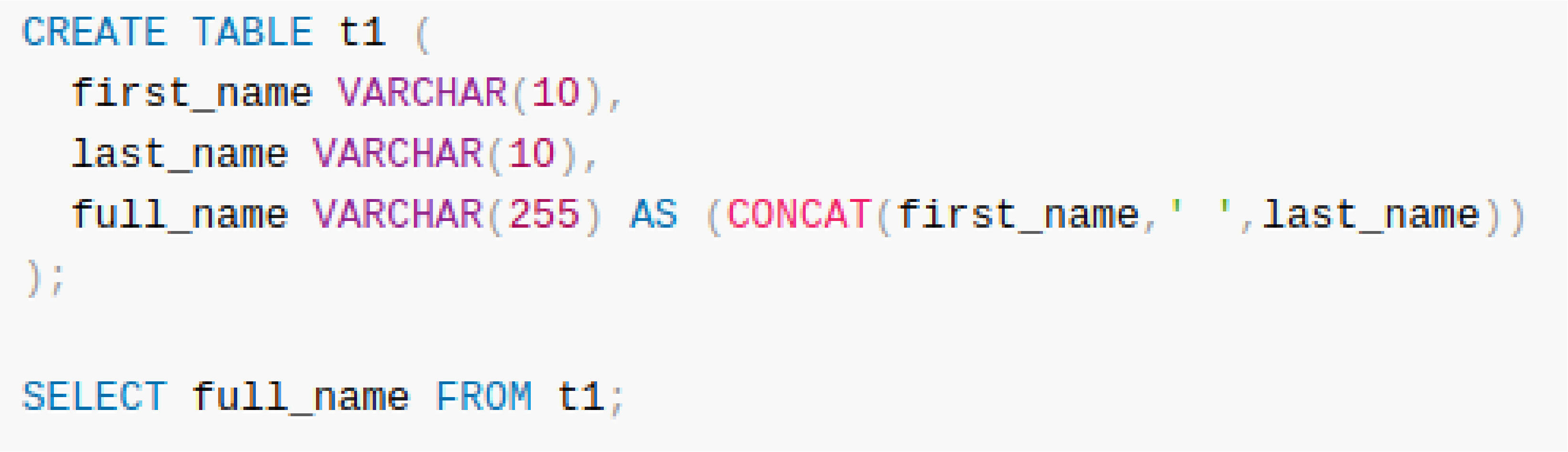
```sql
CREATE TABLE `common_city` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(150) NOT NULL,
  `weight` int(11) NOT NULL,
  `slug` varchar(50) DEFAULT NULL,
  `name_genetive` varchar(150) NOT NULL,
  `name_prepositional` varchar(150) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `common_city_name_27b0439e28f8aaa0_uniq` (`name`),
  KEY  (`weight`),
  FOREIGN KEY (`fund_id`) REFERENCES `funds_fund` (`id`)
) ENGINE=InnoDB;
```

COLUMN DEFENITION



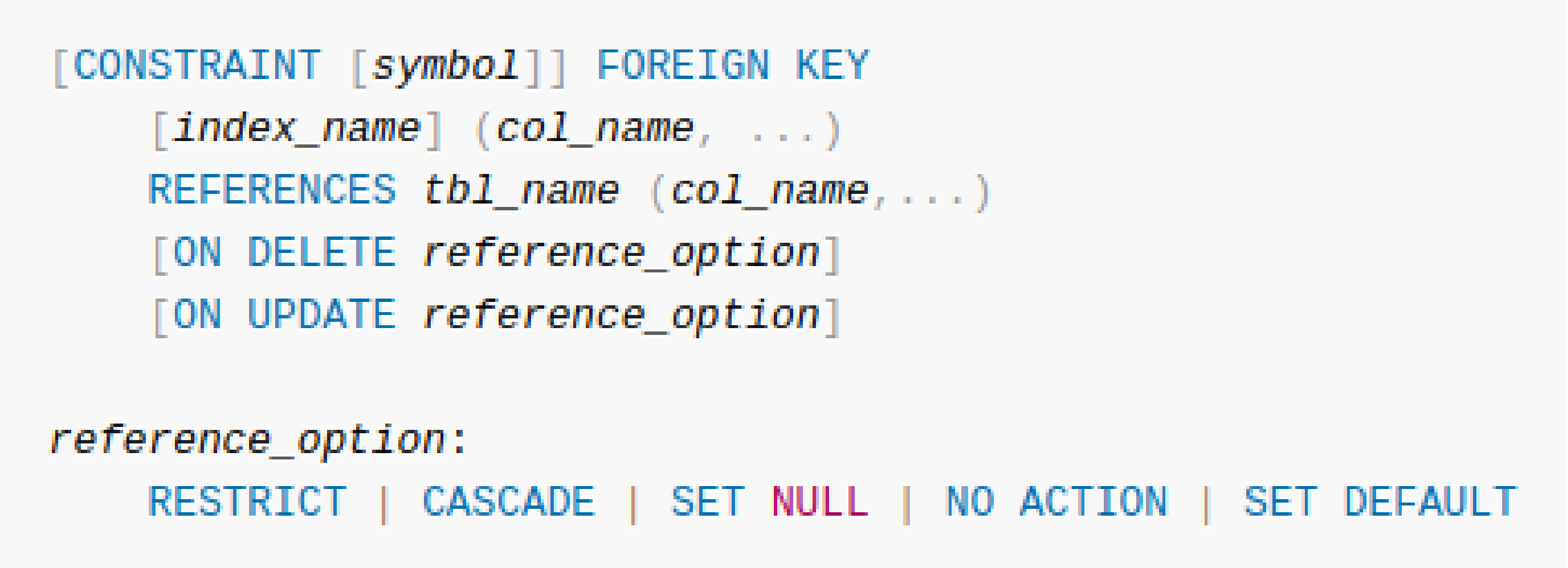
```
column_definition:
    data_type [NOT NULL | NULL] [DEFAULT {literal | (expr)}
      [AUTO_INCREMENT] [UNIQUE [KEY]] [[PRIMARY] KEY]
      [COMMENT 'string']
      [COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]
      [reference_definition]
  | data_type [GENERATED ALWAYS] AS (expression)
      [VIRTUAL | STORED] [NOT NULL | NULL]
      [UNIQUE [KEY]] [[PRIMARY] KEY]
      [COMMENT 'string']
```

GENERATED ALWAYS AS



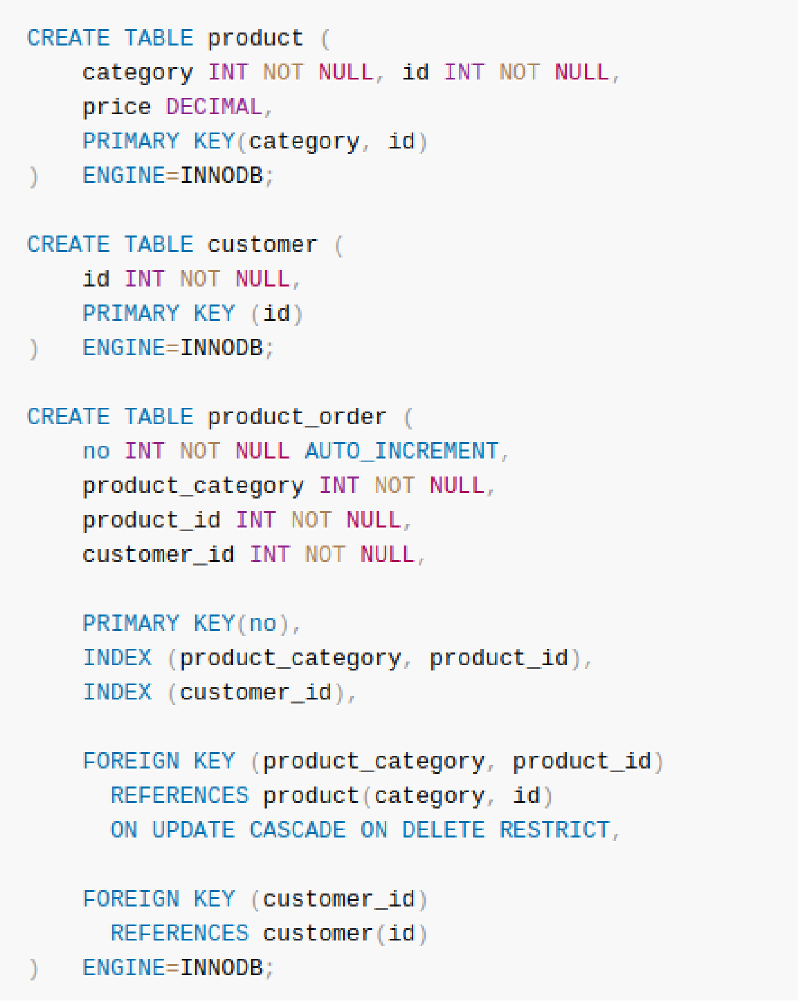
```
CREATE TABLE t1 (
  first_name VARCHAR(10),
  last_name VARCHAR(10),
  full_name VARCHAR(255) AS (CONCAT(first_name,' ',last_name))
);

SELECT full_name FROM t1;
```

FOREIGN KEY Constraints



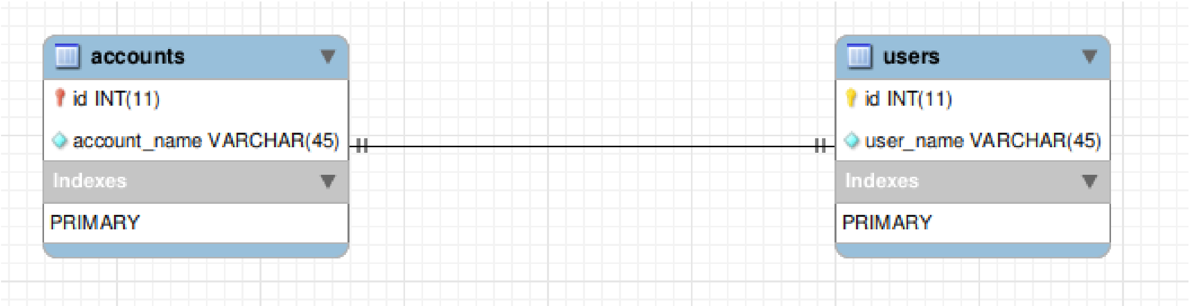
```
[CONSTRAINT [symbol]] FOREIGN KEY
    [index_name] (col_name, ...)
    REFERENCES tbl_name (col_name,...)
    [ON DELETE reference_option]
    [ON UPDATE reference_option]

reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```

FOREIGN KEY Constraints

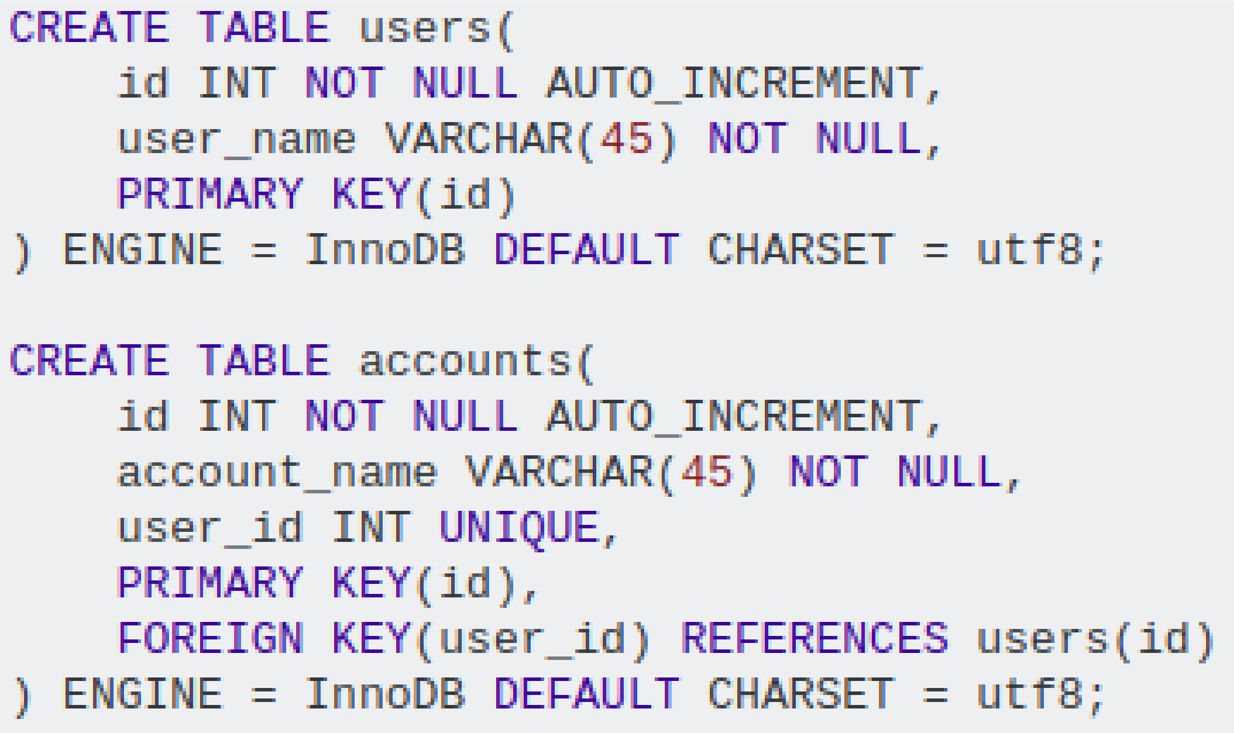


```
CREATE TABLE product (
    category INT NOT NULL, id INT NOT NULL,
    price DECIMAL,
    PRIMARY KEY(category, id)
)   ENGINE=INNODB;

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

CREATE TABLE product_order (
    no INT NOT NULL AUTO_INCREMENT,
    product_category INT NOT NULL,
    product_id INT NOT NULL,
    customer_id INT NOT NULL,

    PRIMARY KEY(no),
    INDEX (product_category, product_id),
    INDEX (customer_id),

    FOREIGN KEY (product_category, product_id)
      REFERENCES product(category, id)
      ON UPDATE CASCADE ON DELETE RESTRICT,

    FOREIGN KEY (customer_id)
      REFERENCES customer(id)
)   ENGINE=INNODB;
```

ONE TO ONE

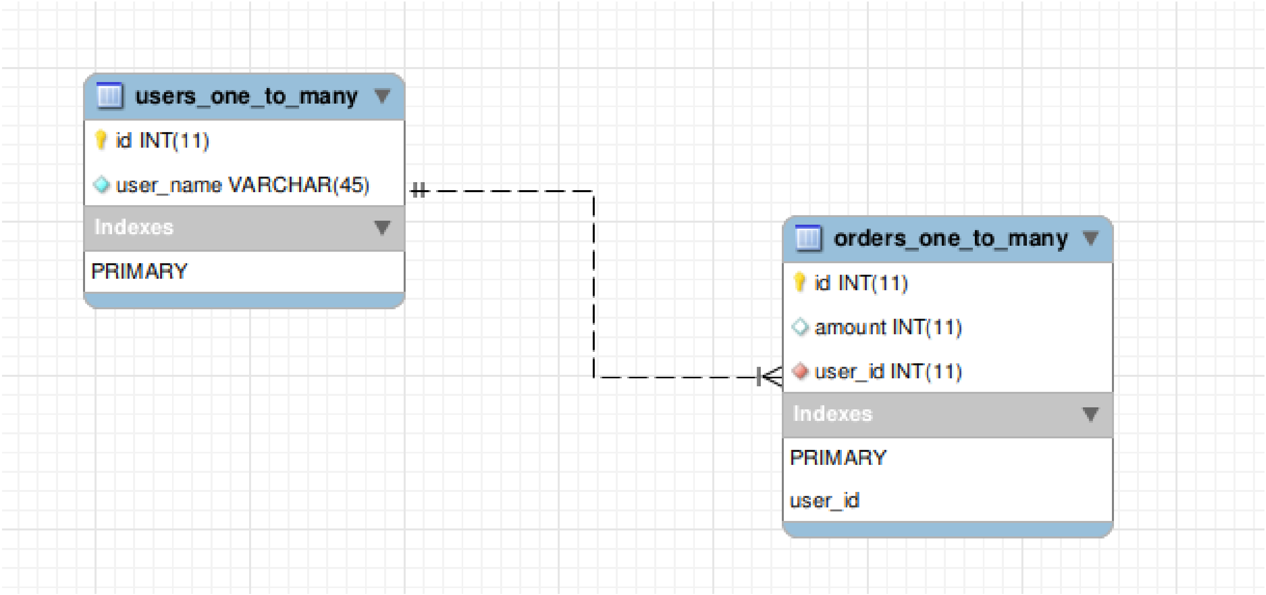

accounts
id INT(11)
account_name VARCHAR(45)
Indexes
PRIMARY
users
id INT(11)
user_name VARCHAR(45)
Indexes
PRIMARY

ONE TO ONE



```
CREATE TABLE users(
    id INT NOT NULL AUTO_INCREMENT,
    user_name VARCHAR(45) NOT NULL,
    PRIMARY KEY(id)
) ENGINE = InnoDB DEFAULT CHARSET = utf8;

CREATE TABLE accounts(
    id INT NOT NULL AUTO_INCREMENT,
    account_name VARCHAR(45) NOT NULL,
    user_id INT UNIQUE,
    PRIMARY KEY(id),
    FOREIGN KEY(user_id) REFERENCES users(id)
) ENGINE = InnoDB DEFAULT CHARSET = utf8;
```

ONE TO MANY



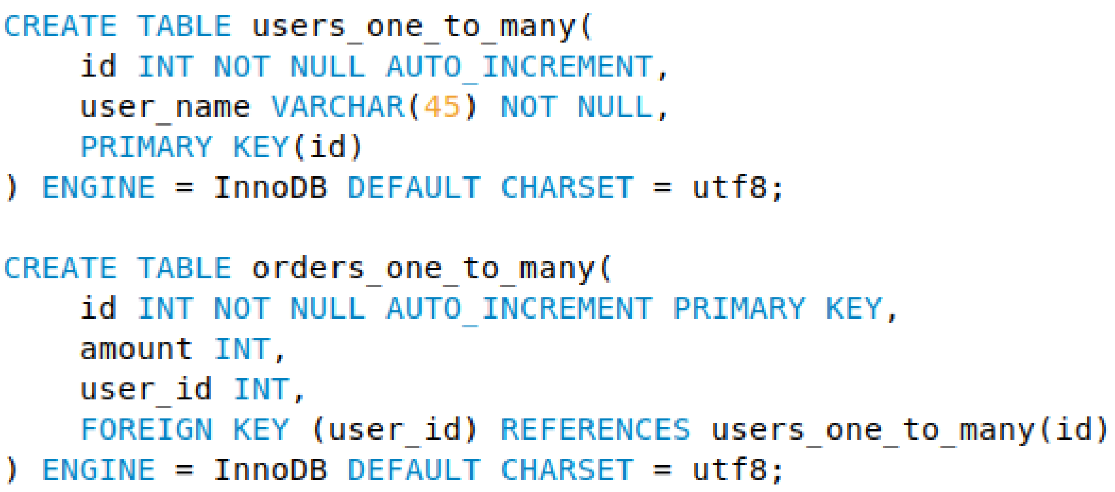
users_one_to_many
- id INT(11)
- user_name VARCHAR(45)

Indexes
- PRIMARY

orders_one_to_many
- id INT(11)
- amount INT(11)
- user_id INT(11)

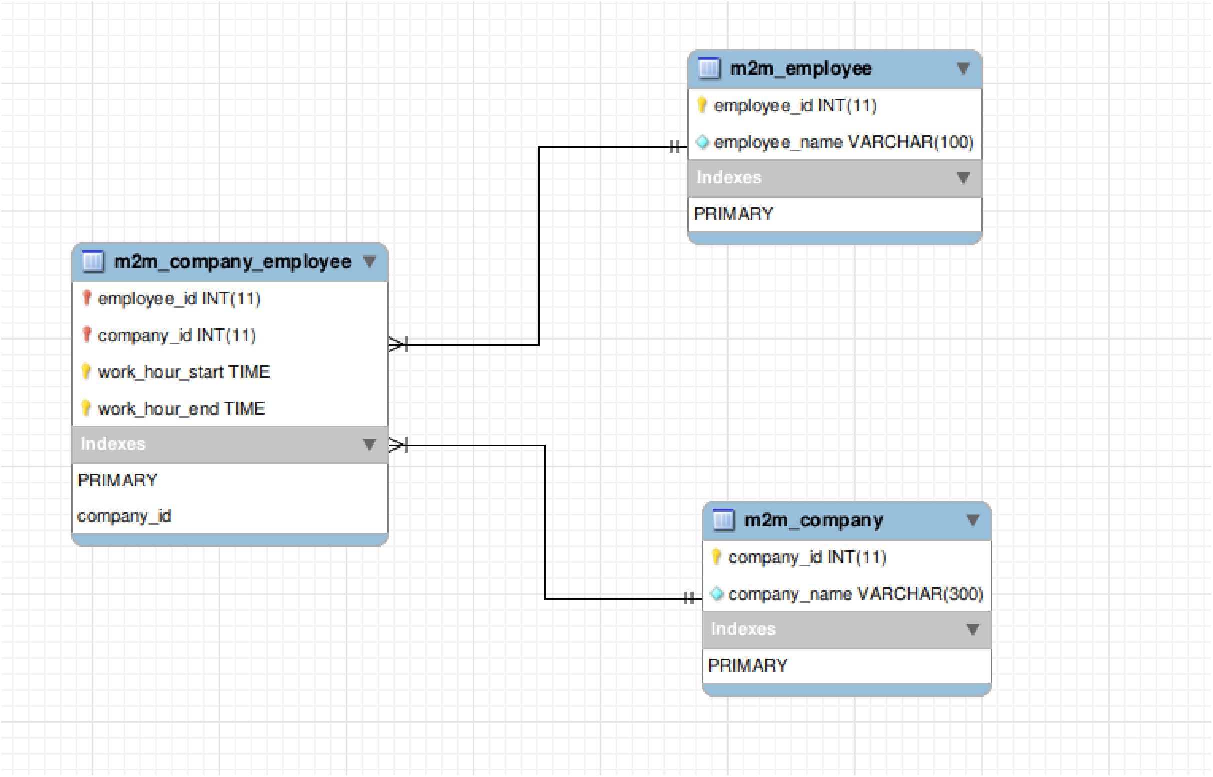
Indexes
- PRIMARY
- user_id

ONE TO MANY



```
CREATE TABLE users_one_to_many(
    id INT NOT NULL AUTO_INCREMENT,
    user_name VARCHAR(45) NOT NULL,
    PRIMARY KEY(id)
) ENGINE = InnoDB DEFAULT CHARSET = utf8;

CREATE TABLE orders_one_to_many(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    amount INT,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES users_one_to_many(id)
) ENGINE = InnoDB DEFAULT CHARSET = utf8;
```

MANY TO MANY



m2m_employee
- employee_id INT(11)
- employee_name VARCHAR(100)

Indexes
- PRIMARY

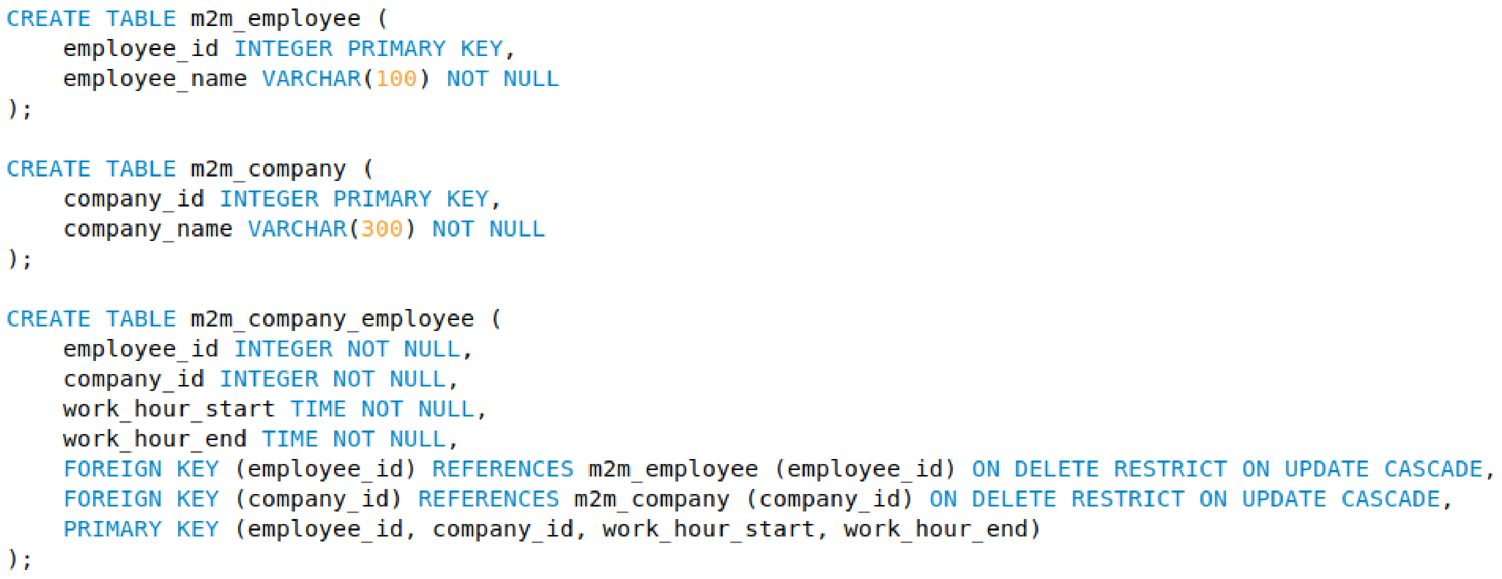
m2m_company_employee
- employee_id INT(11)
- company_id INT(11)
- work_hour_start TIME
- work_hour_end TIME

Indexes
- PRIMARY
- company_id

m2m_company
- company_id INT(11)
- company_name VARCHAR(300)

Indexes
- PRIMARY

MANY TO MANY



```
CREATE TABLE m2m_employee (
    employee_id INTEGER PRIMARY KEY,
    employee_name VARCHAR(100) NOT NULL
);

CREATE TABLE m2m_company (
    company_id INTEGER PRIMARY KEY,
    company_name VARCHAR(300) NOT NULL
);

CREATE TABLE m2m_company_employee (
    employee_id INTEGER NOT NULL,
    company_id INTEGER NOT NULL,
    work_hour_start TIME NOT NULL,
    work_hour_end TIME NOT NULL,
    FOREIGN KEY (employee_id) REFERENCES m2m_employee (employee_id) ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (company_id) REFERENCES m2m_company (company_id) ON DELETE RESTRICT ON UPDATE CASCADE,
    PRIMARY KEY (employee_id, company_id, work_hour_start, work_hour_end)
);
```

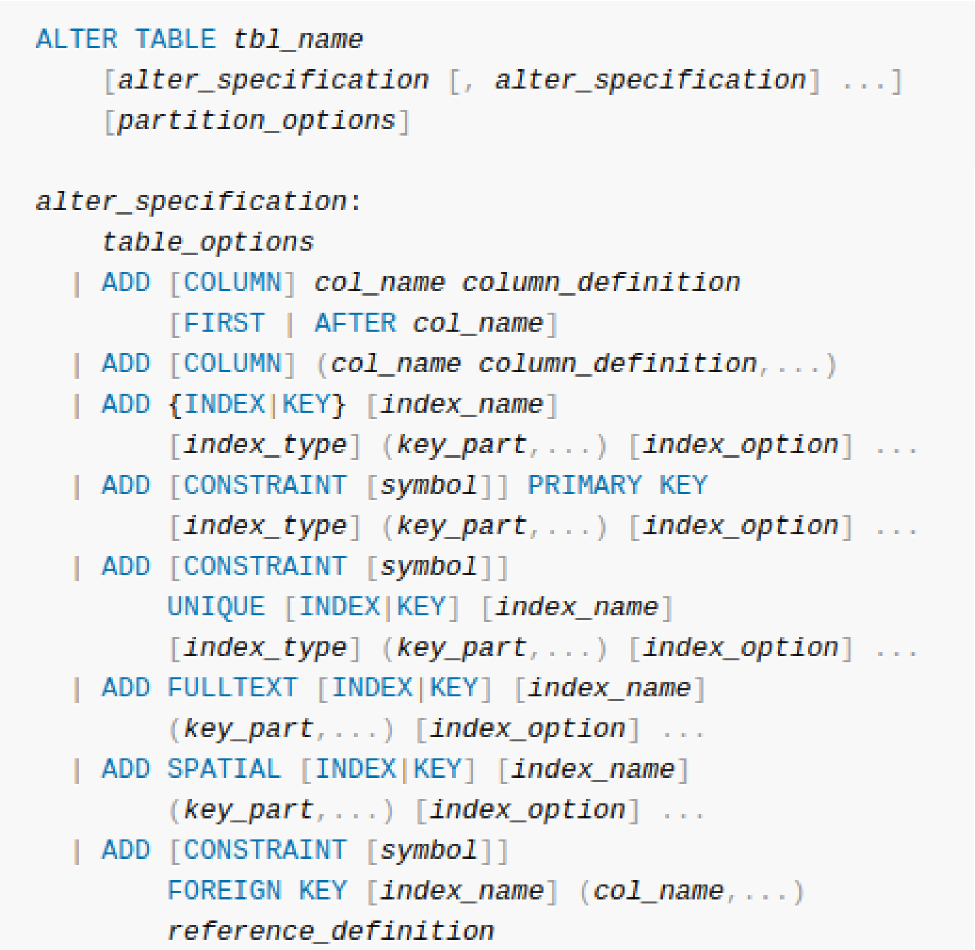
CREATE TABLE Statement Retention

1. SHOW CREATE TABLE tbl_name;

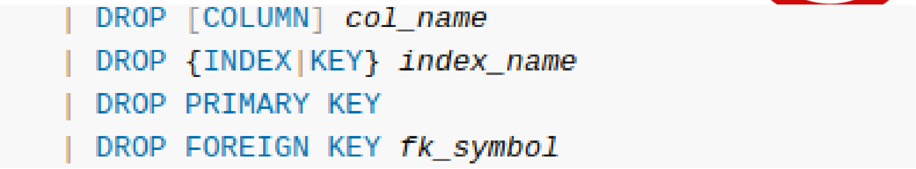
2. DESCRIBE tbl_name;

3. INFORMATION_SCHEMA

ALTER TABLE

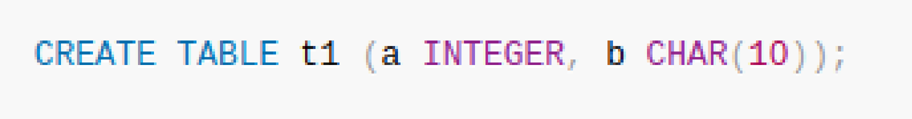


```
ALTER TABLE tbl_name
    [alter_specification [, alter_specification] ...]
    [partition_options]

alter_specification:
    table_options
  | ADD [COLUMN] col_name column_definition
        [FIRST | AFTER col_name]
  | ADD [COLUMN] (col_name column_definition,...)
  | ADD {INDEX|KEY} [index_name]
        [index_type] (key_part,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]] PRIMARY KEY
        [index_type] (key_part,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]]
        UNIQUE [INDEX|KEY] [index_name]
        [index_type] (key_part,...) [index_option] ...
  | ADD FULLTEXT [INDEX|KEY] [index_name]
        (key_part,...) [index_option] ...
  | ADD SPATIAL [INDEX|KEY] [index_name]
        (key_part,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]]
        FOREIGN KEY [index_name] (col_name,...)
        reference_definition
```

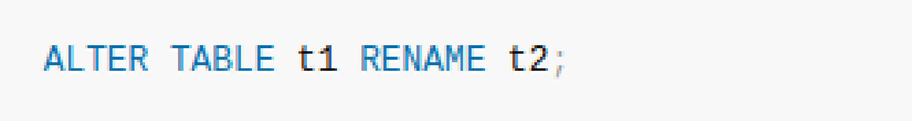


```
  | DROP [COLUMN] col_name
  | DROP {INDEX|KEY} index_name
  | DROP PRIMARY KEY
  | DROP FOREIGN KEY fk_symbol
```

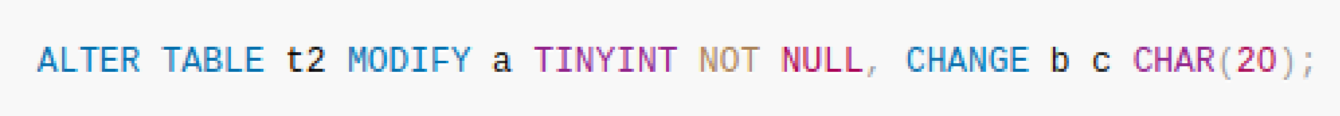
ALTER TABLE



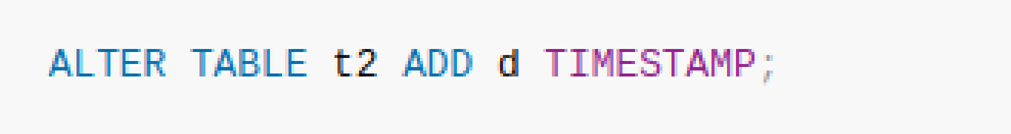
```sql
CREATE TABLE t1 (a INTEGER, b CHAR(10));
```



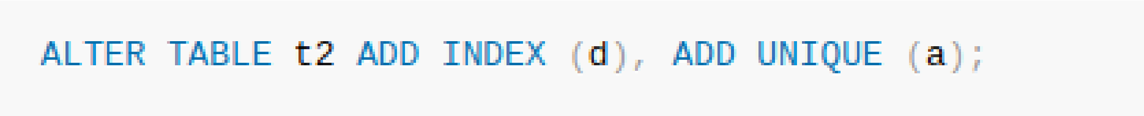
```sql
ALTER TABLE t1 RENAME t2;
```



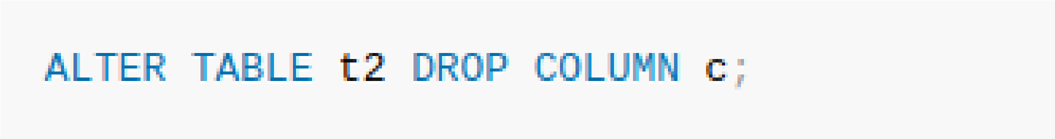
```sql
ALTER TABLE t2 MODIFY a TINYINT NOT NULL, CHANGE b c CHAR(20);
```



```sql
ALTER TABLE t2 ADD d TIMESTAMP;
```

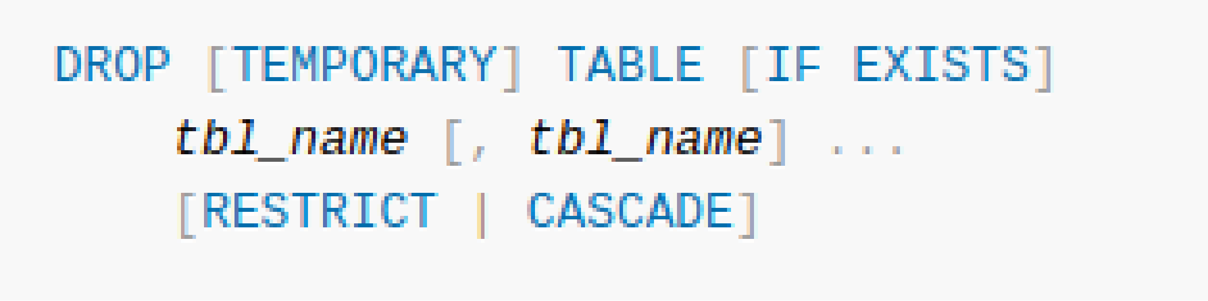


```sql
ALTER TABLE t2 ADD INDEX (d), ADD UNIQUE (a);
```

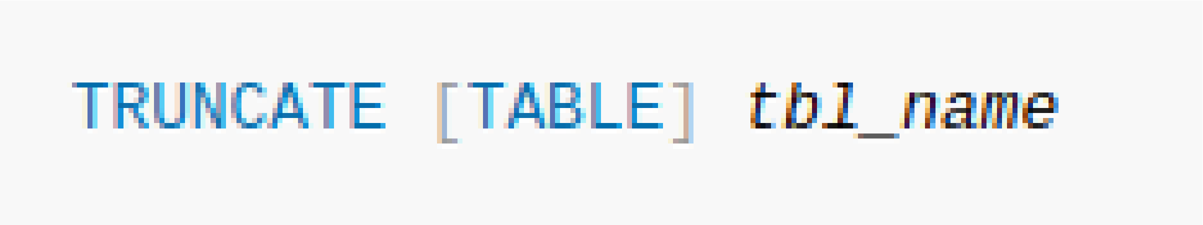


```sql
ALTER TABLE t2 DROP COLUMN c;
```

DROP TABLE, TRUNCATE TABLE

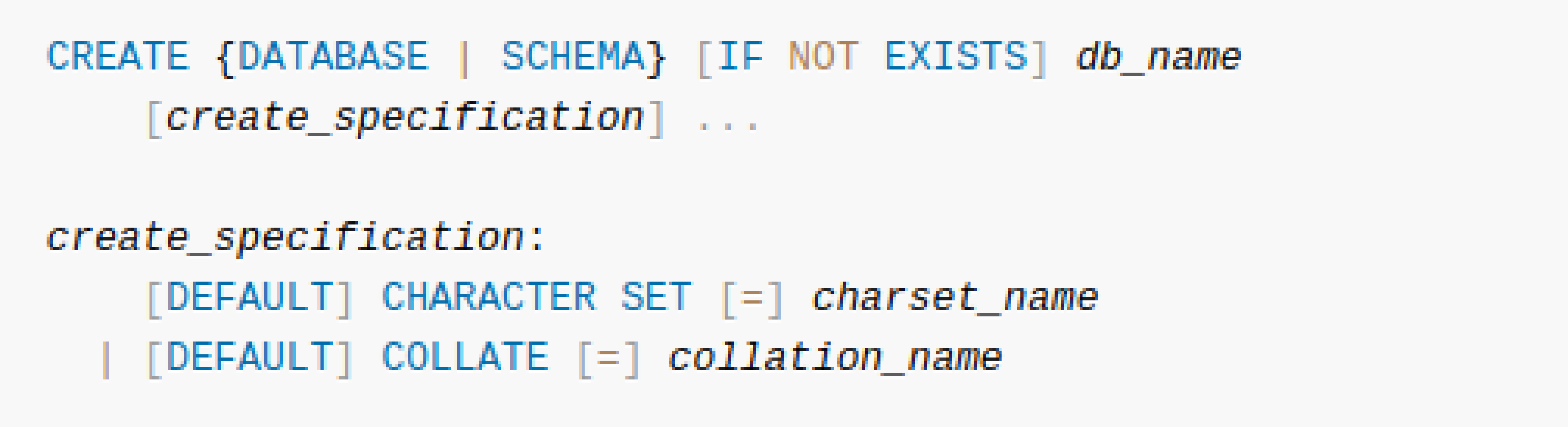


```
DROP [TEMPORARY] TABLE [IF EXISTS]
    tbl_name [, tbl_name] ...
    [RESTRICT | CASCADE]
```



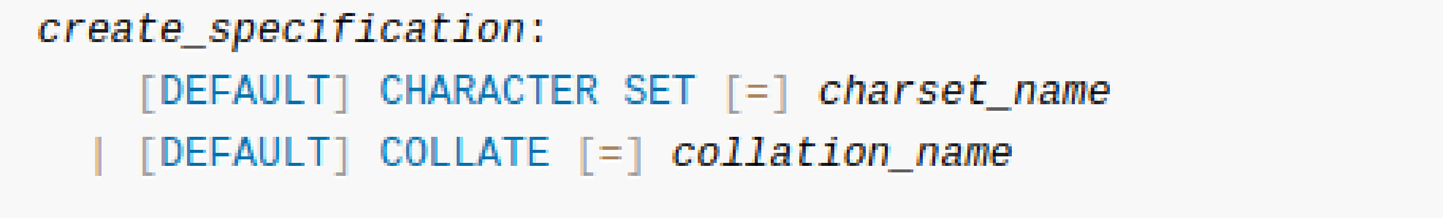
```
TRUNCATE [TABLE] tbl_name
```

CREATE DATABASE



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

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

CHARACTER SET, COLLATE



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

CREATE DATABASE `pets` DEFAULT CHARACTER SET **utf8mb4** COLLATE **utf8mb4_unicode_ci**;

Основы

1. SHOW DATABASES;

2. USE db_name

3. SELECT DATABASE();

Identifiers

Identifiers - Certain objects within MySQL, including database, table, index, column, alias and other object names are known as identifiers.

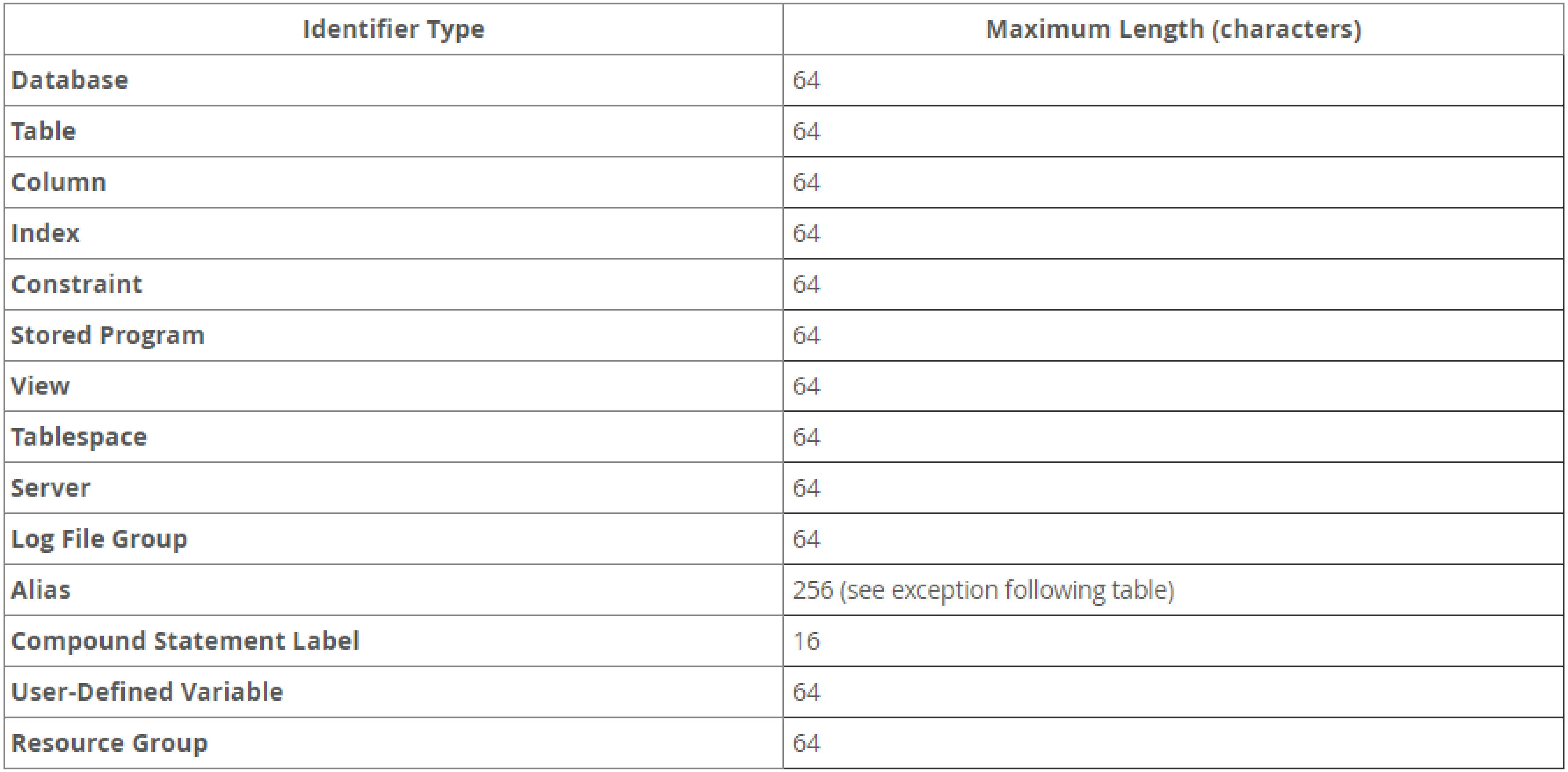
1) Quoted

```
SELECT * FROM `select` WHERE `select`.id > 100;
```

2) Unquoted

```
SELECT * FROM pets WHERE id > 100;
```

Identifiers

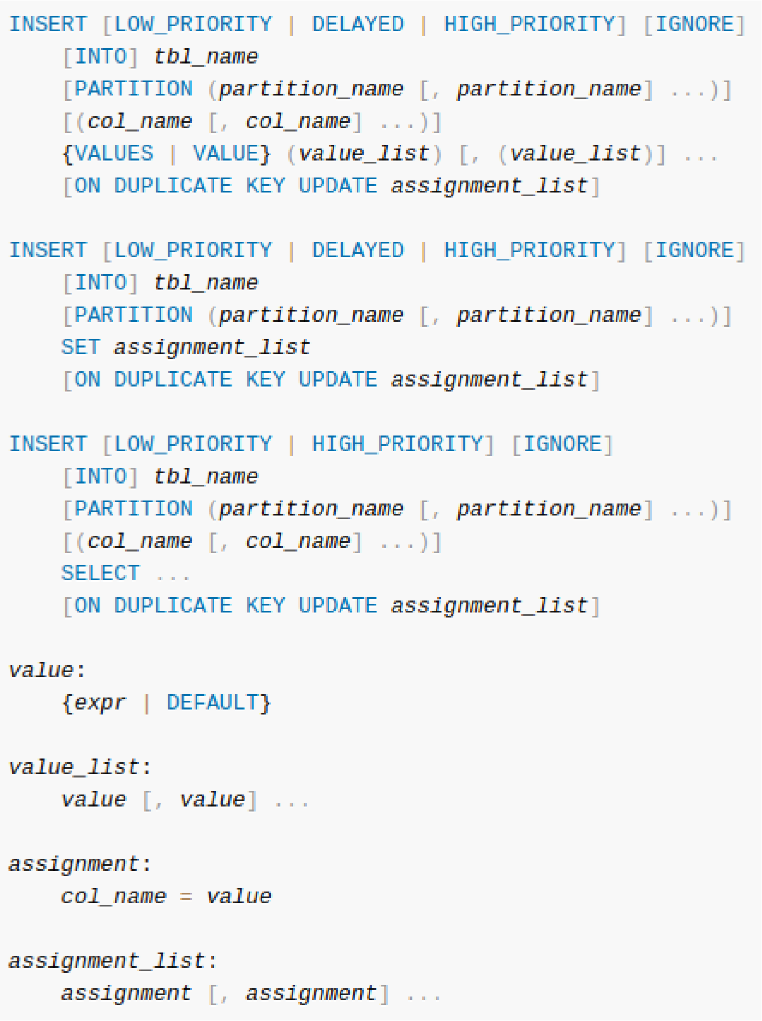


Identifier Type	Maximum Length (characters)
Database	64
Table	64
Column	64
Index	64
Constraint	64
Stored Program	64
View	64
Tablespace	64
Server	64
Log File Group	64
Alias	256 (see exception following table)
Compound Statement Label	16
User-Defined Variable	64
Resource Group	64

DML- Data Manipulation Statements

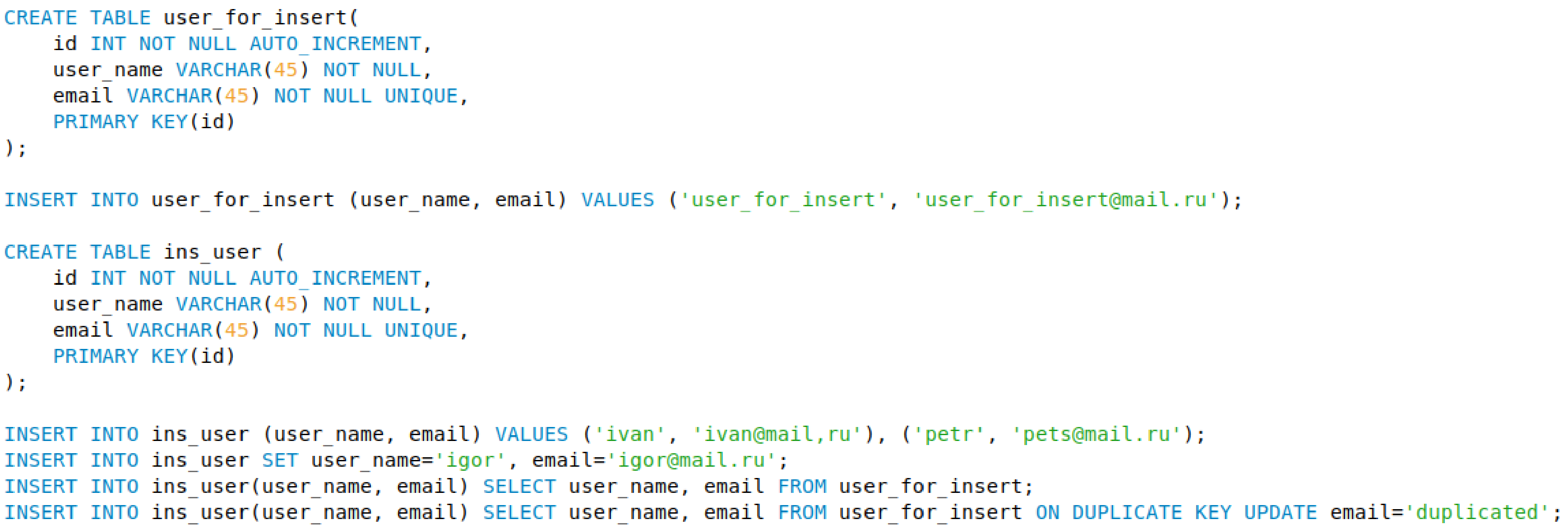
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - Delete records from a database table
- SELECT - retrieve data from a database

INSERT



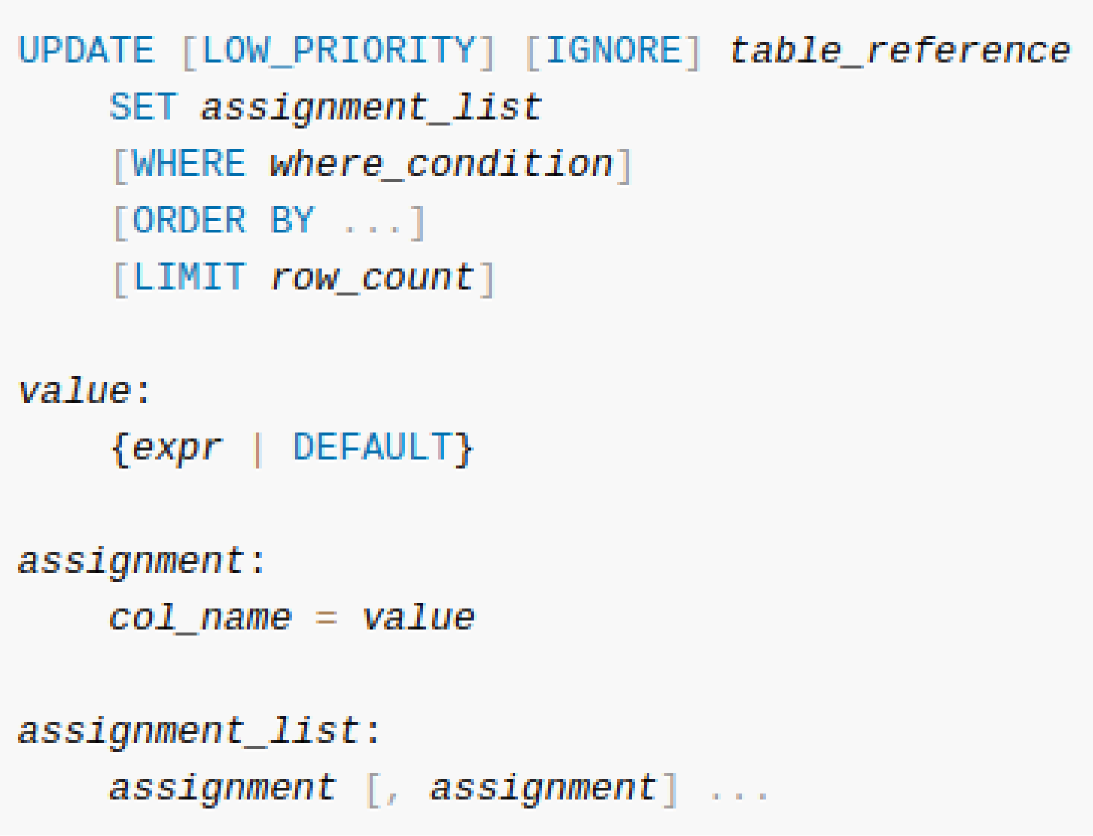
```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name
    [PARTITION (partition_name [, partition_name] ...)]
    [(col_name [, col_name] ...)]
    {VALUES | VALUE} (value_list) [, (value_list)] ...
    [ON DUPLICATE KEY UPDATE assignment_list]

INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name
    [PARTITION (partition_name [, partition_name] ...)]
    SET assignment_list
    [ON DUPLICATE KEY UPDATE assignment_list]

INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name
    [PARTITION (partition_name [, partition_name] ...)]
    [(col_name [, col_name] ...)]
    SELECT ...
    [ON DUPLICATE KEY UPDATE assignment_list]

value:
    {expr | DEFAULT}

value_list:
    value [, value] ...

assignment:
    col_name = value

assignment_list:
    assignment [, assignment] ...
```

INSERT



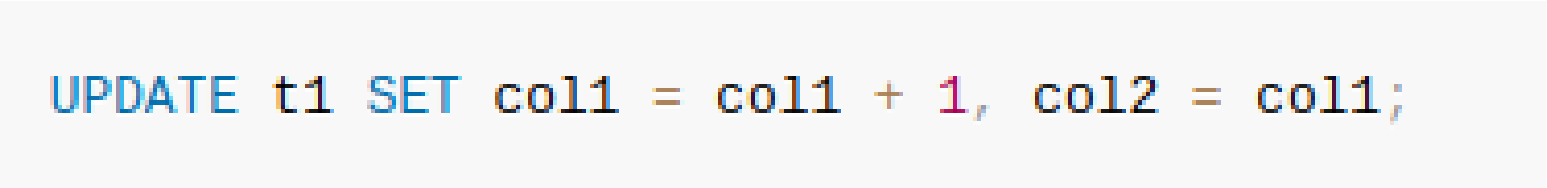
```sql
CREATE TABLE user_for_insert(
    id INT NOT NULL AUTO_INCREMENT,
    user_name VARCHAR(45) NOT NULL,
    email VARCHAR(45) NOT NULL UNIQUE,
    PRIMARY KEY(id)
);

INSERT INTO user_for_insert (user_name, email) VALUES ('user_for_insert', 'user_for_insert@mail.ru');

CREATE TABLE ins_user (
    id INT NOT NULL AUTO_INCREMENT,
    user_name VARCHAR(45) NOT NULL,
    email VARCHAR(45) NOT NULL UNIQUE,
    PRIMARY KEY(id)
);

INSERT INTO ins_user (user_name, email) VALUES ('ivan', 'ivan@mail,ru'), ('petr', 'pets@mail.ru');
INSERT INTO ins_user SET user_name='igor', email='igor@mail.ru';
INSERT INTO ins_user(user_name, email) SELECT user_name, email FROM user_for_insert;
INSERT INTO ins_user(user_name, email) SELECT user_name, email FROM user_for_insert ON DUPLICATE KEY UPDATE email='duplicated';
```

UPDATE



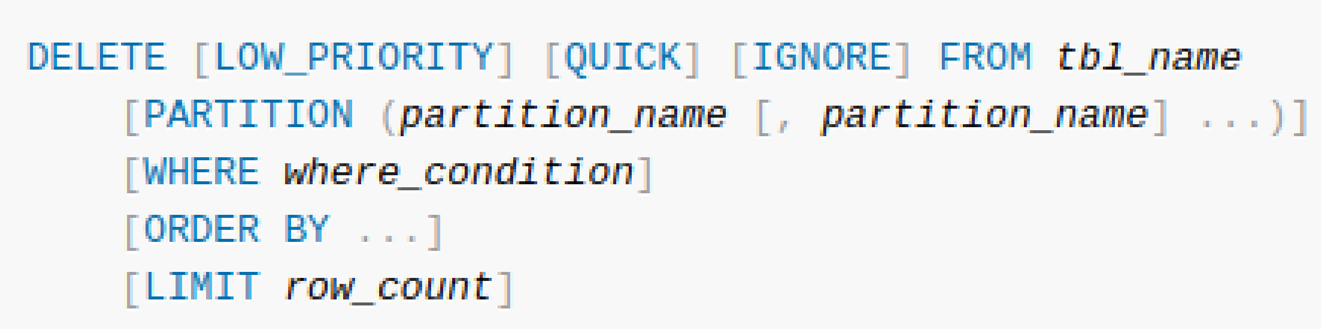
```
UPDATE [LOW_PRIORITY] [IGNORE] table_reference
    SET assignment_list
    [WHERE where_condition]
    [ORDER BY ...]
    [LIMIT row_count]

value:
    {expr | DEFAULT}

assignment:
    col_name = value

assignment_list:
    assignment [, assignment] ...
```

UPDATE



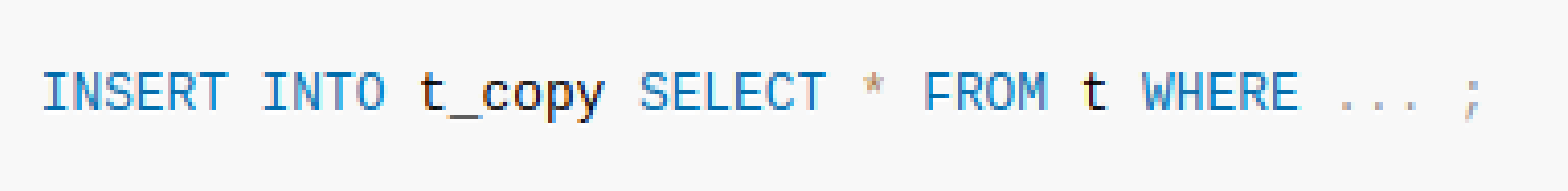
```
UPDATE t1 SET col1 = col1 + 1, col2 = col1;
```

DELETE

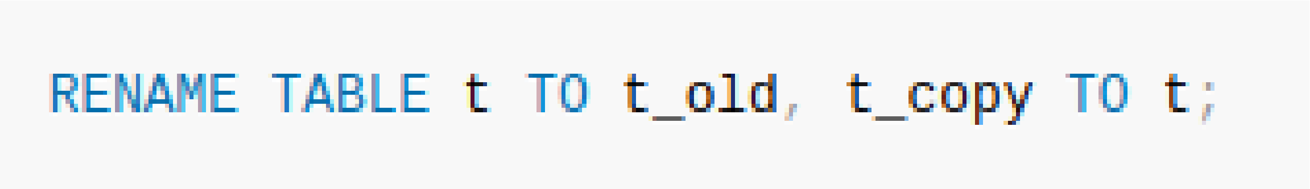


```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name
    [PARTITION (partition_name [, partition_name] ...)]
    [WHERE where_condition]
    [ORDER BY ...]
    [LIMIT row_count]
```

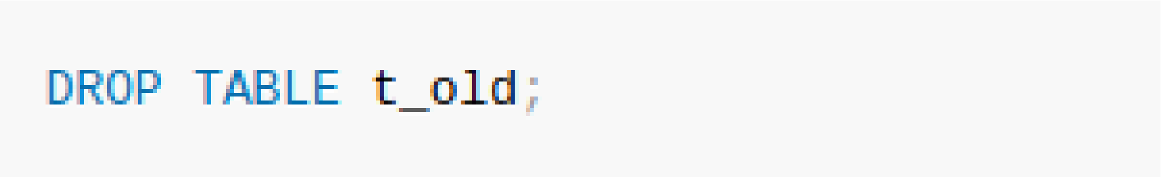
DELETE MANY ROWS (INNO DB)



```
INSERT INTO t_copy SELECT * FROM t WHERE ... ;
```

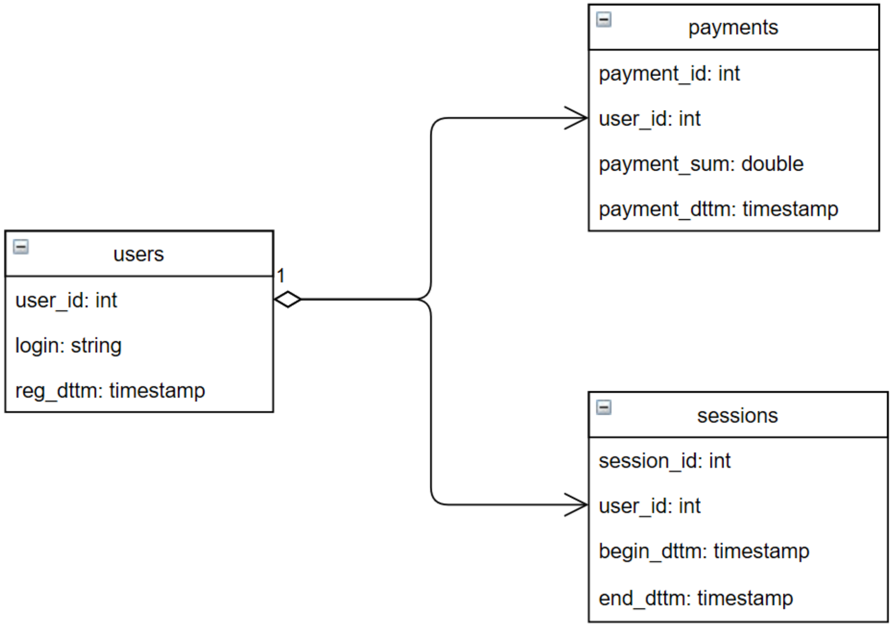


```
RENAME TABLE t TO t_old, t_copy TO t;
```



```
DROP TABLE t_old;
```

Домашнее задание №2



Срок сдачи

03.10.18

ТЕХНОТРЕК

Спасибо за внимание!