

# Создание таблиц (SQL DDL)

Таблицы в базе данных создаются с помощью оператора CREATE. Синтаксис для его использования:

```
CREATE TABLE (  
    {  
        {имя столбца}  
        {тип данных}  
        [значение по умолчанию]  
        [список правил целостности]  
    }+  
)
```

# Типы данных (наиболее распространенные)

- целочисленные значения;
- вещественные значения;
- строки фиксированной длины;
- строки переменной длины;
- дата и время.

# Целочисленные значения

TINYINT 1 байт

SMALLINT 2 байта

INT 4 байта

BIGINT 8 байт

# Вещественные числа

- `FLOAT` 4 байта (`DOUBLE`)  
(числа с плавающей точкой).
- `DECIMAL[(M[,D])]` — неупакованное число с плавающей точкой (хранится в виде строки).
- Для значений, которые будут использованы для вычислений, лучше использовать `FLOAT`.
- Для значений, которые будут использованы для сравнений с другими значениями, лучше использовать `DECIMAL`.

# Строковый тип данных

- Char(n) при хранении дополняются справа пробелами до заданной длины n, если фактическая длина строки меньше n.
- Varchar(n) поле переменной длины — вместе с содержательными данными хранится длина строки, пробелами не дополняется.
- $1 \leq n \leq 255$

# Дата-время

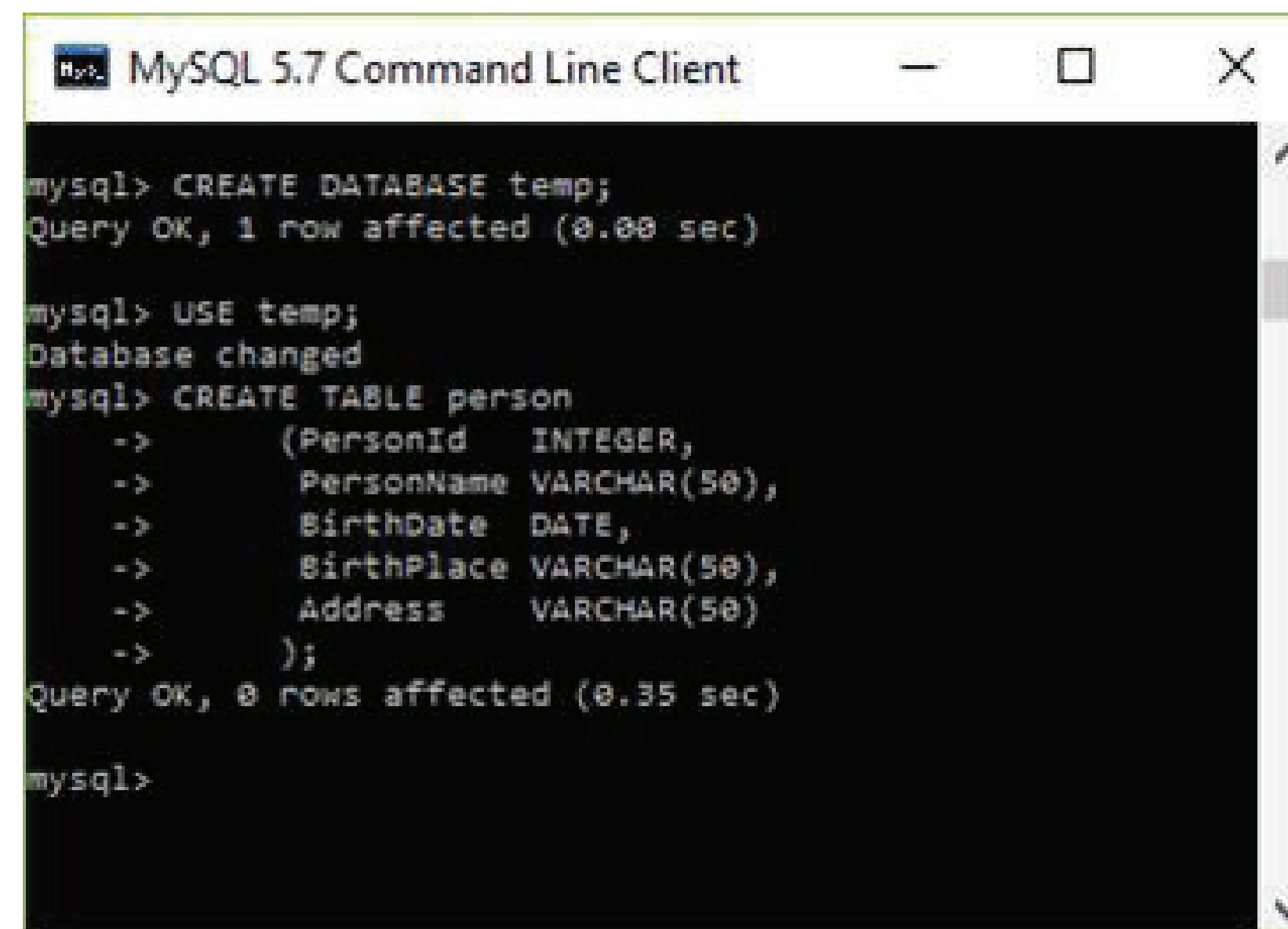
- DATETIME, DATE, TIMESTAMP, TIME и YEAR
- 'YYYY-MM-DD HH:MM:SS'
- Тип столбца **TIMESTAMP** - позволяет автоматически сохранять время изменения/добавления строки.

# Типы данных MySQL

- Реализация SQL в контексте СУБД MySQL (<http://dev.mysql.com/downloads/mysql/>).
- Простейшие типы данных:
  - INTEGER — целые значения;
  - REAL — вещественные значения;
  - CHAR(N) — строки фиксированной длины (n символов);
  - VARCHAR(N) — строки переменной длины (максимально — n символов);
  - DATE — дата и время.

# Пример: создание таблицы (в среде MYSQL)

```
CREATE DATABASE temp;  
USE temp;  
CREATE TABLE person  
    (PersonId    INTEGER,  
     PersonName  VARCHAR(50) ,  
     BirthDate   DATE ,  
     BirthPlace  VARCHAR(50) ,  
     Address     VARCHAR(50)  
    ) ;
```



```
MySQL 5.7 Command Line Client  
mysql> CREATE DATABASE temp;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> USE temp;  
Database changed  
mysql> CREATE TABLE person  
->    (PersonId    INTEGER,  
->     PersonName  VARCHAR(50),  
->     BirthDate   DATE,  
->     BirthPlace  VARCHAR(50),  
->     Address     VARCHAR(50)  
->    );  
Query OK, 0 rows affected (0.35 sec)  
  
mysql>
```



# Просмотр структуры созданной таблицы

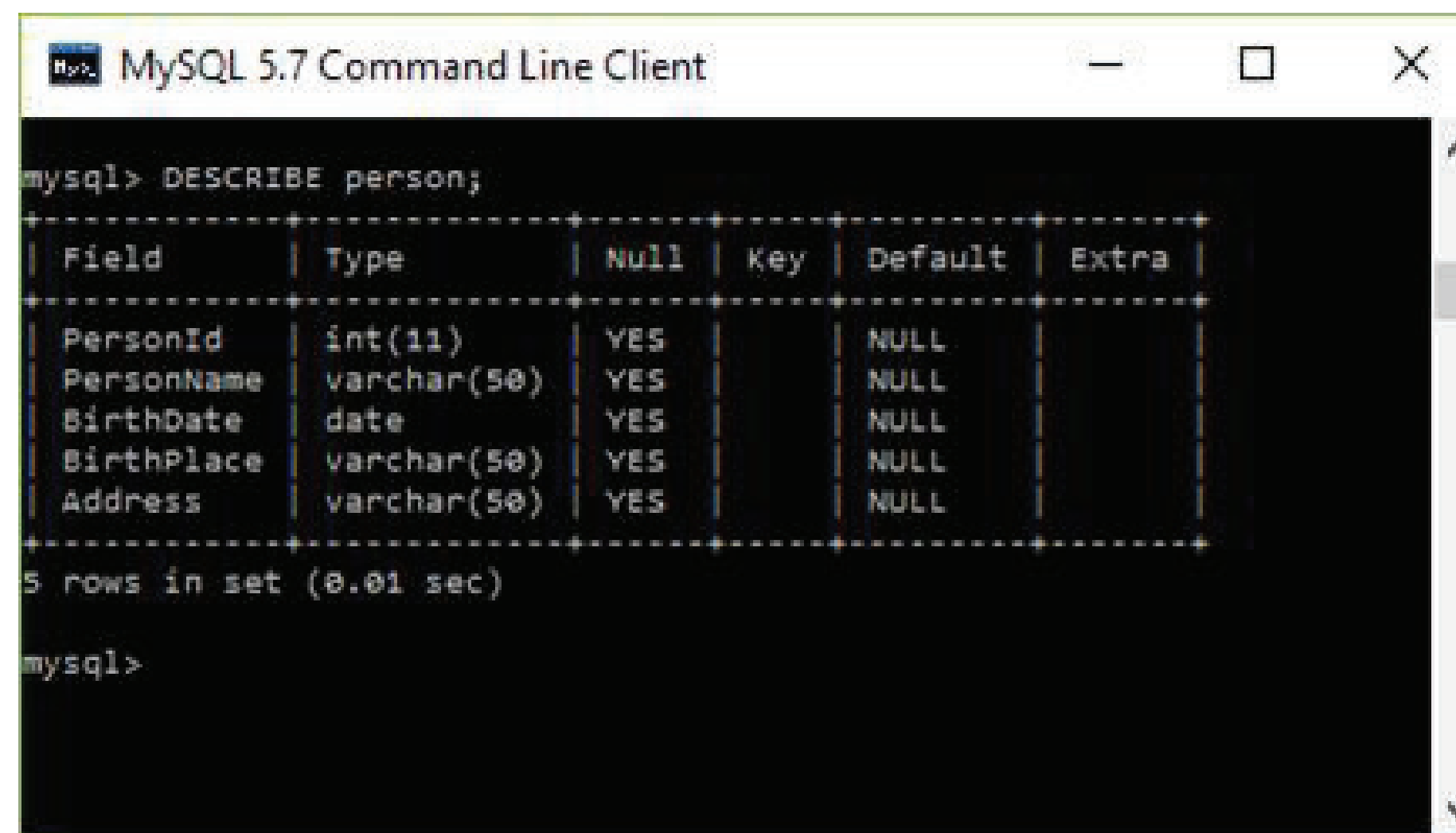
Оператор DESCRIBE.

Синтаксис:

```
DESCRIBE <имя таблицы>
```

# Пример: выполнение команды DESCRIBE

`DESCRIBE person;`



```
mysql> DESCRIBE person;
```

Field	Type	Null	Key	Default	Extra
PersonId	int(11)	YES		NULL	
PersonName	varchar(50)	YES		NULL	
BirthDate	date	YES		NULL	
BirthPlace	varchar(50)	YES		NULL	
Address	varchar(50)	YES		NULL	

```
5 rows in set (0.01 sec)

mysql>
```

# Модификация структуры таблицы

Оператор ALTER TABLE.

Упрощенный синтаксис:

```
ALTER TABLE <имя таблицы>
{{ ADD | DROP | MODIFY } COLUMN
    имя столбца
    тип
    [значение по умолчанию]
    список правил целостности]
}+
```

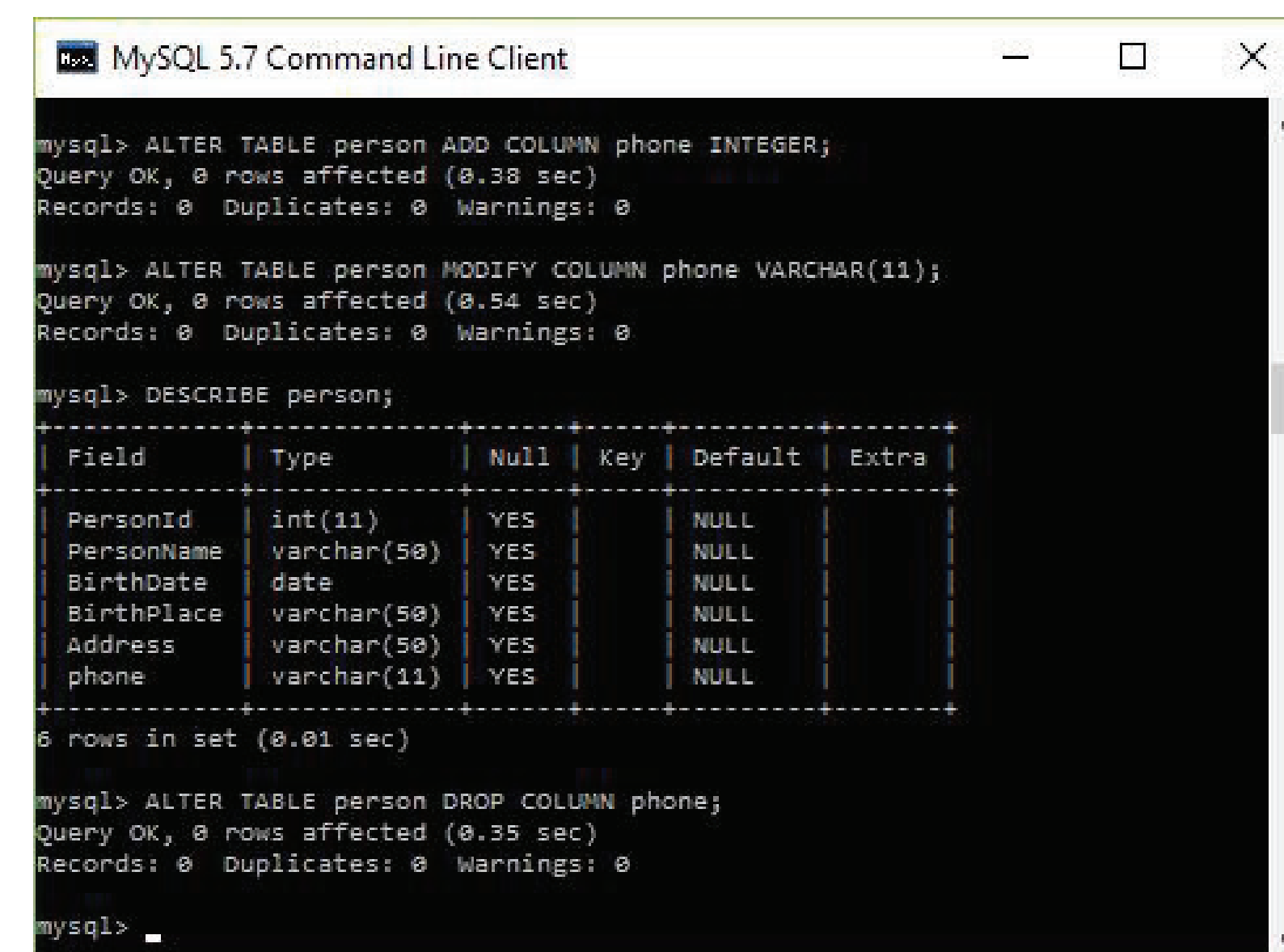
# Пример: модификация структуры таблицы

```
ALTER TABLE person ADD COLUMN phone INTEGER;
```

```
ALTER TABLE person MODIFY COLUMN phone  
VARCHAR(11);
```

```
DESCRIBE person;
```

```
ALTER TABLE person DROP COLUMN phone;
```



```
MySQL 5.7 Command Line Client

mysql> ALTER TABLE person ADD COLUMN phone INTEGER;
Query OK, 0 rows affected (0.38 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE person MODIFY COLUMN phone VARCHAR(11);
Query OK, 0 rows affected (0.54 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESCRIBE person;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| PersonId   | int(11)   | YES  |     | NULL    |       |
| PersonName | varchar(50)| YES  |     | NULL    |       |
| BirthDate  | date      | YES  |     | NULL    |       |
| BirthPlace | varchar(50)| YES  |     | NULL    |       |
| Address    | varchar(50)| YES  |     | NULL    |       |
| phone      | varchar(11)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

mysql> ALTER TABLE person DROP COLUMN phone;
Query OK, 0 rows affected (0.35 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

# Удаление таблиц

- Оператора DROP.
- Синтаксис для его использования:

```
DROP TABLE <имя таблицы>
```

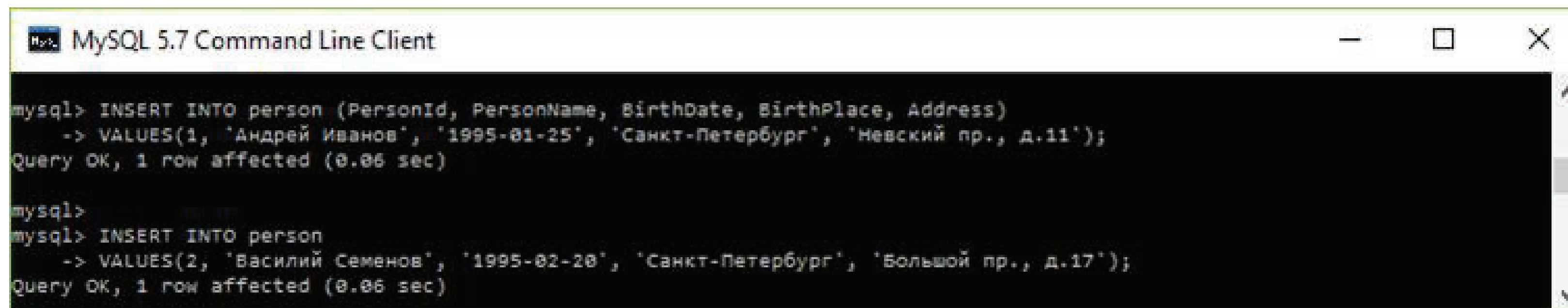
# Добавление записей в таблицу

- Оператор INSERT INTO.
- Синтаксис для его использования:

```
INSERT INTO <table-name> [(column1, ..., columnn)]  
                        values (value1, ..., valuen)  
INSERT INTO <table-name> [(column1, ..., columnn)]  
                        <select-statement>
```

# Пример: добавление записей в таблицу

```
INSERT INTO person (PersonId, PersonName, BirthDate, BirthPlace, Address)
VALUES (1, 'Андрей Иванов', '1995-01-25', 'Санкт-Петербург', 'Невский пр., д.11');
INSERT INTO person
VALUES (2, 'Василий Семенов', '1995-02-20', 'Санкт-Петербург', 'Большой пр., д.17');
```



```
MySQL 5.7 Command Line Client

mysql> INSERT INTO person (PersonId, PersonName, BirthDate, BirthPlace, Address)
-> VALUES(1, 'Андрей Иванов', '1995-01-25', 'Санкт-Петербург', 'Невский пр., д.11');
Query OK, 1 row affected (0.06 sec)

mysql>
mysql> INSERT INTO person
-> VALUES(2, 'Василий Семенов', '1995-02-20', 'Санкт-Петербург', 'Большой пр., д.17');
Query OK, 1 row affected (0.06 sec)
```

# Удаление записей из таблицы

- Два оператора: DELETE и TRUNCATE.
- Синтаксис для использования:

```
DELETE FROM <имя таблицы> [WHERE <condition>]
```

```
TRUNCATE TABLE <имя таблицы>
```



# Пример: удаления записей из таблицы

```
DELETE FROM person WHERE PersonId = 1;  
TRUNCATE TABLE person;
```



```
MySQL 5.7 Command Line Client  
mysql> DELETE FROM person WHERE PersonId = 1;  
Query OK, 1 row affected (0.06 sec)  
  
mysql> TRUNCATE TABLE person;  
Query OK, 0 rows affected (0.20 sec)  
  
mysql> _
```

# Редактирование записей в таблице

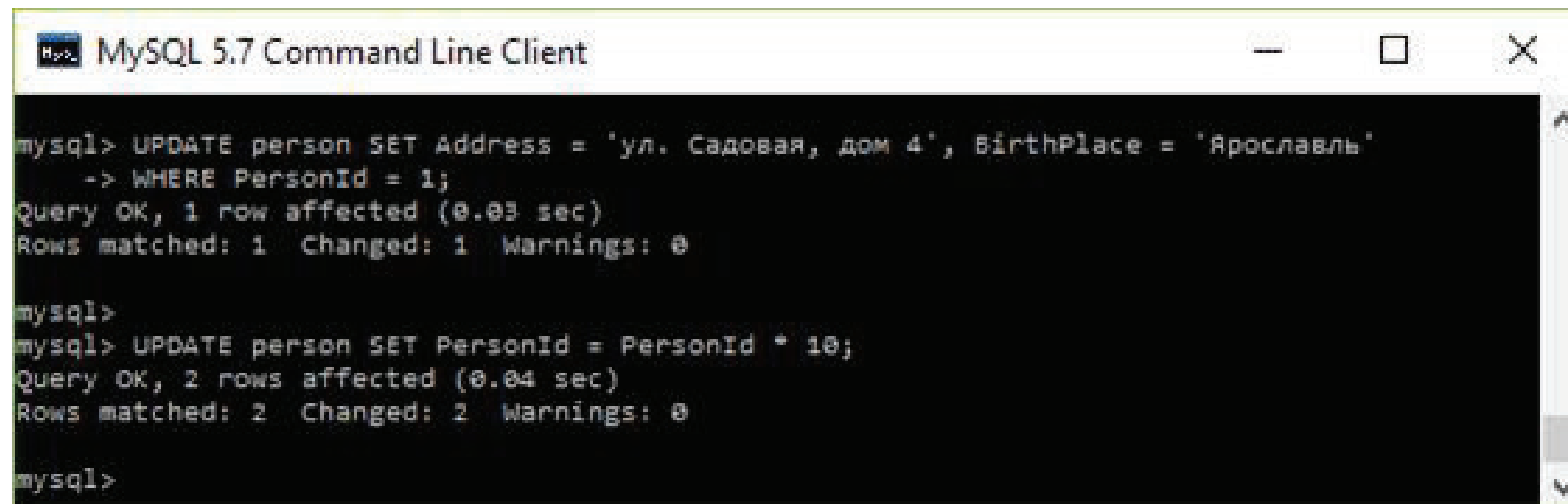
- Оператор UPDATE.
- Синтаксис:

```
UPDATE <имя таблицы>  
    SET column1 = expression1,  
        . . .  
        columnm = expressionm  
[WHERE <condition>]
```

# Пример: редактирование записей в таблице

```
UPDATE person SET Address = 'ул. Садовая, дом 4', BirthPlace = 'Ярославль'  
WHERE PersonId = 1;
```

```
UPDATE person SET PersonId = PersonId * 10;
```



```
MySQL 5.7 Command Line Client  
mysql> UPDATE person SET Address = 'ул. Садовая, дом 4', BirthPlace = 'Ярославль'  
-> WHERE PersonId = 1;  
Query OK, 1 row affected (0.03 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
mysql>  
mysql> UPDATE person SET PersonId = PersonId * 10;  
Query OK, 2 rows affected (0.04 sec)  
Rows matched: 2  Changed: 2  Warnings: 0  
  
mysql>
```