DATABASE PROGRAMING CSC-2113

Lecture 2

Data Definition Language commands(DDL)

Mwavu Rogers
Email: mwavurogers@gmail.com
Phone 0773426328 and 0700497421
Faculty of computing and informatics sciences
DEPARTMENT OF INFORMATION TECHNOLOGY

SUCCEED WE MUST

SUCCEED WE MUST

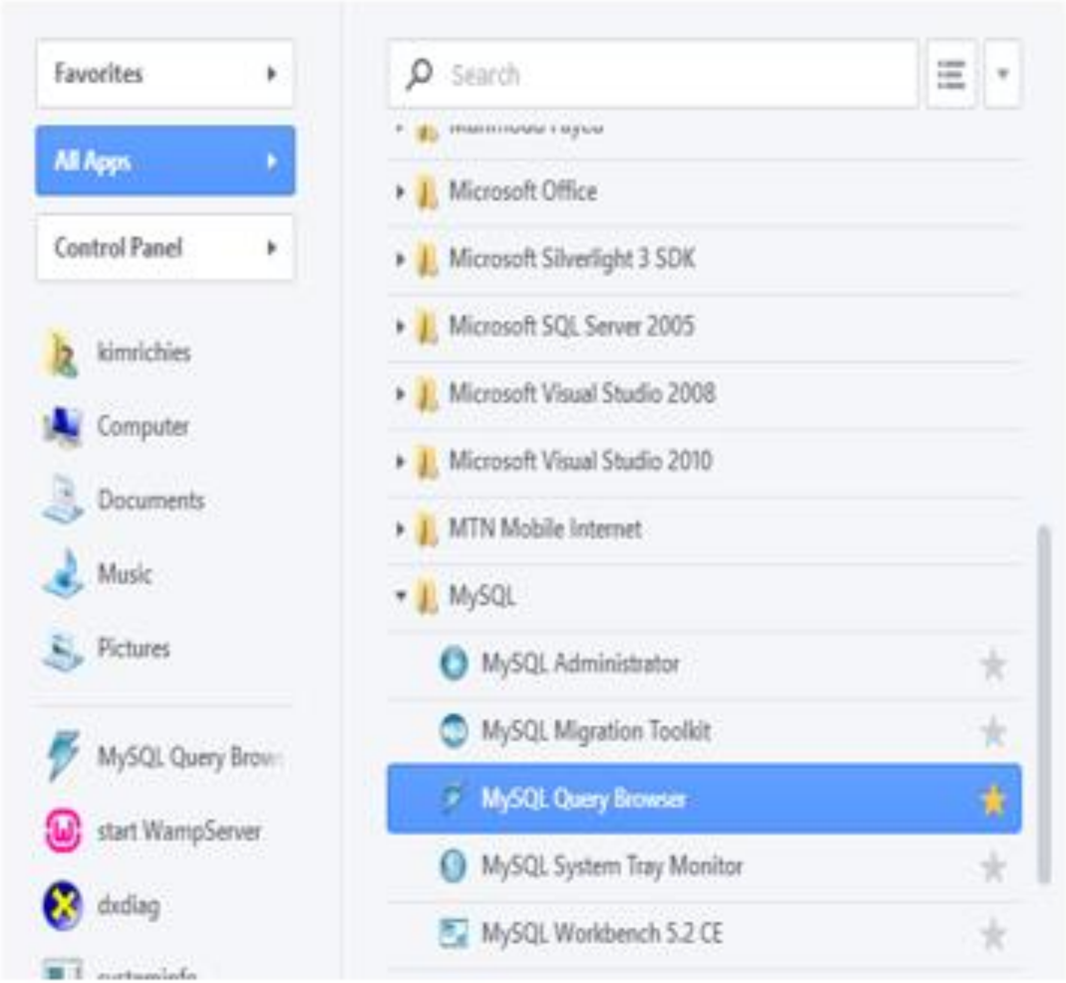
History of SQL

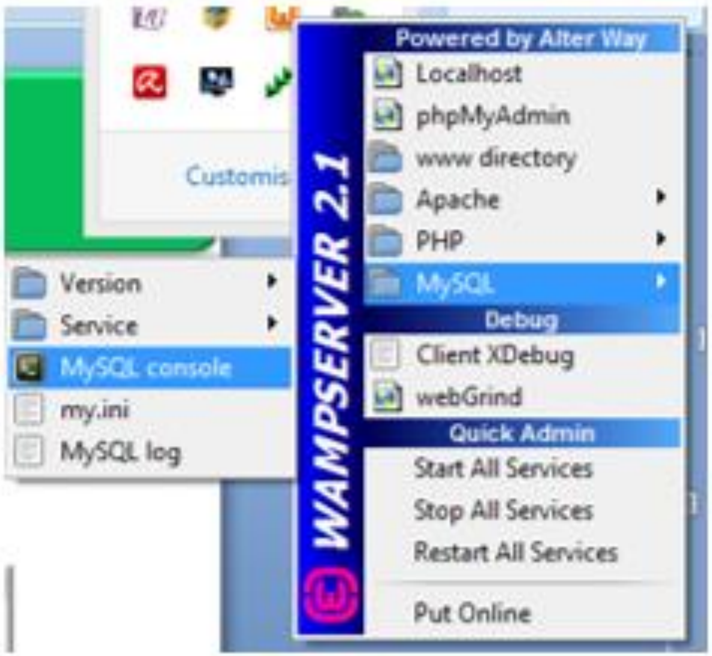
- SQL or **S**tructured **Q**uery **L**anguage was developed in 1970 by IBM as a method of interfacing with **Relational D**ata**b**ase **M**anagement **S**ystems (RDBMS).
- In 1979, Relational Software, Inc. (now Oracle) introduced the first commercially available implementation of SQL
- Today it is the de-facto standard, an ISO standard and ANSI standard.
- This means 99.9% of RDBMS systems utilise SQL as their interfacing language.
- The latest SQL standard was adopted in July 1999 and is often called SQL:99.

Data Definition Language

- Once a Logical and Physical design for a database have been developed, the next step is of course to create the database.
- Of course this can be done with the Tools provided by the vendor but these tools use the commands we shall learn today to Create and Modify the database design.
- You can practice these commands using MySQL and the Command Line Client/MySQL query browser

MySQL and the Command Line Client/MySQL query browser





SQL commands

- SELECT – The most common command. Used to retrieve data from a database
- INSERT, UPDATE, DELETE – These commands are used to enter new rows, change existing rows and remove unwanted rows from tables in the database. They are known as DML or Data Manipulation Language commands.
- CREATE, ALTER, DROP, TRUNCATE – These commands are used to set up, change and remove data structures such as tables. They are known as DDL or Data Definition Language commands
- GRANT , REVOKE – These commands are used to give or remove access rights to the SQL database and the structures within it.

Data Definition Language commands(DDL)

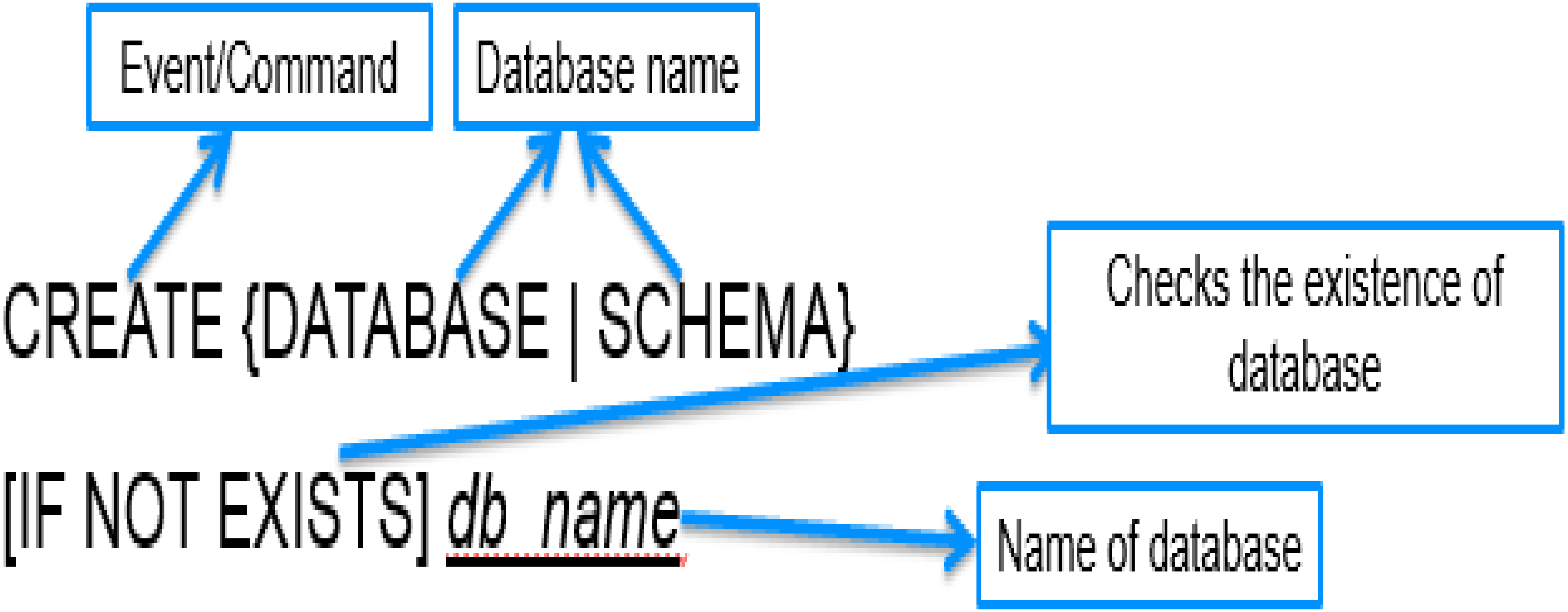
- DDL are commands are used to set up, change and remove data structures such as tables.
- OR
- Data Definition Language commands are commands used for creating and modifying the structure of the database objects, such as schemas, tables, views, indexes, etc. Additionally, it assists in storing the metadata details in the database.

Examples include the following

- CREATE
- ALTER
- DROP
- TRUNCATE

Data Definition Language command - CREATE

- Creating a Database: The syntax for creating a database is as follows:



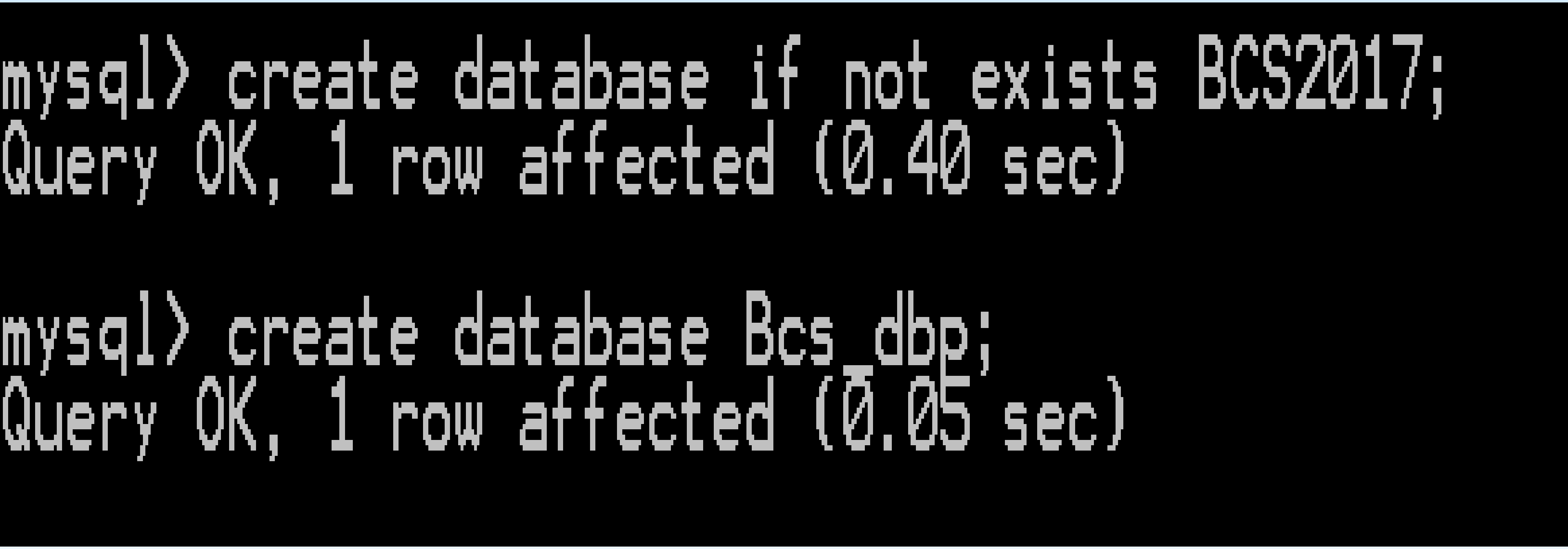
CREATE {DATABASE | SCHEMA}

[IF NOT EXISTS] *db_name*

Using Create SQL command

The **CREATE** statement is used to Create content / objects, this includes

A new Database, **Table**, **View**, **Index** and **Stored Procedure**

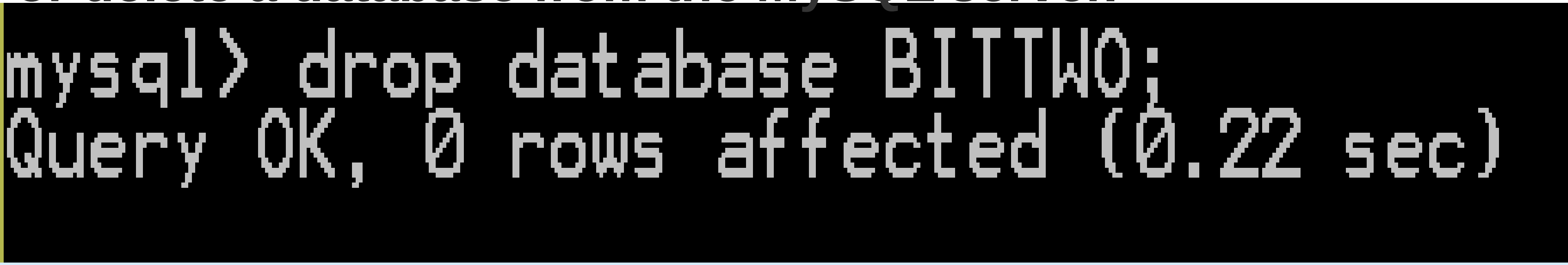


```
mysql> create database if not exists BCS2017;
Query OK, 1 row affected (0.40 sec)

mysql> create database Bcs_dbp;
Query OK, 1 row affected (0.05 sec)
```

SUCCEED WE MUST

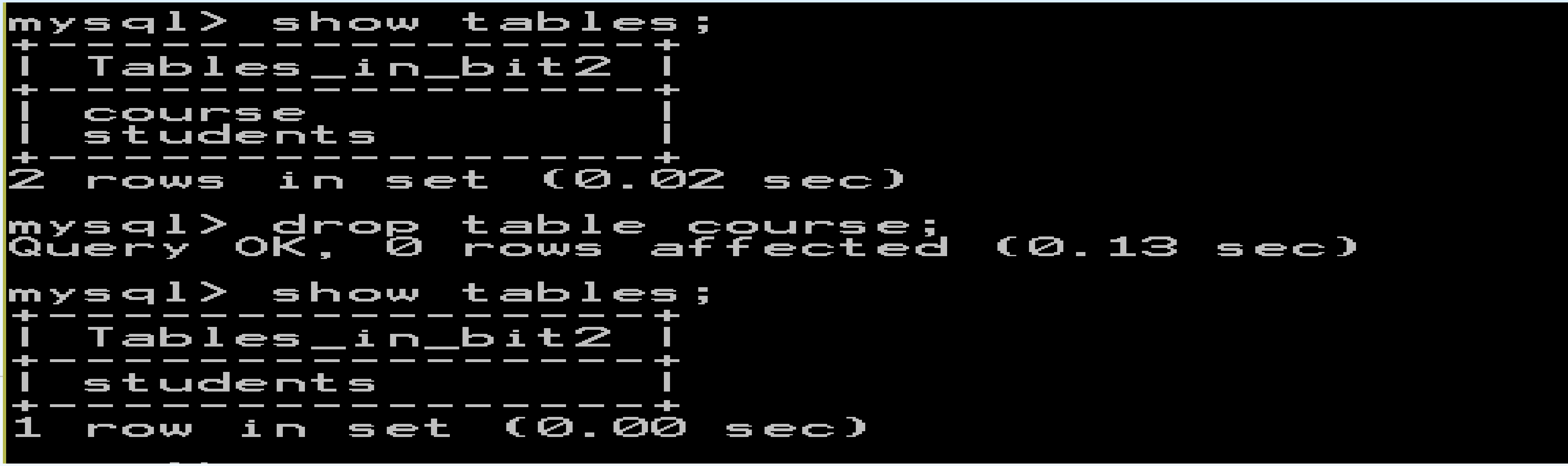
Using Drop SQL Command

The MySQL DROP database statement allows you to remove or delete a database from the MySQL server.



```
mysql> drop database BITTWO;
Query OK, 0 rows affected (0.22 sec)
```

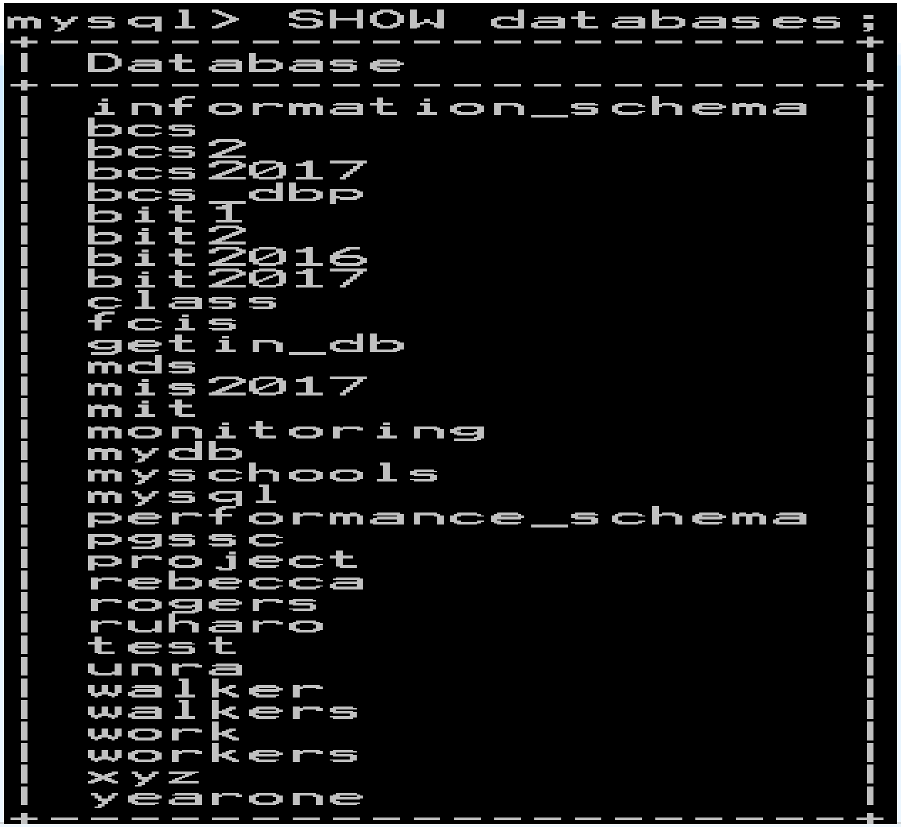
The MySQL DROP TABLE statement allows you to remove or delete a table from the MySQL database.



```
mysql> show tables;
+----------------+
| Tables_in_bit2 |
+----------------+
| course         |
| students       |
+----------------+
2 rows in set (0.02 sec)

mysql> drop table course;
Query OK, 0 rows affected (0.13 sec)

mysql> show tables;
+----------------+
| Tables_in_bit2 |
+----------------+
| students       |
+----------------+
1 row in set (0.00 sec)
```

Use the “Show command” to view the already created databases

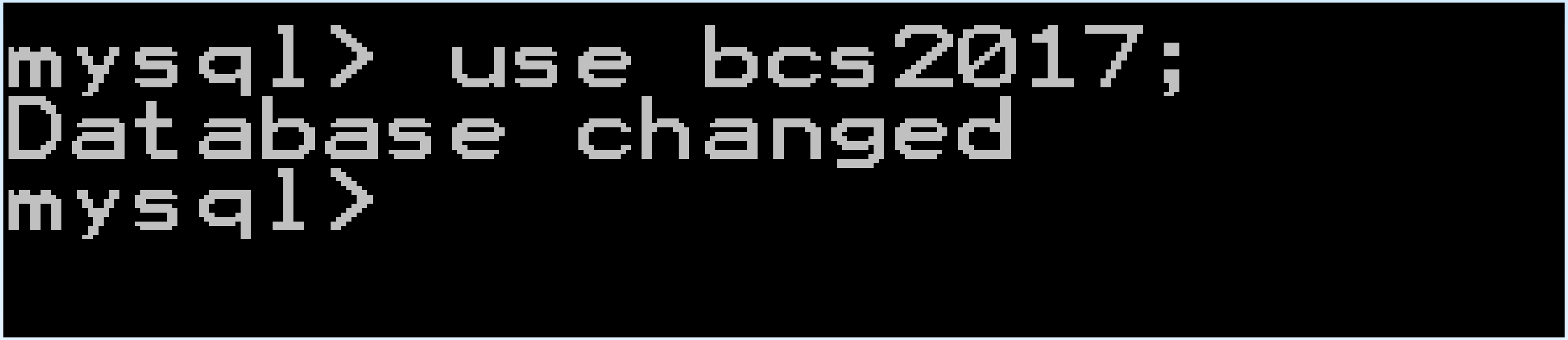
After creating the database Use the “Show command” to view the already created databases



```
mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bcs                |
| bcs2               |
| bcs2017            |
| bcs_dbp            |
| bit1               |
| bit2               |
| bit2016            |
| bit2017            |
| class              |
| fcis               |
| getin_db           |
| mds                |
| mis2017            |
| mit                |
| monitoring         |
| mydb               |
| myschools          |
| mysql              |
| performance_schema |
| pgssc              |
| project            |
| rebecca            |
| rogers             |
| ruharo             |
| test               |
| unra               |
| walker             |
| walkers            |
| work               |
| workers            |
| xyz                |
| yearone            |
+--------------------+
```

Creating a Table

- To create a table, you need to first select the database to use.
- Use the “use database_name” command as below

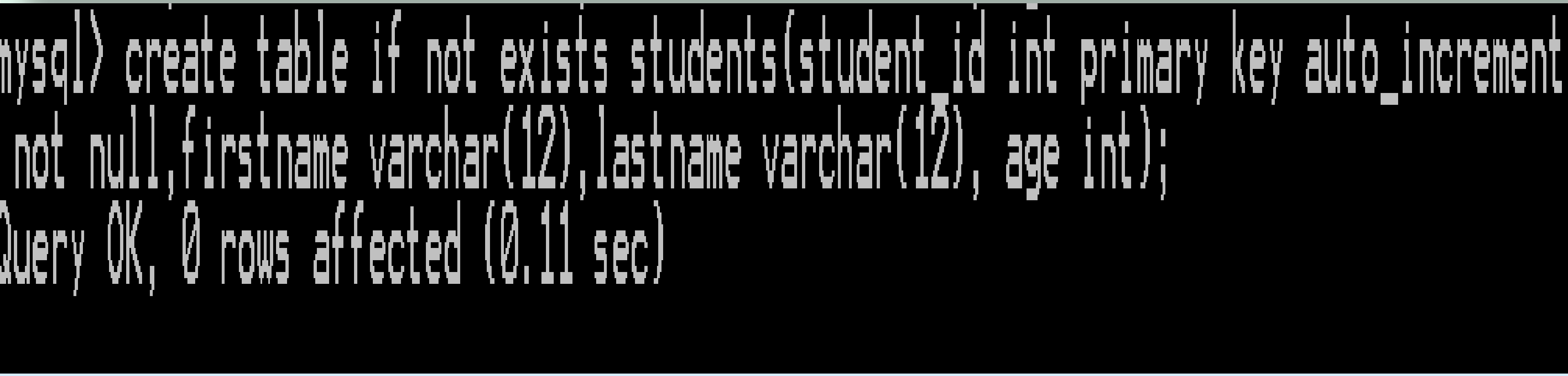


```
mysql> use bcs2017;
Database changed
mysql>
```

Creating a Table: Syntax

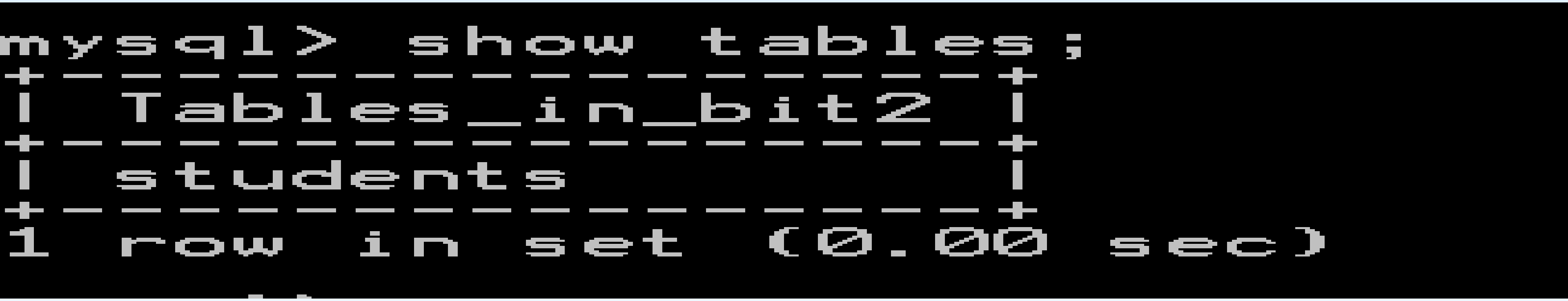
```
CREATE TABLE IF NOT EXISTS database.table_name
 (
        row_1 datatype [NOT NULL | NULL] [AUTO_INCREMENT]
        [UNSIGNED],
        row_2 datatype,
        row_3 datatype etc…
        PRIMARY KEY (row_name)
 )
 ENGINE = database_engine;
```

Example



```
mysql> create table if not exists students(student_id int primary key auto_increment
 not null,firstname varchar(12),lastname varchar(12), age int);
Query OK, 0 rows affected (0.11 sec)
```

We can use the show command to display the tables that are with in the database

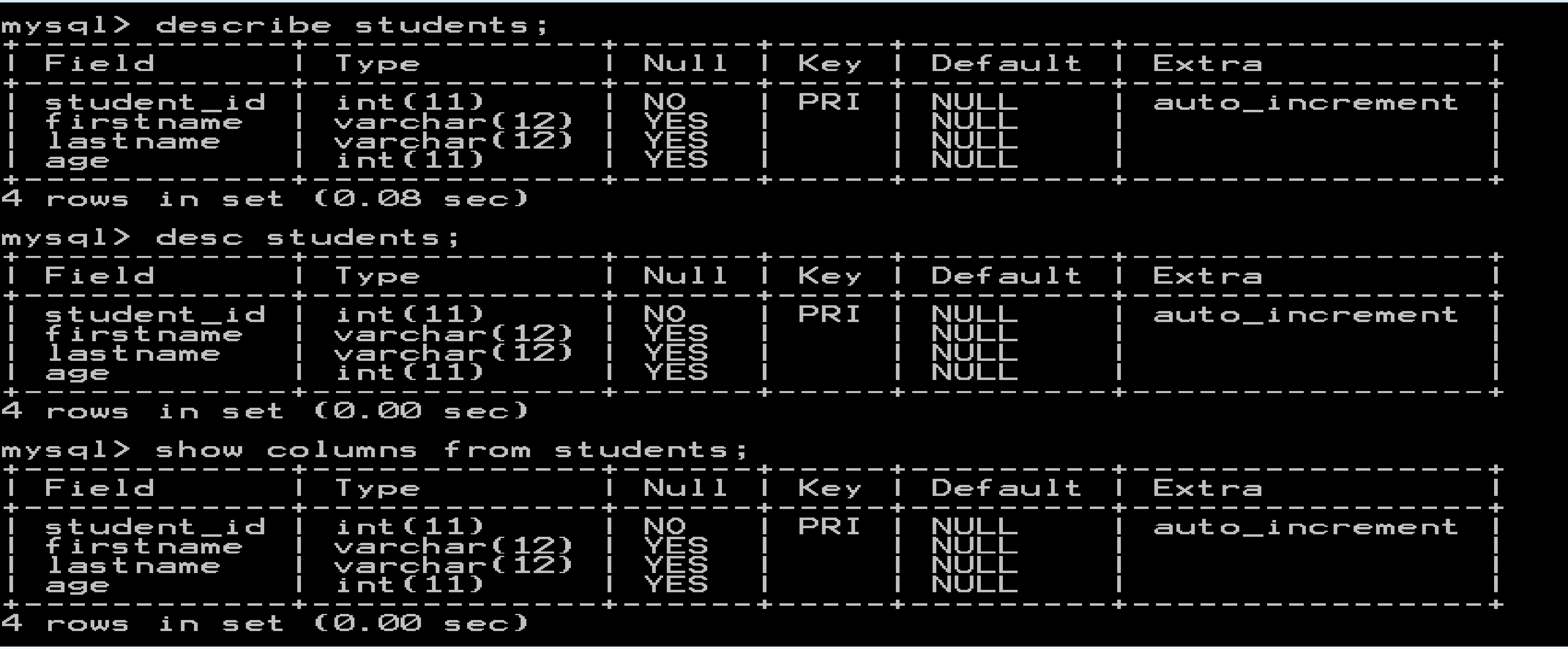


```
mysql> show tables;
+-----------------+
| Tables_in_bit2  |
+-----------------+
| students        |
+-----------------+
1 row in set (0.00 sec)
```

SUCCEED WE MUST

Describing the table

- To describe the table created we can use the following commands
- Describe table_name
- ,desc table_name
- show column from table_name

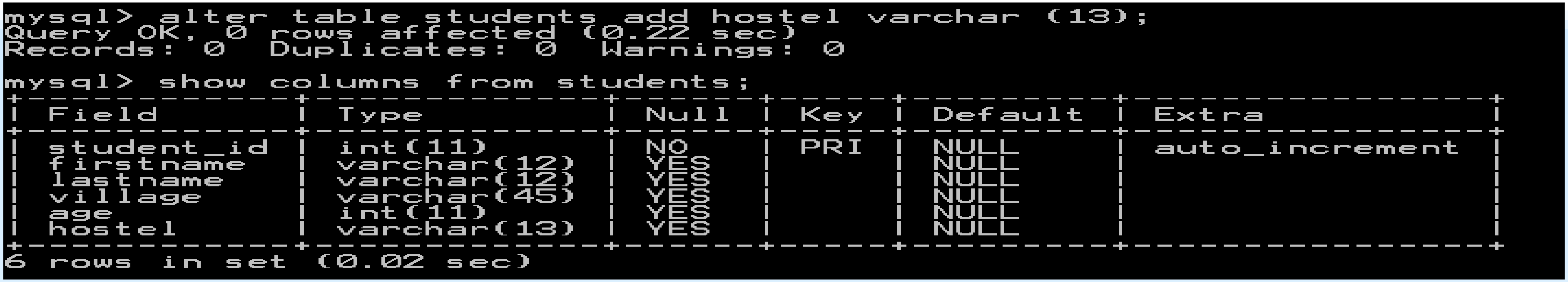


```
mysql> describe students;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| student_id | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname  | varchar(12) | YES  |     | NULL    |                |
| lastname   | varchar(12) | YES  |     | NULL    |                |
| age        | int(11)     | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0.08 sec)

mysql> desc students;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| student_id | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname  | varchar(12) | YES  |     | NULL    |                |
| lastname   | varchar(12) | YES  |     | NULL    |                |
| age        | int(11)     | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

mysql> show columns from students;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| student_id | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname  | varchar(12) | YES  |     | NULL    |                |
| lastname   | varchar(12) | YES  |     | NULL    |                |
| age        | int(11)     | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

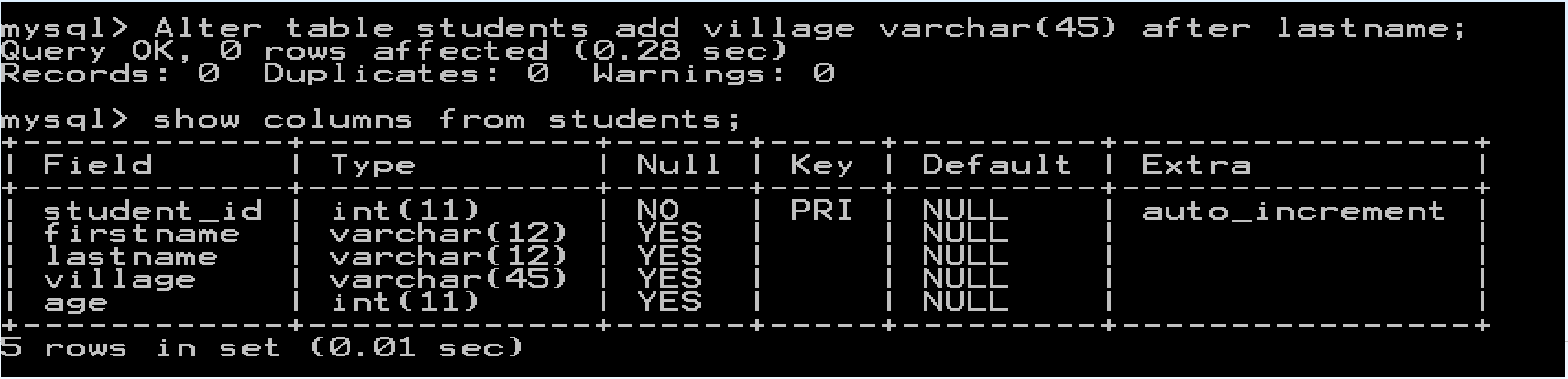
Using Alter SQL Command

- The MySQL **ALTER** command is very useful when you want to change a name of your table, any table field or if you want to add or delete an existing column in a table.

- **Add column in table**

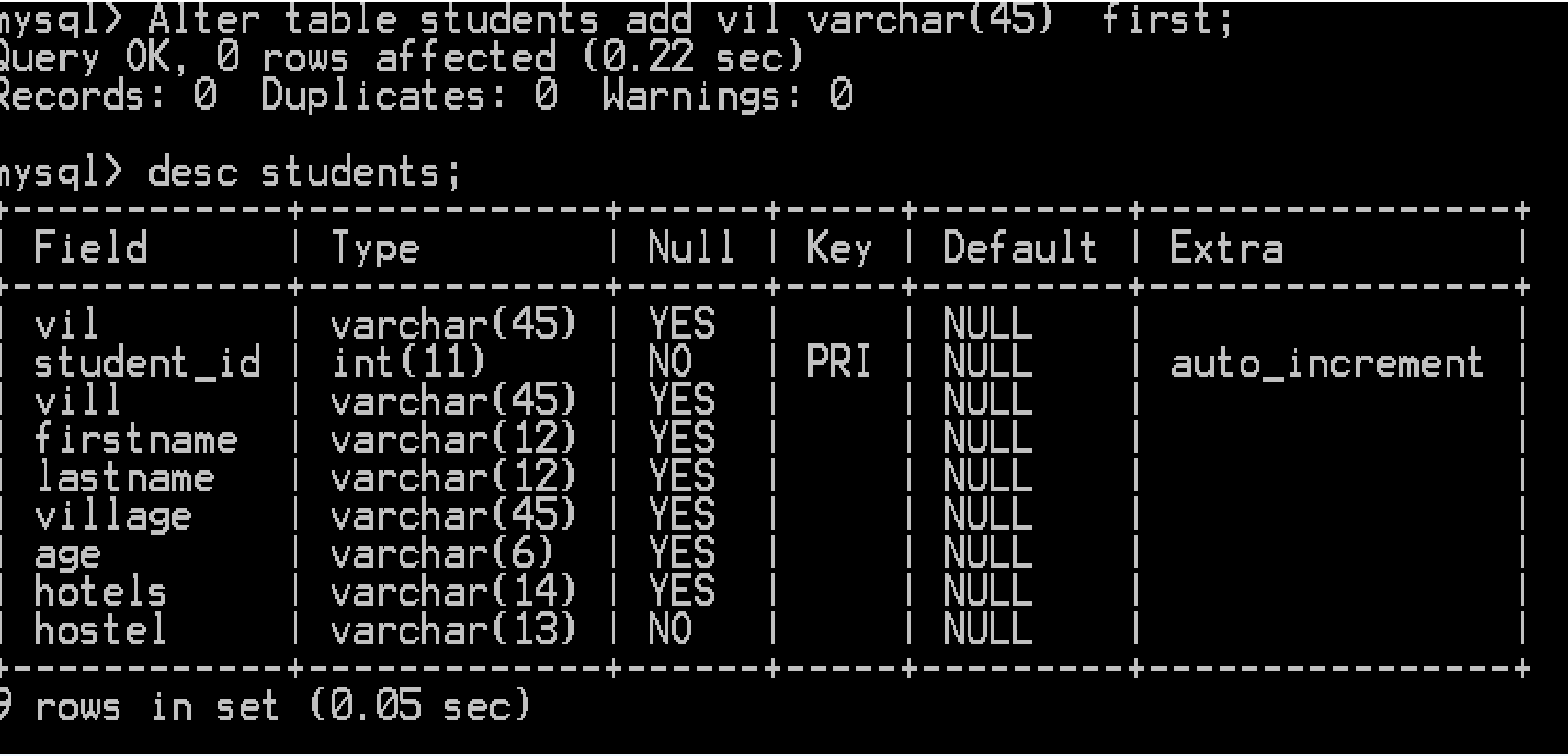


```
mysql> alter table students add hostel varchar (13);
Query OK, 0 rows affected (0.22 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show columns from students;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| student_id  | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname   | varchar(12) | YES  |     | NULL    |                |
| lastname    | varchar(12) | YES  |     | NULL    |                |
| village     | varchar(45) | YES  |     | NULL    |                |
| age         | int(11)     | YES  |     | NULL    |                |
| hostel      | varchar(13) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
6 rows in set (0.02 sec)
```



```
mysql> Alter table students add village varchar(45) after lastname;
Query OK, 0 rows affected (0.28 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show columns from students;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| student_id  | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname   | varchar(12) | YES  |     | NULL    |                |
| lastname    | varchar(12) | YES  |     | NULL    |                |
| village     | varchar(45) | YES  |     | NULL    |                |
| age         | int(11)     | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```

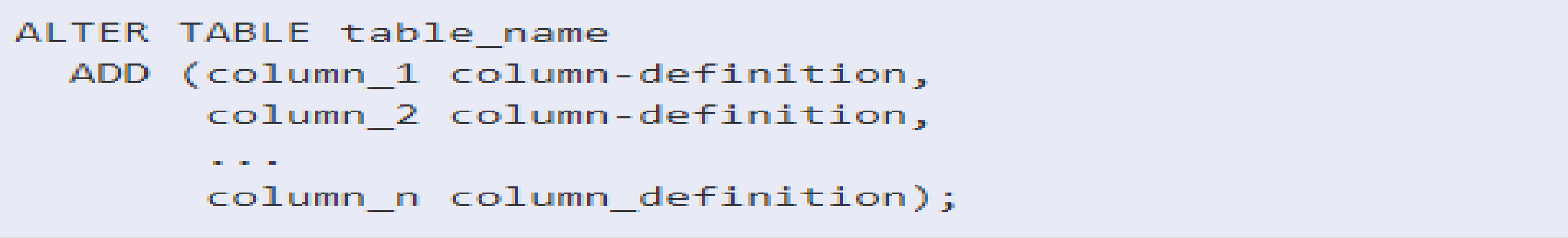
In case you want to place column at the beginning or start of a table, use the "FIRST" statement:



```
mysql> Alter table students add vil varchar(45)  first;
Query OK, 0 rows affected (0.22 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc students;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| vil        | varchar(45) | YES  |     | NULL    |                |
| student_id | int(11)     | NO   | PRI | NULL    | auto_increment |
| vill       | varchar(45) | YES  |     | NULL    |                |
| firstname  | varchar(12) | YES  |     | NULL    |                |
| lastname   | varchar(12) | YES  |     | NULL    |                |
| village    | varchar(45) | YES  |     | NULL    |                |
| age        | varchar(6)  | YES  |     | NULL    |                |
| hotels     | varchar(14) | YES  |     | NULL    |                |
| hostel     | varchar(13) | NO   |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
9 rows in set (0.05 sec)
```

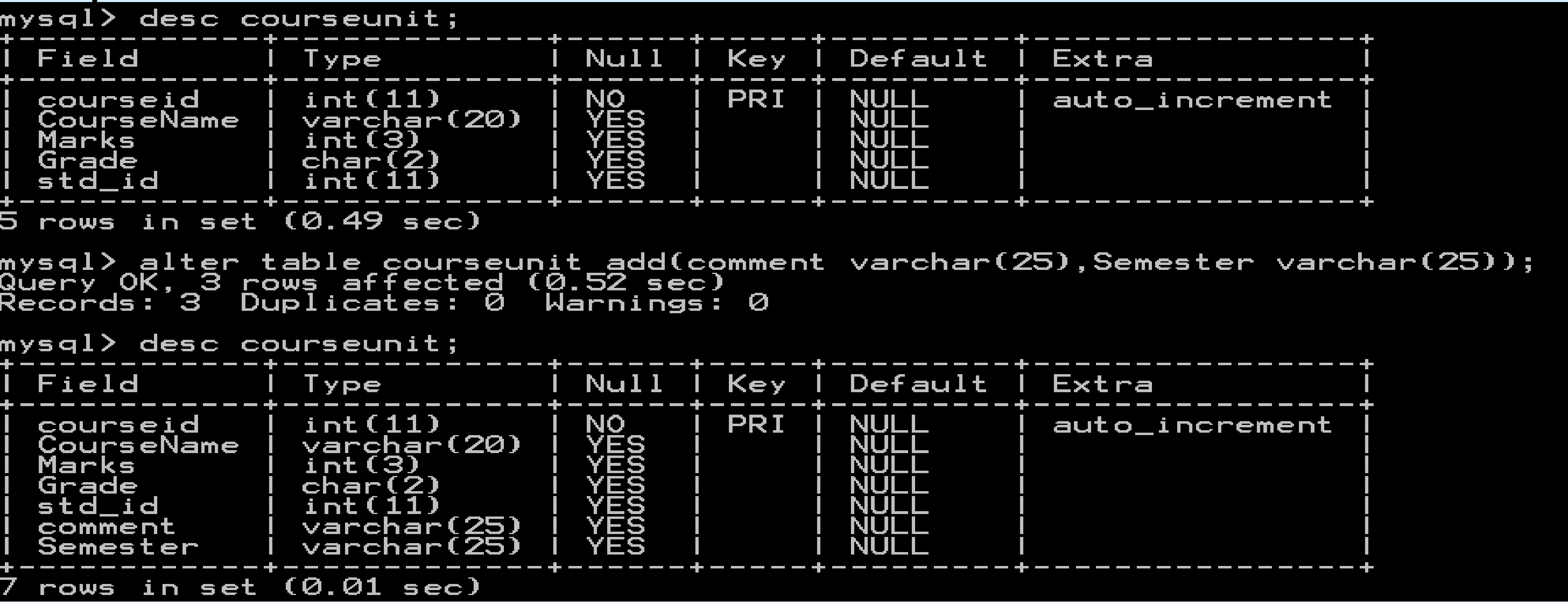
To add multiple columns to an existing table,

- SQL ALTER TABLE syntax is:



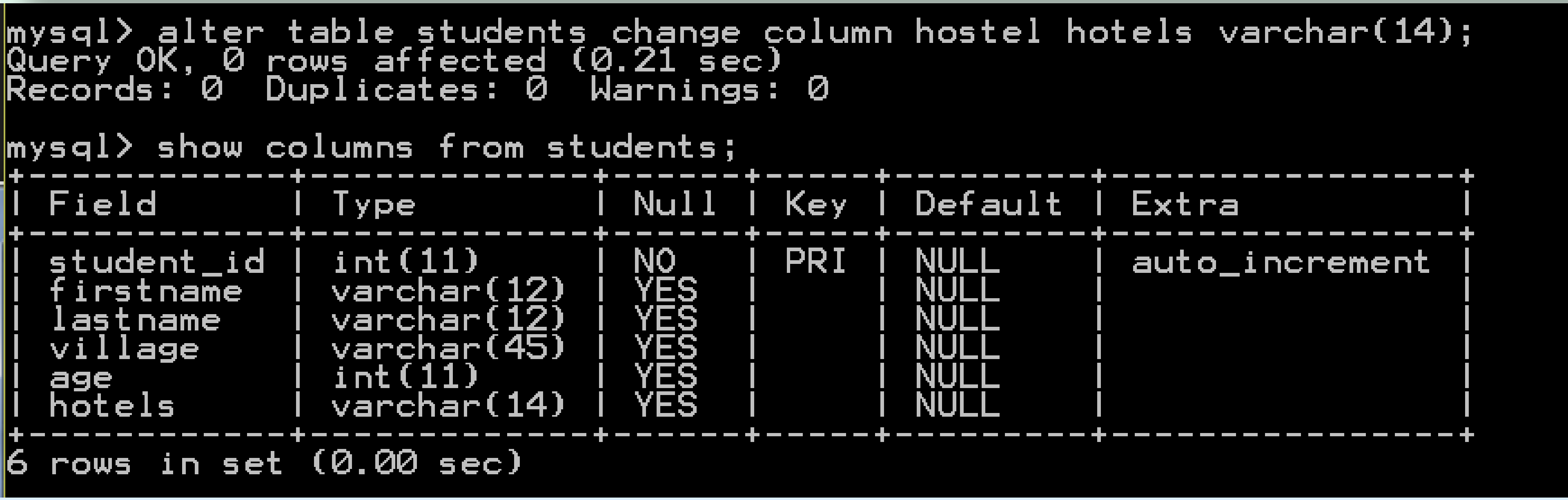
```
ALTER TABLE table_name
   ADD (column_1 column-definition,
        column_2 column-definition,
        ...
        column_n column_definition);
```

Example



```
mysql> desc courseunit;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| courseid      | int(11)     | NO   | PRI | NULL    | auto_increment |
| CourseName    | varchar(20) | YES  |     | NULL    |                |
| Marks         | int(3)      | YES  |     | NULL    |                |
| Grade         | char(2)     | YES  |     | NULL    |                |
| std_id        | int(11)     | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
5 rows in set (0.49 sec)

mysql> alter table courseunit add(comment varchar(25),Semester varchar(25));
Query OK, 3 rows affected (0.52 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> desc courseunit;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| courseid      | int(11)     | NO   | PRI | NULL    | auto_increment |
| CourseName    | varchar(20) | YES  |     | NULL    |                |
| Marks         | int(3)      | YES  |     | NULL    |                |
| Grade         | char(2)     | YES  |     | NULL    |                |
| std_id        | int(11)     | YES  |     | NULL    |                |
| comment       | varchar(25) | YES  |     | NULL    |                |
| Semester      | varchar(25) | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)
```

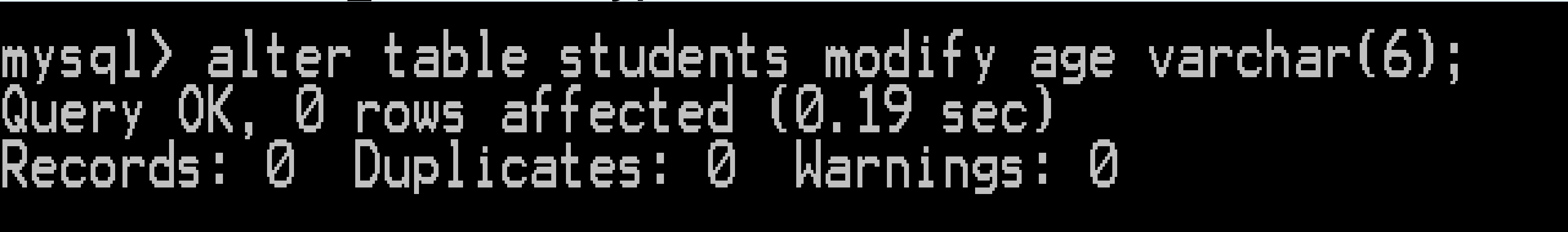
Altering column name



```
mysql> alter table students change column hostel hotels varchar(14);
Query OK, 0 rows affected (0.21 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show columns from students;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| student_id  | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname   | varchar(12) | YES  |     | NULL    |                |
| lastname    | varchar(12) | YES  |     | NULL    |                |
| village     | varchar(45) | YES  |     | NULL    |                |
| age         | int(11)     | YES  |     | NULL    |                |
| hotels      | varchar(14) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

To change the data type of a column in a table, we can use the following Syntax

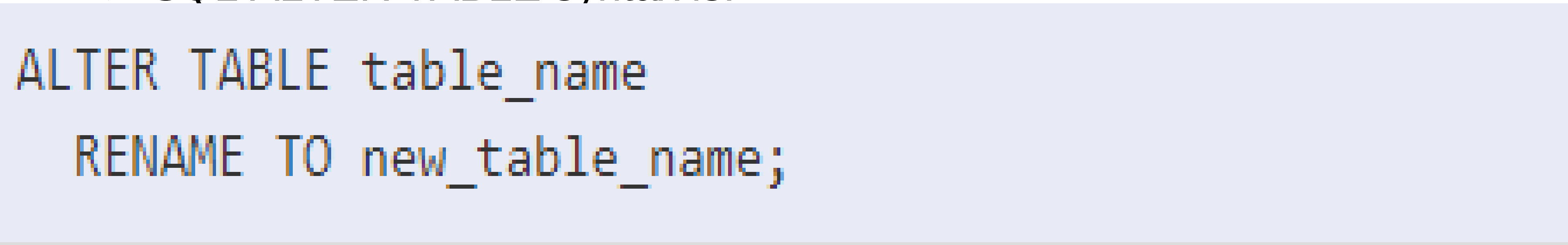
```
ALTER TABLE table_name
MODIFY  column_name datatype;
```



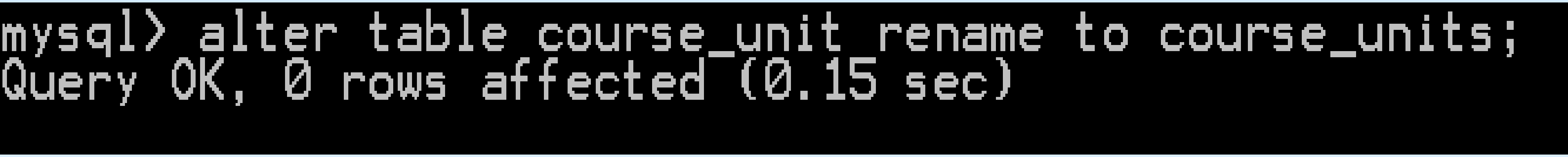
```
mysql> alter table students modify age varchar(6);
Query OK, 0 rows affected (0.19 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

To rename a table,

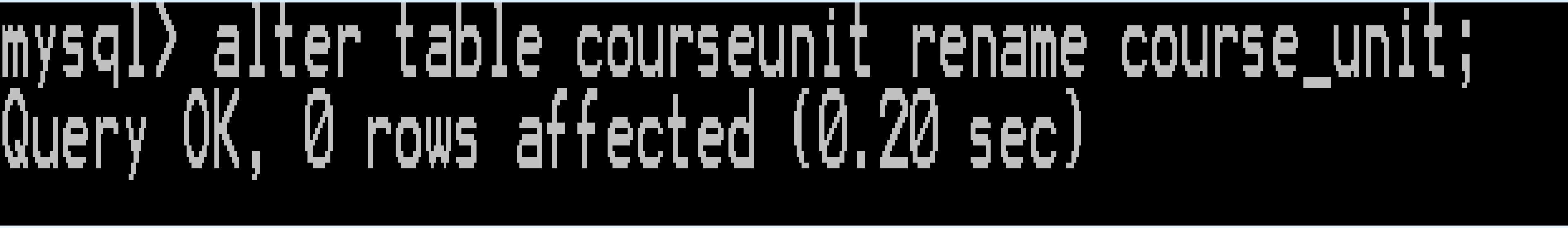
- SQL ALTER TABLE syntax is:



```
ALTER TABLE table_name
  RENAME TO new_table_name;
```



```
mysql> alter table course_unit rename to course_units;
Query OK, 0 rows affected (0.15 sec)
```

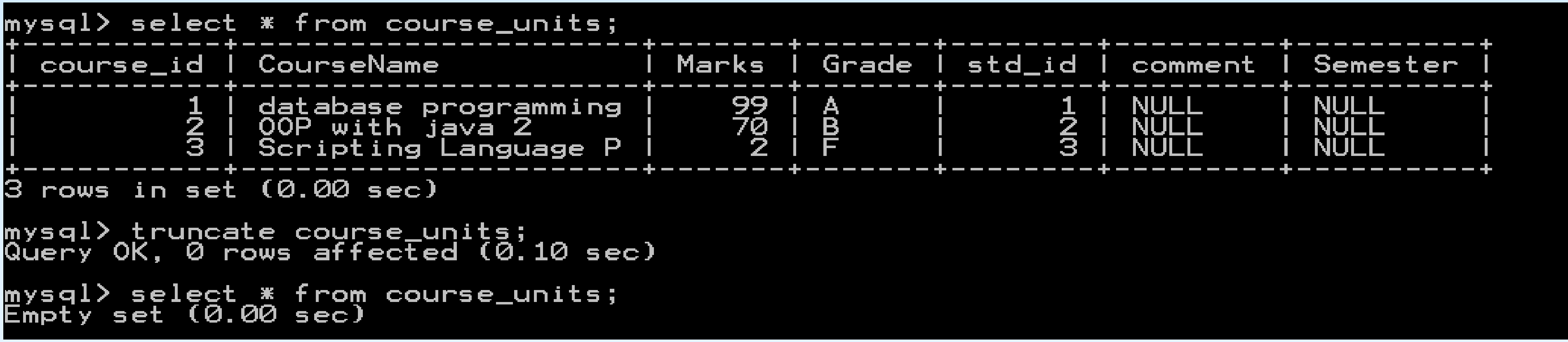


```
mysql> alter table courseunit rename course_unit;
Query OK, 0 rows affected (0.20 sec)
```

Using the truncate command

- The TRUNCATE TABLE statement is a fast, efficient method of deleting all rows in a table. TRUNCATE TABLE is similar to the DELETE statement without a WHERE clause. However, TRUNCATE TABLE is faster and uses fewer system and transaction log resources.

Example Let us first select all the records in the course_units table to see the available records the after we use the truncate command on the course_units table and we see what happens



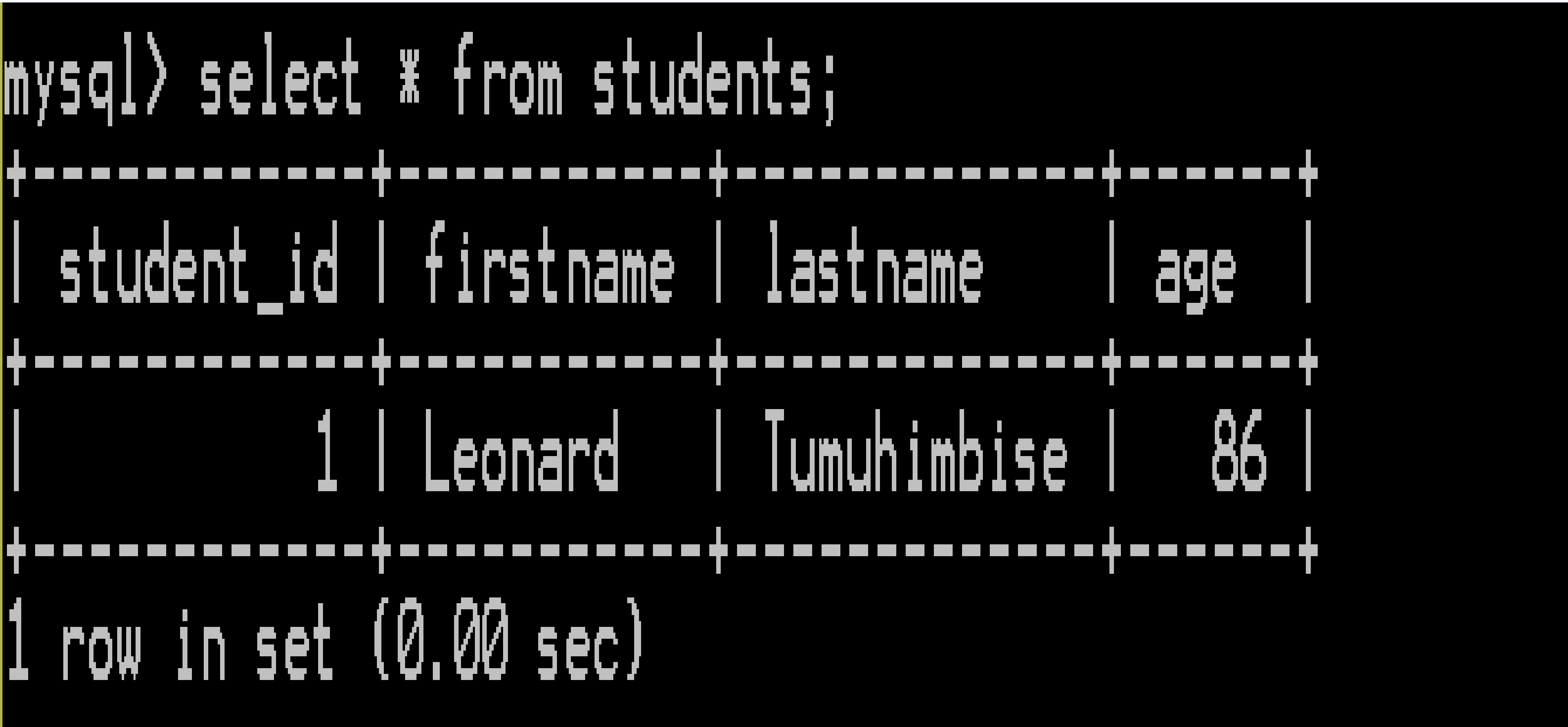
```
mysql> select * from course_units;
+-----------+----------------------+-------+-------+--------+---------+----------+
| course_id | CourseName           | Marks | Grade | std_id | comment | Semester |
+-----------+----------------------+-------+-------+--------+---------+----------+
|         1 | database programming |    99 | A     |      1 | NULL    | NULL     |
|         2 | OOP with java 2      |    70 | B     |      2 | NULL    | NULL     |
|         3 | Scripting Language P |     2 | F     |      3 | NULL    | NULL     |
+-----------+----------------------+-------+-------+--------+---------+----------+
3 rows in set (0.00 sec)

mysql> truncate course_units;
Query OK, 0 rows affected (0.10 sec)

mysql> select * from course_units;
Empty set (0.00 sec)
```

NOTE its important to note that after using the truncate command all the results in the table were Erased or deleted but the table remained as an empty set.

SQL commands

SELECT – The most common command. Used to retrieve data from a database

For example



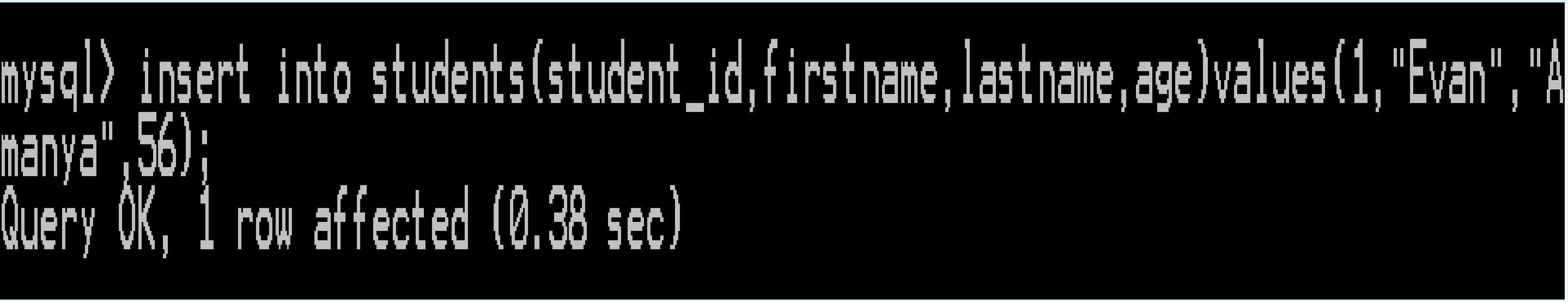
```
mysql> select * from students;
+------------+-----------+-------------+-------+
| student_id | firstname | lastname    | age   |
+------------+-----------+-------------+-------+
|          1 | Leonard   | Tumuhimbise |    86 |
+------------+-----------+-------------+-------+
1 row in set (0.00 sec)
```

DML or Data Manipulation Language commands.

Example of DML or Data Manipulation Language commands include

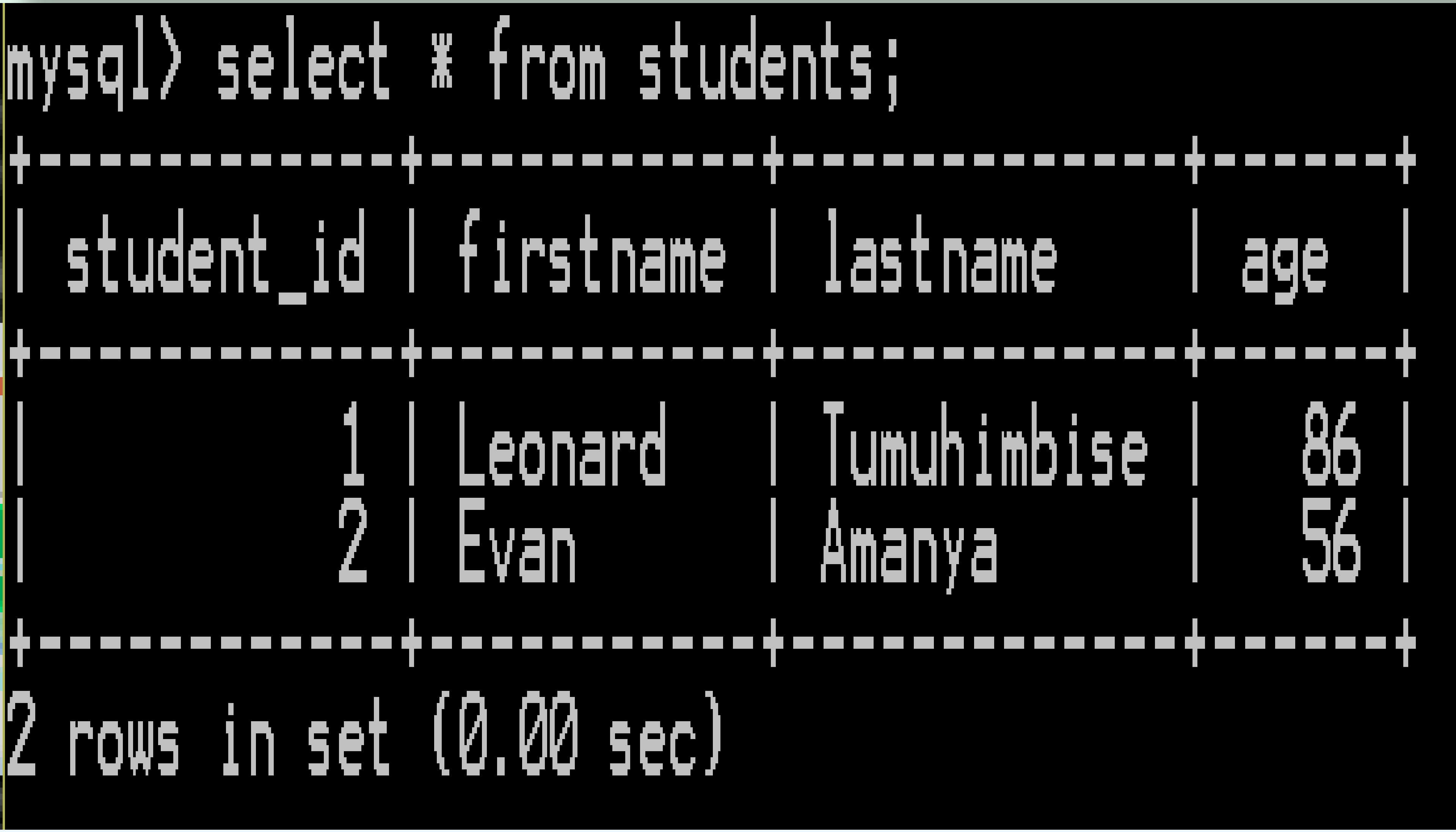
INSERT, UPDATE, DELETE – These commands are used to enter new rows, change existing rows and remove unwanted rows from tables in the database.

An Example for Inserting values in to at table



```
mysql> insert into students(student_id,firstname,lastname,age)values(1,"Evan","A
manya",56);
Query OK, 1 row affected (0.38 sec)
```

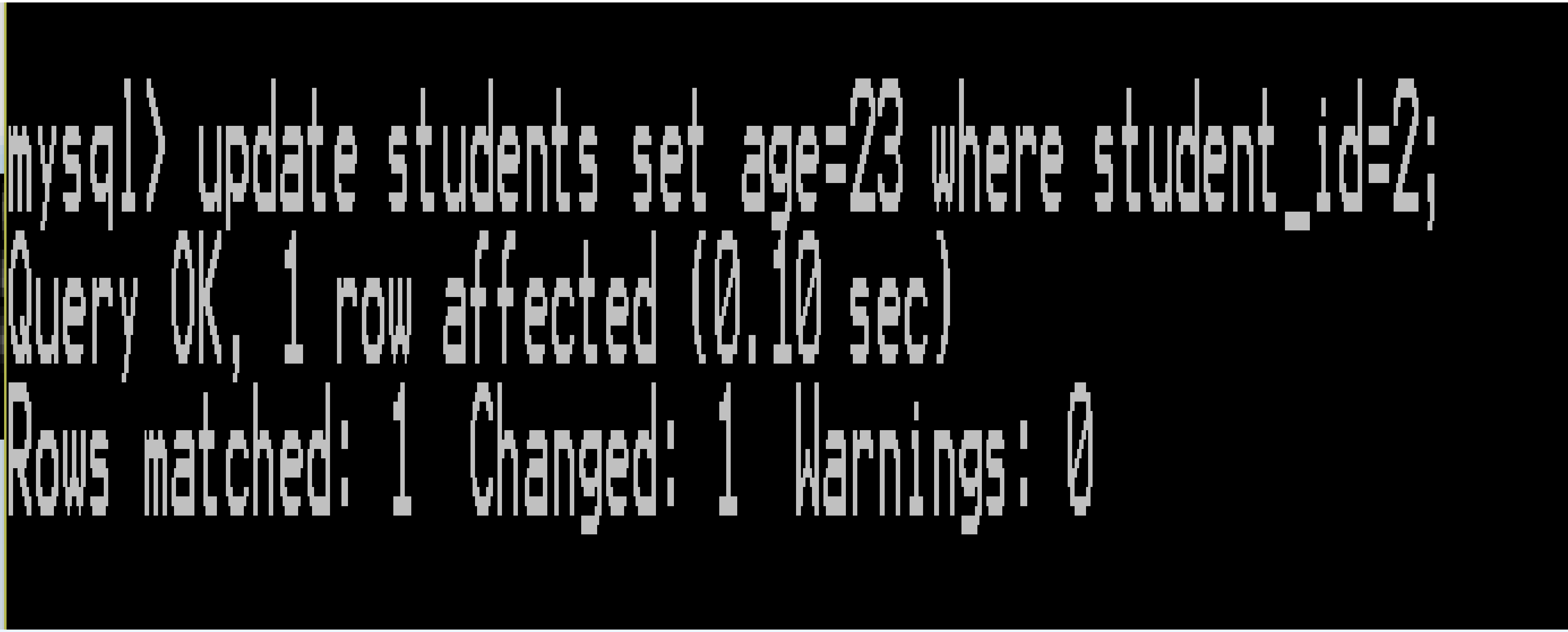
Checking what we have inserted .



```
mysql> select * from students;
+------------+-----------+-------------+------+
| student_id | firstname | lastname    | age  |
+------------+-----------+-------------+------+
|          1 | Leonard   | Tumuhimbise |   86 |
|          2 | Evan      | Amanya      |   56 |
+------------+-----------+-------------+------+
2 rows in set (0.00 sec)
```

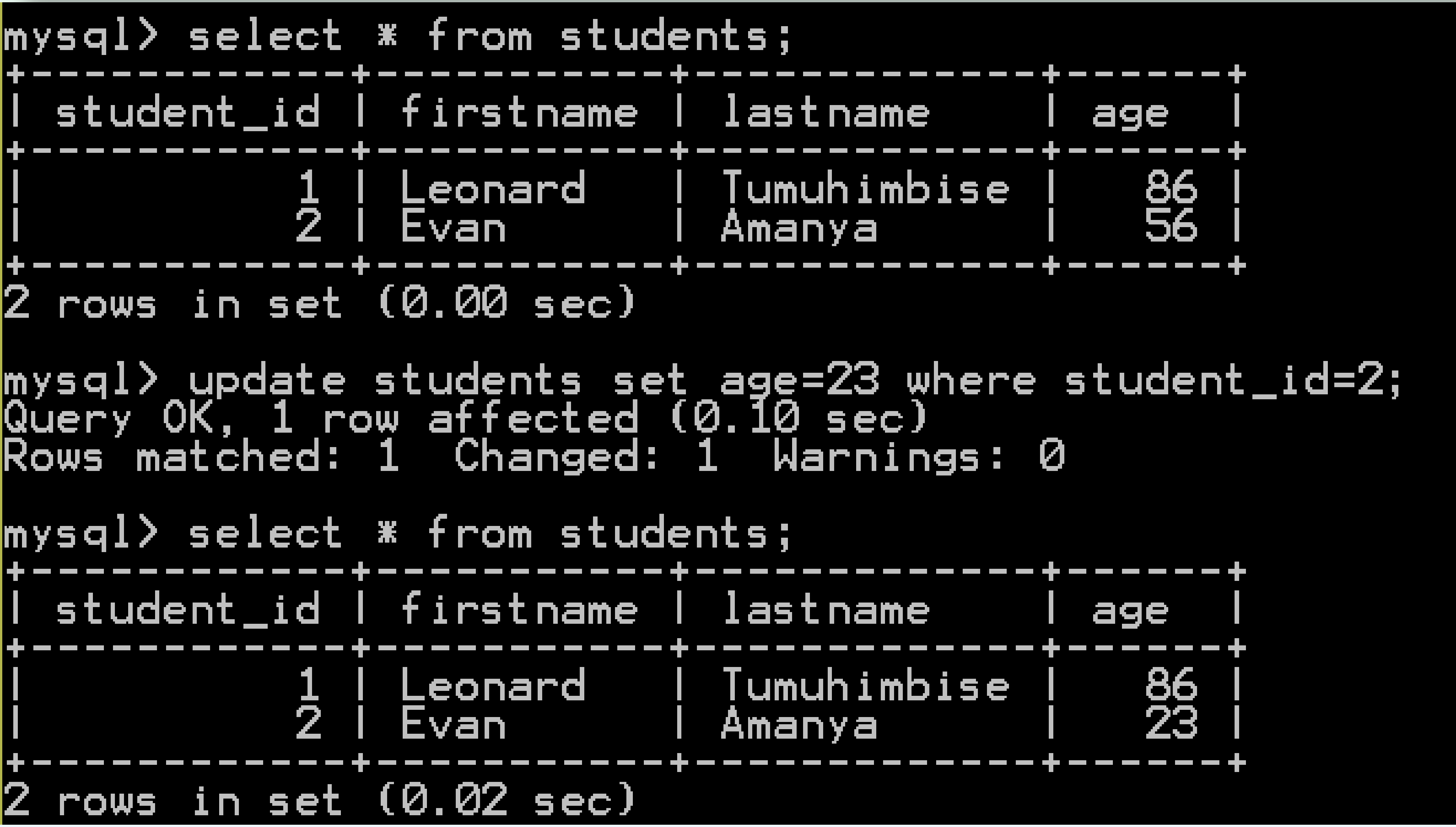
Using the update sql command

- An example where we are changing the age to 21where the student_id=2



```
mysql> update students set age=23 where student_id=2;
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

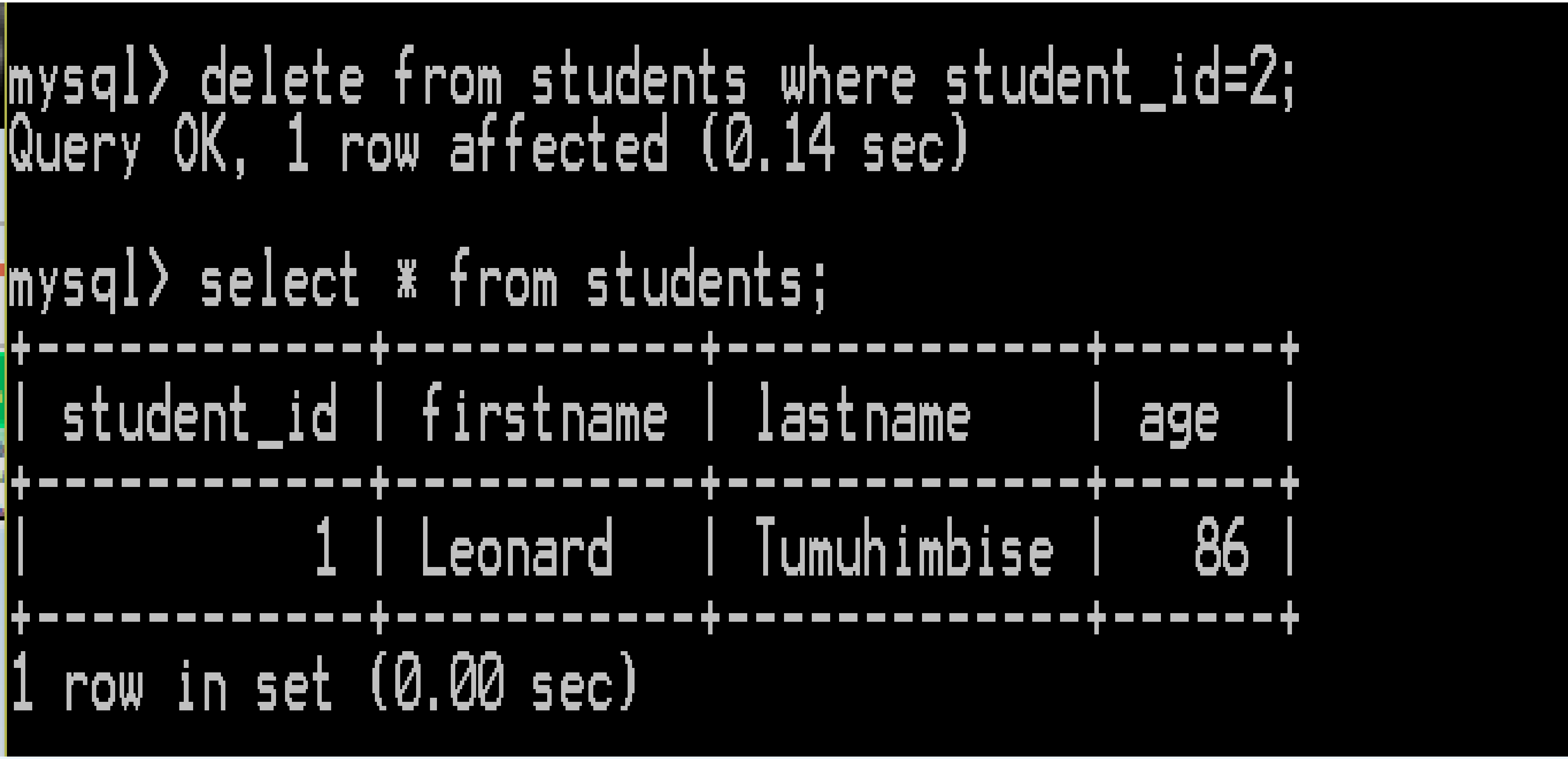
Checking what has been updated



```
mysql> select * from students;
+------------+-----------+-------------+------+
| student_id | firstname | lastname    | age  |
+------------+-----------+-------------+------+
|          1 | Leonard   | Tumuhimbise |   86 |
|          2 | Evan      | Amanya      |   56 |
+------------+-----------+-------------+------+
2 rows in set (0.00 sec)

mysql> update students set age=23 where student_id=2;
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from students;
+------------+-----------+-------------+------+
| student_id | firstname | lastname    | age  |
+------------+-----------+-------------+------+
|          1 | Leonard   | Tumuhimbise |   86 |
|          2 | Evan      | Amanya      |   23 |
+------------+-----------+-------------+------+
2 rows in set (0.02 sec)
```

Using the DELETE sql command

- An example where we are deleting from table student where the student_id=2



```
mysql> delete from students where student_id=2;
Query OK, 1 row affected (0.14 sec)

mysql> select * from students;
+------------+-----------+-------------+------+
| student_id | firstname | lastname    | age  |
+------------+-----------+-------------+------+
|          1 | Leonard   | Tumuhimbise |   86 |
+------------+-----------+-------------+------+
1 row in set (0.00 sec)
```