SUCCEED WE MUST

DATABASE PROGRAMING BIT-2212

Lecture 2

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

SUCCEED WE MUST

History of SQL

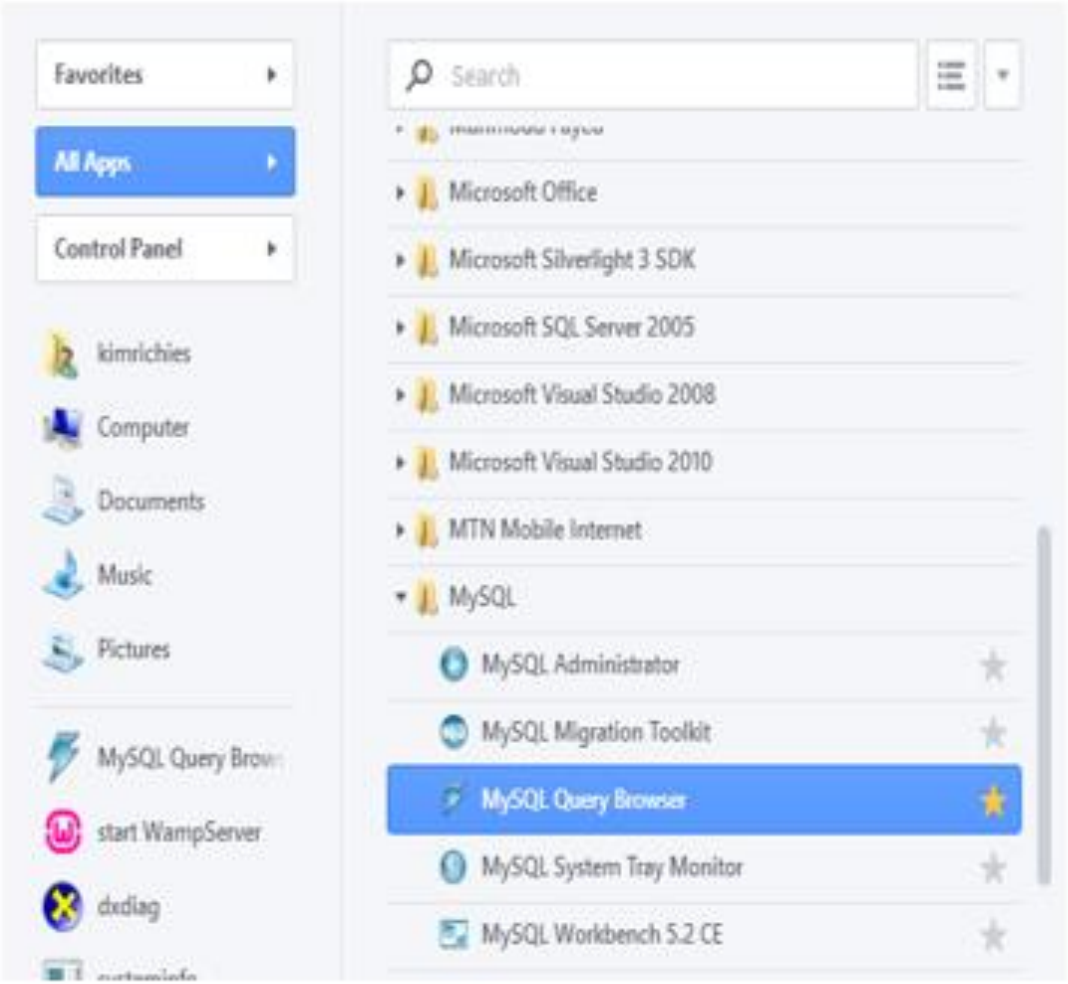
- SQL or **S**tructured **Q**uery **L**anguage was developed in 1970 by IBM as a method of interfacing with **Relational Datab**ase **M**anagement **S**ystems (RDBMS).
- In 1979, Relational Software, Inc. (now Oracle) introduced the first commercially available implementation of SQL
- Today it is the 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.

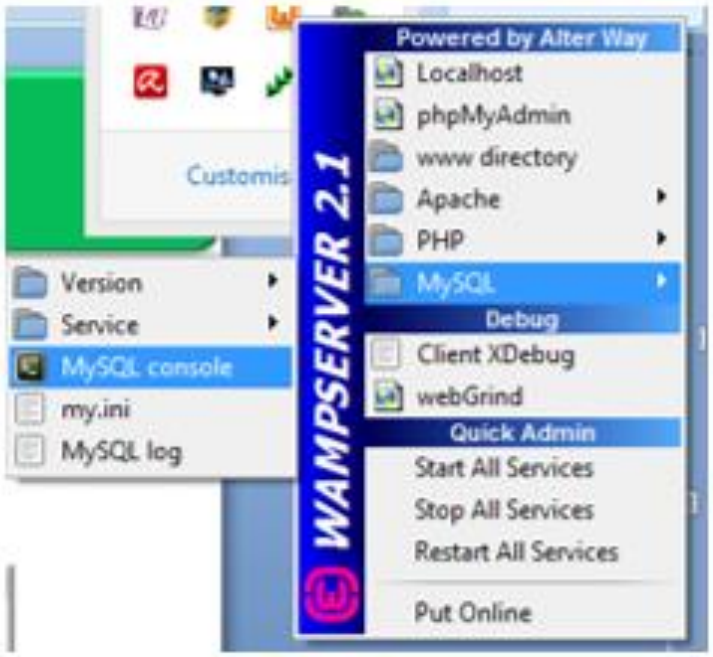
Data Definition Language

- 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

SUCCEED WE MUST

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)

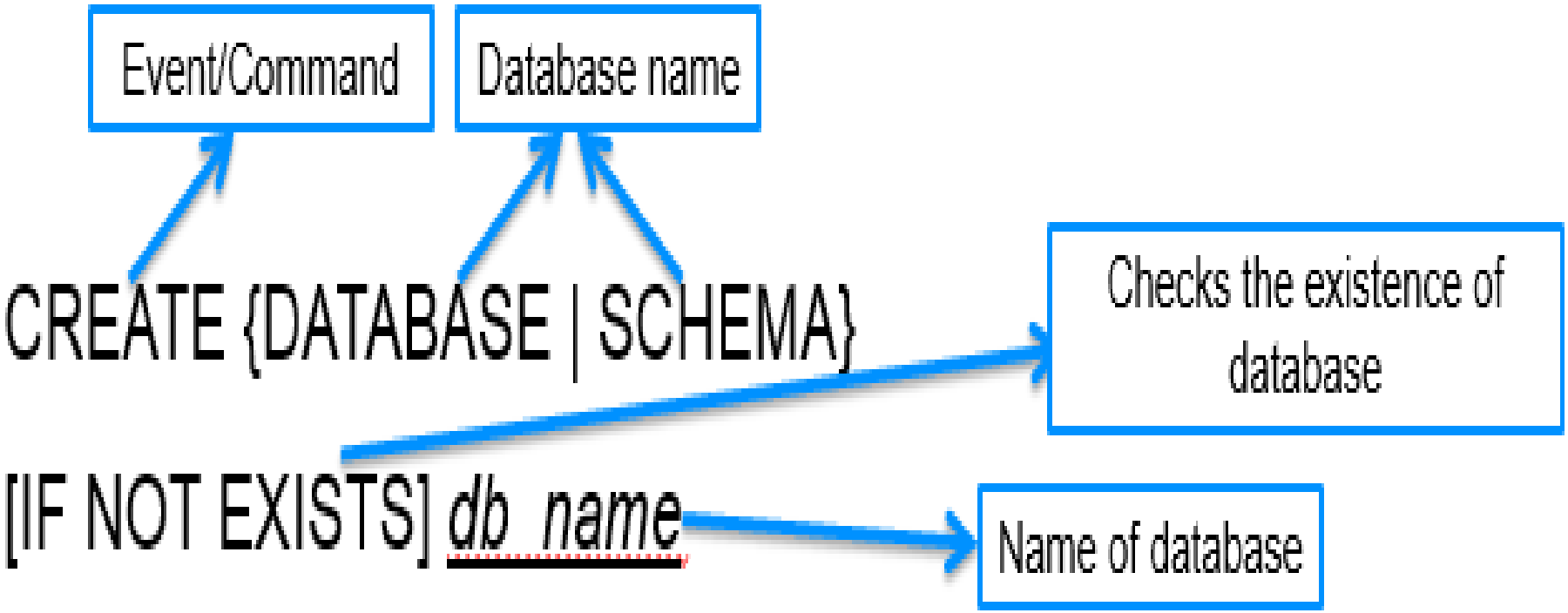
- DDL are commands are used to set up, change and remove data structures such as tables.

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:



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 BIT2;
Query OK, 1 row affected (0.05 sec)
```

```
mysql> create database BITTWO;
Query OK, 1 row affected (0.00 sec)
```

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 |
| 2013bit3_class     |
| data_tracker       |
| evaluation         |
| heada              |
| mte3               |
| mysql              |
| performance_schema |
| test               |
| tests              |
+--------------------+
10 rows in set (0.08 sec)
```

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 bitz;
Database changed
```

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)
```

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.

```
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)
```

```
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)
```

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
```

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.

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)
```

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)
```

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)
```

SUCCEED WE MUST

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)
```