

Data Definition and Datatypes

Managing DBs using IDEs



Mariyan Apostolov

mariyan.apostolov89@gmail.com

CODERDOJO
<BULGARIA>

Table of Content

- Data Types in MySQL
- Database Modelling
- Basic SQL Queries
- Altering Tables
- Deleting Data and Structures



Data Types in MySQL

Numeric, String, Date

Data Types in MySQL

- Numeric data types have certain range
- Their range can be changed if they are:
 - **Signed** - represent numbers both in the positive and negative ranges
 - **Unsigned** - represent numbers only in the positive range
- E.g. signed and unsigned INT:

Signed Range		Unsigned Range	
Min Value	Max Value	Min Value	Max Value
-2147483648	2147483648	0	4294967295

Data Types in MySQL

- **INT** [(M)] [UNSIGNED]
 - TINYINT, SMALLINT, MEDIUMINT, BIGINT

- **DOUBLE** [(M, D)] [UNSIGNED]

Digits stored
for value

Decimals after
floating point

- E.g. DOUBLE[5, 2] – 999.99
- **DECIMAL** [(M, D)] [UNSIGNED] [ZEROFILL]

Data Types in MySQL

- **CHAR** [(M)] - up to 30 characters
- **VARCHAR**(M) – up to 255 characters
- **TEXT** [(M)] – up to 65 535 characters
 - TINYTEXT, MEDIUMTEXT, LONGTEXT
- **BLOB** - Binary Large Object [(M)] - 65 535 (2¹⁶ – 1) characters
 - TINYBLOB, MEDIUMBLOB, LONGBLOB

Column name	Column Type
title	VARCHAR(CHAR)
content	TEXT(LONGTEXT)
picture	BLOB(LONGBLOB)

Data Types in MySQL

- **DATE** - for values with a date part but **no time part**
- **TIME** - for values with time but **no date part**
- **DATETIME** - values that contain both **date and time** parts
- **TIMESTAMP** - both **date and time** parts

Column name	Column Type
birthdate	DATE
last_time_online	TIMESTAMP
start_at	TIME
deleted_on	DATETIME



Database Modelling

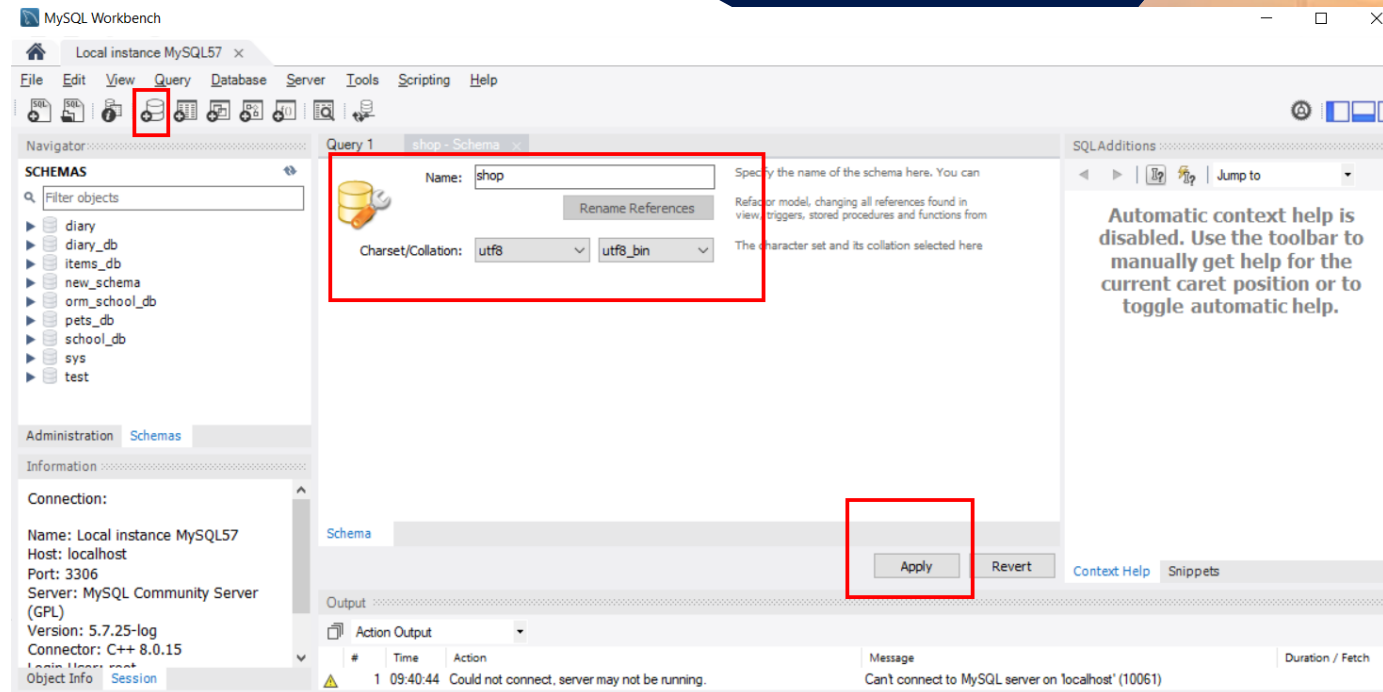
Using GUI Clients

Database Modelling

- We will **manage** databases with **MySQL Workbench**
- Enables us:
 - To **create** a new database
 - To create **objects in the database** (tables, stored procedures, relationships and others)
 - To **change** the properties of objects
 - To **enter records** into the tables

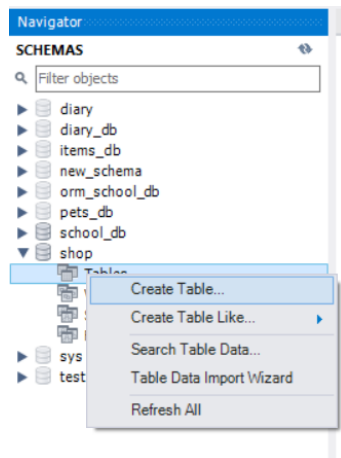
Database Modelling

- Creating a new database
 - Select **Create new schema** from the **command menu**



Database Modelling

- Right click on “Tables”
Select **Create Table**



The 'Create Table' dialog box is shown for the 'products' table in the 'shop' schema. The table name 'products' is highlighted in the 'Table Name' field. The 'Schema' is 'shop' and the 'Engine' is 'InnoDB'. The 'Charset/Collation' is set to 'Default Charset' and 'Default Collation'. The 'Comments' field is empty. The 'Columns' tab is active, showing a table with columns: 'id' (INT, Primary Key, Auto Increment), 'name' (VARCHAR(255)), 'description' (TEXT), 'price' (DECIMAL(10,2)), and 'category' (VARCHAR(255)). The 'id' column is highlighted. The 'Data Type' is 'INT'. The 'Storage' options are 'Virtual' and 'Stored'. The 'Primary Key' and 'Auto Increment' checkboxes are checked. The 'Apply' button is highlighted.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AL	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	



Basic SQL Queries

Data Definition using SQL

Basic SQL Queries

- We communicate with the database engine using SQL
- Queries provide greater **control** and **flexibility**
- To create a database using SQL:

```
CREATE DATABASE shop;
```

- SQL keywords are conventionally **capitalized**

Basic SQL Queries

- Table Creation in SQL

```
CREATE TABLE `shop`.`products`  
(  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `product_name` VARCHAR(45) NOT NULL,  
  `product_owner` VARCHAR(45),  
  PRIMARY KEY (`id`)  
);
```

Table name

Column
name

Custom
properties

Data type

Basic SQL Queries

- Get all information from a table

```
SELECT * FROM `shop`.`products`;
```

- You can limit the columns and number of records

```
SELECT `id`, `product_name` FROM `shop`.`products`  
LIMIT 5;
```

Basic SQL Queries

- Table Customization

- Primary Key

```
id NOT NULL PRIMARY KEY
```

- Auto-Increment (Identity)

```
id INT AUTO_INCREMENT PRIMARY KEY
```

- Unique constraint – no repeating values in entire table

```
email VARCHAR(50) UNIQUE
```

- Default value – if not specified (otherwise set to NULL)

```
balance DECIMAL(10,2) DEFAULT 0
```



Altering Tables

Changing Table Properties
After Creation

Altering Tables

- A table can be changed using the keywords ALTER TABLE

```
ALTER TABLE products;
```

- Add new column

```
ALTER TABLE products  
ADD price DECIMAL;
```

Column name

Data type

Altering Tables

- Delete existing column

```
ALTER TABLE products  
DROP COLUMN price;
```

Column name

- Modify data type of existing column

```
ALTER TABLE products  
MODIFY COLUMN email VARCHAR(100);
```

Column name

New data type

Altering Tables

- Add primary key to existing column

```
ALTER TABLE products  
ADD CONSTRAINT pk_id  
PRIMARY KEY (id);
```

Constraint name

Column name

- Add unique constraint

```
ALTER TABLE products  
ADD CONSTRAINT uq_email  
UNIQUE (email)
```

Constraint name

Column name(s)

Altering Tables

- Set default value

```
ALTER TABLE products  
ALTER COLUMN balance SET DEFAULT 0;
```

Default value

Column name



Deleting Data and Structures

Dropping and Truncating

Deleting Data and Structures

- Deleting structures is called **dropping**
 - You can drop **keys**, **constraints**, **tables** and entire **databases**
- Deleting all data in a table is called **truncating**
- **Both of these actions cannot be undone** – use with caution!

Deleting Data and Structures

- To delete all the entries in a table

```
TRUNCATE TABLE products;
```

Table name

- To drop a table – delete data and structure

```
DROP TABLE products;
```

Table name

- To drop entire database

```
DROP DATABASE shop;
```

Database name

Deleting Data and Structures

- To remove a constraining rule from a column
 - Primary keys, value constraints and unique fields

```
ALTER TABLE products  
DROP CONSTRAINT pk_id;
```

Table name

Constraint name

- To remove **DEFAULT** value (if not specified, revert to NULL)

```
ALTER TABLE products  
ALTER COLUMN balance  
DROP DEFAULT;
```

Table name

Columns name



Thank You.



Mariyan Apostolov



mariyan.apostolov89@gmail.com



<https://www.coderdojo.bg/>

