

Session 17

How to use SQL to create a database

Objectives

- How to work with databases
- How to work with tables
- How to work with indexes
- How to work with users and privileges
- Other skills for creating a database

How to work with database

How to use a database

- How to create a database

```
CREATE DATABASE my_guitar_shop2;
```

- How to create a database only if it does not exist

```
CREATE DATABASE IF NOT EXISTS my_guitar_shop2;
```

- How to select a database

```
USE my_guitar_shop2;
```

- How to drop a database

```
DROP DATABASE my_guitar_shop2;
```

How to work with tables

An introduction to MySQL data types

- Common numeric data types

Type	Description
INT [(size)]	An integer between -2,147,483,648 and 2,147,483,647 where size is the maximum display size. Default size is 11.
TINYINT [(size)]	An integer between -127 and 128 where size is the maximum display size. Default size is 4.
DECIMAL [(p[,s])]	Decimal numbers with fixed precision (p) and scale (s). The maximum precision is 65 and the default is 10. The maximum scale is 30 and the default is 0.

- Common string data types

Type	Description
VARCHAR (size)	Variable-length characters where size is the maximum number of characters. The size argument is required. The maximum size is 65,535.
CHAR [(size)]	Fixed-length characters where size is the number of characters. The default size is 1. The maximum size is 255.
TEXT	Variable-length characters up to a maximum size of 65,535 bytes.

An introduction to MySQL data types (cont.)

- Common date and time data types

Type	Description
DATE	Dates from January 1, 1000 through December 31, 9999. The default format for entry and display is “yyyy-mm-dd”.
TIME	Times in the range -838:59:59 through 838:59:59. The default format for display and entry is “hh:mi:ss”.
DATETIME	Date and time from January 1, 1970 through December 31, 9999. The default format for display and entry is “yyyy-mm-dd hh:mi:ss”.

- MySQL uses the Latin-1-character set, which provides for 256 characters with 1 byte per character.
- The Unicode character set provides for over 65,000 characters, usually with 2 bytes per character.

How to create a table

- The CREATE TABLE statement creates based on the column definitions and column attributes you specify.
- The syntax of the CREATE TABLE statement

```
CREATE TABLE [IF NOT EXISTS] tableName
(
    columnName1    dataType    [columnAttributes] [,
    columnName2    dataType    [columnAttributes]] [,
    columnName3    dataType    [columnAttributes]] ...
)
```


How to create a table (cont.)

- Three common column attributes

Attribute	Description
UNIQUE	Specifies that each value stored in the column must be unique, but allows the column to store NULL values.
NOT NULL	Indicates that the column doesn't accept NULL values. If omitted, the column can accept NULL values.
DEFAULT default_value	Specifies a default value for the column.

- A table without column attributes

```
CREATE TABLE customers
(
    customerID    INT,
    firstName     VARCHAR(60),
    lastName      VARCHAR(60)
);
```

How to create a table (cont.)

- A table with column attributes

```
CREATE TABLE customers
(
    customerID    INT                NOT NULL    UNIQUE,
    firstName     VARCHAR(60)        NOT NULL,
    lastName      VARCHAR(60)        NOT NULL
);
```

- Another table with column attributes

```
CREATE TABLE orders
(
    orderID       INT                NOT NULL    UNIQUE,
    customerID    INT                NOT NULL,
    orderNumber   VARCHAR(50)        NOT NULL,
    orderDate     DATE               NOT NULL,
    orderTotal    DECIMAL(9,2)       NOT NULL,
    paymentTotal  DECIMAL(9,2)       DEFAULT 0
);
```

How to code a primary key

- A primary key requires that each row has unique values for the column or columns that are used for the primary key, and it does not allow NULL values.
- You code a column-level primary key as part of the definition of the column definition.
- You code a table-level primary key as if it is a separate column definition and you name its column or columns within that definition

How to alter a table

- You can use the ALTER TABLE statement to modify the columns of an existing tables

- Statement that renames a table

```
ALTER TABLE products RENAME TO product;
```

- Statement that adds a new column at the end of the table

```
ALTER TABLE customers ADD lastTransactionDate DATE;
```

- Statement that adds a new column after a specified column

```
ALTER TABLE customers ADD lastTransactionDate DATE  
AFTER emailAddress;
```

How to alter a table (cont.)

- Statement that drops a column

```
ALTER TABLE customers DROP lastTransactionDate;
```

- Statement that renames a column

```
ALTER TABLE customers  
CHANGE emailAddress email VARCHAR(255) NOT NULL UNIQUE;
```

- Statement that changes the definition of a column

```
ALTER TABLE customers MODIFY firstName VARCHAR(100) NOT NULL;
```

- Statement that changes the data type of a column

```
ALTER TABLE customers MODIFY firstName CHAR(100) NOT NULL;
```

- Statement that may cause data be lost

```
ALTER TABLE customers MODIFY firstName VARCHAR(8);
```

How to alter a table (cont.)

- Statement that sets the default value of a column

```
ALTER TABLE customers ALTER firstName SET DEFAULT '';
```

- Statement that drops the default value of a column

```
ALTER TABLE customers ALTER firstName DROP DEFAULT;
```

How to drop a table

- You can use the DROP TABLE statement to delete a table

- Statement that drops a table

```
DROP TABLE customers;
```

- Statement that drops a table if it exists

```
DROP TABLE IF EXISTS customers;
```

How to work with indexes

Introduction to database index

- An index improves performance when MySQL searches for rows in the table
- MySQL automatically creates an index for primary keys and for unique constraints
- You can use the CREATE INDEX statement to create indexes for a table

How to create an index

- The syntax of the CREATE INDEX statement

```
CREATE [UNIQUE] INDEX|KEY indexName  
ON tableName (columnName1 [ASC|DESC] [, columnName2 [ASC|DESC]]...)
```

- Statement that creates an index based on a single column

```
CREATE INDEX customerID  
ON orders (customerID);
```

- Statement that creates a unique index

```
CREATE UNIQUE INDEX emailAddress  
ON customers (emailAddress);
```

- Statement that creates an index based on two columns

```
CREATE UNIQUE INDEX customerIDorderNumber  
ON orders (customerID, orderNumber);
```

How to create an index (cont.)

- Statement that creates an index that's sorted in descending order

```
CREATE INDEX orderTotal  
ON orders (orderTotal DESC);
```

- You can use CREATE TABLE statement to create indexes for a table

```
CREATE TABLE customers (  
    customerID      INT          NOT NULL    AUTO_INCREMENT,  
    emailAddress    VARCHAR(255) NOT NULL,  
    firstName       VARCHAR(60)  NOT NULL,  
  
    PRIMARY KEY (customerID),  
    UNIQUE INDEX emailAddress (emailAddress),  
    INDEX firstName (firstName)  
);
```

How to drop an index

- You can use the DROP INDEX statement to drop an index

```
DROP INDEX firstName ON customers;
```

How to work with users and privileges

Introduction about privileges

- The first group of privileges allows the user to work with the data that's stored in the tables of the database.
- The second group of privileges allows the user to modify the structure of the database.
- MySQL makes it possible to specify privileges at the global level, the database level, the table level, and the column level.

Introduction about privileges (cont.)

- Privileges for working data

Privilege	Description
SELECT	Select data from a table.
INSERT	Insert data into a table.
UPDATE	Update data in a table.
DELETE	Delete data from a table.

- Privileges for modifying the database structure

Privilege	Description
CREATE	Create a database or a table.
ALTER	Alter a table.
DROP	Drop a database or a table.
INDEX	Create or drop an index.

Introduction about privileges (cont.)

- Other privileges

Privilege	Description
ALL [PRIVILEGES]	All privileges available at the current level except the GRANT OPTION privilege.
GRANT OPTION	Allows a user to grant his or her privileges to other users.
USAGE	No privileges. It can be used to modify existing accounts without changing privileges for that account.

- The four privilege levels

Level	Example	Description
Global	*.*	All databases and all tables.
Database	music_db.*	All tables on the specified database.
Table	music_db.products	All columns on the specified table.
Column	(listPrice) music_db.products	Only the specified column or columns.

How to create, rename, and drop users

- You use the CREATE USER statement to create a user that has no privileges

- Create a user from a specific host

```
CREATE USER joel@localhost IDENTIFIED BY 'sesame';
```

- Create a user from any host

```
CREATE USER dba IDENTIFIED BY 'sesame';      -- creates 'dba@%'
```

- Rename a user from a specific host

```
RENAME USER joel@localhost TO joelmurach@localhost;
```

How to create, rename, and drop users (cont.)

- Change a user's password

```
GRANT USAGE ON *.*  
TO joelmurach@localhost  
IDENTIFIED BY 'newpassword';
```

- Drop a user from a specific host

```
DROP USER joelmurach@localhost;
```

- Drop a user from any host

```
DROP USER dba;                                -- drops 'dba@%'
```

How to grant privileges

- The WITH GRANT OPTION clause allows the user to grant the privileges for that user other users.
- You can use the asterisk (*) to specify all databases or all tables
- The syntax of the GRANT statement

```
GRANT privilegeList
ON [dbName.]table
TO userName1 [IDENTIFIED BY 'password1'][,
    userName2 [IDENTIFIED BY 'password2']...]
[WITH GRANT OPTION]
```

How to grant privileges (cont.)

- A statement that creates a user with no privileges

```
GRANT USAGE  
ON *.*  
TO joel@localhost IDENTIFIED BY 'sesame';
```

- A statement that creates a user with database privileges

```
GRANT SELECT, INSERT, UPDATE, DELETE  
ON my_guitar_shop2.*  
TO mgs_user@localhost IDENTIFIED BY 'pa55word';
```

How to grant privileges (cont.)

- A statement that creates user with database privileges

```
GRANT SELECT, INSERT, UPDATE, DELETE  
ON my_guitar_shop2.*  
TO mgs_user@localhost IDENTIFIED BY 'pa55word';
```

- A statement that creates a user with global privileges

```
GRANT ALL  
ON *.*  
TO dba IDENTIFIED BY 'supersecret'  
WITH GRANT OPTION;
```

How to grant privileges (cont.)

- A statement that grants table privileges to a user

```
GRANT SELECT, INSERT, UPDATE  
ON my_guitar_shop2.products TO joel@localhost;
```

- A statement that grants database privileges to a user

```
GRANT SELECT, INSERT, UPDATE  
ON my_guitar_shop2.* TO joel@localhost;
```

- A statement that grants global privileges to a user

```
GRANT SELECT, INSERT, UPDATE  
ON *.* TO joel@localhost;
```

How to grant privileges (cont.)

- A statement that grants column privileges to a user

```
GRANT SELECT (productCode, productName, listPrice), UPDATE (description)  
ON my_guitar_shop2.products TO joel@localhost
```

- A statement that uses the current database

```
GRANT SELECT, INSERT, UPDATE, DELETE  
ON customers TO joel@localhost;
```

How to revoke privileges

- You can use the REVOKE statement to revoke privileges from a user
- The syntax of the REVOKE statement for all privileges

```
REVOKE ALL[ PRIVILEGES], GRANT OPTION  
FROM user [, user]
```

- A statement that revokes all privileges from a user

```
REVOKE ALL, GRANT OPTION  
FROM dba;
```


How to revoke privileges (cont.)

- A statement that revokes all privileges from multiple users

```
REVOKE ALL, GRANT OPTION  
FROM dba, joel@localhost;
```

- The syntax of the REVOKE statement for specific privileges

```
REVOKE privilegeList  
ON [dbName.]table  
FROM user [, user]
```

- A statement that revokes specific privi

```
REVOKE UPDATE, DELETE  
ON my_guitar_shop2.customers FROM joel@localhost
```

How to view privileges

- A statement that lists all user

```
SELECT User, Host from mysql.user;
```

The result set

User	Host
dba	%
joel	localhost
mgs_tester	localhost
mgs_user	localhost
pma	localhost
root	localhost

How to view privileges (cont.)

- The syntax of the SHOW GRANTS statement

```
SHOW GRANTS [FOR user]
```

- A statement that shows the privileges for a user from any host

```
SHOW GRANTS FOR dba;
```

The result set

Grants for dba@%

```
GRANT ALL PRIVILEGES ON *.* TO 'dba@%' IDENTIFIED BY PASSWORD  
**90BA3AC0BFDE07AE334CA523CB27167AE33825B9' WITH GRANT OPTION
```

How to view privileges (cont.)

- A statement that shows the privileges for a user with a specific host

```
SHOW GRANTS FOR mgs_user@localhost;
```

The result set

Grants for mgs_user@localhost
GRANT USAGE ON *.* TO 'mgs_user'@'localhost' IDENTIFIED BY PASSWORD 'F71B0AF6B232C58021B6AC63A29FCF13A4E46E59'
GRANT SELECT, INSERT, UPDATE, DELETE ON `my_guitar_shop2`.* TO 'mgs_user'@'localhost'
GRANT SELECT, INSERT, UPDATE, DELETE ON `my_guitar_shop1`.* TO 'mgs_user'@'localhost'

- The statement that shows the privileges for the current user

```
SHOW GRANTS;
```

Other skills for creating a database

How to load data from text files

- Use phpMyAdmin or the LOAD command to load data into a MySQL database from a text file.
- The Import tab for the table named products

The screenshot shows the phpMyAdmin interface for the 'products' table in the 'my_guitar_shop2' database. The 'Import' tab is selected, and the 'Format of imported file' section is expanded, showing the 'CSV using LOAD DATA' option selected. The 'File to import' section shows the location of the text file as 'C:\xampp\htdocs\book\'. The 'Partial import' section has a checkbox for 'Allow the interruption of an import...' which is checked. The 'Format of imported file' section has radio buttons for 'CSV', 'CSV using LOAD DATA' (selected), and 'SQL'. The 'Options' section includes checkboxes for 'Replace table data with file' and 'Ignore duplicate rows', both unchecked. The 'Fields terminated by' field is set to '\t', 'Fields enclosed by' is set to '"', 'Fields escaped by' is set to '\\', 'Lines terminated by' is set to '\n', and 'Column names' is set to 'auto'. The 'Use LOCAL keyword' checkbox is checked. A 'Go' button is at the bottom right.

Server: localhost Database: my_guitar_shop2 Table: products

Browse Structure SQL Search Insert Export Import Operations Empty Drop

File to import

Location of the text file: C:\xampp\htdocs\book\ Browse... (Max: 128 MB)

Character set of the file: utf-8

Imported file compression will be automatically detected from: None, gzip, bzip2, zip

Partial import

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. This might be good way to import large files, however it can break transactions.

Number of records (queries) to skip from start: 0

Format of imported file

☐ CSV

☒ CSV using LOAD DATA

☐ SQL

Options

☐ Replace table data with file

☐ Ignore duplicate rows

Fields terminated by: \t

Fields enclosed by: "

Fields escaped by: \

Lines terminated by: \n

Column names: auto

☒ Use LOCAL keyword

Go

How to load data from text files (cont.)

- A tab-delimited text file that's stored in users.txt

1	John	Smith	jsmith@gmail.com
2	Andrea	Steelman	andi@murach.com
3	Joel	Murach	joelmurach@yahoo.com

- How to use phpMyAdmin to load data from a text file
 1. Start phpMyAdmin, select the database, select the table, and click on the Import tab.
 2. Select the file to import.
 3. Set the options for the import and click the Go button. If you get an error, you can modify the import options and try again.

How to load data from text files (cont.)

- How to use the Windows command prompt to load data from a text file

```
cd \xampp\mysql\bin
mysql -u root -p
Enter password: *****
use my_guitar_shop2;
load data local infile "c:/murach/products.txt" into table products;
exit;
```


How to dump a database to a SQL script

- Use phpMyAdmin or the mysqldump command to generate a SQL script that includes all of the CREATE and INSERT statement needed to create a copy of the database
- How to use phpMyAdmin to dump a database to a SQL script
 1. Start phpMyAdmin, select the database, and click on the Export tab.
 2. Set the options for the SQL script file.
 3. Click on the Go button and save the file.

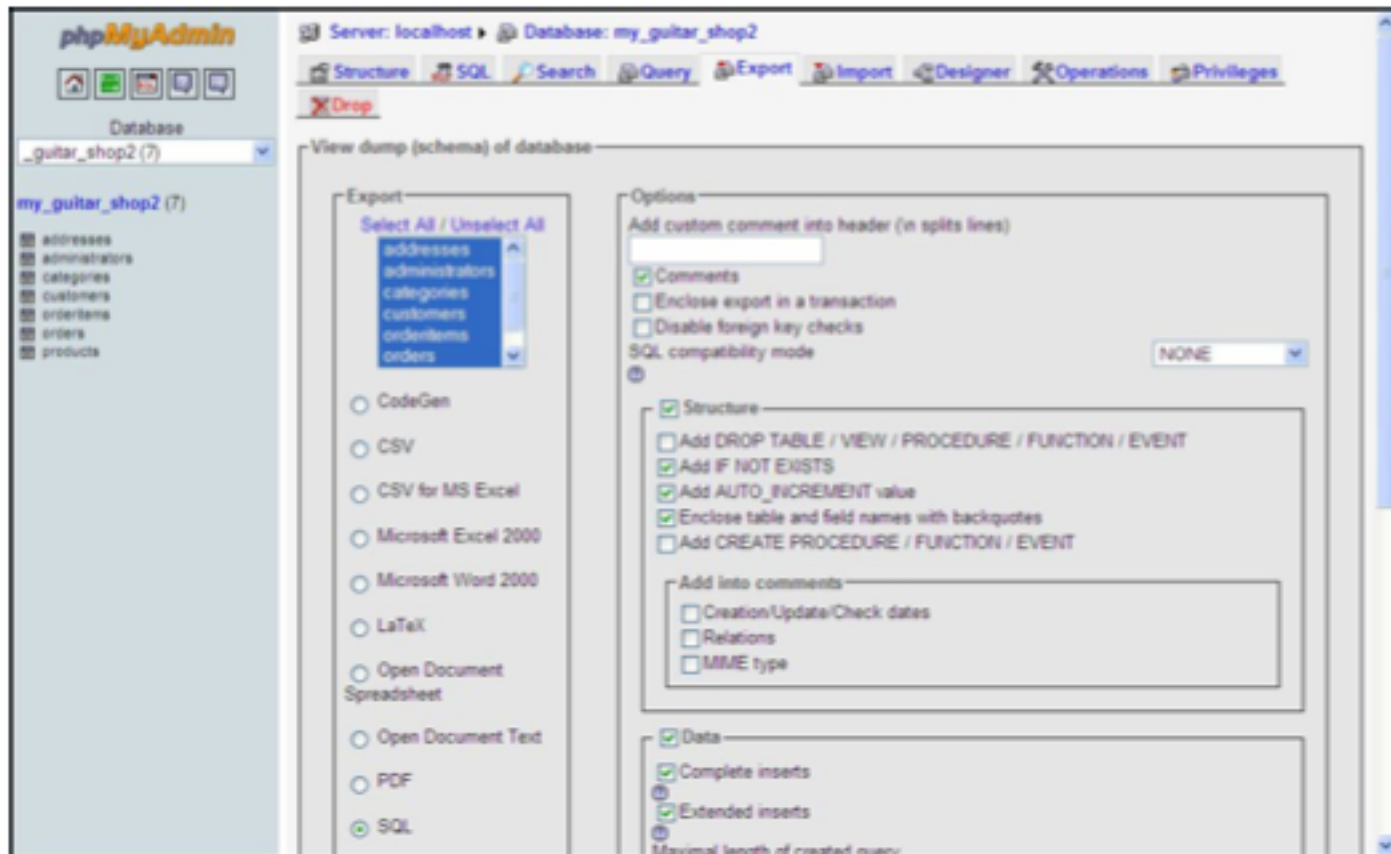
How to dump a database to a SQL script (cont.)

- How to use the Windows command prompt to dump a database to a SQL script

```
cd \xampp\mysql\bin
mysqldump -u root -p my_guitar_shop2 > my_guitar_shop2.sql
Enter password: *****
```

How to dump a database to a SQL script (cont.)

- The Export tab for the database



Summary

- By default MySQL uses the Latin character set, which provides for 256 characters with 1byte per character.
- The CHAR data type is typically used for fixed-length strings. The VARCHAR data type is typically used for variable-length strings.
- The precision of a DECIMAL type indicates the total number of digit can be stored in the data type.
- The CREATE TABLE statement creates a table based on the column definitions and columns attribute specify

Summary (2)

- A primary key requires that each row has a unique value for the column or columns for the primary key, and it doesn't allow NULL values.
- You code a column-level primary key as part of the definition of the column definition
- An index improves performance when MySQL searches for rows in the table
- A user can be granted privileges for working with different part of the data-base
- A script is a file that contain one or more SQL statement.