Setting Up the world Database

Table of Contents

This document describes `world` sample database installation, structure, usage, and history.

The sample data used in the `world` database is Copyright Statistics Finland, http://www.stat.fi/worldinfigures.

For legal information, see the Legal Notices.

For help with using MySQL, please visit the MySQL Forums, where you can discuss your issues with other MySQL users.

Document generated on: 2020-03-31 (revision: 65538)

1 Preface and Legal Notices

This document describes `world` sample database installation, structure, usage, and history.

Legal Notices

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

2 Installation

To enable MySQL users to perform manipulation of the `world` sample database using MySQL, the data set is available as a set of three tables:

- `country`: Information about countries of the world.
- `city`: Information about some of the cities in those countries.
- `countrylanguage`: Languages spoken in each country.

The `world.sql` file containing the `world` database is available for download at https://dev.mysql.com/doc/index-other.html. The file is provided as a compressed `tar` file or Zip archive.

> **Note**
>
> For users of MySQL 5.7 and higher, a `world_x` variant also exists that implements features related to X DevAPI. See Setting Up the world_x Database.

To install the `world` sample database, follow these steps:

1. Download the installation archive to a temporary location such as `C:\temp\` or `/tmp/` and unpack it. Unpacking the archive results in a single file named `world.sql`.

2. Connect to the MySQL server using the `mysql` command-line client with the following command:

   ```
   shell> mysql -u root -p
   ```

   Enter your password when prompted. A non-`root` account can be used, provided that the account has privileges to create new databases.

3. Execute the `world.sql` script to create the database structure and insert the data using the following command:

   ```
   mysql> SOURCE C:/temp/world.sql;
   ```

   Replace the path to the `world.sql` file with the actual path on your system.

   > **Note**
   >
   > On Windows, use slashes rather than backslashes when executing the `SOURCE` command.

4. To confirm that the sample `world` database is installed correctly, execute the following statements. You should see output similar to that shown here.

   ```
   mysql> USE world;
   Database changed

   mysql> SHOW TABLES;
   +-----------------+
   | Tables_in_world |
   +-----------------+
   | city            |
   | country         |
   | countrylanguage |
   +-----------------+
   3 rows in set (0.00 sec)

   mysql> SELECT COUNT(*) FROM city;
   +----------+
   | COUNT(*) |
   +----------+
   | 4079     |
   +----------+
   1 row in set (0.02 sec)

   mysql> SELECT COUNT(*) FROM country;
   +----------+
   | COUNT(*) |
   +----------+
   | 239      |
   +----------+
   1 row in set (0.00 sec)
   ```

You now have the `world` sample database installed.

Another popular sample database is the Sakila database. For additional details, see http://dev.mysql.com/doc/sakila/en/.

3 History

September 2016

Prior releases used mixed-case table names. Because MySQL Shell is case-sensitive, table names are changed to lowercase.

December 2019

The `world.sql` file was changed to address several issues:

- The `world` database used `latin1`, but MySQL as of 8.0 uses a default character set of `utf8mb4`. Converting the database to use `utf8mb4` brings it up to date with MySQL 8.0, while retaining compatibility with older series.

  Conversion of the database to `utf8mb4` was done as follows:

  ```
  USE world;

  -- turn off foreign key checking; otherwise, the ALTER TABLE
  -- statements fail with incompatible foreign key errors.
  SET SESSION foreign_key_checks=0;

  -- convert database and tables to utf8mb4
  ALTER DATABASE world CHARACTER SET utf8mb4;
  ALTER TABLE city CONVERT TO CHARACTER SET utf8mb4;
  ALTER TABLE country CONVERT TO CHARACTER SET utf8mb4;
  ALTER TABLE countrylanguage CONVERT TO CHARACTER SET utf8mb4;

  -- re-enable foreign key checking
  SET SESSION foreign_key_checks=0;
  ```

- MySQL Shell requires a character set of `utf8mb4` for X Protocol connections. Using such a connection to load a `latin1`-encoded `world.sql` file produces errors.

  Using `mysqldump` with options of `--default-character-set=utf8mb4` to set the character set and `--set-charset` so the dump includes `SET NAMES` for that character set writes a `utf8mb4`-encoded dump file. Changing the encoding of `world.sql` to `utf8mb4` permits it to be loaded without errors in MySQL Shell using X Protocol connections.

- MySQL 8.0.17 deprecates these features:

  - Number of digits in floating-point column definitions; for example, `FLOAT(10,2)`.
  - Display width in integer column definitions; for example, `INT(10)`.

  The `world.sql` file used both features, causing load warnings in MySQL 8.0.17 and higher.

  To avoid number of digits in `FLOAT` columns, it is necessary to manually alter the relevant columns. That was done as follows, by altering `FLOAT(M,D)` to `DECIMAL(M,D)`:

  ```
  USE world;

  ALTER TABLE country
    MODIFY SurfaceArea DECIMAL(10,2) NOT NULL DEFAULT '0.00',
    MODIFY LifeExpectancy DECIMAL(3,1) DEFAULT NULL,
    MODIFY GNP DECIMAL(10,2) DEFAULT NULL,
    MODIFY GNPOld DECIMAL(10,2) DEFAULT NULL;
  ALTER TABLE countrylanguage
    MODIFY Percentage DECIMAL(4,1) NOT NULL DEFAULT '0.0';
  ```

  To produce integer column definitions without display widths in the dump file, it suffices to use `mysqldump` from MySQL 8.0.19 or higher.