SQLite to MySQL Migration Plan

This guide provides a step-by-step approach to migrate an existing SQLite database into MySQL.

1. Install MySQL

On Ubuntu/Debian:
sudo apt update
sudo apt install mysql-server -y
sudo systemctl start mysql
sudo systemctl enable mysql

Secure the installation:
sudo mysql_secure_installation

Check MySQL status:
systemctl status mysql

Login to MySQL:
mysql -u root -p

2. Prepare SQLite Database

Ensure you have the SQLite database file, for example: database.sqlite3.

Install SQLite command-line tool:
sudo apt install sqlite3

Dump SQLite schema and data:
sqlite3 database.sqlite3 .dump > sqlite_dump.sql

3. Convert Schema for MySQL

The SQLite dump file is not fully compatible with MySQL. Adjustments are required:
- Replace "AUTOINCREMENT" with "AUTO_INCREMENT".
- Change "INTEGER PRIMARY KEY" to "INT PRIMARY KEY AUTO_INCREMENT".
- Replace "TEXT" with "VARCHAR(255)" where appropriate.
- Remove unsupported SQLite pragmas (e.g., "PRAGMA").
- Ensure date/time fields use DATETIME or TIMESTAMP in MySQL.

Optionally use a tool like `sqlite3-to-mysql`:
pip install sqlite3-to-mysql
sqlite3mysql --sqlite-file database.sqlite3 --mysql-database target_db --mysql-user root
--mysql-password password

4. Create MySQL Database

Login to MySQL and create the target database:
CREATE DATABASE target_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

5. Import Data

If using a manually converted dump:
mysql -u root -p target_db < mysql_ready_dump.sql

If using sqlite3-to-mysql:
The tool automatically migrates tables and data.

6. Verify Migration

- Check table structure in MySQL:
USE target_db;
SHOW TABLES;
DESCRIBE table_name;

- Verify data count matches SQLite:
SELECT COUNT(*) FROM table_name;

7. Update Application Configuration

If your app was using SQLite, update its configuration to point to MySQL instead, e.g.:

In Django settings.py:
DATABASES = {
'default': {
'ENGINE': 'django.db.backends.mysql',
'NAME': 'target_db',
'USER': 'root',
'PASSWORD': 'your_password',
'HOST': 'localhost',
'PORT': '3306',
}
}

In Flask (SQLAlchemy):
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql://root:password@localhost/target_db'

8. Final Notes

- Backup your SQLite database before migration.
- Test the application thoroughly after switching to MySQL.
- Optimize MySQL indexes and schema as needed.