# Requirement-

**Add application-level sharding to the Web Service API from Project 1, partitioning posts across SQLite databases.**

### Databases

Replace the single SQLite database configuration from the previous project with four SQLite databases: three databases holding shards of the table (or tables) containing posts, and one database containing the rest of the tables. You will need to modify your code in order to store connections to all databases [in the application context](http://flask.pocoo.org/docs/1.0/appcontext/" \l "storing-data).

### Shard Key, Mapping, and Data Model

Split your posts into shards based on the thread ID so that all posts for a given thread can be retrieved from a single database. Map the shard key modulo 3 to the database partition.

Since thread ID is used as the shard key, operations involving sharded data (creating a new thread, adding a post to a thread, and listing the posts to a thread) should only require access to a single database partition.

#### Using Thread IDs as Shard Keys

Since the SQLite databases are independent, using integer autoincrement or similar values as primary keys in sharded tables will lead to duplicate values. One way to solve this problem is to use [globally unique identifiers](https://en.wikipedia.org/wiki/Universally_unique_identifier) as keys.

While SQLite does not directly support GUIDs, see the StackOverflow answer [*Proper way to store GUID in sqlite*](https://stackoverflow.com/a/18842491) for an example of configuring the Python sqlite3 module to use [uuid.UUID](https://docs.python.org/2.7/library/uuid.html) objects as primary keys.

### Database Population

You will need to update your Flask CLI init\_db command to create the same tables in each shard and insert test data items into the correct shard.

### Tips

* Not all primary keys in the application need to be GUIDs: only those that may conflict across sharded tables.
* Since SQLite does not directly support GUIDs, you may find it easier to insert sharded test data from Python code rather than directly from a .sql file.