

SQL > Database Manipulation

Create Tables

We'll make some tables using [SQL CREATE TABLE](#) statements. We recommend writing your SQL in some file just to make it easy to save and edit. You can then copy and paste from this file into your sqlite shell to execute the commands.

Below are the three tables we need. After looking at the linked guide above, attempt to write the code yourself and then open the hint and compare the answer to your code.

Add the following attributes to the `movies` table.

- `id`
 - type `INTEGER`
 - primary key
- `name`
 - type `TEXT`
 - default value `NULL`
- `year`
 - type `INTEGER`
 - default value `NULL`
- `rank`
 - type `REAL`
 - default value `NULL`

Hint: CREATE movies

Click to toggle hint 

```
CREATE TABLE movies (  
  id INTEGER PRIMARY KEY,  
  name TEXT DEFAULT NULL,  
  year INTEGER DEFAULT NULL,  
  
  rank REAL DEFAULT NULL  
);
```

WARNING: *Primary key* columns only auto-increment properly if they are of type `INTEGER` . Synonyms such as `INT` cause such columns to behave as normal data fields - [reference](#).

Add the following attributes to the `actors` table.

- `id`
 - type `INTEGER`
 - primary key
- `first_name`
 - type `TEXT`
 - default value `NULL`
- `last_name`
 - type `TEXT`
 - default value `NULL`
- `gender`
 - type `TEXT`
 - default value `NULL`

Hint: CREATE actors

Click to toggle hint ▼

Add the following attributes to the `roles` table.

- `actor_id`
 - type `INTEGER`
- `movie_id`
 - type `INTEGER`
- `role_name`
 - type `TEXT`
 - DEFAULT value `NULL`

Hint: CREATE roles

Click to toggle hint ▼

NOTE: SQLite's `CREATE TABLE` command definition is similar, but not the same, as the Postgres `CREATE TABLE` . Once you learn the concept however, you won't have trouble learning the nuances of the Postgres `CREATE TABLE` command either.

