

# Testing a Database

You can use the same dependency overrides from [Testing Dependencies with Overrides](#) to alter a database for testing.

You could want to set up a different database for testing, rollback the data after the tests, pre-fill it with some testing data, etc.

The main idea is exactly the same you saw in that previous chapter.

## Add tests for the SQL app

Let's update the example from [SQL \(Relational\) Databases](#) to use a testing database.

All the app code is the same, you can go back to that chapter check how it was.

The only changes here are in the new testing file.

Your normal dependency `get_db()` would return a database session.

In the test, you could use a dependency override to return your *custom* database session instead of the one that would be used normally.

In this example we'll create a temporary database only for the tests.

## File structure

We create a new file at `sql_app/tests/test_sql_app.py`.

So the new file structure looks like:

```
.
└─ sql_app
    ├── __init__.py
    ├── crud.py
    ├── database.py
    ├── main.py
    ├── models.py
    ├── schemas.py
    └─ tests
        ├── __init__.py
        └─ test_sql_app.py
```

## Create the new database session

First, we create a new database session with the new database.

For the tests we'll use a file `test.db` instead of `sql_app.db`.

But the rest of the session code is more or less the same, we just copy it.

```
{!../../../docs_src/sql_databases/sql_app/tests/test_sql_app.py!}
```

!!! tip You could reduce duplication in that code by putting it in a function and using it from both `database.py` and `tests/test_sql_app.py`.

For simplicity and to focus on the specific testing code, we are just copying it.

## Create the database

Because now we are going to use a new database in a new file, we need to make sure we create the database with:

```
Base.metadata.create_all(bind=engine)
```

That is normally called in `main.py`, but the line in `main.py` uses the database file `sql_app.db`, and we need to make sure we create `test.db` for the tests.

So we add that line here, with the new file.

```
{!../../../../../docs_src/sql_databases/sql_app/tests/test_sql_app.py!}
```

## Dependency override

Now we create the dependency override and add it to the overrides for our app.

```
{!../../../../../docs_src/sql_databases/sql_app/tests/test_sql_app.py!}
```

!!! tip The code for `override_get_db()` is almost exactly the same as for `get_db()`, but in `override_get_db()` we use the `TestingSessionLocal` for the testing database instead.

## Test the app

Then we can just test the app as normally.

```
{!../../../../../docs_src/sql_databases/sql_app/tests/test_sql_app.py!}
```

And all the modifications we made in the database during the tests will be in the `test.db` database instead of the main `sql_app.db`.