Database Unit Tests

Flath, Dakota Volberg, Devon

September 26, 2023

1 Null-Insert Testing

1.1 Objective

Null testing is the simple procedure of attempting an insert into a table with all fields of the row being null. This will be done to every table that has a mandatory field, which will be all of them, and every single test should result in a failed insert. This test is also weak constraint checking, since null fields will never be within a constraint. When this test is ran, it will attempt to insert entries into each table with all null values.

1.2 Example Test Input

```
INSERT INTO Companies (ID, cName, StockID) VALUES (null, null, null)
```

1.3 Expected Output

We expect the database to reject insertions where every field is a null value.

We also expect the database to reject null inputs on fields with not-null constraints placed on them (primary/foreign keys, other fields based on business rules).

1.3.1 Expected Exceptions

SQLAlchemy

```
DataError
```

2 Connection Failure

2.1 Objective

To test for database connection failure in SQLAlchemy.

2.2 Test Code

```
import logging

from sqlalchemy import create_engine, text
from sqlalchemy.exc import OperationalError

try:
    db_url = "mysql+pymysql://root:incorrect@localhost:13306/null_database"
    engine = create_engine(db_url, pool_size=5, pool_recycle=3600)
    conn = engine.connect()
except OperationalError as err:
    logging.error("Cannot connect to DB %s", err)
    raise err
```

2.3 Expected Output

```
OperationalError: (pymysql.err.OperationalError) (1045, "Access
   denied for user 'root'@'172.17.0.1' (using password: YES)")
```

2.3.1 Expected Exceptions

```
pymysql.err.OperationalError
```