

| Test No | Test Description | Result |
|---------|--|---------------|
| 1 | Database Connection: ensure python connection to db allows queries | <i>PASSED</i> |
| 2 | SQL query for primary key works. Check that sql for "TAB-000", works to return the next value | <i>PASSED</i> |
| 3 | INSERT/UPDATE/DELETE in Single Table. Ensure that table(s) can be edited (following any constraints) | <i>PASSED</i> |
| 4 | INSERT into Db breaking Key constraints. Force INSERT error to confirm constraint functions as intended | <i>PASSED</i> |
| 5 | INSERT into Db breaking UNIQUE constraints. Force INSERT error to confirm constraint functions as intended | <i>PASSED</i> |

| Test No | Test Description | Result |
|---------|--|---------------|
| 6 | Trigger Creation & execution. Confirm creation without error | <i>PASSED</i> |
| 7 | Trigger INSERT/DELETE. Confirm trigger creation & required outputs without error (NEW only for INSERT, and OLD for DELETE) | <i>PASSED</i> |
| 8 | Trigger UPDATE. Confirm trigger creation & required outputs without error (NEW and OLD data pulled) | <i>PASSED</i> |
| 9 | VIEW creation & execution. Ensure data is returned as expected for created VIEW | <i>PASSED</i> |

| Test No | Test Description | Result | Test Data/Screenshots |
|---------|---|--------|-----------------------|
| 1 | <p>Database Connection:</p> <p>Ensure python connection to db allows queries. Setup pytest to connect to the database & run a confirmed script to assert that data has been returned.</p> | PASSED | Below |

```

(.venv) PS C:\Users\eliot\Documents\COM519> pytest -v tests/test_utility_functions.py::test_database_connection_pass
===== test session starts =====
platform win32 -- Python 3.14.0, pytest-9.0.1, pluggy-1.6.0 -- c:\Users\eliot\Documents\COM519\.venv\Scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\eliot\Documents\COM519
collected 1 item

tests/test_utility_functions.py::test_database_connection_pass PASSED

===== 1 passed in 0.81s =====

```

Full test script:

```

def test_database_connection_pass():
    """
    confirm that database connection works & passed back in function
    """

    import database as db

    settings = get_settings_data()

    db_path = os.path.join(
        settings["database_settings"]["database_path"],
        settings["database_settings"]["database"]
    )

    # Create a connection
    conn = db.Database(db_path)

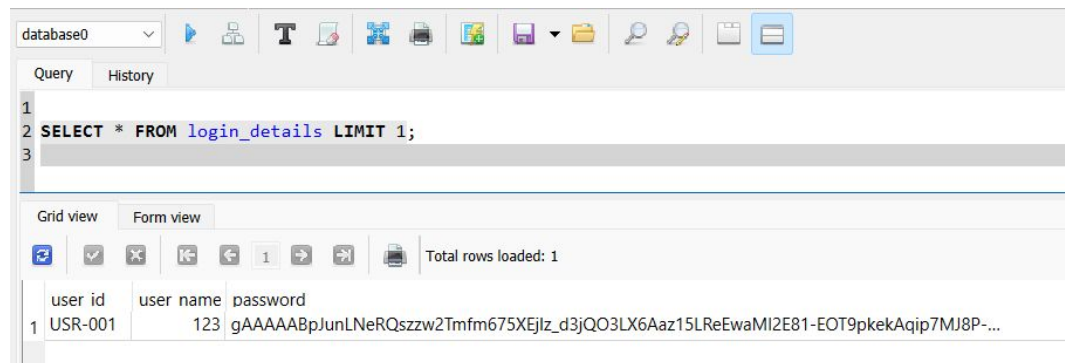
    # Test connection
    test_sql = load_sql_file("database_connection_check.sql")
    result = conn.query(test_sql, ())

    conn.close()

    assert result != None

```

Confirm run from SQLiteStudio



| Test No | Test Description | Result | Test Data/Screenshots |
|---------|--|--------|-----------------------|
| 3 | INSERT/UPDATE/DELETE in Single Table. Ensure that table(s) can be edited (following any constraint requirements - Happy Path) | PASSED | below |

Insert & UPDATE script(s) into SQLiteStudio for user creation:

```

1 INSERT INTO users(
2     user_id,name,address,postcode_id,email,phone_no,primary_garage,access_code,account_creation_date,active_flag)
3 VALUES
4 ('USR-099','Amelia Jones','Flat 4, 22 King's Road, London','PST-001','amelia.jones@email.com','07234 567890','N/A','CUS_USR','2025-10-25',1)
5 ;
6
UPDATE users SET
user_id = "USR-002",name = "Amelia Jones",address= "Flat 4, 22 King's Road, London",
postcode_id = "PST-002",email = "amelia.jones@email.com",phone_no = "07234 567890",primary_garage = "N/A",access_code = "CUS_USR"
,account_creation_date = "2025-10-25",active_flag = 1
WHERE user_id = "USR-002"

```

Insert & UPDATE script(s) into SQLiteStudio for user creation, then delete the user without any constraint errors:

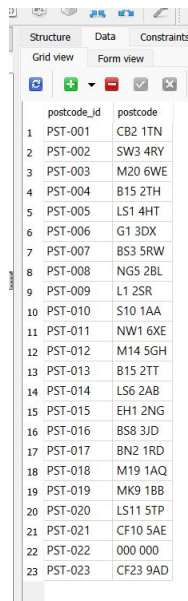
| | | | | |
|---|---------|--------|---------|---|
| 2 | AUD-002 | CREATE | USR-099 | - |
| 3 | AUD-003 | UPDATE | USR-002 | {"name":"Amelia Jones","address":"Flat 4, 22 King's Road, ... |
| 6 | AUD-006 | DELETE | USR-099 | - |
| - | AUD-007 | CREATE | USR-000 | - |

Query finished in 0.000 second(s).

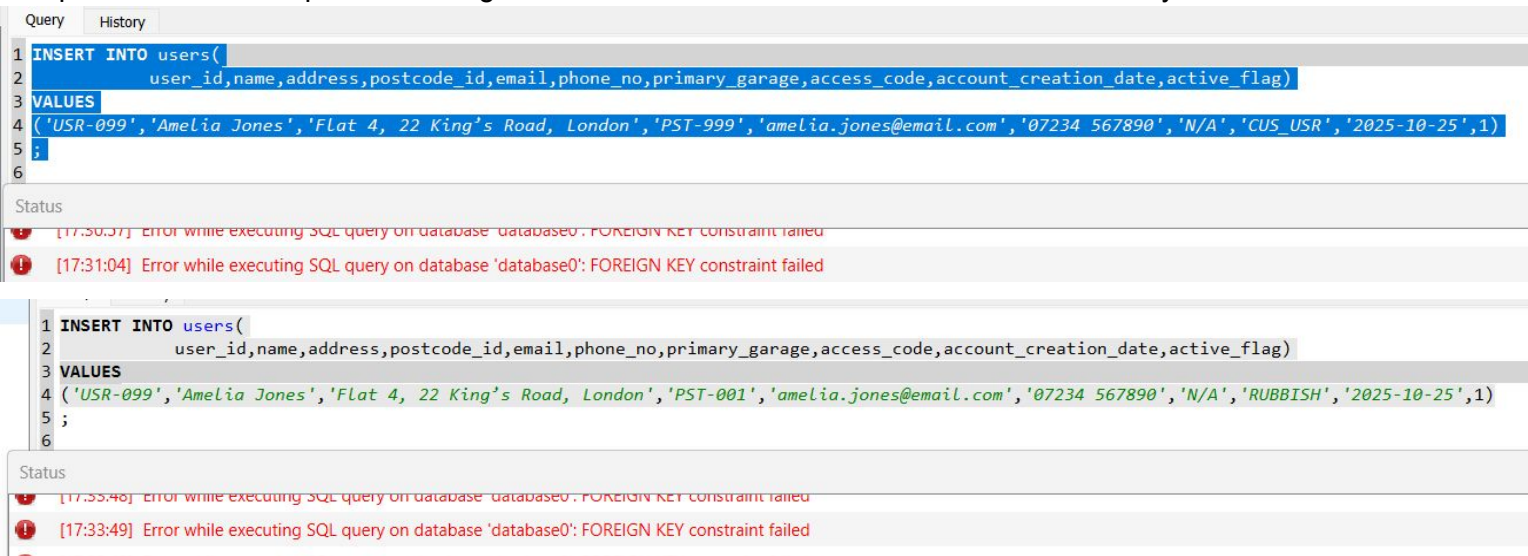
| Test No | Test Description | Result | Test Data/Screenshots |
|---------|---|--------|-----------------------|
| 4 | INSERT into Db breaking Key constraints. Attempt INSERT error to confirm constraint functions as intended. UNIQUE / Non-existence postcode | PASSED | below |

Script to attempt input of postcode 999 which doesn't exist within the postcode table:

Sql return error as expected. Changed to accesscode = "Rubbish" to confirm additionally:



| postcode_id | postcode |
|-------------|----------|
| 1 PST-001 | CB2 1TN |
| 2 PST-002 | SW3 4RY |
| 3 PST-003 | M20 6WE |
| 4 PST-004 | B15 2TH |
| 5 PST-005 | LS1 4HT |
| 6 PST-006 | G1 3DX |
| 7 PST-007 | BS3 5RW |
| 8 PST-008 | NG5 2BL |
| 9 PST-009 | L1 2SR |
| 10 PST-010 | S10 1AA |
| 11 PST-011 | NW1 6XE |
| 12 PST-012 | M14 5GH |
| 13 PST-013 | B15 2TT |
| 14 PST-014 | LS6 2AB |
| 15 PST-015 | EH1 2NG |
| 16 PST-016 | BS8 3JD |
| 17 PST-017 | BN2 1RD |
| 18 PST-018 | M19 1AQ |
| 19 PST-019 | MK9 1BB |
| 20 PST-020 | LS11 5TP |
| 21 PST-021 | CF10 5AE |
| 22 PST-022 | 000 000 |
| 23 PST-023 | CF23 9AD |



Query History

```

1 INSERT INTO users(
2     user_id,name,address,postcode_id,email,phone_no,primary_garage,access_code,account_creation_date,active_flag)
3 VALUES
4 ('USR-099','Amelia Jones','Flat 4, 22 King's Road, London','PST-999','amelia.jones@email.com','07234 567890','N/A','CUS_USR','2025-10-25',1)
5 ;
6

```

Status

[17:30:37] Error while executing SQL query on database 'database0': FOREIGN KEY constraint failed

[17:31:04] Error while executing SQL query on database 'database0': FOREIGN KEY constraint failed

```

1 INSERT INTO users(
2     user_id,name,address,postcode_id,email,phone_no,primary_garage,access_code,account_creation_date,active_flag)
3 VALUES
4 ('USR-099','Amelia Jones','Flat 4, 22 King's Road, London','PST-001','amelia.jones@email.com','07234 567890','N/A','RUBBISH','2025-10-25',1)
5 ;
6

```

Status

[17:33:46] Error while executing SQL query on database 'database0': FOREIGN KEY constraint failed

[17:33:49] Error while executing SQL query on database 'database0': FOREIGN KEY constraint failed

| Test No | Test Description | Result | Test Data/Screenshots |
|---------|--|---------------|-----------------------|
| 5 | INSERT into Db breaking UNQIUE constraints. Force INSERT error to confirm constraint functions as intended. Script attempting to create “USR-001” again | <i>PASSED</i> | below |

Script to attempt input of “USR-001” which already exist within the user table:

Structure

Data

Constraints

Indexes

Triggers

DDL

Grid view

Form view

Sql return error as expected. UNIQUE constraint was broken no INSERT allowed:

| Query | History |
|---|---------|
| <pre> 1 INSERT INTO users(2 user_id,name,address,postcode_id,email,phone_no,primary_garage,access_code,account_creation_date,active_flag) 3 VALUES 4 ('USR-001','Amelia Jones','Flat 4, 22 King's Road, London','PST-001','amelia.jones@email.com','07234 567890','N/A','CUS_USR','2025-10-25',1) 5 ; 6 </pre> | |
| Status | |
| [17:35:40] Error while executing SQL query on database 'database0': UNIQUE constraint failed: users.user_id | |
| [17:35:41] Error while executing SQL query on database 'database0': UNIQUE constraint failed: users.user_id | |

| Test No | Test Description | Result | Test Data/Screenshots |
|---------|--|--------|-----------------------|
| 6 | Trigger Creation & execution. Confirm creation without error | PASSED | below |
| 7 | Trigger INSERT/DELETE. Confirm trigger creation & required outputs without error (NEW only for INSERT, and OLD for DELETE) | PASSED | below |

INSERT and Delete scripts run after trigger successful creation.
Returned success for both trigger creation & scripts, confirmation within audit table that update for changes (INSERT/DELETE) completed:

```
Query History
1 INSERT INTO users(
2     user_id,name,address,postcode_id,email,phone_no,primary_garage,access_code,account_creation_date,active_flag)
3 VALUES
4 ('USR-099', 'Amelia Jones', 'Flat 4, 22 King's Road, London', 'PST-001', 'amelia.jones@email.com', '07234 567890', 'N/A', 'CUS_USR', '2025-10-25
5 ;
6
7 DELETE FROM users WHERE user_id = 'USR-099';
```

| | | | | |
|-----|---------|--------|---------|---|
| 168 | AUD-168 | CREATE | USR-099 | - |
| 169 | AUD-169 | DELETE | USR-099 | {"user_id":"USR-099","name":"Amelia Jones","address":"Flat 4, |

```

1 CREATE TRIGGER trg_users_after_delete
2 AFTER DELETE
3 ON users
4 FOR EACH ROW
5 BEGIN
6     INSERT INTO audit_table (
7         audit_record_id,
8         action_type,
9         primary_key_ref,
10        change_from,
11        change_to,
12        date_of_change,
13        time_of_change
14    )
15    VALUES (
16        'AUD-' || printf('%03d', COALESCE(
17            SELECT MAX(CAST (substr(audit_record_id, 5) AS INTEGER))
18            FROM audit_table
19            ), 0) + 1),
20        'DELETE',
21        OLD.user_id,
22        NULL
23    );
24 END;

```

Grid view Form view

Total rows loaded: 0

Status

[17:39:53] Query finished in 0.012 second(s).

| Test No | Test Description | Result | Test Data/Screenshots |
|---------|---|--------|-----------------------|
| 8 | <p>Trigger UPDATE.</p> <p>Confirm trigger creation & required outputs without error (NEW and OLD data pulled)</p> | PASSED | below |

UPDATE scripts run removing “S” and then returning, after each trigger successfully executed into audit table for update:

```

1 INSERT INTO users(
2     user_id,name,address,postcode_id,email,phone_no,primary_garage,access_code,account_creation_date,active_flag)
3 VALUES
4 ('USR-002','Amelia Jone','Flat 4, 22 King's Road, London','PST-001','amelia.jones@email.com','07234 567890','N/A','CUS_USR','2025-10-25',1)
5 ;
6

```

| | | | | |
|-----|---------|--------|---------|---|
| 166 | AUD-166 | UPDATE | USR-002 | { "user_id": "USR-002", "name": "Amelia Jone", "address": "Flat 4, 22 King's Road, ... |
| 167 | AUD-167 | UPDATE | USR-002 | { "user_id": "USR-002", "name": "Amelia Jones", "address": "Flat 4, 22 King's Road, ... |
| 168 | AUD-168 | CREATE | USR-000 | - |

| audit_record_id | action_type | primary_key_ref | change_from | change_to | date_of_change | time_of_change |
|-----------------|-------------|-----------------|--|--|----------------|----------------|
| AUD-166 | UPDATE | USR-002 | { "user_id": "USR-002", "name": "Amelia Jone", "address": "Flat 4, 22 King's Road, London", "postcode_id": "PST-002", "email": "amelia.jones@email.com", "phone_no": "07234 567890", "primary_garage": "N/A", "access_code": "CUS_USR", "account_creation_date": "2025-10-25", "active_flag": 1 } | { "user_id": "USR-002", "name": "Amelia Jones", "address": "Flat 4, 22 King's Road, London", "postcode_id": "PST-002", "email": "amelia.jones@email.com", "phone_no": "07234 567890", "primary_garage": "N/A", "access_code": "CUS_USR", "account_creation_date": "2025-10-25", "active_flag": 1 } | 2025-12-17 | 17:21:22 |
| AUD-167 | UPDATE | USR-002 | { "user_id": "USR-002", "name": "Amelia Jones", "address": "Flat 4, 22 King's Road, London", "postcode_id": "PST-002", "email": "amelia.jones@email.com", "phone_no": "07234 567890", "primary_garage": "N/A", "access_code": "CUS_USR", "account_creation_date": "2025-10-25", "active_flag": 1 } | { "user_id": "USR-002", "name": "Amelia Jone", "address": "Flat 4, 22 King's Road, London", "postcode_id": "PST-002", "email": "amelia.jones@email.com", "phone_no": "07234 567890", "primary_garage": "N/A", "access_code": "CUS_USR", "account_creation_date": "2025-10-25", "active_flag": 1 } | 2025-12-17 | 17:25:06 |