



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Разработка базы данных для IoT-платформы умный дом»

Студент ИУ7-66Б
(Группа)

(Подпись, дата)

Мамврийский И. С.
(И. О. Фамилия)

Руководитель курсовой работы

(Подпись, дата)

Гаврилова Ю. М.
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

Введение

В настоящее время технологии интернета вещей (IoT) становятся неотъемлемой частью нашей повседневной жизни, и одним из наиболее заметных примеров их применения является IoT-платформа для умного дома. Она объединяет различные устройства в доме, от светильников до умных термостатов, в единую сеть, которая может управляться и контролироваться через интернет.

Основной целью IoT-платформы умного дома является создание интеллектуальной инфраструктуры, способной адаптироваться к потребностям и предпочтениям пользователей. Она обеспечивает возможность контролировать освещение, отопление, кондиционирование воздуха, безопасность и другие аспекты жизни в доме с помощью смартфона или другого устройства с доступом в интернет.

С учетом быстрого развития технологий IoT и роста спроса на умные системы, платформы умного дома становятся все более интегрированными, расширяя свои возможности и предлагая новые функции для улучшения качества жизни пользователей.

Целью курсовой работы является разработка базы данных для IoT-платформы умный дом. Для достижения поставленной цели необходимо выполнить следующие задачи:

Приложение А

Листинг 1 – Реализация функции, используемой для смены статуса устройства в соответствии с текущим состоянием

```
1 CREATE FUNCTION update_status(device_id integer, newStatus
    varchar(10))
2 RETURNS integer AS $$
3 DECLARE
4     current_status varchar(10);
5 BEGIN
6     SELECT status INTO current_status FROM device WHERE deviceid =
        device_id;
7
8     IF NOT FOUND THEN
9         RETURN -1;
10    END IF;
11
12    IF current_status = 'active' AND newStatus = 'active' THEN
13        RETURN -2;
14    END IF;
15
16    IF current_status = 'inactive' AND newStatus = 'inactive' THEN
17        RETURN -3;
18    END IF;
19
20    UPDATE device
21    SET status = newStatus
22    WHERE deviceid = device_id;
23
24    RETURN 0;
25 END;
26 $$ LANGUAGE plpgsql;
```

Приложение Б

Листинг 2 – Создание роли гостя (неавторизованного пользователя)

```
1 CREATE ROLE guest with
2     NOSUPERUSER
3     NOCREATEDB
4     NOCREATEROLE
5     NOINHERIT
6     NOBYPASSRLS
7     NOREPLICATION
8     LOGIN
9     PASSWORD 'guest'
10    connection limit -1;
11
12 GRANT select
13     on public."client"
14     TO guest;
15
16 GRANT INSERT
17     on public."client"
18     TO guest;
```

Листинг 3 – Создание роли авторизованного пользователя

```
1 CREATE ROLE authorized with
2     NOSUPERUSER
3     NOCREATEDB
4     NOCREATEROLE
5     NOINHERIT
6     NOBYPASSRLS
7     NOREPLICATION
8     LOGIN
9     PASSWORD 'guest'
10    CONNECTION LIMIT -1;
11
12 GRANT SELECT
13     ON ALL TABLES IN SCHEMA public
14     TO authorized;
15
16 GRANT INSERT (homeid, coords, name)
17     ON public.home
18     TO authorized;
```

Листинг 4 – Создание роли владельца дома

```
1 CREATE ROLE owner
2     NOSUPERUSER
3     NOCREATEDB
4     NOREPLICATION
5     LOGIN
6     PASSWORD 'owner'
7     CONNECTION LIMIT -1;
8
9 GRANT ALL PRIVILEGES
10    ON ALL TABLES IN SCHEMA public
11    TO owner;
```

Приложение В

```
2024/08/26 14:59:25 github.com/testcontainers/testcontainers-go - Connected to docker:
  Server Version: 20.10.20
  API Version: 1.41
  Operating System: Docker Desktop
  Total Memory: 7859 MB
  Testcontainers for Go Version: v0.32.0
  Resolved Docker Host: unix:///var/run/docker.sock
  Resolved Docker Socket Path: /var/run/docker.sock
  Test SessionID: 2ca83f8c7543b90ef8a97ad139319283f40fcf68eebeaf7ea9e8333f3f815dbd
  Test ProcessID: 75a4a10c-51a5-4bad-b1ae-e33f170b83a4
2024/08/26 14:59:25 🚀 Creating container for image testcontainers/ryuk:0.7.0
2024/08/26 14:59:25 ✅ Container created: 1bca6d5fedfc
2024/08/26 14:59:25 🚀 Starting container: 1bca6d5fedfc
2024/08/26 14:59:26 ✅ Container started: 1bca6d5fedfc
2024/08/26 14:59:26 ⚠️ Waiting for container id 1bca6d5fedfc image: testcontainers/ryuk:0.7.0. Waiting for: &{Port:8080/tcp timeout:<nil> PollInterval:100ms}
2024/08/26 14:59:26 🟡 Container is ready: 1bca6d5fedfc
2024/08/26 14:59:26 🚀 Creating container for image postgres:15.4-alpine
2024/08/26 14:59:27 ✅ Container created: 9bce8e2b5273
2024/08/26 14:59:27 🚀 Starting container: 9bce8e2b5273
2024/08/26 14:59:27 ✅ Container started: 9bce8e2b5273
2024/08/26 14:59:27 ⚠️ Waiting for container id 9bce8e2b5273 image: postgres:15.4-alpine. Waiting for: &{Port:5432/tcp timeout:<nil> PollInterval:100ms}
2024/08/26 14:59:30 🟡 Container is ready: 9bce8e2b5273
2024/08/26 14:59:30 OK 20240822064416_defaulttesttable.sql
2024/08/26 14:59:30 OK 20240822082318_limitation.sql
2024/08/26 14:59:30 OK 20240825153153_functiontest.sql
2024/08/26 14:59:30 goose: no migrations to run. current version: 20240825153153
=== RUN TestCreateClient
=== RUN TestCreateClient/Test1
=== RUN TestCreateClient/Test2
=== RUN TestCreateClient/Test3
--- PASS: TestCreateClient (0.03s)
--- PASS: TestCreateClient/Test1 (0.01s)
--- PASS: TestCreateClient/Test2 (0.01s)
--- PASS: TestCreateClient/Test3 (0.01s)
```

Рисунок 1 – Результаты тестирования работы запросов к базе данных

```
=== RUN TestGetClient
=== RUN TestGetClient/Test1
=== RUN TestGetClient/Test2
=== RUN TestGetClient/Test3
--- PASS: TestGetClient (0.03s)
--- PASS: TestGetClient/Test1 (0.01s)
--- PASS: TestGetClient/Test2 (0.01s)
--- PASS: TestGetClient/Test3 (0.01s)
=== RUN TestUpdateStatusFunc
=== RUN TestUpdateStatusFunc/Test1
=== RUN TestUpdateStatusFunc/Test2
=== RUN TestUpdateStatusFunc/Test3
=== RUN TestUpdateStatusFunc/Test4
=== RUN TestUpdateStatusFunc/Test5
=== RUN TestUpdateStatusFunc/Test6
=== RUN TestUpdateStatusFunc/Test7
=== RUN TestUpdateStatusFunc/Test8
--- PASS: TestUpdateStatusFunc (0.08s)
--- PASS: TestUpdateStatusFunc/Test1 (0.01s)
--- PASS: TestUpdateStatusFunc/Test2 (0.01s)
--- PASS: TestUpdateStatusFunc/Test3 (0.01s)
--- PASS: TestUpdateStatusFunc/Test4 (0.01s)
--- PASS: TestUpdateStatusFunc/Test5 (0.00s)
--- PASS: TestUpdateStatusFunc/Test6 (0.01s)
--- PASS: TestUpdateStatusFunc/Test7 (0.01s)
--- PASS: TestUpdateStatusFunc/Test8 (0.01s)
=== RUN TestCreateHome
=== RUN TestCreateHome/Test1
=== RUN TestCreateHome/Test2
=== RUN TestCreateHome/Test3
--- PASS: TestCreateHome (0.03s)
--- PASS: TestCreateHome/Test1 (0.01s)
--- PASS: TestCreateHome/Test2 (0.01s)
--- PASS: TestCreateHome/Test3 (0.01s)
PASS
ok      github.com/Mamvriyskiy/database_course/main/testsdatabase 5.943s
```

Рисунок 2 – Результаты тестирования работы запросов к базе данных