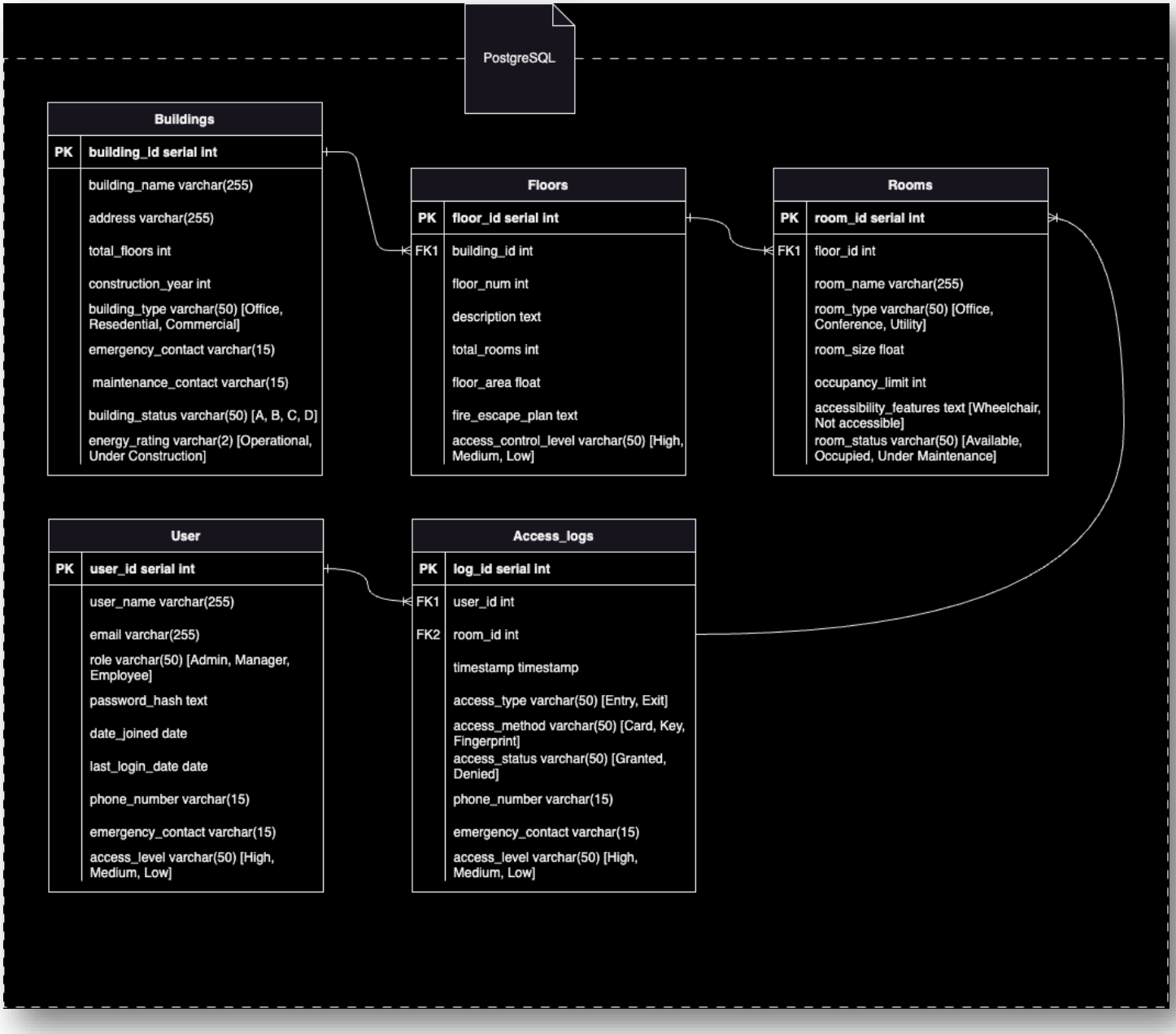
CSE 512: Distributed Database Systems

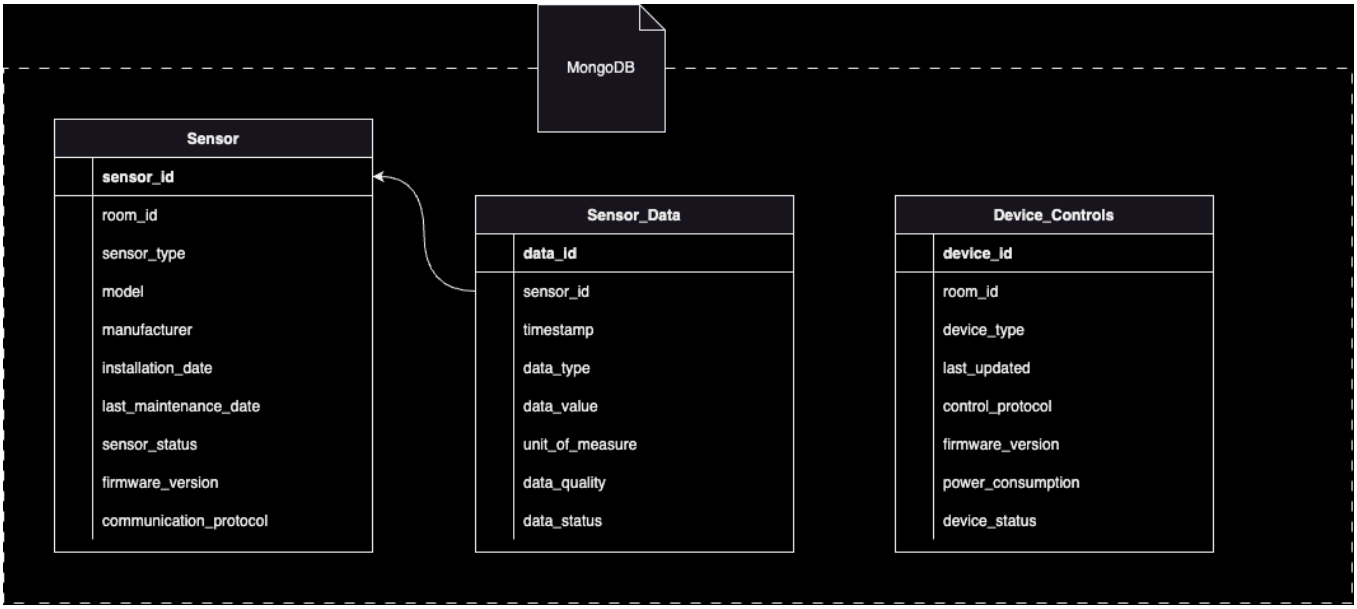
Project: Distributed Database System for a Smart Building

Part – 1

Schema Diagram for Relational Database:



Schema Diagram for NoSQL Database:



Data Distribution Strategy

1. **PostgreSQL Database:**

Tables:

- **Building**
- **Floor**
- **Room**
- **User**
- **Access Log**

Relationships:

- Tables are structured based on the entity types representing physical spaces (**Building**, **Floor**, **Room**) and users (**User**).
- The **Access Log** table captures logs related to user access.

2. **MongoDB Database:**

Collections:

- **Sensor**
- **Sensor Data**
- **Device Control**

Document Structure:

- Data related to devices and sensors is stored in MongoDB collections.

- Devices are categorized into different types (**Thermostat**, **Lighting**, **Air Conditioner**, **CCTV**).
- Sensor data is organized based on different types (**temperature**, **humidity**, **light**, **motion**).
- Each document within a collection represents a specific device or sensor along with its properties.

Justification

- PostgreSQL is used for relational data, capturing information about physical spaces, users, and access logs.
- MongoDB is chosen for its flexibility in handling unstructured data, making it suitable for storing data related to various types of sensors and devices.
- The distribution aligns with the specific strengths of each database system, optimizing performance and ease of data retrieval for the given use cases.

Advantages:

1. **Efficiency:**
    - Each database system is utilized for its specific strengths, optimizing efficiency in data storage and retrieval.
2. **Scalability:**
    - MongoDB's flexible schema allows easy scalability for adding new types of sensors or devices without significant changes to the data structure.
3. **Performance:**
    - PostgreSQL's relational structure is suitable for efficient querying and retrieval of structured data related to physical spaces and users.
4. **Ease of Maintenance:**
    - Logical separation of data types into different databases simplifies maintenance tasks and enhances system manageability.
5. **Suitability:**
    - The chosen strategy is aligned with the specific use cases, providing an optimal balance between relational and non-relational data storage requirements.

Data Retrieval Proof:

Running the following queries to retrieve data from MongoDB:

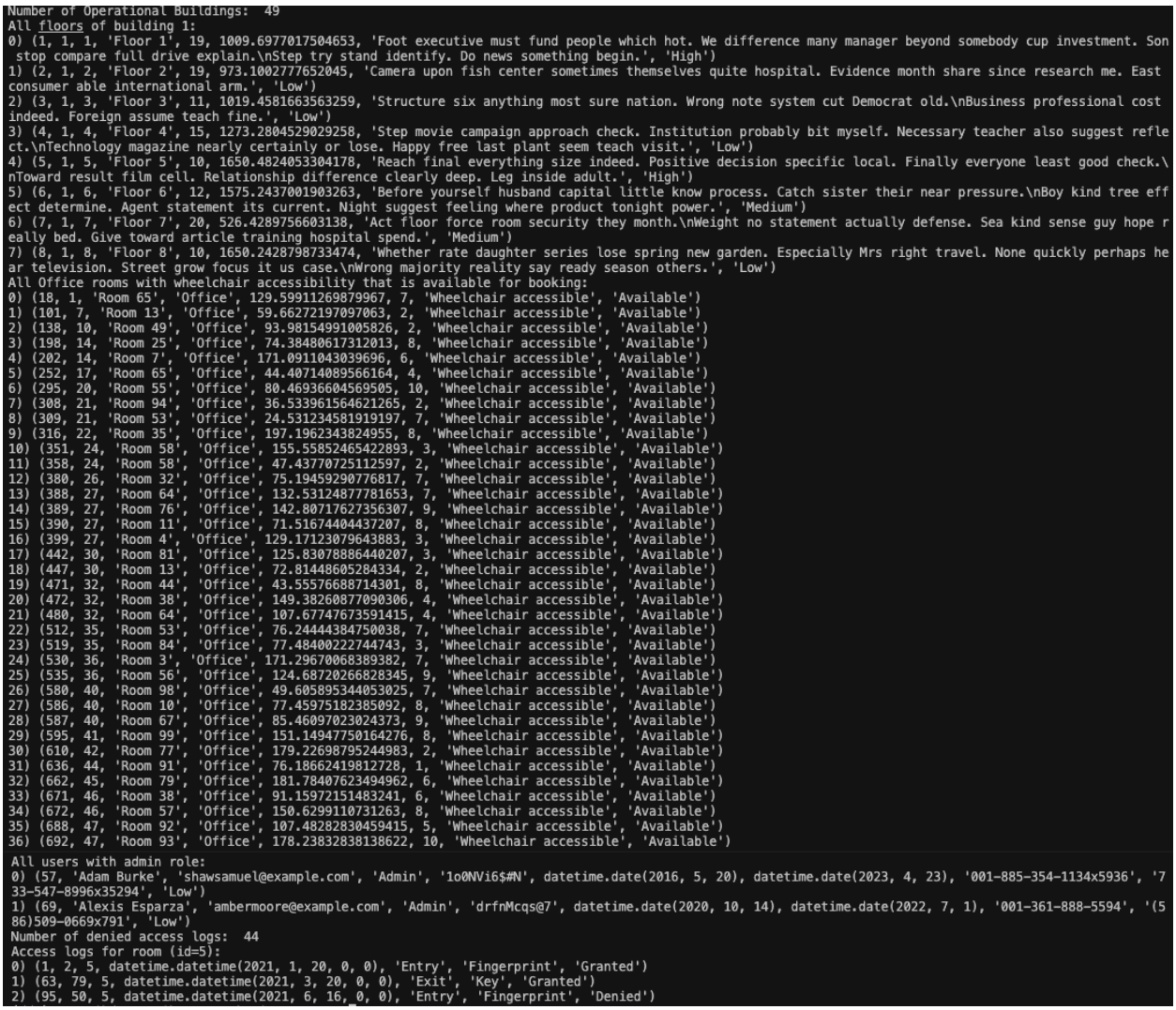
1. Find all sensors with sensor type ‘Temperature’ and model ‘T3000’.
2. Count all inactive sensors.
3. Find Sensor data with id = 158.
4. Count number of poor-quality sensor data.
5. Find all device controls with device type ‘CCTV’ and running Zigbee communication protocol.

```
Connecting to smart_building....
Connected to smart_building
All sensors with sensor type temperature and model T3000:
0. {'_id': ObjectId('6555bd07b5ea54f6f6caa899'), 'sensor_id': 194, 'room_id': 2574, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'AllSe
nsor', 'installation_date': '2021-04-04', 'last_maintenance_date': '2023-03-18', 'sensor_status': 'inactive', 'firmware_version': 'v1.2.3', 'communication
_protocol': 'Bluetooth'}
1. {'_id': ObjectId('6555bd07b5ea54f6f6caa8a5'), 'sensor_id': 536, 'room_id': 320, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'AllSen
sor', 'installation_date': '2022-12-27', 'last_maintenance_date': '2023-04-11', 'sensor_status': 'inactive', 'firmware_version': 'v3.4.5', 'communication_
protocol': 'Zigbee'}
2. {'_id': ObjectId('6555bd07b5ea54f6f6caa8ab'), 'sensor_id': 324, 'room_id': 2209, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'AllSe
nsor', 'installation_date': '2020-09-03', 'last_maintenance_date': '2023-07-02', 'sensor_status': 'active', 'firmware_version': 'v3.4.5', 'communication_p
rotocol': 'Zigbee'}
3. {'_id': ObjectId('6555bd07b5ea54f6f6caa8de'), 'sensor_id': 651, 'room_id': 6697, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Senso
rTech', 'installation_date': '2022-06-30', 'last_maintenance_date': '2023-06-30', 'sensor_status': 'active', 'firmware_version': 'v1.2.3', 'communication_
protocol': 'WiFi'}
4. {'_id': ObjectId('6555bd07b5ea54f6f6caa905'), 'sensor_id': 730, 'room_id': 7259, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'AllSe
nsor', 'installation_date': '2020-08-27', 'last_maintenance_date': '2023-07-08', 'sensor_status': 'active', 'firmware_version': 'v2.3.4', 'communication_p
rotocol': 'Bluetooth'}
5. {'_id': ObjectId('6555bd07b5ea54f6f6caa980'), 'sensor_id': 714, 'room_id': 5895, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Senso
rTech', 'installation_date': '2022-07-10', 'last_maintenance_date': '2023-07-04', 'sensor_status': 'inactive', 'firmware_version': 'v2.3.4', 'communicatio
n_protocol': 'WiFi'}
6. {'_id': ObjectId('6555bd07b5ea54f6f6caa99e'), 'sensor_id': 468, 'room_id': 603, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Sensor
Tech', 'installation_date': '2023-07-28', 'last_maintenance_date': '2023-01-06', 'sensor_status': 'active', 'firmware_version': 'v1.2.3', 'communication_p
rotocol': 'WiFi'}
7. {'_id': ObjectId('6555bd07b5ea54f6f6caa9a1'), 'sensor_id': 979, 'room_id': 4239, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Senso
rTech', 'installation_date': '2020-07-16', 'last_maintenance_date': '2023-02-13', 'sensor_status': 'inactive', 'firmware_version': 'v2.3.4', 'communicatio
n_protocol': 'WiFi'}
8. {'_id': ObjectId('6555bd07b5ea54f6f6caa9b6'), 'sensor_id': 643, 'room_id': 1051, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Senso
rCorp', 'installation_date': '2022-12-17', 'last_maintenance_date': '2023-04-07', 'sensor_status': 'active', 'firmware_version': 'v3.4.5', 'communication_
protocol': 'WiFi'}
9. {'_id': ObjectId('6555bd07b5ea54f6f6caa9c8'), 'sensor_id': 783, 'room_id': 4827, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'AllSe
nsor', 'installation_date': '2022-06-19', 'last_maintenance_date': '2023-04-27', 'sensor_status': 'inactive', 'firmware_version': 'v2.3.4', 'communication
_protocol': 'Zigbee'}
10. {'_id': ObjectId('6555bd07b5ea54f6f6caaa67'), 'sensor_id': 158, 'room_id': 4913, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Sens
orTech', 'installation_date': '2022-12-23', 'last_maintenance_date': '2023-10-11', 'sensor_status': 'active', 'firmware_version': 'v3.4.5', 'communication
_protocol': 'Bluetooth'}
11. {'_id': ObjectId('6555bd07b5ea54f6f6caaa91'), 'sensor_id': 254, 'room_id': 6881, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Sens
orTech', 'installation_date': '2021-02-23', 'last_maintenance_date': '2023-07-05', 'sensor_status': 'inactive', 'firmware_version': 'v1.2.3', 'communicati
on_protocol': 'Zigbee'}
12. {'_id': ObjectId('6555bd07b5ea54f6f6caaa97'), 'sensor_id': 818, 'room_id': 184, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Senso
rCorp', 'installation_date': '2021-01-12', 'last_maintenance_date': '2023-05-07', 'sensor_status': 'active', 'firmware_version': 'v1.2.3', 'communication_
protocol': 'WiFi'}
13. {'_id': ObjectId('6555bd07b5ea54f6f6caaa9d'), 'sensor_id': 459, 'room_id': 4039, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'AllS
ensor', 'installation_date': '2020-05-11', 'last_maintenance_date': '2023-05-01', 'sensor_status': 'active', 'firmware_version': 'v1.2.3', 'communication_
protocol': 'Bluetooth'}
14. {'_id': ObjectId('6555bd07b5ea54f6f6caaaa0'), 'sensor_id': 592, 'room_id': 6765, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'AllS
ensor', 'installation_date': '2021-11-17', 'last_maintenance_date': '2023-05-31', 'sensor_status': 'inactive', 'firmware_version': 'v1.2.3', 'communicatio
n_protocol': 'Zigbee'}
15. {'_id': ObjectId('6555bd07b5ea54f6f6caaaa3'), 'sensor_id': 991, 'room_id': 5108, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Sens
orCorp', 'installation_date': '2021-04-05', 'last_maintenance_date': '2023-06-16', 'sensor_status': 'active', 'firmware_version': 'v1.2.3', 'communication
_protocol': 'Bluetooth'}
```

```
Number of inactive sensors:  112
Sensor Data with id = 158:
0. {'_id': ObjectId('6555bd07b5ea54f6f6caa9bd'), 'data_id': 430, 'sensor_id': 158, 'timestamp': '2020-05-30T18:59:56.823315', 'data_type': 'light', 'data_
value': 533.5610546856316, 'unit_of_measure': 'lux', 'data_quality': 'good', 'data_status': 'unconfirmed'}
1. {'_id': ObjectId('6555bd07b5ea54f6f6caaa68'), 'data_id': 47, 'sensor_id': 158, 'timestamp': '2020-07-15T13:55:09.192848', 'data_type': 'temperature', '
data_value': 71.59093405387006, 'unit_of_measure': 'farhenheit', 'data_quality': 'bad', 'data_status': 'confirmed'}
Number of sensor data with bad quality:  89
All device controls with device type 'CCTV' and running the ZigBee communication protocol:
0. {'_id': ObjectId('6555bd07b5ea54f6f6caa8cb'), 'device_id': 44, 'room_id': 1127, 'device_type': 'CCTV', 'last_updated': '2020-12-25T11:09:55.867978', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v3.4.5', 'power_consumption': 4.487035326316366, 'device_status': 'functioning'}
1. {'_id': ObjectId('6555bd07b5ea54f6f6caa8ce'), 'device_id': 75, 'room_id': 5035, 'device_type': 'CCTV', 'last_updated': '2023-10-03T20:16:54.500080', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v1.2.3', 'power_consumption': 6.569926354351109, 'device_status': 'functioning'}
2. {'_id': ObjectId('6555bd07b5ea54f6f6caa904'), 'device_id': 54, 'room_id': 1066, 'device_type': 'CCTV', 'last_updated': '2020-04-20T13:45:38.502766', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v2.3.4', 'power_consumption': 7.383686183767168, 'device_status': 'not functioning'}
3. {'_id': ObjectId('6555bd07b5ea54f6f6caa907'), 'device_id': 24, 'room_id': 7240, 'device_type': 'CCTV', 'last_updated': '2020-06-25T11:56:14.954854', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v3.4.5', 'power_consumption': 3.0338500605731054, 'device_status': 'functioning'}
4. {'_id': ObjectId('6555bd07b5ea54f6f6caa91c'), 'device_id': 67, 'room_id': 6414, 'device_type': 'CCTV', 'last_updated': '2023-09-25T18:09:53.937490', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v1.2.3', 'power_consumption': 3.9544373952493754, 'device_status': 'not functioning'}
5. {'_id': ObjectId('6555bd07b5ea54f6f6caa946'), 'device_id': 69, 'room_id': 5384, 'device_type': 'CCTV', 'last_updated': '2021-05-18T06:02:17.365906', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v1.2.3', 'power_consumption': 9.874900948716048, 'device_status': 'functioning'}
6. {'_id': ObjectId('6555bd07b5ea54f6f6caa94f'), 'device_id': 40, 'room_id': 139, 'device_type': 'CCTV', 'last_updated': '2021-04-12T18:36:47.439356', 'co
ntrol_protocol': 'Zigbee', 'firmware_version': 'v2.3.4', 'power_consumption': 3.597902027821205, 'device_status': 'functioning'}
7. {'_id': ObjectId('6555bd07b5ea54f6f6caa9ca'), 'device_id': 5, 'room_id': 6737, 'device_type': 'CCTV', 'last_updated': '2021-11-06T12:45:30.946077', 'co
ntrol_protocol': 'Zigbee', 'firmware_version': 'v2.3.4', 'power_consumption': 1.5179685741388238, 'device_status': 'functioning'}
8. {'_id': ObjectId('6555bd07b5ea54f6f6caa9f4'), 'device_id': 44, 'room_id': 1487, 'device_type': 'CCTV', 'last_updated': '2022-06-07T12:32:02.985044', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v3.4.5', 'power_consumption': 1.5179685741388238, 'device_status': 'functioning'}
9. {'_id': ObjectId('6555bd07b5ea54f6f6caaa03'), 'device_id': 70, 'room_id': 341, 'device_type': 'CCTV', 'last_updated': '2021-12-24T15:52:55.553386', 'co
ntrol_protocol': 'Zigbee', 'firmware_version': 'v3.4.5', 'power_consumption': 8.470643376412461, 'device_status': 'not functioning'}
10. {'_id': ObjectId('6555bd07b5ea54f6f6caaa45'), 'device_id': 69, 'room_id': 5501, 'device_type': 'CCTV', 'last_updated': '2021-08-09T06:25:46.763364', '
control_protocol': 'Zigbee', 'firmware_version': 'v2.3.4', 'power_consumption': 2.9951162879452924, 'device_status': 'functioning'}
11. {'_id': ObjectId('6555bd07b5ea54f6f6caaa4b'), 'device_id': 50, 'room_id': 3449, 'device_type': 'CCTV', 'last_updated': '2021-03-30T00:38:20.342397', '
control_protocol': 'Zigbee', 'firmware_version': 'v1.2.3', 'power_consumption': 3.3986953342637474, 'device_status': 'not functioning'}
12. {'_id': ObjectId('6555bd07b5ea54f6f6caaac0'), 'device_id': 16, 'room_id': 3154, 'device_type': 'CCTV', 'last_updated': '2020-05-23T09:06:09.838999', '
control_protocol': 'Zigbee', 'firmware_version': 'v2.3.4', 'power_consumption': 1.8734750230558541, 'device_status': 'functioning'}
13. {'_id': ObjectId('6555bd07b5ea54f6f6caaad8'), 'device_id': 16, 'room_id': 6673, 'device_type': 'CCTV', 'last_updated': '2021-07-22T07:17:40.205775', '
control_protocol': 'Zigbee', 'firmware_version': 'v3.4.5', 'power_consumption': 2.6189672279565936, 'device_status': 'not functioning'}
14. {'_id': ObjectId('6555bd07b5ea54f6f6caaadb'), 'device_id': 72, 'room_id': 8135, 'device_type': 'CCTV', 'last_updated': '2021-09-01T22:32:33.985036', '
control_protocol': 'Zigbee', 'firmware_version': 'v1.2.3', 'power_consumption': 9.118545367412512, 'device_status': 'not functioning'}
```

Running the following queries to retrieve data from PostgreSQL:

1. Count all Operational Buildings.
2. Get all floors in Building id 1.
3. Select users with admin role with name starting with ‘A’.
4. Count number of denied access logs.
5. Select access logs for a specific room.



Disclaimer

The data stored in the database is generated using a random data generator python script “data_generator.py”. The data might vary from the above screenshots if you run the code on your local systems.