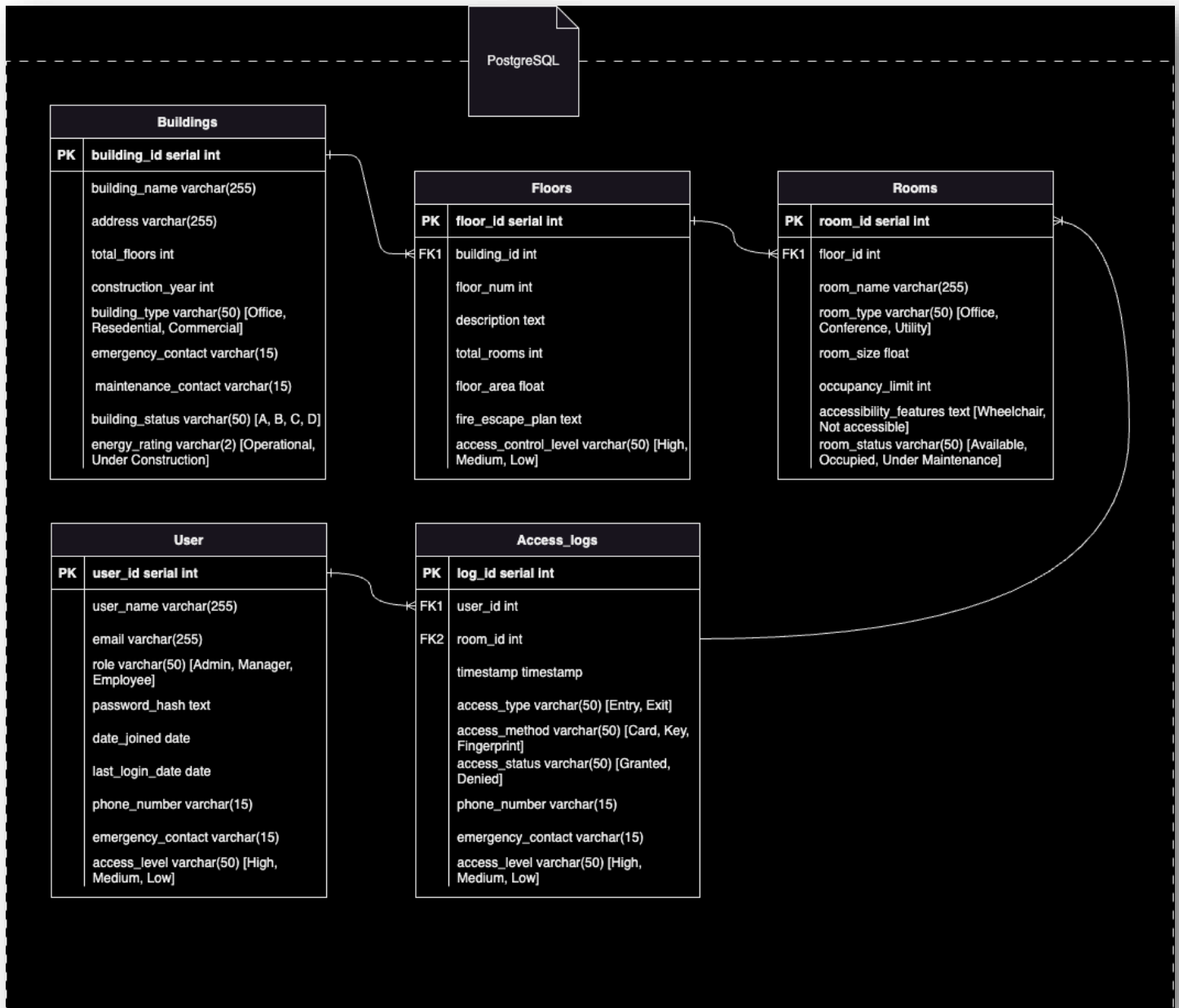


CSE 512: Distributed Database Systems

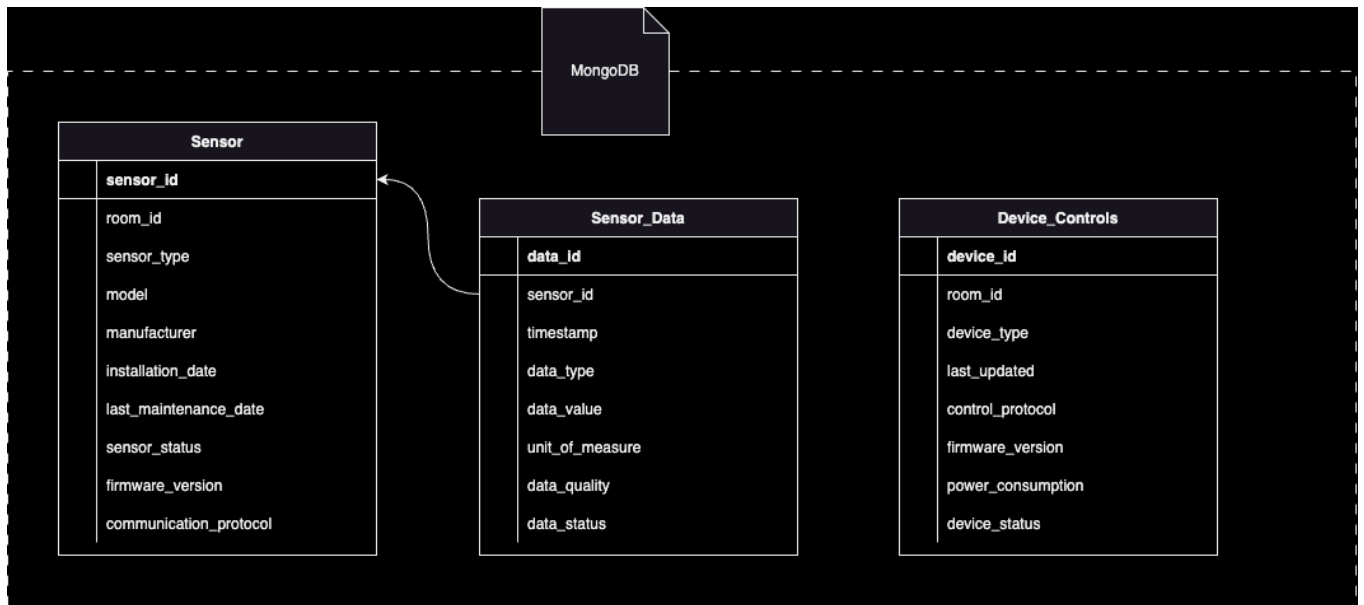
Project: Distributed Database System for a Smart Building

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': 'Sens
orTech', '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': '2022-08-27', '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': 'Sens
orTech', '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': '2022-07-28', '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': 'Sens
orTech', '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': 'Sens
orCorp', 'installation_date': '2022-10-12', '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('6555bd07b5ea54f6f6caa967'), '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('6555bd07b5ea54f6f6caa91'), '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('6555bd07b5ea54f6f6caa97'), 'sensor_id': 818, 'room_id': 184, 'sensor_type': 'temperature', 'model': 'T3000', 'manufacturer': 'Sens
orCorp', 'installation_date': '2022-01-12', 'last_maintenance_date': '2023-05-07', 'sensor_status': 'active', 'firmware_version': 'v1.2.3', 'communication_
protocol': 'WiFi'}
13. {'_id': ObjectId('6555bd07b5ea54f6f6caa9d'), '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('6555bd07b5ea54f6f6caa9a0'), '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('6555bd07b5ea54f6f6caa968'), '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': 9.8749008948716048, '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('6555bd07b5ea54f6f6caa903'), '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('6555bd07b5ea54f6f6caa945'), '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('6555bd07b5ea54f6f6caa94b'), 'device_id': 50, 'room_id': 3449, 'device_type': 'CCTV', 'last_updated': '2021-03-30T00:38:20.342397', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v1.2.3', 'power_consumption': 3.3986953342637474, 'device_status': 'not functioning'}
12. {'_id': ObjectId('6555bd07b5ea54f6f6caa9c0'), 'device_id': 16, 'room_id': 3154, 'device_type': 'CCTV', 'last_updated': '2020-05-23T09:06:09.838999', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v2.3.4', 'power_consumption': 1.8734750230558541, 'device_status': 'functioning'}
13. {'_id': ObjectId('6555bd07b5ea54f6f6caa9d8'), 'device_id': 16, 'room_id': 6673, 'device_type': 'CCTV', 'last_updated': '2021-07-22T07:17:40.205775', 'c
ontrol_protocol': 'Zigbee', 'firmware_version': 'v3.4.5', 'power_consumption': 2.6189672279565936, 'device_status': 'not functioning'}
14. {'_id': ObjectId('6555bd07b5ea54f6f6caa9db'), 'device_id': 72, 'room_id': 8135, 'device_type': 'CCTV', 'last_updated': '2021-09-01T22:32:33.985036', 'c
ontrol_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.

```

Number of Operational Buildings: 49
All floors of building 1:
0) (1, 1, 1, 'Floor 1', 19, 1009.6977017504653, 'Foot executive must fund people which hot. We difference many manager beyond somebody cup investment. Son stop compare full drive explain.\nStep try stand identify. Do news something begin.', 'High')
1) (2, 1, 2, 'Floor 2', 19, 973.1002777652045, 'Camera upon fish center sometimes themselves quite hospital. Evidence month share since research me. East consumer able international arm.', 'Low')
2) (3, 1, 3, 'Floor 3', 11, 1019.4581663563259, 'Structure six anything most sure nation. Wrong note system cut Democrat old.\nBusiness professional cost indeed. Foreign assume teach fine.', 'Low')
3) (4, 1, 4, 'Floor 4', 15, 1273.2804529029258, 'Step movie campaign approach check. Institution probably bit myself. Necessary teacher also suggest reflect.\nTechnology magazine nearly certainly or lose. Happy free last plant seem teach visit.', 'Low')
4) (5, 1, 5, 'Floor 5', 10, 1650.4824053304178, 'Reach final everything size indeed. Positive decision specific local. Finally everyone least good check.\nToward result film cell. Relationship difference clearly deep. Leg inside adult.', 'High')
5) (6, 1, 6, 'Floor 6', 12, 1575.2437001903263, 'Before yourself husband capital little know process. Catch sister their near pressure.\nBoy kind tree effect determine. Agent statement its current. Night suggest feeling where product tonight power.', 'Medium')
6) (7, 1, 7, 'Floor 7', 20, 526.4289756603138, 'Act floor force room security they month.\nWeight no statement actually defense. Sea kind sense guy hope really bed. Give toward article training hospital spend.', 'Medium')
7) (8, 1, 8, 'Floor 8', 10, 1650.2428798733474, 'Whether rate daughter series lose spring new garden. Especially Mrs right travel. None quickly perhaps hear television. Street grow focus it us case.\nWrong majority reality say ready season others.', 'Low')
All Office rooms with wheelchair accessibility that is available for booking:
0) (18, 1, 'Room 65', 'Office', 129.59911269879967, 7, 'Wheelchair accessible', 'Available')
1) (101, 7, 'Room 13', 'Office', 59.66272197097063, 2, 'Wheelchair accessible', 'Available')
2) (138, 10, 'Room 49', 'Office', 93.98154991005826, 2, 'Wheelchair accessible', 'Available')
3) (198, 14, 'Room 25', 'Office', 74.38480617312013, 8, 'Wheelchair accessible', 'Available')
4) (202, 14, 'Room 7', 'Office', 171.0911043039696, 6, 'Wheelchair accessible', 'Available')
5) (252, 17, 'Room 65', 'Office', 44.40714089566164, 4, 'Wheelchair accessible', 'Available')
6) (295, 20, 'Room 55', 'Office', 80.46936604569505, 10, 'Wheelchair accessible', 'Available')
7) (308, 21, 'Room 94', 'Office', 36.533961564621265, 2, 'Wheelchair accessible', 'Available')
8) (309, 21, 'Room 53', 'Office', 24.531234581919197, 7, 'Wheelchair accessible', 'Available')
9) (316, 22, 'Room 35', 'Office', 197.1962343824955, 8, 'Wheelchair accessible', 'Available')
10) (351, 24, 'Room 58', 'Office', 155.55852465422893, 3, 'Wheelchair accessible', 'Available')
11) (358, 24, 'Room 58', 'Office', 47.43770725112597, 2, 'Wheelchair accessible', 'Available')
12) (380, 26, 'Room 32', 'Office', 75.19459290776817, 7, 'Wheelchair accessible', 'Available')
13) (388, 27, 'Room 64', 'Office', 132.5312487781653, 7, 'Wheelchair accessible', 'Available')
14) (389, 27, 'Room 76', 'Office', 142.80717627356307, 9, 'Wheelchair accessible', 'Available')
15) (390, 27, 'Room 11', 'Office', 71.51674404437207, 8, 'Wheelchair accessible', 'Available')
16) (399, 27, 'Room 4', 'Office', 129.17123079643883, 3, 'Wheelchair accessible', 'Available')
17) (442, 30, 'Room 81', 'Office', 125.83078886440207, 3, 'Wheelchair accessible', 'Available')
18) (447, 30, 'Room 13', 'Office', 72.81448605284334, 2, 'Wheelchair accessible', 'Available')
19) (471, 32, 'Room 44', 'Office', 43.55576688714301, 8, 'Wheelchair accessible', 'Available')
20) (472, 32, 'Room 38', 'Office', 149.38260877090306, 4, 'Wheelchair accessible', 'Available')
21) (480, 32, 'Room 64', 'Office', 107.67747673591415, 4, 'Wheelchair accessible', 'Available')
22) (512, 35, 'Room 53', 'Office', 76.24444384750038, 7, 'Wheelchair accessible', 'Available')
23) (519, 35, 'Room 84', 'Office', 77.48400222744743, 3, 'Wheelchair accessible', 'Available')
24) (530, 36, 'Room 3', 'Office', 171.29670068389382, 7, 'Wheelchair accessible', 'Available')
25) (535, 36, 'Room 56', 'Office', 124.68720266828345, 9, 'Wheelchair accessible', 'Available')
26) (580, 40, 'Room 98', 'Office', 49.605895344053025, 7, 'Wheelchair accessible', 'Available')
27) (586, 40, 'Room 10', 'Office', 77.45975182385092, 8, 'Wheelchair accessible', 'Available')
28) (587, 40, 'Room 67', 'Office', 85.46097023024373, 9, 'Wheelchair accessible', 'Available')
29) (595, 41, 'Room 99', 'Office', 151.14947750164276, 8, 'Wheelchair accessible', 'Available')
30) (610, 42, 'Room 77', 'Office', 179.22698795244983, 2, 'Wheelchair accessible', 'Available')
31) (636, 44, 'Room 91', 'Office', 76.18662419812728, 1, 'Wheelchair accessible', 'Available')
32) (662, 45, 'Room 79', 'Office', 181.78407623494962, 6, 'Wheelchair accessible', 'Available')
33) (671, 46, 'Room 38', 'Office', 91.15972151483241, 6, 'Wheelchair accessible', 'Available')
34) (672, 46, 'Room 57', 'Office', 150.6299110731263, 8, 'Wheelchair accessible', 'Available')
35) (688, 47, 'Room 92', 'Office', 107.48282830459415, 5, 'Wheelchair accessible', 'Available')
36) (692, 47, 'Room 93', 'Office', 178.23832838138622, 10, 'Wheelchair accessible', 'Available')
All users with admin role:
0) (57, 'Adam Burke', 'shawsamuel@example.com', 'Admin', '1o0NVi6$#N', datetime.date(2016, 5, 20), datetime.date(2023, 4, 23), '001-885-354-1134x5936', '733-547-8996x35294', 'Low')
1) (69, 'Alexis Esparza', 'ambermoore@example.com', 'Admin', 'drfnMcqs@7', datetime.date(2020, 10, 14), datetime.date(2022, 7, 1), '001-361-888-5594', '(586)509-0669x791', 'Low')
Number of denied access logs: 44
Access logs for room (id=5):
0) (1, 2, 5, datetime.datetime(2021, 1, 20, 0, 0), 'Entry', 'Fingerprint', 'Granted')
1) (63, 79, 5, datetime.datetime(2021, 3, 20, 0, 0), 'Exit', 'Key', 'Granted')
2) (95, 50, 5, datetime.datetime(2021, 6, 16, 0, 0), 'Entry', 'Fingerprint', 'Denied')

```

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.

Compiled By:

[Distributed Nerds](#)

Ahraz Rizvi

Keshava Rajavaram

Poorvik Dharmendra

Sahara Abdi