BIRMINGHAM CITY University

# Project Title: The Design and Database of Smart Home System

**Your name:** Ponamala Gowri Navya

**Your ID:** 24169308

**Course:** CMP7245 Database Design & Development

**School:** Birmingham City University Faculty of Computing, Engineering

**Module Title:** Database Design & Development

**Module Leader**: Dr. Essa Shahra & Dr. Anuoluwapo Adewuyi

Jan, 2025

# Contents

# 1. Critical Evaluation of Database System

## 1.1 Relational Database System

Relational databases simplified data administration by getting rid of redundant and complicated systems. They facilitate the efficiency of processes like selection and joins by organising data into tables. As demonstrated by systems like MORIS, they are generally used because to features like data independence, straightforward query languages, and adaptable storage options. By eliminating duplication, normalisation provides flexibility and adaptability in government, industry, and science while ensuring data integrity.

Advantages:

* Flexibility - Complex queries can be performed using relational algebra or SQL

* Scalability - The relational model supports large-scale systems with multiple table and complex relationships.

Disadvantages:

* Performance Overhead - Maintaining relations and enforcing constraints can introduce overhead, especially for massive datasets.

* Complex schema changes - Modifying schemas in a relational database can be challenging as applications and relationships grow.

## 1.2 NoSQL Database System

NoSQL databases, often known as "Not Only SQL," are non-relational systems made to process massive amounts of unstructured or semi-structured data in a scalable, distributed, and parallel manner. Emerging alongside internet behemoths like Google and Facebook, they prioritise availability and partition tolerance over rigors ACID compliance, making them excellent at real-time applications, massive data, and exploratory analytics. Non-SQL query languages are used by NoSQL systems, which frequently forgo high consistency in favour of increased scalability and flexibility.

Advantages:

* Scalability - NoSQL databases are designed to handle large-scale, distributed and horizontally scalable architectures, making them ideal for big data environments.

* Better to unstructured data - They are well-suited for handling unstructured or semi-structured data, like JSON or XML, which relational databases struggle with.

Disadvantages:

* Limited Consistency - NoSQL databases often sacrifice strong consistency for availability, leading to eventual consistency issues.

* Not Ideal for Complex Queries - NoSQL databases are generally less efficient with complex queries that require joins or aggregations compared to relational databases.

## 1.3 Graph Database System

Graphs are a useful tool for modelling intricate interactions in fields like biology, social networks, and economic applications like fraud detection and trend prediction. Graph databases are more appropriate than relational databases because they are dynamic, heterogeneous, and schema-less. In this paper, an advanced graph database model for complex analysis, GRAD, is presented. Schema-free analysis is supported by GRAD's versatile "load first, model later" data management capabilities. Advanced data structures, graph coherence integrity criteria, algebraic querying operators, and an adaptable Neo4j-based implementation are some of its salient characteristics.

Advantages

* Adaptability: GRAD can be implemented on various database engines, offering flexibility in system design and deployment.

* Ease of Use: Its schema-on-read approach aligns with real-world use cases where data is loaded first and analysed later, offering convenience for analysts.

Disadvantages:

* Lack of Rigid Schema: While schema-less models are flexible, they can lead to inconsistencies or unstructured data if not properly managed.

* Complexity for New Users: Analysts unfamiliar with graph-based data models may find the system challenging to learn and use effectively.

## 1.4 Distributed database System

A Distributed database is not restricted to a single system. It consists of numerous databases that have been distributed in various areas and connected to one another. Data in a distributed database can be kept in several places and controlled by a computer network separately from other places. As a result, end users will find it easier to access the data. This kind of database might be necessary when users from all over the world need to access it. While there are many distributed databases available, some examples are Foundation DB, MySQL, and Apache Ignite.

Advantages

* Improved Reliability and Availability - Data redundancy ensures that even if one node fails, other nodes can provide the data.

* Data Sharing - Local databases maintain control over their data while still being part of the distributed system.

Disadvantages

* Complexity - The design, implementation, and maintenance of distributed systems are significantly more complex than centralized systems.

* Security Risks - Data transmitted between nodes must be secured to prevent interception or unauthorized access.

## 1.5 Object-Oriented Database System

Databases ensure availability, recovery, and multi-user access by managing shared, dependable, and durable data. They currently handle a variety of needs in CAD, AI, and office automation, having previously concentrated on business applications. Integration with logic, functional, and object-oriented programming is being used to address issues like as "impedance mismatch," which occurs when relational systems have trouble integrating with programming languages.

Advantages

* Enhanced Modelling Capabilities - Offers a more intuitive way to model real-world entities and their interactions compared to relational databases.

* Data Reusability and Modularity - Inheritance and encapsulation enable the reuse of existing object structures and behaviours.

Disadvantages

* Complexity - Learning and implementing OODBMS is challenging due to the need for expertise in object-oriented concepts and database management.

* Lack of Standardization - Unlike SQL in relational databases, there is no universally accepted query language or standard for OODBMS.

# 2. Database Design

## 2.1 Entity Relationship Diagram (ERD) (Logical Model)

The structure of the Smart Home System is represented graphically by an Entity-Relationship (ER) diagram, showing entities like users, devices, homes, and their relationships. It highlights how data flows between tables, illustrating connections such as user roles, device schedules, and notifications.

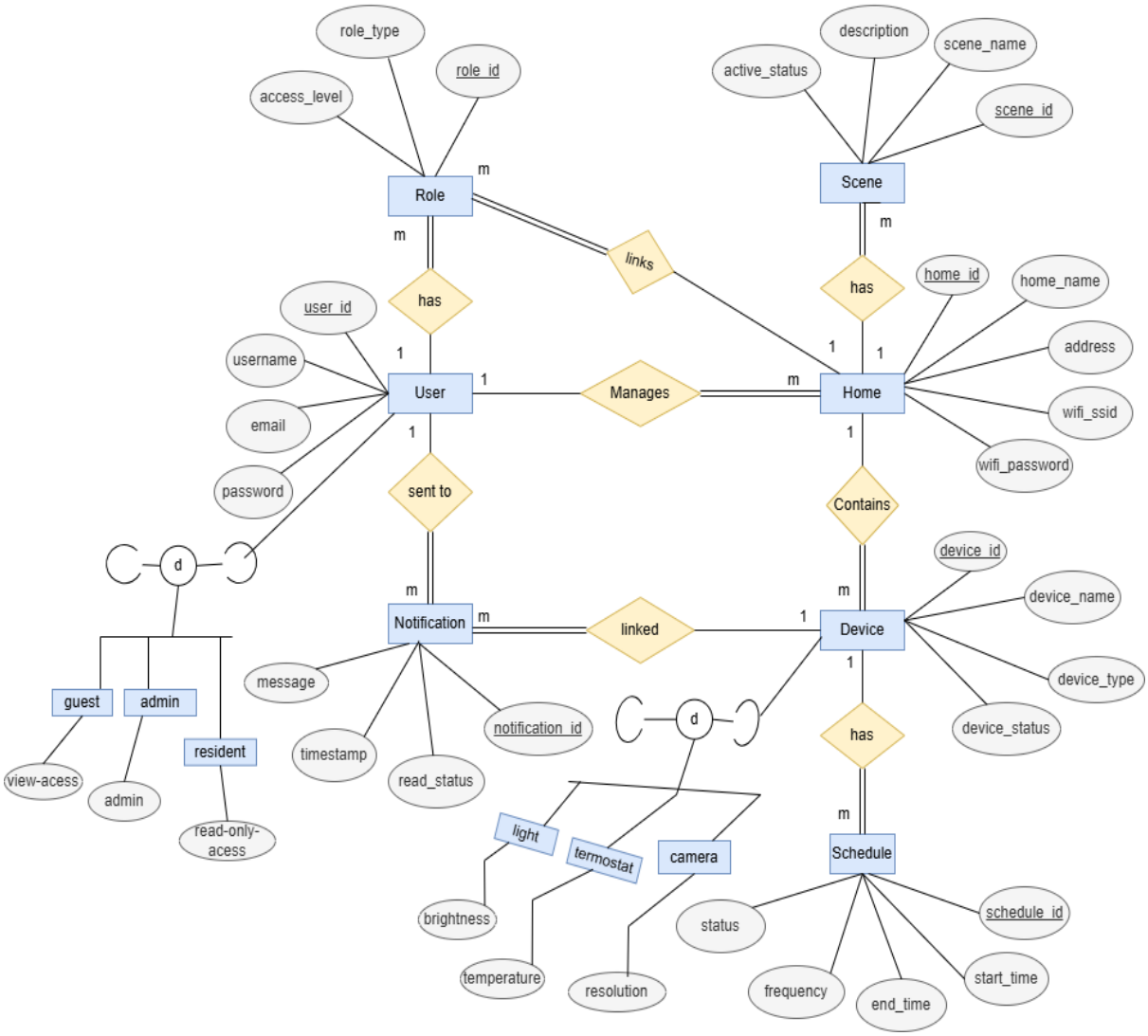



*Figure 1: Entity Relation (ER) Diagram for Smart Home System*

## 2.2 Relational Schema Mapping

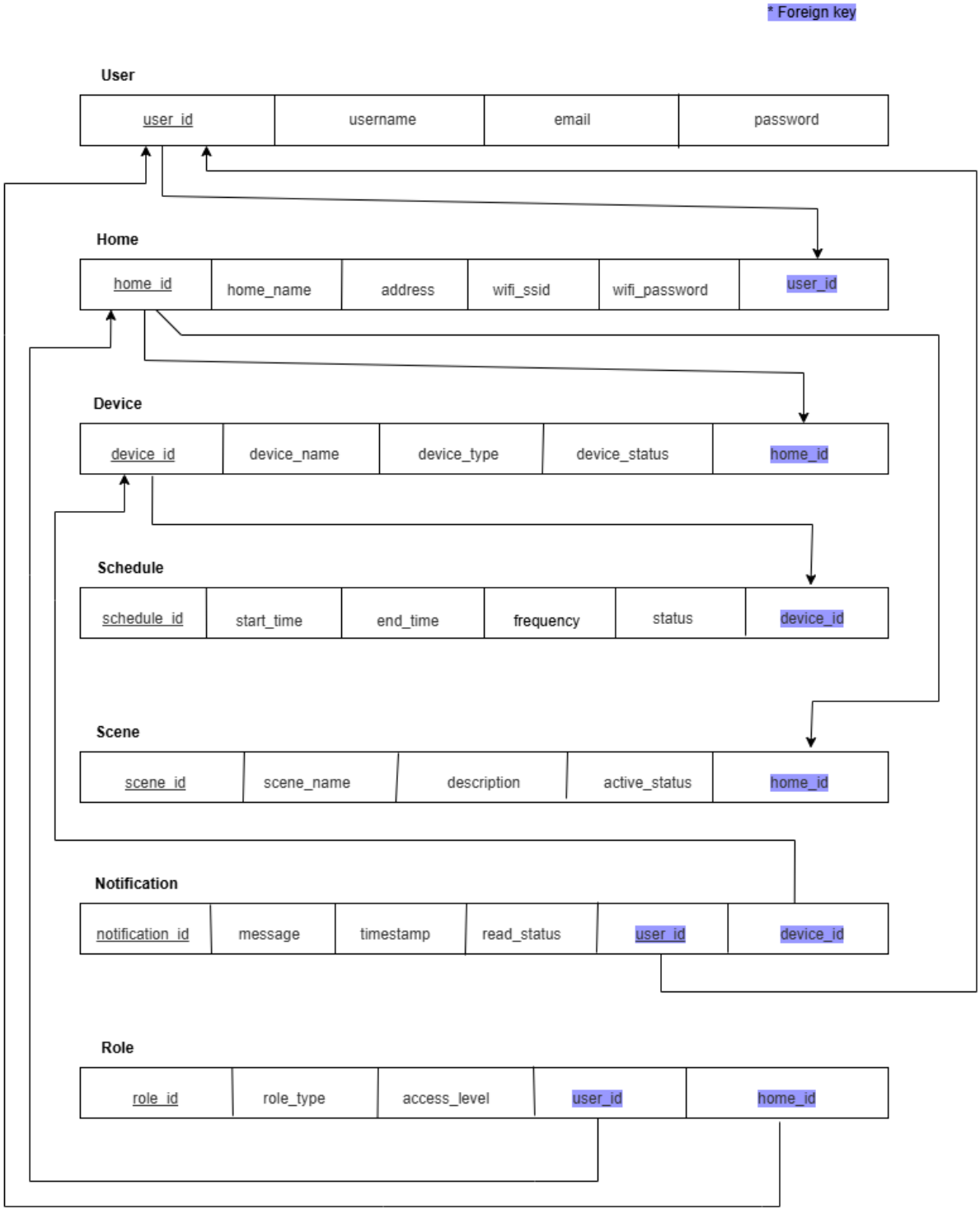



*Figure 2: Schema Diagram for Smart Home System*

# 3. Database Development

## 3.1 Physical Design of the System

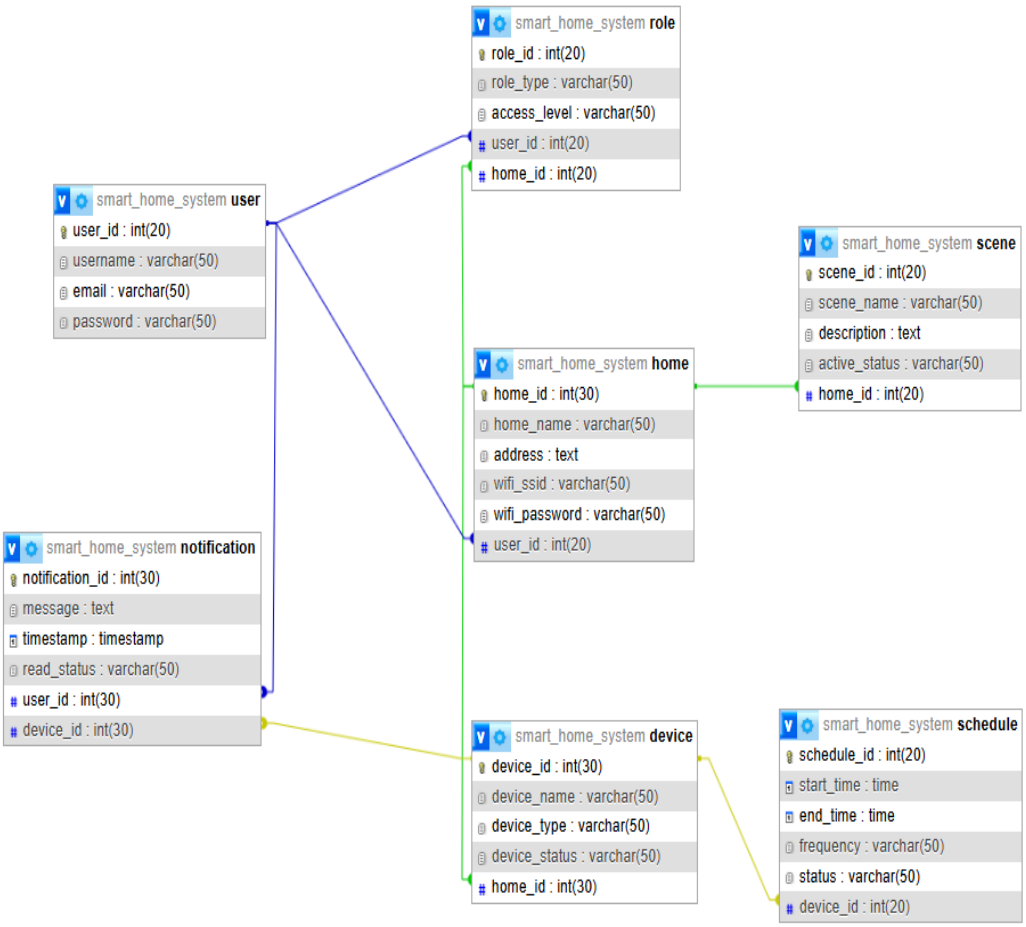



*Figure 3: Physical Diagram of Smart Home System*

The **Smart Home System** integrates devices, users, and schedules for efficient home automation. The **home** table links users and devices to properties, while the **device** table lists smart devices. **Scenes** enable predefined automation, **schedules** manage timed operations, and **roles** define user permissions. **Notifications** keep users informed about system events, ensuring seamless control and automation.

## 3.2 Tables

Creating a Database of Smart home system to store user, home, device, schedule, scene, notification, role tables.

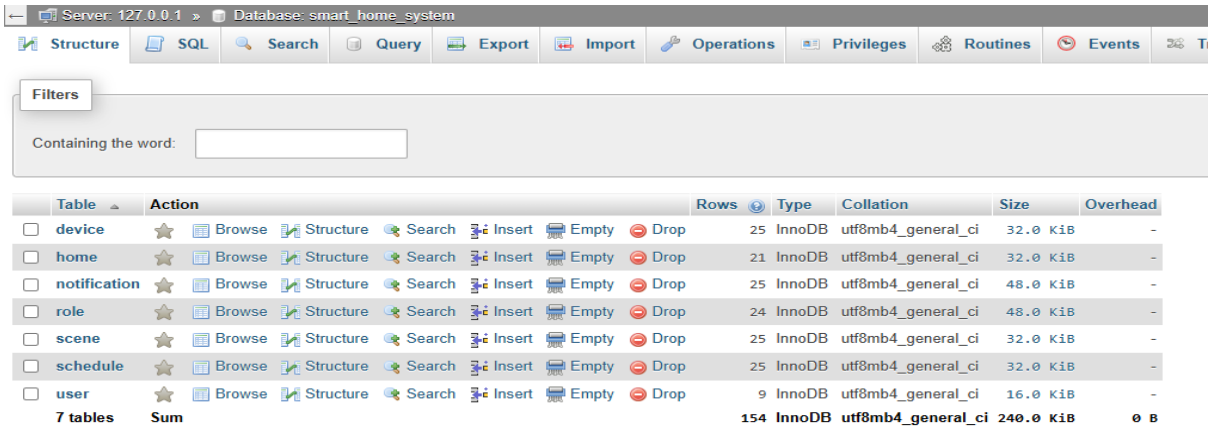



*Figure 4: Database table of Smart Home System*

### *3.2.1 User Table*

The user table stores details about users of the smart home system, including their name, email, and password. Each user is uniquely identified by its user_id.

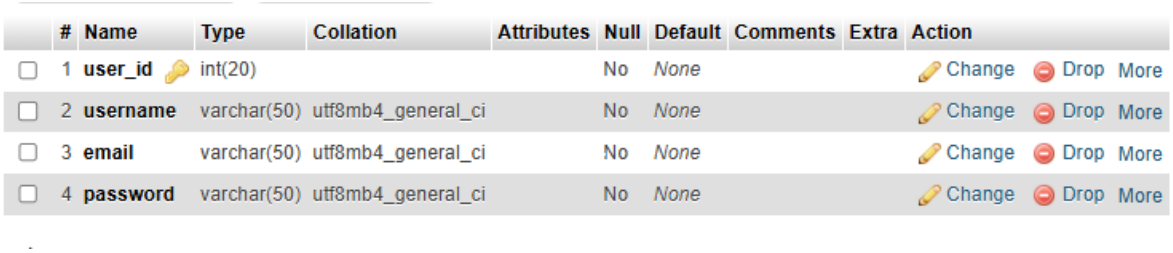



*Figure 5: User table Structure of Smart Home System*

| | | user_id | username | email | password |
|---|---|---|---|---|---|
| ☐ | Edit Copy Delete | 1 | Albert | albert1@mail.com | albert100 |
| ☐ | Edit Copy Delete | 2 | paul | paul2@mail.com | paul200 |
| ☐ | Edit Copy Delete | 3 | jay | jay3@mail.com | jay300 |
| ☐ | Edit Copy Delete | 4 | max | max4@mail.com | max400 |
| ☐ | Edit Copy Delete | 5 | ria | ria5@mail.com | ria500 |
| ☐ | Edit Copy Delete | 6 | ali | ali6@mail.com | ali600 |
| ☐ | Edit Copy Delete | 7 | ira | ira7@mail.com | ira700 |
| ☐ | Edit Copy Delete | 8 | joy | joy8@mail.com | joy800 |
| ☐ | Edit Copy Delete | 9 | ana | ana9@mail.com | ana900 |

*Figure 6: User table of Smart Home System*

### 3.2.2 Home Table

This table consists home's name, address, WiFi credentials, and a reference to the user_id of the homeowner. Each home is uniquely identified by its home_id.

| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | home_id | int(30) | | | No | *None* | | | Change Drop More |
| ☐ | 2 | home_name | varchar(50) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 3 | address | text | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 4 | wifi_ssid | varchar(50) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 5 | wifi_password | varchar(50) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| ☐ | 6 | user_id | int(20) | | | No | *None* | | | Change Drop More |

*Figure 7: Home table Structure of Smart Home System*

Extra options

| | home_id | home_name | address | wifi_ssid | wifi_password | user_id |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | albert's house | 1 kia street xyz 001 | albert's | albert100 | 1 |
| Edit Copy Delete | 2 | paul's home | 2 nest street xyz 002 | paul's | paul200 | 2 |
| Edit Copy Delete | 3 | jay's home | 3 beach road xyz 003 | jay's | jay300 | 3 |
| Edit Copy Delete | 4 | max's house | 4 summerfield street xyz 004 | max's | max400 | 4 |
| Edit Copy Delete | 5 | ria's home | 5 whole street xyz 005 | ria's | ria500 | 5 |
| Edit Copy Delete | 6 | ali's palace | 6 end street xyz 006 | ali's | ali600 | 6 |
| Edit Copy Delete | 7 | ira's cottage | 7 farming street xyz 007 | ira's | ira700 | 7 |
| Edit Copy Delete | 8 | joy's house | 8 fruits street xyz 008 | joy's | joy800 | 8 |
| Edit Copy Delete | 9 | ana's home | 9 bank street xyz 009 | ana's | ana900 | 9 |
| Edit Copy Delete | 10 | albert's building | 10 school street xyz 0010 | albert's1 | albert1000 | 1 |
| Edit Copy Delete | 11 | paul's guest_house | 11 temple road xyz 0011 | paul's1 | paul2000 | 2 |
| Edit Copy Delete | 12 | jay's home | 12 willien street xyz 0012 | jay's1 | jay3000 | 3 |
| Edit Copy Delete | 13 | max's house | 13 center road xyz 0013 | max's1 | max4000 | 4 |
| Edit Copy Delete | 14 | ria's sweet_home | 14 city street xyz 0014 | ria's1 | ria5000 | 5 |
| Edit Copy Delete | 15 | ali's house | 15 park street xyz 0015 | ali's1 | ali6000 | 6 |
| Edit Copy Delete | 16 | ira's home | 16 kits street xyz 0016 | ira's1 | ira7000 | 7 |
| Edit Copy Delete | 18 | ana's palace | 18 moore street xyz 0018 | ana's1 | ana9000 | 9 |
| Edit Copy Delete | 19 | albert's guest_house | 19 market street xyz 0019 | albert's2 | albert10000 | 1 |
| Edit Copy Delete | 20 | max's palace | 20 necklace road xyz 0020 | max's2 | max40000 | 4 |
| Edit Copy Delete | 21 | jay's apatment | 21 old street xyz 0021 | jay's2 | jay30000 | 3 |
| Edit Copy Delete | 22 | paul's gardens | 22 new street xyz 0022 | paul's2 | paul20000 | 2 |

*Figure 8: Home table of Smart Home System*

### 3.2.3 Device Table

The device table records device name, status, and the home_id where the device is installed. Each device has a unique device_id and may belong to only one home.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | device_id | int(30) | | | No | None | | | Change Drop More |
| 2 | device_name | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | device_type | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | device_status | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | home_id | int(30) | | | No | None | | | Change Drop More |

*Figure 9: Device table Structure of Smart Home System*

| | | device_id | device_name | device_type | device_status | home_id |
|---|---|---|---|---|---|---|
| ☐ | Edit Copy Delete | 1 | living room lights | light | on | 1 |
| ☐ | Edit Copy Delete | 2 | bathroom fan | fan | on | 2 |
| ☐ | Edit Copy Delete | 3 | garage lights | light | off | 3 |
| ☐ | Edit Copy Delete | 4 | bed room heater | heater | off | 4 |
| ☐ | Edit Copy Delete | 5 | kitchen fridge | fridge | on | 5 |
| ☐ | Edit Copy Delete | 6 | living room speaker | speaker | on | 6 |
| ☐ | Edit Copy Delete | 7 | backyard lights | light | off | 7 |
| ☐ | Edit Copy Delete | 8 | dining room thermostat | thermostat | on | 8 |
| ☐ | Edit Copy Delete | 9 | main hall thermostat | thermostat | on | 9 |
| ☐ | Edit Copy Delete | 10 | pool lights | light | off | 10 |
| ☐ | Edit Copy Delete | 11 | masterbedroom ac | ac | off | 11 |
| ☐ | Edit Copy Delete | 12 | living room curtains | curtain | off | 12 |
| ☐ | Edit Copy Delete | 13 | study desk lamp | light | on | 13 |
| ☐ | Edit Copy Delete | 14 | store room lights | light | off | 14 |
| ☐ | Edit Copy Delete | 15 | bedroom lights | light | off | 15 |
| ☐ | Edit Copy Delete | 16 | front door camera | camera | on | 16 |
| ☐ | Edit Copy Delete | 17 | garden water pump | pump | off | 1 |
| ☐ | Edit Copy Delete | 18 | outdoor lights | light | on | 18 |
| ☐ | Edit Copy Delete | 19 | kids room light | light | off | 19 |
| ☐ | Edit Copy Delete | 20 | kids room ac | ac | on | 20 |
| ☐ | Edit Copy Delete | 21 | guest room light | light | off | 21 |
| ☐ | Edit Copy Delete | 22 | driveway sensor | sensor | off | 22 |
| ☐ | Edit Copy Delete | 23 | kitchen light | light | on | 9 |
| ☐ | Edit Copy Delete | 24 | wine cellar cooler | cooler | on | 1 |
| ☐ | Edit Copy Delete | 25 | back side camera | camera | on | 5 |

*Figure 10: Device table of Smart Home System*

### 3.2.4 Schedule Table

Manages device operation schedules, including start time, end time, frequency, status and associated device_id.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | schedule_id | int(20) | | | No | None | | | Change Drop More |
| 2 | start_time | time | | | No | None | | | Change Drop More |
| 3 | end_time | time | | | No | None | | | Change Drop More |
| 4 | frequency | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | status | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 6 | device_id | int(20) | | | No | None | | | Change Drop More |

*Figure 11: Schedule table Structure of Smart Home System*

| | schedule_id | start_time | end_time | frequency | status | device_id |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 17:00:00 | 21:00:00 | daily | active | 1 |
| Edit Copy Delete | 2 | 20:00:00 | 21:00:00 | daily | inactive | 2 |
| Edit Copy Delete | 3 | 17:00:00 | 19:00:00 | weekly | inactive | 3 |
| Edit Copy Delete | 4 | 21:00:00 | 23:00:00 | custom | active | 4 |
| Edit Copy Delete | 5 | 06:00:00 | 22:00:00 | daily | active | 5 |
| Edit Copy Delete | 6 | 16:00:00 | 17:30:00 | custom | inactive | 6 |
| Edit Copy Delete | 7 | 17:00:00 | 18:00:00 | custom | inactive | 7 |
| Edit Copy Delete | 8 | 16:00:00 | 20:30:00 | daily | inactive | 8 |
| Edit Copy Delete | 9 | 15:30:00 | 19:30:00 | daily | active | 9 |
| Edit Copy Delete | 10 | 17:00:00 | 17:30:00 | weekly | inactive | 10 |
| Edit Copy Delete | 11 | 21:00:00 | 03:00:00 | custon | active | 11 |
| Edit Copy Delete | 12 | 06:00:00 | 05:00:00 | daily | active | 12 |
| Edit Copy Delete | 13 | 16:00:00 | 19:00:00 | daily | inactive | 13 |
| Edit Copy Delete | 14 | 14:00:00 | 16:00:00 | weekly | active | 14 |
| Edit Copy Delete | 15 | 21:00:00 | 06:00:00 | daily | inactive | 15 |
| Edit Copy Delete | 16 | 19:00:00 | 06:00:00 | daily | active | 16 |
| Edit Copy Delete | 17 | 17:00:00 | 18:00:00 | weekly | active | 17 |
| Edit Copy Delete | 18 | 16:00:00 | 19:00:00 | custom | inactive | 18 |
| Edit Copy Delete | 19 | 17:00:00 | 20:30:00 | daily | active | 19 |
| Edit Copy Delete | 20 | 00:00:00 | 21:00:00 | custom | active | 20 |
| Edit Copy Delete | 21 | 17:00:00 | 21:00:00 | daily | inactive | 21 |
| Edit Copy Delete | 22 | 20:00:00 | 04:00:00 | weekly | inactive | 22 |
| Edit Copy Delete | 23 | 18:00:00 | 17:00:00 | daily | active | 23 |
| Edit Copy Delete | 24 | 20:00:00 | 21:00:00 | custom | inactive | 24 |
| Edit Copy Delete | 25 | 17:00:00 | 16:00:00 | daily | inactive | 25 |

*Figure 12: Schedule table of Smart Home System*

### 3.2.5 Scene Table

Defines automation setups for homes, including scene name, description, status, and linked home_id.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | scene_id | int(20) | | | No | None | | | Change Drop More |
| 2 | scene_name | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | description | text | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | active_status | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | home_id | int(20) | | | No | None | | | Change Drop More |

*Figure 13: Scene table Structure of Smart Home System*

| | scene_id | scene_name | description | active_status | home_id |
|---|---|---|---|---|---|
| Edit Copy Delete | 1 | Evening Comfort | Living room lights and fans are scheduled to creat... | active | 1 |
| Edit Copy Delete | 2 | Morning Freshness | Bathroom fan ensures fresh air daily from 06:00 to... | active | 2 |
| Edit Copy Delete | 3 | Garage Security | Garage lights are set to turn on weekly during eve... | inactive | 3 |
| Edit Copy Delete | 4 | Bedroom Warmth | Bedroom heater is programmed to ensure comfort wit... | active | 4 |
| Edit Copy Delete | 5 | Kitchen Essentials | Fridge operates continuously, ensuring proper cool... | active | 5 |
| Edit Copy Delete | 6 | Living Room Entertainment | Living room speakers and curtains synchronize with... | inactive | 6 |
| Edit Copy Delete | 7 | Backyard Glow | Backyard lights are scheduled weekly to enhance th... | inactive | 7 |
| Edit Copy Delete | 8 | Dining Room Comfort | Dining room thermostat creates a cozy atmosphere d... | active | 8 |
| Edit Copy Delete | 9 | Hall Climate Control | Main hall thermostat regulates temperature for com... | active | 9 |
| Edit Copy Delete | 10 | Poolside Lighting | Pool lights turn on weekly for late evening swims. | inactive | 10 |
| Edit Copy Delete | 11 | Master Bedroom Cool | AC in the master bedroom is programmed to ensure a... | active | 11 |
| Edit Copy Delete | 12 | Smart Curtains | Living room curtains operate daily from 06:00 to 0... | active | 12 |
| Edit Copy Delete | 13 | Study Mode | Study desk lamp operates daily from 16:00 to 19:00... | inactive | 13 |
| Edit Copy Delete | 14 | Storage Room Efficiency | Store room lights are set for weekly checks from 1... | active | 14 |
| Edit Copy Delete | 15 | Nighttime Tranquility | Bedroom lights ensure nighttime comfort with daily... | inactive | 15 |
| Edit Copy Delete | 16 | Home Surveillance | Front door camera operates daily from 19:00 to 06:... | active | 16 |
| Edit Copy Delete | 17 | Garden Maintenance | Garden water pump operates weekly from 17:00 to 18... | active | 17 |
| Edit Copy Delete | 18 | Outdoor Festivities | Outdoor lights are set for custom celebrations, op... | inactive | 18 |
| Edit Copy Delete | 19 | Kids Room Comfort | Kids room AC and lights synchronize to ensure comf... | active | 19 |
| Edit Copy Delete | 20 | Guest Room Preparation | Guest room light is scheduled weekly for evening p... | inactive | 20 |
| Edit Copy Delete | 21 | Kitchen Productivity | Kitchen light is set for daily use from 18:00 to 1... | active | 9 |
| Edit Copy Delete | 22 | Wine Cellar Cooling | Wine cellar cooler ensures proper storage temperat... | active | 1 |
| Edit Copy Delete | 23 | Backyard Security | Backside camera ensures security for the backyard,... | active | 5 |
| Edit Copy Delete | 24 | Driveway Monitoring | Driveway sensor operates weekly for evening and ni... | inactive | 22 |
| Edit Copy Delete | 25 | Late-Night Work Mode | Study desk lamp supports late-night work sessions,... | inactive | 13 |

*Figure 14: Scene table of Smart Home System*

### 3.2.6 *Notification Table*

Tracks system notifications with message content, timestamp, read status, and references to user_id and device_id.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | notification_id | int(30) | | | No | *None* | | | Change Drop More |
| 2 | message | text | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| 3 | timestamp | timestamp | | | No | current_timestamp() | | ON UPDATE CURRENT_TIMESTAMP() | Change Drop More |
| 4 | read_status | varchar(50) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |
| 5 | user_id | int(30) | | | No | *None* | | | Change Drop More |
| 6 | device_id | int(30) | | | No | *None* | | | Change Drop More |

*Figure 15: Notification table Structure of Smart Home System*

| | notification_id | message | timestamp | read_status | user_id | device_id |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | Device update available for Smartphone. | 2024-12-23 10:00:00 | unread | 1 | 1 |
| Edit Copy Delete | 2 | Garage door sensor detected unusual activity. | 2024-12-23 11:00:00 | read | 1 | 2 |
| Edit Copy Delete | 3 | Thermostat detected unusual temperature. | 2024-12-23 12:30:00 | unread | 2 | 3 |
| Edit Copy Delete | 4 | New app installed on Smart TV. | 2024-12-23 14:00:00 | read | 2 | 3 |
| Edit Copy Delete | 5 | Laptop battery health report available. | 2024-12-23 15:45:00 | unread | 3 | 4 |
| Edit Copy Delete | 6 | Smart Fridge requires maintenance. | 2024-12-23 16:00:00 | unread | 1 | 1 |
| Edit Copy Delete | 7 | New security update for Smartphone. | 2024-12-23 17:00:00 | unread | 1 | 1 |
| Edit Copy Delete | 8 | Weekly usage report for Smart TV. | 2024-12-23 18:15:00 | unread | 2 | 3 |
| Edit Copy Delete | 9 | Fridge door left open for too long. | 2024-12-23 19:30:00 | unread | 4 | 5 |
| Edit Copy Delete | 10 | Camera detected motion at front door. | 2024-12-23 20:00:00 | read | 5 | 6 |
| Edit Copy Delete | 11 | New WiFi connection detected on Laptop. | 2024-12-23 21:15:00 | read | 3 | 4 |
| Edit Copy Delete | 12 | Garage lights are scheduled to turn on. | 2024-12-23 22:00:00 | unread | 2 | 2 |
| Edit Copy Delete | 13 | Backup battery level is critically low. | 2024-12-23 23:30:00 | read | 3 | 5 |
| Edit Copy Delete | 14 | Living room speakers require a firmware update. | 2024-12-24 00:45:00 | unread | 1 | 1 |
| Edit Copy Delete | 15 | New thermostat schedule set for evening hours. | 2024-12-24 01:30:00 | unread | 1 | 4 |
| Edit Copy Delete | 16 | Bedroom lights turned off as scheduled. | 2024-12-24 02:00:00 | read | 3 | 1 |
| Edit Copy Delete | 17 | Outdoor security camera detected motion. | 2024-12-24 03:15:00 | read | 5 | 6 |
| Edit Copy Delete | 18 | Water heater is reaching its maintenance period. | 2024-12-24 04:00:00 | read | 4 | 5 |
| Edit Copy Delete | 19 | Smoke detector tested successfully. | 2024-12-24 05:30:00 | unread | 3 | 4 |
| Edit Copy Delete | 20 | WiFi connection lost for Smart TV. | 2024-12-24 06:00:00 | read | 2 | 3 |
| Edit Copy Delete | 21 | New firmware available for Front Door Camera. | 2024-12-24 07:00:00 | unread | 5 | 6 |
| Edit Copy Delete | 22 | Battery replacement needed for backyard sensors. | 2024-12-24 08:15:00 | read | 1 | 4 |
| Edit Copy Delete | 23 | New updates available for all linked devices. | 2024-12-24 09:30:04 | unread | 4 | 5 |
| Edit Copy Delete | 24 | Garden lights turned off after evening schedule. | 2024-12-24 10:00:00 | read | 5 | 2 |
| Edit Copy Delete | 25 | Driveway sensors activated at 22:00. | 2024-12-24 10:45:00 | unread | 4 | 5 |

*Figure 16: Notification table of Smart Home System*

### 3.2.7 Role Table

Defines roles for users including homes such as admin or resident, with details about role permissions and functionality.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | role_id | int(20) | | | No | None | | | Change Drop More |
| 2 | role_type | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | access_level | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | user_id | int(20) | | | No | None | | | Change Drop More |
| 5 | home_id | int(20) | | | No | None | | | Change Drop More |

*Figure 17: Role table Structure of Smart Home System*

| | role_id | role_type | access_level | user_id | home_id |
|---|---|---|---|---|---|
| Edit Copy Delete | 1 | admin | full acess | 1 | 1 |
| Edit Copy Delete | 2 | guest | view_only | 2 | 2 |
| Edit Copy Delete | 3 | editor | edit_content | 3 | 3 |
| Edit Copy Delete | 4 | moderator | moderate_comments | 4 | 4 |
| Edit Copy Delete | 5 | guest | view_only | 5 | 2 |
| Edit Copy Delete | 6 | editor | edit_content | 6 | 5 |
| Edit Copy Delete | 7 | guest | view_only | 7 | 6 |
| Edit Copy Delete | 8 | editor | edit_content | 8 | 7 |
| Edit Copy Delete | 9 | edit_content | moderate_comments | 9 | 8 |
| Edit Copy Delete | 10 | guest | view only | 1 | 9 |
| Edit Copy Delete | 11 | editor | edit_content | 2 | 10 |
| Edit Copy Delete | 12 | viewer | limited access | 3 | 11 |
| Edit Copy Delete | 13 | guest | view_only | 4 | 12 |
| Edit Copy Delete | 14 | editor | edit_content | 5 | 13 |
| Edit Copy Delete | 15 | guest | view_only | 6 | 14 |
| Edit Copy Delete | 16 | editor | edit_content | 7 | 15 |
| Edit Copy Delete | 17 | guest | view_only | 8 | 16 |
| Edit Copy Delete | 18 | editor | edit_content | 9 | 18 |
| Edit Copy Delete | 19 | guest | view_only | 9 | 18 |
| Edit Copy Delete | 20 | editor | edit_content | 1 | 19 |
| Edit Copy Delete | 21 | guest | view_only | 3 | 4 |
| Edit Copy Delete | 22 | editor | edit_content | 6 | 9 |
| Edit Copy Delete | 23 | admin | full acess | 7 | 19 |
| Edit Copy Delete | 24 | admin | full_acess | 2 | 3 |

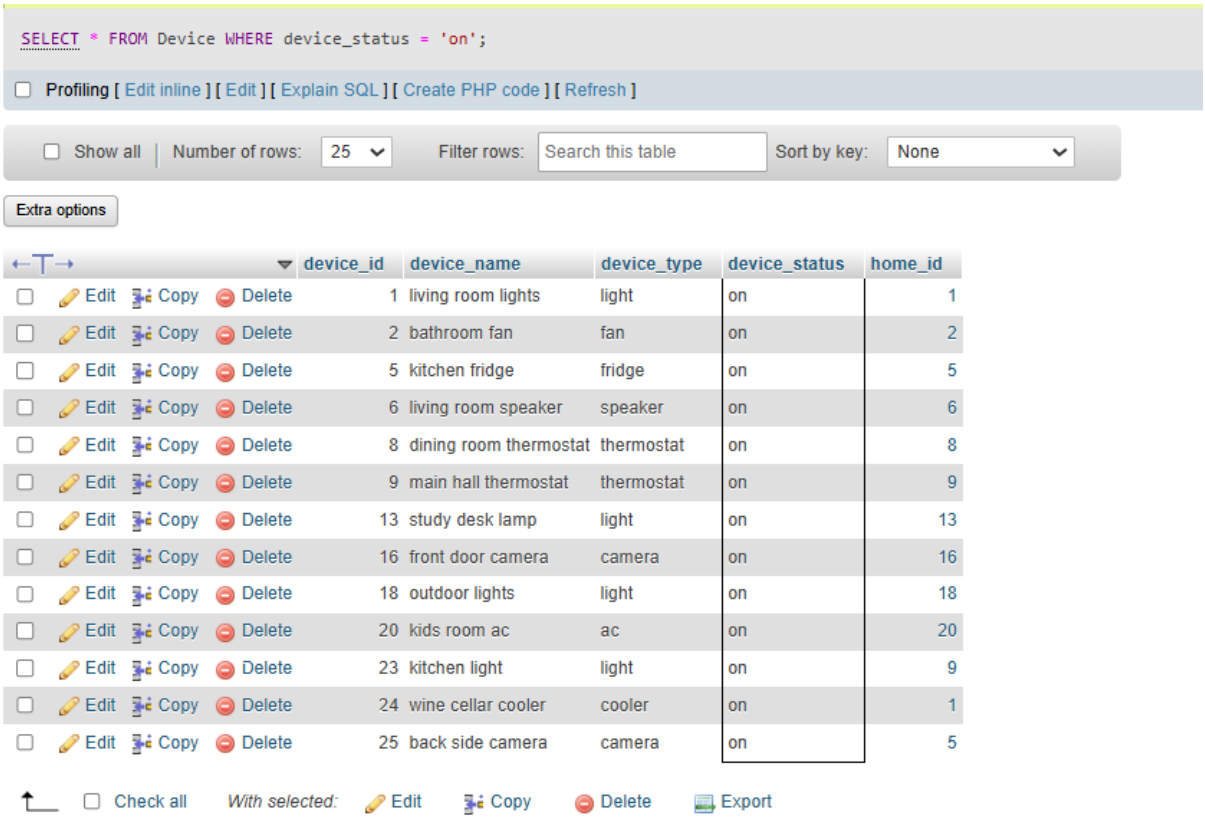
*Figure 18: Role table of Smart Home System*

## 3.3 Queries Examples

**Query 1:** Get all the devices that are currently “ON”

**Answer:**

```
SELECT * FROM Device WHERE device_status = 'on';
```



SELECT * FROM Device WHERE device_status = 'on';

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

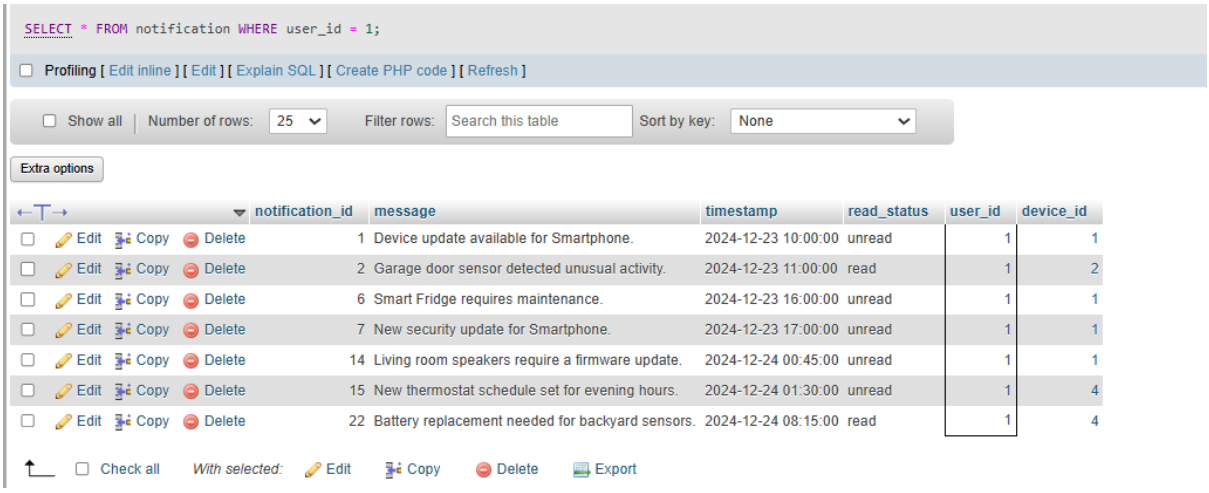
| | device_id | device_name | device_type | device_status | home_id |
|---|---|---|---|---|---|
| Edit Copy Delete | 1 | living room lights | light | on | 1 |
| Edit Copy Delete | 2 | bathroom fan | fan | on | 2 |
| Edit Copy Delete | 5 | kitchen fridge | fridge | on | 5 |
| Edit Copy Delete | 6 | living room speaker | speaker | on | 6 |
| Edit Copy Delete | 8 | dining room thermostat | thermostat | on | 8 |
| Edit Copy Delete | 9 | main hall thermostat | thermostat | on | 9 |
| Edit Copy Delete | 13 | study desk lamp | light | on | 13 |
| Edit Copy Delete | 16 | front door camera | camera | on | 16 |
| Edit Copy Delete | 18 | outdoor lights | light | on | 18 |
| Edit Copy Delete | 20 | kids room ac | ac | on | 20 |
| Edit Copy Delete | 23 | kitchen light | light | on | 9 |
| Edit Copy Delete | 24 | wine cellar cooler | cooler | on | 1 |
| Edit Copy Delete | 25 | back side camera | camera | on | 5 |

Check all With selected: Edit Copy Delete Export

**Query 2:** Gel the notification for a specific user

**Answer:**

```
SELECT * FROM notification WHERE user_id = 1;
```



```
SELECT * FROM notification WHERE user_id = 1;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

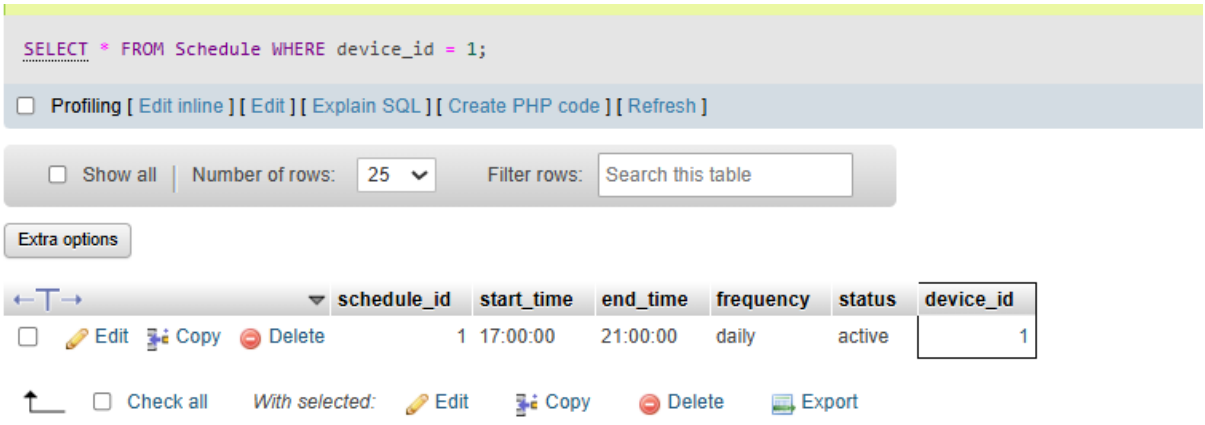
| | notification_id | message | timestamp | read_status | user_id | device_id |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | Device update available for Smartphone. | 2024-12-23 10:00:00 | unread | 1 | 1 |
| Edit Copy Delete | 2 | Garage door sensor detected unusual activity. | 2024-12-23 11:00:00 | read | 1 | 2 |
| Edit Copy Delete | 6 | Smart Fridge requires maintenance. | 2024-12-23 16:00:00 | unread | 1 | 1 |
| Edit Copy Delete | 7 | New security update for Smartphone. | 2024-12-23 17:00:00 | unread | 1 | 1 |
| Edit Copy Delete | 14 | Living room speakers require a firmware update. | 2024-12-24 00:45:00 | unread | 1 | 1 |
| Edit Copy Delete | 15 | New thermostat schedule set for evening hours. | 2024-12-24 01:30:00 | unread | 1 | 4 |
| Edit Copy Delete | 22 | Battery replacement needed for backyard sensors. | 2024-12-24 08:15:00 | read | 1 | 4 |

Check all With selected: Edit Copy Delete Export

**Query 3:** Retrieve all schedules for a specific device

**Answer:**

SELECT * FROM Schedule WHERE device_id = 1;



```
SELECT * FROM Schedule WHERE device_id = 1;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

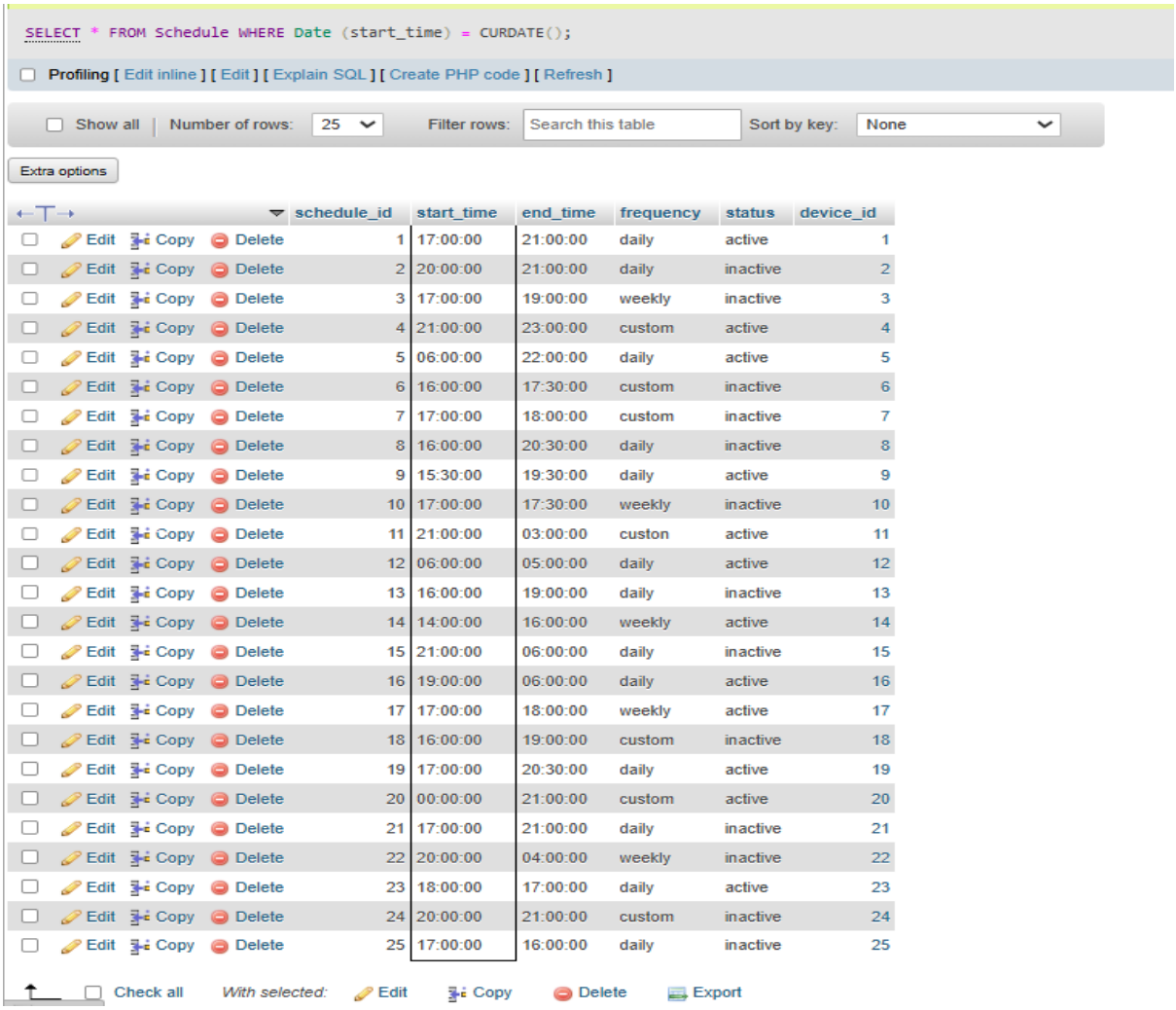
| | schedule_id | start_time | end_time | frequency | status | device_id |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 17:00:00 | 21:00:00 | daily | active | 1 |

Check all With selected: Edit Copy Delete Export

**Query 4:** Get the user who have devices scheduled to run to day

**Answer:**

SELECT * FROM Schedule WHERE Date (start_time) = CURDATE();



```
SELECT * FROM Schedule WHERE Date (start_time) = CURDATE();
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

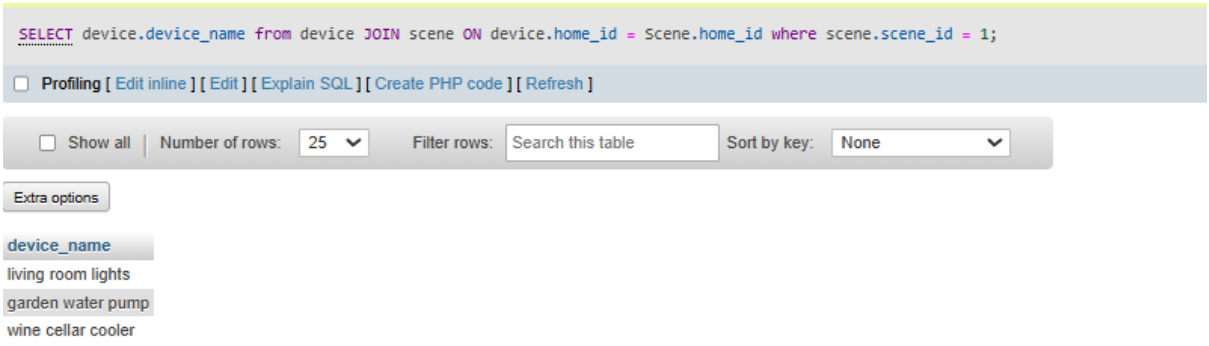
| | schedule_id | start_time | end_time | frequency | status | device_id |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 17:00:00 | 21:00:00 | daily | active | 1 |
| Edit Copy Delete | 2 | 20:00:00 | 21:00:00 | daily | inactive | 2 |
| Edit Copy Delete | 3 | 17:00:00 | 19:00:00 | weekly | inactive | 3 |
| Edit Copy Delete | 4 | 21:00:00 | 23:00:00 | custom | active | 4 |
| Edit Copy Delete | 5 | 06:00:00 | 22:00:00 | daily | active | 5 |
| Edit Copy Delete | 6 | 16:00:00 | 17:30:00 | custom | inactive | 6 |
| Edit Copy Delete | 7 | 17:00:00 | 18:00:00 | custom | inactive | 7 |
| Edit Copy Delete | 8 | 16:00:00 | 20:30:00 | daily | inactive | 8 |
| Edit Copy Delete | 9 | 15:30:00 | 19:30:00 | daily | active | 9 |
| Edit Copy Delete | 10 | 17:00:00 | 17:30:00 | weekly | inactive | 10 |
| Edit Copy Delete | 11 | 21:00:00 | 03:00:00 | custon | active | 11 |
| Edit Copy Delete | 12 | 06:00:00 | 05:00:00 | daily | active | 12 |
| Edit Copy Delete | 13 | 16:00:00 | 19:00:00 | daily | inactive | 13 |
| Edit Copy Delete | 14 | 14:00:00 | 16:00:00 | weekly | active | 14 |
| Edit Copy Delete | 15 | 21:00:00 | 06:00:00 | daily | inactive | 15 |
| Edit Copy Delete | 16 | 19:00:00 | 06:00:00 | daily | active | 16 |
| Edit Copy Delete | 17 | 17:00:00 | 18:00:00 | weekly | active | 17 |
| Edit Copy Delete | 18 | 16:00:00 | 19:00:00 | custom | inactive | 18 |
| Edit Copy Delete | 19 | 17:00:00 | 20:30:00 | daily | active | 19 |
| Edit Copy Delete | 20 | 00:00:00 | 21:00:00 | custom | active | 20 |
| Edit Copy Delete | 21 | 17:00:00 | 21:00:00 | daily | inactive | 21 |
| Edit Copy Delete | 22 | 20:00:00 | 04:00:00 | weekly | inactive | 22 |
| Edit Copy Delete | 23 | 18:00:00 | 17:00:00 | daily | active | 23 |
| Edit Copy Delete | 24 | 20:00:00 | 21:00:00 | custom | inactive | 24 |
| Edit Copy Delete | 25 | 17:00:00 | 16:00:00 | daily | inactive | 25 |

Check all With selected: Edit Copy Delete Export

**Query 5:** Get the devices assigned to a specific scene

**Answer:**

SELECT device.device_name from device JOIN scene ON device.home_id = Scene.home_id where scene.scene_id = 1;



```
SELECT device.device_name from device JOIN scene ON device.home_id = Scene.home_id where scene.scene_id = 1;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

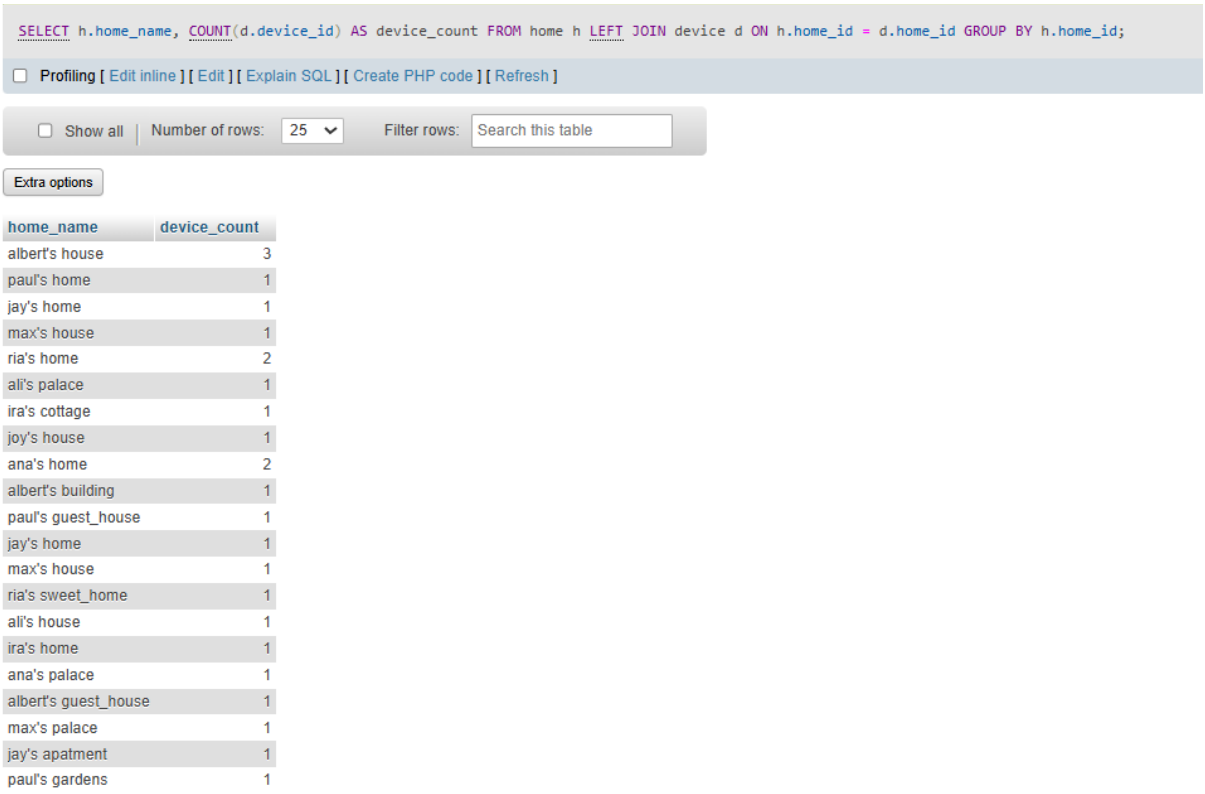
Extra options

| device_name |
|---|
| living room lights |
| garden water pump |
| wine cellar cooler |

**Query 6:** Count all devices in each home

**Answer:**

```
SELECT h.home_name, COUNT(d.device_id) AS device_count
FROM home h
LEFT JOIN device d ON h.home_id = d.home_id
GROUP BY h.home_id;
```



```
SELECT h.home_name, COUNT(d.device_id) AS device_count FROM home h LEFT JOIN device d ON h.home_id = d.home_id GROUP BY h.home_id;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

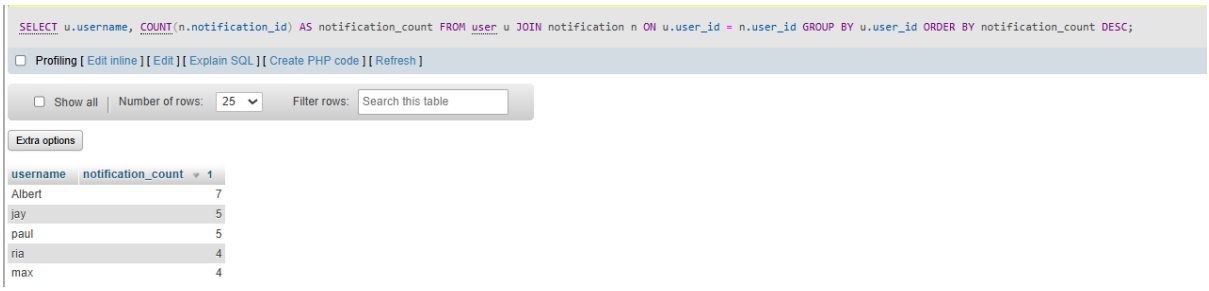
Extra options

| home_name | device_count |
|---|---|
| albert's house | 3 |
| paul's home | 1 |
| jay's home | 1 |
| max's house | 1 |
| ria's home | 2 |
| ali's palace | 1 |
| ira's cottage | 1 |
| joy's house | 1 |
| ana's home | 2 |
| albert's building | 1 |
| paul's guest_house | 1 |
| jay's home | 1 |
| max's house | 1 |
| ria's sweet_home | 1 |
| ali's house | 1 |
| ira's home | 1 |
| ana's palace | 1 |
| albert's guest_house | 1 |
| max's palace | 1 |
| jay's apatment | 1 |
| paul's gardens | 1 |

**Query 7:** Most active users by notifications

**Answer:**

```
SELECT u.username, COUNT(n.notification_id) AS notification_count
FROM user u
JOIN notification n ON u.user_id = n.user_id
GROUP BY u.user_id
ORDER BY notification_count DESC;
```



```
SELECT u.username, COUNT(n.notification_id) AS notification_count FROM user u JOIN notification n ON u.user_id = n.user_id GROUP BY u.user_id ORDER BY notification_count DESC;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

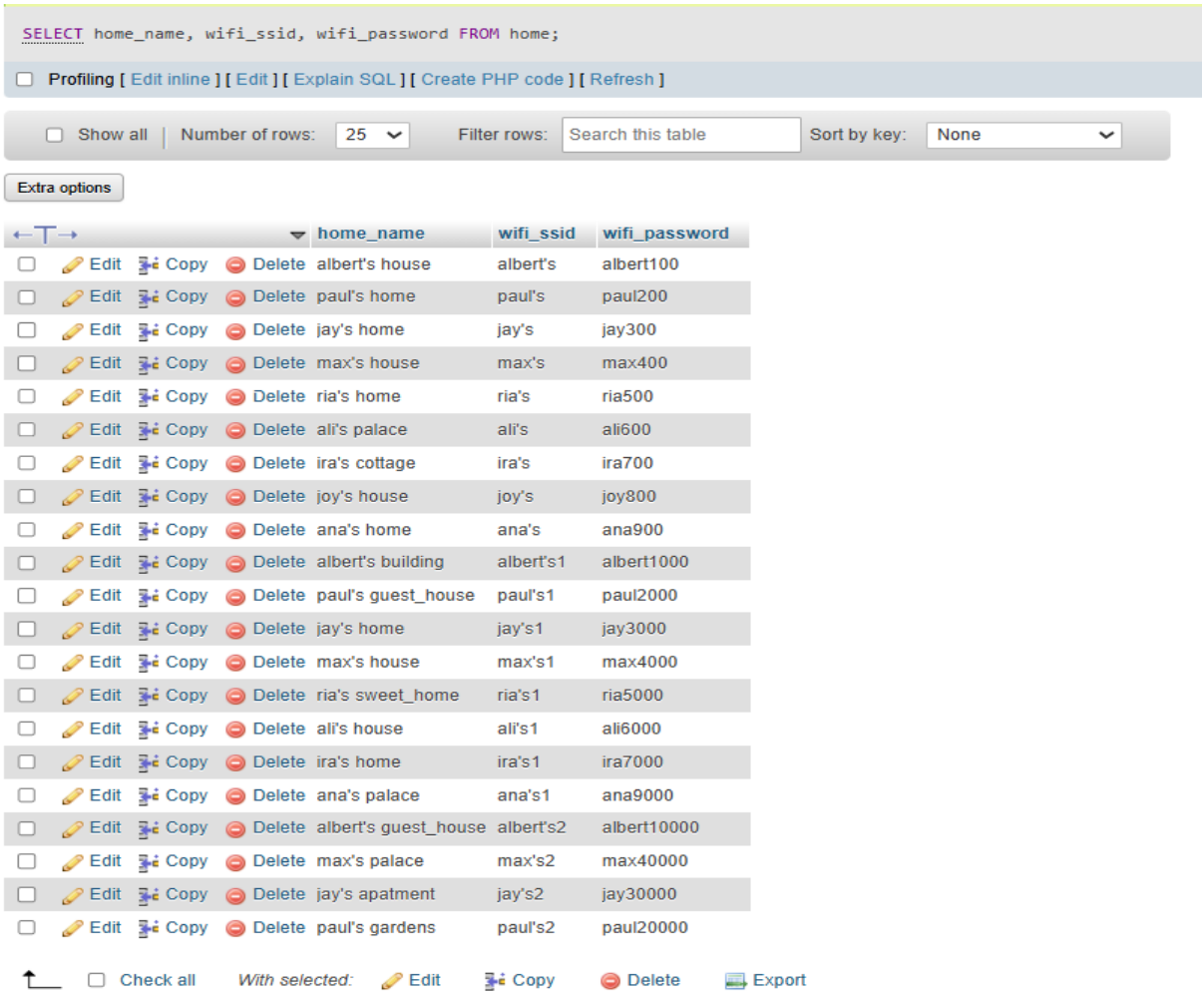
Extra options

| username | notification_count 1 |
|---|---|
| Albert | 7 |
| jay | 5 |
| paul | 5 |
| ria | 4 |
| max | 4 |

**Query 8:** Retrieve home with wifi details

**Answer:**

SELECT home_name, wifi_ssid, wifi_password FROM home;



```
SELECT home_name, wifi_ssid, wifi_password FROM home;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

| | home_name | wifi_ssid | wifi_password |
|---|---|---|---|
| Edit Copy Delete | albert's house | albert's | albert100 |
| Edit Copy Delete | paul's home | paul's | paul200 |
| Edit Copy Delete | jay's home | jay's | jay300 |
| Edit Copy Delete | max's house | max's | max400 |
| Edit Copy Delete | ria's home | ria's | ria500 |
| Edit Copy Delete | ali's palace | ali's | ali600 |
| Edit Copy Delete | ira's cottage | ira's | ira700 |
| Edit Copy Delete | joy's house | joy's | joy800 |
| Edit Copy Delete | ana's home | ana's | ana900 |
| Edit Copy Delete | albert's building | albert's1 | albert1000 |
| Edit Copy Delete | paul's guest_house | paul's1 | paul2000 |
| Edit Copy Delete | jay's home | jay's1 | jay3000 |
| Edit Copy Delete | max's house | max's1 | max4000 |
| Edit Copy Delete | ria's sweet_home | ria's1 | ria5000 |
| Edit Copy Delete | ali's house | ali's1 | ali6000 |
| Edit Copy Delete | ira's home | ira's1 | ira7000 |
| Edit Copy Delete | ana's palace | ana's1 | ana9000 |
| Edit Copy Delete | albert's guest_house | albert's2 | albert10000 |
| Edit Copy Delete | max's palace | max's2 | max40000 |
| Edit Copy Delete | jay's apatment | jay's2 | jay30000 |
| Edit Copy Delete | paul's gardens | paul's2 | paul20000 |

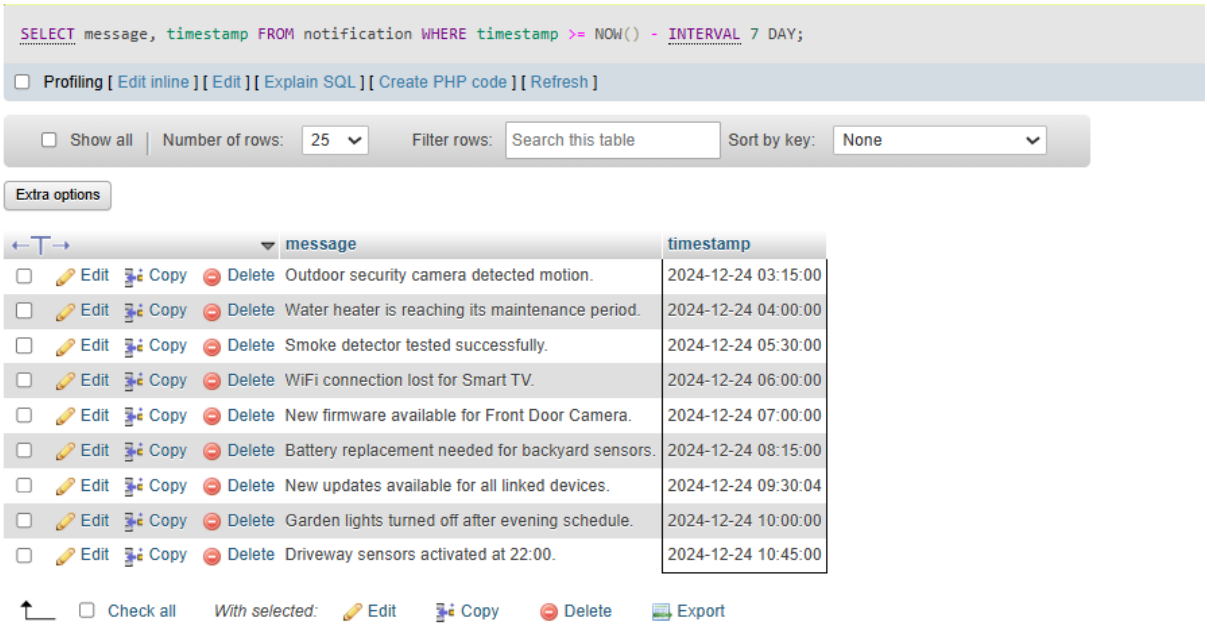
Check all With selected: Edit Copy Delete Export

**Query 9:** Recent notifications (last 7 days)

**Answer:**

```
SELECT message, timestamp FROM notification

WHERE timestamp >= NOW() - INTERVAL 7     DAY;
```



```
SELECT message, timestamp FROM notification WHERE timestamp >= NOW() - INTERVAL 7 DAY;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

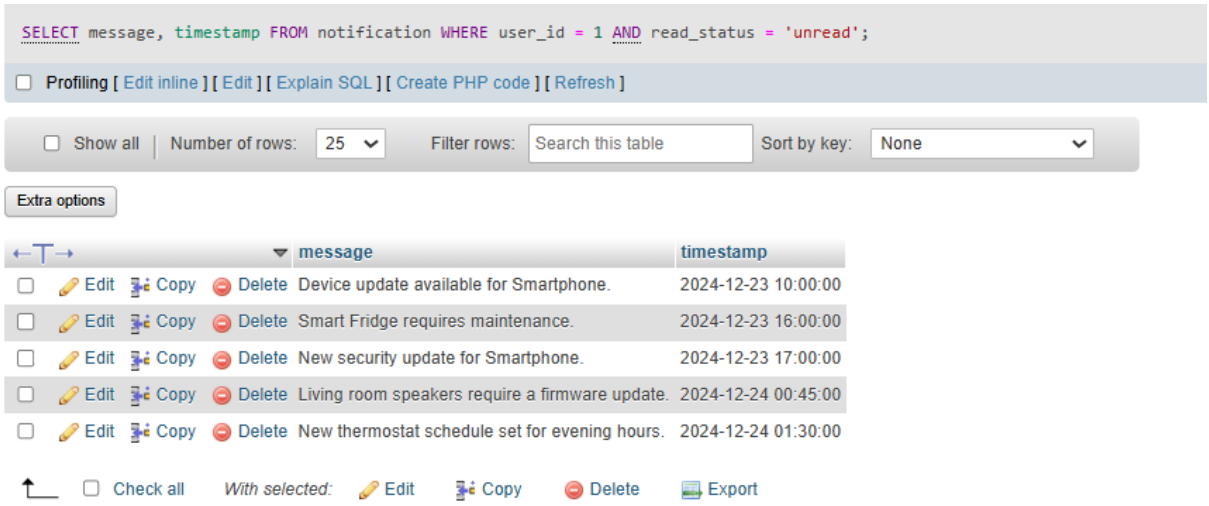
| | message | timestamp |
|---|---|---|
| Edit Copy Delete | Outdoor security camera detected motion. | 2024-12-24 03:15:00 |
| Edit Copy Delete | Water heater is reaching its maintenance period. | 2024-12-24 04:00:00 |
| Edit Copy Delete | Smoke detector tested successfully. | 2024-12-24 05:30:00 |
| Edit Copy Delete | WiFi connection lost for Smart TV. | 2024-12-24 06:00:00 |
| Edit Copy Delete | New firmware available for Front Door Camera. | 2024-12-24 07:00:00 |
| Edit Copy Delete | Battery replacement needed for backyard sensors. | 2024-12-24 08:15:00 |
| Edit Copy Delete | New updates available for all linked devices. | 2024-12-24 09:30:04 |
| Edit Copy Delete | Garden lights turned off after evening schedule. | 2024-12-24 10:00:00 |
| Edit Copy Delete | Driveway sensors activated at 22:00. | 2024-12-24 10:45:00 |

Check all With selected: Edit Copy Delete Export

**Query 10:** List unread notification by user 1

**Answer:**

```
SELECT message, timestamp
FROM notification
WHERE user_id = 1 AND read_status = 'unread';
```



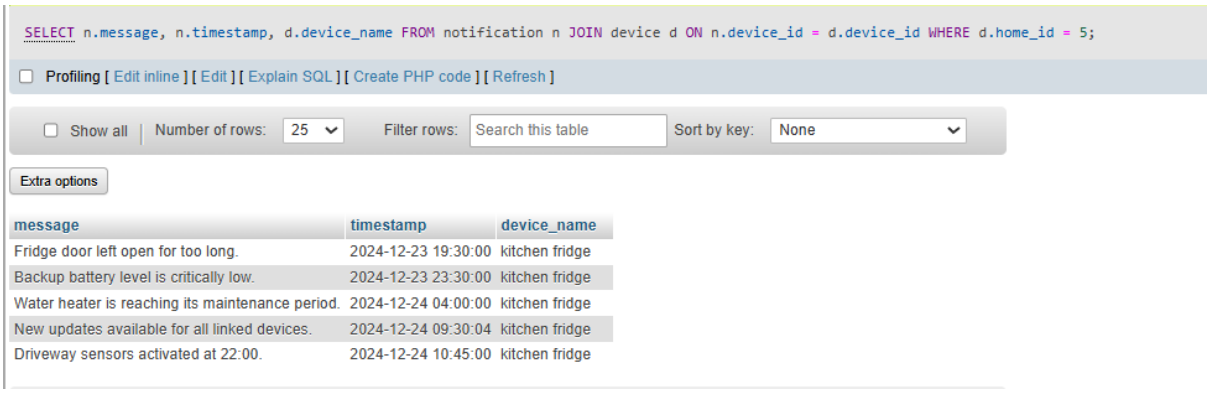

**Query 11:** Notification for all devices in a specific home

**Answer:**

SELECT n.message, n.timestamp, d.device_name

FROM notification n

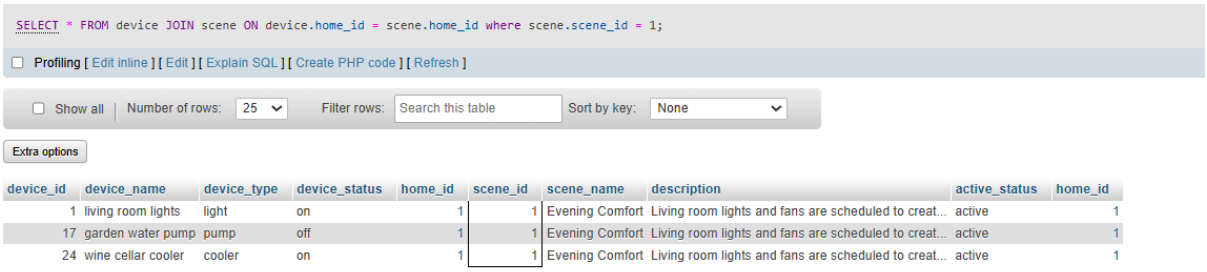
JOIN device d ON n.device_id = d.device_id WHERE d.home_id = 5;




**Query 12:** Get the details of devices that are part of scene

**Answer:**

SELECT * FROM device JOIN scene ON device.home_id = scene.home_id where scene.scene_id = 1;



```
SELECT * FROM device JOIN scene ON device.home_id = scene.home_id where scene.scene_id = 1;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

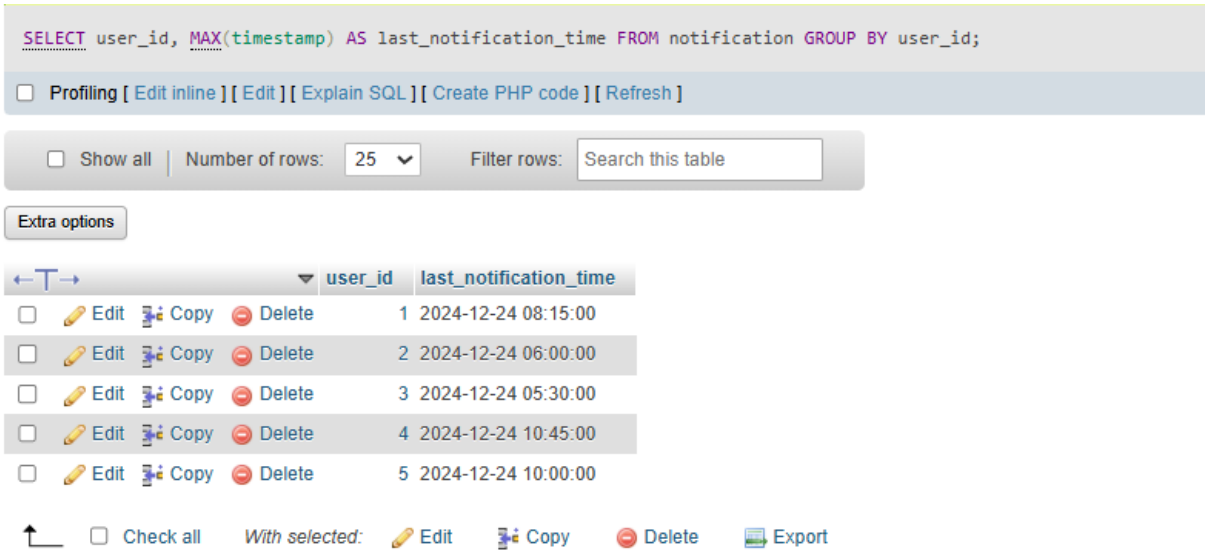
Extra options

| device_id | device_name | device_type | device_status | home_id | scene_id | scene_name | description | active_status | home_id |
|---|---|---|---|---|---|---|---|---|---|
| 1 | living room lights | light | on | 1 | 1 | Evening Comfort | Living room lights and fans are scheduled to creat... | active | 1 |
| 17 | garden water pump | pump | off | 1 | 1 | Evening Comfort | Living room lights and fans are scheduled to creat... | active | 1 |
| 24 | wine cellar cooler | cooler | on | 1 | 1 | Evening Comfort | Living room lights and fans are scheduled to creat... | active | 1 |

**Query 13:** Most Recent Notification for Each User

**Answer:**

```
SELECT user_id, MAX(timestamp) AS last_notification_time
FROM notification
GROUP BY user_id;
```



```
SELECT user_id, MAX(timestamp) AS last_notification_time FROM notification GROUP BY user_id;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

| | user_id | last_notification_time |
|---|---|---|
| Edit Copy Delete | 1 | 2024-12-24 08:15:00 |
| Edit Copy Delete | 2 | 2024-12-24 06:00:00 |
| Edit Copy Delete | 3 | 2024-12-24 05:30:00 |
| Edit Copy Delete | 4 | 2024-12-24 10:45:00 |
| Edit Copy Delete | 5 | 2024-12-24 10:00:00 |

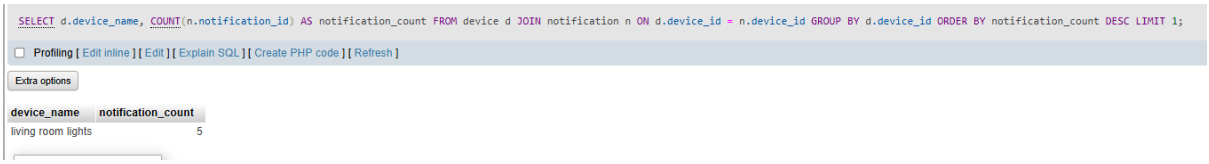
Check all With selected: Edit Copy Delete Export

**Query 14:** Find the Most Frequently Notified Device

**Answer:**

```
SELECT d.device_name, COUNT(n.notification_id) AS notification_count
FROM device d
JOIN notification n ON d.device_id = n.device_id
GROUP BY d.device_id
```

ORDER BY notification_count DESC LIMIT 1;



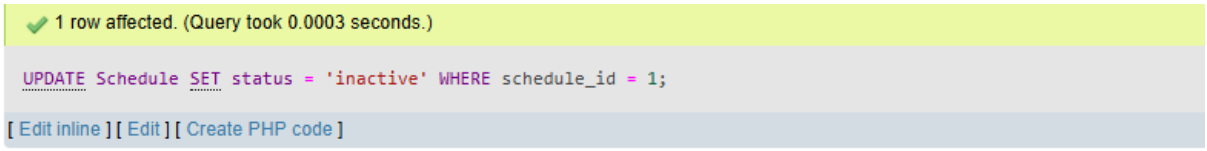

**Query 15:** Deactivate a schedule

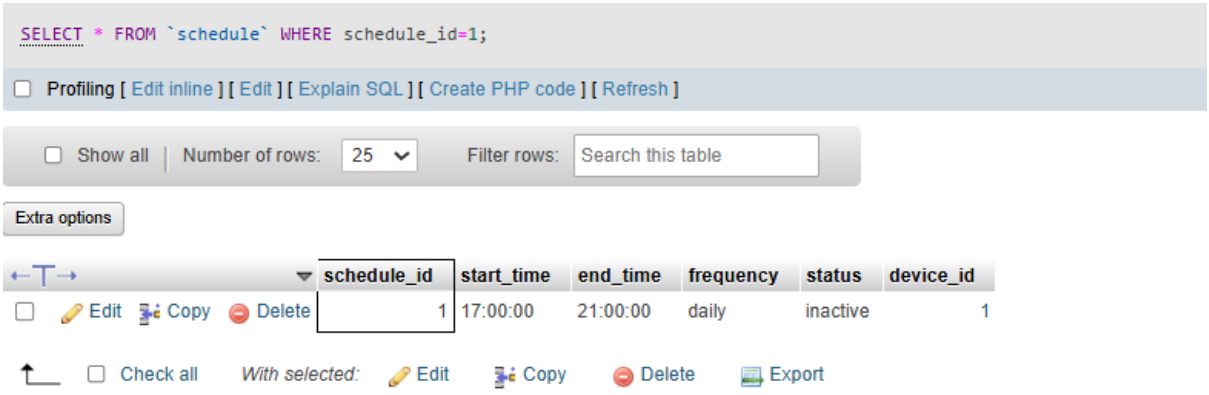
**Answer:**

UPDATE Schedule

SET status = 'inactive'

WHERE schedule_id = 1;




## 4 Security Scenario

A smart home system's security scenario is centred on preserving security across an interconnected network of users, devices, and automated processes. Home and device tables are used to connect devices such as cameras and sensors to their appropriate homes. The scene database contains predetermined security actions, such as locking doors or turning on surveillance cameras, and the

schedule table is used to schedule when they should be performed. Authorised users, whose responsibilities and permissions are specified in the role table, receive alerts regarding questionable activities. This all-inclusive system guarantees effective monitoring, prompt notifications, and controlled access to improve overall home security.

### Role Creation:

Define specific roles that determine the level of access and permissions for users in the system, such as admin, guest, and resident.

```
CREATE ROLE admin;
CREATE ROLE resident;
CREATE ROLE guest;

```

### Grant Privileges to Roles:

Assign privileges to the created roles based on their responsibilities.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON smart_home_system.* TO admin;
GRANT SELECT, INSERT ON smart_home_system.* TO resident;
GRANT SELECT ON smart_home_system.* TO guest;

```

### Create Users:

Create user accounts that will access the smart home system.

```
CREATE USER 'Jay'@'localhost' IDENTIFIED BY 'password123';
CREATE USER 'Albert'@'localhost' IDENTIFIED BY 'password123';
CREATE USER 'Ria'@'localhost' IDENTIFIED BY 'password123';
CREATE USER 'Max'@'localhost' IDENTIFIED BY 'password123';
CREATE USER 'Ira'@'localhost' IDENTIFIED BY 'password123';
CREATE USER 'Ana'@'localhost' IDENTIFIED BY 'password123';

```

### Assign Roles to Users:

Grant roles to the created users to define their permissions.

```
GRANT 'admin_role' TO 'Jay'@'localhost';
GRANT 'admin_role' TO 'Albert'@'localhost';

GRANT 'regular_role' TO 'Ria'@'localhost';
GRANT 'regular_role' TO 'Max'@'localhost';

GRANT 'guest_role' TO 'Ira'@'localhost';
GRANT 'guest_role' TO 'Ana'@'localhost';

```

### Show Grants for Users:

View the privileges and roles granted to a specific user.

```
SHOW GRANTS FOR 'Jay'@'localhost';
SHOW GRANTS FOR 'Ria'@'localhost';
SHOW GRANTS FOR 'albert'@'localhost';
SHOW GRANTS FOR 'Max'@'localhost';
SHOW GRANTS FOR 'Ira'@'localhost';
SHOW GRANTS FOR 'Ana'@'localhost';
```

### Output:

```
SHOW GRANTS FOR 'Jay'@'localhost';
```

Profiling [ Edit inline ] [ Edit ] [ Create PHP code ] [ Refresh ]

Extra options

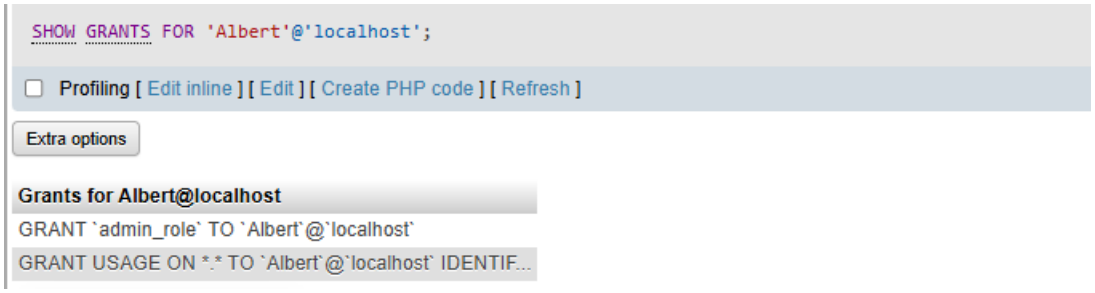
| Grants for Jay@localhost |
| --- |
| GRANT `admin_role` TO `Jay`@`localhost` |
| GRANT USAGE ON *.* TO `Jay`@`localhost` IDENTIFIED... |

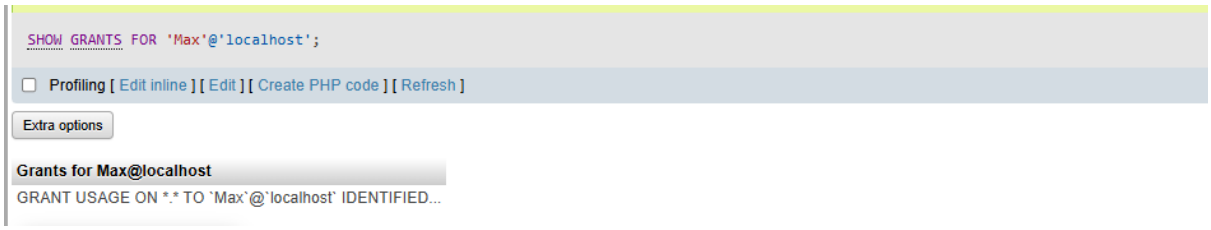
```
SHOW GRANTS FOR 'Ria'@'localhost';
```

Profiling [ Edit inline ] [ Edit ] [ Create PHP code ] [ Refresh ]

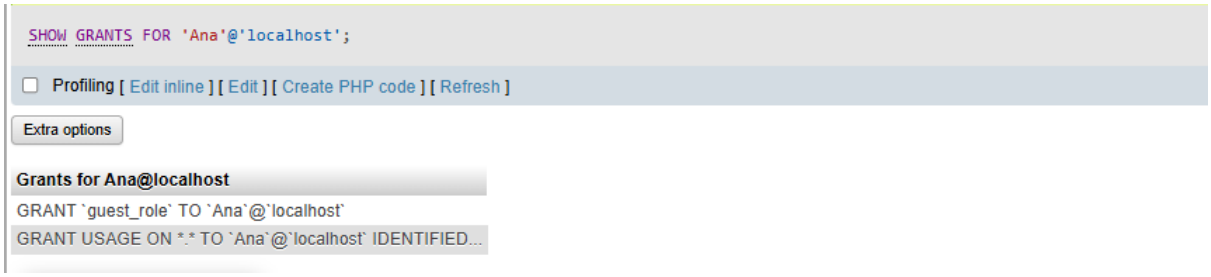
Extra options

| Grants for Ria@localhost |
| --- |
| GRANT `regular_role` TO `Ria`@`localhost` |
| GRANT USAGE ON *.* TO `Ria`@`localhost` IDENTIFIED... |








**Example table of roles assigned:**

| User Name | Role | Privileges |
|---|---|---|
| **Jay** | Admin Role | All privileges on all databases |
| **Albert** | Admin Role | All privileges on all databases |
| **Ria** | Regular Role | SELECT, INSERT, UPDATE, DELETE |
| **Max** | Regular Role | SELECT, INSERT, UPDATE, DELETE |
| **Ira** | Guest Role | SELECT only |
| **Ana** | Guest Role | SELECT only |

**Steps for implementation:**

- **Define Roles:** The admin role has full control over the system, the resident role can manage devices and schedules, and the guest role can view statuses but cannot modify configurations.
- **Grant Privileges:** Assign privileges to each role using GRANT commands to restrict or allow access to specific tables and actions.
- **Create and Assign Users:** Create user accounts and assign appropriate roles to ensure only authorized actions are performed.
- **Audit Roles and Privileges:** Use the SHOW GRANTS command to audit and verify role assignments and permissions for each user.

# 5 Conclusion

Database technologies make it possible for a variety of applications to store, handle, and retrieve data efficiently. Relational databases guarantee organisation and consistency, whereas distributed, object-oriented, graph, and NoSQL databases provide flexibility, scalability, and sophisticated analysis. By addressing a variety of data requirements, they enable contemporary, flexible systems.

The Smart Home System demonstrates a cohesive integration of technology to enhance convenience, security, and automation in modern living spaces. By connecting users, homes, devices, and roles through a well-structured database, the system ensures seamless functionality and efficient management. Features like device scheduling, automated notifications, and role-based privileges provide personalized control while maintaining high security standards. This system exemplifies the potential of smart technology to improve quality of life, making daily activities more manageable and creating a safe, intelligent, and adaptable environment for users.

# 6 References

A **Relational Data Base Management System** Giampio Bracchi, A. Fedeli, P. Paolini Politecnico di Milano, Italy. published: 1 august 1972. Available at: https://dl.acm.org/doi/abs/10.1145/800194.805904.

E.F. Codd, a British computer scientist, introduced the **Advantages and Disadvantages** of relational database model in his groundbreaking 1970 paper titled **"A Relational Model of Data for Large Shared Data Banks."**

**NoSQL Database** A B M Moniruzzaman and Syed Akhter Hossain Department of Computer Science and Engineering Daffodil International University. Available at: https://arxiv.org/pdf/1307.0191. Accessed March 14, 2013.

**Johan Oskarsson**, a developer at Last.fm, reintroduced and popularized the term **Advantages and Disadvantages** of **NoSQL** in early 2009, building on Carlo Strozzi's earlier introduction in 1998.

**Graph Database Modelling -** Amine Ghrab, Oscar Romero, Sabri Skhiri, Alejandro Vaisman, Esteban Zimányi. Available at: https://arxiv.org/abs/1602.00503.

Marijan, B., 2021. Distributed Database (Features, Benefits & Drawbacks). Published: December 2021. Available at: https://phoenixnap.com/kb/distributed-database.

**Object-Oriented Database Systems** Fransois Bancilhon Altair BP 105, 78153 Le Chesnay Cedex France published: March 1988 Available at: https://dl.acm.org/doi/abs/10.1145/308386.308429