**Smart Home**

In our rapidly advancing digital age, the smart home has become a tangible reality, where modern technologies integrate to provide comfort, security, and efficiency in our daily lives. At the heart of this sophisticated system lies a robust and flexible database, capable of managing the vast amounts of data flowing from various smart devices.

This application aims to provide an integrated solution for managing smart home data efficiently and effectively. Through a user-friendly interface, the application allows users to track and analyze their device data, automate routine tasks, and make informed decisions to improve energy consumption and enhance the level of security and comfort in their homes.

Whether you are an early adopter of smart home technologies or just beginning to explore its possibilities, this application provides you with the necessary tools to take full control of your smart home data and make the most of its capabilities.

**Customer Needs Analysis for a Smart Home System:**

A smart home system generally aims to achieve the following for the user:

* **Comfort and Automation:** Facilitating the control of various devices and systems in the home and automating some routine tasks.
* **Security:** Providing advanced security solutions to protect the home and its occupants.
* **Energy Efficiency:** Monitoring and improving energy consumption to reduce costs and conserve the environment.
* **Centralized Control:** Providing a unified interface to control all connected devices and systems.
* **Remote Monitoring:** The ability to monitor the status of the home and control some devices remotely.
* **Custom Scenario Creation:** The ability to create custom scenarios to run a group of devices based on specific conditions (such as "movie mode" that dims the lights and closes the curtains).

**Identifying the Entities:**

Entities are the fundamental elements about which information will be stored in the database. For a Smart Home system, the main entities can include:

* **Device:** Represents any smart device connected to the system (such as a lamp, lock, camera, thermostat, motion sensor, smart curtain, etc.).
  + **Potential Attributes:** DeviceID, DeviceName, DeviceType, DeviceStatus (on/off, open/closed, etc.), Location (Living Room, Kitchen, etc.), Manufacturer, Model.
* **User:** Represents the residents of the home who interact with the system.
  + **Potential Attributes:** UserID, Username, Password, Email, PhoneNumber, Role (Administrator, Regular User).
* **Room:** Represents the different spaces in the home.
  + **Potential Attributes:** RoomID, RoomName.
* **Scenario:** Represents a set of actions performed when a specific trigger occurs (such as "Good Morning" opens the curtains and gradually turns on the lights).
  + **Potential Attributes:** ScenarioID, ScenarioName, Description.
* **Rule:** Represents a condition that leads to the execution of a specific action (such as "If motion is detected at 10 PM, turn on the camera").
  + **Potential Attributes:** RuleID, RuleName, Condition, Action.
* **SensorReading:** Represents the data collected from various sensors (such as temperature, humidity, motion, light).
  + **Potential Attributes:** ReadingID, DeviceID (as a foreign key), SensorType, Value, Unit, Timestamp.
* **ControlLog:** Records the control operations performed by users or the system on the devices.
  + **Potential Attributes:** LogID, UserID (as a foreign key), DeviceID (as a foreign key), Action, Timestamp, Status (Success/Failure).

**Identifying the Users:**

Users of a Smart Home system can be classified into two main categories:

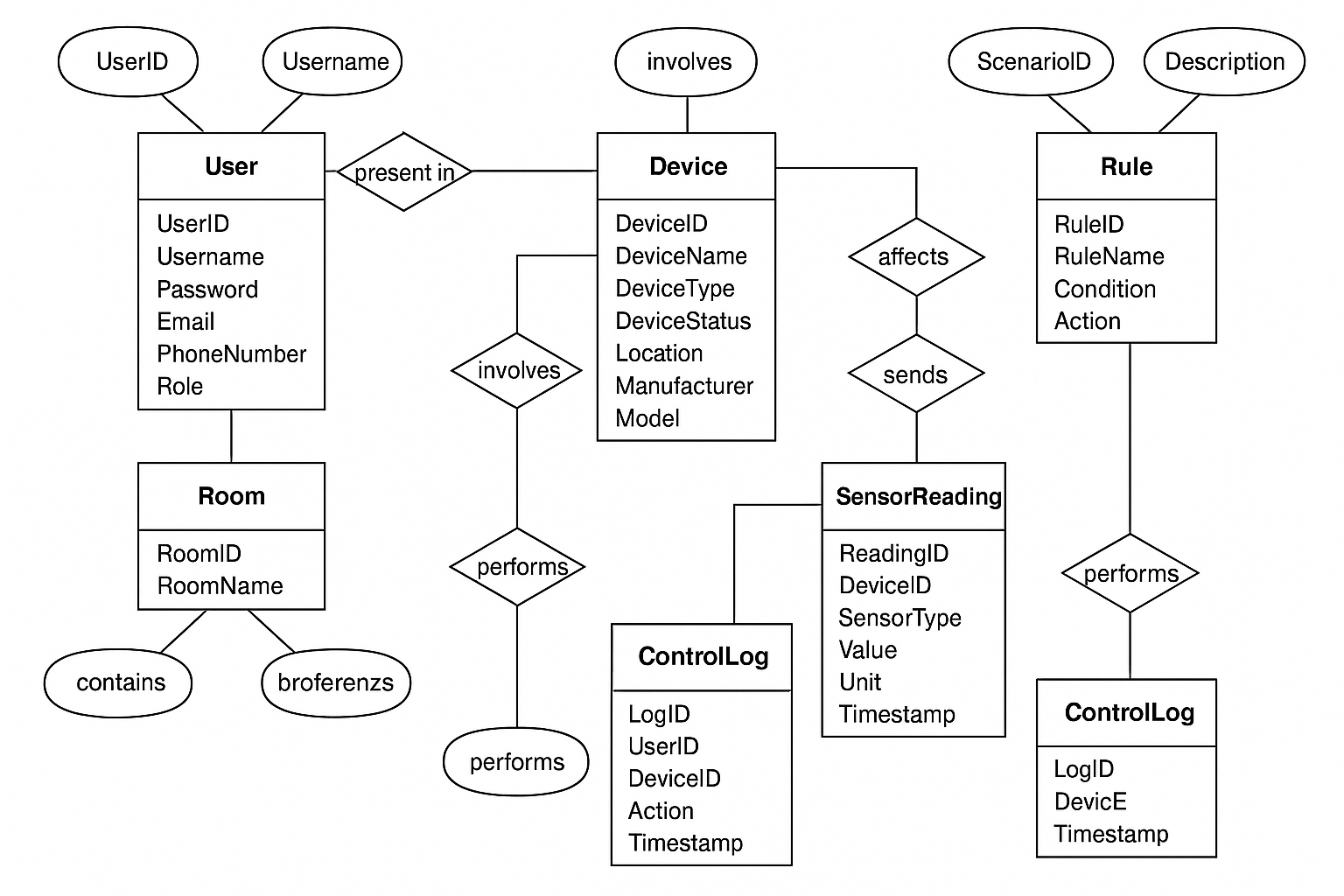
* **End Users:** These are the residents of the home who directly interact with the system to control devices, activate scenarios, and monitor the status. They may have different levels of permissions.
* **Administrator:** There may be one or more users with full privileges to set up the system, add and manage devices and users, and configure scenarios and rules.

**Identifying the Relationships Between Entities:**

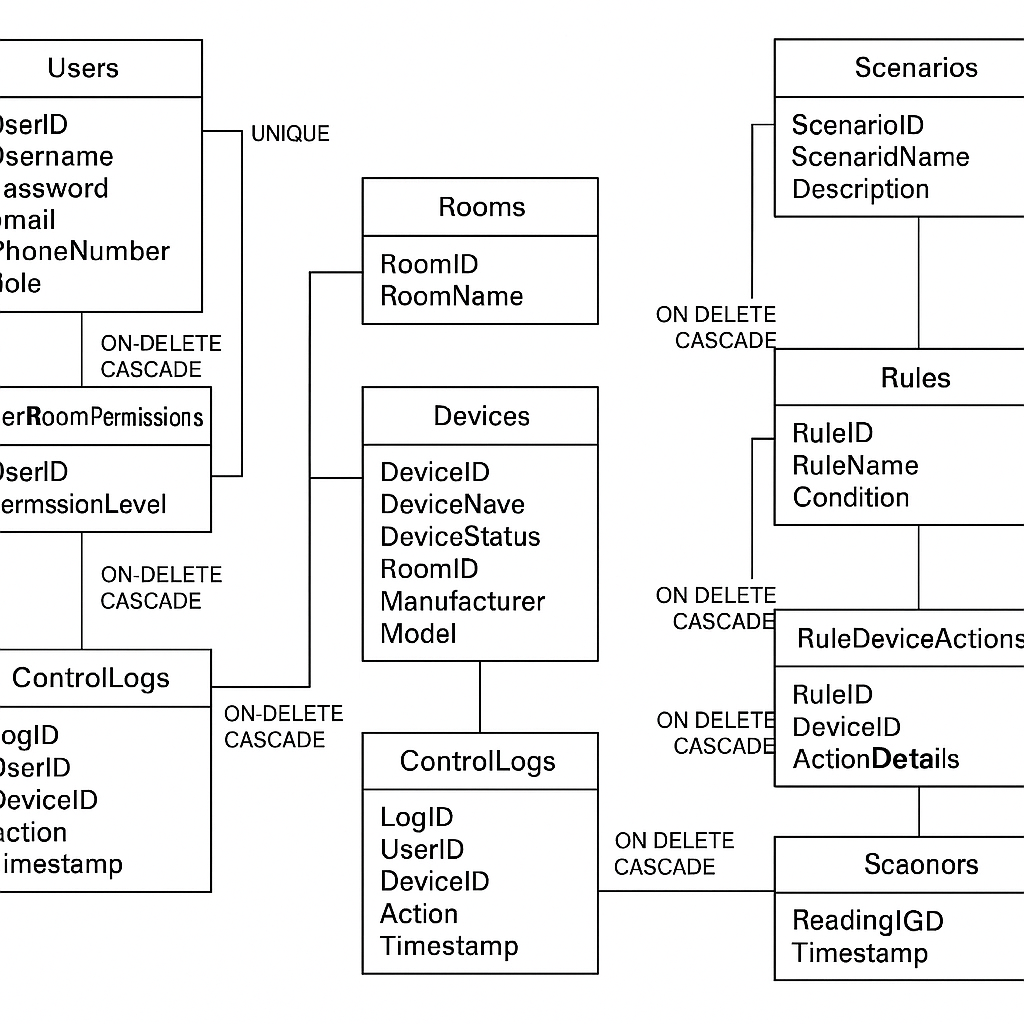
Now, let's define how these entities relate to each other:

* **User - Room:**
  + A user can be present in a specific room (current location relationship).
  + A user can have access permissions to certain rooms (permission relationship).
* **Room - Device:**
  + A room contains many devices (one-to-many relationship: Room 1 --\* Device).
  + A device belongs to one room (many-to-one relationship: Device \*--1 Room).
* **Scenario - Device:**
  + A scenario involves many actions on different devices (many-to-many relationship: Scenario *--* Device). This relationship can be represented by an intermediary table containing (ScenarioID, DeviceID, Action).
* **Rule - Device:**
  + A rule affects one or more devices (one-to-many or many-to-many relationship, depending on the complexity of the rule). It can be represented by a many-to-many relationship using an intermediary table (RuleID, DeviceID, Action).
* **Device - SensorReading:**
  + A device (if it's a sensor) sends many sensor readings (one-to-many relationship: Device 1 --\* SensorReading).
  + A sensor reading belongs to one device (many-to-one relationship: SensorReading \*--1 Device).
* **User - ControlLog:**
  + A user performs many control operations that are logged (one-to-many relationship: User 1 --\* ControlLog).
  + A control log belongs to one user (many-to-one relationship: ControlLog \*--1 User).
* **Device - ControlLog:**
  + Many control operations are recorded for a specific device (one-to-many relationship: Device 1 --\* ControlLog).
  + A control log pertains to one device (many-to-one relationship: ControlLog \*--1 Device).

**ERD:**

****

**Mapping:**

****

**Normalization:**

**1. First Normal Form (1NF):**

* **Condition:** Each column must contain atomic (indivisible) values, and there should be no repeating groups of columns.
* **Table Review:** All the tables we have currently created already satisfy the condition of the First Normal Form, as each column contains only a single value, and there is no clear repetition in groups of columns within the same table.

**2. Second Normal Form (2NF):**

* **Conditions:**
  + The table must be in the First Normal Form (1NF).
  + All non-key attributes must be fully functionally dependent on the entire primary key.
* **Table Review:**
  + **Users Table:** The primary key is UserID. All other columns (Username, Password, Email, PhoneNumber, Role) are fully dependent on UserID. **Therefore, this table is in the Second Normal Form.**
  + **Rooms Table:** The primary key is RoomID. The RoomName column is fully dependent on RoomID. **Therefore, this table is in the Second Normal Form.**
  + **Devices Table:** The primary key is DeviceID. All other columns (DeviceName, DeviceType, DeviceStatus, RoomID, Manufacturer, Model) are fully dependent on DeviceID. **Therefore, this table is in the Second Normal Form.**
  + **Scenarios Table:** The primary key is ScenarioID. The ScenarioName and Description columns are fully dependent on ScenarioID. **Therefore, this table is in the Second Normal Form.**
  + **Rules Table:** The primary key is RuleID. The RuleName, Condition, and Action columns are fully dependent on RuleID. **Therefore, this table is in the Second Normal Form.**
  + **SensorReadings Table:** The primary key is ReadingID. All other columns (DeviceID, SensorType, Value, Unit, Timestamp) are fully dependent on ReadingID. **Therefore, this table is in the Second Normal Form.**
  + **ControlLogs Table:** The primary key is LogID. All other columns (UserID, DeviceID, Action, Timestamp, Status) are fully dependent on LogID. **Therefore, this table is in the Second Normal Form.**
  + **UserRoomPermissions Table:** The composite primary key is (UserID, RoomID). The PermissionLevel column is fully dependent on this composite key (the permission of a specific user in a specific room). **Therefore, this table is in the Second Normal Form.**
  + **ScenarioDeviceActions Table:** The composite primary key is (ScenarioID, DeviceID, ActionDetails). **Here we need to pay attention:** Does ActionDetails depend solely on the entire primary key? In this case, yes, the action details are related to a specific scenario and device. **Therefore, this table is in the Second Normal Form.**
  + **RuleDeviceActions Table:** The composite primary key is (RuleID, DeviceID, ActionDetails). Similarly, the action details are related to a specific rule and device. **Therefore, this table is in the Second Normal Form.**

**3. Third Normal Form (3NF):**

* **Conditions:**
  + The table must be in the Second Normal Form (2NF).
  + There should be no transitive dependencies (no non-key attributes dependent on other non-key attributes).
* **Table Review:**
  + **Users Table:** There are no transitive dependencies. **Therefore, this table is in the Third Normal Form.**
  + **Rooms Table:** There are no transitive dependencies. **Therefore, this table is in the Third Normal Form.**
  + **Devices Table:** The RoomID column is a foreign key referencing the Rooms table. There are no dependencies between other non-key attributes. **Therefore, this table is in the Third Normal Form.**
  + **Scenarios Table:** There are no transitive dependencies. **Therefore, this table is in the Third Normal Form.**
  + **Rules Table:** There are no transitive dependencies. **Therefore, this table is in the Third Normal Form.**
  + **SensorReadings Table:** The DeviceID column is a foreign key. There are no dependencies between other non-key attributes. **Therefore, this table is in the Third Normal Form.**
  + **ControlLogs Table:** The UserID and DeviceID columns are foreign keys. There are no dependencies between other non-key attributes. **Therefore, this table is in the Third Normal Form.**
  + **UserRoomPermissions Table:** All columns are part of the primary key or describe the relationship between the primary keys. **Therefore, this table is in the Third Normal Form.**
  + **ScenarioDeviceActions Table:** All columns are part of the primary key or describe the relationship between the primary keys. **Therefore, this table is in the Third Normal Form.**
  + **RuleDeviceActions Table:** All columns are part of the primary key or describe the relationship between the primary keys. **Therefore, this table is in the Third Normal Form.**

**SQL Code to Create Tables:**

-- Users Table

CREATE TABLE Users (

UserID INT PRIMARY KEY,

Username VARCHAR(255) NOT NULL UNIQUE,

Password VARCHAR(255) NOT NULL,

Email VARCHAR(255) UNIQUE,

PhoneNumber VARCHAR(50),

Role VARCHAR(50)

);

-- Rooms Table

CREATE TABLE Rooms (

RoomID INT PRIMARY KEY,

RoomName VARCHAR(255) NOT NULL

);

-- Devices Table

CREATE TABLE Devices (

DeviceID INT PRIMARY KEY,

DeviceName VARCHAR(255) NOT NULL,

DeviceType VARCHAR(100),

DeviceStatus VARCHAR(50),

RoomID INT,

Manufacturer VARCHAR(255),

Model VARCHAR(255),

FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)

);

-- Scenarios Table

CREATE TABLE Scenarios (

ScenarioID INT PRIMARY KEY,

ScenarioName VARCHAR(255) NOT NULL,

Description TEXT

);

-- Rules Table

CREATE TABLE Rules (

RuleID INT PRIMARY KEY,

RuleName VARCHAR(255) NOT NULL,

Condition TEXT NOT NULL,

Action TEXT NOT NULL

);

-- SensorReadings Table

CREATE TABLE SensorReadings (

ReadingID INT PRIMARY KEY AUTO\_INCREMENT,

DeviceID INT,

SensorType VARCHAR(100),

Value DECIMAL(10, 2),

Unit VARCHAR(50),

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (DeviceID) REFERENCES Devices(DeviceID) ON DELETE CASCADE

);

-- ControlLogs Table

CREATE TABLE ControlLogs (

LogID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

DeviceID INT,

Action VARCHAR(255) NOT NULL,

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

Status VARCHAR(50),

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE SET NULL,

FOREIGN KEY (DeviceID) REFERENCES Devices(DeviceID) ON DELETE CASCADE

);

-- UserRoomPermissions Table

CREATE TABLE UserRoomPermissions (

UserID INT,

RoomID INT,

PermissionLevel VARCHAR(50),

PRIMARY KEY (UserID, RoomID),

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE,

FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID) ON DELETE CASCADE

);

-- ScenarioDeviceActions Table

CREATE TABLE ScenarioDeviceActions (

ScenarioID INT,

DeviceID INT,

ActionDetails VARCHAR(255) NOT NULL,

PRIMARY KEY (ScenarioID, DeviceID, ActionDetails),

FOREIGN KEY (ScenarioID) REFERENCES Scenarios(ScenarioID) ON DELETE CASCADE,

FOREIGN KEY (DeviceID) REFERENCES Devices(DeviceID) ON DELETE CASCADE

);

-- RuleDeviceActions Table

CREATE TABLE RuleDeviceActions (

RuleID INT,

DeviceID INT,

ActionDetails VARCHAR(255) NOT NULL,

PRIMARY KEY (RuleID, DeviceID, ActionDetails),

FOREIGN KEY (RuleID) REFERENCES Rules(RuleID) ON DELETE CASCADE,

FOREIGN KEY (DeviceID) REFERENCES Devices(DeviceID) ON DELETE CASCADE

);

**Defining User Privileges (Examples):**

-- Assuming we have users named 'user1' and 'admin1' in the Users table

-- Grant data viewing permissions to user 'user1'

GRANT SELECT ON Devices TO 'user1';

GRANT SELECT ON SensorReadings TO 'user1';

GRANT SELECT ON ControlLogs TO 'user1';

-- Grant device control permissions to user 'user1'

GRANT UPDATE ON Devices TO 'user1';

-- Note: Additional privileges may be required to execute specific commands, such as turning devices on/off

-- Grant all privileges to user 'admin1' (system administrator)

GRANT ALL PRIVILEGES ON \* TO 'admin1';

-- Alternatively, you can specify privileges more granularly

GRANT SELECT, INSERT, UPDATE, DELETE ON Users TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON Rooms TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON Devices TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON Scenarios TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON Rules TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON SensorReadings TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON ControlLogs TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON UserRoomPermissions TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON ScenarioDeviceActions TO 'admin1';

GRANT SELECT, INSERT, UPDATE, DELETE ON RuleDeviceActions TO 'admin1';

-- Revoke data viewing permissions from user 'user1'

REVOKE SELECT ON Devices FROM 'user1';

-- Display user privileges

SHOW GRANTS FOR 'user1';

SHOW GRANTS FOR 'admin1';

**Example of data**

-- Inserting data into the Users table

INSERT INTO Users (UserID, Username, Password, Email, PhoneNumber, Role) VALUES

(1, 'user1', 'password123', 'user1@example.com', '123-456-7890', 'User'),

(2, 'user2', 'pass456', 'user2@example.com', '987-654-3210', 'User'),

(3, 'admin1', 'adminpass', 'admin1@example.com', '555-123-4567', 'Admin'),

(4, 'user3', 'pass789', 'user3@example.com', '111-222-3333', 'User'),

(5, 'user4', 'pass234', 'user4@example.com', '444-555-6666', 'User'),

(6, 'user5', 'pass567', 'user5@example.com', '777-888-9999', 'User'),

(7, 'editor1', 'editorpass', 'editor1@example.com', '222-333-4444', 'Editor'),

(8, 'user6', 'pass890', 'user6@example.com', '333-444-5555', 'User'),

(9, 'user7', 'pass101', 'user7@example.com', '666-777-8888', 'User'),

(10, 'admin2', 'admin2pass', 'admin2@example.com', '999-000-1111', 'Admin');

-- Inserting data into the Rooms table

INSERT INTO Rooms (RoomID, RoomName) VALUES

(1, 'Living Room'),

(2, 'Kitchen'),

(3, 'Bedroom 1'),

(4, 'Bedroom 2'),

(5, 'Bathroom 1'),

(6, 'Bathroom 2'),

(7, 'Dining Room'),

(8, 'Office'),

(9, 'Garage'),

(10, 'Garden');

-- Inserting data into the Devices table

INSERT INTO Devices (DeviceID, DeviceName, DeviceType, DeviceStatus, RoomID, Manufacturer, Model) VALUES

(1, 'Living Room TV', 'TV', 'Off', 1, 'Samsung', 'QLED-65'),

(2, 'Kitchen Lights', 'Light', 'On', 2, 'Philips', 'Hue-Bulb'),

(3, 'Bedroom 1 Thermostat', 'Thermostat', '20C', 3, 'Nest', 'Learning-3rdGen'),

(4, 'Bedroom 2 Fan', 'Fan', 'Off', 4, 'Honeywell', 'QuietCool'),

(5, 'Bathroom 1 Shower', 'Shower', 'Off', 5, 'Kohler', 'Digital'),

(6, 'Bathroom 2 Mirror Lights', 'Light', 'On', 6, 'Philips', 'Hue-Strip'),

(7, 'Dining Room Chandelier', 'Light', 'Off', 7, 'Osram', 'Crystal'),

(8, 'Office Computer', 'Computer', 'On', 8, 'Dell', 'XPS'),

(9, 'Garage Door', 'Door', 'Closed', 9, 'Chamberlain', 'MyQ'),

(10, 'Garden Sprinkler', 'Sprinkler', 'Off', 10, 'RainBird', 'Smart'),

(11, 'Living Room Soundbar', 'Speaker', 'On', 1, 'Sonos', 'Arc'),

(12, 'Kitchen Refrigerator', 'Refrigerator', 'On', 2, 'LG', 'SmartFridge');

-- Inserting data into the Scenarios table

INSERT INTO Scenarios (ScenarioID, ScenarioName, Description) VALUES

(1, 'Movie Night', 'Dims lights, turns on TV and soundbar'),

(2, 'Good Morning', 'Opens curtains, turns on lights gradually'),

(3, 'Dinner Time', 'Sets dining room lights, plays soft music'),

(4, 'Bedtime', 'Turns off all lights, locks doors'),

(5, 'Party Mode', 'Sets colorful lights, plays music'),

(6, 'Vacation Mode', 'Turns off most devices, activates security system'),

(7, 'Wake Up Alarm', 'Turns on lights and plays music'),

(8, 'Relax Time', 'Dims lights and plays calm music'),

(9, 'Cleaning Mode', 'Turns on all lights'),

(10, 'Security Check', 'Checks all cameras and sensors');

-- Inserting data into the Rules table

INSERT INTO Rules (RuleID, RuleName, Condition, Action) VALUES

(1, 'Motion in Living Room at Night', 'IF motion detected in Living Room AND time is between 22:00 and 06:00', 'Turn on Living Room Lights and Camera'),

(2, 'Temperature Too High', 'IF temperature in Bedroom 1 is greater than 25C', 'Turn on Bedroom 1 Fan'),

(3, 'Door Open Too Long', 'IF Garage Door is open for more than 5 minutes', 'Send notification to user'),

(4, 'Low Battery Warning', 'IF battery level of any device is less than 20%', 'Send low battery notification'),

(5, 'Water Leak Detected', 'IF water leak detected in Bathroom 1', 'Turn off water supply and send alert'),

(6, 'High Humidity', 'IF humidity in Garden is greater than 80%', 'Turn off Garden Sprinkler'),

(7, 'Smoke Detected', 'IF smoke is detected', 'Activate alarm and call emergency services'),

(8, 'Sunrise', 'IF time is sunrise', 'Open Bedroom Curtains'),

(9, 'Sunset', 'IF time is sunset', 'Turn on Living Room Lights'),

(10, 'User Arrives Home', 'IF user arrives home', 'Turn on Welcome Lights and unlock door');

-- Inserting data into the SensorReadings table

INSERT INTO SensorReadings (DeviceID, SensorType, Value, Unit, Timestamp) VALUES

(3, 'Temperature', 22.5, 'C', '2024-07-24 10:00:00'),

(3, 'Temperature', 23.0, 'C', '2024-07-24 10:15:00'),

(10, 'Humidity', 65.0, '%', '2024-07-24 10:30:00'),

(10, 'Humidity', 68.0, '%', '2024-07-24 10:45:00'),

(1, 'Light Level', 100.0, 'lux', '2024-07-24 11:00:00'),

(1, 'Light Level', 150.0, 'lux', '2024-07-24 11:15:00'),

(4, 'Fan Speed', 0.0, 'rpm', '2024-07-24 11:30:00'),

(4, 'Fan Speed', 1200.0, 'rpm', '2024-07-24 11:45:00'),

(5, 'Water Flow', 0.0, 'L/min', '2024-07-24 12:00:00'),

(5, 'Water Flow', 10.5, 'L/min', '2024-07-24 12:15:00'),

(2, 'Light Level', 50, 'lux', '2024-07-24 12:30:00'),

(2, 'Light Level', 200, 'lux', '2024-07-24 12:45:00');

-- Inserting data into the ControlLogs table

INSERT INTO ControlLogs (UserID, DeviceID, Action, Timestamp, Status) VALUES

(1, 1, 'Turn On', '2024-07-24 09:55:00', 'Success'),

(2, 2, 'Turn Off', '2024-07-24 10:05:00', 'Success'),

(1, 3, 'Set Temperature to 22C', '2024-07-24 10:20:00', 'Success'),

(4, 4, 'Turn On', '2024-07-24 11:10:00', 'Success'),

(5, 5, 'Start Shower', '2024-07-24 12:05:00', 'Success'),

(1, 1, 'Turn Off', '2024-07-24 13:00:00', 'Success'),

(3, 8, 'Start Computer', '2024-07-24 08:00:00', 'Success'),

(2, 9, 'Open Garage Door', '2024-07-24 14:00:00', 'Failure'),

(1, 10, 'Start Sprinkler', '2024-07-24 15:00:00', 'Success'),

(1, 2, 'Turn On', '2024-07-24 16:00:00', 'Success'),

(4, 2, 'Turn Off', '2024-07-24 16:01:00', 'Success'),

(4, 12, 'Set Temperature to 5C', '2024-07-24 17:00:00', 'Success');

-- Inserting data into the UserRoomPermissions table

INSERT INTO UserRoomPermissions (UserID, RoomID, PermissionLevel) VALUES

(1, 1, 'Control'),

(1, 2, 'View'),

(2, 2, 'Control'),

(3, 1, 'Control'),

(3, 2, 'Control'),

(3, 3, 'Control'),

(4, 4, 'View'),

(5, 5, 'Control'),

(6, 6, 'View'),

(7, 7, 'Control'),

(8, 8, 'Control'),

(9, 9, 'Control'),

(10, 10, 'Control');

-- Inserting data into the ScenarioDeviceActions table

INSERT INTO ScenarioDeviceActions (ScenarioID, DeviceID, ActionDetails) VALUES

(1, 1, 'Turn On'),

(1, 11, 'Turn On'),

(1, 2, 'Turn Off'),

(2, 2, 'Turn On Gradually'),

(2, 4, 'Turn Off'),

(3, 2, 'Turn On'),

(3, 7, 'Turn On'),

(4, 1, 'Turn Off'),

(4, 2, 'Turn Off'),

(5, 2, 'Set Color to Red'),

(5, 7, 'Set Color to Blue'),

(6, 1, 'Turn Off');

-- Inserting data into the RuleDeviceActions table

INSERT INTO RuleDeviceActions (RuleID, DeviceID, ActionDetails) VALUES

(1, 1, 'Turn On'),

(1, 11, 'Turn On'),

(2, 4, 'Turn On'),

(3, 9, 'Send Notification'),

(4, 0, 'Send Notification'),

(5, 5, 'Turn Off Water'),

(5, 0, 'Send Alert'),

(6, 10, 'Turn Off'),

(7, 0, 'Activate Alarm'),

(7, 0, 'Call Emergency Services'),

(8, 4, 'Open Curtains'),

(9, 1, 'Turn On');

**Inquiries**

-- 1. Simple query to retrieve all users

-- This query retrieves all columns and all rows from the Users table.

SELECT \* FROM Users;

-- 2. Query to retrieve device names and types in a specific room

-- Retrieves the device name and type from the Devices table, and filters the results based on the RoomID.

SELECT DeviceName, DeviceType

FROM Devices

WHERE RoomID = 1; -- You can change RoomID as needed

-- 3. Query to search for devices whose names contain a specific word (text search)

-- Uses the LIKE operator to search for a pattern within the DeviceName column. Uses '%' as a wildcard character representing any string of characters.

SELECT DeviceName

FROM Devices

WHERE DeviceName LIKE '%Light%'; -- Searches for any device whose name contains the word "Light"

-- 4. Query to count the number of devices of each type (aggregate function)

-- Uses the COUNT() function to count the number of devices, and groups the results by device type.

SELECT DeviceType, COUNT(\*) AS NumberOfDevices

FROM Devices

GROUP BY DeviceType;

-- 5. Query to retrieve sensor readings in ascending chronological order

-- Retrieves all columns from the SensorReadings table and orders the results based on the timestamp.

SELECT \*

FROM SensorReadings

ORDER BY Timestamp ASC;

-- 6. Query to retrieve the latest sensor reading for each device

-- Uses the MAX() function to retrieve the latest timestamp for each device.

SELECT DeviceID, MAX(Timestamp) AS LastReadingTime

FROM SensorReadings

GROUP BY DeviceID;

-- 7. Query to retrieve users who have control permission in Room 1

SELECT Users.Username

FROM Users

JOIN UserRoomPermissions ON Users.UserID = UserRoomPermissions.UserID

WHERE UserRoomPermissions.RoomID = 1

AND UserRoomPermissions.PermissionLevel = 'Control';

-- 8. Query to retrieve devices and their status along with the name of the room they are in

SELECT

Devices.DeviceName,

Devices.DeviceStatus,

Rooms.RoomName

FROM

Devices

JOIN

Rooms ON Devices.RoomID = Rooms.RoomID;