User & Asset Database for Enerliving Platform

Doc-Type: Report

Author: Gregory Baltas

Doc-Status: Draft

Version / Revision: v1.0

Date: 31.03.2023

Table of Contents

[User & Asset Database for Enerliving Platform 1](#_Toc131409883)

[Document Version History 2](#_Toc131409884)

[Introduction 3](#_Toc131409885)

[Definitions 3](#_Toc131409886)

[Objectives 3](#_Toc131409887)

[Database Table Modelling 4](#_Toc131409888)

[Overview 4](#_Toc131409889)

[Tables Description 5](#_Toc131409890)

[Admins Table 5](#_Toc131409891)

[Roles Table 5](#_Toc131409892)

[Users Table 5](#_Toc131409893)

[Smart Meters Table 6](#_Toc131409894)

[Assets Table 6](#_Toc131409895)

[Energy Communities Table 6](#_Toc131409896)

[Main Associations Table 7](#_Toc131409897)

[Smart Meter Specs Table 7](#_Toc131409898)

[Smart Meter Tariffs Table 7](#_Toc131409899)

[Future Work 8](#_Toc131409900)

# Document Version History

|  |  |  |
| --- | --- | --- |
| **Version** | **Revision** | **Comments** |
| 1 | 0 | Initial draft – database design & future work |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

This report is dedicated in the description and explanation of the user\_and\_asset database of the **enerliving** platform. The database is based on SQL using PostgreSQL relational database manager system. The implementation code can be found in ICES gitlab [repository](https://git.list.lu/ices/enerliving/databases/user_and_assets) of the enerliving platform. If you don’t have access, please contact the repository manager David FIORELLI (email: david.fiorelli@list.lu). Ensure, also, that you have already an account with LIST’s gitlab.

## Definitions

The definition of main entities of the user\_and\_asset database is provided below:

* **Users:** People that are registered for accessing the enerliving platform. Different users can have different roles (i.e., access rights) to the platform features and functionalities.
* **Assets:** Equipment that refers to energy related devices such as battery energy storage systems, flexible and inflexible loads, photovoltaic generators, electric vehicles etc.
* **Smart meters:** At the point of grid connection usually there is meter (approved and sealed by the system operator) for measuring consumption production. The smart meters refer to these devices not to additional measurement devices**.**
* **Energy Communities:** The collection of one or more assets in a group is called an energy community.

## Objectives

The goal of the user\_and\_asset database is foremost to store the information of the main entities defined above. In addition, to store auxiliary information about the equipment specification, asset usage preferences, electricity tariffs and power contracts (fixed term) and external API credentials such as energy marker API. This process can be considered as the registration of the accounts for users, energy communities, smart meters, assets, external APIs, and their associations within the platform. A key aspect of this registration process is that the unique identifiers of each will be used by the rest of the databases and processes to distinguish incoming, outgoing, and internal data. For instance, when a new user, asset etc. is registered its ID (perhaps using a trigger or lambda function) will be inserted into the other databases responsible for storing real-time (RT) operation, short-term (ST) planning, long-term (LT) and historical data, as shown in Figure 1.

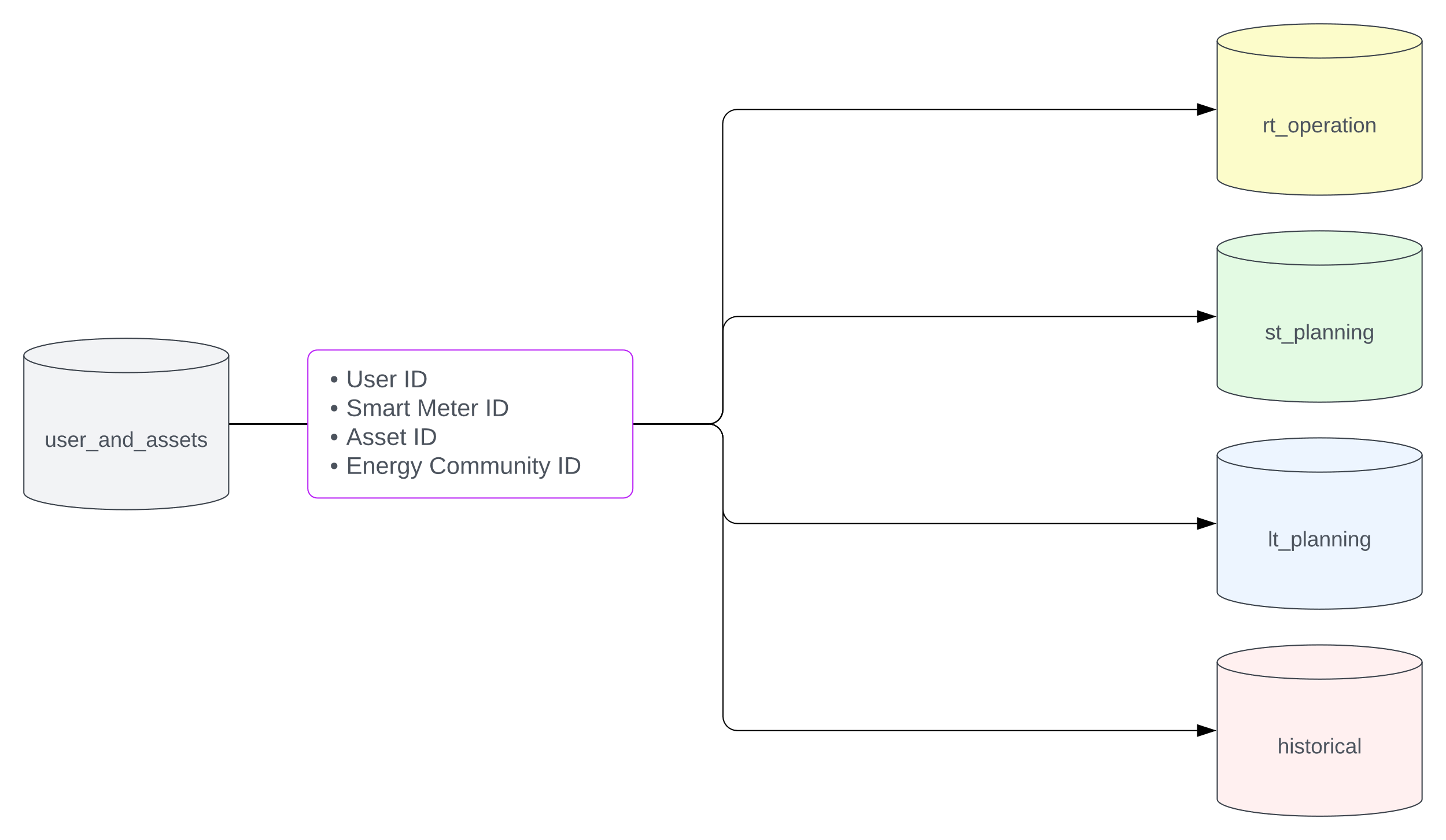


Figure 1: Registered IDs are inserted into the rest of the databases.

# Database Table Modelling

## Overview

Now, the database consists of 9 tables but there will be more added in the future. More details will be given in the future developments section. The overall structure of the database is show in Figure 2. In the following subsection each table will be discussed separately. The following points are however applicable to the database structure in general:

* Except admins table, all other have 4 fields at the last 4 rows: *created\_at,* *created\_by\_admin\_id, updated\_at* and *updated\_by\_admin\_id*. These are for administration purposes and for tracing the changes made to the tables.
* Note that in the figure below, the relationship between the admin and the other tables has been omitted in all but one. Specifically, the *created\_by\_admin\_id* and *updated\_by\_admin\_id* fields are defined in all tables with a foreign key constrain to the primary key of the admins table but the lines indicating that relationship has not been included.
* The primary key (PK) of the **admins**, **users**, **assets**, **smart\_meters** and **energy\_communities** are type Universal Unique Identifier (UUID). This for guaranteeing their uniqueness in all the databases.

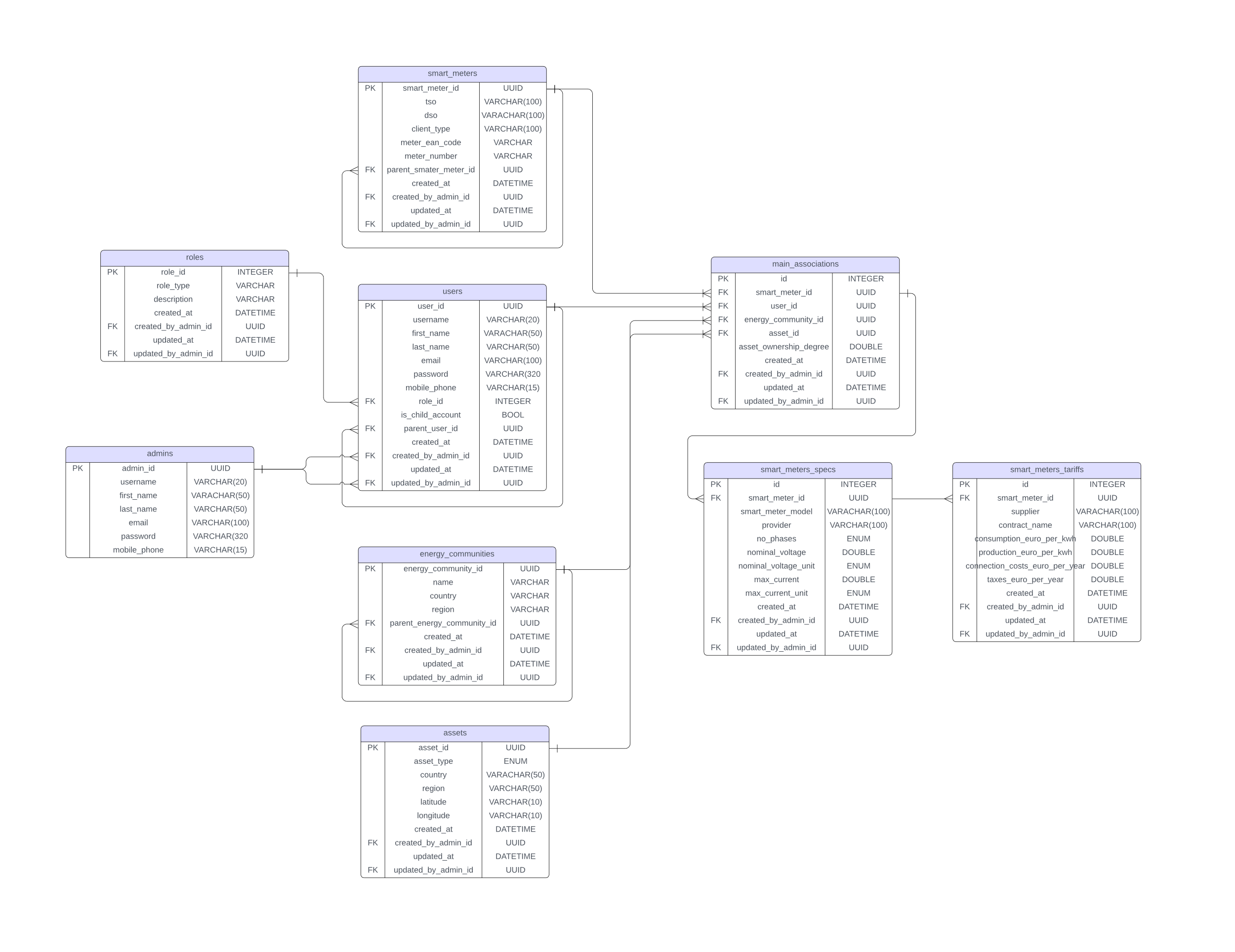


Figure 2: Current structure of user and asset database

## Tables Description

### Admins Table

The **admins** table in Figure 3, holds information about the administrators of the platform. Besides the type of *admin\_id* field that has been discussed before; the other fields are of varchar type with different lengths. All fields, but the *mobile\_phone,* do not permit NULL values, since these are considered, the minimum required information to characterize the admin user. The mobile phone is therefore optional in case a two-step login verification will be added.

Table

Description automatically generated with medium confidence

Figure 3: Admins table

### Roles Table

The roles table in Figure 4, is added for defining the different types of user accounts with different access rights to features and functionalities of the platform. The role\_id is an integer type since it is not necessary to ensure uniquness and it is expected to not exceed ten rows. Although the role\_type field is VARCHAR it will be changed in the future to an ENUM type. The description field is for proving a very brief summary of the different roles.

Thusfar, the roles are:

* Owner: This type of user is, as the name suggestes, the owner of assets. The owner has a wide access to the platform although this can be changed depening on the platforms bussiness model.
* Manager: This type of user is responsible for managing the assets both individually and collectively in the energy community(-ies). This account has the widest access to the platform.

Two additional roles that can be added are:

* Viewer (tentative title): This is a child account of the owner user that has limited access to the platform. Perhaps only to monitoring of the assets and reports.
* Operator: This is a child account of the manager user that has limited access to the platform. Perhaps to the control dashboards of the assets, control parameters high resolution monitoring etc.

A picture containing table

Description automatically generated

Figure 4: Roles table

### Users Table

Due to the dependence to the **admins** and **roles** table, the **users** table in Figure 5 ,has to be defined after. The first seven fields are self-explanatory, like in the **admins** table. The *role\_id* field has a foreign key constraint to the primary key of the **roles** table for indicating the type of the user. The *is\_child\_account* (Boolean) and the *parent\_user\_id* fields are used for allowing the registration of child user accounts (of type viewer or operator for example) and associated with the parent user (main account).

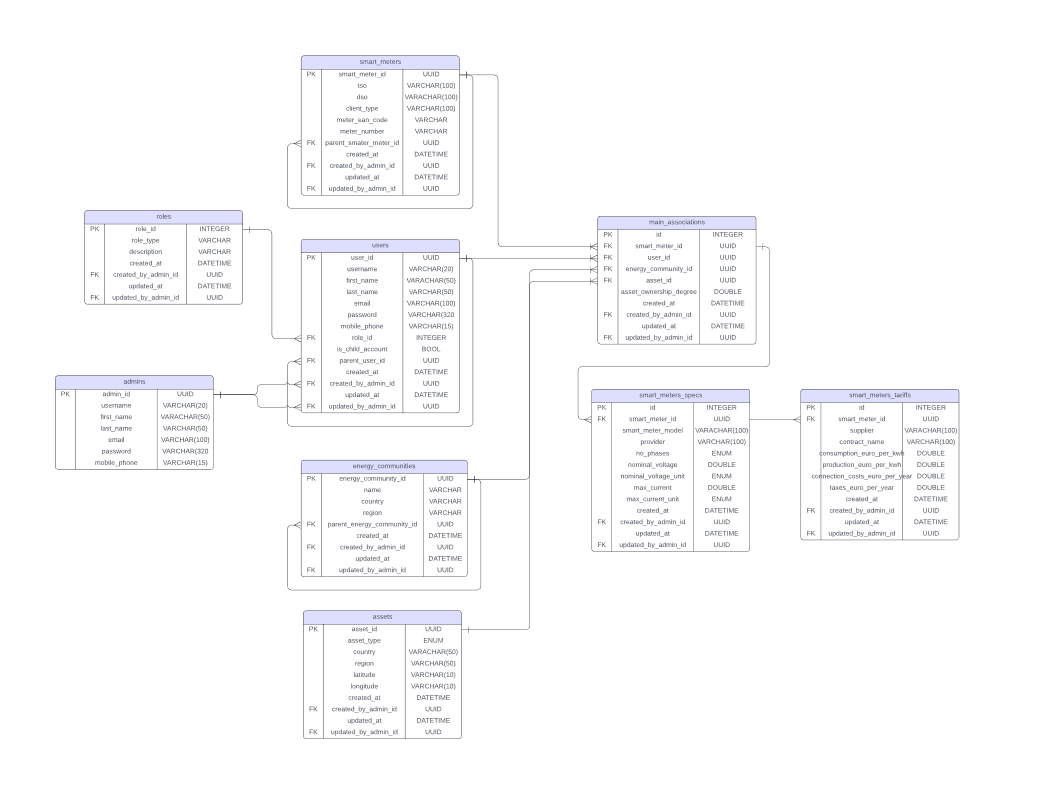


Figure 5: Users Table

### Smart Meters Table

The *tso* and *dso* fields hold information about the transmission system operator and system distribution operator of the network point that the smart meter is connected to. The *client\_type* field is for characterizing the type of energy resources behind the meter, for example, flexible load, flexible load and storage, load and electric vehicle to name a few. Now is VARCHAR type but in a future revision it will be changed to ENUM. The *meter\_ean\_code* and *meter\_number* are identification numbers usually assigned by the system operator’s especially in the case of residential meters. The same philosophy as in the users table is considered in this one for associating smart meters with each other. Specifically, there might be a possibility that some meters face the end-user and another the grid. This configuration can be indicated by having the end-user facing meters associated with the parent meter in the *parent\_smart\_meter\_id* field.

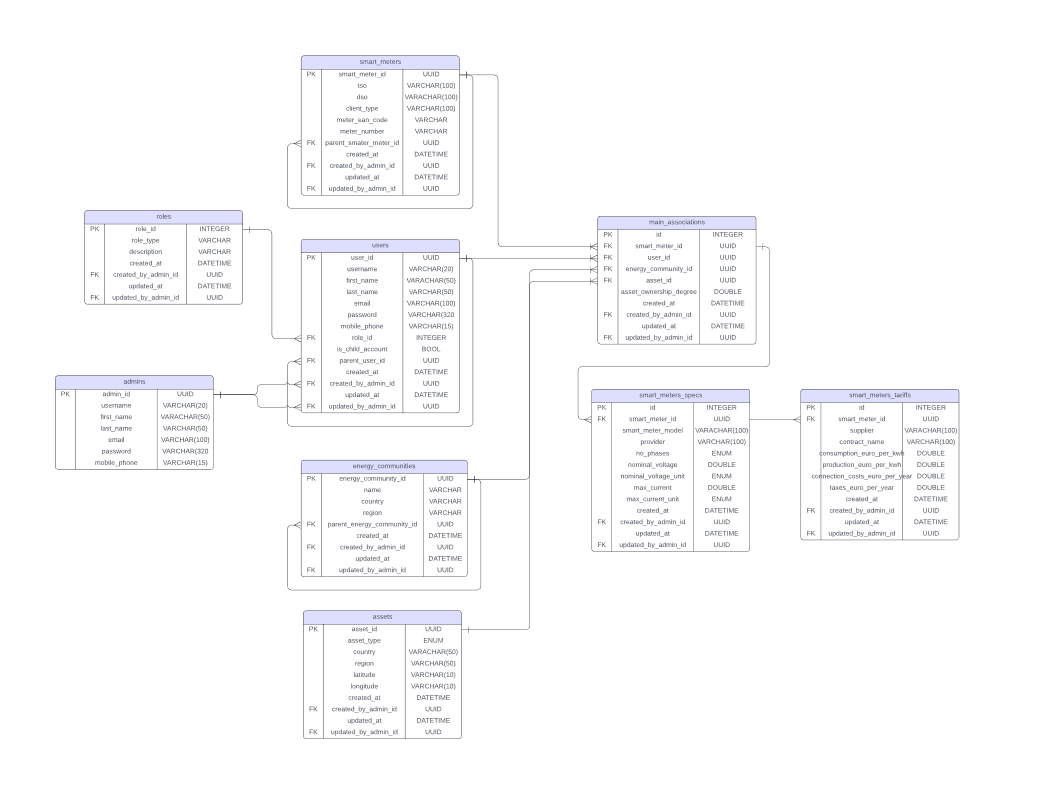


Figure 6: Smart meters table

### Assets Table

The assets table in Figure 7, hold information about the type and geographical location of the asset. The asset\_type field is of ENUM type and can take values of BESS, EV, LOAD, FLEXLOAD, PV, WPP, ELY. The above list can be revised in the future to add or remove values. The field is not nullable.

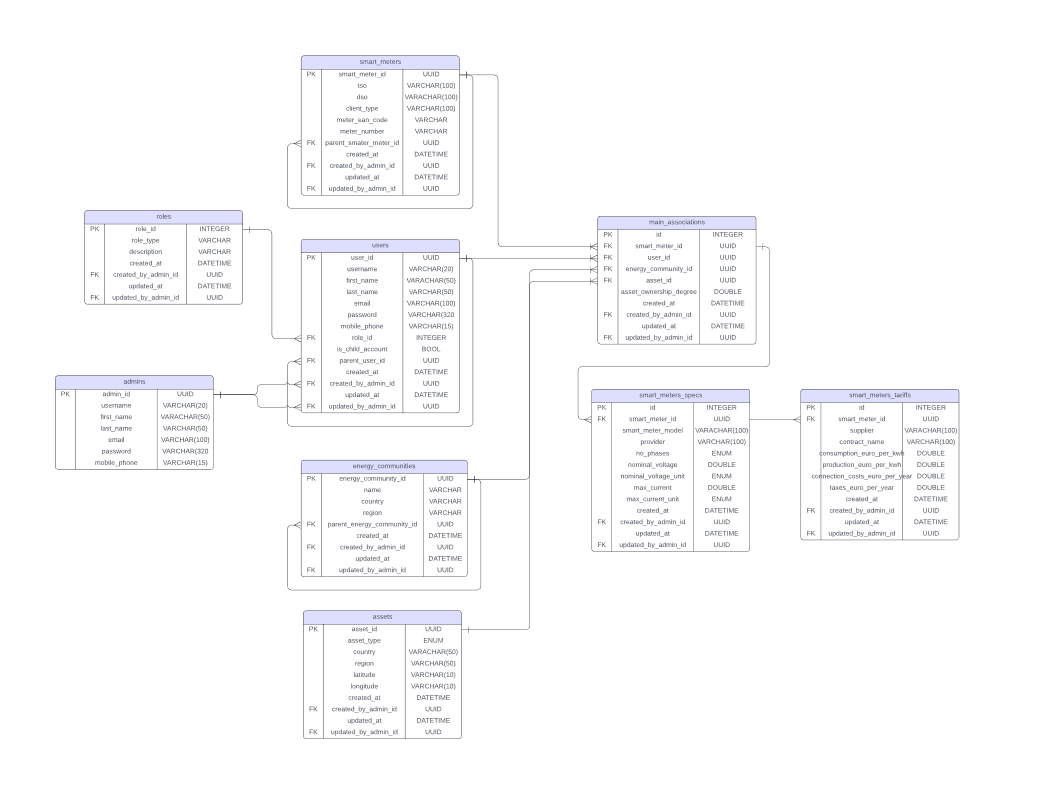


Figure 7: Assets Table

### Energy Communities Table

In general, energy communities are formed from assets that are geographically close to each other. The energy communities table registers the name of the energy community (it can be decided in the future if it will be a keyword or something else) and their location. The name field is not nullable however, the country and region fields are. Here as well it is possible to have nested energy communities indicated by the parent\_energy\_community\_id. If the energy community has not parent then that field is NULL.

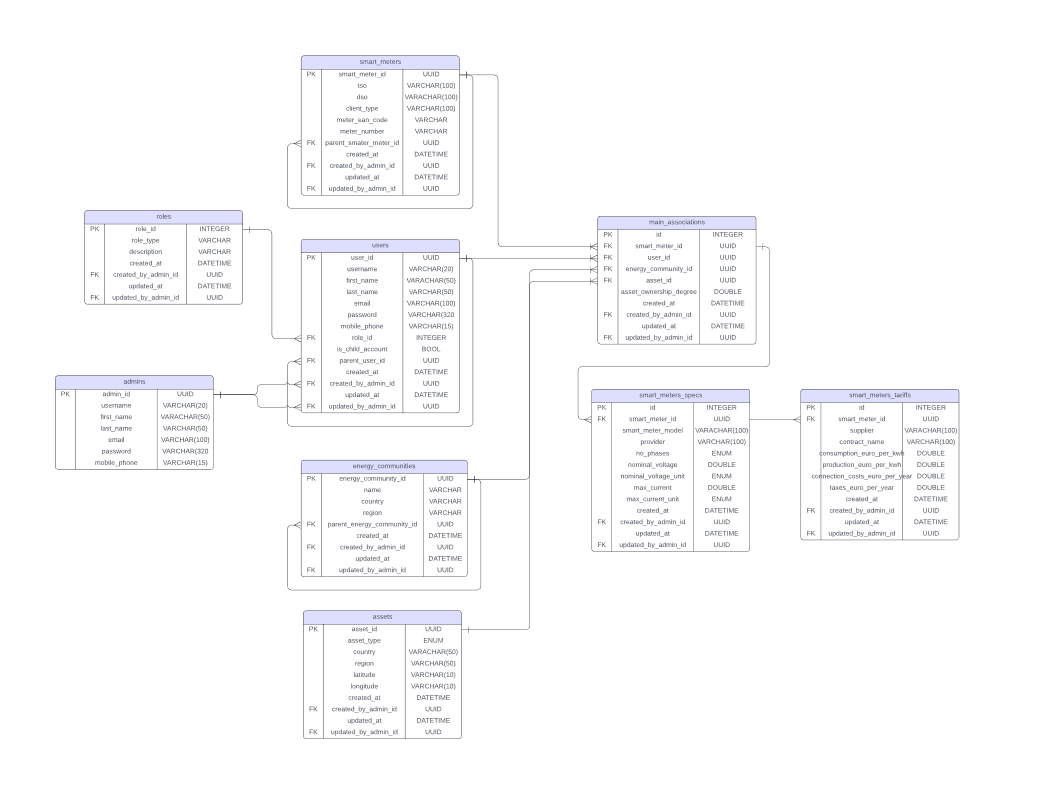


Figure 8: Energy Communities Table

### Main Associations Table

The most important aspect of the user\_and\_assets database is the association of the assets, users, smart meters, and energy communities. This is achieved by the main\_associations table that holds the mapping between the said components. Specifically, the primary keys of these tables are used as foreign key constratints for the respective fields. From the naming convention of each field it is cleary to which table is related to. An additional field called the asset\_ownership\_degree indicates the ownership by multiple users. The permissible values (check constraint) is between 0 and 1. For a single owner that value is 1 and for manager users this value is 0. Yet if the manager has some ownership of assets can have an ownership degree higher that 0.

Query example: Lets assume that we want to find which assets, smart meters and in which energy community the user is. Then with the simple query:

FROM main\_associations SELECT (smart\_meter\_id, energy\_community\_id, asset\_id)

WHERE user\_id = < the id of the user we are interested >;

We can get all the information that we need. If we additionally include join statements in out query we can get not only the IDs but the information stored in the assets, energy\_communites and smart\_meters tables.

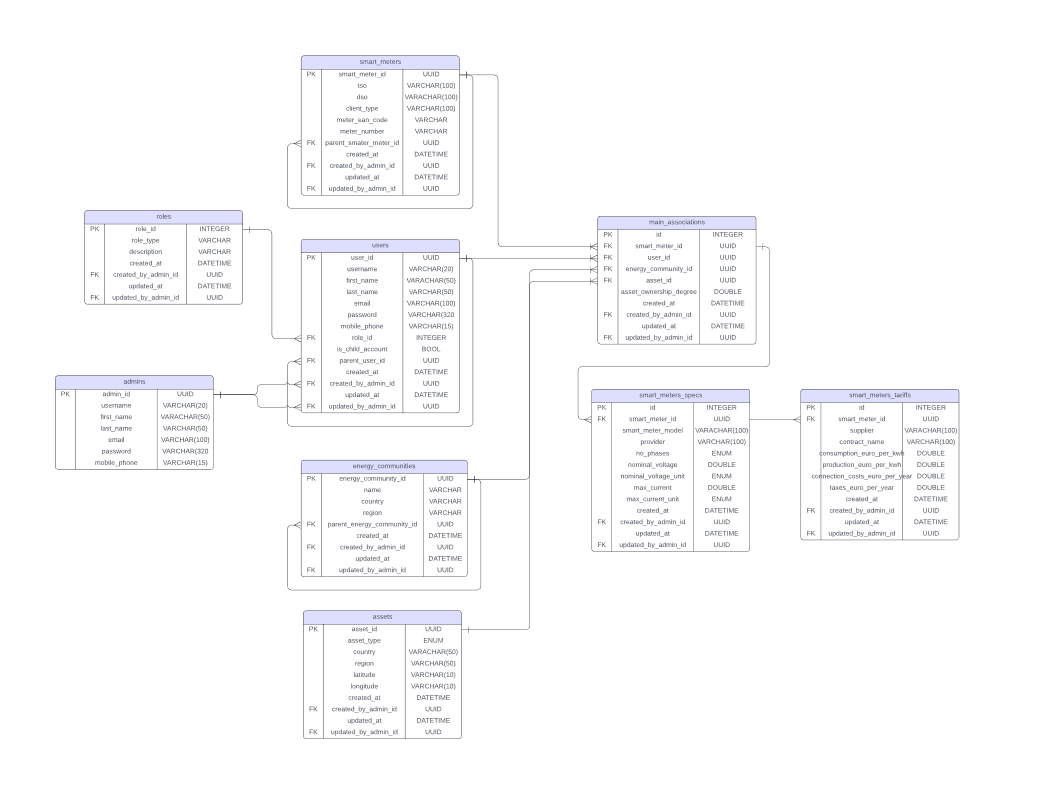


Figure 9: Main Associations Table

### Smart Meter Specs Table

This table is created to hold information about the specifications of the smart meters. For example, what is the model, the provider, and some other information (tentative fields).

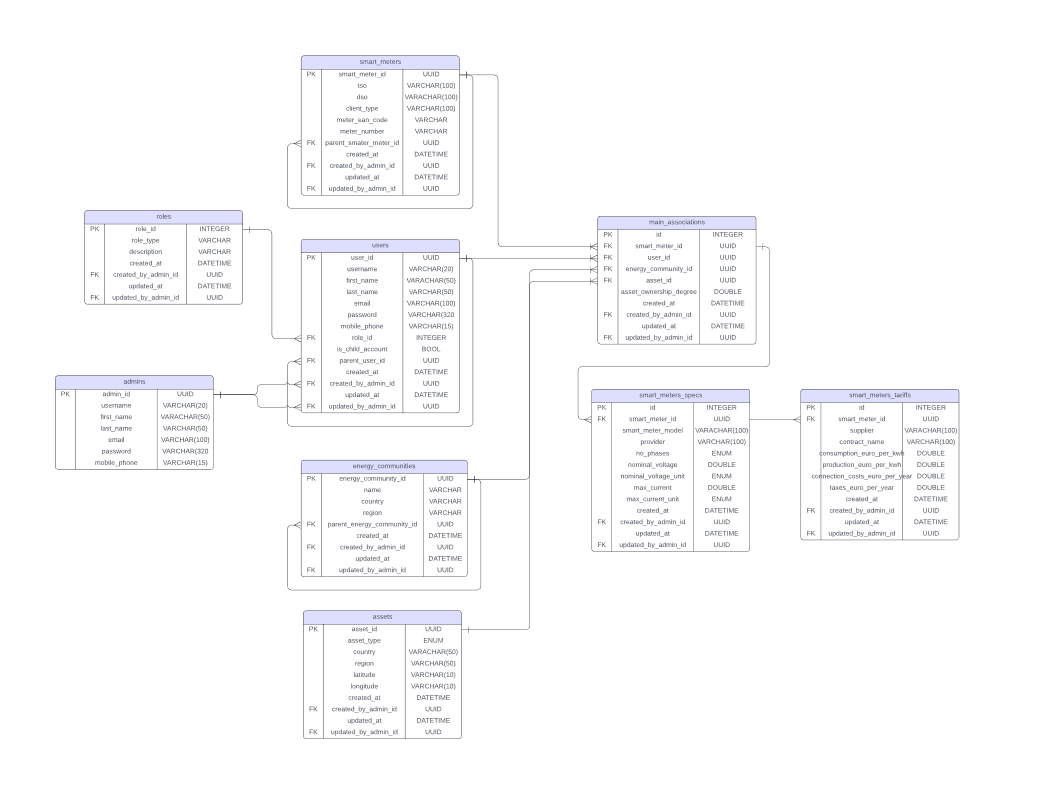


Figure 10: Smart Meter Specs Table

### Smart Meter Tariffs Table

The smart meter tariffs table is responsible for storing information about the electricity charges for injecting or absorbing power, as well as some fixed costs like transmission & distribution costs or taxes. The structure of the table is inspired by a typical breakdown of the electrcicity charges for a residential end-user. However, this can be adjusted to accommodate other type of charging schemes. Perhaps, this table might be more suitable for the rt\_operation database but it has not been finalized yet.

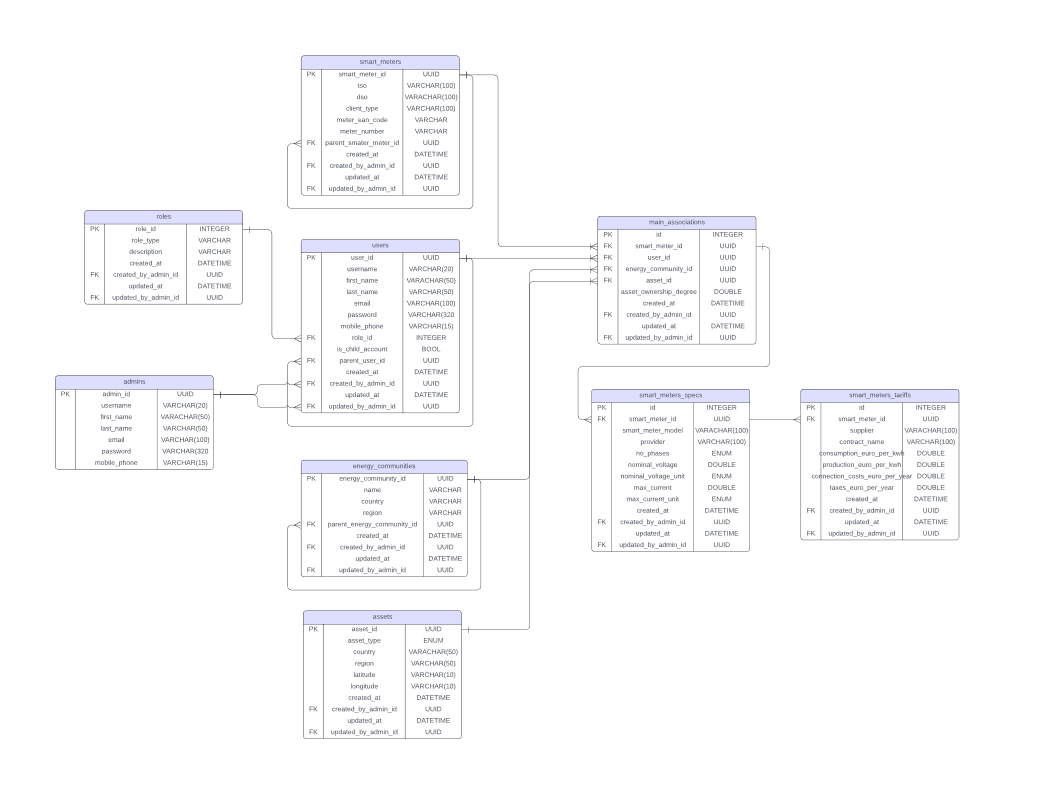


Figure 11: Smart Meter Tariffs Table

# Future Work

There are more tables that will be defined in the future. First of all, a table with the specifications of each asset, this information is related to the physical properties of the hardware equipment and cannot be modified. For example, the characteristics of a PV panel (dimensions, nominal power, etc.) that constitute the PV plant. The exact information that this table should have is not yet defined.

Additionally, the owner should be able to impose his/her preferences on the usage of their assets. This information will be stored in the owner\_preferences, but since this information is not yet clear the table will be in a future implementation.

Fixed term power contracts among assets or users are allowed in some countries. This needs to be clarified for the case of Luxembourg. One possibility is to structure the table as follows:

Table

Description automatically generated

The from\_smart\_meter\_id and to\_smart\_meter\_id indicate the involved parties of the contract. The decision to relate the power contracts to the smart meter was based on our previous decision to concentrate the tariffs and charges to smart meters. The table includes information of the price and volume of the contract and the interval that is valid for. The status field can indicate if the contract is active or expired.

It is expected that the platform will interact with an energy market. This is still premature so development for a future implementation but the relevant tasks is to comply with the documentation of the market API for example NordPOOL [API](https://marketdata.nordpoolgroup.com/) in order to store username password and authentications for accessing their data.

Finally, it has been suggested that it is common for energy suppliers to offer a virtual storage option to end-users. This is still not clear but the once defined a new table called virtual storage can be created with a relationship to the smart\_meters. Additional measurement devices can be registered as behind the meter sensors since we assume that it will be difficult to install such kind of equipment in the grid without the authorisation of the system operator. Nevertheless, if such possibility arise it is possible to register measurement units as a separate component in the database.