



---

# DATA MODELLING

---



Advisor's name: Mohammad Al Hajj

Course Name: Software Engineering

Course Code: CSCI317

Signature:

Instructor Notes:

Date of approval:

MAY 1, 2022

DONE BY ALI HAMED

1220024

# Table of Contents

<b>Over view:</b>	<b>4</b>
<b>Goal:</b>	<b>4</b>
<b>Introduction:</b>	<b>4</b>
<b>What is Data modeling?</b>	<b>4</b>
<b>What is ER diagram?</b>	<b>4</b>
The components and features of an ER diagram:	5
Entity:	5
Associative entity:	5
Cardinality:	5
Primary Key:	6
Foreign Key:	6
Attribute:	6
Data type:	6
<b>Explanation of the tables and what they include:</b>	<b>7</b>
<b>Logical Model Development Stages:</b>	<b>11</b>
<b>An important point to clarify:</b>	<b>11</b>
<b>Context Data model:</b>	<b>11</b>
Part one:	11
Part two:	13
Complete ER Diagram:	14
<b>Key-base data model:</b>	<b>15</b>
Part one:	15
Part two:	16
Part three:	16
Complete ER Diagram:	17
<b>Fully attributed data model:</b>	<b>18</b>
Part one:	18
Part two:	19
Part three:	20
Complete ER Diagram:	21
<b>Normalized data model:</b>	<b>22</b>

First normal form: .....	22
Second normal form: .....	23
Third normal form: .....	24
<b>Conclusion: .....</b>	<b>25</b>
<b>References .....</b>	<b>25</b>

## List of Tables

Table 1 <b>Account permission</b> .....	7
Table 2 <b>Account Rule</b> .....	7
Table 3 <b>Municipality Account</b> .....	7
Table 4 <b>User Account</b> .....	8
Table 5 <b>Dumps</b> .....	8
Table 6 <b>Online Store</b> .....	8
Table 7 <b>Physical Store</b> .....	9
Table 8 <b>Cart</b> .....	9
Table 9 <b>Product Category</b> .....	9
Table 10 <b>Product</b> .....	10
Table 11 <b>Profile</b> .....	10
Table 12 <b>Post</b> .....	10
Table 13 <b>Report</b> .....	10

## Over view:

The problem of waste accumulation is one of the biggest problems that any society may face. Therefore, in order to find an effective mechanism to solve this problem or even contribute to a large extent in solving it, **Clean&Green** application is the perfect project for that.

## Goal:

The idea and purpose of this page is organizing and documenting a system's data using ER diagram as a data modeling.

## Introduction:

### What is Data modeling?

Discover how data modeling uses abstraction to represent and better understand the nature of data flow within an enterprise information system. (IBM, n.d.)

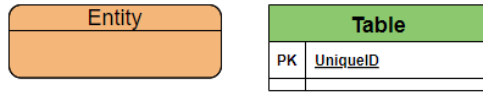
Like any design process, database and information system design begins at a high level of abstraction and becomes increasingly more concrete and specific. Data models can generally be divided into three categories, which vary according to their degree of abstraction. The process will start with a conceptual model, progress to a logical model and conclude with a physical model.

### What is ER diagram?

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs. (lucidchart, n.d.)

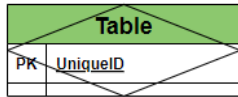
The components and features of an ER diagram:

*Entity:*



A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.

*Associative entity:*



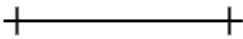
An associative entity associates entities (or elements) within an entity set.

*Cardinality:*

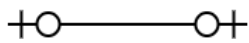
Defines the numerical attributes of the relationship between two entities or entity sets. The three main cardinal relationships are one-to-one, one-to-many, and many-many.

What is used in this graph is:

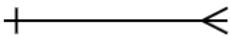
**One-to-One**



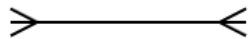
**One or Zero**



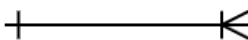
**One-to-Many**



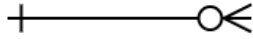
**Many-to-Many**



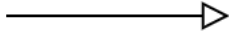
**One-to-One or Many**



**One-to-Zero or one or Many**



### Generalization



### Primary Key:

Primary Key is a set of attributes (or attribute) which uniquely identify the tuples in relation or table. The primary key is a minimal super key, so **there is one and only one primary key in any relationship**. It will be denoted in the table as **(PK)**.

### Foreign Key:

A foreign key is a set of attributes in a table that refers to the primary key of another table. The foreign key links these two tables. It will be denoted in the table as **(FK)**.

### Attribute:

A descriptive property or characteristic of an entity. Synonyms include element, property, and field.

### Example:

Student entity have {First name, Last name, Address, .....}

### Data type:

A property of an attribute that identifies what type of data can be stored in that attribute.

### Example:

Number → Integer

Text → Varchar

## Explanation of the tables and what they include:

### Account permission:

These are the permissions allowed for each account. It constitutes an essential separation point between the user account and the admin account.

*Table 1 Account permission*

Attribute name	Data type	Required	Description
Permission ID	int	yes	To give a number that distinguishes the permissions
Permission Name	varchar	yes	to add name for each permission
Details	varchar	yes	To give some explanation for each permission

### Account Rule:

These are the terms and conditions that the user must abide by and observe during his use of this application.

*Table 2 Account Rule*

Attribute name	Data type	Required	Description
Account rule ID	int	yes	To give a number that distinguishes the Rules
Rule Name	varchar	yes	to add name for each Rule
Details	varchar	yes	To give some explanation for each Rule

### Municipality Account:

It is an account belonging to a municipality and not to a person. It is an official account as it indicates a government department concerned with helping citizens living within the scope of this municipality.

*Table 3 Municipality Account*

Attribute name	Data type	Required	Description
Municipality account ID	int	yes	To give a number that distinguishes the Municipalities
Municipality Name	varchar	yes	to add name for each Municipality
Email	varchar	yes	To add an email to each account
Password	varchar	yes	To add a password to protect the account
Address	varchar	yes	To find out the address of the municipality
Phone Number	int	no	To add a phone number to contact the municipality
Credit card	varchar	no	Add a payment method and receive money

### User Account:

In order for the general people to create an account and benefit from the application.

Table 4 User Account

Attribute name	Data type	Required	Description
User account ID	int	yes	To give a number that distinguishes the Users
User Name	varchar	yes	to add name for each User
Email	varchar	yes	To add an email to each account
Password	varchar	yes	To add a password to protect the account
Address	varchar	no	To find out the address of the User
Phone Number	int	no	To add a phone number to contact the User
Credit card	varchar	no	Add a payment method and receive money

### Dumps:

To know the number and location of garbage containers. And to take advantage of this information to display on the application to help people more.

Table 5 Dumps

Attribute name	Data type	Required	Description
Dumps ID	int	yes	To give a number that distinguishes the Dumps
Dumps Name	varchar	no	to add name for each Dump
Address	varchar	yes	To find out the address of the Dumps
Number of garbage container	int	no	To know or estimate the volume of the landfill

### Online Store:

To add a feature and the ability for all users of all levels to sell items online without the need for a fixed place such as a shop.

Table 6 Online Store

Attribute name	Data type	Required	Description
Online Store ID	int	yes	To give a number that distinguishes the online stores
Store phone number	int	yes	To add phone number to the store
Address	varchar	no	to add address if there
Delivery method	int	yes	to add the way of delivery



### Physical Store:

To allow people who have fixed stores and want to sell materials through these stores and do not prefer selling via the Internet.

Table 7 *Physical Store*

Attribute name	Data type	Required	Description
Physical Store ID	int	yes	To give a number that distinguishes the physical stores
Store phone number	int	no	To add phone number to the store
Address	varchar	yes	to add address if there

### Cart:

It is the place where information about all items that have been purchased or reserved to be purchased at a later time are stored.

Table 8 *Cart*

Attribute name	Data type	Required	Description
Cart ID	int	yes	To give a number that distinguishes the physical stores
Total items price	int	yes	To add the price group of the reserved items
Reserve Date	varchar	yes	To save the date of reserve
Purchase Date	varchar	yes	To save the date of purchase

### Product Category:

To determine the types of materials that can be found in stores.

Table 9 *Product Category*

Attribute name	Data type	Required	Description
Product Category ID	int	yes	To give a number that distinguishes the Product Category
Product Category Name	varchar	yes	to add name for each Product Category
Material	varchar	yes	To set the material type of the product

**Product:**

To identify and distinguish the materials inside each store and add other details to it.

Table 10 *Product*

Attribute name	Data type	Required	Description
Product ID	int	yes	To give a number that distinguishes the Product
Product Name	varchar	yes	to add name for each Product
Price	int	yes	To set a price for the product
Quantity	int	yes	To know the number of pieces

**Profile:**

To add additional personal details about users.

Table 11 *Profile*

Attribute name	Data type	Required	Description
Profile ID	int	yes	To give a number that distinguishes the Profiles
Scientific Specialization	varchar	no	To set a scientific Specialization for the user
work in	varchar	no	To add where the user is working

**Post:**

To get information about posts that users add.

Table 12 *Post*

Attribute name	Data type	Required	Description
Post ID	int	yes	To give a number that distinguishes the Posts
Title	varchar	no	To get the title of the post
Discerption	varchar	no	To add some details to the post

**Report:**

To take information about the report in case there is any problem with the publications.

Table 13 *Report*

Attribute name	Data type	Required	Description
Report ID	int	yes	To give a number that distinguishes the Reports
Message	varchar	yes	To add some details to the report

## Logical Model Development Stages:

An important point to clarify:

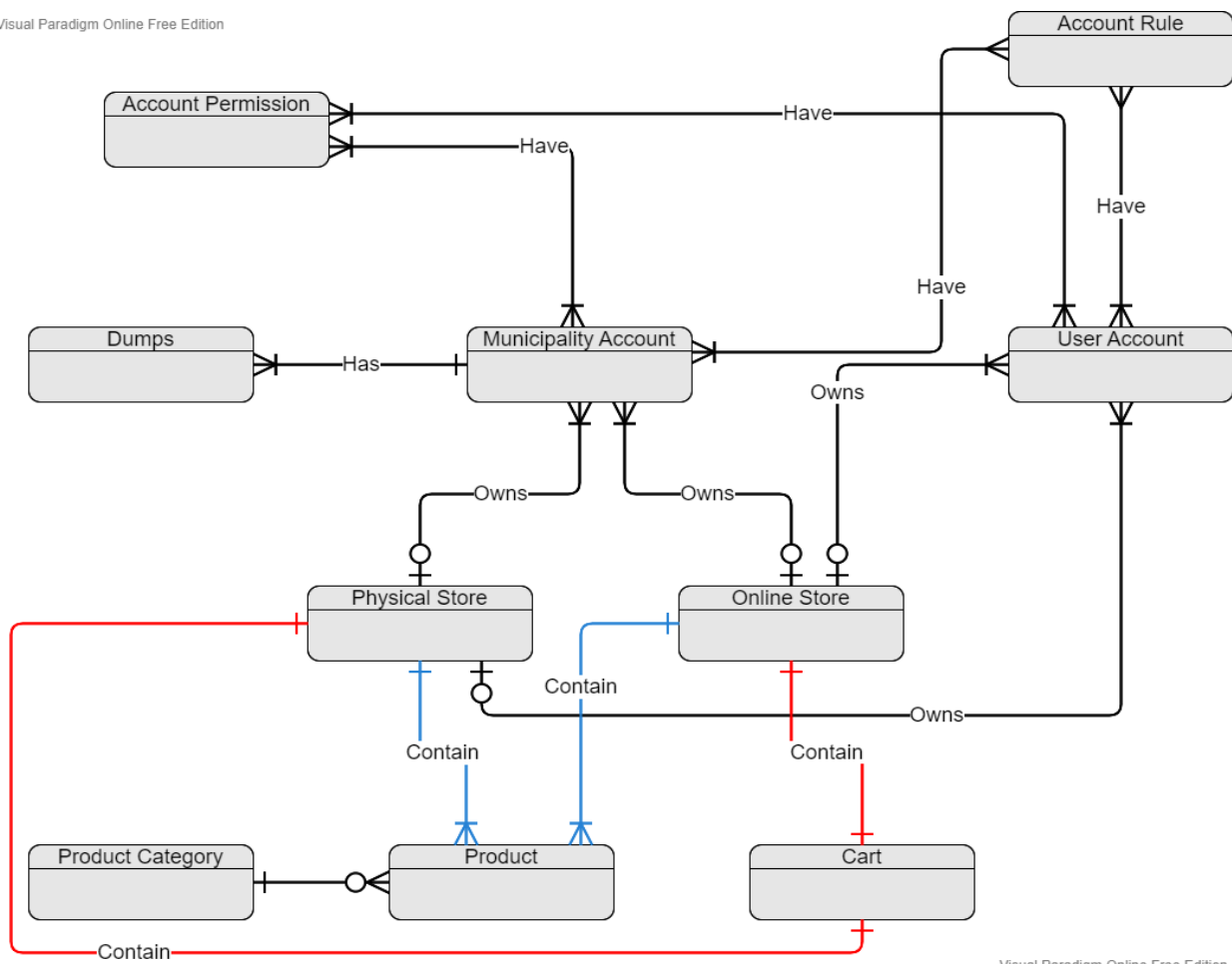
in each section, ER diagrams are segmented into smaller pieces for ease of reading, and at the end of each section the all the pieces are grouped into one final form to show the full scope of the project.

### Context Data model:

A context model (or context modeling) defines how context data are structured and maintained (It plays a key role in supporting efficient context management). It aims to produce a formal or semi-formal description of the context information that is present in a It is used to represent the reusable context information of the components (The top-level classes consist of Operating system, component container, hardware requirement and Software requirement). (wikipedia, n.d.)

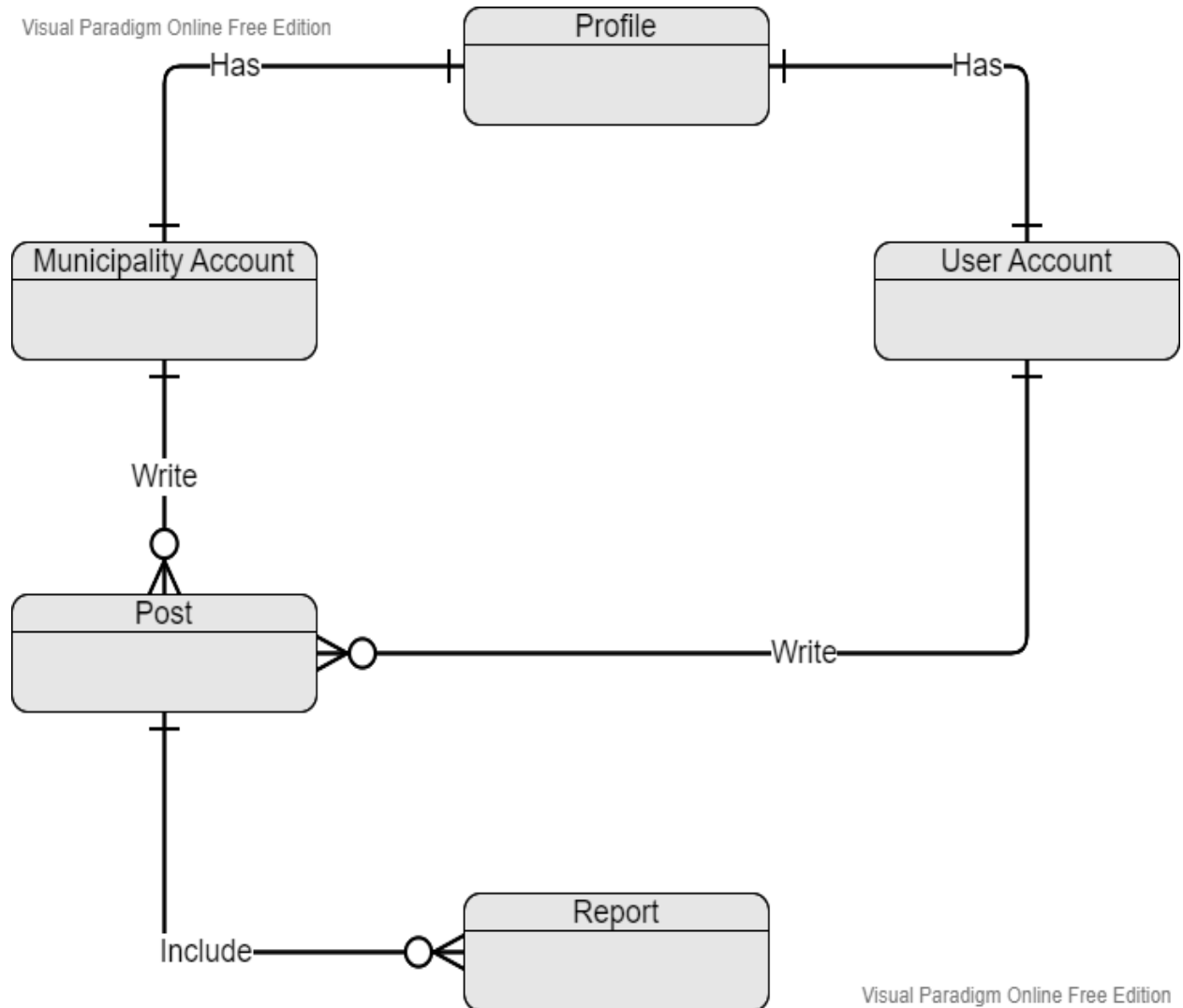
Part one:

Visual Paradigm Online Free Edition



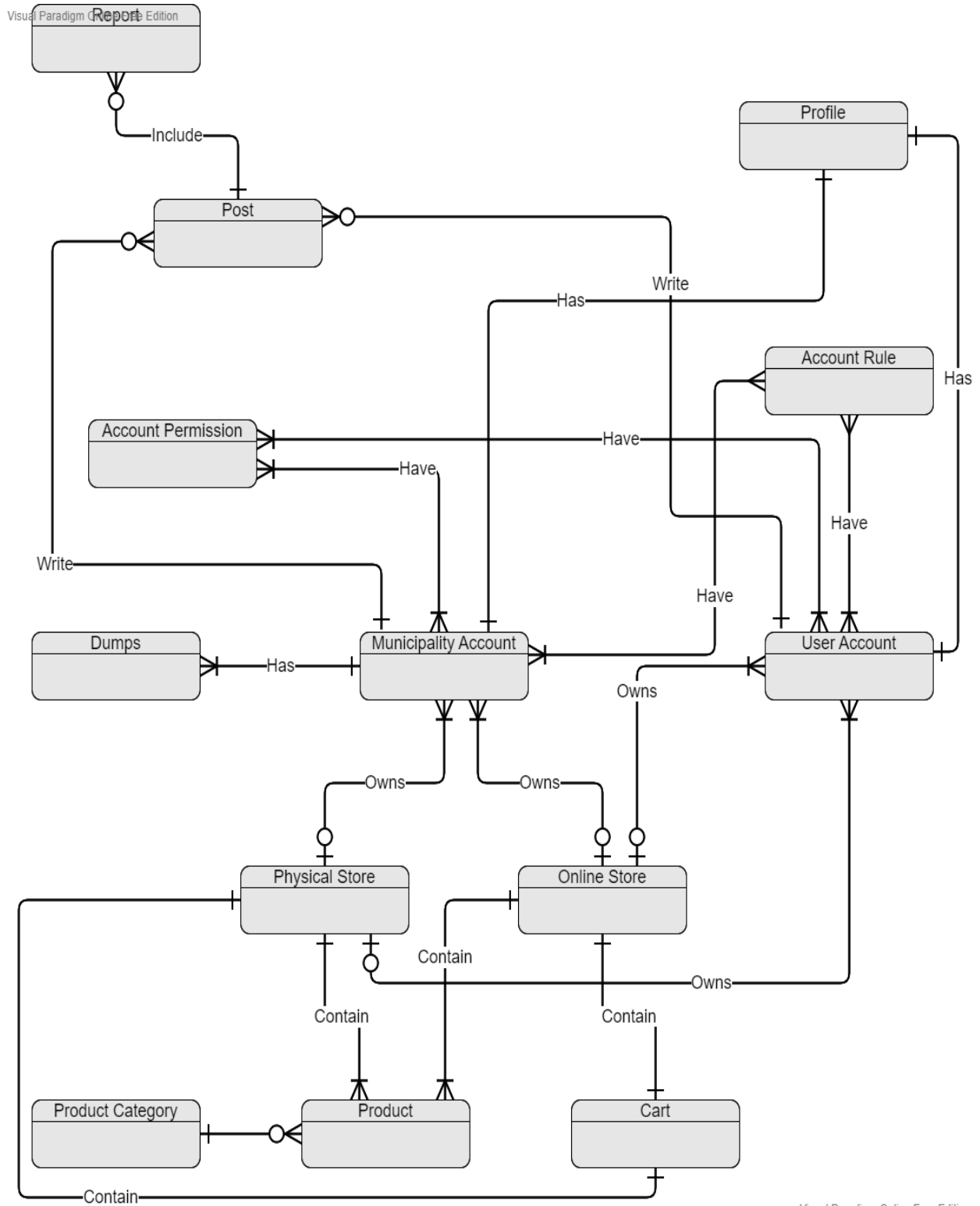
- **The relation between account permission and user account:**  
User account can **have** one or many permissions  
(The same applies to the municipality's account).
- **The relation between account rule and user account:**  
The user account **have** many rules that must be observed.  
(The same applies to the municipality's account).
- **The relation between the municipality account and the dumps:**  
The municipality **has** one or more landfills and it manage them.
- **The relation between the user account and the online store:**  
The user account **owns** one online store or does not own any online store.  
(The same applies to the municipality's account).
- **The relation between the user account and the physical store:**  
The user account **owns** one physical store or does not own any physical store.  
(The same applies to the municipality's account).
- **The relation between the online store and the cart:**  
Each online store **contains** a special shopping cart.  
(The same applies to the physical store).
- **The relation between the product category and the product:**  
Each product is **within** one of the product categories.
- **The relation between the online store and the product:**  
Each online store **contains** its own products.  
(The same applies to the physical store).

Part two:



- **The relation between the user account and the profile:**  
Each user account **has** its own profile.  
(The same applies to the municipality account).
- **The relation between the user account and the post:**  
Each user account can **write** posts if they want to do that.  
(The same applies to the municipality account).
- **The relation between the post and report:**  
Each post can **include** a report or more in case that one or more users do so.

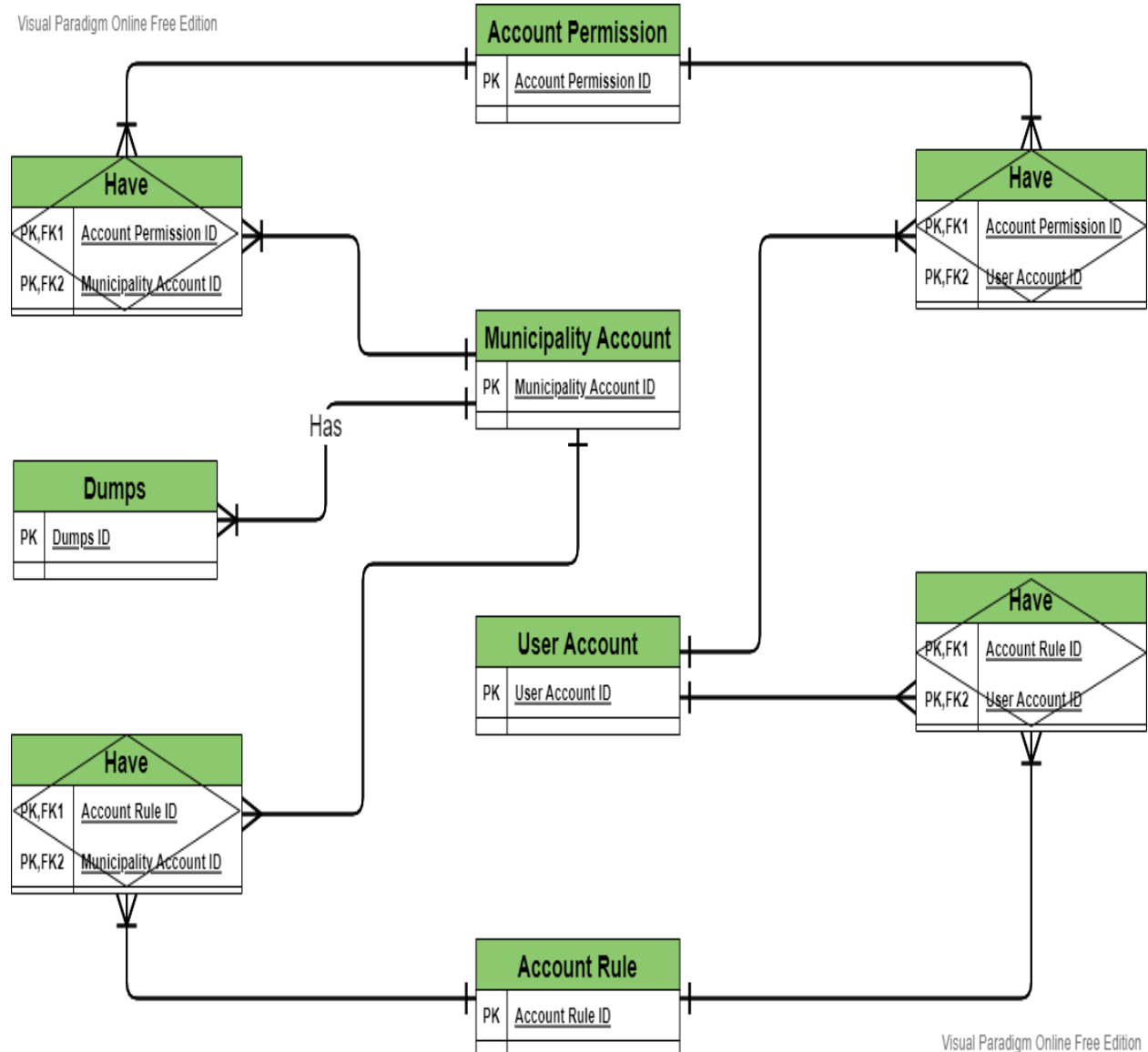
# Complete ER Diagram:



### Key-base data model:

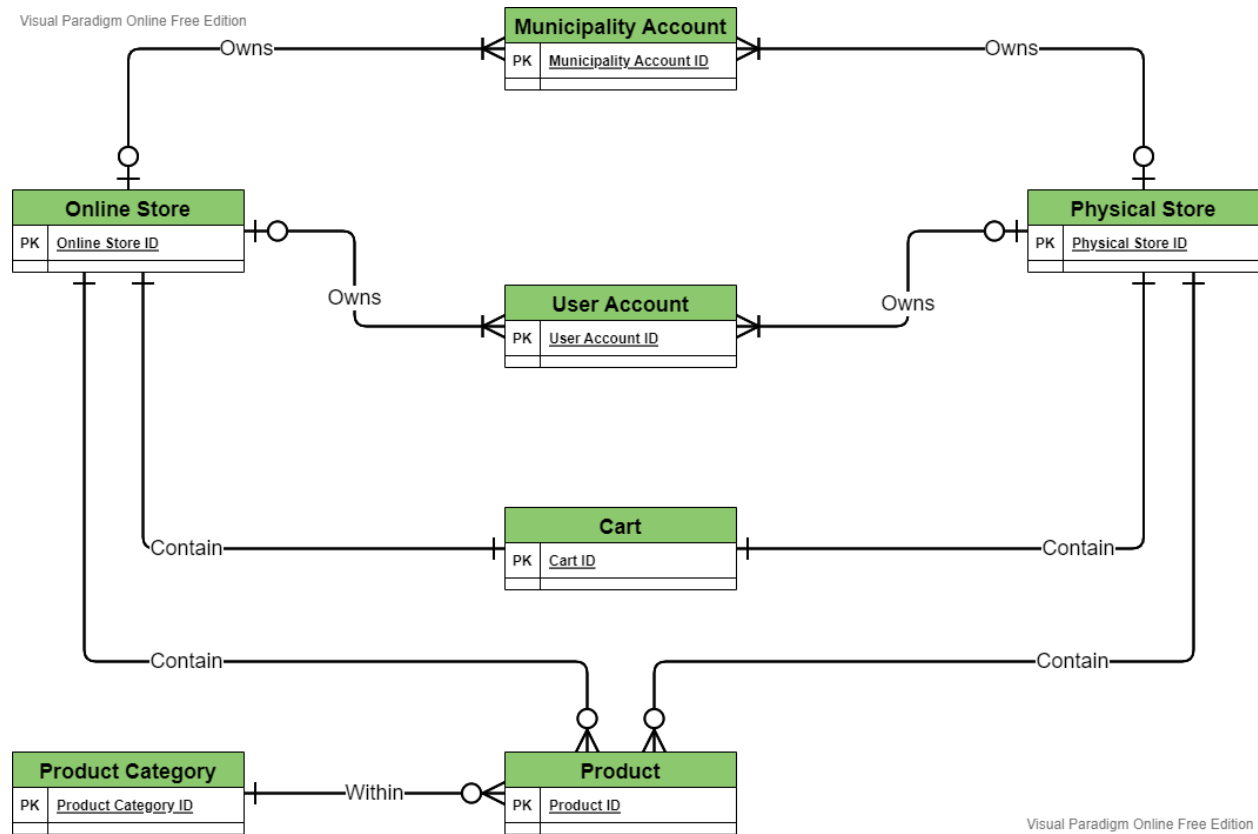
A data model (or data model) is an abstract model that organizes elements of data and standardizes how they relate to one another and to the properties of real-world entities. For instance, a data model may specify that the data element representing a car be composed of a number of other elements which, in turn, represent the color and size of the car and define its owner. (base, n.d.)

### Part one:



## Part two:

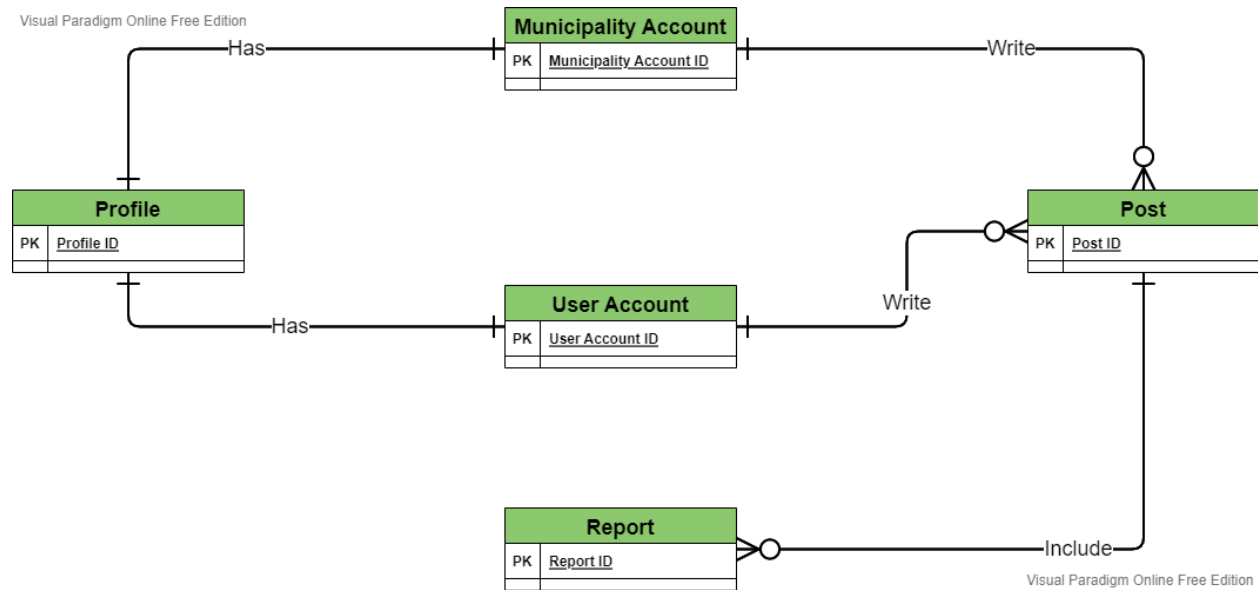
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

## Part three:

Visual Paradigm Online Free Edition

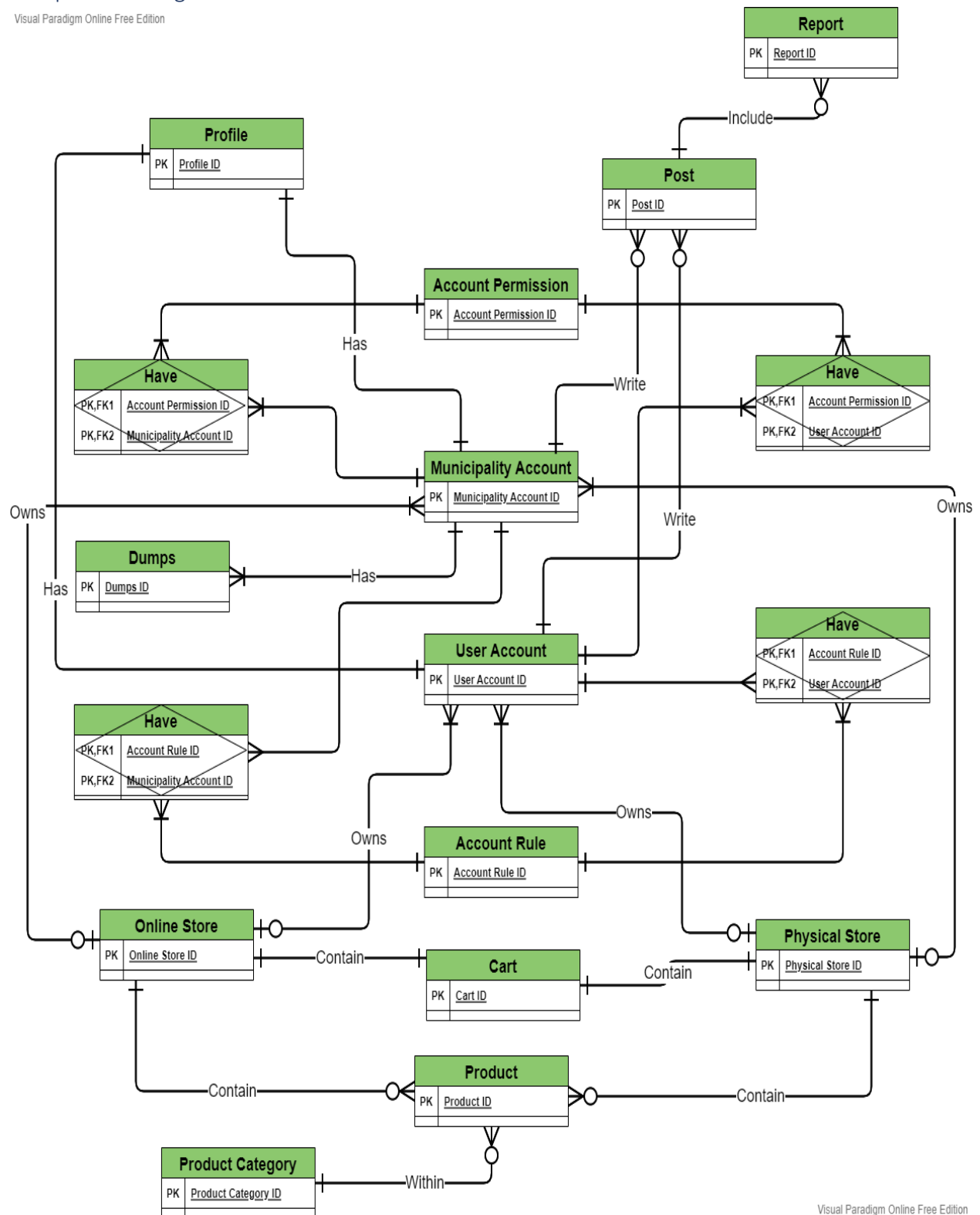


Visual Paradigm Online Free Edition



## Complete ER Diagram:

Visual Paradigm Online Free Edition



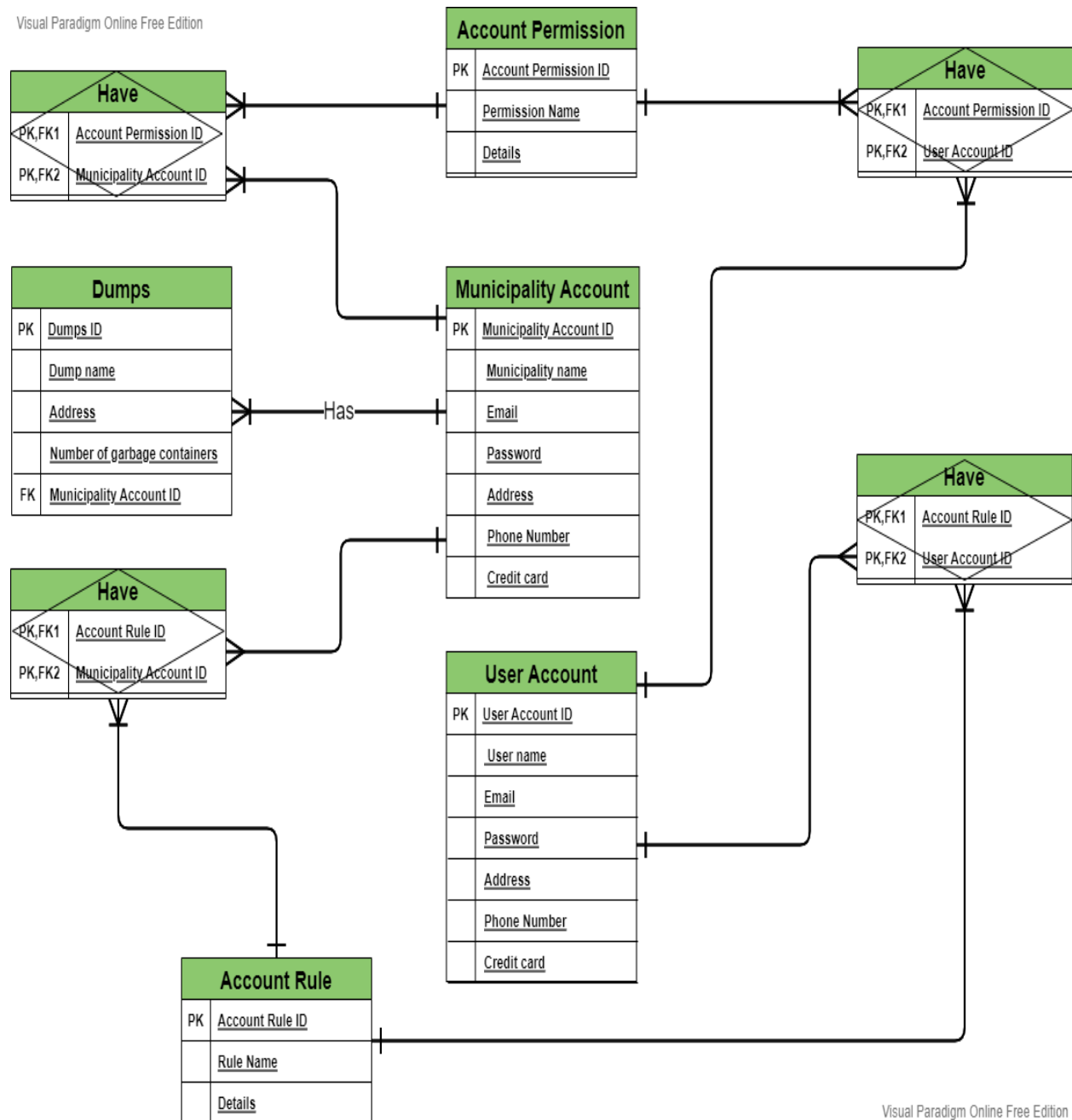
Visual Paradigm Online Free Edition

## Fully attributed data model:

A fully attributed data model contains detailed attributes (descriptions) for every entity within it. The term "database design" can describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. (attributed, n.d.)

### Part one:

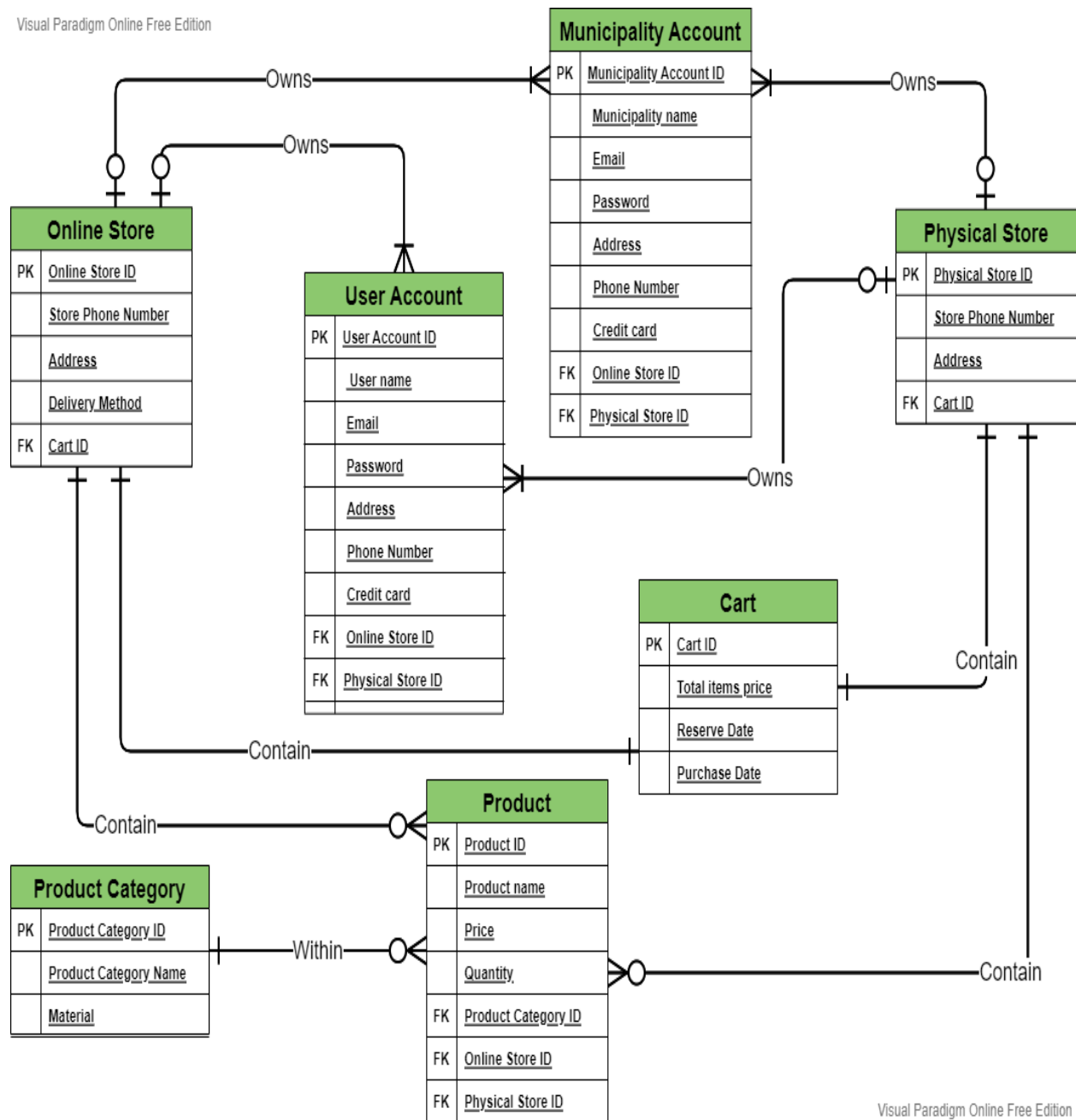
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

## Part two:

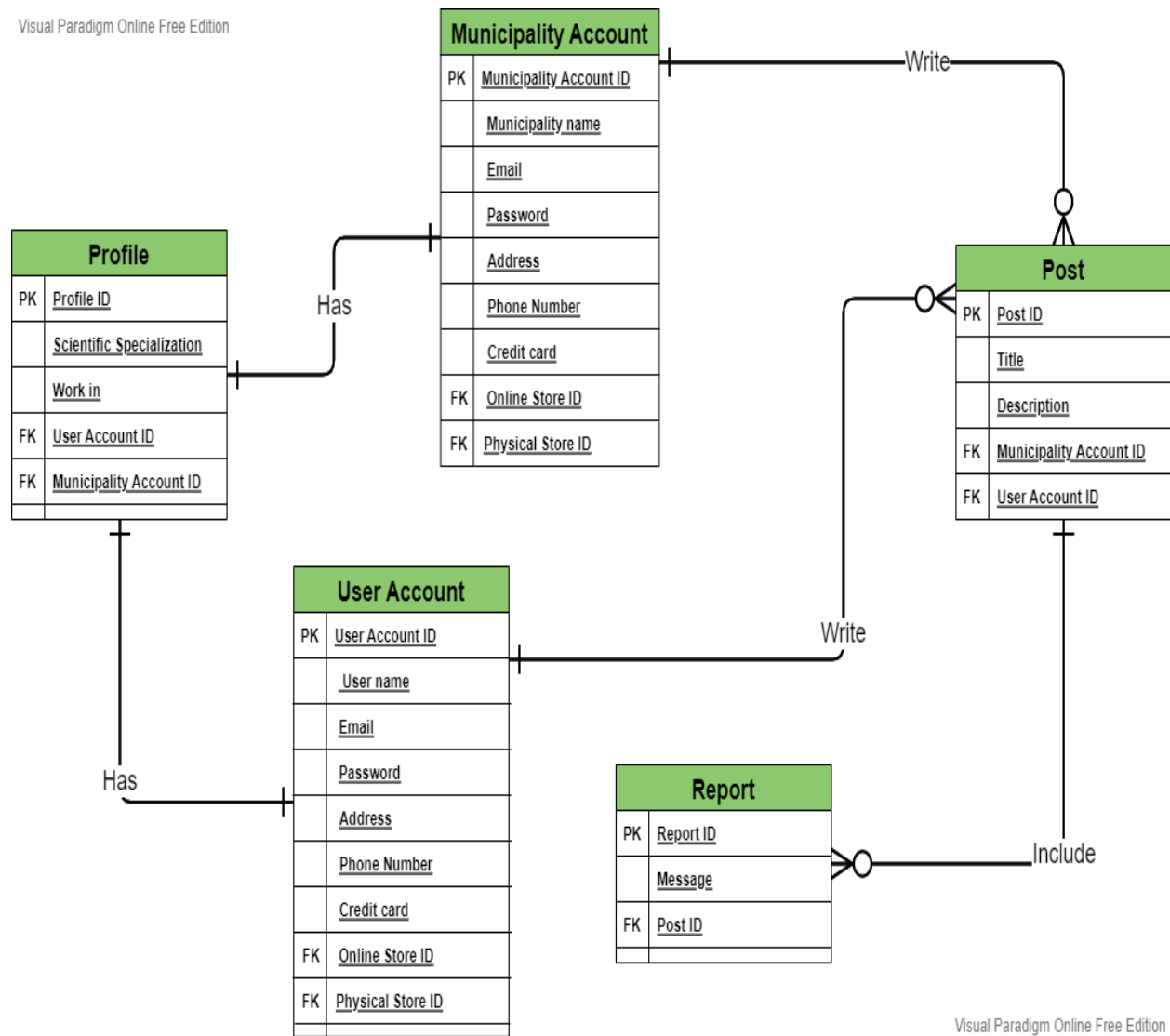
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

### Part three:

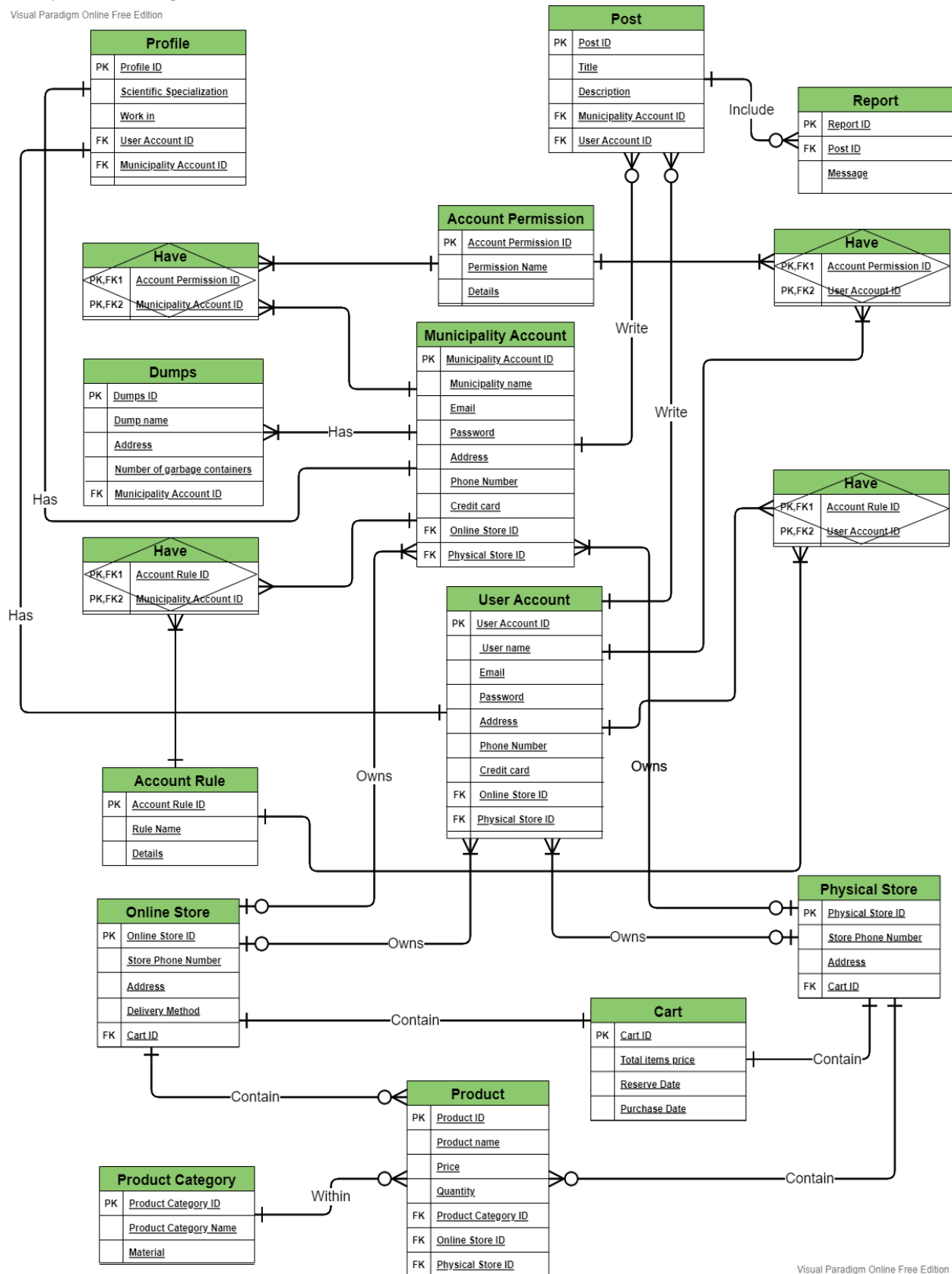
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

## Complete ER Diagram:

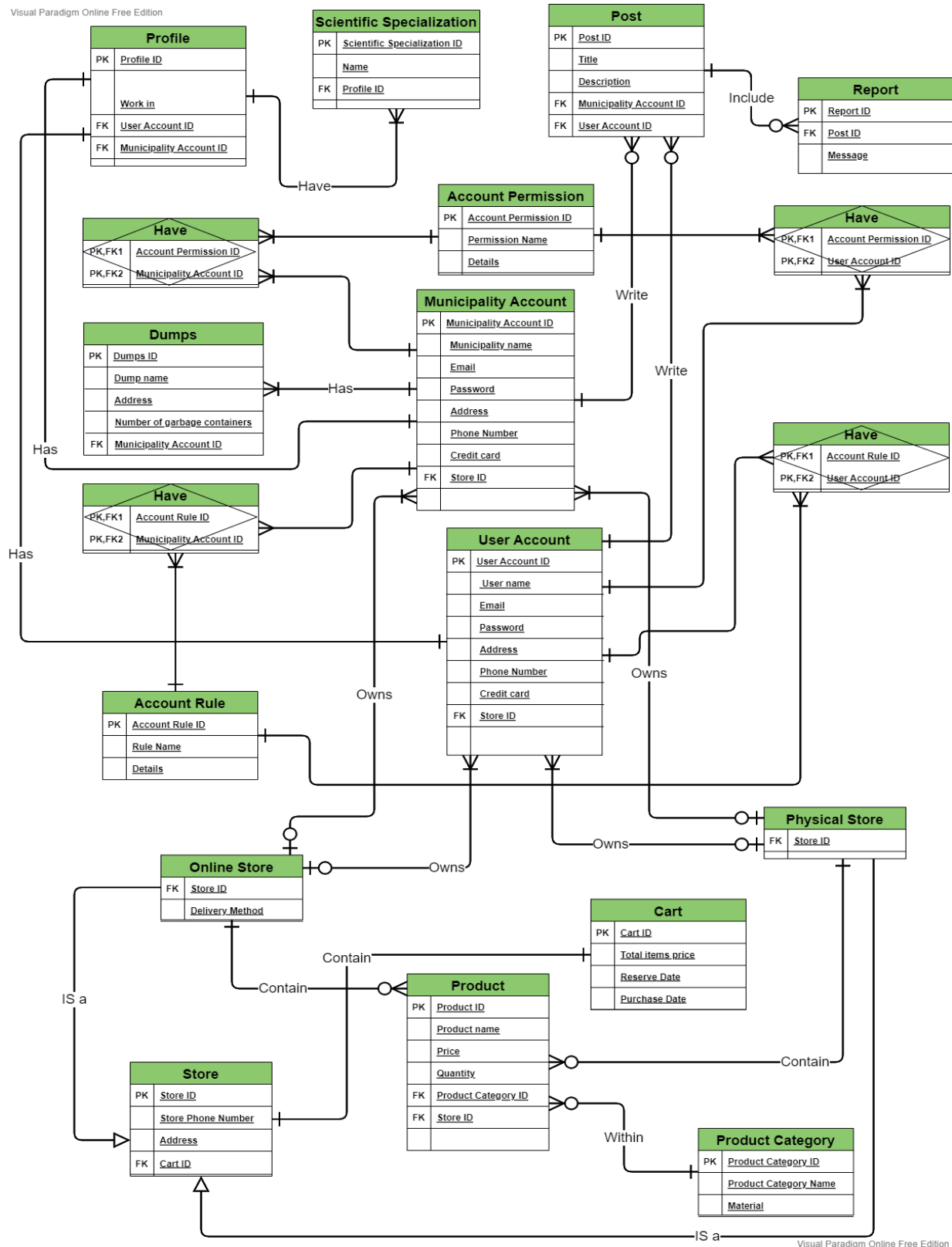
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

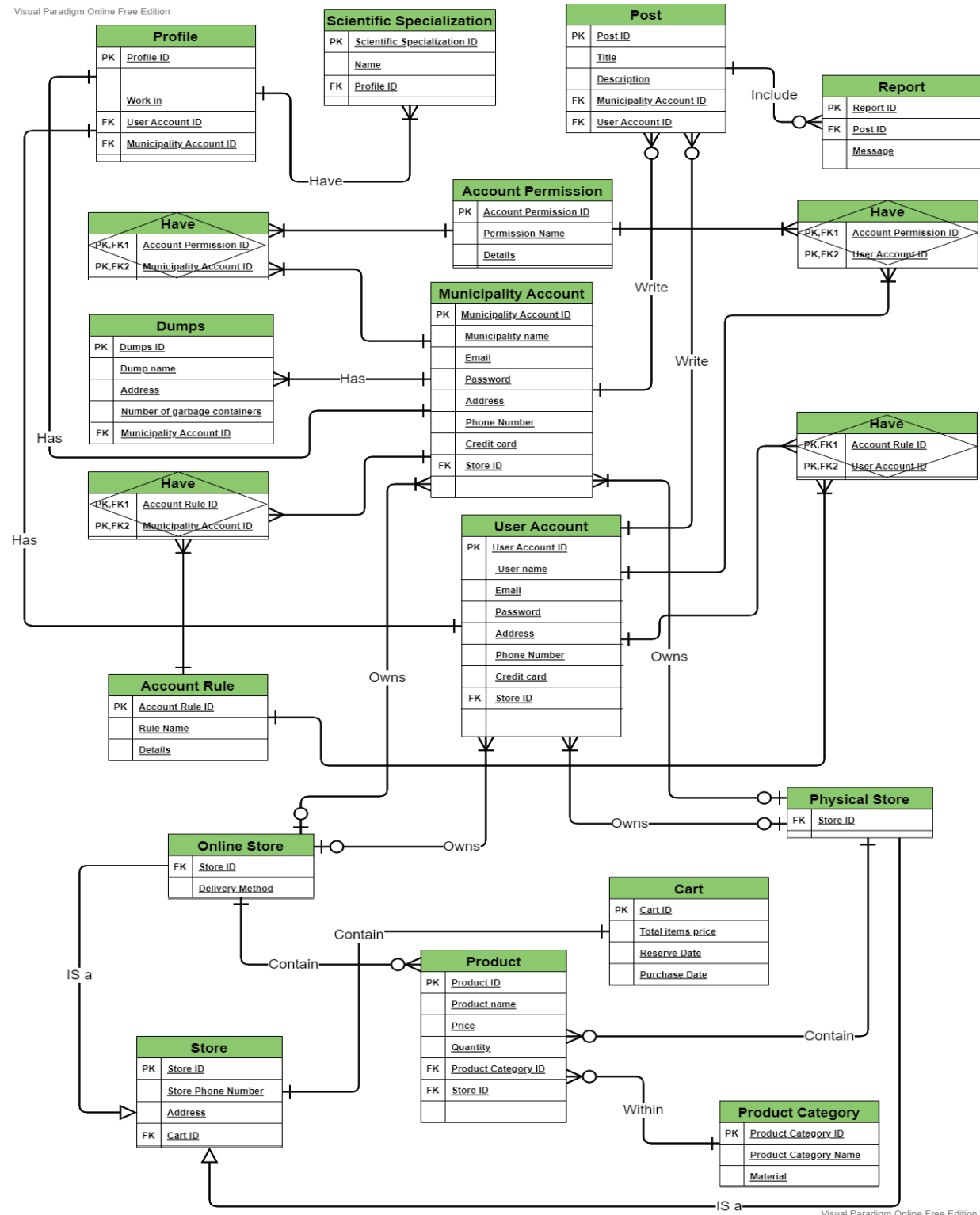


an entity whose nonprimary-key attributes are dependent on the full primary key.



Third normal form:

There is a repetition of data between two entities, namely the online store entity and the physical store entity. To solve the problem, I created a new entity named store that have the common data in it with a relation to the tables.





## Conclusion:

An entity-relationship diagram, or ER diagram, is essential for modeling the data stored in a database. It is the basic design upon which a database is built. ER diagrams specify what data we will store: the entities and their attributes. They also show how entities relate to other entities. Another advantage of ERDs is that they represent the data in a graphical manner. This makes it easier for business folks to understand.

## References

- attributed, w. f. (n.d.). *wikipedia fully attributed*. Retrieved from [https://en.wikipedia.org/wiki/Data\\_modeling#:~:text=A%20fully%20attributed%20data%20model%20contains%20detailed%20attributes,base%20data%20structures%20used%20to%20store%20the%20data](https://en.wikipedia.org/wiki/Data_modeling#:~:text=A%20fully%20attributed%20data%20model%20contains%20detailed%20attributes,base%20data%20structures%20used%20to%20store%20the%20data).
- base, w. k. (n.d.). *wikipedia key base*. Retrieved from [https://en.wikipedia.org/wiki/Data\\_model](https://en.wikipedia.org/wiki/Data_model)
- IBM. (n.d.). *IBM*. Retrieved from IBM: <https://www.ibm.com/cloud/learn/data-modeling>
- lucidchart. (n.d.). *lucidchart*. Retrieved from <https://www.lucidchart.com/pages/er-diagrams>
- Newsletter. (n.d.). *Newsletter*. Retrieved from <https://tdan.com/normalizing-with-entity-relationship-diagramming/4583#:~:text=First%20Normal%20Form%20%281NF%29%20The%20first%20normal%20form,of%20first%20normal%20form%20for%20the%20entity%20type>.
- wikipedia. (n.d.). *wikipedia*. Retrieved from [https://en.wikipedia.org/wiki/Context\\_model](https://en.wikipedia.org/wiki/Context_model)