****

REPUBLIC OF CAMEROON

Peace-Work-Fatherland

MINISTER OF HIGHER EDUCATION

FACULTY OF ENGINEERING AND TECHNOLOGY

REPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie

MINISRE DE L’ENSEIGNEMENT SUPERIEURE

FACULTE D’ENGENIERE ET DE TECHNOLOGIE

**Instructor: DR NKEMENI VALERY**

DEPARMENT OF COMPUTER ENGINEERING

COURSE TITLE: Internet and Mobile Programming

PROGRAMMING

GROUP TWO

**DATABASE DESIGN AND IMPLEMENTATION**

|  |  |
| --- | --- |
| **Names Matricule** | |
| ETOH KIERA EKUTY | FE20A035 |
| GWANA JUSLINE AKAH | FE20A047 |
| LASSI FOKOU BERICE JILDO | FE20A054 |
| YUVEN BRIAN | FE20A125 |
| AJONG NTI MAGCELOUS | FE20A006 |
|  |  |

**OUTLINE**

1. DATABASE DESIGN
2. Introduction
3. Advantages of ER-Diagram
4. ER\_Diagram
5. Relational Schema
6. DATABASE IMPLEMENTATION
7. Definition
8. Advantages of MySQL workbench
9. Choice of DNMS and it’s Advantages
10. Diagrammatic representation after implementation
11. SQL Queries
12. CONCLUSION

**DATABASE DESIGN AND IMPLEMENTATION**

1. **DATABASE DESIGN**

**INTRODUCTION**

An ER diagram, also known as an Entity-Relationship diagram, is a graphical representation of entities and their relationships to each other in a database.

It is used to design and model the structure of a database.. It involves identifying the entities that will be stored in the database, the attributes of those entities, and the relationships between the entities.

**Entities**:

Entities are objects or concepts that exist in the real world and can be represented in a database, such as a person, place, thing, or event. Entities are represented by rectangles. There exist strong , weak and associative Entities.

* A strong entity is an entity that can exist independently
* Weak entity depends on another entity for its existence.
* An associative entity is an entity that is used to represent a relationship between two or more entities.

**Attributes** :

Attributes are characteristics or properties of an entity, such as name, age, or address. Attributes are represented by ovals.

* Multivalued attributes are attributes that can have multiple values, such as a person's hobbies or interests.

**Relationships:**

Define the connections between entities. For example, a person may have a relationship with a company as an employee, or a product may have a relationship with a customer as a purchase. Relationships are represented with the diamond shape.

**The ER-diagram is drawned using the software Modelio**

**THE ADVANTAGES OF USING AN ER DIAGRAM IN DATABASE DESIGN INCLUDE:**

1. Improved communication: An ER diagram helps to communicate the structure of the database to stakeholders, including developers, designers, and end-users.

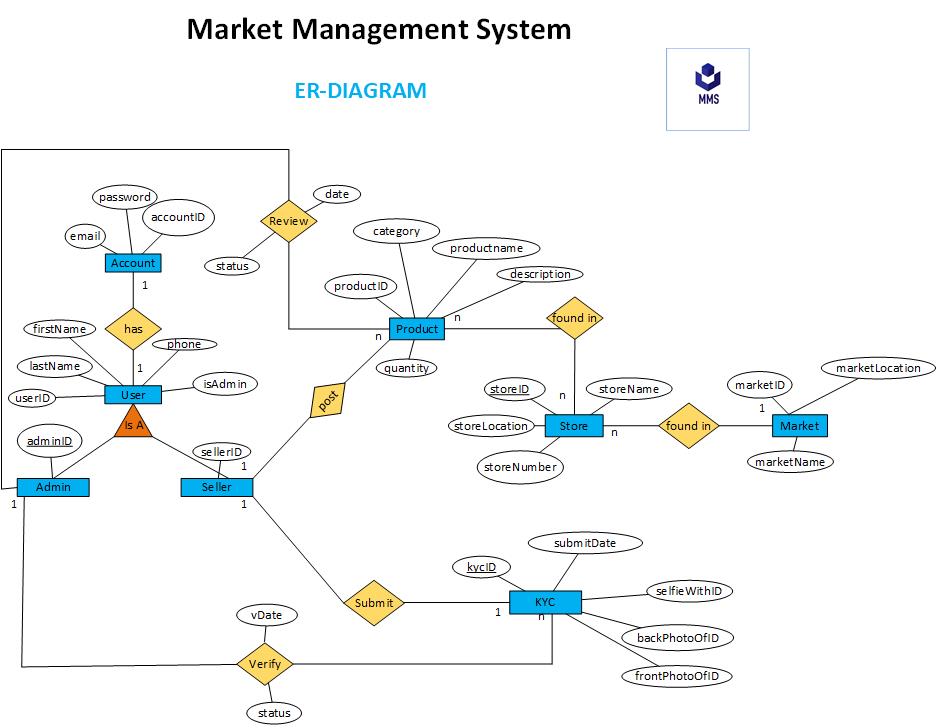
2. Clear representation of data: An ER diagram provides a clear and concise representation of the data in the database, making it easier to understand and work with.

3. Easier maintenance: An ER diagram makes it easier to maintain the database over time, as changes can be made more easily and with greater accuracy.

4. Reduced errors: An ER diagram helps to reduce errors in the database design by providing a clear and structured representation of the data.

5. Scalability: An ER diagram can help to design a database that is scalable and can grow with the needs of the organization over time.

**THE ER\_DIAGRAM**:



**RELATIONAL SCHEMA:**

The relational schema is a formal representation of a database design. It uses tables to represent entities, columns to represent attributes, and primary keys to uniquely identify rows in a table.

The following are the different relational schema for our system:

1. **User**

- UserID (Primary Key)

- FirstName

- LastName

- Phone

- IsAdmin

- ...

2. **Account**

- AccountID (Primary Key)

- UserID (Foreign Key references User.UserID)

- Email

- Password

- ...

3. **Admin**

- AdminID (Primary Key)

- UserID (Foreign Key references User.UserID)

- ...

4. **Seller**

- SellerID (Primary Key)

- UserID (Foreign Key references User.UserID)

- ...

5. **KYC**

- KYCID (Primary Key)

- SellerID (Foreign Key references Seller.SellerID)

- AdminID (Foreign Key references Admin.AdminID)

- FrontPhotoID

- BackPhotoID

- SelfieWithID

- SubmitDate

- VerificationDate

- ...

6. **Product**

- ProductID (Primary Key)

- SellerID (Foreign Key references Seller.SellerID)

- AdminID (Foreign Key references Admin.AdminID)

- StoreID (Foreign Key references Store.StoreID)

- Category

- Description

- ProductName

- Quantity

- ...

7. **Store**

- StoreID (Primary Key)

- SellerID (Foreign Key references Seller.SellerID)

- MarketID (Foreign Key references Market.MarketID)

- StoreName

- Location

- StoreNumber

- ...

8. **Market**

- MarketID (Primary Key)

- MarketName

- Location

1. **DATABASE IMPLEMENTATION**

**DEFINITION**:

The implementation of a database using MySQL Workbench involves creating a data model using an ER diagram, then generating the SQL code to create the tables, columns, and relationships in the database. Once the database is created, data can be inserted, updated, and queried using SQL commands.

**The advantages of using MySQL Workbench for database implementation include:**

1. User-friendly interface: MySQL Workbench has a user-friendly interface that makes it easy to create and manage databases.

2. Visual modeling: MySQL Workbench allows for visual modeling of databases using ER diagrams, which makes it easier to design and understand the structure of the database.

3. SQL code generation: MySQL Workbench can generate SQL code to create tables, columns, and relationships in the database, which saves time and reduces errors.

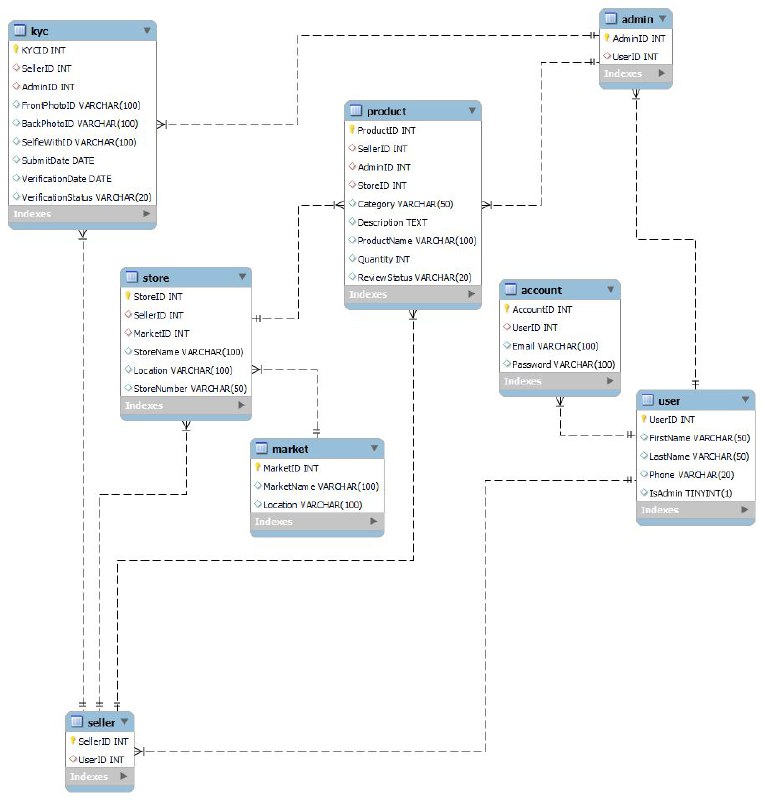
4. Collaboration: MySQL Workbench allows for collaboration among team members working on the same database project, which improves efficiency and accuracy.

5. Integration: MySQL Workbench integrates with other MySQL tools and services, making it easier to manage and maintain the database over time.

**PECEDURES**:

* Open MySQL Workbench.
* Create a new database.
* Import the relational schema into the database.
* Create tables and populate them with data.
* Once the database is implemented, it can be used to store and manage data.

**DIAGRAMATIC STRUCTURE OF DB AFTER IMPLEMENTATION**

****

**SQL QUERIES:**

-- Create User table

CREATE TABLE User (

UserID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Phone VARCHAR(20),

IsAdmin BOOLEAN,

-- Add additional attributes for User table if needed

);

-- Create Account table

CREATE TABLE Account (

AccountID INT PRIMARY KEY,

UserID INT,

Email VARCHAR(100),

Password VARCHAR(100),

FOREIGN KEY (UserID) REFERENCES User(UserID)

-- Add additional attributes for Account table if needed

);

-- Create Admin table

CREATE TABLE Admin (

AdminID INT PRIMARY KEY,

UserID INT,

FOREIGN KEY (UserID) REFERENCES User(UserID)

-- Add additional attributes for Admin table if needed

);

-- Create Seller table

CREATE TABLE Seller (

SellerID INT PRIMARY KEY,

UserID INT,

FOREIGN KEY (UserID) REFERENCES User(UserID)

-- Add additional attributes for Seller table if needed

);

-- Create KYC table

CREATE TABLE KYC (

KYCID INT PRIMARY KEY,

SellerID INT,

AdminID INT,

FrontPhotoID VARCHAR(100),

BackPhotoID VARCHAR(100),

SelfieWithID VARCHAR(100),

SubmitDate DATE,

VerificationDate DATE,

FOREIGN KEY (SellerID) REFERENCES Seller(SellerID),

FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)

-- Add additional attributes for KYC table if needed

);

-- Create Product table

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

SellerID INT,

AdminID INT,

StoreID INT,

Category VARCHAR(50),

Description TEXT,

ProductName VARCHAR(100),

Quantity INT,

FOREIGN KEY (SellerID) REFERENCES Seller(SellerID),

FOREIGN KEY (AdminID) REFERENCES Admin(AdminID),

FOREIGN KEY (StoreID) REFERENCES Store(StoreID)

-- Add additional attributes for Product table if needed

);

-- Create Store table

CREATE TABLE Store (

StoreID INT PRIMARY KEY,

SellerID INT,

MarketID INT,

StoreName VARCHAR(100),

Location VARCHAR(100),

StoreNumber VARCHAR(50),

FOREIGN KEY (SellerID) REFERENCES Seller(SellerID),

FOREIGN KEY (MarketID) REFERENCES Market(MarketID)

-- Add additional attributes for Store table if needed

);

-- Create Market table

CREATE TABLE Market (

MarketID INT PRIMARY KEY,

MarketName VARCHAR(100),

Location VARCHAR(100)

-- Add additional attributes for Market table if needed

);

**CONCLUSION**:

Database design is an important part of the database development process. It involves identifying the entities that will be stored in the database, the attributes of those entities, and the relationships between the entities.

The entity-relationship (ER) diagram is a graphical representation of a database design. The relational schema is a formal representation of a database design. MySQL is a popular open-source relational database management system that can be used to implement database designs.