Database Design Wakefit(Furniture)

Wakefit is an easy-to-use website to explore and customize furniture. It makes shopping simple and secure.

Entities

1. User
2. Admin
3. Product
4. Category
5. Order
6. Cart
7. payment
8. feedback

Functionality :-

1. User Details:

- **Customer Registration and Authentication:** Secure customer registration, login, and password management.
- **Customer Profile Management:** Customers can view and update their personal information, including contact details and address.
- **Customer Roles and Permissions:** Differentiate between roles such as regular customers and premium customers with appropriate permissions.

2. Admin Details:

- **Admin Dashboard:** Centralized control panel for managing the platform.
- **User Management:** Admins can add, edit, or remove users and manage their roles.
- **Content Management:** Admins can manage site content, including banners, announcements, and policies.

3. Product Details:

- **Product Catalog Management:** Add, edit, or remove products in the catalog.
- **Product Details Management:** Update product information, including descriptions, specifications, and images.
- **Inventory Management:** Track stock levels and manage inventory across different locations.

4. Category Details:

• **Category Management:** Create, edit, and delete product categories.

• **Subcategory Management:** Organize products into subcategories for better navigation.

5. Order Details:

• **Order Placement:** Customers can place orders for products.

• **Order History:** Customers can view their past orders and reorder products.

• **Order Management:** Admins can manage and process orders, including cancellations .

6. Shopping Cart Details:

• **Cart Management:** Customers can add, edit, or remove items in their shopping cart.

• **Cart Persistence:** Ensure the cart is saved for logged-in users across sessions.

• **Cart Summary:** Provide a detailed summary of items, prices, and total cost.

• **Discounts and Promotions:** Apply discounts, promo codes, and special offers.

7. Payment Details:

• **Payment Processing:** Secure and reliable processing of payments.

• **Multiple Payment Methods:** Support for various payment methods such as credit cards, UPI, and bank transfers.

• **Payment Confirmation:** Send confirmation notifications to customers upon successful payment.

8. Feedback Details:

• **Customer Feedback:** Allow customers to provide feedback on products and services.

• **Feedback Management:** Admins can view, respond to, and manage customer feedback.

• **Rating System:** Implement a rating system for products and services.

Define Attributes

1. Admin:

Attributes:
aID: Primary key for the admin
aEmail: Email address of the admin.
Password: password for an account admin

- **Relationships:**

  One Admin can handle multiple User (One-to-Many relationship with User table).

2. Category Table:

- **Attributes:**

catID: Primary key for the category.

categoryName: Name of the category.

Descriptions: Brief information about each category

- **Relationships:**

Each category can have multiple products (One-to-Many relationship with Product table).

3. Product Table:

- **Attributes:**

pID: Primary key for the product.

pName: Name of the product.

pDescription: Description of the product.

pPrice: Price of the product.

pImageUrl: URL of the product image.

unitsInStock: Number of units in stock.

quantityAvailable: availability of product

cID: Foreign key referencing the cID in the Category table, indicating the category to which the product belongs.

- **Relationships:**

Each product belongs to one category (Many-to-One relationship with Category table).

- **Relationships:**

Each product can be purchased by multiple customers (One-to-Many relationship with Customer table).

4. User Table:

- **Attributes:**

uID: Primary key for the customer.

uName: Name of the customer.

uEmail: Email address of the customer.

uMobNo: Mobile number of the customer.

uAddress: Address of the customer.

Password: password of account user

- **Relationships:**

Each customer can purchase multiple products (One-to-Many relationship with Product table).

5. Feedback Table:

- **Attributes:**

fID: Primary key for the feedback.

message: Feedback message.

fDate: date of feedback

pID : Foreign key referencing the pID in the Product table, indicating for product who provided for feedback .

uID: Foreign key referencing the uID in the Customer table, indicating the customer who provided the feedback.

- **Relationships:**

Each feedback is provided by one customer (Many-to-One relationship with Customer table).

6. Order Table:

Attributes:

oID: Primary key for the order.

oNumber: Tracking number of the order.

totalAmount: Total price of the order.

oQuantity: Quantity of items in the order.

status: Status of the order.

orderDate: Date when the order was item orderd.

cID: Foreign key referencing the cID in the Cart table, indicating the items in cart who placed the order.

uID: Foreign key referencing the cID in the Customer table, indicating the customer who placed the order.

- **Relationships:**

Each order is placed by one customer (Many-to-One relationship with Customer table).

7. Cart Table:

- **Attributes:**

cID: Primary key for the shopping cart.

unitPrice: Unit price of the product in the shopping cart.

quantity: Quantity of the product in the shopping cart.

pId: Identifier of the product in the shopping cart.

uID: Foreign key referencing the uID in the User table, indicating the user who owns the shopping cart.

pID: Foreign key referencing the pID in the Product table, indicating the product who add in the shopping cart.

- **Relationships:**

Each item in the shopping cart is owned by one customer (Many-to-One relationship with Customer table).

8. Payment Table:

- **Attributes:**

payID: Primary key for the payment.

payMethod: Payment method used.

payDate: Date of the payment.

amount : payable Amount for product.

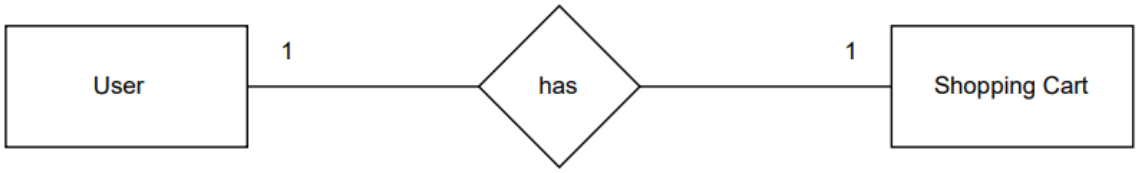
oID: Foreign key referencing the oID in the Order Table, indicating the order to which the payment corresponds.

- **Relationships:**

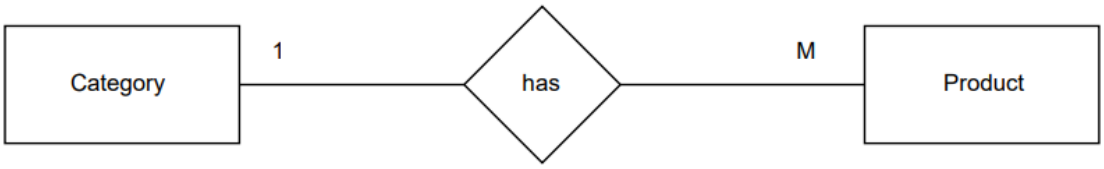
Each payment corresponds to one order (Many-to-One relationship with Order Table).

Let’s see a few examples of relationships:

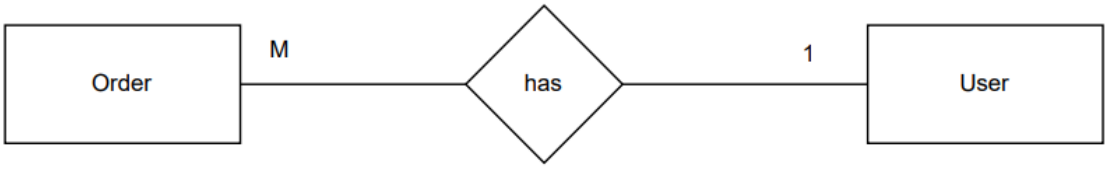
One to One



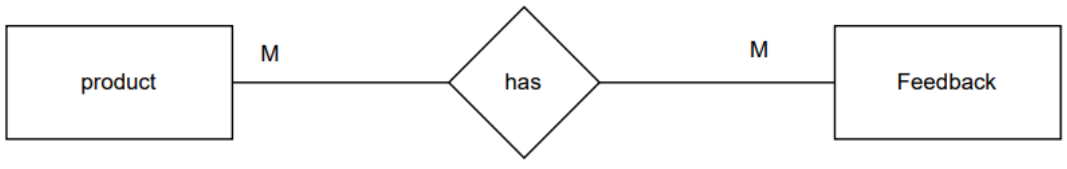
One to Many



Many to One

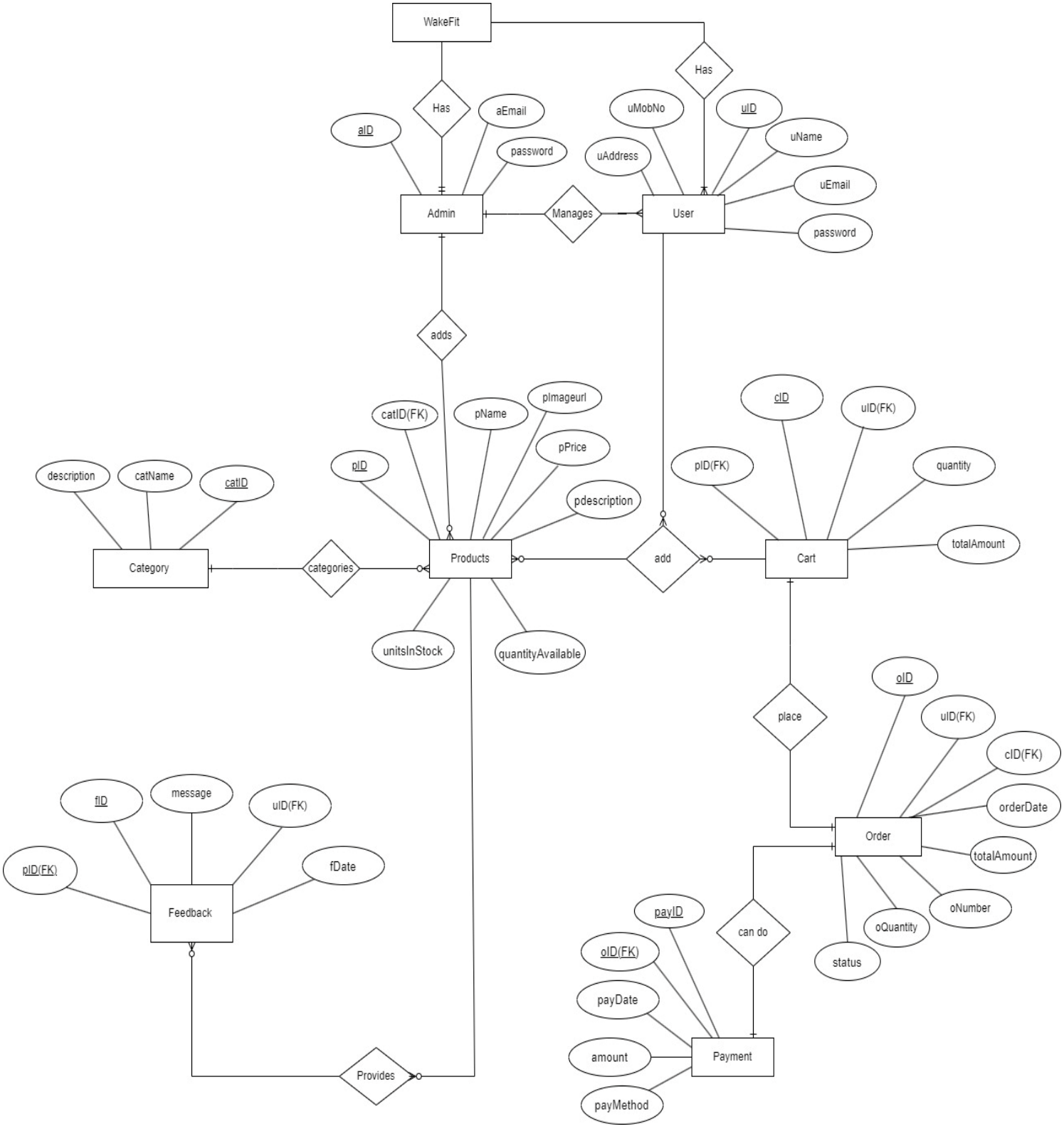


Many to Many



Now, let’s create the ER diagram to visually represent the entities and relationships.

ERD Diagram



In this ERD:

Relationships
Admin(1) --- (m) User
User (1) --- (1) Cart
Cart (1) --- (1) Order
User (1) --- (m) Order
Order (m) --- (m) Product
User (1) --- (m) Review
Review (m) --- (1) Product

4. Creating a Database

Using MySQL server, create a new database for your student management system. You can do this with SQL commands or through the graphical interface.

CREATE DATABASE wakefitFurniture;

5. Using a Database

Before performing any operations on a database, you need to select it using the USE statement:

Create database wakefit;

6. Creating the tables for each entity

USE wakefit;

```
CREATE TABLE Admin (
   aID INT AUTO_INCREMENT PRIMARY KEY,
   aEmail VARCHAR(255) NOT NULL,
   Password VARCHAR(255) NOT NULL
);

CREATE TABLE Category (
   catID INT AUTO_INCREMENT PRIMARY KEY,
   categoryName VARCHAR(255) NOT NULL,
   Descriptions TEXT
);


CREATE TABLE product (
pID INT PRIMARY KEY,
pName VARCHAR(255),
pDescription TEXT,
pUnitPrice FLOAT,    pImageUrl
VARCHAR(255),    active
BOOLEAN,    unitsInStock INT,
```

```
dateCreated DATE,
lastUpdated DATE,
   catID INT,
   FOREIGN KEY (catID) REFERENCES category(catID)
);


CREATE TABLE User (
   uID INT AUTO_INCREMENT PRIMARY KEY,
   uName VARCHAR(255) NOT NULL,
   uEmail VARCHAR(255) NOT NULL,
   uMobNo VARCHAR(15),
   uAddress TEXT,
   Password VARCHAR(255) NOT NULL,
   aID INT,
   FOREIGN KEY (aID) REFERENCES Admin(aID)
);


CREATE TABLE Feedback (
   fID INT AUTO_INCREMENT PRIMARY KEY,
   message TEXT,
   fDate DATE,
   pID INT,
   uID INT,
   FOREIGN KEY (pID) REFERENCES Product(pID),
   FOREIGN KEY (uID) REFERENCES User(uID)
);

CREATE TABLE `Order1` (
   oID INT AUTO_INCREMENT PRIMARY KEY,
   oNumber VARCHAR(255),
   totalAmount DECIMAL(10, 2),
   oQuantity INT,
   status VARCHAR(50),
   orderDate DATE,
   cID INT,
   uID INT,
   FOREIGN KEY (cID) REFERENCES Cart(cID),
   FOREIGN KEY (uID) REFERENCES User(uID)
);


CREATE TABLE Cart (
   cID INT AUTO_INCREMENT PRIMARY KEY,
   unitPrice DECIMAL(10, 2),
   quantity INT,
```

```
    uID INT,
    pID INT,
    FOREIGN KEY (uID) REFERENCES User(uID),
    FOREIGN KEY (pID) REFERENCES Product(pID)
);


CREATE TABLE Payment (
    payID INT AUTO_INCREMENT PRIMARY KEY,
    payMethod VARCHAR(50),
    payDate DATE,
    amount DECIMAL(10, 2),
    oID INT,
    FOREIGN KEY (oID) REFERENCES `Order1`(oID)
);
```

Table Structure

1.Admin

```
mysql> desc admin;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| aID      | int          | NO   | PRI | NULL    | auto_increment |
| aEmail   | varchar(255) | NO   |     | NULL    |                |
| Password | varchar(255) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
```

2.Category

```
mysql> desc category;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| catID        | int          | NO   | PRI | NULL    | auto_increment |
| categoryName | varchar(255) | NO   |     | NULL    |                |
| Descriptions | text         | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
```

3.User

```
mysql> desc  user;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| uID      | int          | NO   | PRI | NULL    | auto_increment |
| uName    | varchar(255) | NO   |     | NULL    |                |
| uEmail   | varchar(255) | NO   |     | NULL    |                |
| uMobNo   | varchar(15)  | YES  |     | NULL    |                |
| uAddress | text         | YES  |     | NULL    |                |
| Password | varchar(255) | NO   |     | NULL    |                |
| aID      | int          | YES  | MUL | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
```

4. Product

```
mysql> desc  product;
+------------------+--------------+------+-----+---------+----------------+
| Field            | Type         | Null | Key | Default | Extra          |
+------------------+--------------+------+-----+---------+----------------+
| pID              | int          | NO   | PRI | NULL    | auto_increment |
| pName            | varchar(255) | NO   |     | NULL    |                |
| pDescription     | text         | YES  |     | NULL    |                |
| pPrice           | decimal(10,2) | NO   |     | NULL    |                |
| pImageUrl        | varchar(255) | YES  |     | NULL    |                |
| unitsInStock     | int          | YES  |     | NULL    |                |
| quantityAvailable | int          | YES  |     | NULL    |                |
| catID            | int          | YES  | MUL | NULL    |                |
+------------------+--------------+------+-----+---------+----------------+
```

5.Order

```
mysql> desc order1;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| oID         | int          | NO   | PRI | NULL    | auto_increment |
| oNumber     | varchar(255) | YES  |     | NULL    |                |
| totalAmount | decimal(10,2) | YES  |     | NULL    |                |
| oQuantity   | int          | YES  |     | NULL    |                |
| status      | varchar(50)  | YES  |     | NULL    |                |
| orderDate   | date         | YES  |     | NULL    |                |
| cID         | int          | YES  | MUL | NULL    |                |
| uID         | int          | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
```

6. Cart

```
mysql> desc cart;
+-----------+---------------+------+-----+---------+----------------+
| Field     | Type          | Null | Key | Default | Extra          |
+-----------+---------------+------+-----+---------+----------------+
| cID       | int           | NO   | PRI | NULL    | auto_increment |
| unitPrice | decimal(10,2) | YES  |     | NULL    |                |
| quantity  | int           | YES  |     | NULL    |                |
| uID       | int           | YES  | MUL | NULL    |                |
| pID       | int           | YES  | MUL | NULL    |                |
+-----------+---------------+------+-----+---------+----------------+
```

7. Payment

```
mysql> desc payment;
+-----------+---------------+------+-----+---------+----------------+
| Field     | Type          | Null | Key | Default | Extra          |
+-----------+---------------+------+-----+---------+----------------+
| payID     | int           | NO   | PRI | NULL    | auto_increment |
| payMethod | varchar(50)   | YES  |     | NULL    |                |
| payDate   | date          | YES  |     | NULL    |                |
| amount    | decimal(10,2) | YES  |     | NULL    |                |
| oID       | int           | YES  | MUL | NULL    |                |
+-----------+---------------+------+-----+---------+----------------+
```

8. Feedback

```
mysql> desc feedback;
+---------+------+------+-----+---------+----------------+
| Field   | Type | Null | Key | Default | Extra          |
+---------+------+------+-----+---------+----------------+
| fID     | int  | NO   | PRI | NULL    | auto_increment |
| message | text | YES  |     | NULL    |                |
| fDate   | date | YES  |     | NULL    |                |
| pID     | int  | YES  | MUL | NULL    |                |
| uID     | int  | YES  | MUL | NULL    |                |
+---------+------+------+-----+---------+----------------+
```