



# Database Application Design

## Overview

In this initial phase of your group project, you will focus on the intricate process of crafting a resilient database application using the entity-relationship (E-R) model. Your tasks include listing essential entity sets with underlined primary keys and delineating associated attributes, clarifying relationships between different entity sets, and constructing a comprehensive E-R diagram. You will also offer detailed rationale behind your design choices, highlighting how each element enhances the overall functionality of the envisioned database application.

---

## Part I: Setting Up Your Group's GitHub Repository

Now that you've formed your groups for the semester, you will create a shared GitHub repository for your group project. This repository will serve as a central workspace for all project materials including design documents, E-R diagrams, presentation slides, database scripts and application code.

Setting up your repository supports collaboration, version control, and transparency in individual contributions—and mirrors how real-world development teams work. All group members are expected to make regular, meaningful contributions to the shared repository. Commit history may be considered alongside peer evaluations when assessing individual participation in the group project.

### **ALL Group Members Must Complete Following:**

- Complete the [Install Your IDE section](#) to install an IDE on your local device that supports web development (e.g., VS Code).
- Complete the [Create A GitHub Account section](#) if you do not already have one.
- Choose **ONE** of the following options:
  - Complete the [Installation & Authentication section](#) to install and authenticate **GitHub Desktop (GUI)** on your device.
  - Complete the [Installation & Authentication section](#) to install and authenticate **Git (CLI)** on your device.

### **ONE Group Member Must Complete Following:**

- ~~Complete the [Create a GitHub Repository for Your Course Projects section](#) to create your group's repository.~~
- ~~Clone the repository to your local device using [GitHub Desktop](#) or [Git CLI](#).~~
- ~~Navigate to and open the repo folder you cloned to your local device (not the one on GitHub.com).~~
- ~~Inside your repo folder, create a new folder named **documentation**.~~
- ~~In your group's repository on GitHub, go to the **Settings** tab → **Collaborators**.~~
- ~~Click **Add people**, then enter your group members' GitHub usernames and send invitations. This step ensures all group members have access to the shared repository.~~

### **ALL Group Members (After Receiving the Invitation) Complete Following:**

- ~~In your GitHub dashboard, click the **Notifications** icon (). Open the invitation message and click **Accept Invitation** to join the repo as a collaborator.~~
- ~~Clone the repository to your local device using [GitHub Desktop](#) or [Git CLI](#).~~
- ~~Open the repository in VS Code and confirm that it contains the expected structure (e.g., the **documentation** folder).~~

## Part II: Database Application Design

Design your group's database application using the E-R model by providing thoughtful responses to the following questions.

1. List the entity sets and their attributes with primary keys underlined in your design.

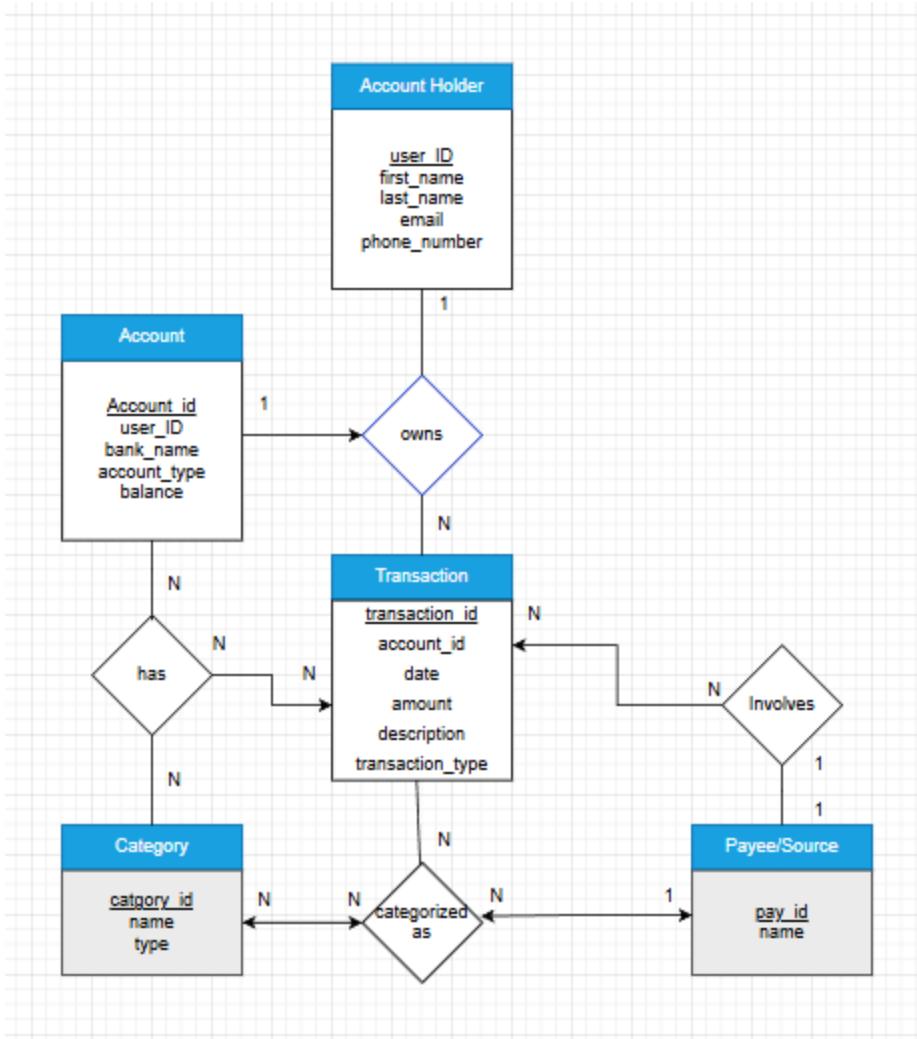
- a. Account entity
  - i. Account\_ID
  - ii. user\_ID
  - iii. Bank\_name
  - iv. Account\_type
  - v. balance
- b. Account Holder
  - i. user\_ID
  - ii. First\_name
  - iii. Last\_name
  - iv. Email
  - v. phone\_number
- c. Transaction
  - i. transaction\_ID
  - ii. Date
  - iii. Amount
  - iv. Description
  - v. transaction\_type
- d. Category
  - i. category\_ID
  - ii. Name
  - iii. type
- e. Payee/Source
  - i. pay\_ID
  - ii. name

2. List the relationships between different entity sets in your design.

- a. Account Holder: The user\_id is the primary key because every person needs a unique identifier. The first name, last name, email, and phone

number all live here because those describe the person, not the account itself.

- b. Each account has an account\_id as the primary key, and a user\_id as a foreign key to link it back to the Account Holder. That relationship shows that one person can own multiple accounts.
  - c. Then we have the Transaction entity: Each transaction has a transaction\_id as the primary key and an account\_id as a foreign key, which links the transaction to a specific account. This makes sense because one account can have many transactions. IE: deposits, withdrawals, purchases, but each transaction belongs to exactly one account. We also store the transaction type, date, amount, and description here.
  - d. Category and Payee/Source help describe transactions: Category acts like a label. It explains what the transaction was for, like groceries, rent, or income. Payee or Source represents who the transaction involved. An example of this is Walmart, Amazon, or an employer. Separating these lets us track spending by category and also see where money is going or coming from.
3. Construct an E-R diagram for your database application. Use the following steps to design your database using the E-R model:
    - Start by identifying essential entity sets to be included in your database.
    - Choose relevant attributes representing the values to be captured in the database for each entity set.
    - Formulate relationship sets among entities, addressing potential redundancy in attributes.
    - Incorporate any necessary constraints (ie, relationship cardinalities, total/partial participation, descriptive attributes, weak entity sets).



4. Provide a detailed explanation of the rationale behind your design choice. Highlight how each element contributes to the overall functionality of your database application.
- With this database, the various relationships, attributes, and entities allow us to organize the flow and categorization of financial data. Holding an Account Holder and Account entity allow us to track a single person and the accounts they have. This also connects to the different transactions that someone can have within an account whether that's income or outgoing costs. The category entity allows us to keep a clean organization of expenses and their types, while the Payee/Source entity allows us to see where the transactions came from.

## Submission

When you're finished, complete the following steps to submit your work:

- Export your document with responses as a **PDF file AND save it inside your documentation** folder. Refer to the following for documentation on how to do this:
  - [Google Docs](#) (*File → Download → PDF Document*)
  - [Microsoft Word](#) (*File → Save As / Export → PDF*)
  - [Pages](#) (*File → Export To → PDF*)
- Export your **E-R diagram** as an **image file** and save it **inside** your **documentation** folder. Be sure your file is clearly named (i.e., **er\_diagram.png**).
- Upload all your changes to GitHub.
  - If you're using **GitHub Desktop (GUI)**, complete the [Uploading Changes \(GitHub Desktop\) section](#) to upload your changes from your local device to GitHub.
  - If you're using **Git (CLI)**, complete the [Uploading Changes \(GitHub CLI\) section](#) to upload your changes from your local device to GitHub.

\***ONE group member\*** must paste the URL of your GitHub repository in the provided textbox in Brightspace. Click the blue *Submit* button to successfully submit your work for this assignment.

## Grading Rubric

You can refer to the **Database Application Design grading rubric** given in Brightspace for this assignment to find details on how your submission will be graded.