Database Group Project Team 11

In our "Budget Application Project", we tried to understand the importance of budget management and creating the best application for our users.

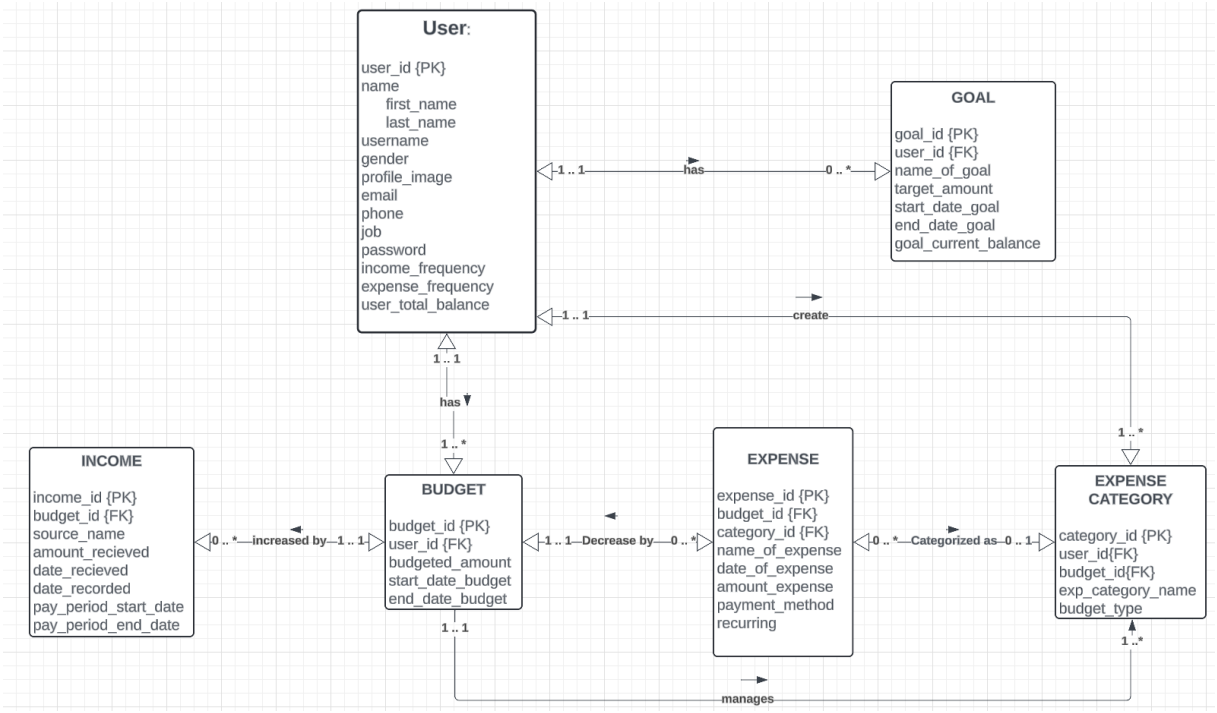
Our Mission Statement:

Our mission is to enable people and businesses to attain financial well-being by providing them with smart budget analysis and effective management. Our dedication lies in offering an extensive and intuitive budget application database that encourages good financial practices, educates decision-making, and eventually improves our users' financial health. Through the use of cutting-edge technology and adaptable features, our goals are to make budgeting easier, encourage transparency, and instill trust in everyone's financial planning.

1. An unnormalized set of data, including
    - A list of the data items being maintained
    - A set of data in a grid or spreadsheet

Column Name	Data Type	Description
first_name	Varchar(255)	User's first name
last_name	Varchar(255)	User's last name
username	Varchar(255)	Username for login
gender	Varchar(255)	Male or Female
email	Varchar(255)	User's email address
phone	Varchar(255)	User's phone number
password	Varchar(255)	User's password
income_frequency	Double	Frequency of income (e.g., monthly income would be listed as 12)
user_total_balance	Double	User's total balance
expense_frequency	Double	Frequency of expenses (e.g., monthly expenses would be listed as 12)
source_name	Varchar(255)	Name of the income source (e.g., Salary, Investment)
amount_received	Double	Amount of income received
date_received	Datetime	Date the income was received
date_recorded	Datetime	Date the income was recorded in the system
pay_period_start_date	Datetime	Start date of the pay period (for recurring income)
pay_period_end_date	Datetime	End date of the pay period (for recurring income)
name_of_expense	Varchar(255)	Name of the expense category (e.g., Groceries, Rent)
exp_category_name	Varchar(255)	Group related categories (e.g., Housing)
budget_type	Varchar(255)	Categorize expenses (fixed, variable, discretionary)
date_of_expense	Datetime	Date of the expense
amount_expense	Double	Amount of the expense
payment_method	Varchar(255)	Payment method used (e.g., Cash, Credit Card)
recurring	Boolean	Yes/No for recurring transaction
budgeted_amount	Double	Budgeted amount for the category during the period
start_date_budget	Datetime	Start date of the budget period
end_date_budget	Datetime	End date of the budget period
name_of_goal	Varchar(255)	Name of the savings goal (e.g., Emergency Fund)
target_amount	Double	Target amount to be saved for the goal
start_date_goal	Datetime	Start date for saving towards the goal
end_date_goal	Datetime	Target end date to reach the goal (optional)
goal_current_balance	Double	Current amount saved towards the goal

2. E-R Diagram for our 3NF database (include the participation/cardinality information... 0...* , 1..1 , etc.)



3. SQL queries for creating tables and inserting data from the UNF database table into the tables created.

User

CREATE TABLE User AS

SELECT first_name, last_name,

username, gender,email,

phone, password, income_frequency, expense_frequency , user_total_balance

FROM UNF;

ALTER TABLE User

ADD COLUMN profile_image MEDIUMBLOB,

ADD COLUMN job VARCHAR(255),

ADD COLUMN user_id BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT;

GOAL

```
CREATE table goal AS

SELECT  name_of_goal, target_amount, start_date_goal, end_date_goal, goal_current_balance

from unf;


ALTER table goal

ADD COLUMN user_id BIGINT NOT NULL,

ADD COLUMN goal_id bigint not null  PRIMARY key AUTO_INCREMENT;


ALTER TABLE goal ADD CONSTRAINT user_id FOREIGN KEY (user_id) REFERENCES user(user_id) ON
UPDATE CASCADE;
```

BUDGET

```
CREATE TABLE Budget  AS

SELECT budgeted_amount, start_date_budget,

end_date_budget

FROM UNF;


ALTER TABLE Budget

ADD COLUMN user_id BIGINT NOT NULL,

ADD COLUMN budget_id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY;


ALTER TABLE Budget

ADD CONSTRAINT PK_budget_user_id FOREIGN KEY (user_id) REFERENCES User (user_id) ON UPDATE
CASCADE;
```

INCOME

CREATE TABLE income AS SELECT source_name, date_received, date_recorded, pay_period_start_date, pay_period_end_date,amount_received FROM UNF;

ALTER TABLE income

ADD COLUMN income_id bigint NOT NULL AUTO_INCREMENT PRIMARY KEY,

ADD COLUMN budget_id BIGINT NOT NULL;

ALTER TABLE INCOME ADD CONSTRAINT FK_income_budget_id FOREIGN KEY (budget_id) REFERENCES Budget (budget_id) ON UPDATE CASCADE;

EXPENSE CATEGORY

CREATE TABLE Expense_Category AS

SELECT budget_type, exp_category_name

FROM UNF;

ALTER TABLE Expense_Category

ADD COLUMN category_id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,

ADD COLUMN budget_id BIGINT NOT NULL,;

ALTER TABLE Expense_Category ADD CONSTRAINT FK_expense_category_user_id FOREIGN KEY (user_id) REFERENCES User (user_id) ON UPDATE CASCADE;

ALTER TABLE Expense_Category ADD CONSTRAINT FK_expense_category_budget_id FOREIGN KEY (budget_id) REFERENCES Budget (budget_id) ON UPDATE CASCADE;

EXPENSE

CREATE TABLE Expense AS

SELECT name_of_expense, date_of_expense, amount_expense,payment_method, recurring

FROM UNF;

```
ALTER TABLE Expense

ADD COLUMN category_id BIGINT NOT NULL,

ADD COLUMN budget_id BIGINT NOT NULL,

ADD COLUMN expense_id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY;


ALTER TABLE expense

ADD CONSTRAINT FK_exps_cat_id FOREIGN KEY (category_id) REFERENCES Expense_Category
(category_id);


ALTER TABLE expense

ADD CONSTRAINT FK_budget_id FOREIGN KEY (budget_id) REFERENCES Budget (budget_id);
```

4. A view to recreate the original dataset from our 3NF Tables (this will be used for comparison purposes to the original dataset we provide as number 1 of this deliverable)

```
CREATE VIEW ComparisonView AS

SELECT

u.first_name,  u.last_name, u.username, u.gender, u.email,  u.phone, u.password, u.income_frequency,
u.expense_frequency,  u.user_total_balance,  g.name_of_goal, g.target_amount,  g.start_date_goal,
g.end_date_goal, g.goal_current_balance,  b.budgeted_amount,  b.start_date_budget,
b.end_date_budget, i.source_name, i.date_received,  i.date_recorded,  i.pay_period_start_date,
i.pay_period_end_date, i.amount_received, ec.exp_category_name,  e.name_of_expense,
e.date_of_expense,  e.amount_expense, e.payment_method, e.recurring

FROM User u

LEFT JOIN Goal g ON u.user_id = g.user_id

LEFT JOIN Budget b ON u.user_id = b.user_id

LEFT JOIN Income i ON  b.budget_id = i.budget_id

LEFT JOIN Expense_Category ec ON b.budget_id = ec.budget_id AND u.user_id = ec.user_id

LEFT JOIN Expense e ON ec.category_id = e.category_id AND u.user_id = e.user_id AND b.budget_id =
ec.budget_id;
```