# Kickstarter Crowdfunding Database Project

## 1. Introduction

This project simulates a Kickstarter-style crowdfunding platform using MySQL. The database stores information about users, projects, rewards, and pledges. It allows creators to list their projects, and backers to pledge money in exchange for rewards. The system can track project progress, funding goals, and backer contributions.

## 2. Objective and Scope

The main objective of this project is to design and implement a relational database that supports the core functions of a crowdfunding platform. The scope includes defining the database schema, inserting sample data, and executing SQL queries to extract insights such as top-funded projects, total raised amounts, and backer lists.

## 3. Database Design

The database consists of the following tables:

• Users – Stores information about users who can be either project creators or backers.  
• Projects – Stores details about each crowdfunding project.  
• Rewards – Stores details of reward tiers for each project.  
• Pledges – Stores backer contributions linked to specific rewards.

The following diagram shows the relationship between the tables:

[ER Diagram Here]

## 4. CREATE TABLE Scripts

CREATE TABLE Users (  
 user\_id INT PRIMARY KEY,  
 name VARCHAR(100),  
 email VARCHAR(100) UNIQUE,  
 role ENUM('creator', 'backer')  
);  
  
CREATE TABLE Projects (  
 project\_id INT PRIMARY KEY,  
 creator\_id INT,  
 title VARCHAR(200),  
 description TEXT,  
 category ENUM('Art', 'Technology', 'Design', 'Music', 'Film'),  
 goal\_amount DECIMAL(10,2),  
 current\_amount DECIMAL(10,2),  
 start\_date DATE,  
 end\_date DATE,  
 status ENUM('ongoing', 'successful', 'failed'),  
 FOREIGN KEY (creator\_id) REFERENCES Users(user\_id)  
);  
  
CREATE TABLE Rewards (  
 reward\_id INT PRIMARY KEY,  
 project\_id INT,  
 description TEXT,  
 amount DECIMAL(10,2),  
 FOREIGN KEY (project\_id) REFERENCES Projects(project\_id)  
);  
  
CREATE TABLE Pledges (  
 pledge\_id INT PRIMARY KEY,  
 project\_id INT,  
 backer\_id INT,  
 reward\_id INT,  
 amount DECIMAL(10,2),  
 pledge\_date DATE,  
 FOREIGN KEY (project\_id) REFERENCES Projects(project\_id),  
 FOREIGN KEY (backer\_id) REFERENCES Users(user\_id),  
 FOREIGN KEY (reward\_id) REFERENCES Rewards(reward\_id)  
);

## 5. Sample Data (INSERT Statements)

INSERT INTO Users (user\_id, name, email, role) VALUES  
(1, 'Alice', 'alice@example.com', 'creator'),  
(2, 'Bob', 'bob@example.com', 'backer'),  
(3, 'Charlie', 'charlie@example.com', 'backer'),  
(4, 'Diana', 'diana@example.com', 'creator');  
  
INSERT INTO Projects (project\_id, creator\_id, title, description, category, goal\_amount, current\_amount, start\_date, end\_date, status) VALUES  
(1, 1, 'Eco-Friendly Notebook', 'Recyclable paper notebooks', 'Design', 5000.00, 2500.00, '2025-07-01', '2025-07-30', 'ongoing'),  
(2, 4, 'Smart Garden', 'Automated indoor plant watering system', 'Technology', 20000.00, 30000.00, '2025-05-01', '2025-06-01', 'successful'),  
(3, 1, 'Solar Charger Bag', 'Backpack with solar panel charger', 'Technology', 30000.00, 10000.00, '2025-06-01', '2025-07-01', 'failed'),  
(4, 4, 'Eco-Friendly Toothbrush', 'Bamboo biodegradable toothbrush', 'Design', 10000.00, 3500.00, '2025-08-01', '2025-08-30', 'ongoing');  
  
INSERT INTO Rewards (reward\_id, project\_id, description, amount) VALUES  
(1, 1, '1 Notebook', 250.00),  
(2, 1, '5 Notebooks', 1000.00),  
(3, 2, 'Small Smart Garden', 5000.00),  
(4, 2, 'Large Smart Garden', 12000.00);  
  
INSERT INTO Pledges (pledge\_id, project\_id, backer\_id, reward\_id, amount, pledge\_date) VALUES  
(1, 1, 2, 1, 250.00, '2025-07-02'),  
(2, 1, 3, 2, 1000.00, '2025-07-05'),  
(3, 2, 2, 3, 5000.00, '2025-05-05'),  
(4, 2, 3, 4, 12000.00, '2025-05-10');

## 6. Example Queries

-- Total raised per project  
SELECT project\_id, title, SUM(amount) AS total\_raised  
FROM Pledges  
JOIN Projects USING (project\_id)  
GROUP BY project\_id, title;  
  
-- Top funded project  
SELECT title, current\_amount  
FROM Projects  
ORDER BY current\_amount DESC  
LIMIT 1;  
  
-- Backers list per project  
SELECT project\_id, title, name AS backer\_name  
FROM Pledges  
JOIN Users ON Pledges.backer\_id = Users.user\_id  
JOIN Projects USING (project\_id)  
ORDER BY project\_id;

## 7. Conclusion

This project demonstrates the use of MySQL to design a relational database for a crowdfunding platform. It covers table creation, data insertion, and example queries for extracting meaningful insights.