Q.2) Donor Management System for Donations, Campaigns, and Beneficiaries

-> This database is designed to manage and track donors, donations, campaigns, and beneficiaries. It helps an organization monitor donations, register donors, track campaign progress, and manage beneficiary details efficiently.

### 1. Database Tables Design

**Table: Donors Table**

**This table stores information about individuals or entities who contribute to campaigns**.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| donor\_id | INT (PK) | Unique identifier for each donor |
| first\_name | VARCHAR(50) | Donor's first name |
| last\_name | VARCHAR(50) | Donor's last name |
| phone\_number | VARCHAR(15) | Donor's contact number |
| email | VARCHAR(100) | Donor's email address |
| address | VARCHAR(255) | Donor's address |

**Table: Donations Table**

**This table records each donation made by a donor, linking it to campaigns and tracking amounts.**

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| donation\_id | INT (PK) | Unique identifier for each donation |
| donor\_id | INT (FK) | References Donors (donor\_id) |
| campaign\_id | INT (FK) | References Campaigns (campaign\_id) |
| donation\_date | DATE | Date of donation |
| amount | DECIMAL(10,2) | Donation amount |

**Table:** **Campaigns Table**

**This table contains details about fundraising campaigns, including their targets and objectives.**

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| campaign\_id | INT (PK) | Unique identifier for each campaign |
| campaign\_name | VARCHAR(100) | Name of the campaign |
| target\_amount | DECIMAL(10,2) | Target donation amount for the campaign |
| total\_raised | DECIMAL(10,2) | Total amount raised (calculated from donations) |
| start\_date | DATE | Campaign start date |
| end\_date | DATE | Campaign end date |

**Table: Beneficiaries Table**

**This table lists the individuals or groups who benefit from the donations made during campaigns.**

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| beneficiary\_id | INT (PK) | Unique identifier for each beneficiary |
| name | VARCHAR(100) | Name of the beneficiary |
| description | TEXT | Description of the beneficiary (e.g., project description) |
| campaign\_id | INT (FK) | References Campaigns (campaign\_id) |

### 2. Constraints for Referential Integrity

### **Foreign Keys:**

### **donor\_id in Donations references Donors(donor\_id)**

### **campaign\_id in Donations references Campaigns(campaign\_id)**

### **campaign\_id in Beneficiaries references Campaigns(campaign\_id)**

### **Primary Keys:**

### **donor\_id in Donors table**

### **donation\_id in Donations table**

### **campaign\_id in Campaigns table**

### **beneficiary\_id in Beneficiaries table**

### **Check Constraints:**

### **Ensure amount in Donations is a non-negative value.**

### **Ensure total\_raised in Campaigns does not exceed target\_amount.**

### 3. Stored Procedures

#### ****a. Register Donation****

This stored procedure registers a new donation from a donor, links it to a campaign, and updates the campaign’s total raised amount.

CREATE PROCEDURE RegisterDonation(IN donorId INT, IN campaignId INT, IN donationAmount DECIMAL(10,2), IN donationDate DATE)

BEGIN

DECLARE totalRaised DECIMAL(10,2);

-- Insert the donation record

INSERT INTO Donations (donor\_id, campaign\_id, donation\_date, amount)

VALUES (donorId, campaignId, donationDate, donationAmount);

-- Update the total amount raised for the campaign

SELECT total\_raised INTO totalRaised FROM Campaigns WHERE campaign\_id = campaignId;

SET totalRaised = totalRaised + donationAmount;

UPDATE Campaigns SET total\_raised = totalRaised WHERE campaign\_id = campaignId;

END;

#### ****a. Process Campaign Completion****

This stored procedure handles the completion of a campaign by checking if the target was met and updating the campaign’s status.

#### CREATE PROCEDURE ProcessCampaignCompletion(IN campaignId INT)

#### BEGIN

#### DECLARE targetAmount DECIMAL(10,2);

#### DECLARE totalRaised DECIMAL(10,2);

#### -- Get the target amount and total raised amount

#### SELECT target\_amount, total\_raised INTO targetAmount, totalRaised

#### FROM Campaigns WHERE campaign\_id = campaignId;

#### 

#### -- Update the campaign status based on whether the target was met

#### IF totalRaised >= targetAmount THEN

#### UPDATE Campaigns SET status = 'Completed' WHERE campaign\_id = campaignId;

#### ELSE

#### UPDATE Campaigns SET status = 'Failed' WHERE campaign\_id = campaignId;

#### END IF;

#### END;

#### c. Register New Campaign

This procedure registers a new campaign, initializing the total raised amount to zero.

### CREATE PROCEDURE RegisterCampaign(IN campaignName VARCHAR(100), IN targetAmount DECIMAL(10,2), IN startDate DATE, IN endDate DATE)

### BEGIN

### INSERT INTO Campaigns (campaign\_name, target\_amount, total\_raised, start\_date, end\_date)

### VALUES (campaignName, targetAmount, 0, startDate, endDate);

### END;

### 4. Triggers

**a. Trigger on Donation (Update Campaign Total)**

**This trigger ensures that every time a donation is made, the total raised amount for the associated campaign is automatically updated.**

#### CREATE TRIGGER AfterDonationInsert

#### AFTER INSERT ON Donations

#### FOR EACH ROW

#### BEGIN

#### DECLARE totalRaised DECIMAL(10,2);

#### -- Get the current total raised for the campaign

#### SELECT total\_raised INTO totalRaised FROM Campaigns WHERE campaign\_id = NEW.campaign\_id;

#### -- Update the total raised for the campaign

#### SET totalRaised = totalRaised + NEW.amount;

#### UPDATE Campaigns SET total\_raised = totalRaised WHERE campaign\_id = NEW.campaign\_id;

#### END;

**b. Trigger on Campaign Completion (Update Campaign Status)**

**This trigger checks if the target amount has been met after each donation and updates the campaign status accordingly.**

CREATE TRIGGER AfterDonationUpdate

AFTER UPDATE ON Donations

FOR EACH ROW

BEGIN

DECLARE targetAmount DECIMAL(10,2);

DECLARE totalRaised DECIMAL(10,2);

-- Check if the campaign's target amount is met

SELECT target\_amount, total\_raised INTO targetAmount, totalRaised FROM Campaigns WHERE campaign\_id = NEW.campaign\_id;

IF totalRaised >= targetAmount THEN

UPDATE Campaigns SET status = 'Completed' WHERE campaign\_id = NEW.campaign\_id;

ELSE

UPDATE Campaigns SET status = 'Ongoing' WHERE campaign\_id = NEW.campaign\_id;

END IF;

END;

### 5. SQL Queries for Reports

#### a. Generate Donation Trends

#### SELECT donation\_date, SUM(amount) AS total\_donated

#### FROM Donations

#### GROUP BY donation\_date

#### ORDER BY donation\_date;

#### b. Generate Campaign Success Report

#### SELECT campaign\_name, target\_amount, total\_raised,

#### (total\_raised / target\_amount) \* 100 AS success\_rate

#### ORDER BY success\_rate DESC;

#### c. Analyze Donations by Donor

SELECT d.first\_name, d.last\_name, SUM(do.amount) AS total\_contributed

FROM Donations do

JOIN Donors d ON do.donor\_id = d.donor\_id

GROUP BY d.donor\_id

ORDER BY total\_contributed DESC;

d. **Generate Campaign Success Report**

SELECT b.name, b.description

FROM Beneficiaries b

JOIN Campaigns c ON b.campaign\_id = c.campaign\_id

WHERE c.campaign\_id = 1;

**Conclusion**

This Donor Management System is structured to handle all aspects of fundraising campaigns, from registering donors to tracking donations and managing beneficiaries. It includes:

* Efficient management of donor and campaign data.
* Automatic updates of campaign totals with triggers.
* Insightful reports on donation trends, campaign success, and donor contributions.