GREENBACKS

12.06.2022

Team 3
Arun Kumar Reddy Jinna
Naga Sai Bhavana Davuluri
Harika Reddy Puli
Rajyalaxmi Kondapalli

Overview

We all know, How important is 1842 Day for Villanova? It is called the annual day of giving as our entire Villanova community celebrates being Villanovans by raising funds that help us dream big, reach heights, and be united as a community. Fundraising is an essential component for any nonprofit organization like ours. In formal terms, Fundraising means "An ongoing process of seeking funds or donations made voluntarily which helps an organization to grow and meet the needs of the community". There are approaches to fundraising like Online or Digital Fundraising, Virtual Fundraising, Event-Based Fundraising, Auctioning, Pledges, etc.., As the ultimate goal for any fundraising project is to make every dollar count, we name our team ***GREENBACKS***.

Goals

1. Creating a Fundraising Database for a Non-profit Organization like Villanova.
2. Performing the CRUD operations over the database.
3. Having a simple giving form in the front end where users can make donations.

Entities

For our database, the following are the main entities and their respective attributes that we will begin working on,

- **ACTIVE_STUDENT** - This entity contains all the information related to the current or active students in the university. Ideally, this relationship should be refreshed every semester, and graduated students should be referred to as Alumni. ACTIVE_STUDENT will have the following attributes,
    - Student_ID
    - First_Name
    - Last_Name
    - Email
    - Start_year
    - End_year
    - Parent_Name
    - Department
    - Mobile

- **ALUMNI** - This is a master table that contains the information of all the students who have attended the university. Typically, this entity is updated based on the ACTIVE_STUDENT entity accordingly.
    - Alumni_ID
    - First_Name
    - Last_Name
    - Email
    - Graduation_year
    - Address
    - Mobile
- **PARENTS** - This entity contains the information of all the student's parents. They may be parents of the ACTIVE_STUDENT or ALUMNI. There can be a case like a PARENT now is also an ALUMNI.
    - Parent_ID
    - First_Name
    - Last_Name
    - Child_student_ID
    - Child_Email
    - Address
    - Phone
    - Alumni_status
- **CAMPAIGN** - Different campaigns are run year-long across the university in order to raise funds. These campaigns can be generalized for a common purpose or can be aimed at specific goals. This entity contains all the campaign information that the university conducts.
    - Campaign_ID
    - Campaign_name
    - Campaign_type
    - Start_date
    - End_date
    - Semester
    - Description
- **DONATIONS** - We assume that all parents, alumni, and active students can make a gift. All the donations received for different campaigns from different users are stored in DONATIONS. With this table we can track the total gifts contributed to the university or towards a specific campaign.
    - Transaction_ID
    - Transaction_date
    - First_Name
    - Last_Name
    - Email

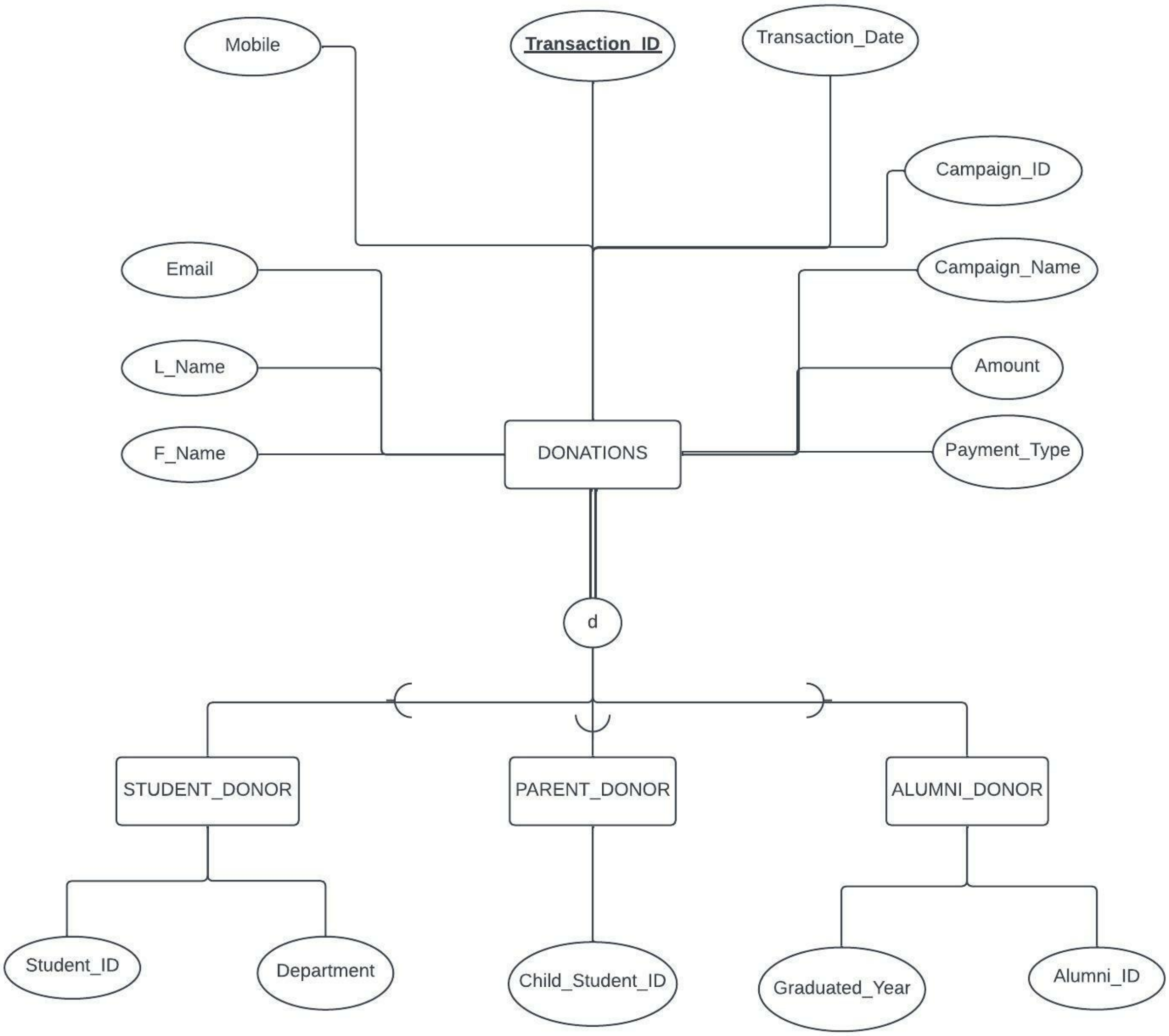
- Applicable ID
- Donor_Status
- Mobile
- Campaign_ID
- Campaign_name
- Amount
- Payment_type

Specifications

- ➔ There can be a case where a user in the PARENT relation who is a parent to the current student and also an Alumni. To identify such cases we have a flag called Alumni_status in PARENT relation in order to map them with ALUMNI relation when needed.
- ➔ The Child_student_ID in PARENT relation helps us to establish a relationship between PARENT and ACTIVE_STUDENT relations. This can also be used to reference ALUMNI relations as there can be cases where a parent can make donations to the university even if their child is not an active student.
- ➔ A single campaign with the same name and ID can be conducted multiple times in a year. So in order to track them individually if needed, we have an attribute called Semester in CAMPAIGN. Assuming that, a campaign can run no more than once across semesters i.e., Spring, Summer & Fall.
- ➔ Campaign_type in CAMPAIGN refers to different types of campaigns taking place in the university like Email, Event-based, Mailing, etc..,

Specialization/Generalization

We can see that the DONATIONS table has an Applicable_ID and Donor_Status that identifies who is the donor. Since all the attributes are the same except for the Donor_Status flag, we are using specialization/generalization concepts here to reduce redundancy and optimize this by saving storage space. We are Specializing this master table into three tables i.e., STUDENT_DONOR, PARENT_DONOR, and ALUMNI_DONOR containing all the transaction details of every donor based on the donor status. After generalization, the schema is,



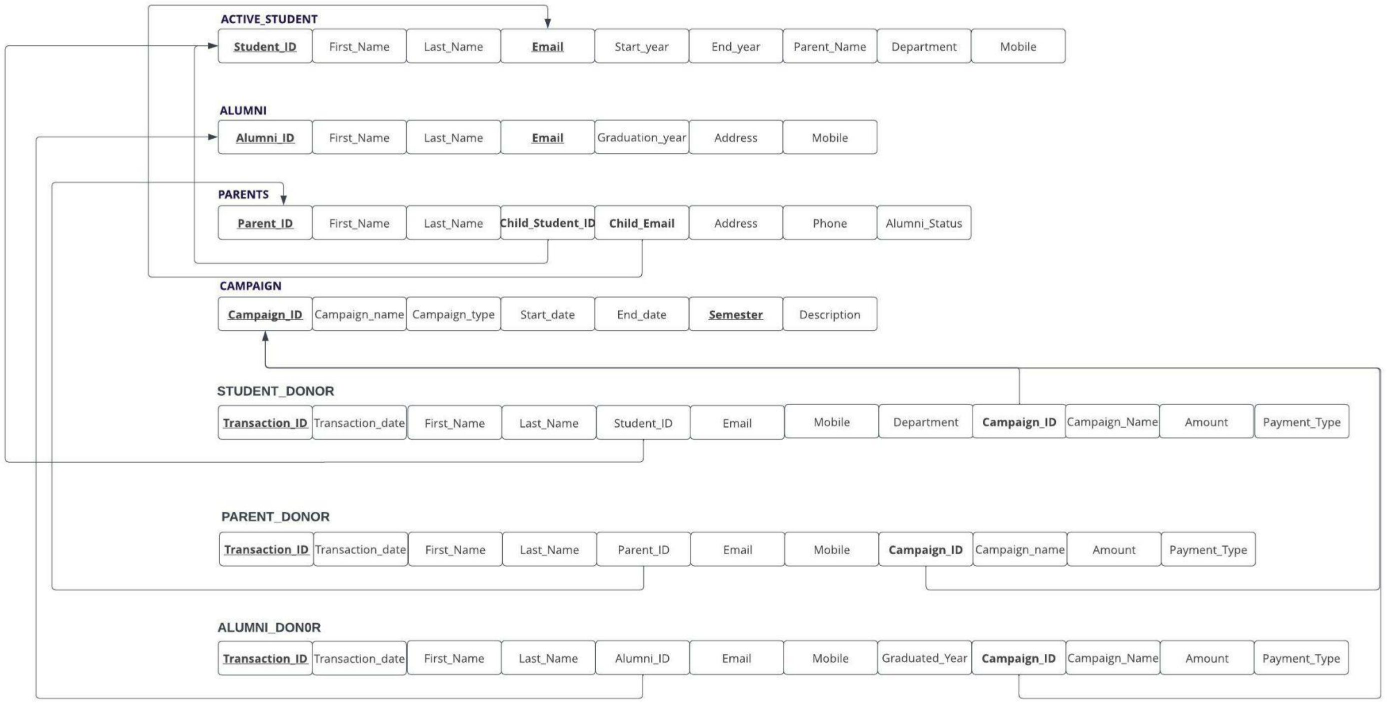
After implementing generalization, the reformed/ new tables will look like this,

.

- **STUDENT_DONOR** - All the students i.e., active students who make any donations to any campaign will be stored in this table.
    - Transaction_ID
    - Transaction_date
    - First_Name
    - Last_Name
    - Student_ID
    - Email
    - Mobile

    - Department
    - Campaign_ID
    - Campaign_name
    - Amount
    - Payment_type
- **PARENT_DONOR** - All the parents of both the active students and alumni who make any donations to any campaign will be stored in this table.
    - Transaction_ID
    - Transaction_date
    - First_Name
    - Last_Name
    - Parent_ID
    - Email
    - Mobile
    - Campaign_ID
    - Campaign_name
    - Amount
    - Payment_type
- **ALUMNI_DONOR** - All the alumni who make any donations to any campaign will be stored in this table.
    - Transaction_ID
    - Transaction_date
    - First_Name
    - Last_Name
    - Alumni_ID
    - Email
    - Mobile
    - Graduated_Year
    - Campaign_ID
    - Campaign_name
    - Amount
    - Payment_type

Referential Integrity Diagram



ACTIVE_STUDENT

Student_ID	First_Name	Last_Name	Email	Start_year	End_year	Parent_Name	Department	Mobile

ALUMNI

Alumni_ID	First_Name	Last_Name	Email	Graduation_year	Address	Mobile

PARENTS

Parent_ID	First_Name	Last_Name	Child_Student_ID	Child_Email	Address	Phone	Alumni_Status

CAMPAIGN

Campaign_ID	Campaign_name	Campaign_type	Start_date	End_date	Semester	Description

STUDENT_DONOR

Transaction_ID	Transaction_date	First_Name	Last_Name	Student_ID	Email	Mobile	Department	Campaign_ID	Campaign_Name	Amount	Payment_Type

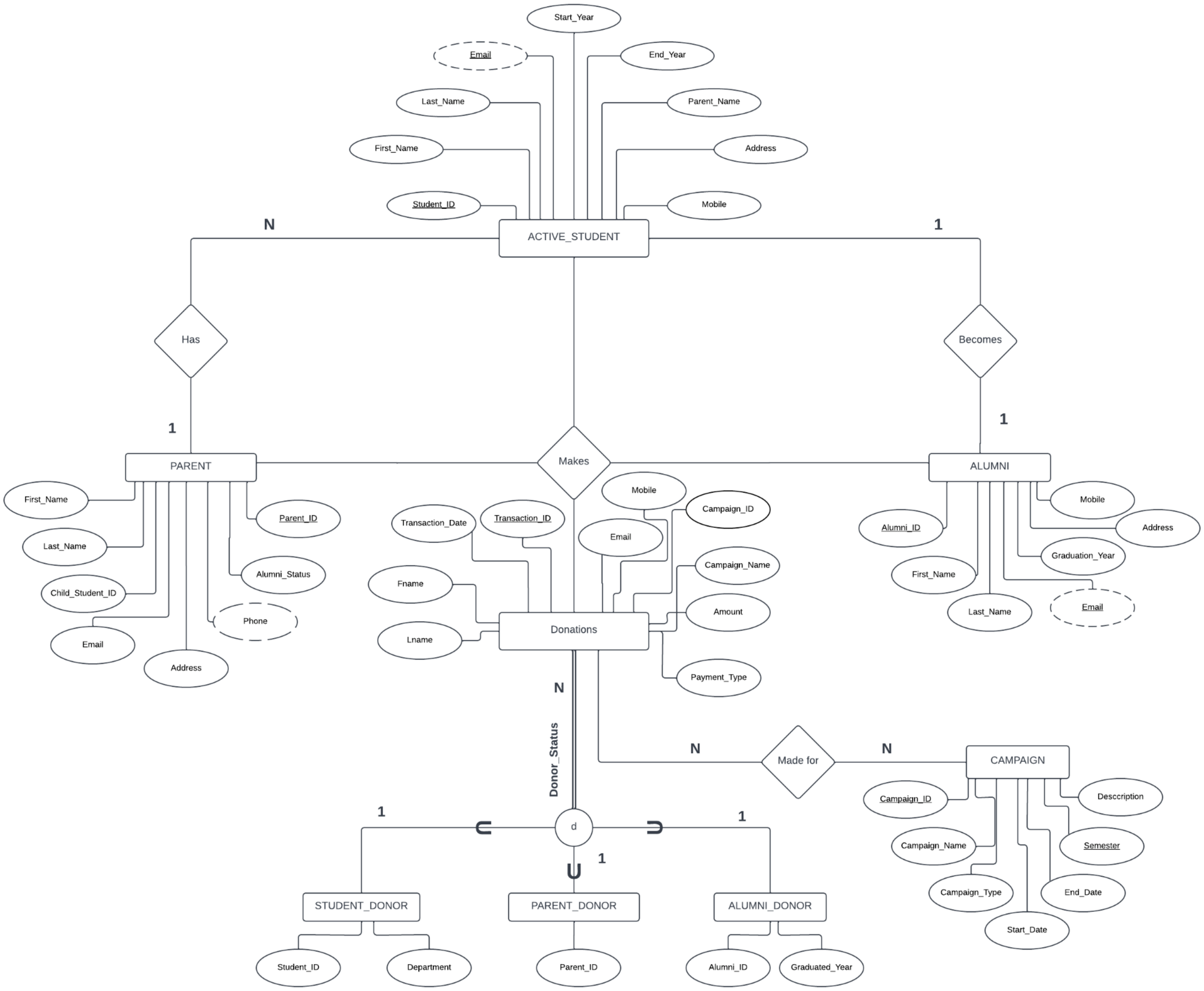
PARENT_DONOR

Transaction_ID	Transaction_date	First_Name	Last_Name	Parent_ID	Email	Mobile	Campaign_ID	Campaign_name	Amount	Payment_Type

ALUMNI_DON0R

Transaction_ID	Transaction_date	First_Name	Last_Name	Alumni_ID	Email	Mobile	Graduated_Year	Campaign_ID	Campaign_Name	Amount	Payment_Type

Enhance Entity Relationship Diagram



Normalization

1 NF

ACTIVE_STUDENT

Student_ID	First_Name	Last_Name	**Email**	Start_year	End_year	Parent_Name	Department	Mobile

ALUMNI

Alumni_ID	First_Name	Last_Name	**Email**	Graduation_year	Address	Mobile

PARENTS

Parent_ID	First_Name	Last_Name	Child_Student_ID	**Child_Email**	Address	Phone	Alumni_Status

STUDENT_DONOR

Transaction_ID	Transaction_date	First_Name	Last_Name	Student_ID	Email	Mobile	Department	**Campaign_ID**	**Campaign_Name**	Amount	Payment_Type

PARENT_DONOR

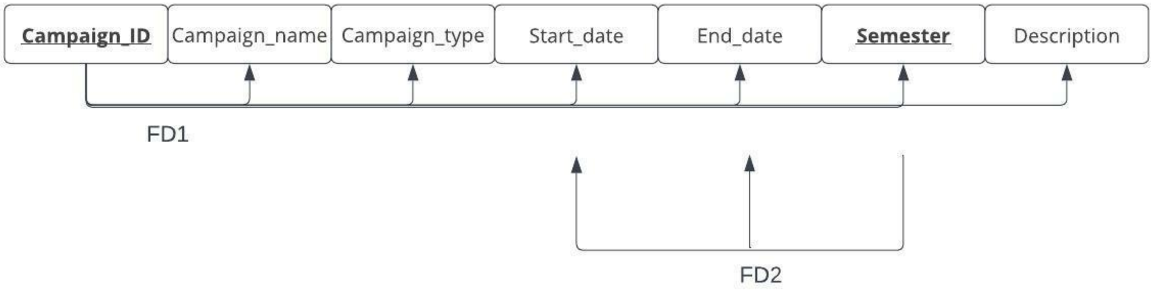
Transaction_ID	Transaction_date	First_Name	Last_Name	Parent_ID	Email	Mobile	**Campaign_ID**	**Campaign_name**	Amount	Payment_Type

ALUMNI_DON0R

Transaction_ID	Transaction_date	First_Name	Last_Name	Alumni_ID	Email	Mobile	Graduated_Year	**Campaign_ID**	**Campaign_Name**	Amount	Payment_Type

2NF

CAMPAIGN



Campaign_ID	Campaign_name	Campaign_type	Description

Sem	**Campaign_ID**	Start_date	End_date

User Stories

- ➢ As Digital Fundraiser, I want to find out which payment type i.e., (Credit Card, Cash, or Apple Pay) has received more donations.
- ➢ As a Digital Fundraiser, I want to view all the campaign information along with their start and end dates
- ➢ As a Digital Fundraiser, I want to find out in which semester did we get more gifts and more money
- ➢ All start dates and end dates for a campaign should match the season/semester. If a new insertion has dates beyond or before the specific season, a trigger should fire
- ➢ When a donor tries to donate to a campaign that is not live i.e. if the transaction date does not lie in between the start and end dates of a campaign. It means the campaign is closed and if a donor tries to donate to closed campaigns, a trigger should fire
- ➢ As a Digital Fundraiser, I want to find out the parents who are alumni and who are also parent to active student. I want their details along with their alumni id.
- ➢ As a Digital Fundraiser, I want to campaign breakdown. How much did the students, parents & alumni donate to each campaign?
- ➢ As a Digital Fundraiser, I want to find out which campaign did the best and which did the worst
- ➢ As a Digital Fundraiser, I want to find out how many dollars students contribute
- ➢ As a Digital Fundraiser, I want to find out how many dollars parents contribute
- ➢ As a Digital Fundraiser, I want to find out how many dollars alumni contribute