McGill | DESAUTELS

Master of Management in Analytics

INSY 661
Databases and Distributed Systems

GROUP PROJECT

"Connecting you through causes that matter"

Contents

PART A

1. Business overview

In recent years, online crowdfunding platforms have risen in popularity as a means for people to raise money, exchange critical resources, and connect with others through participating in meaningful causes. As of 2015, over $34 billion USD in funds have been exchanged worldwide through such "alternative financing" methods[1]. Prominent examples include Kickstarter, an alternative to traditional venture capital whereby creative projects and businesses can receive seed funding from "backers", and GoFundMe, the original crowdfunding platform for supporting all causes from philanthropy, education, to personal causes. As such, there exists a significant market for the development and maintenance of agile database systems which can handle the large amounts of transactions exchanged around the world each day.

The goal of this project is to create a database which will be used to support the storage, manipulation, and handling of data for our company, *Connect the Dots*. Our platform provides a centralized place for individuals, businesses, and organizations to advertise causes they wish to crowdfund for, while connecting potential donors to causes they are passionate about. One key goal of *Connect the Dots* is to allow campaign organizers to reach a wider audience than they would reach through traditional fundraising methods. This is done by using a matching system, whereby potential donors and organizers are matched based on a campaign's category and a donor's preferences categories, while using past donation preference data to expand the donation pool through targeting likely future donors. Another key service provided by *Connect the Dots* is its volunteer campaign and donation options. Campaign organizers looking for non-monetary aid (such as assistance on a community project) can request volunteer hours instead, and donors can donate their time and assistance to a campaign. This function facilitates community building and broadens access by attracting users that may be looking to participate in or commit to campaigns in a non-financial way.

[1]From: Cambridge Judge Business School: Cambridge Centre for Alternative Finance". Jbs.cam.ac.uk.

1.1. Business Features & Functionalities

1. Campaign organizers (individuals/companies) must create user profiles where they can manage campaigns (category, duration, goal amount, timeframe, etc.) and raise funds. Donors (individuals/companies) will create user profiles where they can find and donate to campaigns.
2. Creating an account requires that donors and campaign organizers input financial information in advance to ensure the integrity of transactions made through the platform and to protect from false campaigns and donor commitments.
3. Campaign organizers can pay for additional features, such as being able to have their campaign(s) 'featured' on the website landing page in order of priority. Campaign priority can be determined by our system based on a number of factors, including time left until the end of the campaign and how close a campaign is to its goal.
4. The platform will help match campaigns with potential donors by analyzing their profile, category preferences and donation history.
5. To ensure optimal matching of donor preferences with campaign types, campaigns can list themselves under many different campaign categories (i.e. environment, humanitarian aid, education), while donors can also specify which preferences they are interested in.

1.2. Mission Statement

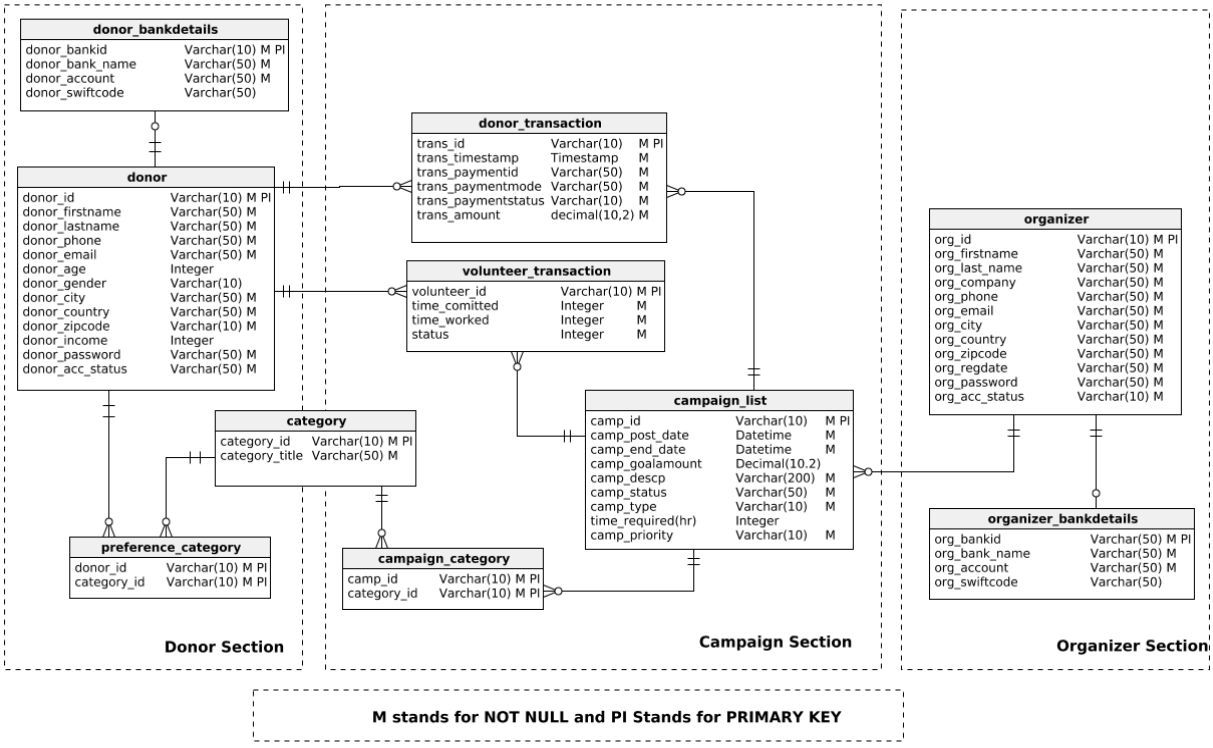
The purpose of the *Connect the Dots* database system is to maintain data that will be used and generated to facilitate the exchange of donations and volunteer hours between donors and campaign organizers. The fundraising platform not only exists as a physical transaction platform, but will capture information on prospective donors, volunteers, and campaign organizers to create a user experience which allows for donors to and campaigns to be meaningfully matched to each other based on their mutually specified categories. Overall, Connect the Dots aims to facilitate the exchange of resources to support personal, business, community, and philanthropic initiatives, and better help connect people from around the world through causes that matter to them.

PART B

2. Entity Relationship Diagram (ERD)

2.1 Assumptions

1. If an individual or organization wants to join the crowdsourcing platform as a donor and an organizer, they must create separate accounts.
2. The organizer may or may not create a campaign on the portal.
3. The donor and organizer may or may not provide their bank details.
4. The donor can donate to more than one campaign.
5. The donor can either donate money or volunteer (depending on the requirements of the campaign).
6. The campaign created by the organizer should have at least one category associated to it.
7. While registering as a donor, the donor may or may not specify the category they prefer to donate to (preferred category).



3. DATA DICTIONARY

3.1 Description of the Entities

Entity Name	Description	Aliases	Occurrence
donor	Contains information needed to identify individuals or organizations registered as campaign donors	donor	One donor may make many transactions (optional). One donor may specify multiple campaign cause categories (optional).
organizer	Contains information needed to identify individuals or organizations registered as campaign organizers	organizer	One organizer can run many campaigns (optional). One organizer can have only one set of bank details (optional).
donor_bankdetails	Contains donor bank information necessary for the donor to donate to campaigns	donor bank details	One donor may have many sets of bank details (optional). One set of bank details must belong to only one organizer (mandatory).
organizer_bankdetails	Contains bank information necessary for the organizer to receive campaign funds	organizer bank details	One organizer may have many sets of bank details (optional). One set of bank details must belong to only one organizer (mandatory).
category	Contains an entire set of campaign categories used to match donor preferences to relevant campaign causes/categories	campaign match	One campaign category may be affiliated with many donor preferences and campaign types (optional). A preference category which has been listed must be affiliated with one donor (mandatory). A campaign category which has been listed must be affiliated with one campaign (mandatory).
preference_category	Contains information the donor and their campaign preferences used to match donor preferences to existing campaign categories.	donor preference	One donor may specify many campaign preferences (optional). Each preference which has been specified must have been done so by one donor (mandatory).

campaign_category	Contains information on the campaign organizer and the campaign category(ies) the organizer has listed itself under used to match campaigns with donors preferences.	organizer campaign category	One campaign organizer may specify many categories its campaign is relevant to (optional). Each category which has been specified must have been done so by one organizer's campaign (mandatory).
campaign_list	Contains information detailing individual fundraising campaigns	available campaigns	One campaign may have many donation transactions made to it (optional). One campaign may belong to many categories but must belong to at least one (mandatory). One campaign can only belong to one organizer (mandatory).
donation_transaction	Contains financial data on transactions made from donors to campaigns	monetary transaction	One donation transaction may have been made to one campaign (mandatory). One donation transaction may only have been made by one donor (optional).
volunteer_transaction	Contains information on a non-monetary, volunteer-based transaction, including volunteer hours committed and completed from the donor to the campaign and the status of the "transaction".	volunteer work transaction	One donor may commit to and complete many volunteer transactions (optional). Each volunteer transaction must belong to one donor (mandatory). One campaign may receive many volunteer transactions (optional). One volunteer transaction can only be given to one campaign (mandatory).

3.2 Description of the Attributes

Entity Name	Attributes	Description	Data Type	Nulls	Multi-valued	Derived	Default
donor	**donor_id**	unique id for each donor	VARCHAR (10)	NO	NO	NO	NONE
	donor_firstname	first name of donor	VARCHAR (50)	NO	NO	NO	NONE
	donor_lastname	last name of donor	VARCHAR (50)	NO	NO	NO	NONE

	donor_phone	phone number	VARCHAR (50)	NO	NO	NO	NONE
	donor_email	donor email address	VARCHAR (50)	NO	NO	NO	NONE
	donor_age	donor age	INT	YES	NO	NO	NONE
	donor_gender	donor gender	VARCHAR (10)	YES	NO	NO	NONE
	donor_city	city of location	VARCHAR (50)	NO	NO	NO	NONE
	donor_country	country of location	VARCHAR (50)	NO	NO	NO	NONE
	donor_zipcode	zip code	VARCHAR (50)	NO	NO	NO	NONE
	donor_income	income of donor	INT	YES	NO	NO	NONE
	donor_password	password for login	VARCHAR (50)	NO	NO	NO	NONE
	donor_account_status	donor account status	VARCHAR (50)	NO	NO	NO	ACTIVE
donor_transaction	**trans_id**	unique id for each transaction	VARCHAR (10)	NO	NO	NO	NONE
	trans_timestamp	date and time when user made transaction	TIMESTAMP	NO	NO	NO	NONE
	trans_paymentid	unique payment id	VARCHAR (50)	NO	NO	NO	NONE
	trans_paymentmode	transaction methods Visa	VARCHAR (50)	NO	NO	NO	NONE
	trans_paymentstatus	status of payment (pending/completed/canceled)	VARCHAR (10)	NO	NO	NO	NONE
	trans_amount	amount paid by donor	DECIMAL(10,2)	NO	NO	NO	NONE
category	**category_id**	unique id for each category	VARCHAR (10)	NO	NO	NO	NONE

	category_title	names of category	VARCHAR (50)	NO	NO	NO	NONE
preference_category	**donor_id**	unique id for each donor	VARCHAR (10)	NO	NO	NO	NONE
	category_id	id for each category	VARCHAR (10)	NO	NO	NO	NONE
campaign_category	**camp_id**	unique id for each campaign organizer	VARCHAR (10)	NO	NO	NO	NONE
	category_id	id for each category	VARCHAR (10)	NO	NO	NO	NONE
campaign_list	**camp_id**	unique id for campaign	VARCHAR (10)	NO	NO	NO	NONE
	camp_post_date	date campaign posted	DATETIME	NO	NO	NO	NONE
	camp_end_date	date campaign ends	DATETIME	NO	NO	NO	NONE
	camp_descp	description of the campaign	VARCHAR (200)	NO	NO	NO	NONE
	camp_status	status of campaign (on-going/ended/plan)	VARCHAR (50)	NO	NO	NO	NONE
	camp_type	type of campaign	VARCHAR (10)	NO	NO	NO	NONE
	camp_goalamount	target amount of campaign	DECIMAL(10,2)	YES	NO	NO	NONE
	time_required(hr)	amount of time the campaign will be listed for	INT	YES	NO	NO	NONE
	camp_priority	low/medium/high priority options	VARCHAR (50)	NO	NO	NO	NONE
organizer	**org_id**	unique id for organizer	VARCHAR (10)	NO	NO	NO	NONE
	org_firstname	first name of organizer	VARCHAR (50)	NO	NO	NO	NONE
	org_last_name	last name of	VARCHAR	NO	NO	NO	NONE

		organizer	(50)				
	org_company	company name of organizer	VARCHAR (50)	NO	NO	NO	NONE
	org_phone	phone number	LONGINT				NONE
	org_email	email for organizer	VARCHAR (50)	NO	NO	NO	NONE
	org_city	located city of organizer	VARCHAR (50)	NO	NO	NO	NONE
	org_country	state of organizer	VARCHAR (50)	NO	NO	NO	NONE
	org_zipcode	zip code of organizer	VARCHAR (50)	NO	NO	NO	NONE
	org_regdate	registration date of organizer	VARCHAR (50)	NO	NO	NO	NONE
	org_password	organizer password	VARCHAR (50)	NO	NO	NO	NONE
	org_acc_status	organizer account status	VARCHAR (10)	NO	NO	NO	ACTIVE
organizer_bankdetails	**org_bankid**	unique id defined by us	VARCHAR (50)	NO	NO	NO	NONE
	org_bank_name	name of the bank	VARCHAR (50)	NO	NO	NO	NONE
	org_account	organizer's account number	VARCHAR (50)	NO	NO	NO	NONE
	org_swiftcode	SWIFT code for bank	VARCHAR (50)	YES	NO	NO	NONE
donor_bankdetails	**donor_bankid**	unique id defined by us	VARCHAR (50)	NO	NO	NO	NONE
	donor_bankname	name of bank	VARCHAR (50)	NO	NO	NO	NONE
	donor_account	donor's account number	VARCHAR (50)	NO	NO	NO	NONE
	donor_swiftcode	SWIFT code for bank	VARCHAR (50)	YES	NO	NO	NONE

4. Relational Schema

__bold__ indicates primary key
highlight indicates foreign key

donor(__donor_id__, donor_firstname, donor_lastname, donor_phone, donor_email, donor_age, donor_gender, donor_city, donor_country, donor_zipcode, donor_income, donor_password, donor_account_status)
PRIMARY KEY: donor_id

organizer(__org_id__, org_firstname, org_last_name, org_company, org_phone, org_email, org_city, org_country, org_zipcode, org_regdate, org_password, org_acc_status)
PRIMARY KEY: org_id

campaign_list(__camp_id__, camp_post_date, camp_end_date, camp_descp, camp_status, camp_type, camp_goalamount, time_required, camp_priority, org_id)
PRIMARY KEY: camp_id
FOREIGN KEY: org_id REFERENCES organizer (org_id)

donor_bankdetails(__donor_bankid__, donor_bankname, donor_account, donor_swiftcode, donor_id)
PRIMARY KEY: donor_bankid
FOREIGN KEY: donor_id REFERENCES donor (donor_id)

organizer_bankdetails(__org_bankid__, org_bank_name, org_account, org_swiftcode, org_id)
PRIMARY KEY: org_bankid
FOREIGN KEY: org_id REFERENCES organizer (org_id)

category(__category_id__, category_title)
PRIMARY KEY: category_id

campaign_category (__camp_id, category_id__,)
PRIMARY KEY: (camp_id, category_id)
FOREIGN KEY: camp_id REFERENCES campaign_list (camp_id)
FOREIGN KEY: category_id REFERENCES category (category_id)

volunteer_transaction(__volunteer_id__, time_comitted, time_worked, volunteer_status, donor_id, camp_id)
PRIMARY KEY: volunteer_id
FOREIGN KEY: camp_id REFERENCES campaign (camp_id)
FOREIGN KEY: donor_id REFERENCES donor (donor_id)

donor_transaction(**trans_id**, trans_timestamp, trans_paymentid, trans_paymentmode, trans_paymentstatus, trans_amount, donor_id, camp_id)
PRIMARY KEY: trans_id
FOREIGN KEY: donor_id REFERENCES donor (donor_id)
FOREIGN KEY: camp_id REFERENCES campaign_list (camp_id)

preference_category (**donor_id, category_id**)
PRIMARY KEY: (donor_id, category_id)
FOREIGN KEY: donor_id REFERENCES donor (donor_id)
FOREIGN KEY: category_id REFERENCES category (category_id)

PART C

1. DDL Statements

```
/* CREATE DONOR TABLE
CONTAINS INFORMATION NEEDED TO IDENTIFY INDIVIDUALS OR
ORGANIZATIONS REGISTERED AS CAMPAIGN DONORS */

CREATE TABLE IF NOT EXISTS donor
  (
     donor_id             VARCHAR(10) NOT NULL,
     donor_firstname      VARCHAR(50) NOT NULL,
     donor_lastname       VARCHAR(50) NOT NULL,
     donor_phone          VARCHAR(50) NOT NULL,
     donor_email          VARCHAR(50) NOT NULL,
     donor_age            INT,
     donor_gender         VARCHAR(10),
     donor_city           VARCHAR(50) NOT NULL,
     donor_country        VARCHAR(50) NOT NULL,
     donor_zipcode        VARCHAR(50) NOT NULL,
     donor_income         INT NULL,
     donor_password       VARCHAR(50) NOT NULL,
     donor_account_status VARCHAR(50) NOT NULL,
     PRIMARY KEY (`donor_id`)
  );



/* CREATE ORGANIZER TABLE
CONTAINS INFORMATION NEEDED TO IDENTIFY INDIVIDUALS OR
ORGANIZATIONS REGISTERED AS CAMPAIGN ORGANIZERS*/

CREATE TABLE IF NOT EXISTS organizer
  (
     org_id          VARCHAR(10) NOT NULL,
     org_firstname   VARCHAR(50) NOT NULL,
     org_last_name   VARCHAR(50) NOT NULL,
     org_company     VARCHAR(50) NOT NULL,
     org_phone       VARCHAR(50) NOT NULL,
     org_email       VARCHAR(50) NOT NULL,
     org_city        VARCHAR(50) NOT NULL,
     org_country     VARCHAR(50) NOT NULL,
```

```
    org_zipcode     VARCHAR(50) NOT NULL,
    org_regdate     DATE NOT NULL,
    org_password    VARCHAR(100) NOT NULL,
    org_acc_status  INTEGER NOT NULL,
    PRIMARY KEY (`org_id`)
  );
```

```
/* CREATE CAMPAIGN LIST TABLE
CONTAINS INFORMATION DETAILING INDIVIDUAL FUNDRAISING CAMPAIGNS*/

CREATE TABLE IF NOT EXISTS campaign_list
  (
    camp_id          VARCHAR(10) NOT NULL,
    camp_post_date   DATE NOT NULL,
    camp_end_date    DATE NOT NULL,
    camp_descp       VARCHAR(200) NOT NULL,
    camp_status      INT NOT NULL,
    camp_type        VARCHAR(50) NOT NULL,
    camp_goalamount  INT NULL,
    time_required    INT NULL,
    camp_priority    VARCHAR(50) NOT NULL,
    org_id           VARCHAR(10),
    PRIMARY KEY (`camp_id`),
    FOREIGN KEY (`org_id`) REFERENCES organizer (org_id)
  );
```

```
/* CREATE DONOR BANK DETAILS TABLE
CONTAINS BANK INFORMATION NECESSARY FOR THE DONOR TO PAY
DONATIONS TO CAMPAIGNS*/

CREATE TABLE IF NOT EXISTS `donor_bankdetails`
  (
    `donor_bankid`     VARCHAR(10) NOT NULL,
    `donor_bankname`   VARCHAR(50) NOT NULL,
    `donor_account`    VARCHAR(50) NOT NULL,
    `donor_swiftcode`  VARCHAR(50),
    `donor_id`         VARCHAR(10) NOT NULL,
    PRIMARY KEY (`donor_bankid`),
    FOREIGN KEY (`donor_id`) REFERENCES donor(donor_id)
  );
```

```
/* CREATE ORGANIZER BANK DETAILS TABLE
CONTAINS BANK INFORMATION NECESSARY FOR THE ORGANIZER TO RECEIVE
CAMPAIGN FUNDS */
```

```
CREATE TABLE IF NOT EXISTS organizer_bankdetails
  (
     org_bankid    VARCHAR(10),
     org_bank_name VARCHAR(50),
     org_account   VARCHAR(50) NULL,
     org_swiftcode VARCHAR(50),
     org_id        VARCHAR(10),
     PRIMARY KEY (`org_bankid`),
     FOREIGN KEY (`org_id`) REFERENCES organizer(org_id)
  );



/* CREATE CATEGORY TABLE
CONTAINS DIFFERENT CATEGORIES TO WHICH CAMPAIGNS BELONG */

CREATE TABLE `category`
  (
     `category_id`    VARCHAR(10) NOT NULL,
     `category_title` VARCHAR(50) NOT NULL,
     PRIMARY KEY (`category_id`)
  );



/* CREATE CAMPAIGN CATEGORY TABLE
MATCHES THE CAMPAIGN WITH THE CATEGORIES IT BELONGS TO */

CREATE TABLE `campaign_category`
  (
     `camp_id`     VARCHAR(10) NOT NULL,
     `category_id` VARCHAR(10) NOT NULL,
     PRIMARY KEY (`camp_id`, `category_id`),
     FOREIGN KEY (`camp_id`) REFERENCES campaign_list
(`camp_id`),
     FOREIGN KEY (`category_id`) REFERENCES category
(`category_id`)
  );



/* CREATE VOLUNTEER TRANSACTION TABLE
CONTAINS DATA ON VOLUNTEER TRANSACTIONS MADE BY DONORS TO
VOLUNTEER CAMPAIGNS */

CREATE TABLE IF NOT EXISTS volunteer_transaction
  (
     volunteer_id     VARCHAR(10) NOT NULL,
     time_committed   INTEGER NOT NULL,
```

```
    time_worked      INTEGER NOT NULL,
    volunteer_status INTEGER NOT NULL,
    donor_id         VARCHAR(10) NOT NULL,
    camp_id          VARCHAR(10) NOT NULL,
    PRIMARY KEY (`volunteer_id`),
    FOREIGN KEY (`camp_id`) REFERENCES campaign_list
(`camp_id`),
    FOREIGN KEY (`donor_id`) REFERENCES donor (`donor_id`)
  );


/* CREATE  DONOR TRANSACTION TABLE
CONTAINS FINANCIAL DATA ON TRANSACTIONS MADE FROM DONORS TO
CAMPAIGNS*/

CREATE TABLE IF NOT EXISTS donor_transaction
  (
    trans_id            VARCHAR(10) NOT NULL,
    trans_timestamp     TIMESTAMP NOT NULL,
    trans_paymentid     VARCHAR(50) NOT NULL,
    trans_paymentmode   VARCHAR(50) NOT NULL,
    trans_paymentstatus VARCHAR(50) NOT NULL,
    trans_amount        DECIMAL(10, 2) NOT NULL,
    camp_id             VARCHAR(10) NOT NULL,
    donor_id            VARCHAR(10) NOT NULL,
    PRIMARY KEY (`trans_id`),
    FOREIGN KEY (`camp_id`) REFERENCES campaign_list
(`camp_id`),
    FOREIGN KEY (`donor_id`) REFERENCES donor (`donor_id`)
  );


/* CREATE PREFERENCE CATEGORY TABLE
 MATCHES THE DONOR WITH THE PREFERRED CATEGORIES HE/SHE HAS
CHOSEN*/

CREATE TABLE IF NOT EXISTS preference_category
  (
    category_id VARCHAR(10) NOT NULL,
    donor_id    VARCHAR(10) NOT NULL,
    PRIMARY KEY (`donor_id`, `category_id`),
    FOREIGN KEY (`donor_id`) REFERENCES donor (`donor_id`),
    FOREIGN KEY (`category_id`) REFERENCES category
(`category_id`)
  );
```

2. DML Statements

**Please see attached SQL file for DML statements*

PART D

LIST OF QUERIES

QUERY 1: Campaigns and categories represented
QUERY 2: Total funds and volunteer hours processed through the database
QUERY 3: Registered users vs. platform utilization
QUERY 4: Information on donors contributing both money and volunteer time to a campaign
QUERY 5: Donor campaign category preferences
QUERY 6: Donor campaign preferences: 40-60 age bracket
QUERY 7: Average amount of funds donated to campaigns by category
QUERY 8: Goal completion percentage for volunteer campaigns
QUERY 9: Most popular donor transaction payment method
QUERY 10: Most commonly listed campaign category
QUERY 11: Which gender (male or female) has volunteered more time?
QUERY 12: Name of donor who has donated the highest amount of money in total
QUERY 13: List of donors who completed donations transactions of a total value higher than the average amount donated across all transactions in our database
QUERY 14: List of all active campaigns along with the number of donors whose category preference matches the category to which the campaign belongs
QUERY 15: Time remaining before a campaign's closing date
QUERY 16: Estimation of number of additional volunteers needed for a campaign to reach its required time goal
QUERY 17: Estimation of donors needed to reach campaign target
QUERY 18: Summary of total number of volunteers and total sum of money donated per country
QUERY 19: Comparison of total amount of money donated vs. total hours volunteered for each donor
QUERY 20: Change in donor category preferences as income increases
QUERY 21: Campaign category that is most preferred by donors and the number of users who selected the category as a preference
QUERY 22: Setting campaign priority levels based on number of days left till campaign ends
QUERY 23: Match donors to new campaigns based on past preferences

***

QUERY 1: Campaigns and categories represented

We begin by obtaining a broad overview of the Connect the Dots database. Since the core function of this database is to connect donors and organizers through campaigns, we began the analysis by examining the variety of campaigns available and the categories under which they fall. We find that campaigns belong to variety of different categories, including Arts & Culture, Environment, and Education.

```
SELECT CL.camp_descp,
       C.category_title
FROM   category C,
       campaign_list CL,
       campaign_category
WHERE  C.category_id = campaign_category.category_id
       AND campaign_category.camp_id = CL.camp_id;
```

CAMP_DESCP	CATEGORY_TITLE
Buy laptops for underfunded schools	Arts & Culture
Help support Sarah's college education	Education
Build a playground for a local school	Education
Buy laptops for underfunded schools	Education
Help fund the Galapagos Tortoise breeding prog...	Environment
Join the community trash pickup	Environment
Fund Dan's kidney transplant surgery	Health & Medical
Assist with covid-19 testing	Health & Medical
Help fund the Galapagos Tortoise breeding prog...	Health & Medical
Donate to the World Health Organization's fund...	Health & Medical

QUERY 2: Total funds and volunteer hours processed through the database

Continuing on with the broad overview, it is useful to find out how much money and how many hours have been processed through the database already. This will give a sense of how much value has been delivered to donors and campaign organizers thus far. We find that in total, over $2.4 million dollars has been donated through Connect the Dots, and over 3000 hours have been volunteered.

```
SELECT Sum(D.trans_amount) AS TOTAL_DONATED,
       Sum(V.time_worked)  AS TOTAL_VOLUNTEERED_HRS
FROM   donor_transaction D,
       volunteer_transaction V;
```

TOTAL_DONATED	TOTAL_VOLUNTEERED_HRS
2405246.86	3151

QUERY 3: Registered users vs. platform utilization

It is valuable to know how the platform is being utilized -- namely, how many of the donors with an account have yet to make a monetary donation. Fortunately, we find that only one donor with an account has not yet made a transaction.

```
SELECT donor.donor_id
FROM   donor
WHERE  NOT EXISTS(SELECT *
                  FROM   donor_transaction
                  WHERE  donor.donor_id =
donor_transaction.donor_id);
```

	DONOR_ID
▶	D009

QUERY 4: Information on donors contributing both money and volunteer time to a campaign

It's possible that a donor might not only want to donate time or money to a specific campaign, but might want to contribute both. We examined which donors fit this criteria, and found that two donors contributed both time and money to help build a playground for a local school.

```
SELECT T4.donor_id,
       T4.donor_firstname,
       T4.donor_lastname,
       T4.donor_age,
       T4.donor_gender,
       T4.donor_country,
       T3.camp_id,
       T3.camp_descp,
       T3.camp_type
FROM   volunteer_transaction T1
       JOIN donor_transaction T2
         ON T1.donor_id = T2.donor_id
       JOIN campaign_list T3
         ON T1.camp_id = T3.camp_id
       JOIN donor T4
         ON T1.donor_id = T4.donor_id
WHERE  T1.camp_id = T2.camp_id;
```

	DONOR_ID	DONOR_FIRSTNAME	DONOR_LASTNAME	DONOR_AGE	DONOR_GENDER	DONOR_COUNTRY	CAMP_ID	CAMP_DESCP	CAMP_TYPE
▶	D005	Clim	Risborough	80	M	Bahamas	C005	Build a playground for a local school	both
	D010	Debora	Richemond	49	F	South Korea	C005	Build a playground for a local school	both

QUERY 5: Donor campaign category preferences

A key function of the Connect the Dots crowdfunding platform is its ability to connect donors to campaigns through matching donor preferences to campaign categories. As such, we examined which campaign categories donors preferred to get an overview of donor preferences. We found that frequently, donors specified multiple preferences.

```
SELECT DISTINCT T1.donor_id,
                T4.donor_firstname,
                T4.donor_lastname,
                T3.category_id,
                T3.category_title
FROM   preference_category T1
       JOIN campaign_category T2
         ON T1.category_id = T2.category_id
       JOIN category T3
         ON T1.category_id = T3.category_id
       JOIN donor T4
         ON T1.donor_id = T4.donor_id
ORDER  BY T1.donor_id;
```

DONOR_ID	DONOR_FIRSTNAME	DONOR_LASTNAME	CATEGORY_ID	CATEGORY_TITLE
D003	Michal	Matitiaho	CAT013	Animal and Humane
D004	Valida	Wilshaw	CAT006	Human Services
D004	Valida	Wilshaw	CAT013	Animal and Humane
D005	Clim	Risborough	CAT007	Golbal
D005	Clim	Risborough	CAT013	Animal and Humane
D006	Bryn	Newbold	CAT011	Miscellaneous
D007	Betsy	Will	CAT007	Golbal
D007	Betsy	Will	CAT008	Social Action
D008	Griswold	McKelvey	CAT009	Community
D008	Griswold	McKelvey	CAT010	Faith and Mission

QUERY 6: Donor campaign preferences: 40-60 age bracket

If we wanted to examine donor preferences on a deeper level, segmenting by age demographic can give us more insights into which ages prefer what. Here, we selected donor preferences for donors aged 40-60.

```
SELECT T2.category_title
FROM   campaign_category T1
       JOIN category T2
         ON T1.category_id = T2.category_id
WHERE  camp_id IN (SELECT DISTINCT camp_id
                   FROM   donor_transaction
                   WHERE  donor_id IN (SELECT donor_id
                                       FROM   donor
```

```
                                        WHERE  donor_age > 40
                                               AND donor_age <60))
GROUP  BY T2.category_title;
```

	CATEGORY_TITLE
▶	Education
	Environment
	Health & Medical
	Human Services
	Golbal
	Social Action
	Community
	Faith and Mission
	Animal and Humane

QUERY 7: Average amount of funds donated to campaigns by category

Are certain campaign categories attract higher donation amounts? We examined the average donation while grouping by each campaign category and found that overall, environmental campaign causes drew the highest donations, averaging $25,000 dollars.

```
SELECT T3.*,
       T4.avg_donation
FROM   category T3
       JOIN (SELECT T1.category_id,
                    Avg(total_donation) AS AVG_DONATION
             FROM   campaign_category T1
                    RIGHT JOIN (SELECT camp_id,
                                       Sum(trans_amount) AS
TOTAL_DONATION
                                FROM   donor_transaction
                                GROUP  BY camp_id) T2
                            ON T1.camp_id = T2.camp_id
             GROUP  BY category_id) T4
         ON T3.category_id = T4.category_id
ORDER  BY T4.avg_donation DESC;
```

	CATEGORY_ID	CATEGORY_TITLE	AVG_DONATION
▶	CAT001	Arts & Culture	7851.260000
	CAT002	Education	10950.420000
	CAT003	Environment	25139.490000
	CAT004	Health & Medical	16108.362000
	CAT005	Active Duty & Veterans	6378.190000
	CAT006	Human Services	13097.680000
	CAT007	Golbal	16162.370000
	CAT008	Social Action	14219.263333
	CAT009	Community	11226.900000
	CAT010	Faith and Mission	12661.810000

QUERY 8: Goal completion percentage for volunteer campaigns

In order to display on the interface how close campaigns are to reaching their goal, it can be useful to calculate the ratio of time completed to time required for each campaign. Here, represented progress towards campaign goal completion as a percentage, and found that out of all the volunteers, none were yet completed. The closest campaign to completion was a call for COVID testing volunteers, at 98%.

```
SELECT T1.*,
       T2.time_required,
       Concat(( time_completed / T2.time_required * 100 ), '%')
AS
       PERCENT_FINISHED,
       T2.camp_descp
FROM   (SELECT camp_id,
               Sum(time_worked) AS TIME_COMPLETED
        FROM   volunteer_transaction
        GROUP  BY camp_id) T1
       LEFT JOIN campaign_list T2
              ON T1.camp_id = T2.camp_id
ORDER  BY T1.camp_id;
```

CAMP_ID	TIME_COMPLETED	TIME_REQUIRED	PERCENT_FINISHED	CAMP_DESCP
C003	52	53	98.1132%	Assist with covid-19 testing
C005	46	150	30.6667%	Build a playground for a local school
C008	21	99	21.2121%	Help Annie rebuild her home
C011	5	100	5.0000%	Join the community trash pickup
C013	10	200	5.0000%	Volunteer at the Dawson's community center
C015	3	100	3.0000%	Help send aid and resources to those affected ...

QUERY 9: Most popular donor transaction payment method

What payment method is most popular amongst donors? Knowing this information might be used as rationale to create new platform functionalities such as an easy checkout option. We find that the most popular checkout methods are Mastercard (26% of transactions), followed by Visa (~22% of transactions).

```
CREATE temporary TABLE payment_mode
  SELECT trans_paymentmode,
         Count(trans_paymentmode) AS NUMBER_OF_USER
  FROM   donor_transaction
  GROUP  BY trans_paymentmode
  ORDER  BY number_of_user DESC;

SELECT trans_paymentmode, NUMBER_OF_USER, Round(number_of_user /
23 * 100, 2) AS percent
```

```
FROM payment_mode;
```

	trans_paymentmode	NUMBER_OF_USER	percent
▶	mastercard	6	26.09
	visa	5	21.74
	jcb	3	13.04
	diners-club	3	13.04
	unionpay	2	8.70
	visa-electron	1	4.35

QUERY 10: Most commonly listed campaign category

Which campaign category is most selected by campaign organizers? By selecting the number of campaigns and grouping them by the categories they identify under, we find that the highest number of campaigns have identified themselves under the "community" category.

```
SELECT category_id,
       category_title,
       Max(number_of_campaigns) AS MAX_CAMPAIGN_NUMBER
FROM   (SELECT C.category_id,
               C.category_title,
               Count(CC.camp_id) AS NUMBER_OF_CAMPAIGNS
        FROM   category AS C
               INNER JOIN campaign_category AS CC
                       ON C.category_id = CC.category_id
        GROUP  BY C.category_id
        ORDER  BY number_of_campaigns DESC) AS
NUMBER_OF_CAMPAIGNS_TABLE;
```

	CATEGORY_ID	CATEGORY_TITLE	MAX_CAMPAIGN_NUMBER
▶	CAT009	Community	8

QUERY 11: Which gender (male or female) has volunteered more time?

We are interested in seeing whether a particular gender (male or female) donates more time (volunteering) than the other gender. Our data shows that Men have volunteered more hours than Women. Overall, Men have volunteered a total of 93 hours, while Women have volunteered a total of 59 hours.

```
SELECT donor_gender,
       Sum(time_committed) AS total_time_committed
FROM   donor D
```

```
        INNER JOIN volunteer_transaction V
                ON D.donor_id = V.donor_id
GROUP   BY donor_gender
ORDER   BY total_time_committed DESC
```

donor_gender	total_time_committed
M	93
F	59

QUERY 12: Name of the donor who has donated the highest amount of money in total

Donors who come across our platform tend to be very generous. We are curious to see which donor has donated the largest overall sum of money since they registered with our platform. Clim Risborough has donated $26,343.04!

```
SELECT donor_id,
       donor_firstname,
       donor_lastname,
       Max(total_transaction_amount) AS MAX_AMOUNT
FROM   (SELECT DT.donor_id,
               D.donor_firstname,
               D.donor_lastname,
               Sum(DT.trans_amount) AS TOTAL_TRANSACTION_AMOUNT
        FROM   donor_transaction AS DT
               INNER JOIN donor AS D
                       ON DT.donor_id = D.donor_id
        WHERE  DT.trans_paymentstatus = 'COMPLETED'
        GROUP  BY DT.donor_id
        ORDER  BY total_transaction_amount DESC) AS
       TOTAL_TRANSACTION_AMOUNT_TABLE;
```

DONOR_ID	DONOR_FIRSTNAME	DONOR_LASTNAME	MAX_AMOUNT
D005	Clim	Risborough	26343.04

QUERY 13: List of donors who completed donations transactions of a total value higher than the average amount donated across all transactions

We have calculated that the average amount of money that has been donated by our donors is $5,012.84. This query will filter and order the data to show us a list of the donors who have donated more than this average. We may be able to use this information to send thank you cards to our donors for their generosity.

```
SELECT donor.donor_id,
       Concat(donor_firstname, ' ', donor_lastname) AS
DONOR_NAME,
       Sum(trans_amount) AS SUM_AMOUNT,
       (SELECT Avg(trans_amount)
        FROM   donor_transaction
        WHERE  trans_paymentstatus = 'COMPLETED')AS
AVERAGE_AMOUNT
FROM   donor_transaction,
       donor
WHERE  donor.donor_id = donor_transaction.donor_id
       AND trans_paymentstatus = 'COMPLETED'
GROUP  BY donor_transaction.donor_id
HAVING Sum(trans_amount) > average_amount
ORDER  BY sum_amount DESC;
```

DONOR_ID	DONOR_NAME	SUM_AMOUNT	AVERAGE_AMOUNT
D005	Clim Risborough	26343.04	5012.838000
D010	Debora Richemond	19500.00	5012.838000
D006	Bryn Newbold	15972.36	5012.838000
D008	Griswold McKelvey	12923.68	5012.838000
D002	Palm Worsfield	8564.25	5012.838000
D007	Betsy Will	8075.24	5012.838000

QUERY 14: List of all the active campaigns along with the number of donors whose category preference matches the category to which the campaign belongs

Our overarching aim is to match campaigns to the right donors according to their category preferences. This query will show us how many donors would likely be interested in donating to a certain campaign based on if the category(ies) the campaign is classified in matches the category preference(s) of a donor. This information will help us gauge how much traffic a certain campaign may be getting. For example, campaigns C011 and C012 only have 1 match, so they may not receive as many donations as C005 which has 7 donor matches.

```
SELECT campaign_list.camp_id,
       camp_descp,
```

```
       Count(DISTINCT donor_id) AS NUMBER_OF_PREFERENCE
FROM   campaign_list,
       campaign_category,
       preference_category
WHERE  campaign_category.category_id =
preference_category.category_id
       AND campaign_list.camp_id = campaign_category.camp_id
GROUP  BY campaign_list.camp_id
ORDER  BY campaign_list.camp_id ASC;
```

CAMP_ID	CAMP_DESCP	NUMBER_OF_PREFERENCE
C001	Help support Sarah's college education	5
C002	Fund Dan's kidney transplant surgery	3
C003	Assist with covid-19 testing	5
C004	Help fundraise medical supplies for a new comm...	4
C005	Build a playground for a local school	7
C006	Help fund the Galapagos Tortoise breeding prog...	2
C007	Donate to the World Health Organization's fund...	2
C008	Help Annie rebuild her home	4
C009	Buy laptops for underfunded schools	2
C010	Help fund Karen's rehabilitation therapy	3
C011	Join the community trash pickup	1
C012	Buy prosthetics for a military veteran	1
C013	Volunteer at the Dawson's community center	4
C014	Help John overcome cystic fibrosis	4

***QUERY 15*: Time remaining before a campaign's closing date**

This query calculates the total amount of time remaining for a campaign before it closes, based on the difference between the current date and the campaign's indicated end date. We will use this information to dictate which campaigns we boost on our homepage. For example, C014 has only 13 days left, so we might boost it with more urgency that C003 which has 107 days left.

```
SELECT camp_id,
       camp_descp,
       camp_type,
       Datediff(camp_end_date, Sysdate()) AS DAYS
FROM   campaign_list
WHERE  camp_status = 1
ORDER  BY days DESC;
```

CAMP_ID	CAMP_DESCP	CAMP_TYPE	DAYS
C003	Assist with covid-19 testing	volunteer	107
C009	Buy laptops for underfunded schools	monetary	99
C012	Buy prosthetics for a military veteran	monetary	71
C013	Volunteer at the Dawson's community center	volunteer	25
C005	Build a playground for a local school	both	15
C010	Help fund Karen's rehabilitation therapy	monetary	15
C014	Help John overcome cystic fibrosis	monetary	13
C007	Donate to the World Health Organization's fund...	monetary	0

QUERY 16: Estimation of number of additional volunteers needed for a campaign to reach its required time goal

An organizer may want to estimate how many more volunteers they need in order to reach their campaign goal. To obtain this insight, we will calculate the average amount of time currently being donated by the volunteers who have committed to their campaign. Based on this average time value, we will determine how many more volunteers will be needed to complete the remaining hours, with the assumption that future volunteers commit to a similar amount of time as the average time.

```
SELECT campaign_list.camp_id,
       Ceil(( campaign_list.time_required - Sum(time_worked) ) /
            Avg(time_worked)) AS
       VOL_STILL_REQUIRED,
       Count(volunteer_id)
       AS NUM_VOLUNTEERS
FROM   campaign_list,
       volunteer_transaction
WHERE  campaign_list.camp_id = volunteer_transaction.camp_id
GROUP  BY campaign_list.camp_id;
```

CAMP_ID	VOL_STILL_REQUIRED	NUM_VOLUNTEERS
C003	1	6
C005	16	7
C008	12	3
C011	19	1
C013	19	1
C015	33	1

QUERY 17: Estimation of donors needed to reach campaign target

Based on the previous data stored in the database, we can conduct some useful predictive analysis which will help estimate trends for current and future campaigns. In this instance, we can predict how many donors are still needed for each campaign for it to reach its goal by looking at a campaign's donation history. By looking at the average donation amount per user for each campaign, we can estimate how many more donors are needed to fulfill that campaign's goal with the assumption that future donors donate similar amounts.

```
SELECT campaign_list.camp_id,
       camp_descp,
       CEIL(( camp_goalamount - Sum(trans_amount) ) /
Avg(trans_amount)) AS
       MORE_DONOR
FROM   donor_transaction,
       campaign_list
WHERE  trans_paymentstatus = 'COMPLETED'
       AND campaign_list.camp_id = donor_transaction.camp_id
       AND camp_status = 1
GROUP  BY donor_transaction.camp_id;
```

	CAMP_ID	CAMP_DESCP	MORE_DONOR
▶	C002	Fund Dan's kidney transplant surgery	20
	C005	Build a playground for a local school	5
	C012	Buy prosthetics for a military veteran	1
	C014	Help John overcome cystic fibrosis	11
	C010	Help fund Karen's rehabilitation therapy	0
	C007	Donate to the World Health Organization's fund...	34
	C009	Buy laptops for underfunded schools	2

QUERY 18: Summary of the total number of volunteers and the total sum of money donated per country

We are interested in seeing the different donation patterns based on country. This information will be useful when designing our marketing strategies. We can see that the Bahamas ranks the highest in terms of total amount of money donated, while Canada ranks the highest for total number of volunteers. In the future, we could enhance our marketing in Argentina to attract more active donors and volunteers from the country.

```
SELECT DISTINCT donor_country,
                Sum(trans_amount) AS DONATION_AMOUNT,
                Count(volunteer_transaction.volunteer_id) AS
NUM_VOLUNTEER
FROM   volunteer_transaction,
       donor,
       donor_transaction
WHERE  volunteer_transaction.donor_id = donor.donor_id
       AND donor_transaction.donor_id = donor.donor_id
GROUP  BY donor_country
ORDER  BY donation_amount DESC;
```

DONOR_COUNTRY	DONATION_AMOUNT	NUM_VOLUNTEER
Bahamas	52686.08	8
South Korea	52641.36	8
Canada	41149.23	13
Bangladesh	34157.50	4
USA	24225.72	6
France	15972.36	3
China	15500.00	2
Argentina	2000.00	1

QUERY 19: Comparison of total amount of money donated vs. total hours volunteered for each donor

The output of this query is useful in giving us an overview of the donor activity on our site. With this information, we can gauge our donors' willingness to donate their time and/or money. For instance, we can see that donor D009 has donated quite a few hours (11hrs) but has not donated any money.

```
SELECT T1.donor_id,
       T1.donor_firstname,
       T1.donor_lastname,
       T2.total_work_done,
       T3.total_donation
FROM   donor T1
       LEFT JOIN (SELECT donor_id,
                         Sum(time_worked) AS TOTAL_WORK_DONE
                  FROM   volunteer_transaction
                  GROUP  BY donor_id) T2
              ON T1.donor_id = T2.donor_id
       LEFT JOIN (SELECT donor_id,
                         Sum(trans_amount) AS TOTAL_DONATION
                  FROM   donor_transaction
                  GROUP  BY donor_id) T3
              ON T1.donor_id = T3.donor_id;
```

DONOR_ID	DONOR_FIRSTNAME	DONOR_LASTNAME	TOTAL_WORK_DONE	TOTAL_DONATION
D001	Ludwig	Mates	10	2000.00
D002	Palm	Worsfield	16	17078.75
D003	Michal	Matitiaho	6	2378.19
D004	Valida	Wilshaw	11	15500.00
D005	Clim	Risborough	22	26343.04
D006	Bryn	Newbold	5	15972.36
D007	Betsy	Will	12	8075.24
D008	Griswold	McKelvey	29	12923.68
D009	Bertrando	Ruddiforth	11	NULL
D010	Debora	Richemond	15	26320.68

QUERY 20: Change in donor category preferences as income increases

This query will provide us with information on which categories our donors prefer, given their income bracket. We can see that "community" and "faith & mission" are important to our wealthiest donors ($200k+) while our donors who fall within the lower income bracket (0~$100k) are more drawn to "arts & culture" and "education" campaigns.

```
SELECT c.category_id,
       c.category_title AS CATEGORY_NAME,
       Count(CASE
               WHEN donor_income < 100000
                    AND donor_income > 0 THEN d.donor_id
             END)       AS '0 ~ 100k',
       Count(CASE
               WHEN donor_income >= 100000
                    AND donor_income < 200000 THEN d.donor_id
             END)       AS '100k ~ 200k',
       Count(CASE
               WHEN donor_income >= 200000 THEN d.donor_id
             END)       AS 'above 200k'
FROM   category c
       LEFT JOIN preference_category pc
              ON c.category_id = pc.category_id
       JOIN donor d
         ON pc.donor_id = d.donor_id
GROUP  BY c.category_id
ORDER  BY c.category_id ASC;
```

category_id	CATEGORY_NAME	0 ~ 100k	100k ~ 200k	above 200k
CAT001	Arts & Culture	1	2	0
CAT002	Education	1	0	0
CAT003	Environment	1	0	0
CAT004	Health & Medical	1	1	0
CAT005	Active Duty & Veterans	0	1	0
CAT006	Human Services	0	1	0
CAT007	Golbal	0	2	0
CAT008	Social Action	0	1	0
CAT009	Community	1	0	2
CAT010	Faith and Mission	0	0	2
CAT011	Miscellaneous	0	0	1
CAT012	Disaster Relief	1	0	0
CAT013	Animal and Humane	0	4	0

QUERY 21: Campaign category that is most preferred by donors and the number of users who selected the category as a preference

We would like to find out the campaign category that was chosen most among donors as their preference, as well as the number of donors who actually chose that category. With this information, we will be able to better customize the campaigns that we display on our homepage and showcase campaigns from the top category that more donors would be attracted to.
We find that the category that donors prefer the most is "community". 3 donors have listed this as their preference, so we will aim to showcase more community-driven campaigns on our homepage.

```
SELECT category_id,
       category_title,
       Max(number_of_donors) AS MAX_DONOR_NUMBER
FROM   (SELECT C.category_id,
               C.category_title,
               Count(PC.donor_id) AS NUMBER_OF_DONORS
        FROM   category AS C
               INNER JOIN preference_category AS PC
                       ON C.category_id = PC.category_id
        GROUP  BY C.category_id
        ORDER  BY number_of_donors DESC) AS
NUMBER_OF_DONORS_TABLE;
```

	CATEGORY_ID	CATEGORY_TITLE	MAX_DONOR_NUMBER
▶	CAT009	Community	3

QUERY 22: Setting campaign priority levels based on number of days left till campaign ends

A key service provided by Connect the Dots *is its ability to give campaigns extra visibility based on priority factors such as time left in a campaign or amount of funds still needed. Here, we can assign campaigns different priorities based on the difference between the current date and the campaign end date (i.e. the amount of time left before a campaign ends). We assign campaigns with more than 50 days left a "low" priority, campaigns with 20-50 days left a "moderate" priority, and campaigns with less than 20 days left a "high" priority. As campaigns near their end date, their priorities will automatically shift. This is important as it allows us to increase advertising for high priority campaigns, such as by placing them on the front page of our website.*

```
SELECT camp_id,
       camp_descp,
       camp_type,
       Datediff(camp_end_date, Sysdate()) AS DAYS_LEFT,
       CASE
         WHEN ( Datediff(camp_end_date, Sysdate()) > 0
```

```
                AND Datediff(camp_end_date, Sysdate()) <= 20 )
THEN 'HIGH'
      WHEN ( Datediff(camp_end_date, Sysdate()) < 0)
THEN 'CAMPAIGN_ENDED'
          WHEN ( Datediff(camp_end_date, Sysdate()) > 20
                AND Datediff(camp_end_date, Sysdate()) < 50 )
THEN 'MODERATE'
          WHEN Datediff(camp_end_date, Sysdate()) >= 50 THEN 'LOW'
          ELSE 'ERROR'
        END                                    AS PRIORITY
FROM    campaign_list
WHERE   camp_status = 1;
```

	camp_id	camp_descp	camp_type	DAYS_LEFT	PRIORITY
▶	C002	Fund Dan's kidney transplant surgery	monetary	-6	CAMPAIGN_ENDED
	C003	Assist with covid-19 testing	volunteer	106	LOW
	C005	Build a playground for a local school	both	14	HIGH
	C007	Donate to the World Health Organization's fund...	monetary	-1	CAMPAIGN_ENDED
	C009	Buy laptops for underfunded schools	monetary	98	LOW
	C010	Help fund Karen's rehabilitation therapy	monetary	14	HIGH
	C012	Buy prosthetics for a military veteran	monetary	70	LOW
	C013	Volunteer at the Dawson's community center	volunteer	24	MODERATE
	C014	Help John overcome cystic fibrosis	monetary	12	HIGH

QUERY 23: Match donors to new campaigns based on past donations

In order to better match donors with causes that donors are passionate about donating to, it can be vital to use past donation history in order to make future recommendations to donors. This targeted recommendation system can be useful as it both ensures that campaigns get increased visibility and exposure, while ensuring that donors are not inundated with campaigns they may find irrelevant. Targeted advertising will become especially important as the database grows and the number of campaigns and donation categories increases. Subsequently, we used a donor's past donations (both monetary and voluntary) to match them to new campaigns they might be interested in and previously had not donated to, based on their category preferences.

```
SELECT
   X.DONOR_ID,
   X.DONOR_FIRSTNAME,
   Y.CATEGORY_TITLE,
   X.CATEGORY_ID AS INTERESTED_IN,
   X.CAMP_ID AS RECOMMEND_CAMP_ID,
   Z.CAMP_DESCP AS RECOMMEND_CAMP_DESCP,
   Z.CAMP_END_DATE
FROM
```

```
(
   SELECT
      T4.CAMP_ID,
      T3.*,
      CONCAT(DONOR_ID, CAMP_ID) AS COMBINATION
   FROM
      (
         SELECT
            T2.*,
            T1.DONOR_FIRSTNAME
         FROM
            PREFERENCE_CATEGORY T2
            JOIN
               DONOR T1
               ON T2.DONOR_ID = T1.DONOR_ID
      )
      T3
      JOIN
         CAMPAIGN_CATEGORY T4
         ON T3.CATEGORY_ID = T4.CATEGORY_ID
   WHERE
      CONCAT(DONOR_ID, CAMP_ID) NOT IN
      (
         SELECT
            CONCAT(A.DONOR_ID, B.CAMP_ID)
         FROM
            DONOR A
            JOIN
               DONOR_TRANSACTION B
               ON A.DONOR_ID = B.DONOR_ID
            UNION
            SELECT
               CONCAT(C.DONOR_ID, D.CAMP_ID)
            FROM
               DONOR C
               JOIN
                  VOLUNTEER_TRANSACTION D
                  ON C.DONOR_ID = D.DONOR_ID
      )
)
X
JOIN
   CATEGORY Y
   ON X.CATEGORY_ID = Y.CATEGORY_ID
JOIN
   CAMPAIGN_LIST Z
   ON Z.CAMP_ID = X.CAMP_ID
ORDER BY X.DONOR_ID, X.CATEGORY_ID, X.CAMP_ID;
```

donor_id	DONOR_FIRSTNAME	CATEGORY_TITLE	INTERESTED_IN	RECOMMEND_CAMP_ID	RECOMMEND_CAMP_DESCP	CAMP_END_DATE
D001	Ludwig	Arts & Culture	CAT001	C009	Buy laptops for underfunded schools	2020-12-23
D001	Ludwig	Education	CAT002	C001	Help support Sarah's college education	2020-02-08
D001	Ludwig	Education	CAT002	C009	Buy laptops for underfunded schools	2020-12-23
D001	Ludwig	Environment	CAT003	C006	Help fund the Galapagos Tortoise breeding prog...	2020-04-21
D001	Ludwig	Environment	CAT003	C011	Join the community trash pickup	2020-06-25
D001	Ludwig	Health & Medical	CAT004	C002	Fund Dan's kidney transplant surgery	2020-09-10
D001	Ludwig	Health & Medical	CAT004	C006	Help fund the Galapagos Tortoise breeding prog...	2020-04-21
D001	Ludwig	Health & Medical	CAT004	C007	Donate to the World Health Organization's fund...	2020-09-15
D001	Ludwig	Health & Medical	CAT004	C010	Help fund Karen's rehabilitation therapy	2020-09-30
D001	Ludwig	Health & Medical	CAT004	C014	Help John overcome cystic fibrosis	2020-09-28
D001	Ludwig	Health & Medical	CAT004	C015	Help send aid and resources to those affected ...	2020-08-19
D001	Ludwig	Disaster Relief	CAT012	C007	Donate to the World Health Organization's fund...	2020-09-15
D001	Ludwig	Disaster Relief	CAT012	C015	Help send aid and resources to those affected ...	2020-08-19
D002	Palm	Arts & Culture	CAT001	C009	Buy laptops for underfunded schools	2020-12-23
D002	Palm	Health & Medical	CAT004	C007	Donate to the World Health Organization's fund...	2020-09-15
D002	Palm	Health & Medical	CAT004	C010	Help fund Karen's rehabilitation therapy	2020-09-30
D002	Palm	Health & Medical	CAT004	C014	Help John overcome cystic fibrosis	2020-09-28
D002	Palm	Health & Medical	CAT004	C015	Help send aid and resources to those affected ...	2020-08-19
D002	Palm	Animal and Humane	CAT013	C014	Help John overcome cystic fibrosis	2020-09-28
D003	Michal	Arts & Culture	CAT001	C009	Buy laptops for underfunded schools	2020-12-23
D003	Michal	Active Duty & Ve...	CAT005	C015	Help send aid and resources to those affected ...	2020-08-19

PART E

1. Overview of Project Experience

Overall insights:

Our group learned many things from each other throughout the project experience. First and foremost, we found the entire project to be an iterative process. After determining the structure of our initial ERD, Logical model, and DDL, we found ourselves going back and forth between these steps and adjusting our data models between each deliverable, based on the feedback given to us and our own evolving conception of our database. To improve individual learning, our group agreed that we would each write a set number of queries and then reconvene to discuss them together. This was a beneficial process, as after coming back together, we found certain question types would be attempted by multiple members in different ways. This in turn gave us multiple ways to approach each question to obtain the same output, and allowed us to collectively optimize our queries to consider alternative methods and consequently find the best way to approach each problem. Given the variety of data processing architectures available, we also learned that it was important to test our queries in different systems (i.e. PHPMyAdmin vs. MySQL Workbench) to ensure that our queries worked with the different settings of each processing system. This was an important lesson learned, given that in real organizational scenarios, clients may operate on different systems. Our team learned to test each other's queries to ensure that each query worked for each member of our team. This also allowed us to better understand queries that other teammates had written, and also provided an opportunity for us to critique and improve upon each other's queries.

Alternative data models considered:

We also found ourselves adjusting our data models and utilizing normalization to improve our database between each deliverable. While our initial ERD included M:N relationships between donors and category preferences and campaigns and campaign categories (see below figure), we found ourselves breaking these relationships up into 1:M relationships in order to represent relationship attributes -- as to more clearly organize our data and make it easier to query and access it. Our models also changed based on our evolving perception of the business functionalities of our database. We expanded our tables further after adding “volunteer” as an additional donation project in order to give variety to our project and an additional level of complexity upon which we could develop more nuanced queries. Since volunteer donations and monetary donations involved fundamentally different data types (time vs. money), we decided to use two

separate entities and relationships to represent them. As we developed our "business case", we also decided to add additional entities to make our database more realistic given the context of our business functionalities; for instance, we added donor bank details for security and practical reasons (i.e. to allow for donors to easily donate to multiple campaigns without having to input their details every time, and to allow for easy refunds in case of fraudulent campaigns). As we learned about additional data types, we also altered the pre-existing data types that we had chosen. For instance, we chose to replace our Datetime type attributes with Timestamp type attributes so that our data would be updated in real time. This was specifically useful for calculations such as calculating time until the end of a campaign.

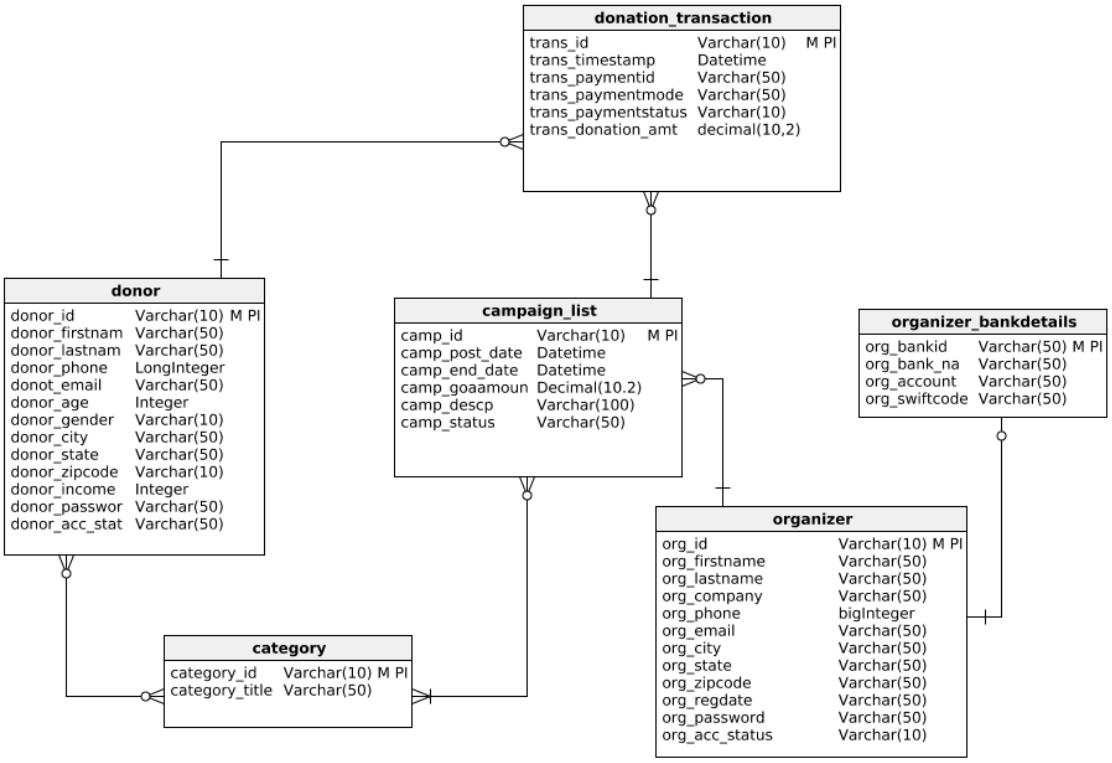


Figure 1 - First ERD created

2. Complex Queries Explained

QUERY 22: Setting campaign priority levels based on number of days left till campaign ends

This query assigns campaigns having different end dates with priority levels determined based on the difference between the expected end date and today's date (which is obtained by using SYSDATE() command). When the difference is between 0 to 20 days, priority is set to high; when the difference is between 20 to 50 days, priority is set to medium; when the difference is more than 50 days, the priority is set to low. This query is challenging and interesting as it generates a new column for categorizing each campaign based on three different conditions. The desired result is achieved by using the CASE statement, in which the three conditions for priority levels are set, and returned into a column named 'PRIORITY':

```
SELECT camp_id,
       camp_descp,
       camp_type,
       Datediff(camp_end_date, Sysdate()) AS DAYS_LEFT
       /*Calculate date difference between end date and today's date*/,

      CASE
          /*Condition for high priority:*/
          WHEN ( Datediff(camp_end_date, Sysdate()) > 0
                 AND Datediff(camp_end_date, Sysdate()) <= 20 )
THEN 'HIGH'

          /*Condition for medium priority:*/
          WHEN ( Datediff(camp_end_date, Sysdate()) > 20
                 AND Datediff(camp_end_date, Sysdate()) < 50 )
THEN 'MODERATE'

          /*Condition for low priority:*/
          WHEN Datediff(camp_end_date, Sysdate()) >= 50 THEN 'LOW'

            /*Condition for campaign ended:*/
            WHEN ( Datediff(camp_end_date, Sysdate()) < 0)
THEN 'CAMPAIGN_ENDED'

          ELSE 'ERROR'
       END                                  AS PRIORITY
```

/*Returns the priority levels back into a new column named 'PRIORITY'*/

```
FROM   campaign_list
```
/***Extracts from the table 'campaign_list' which contains all campaigns and their end dates***/

```
WHERE  camp_status = 1;
```
/***Specifies constraints that must be an ongoing campaign***/

***

QUERY 23: Match donors to new campaigns based on past donations

This query will give recommendations to donors about campaigns they might be interested in based on their preferences and donation history. This query is interesting and challenging since lots of concatenations are involved. Besides, it also involves how to exclude undesired rows from the result. The Campaign IDs and Donor IDs observed in all transaction records, both monetary and voluntary transactions, are concatenated together using CONCAT() statement to form a new column with unique values to identify the campaigns that donors have already donated to. The same concatenation is done on Campaign IDs and Donor IDs based on the donor preference table to determine all campaigns that donors might be interested in. After that, a WHERE NOT IN statement is used to exclude all campaigns that a donor has donated from the campaigns that donor might be interested in, and what is left is the campaigns that donors are interested but have not donated yet. A final selection is then made to add additional information like donor name, campaign descriptions and end dates, and categories campaigns belong to and corresponding category descriptions.

Comments and explanations based on the query:

```
SELECT /*Display detailed information about donors, campaigns, and categories*/
   X.DONOR_ID,
   X.DONOR_FIRSTNAME,
   Y.CATEGORY_TITLE,
   X.CATEGORY_ID AS INTERESTED_IN,
   X.CAMP_ID AS RECOMMEND_CAMP_ID,
   Z.CAMP_DESCP AS RECOMMEND_CAMP_DESCP,
   Z.CAMP_END_DATE
FROM
   (
      SELECT
```

```
        T4.CAMP_ID,
        T3.*,
        CONCAT(DONOR_ID, CAMP_ID) AS COMBINATION
```

/*Concatenate camp_id with donor_id based on preference_category and campaign_category, which displays all possible campaigns that might be interesting for each donor*/

```
    FROM
        (
            SELECT
                T2.*,
                T1.DONOR_FIRSTNAME
            FROM
                PREFERENCE_CATEGORY T2
                JOIN
                    DONOR T1
                    ON T2.DONOR_ID = T1.DONOR_ID
        )
        T3
        JOIN
            CAMPAIGN_CATEGORY T4
            ON T3.CATEGORY_ID = T4.CATEGORY_ID
    WHERE
        CONCAT(DONOR_ID, CAMP_ID) NOT IN
```

/*Statement to exclude all posted donations from all possible donations for donors*/

```
        (
            SELECT
                CONCAT(A.DONOR_ID, B.CAMP_ID)
            FROM
                DONOR A
                JOIN
                    DONOR_TRANSACTION B
                    ON A.DONOR_ID = B.DONOR_ID
```

/*Concatenate camp_id with donor_id based on donor_transaction table (monetary)*/

```
                UNION /*Unionize the two concatenations*/
                SELECT
                    CONCAT(C.DONOR_ID, D.CAMP_ID)
                FROM
                    DONOR C
                    JOIN
                        VOLUNTEER_TRANSACTION D
                        ON C.DONOR_ID = D.DONOR_ID
        )
```

/*Concatenate camp_id with donor_id based on volunteer_transaction table (voluntary)*/

```
    )
    X
```

```
JOIN
   CATEGORY Y
   ON X.CATEGORY_ID = Y.CATEGORY_ID
JOIN
   CAMPAIGN_LIST Z
   ON Z.CAMP_ID = X.CAMP_ID
ORDER BY X.DONOR_ID, X.CATEGORY_ID, X.CAMP_ID;
```

PART F

APPENDIX

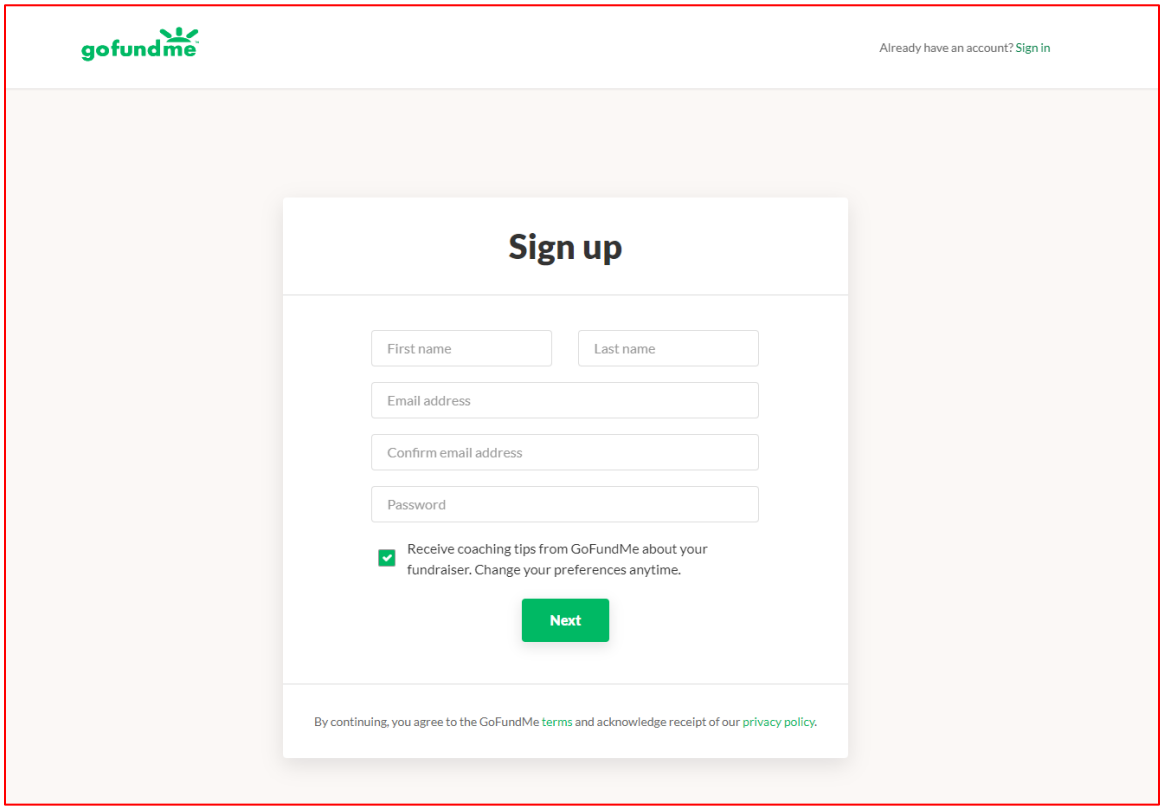


Image 1: Sample of data collected when donors and organizers create an account (e.g. 'first_name', 'last_name', 'email_address', 'password')

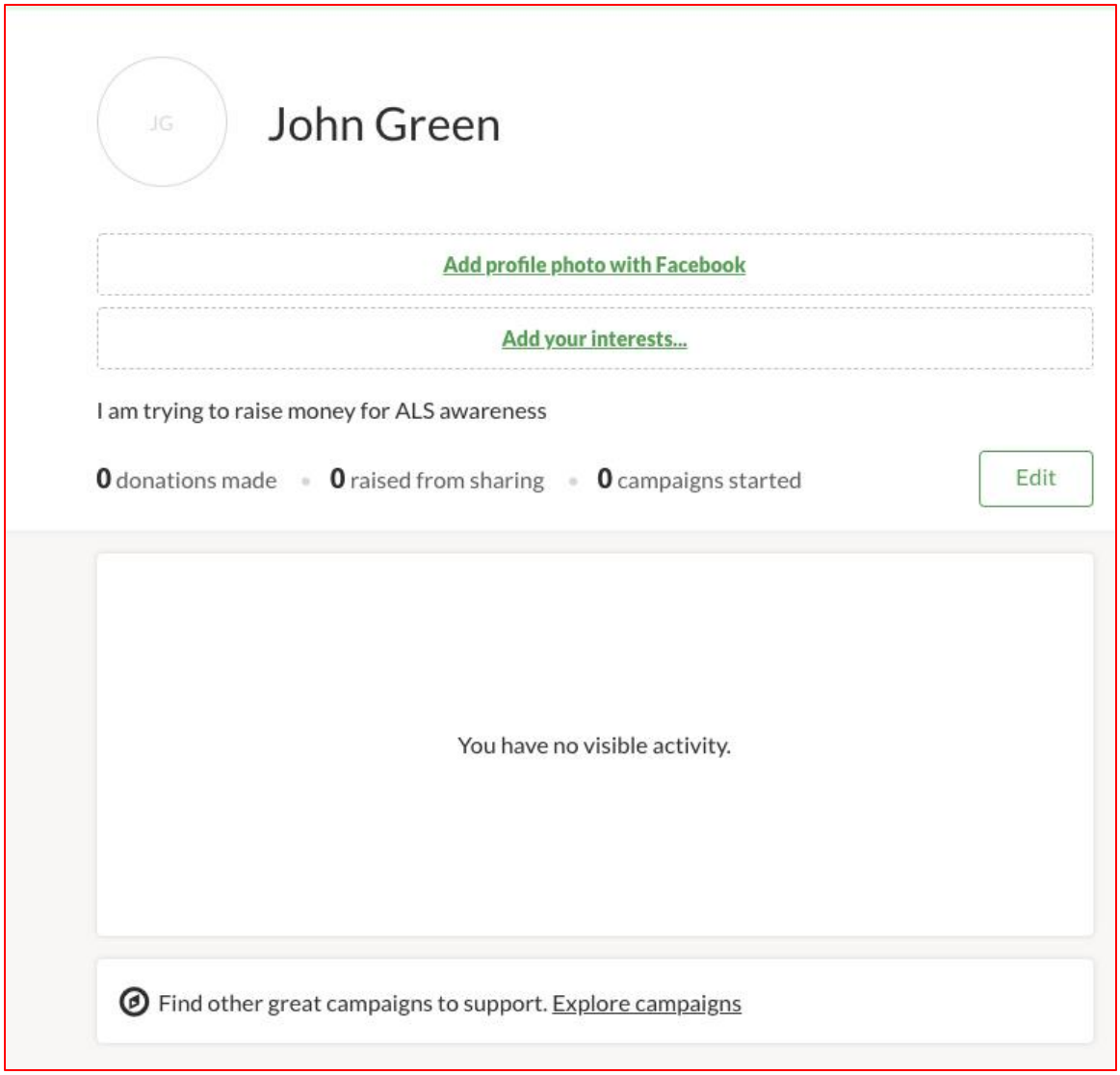


Image 2: Sample of information tracked in the donor account (e.g. 'campaigns started')

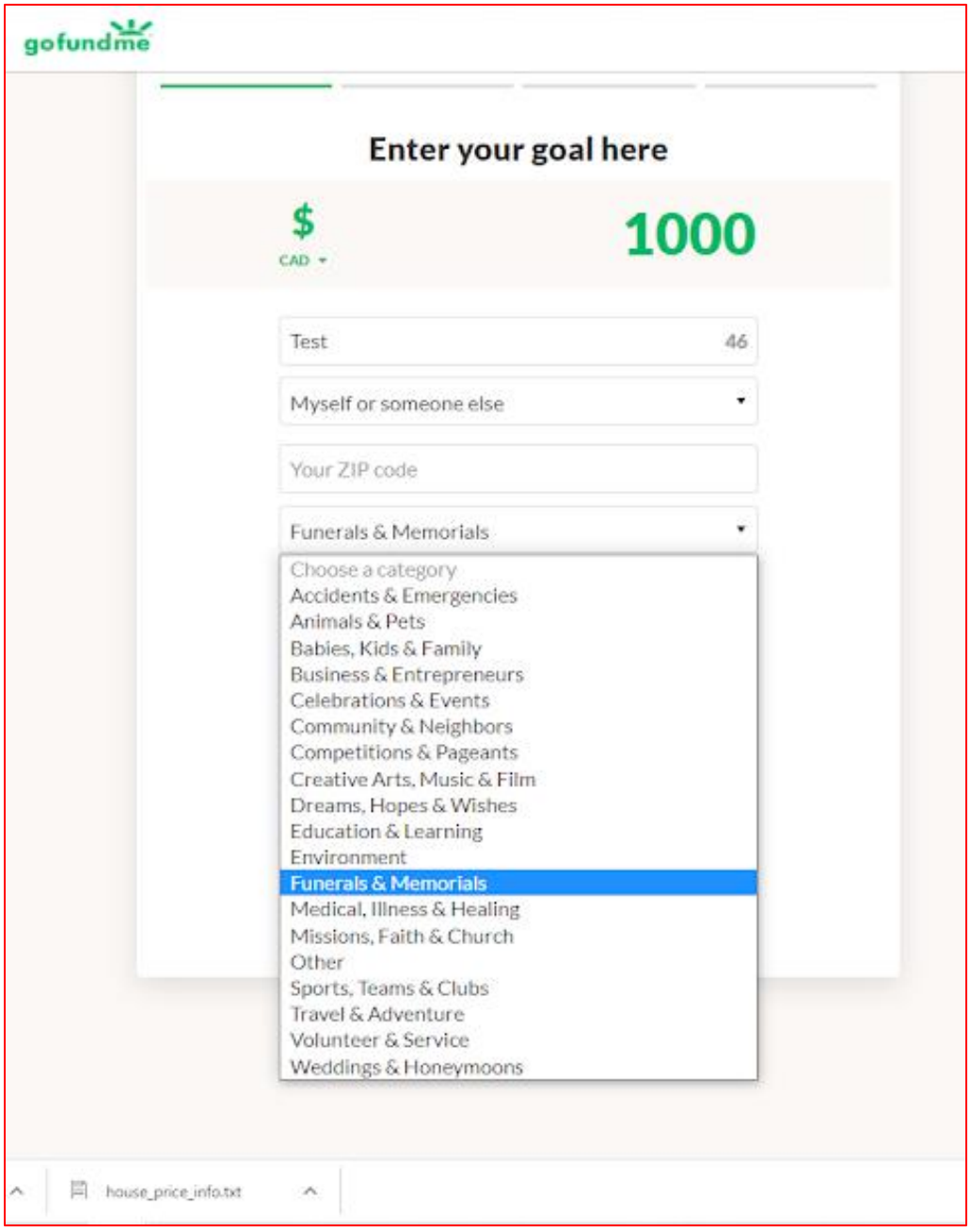


Image 3: Sample of data entered when creating a campaign (e.g. 'title', 'category', 'campaign_goal')

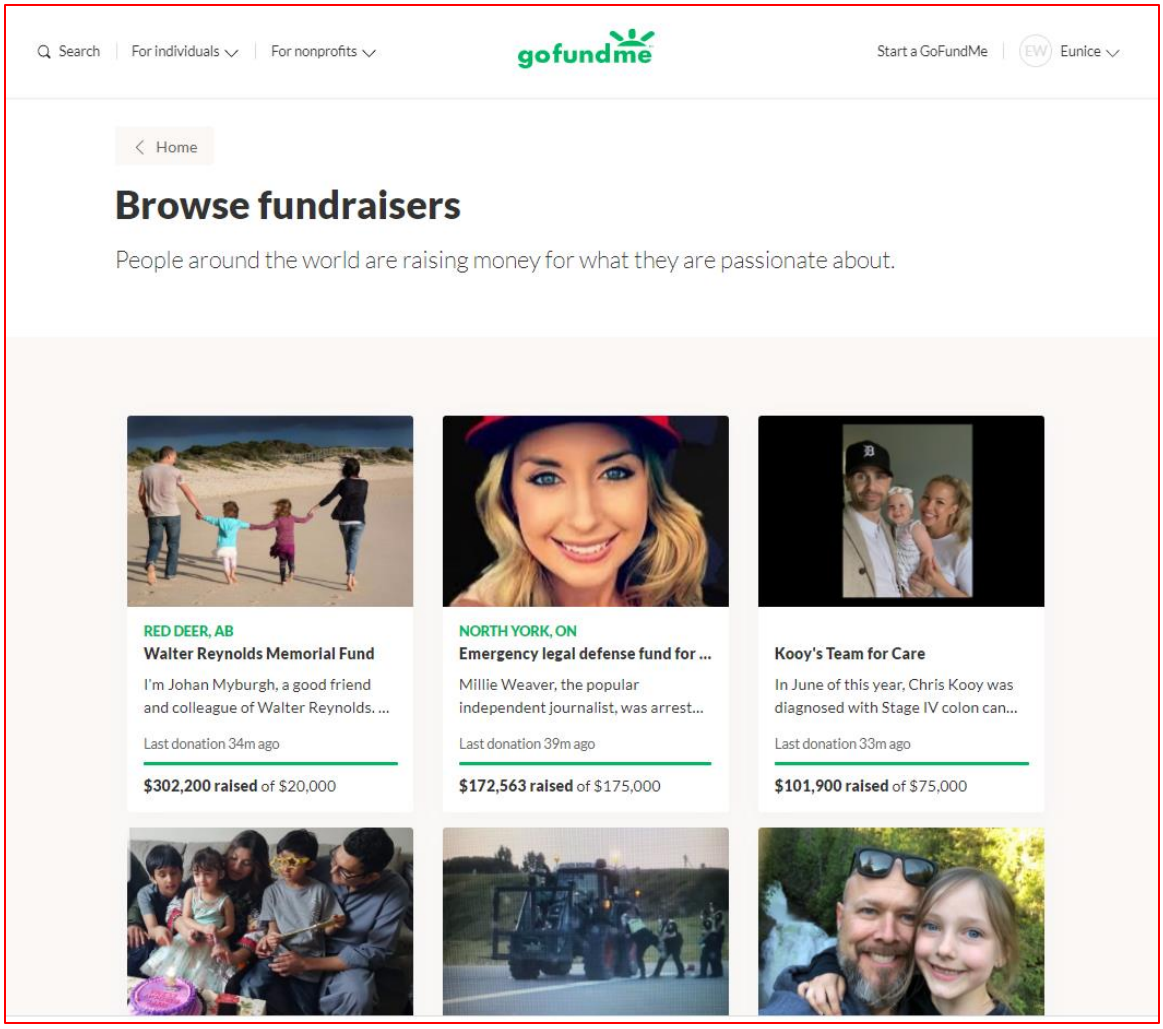


Image 4: Sample data displayed for campaigns (e.g. 'title', 'description', 'campaign_post_date', 'campaign_status'), as well as derived calculations such as amount raised of total amount