8.5.2

Create an Entity Relationship Diagram

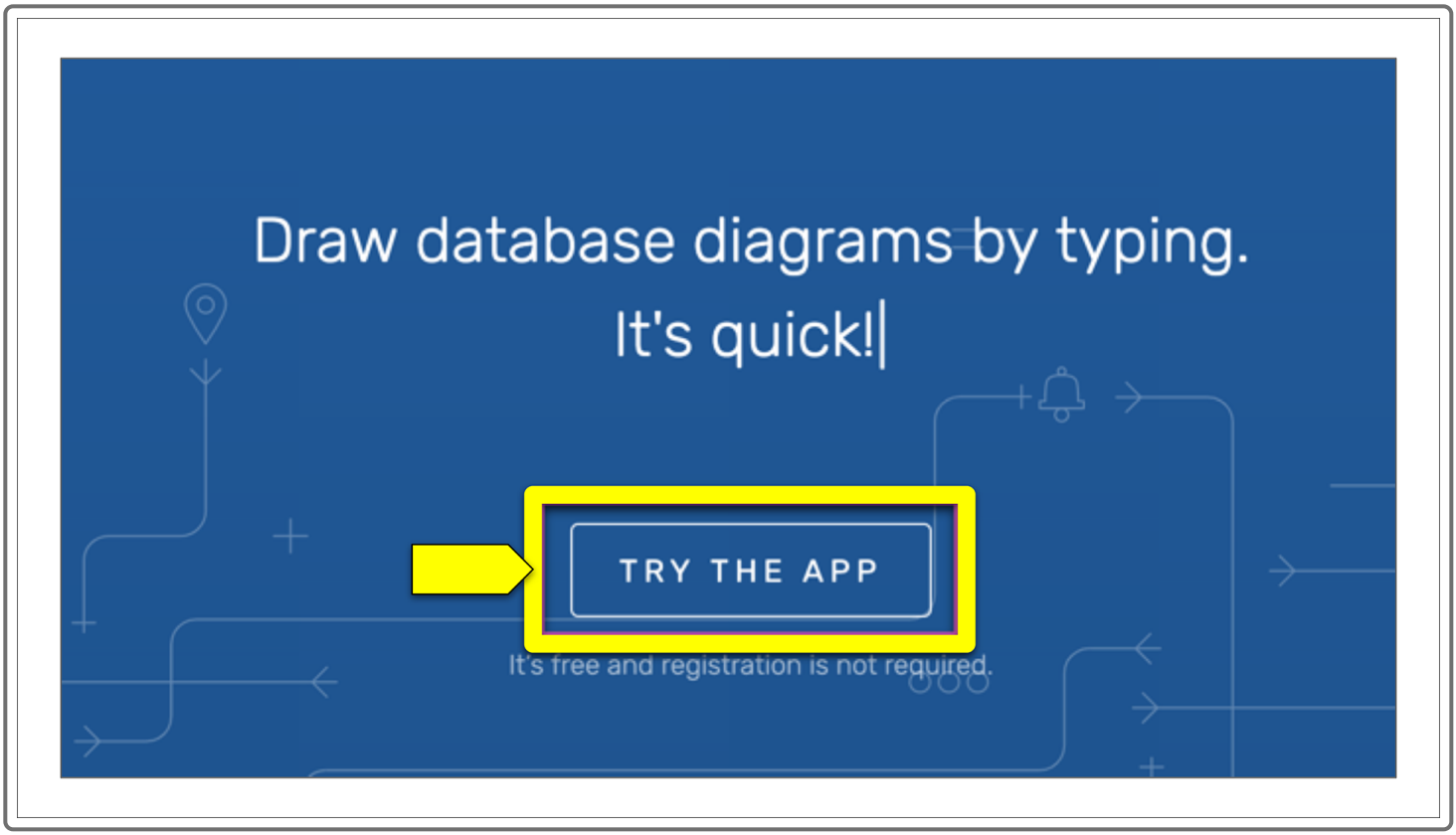
You and Britta need to create a map of the database and table schema by using an entity relationship diagram (ERD). To create the ERD, you'll use Quick Database Diagrams (Quick DBD).

With Quick DBD, you and Britta can model the data, which will allow you to establish the relationships among the four datasets.

REWIND

Quick DBD creates clean and comprehensible ERDs that we can easily export as image files.

To get started, go to the Quick DBD website (http://quickdatabasediagrams.com) , and then click the "Try the App" button.

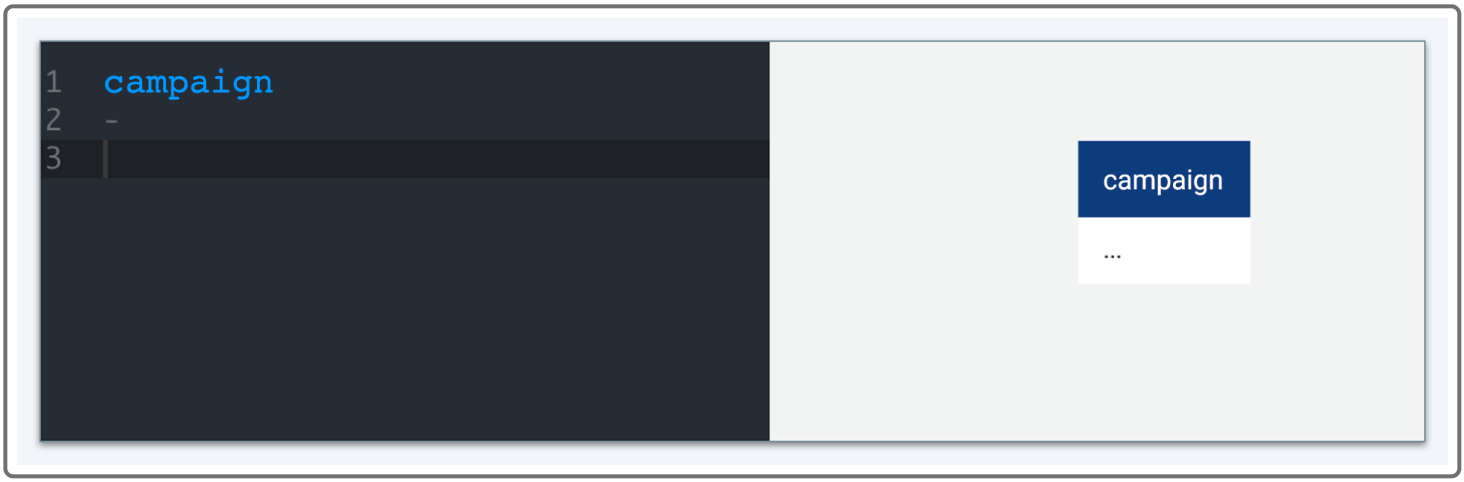


When the text editor and canvas appear, clear the contents in the text editor. We'll be creating the database schema from scratch.

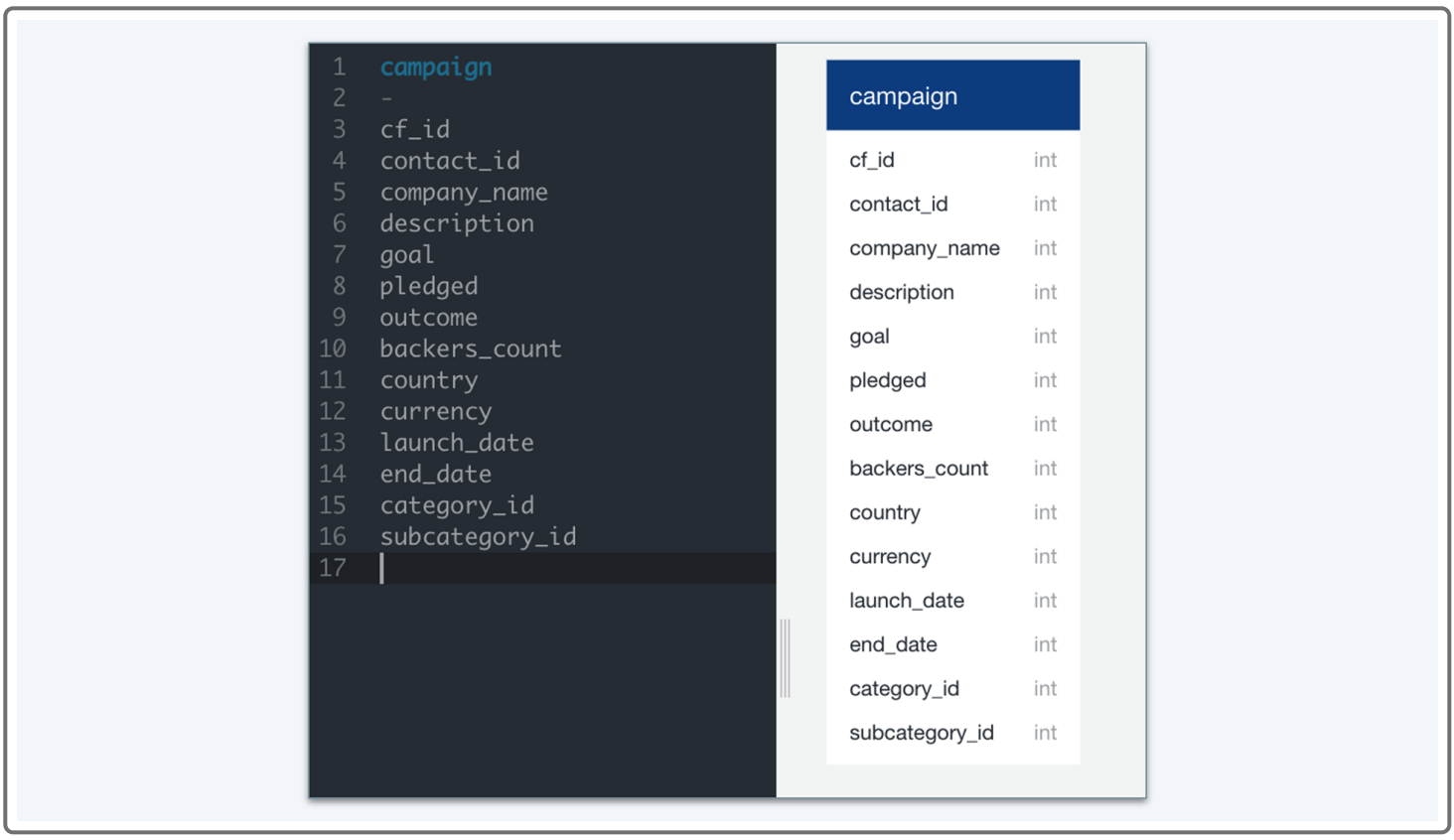
Create a Conceptual Diagram

First, we'll create the schema from the `campaign.csv` dataset for the `campaign` table. Begin by entering "campaign" (without the quotes) on the first line, press Enter, add a hyphen on the second line, and then press Enter.

The following image shows your conceptual `campaign` table so far:



Next, add placeholders for the table columns, as the following image shows:



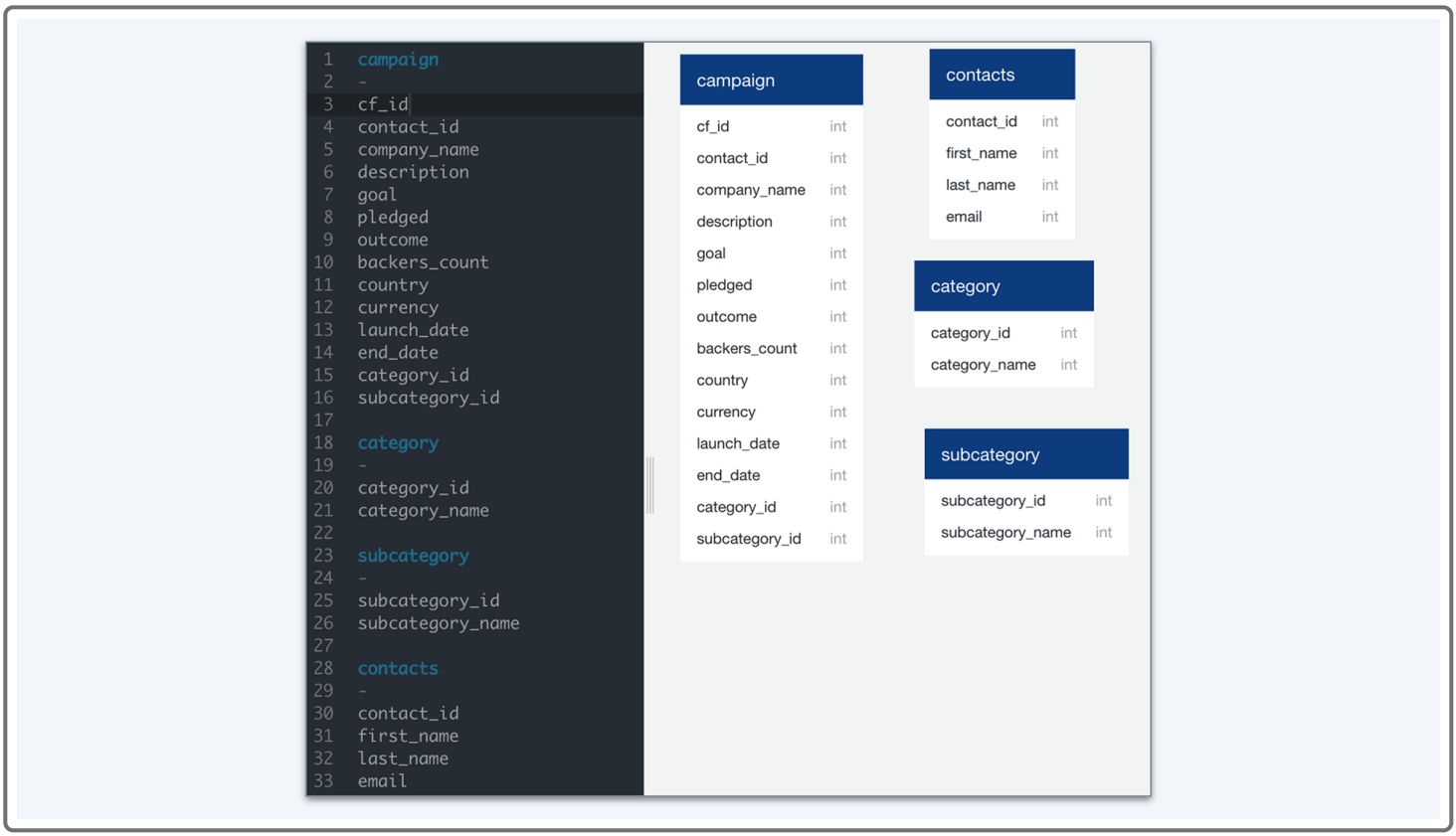
The canvas should contain the conceptual `campaign` table that includes the 14 column names, each with a type of `int`.

Practice adding tables yourself by adding the conceptual `category`, `subcategory`, and `contacts` tables in the following Skill Drill:

SKILL DRILL

After the `campaign` table, create the conceptual `category`, `subcategory`, and `contacts` tables. Make sure to add a blank line before adding each new table by pressing Enter.

The following image shows all four conceptual tables in Quick DBD:



As the preceding image shows, the canvas now has the following three additional tables:

- The `category` table with the following column names:
  - category_id
  - category_name
- The `subcategory` table with the following column names:
  - subcategory_id
  - subcategory_name
- The `contacts` table with the following column names:

- contact_id
- first_name
- last_name
- email

And, the canvas shows that all the columns in all the tables have a type of `int`.

Turn the Conceptual Diagram into a Logical Diagram

Next, we'll assign data types to the tables to turn the conceptual diagram into a logical diagram.

REWIND

A logical diagram has all the information that a conceptual diagram does. But, the table is updated to include data types, primary keys, and foreign keys.

To do this, we'll first update the schema for the `campaign` table. We know that the primary key is the "cf_id" column. And, we know that the foreign keys are the "contact_id", "category_id", and "subcategory_id" columns. So, we'll add this information in the text editor.

In the text editor, update the `campaign` table by adding syntax as follows:

- On Line 3 (the line that has cf_id), add a space and then "int PK" (without the quotes) to the end of the line.
- On Line 4 (the line that has contact_id), add a space and then "int FK - contacts.contact_id" (without the quotes) to the end of the line.
- On Line 15 (the line that has category_id), add "varchar(10) FK >- category.category_id" (without the quotes) to the end of the line.
- On Line 16 (the line that has subcategory_id), add "varchar(10) FK >- subcategory.subcategory_id" (without the quotes) to the end of the line.

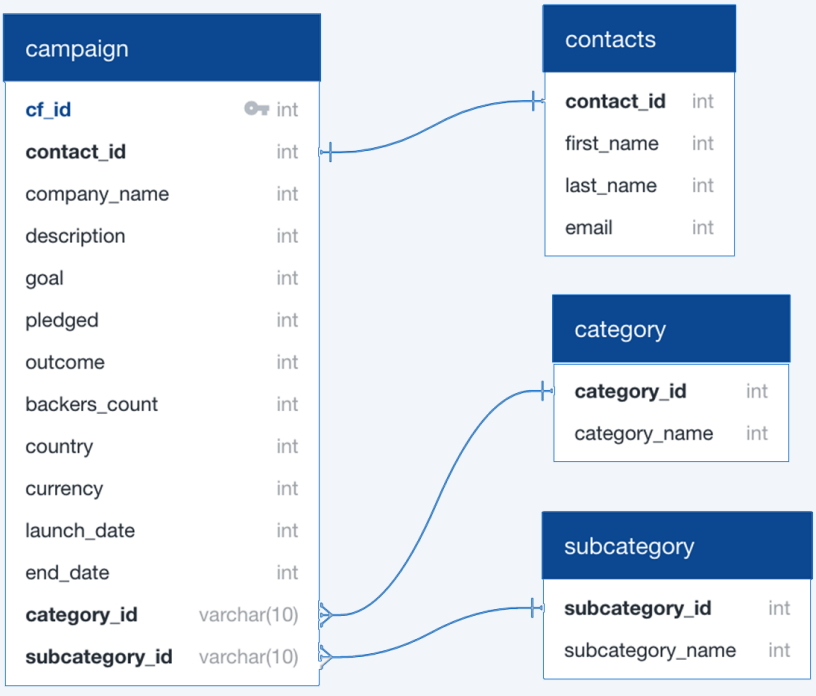
NOTE

PK stands for primary key, and FK stands for foreign key.

The following image shows the primary and foreign key relationships in the text editor:

```
campaign
-
cf_id int PK
contact_id int FK - contacts.contact_id
company_name int
description int
goal int
pledged int
outcome int
backers_count int
country int
currency int
launch_date int
end_date int
category_id varchar(10) FK >- category.category_id
subcategory_id varchar(10) FK >- subcategory.subcategory_id
```

And, the following image shows the relationship diagram in the canvas:



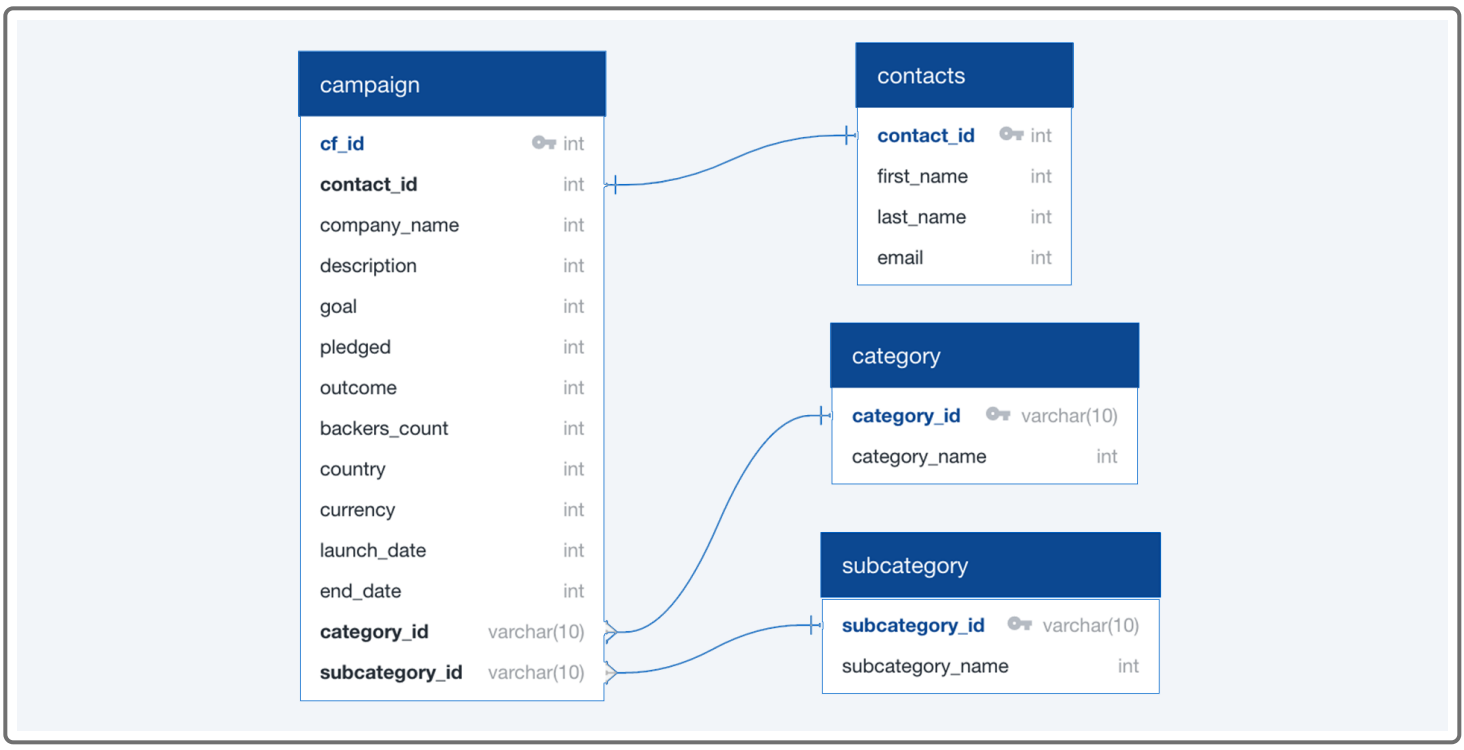
As the preceding image shows, the campaign table has a one-to-one relationship with the contacts table through the "contact_id" column, where each contact_id appears only once in both tables. But, there is a one-to-many relationship between the category and subcategory tables and the campaign table on the "category_id" and "subcategory_id" columns, respectively. This is because each category_id and subcategory_id appears more than once in the campaign table.

Practice assigning the primary keys for the remaining three tables yourself in the following Skill Drill:

SKILL DRILL

Earlier, we determined the primary keys in the `category`, `subcategory`, and `contacts` datasets. Assign the columns in those datasets as the primary keys with the appropriate data types.

The following image shows the relationship diagram in the canvas after you've assigned the primary key and data type to the `category`, `subcategory`, and `contacts` table:



As the preceding image shows, the primary key in contacts, category, and subcategory tables are the "contact_id", "category_id" and "subcategory_id" columns, respectively.

Before moving on, check your knowledge in the following assessment:

Next, we'll assign data types to the remaining columns. We'll start with the `campaign` table.

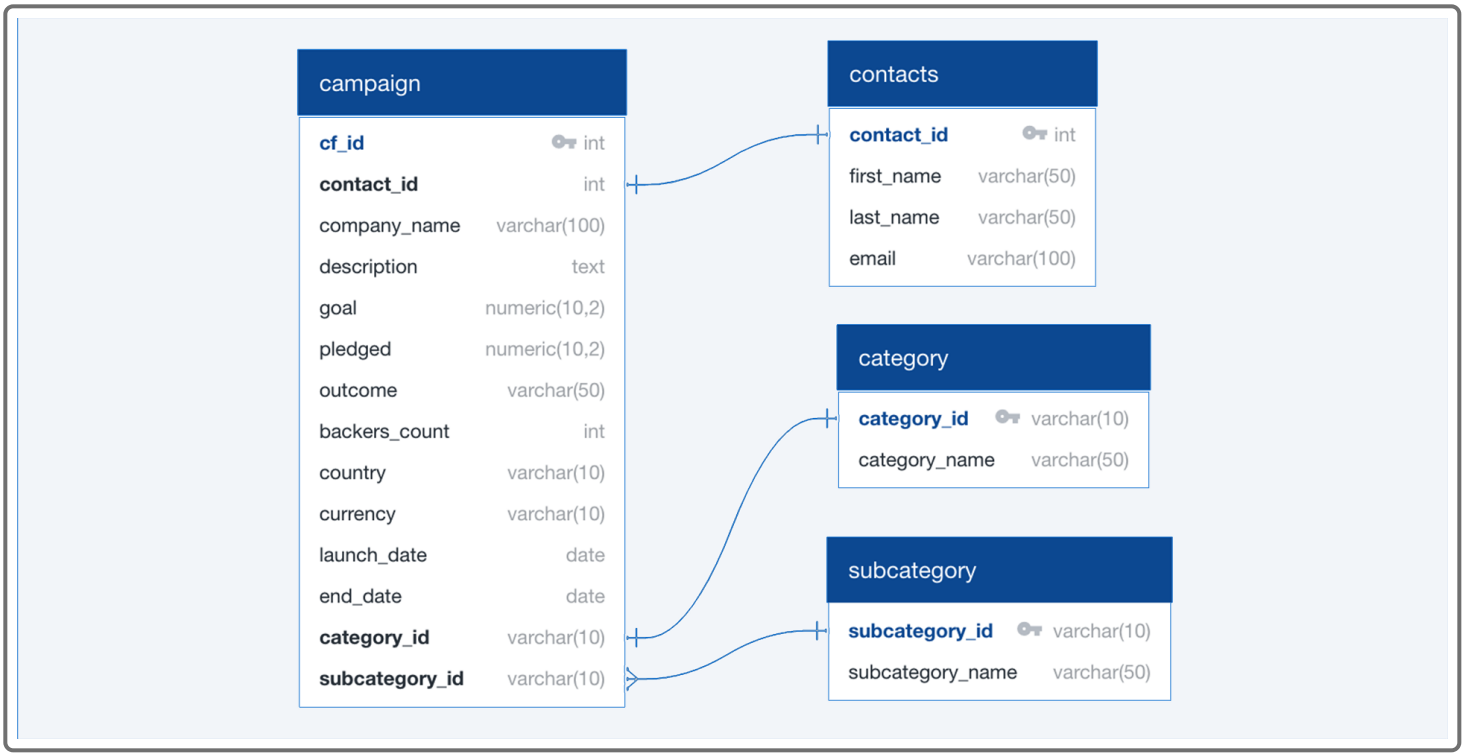
In the text editor, in the `campaign` table, cast data types to columns as follows:

- The "company_name" column is long, so cast this as "varchar(100)."
- The "description" column is also long, so cast this as "text."
- The "goal" and "pledged" columns are both currency, so cast this as "numeric(10,2)."
    - This allows a decimal number with a maximal total precision of 10 digits—8 before the decimal and 2 after.
    - The largest number that this data type can accommodate is 99,999,999.99.
- Cast the outcome column "varchar(50)."
- Leave the backers_count column as "int."
- Cast the country and currency columns as "varchar(10)."
- Cast the launch_date and end_date columns as "date."

In the remaining columns in the `category`, `subcategory`, and `contacts` tables, cast data types to columns as follows:

- Cast the category_name and subcategory_name columns as "varchar(50)."
- Cast the first_name and last_name columns as "varchar(50)."
- Cast the email column as "varchar(100)."

The following image shows the resulting relationship diagram in the canvas:



The database schema is now complete! Save your relationship entity diagram as `crowdfunding_db_schema.png` and your database schema as a PostgreSQL file, and rename it, `crowdfunding_db_schema.sql`. Next, you'll use the SQL code in the `crowdfunding_db_schema.sql` file to create the crowdfunding database and tables in pgAdmin.