Module 8 Challenge

Start Assignment

* **Due** Thursday by 11:59pm

* **Points** 100

* **Submitting** a text entry box or a website url

**Background**

Independent Funding loves what you and Britta have done with both the crowdfunding ETL project and the SQL data analysis. But, you still have more to do! The company has just received a new dataset that contains information about the backers who’ve pledged to the live projects. And, Independent Funding wants you and Britta to perform both an ETL process on this dataset and a data analysis by using SQL queries.

In this Challenge, you’ll use Python, Pandas, and Jupyter notebooks to do the extract and transform phases. Specifically, you’ll extract and transform the backers’ contact information from a CSV file to create a DataFrame that will be exported as a CSV file. You'll then do the load phase. Specifically, you’ll use the dataset to create an ERD and a table schema for creating a new table in the crowdfunding\_db database. And, you’ll upload the CSV file that contains the backers’ information into this table. Finally, you’ll perform a data analysis on the crowdfunding\_db database by using SQL queries.

**What You're Creating**

This new assignment consists of four technical analysis deliverables. You will submit the following:

* Deliverable 1: Extract Data
* Deliverable 2: Transform and Clean Data
* Deliverable 3: Create an ERD and Table Schema, and Load Data
* Deliverable 4: SQL Analysis

**Files**

[backer\_info.csvLinks to an external site.](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/v2/module_8/backer_info.csv)

[Extract-Transform starter codeLinks to an external site.](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/v2/module_8/Extract-Transform_starter_code.ipynb)

[SQL analysis starter codeLinks to an external site.](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/v2/module_8/SQL_Analysis_starter_code.sql)

**Deliverable 1: Extract Data (35 pt)**

**Deliverable 1 Instructions**

Using your knowledge of Python, Pandas, and the extract and transform phases of ETL, you’ll extract the raw data and add it to a DataFrame for the transform phase.

**REWIND**

For this deliverable, you have done the following in this module:

* **Lesson 8.2.1:** Opening and inspecting a dataset
* **Lesson 8.2.3:** Creating a DataFrame
* **Lesson 8.4.3:** Using regular expressions for data extraction

1. Download the backer\_info.csv and Extract-Transform\_starter\_code.ipynb files into your Crowdfunding-ETL folder. Then rename the Extract-Transform\_starter\_code.ipynb file to Extract-Transform\_final\_code.ipynb.
2. Import the backer\_info.csv file into a DataFrame.
3. Note that for this deliverable you have two options for extracting the data: use Python dictionary methods or use regular expressions. Choose your option.
4. To use use Python dictionary methods, complete the following steps:
   * Iterate through the DataFrame, and convert each row to a dictionary.
   * Iterate through each dictionary, and do the following:
     + Extract the dictionary values from the keys by using Python list comprehensions.
     + Add the values for each row to a new list.
5. To use regular expressions, complete the following step:
   * Extract the "backer\_id", "cf\_id", "name", and "email" string values.
     + If you’d like a hint for extracting the alphanumeric "backer\_id" and the numeric "cf\_id" identification numbers, that’s totally okay. If not, that’s great too. You can always revisit this later if you change your mind.

**HINT**

1. Create a new DataFrame with the retrieved data.
2. Confirm that your DataFrame matches the following image:

In the preceding image, notice that the DataFrame has "backer\_id", "cf\_id", "name", and "email" columns. And, each column contains the extracted data from the backer\_info.csv file.

**Deliverable 1 Requirements**

* The alphanumeric "backer\_id" string identification number is extracted without extra characters. **(5 pt)**
* The numeric "cf\_id" string identification number is extracted without extra characters. **(5 pt)**
* ​The "name" string value is extracted without extra characters. **(5 pt)**
* ​The "email" string value is extracted without extra characters. **(5 pt)**
* ​A DataFrame is created with the following columns: "backer\_id", "cf\_id", "name", and "email". **(5 pt)**
* Each column in the DataFrame contains the appropriate data. **(10 pt)**

**Deliverable 2: Transform and Clean Data (30 pt)**

**Deliverable 2 Instructions**

Using your knowledge of Python, Pandas, and data cleaning strategies, you’ll transform the data via formatting, splitting, converting data types, and restructuring to create a DataFrame that can be loaded into a postgreSQL database as a CSV file.

**REWIND**

For this deliverable, you have done the following in this module:

* **Lesson 8.3.1:** Data cleaning strategies
* **Lesson 8.3.5:** Creating a DataFrame

1. Continuing to use the Extract-Transform\_final\_code.ipynb file, check the data types of the columns, and convert the "cf\_id" column to int64 if necessary.
2. Split the names in the "name" column into first and last names, and add them to "first\_name" and the "last\_name" column, respectively, in the DataFrame.
3. Drop the "name" column in the DataFrame.
4. Reorder the columns so that your DataFrame matches the following image:

In the preceding image, notice that the DataFrame has the columns in the following order: "backer\_id", "cf\_id", "first\_name", "last\_name" and "email".

1. Export the DataFrame as backers.csv.

**Deliverable 2 Requirements**

You will earn a perfect score for Deliverable 2 by completing all requirements below:

* The "cf\_id" column is converted to int64. **(5 pt)**
* The "name" column is split into "first\_name" and "last\_name" columns that are added to the DataFrame. **(10 pt)**
* The "name" column is dropped from the DataFrame. **(5 pt)**
* The columns are reordered. **(5 pt)**
* The DataFrame is exported as backers.csv. **(5 pt)**

**Deliverable 3: Create an ERD and a Table Schema and Load the Data (20 pt)**

**Deliverable 3 Instructions**

Using the ERD that you created in this module, you’ll create a backers table that has primary and foreign keys based on the summary information about the backers.csv dataset. You'll then export the updated database schema as a PostgreSQL file and use it to create the backers table in the crowdfunding\_db database. Finally, you'll use pgAdmin to upload the backers.csv file into the backers table.

**REWIND**

For this deliverable, you have done the following in this module:

* **Lesson 8.5.1:** Identifying data relationships
* **Lesson 8.5.2:** Creating an entity relationship diagram
* **Lesson 8.5.3:** Creating a database and tables
* **Lesson 8.5.4:** Loading the data

1. Using the information from the backers.csv file, open the crowdfunding\_db ERD that you created in this module and create a backers table.
   * Make sure that each column has the appropriate data type, that the table has a primary and a foreign key, and that the foreign key references the relevant table.
2. Save the crowdfunding relationship diagram as crowdfunding\_db\_relationships.png and the updated schema as a PostgreSQL file named crowdfunding\_db\_schema.sql.
3. Using the crowdfunding\_db\_schema.sql PostgreSQL file, copy the schema for the backers table and the ALTER TABLE statement to add the foreign key constraint, paste them in the pgAdmin query editor, and then run the query.
4. Refresh your schema in the crowdfunding\_db database to confirm that the backers table was created.
5. Upload the backers.csv file into the backers table.
6. To check the import, enter the following in the query editor, and then run the query:

SELECT \* FROM backers;

1. Confirm that the first 10 rows of the backers table match the following image:

In the preceding image, notice that the backers table should mirror the CSV file.

1. Save crowdfunding\_db\_relationships.png and crowdfunding\_db\_schema.sql in your Crowdfunding-ETL GitHub repository.

**Deliverable 3 Requirements**

You will earn a perfect score for Deliverable 2 by completing all requirements below:

* The crowdfunding\_db relationship diagram has five tables, and the diagram is saved as crowdfunding\_db\_relationships.png. **(5 pt)**
* The crowdfunding\_db\_schema.sql file contains the table schema and the ALTER TABLE statement for each of the five tables. **(10 pt)**
* The backers.csv file is imported into the backers table without any errors. **(5 pt)**

**Deliverable 4: SQL Analysis—Bonus (15 pt)**

**NOTE**

You can earn bonus pt for completing the SQL queries in this deliverable.

**Deliverable 4 Instructions**

Using your knowledge of SQL, you’ll perform a data analysis on the crowdfunding\_db database.

**REWIND**

For this deliverable, you have done the following in a previous module:

* [Lesson 7.3.1](https://courses.bootcampspot.com/courses/1001/pages/7-dot-3-1-query-dates): Create new tables with the INTO statement
* [Lesson 7.3.1](https://courses.bootcampspot.com/courses/1001/pages/7-dot-3-1-query-dates): Filter queries with the WHERE clause
* [Lesson 7.3.3](https://courses.bootcampspot.com/courses/1001/pages/7-dot-3-3-joins-in-action): Use the INNER JOIN clause to join two tables on a primary key
* [Lesson 7.3.3](https://courses.bootcampspot.com/courses/1001/pages/7-dot-3-3-joins-in-action): Use the ON () clause
* [Lesson 7.3.3](https://courses.bootcampspot.com/courses/1001/pages/7-dot-3-3-joins-in-action): Use an alias instead of a full table name
* [Lesson 7.3.4](https://courses.bootcampspot.com/courses/1001/pages/7-dot-3-4-use-count-group-by-and-order-by): Use the GROUP BY andORDER BY` clauses
* [Lesson 7.3.4](https://courses.bootcampspot.com/courses/1001/pages/7-dot-3-4-use-count-group-by-and-order-by): Use the COUNT() function to retrieve the total number of rows that match a specified criteria

1. Download the ETL\_SQL\_starter\_code.sql file, and rename it to crowdfunding\_SQL\_Analysis.sql.
2. Write a SQL query that retrieves the number of backer\_counts in descending order for each “cf\_id” for all the "live" campaigns.
3. Write a SQL query that uses the backers table to confirm the results from Step 2.
4. You and Britta receive notification from her boss, who wants to send an email to each contact of every live campaign to inform them of how much of the goal remains. To allow them to do so, complete the following steps:
   * Write a SQL query that creates a new table named email\_contacts\_remaining\_goal\_amount that contains the first name of each contact, the last name, the email address, and the remaining goal amount (as "Remaining Goal Amount") in descending order for each live campaign.
   * Confirm that the table matches the following image:

In the preceding image, notice that the "email\_contacts\_remaining\_goal\_amount" contains the remaining goal amount for the live campaigns in descending order, and the first and last name, and email address of each contact .

* + Export the table as email\_contacts\_remaining\_goal\_amount.csv.

1. Britta informs you that her boss also wants to send an email to each backer to let them know how much of the goal remains for each live campaign that they’ve pledged. To allow them to do so, complete the following steps:
   * Write a SQL query that creates a new table named email\_backers\_remaining\_goal\_amount that contains the email addresses of the backers in descending order, the first and the last name of each backer, the cf\_id, the company name, the description, the end date of the campaign, and the remaining amount of the campaign goal as "Left of Goal".
   * Confirm that your table matches the following image:
   * Export the table as email\_backers\_remaining\_goal\_amount.csv.
2. Add your queries to the crowdfunding\_SQL\_Analysis.sql file, and then save the file in your Crowdfunding-ETL GitHub repository.

**Deliverable 4 Requirements**

You will earn a perfect score for Deliverable 4 by completing all requirements below:

* A SQL query is written and successfully executed that retrieves the number of backer\_counts in descending order for each cf\_id and for all the live campaigns. **(2.5 pt)**
* A SQL query is written and successfully executed that retrieves the number of backers in descending order for each cf\_id from the backers table. **(2.5 pt)**
* A SQL query is written and successfully executed to create the email\_contacts\_remaining\_goal\_amount table, and the table is exported as email\_contacts\_remaining\_goal\_amount.csv. **(5 pt)**
* A SQL query is written and successfully executed to create the email\_backers\_remaining\_goal\_amount table, and the table is exported as email\_contacts\_remaining\_goal\_amount.csv. **(5 pt)**

**Submission**

Once you’re ready to submit, make sure to check your work against the rubric to ensure you meet the requirements for this Challenge one final time. It’s easy to overlook items when you’re in the zone!

As a reminder, the deliverables for this Challenge are as follows:

* Deliverable 1: Extract Data
* Deliverable 2: Transform and Clean Data
* Deliverable 3: Create an ERD and a Table Schema, and the Load Data
* Deliverable 4: SQL Analysis—Bonus

**IMPORTANT**

Don’t clear the output of your Jupyter Notebook files. Doing so will result in a lower score.

Before submitting the URL to your Crowdfunding-ETL GitHub repository make sure it contains the following files:

* The Extract-Transform\_final\_code.ipynb file
* All five CSV files, including the backers.csv file
* The ERD logical diagram, which includes the backers table saved as crowdfunding\_db\_relationships.png
* The database schema, which includes the backers table saved as crowdfunding\_db\_schema.sql
* The bonus SQL queries, saved as crowdfunding\_SQL\_Analysis.sql
* The email\_contacts\_remaining\_goal\_amount table, saved as email\_contacts\_remaining\_goal\_amount.csv
* The email\_backers\_remaining\_goal\_amount table, saved as email\_backers\_remaining\_goal\_amount.csv
* A README.md file that describes the purpose of the repository. Although this Challenge doesn’t include a graded written analysis, we encourage you to add a brief description of your project, which is a good practice.

To submit your Challenge assignment in Bootcamp Spot, click Start Assignment, click the Website URL tab, add the URL for the solution to your Crowdfunding-ETL project in your GitHub repository, and then click Submit. Comments are disabled for graded submissions in Bootcamp Spot. If you have questions about your feedback, notify the instructional staff or your Student Success Manager. If you want to resubmit your work for an improved grade, you can use the **Re-Submit Assignment** button to upload new links. You can resubmit your work up to three times for a total of four submissions.

**IMPORTANT**

When you receive feedback on your Challenge, make any suggested updates or adjustments to your work. Then add this week’s Challenge to your professional portfolio.

**NOTE**

You are allowed to miss up to two Challenge assignments and still earn your certificate. If you wish to skip this assignment, click Submit then indicate you are skipping by typing “I choose to skip this assignment” in the text box.

[Previous](https://courses.bootcampspot.com/courses/2457/modules/items/791776)[Next](https://courses.bootcampspot.com/courses/2457/modules/items/791785)

© 2022 edX Boot Camps LLC