# Investor Portfolio Builder

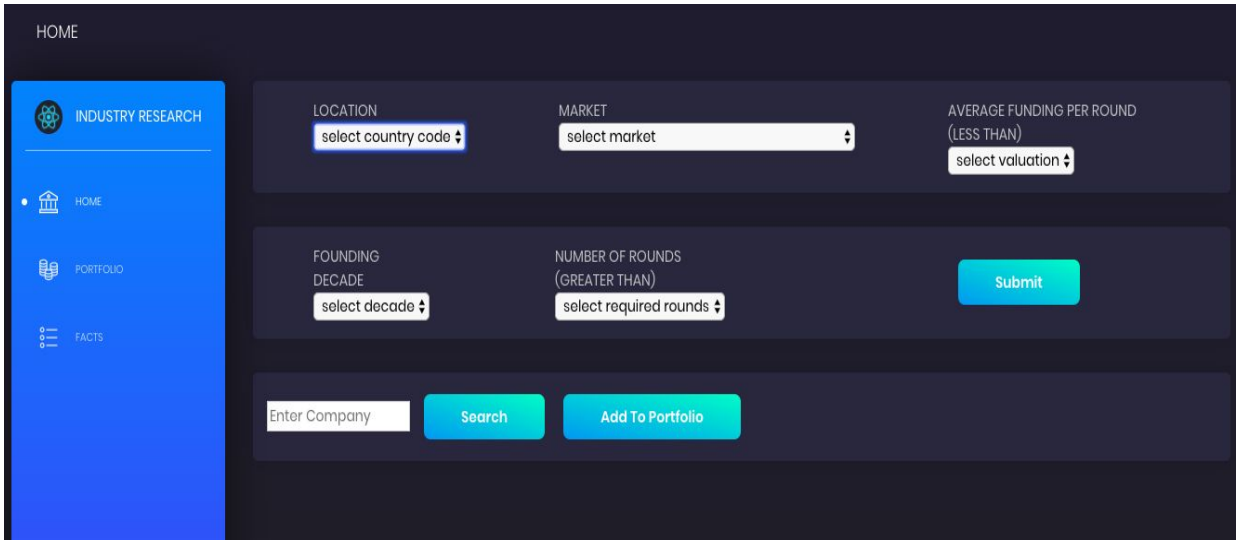
## 1. Introduction:

- **List of Group Members**
  - Anusha Chheda
  - Arnav Joshi
  - Daniel Ssemanda
  - Sarthak Jain

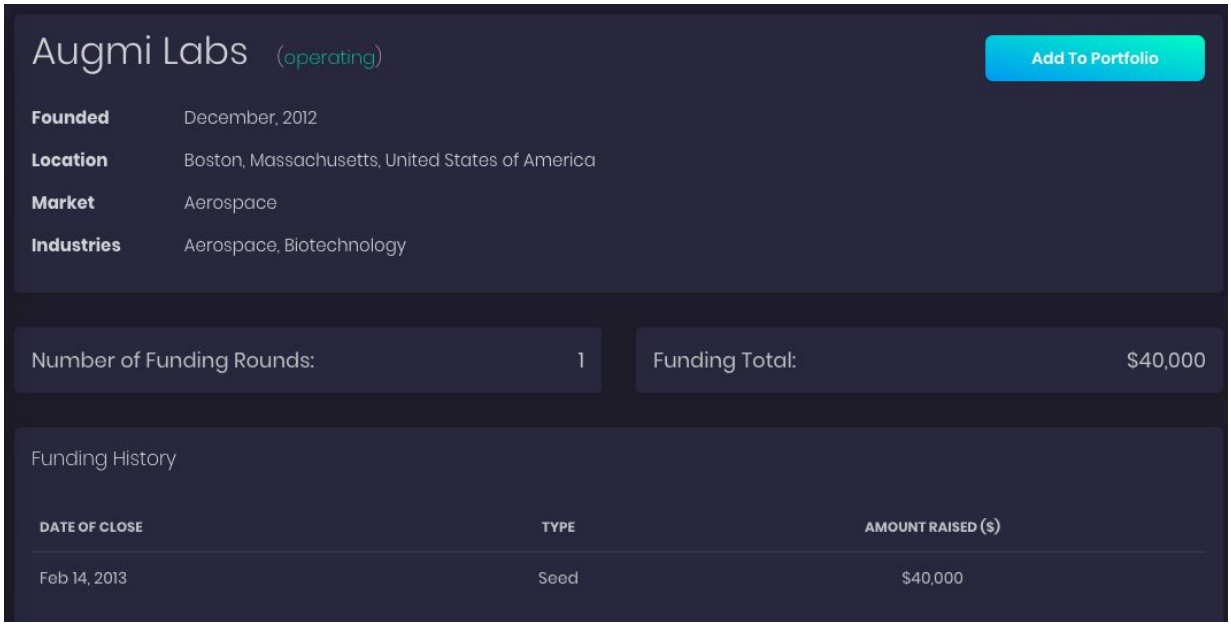
- **Target Problem**
  - There is a lot of data available through several media outlets about companies that are publicly listed on stock exchanges. However Venture Capitalists, business research analysts and other investors might find it difficult to obtain information about startups and companies not publicly listed.

- **Project Goals**
  - Our database holds information about startups from the early 1900s to very recent startups in 2014.
  - The goal of our website is to ensure that investors can get sufficient information about the trends in the startup space over the years.
  - We also want investors to look at company profiles i.e details about the company industry or location and details about the funding rounds that they carried out.
  - Investors can then add the companies that they find interesting to their portfolios. Our facts page attempts to highlight some such companies with interesting characteristics.

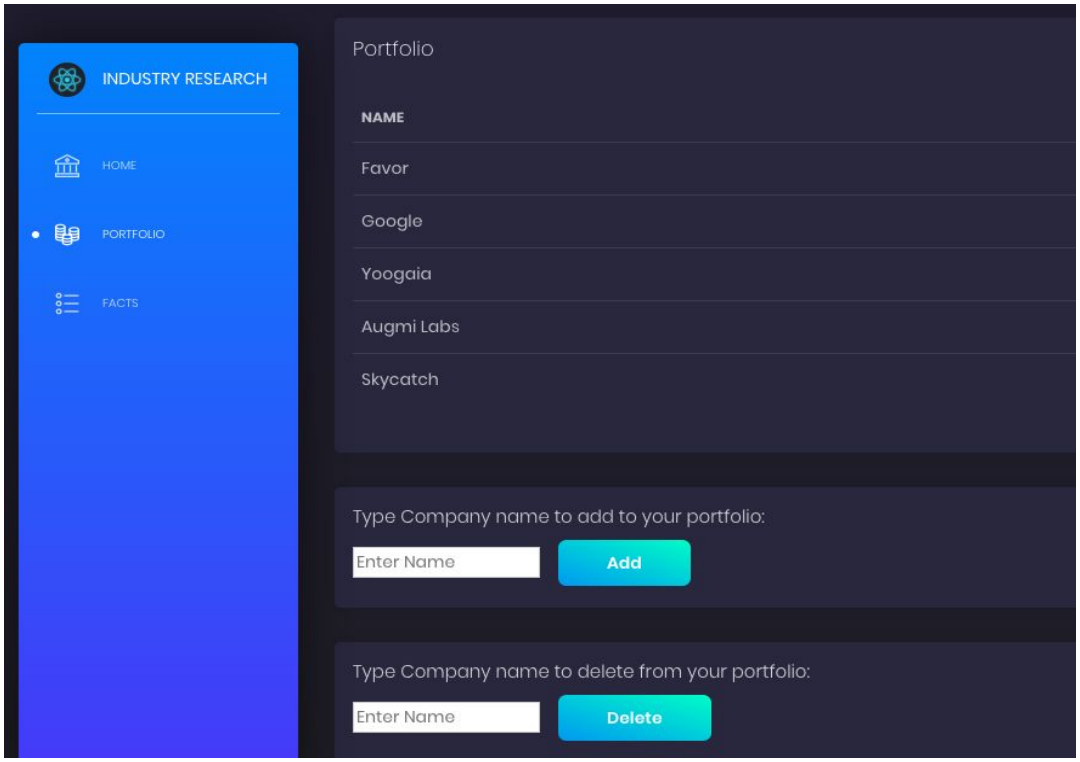
- **Description of Application Functionality**
  1. Home page:
     - The home page allows the user to filter companies based on country, market, average money raised in a funding round, the decade in which the company was founded and the number of funding rounds the company has carried out.
     - There is also an option to search company by name and view its profile.



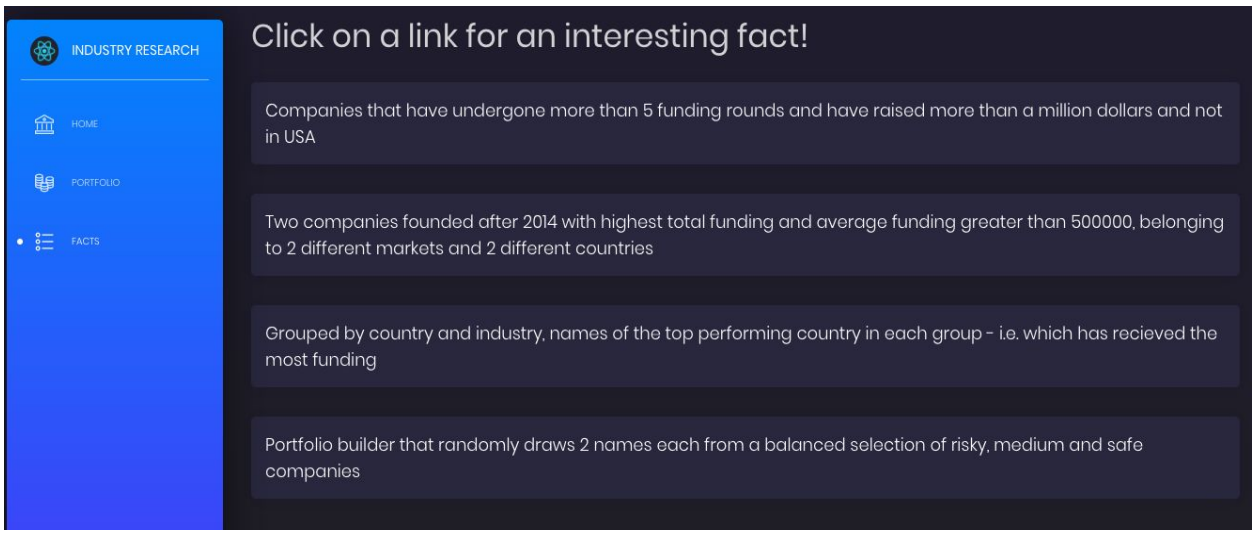
2. Company Profile Page:
    - On carrying out a filter search or on searching a company by its name, we can view the company profile that provides information about the company's status, founded date, location, market, industries, and details about funding rounds.




3. Portfolio:
    - Users can add the companies that the find interesting to their portfolio.
    - There are buttons on the home page, on the page that displays the companies after doing a filtered search, and on the company profile page that allow users to add companies to their portfolio.



4. Facts page
    - This page runs complex queries in order to suggest names of companies that have some interesting characteristics and might be worth adding to a user's portfolio.




## 2. Architecture

- **List of technologies used:**
    - Google Colab (Data Cleaning)
    - Python Pandas (Data Cleaning)
    - AWS RDS (Database instance)
    - Node.js (Server Side)
    - React.js (Client Side)
    - Oracle SQL Developer (Query testing)

- **Description of System Application/ Architecture**

  **client/public/**

  index.html - The root HTML file that React renders all of its components on.

  **client/src**

  index.js - The javascript file first invoked when the React script is started.

  **client/src/views/**

  This folder contains the files Home.js, Portfolio,js, and Facts.js which display the three main pages of the application. It also contains TableView.js which displays a list of companies and SingleCompany.js which displays the company profile for a single company.

  **client/src/routes.js**

  This file declares all the routes to the different pages such as the Home page or the portfolio page.

**client/package.json and server/package.json**

All the dependencies for the client and server are declared here.

**server/**

db_config.js - Configuration file for connecting with database.
index.js - Starts the server, links HTTP requests with their handlers in routes.js.
routes.js - Contains functions that handle HTTP requests and return responses to the client.
queryExecuter.js - Takes a query from the routes.js file, establishes a connection to the database and then executes the query.

## 3. Data

- **Link to the source of the dataset used:**
  http://info.crunchbase.com/about/crunchbase-data-exports/

- **Description of the datasets**
  The data has two main tables that we used
  - Companies Table: Contains details about name, industry details, current status, company location, funding and founding.
  - Rounds Table: Contains details about when and what type of funding the company received, which company carried out the funding round and what the funding round type was.

- **Relevant Summary Statistics**
  - Companies Table: 49,438 rows, 18 attributes - 24,454 rows post cleaning.
  - Rounds Table: 83,871 rows, 16 attributes - 46,406 rows post cleaning.

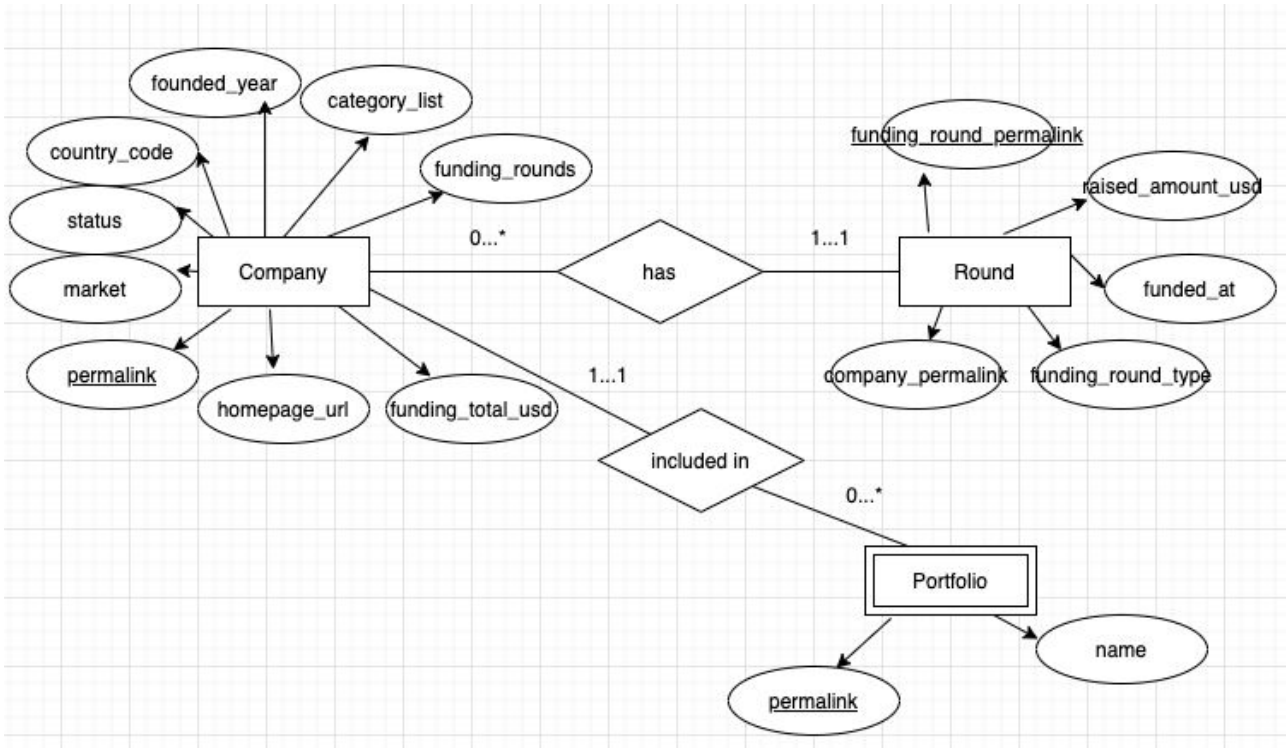
- **Explanation of how we use the data**
  - Companies Table: The data is used in the filtered search and search of company by name on the homepage. This data allows us to display the company profile and allows users to add companies to their portfolio. The queries on the facts page also utilise a lot of information for filtering through companies that is provided by this table.
  - Rounds Table: The data used in the filtered search (number of funding rounds) can be obtained using this table. This table is also used to provide information about the funding rounds of a company in the company profile. Many queries on the facts page have conditions on the number of funding rounds or the average amount raised in funding rounds which can be obtained using this table.

## 4. Database:

- **Data cleaning and ingestion procedure**
  - We used google colab and python pandas to pre-process the data.
  - We ensured that the company and round tables have no null values and kept only those companies for which all the rounds of funding are accounted for.
  - We dropped all duplicate rows and converted all the columns to the relevant data types.
  - We then downloaded the cleaned data files as .csv files and used Oracle SQL Developer to ingest the data into the database.

- **Entity Resolution Efforts**
    - The company table and the rounds table naturally provided us with two entities that we could use- company and round
    - We also found that the data had primary keys for both those entities - permalink for company and funding_round_permalink for the rounds
    - We also decided to create a weak entity portfolio which contains several companies, so that we could allow users to save the companies that they find interesting

- **ER Diagram**




- **Normal Form And Justification:**
    - All the entities are in 3NF.
    - Entities are clearly in 1NF as the primary keys for all the entities ensure that the combination of all the columns make a unique row every single time.
    - Entities are in 2NF as the attributes in each entity only seek to describe some characteristic particular to that entity. For example, the company entity has status which describes the status of the company but the round entity does not have it as it does not describe a funding round.
    - Entities are in 3NF as there are no transitional dependencies. This means that knowing the value of any one or two of the attributes in an entity cannot help us find out what the value of a third attribute is going to be. For example, knowing the market and founded_year of the company does not tell us anything about the country_code of the company.

## 5. Queries

Examples of some of the complex queries used in our application:

**a) Query to display a highlight of two companies having raised the highest total funds greater than $500,000, and belong to 2 different markets and 2 different countries and founded after 2014 in the facts page.**

```
WITH A AS(
 SELECT c.country_code, c.market, c.permalink
 FROM Company c
 WHERE c.founded_year >= 2014 AND c.country_code != 'USA'),
 B AS(
 SELECT A.permalink, SUM(r.raised_amount_usd) AS total_amt, AVG(r.raised_amount_usd) AS avg_amount
 FROM A JOIN Round r ON A.permalink = r.company_permalink
 GROUP BY A.permalink
 HAVING SUM(r.raised_amount_usd) >= 500000 AND AVG(r.raised_amount_usd) < 500000),
 C AS(
 SELECT MAX(B.total_amt) AS max_val
 FROM B JOIN Company c ON B.permalink = C.permalink
 ),
 D AS(
 SELECT B.permalink, c.name, c.market AS market_one, c.country_code AS country_one, B.total_amt,
C.max_val
 FROM B, C, Company c
 WHERE B.total_amt = C.max_val AND B.permalink = c.permalink
 ),
 E AS(
 SELECT MAX(B.total_amt) as max_val_2
 FROM B, D, Company c
 WHERE c.country_code NOT IN D.country_one AND c.market NOT IN D.market_one AND B.permalink =
c.permalink
 ),
 F AS(
 SELECT B.permalink, c.name, c.market AS market_two, c.country_code AS country_two
 FROM B, E, Company c
 WHERE E.max_val_2 = B.total_amt AND B.permalink = c.permalink)
 SELECT D.name, D.permalink
 FROM D JOIN B ON D.permalink = B.permalink
 UNION
 SELECT F.name, F.permalink
 FROM F JOIN B ON F.permalink = B.permalink;
```

**b) Grouped by country and industry, query that provides names of the top performing country in each group - i.e. which has recieved the most funding to provide additional information to the investor.**

```
WITH countries AS (
  SELECT DISTINCT country_code FROM company
 ),
 industries AS (
  SELECT DISTINCT market FROM company
 ),
 combo AS (
  SELECT * FROM countries, industries
 ),
 filled_table AS (
  SELECT combo.country_code, combo.market AS "INDUSTRY", c.name, c.permalink, c.funding_total_usd
  FROM combo JOIN company c ON combo.country_code = c.country_code AND combo.market = c.market
  WHERE ROWNUM <= 20 AND c.funding_total_usd >= ALL (
    SELECT company.funding_total_usd
    FROM company
    WHERE company.country_code = c.country_code AND company.market = c.market
  )
  ORDER BY combo.country_code, combo.market
 )
 SELECT f.name, f.permalink FROM filled_table f;
```

**c) Advanced portfolio builder that randomly draw 2 names each from a balanced selection of risky, medium and safe companies in that order where in each category has certain defined parameters.**

```
WITH risky
    AS (SELECT NAME, permalink,
               funding_total_usd
        FROM   company
        WHERE  status = 'operating'
               AND funding_total_usd < 0.75 * (SELECT Avg(funding_total_usd)
                                               FROM   company)
               AND founded_year >= 2012
               AND ( market = 'Biotechnology'
                     OR market = 'Clean Technology'
                     OR market = 'Health and Wellness'
                     OR market = 'Solar'
                     OR market = 'Biotechnology'
                     OR market = 'Startups' )),
    safe
    AS (SELECT NAME, permalink,
               funding_total_usd
        FROM   company
        WHERE  status = 'operating'
               AND funding_total_usd >= 0.75 * (SELECT Avg(funding_total_usd)
                                                FROM   company)
               AND funding_rounds >= 6
               AND country_code = 'USA'
               AND ( market = 'Software'
                     OR market = 'Consulting'
                     OR market = 'Universities'
                     OR market = 'Utilities'
                     OR market = 'Education'
                     OR market = 'Colleges'
                     OR market LIKE 'Big Data%' )),
    medium
    AS (SELECT NAME, permalink,
               funding_total_usd
        FROM   company
        WHERE  status = 'operating'
               AND funding_total_usd > (SELECT Avg(funding_total_usd)
                                        FROM   company)
               AND funding_rounds >= 3
               AND country_code = 'USA'
               AND market NOT LIKE '%Travel%'),
    p
    AS ((SELECT *
         FROM   (SELECT *
                 FROM   risky
                 ORDER  BY dbms_random.value)
         WHERE  rownum <= 2)
        UNION
        (SELECT *
         FROM   (SELECT *
                 FROM   medium
                 ORDER  BY dbms_random.value)
         WHERE  rownum <= 2)
         UNION
        (SELECT *
         FROM   (SELECT *
                 FROM   safe
                 ORDER  BY dbms_random.value)
         WHERE  rownum <= 2))
SELECT name, permalink
FROM   p;
```

**d) Query to filter companies based on user entered parameters such as country code, market, founding decade, funding rounds and amount raised.**

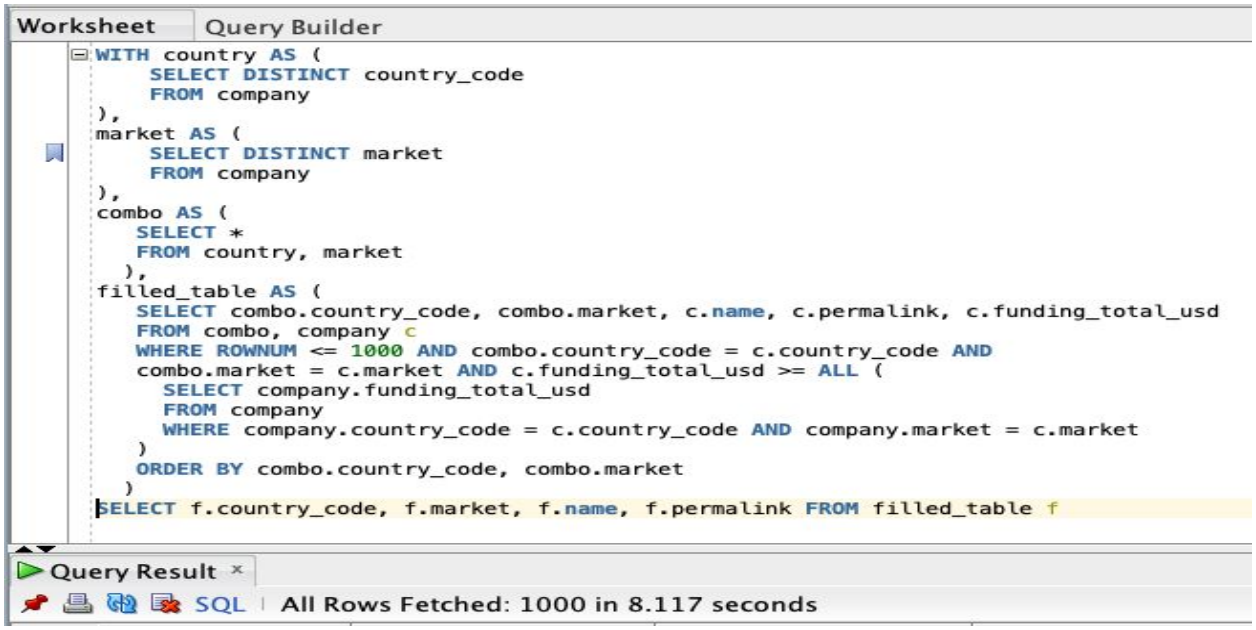
```
 SELECT DISTINCT name, permalink
FROM company
WHERE (ROWNUM <= 20) `;
if (country !== "select country code" && country !== 'null'){
  query = query + `AND (country_code = '${country}') `
}
if (year !== "select decade" && year !== 'null'){
  query = query + `AND ((FLOOR(founded_year/10)*10) >= '${year}') `
}
if (rounds !== "select required rounds" && rounds !== 'null'){
  query = query + `AND (funding_rounds >= '${rounds}') `
}
if (market !== "select market" && market !== 'null'){
  query = query + `AND (market = '${market}') `
}
if (valuation !== "select valuation" && valuation !== 'null'){
  query = query + `AND ((funding_total_usd/funding_rounds) <= '${valuation}') ;
```

**e) Query to filter out companies that have undergone more than 5 funding rounds and have raised more than a million dollars and not in USA.**

```
WITH A AS(
SELECT c.NAME, c.PERMALINK
FROM Company c JOIN Round r ON c.PERMALINK = r.COMPANY_PERMALINK
WHERE c.COUNTRY_CODE != 'USA'
GROUP BY c.PERMALINK, c.NAME
HAVING COUNT(r.FUNDING_ROUND_PERMALINK) > 5 AND SUM(r.RAISED_AMOUNT_USD) > 1000000)
SELECT c.NAME, c.PERMALINK
FROM A JOIN Company c ON c.PERMALINK = A.PERMALINK;
```

## 6. Performance evaluation

- **Recorded timings**
  Unoptimized:



Optimized:

Worksheet | Query Builder

```
CREATE INDEX comp_market ON company(country_code,market);
WITH combo AS (
   SELECT country_code, market FROM company
   GROUP BY country_code, market
),
filled_table AS (
   SELECT combo.country_code, combo.market, c.name, c.permalink, c.funding_total_usd
   FROM combo JOIN company c ON combo.country_code = c.country_code AND combo.market = c.market
   WHERE ROWNUM <= 1000 AND c.funding_total_usd >= ALL (
     SELECT company.funding_total_usd
     FROM company
     WHERE company.country_code = c.country_code AND company.market = c.market
   )
   ORDER BY combo.country_code, combo.market
)
SELECT f.country_code, f.market, f.name, f.permalink FROM filled_table f;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 1000 in 5.181 seconds

| | COUNTRY_CODE | MARKET | NAME | PERMALINK |
|---|---|---|---|---|
| 1 | ARE | Curated Web | HelloBooks | /organization/hellobooks |
| 2 | ARE | Fashion | GlamBox | /organization/glambox |
| 3 | ARE | Hardware + Software | Redington | /organization/redington |

- **Description of events being timed:**
  - We are executing a query that gets 1000 rows, where we select the companies with the highest funding total, grouped by country and market (this query corresponds to Fact3 on the Facts page).

- **Explanation of why optimizations improved the timings:**
  - The first optimization that we did was selecting country and market using a group by from the company table instead of selecting country and market separately. This improves the timings because instead of running through the company table twice and selecting distinct countries first and then distinct markets, we just go through the companies table once and using a group by on both country and market, to get distinct values.
  - The second optimization that we did was converting inefficient cross products to joins. The unoptimized query selects from two tables combo and company and puts the join condition in the where clause, which leads to a cross product between the tables and a lot of comparisons. Joining the tables by putting the join condition using the ‘JOIN’ keyword and not in the ‘WHERE’ clause reduces the number of comparisons, to optimize the query.
  - The third optimization that we did was creating an index on (country_code, market). As we group by both these attributes, the index speeds up our query.

## 7. Technical challenges faced

- **Ingesting data into the Oracle Database**
  - This process took many hours and even within this process it was possible for the load to fail after a few thousand rows so one would need to start again.

- **Connecting to the Oracle Database using node**
  - The official instructions we found online to set up our computers to use the ‘oracledb’ node package were incorrect, leading to major problems.
  - All group members had different versions of the Mac OSX operating system, which ended up requiring slightly different processes to get connected.
  - We needed extensive debugging and TA assistance to get connected.

- **Designing the front end**
    - No one on the team had extensive experience designing a web front end
    - We were still new to React JS and working with a React design template. Had to learn a lot of new details about the framework as we continued to build the project.

- **Working with our AWS database instance**
    - We underestimated the amount of space our table would take and ran out of storage on our AWS instance. We eventually had to increase the allocated storage for our application to work again.

## 7. Extra credit features

- **Link to the company website**
    - On clicking the name of a company, it redirects to the web and finds the company webpage and loads it.