

# Extract Transform Load

## Group Members:

- Grace Arhin
- Josh Mills
- Rodney Davermann
- Chithra Sundaram
- Jennifer Gaddie

**Question:** Create a forecasting model for the occupation that is most likely to win the next Bachelor/Bachelorette.



# ETL Process to get the data for the winner

## Extract

- Import needed data from two sources: Kaggle and Data World
- Both Files were CSV
- Prior to transform, we examined the data, and became familiar with it

## Transform

- Concatenate season with name to create a primary/foreign key
- Make the key all Upper Case
- Separate the id from last name to further make a primary key
- Replace the the last name that says none to X so we could make a primary key
- Took just the first letter of the last name so we could make a primary key
- Concatenate everything as a final key
- Take the first part of contestant id off because one sheet starts with 01 and the other sheet starts with 1
- Made it so both keys had the same format ex. 01\_firstname\_lastname check conid2 at the end
- Drop unneeded duplicated columns
- Split hometown column into cities and state

## Load

- Convert files back to csv so we can load into Postgres
- Built EDR to confirm correct SQL and Primary Key
- Created Database, and Tables to load to Postgres
- Use SQL Alchemy to pull from Postgres
- Check for tables to make sure they are available
- **NOTE:** We chose this path to allow for ease of data review later, and to allow for future test against additional hypothesis testing

| contestant_details |             |
|--------------------|-------------|
| Unnamed            | int         |
| Age                | int         |
| Eliminated         | varchar(50) |
| Name               | varchar(50) |
| Occupation         | varchar(50) |
| Outcome            | varchar(50) |
| Season             | int         |
| id2                | varchar(50) |
| last_name          | varchar(50) |
| first_letter       | char        |
| contestant_id      | varchar(50) |
| City               | varchar(50) |
| State              | varchar(50) |

| show_outcomes  |             |
|----------------|-------------|
| SHOW           | varchar(50) |
| SEASON         | int         |
| CONTESTANT     | varchar(50) |
| ELIMINATION-1  | varchar(50) |
| ELIMINATION-2  | varchar(50) |
| ELIMINATION-3  | varchar(50) |
| ELIMINATION-4  | varchar(50) |
| ELIMINATION-5  | varchar(50) |
| ELIMINATION-6  | varchar(50) |
| ELIMINATION-7  | varchar(50) |
| ELIMINATION-8  | varchar(50) |
| ELIMINATION-9  | varchar(50) |
| ELIMINATION-10 | varchar(50) |
| DATES-1        | varchar(50) |
| DATES-2        | varchar(50) |
| DATES-3        | varchar(50) |
| DATES-4        | varchar(50) |
| DATES-5        | varchar(50) |
| DATES-6        | varchar(50) |
| DATES-7        | varchar(50) |
| DATES-8        | varchar(50) |
| DATES-9        | varchar(50) |
| DATES-10       | varchar(50) |
| conid          | varchar(50) |
| conid2         | varchar(50) |

ERD Diagram  
screenshot

[etl\\_project github link](#)