*Notes:*

- *Read this entire take-home before starting.*
- *Please respond with a github (or similar) repo or gist.*
- *For Part 1&2, the research production process is intentionally left ambiguous. Feel free to make assumptions about our business from the data set.*
- *We are looking for your SQL skills, use of best practices, your thinking and creativity and the ability to add your own ideas.*

# Part 1: Analysis

Analyze and extract any interesting insights you can derive from the data set attached (each row represents the assignment of a job in our research queue, including some data about the analyst who received the assignment and the current state of the research queue). What can you infer? What do you think this means for us from a business perspective? Please use SQL and include your queries.
Optional: include a link or file with the database you've created from the attached file.

# Part 2: Data modeling

Assuming we are starting with the data set from Part 1 as our raw data table, how would you model this data in a data warehouse for analytical purposes? What tables would you create? What kinds of questions do you imagine business users would want to ask of this data, and how would they express them in your data model? Please use whatever tools you are comfortable with to answer this question and whatever flavor of SQL you are most familiar with.

# Part 3: SQL

Assume we are using these two tables:

customers table:

| customer_nbr | customer_name |
|---|---|
| 1 | Jim Brown |
| 2 | Jeff Gordon |
| 3 | Peter Green |
| 4 | Julie Peters |

orders table:

| order_nbr | order_date | customer_nbr | order_amt |
|---|---|---|---|
| 1 | 2008-10-01 | 1 | 15.50 |
| 2 | 2008-12-15 | 2 | 25.00 |
| 3 | 2009-01-02 | 1 | 18.00 |
| 4 | 2009-02-20 | 3 | 10.25 |
| 5 | 2009-03-05 | 1 | 30.00 |

Imagine your product manager comes to you with a query that she's not sure is correct. You do your best to guess what she was intending to query for and realize there are multiple issues.

Please explain each issue and include a fixed query addressing all issues.

```
SELECT
      customers.customer_name,
      SUM(COALESCE(orders.order_amt, 0)) AS total_2009
FROM
      customers
      LEFT OUTER JOIN orders ON (
            customers.customer_nbr = orders.customer_nbr
      )
WHERE
      orders.order_date >= '20090101'
GROUP BY
      customers.customer_name
```