Data Engineering interview assignment

Thank you for investing time to interview with BigSpring. This is a confidential document. Do not share.

We are a product driven early stage company building a B2B2C product. Our decision making backbone is data and a core value proposition is using data to drive ROI calculations. We are also setting up the right frameworks to build ML models on top of our data to scale out new use cases. A great Data Engineering team is critical to these goals.

Summary

In this assignment, you'll take a sample PostgreSQL database called Pagila and build a functional ETL-like workflow to create a summary roll-up table into a new database. You are welcome to use any tool like Airflow, Airbyte, DBT or custom scripts.

Setup

- Clone, setup and run the Pagila sample database locally. Docker process is ideal but go for what works for you.
    - Hint: if you would like to connect to the docker instance using tools like DBeaver pass in `-p 5432:5432` to the `docker run` command.
- Explore the schema and the tables in this sample database.
- An important table is `rental`

```
select * from rental r
```

rental 1

select * from rental r | Enter a SQL expression to filter results (use Ctrl+Space)

	rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	last_update
1	2	2005-05-24 17:54:33	1,525	459	2005-05-28 14:40:33	1	2020-02-15 21:30:53
2	3	2005-05-24 18:03:39	1,711	408	2005-06-01 17:12:39	1	2020-02-15 21:30:53
3	4	2005-05-24 18:04:41	2,452	333	2005-06-02 20:43:41	2	2020-02-15 21:30:53
4	5	2005-05-24 18:05:21	2,079	222	2005-06-01 23:33:21	1	2020-02-15 21:30:53
5	6	2005-05-24 18:08:07	2,792	549	2005-05-26 20:32:07	1	2020-02-15 21:30:53
6	7	2005-05-24 18:11:53	3,995	269	2005-05-29 15:34:53	2	2020-02-15 21:30:53
7	8	2005-05-24 18:31:46	2,346	239	2005-05-27 18:33:46	2	2020-02-15 21:30:53
8	9	2005-05-24 19:00:40	2,580	126	2005-05-27 19:22:40	1	2020-02-15 21:30:53
9	10	2005-05-24 19:02:21	1,824	399	2005-05-31 17:44:21	2	2020-02-15 21:30:53
10	11	2005-05-24 19:09:02	4,443	142	2005-06-02 15:56:02	2	2020-02-15 21:30:53
11	12	2005-05-24 19:19:27	1,584	261	2005-05-30 00:44:27	2	2020-02-15 21:30:53
12	13	2005-05-24 19:22:55	2,294	334	2005-05-29 23:28:55	1	2020-02-15 21:30:53
13	14	2005-05-24 19:31:15	2,701	446	2005-05-25 21:56:15	1	2020-02-15 21:30:53
14	15	2005-05-24 19:39:22	3,049	319	2005-06-02 22:30:22	1	2020-02-15 21:30:53
15	16	2005-05-24 19:43:11	389	316	2005-05-25 23:42:11	2	2020-02-15 21:30:53
16	17	2005-05-24 20:06:36	830	575	2005-05-26 19:43:36	1	2020-02-15 21:30:53

- Create a new database that will be the Rollup data

- In this new database, create a new table that will show the outstanding rentals for each week. Define a week in any reasonable way and state your assumptions.
    - Feel free to make minor changes to the schema as long the basic functional use cases are met or to highlight some new use cases.
    - The goal is to see how many movies were out for rental for a given week and how many were returned that week.
    - Sample schema and made up data below.

WeekBeginning	OutstandingRentals	ReturnedRentals
2018-01-08	800	200
2018-01-15	750	222

- Populate this new database table using a ETL tool, script, process of your choice
- Look for ways to do this in an incremental way so each weekly run will only update newly relevant rows.

Goals

You'll end up with 2 databases (one input and one output) and a set of code/scripts/config to load the output database. You'll have some high level documentation on how to set up the run process on another machine.

We are eager to see a mix of your infra setup/management, database exploration, SQL capabilities and workflow setup/management.

This table might drive a ML prediction system that will try to predict inventory that is needed per title. Please use your creativity to find efficient and interesting tricks and alternatives.

Non-Goals

You do not have to run these 2 databases in any production caliber environments. The process can be run manually in your local machine.

Deliverables

Please push any and all code, configurations, diagrams, README or details into this repo a day before we meet for the debrief. During the deep dive, plan to do a live demo and walk through your setup and the codebase.