CS 327E Final Project: Milestone 2, due Wednesday 12/06. **Due date is not flexible.**

This milestone has three sections. The first asks you to set up your environment for Trino. The second asks you to implement three federated queries that span multiple databases. The third asks you to evaluate Trino and summarize its trade-offs.

**Part 1: Deployment**

- Follow our setup guide to deploy Trino and MongoDB. Note: We are deploying MongoDB instead of using Atlas because Trino is currently incompatible with Atlas.

- Go back to `final-project-mongodb.ipynb` from Milestone 1. Change the `HOST` and `url` to point to your new MongoDB server. The `HOST` should be set to the Private IP of your Trino VM. The `url` should look like this:
  `url = "mongodb://{}:{}".format(HOST, PORT)`
  Do **not** change the port number. It needs to be 27017. Refer to this notebook for more details.

- Re-run the contents of your notebook to create the `final_project` collection and populate it with the ticketing data from Milestone 1.

- If you still have your college datasets in MySQL, Postgres, and BigQuery, go through this notebook to verify that you can federate to each of your databases. This is an optional step.

**Part 2: Federated Queries**

Create a Jupyter notebook and name it `final-project-trino.ipynb`. Using the Trino cli, implement the following three federated queries in your notebook. Refer to this notebook for code samples.

*Q1: Who are the shoppers who have reservations in a different city from their own city and who paid for their reservations in US dollars? For those shoppers, return their cust_id, email, along with their reservation count. Order the results by cust_id. Limit the results to 5 records.*

*Q2: Who are the shoppers who have bought non-stop tickets on American Airlines with a departure date between 01/01/2020 and 12/31/2024? Return the shopper's cust_id, email, dep_airport and dep_date. Order the results by dep_date. Limit the results to 5 records.*

*Q3: Which shoppers have reservations and tickets with matching dates such that the arr_date and dep_date are equal between reservations and ticketing? For such shoppers, sum up their pmt_amt and tik_amt and return this sum as trip_amount. Also, return the shoppers' cust_id, email, pmt_amt, tik_amt. Order the results by trip_amount in descending order. Limit the results to 5 records.*

**Part 3: Trino Evaluation**

Now that you have seen how to implement cross-database joins, it's time to conduct a more thorough evaluation of Trino. Think of other tests you can run to gain a more complete picture of Trino's functionality. Come up with some tests and run them from your notebook. You want to test a variety of scenarios that include DDL, DML, and SELECT statements. Then, write a short paragraph to summarize your findings.

Your summary needs to be specific and supported by your test results. For example: "CREATE TABLE statement X against MySQL resulted in Y" or "BEGIN TRANSACTION and COMMIT" against Postgres resulted in Z". Write your summary as a Markdown comment on the last cell of the notebook.

CS 327E Final Project Milestone 2 Rubric
**Due Date: 12/06/23**

| | |
|---|---|
| Part 1: Deploy Trino and MongoDB on a Compute Engine VM as per our guide<br>**-10** Trino service does not start on VM<br>**-10** mongod is not active on VM | 20 |
| Part 1: Update the HOST and url variables in `final-project-mongodb.ipynb` and re-run the cells to populate the ticketing collection with 100 documents.<br>**-10** MongoDB server on Trino VM does not contain a ticketing collection<br>**-8** Ticketing collection is empty or is missing documents | 10 |
| Part 2: Implement federated queries Q1, Q2, and Q3.<br>**-3** for each missing or incorrect join clause<br>**-2** for each missing or incorrect where clause<br>**-2** for each missing or incorrect select clause<br>**-2** for each missing or incorrect order by clause<br>**-1** for each missing or incorrect limit clause | 30 |
| Part 3: Test a wide variety of scenarios with query federation to understand trade-offs.<br>**-10** for ignoring or missing at least one major scenario<br>**-10** for ignoring or missing at least one database system<br>**-10** for lack of logical progression or organization<br>**-10** for summary that is vague or unsupported | 40 |
| `final-project-trino.ipynb` and `final-project-mongodb.ipynb` pushed to your group's private repo on GitHub. Your milestone **will not** be graded without this submission. | **Required** |
| `submission.json` submitted into Canvas. Your project **will not** be graded without this submission. The file should have the following schema:<br><br>`{`<br>`  "commit-id": "your most recent commit ID from GitHub",`<br>`  "project-id": "your project ID from GCP"`<br>`}`<br><br>Example:<br><br>`{`<br>`  "commit-id": "dab96492ac7d906368ac9c7a17cb0dbd670923d9",`<br>`  "project-id": "some-project-id"`<br>`}` | **Required** |
| **Total Credit:** | **100** |