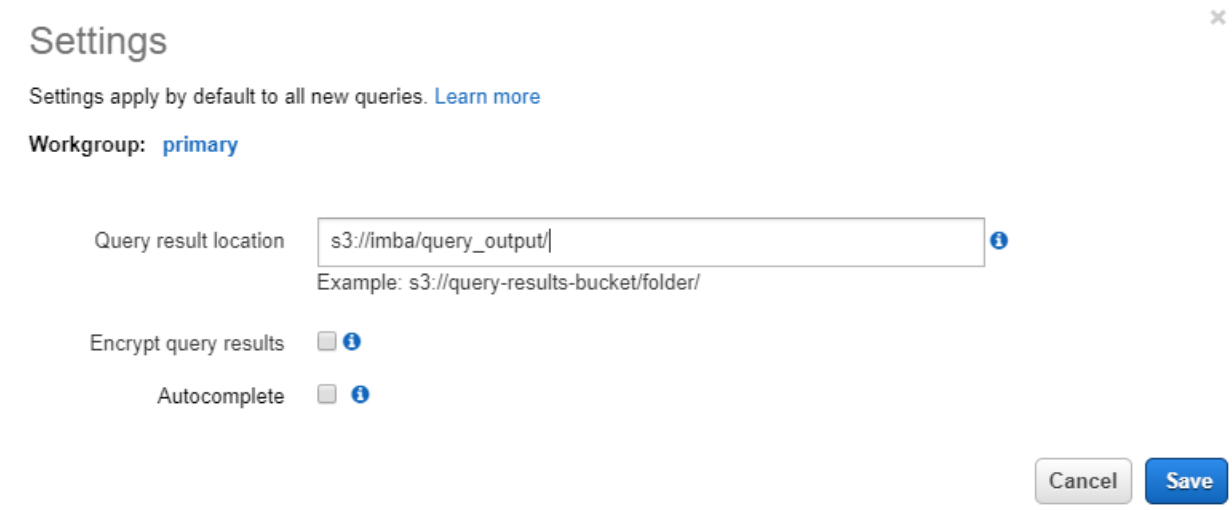
Project instructions

AWS Glue

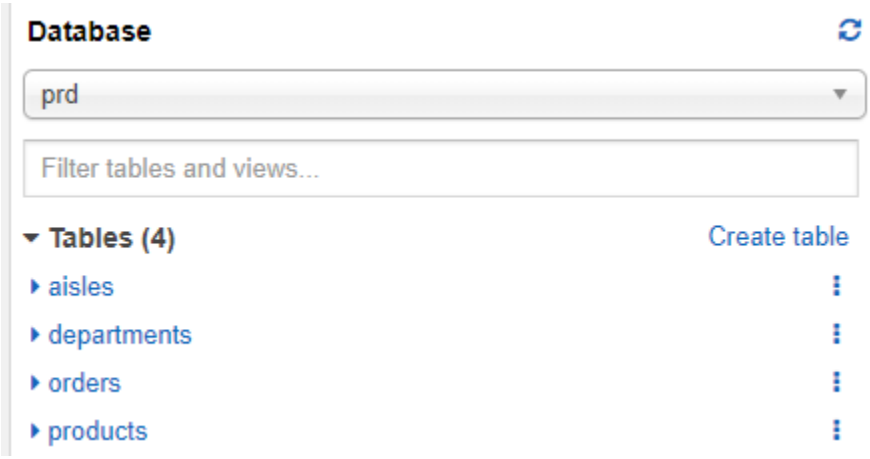
1. Download imba.zip from share drive, unzip all the files.
2. Create an s3 bucket with name imba_<put your name here> then create the following folders:
   data/aisles
   data/departments
   data/orders
   data/products
   data/order_products
3. Upload files to the corresponding directory (note: gzip both order_products__prior.csv and order_products__train.csv in gitbash, and upload all of them to data/order_products).
4. Go to AWS Glue service, click on Crawlers and Add crawler.
5. Give the crawler a name, e.g. imba, click next.
6. Choose Data stores as Crawler source type, click next.
7. Choose s3 as data store and specify the Include path as: s3://ibma_<put your name here>/data, click next.
8. Do not add another data store and click next.
9. Select Create an IAM role and type a name (any name will do) in the text box, click next.
10. Specify Frequency as Run on demand, click next.
11. Create a new database by clicking Add database, name it as prod, click next.
12. Review the options and click Finish.
13. Select the crawler you just created and click Run crawler button.

AWS Athena

1. Go to AWS Athena service, click Settings button located on the top right of the page, type in the Query result location as s3://<your s3 bucket>/query_output/:



2. Go to AWS Athena service, select prod data on the left pane, you should see four tables are created.



3. Check the fields for each table by expanding the table, as shown below:

4. You might notice products table is malformed, this is because crawler has problems to recognize csv files with double quote in content. Instead, manually create the table by running below command in query pane(remember to update the s3 location to your file location `'s3://imba_<put your name here>/data/products'`):
   a. First drop the table by running query: DROP TABLE IF EXISTS products;
   b. Now re-create the table by running below query:

```
CREATE EXTERNAL TABLE `products`(
  `product_id` string COMMENT 'from deserializer',
  `product_name` string COMMENT 'from deserializer',
  `aisle_id` string COMMENT 'from deserializer',
  `department_id` string COMMENT 'from deserializer')
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  'escapeChar'='\\',
  'quoteChar'='\"',
  'separatorChar'=',')
STORED AS INPUTFORMAT
```

```
  'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

  's3://imba_<put your name here>/data/products'

TBLPROPERTIES (

  'skip.header.line.count'='1')
```

5. Do some exploration on these tables and make sure you understand the context of them.
6. Design a query which join orders table and order_products table together, and filter on eval_set = ‘prior’