

# 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 or any other way you like, and upload the gzip files 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://**imba-<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:// **imba-<put your name here>**/query\_output/:

## Settings

Settings apply by default to all new queries. [Learn more](#)

Workgroup: **primary**

Query result location  ⓘ  
Example: s3://query-results-bucket/folder/


Encrypt query results ☐ ⓘ

Autocomplete ☐ ⓘ

Cancel

Save

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

**Database** 

prd

Filter tables and views...

▼ Tables (4)

Create table

▶ aisles

⋮

▶ departments

⋮

▶ orders

⋮

▶ products

⋮

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

▼ **Tables (4)**

Create table

▼ aisles

⋮

aisle\_id (bigint)

aisle (string)

▼ departments

⋮

department\_id (bigint)

department (string)

▼ orders

⋮

order\_id (bigint)

user\_id (bigint)

eval\_set (string)

order\_number (bigint)

order\_dow (bigint)

order\_hour\_of\_day (bigint)

days\_since\_prior\_order (double)

▶ products

⋮

4. Do some exploration on these tables and make sure you understand the context of them.
5. Design a query which join orders table and order\_products table together, and filter on eval\_set = 'prior'