

## AWS Glue DataBrew

### Create dataset

1. Open AWS Glue DataBrew service from aws console
2. Click on DATASETS on the navigation bar on the left, and click “Connect new dataset”.
3. Name the dataset name as “order-products-prior”, select “Amazon S3” as data source and put the s3 location for your order\_products\_prior table (s3://imba/features/order\_products\_prior/), select PARQUET as file type and then click “Create dataset”.
4. You should see a new Dataset is created:

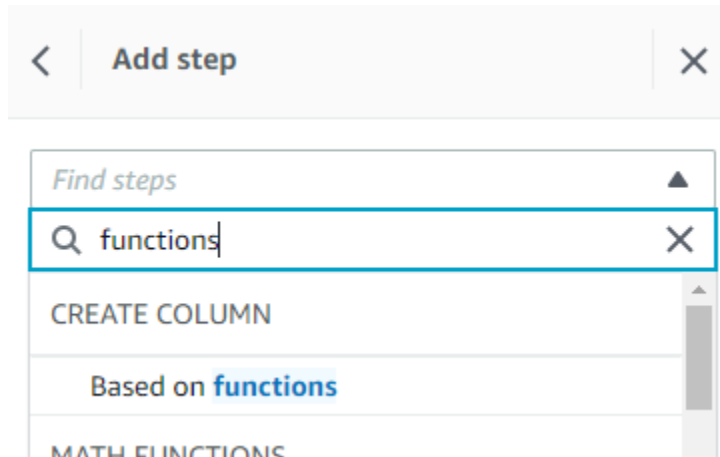
| <input type="checkbox"/> | Dataset name         | Data type | Data profile | Source | Location   |
|--------------------------|----------------------|-----------|--------------|--------|--|
| <input type="checkbox"/> | order-products-prior | parquet   | -            | S3     | s3://imba/features/order_products_prior/ <a href="#">🔗</a> |

### Create Projects

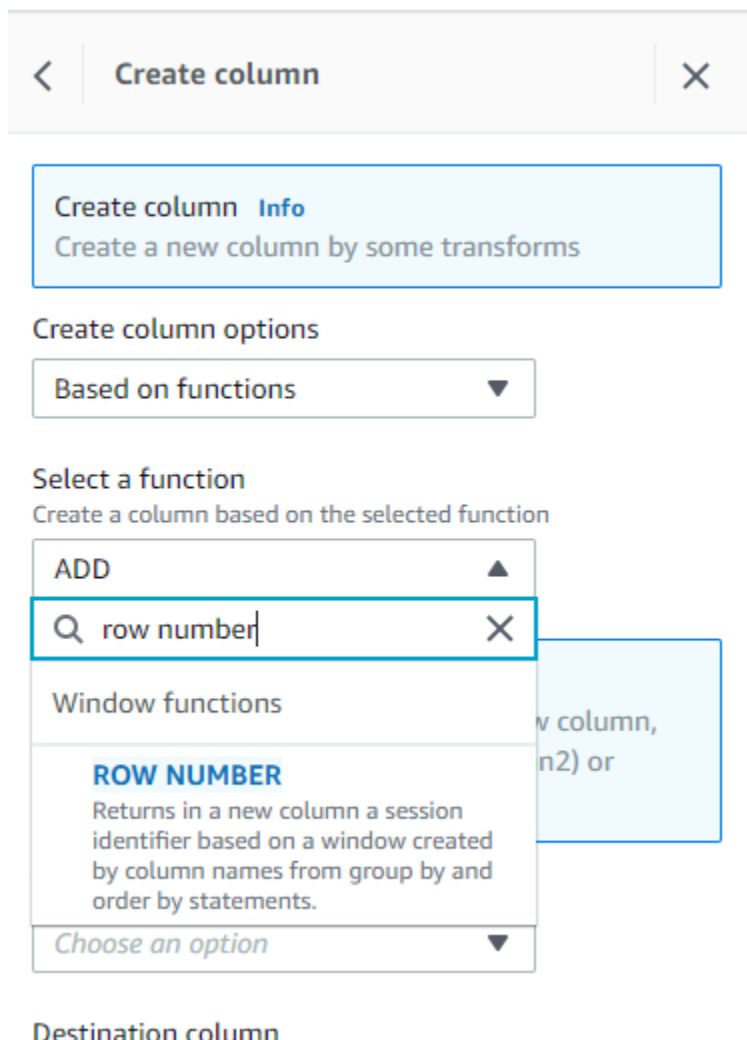
1. Select PROJECTS from the navigation bar from the left, click Create project.
2. Put the project as “prd\_features”, enter a Recipe name as “prd-features-recipe”.
3. Select the dataset you just created: order-products-prior.
4. Under Permissions, select Create new IAM role, put the role suffix as “imba” and click Create project.
5. Wait until the project creation is complete.

### Create recipe

1. Click Add step from the right, search functions in the search bar and select Based on functions, see below:



2. From Select a function, search row number in the search bar and select the "ROW NUMBER" function, see below:



3. Select order\_number as order by column and user\_id, product\_id as group by columns, name the Destination column as product\_seq\_time, see below:

Name of columns to order by with - optional

order\_number ▼

Name of columns to group by with

▼

# user\_id ✕

# product\_id ✕

Destination column

Name of the column created with extracted values

product\_seq\_time

Valid characters are alphanumeric, underscore, and space

Apply transform to

☒ All rows (500 rows)

Transformation will be applied to all rows  
in the dataset

4. Click Apply to apply the step.
5. Add another step, select Change type:

< Add step ✕

Find steps ▼

☰ COLUMN ACTIONS Info

Rename

Change type

Move column

Duplicate

Delete

Change the column product\_seq\_time to string and click Apply:

< **Change data type** ×

Change data type [Info](#)  
Change the data type of column

Source column  
product\_seq\_time ▼

Change type to  
ABC string ▼

Text or character values

[Preview changes](#)

Cancel

Delete step

Apply

6. Add another step, search and select Categorical mapping or relabeling function from the “Find steps” dropdown list:

< **Add step** ×

Find steps ▲

Q map

×

CATEGORICAL DATA ENCODING

Categorical **map**ping or relabeling

Move column

Duplicate

Delete

Make the configuration similar to below and click Apply:



## Categorically map column



### Categorical mapping [Info](#)

Map one or more of your categorical values to numeric or other values

#### Source column

Select a column to perform categorical mapping

product\_seq\_time



#### Mapping options

- ☐ Map top 1 values
- ☐ Map all values (4 values)
- ☒ Custom map values

#### Map values

☒ Map values to numeric values

Existing values

New value

1

1

[+ Add another map value](#)

#### Other values

- ☒ Map all other values 0
- ☐ Delete all rows with other values
- ☐ Keep all others values the same

#### Destination column

Name of the column created with mapped values

prod\_first

#### Apply transform to

- ☒ All rows (500 rows)  
Transformation will be applied to all rows in the dataset
- ☐ Filtered rows - 0 filters applied (500/500 rows)  
Transformation will be applied to filtered rows in the grid

[Preview changes](#)

Cancel

Apply

7. Use Categorical mapping again to create another column called `prod_second` and click Apply:

< Categorical map column ×

Categorical mapping [Info](#)  
Map one or more of your categorical values to numeric or other values

**Source column**  
Select a column to perform categorical mapping  
product\_seq\_time ▼

**Mapping options**  
☐ Map top 5 values  
☐ Map all values (4 values)  
☒ Custom map values

**Map values** ☒ Map values to numeric values

| Existing values | New value |
|-----------------|-----------|
| 2               | 1         |

+ Add another map value

**Other values**  
☒ Map all other values 0  
☐ Delete all rows with other values  
☐ Keep all others values the same

**Destination column**  
Name of the column created with mapped values  
prod\_second


**Apply transform to**  
☒ All rows (500 rows)  
Transformation will be applied to all rows in the dataset  
☐ Filtered rows - 0 filters applied (500/500 rows)  
Transformation will be applied to filtered rows in the grid


[Preview changes](#)


Cancel Delete step Apply


8. Add another step, scroll down and select “Group by and aggregate columns” (note: sometimes you need to search SUM aggregate functions to make the “GROUP BY” step available):

< **Add step** ×

 **CREATE COLUMN** [Info](#)  
Based on functions  
Create a flag column

 **NEST-UNNEST**  
Unnest  
Nest

 **PIVOT** [Info](#)  
Pivot - rows to columns  
Unpivot - columns to rows  
Transpose - columns to rows

 **GROUP BY** [Info](#)  
**Group by and aggregate columns**

9. Have below configuration ready in this step and click finish:

**Group**

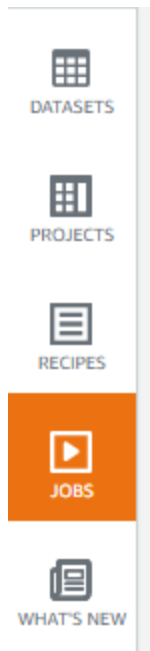
**Column list**  
Add column with aggregation for your grouped table

|    | Column name     | Aggregate  | New column name    | New column type |                        |
|----|-----------------|------------|--------------------|-----------------|------------------------|
| :: | # product_id ▼  | Group by ▼ | product_id         | # Long ▼        | <a href="#">Remove</a> |
| :: | # product_id ▼  | Count ▼    | prod_orders        | # Integer ▼     | <a href="#">Remove</a> |
| :: | # reordered ▼   | Sum ▼      | prod_reorders      | # Integer ▼     | <a href="#">Remove</a> |
| :: | # prod_first ▼  | Sum ▼      | prod_first_orders  | # Integer ▼     | <a href="#">Remove</a> |
| :: | # prod_second ▼ | Sum ▼      | prod_second_orders | # Integer ▼     | <a href="#">Remove</a> |

Add another column

**Group type**  
☒ Group as new table (replaces all existing columns with new columns)

10. Upon completion of all steps, navigate to JOBS on the left and click Create job:



11. Name the job as “prd-features-job”, for Job input please select Project and choose the “prd-features” project you just created.
12. For Job output settings, choose PARQUET as File type and enter the S3 location as: s3://**imba**/features/prd\_feature\_db/.
13. Leave everything else as default and go to Permissions at the bottom, re-use the role AWSGlueDataBrewServiceRole-imba.
14. Click Settings and select Replace output files for each job run:



Settings

×

File output storage

☐ Create a new folder for each job run
 

Under specified S3 path, a new folder will be created for each job run and for each output file type. The output folder and file name contains job name and job run time. Example:  
 s3://bucket/myfolder/jobname\_10may2020\_timestamp/filetype\_compression/jobname\_10may2020\_timestamp\_part1.csv

☒ Replace output files for each job run
 

Flat output files will be created under the specified S3 path. For every job run, the previous output files will be replaced with files from the latest job run. You can enable bucket versioning to be able to restore previous file versions. Example:  
 s3://bucket/myfolder/jobname\_part1.csv

Custom partition by column values

Partition by unique values of columns. The file is partitioned and stored in a folder path based on the order of columns provided. Example: A file partitioned by Column A and Column B is stored at this S3 path: s3://output file path.../Column A/Column B/

Cancel

Save

15. Open a new AWS console tab in browser and go to IAM, select the role `AWSGlueDataBrewServiceRole-imba` and add `AmazonS3FullAccess` permission to it:

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

▼ Permissions policies (4 policies applied)

Policy name ▼

☐ AmazonS3FullAccess

16. Click Create and run job, you should see a few files are created in:  
`s3://imba/features/prd_feature_db/`
17. Repeat the process for `user_features_1`, `user_features_2` and `up_features` based on the SQL queries from project part 2.

## **AWS glue development endpoint**

Develop a notebook using glue development endpoint, which achieves below:

1. Join up\_features, prd\_features, user\_features\_1 and user\_features\_2 into one dataframe
2. Write the output as a single csv file to s3 bucket.