**Glue Job Name :** build\_order\_caa\_627

We are creating a glue job from the console.

**Reusing:**

* + We are using some pieces of ETL code generated by Glue.
  + We are directly using the data source and apply mapping glue data frame code what’s been generated

**Modification:**

Converting to spark dataframe

* + We are converting the glue frame it to the spark dataframe.

Variables

* + **prefix='order‘** - We are passing this variable to be added as a prefix dynamically to build the key column name and partition column name
  + **key\_list = ['order\_id','customer\_id']** - This particular list of variables on which key columns will be built
  + **job\_name=args['JOB\_NAME']** – This is job name
  + **partition\_col="order\_purchase\_timestamp“** - This is where we are specifying the column on which we will be doing the partition
  + **audit\_cols=['foundation\_program\_nam','foundation\_ins\_dt']** – These are the audit column names

Function calling

We are currently calling three separate functions

* **hash\_primary\_key** - This particular function will build the hashed key out of the primary key columns .
* **add\_auditcol** – This will add the audit columns to the dataframe 'foundation\_program\_nam','foundation\_ins\_dt'
* **add\_year\_partition** : This particular function creates the year column on which it will be partitioned while writing to target. We may have different partitioning logic. So it’s not always we will be using this function.

Converting back to dynamic dataframe

* + We are converting the spark data frame back to the dynamic dataframe.

Writing to Sink

* + We are finally write the data to the foundation bucket with this final step

**One thing to add here is that, it’s a job in progress. So it’s subjected to changes.**