



Amazon Web Services Data Engineering Immersion Day

Extract, Transform and Load Data Lake with Glue

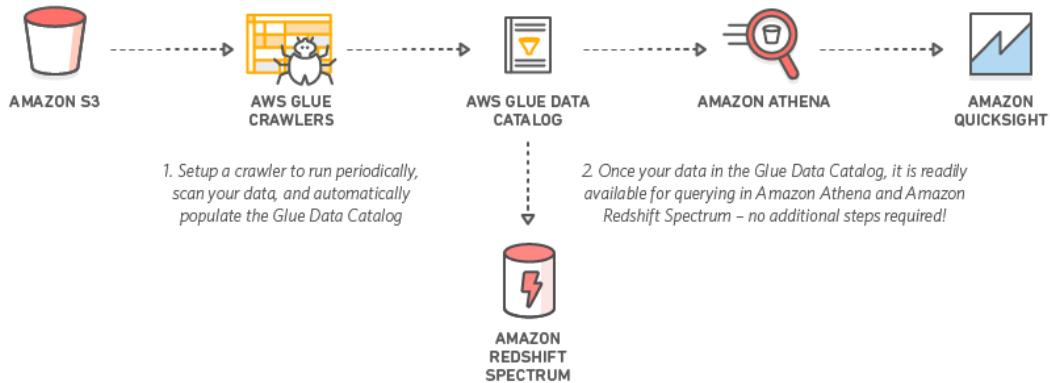
Jun 2019

Table of Contents

<i>Introduction</i>	2
Prerequisites:	2
Tasks Completed in this Lab:	2
Getting Started	2
Create Glue Crawler for initial full load data	3
Data Validation Exercise	8
Data ETL Exercise	9
Create Crawler for Parquet Files	14
Next Steps	17
Create Crawler for ongoing replication (optional)	17

Introduction

This lab will give you an understanding of the AWS Glue – a fully managed data catalog and ETL service, as well as Athena and Quicksight for querying and visualization the data you import.



Prerequisites:

The DMS Lab is a prerequisite for this lab.

Tasks Completed in this Lab:

In this lab you will be completing the following tasks:

1. [Create Glue crawler for initial data](#)
2. [Create Glue crawler for ongoing replication \(optional\)](#)
3. [Create Glue ETL to transform CSV data to Parquet format](#)

Getting Started

Navigate to the AWS Glue service.

AWS services

glue

AWS Glue
AWS Glue is a fully managed ETL (extract, transform, and load) service

EC2 Database Migration Service S3

CloudWatch AWS Glue

> All services

Create Glue Crawler for initial full load data

1. On the AWS Glue menu, select **Crawlers**.

The screenshot shows the AWS Glue service interface. On the left, there's a sidebar with navigation links: AWS Glue, Data catalog, Databases, Tables, Connections, **Crawlers** (which is selected and highlighted in orange), Classifiers, ETL, Jobs, Triggers, and Dev endpoints. The main area is titled "Crawlers" and contains a sub-header: "A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog." Below this is a search bar with buttons for "Add crawler", "Run crawler", "Action", and "Filter by attributes". A message says "Showing: 0 - 0 < > 🔍 ⓘ". The main table has columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. A large blue button labeled "Add crawler" is visible. A message at the top right says "You don't have any crawlers yet."

2. Click **Add crawler**.
3. Enter the crawler name for initial data load. This name should be descriptive and easily recognized (e.g., "glue-lab-crawler").
4. Optionally, enter the description. This should also be descriptive and easily recognized and **Click Next**.

This is a screenshot of the "Add crawler" wizard. The title bar says "Add crawler". On the left, a sidebar lists steps: **Crawler info** (selected), Crawler source type, Data store, IAM Role, Schedule, Output, and Review all steps. The main panel is titled "Add information about your crawler". It contains a "Crawler name" field with "glue-lab-crawler" entered. Below it are two optional sections: "Tags, description, security configuration, and classifiers (optional)" and "Catalog options (optional)". At the bottom right is a "Next" button.

5. Choose **Crawler Source Type** as Data Source and **Click Next**

This is a screenshot of the "Add crawler" wizard, Step 2. The title bar says "Add crawler". The sidebar still shows "Crawler info" is selected. The main panel is titled "Specify crawler source type". It contains a note: "Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores." Below this is a "Crawler source type" section with two radio buttons: "Data stores" (selected) and "Existing catalog tables". At the bottom right are "Back" and "Next" buttons.

6. On the **Add a data store** page, make the following selections:

- For Choose a data store, click the drop-down box and select **S3**.
 - For Crawl data in, select **Specified path in my account**.
 - For **Include path**, click the little folder icon, browse to the tickets folder e.g., "s3://xxxxx/tickets"
- 7. Click Next.**

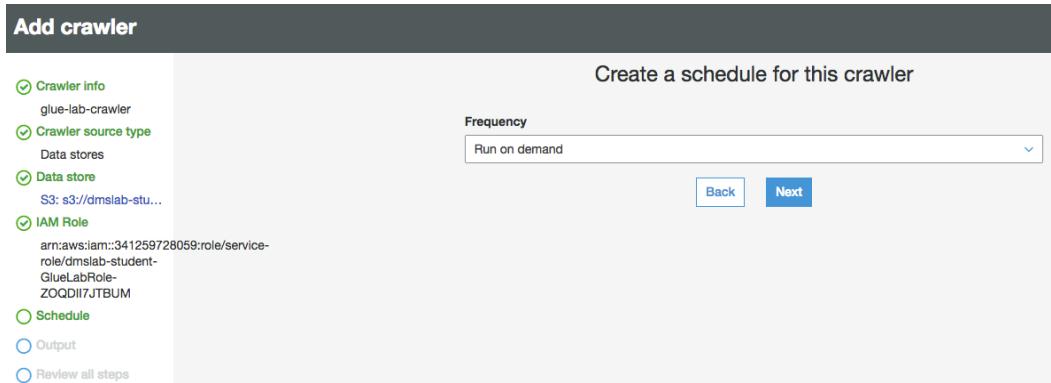
- 8. On the Add another data store page, select **No.** and Click Next.**

- 9. On the Choose an IAM role page, make the following selections:**

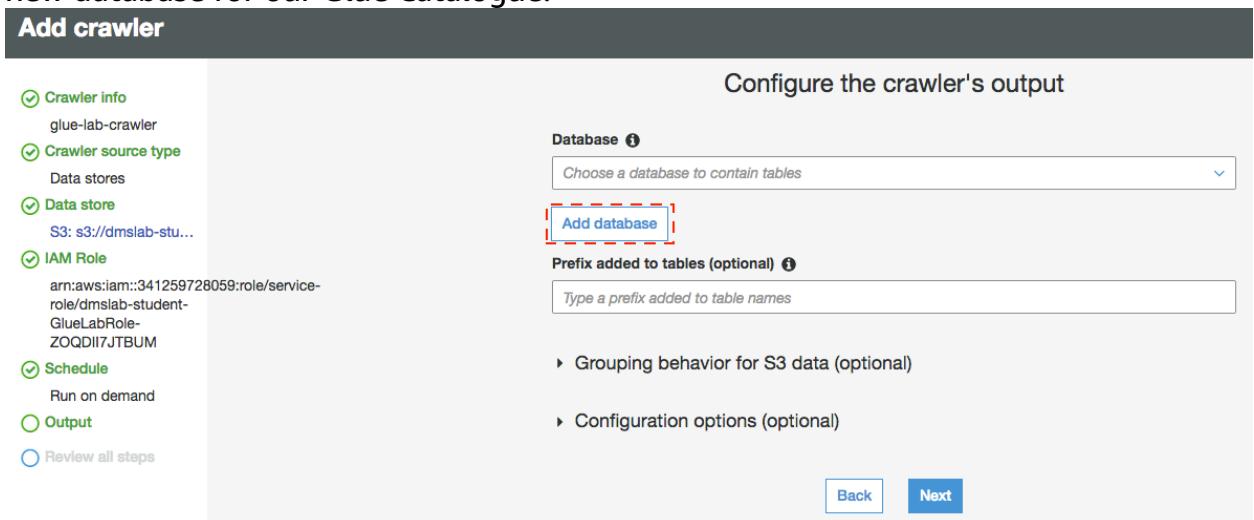
- Select **Choose an existing IAM role**.
- For **IAM role**, select the **GlueLabRole** that was created at the initial environment setup

- 10. Click Next.**

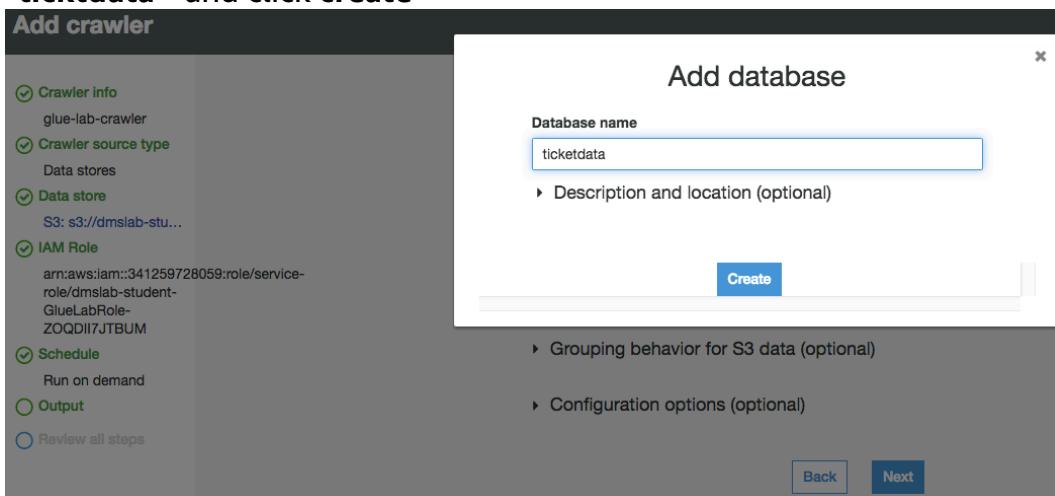
- 11. On the Create a schedule for this crawler page, for Frequency, select **Run on demand** and Click Next.**



12. On the Configure the crawler's output page, click **Add database** to create a new database for our Glue Catalogue.



13. Give Catalog database name as per your convenient choice for example "ticketdata" and click **create**



14. For Prefix added to tables (optional), leave the field empty.

15. For Configuration options (optional), select **Add new columns only** and keep the remaining default configuration options and Click **Next**.

Add crawler

Configure the crawler's output

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule
Run on demand

Output

Review all steps

Database ticketdata

Prefix added to tables (optional) Type a prefix added to table names

Grouping behavior for S3 data (optional)

Configuration options (optional)

During the crawler run, all schema changes are logged.

When the crawler detects schema changes in the data store, how should AWS Glue handle table updates in the data catalog?

Update the table definition in the data catalog.
 Add new columns only!
 Ignore the change and don't update the table in the data catalog. ⓘ

Update all new and existing partitions with metadata from the table. ⓘ

How should AWS Glue handle deleted objects in the data store?

Delete tables and partitions from the data catalog.
 Ignore the change and don't update the table in the data catalog.
 Mark the table as deprecated in the data catalog. ⓘ

Back **Next**

16. Review the summary page noting the Include path and Database output and Click **Finish**. The crawler is now ready to run.

Crawler info

Name: glue-lab-crawler
Tags: -
Use Lake Formation Data Catalog: false

Data stores

Data store: S3
Include path: s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets
Exclude patterns:

IAM role

IAM role: arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule

Schedule: Run on demand

Output

Database: ticketdata
Prefix added to tables (optional): false
Create a single schema for each S3 path: false
Configuration options:
 Add new columns only!
 Update the table definition in the data catalog for all data stores except S3. For tables that map to S3 data, add new columns only.
 Update the table definition in the data catalog for all data stores except S3. For tables that map to S3 data, add new columns only.
Object deletion in the data store: Mark the table as deprecated in the data catalog.

Back **Finish**

17. Click **Run it now**.

AWS Glue

Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
Settings

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler glue-lab-crawler was created to run on demand. [Run it now!](#)

Add crawler Run crawler Action Filter or search for crawlers... User preferences Showing: 1 - 1

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-crawler		Glue	Ready		0 secs	0 secs	0	0

Crawler will change status from starting to stopping, wait until crawler comes back to ready state, you can see that it has created 15 tables.

AWS Glue

Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
Settings

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler "glue-lab-crawler" completed and made the following changes: 15 tables created, 0 tables updated. See the tables created in database ticketdata.

Add crawler Run crawler Action Filter or search for crawlers... User preferences Showing: 1 - 1

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-crawler		Glue	Ready	Logs	1 min	1 min	0	15

18. In the AWS Glue navigation pane, click **Databases > Tables**. (You can also click the database name (e.g., "ticketdata" to browse the tables.).

AWS Glue

Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
Settings

ETL
Jobs
ML Transforms
Triggers
Dev endpoints
Notebooks

Security
Security configurations

Tutorials
Add crawler
Explore table

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Add tables Action Filter or search for tables... Save view

Name	Database	Location	Classification	Last updated
mlb_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
name_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
nfl_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
nfl_stadium_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
person	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
player	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
seat	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
seat_type	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
sport_division	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
sport_league	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
sport_location	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
sport_team	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
sporting_event	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7
ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/tickets/dms_sam...	csv	30 May 2019 9:37 AM UTC-7

i Data Validation Exercise

1. Within the Tables section of your ticketdata database, click the person table.

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Settings, ETL, Jobs, ML Transforms, Triggers, Dev endpoints, and Notebooks. The 'Tables' link is currently selected. The main area is titled 'Tables' and contains a table with the following columns: Name, Database, Location, Classification, and Last updated. There are ten rows in the table, each representing a different table in the 'ticketdata' database. The 'person' table is highlighted with a red dashed border around its row.

Name	Database	Location	Classification	Last updated
mlb_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
name_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
nfl_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
nfl_stadium_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
person	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
player	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
seat	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
seat_type	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
sport_division	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...
sport_league	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw...	csv	30 May 2019 9:15 AM ...

You may have noticed that some tables (such as person) have column headers such as **col0,col1,col2,col3**. In absence of headers or when the crawler cannot determine the header type, default column headers are specified.

This exercise uses the person table as an example of how to resolve this issue.

2. Click **Edit Schema** on the top right side.

The screenshot shows the 'Edit table' view for the 'person' table. At the top, there are buttons for 'Edit table' and 'Delete table'. To the right, there are buttons for 'View properties', 'Compare versions', and 'Edit schema', with 'Edit schema' also highlighted by a red dashed border. The main area displays the table's properties and schema. The properties section includes fields like Name (person), Description (ticketdata), Classification (csv), Location (s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sample/person/), Connection (No), Deprecated (No), Last updated (Thu May 30 09:15:02 GMT-700 2019), Input format (org.apache.hadoop.mapred.TextInputFormat), Output format (org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat), Serde serialization lib (org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe), and Serde parameters (field.delim ,). The 'Table properties' section shows various configuration settings. Below these, the 'Schema' section lists four columns: col0, col1, col2, and col3, each with a data type of string. A red dashed border highlights the 'Column name' column.

3. In the Edit Schema section, double-click **col0** (column name) to open edit mode. Type "id" as the column name.

4. Repeat the preceding step to change the remaining column names to match those shown in the following figure.

Column name	Data type	Key	Comment
1 id	string		x
2 full_name	string		x
3 last_name	string		x
4 first_name	string		x

5. Click **Save**.

Data ETL Exercise

1. In the left navigation pane, under **ETL**, click **Jobs**, and then click **Add job**.

Name	ETL language	Script location	Last modified	Job bookmark
You don't have any jobs defined yet.				
Add job				

2. On the Job properties page, make the following selections:
 - For **Name**, type **Glue-Lab-SportTeamParquet**.
 - For **IAM role**, choose existing **GlueLabRole**
 - For **Type**, Select **Spark**
 - Choose **Python 3** in Glue Version
 - For **This job runs**, select **A proposed script generated by AWS Glue**.
 - For **Script file name**, type **Glue-Lab-SportTeamParquet**.
 - Keep the rest settings as **default**.
3. Click **Next**.

Configure the job properties

Name	Glue-Lab-SportTeamParquet
IAM role ⓘ	module-3fccddd609114925bf8094186f40267-GlueLabRole-CTDC8071AG1E
Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job. Create IAM role .	
Type	Spark
Glue version	Spark 2.4, Python 3 (Glue version 1.0)
This job runs	<input checked="" type="radio"/> A proposed script generated by AWS Glue ⓘ <input type="radio"/> An existing script that you provide <input type="radio"/> A new script to be authored by you
Script file name	Glue-Lab-SportTeamParquet
S3 path where the script is stored	s3://aws-glue-scripts-773303173141-us-east-1/admin
Temporary directory ⓘ	s3://aws-glue-temporary-773303173141-us-east-1/admin

4. On the **Data source** page, select **sport_team** and Click **Next**.

Add job

Job properties
Glue-Lab-SportTeamParquet

Data source

Transform type

Data target

Schema

Choose a data source

Name	Database	Location	Classification
sport_team	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cv3/tick...	csv
sportstickets_dms_sample_sport_team	onprem-db	sportstickets.dms_sample.sport_team	postgresql

Showing: 1 - 2 < >

Back **Next**

5. On the **Choose a transformation type** page, select **change schema**

Add job

Job properties
Glue-Lab-SportTeamParquet

Data source
sport_team

Transform type
Change schema

Data target

Schema

Choose a transform type

Change schema
Change schema of your source data and create a new target dataset

Find matching records
Use machine learning to find matching records within your source data

Back **Next**

6. On the **Choose a data target** page, select **Create tables in your data target**.
7. For Data store, select **Amazon S3**.
8. For Format, select **Parquet**.
9. For Target path, create a *new folder dms_parquet* for the table **sport_team** at the end of the path, ie. **s3://<bucketname>/tickets/dms_parquet/sport_team** (an empty folder), to store the results produced by the ETL Job *Glue-Lab-SportTeamParquet*

10. Click **Next**.

Data source
sport_location

Transform type
Change schema

Data target

Schema

Create tables in your data target
 Use tables in the data catalog and update your data target

Data store
Amazon S3

Format
Parquet

Target path
s3bucket-1eegnc2tj056l/tickets/dms_parquet/sport_team

Back **Next**

11. Click the **target Data type** to edit the **id** schema mapping. In **String type** pop-up window Select **double** from **Column type** drop down and click **update**.

Add job

Job properties
Glue-Lab-SportTeamParquet

Data source
sport_team

Transform type
Change schema

Data target
s3://dmslabstudent...

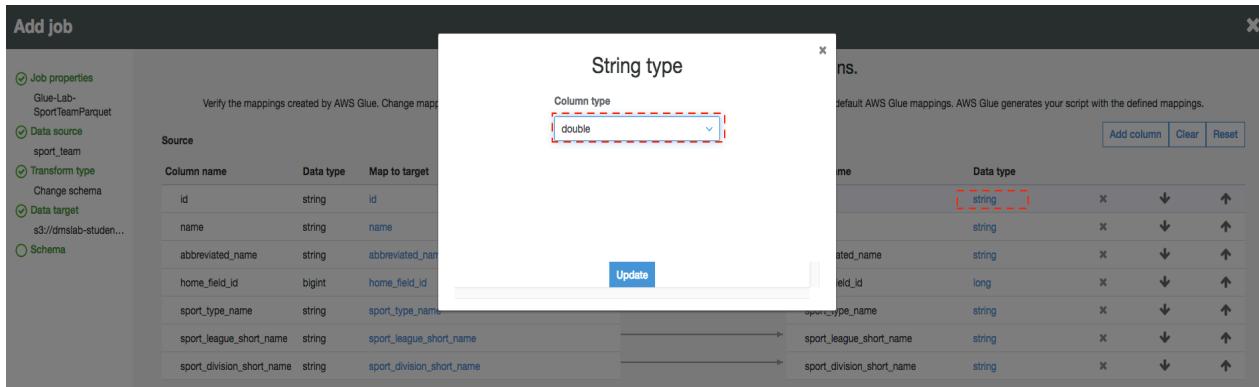
Schema

Map the source columns to target columns.

Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

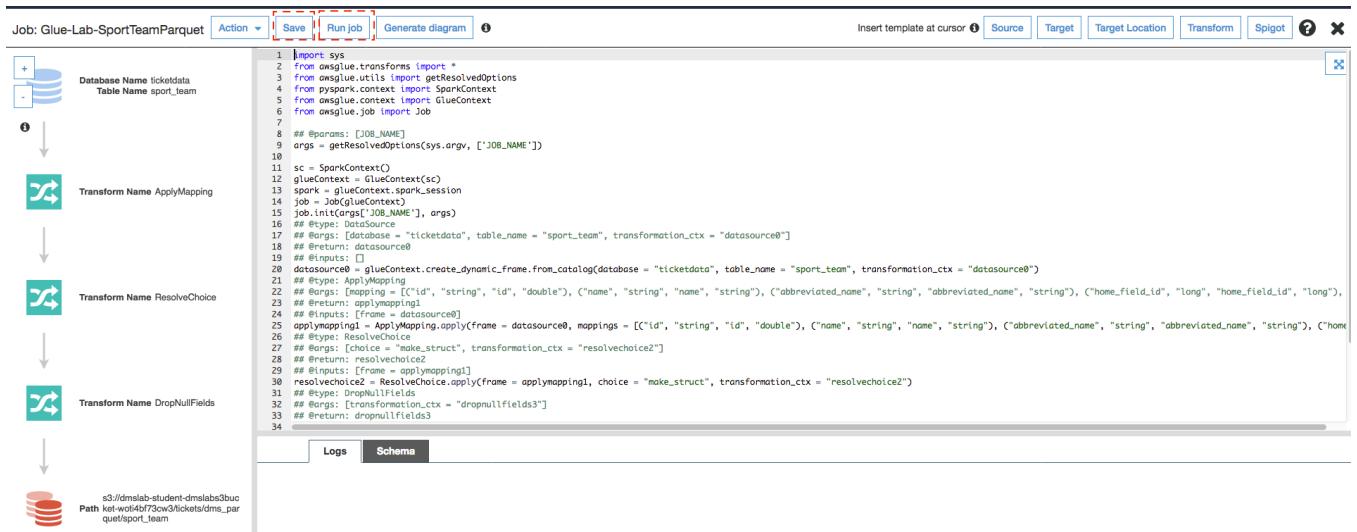
Source			Target		
Column name	Data type	Map to target	Column name	Data type	
id	string	id	id	string	x ↴ ↺
name	string	name	name	string	x ↴ ↺
abbreviated_name	string	abbreviated_name	abbreviated_name	string	x ↴ ↺
home field id	bigint	home field id	home field id	long	x ↴ ↺

Add column **Clear** **Reset**



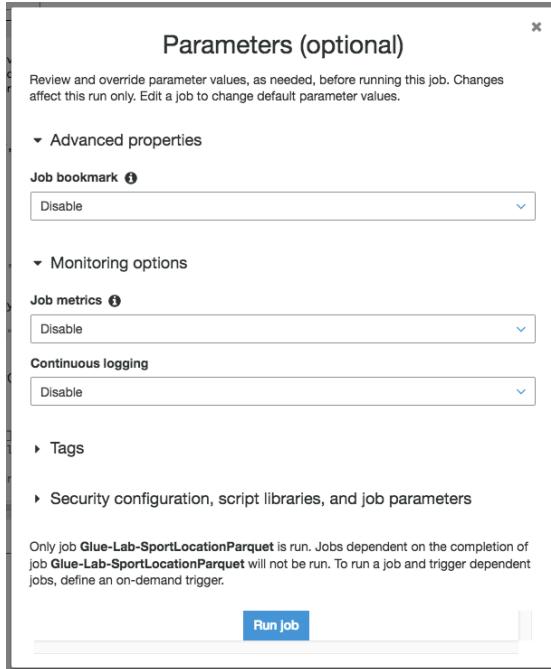
12. click **Save job and edit script**.

13. View the job. (This screen provides you with the ability to customize this script as required.) Click **Save** and then **Run Job**.



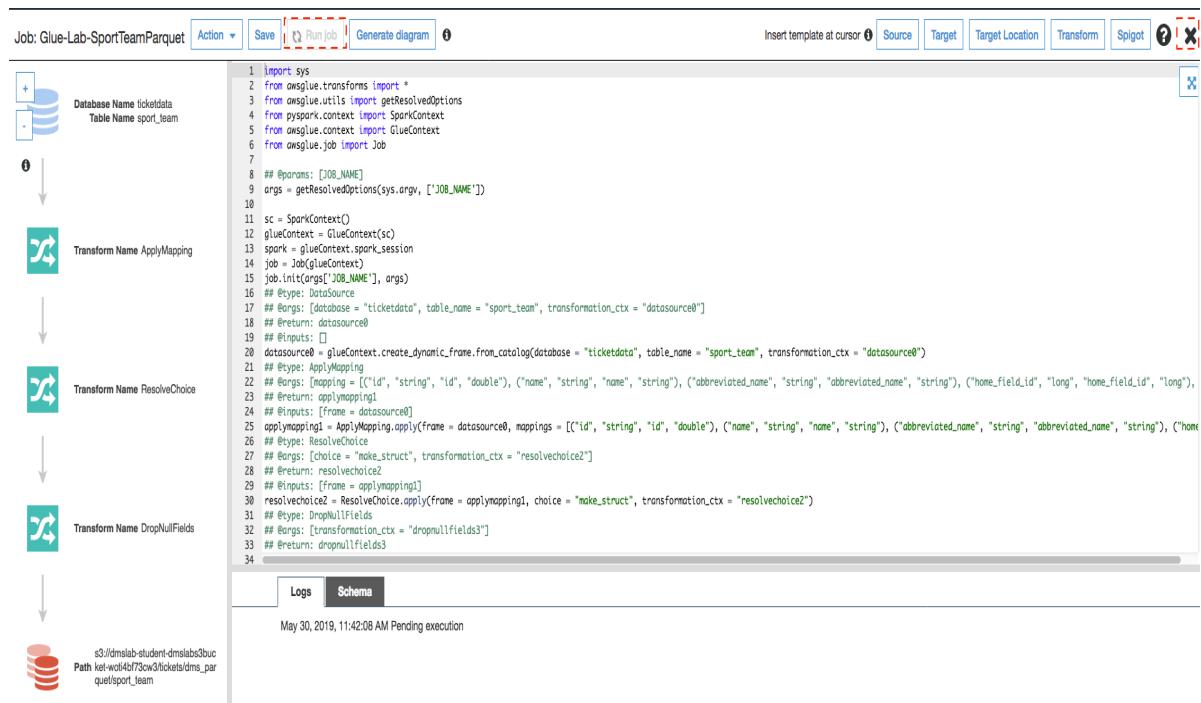
14. In Parameters option,

- a. you can leave **Job bookmark** as **Disable**. AWS Glue tracks data that has already been processed during a previous run of an ETL job by persisting state information from the job run.
 - b. You can leave the **Job metrics** option **Disable**. You can collect metrics about AWS Glue jobs and visualize them on the AWS Glue with job metrics.



15. Click Run Job

16. You will see job is now running as **Run job** button got disable. Click the cross button located on the top right corner to close the window to return to the ETL jobs.



17. Click your job to view history and verify that it ran successfully.

The screenshot shows the AWS Glue Jobs console. On the left, there's a navigation menu with options like AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Settings, ETL, Jobs, ML Transforms, and Triggers. The main area is titled 'Jobs' and contains a table with one row. The row details a job named 'Glue-Lab-SportTeamParquet' of type 'Spark' with 'Glue' as the catalog type and 'python' as the ETL language. The script location is 's3://aws-glue-scripts-341259728...', last modified on '30 May 2019 11:31 AM UTC-7', and it has a 'Disable' status. Below the table are tabs for History, Details, Script, and Metrics, with 'History' selected. A sub-table under 'History' shows a single run with ID 'jr_66bcc40bbaedc9de...', status 'Running', logs and error logs, maximum capacity of 10, execution time of 0 secs, timeout of 2880 mins, and triggered by '30 May 2019 ...'. At the bottom right, there are buttons for 'View run metrics' and 'Edit'.

IMPORTANT: We will **repeat** the preceding steps to create **4 more new ETL Jobs** to transform the additional tables from CSV to Parquet format.

#	Job Name & Script Filename	Source Table	S3 Target Path
1	Glue-Lab-SportLocationParquet	sport_location	dms_parquet/sport_location
2	Glue-Lab-SportingEventParquet	sporting_event	dms_parquet/sporting_event <i>(require 2 data type change, see below)</i>
3	Glue-Lab-SportingEventTicketParquet	sporting_event_ticket	dms_parquet/sporting_event_ticket <i>(require 3 data type change, see below)</i>
4	Glue-Lab-PersonParquet	person	dms_parquet/person <i>(require 1 data type change, see below)</i>

To enable us to join these tables, we will also update the **target data types** in the schema.

#	Table	Column	Source Data Type	Target Data Type
1	sporting_event	start_date_time	STRING	TIMESTAMP
2	sporting_event	start_date	STRING	DATE
3	sporting_event_ticket	id	STRING	DOUBLE
4	sporting_event_ticket	sporting_event_id	STRING	DOUBLE
5	sporting_event_ticket	tickerholder_id	STRING	DOUBLE
6	person	id	STRING	DOUBLE

Once these jobs have completed, we can create a crawler to detect metadata information for these parquet files.

Create Crawler for Parquet Files

1. In the AWS Glue navigation menu, click **Crawlers**, and then click **Add crawler**.

AWS Glue

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.									
User preferences									
Showing: 1 - 2 < > ⌂ ⌂									
Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added	
glue-lab-cdc-crawler		Glue	Ready	Logs	1 min	1 min	0	2	
glue-lab-crawler		Glue	Ready	Logs	1 min	1 min	0	15	

2. For Crawler name, type **glue-lab-parquet-crawler** and Click **Next**.

Add crawler

Add information about your crawler

Crawler info

Crawler source type

Data store

IAM Role

Schedule

Output

Review all steps

Crawler name
glue-lab-parquet-crawler

Tags, description, security configuration, and classifiers (optional)

Catalog options (optional)

Next

3. In next screen **crawler source type**, select **Data stores** and click **Next**.
4. Choose **S3** as data store type.
5. For Crawl data in, select **Specified path in my account**.
6. For Include path, specify the **S3 root path** that contains the all parquet files e.g., s3://<bucketname>/tickets/dms_parquet
7. Click **Next**.

Add a data store

Choose a data store
S3

Crawl data in

Specified path in my account

Specified path in another account

Include path
s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_parquet

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Exclude patterns (optional)

Back **Next**

8. For **Add another data store**, select **No** and Click **Next**
9. On the Choose an IAM role page, select **Choose an existing IAM role**.
10. For **IAM role**, select the role "xxxx-GlueLabRole-xxxx" from drop down and Click **Next**.
11. Select **Run On Demand** in Frequency and Click **Next**.
12. Choose the existing **ticktdata** database as the crawler's output database
13. In the **Prefix added to tables (optional)**, type **parquet_**

Add crawler

- Crawler info
glue-lab-parquet-crawler
- Crawler source type
Data stores
- Data store
S3: s3://dmslab-stu...
- IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
- Schedule
Run on demand
- Output
- Review all steps

Configure the crawler's output

Database ticketdata
 [Add database](#)

Prefix added to tables (optional) parquet_
 [Grouping behavior for S3 data \(optional\)](#)
[Configuration options \(optional\)](#)

[Back](#)
Next

14. Review the summary page and click **Finish**.

Crawler info

Name	glue-lab-parquet-crawler
Tags	-
Use Lake Formation Data Catalog	false

IAM role

IAM role	arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
----------	-------------------------------------------------------------------------------------

Schedule

Schedule	Run on demand
----------	---------------

Output

Database	ticketdata
Prefix added to tables (optional)	parquet_
Create a single schema for each S3 path	false
Configuration options	

[Back](#)
Finish

15. On the notification bar, click **Run it now**.

Once your crawler has finished running, you should report that 5 tables were added.

AWS Glue

- Data catalog
- Databases
- Tables
- Connections
- Crawlers**
- Classifiers
- Settings

ETL

- Jobs

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler "glue-lab-parquet-crawler" completed and made the following changes: 5 tables created, 0 tables updated. See the tables created in database ticketdata.

User preferences
Showing: 1 - 3 < > ⌂ ⓘ

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-cdc-cra...		Glue	Ready	Logs	1 min	1 min	0	2
glue-lab-crawler		Glue	Ready	Logs	1 min	1 min	0	15
glue-lab-parquet...		Glue	Ready	Logs	1 min	1 min	0	5

16

Confirm you can see the tables:

1. In the left navigation pane, click **Tables**.
2. Add the filter "parquet" to return the newly created tables.

Name	Database	Location	Classification	Last updated
mib_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
name_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
nfl_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
nfl_stadium_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
parquet_person	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	parquet	30 May 2019 3:33 PM ...
parquet_sport_location	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	parquet	30 May 2019 3:33 PM ...
parquet_sport_team	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	parquet	30 May 2019 3:33 PM ...
parquet_sporting_event	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	parquet	30 May 2019 3:33 PM ...
parquet_sporting_event...	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	parquet	30 May 2019 3:33 PM ...
person	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 2:21 PM ...
player	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
seat	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
seat_type	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
sport_division	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
sport_league	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...
sport_location	ticketdata	s3://dmslab-student-dmslabs3bucket-wot4bf73cw...	csv	30 May 2019 9:37 AM ...

Next Steps

In next lab, we will complete the following tasks:

- Query data and create a View with Athena
- Build a dashboard with QuickSight

Create Crawler for ongoing replication (optional)

Now, let's repeat this process to load the data from change data capture.

1. On the AWS Glue menu, select Crawlers.

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
You don't have any crawlers yet.							
Add crawler							

2. Click **Add crawler**.

3. Enter the crawler name for ongoing replication. This name should be descriptive and easily recognized (e.g., " glue-lab-cdc-crawler").
4. Optionally, enter the description. This should also be descriptive and easily recognized and Click **Next**.

Add crawler

Add information about your crawler	
<input checked="" type="radio"/> Crawler info <input type="radio"/> Crawler source type <input type="radio"/> Data store <input type="radio"/> IAM Role <input type="radio"/> Schedule <input type="radio"/> Output <input type="radio"/> Review all steps	Crawler name <input type="text" value="glue-lab-cdc-crawler"/> ▶ Tags, description, security configuration, and classifiers (optional) ▶ Catalog options (optional)
<input type="button" value="Next"/>	

5. Choose **Crawler Source Type** as **Data Source** and Click **Next**

Add crawler

Specify crawler source type	
<input checked="" type="radio"/> Crawler info glue-lab-crawler <input checked="" type="radio"/> Crawler source type <input type="radio"/> Data store <input type="radio"/> IAM Role <input type="radio"/> Schedule <input type="radio"/> Output <input type="radio"/> Review all steps	Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores. Crawler source type <input checked="" type="radio"/> Data stores <input type="radio"/> Existing catalog tables
<input type="button" value="Back"/> <input type="button" value="Next"/>	

6. On the Add a data store page, make the following selections:
 - a. For Choose a data store, click the drop-down box and select S3.
 - b. For Crawl data in, select Specified path in my account.
 - c. For Include path, enter the target folder for your DMS ongoing replication, e.g., "s3://dmslab-student-dmslabs3bucket-woti4bf73cw3/cdc/dms_sample"
7. Click **Next**.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store

IAM Role

Schedule

Output

Review all steps

Add a data store

Choose a data store
S3

Crawl data in
 Specified path in my account
 Specified path in another account

Include path
s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/cdc/dms_sample

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Exclude patterns (optional)

[Back](#) [Next](#)

8. On the Add another data store page, select No and Click Next.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role

Schedule

Output

Review all steps

Add another data store

Yes
 No

Chosen data stores
S3: s3://dmslab-stud...

[Back](#) [Next](#)

9. On the Choose an IAM role page, make the following selections:

- Select **Choose an existing IAM role**.
- For IAM role, select <stackname>-GlueLabRole-<RandomString>. E.g. "dmslab-student-GlueLabRole-ZOQDII7JTBUM"

10. Click Next.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)
dmslab-student-GlueLabRole-ZOQDII7JTBUM

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

- s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/cdc/dms_sample

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

11. On the Create a schedule for this crawler page, for Frequency, select **Run on demand** and Click Next.

Add crawler

- Crawler info
glue-lab-cdc-crawler
- Crawler source type
Data stores
- Data store
S3: s3://dmslab-stu...
- IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
- Schedule
Run on demand
- Output
- Review all steps

Create a schedule for this crawler

Frequency

[Back](#) [Next](#)

12. On the Configure the crawler's output page, select the existing Database for crawler output (e.g., "ticketdata").
13. For Prefix added to tables (optional), specify "cdc_"
14. For Configuration options (optional), keep the default selections and click Next.

Add crawler

- Crawler info
glue-lab-cdc-crawler
- Crawler source type
Data stores
- Data store
S3: s3://dmslab-stu...
- IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
- Schedule
Run on demand
- Output
- Review all steps

Configure the crawler's output

Database [?](#)

[Add database](#)

Prefix added to tables (optional) [?](#)

► Grouping behavior for S3 data (optional)

▼ Configuration options (optional)

During the crawler run, all schema changes are logged.
When the crawler detects schema changes in the data store, how should AWS Glue handle table updates in the data catalog?

Update the table definition in the data catalog.
 Add new columns only.
 Ignore the change and don't update the table in the data catalog. [?](#)

Update all new and existing partitions with metadata from the table. [?](#)

How should AWS Glue handle deleted objects in the data store?

Delete tables and partitions from the data catalog.
 Ignore the change and don't update the table in the data catalog.
 Mark the table as deprecated in the data catalog. [?](#)

[Back](#) [Next](#)

15. Review the summary page noting the Include path and Database target and Click **Finish**. The crawler is now ready to run.

Add crawler

- Crawler info
glue-lab-cdc-crawler
- Crawler source type
Data stores
- Data store
S3: s3://dmslab-stu...
- IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDIIJTBUM
- Schedule
Run on demand
- Output
ticketdata
- Review all steps

Crawler info

Name	glue-lab-cdc-crawler
Tags	-
Use Lake Formation Data Catalog	

IAM role

IAM role	arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDIIJTBUM
----------	------------------------------------------------------------------------------------

Schedule

Schedule	Run on demand
----------	---------------

Output

Database	ticketdata
Prefix added to tables (optional)	
Create a single schema for each S3 path	false
▼ Configuration options	
Schema updates in the data store	Update the table definition in the data catalog.
Object deletion in the data store	Mark the table as deprecated in the data catalog.

[Back](#) [Finish](#)

16. Click Run it now.

AWS Glue Data catalog Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler glue-lab-cdc-crawler was created to run on demand. [Run it now!](#)

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-cdc-cra...	Glue	Ready	0 secs	0 secs	0	0	0	0
glue-lab-crawler	Glue	Ready	Logs	1 min	1 min	0	0	15

17. When the crawler is completed, you can see it has "Status" as Ready, Crawler will change status from starting to stopping, wait until crawler comes back to ready state, you can see that it has created 2 tables.

AWS Glue Data catalog Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawler "glue-lab-cdc-crawler" completed and made the following changes: 2 tables created, 0 tables updated. See the tables created in database ticketdata.

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-cdc-cra...	Glue	Ready	Logs	1 min	1 min	0	0	2
glue-lab-crawler	Glue	Ready	Logs	1 min	1 min	0	0	15

18. Click the database name (e.g., "ticketdata") to browse the tables. Specify "cdc" as the filter to list only newly imported tables.

Name	Database	Location	Classification	Last updated
ticket_purchase_hist_95f83e3d8...	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/cdc/dms_samp...	csv	30 May 2019 10:38 AM UTC-7
sporting_event_ticket_1bb4a008...	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/cdc/dms_samp...	csv	30 May 2019 10:38 AM UTC-7
sport_team	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7
player	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7
seat_type	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7
mlb_data	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7
sport_location	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7
ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7
sport_league	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7
sporting_event	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_sa...	csv	30 May 2019 9:37 AM UTC-7

You can repeat same steps for CDC data as you preformed for initial full load data which include:

- Create folder structure in S3 bucket to store CDC parquet file.
- Create and Run ETL job to convert csv data into parquets format.
- Create and run another crawler to create data catalog for CDC parquet files.

When you are building an enterprise use cases, it's become important to automate entire pipeline and add notification. Please refer below blogs to try out end to end servlets datalike automation:

Build and automate a serverless data lake using an AWS Glue trigger for the Data Catalog and ETL jobs:

<https://aws.amazon.com/blogs/big-data/build-and-automate-a-serverless-data-lake-using-an-aws-glue-trigger-for-the-data-catalog-and-etl-jobs/>