



Amazon Web Services Data Engineering Immersion Day

Extract, Transform and Load Data Lake with Glue

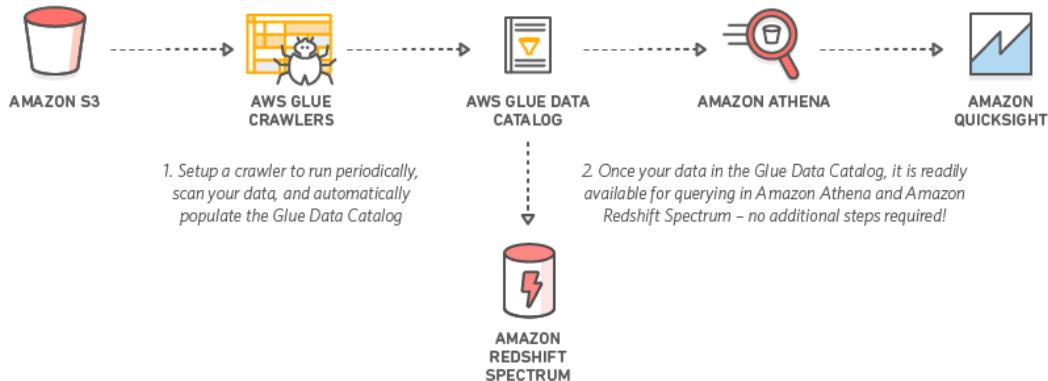
Jun 2019

Table of Contents

<i>Introduction</i>	2
Prerequisites:	2
Tasks Completed in this Lab:	2
Getting Started.....	2
<i>Create Glue Crawler for initial full load data</i>	3
<i>Data Validation Exercise</i>	8
<i>Data ETL Exercise</i>	9
<i>Create Crawler for Parquet Files</i>	14
<i>Next Steps</i>	18
<i>Create Crawler for ongoing replication (optional)</i>	18

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.

The screenshot shows the AWS Management Console search results for the term "glue".

The search bar at the top contains the text "glue".

The results list shows the "AWS Glue" service card, which is highlighted. The card describes AWS Glue as a fully managed ETL (extract, transform, and load) service.

Below the search bar, there are other service icons and names:

- EC2
- Database Migration Service
- S3
- CloudWatch
- AWS Glue

At the bottom left, there is a link to "All services".

Create Glue Crawler for initial full load data

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

The screenshot shows the AWS Glue interface with the 'Crawlers' section selected. The left sidebar includes options like Data catalog, Databases, Tables, Connections, Crawlers (which is highlighted), Classifiers, ETL, Jobs, Triggers, and Dev endpoints. The main area is titled 'Crawlers' with a sub-instruction: '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.' It features a search bar with 'Add crawler', 'Run crawler', 'Action', and a 'Filter by attributes' dropdown. A message says 'Showing: 0 - 0' and there's a 'Search' icon. Below is a table header with columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. A note says 'You don't have any crawlers yet.' with an 'Add crawler' button.

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 has 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 is a 'Crawler source type' section with 'Data stores' selected. At the bottom are 'Back' and 'Next' buttons.

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**.

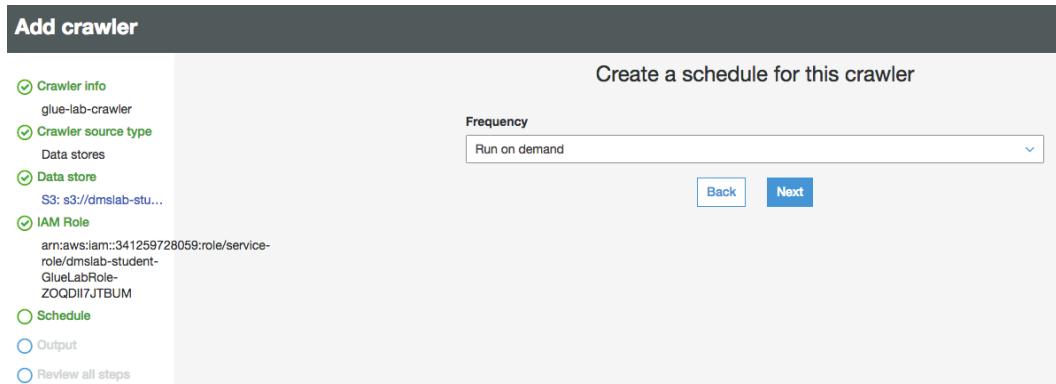
- b. For Crawl data in, select **Specified path in my account**.
 - c. For Include path, browse to the target folder for your DMS initial export, e.g., "s3://dmslab-student-dmslabs3bucket-woti4bf73cw3/tickets
7. Click **Next**.

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

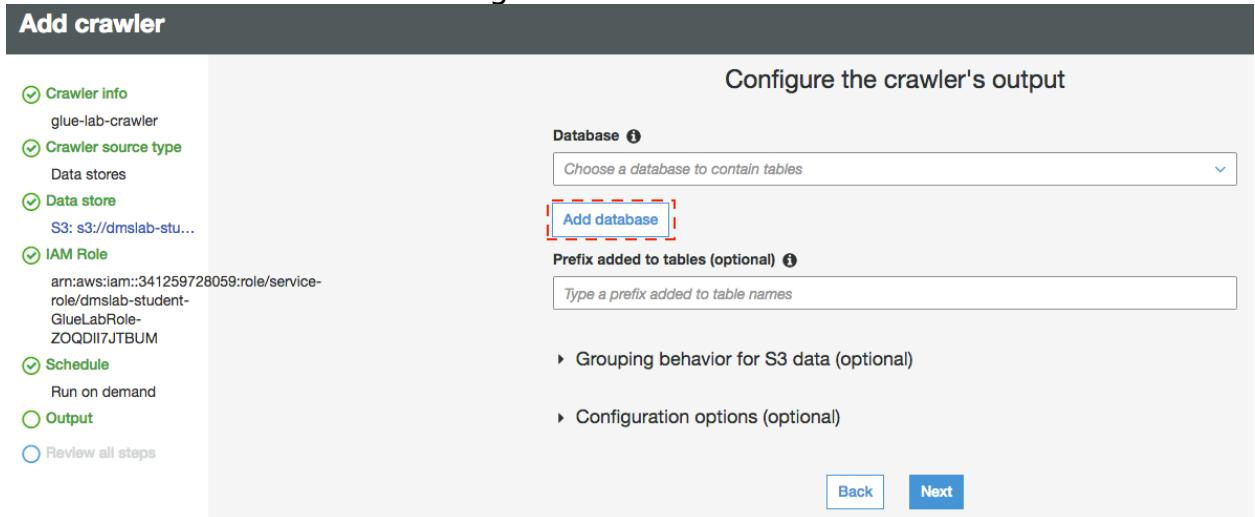
9. On the **Choose an IAM role** page, make the following selections:
- a. Select **Choose an existing IAM role**.
 - b. For **IAM role**, select **<stackname>-GlueLabRole-<RandomString>** created from the AWS CloudFormation template during the student lab.
For example "dmslab-student-GlueLabRole-ZOQDII7JTBUM"

10. Click **Next**.

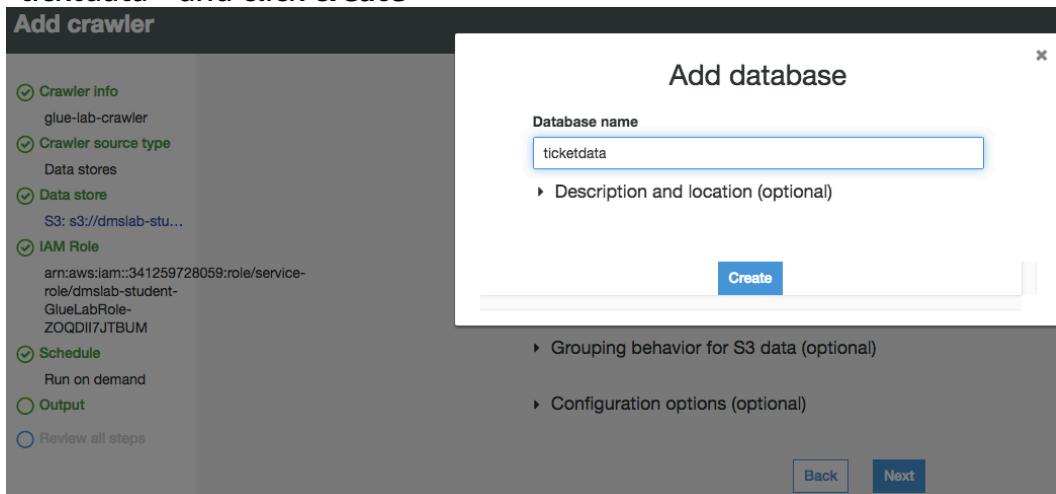
11. On the Create a schedule for this crawler page, for Frequency, select **Run on demand** and Click **Next**.



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-ZOQDI7JTBUM

Schedule

Run on demand

Output

Review all steps

Database

ticketdata

Add database

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-wotl4bf73cw3/tickets
Exclude patterns:

IAM role

IAM role: arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDI7JTBUM

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 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.

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 Name : glue-lab 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.

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 Name : glue-lab 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.).

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 Database : ticketdata 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
mbl_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 in 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. The left sidebar is identical to the previous screenshot. The main area shows the table details and schema. The 'Edit schema' button is highlighted with a red dashed border. The schema table has four columns: Column name, Data type, Partition key, and Comment. The first four rows are highlighted with a red dashed border.

Column name	Data type	Partition key	Comment
1	string		
2	string		
3	string		
4	string		

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.

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a navigation pane with options like Services, Resource Groups, AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Settings, ETL, and Jobs. The main area shows a table named 'person'. At the top right, it says 'Last updated 25 Sep 2018 Table Version (Current version)'. Below that is an 'Edit schema' button. A yellow circle highlights the 'Column name' column header. The table data is as follows:

	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**.

i Data ETL Exercise

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

The screenshot shows the AWS Glue ETL Jobs page. The left navigation pane includes 'AWS Glue', 'Data catalog', 'Databases', 'Tables', 'Connections', 'Crawlers', 'Classifiers', 'ETL', 'Jobs' (which is selected and highlighted in orange), 'Triggers', and 'Dev endpoints'. The main area is titled 'Jobs' with a sub-instruction: 'A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.' It features a search bar ('Filter by attributes') and a table with columns: Name, ETL language, Script location, Last modified, and Job bookmark. A message says 'You don't have any jobs defined yet.' with a blue 'Add job' button below it. At the top right, there are 'User preferences' and a 'Showing: 0 - 0' indicator.

2. On the Job properties page, make the following selections:
 - For **Name**, type **Glue-Lab-SportTeamParquet**.
 - For **IAM role**, choose existing role "dmslab-student-GlueLabRole-ZOQDII7JTBUM"
 - For **Type**, Select **Spark**
 - For **This job runs**, select **A proposed script generated by AWS Glue**.
 - For **ETL language**, select **Python**.
 - For **Script file name**, type **Glue-Lab-SportTeamParquet**.
 - For **S3 path where the script is stored**, provide a unique Amazon S3 path to store the scripts. (You can keep the default for this lab.)

- h. For **Temporary directory**, provide a unique Amazon S3 directory for a temporary directory. (You can keep the default for this lab.)

3. Click **Next**.

Configure the job properties

Name	Glue-Lab-SportTeamParquet
IAM role	dmslab-student-GlueLabRole-ZOQDII7JTBUM
Type	Spark
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
ETL language	<input checked="" type="radio"/> Python <input type="radio"/> Scala
Script file name	Glue-Lab-SportTeamParquet
S3 path where the script is stored	s3://aws-glue-scripts-341259728059-us-east-1/admin
Temporary directory	s3://aws-glue-temporary-341259728059-us-east-1/admin
▶ Advanced properties ▶ Monitoring options ▶ Tags (optional)	

4. On the Choose your data sources page, select **sport_team** and Click **Next**.

Add job

<input checked="" type="checkbox"/> Job properties Glue-Lab-SportTeamParquet	Choose a data source search : sport_team Filter or search for tables...												
<input type="checkbox"/> Data source <input type="checkbox"/> Transform type <input type="checkbox"/> Data target <input type="checkbox"/> Schema	Showing: 1 ~ 2 < > <table border="1"> <thead> <tr> <th>Name</th> <th>Database</th> <th>Location</th> <th>Classification</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="radio"/> sport_team</td> <td>ticketdata</td> <td>s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tick... csv</td> <td></td> </tr> <tr> <td><input type="radio"/> sportstickets_dms_sample_sport_team</td> <td>onprem-db</td> <td>sportstickets.dms_sample.sport_team</td> <td>postgresql</td> </tr> </tbody> </table>	Name	Database	Location	Classification	<input checked="" type="radio"/> sport_team	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tick... csv		<input type="radio"/> sportstickets_dms_sample_sport_team	onprem-db	sportstickets.dms_sample.sport_team	postgresql
Name	Database	Location	Classification										
<input checked="" type="radio"/> sport_team	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tick... csv											
<input type="radio"/> sportstickets_dms_sample_sport_team	onprem-db	sportstickets.dms_sample.sport_team	postgresql										

[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 your data targets page, select **Create tables in your data target**.
7. For Data store, select **Amazon S3**.
8. For Format, select **Parquet**.
9. For Target path, choose a new location adjacent to your CSV files (without any existing objects) to store the results e.g., "s3://dmslab-student-dmslabs3bucket-woti4bf73cw3/tickets/dms_parquet/sport_team"

Follow this user guide to create folder structure in your S3 bucket to store parquet file - <https://docs.aws.amazon.com/AmazonS3/latest/user-guide/create-folder.html>

10. Click **Next**.

Add job

Job properties
Glue-Lab-SportTeamParquet

Data source
sport_team

Transform type
Change schema

Data target

Schema

Choose a data target

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-woti4bf73cw3/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.

12.

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with a tree view of the job properties, including 'Job properties', 'Data source', 'Transform type', 'Data target', and 'Schema'. The 'Schema' section is expanded, showing a table with columns: 'Column name', 'Data type', and 'Map to target'. The table rows correspond to the columns in the main configuration area. In the center, a modal window titled 'String type' is open, showing a dropdown menu for 'Column type' with 'double' selected. Below the dropdown is a blue 'Update' button. To the right of the modal, the main configuration area shows a table with columns: 'Name' and 'Data type'. The 'Data type' column contains several 'string' entries, each with a red dashed box around it. There are also 'long' and 'double' entries. On the far right of the table are four icons: a downward arrow, an upward arrow, a delete icon, and a refresh icon. At the top right of the modal, there are three buttons: 'Add column', 'Clear', and 'Reset'.

Map the source columns to target columns.

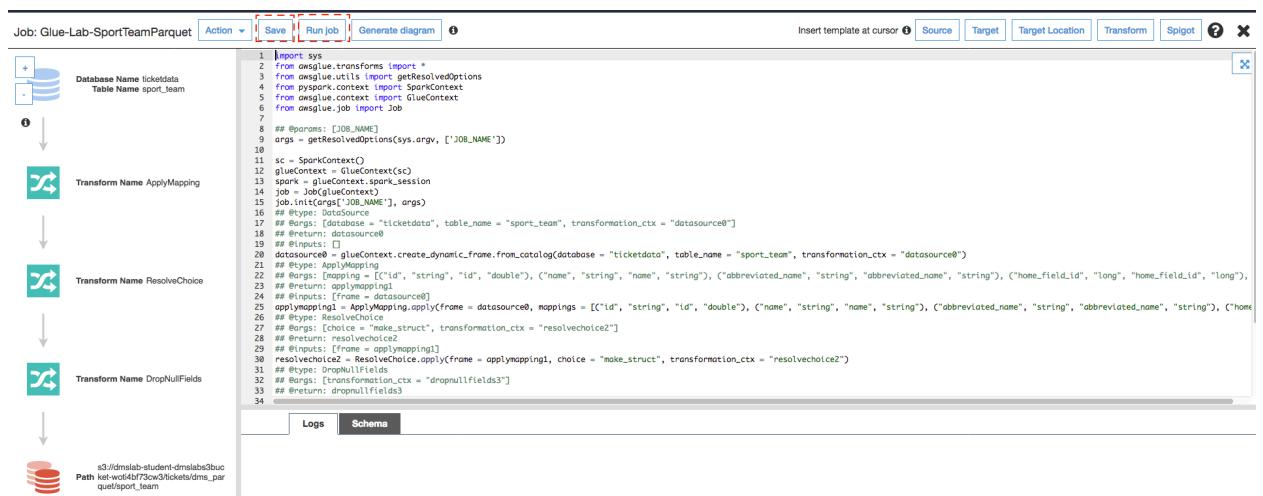
Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source			Target				
Column name	Data type	Map to target	Column name	Data type			
id	string	id	id	double	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	↓	↑
sport_type_name	string	sport_type_name	sport_type_name	string	x	↓	↑
sport_league_short_name	string	sport_league_short_name	sport_league_short_name	string	x	↓	↑
sport_division_short_name	string	sport_division_short_name	sport_division_short_name	string	x	↓	↑

[Back](#) [Save job and edit script](#)

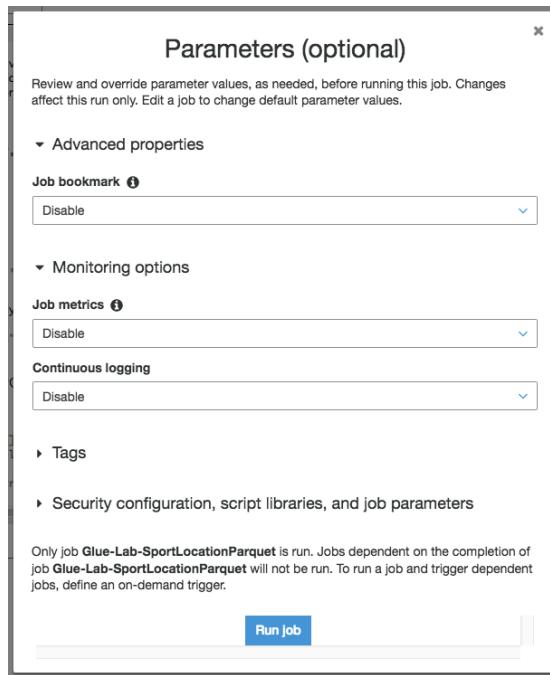
13. click Save job and edit script.

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



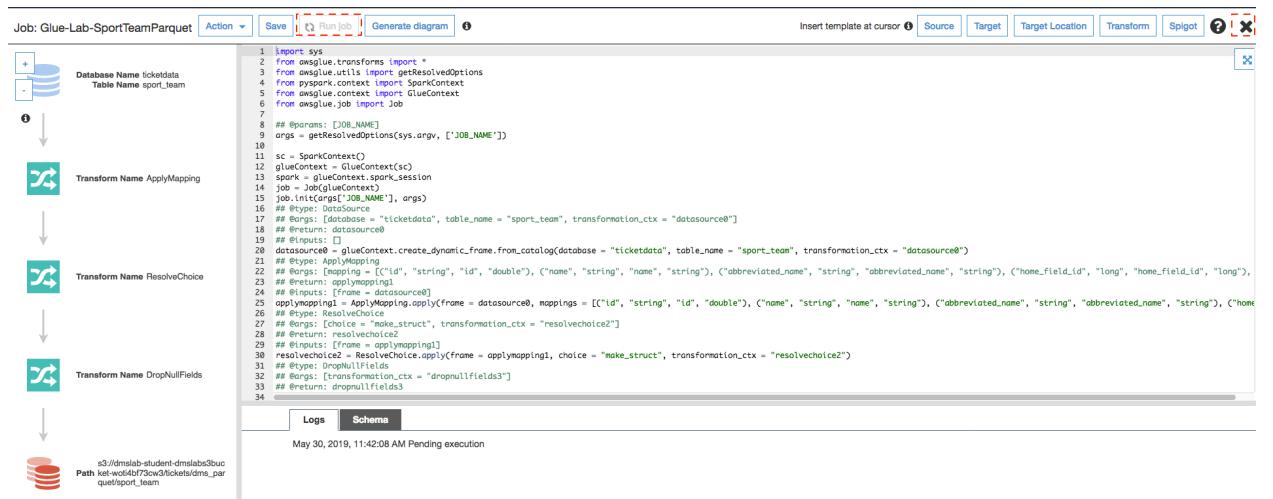
15. In Parameters option,

- 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.
- 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.



16. Click Run Job

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



18. 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 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' which is a Spark job of type 'Glue' written in 'python'. It was last modified on '30 May 2019 11:31 AM UTC-7' and has a status of 'Disable'. Below the table, there are tabs for 'History', 'Details', 'Script', and 'Metrics'. Under the 'History' tab, there's another table showing a single run ID. This run is listed as 'Running' with a timestamp of '30 May 2019 ...'. The 'Logs' and 'Error logs' columns show 'Logs' and 'Error logs' respectively. Other columns include 'Retry attempt', 'Run status', 'Error', 'Logs', 'Error logs', 'Maximum capacity', 'Execution time', 'Timeout', 'Delay', 'Triggered by', 'Start time', and 'End time'. At the top right of the main table, there are filters and pagination controls.

We will repeat the preceding steps to create new ETL Jobs to transform the additional tables.

Job Name & Script Filename	Source Table	S3 Target Path
Glue-Lab-SportLocationParquet	sport_location	dms_parquet/sport_location
Glue-Lab-SportingEventParquet	sporting_event	dms_parquet/sporting_event
Glue-Lab-SportingEventTicketParquet	sporting_event_ticket	dms_parquet/sporting_event_ticket
Glue-Lab-PersonParquet	person	dms_parquet/person

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
sporting_event_ticket	id	STRING	DOUBLE
sporting_event_ticket	sporting_event_id	STRING	DOUBLE
sporting_event_ticket	tickerholder_id	STRING	DOUBLE
sporting_event	start_date_time	STRING	TIMESTAMP
sporting_event	start_date	STRING	DATE
person	id	STRING	DOUBLE

Once these jobs have completed, we can create a crawler to index these parquet files.

Create Crawler for Parquet Files

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

The screenshot shows the AWS Glue Crawlers console. On the left, there's a navigation menu with options like Data catalog, Databases, Tables, Connections, Crawlers (which is selected and highlighted in orange), and Classifiers. The main area is titled 'Crawlers' and contains a table with two rows. The rows detail crawlers named 'glue-lab-cdc-crawler' and 'glue-lab-crawler', both of which are 'Ready' and using the 'Glue' catalog type. They have a median runtime of 1 min, 0 tables updated, and 2 tables added. There are tabs for 'Add crawler', 'Run crawler', 'Action', and a search bar. At the top right, there are filters and pagination controls.

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

Add crawler

Add information about your crawler

Crawler name: glue-lab-parquet-crawler

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

Catalog options (optional)

Next

Step navigation: Crawler info (selected), Crawler source type, Data store, IAM Role, Schedule, Output, Review all steps.

- In next screen **Specify crawler source type**, select **Data Source** as choice for **Crawler resource type** and click **Next**.
- For Choose a data store, select **S3**.
- For Crawl data in, select Specified path in account.
- For Include path, specify the S3 Path that contains the nested parquet files e.g., s3://dmslab-student-dmsslabs3bucket-wotl4bf73cw3/tickets/dms_parquet
- 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-dmsslabs3bucket-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

- For Add another data store, select **No** and Click **Next**.

Add crawler

Crawler info: glue-lab-parquet-crawler

Crawler source type: Data stores

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

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

Add another data store:

- Yes
- No

Back Next

- On the Choose an IAM role page, select **Choose an existing IAM role**.

10. For IAM role, select the existing role "dmslab-student-GlueLabRole-ZOQDII7JTBUM" and Click **Next**.

Add crawler

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 [?](#)

This role must provide permissions similar to the AWS managed policy, [AWSGlueServiceRole](#), plus access to your data stores.

- s3://dms-lab-george/tickets/dms_sample_parquet

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

Back **Next**

11. For Frequency, select Run On Demand and Click **Next**.

Add crawler

Create a schedule for this crawler

Frequency

Back **Next**

12. For the crawler's output database, choose your existing database.

13. For the Prefix added to tables (optional), type "**parquet_**"

Add crawler

Configure the crawler's output

Database [?](#)

Add database

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

► 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.

The screenshot shows the AWS Glue Data Catalog interface. In the left navigation pane, 'Tables' is selected under 'Data catalog'. A notification bar at the top right indicates: 'Crawlers Crawler "glue-lab-parquet-crawler" completed and made the following changes: 5 tables created, 0 tables updated. See the tables created in database ticketdata.' Below this, the crawler list table shows three crawlers: 'glue-lab-cdc-cra...', 'glue-lab-crawler', and 'glue-lab-parquet...'. The 'glue-lab-parquet...' row is highlighted with a blue background, indicating it is the active crawler.

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.

The screenshot shows the AWS Glue Data Catalog 'Tables' page. The left navigation pane shows 'Tables' selected under 'Data catalog'. The main table lists various tables in the 'ticketdata' database. A red dashed box highlights five specific tables: 'parquet_person', 'parquet_sport_location', 'parquet_sport_team', 'parquet_sporting_event', and 'parquet_sporting_event.'. These tables were created by the 'glue-lab-parquet...' crawler and are listed in the 'ticketdata' database.

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.

The screenshot shows the AWS Glue interface with the 'Crawlers' section selected. On the left, there is a sidebar with navigation links: Data catalog, Databases, Tables, Connections, Crawlers (which is highlighted), Classifiers, ETL, Jobs, Triggers, and Dev endpoints. The main area has a title 'Crawlers' with a sub-instruction: '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 are buttons for 'Add crawler', 'Run crawler', 'Action', and a search bar. A message says 'Showing: 0 - 0 < > ⓘ ⓘ'. A table header row includes columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. A message below the table states 'You don't have any crawlers yet.' with a 'Add crawler' button.

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**.

The screenshot shows the 'Add crawler' wizard. The top bar is dark grey with the title 'Add crawler'. The left sidebar has radio buttons for 'Crawler info' (selected), 'Crawler source type', 'Data store', 'IAM Role', 'Schedule', 'Output', and 'Review all steps'. The main right panel has a title 'Add information about your crawler'. It contains a 'Crawler name' field with the value 'glue-lab-cdc-crawler'. Below it are two expandable sections: 'Tags, description, security configuration, and classifiers (optional)' and 'Catalog options (optional)'. At the bottom right is a blue 'Next' button.

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

Add crawler

Crawler info
glue-lab-crawler

Crawler source type

Data store

IAM Role

Schedule

Output

Review all steps

Specify crawler source type

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

Data stores

Existing catalog tables

Back **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-woti4bf73cw3/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

Back **Next**

Chosen data stores

S3: s3://dmslab-stud...

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

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 [?](#)
 [Change](#)

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

- s3://dmslab-student-dmslabs3bucket-wot4bf73cw3/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

Create a schedule for this crawler

Frequency
 [Change](#)

[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

Configure the crawler's output

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
ticketdata

Review all steps

Database
ticketdata

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. ⓘ

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-ZOQDII7JTBUM

Schedule
Run on demand

Output
ticketdata

Review all steps

Crawler info

Name	glue-lab-cdc-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)	
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.

16. Click **Run it now**.

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.

Crawler glue-lab-cdc-crawler was created to run on demand.

Showing: 1 - 2 < >

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		
glue-lab-crawler	Glue	Ready	Logs	1 min	1 min	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.

The screenshot shows the AWS Glue Crawler list page. On the left, there's a sidebar with navigation links: AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers (which is selected), Classifiers, Settings, and ETL. The main content area has a heading 'Crawlers' with a sub-instruction: '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, a message box states: 'Crawler "glue-lab-cdc-crawler" completed and made the following changes: 2 tables created, 0 tables updated. See the tables created in database ticketdata.' There are buttons for 'Add crawler', 'Run crawler', and 'Action'. A search bar filters results by 'Name : glue-lab'. The table lists two crawlers:

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

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

The screenshot shows the AWS Glue Tables list page. The sidebar includes links for AWS Glue, Data catalog, Databases, Tables (selected), Connections, Crawlers, Classifiers, Settings, ETL, Jobs, ML Transforms, Triggers, Dev endpoints, and Notebooks. The main area has a heading 'Tables' with a sub-instruction: '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.' A message box says: 'Tables in database ticketdata' and 'Filter or search for tables...'. The table lists tables in the 'ticketdata' database, with three specific ones highlighted with red boxes:

Name	Database	Location	Classification	Last updated
ticket_purchase_hist_95fb3e3dB...	ticketdata	s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/cdc/dms_samp...	csv	30 May 2019 10:38 AM UTC-7
sporting_event_ticket_1bb4a00B...	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
mib_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/>