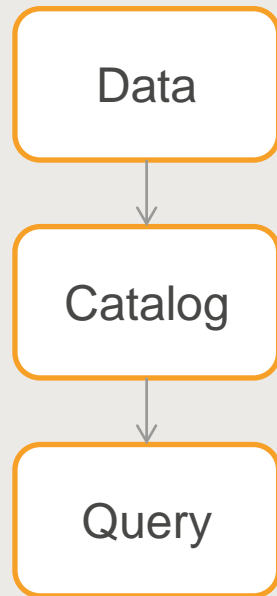


Glue Catalog Management

Schema Changes, Formats, Partitioning

Handling Changes to Data

- Rename columns
- Add columns
- Remove columns
- Reorder columns
- Change data type



Dependencies

- Keep the catalog in Glue up-to-date
 - Schedule Crawler to run periodically for automated update
 - Manual schema changes
- Querying systems compatibility
 - Athena and other systems may not work properly when schema or data changes
 - Data Format plays an important role

References: <https://docs.aws.amazon.com/athena/latest/ug/handling-schema-updates-chapter.html>

How Athena Queries

Access by column index - CSV, TSV, ORC (default), Parquet (optional)

Access by column name – Parquet (default), ORC (optional)

Reference:

Athena - format comparison, handling schema updates

<https://docs.aws.amazon.com/athena/latest/ug/handling-schema-updates-chapter.html>

Athena uses column index for CSV, TSV

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Shuffled – Index maps to wrong columns

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4
petal_width	sepal_width	petal_length	sepal_length	class
0.2	3.5	1.4	5.1	Iris-setosa
0.2	3	1.4	4.9	Iris-setosa
0.2	3.2	1.3	4.7	Iris-setosa
0.2	3.1	1.5	4.6	Iris-setosa
0.2	3.6	1.4	5	Iris-setosa

New data
Columns
Shuffled

Missing Data – Index maps to wrong columns

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4
sepal_width	petal_length	petal_width	class	
3.5	1.4	0.2	Iris-setosa	
3	1.4	0.2	Iris-setosa	
3.2	1.3	0.2	Iris-setosa	
3.1	1.5	0.2	Iris-setosa	
3.6	1.4	0.2	Iris-setosa	

*New data
has missing
columns*

New Columns - Index maps to wrong columns

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4	5
sepal_length	sepal_width	petal_length	petal_width	sepal_area	class
5.1	3.5	1.4	0.2	17.9	Iris-setosa
4.9	3	1.4	0.2	14.7	Iris-setosa
4.7	3.2	1.3	0.2	15.0	Iris-setosa
4.6	3.1	1.5	0.2	14.3	Iris-setosa
5	3.6	1.4	0.2	18.0	Iris-setosa

*New data
in the middle*

CSV works well if you preserve order of existing columns – add new columns at the end

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4	5
sepal_length	sepal_width	petal_length	petal_width	class	sepal_area
5.1	3.5	1.4	0.2	Iris-setosa	17.9
4.9	3	1.4	0.2	Iris-setosa	14.7
4.7	3.2	1.3	0.2	Iris-setosa	15.0
4.6	3.1	1.5	0.2	Iris-setosa	14.3
5	3.6	1.4	0.2	Iris-setosa	18.0

*New data
at the end*

CSV supports renaming columns in schema

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4
sepal_len_cm	sepal_wid_cm	petal_len_cm	petal_wid_cm	plant_type
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

*Rename
columns in
the Glue
Catalog*

Summary: Athena and CSV/TSV Format

- CSV/TSV columns are accessed by index
- When you have a mix of old data and new data
 - Ordering of columns need to be preserved
 - Add columns at the end
 - Removing columns not supported - i.e. new data with missing columns
 - Data type change would require testing
 - Columns can be renamed
- CSV/TSV is suitable if you can add columns at the end and not remove any existing columns

Athena uses column name for Parquet

Shuffled – Access by name works!

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4
petal_width	sepal_width	petal_length	sepal_length	class
0.2	3.5	1.4	5.1	Iris-setosa
0.2	3	1.4	4.9	Iris-setosa
0.2	3.2	1.3	4.7	Iris-setosa
0.2	3.1	1.5	4.6	Iris-setosa
0.2	3.6	1.4	5	Iris-setosa

*New data
Columns
Shuffled*

Missing Data – Access by name works!

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4
sepal_width	petal_length	petal_width	class	
3.5	1.4	0.2	Iris-setosa	
3	1.4	0.2	Iris-setosa	
3.2	1.3	0.2	Iris-setosa	
3.1	1.5	0.2	Iris-setosa	
3.6	1.4	0.2	Iris-setosa	

*New data
has missing
columns*

New Columns – Access by name works!

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4	5
sepal_length	sepal_width	petal_length	petal_width	sepal_area	class
5.1	3.5	1.4	0.2	17.9	Iris-setosa
4.9	3	1.4	0.2	14.7	Iris-setosa
4.7	3.2	1.3	0.2	15.0	Iris-setosa
4.6	3.1	1.5	0.2	14.3	Iris-setosa
5	3.6	1.4	0.2	18.0	Iris-setosa

*New data
in the middle*

Parquet handles lot of common schema updates

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4	5
sepal_length	sepal_width	petal_length	petal_width	class	sepal_area
5.1	3.5	1.4	0.2	Iris-setosa	17.9
4.9	3	1.4	0.2	Iris-setosa	14.7
4.7	3.2	1.3	0.2	Iris-setosa	15.0
4.6	3.1	1.5	0.2	Iris-setosa	14.3
5	3.6	1.4	0.2	Iris-setosa	18.0

*New data
at the end*

Parquet - schema column rename not supported.

0	1	2	3	4
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

Old data

0	1	2	3	4
sepal_len_cm	sepal_wid_cm	petal_len_cm	petal_wid_cm	plant_type

*Renamed
columns in the
Glue Catalog -
won't retrieve
any data*

Summary: Athena and Parquet Format

- Columns are accessed by column name
- When you have a mix of new and old data:
 - Columns can be removed – i.e. new data with missing columns
 - Columns can be reordered
 - Column rename not supported (it will pull new data. old column not retrieved)
- Data type change would require testing
- Efficient format for high performance and cost-effective access

Lab – Athena and CSV

- New data that matches existing schema
- Changes to columns:
 - Reordering
 - Adding
 - Removing
 - Renaming
- Iris dataset

Lab – Athena and PARQUET

- New data that matches existing schema
- Changes to columns:
 - Reordering
 - Adding
 - Removing
 - Renaming
- Iris dataset

Organization of data in S3 Data Lake

Glue and Athena

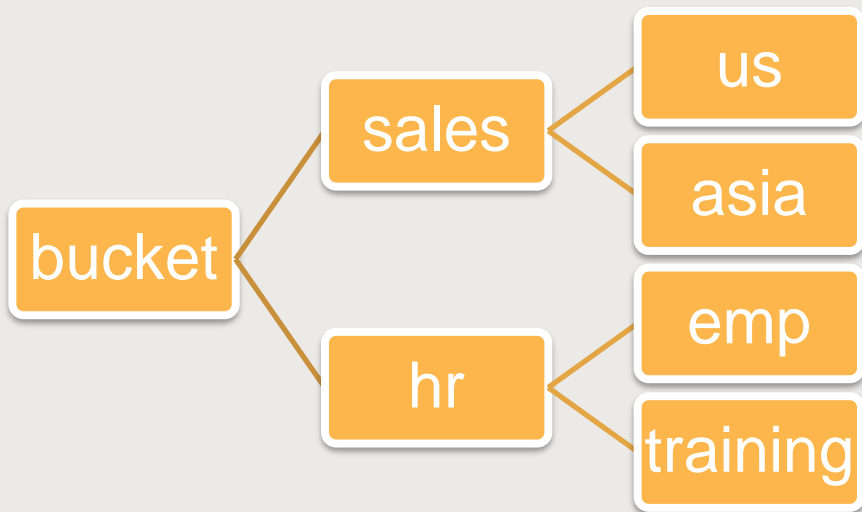
Glue Crawler can discover metadata from complex folder structure and create tables

However, Athena and other applications expect data to be similar in a particular S3 path

Don't mix and match files that are structurally different in the same folder – keep files that have similar schema

Glue Crawler

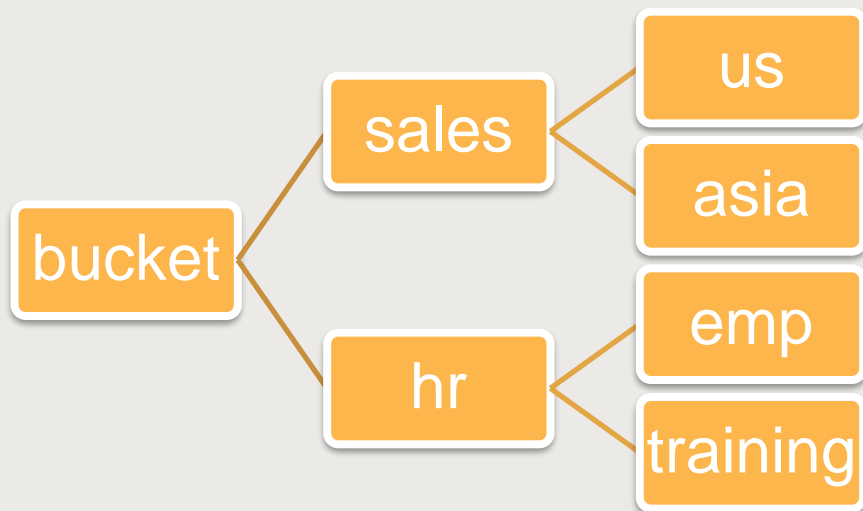
S3 Folder Structure



- Classifies data based on format, schema
- Groups data into tables, partitions
- Crawler can create multiple tables from the same S3 prefix

Glue Crawler – Athena Best Practices

S3 Folder Structure



- Create crawlers with more specific S3 path

Sales: S3://bucket/sales/

HR-Emp: s3://bucket/hr/emp/

HR-Training: s3://bucket/hr/training/

Organize folders by Data Classification

Tier 1 – Protected Data

- Information for internal use
- Vendor bank account information
- Information for internal use only

Tier 2 – Restricted Data

- Sales and marketing data, executed contracts, receipts
- Employee HR records

Tier 3 – Highly Strategic

- Trade secret
- Pricing information
- Merger/acquisition information
- Proprietary Process
- Inventions prior to patent
- Public disclosure could cause severe or catastrophic legal, financial, or reputational damage

Organize by data classification

- Enforce principles of least privileged access
- Separate buckets by class
- Separate top-level folders by class
- Classification of a particular data can change in the future

Partitioning – big data systems

Improve performance

Reduce querying costs

Prevent GET [request rate-limit errors](#) in S3

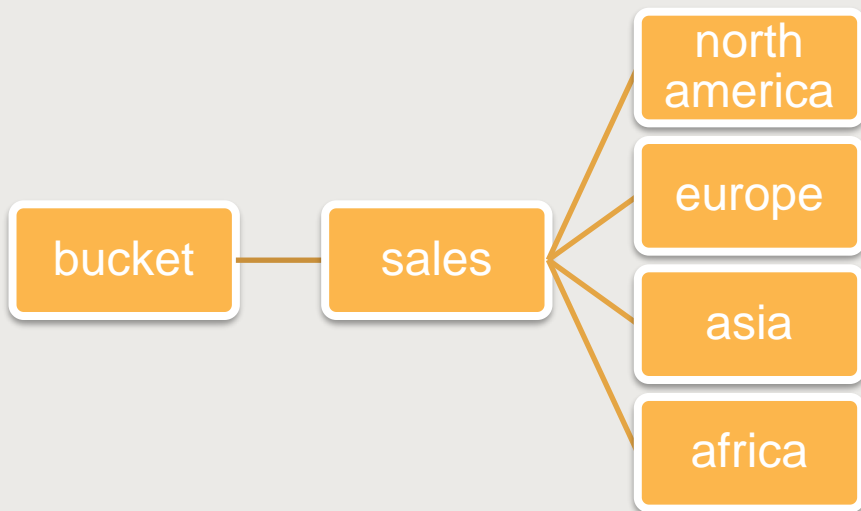
References:

<https://aws.amazon.com/blogs/big-data/work-with-partitioned-data-in-aws-glue/>

<https://docs.aws.amazon.com/athena/latest/ug/partitions.html>

Glue Crawler – Partitions

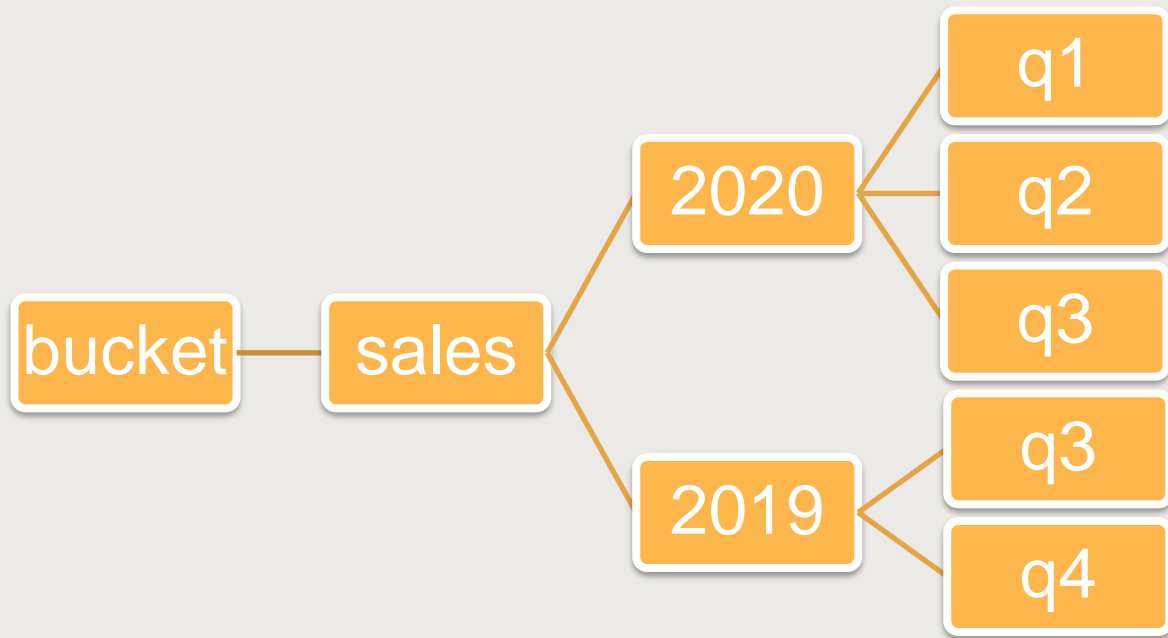
S3 Folder Structure – Partition By Region



- Crawler automatically partitions table based on folder structure
- When region is specified in where clause, Athena will scan only files in that region

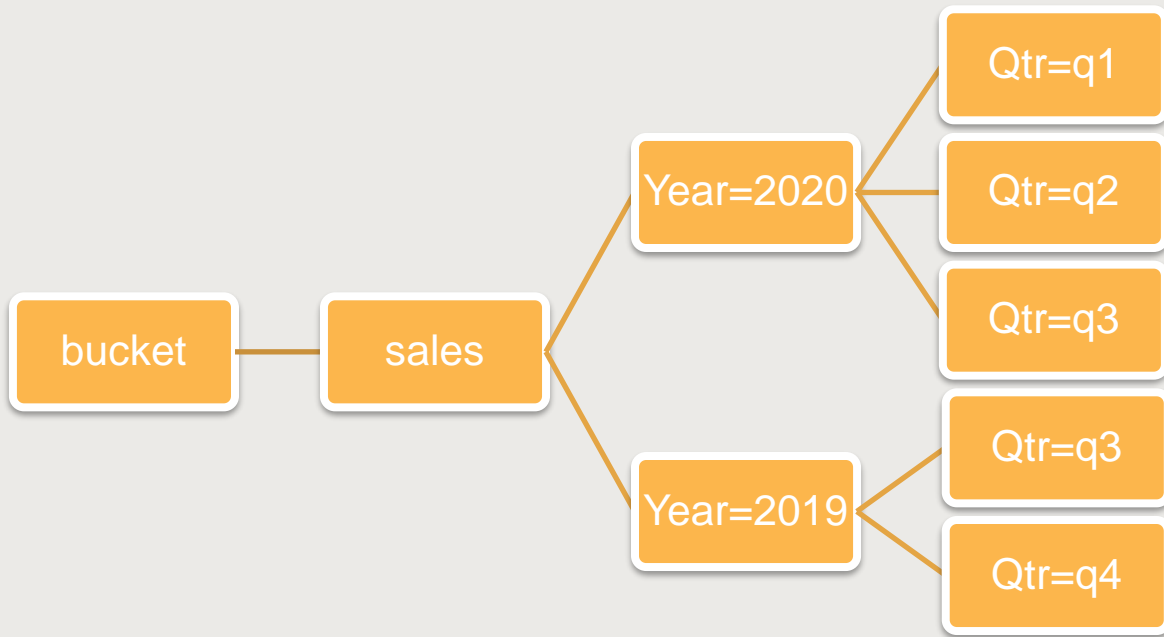
Glue Crawler – Partition based on Time

Partition By Year and Quarter



Glue Crawler – Hive Compatible Partitions

Key-value name for folders



Add, Remove Partitions

Glue Catalog needs to be updated when you add or remove partitions!

Type	Behavior
Crawler	Run crawler again to update catalog (slowest) Scheduled (hourly, daily, monthly, or custom) Hive-compatible and regular partitions Updates catalog to add and remove partitions
MSCK Command	Add or remove Hive compatible partitions (faster) Updates catalog to add and remove partitions On-demand, event-based
ADD, DROP PARTITION	Low-level Table command (fastest) You need to specify which partition to add and remove On-demand, event-based

Table and Partition Schema

Glue maintains separate schema for table and partitions

These two schemas can start drifting

You can ask Glue Crawler to maintain a single schema at table level and propagate to all partitions.

Summary

Organize folders such that lowest level folders contain files of similar schema

Crawlers – do not specify a top-level path. Specify more precise path

Ensure each S3 path maps to exactly a single table in the catalog (for Athena, Redshift Spectrum, EMR compatibility)

Organize bucket or top-level folder by data classification

Use partitioning to optimize for performance and cost

Partitioned Table – handling new partitions

- Use crawler to detect and add (easiest)
- Run `MSCK REPAIR TABLE orders` command
 - Works for hive compatible partitions
 - Scans source for partition changes and updates the table definition
- Run Add Partition command
 - Works for any folder structure
 - You need to build logic to add-remove partitions

```
ALTER TABLE orders ADD PARTITION (dt='2015-01-01')  
LOCATION 's3://athena-examples-us-west/elb/plaintext/2015/01/01/'
```

Lab – Partitioned Table

- Organize S3 data by plant type
- Configure crawler for the partitioned folder
- Manage partitions (using Crawler, ADD PARTITION)

```
ALTER TABLE iris_partitioned
```

```
ADD PARTITION (partition_0='versicolor') location  
's3://bucket/iris/partitioned/versicolor/';
```

- Query partitioned table using Athena

Lab – Manual Table Creation and Maintenance

Create a table

Configure Glue Crawler to maintain manually created schema

[see next page for create table statement]

Create Table

```
CREATE EXTERNAL TABLE `iris_manual`(  
  `sepal_length` double,  
  `sepal_width` double,  
  `petal_length` double,  
  `petal_width` double,  
  `class` string)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
LOCATION 's3://bucket/iris/partitioned/'  
TBLPROPERTIES ('skip.header.line.count'='1', 'typeOfData'='file','classification'='csv')
```

Lab – Hive Compatible Folders

Create a partitioned table

Use **MSCK REPAIR** TABLE Command

Configure Glue Crawler to maintain schema

ADD PARTITION Command



Chandra Lingam

57,000+ Students



For AWS self-paced video courses, visit:

<https://www.cloudwavetraining.com/>

