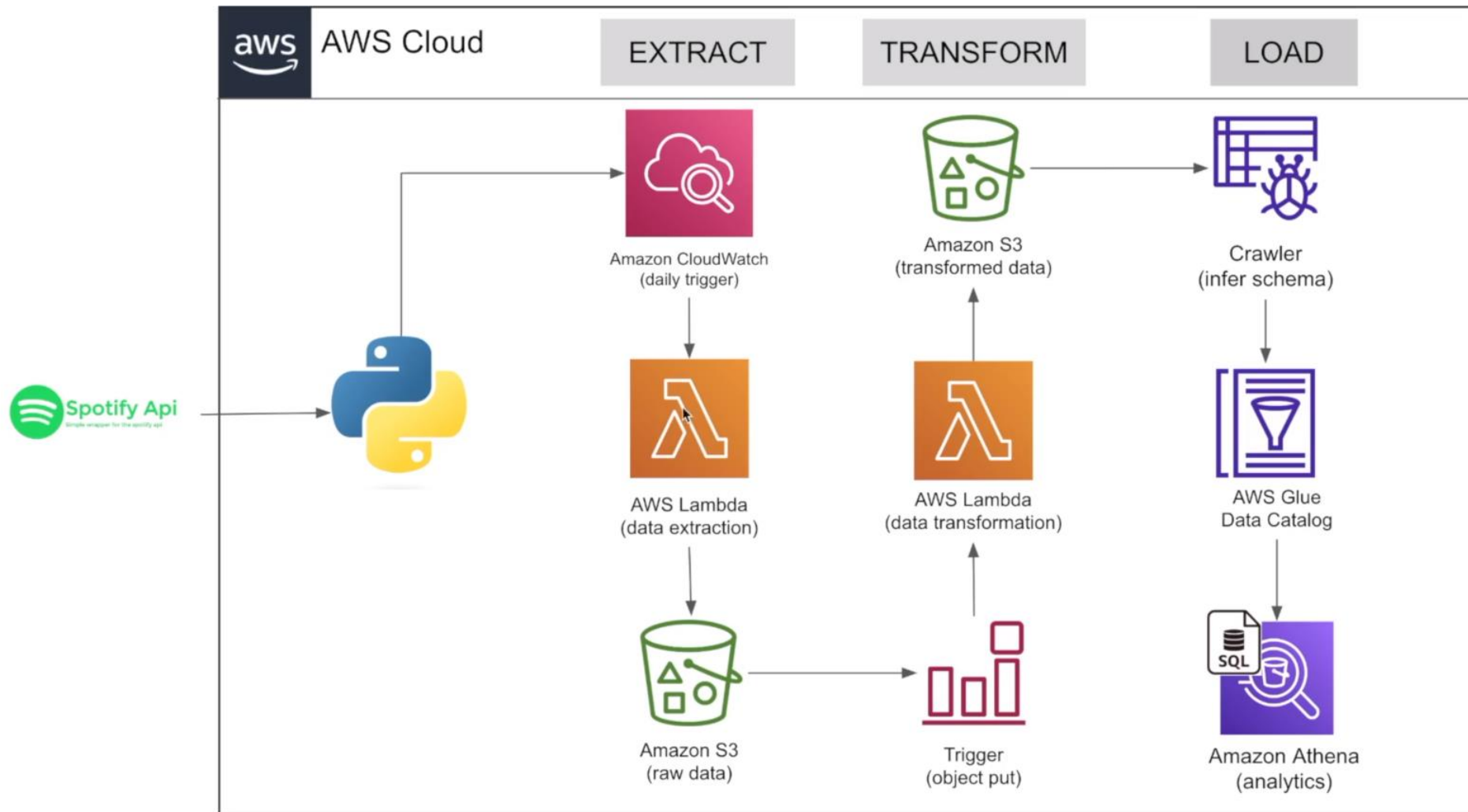


Architecture OF ETL Pipeline Created Using AWS



AWS Lambda ✕

- Dashboard
- Applications
- Functions
- ▼ Additional resources
 - Code signing configurations
 - Layers
 - Replicas
- ▼ Related AWS resources
 - Step Functions state machines

[Lambda](#) > Functions

Functions (2)

Last fetched 3 hours ago 🔄 **Actions** ▾ **Create function**

🔍 Filter by tags and attributes or search by keyword < 1 > ⚙️

<input type="checkbox"/>	Function name ▾	Description ▾	Package type ▾	Runtime ▾	Last modified ▾
<input type="checkbox"/>	spotify_api_data_extraction	-	Zip	Python 3.10	4 hours ago
<input type="checkbox"/>	spotify_transformation_load_function	-	Zip	Python 3.10	4 hours ago

We will use this function to Extract Data from Spotify using the Spotify(API)

We will use this function to transform and load the data into s3 Bucket

BLACKBOX AI

Spotify_api_data_extraction (Function)

The screenshot displays the AWS Lambda console interface for the function 'spotify_api_data_extraction'. The 'Code source' tab is active, showing the Python code for the lambda handler. The code imports necessary libraries (json, os, spotify, boto3, datetime) and defines a lambda_handler function. It retrieves environment variables for client_id and client_secret, initializes a Spotify client, fetches playlist tracks, and uploads the data as a JSON file to an S3 bucket. A blue arrow points from the text annotation to the Spotify client initialization code.

Code source Info Upload from ▾

File Edit Find View Go Tools Window Test ▾ Deploy

Go to Anything (Ctrl-P)

Environment

- spotify_api_data_ex
- lambda_function.py

```
1 import json
2 import os
3 import spotify
4 from spotify.oauth2 import SpotifyClientCredentials
5
6
7 import boto3 # package that connects you with the Amazon services
8 from datetime import datetime
9 def lambda_handler(event, context):
10
11     client_id=os.environ.get('client_id')
12     client_secret=os.environ.get('client_secret')
13
14     client_credentials_manager = SpotifyClientCredentials(client_id=client_id,client_secret=client_secret)
15     sp= spotify.Spotify(client_credentials_manager = client_credentials_manager)
16     playlist_link= "https://open.spotify.com/playlist/37i9dQZEVXBNG2KDCfKOF"
17     playlist_URI=playlist_link.split("/")[-1].split('?')[0]
18     spotify_data = sp.playlist_tracks(playlist_URI)
19     print(spotify_data)
20     client=boto3.client('s3')
21
22     filename="spotify_raw_" + str(datetime.now()) + ".json"
23     client.put_object(
24         Bucket="spotify-etl-project-gurpartee",
25         Key="raw_data/to_processed/" + filename, #Path where you want to store your data
26         Body=json.dumps(spotify_data)
27     )#This will convert entire data into JSON Dictionary Format
```

This code will extract the data from Spotify using spotify(API)

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Spotify_transformation_load_function (Function)

The screenshot shows the AWS Lambda console interface for the function `spotify_transformation_load`. The code source is displayed in a Python editor. The code defines three functions: `album`, `artist`, and `songs`, which extract data from Spotify data. Arrows point from the text on the right to the function definitions in the code.

```
1 import json
2 import boto3
3 from datetime import datetime
4 from io import StringIO # FOR THE CSV FILES
5 import pandas as pd
6
7 def album(data):
8     album_list = []
9     for row in data['items']:
10         album_id = row['track']['album']['id']
11         album_name = row['track']['album']['name']
12         album_release_date = row['track']['album']['release_date']
13         album_total_tracks = row['track']['album']['total_tracks']
14         album_url = row['track']['album']['external_urls']['spotify']
15         album_element = {'album_id': album_id, 'name': album_name, 'release_date': album_release_date,
16                         'total_tracks': album_total_tracks, 'url': album_url}
17         album_list.append(album_element)
18     return album_list
19
20
21 def artist(data):
22     artist_list = []
23     for row in data['items']:
24         for key, value in row.items():
25             if key == "track":
26                 for artist in value['artists']:
27                     artist_dict = {'artist_id': artist['id'], 'artist_name': artist['name'], 'external_url': artist['href']}
28                     artist_list.append(artist_dict)
29
30     return artist_list
31
32
33 def songs(data):
34     song_list = []
35     for row in data['items']:
36         song_id = row['track']['id']
37         song_name = row['track']['name']
```

The 3 functions namely **album**, **artist** and **songs** will extract data such as `album_name`, `artist_name` etc from the spotify data.

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

- spotify_transformat
 - lambda_function.py

```
31
32
33 def songs(data):
34     song_list = []
35     for row in data['items']:
36         song_id = row['track']['id']
37         song_name = row['track']['name']
38         song_duration = row['track']['duration_ms']
39         song_url = row['track']['external_urls']['spotify']
40         song_popularity = row['track']['popularity']
41         song_added = row['added_at']
42         album_id = row['track']['album']['id']
43         artist_id = row['track']['album']['artists'][0]['id']
44         song_element = {'song_id': song_id, 'song_name': song_name, 'duration_ms': song_duration, 'url': song_url,
45                        'popularity': song_popularity, 'song_added': song_added, 'album_id': album_id,
46                        'artist_id': artist_id}
47     song_list.append(song_element)
48
49
50 return song_list
51
52
53
54
55 def lambda_handler(event, context):
56     s3 = boto3.client('s3') # Creating the Object
57     Bucket = "spotify-etl-project-gurpartee"
58     Key = "raw_data/to_processed" # Path where the files are stored
59
60     # This is a function
61     # print(s3.list_objects(Bucket=Bucket, Prefix=Key)['Contents'])
62
63     spotify_data = []
64     spotify_keys = []
65
66     # Here we are using Forloop because we have List and inside the list we
67     # we have Multiple Dictionaries
```

1:1 Python Spaces: 4

Code source Info

Upload from ▾

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

spotify_transformat
lambda_function.py

```
52  
53  
54  
55 def lambda_handler(event, context):  
56     s3= boto3.client('s3') # Creating the Object  
57     Bucket="spotify-etl-project-gurparteek"  
58     Key="raw_data/to_processed" # Path where the files are stored  
59  
60     #This is a function  
61     #print(s3.list_objects(Bucket=Bucket,Prefix=Key)['Contents'])  
62  
63     spotify_data=[]  
64     spotify_keys=[]  
65  
66     # Here we are using Forloop because we have List and inside the list we  
67     # we have Mutiple Dictionaries  
68     for file in s3.list_objects(Bucket=Bucket,Prefix=Key)['Contents']:  
69         #print(file['Key'])  
70         file_key=file['Key']  
71         if file_key.split('.')[1] == "json": # Means it will pick only Json files  
72  
73         # Inside the s3 object we have this function availbale  
74         response =s3.get_object(Bucket = Bucket,Key = file_key)  
75         content=response['Body']  
76         jsonObject= json.loads(content.read())  
77         spotify_data.append(jsonObject)  
78         spotify_keys.append(file_key)  
79         # print(jsonObject)  
80  
81     for data in spotify_data:  
82         album_list= album(data)  
83         artist_list= artist(data)  
84         song_list= songs(data)  
85  
86     # print(album_list)  
87     album_df = pd.DataFrame.from_dict(album_list)  
88     album_df = album_df.drop_duplicates(subset=['album_id'])  
89
```

1:1 Python Spaces: 4

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now we will store the extracted data inside the (raw_data/to_processed) Bucket .

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

lambda_function x Environment Var x Execution results x

Environment

spotify_transformat

lambda_function.py

```
80
81 for data in spotify_data:
82     album_list= album(data)
83     artist_list= artist(data)
84     song_list= songs(data)
85
86     # print(album_list)
87     album_df = pd.DataFrame.from_dict(album_list)
88     album_df = album_df.drop_duplicates(subset=['album_id'])
89
90     artist_df = pd.DataFrame.from_dict(artist_list)
91     artist_df = artist_df.drop_duplicates(subset=['artist_id'])
92
93     song_df = pd.DataFrame.from_dict(song_list)
94
95     album_df['release_date'] = pd.to_datetime(album_df['release_date'])
96     song_df['song_added'] = pd.to_datetime(song_df['song_added'])
97
98
99     songs_key= "transformed_data/songs_data/songs_transformed" + str(datetime.now()) + ".csv"
100    song_buffer=StringIO()# For string conversion
101    song_df.to_csv(song_buffer,index=False) # Will finally covert
102    song_content= song_buffer.getvalue()
103    s3.put_object(Bucket=Bucket,Key=songs_key,Body= song_content)
104
105    album_key= "transformed_data/album_data/album_transformed" + str(datetime.now()) + ".csv"
106    album_buffer=StringIO()# For string conversion
107    album_df.to_csv(album_buffer,index=False) # Will finally covert
108    album_content= album_buffer.getvalue()
109    s3.put_object(Bucket=Bucket,Key=album_key,Body= album_content)
110
111    artist_key= "transformed_data/artist_data/artist_transformed" + str(datetime.now()) + ".csv"
112    artist_buffer=StringIO()# For string conversion
113    artist_df.to_csv(artist_buffer,index=False) # Will finally covert # Index false because glue caller will not be able to access the entire schema
114    artist_content= artist_buffer.getvalue()
115    s3.put_object(Bucket=Bucket,Key=artist_key,Body= artist_content)
116
```

Now we will transform the
JSON data into pandas
Dataframe format.

1:1 Python Spaces: 4

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Tables - AWS Glue Co x spotify-etl-project-gu x Functions - Lambda x spotify_api_data_extro x spotify_transformatio x Query editor | Athen x Jeremy Zucker - c x +

https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/spotify_transformation_load_function?tab=code

aws Services Search [Alt+S] N. Virginia Gurpartee Gill

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

- spotify_transformat
- lambda_function.py

```
105 album_key= "transformed_data/album_data/album_transformed" + str(datetime.now()) + ".csv"
106 album_buffer=StringIO()# For string conversion
107 album_df.to_csv(album_buffer,index=False) # Will finally covert
108 album_content= album_buffer.getvalue()
109 s3.put_object(Bucket=Bucket,Key=album_key,Body= album_content)
110
111 artist_key= "transformed_data/artist_data/artist_transformed" + str(datetime.now()) + ".csv"
112 artist_buffer=StringIO()# For string conversion
113 artist_df.to_csv(artist_buffer,index=False) # Will finally covert # Index false because glue caller will not be able to access the entire schema
114 artist_content= artist_buffer.getvalue()
115 s3.put_object(Bucket=Bucket,Key=artist_key,Body= artist_content)
116
117
118
119 s3_resource = boto3.resource('s3')
120 for key in spotify_keys:
121     copy_source = {
122         'Bucket':Bucket,
123         'Key':key
124     }
125     #Targated Bucket
126 s3_resource.meta.client.copy(copy_source,Bucket,'raw_data/processed/' + key.split("/")[-1])
127 s3_resource.Object(Bucket,key).delete()
128
```

This will store the transformed data into (raw_data/processed) Bucket.

1:1 Python Spaces: 4

s3 Bucket



Amazon S3

Tables - AWS Glue Co

S3 buckets | S3 | Glob

Functions - Lambda

spotify_api_data_ext

spotify_transformati

Query editor | Athen

Jeremy Zucker - c

← → ↺ 🏠 🔍 https://s3.console.aws.amazon.com/s3/buckets?region=us-east-1 ☆ 🗂 📄 6 ⋮

aws Services 🔍 Search [Alt+S] 📧 🔔 ⓘ ⚙ Global ▾ Gurpartee Gill ▾

Amazon S3 ✕

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 7

► AWS Marketplace for S3

Amazon S3 > Buckets ⓘ

► Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

Buckets (1) Info

Refresh

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

🔍 Find buckets by name

< 1 > ⚙

	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
○	spotify-etl-project-gurpartee	US East (N. Virginia) us-east-1	Bucket and objects not public	October 26, 2023, 18:02:39 (UTC+05:30)

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

▼ Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 7

- ▶ AWS Marketplace for S3

[Amazon S3](#) > [Buckets](#) > spotify-etl-project-gurparteeek

[spotify-etl-project-gurparteek](#) [Info](#)

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

 Copy S3 URI

URL



Download



A




Actions

Create folder



load

Find objects by prefix

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	 raw_data/	Folder	-	-	-
<input type="checkbox"/>	 transformed_data/	Folder	-	-	-
<input type="checkbox"/>	 Unsaved/	Folder	-	-	-

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 7

AWS Marketplace for S3

Amazon S3

Buckets

spotify-etl-project-gurpartee

raw_data/

raw_data/

Copy S3 URI

Objects

Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

🔄

Copy S3 URI

Copy URL

Download

Open

Delete

Actions ▾

Create folder

Upload

Find objects by prefix

< 1 > ⚙

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	processed/	Folder	-	-	-
<input type="checkbox"/>	to_processed/	Folder	-	-	-

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 7

AWS Marketplace for S3

Amazon S3 > Buckets > spotify-etl-project-gurpartee > transformed_data/

transformed_data/

Copy S3 URI

Objects Properties

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	album_data/	Folder	-	-	-
<input type="checkbox"/>	artist_data/	Folder	-	-	-
<input type="checkbox"/>	songs_data/	Folder	-	-	-



Crawler

Browser tabs: Crawlers - AWS Glue, spotify-etl-project-gu..., Functions - Lambda, spotify_api_data_extro..., spotify_transformatio..., Query editor | Athen..., Gucci Chick - You...

Browser address bar: https://us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data-catalog/crawlers

AWS Glue Services Search [Alt+S]

AWS Glue

- Getting started
- ETL jobs
 - Visual ETL
 - Notebooks
 - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- Data Catalog**
 - Databases
 - Tables
 - Stream schema registries
 - Schemas
 - Connections
 - Crawlers**
 - Classifiers
 - Catalog settings
- Data Integration and ETL
- Legacy pages

What's New Documentation AWS Marketplace

Crawler successfully starting
The following crawler is now starting: "spotify_album_crawler"

[AWS Glue](#) > Crawlers

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.

Crawlers (3) Info Last updated (UTC) October 27, 2023 at 13:57:27 Refresh Action Run Create crawler

View and manage all available crawlers.

<input type="checkbox"/>	Name	State	Schedule	Last run	Last run timesta...	Log	Table changes fro...
<input type="checkbox"/>	spotify_album_craw...	✓ Ready		✓ Succeeded	October 27, 2023 a...	View log	1 created
<input type="checkbox"/>	spotify_artists_craw...	✓ Ready		✓ Succeeded	October 27, 2023 a...	View log	1 created
<input type="checkbox"/>	spotify_songs_crawler	✓ Ready		✓ Succeeded	October 27, 2023 a...	View log	1 created

Here we have created 3 crawlers(album,artists,songs)

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Amazon Athena

Query editor | Athena

https://us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#/query-editor/history/e610ca24-9df2-4d81-95ab-238cd321770c

Services Search [Alt+S]

Amazon Athena > Query editor

Editor Recent queries Saved queries Settings

Workgroup primary

Data

Data source: AwsDataCatalog

Database: spotify_db

Tables and views: Create Filter tables and views

Tables (3)

- album_data
- artist_data
- songs_data

Views (0)

Query 2 Query 3 Query 4 Query 5

```
1 SELECT name,SUM(total_tracks) FROM "spotify_db"."album_data"
2 group by name
3 order by SUM(total_tracks) desc;
```

SQL Ln 3, Col 34

Run again Explain Cancel Clear Create

Reuse query results up to 60 minutes ago

Query results Query stats

Completed Time in queue: 113 ms Run time: 418 ms Data scanned: 12.73 KB

Here you can write SQL queries

Now, you can see 3 different Tables (album,artists,songs)

This is the preview of artists table.

Query results

Query stats

Completed

Time in queue: 190 msRun time: 455 msData scanned: 8.97 KB

Results (10)

Copy

Download results

Search rows

<1>

⚙

#	artist_id	artist_name	external_url
1	4q3ewBCX7sLwd24euv69X	Bad Bunny	https://api.spotify.com/v1/artists/4q3ewBCX7sLwd24euv69X
2	6HaGTQPmzraVmaVxvz6EUc	Jung Kook	https://api.spotify.com/v1/artists/6HaGTQPmzraVmaVxvz6EUc
3	3MdXrJWsbVzdn6fe5JYkSQ	Latto	https://api.spotify.com/v1/artists/3MdXrJWsbVzdn6fe5JYkSQ
4	5cj0LLjcoR7YOSnhnX0Po5	Doja Cat	https://api.spotify.com/v1/artists/5cj0LLjcoR7YOSnhnX0Po5
5	2LRolwIKmHjgvigdNGBHNo	Feid	https://api.spotify.com/v1/artists/2LRolwIKmHjgvigdNGBHNo
6	45dkTj5sMRSjrmBSBeiHym	Tate McRae	https://api.spotify.com/v1/artists/45dkTj5sMRSjrmBSBeiHym
7	0jbo7KFNMilkfBR6ih0yhm	iñigo quintero	https://api.spotify.com/v1/artists/0jbo7KFNMilkfBR6ih0yhm
8	3qsKSpcV3ncke3hw52JSMB	Young Miko	https://api.spotify.com/v1/artists/3qsKSpcV3ncke3hw52JSMB
9	06HL4z0CvFAxyc27GXpf02	Taylor Swift	https://api.spotify.com/v1/artists/06HL4z0CvFAxyc27GXpf02
10	7uMDnSZyUYNBPLhPMNuaM2	Kenya Grace	https://api.spotify.com/v1/artists/7uMDnSZyUYNBPLhPMNuaM2

This is the preview
of songs table

Completed Time in queue: 307 ms Run time: 879 ms Data scanned: 8.54 KB					
Results (10) Copy Download results					
<input type="text" value="Search rows"/>					
#	song_id	song_name	duration_ms	url	
1	4MjDJD8cW7iVeWInc2Bdyj	MONACO	267194	https://open.spotify.com/track/4MjDJD8cW7iVeWInc2Bdyj	
2	7x9aauaA9cu6tyfpHnqDLo	Seven (feat. Latto) (Explicit Ver.)	184400	https://open.spotify.com/track/7x9aauaA9cu6tyfpHnqDLo	
3	56y1jOTK0XSvJzVv9vHQBK	Paint The Town Red	230480	https://open.spotify.com/track/56y1jOTK0XSvJzVv9vHQBK	
4	7iQXYTyuG13aoeHxGG28Nh	PERRO NEGRO	162767	https://open.spotify.com/track/7iQXYTyuG13aoeHxGG28Nh	
5	3rUGC1vUpkDG9CZFHMur1t	greedy	131872	https://open.spotify.com/track/3rUGC1vUpkDG9CZFHMur1t	
6	2HafqoJbgXdtjwCOvNEF14	Si No Estás	184061	https://open.spotify.com/track/2HafqoJbgXdtjwCOvNEF14	
7	3nNmRE0DxHC6ZaKkrpUumS	FINA	216327	https://open.spotify.com/track/3nNmRE0DxHC6ZaKkrpUumS	
8	1BxfuPKGuaTgP7aM0Bbdwr	Cruel Summer	178426	https://open.spotify.com/track/1BxfuPKGuaTgP7aM0Bbdwr	
9	5mjYQaktjmjcMKcUlcqz4s	Strangers	172964	https://open.spotify.com/track/5mjYQaktjmjcMKcUlcqz4s	
10	01qFKNWq73UfEsII0GvumE	3D (feat. Jack Harlow)	201812	https://open.spotify.com/track/01qFKNWq73UfEsII0GvumE	