TIme to Query Data From Athena

In [ ]:

```
%store -r ingest_create_athena_table_csv_passed
```

In [ ]:

```
try:
    ingest_create_athena_table_csv_passed
except NameError:
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL PREVIOUS NOTEBOOKS.  You did not register th
    print("++++++++++++++++++++++++++++++++++++++++++++++")
```

In [ ]:

```
print(ingest_create_athena_table_csv_passed)
```

```
True
```

In [ ]:

```
if not ingest_create_athena_table_csv_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL PREVIOUS NOTEBOOKS.  You did not register th
    print("++++++++++++++++++++++++++++++++++++++++++++++")
else:
    print("[OK]")
```

```
[OK]
```

In [ ]:

```
%store
```

```
Stored variables and their in-db values:
data_path                                         -> '/root/AAI-540/Module2/csv'
ingest_create_athena_db_mod2_passed               -> True
ingest_create_athena_table_csv_passed             -> True
s3_private_path_csv                               -> 's3://sagemaker-us-east-1-00460
8622582/module2_dat
setup_dependencies_mod2_passed                    -> True
setup_s3_bucket_passed                            -> True
```

Setup

In [ ]:

```
import sagemaker
import boto3

sess = sagemaker.Session()
bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = boto3.Session().region_name

sm = boto3.Session().client(service_name="sagemaker", region_name=region)
```

In [ ]:

```
import awswrangler as wr
```

Query From Glue Catalog

In [ ]:

```
database_name = "mod2_db"
table_name_csv = "music"
```

In [ ]:

```
for table in wr.catalog.get_tables(database="mod2_db"):
    print(table["Name"])
```

```
music
```

Query From Athena

In [ ]:

```
%%time
df = wr.athena.read_sql_query(sql="SELECT * FROM {}.{} LIMIT 5000".format(database_
```

```
CPU times: user 467 ms, sys: 34.1 ms, total: 501 ms
Wall time: 2.86 s
```

In [ ]:

```
df.head()
```

Out[ ]:

	track_id	artists	popularity	duration_ms	explicit	danceabili
0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	73	230666	False	0.67
1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	55	149610	False	0.42
2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	57	210826	False	0.43
3	6lfxq3CG4xtTiEg7opyCyx	Kina Grannis	71	201933	False	0.26
4	5vjLSffimiIP26QG5WcN2K	Chord Overstreet	82	198853	False	0.6

QUERY Tasks

1. List artist, track_name, and popularity for songs that have a popularity greater than or equal to 99

In [ ]:

```
%%time
df = wr.athena.read_sql_query(sql="SELECT artists, popularity FROM {} WHERE popular
```

```
CPU times: user 688 ms, sys: 27 ms, total: 715 ms
Wall time: 2.74 s
```

In [ ]:

```
df.head()
```

Out[ ]:

	artists	popularity
0	Sam Smith;Kim Petras	100
1	Bizarrap;Quevedo	99
2	Sam Smith;Kim Petras	100

2. List artists with an average popularity of 92

In [ ]:
```
%%time
df = wr.athena.read_sql_query(sql="SELECT artists, AVG(popularity) AS avg_popularit
```

```
CPU times: user 460 ms, sys: 40.7 ms, total: 501 ms
Wall time: 2.84 s
```

In [ ]:
```
df.head()
```

Out[ ]:

	artists	avg_popularity
0	Harry Styles	92.0
1	Rema;Selena Gomez	92.0

3.List the Top 10 most energetic genres

In [ ]:
```
%%time
df = wr.athena.read_sql_query(sql="SELECT AVG(energy) as avg_energy, track_genre FR
```

```
CPU times: user 443 ms, sys: 36.7 ms, total: 480 ms
Wall time: 2.77 s
```

In [ ]:
```
df.head(10)
```

Out[ ]:

	avg_energy	track_genre
0	0.931470	death-metal
1	0.924201	grindcore
2	0.914220	metalcore
3	0.910971	happy
4	0.901246	hardstyle
5	0.876617	drum-and-bass
6	0.874897	black-metal
7	0.874003	heavy-metal
8	0.871237	party
9	0.868677	j-idol

4. How many tracks is Bad Bunny On?

In [ ]:
```
%%time
df = wr.athena.read_sql_query(sql="SELECT COUNT(*) AS bb_count FROM {}.{} WHERE art
```

```
CPU times: user 440 ms, sys: 20.3 ms, total: 460 ms
Wall time: 2.75 s
```

In [ ]:

```
df.head()
```

Out[ ]:

	bb_count
0	48

5. Show the Top 10 genres in terms of popularity sorted by their most popular track

In [ ]:

```
%%time
df = wr.athena.read_sql_query(sql="SELECT MAX(popularity) as max_popularity, track_
```

```
CPU times: user 655 ms, sys: 40.9 ms, total: 696 ms
Wall time: 3.04 s
```

In [ ]:

```
df.head(10)
```

Out[ ]:

	max_popularity	track_genre
0	100	dance
1	100	pop
2	99	hip-hop
3	98	reggae
4	98	reggaeton
5	98	latino
6	98	latin
7	98	edm
8	96	piano
9	96	rock

Release Resources

In [ ]:

```
%%html

<p><b>Shutting down your kernel for this notebook to release resources.</b></p>
<button class="sm-command-button" data-commandlinker-command="kernelmenu:shutdown"

<script>
try {
    els = document.getElementsByClassName("sm-command-button");
    els[0].click();
}
catch(err) {
    // NoOp
}
</script>
```