

Question 1

Amazon S3 > Buckets > bigdata-dzf-2 > bigdata_hw/

bigdata_hw/

Copy S3 URI

Objects Properties

Objects (1)

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

< 1 > ⚙

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Crimes_2001_to_Present.csv	csv	February 6, 2023, 18:47:55 (UTC-06:00)	1.7 GB	Standard

Question 2 and 3

Query 1 Query 2

+ ▼

```
1 show databases;
2 create external table chicago_crimes (
3   id string,
4   case_number string,
5   occurred_at string,
6   block string,
7   iucr string,
8   primary_type string,
9   description string,
10  location_description string,
11  arrest string,
12  domestic string,
13  beat string,
14  district string,
15  ward string,
16  community_area string,
17  fbi_code string,
18  x_coordinate string,
19  y_coordinate string,
20  year string,
21  updated_on string,
22  latitude string,
23  longitude string,
24  location string
25 )
26 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
27 STORED AS TEXTFILE
28 LOCATION 's3://bigdata-dzf-2/bigdata_hw/'
29 tblproperties("skip.header.line.count"="1");
```

SQL Ln 18, Col 25

⌕ ⌕ ⚙

Run again Explain Cancel Clear Create

Reuse query results
*Athena engine version 3 only

Query results Query stats

Completed

Time in queue: 46 ms Run time: 291 ms Data scanned: -

Query successful.

Question 4

Query 1 : X

Query 2 : X

+ ▼

1 select min(occurred_at) from chicago_crimes;

SQL Ln 1, Col 38

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results
*Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 99 ms Run time: 1.081 sec Data scanned: 1.70 GB

Results (1)

Copy

Download results

Search rows

< 1 > ⚙

▼ _col0 ▼

1 01/01/2001 01:00:00 AM

Query 1 : X

Query 2 : X

+ ▼

1 select max(occurred_at) from chicago_crimes;

SQL Ln 1, Col 11

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results
*Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 121 ms Run time: 997 ms Data scanned: 1.70 GB

Results (1)

Copy

Download results

Search rows

< 1 > ⚙

▼ _col0 ▼

1 12/31/2022 12:59:00 PM

Question 5

Query 1 : X

Query 2 : X

+ ▼

1 select * from (select primary_type, count(*) as cnt from chicago_crimes group by primary_type) t order by cnt desc limit 5;

SQL Ln 1, Col 124

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results
*Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 195 ms Run time: 1.391 sec Data scanned: 1.70 GB

Results (5)

Copy

Download results

Search rows

< 1 > ⚙

▼ primary_type ▼ cnt ▼

1 THEFT 1630499

2 BATTERY 1413736

3 CRIMINAL DAMAGE 881147

4 NARCOTICS 745909

5 ASSAULT 502622

Query 1 : X

Query 2 : X

+▼

1

select * from (select primary_type, count(*) as cnt from chicago_crimes group by primary_type) t order by cnt limit 5;

SQL Ln 1, Col 119

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results

Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 107 ms

Run time: 1.1 sec

Data scanned: 1.70 GB

Results (5)

Copy

Download results

Search rows

< 1 > ⚙

# ▼	primary_type ▼	cnt ▼
1	DOMESTIC VIOLENCE	1
2	NON-CRIMINAL (SUBJECT SPECIFIED)	9
3	RITUALISM	24
4	NON - CRIMINAL	38
5	HUMAN TRAFFICKING	99

Question 6

Query 1 : X

Query 2 : X

+▼

1

select * from (select location_description, count(*) as cnt from chicago_crimes where primary_type = 'HOMICIDE' group by location_description) t order by cnt desc limit 5;

SQL Ln 1, Col 172

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results

Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 200 ms

Run time: 1.636 sec

Data scanned: 1.70 GB

Results (5)

Copy

Download results

Search rows

< 1 > ⚙

# ▼	location_description ▼	cnt ▼
1	STREET	6229
2	AUTO	1353
3	APARTMENT	1050
4	ALLEY	775
5	HOUSE	637

Question 7

Query 1 : X

Query 2 : X

+ | ▾

1 select * from (select district, count(*) as cnt from chicago_crimes group by district) t order by cnt desc limit 5;

SQL Ln 1, Col 116

Run again

Explain

Cancel

Clear

Create ▾

Reuse query results

Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 220 ms

Run time: 1.286 sec

Data scanned: 1.70 GB

Results (5)

Copy

Download results

Search rows

< 1 > ⚙

# ▾	district ▾	cnt ▾
1	008	500877
2	011	482400
3	006	437951
4	007	434837
5	025	427382

Query 1 : X

Query 2 : X

+ | ▾

1 select * from (select district, count(*) as cnt from chicago_crimes group by district) t order by cnt limit 5;

SQL Ln 1, Col 102

Run again

Explain

Cancel

Clear

Create ▾

Reuse query results

Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 100 ms

Run time: 1.027 sec

Data scanned: 1.70 GB

Results (5)

Copy

Download results

Search rows

< 1 > ⚙

# ▾	district ▾	cnt ▾
1	2123	1
2	1813	1
3	1225	1
4	0723	1
5	1324	1

Question 8

Query 1Query 2Query 3Query 5

```
1 select sum(cnt)/count(time) from (select date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'),'%Y-%m-%d') as time, count(*) as cnt from chicago_crimes
2 where primary_type = 'ASSAULT'
3 and date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'),'%Y-%m-%d') >= '2021-01-01' and date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'
4 group by date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'),'%Y-%m-%d'));
```

SQLLn 1, Col 15

Run againExplainCancelClearCreate

Reuse query results
*Athena engine version 3 only

Query resultsQuery stats

CompletedTime in queue: 119 msRun time: 1.129 secData scanned: 1.70 GB

Results (1)

Search rows

<1>

#	_col0
1	55

Query 1Query 2Query 3Query 5

```
1 select sum(cnt)/count(time) from (select date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'),'%Y-%m-%d') as time, count(*) as cnt from chicago_crimes
2 where primary_type = 'ASSAULT'
3 and date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'),'%Y-%m-%d') >= '2020-01-01' and date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'
4 group by date_format(date_parse(SUBSTRING(occurred_at, 1, 10),'%m/%d/%Y'),'%Y-%m-%d'));
```

SQLLn 3, Col 187

Run againExplainCancelClearCreate

Reuse query results
*Athena engine version 3 only

Query resultsQuery stats

CompletedTime in queue: 127 msRun time: 1.189 secData scanned: 1.70 GB

Results (1)

Search rows

<1>

#	_col0
1	49

The average number in 2021 is 55, and in 2020 is 49. So compared to prior year, the average number of assaults is increased.

Question 9

Query 1Query 2Query 3Query 5

```
1 create table chicago_crimes_parquet
2 with (format = 'parquet')
3 as select * from chicago_crimes;
```

SQLLn 3, Col 33

Run againExplainCancelClearCreate

Reuse query results
*Athena engine version 3 only

Query resultsQuery stats

CompletedTime in queue: 94 msRun time: 8.63 secData scanned: 1.70 GB

Query successful.

Question 10

Query 1 : X

Query 5 : X

+ ▼

1 EXPLAIN ANALYZE

2 select max(occurred_at) from chicago_crimes;

SQL Ln 2, Col 45

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results

Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 96 ms

Run time: 1.328 sec

Data scanned: 1.70 GB

Query Plan

Fragment 1

CPU: 12.91ms, Input: 55 rows (1.45kB), Data Scanned: 0B; per task: std.dev.: 0.00, Output: 1 row (27B)

Output layout: [max]

- Aggregate(FINAL) => [[max]]

CPU: 2.00ms (0.01%), Output: 1 row (27B)

Input avg.: 55.00 rows, Input std.dev.: 0.00%

max := "max"("max_4")

- LocalExchange[SINGLE] () => [[max_4]]

CPU: 2.00ms (0.01%), Output: 55 rows (1.45kB)

Input avg.: 13.75 rows, Input std.dev.: 51.78%

- RemoteSource[2] => [[max_4]]

CPU: 3.00ms (0.01%), Output: 55 rows (1.45kB)

Input avg.: 13.75 rows, Input std.dev.: 51.78%

Fragment 2

CPU: 23.82s, Input: 7728392 rows (199.00MB), Data Scanned: 1.70GB; per task: std.dev.: 36698.29, Output: 55 rows (1.45kB)

Output layout: [max_4]

- Aggregate(PARTIAL) => [[max_4]]

CPU: 327.00ms (1.37%), Output: 55 rows (1.45kB)

Input avg.: 140516.22 rows, Input std.dev.: 6.26%

max_4 := "max"("occurred_at")

- TableScan[awsdatacatalog:HiveTableHandle(schemaName=crime, tableName=chicago_crimes, analyzePartitionValues=Optional.empty), grouped = false] => [[occurred_at]]

CPU: 23.49s (98.60%), Output: 7728392 rows (199.00MB)

Input avg.: 140516.22 rows, Input std.dev.: 6.26%

LAYOUT: crime.chicago_crimes

occurred_at := occurred_at:string:2:REGULAR

Query 1 : X

Query 5 : X

+ ▼

1 EXPLAIN ANALYZE

2 select max(occurred_at) from chicago_crimes_parquet;

SQL Ln 2, Col 52

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results

Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 185 ms

Run time: 1.121 sec

Data scanned: 26.59 MB

Query Plan

Fragment 1

CPU: 3.57ms, Input: 26 rows (702B), Data Scanned: 0B; per task: std.dev.: 0.00, Output: 1 row (27B)

Output layout: [max]

- Aggregate(FINAL) => [[max]]

CPU: 1.00ms (0.03%), Output: 1 row (27B)

Input avg.: 26.00 rows, Input std.dev.: 0.00%

max := "max"("max_4")

- LocalExchange[SINGLE] () => [[max_4]]

CPU: 0.00ms (0.00%), Output: 26 rows (702B)

Input avg.: 6.50 rows, Input std.dev.: 95.77%

- RemoteSource[2] => [[max_4]]

CPU: 0.00ms (0.00%), Output: 26 rows (702B)

Input avg.: 6.50 rows, Input std.dev.: 95.77%

Fragment 2

CPU: 3.40s, Input: 7728392 rows (199.00MB), Data Scanned: 26.59MB; per task: std.dev.: 11554.00, Output: 26 rows (702B)

Output layout: [max_4]

- Aggregate(PARTIAL) => [[max_4]]

CPU: 612.00ms (17.98%), Output: 26 rows (702B)

Input avg.: 297245.85 rows, Input std.dev.: 9.98%

max_4 := "max"("occurred_at")

- TableScan[awsdatacatalog:HiveTableHandle(schemaName=crime, tableName=chicago_crimes_parquet, analyzePartitionValues=Optional.empty), grouped = false] => [[occurred_at]]

CPU: 2.79s (81.99%), Output: 7728392 rows (199.00MB)

Input avg.: 297245.85 rows, Input std.dev.: 9.98%

LAYOUT: crime.chicago_crimes_parquet

occurred_at := occurred_at:string:2:REGULAR

Parquet format is more efficient than normal format when extracting the data from the tables.

Question 11

Query 1 : X


Query 5 : X

+ ▼

```
1 create table crimes_for_download as
2 SELECT
3     primary_type,
4     community_area,
5     count(*) as count
6 FROM chicago_crimes
7 GROUP BY primary_type, community_area;
```

SQL Ln 7, Col 39

Run again

Explain 

Cancel

Clear

Create ▼

☐ Reuse query results
*Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 98 ms Run time: 2.603 sec Data scanned: 1.70 GB

Query successful.

I preview the table and download the result. The resulting file is csv.