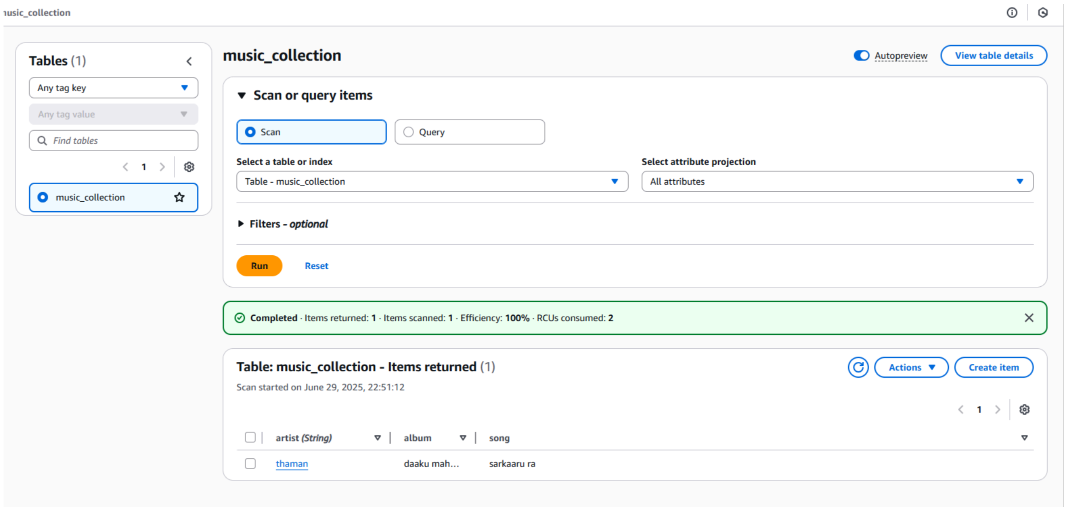
partiQL

DynamoDB Table :



music_collection

● Scan ○ Query

Select a table or index: Table - music_collection

Select attribute projection: All attributes

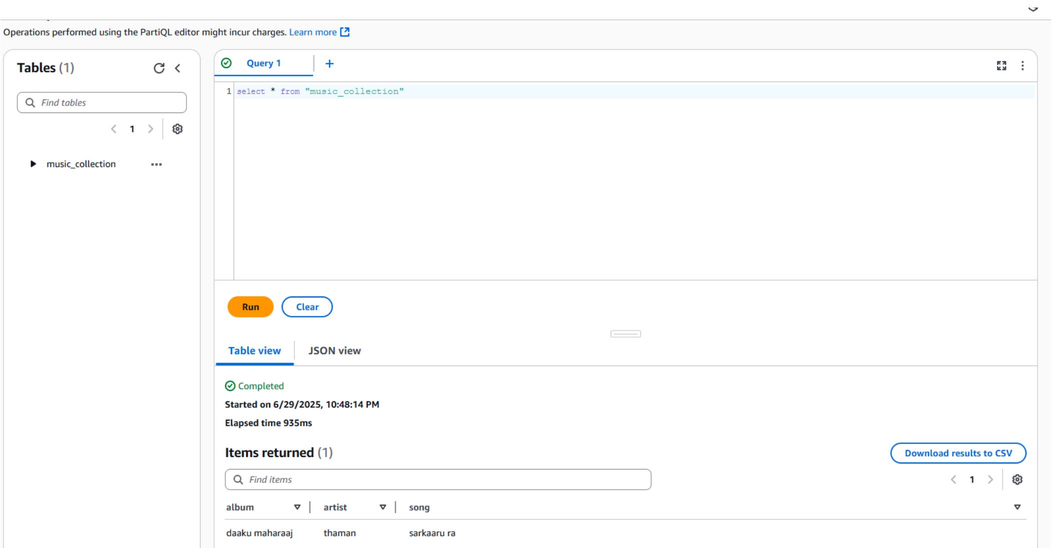
Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 2

Table: music_collection - Items returned (1)

Scan started on June 29, 2025, 22:51:12

artist (String)	album	song
thaman	daaku mah...	sarkaaru ra

Select (Read) :



Operations performed using the PartiQL editor might incur charges. Learn more

```
select * from "music_collection"
```

Completed

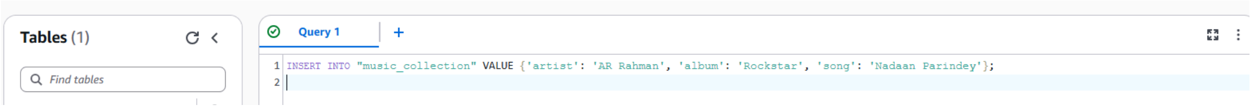
Started on 6/29/2025, 10:48:14 PM

Elapsed time 935ms

Items returned (1)

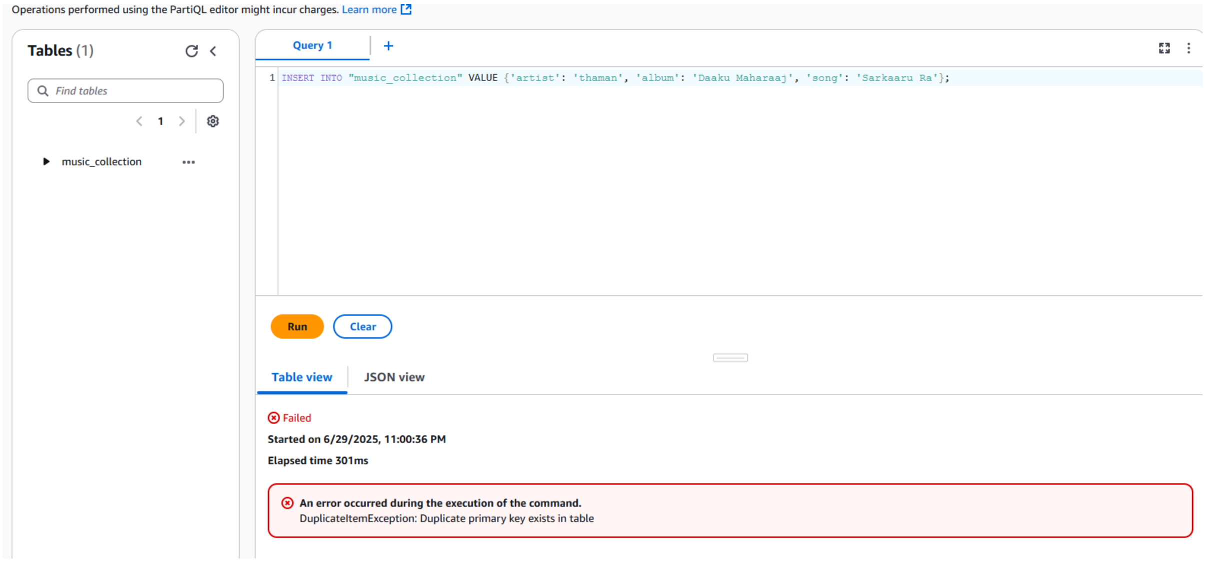
album	artist	song
daaku maharaaj	thaman	sarkaaru ra

Insert (Create):



```
INSERT INTO "music_collection" VALUE {'artist': 'AR Rahman', 'album': 'Rockstar', 'song': 'Nadaan Parindey'};
```

Duplicate primary key :



Operations performed using the PartiQL editor might incur charges. Learn more

```
INSERT INTO "music_collection" VALUE {'artist': 'thaman', 'album': 'Daaku Maharaaj', 'song': 'Sarkaaru Ra'};
```

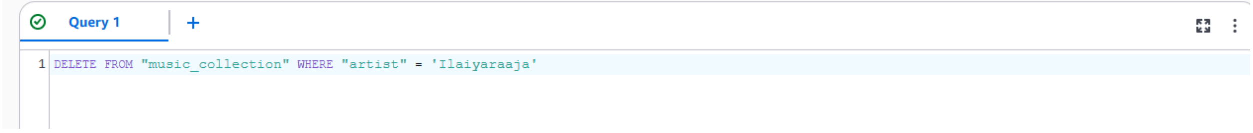
Failed

Started on 6/29/2025, 11:00:36 PM

Elapsed time 301ms

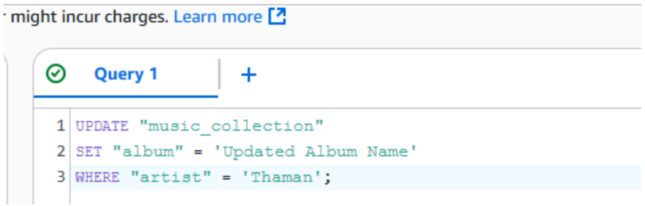
An error occurred during the execution of the command.
DuplicateItemException: Duplicate primary key exists in table

Delete :



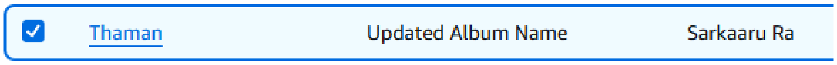
```
DELETE FROM "music_collection" WHERE "artist" = 'Ilaiyaraaja'
```

Update :



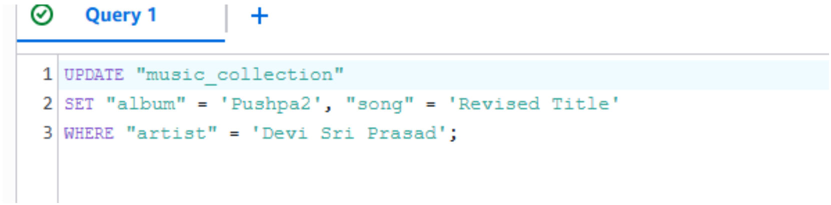
might incur charges. Learn more

```
UPDATE "music_collection"
SET "album" = 'Updated Album Name'
WHERE "artist" = 'Thaman';
```

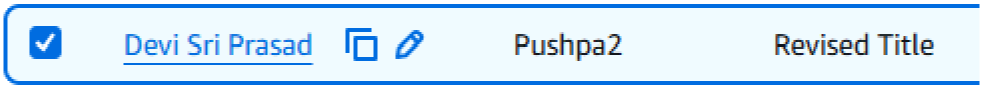


☑	Thaman	Updated Album Name	Sarkaaru Ra

Update multiple fields based condition :

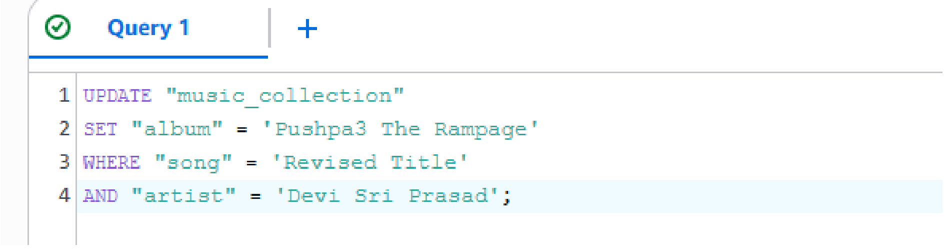


```
UPDATE "music_collection"
SET "album" = 'Pushpa2', "song" = 'Revised Title'
WHERE "artist" = 'Devi Sri Prasad';
```

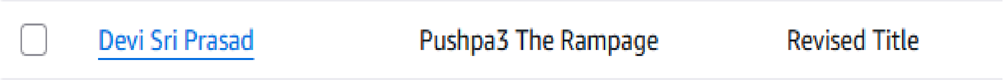


☑	Devi Sri Prasad	Pushpa2	Revised Title

Update based on multiple conditions :

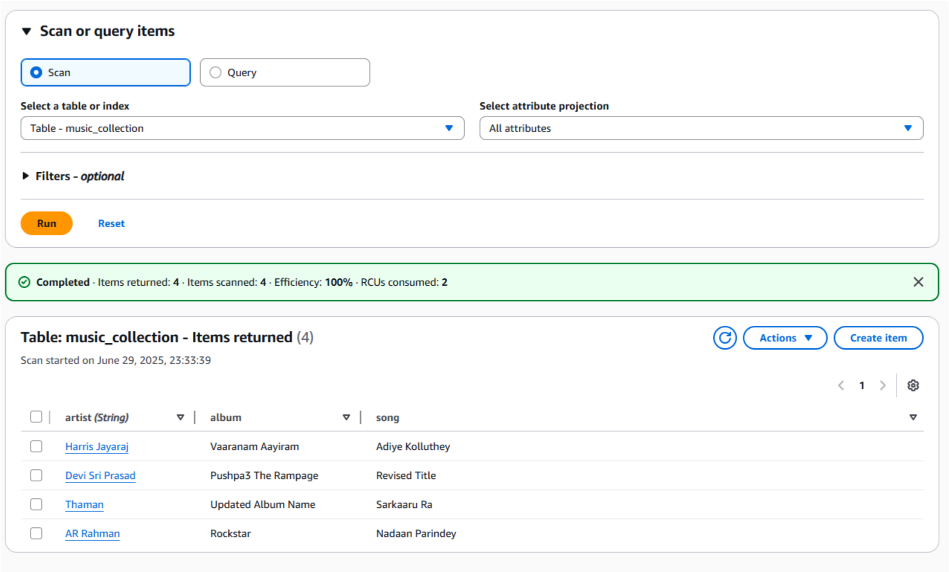


```
UPDATE "music_collection"
SET "album" = 'Pushpa3 The Rampage'
WHERE "song" = 'Revised Title'
AND "artist" = 'Devi Sri Prasad';
```



□	Devi Sri Prasad	Pushpa3 The Rampage	Revised Title

Scan :



▼ Scan or query items

● Scan ○ Query

Select a table or index: Table - music_collection

Select attribute projection: All attributes

▶ Filters - *optional*

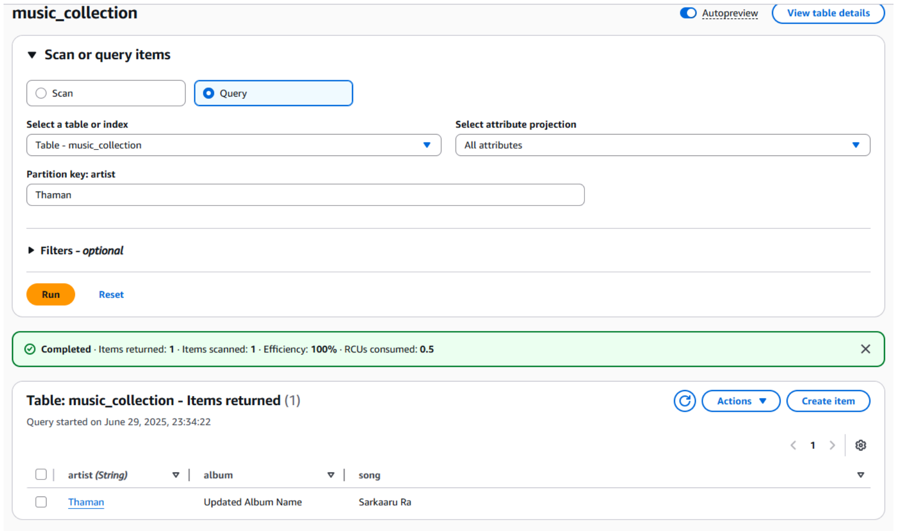
Completed · Items returned: 4 · Items scanned: 4 · Efficiency: 100% · RCUs consumed: 2

Table: music_collection - Items returned (4)

Scan started on June 29, 2025, 23:33:39

□	artist (String)	album	song
□	Harris Jayaraj	Vaaranam Aayiram	Adiye Kolluthey
□	Devi Sri Prasad	Pushpa3 The Rampage	Revised Title
□	Thaman	Updated Album Name	Sarkaaru Ra
□	AR Rahman	Rockstar	Nadaan Parindey

Query :



music_collection

▼ Scan or query items

○ Scan ● Query

Select a table or index: Table - music_collection

Select attribute projection: All attributes

Partition key: artist: Thaman

▶ Filters - *optional*

Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 0.5

Table: music_collection - Items returned (1)

Query started on June 29, 2025, 23:34:22

□	artist (String)	album	song
□	Thaman	Updated Album Name	Sarkaaru Ra

In PartiQL, `Query` and `Scan` are two operations used to retrieve data, but they are very different in efficiency and purpose:

🔍 Query

- **Purpose: Retrieve specific items using the partition key (and optional sort key)**
- **Efficient: Looks up items by index, so it's fast and cost-effective**

Example

```
SELECT * FROM "music_collection" WHERE "artist" = 'Ilaiyaraaja'
```

This is a Query operation (since `artist` is the partition key).

✅ Best used when:

- **You know the partition key value**
- **Your access patterns are predictable**

🧹 Scan

- **Purpose: Reads every item in the table (full table scan**
- **Slower and costlier: Especially with large tables.**

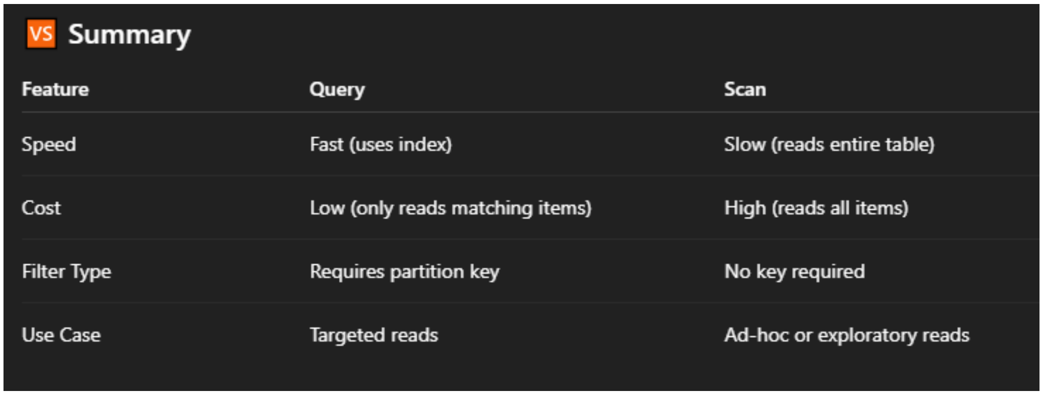
Example:

```
SELECT * FROM "music_collection"
```

This is a Scan operation — reads all items, no filters

❌ Avoid for:

- **Large tables with tight latency or cost budgets.**



VS Summary

Feature	Query	Scan
Speed	Fast (uses index)	Slow (reads entire table)
Cost	Low (only reads matching items)	High (reads all items)
Filter Type	Requires partition key	No key required
Use Case	Targeted reads	Ad-hoc or exploratory reads

You cannot create a DynamoDB table using PartiQL. PartiQL in AWS **only supports data manipulation (CRUD)**, not schema definition for DynamoDB (no create, alter, drop)

Examples :

SELECT * FROM "music_collection"

WHERE "artist" = 'AR Rahman' AND "song" >= 'N'

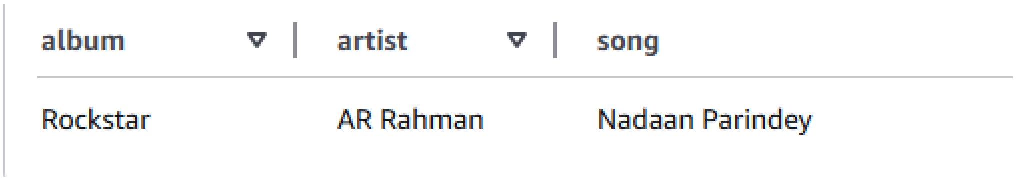
SELECT * FROM "music_collection"

WHERE "artist" = 'AR Rahman' AND "song" BETWEEN 'A' AND 'Z'

SELECT * FROM "music_collection"

WHERE "artist" = 'AR Rahman' AND BEGINS_WITH("song", 'Nadaan')

Same output for above three :



album	artist	song
Rockstar	AR Rahman	Nadaan Parindey