

# Part 1:

## Creating Role and assigning policy

The screenshot displays the AWS IAM console interface for the role `role_rr3493`. The left sidebar shows the navigation menu with sections like Identity and Access Management (IAM), Access management, and Access reports. The main content area shows the role details, including its ARN, creation date, and last activity. Below the summary, there are tabs for Permissions, Trust relationships, Tags, Access Advisor, and Revoke sessions. The Permissions tab is active, showing a list of attached policies, including `AmazonS3ReadOnlyAccess`.

**Identity and Access Management (IAM)**

Search IAM

Dashboard

▼ Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

▼ Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies

**role\_rr3493** [Info](#) [Delete](#)

Allows Redshift clusters to call AWS services on your behalf.

**Summary** [Edit](#)

Creation date	ARN
April 21, 2024, 20:14 (UTC-07:00)	<a href="#">arn:aws:iam::891376914165:role/role_rr3493</a>
Last activity	Maximum session duration
-	1 hour

**Permissions** | Trust relationships | Tags | Access Advisor | Revoke sessions

**Permissions policies (1)** [Info](#) [Refresh](#) [Simulate](#) [Remove](#) [Add permissions](#)

You can attach up to 10 managed policies.

Search  Filter by Type [All types](#)

<input type="checkbox"/>	<a href="#">Policy name</a>	Type	Attached entities
<input type="checkbox"/>	<a href="#">AmazonS3ReadOnlyAccess</a>	AWS managed	1

► [Permissions boundary \(not set\)](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Creating Cluster:

Amazon Redshift | us-east-1

us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#create-cluster

Amazon Redshift > Clusters > Create cluster

Create cluster [info](#)

Looking for free trial? Try Redshift Serverless. First-time Redshift Serverless customers receive a \$300 credit to use in their account. [Launch Redshift Serverless](#)

**Cluster configuration**

**Cluster identifier**  
This is the unique key that identifies a cluster.  
  
The identifier must be from 1-63 characters. Valid characters are a-z (lowercase only) and - (hyphen).

**Choose the size of the cluster**  
☒ I'll choose  
☐ Help me choose

**Node type** [info](#)  
Choose a node type that meets your CPU, RAM, storage capacity, and drive type requirements.

**Number of nodes**  
Enter the number of nodes that you need.  
  
Range (1-32)

**Configuration summary** [info](#)  
dc2.large | 2 nodes

**\$365.00/month**  
Estimated on-demand compute price  
Save more than 60% of your costs by purchasing reserved nodes. [Learn more about pricing](#)

**320 GB**  
Total compressed storage  
The total storage capacity for the cluster if you deploy the number of nodes that you chose.

## Selecting checkbox to load the Tickit data

Chrome File Edit View History Bookmarks Profiles Tab Window Help

us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#create-cluster

Amazon Redshift > Clusters > Create cluster

Create cluster [info](#)

**Sample data** [info](#)

☒ Load sample data  
Load sample data to your Redshift cluster to start using the query editor to query data.

**Tickit (28 MB)**  
Tickit is the sample data set that uses a sample database called TICKET. Tickit contains individual sample data files: two fact tables and five dimensions.

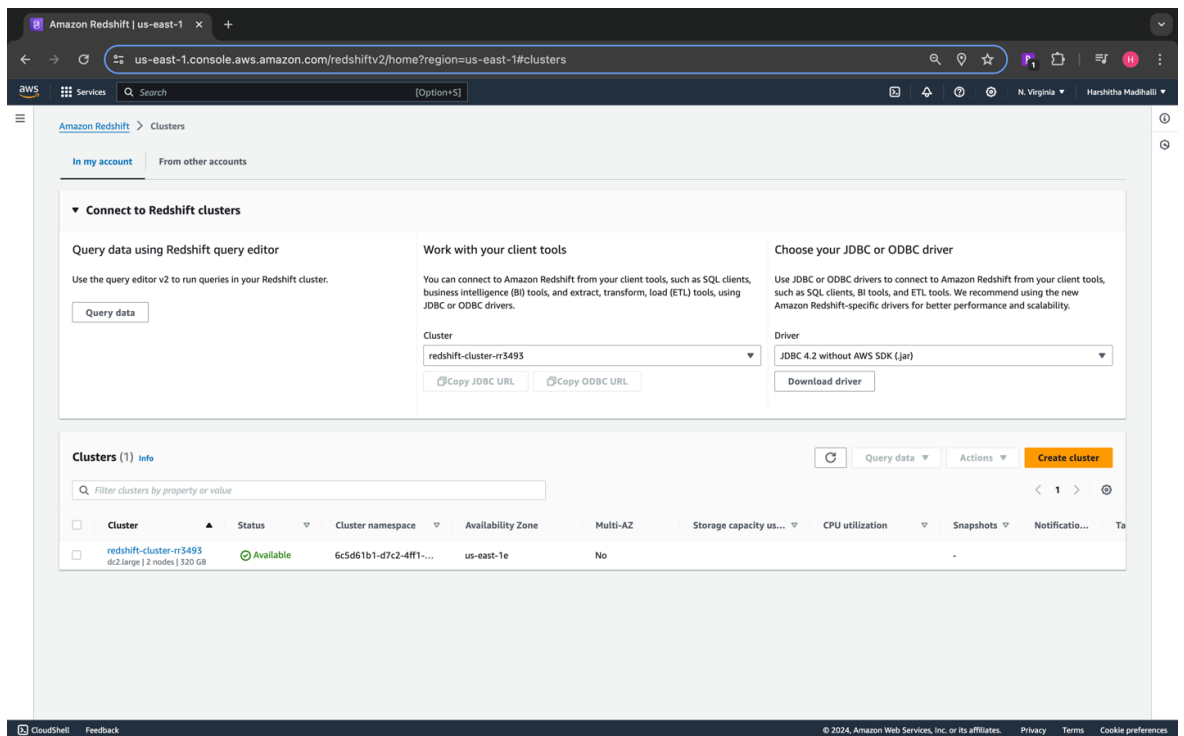
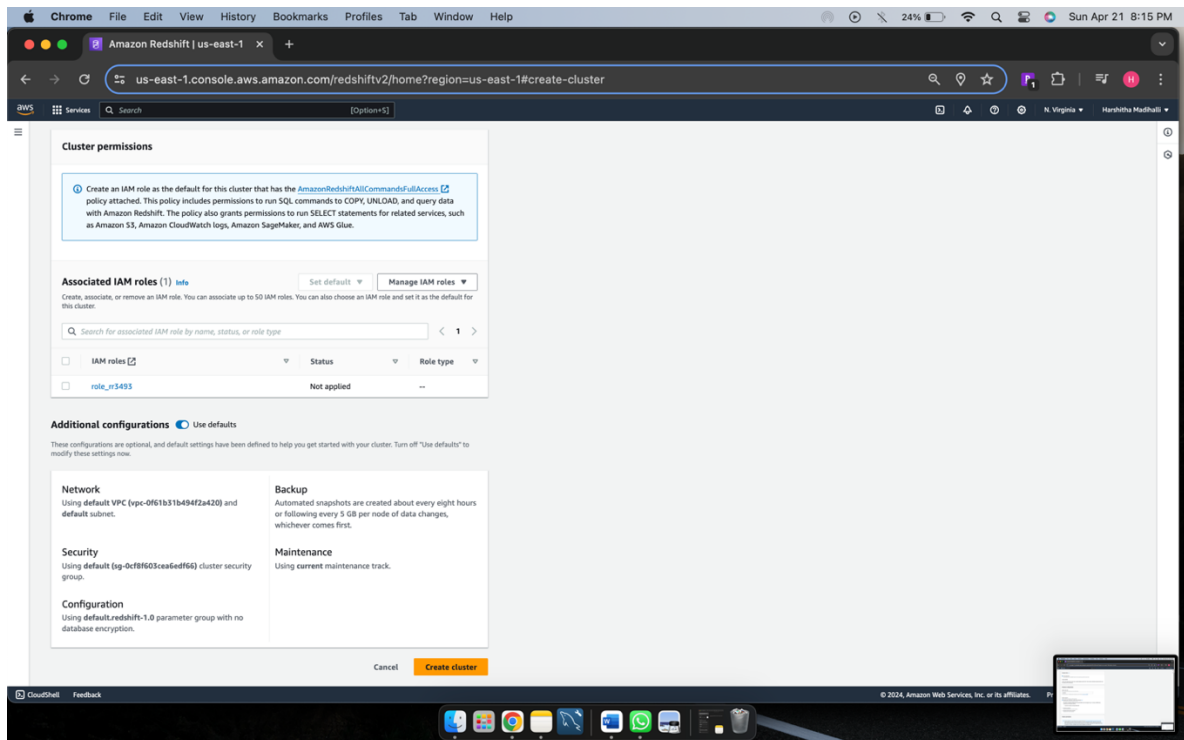
**Database configurations**

**Admin user name**  
Enter a login ID for the admin user of your DB instance.  
  
The name must be 1-128 alphanumeric characters, and it can't be a [reserved word](#).

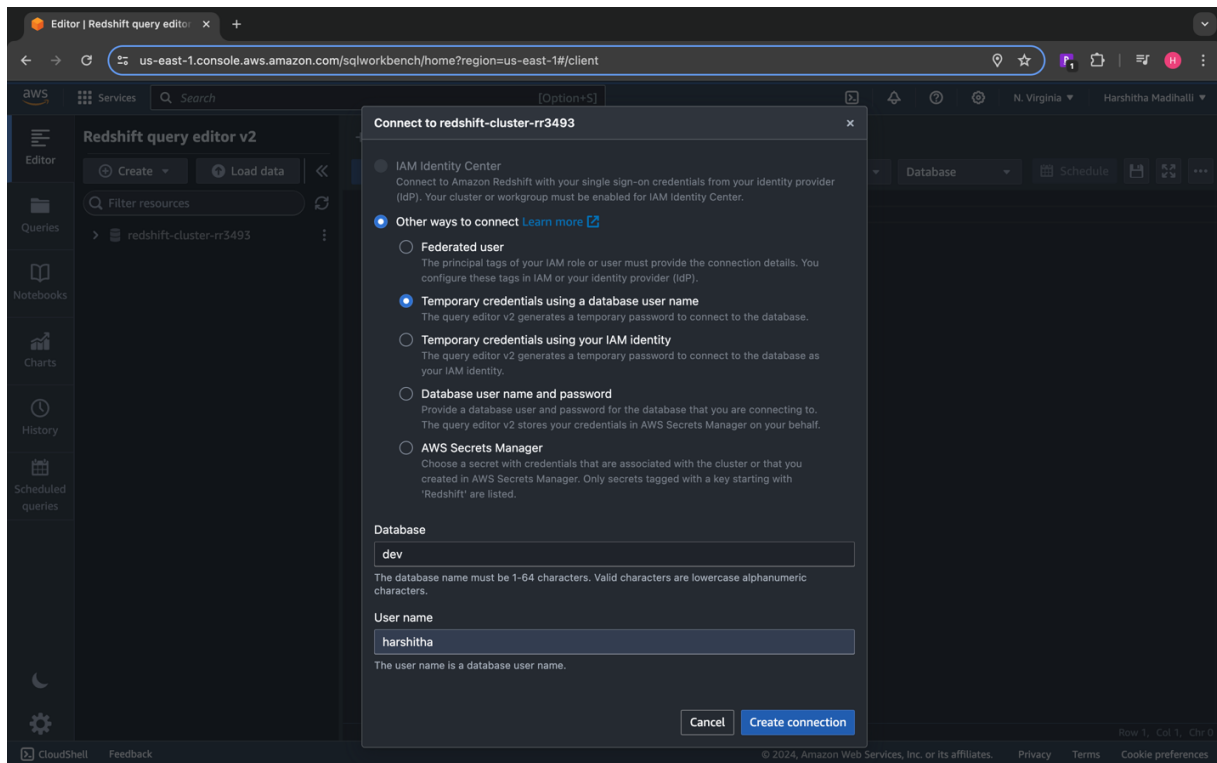
**Admin password**  
Select an option to manage your admin password.  
☒ Manage admin credentials in AWS Secrets Manager [info](#)  
AWS manages a CMK key that encrypts your data.  
Your data is encrypted by default with a key that AWS owns and manages for you. To choose a different key, customize your encryption settings.  
☐ Customize encryption settings (advanced)  
☐ Generate a password  
Amazon Redshift generates an admin password.  
☐ Manually add the admin password  
Manually enter the admin password.

**Cluster permissions**

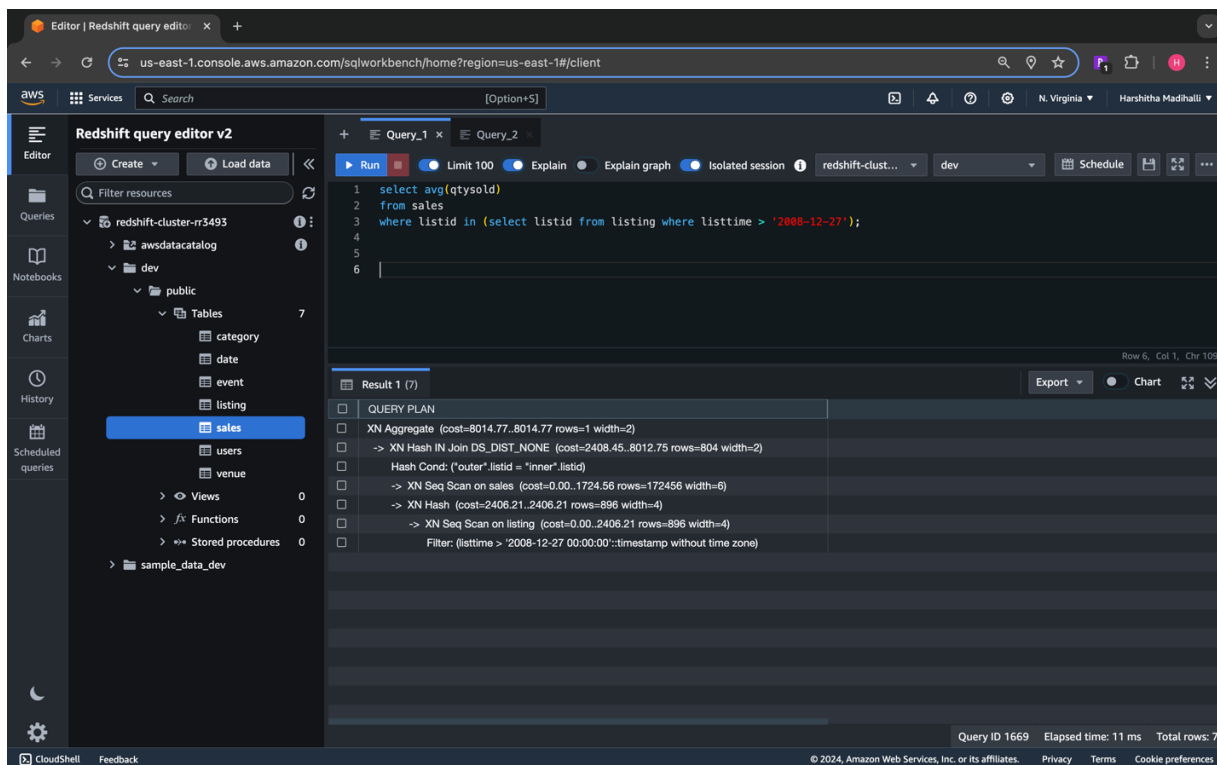
Create an IAM role as the default for this cluster that has the [AmazonRedshiftKDCCommandsFullAccess](#) policy attached. This policy includes permissions to run SQL commands to COPY, UNLOAD, and query data with Amazon Redshift. The policy also grants permissions to run SELECT statements for related services, such as Amazon S3, Amazon CloudWatch logs, Amazon SageMaker, and AWS Glue.



## Connecting to Database:



## Query 1:



## Query 2:

The screenshot displays the AWS Redshift Query Editor v2 interface. The left sidebar shows the 'Tables' section with 'sales' selected. The main editor area contains the following SQL query:

```
1 select avg(qtysold)
2 from sales
3 where saletime > '2008-12-27'
4 and listid in (select listid from listing where listtime > '2008-12-27');
5
```

Below the query, the 'Result 1 (8)' section shows the query plan. The plan details the execution steps and their costs:

- QUERY PLAN
- XN Aggregate (cost=4588.76..4588.76 rows=1 width=2)
- > XN Hash IN Join DS\_DIST\_NONE (cost=2408.45..4588.74 rows=6 width=2)
- Hash Cond: ("outer".listid = "inner".listid)
- > XN Seq Scan on sales (cost=0.00..2155.70 rows=1090 width=6)
- Filter: (saletime > '2008-12-27 00:00:00':timestamp without time zone)
- > XN Hash (cost=2408.21..2408.21 rows=896 width=4)
- > XN Seq Scan on listing (cost=0.00..2408.21 rows=896 width=4)
- Filter: (listtime > '2008-12-27 00:00:00':timestamp without time zone)

At the bottom right, the status bar indicates: Query ID 1682, Elapsed time: 65 ms, Total rows: 8.

- Two queries give the same output.
- Query 2 costs around 4588 and Query 1 around 80144, hence Query 2 is more cost efficient than Query 1.

# Part 2:

## Creating a project

Inbox - hm@horizon.csueas...x

New Project - horizon.csueas...x

Dashboardx

Homepagex

+

console.cloud.google.com/projectcreate?previousPage=%2Fwelcome%2Fnew%3ForganizationId%3D563623475875&organizationId=563623475875

☆P3|📄|📥|🔔|⋮

Google Cloud

Search (/) for resources, docs, products, and more

🔍 Search

✦📄1?⋮H

New Project

⚠️

You have 7 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \*

My Project-rr3493

?

Project ID: my-project-rr3493. It cannot be changed later. [EDIT](#)

Organization \*

horizon.csueastbay.edu

▼?

Select an organization to attach it to a project. This selection can't be changed later.

Location \*

horizon.csueastbay.edu

BROWSE

Parent organization or folder

CREATE

CANCEL

## Creating a dataset

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar shows the Google Cloud logo, the project name 'My Project-rr3493', and a search bar. The left sidebar contains a menu with categories like Analysis, Migration, and Administration. The main content area is divided into three sections: Explorer, Welcome to BigQuery Studio, and Job history. The Explorer section shows a tree view of resources for the project 'my-project-rr3493', including Queries, Notebooks, Data canvases, and External connections. The 'Welcome to BigQuery Studio' section features a 'Get started' area with buttons for 'CREATE SQL QUERY', 'CREATE PYTHON NOTEBOOK', and 'CREATE DATA CANVAS'. Below this, there are two demo queries: 'Try the Google Trends Demo Query' and 'Try the Colab Demo Notebook'. The 'Job history' section is visible at the bottom right.

The 'Create dataset' dialog box is open on the right side of the screen. It contains the following fields and options:

- Project ID \***: my-project-rr3493 (with a 'CHANGE' link)
- Dataset ID \***: rr3493\_ticket (with a note: 'Letters, numbers, and underscores allowed')
- Location type**:
  - ☐ Region: Specify a region to colocate your datasets with other Google Cloud services.
  - ☒ Multi-region: Allow BigQuery to select a region within a group to achieve higher quota limits.
- Multi-region \***: US (multiple regions in United States) (dropdown menu)
- Default table expiration**:
  - ☐ Enable table expiration
  - Default maximum table age: [text input] Days
- Advanced options**: (collapsed section)
- Buttons**: CREATE DATASET, CANCEL

## Creating Listing Table

**Create table**

**Source**

Create table from  
Upload

Select file \*  
listings\_csv-1.csv X [BROWSE](#) ?

File format  
CSV

**Destination**

Project \*  
my-project-rr3493 [BROWSE](#)

Dataset \*  
rr3493\_tickit

Table \*  
listing

Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, and spaces are allowed.

Table type  
Native table

**Schema**

☒ Auto detect

! Schema will be automatically generated.

[CREATE TABLE](#) [CANCEL](#)

## Creating Sales Table

**Create table**

**Source**

Create table from  
Upload

Select file \*  
sales\_csv-1.csv X [BROWSE](#) ?

File format  
CSV

**Destination**

Project \*  
my-project-rr3493 [BROWSE](#)

Dataset \*  
rr3493\_tickit

Table \*  
sales

Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, and spaces are allowed.

Table type  
Native table

**Schema**

☒ Auto detect

! Schema will be automatically generated.

[CREATE TABLE](#) [CANCEL](#)



## Listing Table:

The screenshot displays the Google Cloud BigQuery console interface. The left sidebar contains navigation menus for Analysis, Migration, and Administration. The central Explorer pane shows the project hierarchy, with the 'listing' table selected under the 'rr3493\_tickit' dataset. The right pane shows the 'listing' table's schema, including fields like listid, sellerid, eventid, dateid, numtickets, priceperticket, totalprice, and listtime. Below the schema, there are buttons for 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'. The bottom section shows the 'Job history' with a 'REFRESH' button.

**Listing Table Schema:**

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
listid	INTEGER	NULLABLE	-	-	-	-	-
sellerid	INTEGER	NULLABLE	-	-	-	-	-
eventid	INTEGER	NULLABLE	-	-	-	-	-
dateid	INTEGER	NULLABLE	-	-	-	-	-
numtickets	INTEGER	NULLABLE	-	-	-	-	-
priceperticket	INTEGER	NULLABLE	-	-	-	-	-
totalprice	INTEGER	NULLABLE	-	-	-	-	-
listtime	TIMESTAMP	NULLABLE	-	-	-	-	-

## Sales Table:

The screenshot displays the Google Cloud BigQuery console interface, similar to the Listing Table view. The central Explorer pane shows the 'sales' table selected under the 'rr3493\_tickit' dataset. The right pane shows the 'sales' table's schema, including fields like salesid, listid, sellerid, buyerid, eventid, dateid, qtytsold, pricepalid, commission, and saletime. Below the schema, there are buttons for 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'. The bottom section shows the 'Job history' with a 'REFRESH' button.

**Sales Table Schema:**

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
salesid	INTEGER	NULLABLE	-	-	-	-	-
listid	INTEGER	NULLABLE	-	-	-	-	-
sellerid	INTEGER	NULLABLE	-	-	-	-	-
buyerid	INTEGER	NULLABLE	-	-	-	-	-
eventid	INTEGER	NULLABLE	-	-	-	-	-
dateid	INTEGER	NULLABLE	-	-	-	-	-
qtytsold	INTEGER	NULLABLE	-	-	-	-	-
pricepalid	INTEGER	NULLABLE	-	-	-	-	-
commission	FLOAT	NULLABLE	-	-	-	-	-
saletime	TIMESTAMP	NULLABLE	-	-	-	-	-

# Query 1:

Explorer

+ ADD

IK

Type to search

Viewing resources.

SHOW STARRED ONLY

my-project-rr3493

Queries

Notebooks

Data canvases

External connections

rr3493\_tickit

listing

sales

Untitled query

RUN

SAVE

DOWNLOAD

SHARE

SCHEDULE

MORE

Quer.

```
#query1
select sum(qtysold)
from my-project-rr3493.rr3493_tickit.sales as S1, my-project-rr3493.rr3493_tickit.listing as L1
where S1.listid = L1.listid
and L1.listtime > '2008-12-26';
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row

f0\_

46

SUMMARY

Nothing currently selected

## Details:

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Elapsed time

Slot time consumed

Bytes shuffled

Bytes spilled to disk

317 ms

295 ms

1.79 KB

0 B

SHOW AVERAGE TIME

SHOW MAXIMUM TIME

Stages

Working timing

Rows

S00: Input

Wait:

Read:

Compute:

Write:

1 ms

32 ms

26 ms

10 ms

Records read: 192497

Records written: 202

S02: Join+

Wait:

Read:

Compute:

Write:

15 ms

10 ms

26 ms

1 ms

Records read: 172658

Records written: 1

S03: Output

Wait:

Read:

Compute:

Write:

11 ms

0 ms

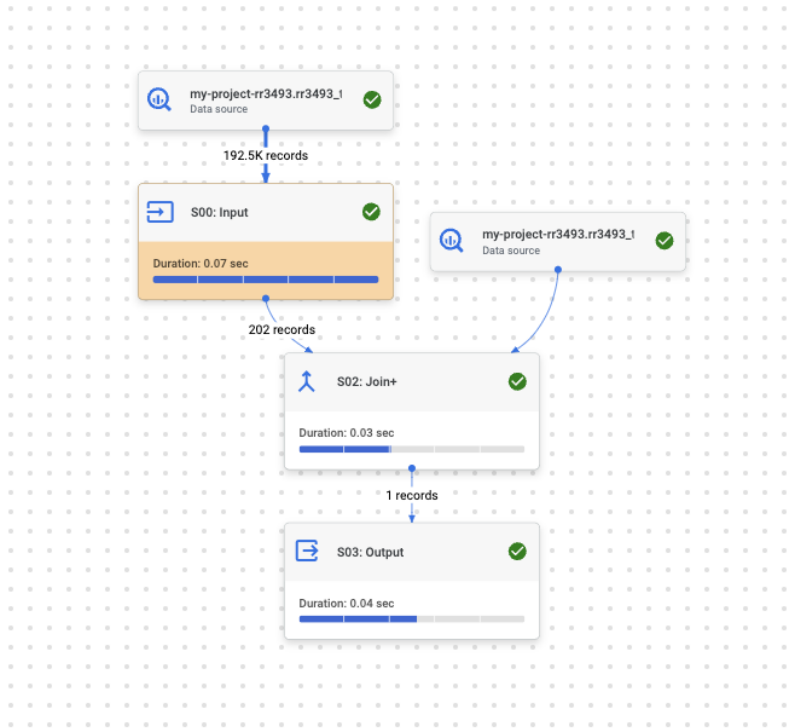
6 ms

3 ms

Records read: 1

Records written: 1

## Graph:



## Query 2:

Untitled query

Run Save Download Share Schedule More

Query completed.

```
#query2
select sum(qtysold)
from my-project-rr3493.rr3493_tickit.sales as S1
where S1.saletime > '2008-12-25'
and S1.listid in (select listid from my-project-rr3493.rr3493_tickit.listing as L1
where L1.listtime > '2008-12-26');
7
```

Query results

Save Results Explore Data

Job Information Results Chart JSON Execution Details Execution Graph

Row	f0_
1	12

## Details:

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Elapsed time

Slot time consumed

Bytes shuffled

Bytes spilled to disk

304 ms

174 ms

1.79 KB

0 B

SHOW AVERAGE TIME

SHOW MAXIMUM TIME

Stages

Working timing

Rows

▶ S00: Input

Wait:

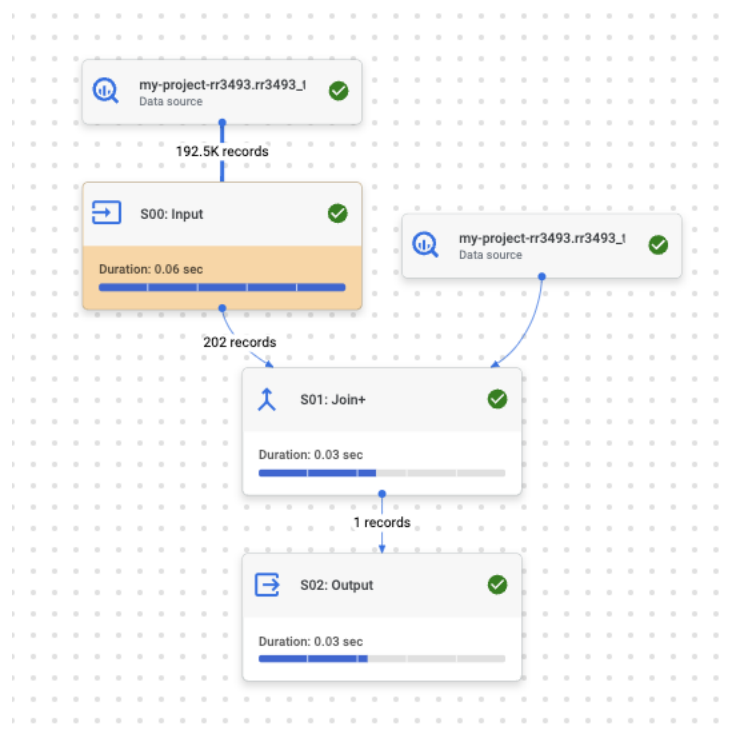
Read:

Compute:

Write:

</

## Graph:



- Both queries have different approach which give the same output.
- Query 2 is more efficient than Query 1 in terms of their elapsed time and slot time consumption.