

Load Data from S3

To begin with the Lab

Summary of the Lab

In this lab, you learn how to **load data from Amazon S3 into Amazon Redshift**. First, an S3 bucket is created, and a CSV file (e.g., orders.csv) is uploaded. Then, a target table is created in Redshift using the **CREATE TABLE** command, and data is imported using the **COPY** command, specifying the S3 path, IAM role, region, and delimiter options. Alternatively, Redshift's graphical **Load Data wizard** can be used to import data without SQL by selecting the S3 bucket, configuring file settings, and assigning roles. The data successfully loads into the Redshift table.

- Setting Up an S3 Bucket
- A new S3 bucket is created to store the data that will be loaded into Redshift.
- Default settings are used for simplicity.
- Public access restrictions are disabled for this demonstration.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

- After creation, the bucket appears under the specified region (e.g., US East 1).
- A CSV file (such as orders.csv) is uploaded to the bucket.

redshift-order-table [Info](#)

[Objects](#) [Metadata](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (1) [Actions](#) [Create folder](#) [Upload](#)

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](#)

Name	Type	Last modified	Size	Storage class
List+of+Orders.csv	csv	November 13, 2025, 22:18:28 (UTC+05:30)	23.1 KB	Standard

- Loading Data into Redshift
- Data loading is performed using the **COPY** command, a standard SQL command in data warehouses.
- The **COPY** command copies data from S3 into a Redshift table.

- Before executing the COPY command, a target table must exist in the database.
- The command structure includes the table name, S3 file path, IAM role, region, delimiter, and header options.

```

1  COPY public.orders
2  FROM 's3://redshift-order-table>List+of+Orders.csv'
3  IAM_ROLE 'arn:aws:iam::463646775279:role/service-role/AmazonRedshift-CommandsAccessRole-20251113T225531'
4  DELIMITER ','
5  IGNOREHEADER 1
6  REGION 'us-east-1';

```

```

COPY public.orders
FROM 's3://redshift-order-table>List+of+Orders.csv'
IAM_ROLE 'arn:aws:iam::463646775279:role/service-role/AmazonRedshift-
CommandsAccessRole-20251113T225531'
DELIMITER ','
IGNOREHEADER 1
REGION 'us-east-1';

```

- Creating the Target Table
- A table must be created before using the COPY command.
- The CREATE TABLE command is used to define the table structure, specifying columns and data types.
- The table is created under the default “public” schema if no other schema is specified.
- The new table can be verified in the Redshift Query Editor’s schema view.

```

8  CREATE TABLE orders (
9    order_id VARCHAR(20),
10   order_date VARCHAR(20),
11   customer_name VARCHAR(100),
12   state VARCHAR(50),
13   city VARCHAR(50)
14 );

```

```

CREATE TABLE orders (
  order_id VARCHAR(20),
  order_date VARCHAR(20),
  customer_name VARCHAR(100),
  state VARCHAR(50),
  city VARCHAR(50)
);

```

- Here we created the table first and then we have updated the data from s3 to table.
- We used the above commands to create table and load the data from the s3 bucket

Result 1

Summary

Returned rows: 0
Elapsed time: 473ms
Result set query:

```
/* RQEY2-AIqkUHqcdZ */
CREATE TABLE orders (
    order_id VARCHAR(20),
    order_date VARCHAR(20),
    customer_name VARCHAR(100),
    state VARCHAR(50),
    city VARCHAR(50)
)
```

+ Untitled 2 x

Result 1

Summary

Info:
• Load into table 'orders' completed, 560 record(s) loaded successfully.

Returned rows: 0
Query ID: 3400
Elapsed time: 17.6s
Result set query:

```
/* RQEY2-C62SgX0zvv */
COPY public.orders
FROM 's3://redshift-order-table>List+of+Orders.csv'
IAM_ROLE 'arn:aws:iam::463646775279:role/service-role/AmazonRedshift-CommandsAccessRole-20251113T225531'
DELIMITER ','
IGNOREHEADER 1
REGION 'us-east-1'
```

Redshift query editor v2

Untitled 2 x Untitled 1 x

Run Limit 100 Explain Isolated session redshift-clust... dev Schedule

Filter resources

redshift-cluster-1 native databases (2) dev public Tables category date event listing orders sales users venue Views Functions

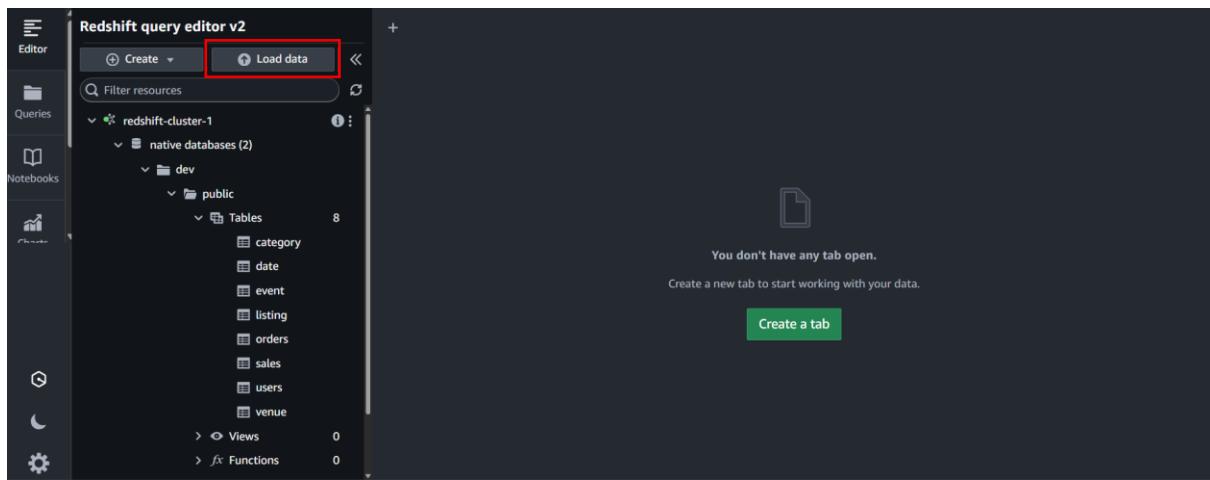
```
1 SELECT
2   *
3 FROM
4   "dev"."public"."orders";
```

Result 1 (100)

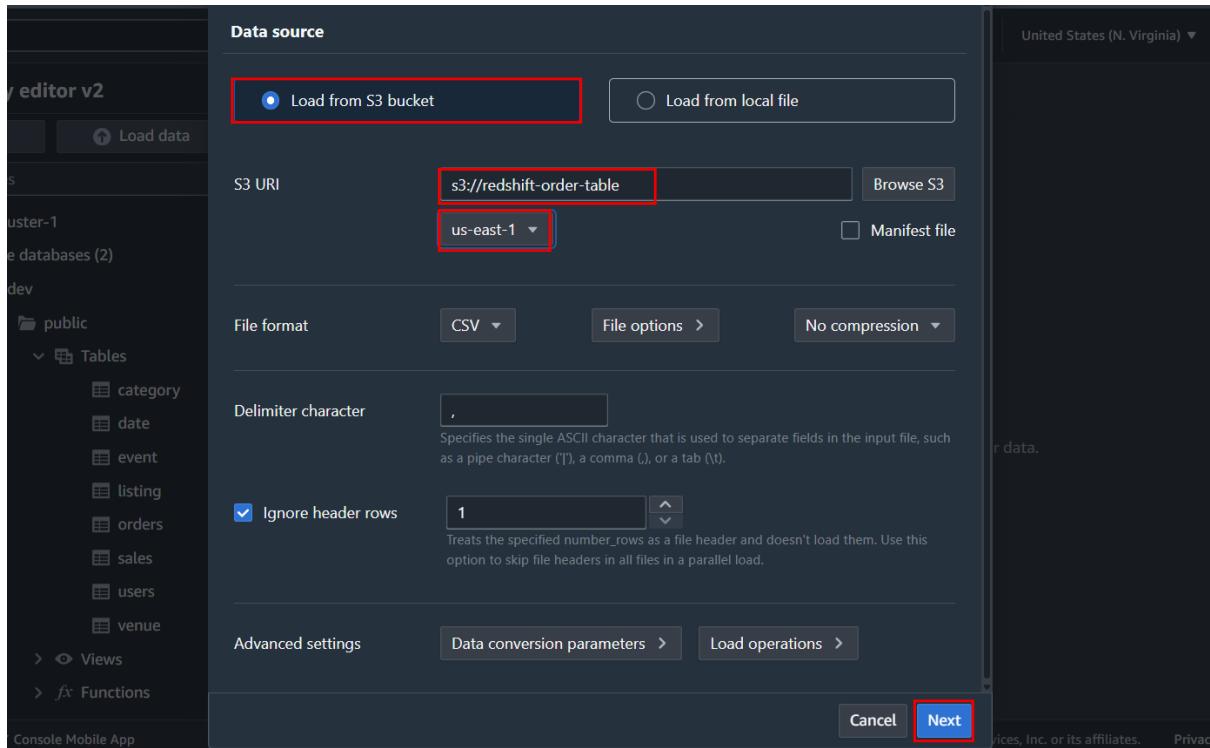
	order_id	order_date	customer_name	state	city
1	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad
2	B-25602	01-04-2018	Pearl	Maharashtra	Pune
3	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal
4	B-25604	03-04-2018	Divsha	Rajasthan	Jaipur
5	B-25605	05-04-2018	Kasheen	West Bengal	Kolkata
6	B-25606	06-04-2018	Hazel	Karnataka	Bangalore
7	B-25607	06-04-2018	Sonakshi	Jammu and Kashmir	Kashmir

Row 1, Col 1, Chr 49
Query ID 3449 Elapsed time: 4889 ms Total rows: 100

- Alternative Method – Using the Redshift Interface
- Redshift provides a graphical method to load data using a built-in wizard.
- The “Load Data” option in Query Editor allows users to import data without writing SQL.



- Selecting “Load data from S3.”
- Choosing the S3 bucket and region.
- Configuring file options like delimiter and headers.



- Selecting whether to load into an existing table or create a new one.
- Defining column names, data types, and constraints (e.g., primary key or not null).
- Selecting the IAM role for access permissions.
- After setup, Redshift automatically generates and executes the COPY command.

Load existing table
Load data into an existing table

Load new table
Create table with detected schema

Cluster or workgroup: redshift-cluster-1 Database: dev Schema: public Table: order2

IAM role: arn:aws:iam::463646775279:role/service-role/AmazonRedshift-CommandsAccessRole-20251113T225531

Columns Table details

Selected S3 file: List+of+Orders.csv + Add column

Column name	Data type	Encoding
order_id (P)	VARCHAR	No selection
order_date	VARCHAR	No selection
customer_name	VARCHAR	No selection
state	VARCHAR	No selection
city	VARCHAR	No selection

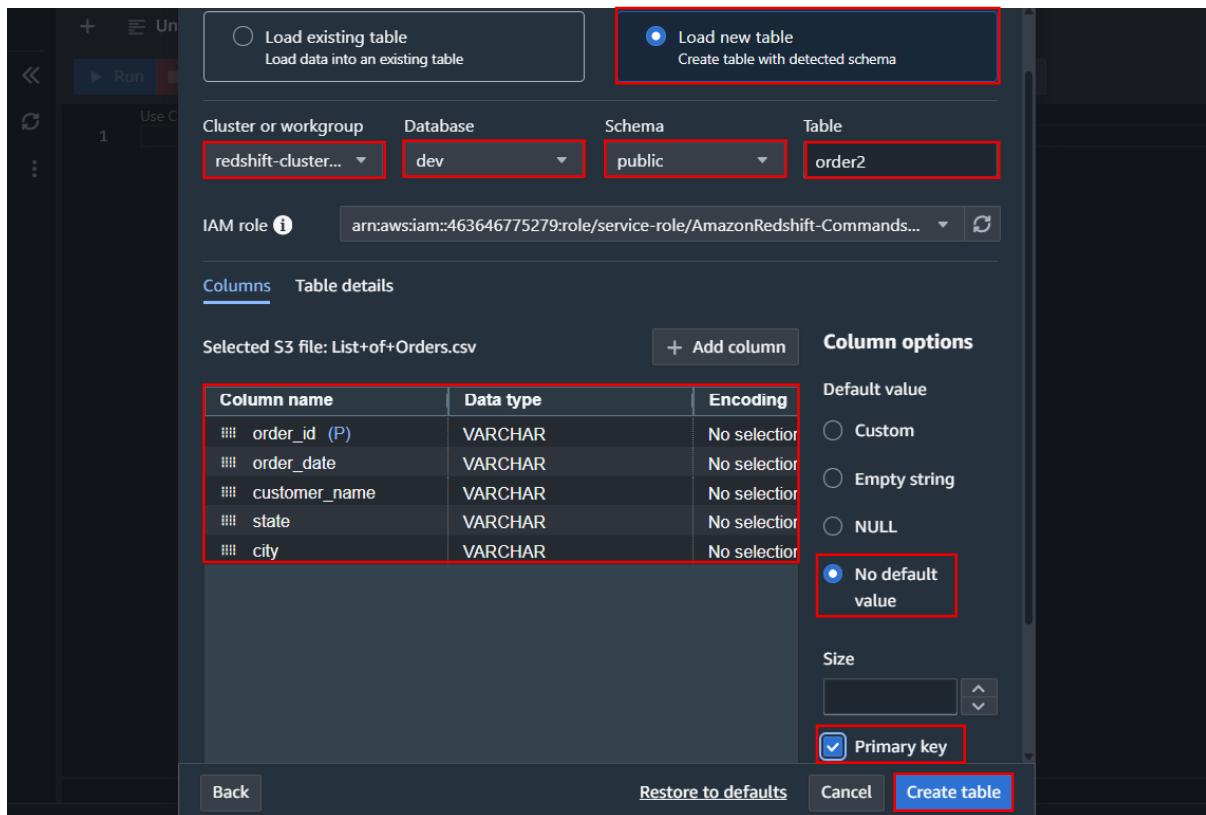
Column options

Default value: No default value
 Custom
 Empty string
 NULL

Size:

Primary key

Back Restore to defaults Cancel Create table



Load data

Review an order2 table is created successfully.

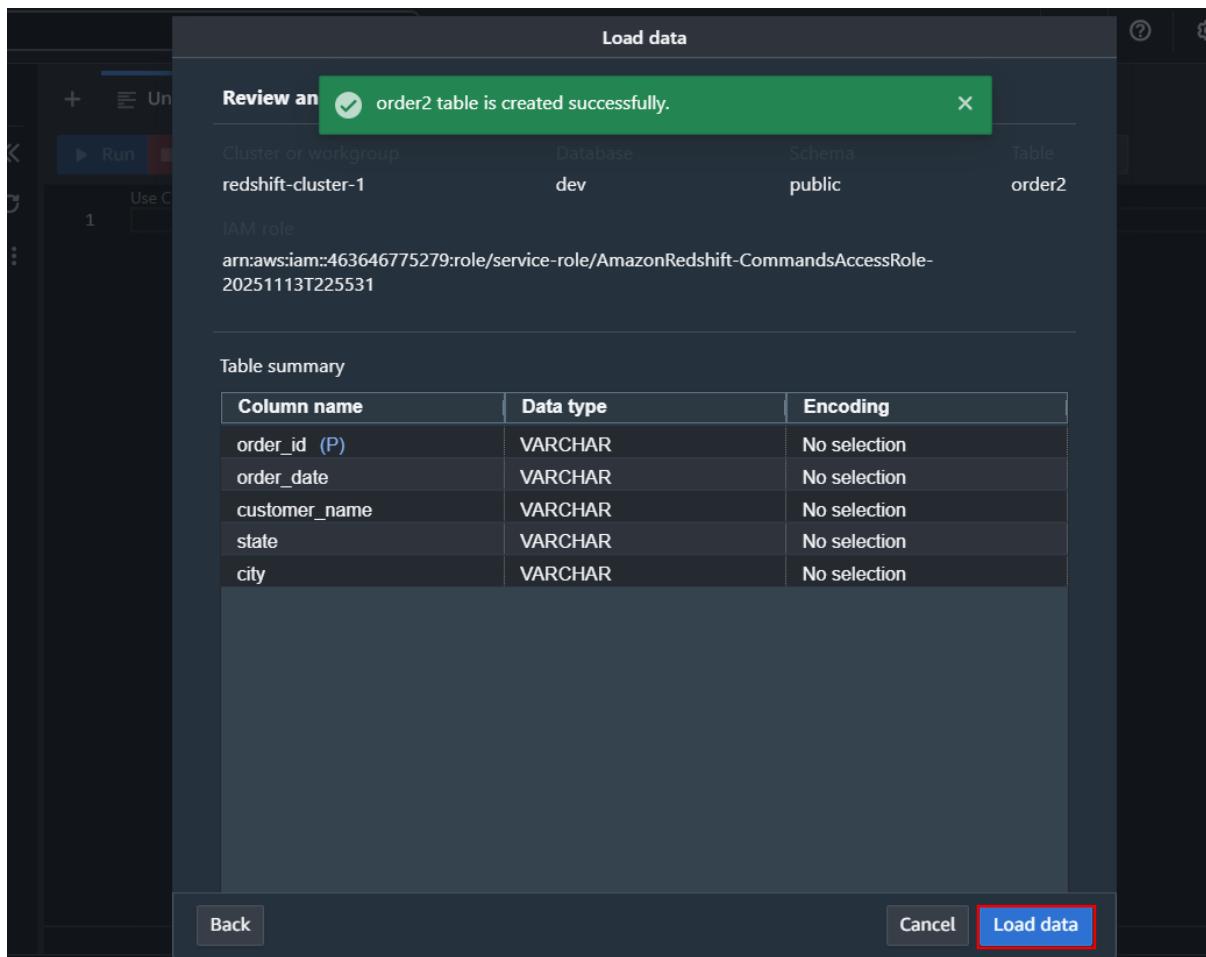
Cluster or workgroup: redshift-cluster-1 Database: dev Schema: public Table: order2

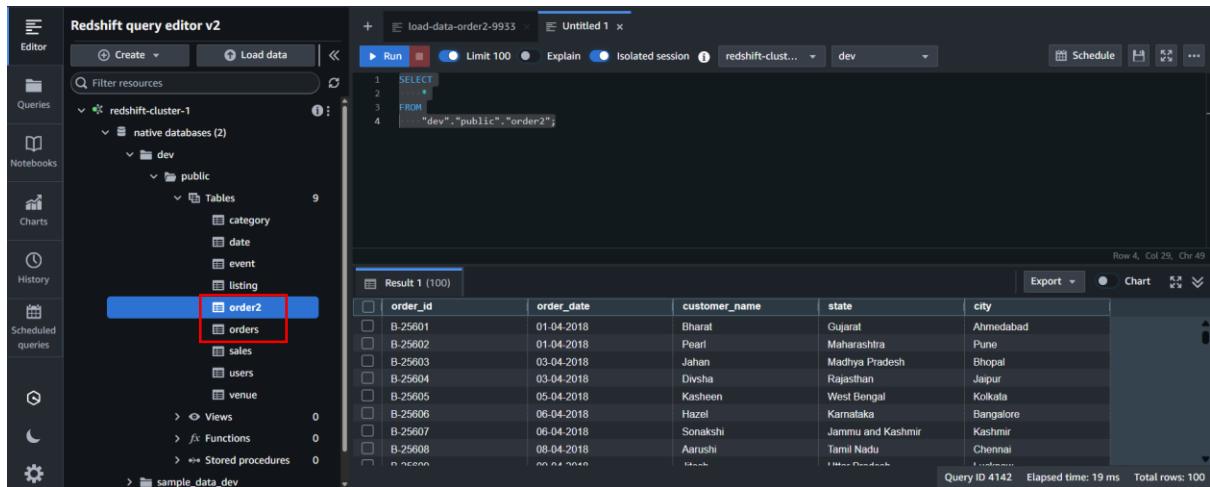
IAM role: arn:aws:iam::463646775279:role/service-role/AmazonRedshift-CommandsAccessRole-20251113T225531

Table summary

Column name	Data type	Encoding
order_id (P)	VARCHAR	No selection
order_date	VARCHAR	No selection
customer_name	VARCHAR	No selection
state	VARCHAR	No selection
city	VARCHAR	No selection

Back Cancel Load data





The screenshot shows the Redshift query editor interface. On the left, the sidebar displays the database structure under 'redshift-cluster-1' / 'dev' / 'public'. The 'order2' table is highlighted with a red box. The main panel shows a query editor with the following SQL code:

```
1 SELECT
2 *
3 FROM
4 "dev"."public"."order2";
```

The results pane displays a table titled 'Result 1 (100)' with 100 rows of data. The columns are: order_id, order_date, customer_name, state, and city. The data is as follows:

	order_id	order_date	customer_name	state	city
1	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad
2	B-25602	01-04-2018	Pearl	Maharashtra	Pune
3	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal
4	B-25604	03-04-2018	Divsha	Rajasthan	Japur
5	B-25605	05-04-2018	Kasheen	West Bengal	Kolkata
6	B-25606	06-04-2018	Hazel	Karnataka	Bangalore
7	B-25607	06-04-2018	Sonakshi	Jammu and Kashmir	Kashmir
8	B-25608	08-04-2018	Aarushi	Tamil Nadu	Chennai
99	B-25609	09-04-2018	Shank	Uttar Pradesh	Lucknow

At the bottom right of the results pane, it says 'Query ID 4142 Elapsed time: 19 ms Total rows: 100'.

- You can see the table order2 with the data uploaded from the s3 bucket.