

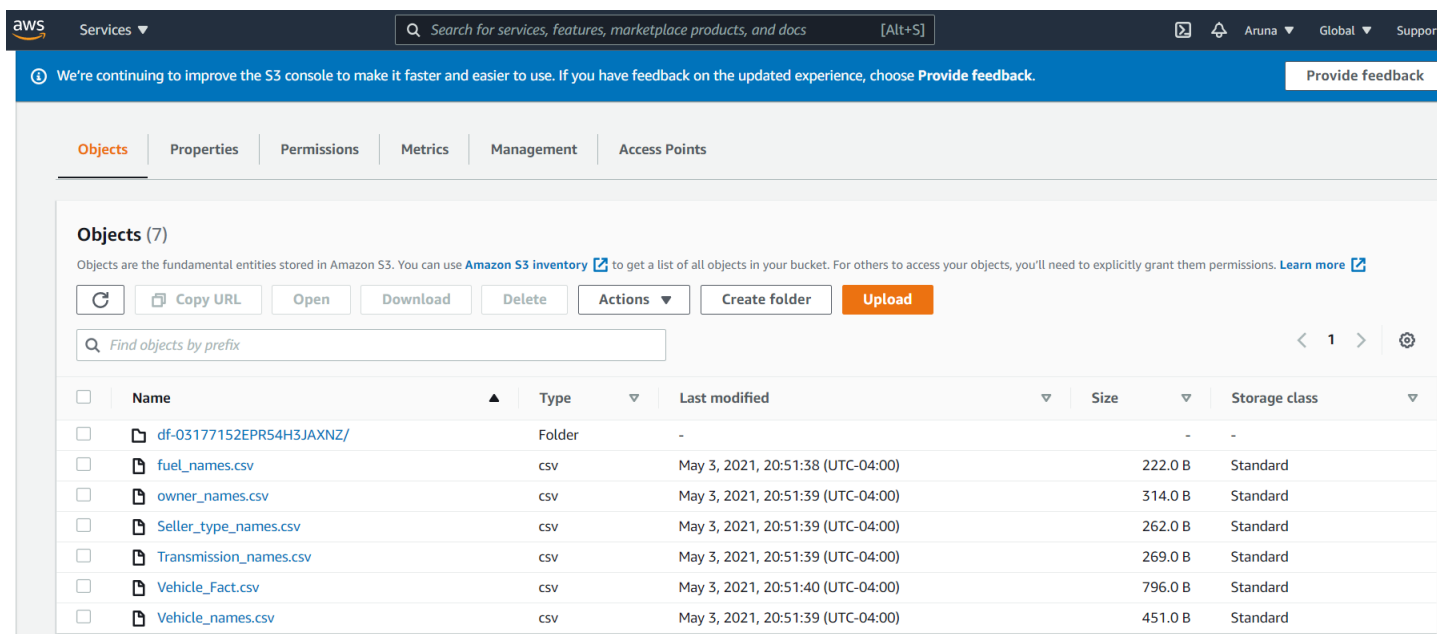
Datawarehouse Engineer take home

With the given problem statement, I have taken Vehicles dataset from Kaggle and decided to Use **Amazon S3** as **Storage layer** and **Amazon redshift** for **Datawarehouse layer** and Amazon Redshift can be connected to **Tableau** for **Reporting Purpose**.

And for scheduling this entire flow I used **Amazon Data Pipeline**.

I separated the data from Vehicles dataset (source : <https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho>) into Fact Table and Dimension Tables and uploaded to Amazon S3 as shown below.

The list of tables are as follows



	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	df-03177152EPR54H3JAXNZ/	Folder	-	-	-
<input type="checkbox"/>	fuel_names.csv	csv	May 3, 2021, 20:51:38 (UTC-04:00)	222.0 B	Standard
<input type="checkbox"/>	owner_names.csv	csv	May 3, 2021, 20:51:39 (UTC-04:00)	314.0 B	Standard
<input type="checkbox"/>	Seller_type_names.csv	csv	May 3, 2021, 20:51:39 (UTC-04:00)	262.0 B	Standard
<input type="checkbox"/>	Transmission_names.csv	csv	May 3, 2021, 20:51:39 (UTC-04:00)	269.0 B	Standard
<input type="checkbox"/>	Vehicle_Fact.csv	csv	May 3, 2021, 20:51:40 (UTC-04:00)	796.0 B	Standard
<input type="checkbox"/>	Vehicle_names.csv	csv	May 3, 2021, 20:51:39 (UTC-04:00)	451.0 B	Standard

From the Amazon S3 as the input layer, I created a cluster in Amazon Redshift **redshift-cluster-1** with Node type as dc2.large and Number of nodes as 1.

Amazon Redshift > Clusters > redshift-cluster-1

redshift-cluster-1

Actions Edit Add partner integration Query cluster

General information

Cluster identifier redshift-cluster-1	Status Available	Node type dc2.large	Endpoint redshift-cluster-1.catask0zlrk.us-east-1.reds...
Cluster namespace b0b4b333-7344-4a68-b3dd-682f32e74dbc	Date created May 03, 2021, 10:29(UTC-04:00)	Number of nodes 1	JDBC URL jdbc:redshift://redshift-cluster-1.catask0zlrk....
	Storage used 0.05% (0.08 of 160 GB used)	AQUA Not available	ODBC URL Driver={Amazon Redshift (x64)}; Server=reds...

Cluster performance Query monitoring Schedules Maintenance Properties

Recommendations (0)
To improve performance and decrease operating costs, recommendations are provided by the Amazon Redshift Advisor.

Created a IAM Cluster role Redshift_Role_S3 which contains AmazonS3ReadOnlyAccess

console.aws.amazon.com/iam/home?region=us-east-1#/roles/Redshift_Role_S3

Summary

Role ARN: arn:aws:iam::640386721955:role/Redshift_Role_S3

Role description: Allows Redshift clusters to call AWS services on your behalf. | Edit

Instance Profile ARNs: /

Path: /

Creation time: 2021-05-03 10:21 EDT

Last activity: 2021-05-03 20:59 EDT (Today)

Maximum session duration: 1 hour | Edit

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (1 policy applied)

Attach policies Add inline policy

Policy name	Policy type
AmazonS3ReadOnlyAccess	AWS managed policy

Permissions boundary (not set)

Generate policy based on CloudTrail events

And I associated this IAM role to the cluster I created.

After creating the cluster in Redshift, I started creating Tables in Redshift with the same structure of the files that I loaded in S3 and I loaded data into tables using the copy command.

While loading the data to tables using copy command , I gave the input path as S3 Bucket and also given Aws_Iam_Role and ignored the header

Tables that I created: one Fact Table and 5 Dimension Tables:

```
create table if not exists vehicle_fact(Vehicle_id nvarchar(100),
fuel_id nvarchar(20),
seller_type_id nvarchar(20),
transmission_id nvarchar(20),
owner_id nvarchar(50),
year nvarchar(10),
selling_price decimal(20,2),
km_driven decimal(20,2));
```

=====

```
copy vehicle_fact from 's3://redshiftfilestorage/Vehicle_Fact.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

=====

```
create table if not exists Vehicle_names(Vehicle_id nvarchar(10),
Vehicle_name nvarchar(200));
```

=====

```
copy Vehicle_names from 's3://redshiftfilestorage/Vehicle_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

=====

```
create table if not exists Transmission_names(transmission_id
nvarchar(10),
transmission_name nvarchar(200));
```

=====

```
copy Transmission_names from 's3://redshiftfilestorage/Transmission_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

=====

```
create table if not exists Seller_type_names(seller_type_id nvarchar(10),
seller_type_names nvarchar(200));
```

=====

```
copy Seller_type_names from 's3://redshiftfilestorage/Seller_type_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

=====

```
create table if not exists owner_names(owner_id nvarchar(10),
owner_name nvarchar(200));
```

=====

```
copy owner_names from 's3://redshiftfilestorage/owner_names.csv'
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
delimiter ',' region 'us-east-1'
IGNOREHEADER 1;
```

=====

```
create table if not exists fuel_names(fuel_id nvarchar(10),
```

```
fuel_names nvarchar(200));
```

```
=====
```

```
copy fuel_names from 's3://redshiftfilestorage/fuel_names.csv'
```

```
credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
```

```
delimiter ',' region 'us-east-1'
```

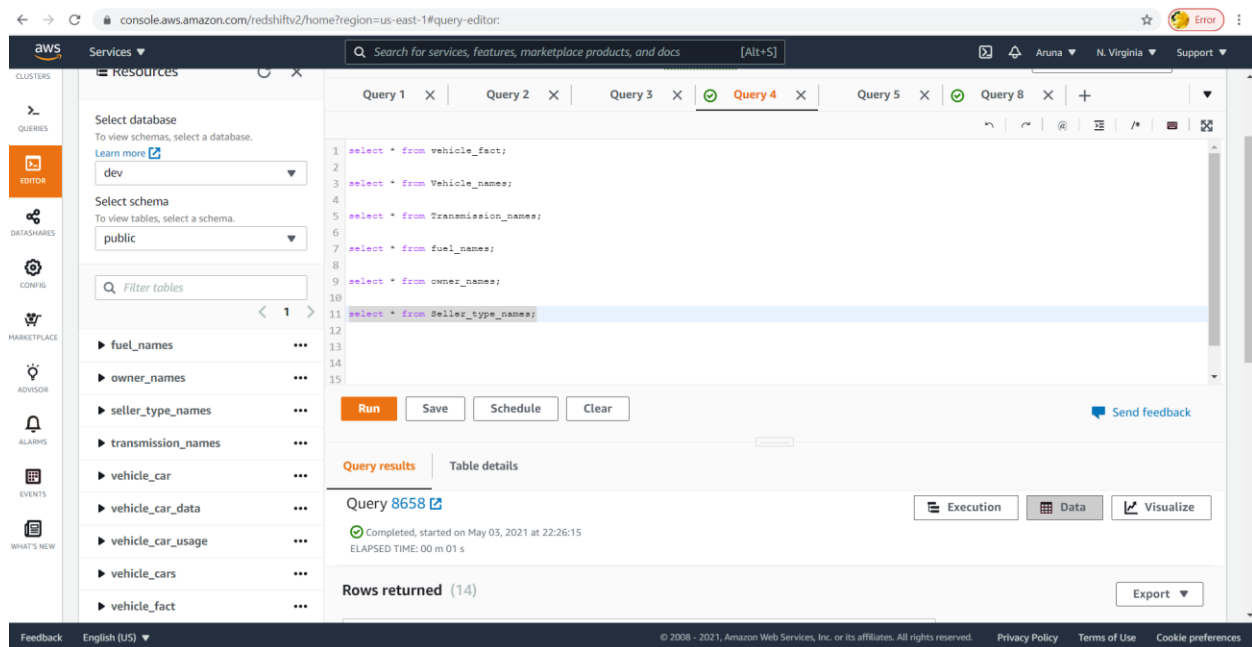
```
IGNOREHEADER 1;
```

```
=====
```

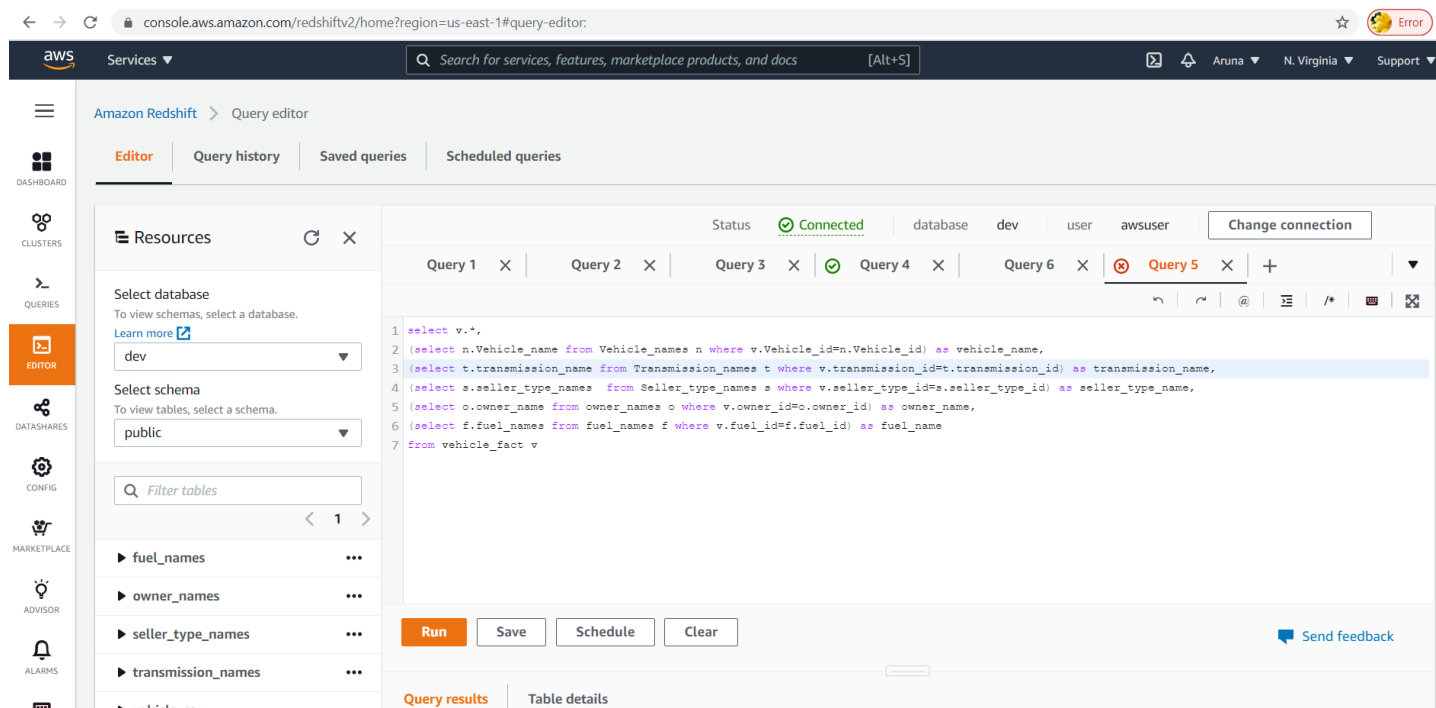
The screenshot displays the AWS Redshift console interface. On the left sidebar, the 'EDITION' tab is selected. The main panel is divided into two sections: 'Resources' on the left and a SQL query editor on the right. The 'Resources' section shows a list of tables under the 'public' schema, including 'fuel_names', 'owner_names', 'seller_type_names', 'transmission_names', 'vehicle_car', 'vehicle_car_data', and 'vehicle_car_usage'. The SQL query editor contains the following code:

```
1 create table if not exists Seller_type_names(seller_type_id nvarchar(10),
2 seller_type_names nvarchar(200));
3
4
5 copy Seller_type_names from 's3://redshiftfilestorage/Seller_type_names.csv'
6 credentials 'aws_iam_role=arn:aws:iam::640386721955:role/Redshift_Role_S3'
7 delimiter ',' region 'us-east-1'
8 IGNOREHEADER 1;
9
10
11
12 create table if not exists owner_names(owner_id nvarchar(10),
13 owner_name nvarchar(200));
14
15
```

Below the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Query results' tab is selected, showing the query execution status: 'Completed, started on May 03, 2021 at 22:26:43' and 'ELAPSED TIME: 02 m 13 s'. The 'Table details' tab is also visible.



Using the Star Schema concept, I joined the fact table to the dimension tables to get the dimension names.



Now, this data after Dimensional Modelling is available for reporting.


Data Pipeline scheduling:

=====

I scheduled this entire data pipeline that is loading the data from Amazon s3 to Amazon redshift using the copy statement daily using Amazon Data Pipeline.




I created a pipeline called s3toredshiftcopy which takes the data from S3 Bucket and load into Redshift tables everyday.

Create Pipeline

 You can create pipeline using a template or build one using the Architect page.

Name	<input type="text" value="s3toredshiftcopy"/>
Description (optional)	<input type="text" value="automating copy statement from input s3 folder to redshift table"/>
Source	<div><input checked="" type="radio"/> Build using a template <div><div>Load data from S3 into Redshift</div><div>▼</div></div><div><input type="radio"/> Import a definition <input type="radio"/> Build using Architect</div></div>

Parameters

Redshift password	<input type="password" value="....."/>
Redshift security group(s)	<input type="text" value="default"/> 
Redshift database name	<input type="text" value="dev"/>
Redshift username	<input type="text" value="awsuser"/>
Create table SQL query (optional)	<input type="text" value="transmission varchar(50),
owner varchar(50);"/> 
Table insert mode	<div><div>OVERWRITE_EXISTING</div><div>▼ </div></div>

Redshift table name

Input S3 folder

Redshift JDBC connection string

Copy options (optional)

? +

x +

x +

Primary keys (optional) ? +

Schedule

i You can run your pipeline once or specify a schedule. [More](#)

Run ☒ on pipeline activation
☐ on a schedule

Pipeline Configuration

Logging ☒ Enabled
☐ Disabled

Copy execution logs to S3. [More](#)

S3 location for logs

Pipeline Configuration

Logging ☒ Enabled
☐ Disabled

Copy execution logs to S3. [More](#)

S3 location for logs

Security/Access

IAM roles ☒ Default
☐ Custom

IAM Roles let you control permissions for AWS Data Pipeline and your EC2 applications. [More](#)

Tags

i Add up to 10 tags to your pipeline. These tags will be applied to the pipeline as well as any resources created by the pipeline. A tag consists of a case-sensitive key-value pair. [Learn more](#)

Key	Value (Optional)
<input type="text" value="Add key to create"/>	<input type="text"/>

This pipeline launches an Amazon EC2 instance (t1.micro) in your account on every scheduled execution of the pipeline. [Normal service charges](#) for this resource will apply in addition to charges for other AWS services used by this pipeline.

[Cancel](#) [Edit in Architect](#) [Activate](#)

How would you rate your experience with this service console? ☆ ☆ ☆ ☆ ☆

aws Services [Alt+S] Aruna N. Virginia Support

Data Pipeline > Execution Details: s3toredshiftcopy (df-03177152EPR54H3JAXNZ) [Data Pipeline Help](#)

[Edit Pipeline](#) [Rerun](#) [Cancel](#) [Mark Finished](#)

Show components in state with between UTC and UTC [Apply](#)

Filter: 1 instances (all loaded)

Component Name	Schedule Interval (UTC)	Type	Status	Execution Start (UTC)	Execution End (UTC)	Attempt
<input checked="" type="checkbox"/> ▶ RedshiftLoadActivity	2021-05-03 16:58:22 - 2021-05-03 16:58:22	RedshiftCopyActivity	WAITING_FOR_RUNNER	2021-05-03 16:58:25	-	1 of 3