### **INTEGRATION OBJECT**

//DB & SCHEMA

USE AZURE\_DATA.JSON\_DATA;

//CREATE INTEGRATION- ENABLES COMMUNICATION TO AZURE

**CREATE STORAGE INTEGRATION AZURE\_INT**

**TYPE=EXTERNAL\_STAGE**

**STORAGE\_PROVIDER=AZURE**

**ENABLED=TRUE**

**AZURE\_TENANT\_ID='4fc3d464-9f48-4ce1-85b9-3eeeef5d76ca' //PARENT MANAGEMENT GROUP IN AZURE SUBSCRIPTIONS IAM**

**STORAGE\_ALLOWED\_LOCATIONS=('azure://snowflakeevdata.blob.core.windows.net/snowflake-ev-demo/json/');**

DESC INTEGRATION AZURE\_INT

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AZURE\_CONSENT\_URL https://login.microsoftonline.com/4fc3d464-9f48-4ce1-85b9-3eeeef5d76ca/oauth2/authorize?client\_id=7df75a2e-58de-4247-b408-8d32f19eb822&response\_type=code

AZURE\_MULTI\_TENANT\_APP\_NAME a5cq6gsnowflakepacint\_1741516020271

CLICK ON THE LINK & CREATE ROLE ASSIGNMENT WITH DATA BLOB CONTRIBUTOR AND SEARCH APP\_NAME IN MEMBERS

\*\*\*\*\*\*\*\*\*\*\*/

### **FILE FORMAT**

//CREATE FILE FORMAT

**CREATE OR REPLACE FILE FORMAT AZURE\_DATA.JSON\_DATA.JSON\_FORMAT**

**TYPE='json';**

### **STAGE**

**CREATE OR REPLACE STAGE AZURE\_DATA.JSON\_DATA.AZURE\_STAGE**

**STORAGE\_INTEGRATION=AZURE\_INT**

**URL= 'azure://snowflakeevdata.blob.core.windows.net/snowflake-ev-demo/json/ElectricVehiclePopulationData.json'**

**FILE\_FORMAT=JSON\_FORMAT;**

### **RAW TABLE**

**CREATE OR REPLACE TABLE EVTABLE\_RAW(**

**JSON\_DATA VARIANT**

**)**

**COPY INTO EVTABLE\_RAW**

**FROM @AZURE\_DATA.JSON\_DATA.AZURE\_STAGE**

SELECT \* FROM EVTABLE\_RAW

### **RELATIONAL TABLE**

CREATE OR REPLACE TABLE EVTABLE\_FINAL

(

sid VARCHAR(20),

id VARCHAR(100),

position VARCHAR(100),

created\_at DATE,

created\_meta DATE ,

updated\_at DATE,

updated\_meta DATE,

vin VARCHAR(20) PRIMARY KEY,

county VARCHAR(20),

city VARCHAR(100),

State VARCHAR(20),

POSTAL\_CODE VARCHAR(5) ,

MODEL\_YEAR VARCHAR(20),

MAKE VARCHAR(20) ,

MODEL VARCHAR(100) ,

ELECTRICAL\_VEHICLE\_TYPE VARCHAR(100),

CAFV\_ELIGIBILITY VARCHAR(100) ,

ELECTRIC\_RANGE INT DEFAULT 0,

BASE\_MSRP INT DEFAULT 0,

LEGISLATIVE\_DISTRICT VARCHAR(20),

DOL\_VEHICLE\_ID VARCHAR(20) ,

VEHICLE\_LOCATION VARCHAR(100) ,

ELECTRIC\_UTILITY VARCHAR(1000) ,

CENSUS\_TRACT\_2020 VARCHAR(20) ,

COUNTIES NUMBER,

CONGRESSIONAL\_DISTRICTS NUMBER,

LEGISLATIVE\_DISTRICT\_BOUNDARY NUMBER

);

INSERT INTO EVTABLE\_FINAL

SELECT

f.value[0]::VARCHAR(20) sid,

f.value[1]::VARCHAR(100) id,

f.value[2]::VARCHAR(100) position,

TO\_TIMESTAMP(f.value[3]::INT) created\_at,

TO\_TIMESTAMP(f.value[4]::INT) created\_meta,

TO\_TIMESTAMP(f.value[5]::INT) updated\_at,

TO\_TIMESTAMP(f.value[6]::INT) updated\_meta,

//f.value[7] meta,

f.value[8]::VARCHAR(20) vin,

f.value[9]::VARCHAR(20) county,

f.value[10]::VARCHAR(100) city,

f.value[11]::VARCHAR(20) State,

f.value[12]::VARCHAR(5) POSTAL\_CODE,

f.value[13]::VARCHAR(4) MODEL\_YEAR,

f.value[14]::VARCHAR(20) MAKE,

f.value[15]::VARCHAR(100) MODEL,

f.value[16]::VARCHAR(100) ELECTRICAL\_VEHICLE\_TYPE,

f.value[17]::VARCHAR(100) CAFV\_ELIGIBILITY,

f.value[18]::INT ELECTRIC\_RANGE,

f.value[19]::INT BASE\_MSRP,

f.value[20]::VARCHAR(20) LEGISLATIVE\_DISTRICT,

f.value[21]::VARCHAR(20) DOL\_VEHICLE\_ID,

f.value[22]::VARCHAR(100) VEHICLE\_LOCATION,

f.value[23]::VARCHAR(1000) ELECTRIC\_UTILITY,

F.VALUE[24]::VARCHAR(20) CENSUS\_TRACT\_2020,

F.VALUE[25]::NUMBER COUNTIES,

F.VALUE[26]::NUMBER CONGRESSIONAL\_DISTRICTS,

F.VALUE[27]::NUMBER LEGISLATIVE\_DISTRICT\_BOUNDARY

FROM

TABLE(FLATTEN(input => PARSE\_JSON(SELECT \* FROM EVTABLE\_RAW)::VARIANT:data)) AS F

;

### **SCHEMA VALIDATION**

1ST Time Load

CREATE OR REPLACE TABLE EVTABLE\_COLUMNS

(

COLUMN\_ID VARCHAR(10),

COLUMN\_NAME VARCHAR(20),

COLUMN\_POSITION INT,

COLUMN\_DESCRIPTION VARCHAR(1000),

COLUMN\_DATATYPE VARCHAR(10)

);

Insert following table

| COLUMN\_ID | COLUMN\_NAME | COLUMN\_POSITION | COLUMN\_DESCRIPTION | COLUMN\_DATATYPE |
| --- | --- | --- | --- | --- |
| 561974342 | VIN (1-10) | 1 | The 1st 10 characters of each vehicle's Vehicle Identification Number (VIN). | text |
| 561974343 | County | 2 | The county in which the registered owner resides. | text |
| 561974344 | City | 3 | The city in which the registered owner resides. | text |
| 561974345 | State | 4 | The state in which the registered owner resides. | text |
| 561974346 | Postal Code | 5 | The 5 digit zip code in which the registered owner resides. | text |
| 561974347 | Model Year | 6 | The model year of the vehicle, determined by decoding the Vehicle Identification Number (VIN). | text |
| 561974348 | Make | 7 | The manufacturer of the vehicle, determined by decoding the Vehicle Identification Number (VIN). | text |
| 561974349 | Model | 8 | The model of the vehicle, determined by decoding the Vehicle Identification Number (VIN). | text |
| 561974350 | Electric Vehicle Type | 9 | This distinguishes the vehicle as all electric or a plug-in hybrid. | text |
| 561974351 | Clean Alternative Fuel Vehicle (CAFV) Eligibility | 10 | This categorizes vehicle as Clean Alternative Fuel Vehicles (CAFVs) based on the fuel requirement and electric-only range requirement in House Bill 2042 as passed in the 2019 legislative session. | text |
| 561974352 | Electric Range | 11 |  | number |
| 561974353 | Base MSRP | 12 | This is the lowest Manufacturer's Suggested Retail Price (MSRP) for any trim level of the model in question. | number |
| 561974354 | Legislative District | 13 | The specific section of Washington State that the vehicle's owner resides in, as represented in the state legislature. | number |
| 561974355 | DOL Vehicle ID | 14 | Unique number assigned to each vehicle by Department of Licensing for identification purposes. | text |
| 561974356 | Vehicle Location | 15 | The center of the ZIP Code for the registered vehicle. | point |
| 561974358 | Electric Utility | 17 |  | text |
| 561974362 | 2020 Census Tract | 18 |  | text |
| 561974360 | Counties | 19 |  | number |
| 561974361 | Congressional Districts | 20 |  | number |
| 561974359 | WAOFM - GIS - Legislative District Boundary | 21 |  | number |

**CHECKING FOR NEW COLUMNS**

SELECT COUNT(\*) FROM (

SELECT COLUMN\_ID, COLUMN\_NAME FROM EVTABLE\_COLUMNS C

WHERE **NOT EXISTS** ( SELECT

G.VALUE:"id"::INT AS COLUMN\_ID,

G.VALUE:"name"::VARCHAR(50) AS COLUMN\_NAME

FROM EVTABLE\_RAW x,

LATERAL FLATTEN(input => x.json\_data:meta) AS F,

LATERAL FLATTEN(INPUT => F.VALUE:"columns") as G

WHERE G.VALUE:"position"::INT>0 OR G.VALUE:"id"::INT>0 AND **G.VALUE:"id"::INT=C.COLUMN\_ID**));

### **SUBMISSION TABLE CREATION**

CREATE OR REPLACE TABLE APPROVALS(

REVIEWED\_AT DATETIME,

REVIEWED\_AUTOMATICALLY BOOLEAN,

STATE VARCHAR(20),

SUBMISSION\_ID INT,

SUBMISSION\_OBJECT VARCHAR(50),

SUBMISSION\_OUTCOME VARCHAR(50),

SUBMITTED\_AT DATETIME,

WORKFLOW\_ID INT,

PERMISSION\_TYPE VARCHAR(20),

FAILURE\_COUNT INT,

COUTCOME\_STATUS VARCHAR(30),

SUBMITTER\_ID VARCHAR(50),

SUBMITTER\_NAME VARCHAR(50)

)

INSERT INTO APPROVALS (

SELECT

TO\_TIMESTAMP(F.value:reviewedAt::INT) AS REVIEWED\_AT,

F.value:reviewedAutomatically::BOOLEAN AS REVIEWED\_AUTOMATICALLY,

F.value:state::VARCHAR(20) AS STATE,

F.value:submissionId::INT AS SUBMISSION\_ID,

F.value:submissionObject::VARCHAR(50) AS SUBMISSION\_OBJECT,

F.value:submissionOutcome::VARCHAR(50) AS SUBMISSION\_OUTCOME,

TO\_TIMESTAMP(F.value:submittedAt::INT) AS SUBMITTED\_AT,

F.value:workflowId::INT AS WORKFLOW\_ID,

F.value:submissionDetails.permissionType::VARCHAR(20) AS PERMISSION\_TYPE,

F.value:submissionOutcomeApplication.failureCount::INT AS FAILURE\_COUNT,

F.value:submissionOutcomeApplication.status::VARCHAR(30) AS COUTCOME\_STATUS,

F.value:submitter.id::VARCHAR(50) AS SUBMITTER\_ID,

F.value:submitter.displayName::VARCHAR(50) AS SUBMITTER\_NAME

FROM EVTABLE\_RAW x,

LATERAL FLATTEN(input => x.JSON\_DATA:meta.view.approvals) AS F)