-- RAW.CONFORMED.DATEDIMENSIONRAW.CONFORMED.DATEDIMENSION

-- create databases

create database if not exists analytics;

create database if not exists raw;

--create schemas

create schema if not exists analytics.chinook;

create schema if not exists raw.chinook;

create schema if not exists raw.conformed;

-- define file formats

create or replace file format RAW.PUBLIC.PARQUET

TYPE = parquet

REPLACE\_INVALID\_CHARACTERS = TRUE;

create or replace file format RAW.PUBLIC.JSONARRAY

TYPE = json

STRIP\_OUTER\_ARRAY = TRUE;

CREATE OR REPLACE FILE FORMAT RAW.PUBLIC.JSON

TYPE = 'JSON'

STRIP\_OUTER\_ARRAY = FALSE;

create or replace file format RAW.PUBLIC.CSVHEADER

TYPE = 'csv'

FIELD\_DELIMITER = ','

SKIP\_HEADER=1;

create or replace file format RAW.PUBLIC.CSV

TYPE = csv

FIELD\_DELIMITER = ','

PARSE\_HEADER = FALSE

SKIP\_HEADER = 0;

-- create stages

-- varying file formats

CREATE or replace STAGE RAW.PUBLIC.externalworld\_files

URL = 'azure://externalworld.blob.core.windows.net/files/';

-- these are all parquet file formats

CREATE or replace STAGE RAW.PUBLIC.externalworld\_database

URL = 'azure://externalworld.blob.core.windows.net/database/'

FILE\_FORMAT = RAW.PUBLIC.PARQUET ;

-- stage the date dimension

CREATE or REPLACE TABLE raw.conformed.datedimension (

datekey int

,date date

,datetime timestamp

,year int

,quarter int

,quartername varchar(2)

,month int

,monthname varchar(3)

,day int

,dayofweek int

,dayname varchar(3)

,weekday varchar(1)

,weekofyear int

,dayofyear int

) AS

WITH CTE\_MY\_DATE AS (

SELECT DATEADD(DAY, SEQ4(), '1997-01-01 00:00:00') AS MY\_DATE

FROM TABLE(GENERATOR(ROWCOUNT=>365\*30))

)

SELECT

REPLACE(TO\_DATE(MY\_DATE)::varchar,'-','')::int as datekey,

TO\_DATE(MY\_DATE) as date

,TO\_TIMESTAMP(MY\_DATE) as datetime

,YEAR(MY\_DATE) as year

,QUARTER(MY\_DATE) as quarter

,CONCAT('Q', QUARTER(MY\_DATE)::varchar) as quartername

,MONTH(MY\_DATE) as month

,MONTHNAME(MY\_DATE) as monthname

,DAY(MY\_DATE) as day

,DAYOFWEEK(MY\_DATE) as dayofweek

,DAYNAME(MY\_DATE) as dayname

,case when DAYOFWEEK(MY\_DATE) between 1 and 5 then 'Y' else 'N' end as weekday

,WEEKOFYEAR(MY\_DATE) as weekofyear

,DAYOFYEAR(MY\_DATE) as dayofyear

FROM CTE\_MY\_DATE

;

-------customer\_track\_reviews

-- Create or replace the table

CREATE OR REPLACE TABLE RAW.CHINOOK.customer\_track\_reviews

(

Customerid INT,

Trackid INT,

Sentiment VARCHAR

);

-- Copy the JSON data into the table using MATCH\_BY\_COLUMN\_NAME

COPY INTO RAW.CHINOOK.customer\_track\_reviews

FROM '@RAW.PUBLIC.externalworld\_files/chinook.customer-track-reviews.json'

FILE\_FORMAT = (FORMAT\_NAME = RAW.PUBLIC.JSONARRAY)

MATCH\_BY\_COLUMN\_NAME = CASE\_INSENSITIVE;

-- stage chinook customer

create or replace table RAW.chinook.customer

(

customerid int,

firstname varchar,

lastname varchar,

company varchar,

address varchar,

city varchar,

state varchar,

country varchar,

postalcode varchar,

phone varchar,

fax varchar,

email varchar,

supportrepid int

);

copy into RAW.chinook.customer

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.customer.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook invoice

CREATE OR REPLACE TABLE RAW.chinook.invoice

(

invoiceid INT,

customerid INT,

invoicedate varchar,

billingaddress VARCHAR,

billingcity VARCHAR,

billingstate VARCHAR,

billingcountry VARCHAR,

billingpostalcode VARCHAR,

total float

);

COPY INTO RAW.chinook.invoice

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.invoice.parquet'

FILE\_FORMAT = (TYPE = 'PARQUET')

MATCH\_BY\_COLUMN\_NAME = 'CASE\_INSENSITIVE';

---- stage chinook invoiceline

create or replace table RAW.chinook.invoiceline

(

invoicelineid int,

invoiceid int,

trackid int,

unitprice float ,

quantity float

);

copy into RAW.chinook.invoiceline

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.invoiceline.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook album

create or replace table RAW.chinook.album

(

albumid int,

title varchar,

artistid int

);

copy into RAW.chinook.album

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.album.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook playlist

create or replace table RAW.chinook.playlist

(

playlistid int,

name varchar

);

copy into RAW.chinook.playlist

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.playlist.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook playlisttrack

create or replace table RAW.chinook.playlistTrack

(

playlistid int,

trackid int

);

copy into RAW.chinook.playlistTrack

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.playlisttrack.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook Track

create or replace table RAW.chinook.track

(

trackid int,

Name varchar,

albumid int,

mediatypeid int,

genreid varchar,

composer varchar,

milliseconds float,

bytes float,

unitprice float

);

copy into RAW.chinook.track

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.track.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook artist

create or replace table RAW.chinook.artist

(

artistid int,

Name varchar

);

copy into RAW.chinook.artist

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.artist.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook mediatype

create or replace table RAW.chinook.mediatype

(

mediatypeid int,

Name varchar

);

copy into RAW.chinook.mediatype

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.mediatype.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook genre

create or replace table RAW.chinook.genre

(

genreid int,

Name varchar

);

copy into RAW.chinook.genre

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.genre.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

---- stage chinook employee

create or replace table RAW.chinook.employee

(

employeeid int,

lastName varchar,

firstname varchar,

title varchar,

reportsto int,

birthdate varchar,

hiredate varchar,

address varchar,

city varchar,

state varchar,

country varchar,

postalcode varchar,

phone varchar,

fax varchar,

email varchar

);

copy into RAW.chinook.employee

FROM '@"RAW"."PUBLIC"."EXTERNALWORLD\_DATABASE"/chinook.employee.parquet'

MATCH\_BY\_COLUMN\_NAME='CASE\_INSENSITIVE';

-------------------------------------------------------------------

select TABLE\_CATALOG,TABLE\_SCHEMA,TABLE\_NAME,ROW\_COUNT

from RAW.INFORMATION\_SCHEMA.TABLES

where TABLE\_SCHEMA='CHINOOK' order by TABLE\_NAME;