SNOWFLAKE SQL SCRIPT

CREATE OR REPLACE STORAGE INTEGRATION PBI\_Integration

TYPE = EXTERNAL\_STAGE

STORAGE\_PROVIDER = 'S3'

ENABLED = TRUE

STORAGE\_AWS\_ROLE\_ARN = ‘XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX’

STORAGE\_ALLOWED\_LOCATIONS = ('s3://myhranalyticsproject/')

COMMENT = 'Optional Comment'

//description Integration Object

desc integration PBI\_Integration

//drop integration PBI\_Integration

//drop database POWERBI

CREATE OR REPLACE DATABASE POWERBI\_HRANALYTICS;

//create schema PBI\_Data;

CREATE OR REPLACE SCHEMA POWERBI\_HRANALYTICS.PBI\_Data;

USE SCHEMA POWERBI\_HRANALYTICS.PBI\_Data;

CREATE OR REPLACE TABLE Employee (

  EmployeeID VARCHAR,

  FirstName VARCHAR,

  LastName VARCHAR,

  Gender VARCHAR,

  Age NUMBER(3,0),

  BusinessTravel VARCHAR,

  Department VARCHAR,

  DistanceFromHome\_KM NUMBER(5,2),

  State VARCHAR,

  Ethnicity VARCHAR,

  Education NUMBER(1,0),

  EducationField VARCHAR,

  JobRole VARCHAR,

  MaritalStatus VARCHAR,

  Salary NUMBER(10,2),

  StockOptionLevel NUMBER(1,0),

  OverTime VARCHAR,

  HireDate DATE,

  Attrition VARCHAR,

  YearsAtCompany NUMBER(2,0),

  YearsInMostRecentRole NUMBER(2,0),

  YearsSinceLastPromotion NUMBER(2,0),

  YearsWithCurrManager NUMBER(2,0)

);

select \* from EMPLOYEE

//drop database test;

create stage POWERBI\_HRANALYTICS.PUBLIC.PBI\_STAGE2

url = 's3://myhranalyticsproject'

storage\_integration = PBI\_Integration

//desc stage s1

//drop stage PowerBI.PBI\_Data.pbi\_stage2;

copy into EMPLOYEE

from @pbi\_stage2/Employee.csv

file\_format = (type=csv field\_delimiter=',' skip\_header=1 )

on\_error = 'continue'

list @pbi\_stage2

CREATE OR REPLACE TABLE EducationLevel (

EducationLevelID Number,

EducationLevel VARCHAR

);

Select \* from EDUCATIONLEVEL

copy into EducationLevel

from @pbi\_stage2/EducationLevel.csv

file\_format = (type=csv field\_delimiter=',' skip\_header=1)

on\_error = 'continue'

CREATE TABLE SatisfiedLevel (

SatisfactionID Number,

SatisfactionLevel VARCHAR

);

SELECT \* FROM SatisfiedLevel

//DROP TABLE SatisfactionLevel

copy into SatisfiedLevel

from @pbi\_stage2/SatisfiedLevel.csv

file\_format = (type=csv field\_delimiter=',',skip\_header=1)

on\_error = 'continue'

CREATE TABLE RatingLevel (

RatingID Number,

RatingLevel VARCHAR

)

copy into RatingLevel

FROM @pbi\_stage2/RatingLevel.csv

file\_format = (type=csv field\_delimiter=',', skip\_header=1)

on\_error='continue'

SELECT \* FROM RATINGLEVEL

CREATE TABLE PerformanceRating (

PerformanceID VARCHAR,

EmployeeID VARCHAR,

ReviewDate Date,

EnvironmentSatisfaction Number,

JobSatisfaction Number,

RelationshipSatisfaction Number,

TrainingOpportunitiesWithinYear Number,

TrainingOpportunitiesTaken Number,

WorkLifeBalance Number,

SelfRating Number,

ManagerRating Number

)

copy into PerformanceRating

FROM @pbi\_stage2/PerformanceRating.csv

file\_format = (type=csv field\_delimiter=',', skip\_header=1)

on\_error='continue'

SELECT \* FROM PERFORMANCERATING

SELECT CURRENT\_WAREHOUSE();

CREATE OR REPLACE TABLE DimDate AS

WITH date\_bounds AS (

    SELECT

        DATE\_TRUNC('year', MIN(HireDate)) AS min\_year\_start,

        DATEADD(day, -1, DATEADD(year, 1, DATE\_TRUNC('year', MAX(HireDate)))) AS max\_year\_end

    FROM Employee

),

calendar AS (

    SELECT

        DATEADD(day, SEQ4(), (SELECT min\_year\_start FROM date\_bounds)) AS Date

    FROM TABLE(GENERATOR(ROWCOUNT => 10000))

    WHERE DATEADD(day, SEQ4(), (SELECT min\_year\_start FROM date\_bounds))

          <= (SELECT max\_year\_end FROM date\_bounds)

)

SELECT

  Date,

  YEAR(Date) AS Year,

  DATE\_TRUNC('year', Date) AS Year\_Start,

  LAST\_DAY(Date, 'year') AS Year\_End,

  MONTH(Date) AS Month\_Number,

  DATE\_TRUNC('month', Date) AS Month\_Start,

  LAST\_DAY(Date, 'month') AS Month\_End,

  DAY(LAST\_DAY(Date, 'month')) AS Days\_In\_Month,

  (YEAR(Date) \* 100 + MONTH(Date)) AS YearMonthNumber,

  TO\_CHAR(Date, 'YYYY-Mon') AS YearMonthName,

  DAY(Date) AS Day\_Number,

  RTRIM(TO\_CHAR(Date, 'Day')) AS Day\_Name,

  SUBSTR(RTRIM(TO\_CHAR(Date, 'Day')), 1, 3) AS Day\_Name\_Short,

  ((DAYOFWEEKISO(Date) % 7) + 1) AS Day\_Of\_Week,

  RTRIM(TO\_CHAR(Date, 'Month')) AS Month\_Name,

  TO\_CHAR(Date, 'Mon') AS Month\_Name\_Short,

  QUARTER(Date) AS Quarter,

  'Q' || QUARTER(Date) AS Quarter\_Name,

  (YEAR(Date) \* 10 + QUARTER(Date)) AS YearQuarterNumber,

  TO\_CHAR(Date, 'YYYY') || ' Q' || QUARTER(Date) AS YearQuarterName,

  DATE\_TRUNC('quarter', Date) AS Quarter\_Start,

  LAST\_DAY(Date, 'quarter') AS Quarter\_End,

  CASE

    WHEN (DATEDIFF(day, DATE\_TRUNC('year', Date), Date) + 1)

         <= (8 - ((DAYOFWEEKISO(DATE\_TRUNC('year', Date)) % 7) + 1))

    THEN 1

    ELSE TRUNC(((DATEDIFF(day, DATE\_TRUNC('year', Date), Date) + 1)

                - (8 - ((DAYOFWEEKISO(DATE\_TRUNC('year', Date)) % 7) + 1)) - 1) / 7) + 2

  END AS Week\_Number,

  DATEADD(day, 1 - ((DAYOFWEEKISO(Date) % 7) + 1), Date) AS Week\_Start,

  DATEADD(day, 7 - ((DAYOFWEEKISO(Date) % 7) + 1), Date) AS Week\_End,

  YEAR(DATEADD(month, -3, Date)) AS FiscalYear,

  QUARTER(DATEADD(month, -3, Date)) AS FiscalQuarter,

  MONTH(DATEADD(month, -3, Date)) AS FiscalMonth

FROM calendar

ORDER BY Date;