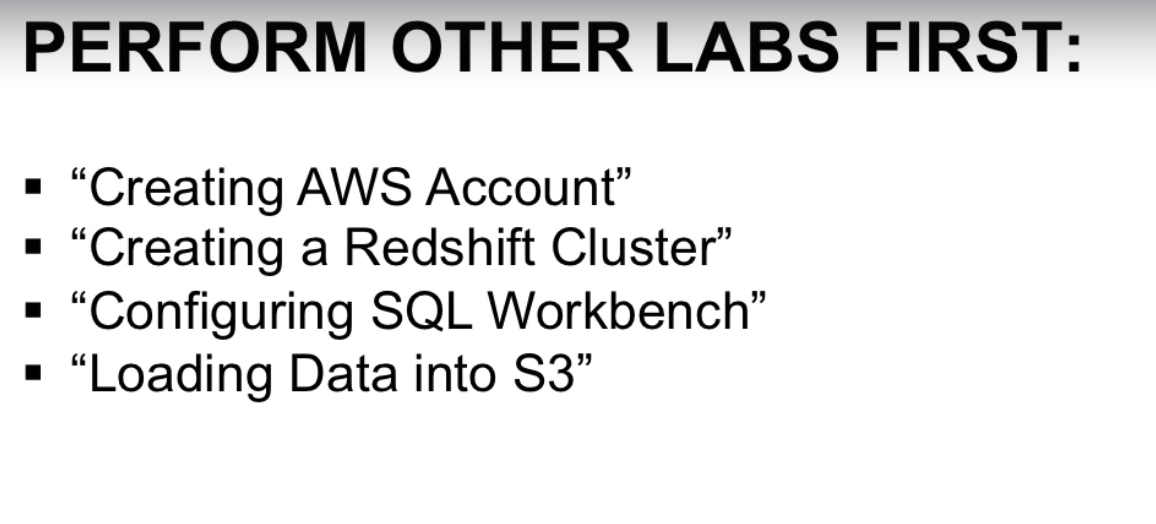
**Loading Data into RedShift**

Pre-Requisites-



Links Referred-

<http://docs.aws.amazon.com/redshift/latest/dg/tutorial-loading-data-download-files.html>

DIRECT ACCESS TO THE DOWNLOAD IS AT:

<https://s3.amazonaws.com/awssampledb/LoadingDataSampleFiles.zip>

// UPLOAD THESE TO YOUR S3 BUCKET

**Step1-**

Run the following script on your **SQL Workbench**

select distinct(tablename) from pg\_table\_def where schemaname = 'public';

// If that shows tables (it shouldn't), then you need to issue some drop statements

drop table part;

drop table supplier;

drop table customer;

drop table dwdate;

drop table lineorder;

CREATE TABLE part

(

p\_partkey INTEGER NOT NULL,

p\_name VARCHAR(22) NOT NULL,

p\_mfgr VARCHAR(6),

p\_category VARCHAR(7) NOT NULL,

p\_brand1 VARCHAR(9) NOT NULL,

p\_color VARCHAR(11) NOT NULL,

p\_type VARCHAR(25) NOT NULL,

p\_size INTEGER NOT NULL,

p\_container VARCHAR(10) NOT NULL

);

CREATE TABLE supplier

(

s\_suppkey INTEGER NOT NULL,

s\_name VARCHAR(25) NOT NULL,

s\_address VARCHAR(25) NOT NULL,

s\_city VARCHAR(10) NOT NULL,

s\_nation VARCHAR(15) NOT NULL,

s\_region VARCHAR(12) NOT NULL,

s\_phone VARCHAR(15) NOT NULL

);

CREATE TABLE customer

(

c\_custkey INTEGER NOT NULL,

c\_name VARCHAR(25) NOT NULL,

c\_address VARCHAR(25) NOT NULL,

c\_city VARCHAR(10) NOT NULL,

c\_nation VARCHAR(15) NOT NULL,

c\_region VARCHAR(12) NOT NULL,

c\_phone VARCHAR(15) NOT NULL,

c\_mktsegment VARCHAR(10) NOT NULL

);

CREATE TABLE dwdate

(

d\_datekey INTEGER NOT NULL,

d\_date VARCHAR(19) NOT NULL,

d\_dayofweek VARCHAR(10) NOT NULL,

d\_month VARCHAR(10) NOT NULL,

d\_year INTEGER NOT NULL,

d\_yearmonthnum INTEGER NOT NULL,

d\_yearmonth VARCHAR(8) NOT NULL,

d\_daynuminweek INTEGER NOT NULL,

d\_daynuminmonth INTEGER NOT NULL,

d\_daynuminyear INTEGER NOT NULL,

d\_monthnuminyear INTEGER NOT NULL,

d\_weeknuminyear INTEGER NOT NULL,

d\_sellingseason VARCHAR(13) NOT NULL,

d\_lastdayinweekfl VARCHAR(1) NOT NULL,

d\_lastdayinmonthfl VARCHAR(1) NOT NULL,

d\_holidayfl VARCHAR(1) NOT NULL,

d\_weekdayfl VARCHAR(1) NOT NULL

);

CREATE TABLE lineorder

(

lo\_orderkey INTEGER NOT NULL,

lo\_linenumber INTEGER NOT NULL,

lo\_custkey INTEGER NOT NULL,

lo\_partkey INTEGER NOT NULL,

lo\_suppkey INTEGER NOT NULL,

lo\_orderdate INTEGER NOT NULL,

lo\_orderpriority VARCHAR(15) NOT NULL,

lo\_shippriority VARCHAR(1) NOT NULL,

lo\_quantity INTEGER NOT NULL,

lo\_extendedprice INTEGER NOT NULL,

lo\_ordertotalprice INTEGER NOT NULL,

lo\_discount INTEGER NOT NULL,

lo\_revenue INTEGER NOT NULL,

lo\_supplycost INTEGER NOT NULL,

lo\_tax INTEGER NOT NULL,

lo\_commitdate INTEGER NOT NULL,

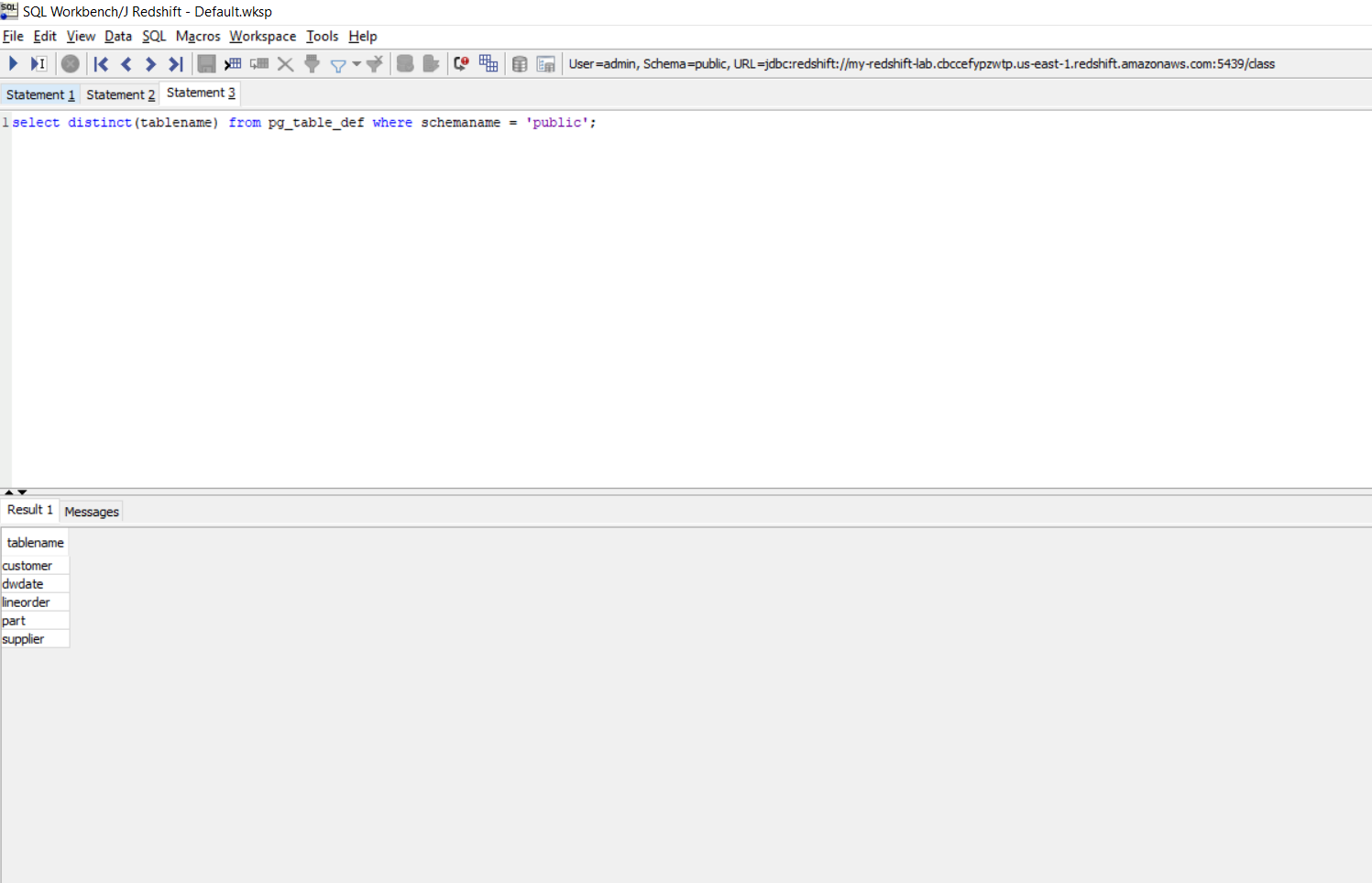
lo\_shipmode VARCHAR(10) NOT NULL

);

select distinct(tablename) from pg\_table\_def where schemaname = 'public';

//should now show tables.

After you have run the above script you should be able to see the output as shown in IMG 1.1



IMG 1.1

Step2-

Run the following script on your SQL Workbench

**set wlm\_query\_slot\_count to 2;**

(This will create 2 query slots, one which will allow us to run our copy commands and then another if we run into any issues)

Step3- **Running different Copy Commands with various options**

This is the Basic Copy Command structure -

**copy table**

**from 's3://[YOUR-BUCKET-NAME]/data-load/[KEY-PREFIX]'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]**

**options;**

**Note**-

* This is the basic structure of your copy command, your options will vary
* You have to replace [YOUR-BUCKET-NAME] with Bucket name you have given on your S3 while creating the bucket, same goes with folder name and replace the [YOUR-ACCESS-KEY] with the Access Key in your **Credentials.csv**(when you created the IAM user) and same with [YOUR-SECRET-ACCESS-KEY]

// FOR YOUR SPECIFIC TABLES & BUCKET [THIS LOAD WILL FAIL]

**copy part**

**from 's3://[YOUR-BUCKET-NAME]/ [YOUR-FOLDER-NAME]/part-csv.tbl'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-AWS-SECRET-ACCESS-KEY];**

// FOR YOUR SPECIFIC TABLES & BUCKET [THIS LOAD WILL SUCCEED]

**copy part**

**from 's3:// [YOUR-BUCKET-NAME]/[YOUR-FOLDER-NAME]/part-csv.tbl'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-AWS-SECRET-ACCESS-KEY]'**

csv

null as '\000';

// IN 2ND TAB ("Statement 2" in SQL Workbench), RUN THE FOLLOWING TO SEE WHY YOU FAILED LOAD

**select query, substring(filename,22,25) as filename,line\_number as line,**

**substring(colname,0,12) as column, type, position as pos, substring(raw\_line,0,30) as line\_text,**

**substring(raw\_field\_value,0,15) as field\_text, substring(err\_reason,0,45) as reason**

**from stl\_load\_errors**

**order by query desc**

**limit 10;**

// LOAD THE SUPPLIER DATA [IT COMES FROM AN AWS-SUPPORTED BUCKET IN THE 'us-east-1' REGION]

**copy supplier**

**from 's3://awssampledbuswest2/ssbgz/supplier.tbl'**

**credentials 'aws\_access\_key\_id=AKIAILSHMHSSQGL23HTA;aws\_secret\_access\_key=** **+Wko3dU6C20Xoeb3gL0i/5RxhHoW3rvuLDAfKB5V'**

**delimiter '|'**

**gzip**

**region 'us-east-1';**

// CUSTOMER DATA IS FIXED-WIDTH, SO TRY TO ADD THAT

**copy customer**

**from 's3://smasand5640/data-load/customer-fw.tbl'**

**credentials 'aws\_access\_key\_id=AKIAILSHMHSSQGL23HTA;aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**fixedwidth 'c\_custkey:10, c\_name:25, c\_address:25, c\_city:10, c\_nation:15, c\_region :12, c\_phone:15,c\_mktsegment:10'**

**maxerror 10;**

// LOAD THE CUSTOMER DATA [IT'S FIXED-WIDTH, SO USE MANIFEST -- THIS WILL FAIL BECAUSE OF BAD DATA]

**copy customer**

**from 's3://smasand5640/data-load/customer-fw.tbl'**

**credentials 'aws\_access\_key\_id=AKIAILSHMHSSQGL23HTA;aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**fixedwidth 'c\_custkey:10, c\_name:25, c\_address:25, c\_city:10, c\_nation:15, c\_region :12, c\_phone:15,c\_mktsegment:10';**

// NOW THIS SHOULD WORK ['acceptinvchars' only works on VARCHAR cols, but replaces any bad chars with whatever you specify]

**copy customer**

**from 's3://[YOUR-BUCKET-NAME]/[YOUR-FOLDER-NAME]/customer-fw-manifest'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**fixedwidth 'c\_custkey:10, c\_name:25, c\_address:25, c\_city:10, c\_nation:15, c\_region :12, c\_phone:15,c\_mktsegment:10'**

**maxerror 10**

acceptinvchars as '^'

manifest;

// WE STILL HAVE DATA ERRORS, USE A MANIFEST TO AVOID UNWANTED TABLES

// [THIS BUCKET FOLDER CONTAINS A BUNCH OF FILES LIKE .log, .bak THAT WE DON'T WANT TO LOAD]

// **CREATE A MANIFEST**

**{**

**"entries": [**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-000"},**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-001"},**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-002"},**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-003"},**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-004"},**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-005"},**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-006"},**

**{"url":"s3://smasand5640/data-load/customer-fw.tbl-007"}**

**]**

**}**

// SAVE THIS AS 'customer-fw-manifest' & UPLOAD TO YOUR S3 BUCKET

Step 4- **Copying the data from S3 Buckets hosted by AWS**

**// select count(\*) from customer;**

**select distinct(tablename) from pg\_table\_def where schemaname = 'public';**

**select count(\*) from customer;**

**select count(\*) from dwdate;**

**select count(\*) from lineorder;**

**select count(\*) from part;**

**select count(\*) from supplier;**

**delete from customer;**

**delete from part;**

**delete from supplier;**

// NOW GO AHEAD & RE-CREATE THOSE TABLES & PERFORM ALL THE LOADS BELOW, ONE AT A TIME!

// COPY dwdate

**copy dwdate from 's3://awssampledbuswest2/ssbgz/dwdate'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**gzip**

**compupdate off**

// COPY part

**copy part from 's3://awssampledbuswest2/ssbgz/part'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**gzip**

**compupdate off**

// COPY supplier

**copy supplier from 's3://awssampledbuswest2/ssbgz/supplier'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**gzip**

**compupdate off**

// COPY customer

**copy customer from 's3://awssampledbuswest2/ssbgz/customer'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**gzip**

**compupdate off**

// COPY lineorder (has 600,000,000 records... gonna be very slow)

**copy lineorder from 's3://awssampledbuswest2/ssbgz/lineorder'**

**credentials 'aws\_access\_key\_id=[YOUR-ACCESS-KEY];aws\_secret\_access\_key=[YOUR-SECRET-ACCESS-KEY]'**

**gzip**

**compupdate off**

// LINEORDER TAKES A LOOOONG TIME TO COPY

// CHECK STL\_LOAD\_COMMITS TABLE TO SEE THE LAST COMMIT:

**select query, trim(filename) as file, curtime as updated**

**from stl\_load\_commits order by updated desc;**

// CHECK DATA DISTRIBUTION ON ALL TABLES

**select slice, col, num\_values, minvalue, maxvalue**

**from svv\_diskusage**

**where name='customer' and col=0**

**order by slice,col;**

// LOOK FOR DISK SPILLS

**select query, step, rows, workmem, label, is\_diskbased**

**from svl\_query\_summary**

**where query = [YOUR-QUERY-ID]**

**order by workmem desc;**

// CHECK COLUMN, DISTKEY, SORTKEY FOR A GIVEN TABLE

**select "column", type, encoding, distkey, sortkey, "notnull"**

**from pg\_table\_def**

**where tablename = 'lineorder';**