­

**Performance Assessment for D211: Advanced Data Acquisition  
Part 1: Data Dashboards**

Christopher Fischer - 011933891

College of Information Technology, Western Governors University

September 19, 2024

Performance Assessment for D211: Advanced Data Acquisition – Part 1

# A1 – Data Sets

The following datasets comprise the inputs to my Churn Dashboard:

1. The ***churn*** PostgreSQL database provided by WGU faculty
   1. This database contains demographics, account information, and service information for a telecom company’s 10,000 customers, some of whom have canceled their services (churned).
   2. This database will be created from a backup file included with my submission.
2. A ***stateregion*** table added to the PostgreSQL database
   1. I created this data using the company’s grouping of states into regions.
   2. This table will be created from a SQL script included with my submission.
3. ***Broadband\_data\_opendatachallenge.csv*** available on [GitHub](https://github.com/BroadbandNow/Open-Data)
   1. This file contains population, broadband service availability, and pricing data for most US zip codes.
   2. This file is included in my submission and is joined to my custom SQL query within Tableau.

# A2 – Installation Instructions

Please follow these steps to install my dashboard.

## Environment Preparation

1. Launch the standard ***LabOnDemand*** environment for D211
2. Unzip the file attached to my submission and place these four files in the “c:\users\LabUser\Desktop” folder
   1. D211\_churn.twbx
   2. churn.backup
   3. stateregion.sql
   4. broadband\_data\_openchallenge.csv

## Data Preparation

1. Open ***pgadmin4***
2. Connect to the localhost server on port 5432 with:
   1. Username: postgres
   2. Password: Passw0rd!
3. If no ***churn*** database is available, follow these steps:
   1. Right-click on “Databases”
   2. Click “Create” then “Database…”
   3. Enter “churn” for database name
   4. Click “Save”
   5. Right-click on “Databases”
   6. Click “Refresh”
   7. Right-click on the “Churn” database
   8. Click “Restore”
   9. Click the “…” next to “Filename”
   10. Navigate to “c:\users\LabUser\Desktop”
   11. Select “churn.backup”
   12. Click “Restore”
   13. When the restore process is complete, close the informational dialog box
4. Connect to the ***churn*** database
5. Open a query window by clicking “Tools”, then “Query Tool”
6. Click the Open folder button
7. Navigate to “c:\users\LabUser\Desktop”
8. Select “stateregion.sql”
9. Execute the script by clicking the Run triangle button
10. Verify successful completion by clicking on “Schemas”, then “Public”, then “Tables” in the tree view and confirm that the “stateregion” table was created
11. Close ***pgadmin4***

## Dashboard Preparation:

1. Open ***Tableau***
2. Click “File”, then “Open”
3. Navigate to “c:\users\LabUser\Desktop”
4. Select “D211\_churn.twbx”
5. If a Custom SQL Warning dialog is displayed, click “Yes”
6. Enter the appropriate PostgreSQL credentials:
   1. Server: localhost
   2. Port: 5432
   3. Database: churn
   4. Authentication: username and password
   5. Username: postgres
   6. Password: Passw0rd!
7. Click “Sign in”

# A3 – Navigation Instructions

Please follow these steps to have the best experience navigating my dashboard.

1. Click on ‘Churn Dashboard” along the bottom of the Tableau window
2. Enter presentation mode by pressing the <F7> key or clicking the corresponding toolbar button

Additional navigation aids are provided within the dashboard. These can be accessed by clicking the large “Help” button in the upper right corner of the dashboard.

A screenshot of a computer

Description automatically generated

After referencing the help page, viewers may go back to the dashboard by clicking the “Return” button.

A white background with black and white clouds

Description automatically generated with medium confidence

# A4 – SQL Supporting the Dashboard

## SQL to create the *stateregion* table

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

-- Table structure for stateregion

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

DROP TABLE IF EXISTS "public"."stateregion";

CREATE TABLE "public"."stateregion" (

"state" text COLLATE "pg\_catalog"."default",

"region" text COLLATE "pg\_catalog"."default");

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

-- Records of stateregion

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

INSERT INTO "public"."stateregion" VALUES ('ME', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('VT', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('NH', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('MA', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('RI', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('CT', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('NY', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('NJ', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('PA', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('DE', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('MD', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('DC', 'Northeast');

INSERT INTO "public"."stateregion" VALUES ('KY', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('WV', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('VA', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('TN', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('NC', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('AR', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('LA', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('MS', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('AL', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('GA', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('SC', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('FL', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('PR', 'Southeast');

INSERT INTO "public"."stateregion" VALUES ('OH', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('MI', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('IN', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('IL', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('MO', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('KS', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('NE', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('SD', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('ND', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('MN', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('WI', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('IA', 'Midwest');

INSERT INTO "public"."stateregion" VALUES ('OK', 'Southwest');

INSERT INTO "public"."stateregion" VALUES ('TX', 'Southwest');

INSERT INTO "public"."stateregion" VALUES ('NM', 'Southwest');

INSERT INTO "public"."stateregion" VALUES ('AZ', 'Southwest');

INSERT INTO "public"."stateregion" VALUES ('CO', 'Southwest');

INSERT INTO "public"."stateregion" VALUES ('MT', 'West');

INSERT INTO "public"."stateregion" VALUES ('WY', 'West');

INSERT INTO "public"."stateregion" VALUES ('UT', 'West');

INSERT INTO "public"."stateregion" VALUES ('ID', 'West');

INSERT INTO "public"."stateregion" VALUES ('NV', 'West');

INSERT INTO "public"."stateregion" VALUES ('CA', 'West');

INSERT INTO "public"."stateregion" VALUES ('OR', 'West');

INSERT INTO "public"."stateregion" VALUES ('WA', 'West');

INSERT INTO "public"."stateregion" VALUES ('AK', 'West');

INSERT INTO "public"."stateregion" VALUES ('HI', 'West');

## Custom SQL used for Tableau Datasource

SELECT

c.customer\_id,

c.lat,

c.lng,

c.age,

c.marital,

c.gender,

c.tenure,

c.monthly\_charge,

c.income,

c.churn,

l.city,

l.zip,

l.county,

l.state,

s.region,

ct.duration,

p.payment\_type

FROM

customer c

INNER JOIN location l ON l.location\_id = c.location\_id

INNER JOIN stateregion s ON s.state = l.state

INNER JOIN contract ct ON ct.contract\_id = c.contract\_id

INNER JOIN payment p ON p.payment\_id = c.payment\_id

WHERE l.state NOT IN ('AK','HI','PR');