CPSC 471
T01-G05 Final Demo

December 2, 2025
Sam Safe, Aniket Bulusu, Sultan Alzoghaibi

Summary of Project

- ❏ Primary question: Trends in voting turnout and preferences based on ward-level characteristics in the city of Calgary.
- ❏ Data Driven Application
- ❏ Displaying a GeoJSON map with all wards & associated characteristics
- ❏ Users can explore how different ward characteristics influence either voter turnout or winners by selecting the specific factors they want to analyze.
- ❏ A meaningful data visualization will pop up and display the user's inquiries
  - ❏ Does an increase in crime rate have higher voter turnout?
  - ❏ Do wards with more recreational communities have a higher voter turnout?

Prominent Changes

- ❏ Convert relational schema to database schema .sql file
- ❏ Focus on voter turnout
- ❏ Create a loading script in Python that populates schema database in PostgreSQL
  - ❏ Use pgAdmin to do this through GUI as well
  - ❏ Data cleaned for easier access
- ❏ Implemented UI using the Dash framework
- ❏ Add visualizations via Plotly, Pandas, statsmodels
- ❏ Implement geospatial data visualization using Folium, GeoPandas, Shapely
- ❏ Use SQLAlchemy ORM to bake in functions for SQL queries
- ❏ Dockerized the entire project to simplify launch across platforms

Opportunities for Growth | Major Entities & Datasets

OPPORTUNITIES FOR GROWTH:

- Finding a meaningful way of presenting the data instead of just having several visualizations for the user to look at. We took the opportunity to regroup, re-discuss our thoughts of what the user would experience when looking through our project.
- Implemented a GeoJSON map to give users a pleasing visual to have as reference when filtering amongst wards
- Include a dedicated reference section with legible citations giving structure and format towards our final submission

MAJOR ENTITIES & DATASETS

- _Ward_Election_Results was our most important dataset as that contained all data regarding voting by ward and different types of candidates
- All datasets must contain Ward as that was the foreign key to join our tables and have the appropriate data for our visualizations
- The datasets from the milestone demo have not changed at all

Project Progress | Database

❑   Take Open Calgary Data Portal Data → Filter dataset for relevant data

Project Progress | Database Cont.

❏ Do various data mining techniques → Filter through addresses of stops →
Categorize by Ward, Active_Stops, Inactive_Stops, Total_Stops

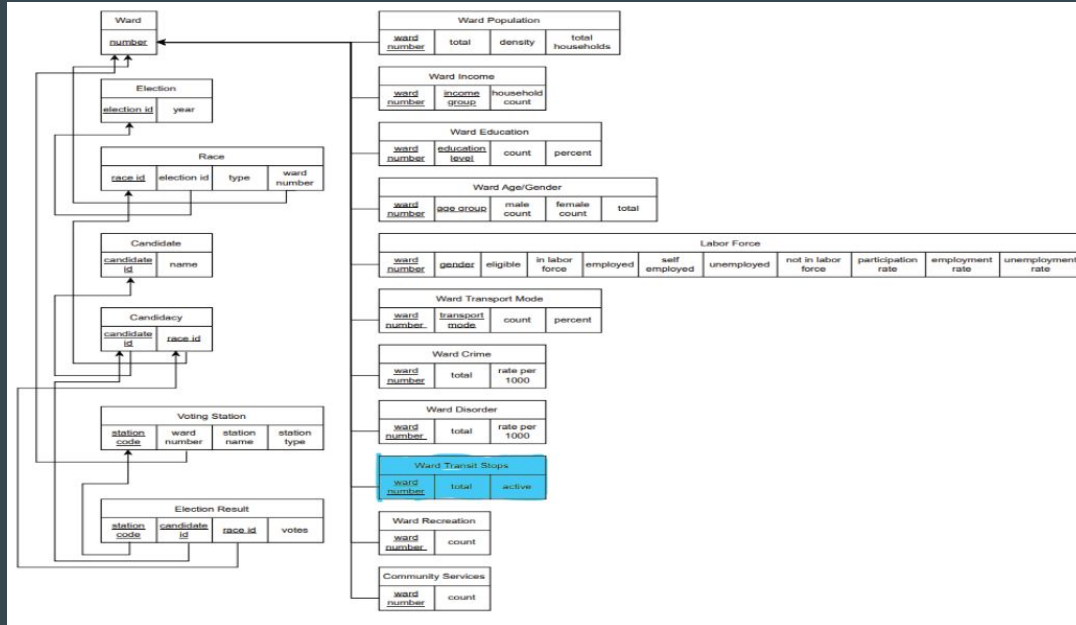| ward_num	active_stop	inactive_st	total_stops
1	452	81	533
2	405	28	433
3	272	18	290
4	527	66	593
5	465	139	604
6	389	72	461
7	456	222	678
8	392	166	558
9	679	212	891
10	495	62	557
11	522	113	635
12	458	112	570
13	292	42	334
14	425	55	480

```sql
CREATE TABLE ward_transit_stops (
    ward_number INTEGER PRIMARY KEY REFERENCES ward(ward_number),
    total INTEGER NOT NULL,
    active INTEGER NOT NULL,
    inactive INTEGER
);
```

```python
# data standardiser
def load_csv(filename):
    df = pd.read_csv(DATA_DIR / filename)
    df.columns = (
        df.columns.str.strip()          You, 2
        .str.lower()
        .str.replace(" ", "_")
        .str.replace("/", "_")
    )
    return df
```

```python
def load_ward_transit_stops(engine):
    print("Loading transport stops...")
    df = load_csv("_Ward_Transit_Stops.csv")
    transit_df = pd.DataFrame ({
        'ward_number': df['ward_num'],
        'total': df['total_stops'],
        'active': df['active_stops'],
        'inactive': df['inactive_stops']
    })
    transit_df.to_sql('ward_transit_stops', engine
```

Project Progress | Database Cont.

❏ Just like the other datasets, everything had gone through a similar process like this and translated to the database schema with Ward Number being a <u>foreign key</u>

Project Progress | Database Cont.

Ward	Attraction	Commerci	Communit	Court	Hospital	Library	PHS Clinic	Social Dev	Visitor Info	Total
1	0	0	5	0	0	2	0	0	0	7
2	0	0	3	0	0	2	1	0	0	6
3	0	0	0	0	0	1	0	0	0	1
4	0	0	12	0	0	2	1	1	0	16
5	2	0	2	0	0	1	0	0	1	6
6	2	0	8	0	0	1	1	0	0	12
7	22	0	13	9	2	2	1	2	3	54
8	5	0	15	0	0	3	1	1	1	26
9	6	2	12	0	0	1	1	3	0	25
10	0	0	9	0	1	1	1	0	0	12
11	1	1	16	0	1	3	1	1	0	24
12	0	0	1	0	1	1	0	0	0	3
13	1	0	3	0	0	1	0	0	0	5
14	1	0	8	0	0	0	1	0	0	10

```
1  "Ward","VotingStationCode","VotingStation","VotingStationType","OfficeType","CandidateName","Votes"
2  "1","101","William D. Pratt School","Regular","MAYOR","IAN CHIANG","3"
3  "1","102","Journey Church","Regular","MAYOR","IAN CHIANG","5"
4  "1","103","Royal Oak School","Regular","MAYOR","IAN CHIANG","5"
5  "1","104","Eric Harvie  School","Regular","MAYOR","IAN CHIANG","5"
6  "1","105","Tuscany School","Regular","MAYOR","IAN CHIANG","2"
7  "1","106","Monsignor E. L. Doyle Elementary School","Regular","MAYOR","IAN CHIANG","1"
8  "1","107","Crowchild Twin Arena Association","Regular","MAYOR","IAN CHIANG","0"
9  "1","108","Silver Springs School","Regular","MAYOR","IAN CHIANG","3"
10 "1","109","F.E. Osborne School","Regular","MAYOR","IAN CHIANG","3"
11 "1","110","Varsity Acres Elementary School","Regular","MAYOR","IAN CHIANG","2"
12 "1","111","Thomas B Riley School","Regular","MAYOR","IAN CHIANG","1"
13 "1","112","Bowness Community Association","Regular","MAYOR","IAN CHIANG","3"
14 "1","113","Crestmont Residents Hall","Regular","MAYOR","IAN CHIANG","0"
15 "1","114","Twelve Mile Coulee School","Regular","MAYOR","IAN CHIANG","0"
16 "1","115","Valley Ridge Golf Course","Regular","MAYOR","IAN CHIANG","4"
17 "1","160","Frank Wills Memorial Hall (CSSD)","Regular","MAYOR","IAN CHIANG","0"
18 "1","161","Cochrane Alliance Church (CSSD)","Regular","MAYOR","IAN CHIANG","0"
19 "1","162","Cochrane Ranche House (CSSD)","Regular","MAYOR","IAN CHIANG","0"
20 "2","201","Bearspaw Christian School","Regular","MAYOR","IAN CHIANG","3"
21 "2","202","Symons Valley United Church","Regular","MAYOR","IAN CHIANG","5"
```

```
CREATE TABLE community_services (
    ward_number INTEGER NOT NULL REFERENCES ward(ward_number),
    service_type VARCHAR(100) NOT NULL,
    count INTEGER NOT NULL,
    PRIMARY KEY (ward_number, service_type)
);
```

- ❏ Election
- ❏ Race (councillor/mayor)
- ❏ Candidate
- ❏ Candidacy (race/candidate)
- ❏ Station
- ❏ Result

Project Progress | Other Technical Aspects

❏ Use references for Dash and Plotly to build an aesthetic application frontend

❏ Dockerized containers for the project with automated entry point script following tutorials and online guides

❏ GeoJSON and data processing pipeline through pandas and WKT

❏ Interactive dashboard with Dash uses Pythonic callbacks to dynamically update visualizations

Supplemental Learning

- ❏ Connecting a Frontend (Dashboard) to our Backend (Data Visualizations & Queries
- ❏ Specific visualization and associated query must be called so the user can see the correct display of what we want to showcase
- ❏ Data pipeline process had to go through various steps (CSV -> cleaned by pandas -> Postgres -> Dash queries using SQL)
- ❏ Learning experience on how data can be connected to a dashboard through different types of methods.
- ❏ Implement GeoJSON map for front-end

Team Dynamics

- ❏ We are communicating well making sure we kept tentative group deadlines so we have that much needed time to tweek any aspects of our project
- ❏ Regarding team meetings, sometimes we get in a little debuggle of how we want to move forward, but we always hear everyone's thoughts and opinions and find a general middle ground of how we want progress in our project
- ❏ We have learned that through peer evaluations, in-person meetups, or team meetings that we all have a voice and that if something is not quite how we envision, we make it vocal and heard so that we can correct those mistakes and have an even more fully fledged and successful project. This was always a very good aspect to have in our team as it always showed that we had an attention to detail in how our workflow was structured and ultimately produce a clean, structured, and successful project.

References

❏ NeuralNine. (2024, September 13). Build Data Apps in Python with Plotly Dash. YouTube. https://www.youtube.com/watch?v=pLU7ZLPhyX8

❏ Colgan, M. (2020, July 7). Developing and Deploying Data-Driven Apps. YouTube. https://www.youtube.com/watch?v=KreNYemH198

❏ Hello dash. Plotly. (n.d.). https://dash.plotly.com/layout

❏ Guides. Docker Documentation. (2024, December 20). https://docs.docker.com/guides/

❏ PostgreSQL 18.1 documentation. PostgreSQL Documentation. (2025, November 13). https://www.postgresql.org/docs/current/index.html

❏ SQLALCHEMY 2.0 documentation. SQLAlchemy Unified Tutorial - SQLAlchemy 2.0 Documentation. (n.d.). https://docs.sqlalchemy.org/en/20/tutorial/index.html

❏ mCoding. (2024, August 2). Docker Tutorial for Beginners. YouTube. https://www.youtube.com/watch?v=b0HMimUb4f0

❏ A practical introduction to databases — a practical introduction to databases. (n.d.). https://runestone.academy/ns/books/published/practical_db/index.html

❏ W3schools.com. W3Schools Online Web Tutorials. (n.d.-a). https://www.w3schools.com/postgresql/index.php

❏ Ark Coding. (n.d.). Streamlit vs Dash - Which one is better? Interactive Dashboard with Python. YouTube. https://www.youtube.com/watch?v=tXHXDRog37
❏ W3Schools. (n.d.). *Matplotlib Pyplot*. Www.w3schools.com. https://www.w3schools.com/python/matplotlib_pyplot.asp

❏ *NumPy Tutorial*. (n.d.). Www.w3schools.com. https://www.w3schools.com/python/numpy/default.asp