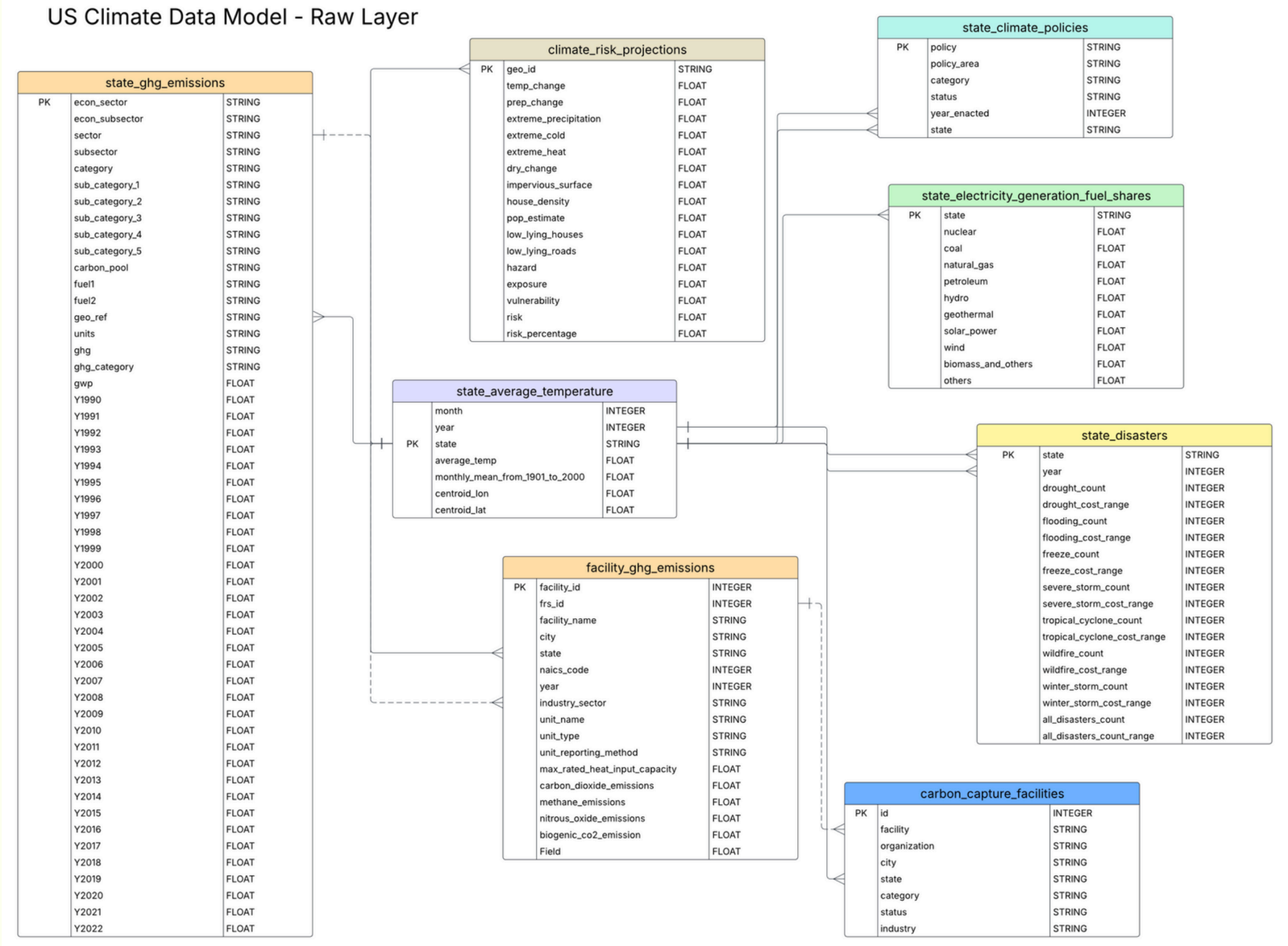
U.S. CLIMATE DATA WAREHOUSE

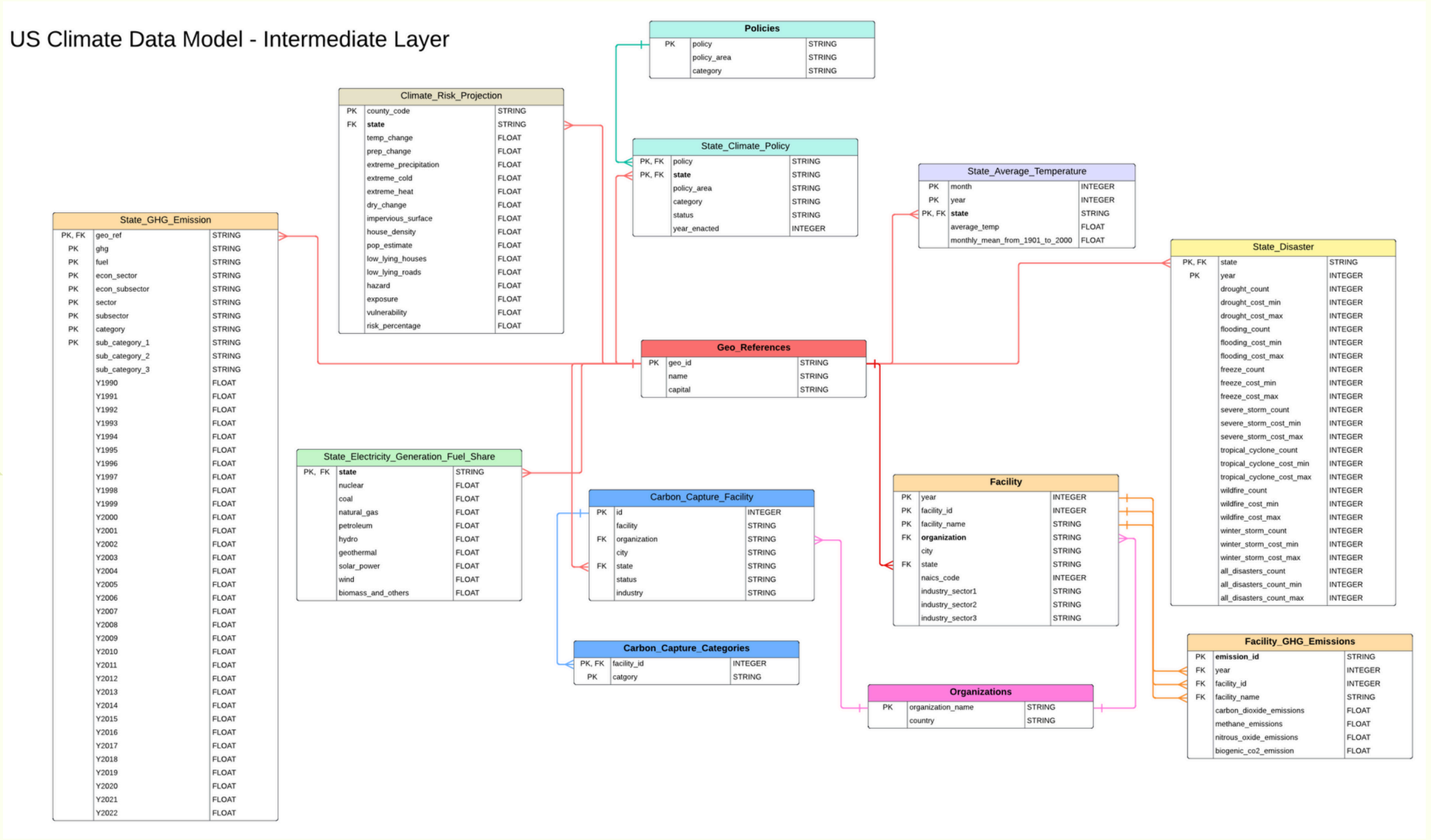
Erica Zhao, Kiara Foght

RAW DATA

- EPA
- Kaggle
- Climate XChange
- NOAA
- NEI
- BBI International
- NASA



INT ERD


US Climate Data Model - Intermediate Layer
Policies
PK policy STRING
policy_area STRING
category STRING
Climate_Risk_Projection
PK county_code STRING
FK state STRING
temp_change FLOAT
prep_change FLOAT
extreme_precipitation FLOAT
extreme_cold FLOAT
extreme_heat FLOAT
dry_change FLOAT
impervious_surface FLOAT
house_density FLOAT
pop_estimate FLOAT
low_lying_houses FLOAT
low_lying_roads FLOAT
hazard FLOAT
exposure FLOAT
vulnerability FLOAT
risk_percentage FLOAT
State_Climate_Policy
PK, FK policy STRING
PK, FK state STRING
policy_area STRING
category STRING
status STRING
year_enacted INTEGER
State_Average_Temperature
PK month INTEGER
PK year INTEGER
PK, FK state STRING
average_temp FLOAT
monthly_mean_from_1901_to_2000 FLOAT
State_GHG_Emission
PK, FK geo_ref STRING
PK ghg STRING
PK fuel STRING
PK econ_sector STRING
PK econ_subsector STRING
PK sector STRING
PK subsector STRING
PK category STRING
PK sub_category_1 STRING
sub_category_2 STRING
sub_category_3 STRING
Y1990 FLOAT
Y1991 FLOAT
Y1992 FLOAT
Y1993 FLOAT
Y1994 FLOAT
Y1995 FLOAT
Y1996 FLOAT
Y1997 FLOAT
Y1998 FLOAT
Y1999 FLOAT
Y2000 FLOAT
Y2001 FLOAT
Y2002 FLOAT
Y2003 FLOAT
Y2004 FLOAT
Y2005 FLOAT
Y2006 FLOAT
Y2007 FLOAT
Y2008 FLOAT
Y2009 FLOAT
Y2010 FLOAT
Y2011 FLOAT
Y2012 FLOAT
Y2013 FLOAT
Y2014 FLOAT
Y2015 FLOAT
Y2016 FLOAT
Y2017 FLOAT
Y2018 FLOAT
Y2019 FLOAT
Y2020 FLOAT
Y2021 FLOAT
Y2022 FLOAT
Geo_References
PK geo_id STRING
name STRING
capital STRING
State_Disaster
PK, FK state STRING
PK year INTEGER
drought_count INTEGER
drought_cost_min INTEGER
drought_cost_max INTEGER
flooding_count INTEGER
flooding_cost_min INTEGER
flooding_cost_max INTEGER
freeze_count INTEGER
freeze_cost_min INTEGER
freeze_cost_max INTEGER
severe_storm_count INTEGER
severe_storm_cost_min INTEGER
severe_storm_cost_max INTEGER
tropical_cyclone_count INTEGER
tropical_cyclone_cost_min INTEGER
tropical_cyclone_cost_max INTEGER
wildfire_count INTEGER
wildfire_cost_min INTEGER
wildfire_cost_max INTEGER
winter_storm_count INTEGER
winter_storm_cost_min INTEGER
winter_storm_cost_max INTEGER
all_disasters_count INTEGER
all_disasters_count_min INTEGER
all_disasters_count_max INTEGER
State_Electricity_Generation_Fuel_Share
PK, FK state STRING
nuclear FLOAT
coal FLOAT
natural_gas FLOAT
petroleum FLOAT
hydro FLOAT
geothermal FLOAT
solar_power FLOAT
wind FLOAT
biomass_and_others FLOAT
Carbon_Capture_Facility
PK id INTEGER
facility STRING
FK organization STRING
city STRING
FK state STRING
status STRING
industry STRING
Facility
PK year INTEGER
PK facility_id INTEGER
PK facility_name STRING
FK organization STRING
city STRING
FK state STRING
naics_code INTEGER
industry_sector1 STRING
industry_sector2 STRING
industry_sector3 STRING
Carbon_Capture_Categories
PK, FK facility_id INTEGER
PK catgory STRING
Organizations
PK organization_name STRING
country STRING
Facility_GHG_Emissions
PK emission_id STRING
FK year INTEGER
FK facility_id INTEGER
FK facility_name STRING
carbon_dioxide_emissions FLOAT
methane_emissions FLOAT
nitrous_oxide_emissions FLOAT
biogenic_co2_emission FLOAT

ORGANIZATION TABLE

Use LLM to Identify Organizations

We used an LLM to identify organizations by querying facilities without an assigned organization and extracting names based on facility details. CCF and GHG facilities were processed separately, and results were stored in a temporary table.

Normalize Organization Names

Next, we normalized organization names by extracting distinct names, using the LLM to standardize variations (e.g., "3M" vs. "3M Company"), and appending country information where possible. The standardized names were stored in a mapping table.

Create the Final Organization Table

We then created the final Organization Table by merging CCF and GHG organizations, removing duplicates, and assigning primary keys for data integrity.

Validate and Finalize the Organization Table

Finally, we validated and finalized the table by ensuring name uniqueness, creating the final BigQuery table, and cleaning up temporary tables to optimize storage.

ORGANIZATION TABLE

```
prompt_ghg = """Given a facility from the GHG emissions dataset:
facility_id, facility_name, city, state, naics_code, industry_sector1,2,3.

Identify the organization that owns or operates this facility, or return null if unknown.
Return EXACTLY one JSON line:
{
  "facility_id": <string>,
  "organization_name": <string or null>
}
No extra text or explanation.
"""
```

01

Prompt to Find Organization

02

Create Check-Point Table

```
batch_size = 50
min_batch_size = 5
sleep_time = 5
max_retries = 5

results_ghg = []
i = 0

while i < len(df_ghg):
    try:
        batch_df = df_ghg.iloc[i:i + batch_size]
        batch_results = [find_org_for_ghg(row) for _, row in batch_df.iterrows()]

        df_batch = pd.DataFrame(batch_results)
        pandas_gbq.to_gbq(
            df_batch,
            "us_climate_int.tmp_ghg_facilities_llm_org_checkpoint",
            project_id=project_id,
            if_exists="append"
        )

        results_ghg.extend(batch_results)
        i += batch_size
        print(f"Processed {i}/{len(df_ghg)} records.")

        time.sleep(sleep_time)

    except GoogleAPIError as e:
        print(f"Quota error encountered: {e}. Retrying with backoff...")

        for retry in range(1, max_retries + 1):
            time.sleep(sleep_time * retry)
            print(f"Retrying (attempt {retry}/{max_retries})...")
            try:
                batch_df = df_ghg.iloc[i:i + batch_size]
                batch_results = [find_org_for_ghg(row) for _, row in batch_df.iterrows()]

                df_batch = pd.DataFrame(batch_results)
                pandas_gbq.to_gbq(
                    df_batch,
                    "us_climate_int.tmp_ghg_facilities_llm_org_checkpoint",
                    project_id=project_id,
                    if_exists="append"
                )

                results_ghg.extend(batch_results)
                i += batch_size
                print(f"Processed {i}/{len(df_ghg)} records after retry.")

                break

            except GoogleAPIError:
                if retry == max_retries:
                    print("Max retries reached. Reducing batch size.")
                    batch_size = max(batch_size // 2, min_batch_size)
                    if batch_size == min_batch_size:
                        print("Minimum batch size reached. Exiting.")
                        break
```

```
sql_orgs = """
SELECT DISTINCT organization_name
FROM us_climate_int.tmp_ccf_facilities_llm_org_checkpoint
WHERE organization_name IS NOT NULL

UNION DISTINCT

SELECT DISTINCT organization
FROM us_climate_stg.carbon_capture_facilities
WHERE organization IS NOT NULL
"""
```

03

Prompt to Normalize Organization

UNIVERSAL IDENTIFIER

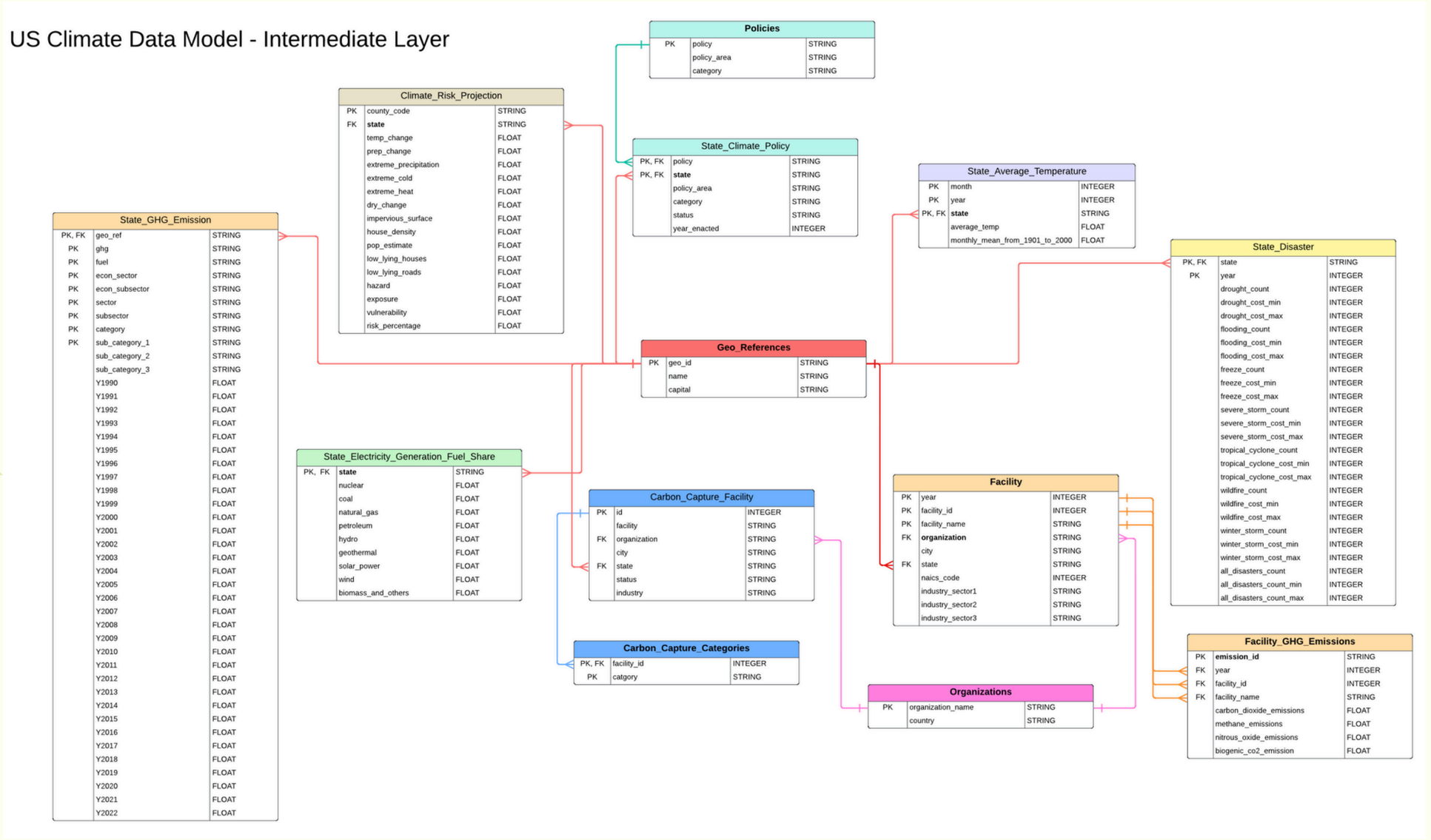
STATE

From the raw data, only saw a way to create relationships between tables through state field.

- Climate Risk Projections table only had county code
- Mix of state abbreviations and spelled out
- Had data for U.S. territories and some Canadian provinces

Created Geo References table using LLM to match state/territories/provinces abbreviation to full name and vice versa. Also used it to find the capitals.

INT ERD


US Climate Data Model - Intermediate Layer
Policies
PK policy STRING
policy_area STRING
category STRING
Climate_Risk_Projection
PK county_code STRING
FK state STRING
temp_change FLOAT
prep_change FLOAT
extreme_precipitation FLOAT
extreme_cold FLOAT
extreme_heat FLOAT
dry_change FLOAT
impervious_surface FLOAT
house_density FLOAT
pop_estimate FLOAT
low_lying_houses FLOAT
low_lying_roads FLOAT
hazard FLOAT
exposure FLOAT
vulnerability FLOAT
risk_percentage FLOAT
State_Climate_Policy
PK, FK policy STRING
PK, FK state STRING
policy_area STRING
category STRING
status STRING
year_enacted INTEGER
State_Average_Temperature
PK month INTEGER
PK year INTEGER
PK, FK state STRING
average_temp FLOAT
monthly_mean_from_1901_to_2000 FLOAT
State_GHG_Emission
PK, FK geo_ref STRING
PK ghg STRING
PK fuel STRING
PK econ_sector STRING
PK econ_subsector STRING
PK sector STRING
PK subsector STRING
PK category STRING
PK sub_category_1 STRING
sub_category_2 STRING
sub_category_3 STRING
Y1990 FLOAT
Y1991 FLOAT
Y1992 FLOAT
Y1993 FLOAT
Y1994 FLOAT
Y1995 FLOAT
Y1996 FLOAT
Y1997 FLOAT
Y1998 FLOAT
Y1999 FLOAT
Y2000 FLOAT
Y2001 FLOAT
Y2002 FLOAT
Y2003 FLOAT
Y2004 FLOAT
Y2005 FLOAT
Y2006 FLOAT
Y2007 FLOAT
Y2008 FLOAT
Y2009 FLOAT
Y2010 FLOAT
Y2011 FLOAT
Y2012 FLOAT
Y2013 FLOAT
Y2014 FLOAT
Y2015 FLOAT
Y2016 FLOAT
Y2017 FLOAT
Y2018 FLOAT
Y2019 FLOAT
Y2020 FLOAT
Y2021 FLOAT
Y2022 FLOAT
Geo_References
PK geo_id STRING
name STRING
capital STRING
State_Disaster
PK, FK state STRING
PK year INTEGER
drought_count INTEGER
drought_cost_min INTEGER
drought_cost_max INTEGER
flooding_count INTEGER
flooding_cost_min INTEGER
flooding_cost_max INTEGER
freeze_count INTEGER
freeze_cost_min INTEGER
freeze_cost_max INTEGER
severe_storm_count INTEGER
severe_storm_cost_min INTEGER
severe_storm_cost_max INTEGER
tropical_cyclone_count INTEGER
tropical_cyclone_cost_min INTEGER
tropical_cyclone_cost_max INTEGER
wildfire_count INTEGER
wildfire_cost_min INTEGER
wildfire_cost_max INTEGER
winter_storm_count INTEGER
winter_storm_cost_min INTEGER
winter_storm_cost_max INTEGER
all_disasters_count INTEGER
all_disasters_count_min INTEGER
all_disasters_count_max INTEGER
State_Electricity_Generation_Fuel_Share
PK, FK state STRING
nuclear FLOAT
coal FLOAT
natural_gas FLOAT
petroleum FLOAT
hydro FLOAT
geothermal FLOAT
solar_power FLOAT
wind FLOAT
biomass_and_others FLOAT
Carbon_Capture_Facility
PK id INTEGER
facility STRING
FK organization STRING
city STRING
FK state STRING
status STRING
industry STRING
Facility
PK year INTEGER
PK facility_id INTEGER
PK facility_name STRING
FK organization STRING
city STRING
FK state STRING
naics_code INTEGER
industry_sector1 STRING
industry_sector2 STRING
industry_sector3 STRING
Carbon_Capture_Categories
PK, FK facility_id INTEGER
PK catgory STRING
Organizations
PK organization_name STRING
country STRING
Facility_GHG_Emissions
PK emission_id STRING
FK year INTEGER
FK facility_id INTEGER
FK facility_name STRING
carbon_dioxide_emissions FLOAT
methane_emissions FLOAT
nitrous_oxide_emissions FLOAT
biogenic_co2_emission FLOAT

THANK YOU