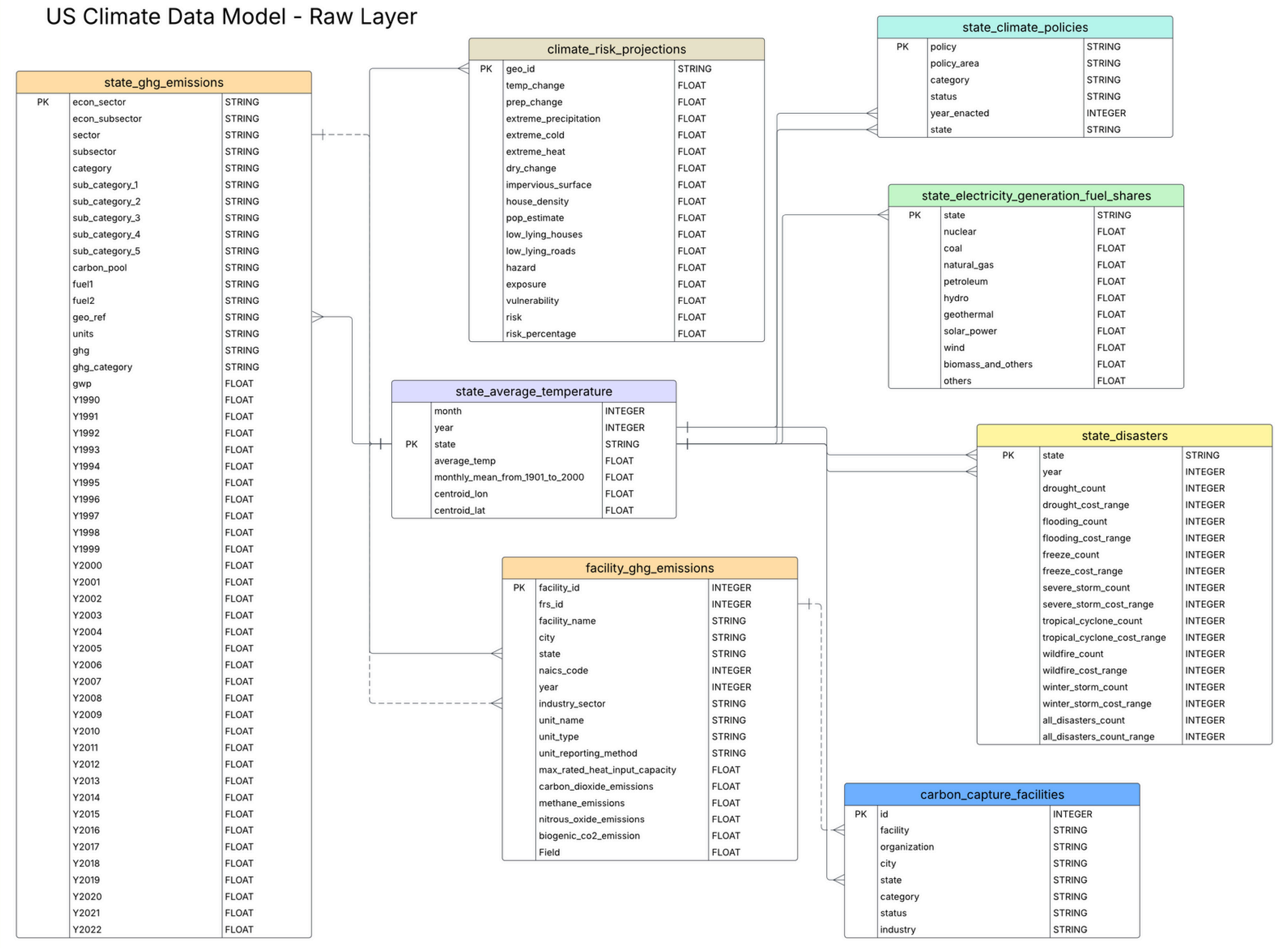
U.S. CLIMATE DATA WAREHOUSE

Erica Zhao, Kiara Foght

RAW DATA

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



US Climate Data Model - Raw Layer

state_ghg_emissions

PK	econ_sector	STRING
	econ_subsector	STRING
	sector	STRING
	subsector	STRING
	category	STRING
	sub_category_1	STRING
	sub_category_2	STRING
	sub_category_3	STRING
	sub_category_4	STRING
	sub_category_5	STRING
	carbon_pool	STRING
	fuel1	STRING
	fuel2	STRING
	geo_ref	STRING
	units	STRING
	ghg	STRING
	ghg_category	STRING
	gwp	FLOAT
	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

climate_risk_projections

PK	geo_id	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	FLOAT
	risk_percentage	FLOAT

state_climate_policies

PK	policy	STRING
	policy_area	STRING
	category	STRING
	status	STRING
	year_enacted	INTEGER
	state	STRING

state_electricity_generation_fuel_shares

PK	state	STRING
	nuclear	FLOAT
	coal	FLOAT
	natural_gas	FLOAT
	petroleum	FLOAT
	hydro	FLOAT
	geothermal	FLOAT
	solar_power	FLOAT
	wind	FLOAT
	biomass_and_others	FLOAT
	others	FLOAT

state_average_temperature

	month	INTEGER
	year	INTEGER
PK	state	STRING
	average_temp	FLOAT
	monthly_mean_from_1901_to_2000	FLOAT
	centroid_lon	FLOAT
	centroid_lat	FLOAT

state_disasters

PK	state	STRING
	year	INTEGER
	drought_count	INTEGER
	drought_cost_range	INTEGER
	flooding_count	INTEGER
	flooding_cost_range	INTEGER
	freeze_count	INTEGER
	freeze_cost_range	INTEGER
	severe_storm_count	INTEGER
	severe_storm_cost_range	INTEGER
	tropical_cyclone_count	INTEGER
	tropical_cyclone_cost_range	INTEGER
	wildfire_count	INTEGER
	wildfire_cost_range	INTEGER
	winter_storm_count	INTEGER
	winter_storm_cost_range	INTEGER
	all_disasters_count	INTEGER
	all_disasters_count_range	INTEGER

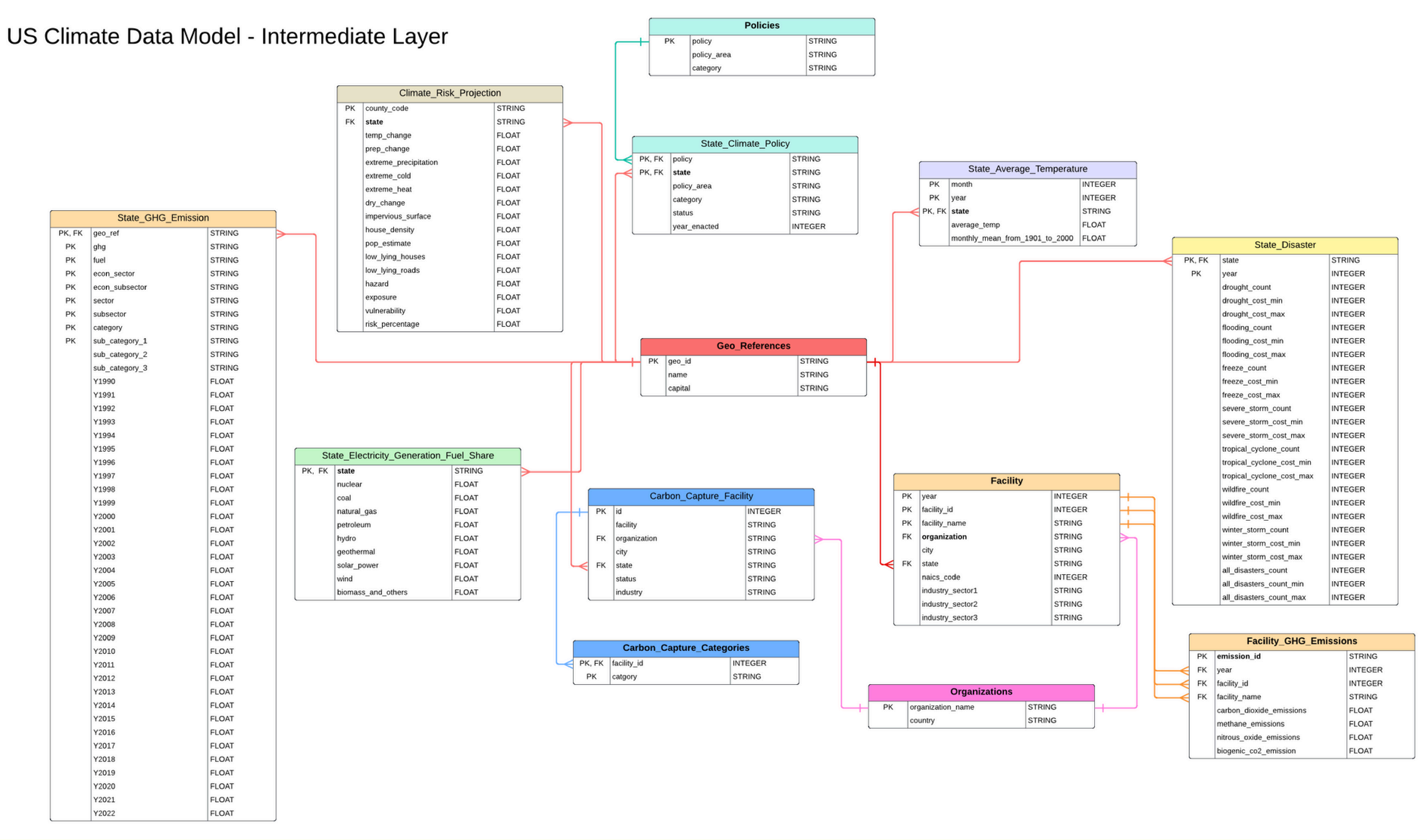
facility_ghg_emissions

PK	facility_id	INTEGER
	frs_id	INTEGER
	facility_name	STRING
	city	STRING
	state	STRING
	naics_code	INTEGER
	year	INTEGER
	industry_sector	STRING
	unit_name	STRING
	unit_type	STRING
	unit_reporting_method	STRING
	max_rated_heat_input_capacity	FLOAT
	carbon_dioxide_emissions	FLOAT
	methane_emissions	FLOAT
	nitrous_oxide_emissions	FLOAT
	biogenic_co2_emission	FLOAT
	Field	FLOAT

carbon_capture_facilities

PK	id	INTEGER
	facility	STRING
	organization	STRING
	city	STRING
	state	STRING
	category	STRING
	status	STRING
	industry	STRING

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

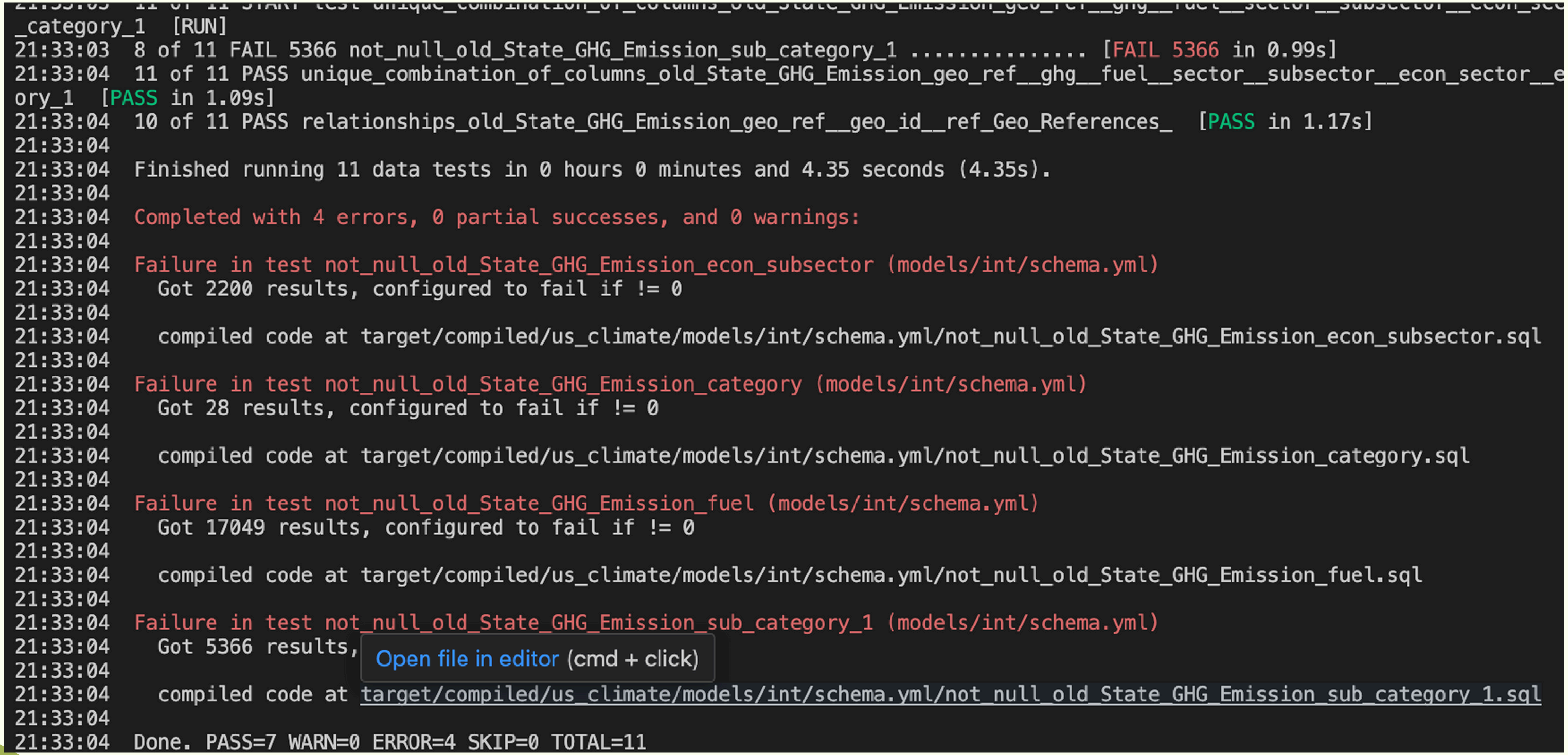
PK	organization_name	STRING
	country	STRING

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

DBT CHALLENGES

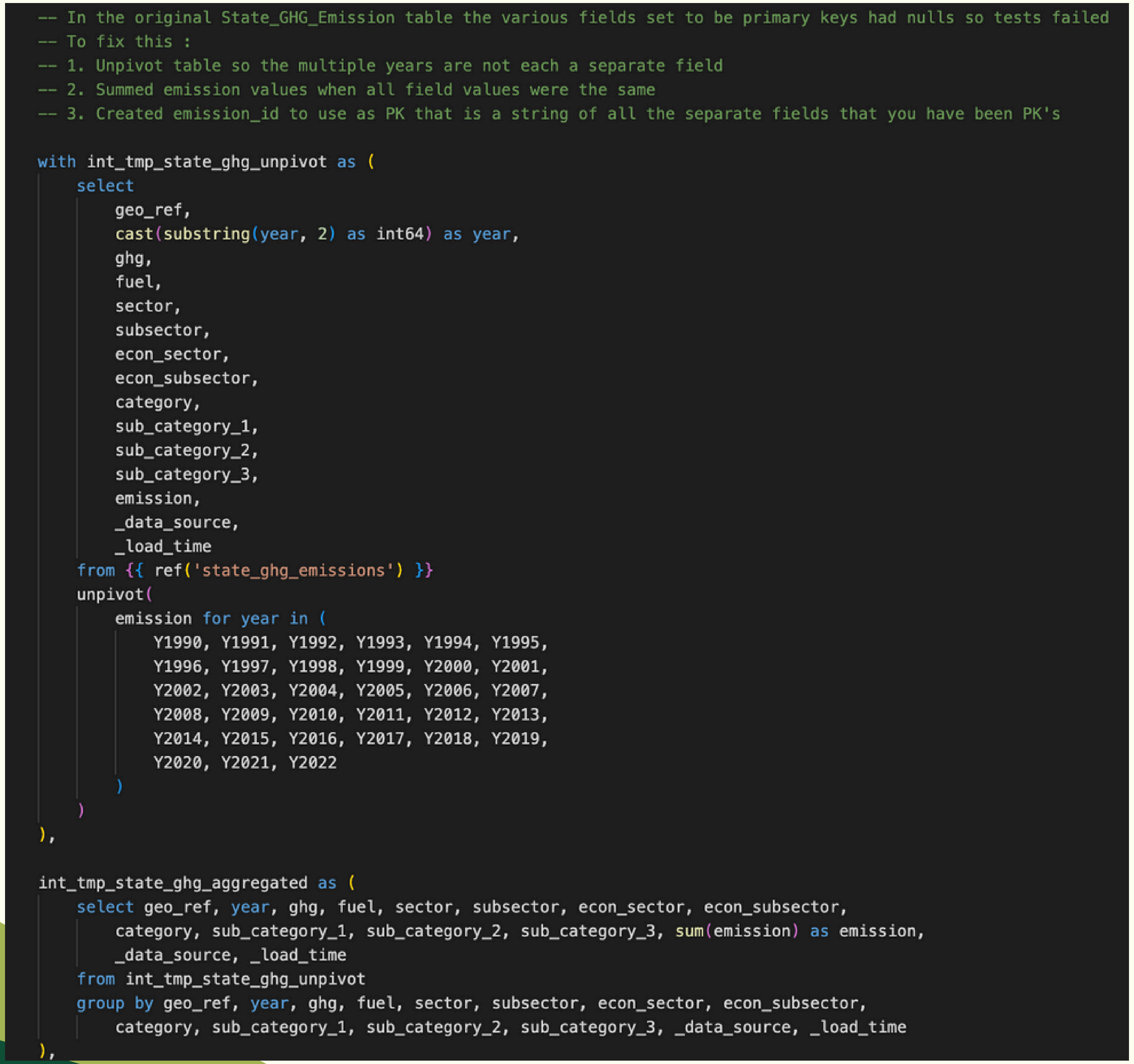
old_State_GHG_Emission



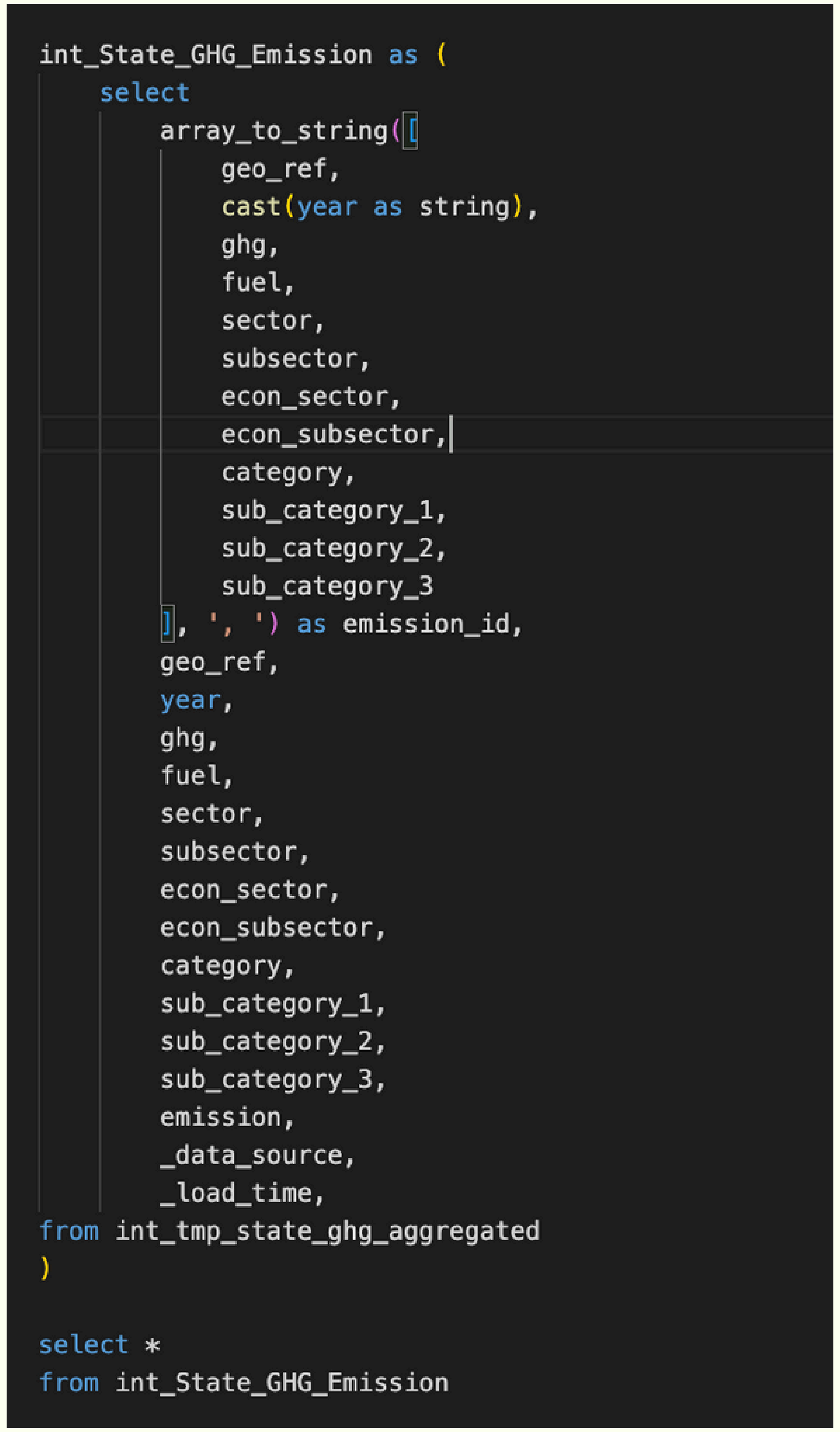
```
21:33:03  11 of 11 START test unique_combination_of_columns_old_State_GHG_Emission_geo_ref__ghg__fuel__sector__subsector__econ_sec
_category_1  [RUN]
21:33:03  8 of 11 FAIL 5366 not_null_old_State_GHG_Emission_sub_category_1 ............... [FAIL 5366 in 0.99s]
21:33:04  11 of 11 PASS unique_combination_of_columns_old_State_GHG_Emission_geo_ref__ghg__fuel__sector__subsector__econ_sector__e
ory_1  [PASS in 1.09s]
21:33:04  10 of 11 PASS relationships_old_State_GHG_Emission_geo_ref__geo_id__ref_Geo_References_  [PASS in 1.17s]
21:33:04
21:33:04  Finished running 11 data tests in 0 hours 0 minutes and 4.35 seconds (4.35s).
21:33:04
21:33:04  Completed with 4 errors, 0 partial successes, and 0 warnings:
21:33:04
21:33:04  Failure in test not_null_old_State_GHG_Emission_econ_subsector (models/int/schema.yml)
21:33:04    Got 2200 results, configured to fail if != 0
21:33:04
21:33:04    compiled code at target/compiled/us_climate/models/int/schema.yml/not_null_old_State_GHG_Emission_econ_subsector.sql
21:33:04
21:33:04  Failure in test not_null_old_State_GHG_Emission_category (models/int/schema.yml)
21:33:04    Got 28 results, configured to fail if != 0
21:33:04
21:33:04    compiled code at target/compiled/us_climate/models/int/schema.yml/not_null_old_State_GHG_Emission_category.sql
21:33:04
21:33:04  Failure in test not_null_old_State_GHG_Emission_fuel (models/int/schema.yml)
21:33:04    Got 17049 results, configured to fail if != 0
21:33:04
21:33:04    compiled code at target/compiled/us_climate/models/int/schema.yml/not_null_old_State_GHG_Emission_fuel.sql
21:33:04
21:33:04  Failure in test not_null_old_State_GHG_Emission_sub_category_1 (models/int/schema.yml)
21:33:04    Got 5366 results,
21:33:04
21:33:04    compiled code at target/compiled/us_climate/models/int/schema.yml/not_null_old_State_GHG_Emission_sub_category_1.sql
21:33:04
21:33:04  Done. PASS=7 WARN=0 ERROR=4 SKIP=0 TOTAL=11
```

DBT FIX

State_GHG_Emission



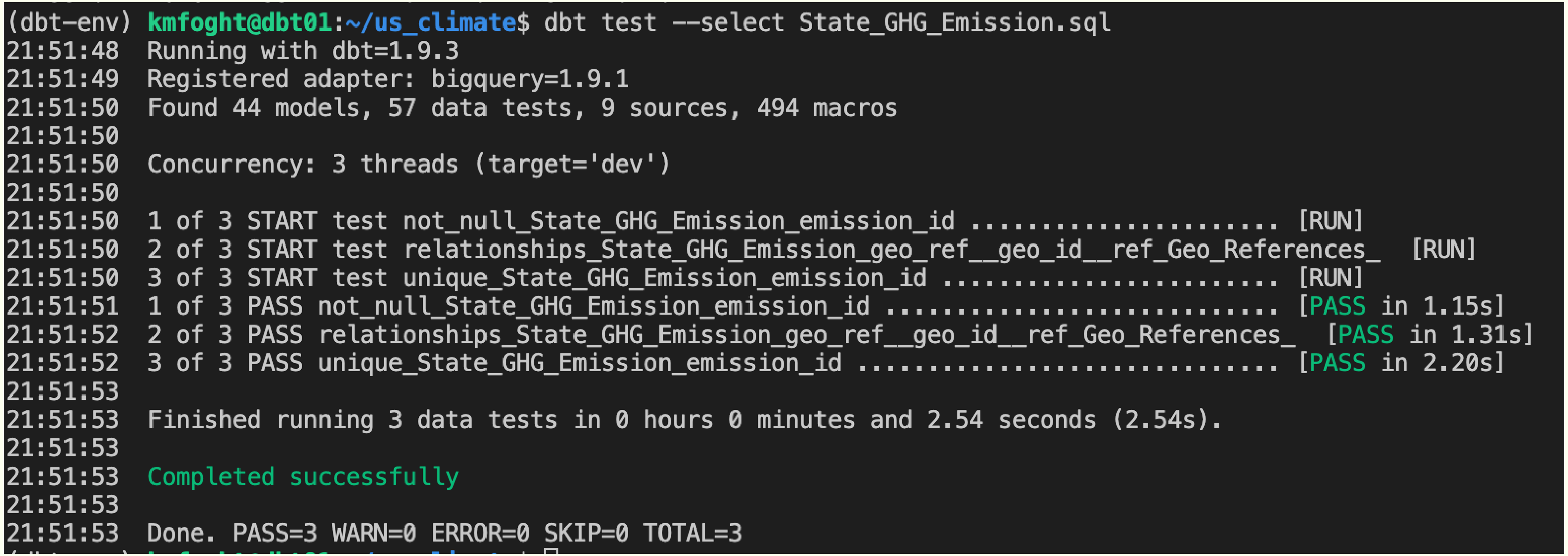
```
-- In the original State_GHG_Emission table the various fields set to be primary keys had nulls so tests failed
-- To fix this :
-- 1. Unpivot table so the multiple years are not each a separate field
-- 2. Summed emission values when all field values were the same
-- 3. Created emission_id to use as PK that is a string of all the separate fields that you have been PK's

with int_tmp_state_ghg_unpivot as (
    select
        geo_ref,
        cast(substring(year, 2) as int64) as year,
        ghg,
        fuel,
        sector,
        subsector,
        econ_sector,
        econ_subsector,
        category,
        sub_category_1,
        sub_category_2,
        sub_category_3,
        emission,
        _data_source,
        _load_time
    from {{ ref('state_ghg_emissions') }}
    unpivot(
        emission for year in (
            Y1990, Y1991, Y1992, Y1993, Y1994, Y1995,
            Y1996, Y1997, Y1998, Y1999, Y2000, Y2001,
            Y2002, Y2003, Y2004, Y2005, Y2006, Y2007,
            Y2008, Y2009, Y2010, Y2011, Y2012, Y2013,
            Y2014, Y2015, Y2016, Y2017, Y2018, Y2019,
            Y2020, Y2021, Y2022
        )
    )
),

int_tmp_state_ghg_aggregated as (
    select geo_ref, year, ghg, fuel, sector, subsector, econ_sector, econ_subsector,
        category, sub_category_1, sub_category_2, sub_category_3, sum(emission) as emission,
        _data_source, _load_time
    from int_tmp_state_ghg_unpivot
    group by geo_ref, year, ghg, fuel, sector, subsector, econ_sector, econ_subsector,
        category, sub_category_1, sub_category_2, sub_category_3, _data_source, _load_time
),
```



```
int_State_GHG_Emission as (
    select
        array_to_string([
            geo_ref,
            cast(year as string),
            ghg,
            fuel,
            sector,
            subsector,
            econ_sector,
            econ_subsector,
            category,
            sub_category_1,
            sub_category_2,
            sub_category_3
        ], ', ') as emission_id,
        geo_ref,
        year,
        ghg,
        fuel,
        sector,
        subsector,
        econ_sector,
        econ_subsector,
        category,
        sub_category_1,
        sub_category_2,
        sub_category_3,
        emission,
        _data_source,
        _load_time,
    from int_tmp_state_ghg_aggregated
)

select *
from int_State_GHG_Emission
```

DBT FIX

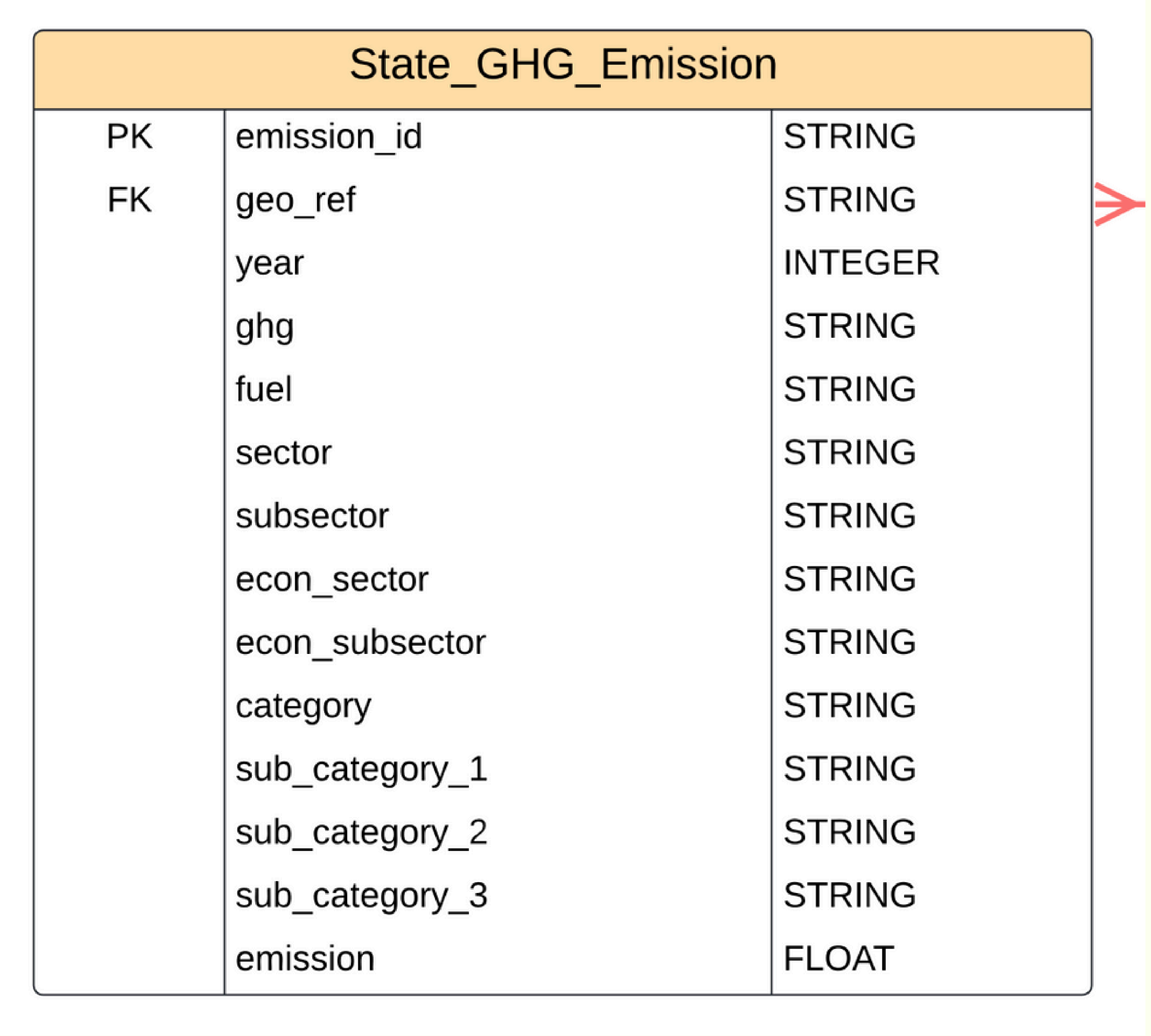
State_GHG_Emission



```
(dbt-env) kmfoght@dbt01:~/us_climate$ dbt test --select State_GHG_Emission.sql
21:51:48  Running with dbt=1.9.3
21:51:49  Registered adapter: bigquery=1.9.1
21:51:50  Found 44 models, 57 data tests, 9 sources, 494 macros
21:51:50
21:51:50  Concurrency: 3 threads (target='dev')
21:51:50
21:51:50  1 of 3 START test not_null_State_GHG_Emission_emission_id ....................... [RUN]
21:51:50  2 of 3 START test relationships_State_GHG_Emission_geo_ref__geo_id__ref_Geo_References_  [RUN]
21:51:50  3 of 3 START test unique_State_GHG_Emission_emission_id ......................... [RUN]
21:51:51  1 of 3 PASS not_null_State_GHG_Emission_emission_id ............................. [PASS in 1.15s]
21:51:52  2 of 3 PASS relationships_State_GHG_Emission_geo_ref__geo_id__ref_Geo_References_  [PASS in 1.31s]
21:51:52  3 of 3 PASS unique_State_GHG_Emission_emission_id ............................... [PASS in 2.20s]
21:51:53
21:51:53  Finished running 3 data tests in 0 hours 0 minutes and 2.54 seconds (2.54s).
21:51:53
21:51:53  Completed successfully
21:51:53
21:51:53  Done. PASS=3 WARN=0 ERROR=0 SKIP=0 TOTAL=3
```

DBT

Final State_GHG_Emission



State_GHG_Emission

PK	emission_id	STRING
FK	geo_ref	STRING
	year	INTEGER
	ghg	STRING
	fuel	STRING
	sector	STRING
	subsector	STRING
	econ_sector	STRING
	econ_subsector	STRING
	category	STRING
	sub_category_1	STRING
	sub_category_2	STRING
	sub_category_3	STRING
	emission	FLOAT

ORGANIZATION TABLE

Use LLM to Identify Organizations

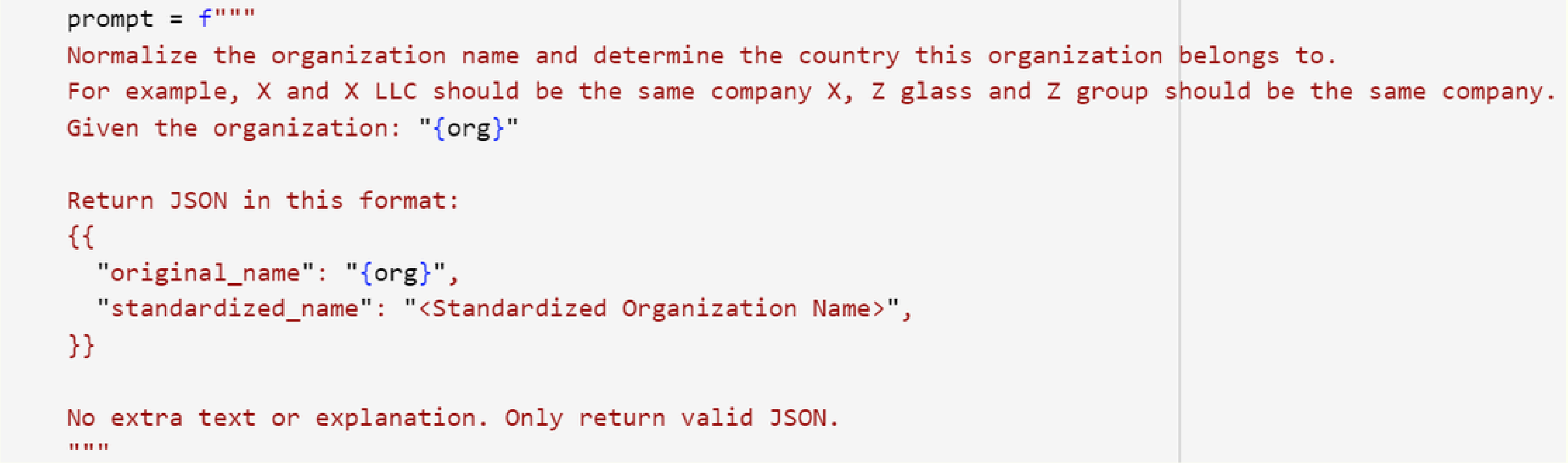
Normalize Organization Names

Create the Final Organization Table

Validate and Finalize the Organization Table

LLM

Used a large language model to semantically evaluate organization names and return the most representative or distinguish between unrelated ones.



```
prompt = f"""
Normalize the organization name and determine the country this organization belongs to.
For example, X and X LLC should be the same company X, Z glass and Z group should be the same company.
Given the organization: "{org}"

Return JSON in this format:
{{
  "original_name": "{org}",
  "standardized_name": "<Standardized Organization Name>",
}}

No extra text or explanation. Only return valid JSON.
"""
```

VS

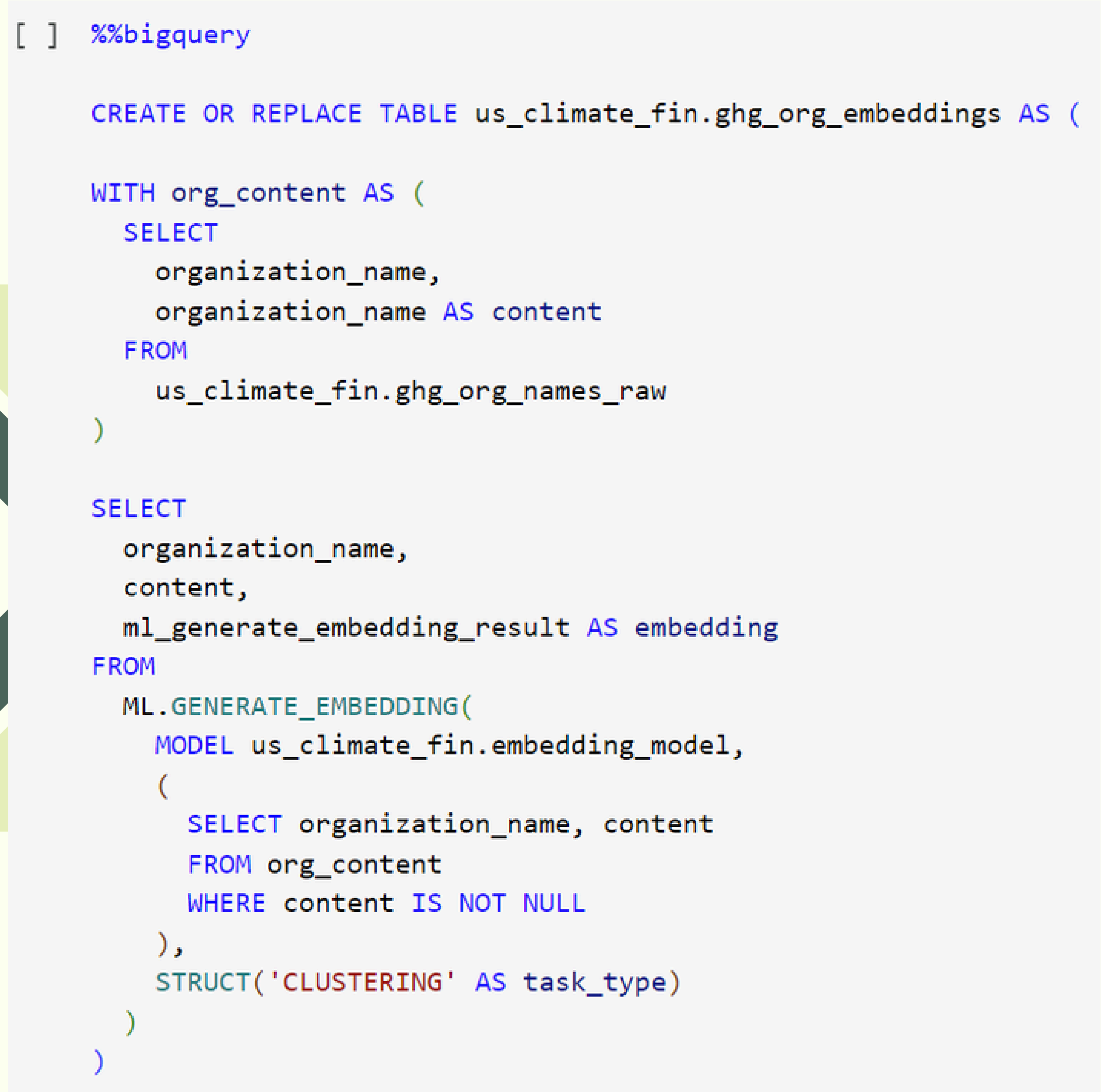
Fuzzy Matching

Converted names into embeddings and grouped similar ones using cosine distance, then reviewed borderline cases with LLM.

FUZZY MATCHING

01

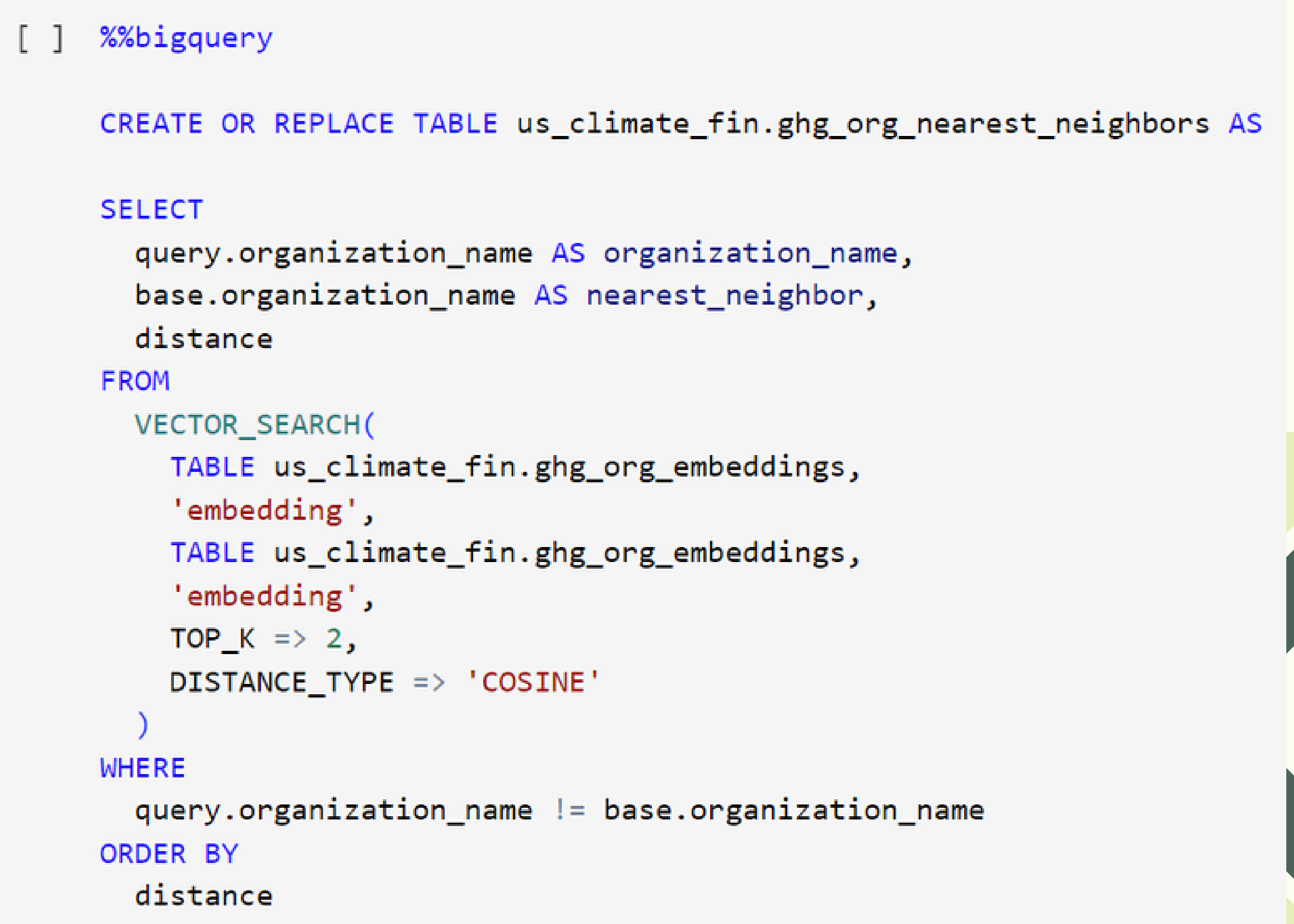
Generate Text Embeddings



```
[ ]  %%bigquery

     CREATE OR REPLACE TABLE us_climate_fin.ghg_org_embeddings AS (

     WITH org_content AS (
       SELECT
         organization_name,
         organization_name AS content
       FROM
         us_climate_fin.ghg_org_names_raw
     )

     SELECT
       organization_name,
       content,
       ml_generate_embedding_result AS embedding
     FROM
       ML.GENERATE_EMBEDDING(
         MODEL us_climate_fin.embedding_model,
         (
           SELECT organization_name, content
           FROM org_content
           WHERE content IS NOT NULL
         ),
         STRUCT('CLUSTERING' AS task_type)
       )
     )
```

02

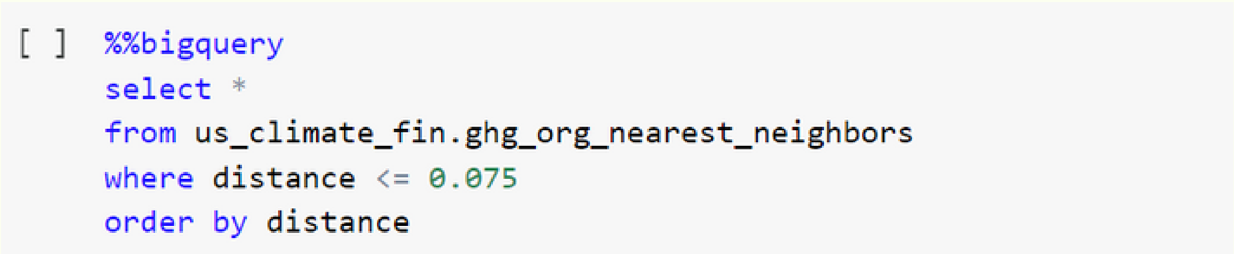
Perform Nearest Neighbor Search



```
[ ]  %%bigquery

     CREATE OR REPLACE TABLE us_climate_fin.ghg_org_nearest_neighbors AS

     SELECT
       query.organization_name AS organization_name,
       base.organization_name AS nearest_neighbor,
       distance
     FROM
       VECTOR_SEARCH(
         TABLE us_climate_fin.ghg_org_embeddings,
         'embedding',
         TABLE us_climate_fin.ghg_org_embeddings,
         'embedding',
         TOP_K => 2,
         DISTANCE_TYPE => 'COSINE'
       )
     WHERE
       query.organization_name != base.organization_name
     ORDER BY
       distance
```

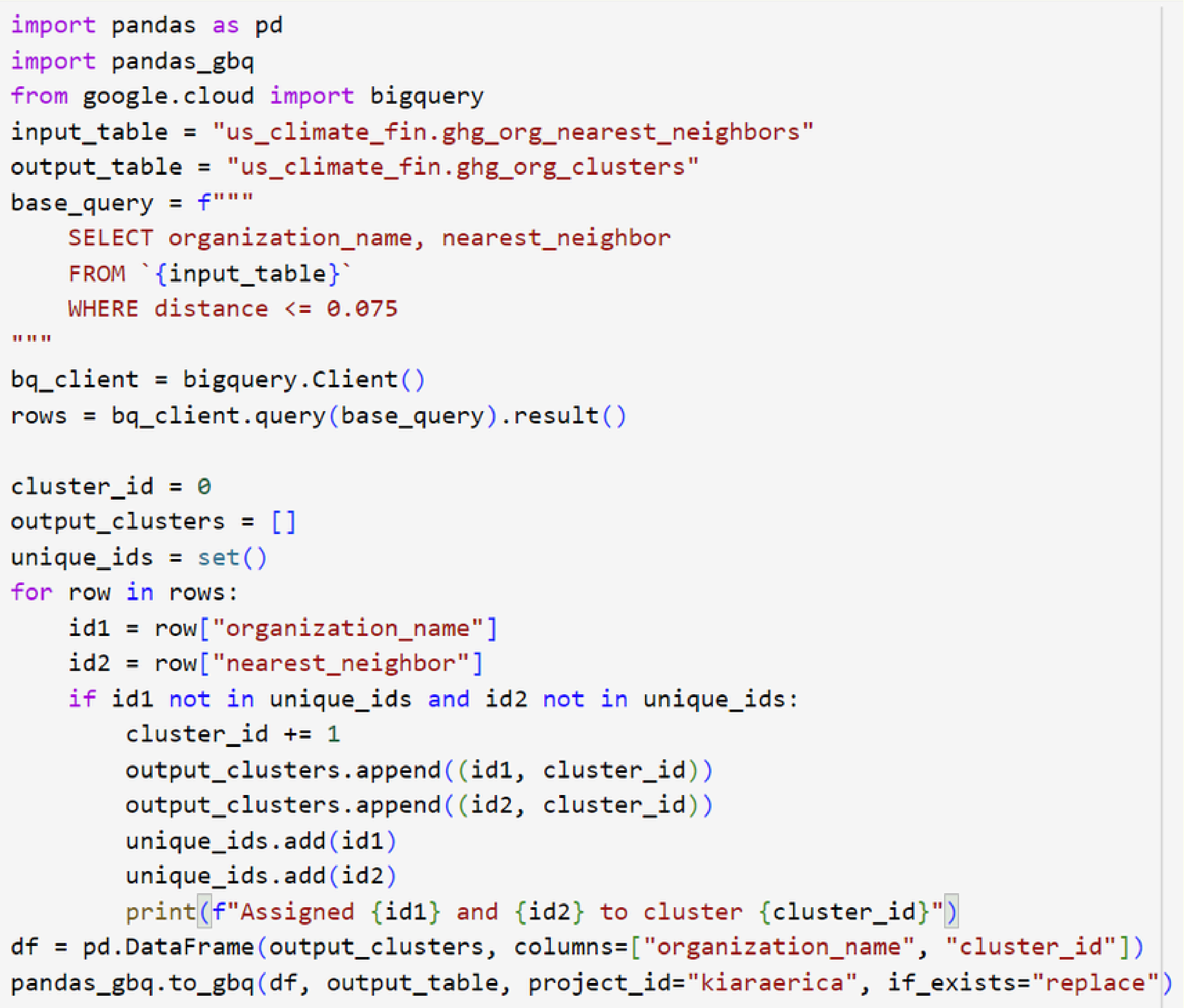
03

Apply a Distance Threshold



```
[ ]  %%bigquery
     select *
     from us_climate_fin.ghg_org_nearest_neighbors
     where distance <= 0.075
     order by distance
```

FUZZY MATCHING



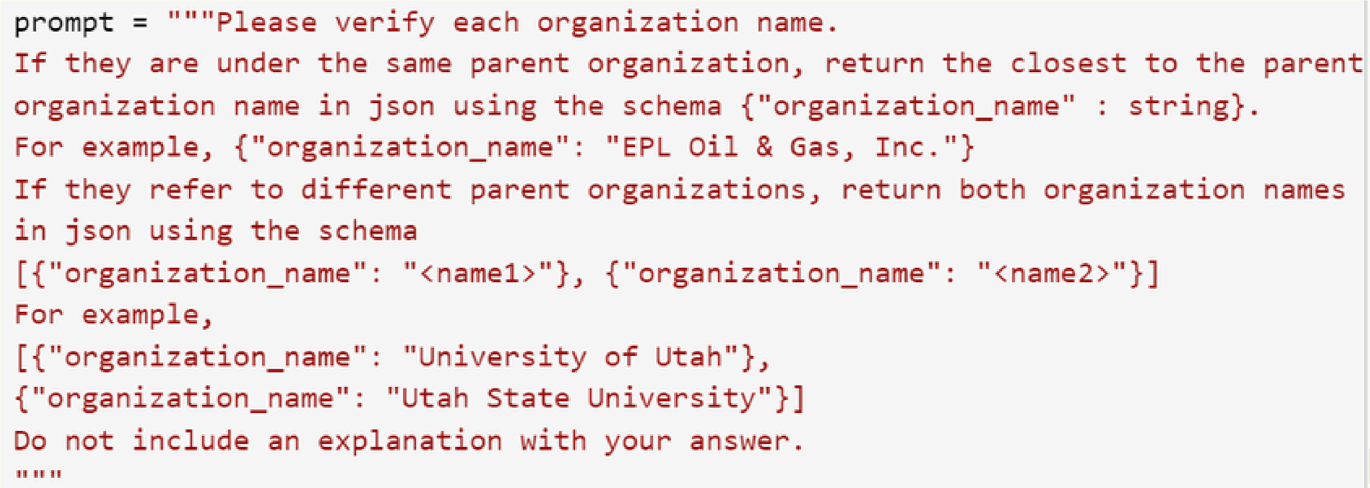
```
import pandas as pd
import pandas_gbq
from google.cloud import bigquery
input_table = "us_climate_fin.ghg_org_nearest_neighbors"
output_table = "us_climate_fin.ghg_org_clusters"
base_query = f"""
    SELECT organization_name, nearest_neighbor
    FROM `{input_table}`
    WHERE distance <= 0.075
"""
bq_client = bigquery.Client()
rows = bq_client.query(base_query).result()

cluster_id = 0
output_clusters = []
unique_ids = set()
for row in rows:
    id1 = row["organization_name"]
    id2 = row["nearest_neighbor"]
    if id1 not in unique_ids and id2 not in unique_ids:
        cluster_id += 1
        output_clusters.append((id1, cluster_id))
        output_clusters.append((id2, cluster_id))
        unique_ids.add(id1)
        unique_ids.add(id2)
        print(f"Assigned {id1} and {id2} to cluster {cluster_id}")
df = pd.DataFrame(output_clusters, columns=["organization_name", "cluster_id"])
pandas_gbq.to_gbq(df, output_table, project_id="kiaraerica", if_exists="replace")
```

04

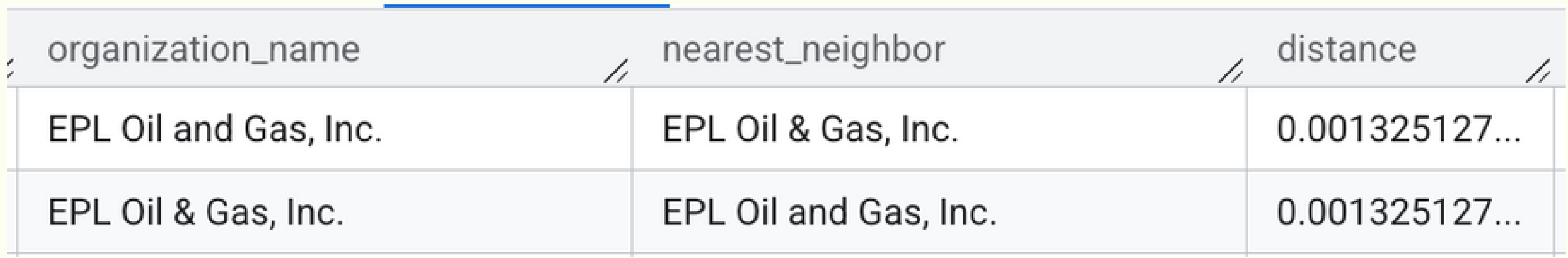
Assign Cluster IDs

05

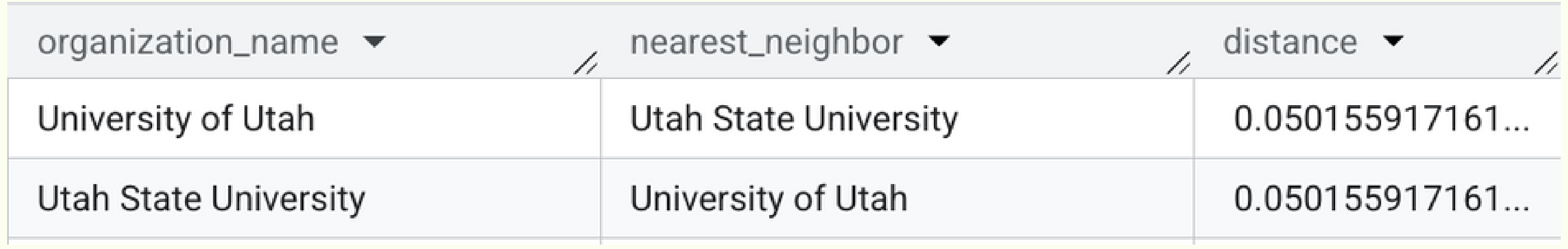
Choose Standardized Name



```
prompt = """Please verify each organization name.
If they are under the same parent organization, return the closest to the parent
organization name in json using the schema {"organization_name" : string}.
For example, {"organization_name": "EPL Oil & Gas, Inc."}
If they refer to different parent organizations, return both organization names
in json using the schema
[{"organization_name": "<name1>"}, {"organization_name": "<name2>"}]
For example,
[{"organization_name": "University of Utah"},
{"organization_name": "Utah State University"}]
Do not include an explanation with your answer.
"""
```



organization_name	nearest_neighbor	distance
EPL Oil and Gas, Inc.	EPL Oil & Gas, Inc.	0.001325127...
EPL Oil & Gas, Inc.	EPL Oil and Gas, Inc.	0.001325127...



organization_name	nearest_neighbor	distance
University of Utah	Utah State University	0.050155917161...
Utah State University	University of Utah	0.050155917161...

COMPARISION

This method captured duplicates missed by traditional string matching, especially those with abbreviations or inconsistent formatting.

This let us group similar names before invoking the LLM.

However, to ensure reasonable recall, we had to use a relatively large distance threshold, which introduced some false positives and still missed a few known duplicates.

Fine-tune the trade-off & Link the normalized organization names back to their corresponding facility IDs

THANK YOU