**UN Dataset and SQL queries**

Created a dataset from the given JSON schema and added party id, project ID columns to each table for making relationship between the tables executing sql queries.

**Query 1)**

Total climate finance provided by each party per year

This query uses a SUM aggregation and a GROUP BY clause to calculate the total climate finance (both bilateral and multilateral) provided by each party for each inventory year. It also uses a JOIN to link the two tables.

SELECT p.party, p.inventory\_year, SUM(p.amount\_climate\_specific\_face\_value) AS total\_bilateral\_finance, SUM(m.amount\_face\_value) AS total\_multilateral\_finance FROM sp\_provided\_bilateral AS p JOIN support\_provided\_via\_multilateral\_channels AS m ON p.party\_id = m.party\_id AND p.inventory\_year = m.inventory\_year GROUP BY p.party, p.inventory\_year ORDER BY p.party, p.inventory\_year;

**Query 2)**

Average amount of mobilized support for Mitigation projects

This query calculates the average mobilized amount for projects specifically tagged for mitigation. It uses an AVG aggregation and a WHERE clause for filtering.

SELECT party, AVG(amount\_mobilized\_face\_value) AS avg\_mobilized\_mitigation\_finance FROM support\_mobilized\_through\_public\_interventions WHERE type\_of\_support = 'Mitigation' GROUP BY party ORDER BY avg\_mobilized\_mitigation\_finance DESC;

**Query 3)**

Total number of projects by type of support and year

This query uses COUNT and GROUP BY on a single table to provide a simple overview of the number of projects classified by their type of support.

SELECT inventory\_year, type\_of\_support, COUNT(project\_id) AS number\_of\_projects FROM tdts\_provided GROUP BY inventory\_year, type\_of\_support ORDER BY inventory\_year, number\_of\_projects DESC;

**Query 4)**

Projects focused on Agriculture in 2022

This query uses a WHERE clause to filter projects based on both the sector and inventory year.

SELECT party, project\_title, type\_of\_support FROM sp\_provided\_bilateral WHERE sector = 'Agriculture' AND inventory\_year = 2022 ORDER BY party;

**Query 5)**

Count of Projects by Sector

This query counts the number of projects needed in each sector

SELECT sector, COUNT(project\_id) AS number\_of\_projects FROM support\_needed\_ftc GROUP BY sector ORDER BY number\_of\_projects DESC;

**Query 6)**

Matching Needed Projects with Received Support This query uses a JOIN to find which projects that a country needed actually received support, linking the support\_needed\_ftc and support\_received\_implementation tables on project\_id and party\_id.

SELECT sn.party, sn.activity\_title AS needed\_activity, sn.estimated\_amount, sr.amount\_received FROM support\_needed\_ftc AS sn JOIN support\_received\_implementation AS sr ON sn.project\_id = sr.project\_id AND sn.party\_id = sr.party\_id WHERE sn.party\_id = 205 -- Brazil ORDER BY sr.amount\_received DESC;

**Query 7)**

Listing all Needed and Received Technology Projects This query uses a JOIN to list all technology transfer projects that were both requested (tdts\_needed) and received (tdts\_received), and sorts them alphabetically by the country

SELECT tn.party, tn.activity\_title AS needed\_activity, tr.activity\_title AS received\_activity, tr.time\_frame, tr.impact\_summary FROM tdts\_needed AS tn JOIN tdts\_received AS tr ON tn.project\_id = tr.project\_id ORDER BY tn.party;