✅ Task 3: Perform a Transformation on BigQuery Data

🎯 Goal

Use the `ST_GEOGPOINT(longitude, latitude)` function to create geographic point data for customers and distribution centers, then calculate the **distance** between each customer and their **closest center**.

🔹 Step 1: Load and Transform the `centers` Table

✅ SQL Query:

```
SQL
-- Load the centers table from public dataset and include geography
transformation
CREATE OR REPLACE TABLE
  `thelook_ecommerce.centers` AS
SELECT
  id,
  name,
  latitude,
  longitude,
  ST_GEOGPOINT(dcenters.longitude, dcenters.latitude) AS
point_location
FROM
  `bigquery-public-data.thelook_ecommerce.distribution_centers` AS
dcenters;
```

✅ Correct Outcome:

- Table `centers` is created in the `thelook_ecommerce` dataset.
- Includes a new column `point_location` of type `GEOGRAPHY`.

📝 Feedback Tip:

If validation fails:

- Ensure the dataset name is **exactly**`thelook_ecommerce`.
- Confirm the column `point_location` is present and correctly typed.

🔹 Step 2: Load and Transform the `customers` Table

✅ SQL Query:

SQL

```
-- Load the customers table from public dataset and include
geography transformation
CREATE OR REPLACE TABLE
  `thelook_ecommerce.customers` AS
SELECT
  id,
  first_name,
  last_name,
  email,
  age,
  gender,
  state,
  street_address,
  postal_code,
  city,
  country,
  traffic_source,
  created_at,
  latitude,
  longitude,
  ST_GEOGPOINT(users.longitude, users.latitude) AS point_location
FROM
  `bigquery-public-data.thelook_ecommerce.users` AS users;
```

Afficher plus de lignes

✅ Correct Outcome:

- Table `customers` is created in the `thelook_ecommerce` dataset.
- Includes a `point_location` column of type `GEOGRAPHY`.

🔷 Step 3: Calculate Distance to Closest Center

✅ SQL Query:

SQL

```
SELECT
  customers.id AS customer_id,
  (
    SELECT
      MIN(ST_DISTANCE(centers.point_location,
customers.point_location)) / 1000
    FROM
      `thelook_ecommerce.centers` AS centers
  ) AS distance_to_closest_center
```

```
FROM
  `thelook_ecommerce.customers` AS customers;
```

✅ Correct Outcome:

- Returns a list of `customer_id` and their **distance in kilometers** to the **nearest distribution center**.
- Uses a **scalar subquery** with `ST_DISTANCE` and `MIN()` to find the closest center.

💾 Step 4: Save the Query

- Click **Save** → **Save query**
- Name: `Calculate Customer Distance to Closest Center` ✅
- Region: Select your region
- Click **Save**

📝 Final Feedback Tips:

- If the **Check my progress** button doesn't validate:
  - Make sure both tables (`customers`, `centers`) are correctly created and populated.
  - Ensure the `ST_GEOGPOINT` function is used correctly.
  - Confirm the query calculates **distance in kilometers** using `/1000`.