

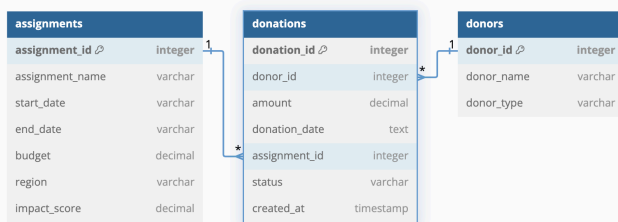


GoodThought NGO has been a catalyst for positive change, focusing its efforts on education, healthcare, and sustainable development to make a significant difference in communities worldwide. With this mission, GoodThought has orchestrated an array of assignments aimed at uplifting underprivileged populations and fostering long-term growth.

This project offers a hands-on opportunity to explore how data-driven insights can direct and enhance these humanitarian efforts. In this project, you'll engage with the GoodThought PostgreSQL database, which encapsulates detailed records of assignments, funding, impacts, and donor activities from 2010 to 2023. This comprehensive dataset includes:

- **Assignments** : Details about each project, including its name, duration (start and end dates), budget, geographical region, and the impact score.
- **Donations** : Records of financial contributions, linked to specific donors and assignments, highlighting how financial support is allocated and utilized.
- **Donors** : Information on individuals and organizations that fund GoodThought's projects, including donor types.

Refer to the below ERD diagram for a visual representation of the relationships between these data tables:



You will execute SQL queries to answer two questions, as listed in the instructions. Good luck!

 Projects Data DataFrame as highest\_donation\_assignments

-- List the top five assignments based on total value of donations, categorized by donor type. The output should include four columns: 1) assignment\_name, 2) region, 3) rounded\_total\_donation\_amount rounded to two decimal places, and 4) donor\_type, sorted by rounded\_total\_donation\_amount in descending order. Save the result as highest\_donation\_assignments.

```
SELECT a.assignment_name,
       a.region,
       ROUND(SUM(d1.amount),2) AS rounded_total_donation_amount,
       d2.donor_type
FROM assignments as a
INNER JOIN donations as d1
USING (assignment_id)
INNER JOIN donors as d2
USING (donor_id)
GROUP BY a.assignment_name, a.region, d2.donor_type
ORDER BY rounded_total_donation_amount DESC
LIMIT 5;
```

index	...	↑↓	assignment_name	...	↑↓	region	...	↑↓	rounded_total_donation_amount
		0	Assignment_3033			East			
		1	Assignment_300			West			
		2	Assignment_4114			North			
		3	Assignment_1765			West			
		4	Assignment_268			East			

Rows: 5

[Expand Table](#)

 Projects Data DataFrame as top\_regional\_impact\_assignments

-- Identify the assignment with the highest impact score in each region, ensuring that each listed assignment has received at least one donation. The output should include four columns: 1) assignment\_name, 2) region, 3) impact\_score, and 4) num\_total\_donations, sorted by region in ascending order. Include only the highest-scoring assignment per region, avoiding duplicates within the same region. Save the result as top\_regional\_impact\_assignments.

```
WITH dc AS (
SELECT assignment_id,
       COUNT(donation_id) AS num_total_donations
FROM donations
GROUP BY assignment_id
),
ranking AS (
SELECT assignment_name,
       region, impact_score, assignment_id,
       ROW_NUMBER() OVER(PARTITION BY region ORDER BY impact_score DESC) AS rank
FROM assignments
)
SELECT ranking.assignment_name, ranking.region, ranking.impact_score, dc.num_total_donations
FROM dc
JOIN ranking
USING (assignment_id)
WHERE ranking.rank = 1
GROUP BY ranking.region, ranking.assignment_name, ranking.impact_score, dc.num_total_donations
```

...	↑↓	assignme...	...	↑↓	...	↑↓	impa...	...	↑↓	num_total_donat...	...	↑↓
	0	Assignment_316			East		10		2			
	1	Assignment_2253			North		9.99		1			
	2	Assignment_3547			South		10		1			
	3	Assignment_3764			West		9.99		1			

Rows: 4

[Expand Table](#)