

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 df

```
SELECT *
FROM public.assignments
LIMIT 5;
```

...	↑↓	assign...	...	↑↓	assignme...	...	↑↓	start_date	...	↑↓	end_date	...	↑↓	b	...	↑↓	...	↑↓	impa...	...
	0			1	Assignment_1			2021-10-17T00:00:00.000			2021-12-04T00:00:00.000			-32322.03			West			
	1			2	Assignment_2			2020-10-26T00:00:00.000			2020-11-28T00:00:00.000			57278.4			South			
	2			3	Assignment_3			2021-08-11T00:00:00.000			2022-03-17T00:00:00.000			40414.51			West			
	3			4	Assignment_4			2021-11-22T00:00:00.000			2022-05-17T00:00:00.000			31732.48			East			
	4			5	Assignment_5			2020-11-22T00:00:00.000			2021-07-10T00:00:00.000			13548.22			North			

Rows: 5

Projects Data DataFrame as df1

```
SELECT *
FROM public.donations
LIMIT 5;
```

i.	...	↑↓	donatio...	...	↑↓	d...	...	↑↓	c	...	↑↓	donation_date	...	↑↓	assignment_...	...	↑↓	
	0				1			2733			271.36	2021-08-21T00:00:00.000					4226	
	1				2			2608			251.49	2021-10-15T00:00:00.000					1323	
	2				3			1654			528.38	2020-03-03T00:00:00.000					4881	
	3				4			3265			730.36	2021-02-06T00:00:00.000					1237	
	4				5			4932			285.96	2022-03-05T00:00:00.000					1626	

Rows: 5

Projects Data DataFrame as df2

```
SELECT *
FROM public.donors
LIMIT 5;
```

index	...	↑↓	donor_id	...	↑↓	donor_name	...	↑↓	donor_type	...	↑↓
		0			1	Donor_1			Individual		
		1			2	Donor_2			Organization		
		2			3	Donor_3			Individual		
		3			4	Donor_4			Organization		
		4			5	Donor_5			Organization		

Rows: 5

Projects Data DataFrame as highest_donation_assignments

```
-- highest_donation_assignments
SELECT
  a.assignment_name,
  a.region,
  ROUND(SUM(d.amount), 2) AS rounded_total_donation_amount,
  dr.donor_type
FROM assignments AS a
```

```
INNER JOIN donations AS d
USING(assignment_id)
INNER JOIN donors AS dr
USING(donor_id)
GROUP BY assignment_name, region, donor_type
ORDER BY rounded_total_donation_amount DESC
LIMIT 5;
```

i	...	↑↓	assignment_na...	...	↑↓	r	...	↑↓	rounded_total_donation_amount	...	↑↓	donor_type	...	↑↓
	0		Assignment_3033			East			3840.66			Individual		
	1		Assignment_300			West			3133.98			Organization		
	2		Assignment_4114			North			2778.57			Organization		
	3		Assignment_1765			West			2626.98			Organization		
	4		Assignment_268			East			2488.69			Individual		

Rows: 5

Projects Data DataFrame as top_regional_impact_assignments

```
-- top_regional_impact_assignments
SELECT
    a.assignment_name,
    a.region,
    a.impact_score,
    d.num_total_donations
FROM assignments AS a
INNER JOIN (
    SELECT assignment_id, COUNT(*) AS num_total_donations
    FROM donations
    WHERE amount <> 0
    GROUP BY assignment_id
) AS d
USING(assignment_id)
INNER JOIN (
    SELECT assignment_id, ROW_NUMBER() OVER(PARTITION BY region ORDER BY impact_score DESC) AS rank
    FROM assignments
) AS sub_rank
USING(assignment_id)
WHERE
    rank = 1
ORDER BY region ASC
```

...	↑↓	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