

PROJECT: IMPACT ANALYSIS OF GOODTHOUGHT NGO INITIATIVES



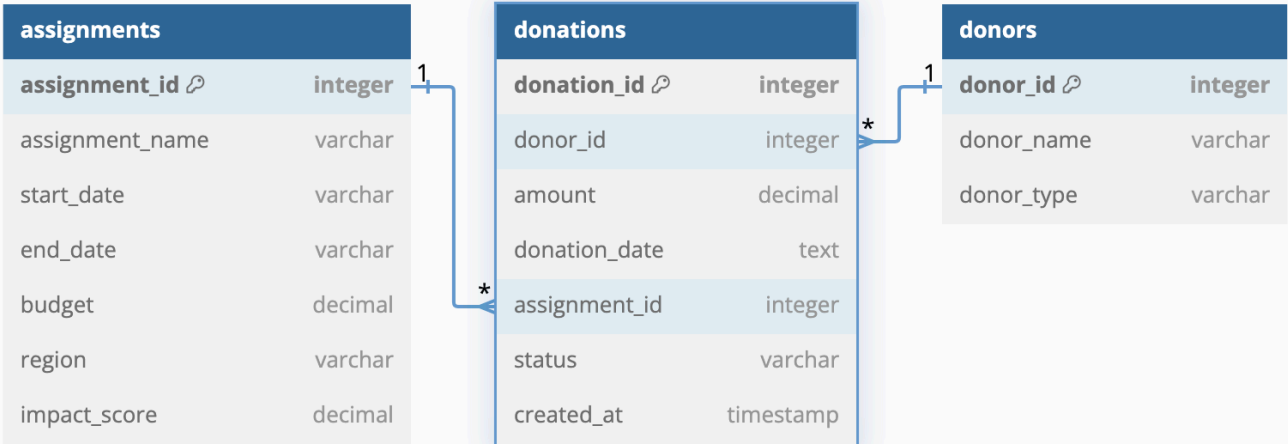


GoodThought NGO has been a catalyst for positive change, focusing its efforts on education, healthcare, and sustainable development to make a significant difference in communities worldwide. With this mission, GoodThought has orchestrated an array of assignments aimed at uplifting underprivileged populations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

```
-- Identifying the top five assignments based on total value of donations categorized by donor type highest_donation_assignments

--- Declaring the CTE
WITH cte AS (
    SELECT public.donors.donor_type, public.donations.assignment_id, SUM(amount) AS total_amount
    FROM public.donors
    INNER JOIN public.donations
    ON public.donations.donor_id = public.donors.donor_id
    GROUP BY public.donors.donor_type, public.donations.assignment_id
)

--- Selecting data
SELECT public.assignments.assignment_name, public.assignments.region, ROUND(cte.total_amount, 2) AS rounded_total_donation_amount, cte.donor_type
FROM public.assignments

--- Joining tables
INNER JOIN cte
    ON public.assignments.assignment_id = cte.assignment_id
ORDER BY rounded_total_donation_amount DESC
LIMIT 5
```

| ... | ↑↓ | assignment_name | ... | ↑↓ | region | ... | ↑↓ | 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

 Expand

 Projects Data DataFrame as `top_regional_impact_assignments`

```
-- Identifying the top assignments by impact in each region top_regional_impact_assignments

-- Building the first CTE to calculate the total number of donations received for each assignment,
WITH donation_counts AS (
  SELECT
    assignment_id,
    COUNT(donation_id) AS num_total_donations
  FROM
    donations
  GROUP BY
    assignment_id
),

-- Building the second CTE to rank assignments within each region
ranked_assignments AS (
  SELECT
    a.assignment_name,
    a.region,
    a.impact_score,
    dc.num_total_donations,
    ROW_NUMBER() OVER (PARTITION BY a.region ORDER BY a.impact_score DESC) AS rank_in_region
  FROM
    assignments a
  JOIN
    donation_counts dc ON a.assignment_id = dc.assignment_id
  WHERE
    dc.num_total_donations > 0
)
SELECT
  assignment_name,
  region,
  impact_score,
  num_total_donations
FROM
  ranked_assignments
WHERE
  rank_in_region = 1
ORDER BY
  region ASC;
```

| ... | ↑↓ | assignment_name | ... | ↑↓ | region | ... | ↑↓ | impact_score | ... | ↑↓ | num_total_donations | ... | ↑↓ |
|-----|----|-----------------|-----|----|--------|-----|----|--------------|-----|----|---------------------|-----|----|
| | 0 | Assignment_316 | | | East | | | 10 | | | 2 | | |
| | 1 | Assignment_2253 | | | North | | | 9.99 | | | 1 | | |
| | 2 | Assignment_3547 | | | South | | | 10 | | | 1 | | |
| | 3 | Assignment_2794 | | | West | | | 9.99 | | | 2 | | |

 Projects Data DataFrame as `df`

```
-- Initial answer
-- Identifying the top assignments by impact in each region top_regional_impact_assignments
-- Building the first CTE to calculate the total number of donations received for each assignment,
WITH cte1 AS (
  SELECT public.assignments.assignment_name, public.assignments.assignment_id, COUNT(public.donations.donation_id) AS num_total_donations
  FROM public.assignments
  INNER JOIN public.donations
  ON public.assignments.assignment_id = public.donations.assignment_id
  GROUP BY public.assignments.assignment_id
),

-- Building the second CTE to rank assignments within each region
cte2 AS (
  SELECT public.assignments.region, public.assignments.assignment_id, RANK() OVER
  (PARTITION BY public.assignments.region
  ORDER BY public.assignments.impact_score DESC) AS assignment_rank
  FROM public.assignments
)

-- Main SQL Query
SELECT a.assignment_name, a.region, a.impact_score, c1.num_total_donations
FROM public.assignments a
JOIN cte1 c1 ON a.assignment_id = c1.assignment_id
JOIN cte2 c2 ON a.assignment_id = c2.assignment_id
WHERE c2.assignment_rank = 1
ORDER BY a.region;
```

| in... | ... | ↑↓ | assignment_name | ... | ↑↓ | region | ... | ↑↓ | impact_score | ... | ↑↓ | num_total_donations | ... | ↑↓ | |
|-------|-----|----|-----------------|-----|----|--------|-----|----|--------------|-----|----|---------------------|-----|----|--|
| | | 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 | | | |
| | | 4 | Assignment_2794 | | | West | | | 9.99 | | | 2 | | | |

Rows: 5

 Expand