QUANTIC UNIVERSITY

PRINCIPLES OF BUSINESS ANALYTICS PROJECT

-

Activity Tracking & Donor Reporting - A Relational Data Solution for INSO in Somalia

Prepared by: Ilyas Abdilahi Abdirahman

Email: Ilyaas1st@gmail.com

Date: 27 September 2025

Introduction

This report presents a practical data management solution for INSO's recurring briefings across cities in Somalia. The current process depends on spreadsheets. It causes late submissions, duplicate counting, and missing evidence for donor reports. I have thought implementation of a small relational database in PostgreSQL that organizes events, attendance, submissions, and late uploads of required documents. This design reduces errors, improves timeliness, and speeds up quarterly reporting.

Business Scenario & Client Context

INSO runs three types of briefings: Security Roundtables, Orientation Briefings, and Country Directors' Briefings. These take place in Hargeisa, Mogadishu, Garowe, Kismayo, Baidoa, Bosaso, Beletweyne, Erigavo, and Burco. Mogadishu, Hargeisa, and Garowe hold bi-weekly security roundtables. The other cities hold monthly sessions. Country Directors' Briefing is monthly, and Orientation Briefing is bi-monthly. Attendance normally ranges from 10 to 35 participants.

Donor agreements require a quarterly report with evidence. That evidence usually includes a signed attendance list, slides, and minutes. When each office sends separate spreadsheets and files, it is hard to consolidate and validate quickly. The new database provides one source of truth, so leadership can check progress and quality in near real time.

Problems & Root Causes

I. Duplicate work and double counting because each office maintains its own files.
II. Late submissions due to email delays and unclear templates.
III. Missing evidence files or wrong file names, which slow donor reporting.
IV. Manual aggregation for quarterly reports, which takes days and is error prone.

Objectives and Success Criteria

Objectives:

a) Centralize event, attendance, and evidence in one database.

b) Track timeliness against a 7-day from event to first submission.

c) Monitor completeness of required documents for each submission.

d) Produce donor-ready tables grouped by city, period, and meeting type.

Success criteria:

- Higher on-time submission rate each quarter.
- Fewer missing documents before the donor deadline.
- Reporting effort reduced from days to hours.

Data source and selection

I am going to use hybrid simulated data for location names and public partner lists. Internal data (events, attendance, and uploads) is entered by focal persons or loaded from simple spreadsheets.

Staging inputs:

a. Cities and place names for Somalia (used to validate the location master).

b. Partner lists to standardize organization names and types (LNGO, INGO, UN, Donor).

c. Operational sheets that record event dates, attendance, and filenames for evidence.

Relational Schema

The schema has master tables for location, meeting_type, period_type, partner_type, partner_status, and required_doc_type. Organization and participant tables link to these masters. The event table stores the core activity. Attendance is a many-to-many link between events and participants. The submission table records the donor package for an event, and the upload table stores each required document linked to the submission. An event_ingest_log keeps the filename that created an event.

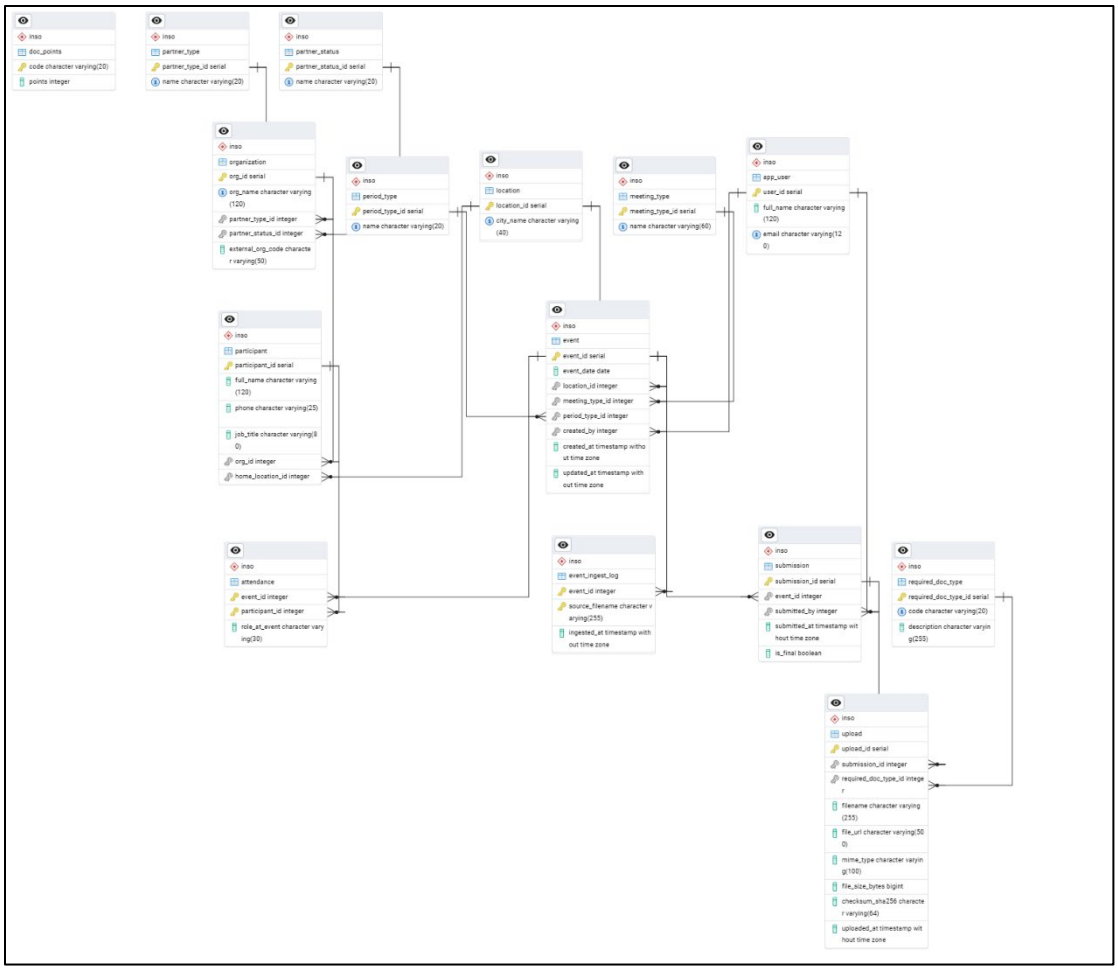
Important constraints and rules:

- Foreign keys keep values valid across tables.
- A unique pair (submission_id, required_doc_type_id) prevents duplicate uploads for the same type.

Here is the link of the data definition language file in GitHub and first page snip;

```sql
-- Create the schema and tables in a single code block
CREATE SCHEMA inso;

-- Master / Reference Tables

CREATE TABLE inso.location (
    location_id     SERIAL PRIMARY KEY,
    city_name       VARCHAR(40) NOT NULL UNIQUE
    -- e.g., Hargeisa, Mogadishu, Garowe, Kismayo, Baidoa, Bosaso, Beletweyne, Erigavo, Burco
);

CREATE TABLE inso.meeting_type (
    meeting_type_id SERIAL PRIMARY KEY,
    name            VARCHAR(60) NOT NULL UNIQUE
    -- e.g., Security Roundtable, Orientation Briefing, Country Directors Briefing
);

CREATE TABLE inso.period_type (
    period_type_id  SERIAL PRIMARY KEY,
    name            VARCHAR(20) NOT NULL UNIQUE
    -- expected values: weekly, bi-weekly, monthly, bi-monthly
);

CREATE TABLE inso.partner_type (
    partner_type_id SERIAL PRIMARY KEY,
    name            VARCHAR(20) NOT NULL UNIQUE
    -- LNGO, INGO, UN, Donor
);

CREATE TABLE inso.partner_status (
    partner_status_id SERIAL PRIMARY KEY,
    name              VARCHAR(20) NOT NULL UNIQUE
    -- Registered, Unregistered
);

-- Organizations & People

CREATE TABLE inso.organization (
    org_id            SERIAL PRIMARY KEY,
    org_name          VARCHAR(120) NOT NULL,
    partner_type_id   INT REFERENCES inso.partner_type (partner_type_id) ON DELETE SET NULL,
    partner_status_id INT REFERENCES inso.partner_status (partner_status_id) ON DELETE SET NULL,
    external_org_code VARCHAR(50),
    UNIQUE (org_name)
);

CREATE TABLE inso.participant (
    participant_id   SERIAL PRIMARY KEY,
    full_name        VARCHAR(120) NOT NULL,
    phone            VARCHAR(25),
    job_title        VARCHAR(80),
    org_id           INT REFERENCES inso.organization (org_id) ON DELETE SET NULL,
    home_location_id INT REFERENCES inso.location (location_id) ON DELETE SET NULL
);

CREATE TABLE inso.app_user (
    user_id   SERIAL PRIMARY KEY,
    full_name VARCHAR(120) NOT NULL,
    email     VARCHAR(120) NOT NULL UNIQUE
    -- focal persons entering data
);

```

https://github.com/Ilyas1st/MSBA_Principles_of_BA_project



Implementation

The database is created from one Data Definition language DDL file. Another data file adds Somali cities, organizations, and sample events.

Then used ETL load masters, load organizations and participants, then events, attendance, submissions, and uploads.

Data quality checks:

a. Required fields are not null (e.g., event_date, city, meeting_type).
b. Valid code values for meeting and period types.
c. No duplicate attendance for the same person and event.

Analytics and SQL code that can provide insights to donors

The analytics are grouped as basic, intermediate, and advanced. Each one maps to a management decision.

The SQL is clean and easy to maintain.

https://github.com/Ilyas1st/MSBA_Principles_of_BA_project

Query | Query History

```sql
-- List all partner organizations
SELECT * FROM inso.organization;

-- All events with their city and meeting type
SELECT e.event_id, e.event_date, l.city_name, mt.name AS meeting_type
FROM inso.event e
JOIN inso.location l      ON l.location_id = e.location_id
JOIN inso.meeting_type mt ON mt.meeting_type_id = e.meeting_type_id;

-- Submissions joined to their events
SELECT s.submission_id, e.event_id, e.event_date, s.is_final
FROM inso.submission s
JOIN inso.event e ON e.event_id = s.event_id;

-- Full names of participants whose organizations are LNGOs
SELECT p.full_name
FROM inso.participant p
JOIN inso.organization o   ON o.org_id = p.org_id
JOIN inso.partner_type pt  ON pt.partner_type_id = o.partner_type_id
WHERE pt.name = 'LNGO';

-- Event IDs and dates of Security Roundtable events that are bi-weekly
SELECT e.event_id, e.event_date
FROM inso.event e
JOIN inso.meeting_type mt ON mt.meeting_type_id = e.meeting_type_id
JOIN inso.period_type  pt ON pt.period_type_id  = e.period_type_id
WHERE mt.name = 'Security Roundtable'
  AND pt.name = 'bi-weekly';

-- Submission and filenames for Mogadishu Country Directors Briefing on 2025-09-10
SELECT s.submission_id, u.filename
FROM inso.event e
JOIN inso.location l ON l.location_id = e.location_id
JOIN inso.submission s ON s.event_id = e.event_id
JOIN inso.upload u     ON u.submission_id = s.submission_id
JOIN inso.meeting_type mt ON mt.meeting_type_id = e.meeting_type_id
WHERE l.city_name = 'Mogadishu'
  AND mt.name = 'Country Directors Briefing'
  AND e.event_date = DATE '2025-09-10';

-- Top Attendees
SELECT
  p.full_name,
  COUNT(*)              AS events_attended,
  MIN(e.event_date)     AS first_attended,
  MAX(e.event_date)     AS last_attended
FROM inso.attendance a
JOIN inso.participant p ON p.participant_id = a.participant_id
JOIN inso.event e       ON e.event_id = a.event_id
GROUP BY p.full_name
ORDER BY events_attended DESC, last_attended DESC, p.full_name
LIMIT 5;
```

Query | Query History

```sql
LIMIT 5;

--Security Roundtable reach by city per partner type
WITH sr AS (
  SELECT a.participant_id, e.location_id
  FROM inso.attendance a
  JOIN inso.event e      ON e.event_id = a.event_id
  JOIN inso.meeting_type mt ON mt.meeting_type_id = e.meeting_type_id
  WHERE mt.name = 'Security Roundtable'
),
mix AS (
  SELECT
    l.city_name,
    COALESCE(pt.name, 'Unknown type') AS partner_type,
    COUNT(DISTINCT p.participant_id)  AS participants,
    COUNT(DISTINCT o.org_id)          AS orgs
  FROM sr
  JOIN inso.participant p ON p.participant_id = sr.participant_id
  LEFT JOIN inso.organization o ON o.org_id = p.org_id
  LEFT JOIN inso.partner_type pt ON pt.partner_type_id = o.partner_type_id
  JOIN inso.location l ON l.location_id = sr.location_id
  GROUP BY l.city_name, pt.name
)
SELECT
  m.city_name,
  m.partner_type,
  m.orgs,
  m.participants,
  -- share of orgs within the city (kept simple; no window functions needed)
  ROUND(100.0 * m.orgs / NULLIF((
      SELECT SUM(orgs) FROM mix WHERE city_name = m.city_name
  ),0), 1) AS pct_orgs_in_city
FROM mix m
ORDER BY m.city_name, m.partner_type;

-- Distinct organization names that attended a Security Roundtable
SELECT DISTINCT o.org_name
FROM inso.attendance a
JOIN inso.event e         ON e.event_id = a.event_id
JOIN inso.meeting_type mt ON mt.meeting_type_id = e.meeting_type_id
JOIN inso.participant p   ON p.participant_id = a.participant_id
JOIN inso.organization o  ON o.org_id = p.org_id
WHERE mt.name = 'Security Roundtable';

-- Maximum and minimum attendance (only events with some attendance)
SELECT
  MAX(att_count) AS max_attendance,
  MIN(att_count) AS min_attendance
FROM (
  SELECT e.event_id, COUNT(a.participant_id) AS att_count
  FROM inso.event e
  JOIN inso.attendance a ON a.event_id = e.event_id
  GROUP BY e.event_id
) AS event_attendance;

-- Highest attendance event(s) for each meeting type
SELECT meeting_type, event_id, event_date, att_count
FROM (
  SELECT
    mt.name AS meeting_type,
    e.event_id,
    e.event_date,
    COUNT(a.participant_id) AS att_count,
    RANK() OVER (PARTITION BY mt.name ORDER BY COUNT(a.participant_id) DESC) AS rnk
  FROM inso.event e
  JOIN inso.meeting_type mt ON mt.meeting_type_id = e.meeting_type_id
  LEFT JOIN inso.attendance a ON a.event_id = e.event_id
  GROUP BY mt.name, e.event_id, e.event_date
) t
WHERE rnk = 1
ORDER BY meeting_type, event_date;
```

Decision value for Leadership

Timeliness: leadership can see which cities or types are slipping and act before the quarter ends.

Coverage: understand where to add extra sessions or outreach to specific partner types.

Compliance: measure evidence completeness and fix gaps before submission to donors.

Efficiency: reduce time on manual spreadsheet work and focus on analysis.

Risks and possible mitigations

We will protect personal data by using role-based access and limiting exports of phone numbers. To reduce data-quality errors, we will keep master lists up to date and add simple validation in the data-entry forms. We will also address capacity and training by providing a short user guide for focal persons. As the system grows, we will plan for performance by adding indexes and, later, partitions when data volume increases.

Improvements and roadmap

Near term (0–3 months):

- Add donor-specific summary views that match each template exactly.
- Create a low-code entry form so offices can enter events directly with validation.
- Add automated checks for duplicate events and missing uploads.

.