Proposal for National In-Service Training

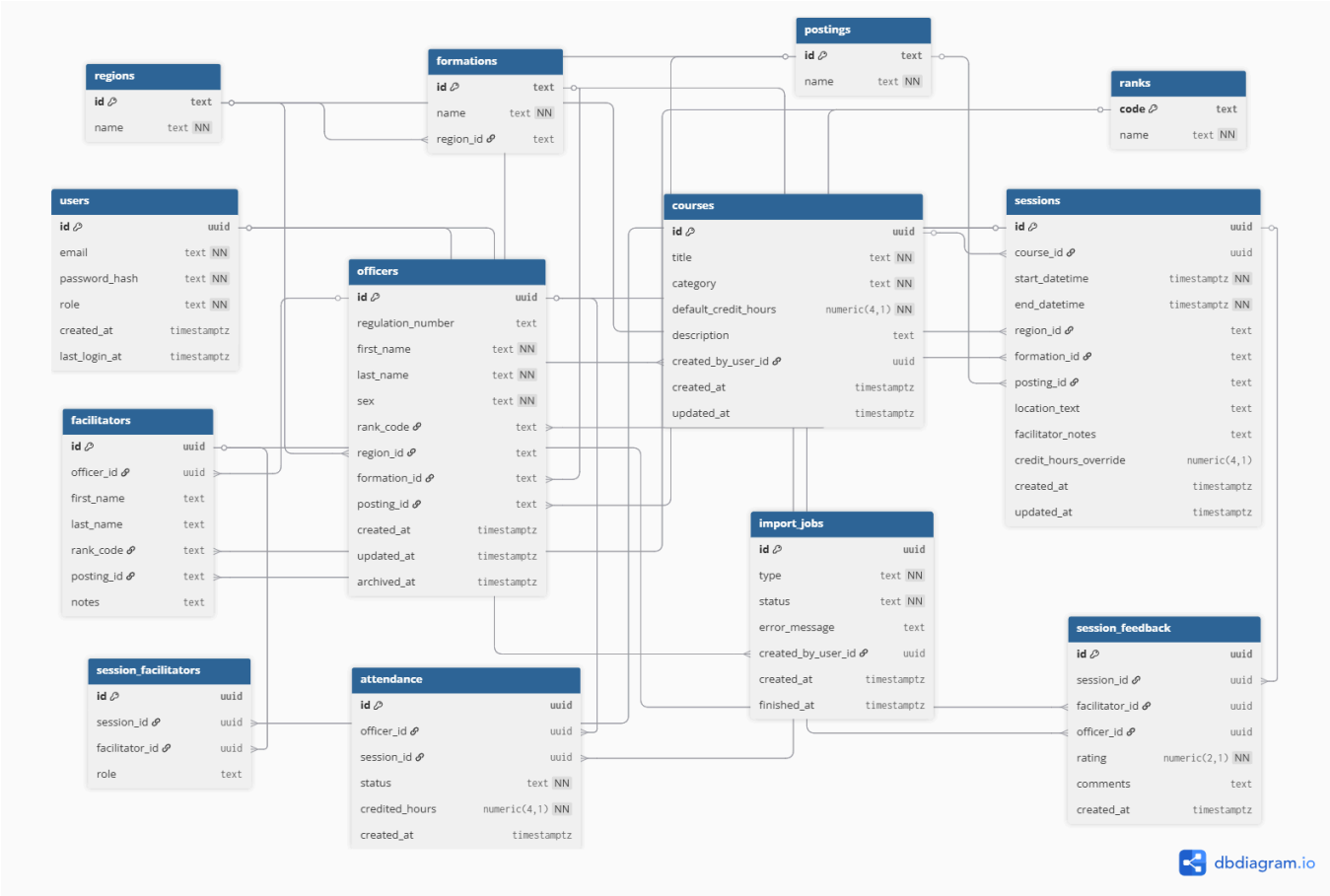
Addie Vasquez & Mickali Garbutt

2020151434 & 2020151994

CMPS4191 - Advance Web Development

20/10/2025

Entity Relationship Diagram



ERD Breakdown: National In-Service Training Database

1. Lookup Tables

Lookup tables define reference data for consistent usage across the system.

- **regions**
  - Primary key: id (e.g., "northern").
  - Attributes: name.
  - Purpose: Represents geographical regions where formations and sessions are located.
- **formations**
  - Primary key: id (e.g., "corozal").
  - Foreign key: region_id → regions.id.
  - Attributes: name.
  - Purpose: Subdivision within a region, linked back to a region.
- **postings**
  - Primary key: id (e.g., "CIB").

    - Attributes: name.
    - Purpose: Current assignment/department of an officer or facilitator.
- **ranks**
    - Primary key: code (e.g., "INSP").
    - Attributes: name.
    - Purpose: Defines police officer ranks for consistent classification.

2. Core Entities

- **users**
    - Primary key: id.
    - Attributes: email (unique), password_hash, role (enum: admin/contributor/viewer), timestamps.
    - Purpose: Represents system users who manage courses, import data, or administer the system.
- **officers**
    - Primary key: id.
    - Attributes: regulation_number (unique, optional), first_name, last_name, sex, timestamps.
    - Foreign keys:
        - rank_code → ranks.code
        - region_id → regions.id
        - formation_id → formations.id
        - posting_id → postings.id
    - Purpose: Stores personnel/officer records participating in courses.
- **courses**
    - Primary key: id.
    - Attributes: title, category (enum: mandatory/elective/instructor), default_credit_hours, description.
    - Foreign key: created_by_user_id → users.id.
    - Purpose: Defines training courses.
- **sessions**
    - Primary key: id.
    - Attributes: start_datetime, end_datetime, location_text, facilitator_notes, credit_hours_override.
    - Foreign keys:
        - course_id → courses.id
        - region_id → regions.id (nullable)
        - formation_id → formations.id
        - posting_id → postings.id
    - Purpose: Specific offerings of courses (scheduled events).
- **facilitators**
    - Primary key: id.

    - Attributes: first_name, last_name, notes.
    - Foreign keys:
        - officer_id → officers.id (nullable: facilitator may or may not be an officer).
        - rank_code → ranks.code
        - posting_id → postings.id
    - Purpose: Instructors for sessions, either officers or external personnel.
- **attendance** *(junction table)*
    - Primary key: id.
    - Attributes: status (enum: attended/absent/excused), credited_hours, timestamps.
    - Foreign keys:
        - officer_id → officers.id
        - session_id → sessions.id
    - Constraint: UNIQUE(officer_id, session_id) prevents duplicate attendance records.
    - Purpose: Tracks which officers attended which sessions and hours credited.
- **import_jobs**
    - Primary key: id.
    - Attributes: type (officers|courses|attendance), status (pending|running|failed|completed), error_message, timestamps.
    - Foreign key: created_by_user_id → users.id.
    - Purpose: Tracks batch data import operations initiated by users.

3. Junction Entities

- **session_facilitators** *(junction table)*
    - Primary key: id.
    - Foreign keys:
        - session_id → sessions.id
        - facilitator_id → facilitators.id
    - Attributes: role (e.g., lead, assistant).
    - Purpose: Allows sessions to have multiple facilitators, and facilitators to be assigned to multiple sessions.

- **session_feedback (junction table)**
    - Primary key: id.
    - Foreign keys:
        - session_id → sessions.id
        - facilitator_id → facilitators.id
        - officer_id → officers.id
    - Attributes:
        - rating (numeric, e.g., 1.0–5.0)
        - comments (optional text feedback)
        - created_at (timestamp)
    - **Purpose:** Enables officers who attend a session to provide feedback on specific facilitators. This design supports performance tracking and reporting by allowing each facilitator's average rating to be calculated over time.

4. Relationships Summary

- **Regions** have many **formations**, and may contain officers or sessions.
- **Formations** belong to one region, and can have officers or sessions.
- **Postings** categorize both officers and facilitators.
- **Ranks** apply to officers and facilitators.
- **Users** create courses and import jobs.
- **Courses** contain multiple sessions.
- Sessions occur in specific regions/formations/postings, and can have multiple facilitators (through the session_facilitators junction table).
- **Officers** attend sessions (through attendance).
- Facilitators may or may not be officers, but are linked through session_facilitators.
- **Officers** can rate facilitators for each session (through session_feedback).

5. Normalization & Constraints

- **1NF**: All attributes are atomic (no repeating groups).
- **2NF**: Composite keys avoided; all non-key attributes depend fully on the primary key.
- **3NF**: No transitive dependencies; e.g., region_id in formations prevents duplication of region data.
- **Constraints**:
    - Foreign keys enforce referential integrity (officers must belong to valid rank, formation, etc.).
    - Unique constraints prevent duplicate officer attendance or duplicate course creator emails.
    - Nullable FKs (e.g., facilitator may not have officer_id) allow flexibility.
    - Ratings use numeric validation (1–5) to ensure consistent scoring.

National In-Service Training Database: A Review of Design, Limitations, and Future Enhancements

The design of the National In-Service Training Database and API focuses on keeping things organized, secure, and easy to scale as the system grows. We used Go for its speed and concurrency features, and PostgreSQL for reliable data handling. The database is divided into lookup tables, core entities, and junction tables to avoid repetition and keep relationships clear. Lookup tables like regions, formations, and ranks store reference data that stays consistent across the system. Core entities such as officers, sessions, and courses represent the main parts of the training process, while junction tables (like attendance and session_facilitators) handle many-to-many relationships. The API supports full CRUD operations, secure login with bcrypt password hashing, email-based account activation, password reset, and JWT token authentication. Background jobs are handled using goroutines, so sending emails doesn't slow down the API. Features like rate limiting, CORS configuration, and graceful shutdown were added to improve reliability and security.

One limitation of the current system is that feedback and attendance tracking are quite simple. For example, feedback is stored mostly as numeric ratings, so it doesn't capture detailed comments or suggestions that could help improve future training sessions. There's also no built-in way to track historical changes, such as when an officer moves between postings or ranks, which could be useful for long-term analysis. Some foreign key relationships are optional, which gives flexibility but also means validation must be handled carefully to keep data clean. At the API level, we haven't yet added deeper performance tuning like extra indexes or caching for when the database gets very large.

In the future, we'd like to expand the feedback system to include comments or surveys that give more context about training quality. Adding role-based access control (RBAC) would help manage different user permissions more effectively. We also plan to include historical tracking

for officer records and possibly dashboards for visualizing attendance, performance, and training outcomes over time. Integrating with external HR or education systems could make the platform even more useful, and improving our automated testing and deployment process (e.g., through CI/CD or Docker) would make it easier to maintain long-term.