Assignment 3 – Advanced PostgreSQL

1. Views & Materialized Views
    a. Create a **VIEW** that displays:
        i. user_id
        ii. full_name (first_name + last_name)
        iii. email
        iv. role_name
    b. Create a **VIEW** that shows only **active users** created in the last 30 days.
    c. Create a **MATERIALIZED VIEW** that stores:
        i. role_name
        ii. total_users per role
    d. Write a query to **refresh** the materialized view.
2. Functions
    a. Create a **SQL function** that:
        i. Accepts a role_id as input
        ii. Returns the total number of users for that role
    b. Create a **PL/pgSQL function** that:
        i. Accepts a user_id
        ii. Returns the user's full name
    c. Create a function that:
        i. Accepts birth_date
        ii. Returns calculated age
    d. Create a function that:
        i. Returns all users created **today**
    e. Demonstrate usage of:
        i. IN parameters
        ii. OUT parameters
        iii. RETURN TABLE
3. Stored Procedures
    a. Create a **stored procedure** to:
        i. Insert a new user

    ii. Validate email uniqueness
    iii. Set created_date automatically
  b. Create a stored procedure that:
    i. Soft deletes a user (is_deleted = true)
  c. Create a stored procedure that:
    i. Updates user role
  d. Logs old role and new role into an audit table
4. Triggers
  a. Create a trigger that:
    i. Automatically updates modified_date on UPDATE of users table
  b. Create a trigger that:
    i. Prevents deletion of users
    ii. Raises an exception if DELETE is attempted
  c. Create an **AFTER INSERT** trigger to:
    i. Log user creation into an audit table
5. Cursors
  a. Write a **PL/pgSQL block** using a cursor to:
    i. Loop through all users
    ii. Print user_id and email using RAISE NOTICE
6. Jobs / Scheduling
  a. Write a job to:
    i. Refresh a materialized view every hour