Assignment 2: PostgreSQL Queries & Constraints

General Instructions

- Use PostgreSQL syntax only
- Follow snake_case naming convention
- Do not include query outputs
- Focus on correct SQL logic
- Database: homecare_db
- Schema: hc

Available Tables

- users
- roles
- service_types
- categories
- sub_categories

1. Aggregate Functions (COUNT, SUM, AVG, MIN, MAX)
    a. Count the total number of users
    b. Count the number of active users
    c. Count the number of users per role
    d. Find the minimum and maximum birth_date of users
    e. Calculate the average age of users
    f. Count total categories per service type
    g. Count total sub-categories per category
2. GROUP BY & HAVING
    a. Fetch the number of users grouped by role
    b. Fetch roles having more than 2 users
    c. Fetch service types having more than 3 categories
    d. Fetch categories having at least 2 sub-categories
    e. Fetch users grouped by birth year
    f. Fetch birth years having more than 5 users
3. Joins
    a. INNER JOIN:
        i. Fetch categories with service type names
    b. LEFT JOIN:
        i. Fetch all service types including those without categories

    c. RIGHT JOIN:
        i. Fetch all categories even if they have no sub-categories
4. EXISTS
    a. Fetch roles that have at least one user
    b. Fetch service types that have categories
    c. Fetch categories that have sub-categories
    d. Fetch users whose role exists
5. Common Table Expression (CTE)
    a. Use a CTE to count the number of categories per service type
6. Constraints
    a. UNIQUE Constraint:
        i. Ensure users.email is unique
    b. FOREIGN KEY Constraints:
        i. users → roles
        ii. categories → service_types
        iii. sub_categories → categories
    c. CHECK Constraints:
        i. Mobile number must have a valid length
        ii. birth_date must be a past date
7. Indexes
    a. Create an index on users.email
8. Date & Time Handling
    a. Fetch users created today
    b. Fetch users created in the last 30 days
    c. Extract year from users.birth_date
    d. Calculate age of each user
    e. Group users by birth year
9. Window Functions (ROW_NUMBER, RANK, LAG, LEAD)
    a. Assign row numbers to users within each role based on created_date
    b. Rank roles based on number of users
    c. Use LAG to fetch previous user creation date per role
    d. Use LEAD to fetch next user creation date per role
    e. Fetch the second oldest user per role