Laboratory Work #2: Advanced DDL Operations

Topic: Database Creation, Table Management & Data Types
Save all SQL commands in a file named `lab2_advanced_ddl.sql`

Part 1: Multiple Database Management

Task 1.1: Database Creation with Parameters

Create the following databases with specific configurations:

1. **Create database `university_main`**
   - Set owner to current user
   - Use template0 as template
   - Set encoding to UTF8
2. **Create database `university_archive`**
   - Set connection limit to 50
   - Use template0 as template
3. **Create database `university_test`**
   - Mark as template database (istemplate = true)
   - Set connection limit to 10

Task 1.2: Tablespace Operations

1. **Create tablespace `student_data`**
   - Location: '/data/students'
2. **Create tablespace `course_data`**
   - Location: '/data/courses'
   - Set owner to current user
3. **Create database `university_distributed`**
   - Use `student_data` tablespace
   - Set encoding to LATIN9

Part 2: Complex Table Creation

Task 2.1: University Management System

Using the `university_main` database, create the following tables with appropriate data types:

Table: students

- `student_id` (auto-incrementing integer, primary key)
- `first_name` (variable string, max 50 characters)
- `last_name` (variable string, max 50 characters)

- email (variable string, max 100 characters)
- phone (fixed string, exactly 15 characters)
- date_of_birth (date only)
- enrollment_date (date only)
- gpa (decimal number with 2 decimal places)
- is_active (boolean type)
- graduation_year (small integer)

Table: professors

- professor_id (auto-incrementing integer, primary key)
- first_name (variable string, max 50 characters)
- last_name (variable string, max 50 characters)
- email (variable string, max 100 characters)
- office_number (variable string, max 20 characters)
- hire_date (date only)
- salary (large decimal number with 2 decimal places)
- is_tenured (boolean type)
- years_experience (integer)

Table: courses

- course_id (auto-incrementing integer, primary key)
- course_code (fixed string, exactly 8 characters)
- course_title (variable string, max 100 characters)
- description (unlimited text)
- credits (small integer)
- max_enrollment (integer)
- course_fee (decimal with 2 decimal places)
- is_online (boolean type)
- created_at (timestamp without timezone)

Task 2.2: Time-based and Specialized Tables

Table: class_schedule

- schedule_id (auto-incrementing integer, primary key)
- course_id (integer)
- professor_id (integer)
- classroom (variable string, max 20 characters)
- class_date (date only)
- start_time (time without timezone)
- end_time (time without timezone)
- duration (interval type)

Table: student_records

- `record_id` (auto-incrementing integer, primary key)
- `student_id` (integer) - `course_id` (integer)
- `semester` (variable string, max 20 characters)
- `year` (integer)
- `grade` (fixed string, exactly 2 characters)
- `attendance_percentage` (decimal with 1 decimal place)
- `submission_timestamp` (timestamp with timezone)
- `last_updated` (timestamp with timezone)

Part 3: Advanced ALTER TABLE Operations

Task 3.1: Modifying Existing Tables

Perform the following modifications on your tables:

Modify students table:

1. Add column `middle_name` (variable string, max 30 characters)

2. Add column `student_status` (variable string, max 20 characters)

3. Change data type of phone from char(15) to varchar(20)

4. Set default value 'ACTIVE' for `student_status` column

5. Change gpa column to have default value 0.00

Modify professors table:

1. Add column `department_code` (fixed string, exactly 5 characters)

2. Add column `research_area` (unlimited text)

3. Change data type of `years_experience` to smallint

4. Set default value for `is_tenured` to false

5. Add column `last_promotion_date` (date only)

Modify courses table:

1. Add column `prerequisite_course_id` (integer)

2. Add column `difficulty_level` (smallint)

3. Change `course_code` from char(8) to varchar(10)

4. Set default value 3 for `credits` column

5. Add column `lab_required` (boolean) with default false

Task 3.2: Column Management Operations

For class_schedule table:

1. Add column `room_capacity` (integer)

2. Drop column `duration`

3. Add column `session_type` (variable string, max 15 characters)

4. Change `classroom` data type to varchar(30)

5. Add column `equipment_needed` (unlimited text)

For student_records table:

1. Add column `extra_credit_points` (decimal with 1 decimal place)

2. Change `grade` from char(2) to varchar(5)

3. Set default value 0.0 for `extra_credit_points`

4. Add column `final_exam_date` (date only)

5. Drop column `last_updated`

Part 4: Table Relationships and Management

Task 4.1: Additional Supporting Tables

Create these additional tables:

Table: departments

- `department_id` (auto-incrementing integer, primary key)
- `department_name` (variable string, max 100 characters)
- `department_code` (fixed string, exactly 5 characters)
- `building` (variable string, max 50 characters)
- `phone` (variable string, max 15 characters)
- `budget` (large decimal with 2 decimal places)
- `established_year` (integer)

Table: library_books

- book_id (auto-incrementing integer, primary key)
- isbn (fixed string, exactly 13 characters)
- title (variable string, max 200 characters)
- author (variable string, max 100 characters)
- publisher (variable string, max 100 characters)
- publication_date (date only)
- price (decimal with 2 decimal places)
- is_available (boolean)
- acquisition_timestamp (timestamp without timezone)

Table: student_book_loans

- loan_id (auto-incrementing integer, primary key)
- student_id (integer)
- book_id (integer)
- loan_date (date only)
- due_date (date only)
- return_date (date only)
- fine_amount (decimal with 2 decimal places)
- loan_status (variable string, max 20 characters)

Task 4.2: Table Modifications for Integration

1. **Add foreign key columns** (just add the columns, don't create relationships yet):
   - Add department_id to professors table (integer)
   - Add advisor_id to students table (integer)
   - Add department_id to courses table (integer)
2. **Create lookup tables**:

Table: grade_scale

- grade_id (auto-incrementing integer, primary key)
- letter_grade (fixed string, exactly 2 characters)
- min_percentage (decimal with 1 decimal place)
- max_percentage (decimal with 1 decimal place)
- gpa_points (decimal with 2 decimal places)

Table: semester_calendar

- semester_id (auto-incrementing integer, primary key)
- semester_name (variable string, max 20 characters)
- academic_year (integer)
- start_date (date only) - end_date (date only)
- registration_deadline (timestamp with timezone)

- `is_current` (boolean)

Part 5: Table Deletion and Cleanup

Task 5.1: Conditional Table Operations

Write SQL commands to safely perform these operations:

1. **Drop tables if they exist**:
   - Drop `student_book_loans` table only if it exists
   - Drop `library_books` table only if it exists
   - Drop `grade_scale` table only if it exists
2. **Recreate one of the dropped tables** with modified structure:
   - Recreate `grade_scale` table with an additional column `description` (unlimited text)
3. **Drop and recreate with CASCADE**:
   - Drop `semester_calendar` table with CASCADE option
   - Recreate it with the same structure

Task 5.2: Database Cleanup

1. **Database operations**:
   - Drop `university_test` database if it exists
   - Drop `university_distributed` database if it exists
   - Create new database `university_backup` using `university_main` as template

Deliverables

Required Files:

1. **`lab2_advanced_ddl.sql`** - All SQL commands in sequential order

Submission Format:

- Code should execute without errors when run in sequence

Grading Criteria:

- **Database Operations** (20%): Correct database and tablespace creation
- **Table Creation** (30%): Proper use of data types and table structure
- **ALTER Operations** (30%): Successful table modifications
- **Table Management** (15%): Additional tables and relationships
- **Cleanup Operations** (5%): Proper deletion and recreation

Data Types You Must Use:

From the lecture materials, ensure you use these PostgreSQL data types:

- **Integers**: smallint, integer, bigint
- **Auto-increment**: serial, smallserial
- **Floating-point**: real, double precision
- **Character**: varchar(n), char(n), text
- **Date/Time**: date, time, timestamp (with/without timezone), interval
- **Boolean**: boolean
- **Decimal**: numeric/decimal with precision