-- Task 1.1: Database Creation with Parameters

CREATE DATABASE university\_main

OWNER postgres

TEMPLATE template0

ENCODING 'UTF8';

CREATE DATABASE university\_archive

CONNECTION LIMIT 50

TEMPLATE template0;

CREATE DATABASE university\_test

IS\_TEMPLATE true

CONNECTION LIMIT 10;

-- Task 1.2: Tablespace Operations

CREATE TABLESPACE student\_data

LOCATION '/tmp/students';

CREATE TABLESPACE course\_data

OWNER postgres

LOCATION '/tmp/courses';

CREATE DATABASE university\_distributed

TABLESPACE student\_data

ENCODING 'LATIN9';

\c university\_main;

-- Task 2.1: University Management System

CREATE TABLE students (

student\_id SERIAL PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone CHAR(15),

date\_of\_birth DATE,

enrollment\_date DATE,

gpa NUMERIC(3,2),

is\_active BOOLEAN,

graduation\_year SMALLINT

);

CREATE TABLE professors (

professor\_id SERIAL PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

office\_number VARCHAR(20),

hire\_date DATE,

salary NUMERIC(12,2),

is\_tenured BOOLEAN,

years\_experience INTEGER

);

CREATE TABLE courses (

course\_id SERIAL PRIMARY KEY,

course\_code CHAR(8),

course\_title VARCHAR(100),

description TEXT,

credits SMALLINT,

max\_enrollment INTEGER,

course\_fee NUMERIC(10,2),

is\_online BOOLEAN,

created\_at TIMESTAMP WITHOUT TIME ZONE

);

-- Task 2.2: Time-based and Specialized Tables

CREATE TABLE class\_schedule (

schedule\_id SERIAL PRIMARY KEY,

course\_id INTEGER,

professor\_id INTEGER,

classroom VARCHAR(20),

class\_date DATE,

start\_time TIME WITHOUT TIME ZONE,

end\_time TIME WITHOUT TIME ZONE,

duration INTERVAL

);

CREATE TABLE student\_records (

record\_id SERIAL PRIMARY KEY,

student\_id INTEGER,

course\_id INTEGER,

semester VARCHAR(20),

year INTEGER,

grade CHAR(2),

attendance\_percentage NUMERIC(4,1),

submission\_timestamp TIMESTAMP WITH TIME ZONE,

last\_updated TIMESTAMP WITH TIME ZONE

);

-- Task 3.1: Modifying Existing Tables

-- Students

ALTER TABLE students ADD middle\_name VARCHAR(30);

ALTER TABLE students ADD student\_status VARCHAR(20) DEFAULT 'ACTIVE';

ALTER TABLE students ALTER COLUMN phone TYPE VARCHAR(20);

ALTER TABLE students ALTER COLUMN gpa SET DEFAULT 0.00;

-- Professors

ALTER TABLE professors ADD department\_code CHAR(5);

ALTER TABLE professors ADD research\_area TEXT;

ALTER TABLE professors ALTER COLUMN years\_experience TYPE SMALLINT;

ALTER TABLE professors ALTER COLUMN is\_tenured SET DEFAULT false;

ALTER TABLE professors ADD last\_promotion\_date DATE;

-- Courses

ALTER TABLE courses ADD prerequisite\_course\_id INTEGER;

ALTER TABLE courses ADD difficulty\_level SMALLINT;

ALTER TABLE courses ALTER COLUMN course\_code TYPE VARCHAR(10);

ALTER TABLE courses ALTER COLUMN credits SET DEFAULT 3;

ALTER TABLE courses ADD lab\_required BOOLEAN DEFAULT false;

-- Task 3.2: Column Management Operations

-- Class Schedule

ALTER TABLE class\_schedule ADD room\_capacity INTEGER;

ALTER TABLE class\_schedule DROP COLUMN duration;

ALTER TABLE class\_schedule ADD session\_type VARCHAR(15);

ALTER TABLE class\_schedule ALTER COLUMN classroom TYPE VARCHAR(30);

ALTER TABLE class\_schedule ADD equipment\_needed TEXT;

-- Student Records

ALTER TABLE student\_records ADD extra\_credit\_points NUMERIC(4,1) DEFAULT 0.0;

ALTER TABLE student\_records ALTER COLUMN grade TYPE VARCHAR(5);

ALTER TABLE student\_records ADD final\_exam\_date DATE;

ALTER TABLE student\_records DROP COLUMN last\_updated;

-- Task 4.1: Additional Supporting Tables

CREATE TABLE departments (

department\_id SERIAL PRIMARY KEY,

department\_name VARCHAR(100),

department\_code CHAR(5),

building VARCHAR(50),

phone VARCHAR(15),

budget NUMERIC(15,2),

established\_year INTEGER

);

CREATE TABLE library\_books (

book\_id SERIAL PRIMARY KEY,

isbn CHAR(13),

title VARCHAR(200),

author VARCHAR(100),

publisher VARCHAR(100),

publication\_date DATE,

price NUMERIC(10,2),

is\_available BOOLEAN,

acquisition\_timestamp TIMESTAMP WITHOUT TIME ZONE

);

CREATE TABLE student\_book\_loans (

loan\_id SERIAL PRIMARY KEY,

student\_id INTEGER,

book\_id INTEGER,

loan\_date DATE,

due\_date DATE,

return\_date DATE,

fine\_amount NUMERIC(8,2),

loan\_status VARCHAR(20)

);

-- Task 4.2: Integration Columns

ALTER TABLE professors ADD department\_id INTEGER;

ALTER TABLE students ADD advisor\_id INTEGER;

ALTER TABLE courses ADD department\_id INTEGER;

-- Lookup Tables

CREATE TABLE grade\_scale (

grade\_id SERIAL PRIMARY KEY,

letter\_grade CHAR(2),

min\_percentage NUMERIC(4,1),

max\_percentage NUMERIC(4,1),

gpa\_points NUMERIC(3,2)

);

CREATE TABLE semester\_calendar (

semester\_id SERIAL PRIMARY KEY,

semester\_name VARCHAR(20),

academic\_year INTEGER,

start\_date DATE,

end\_date DATE,

registration\_deadline TIMESTAMP WITH TIME ZONE,

is\_current BOOLEAN

);

-- Task 5.1: Conditional Table Operations

DROP TABLE IF EXISTS student\_book\_loans;

DROP TABLE IF EXISTS library\_books;

DROP TABLE IF EXISTS grade\_scale;

-- Recreate grade\_scale with description

CREATE TABLE grade\_scale (

grade\_id SERIAL PRIMARY KEY,

letter\_grade CHAR(2),

min\_percentage NUMERIC(4,1),

max\_percentage NUMERIC(4,1),

gpa\_points NUMERIC(3,2),

description TEXT

);

-- Drop and recreate semester\_calendar with CASCADE

DROP TABLE IF EXISTS semester\_calendar CASCADE;

CREATE TABLE semester\_calendar (

semester\_id SERIAL PRIMARY KEY,

semester\_name VARCHAR(20),

academic\_year INTEGER,

start\_date DATE,

end\_date DATE,

registration\_deadline TIMESTAMP WITH TIME ZONE,

is\_current BOOLEAN

);

-- Task 5.2: Database Cleanup

DROP DATABASE IF EXISTS university\_test;

DROP DATABASE IF EXISTS university\_distributed;

CREATE DATABASE university\_backup

TEMPLATE university\_main;