1.1

CREATE DATABASE university\_main

OWNER = postgres

TEMPLATE = template0

ENCODING = 'UTF8';

CREATE DATABASE university\_archive

TEMPLATE = template0

CONNECTION LIMIT = 50;

CREATE DATABASE university\_test

CONNECTION LIMIT = 10

IS\_TEMPLATE = true;

1.2

CREATE TABLESPACE student\_data

LOCATION 'C:\Users\Asus\Downloads\data\students';

CREATE TABLESPACE course\_data

OWNER CURRENT\_USER

LOCATION '/data/courses';

CREATE DATABASE university\_distributed

WITH OWNER = CURRENT\_USER

TABLESPACE = student\_data

ENCODING = 'LATIN9';

2.1

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 CHAR(20),

hire\_date DATE,

salary NUMERIC(12,2),

is\_tenured BOOLEAN,

years\_experience INT

);

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)

);

2.2

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

);

3.1

ALTER TABLE students

ADD COLUMN middle\_name VARCHAR(30);

ALTER TABLE students

ADD COLUMN 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;

2. ALTER TABLE professors

ADD COLUMN department\_code CHAR(5);

ALTER TABLE professors

ADD COLUMN 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 COLUMN last\_promotion\_date DATE;

3. ALTER TABLE courses

ADD COLUMN prerequisite\_course\_id INTEGER;

ALTER TABLE courses

ADD COLUMN 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 COLUMN lab\_required BOOLEAN DEFAULT false;

3.2

ALTER TABLE class\_schedule

ADD COLUMN room\_capacity INTEGER;

ALTER TABLE class\_schedule

DROP COLUMN duration;

ALTER TABLE class\_schedule

ADD COLUMN session\_type VARCHAR(15);

ALTER TABLE class\_schedule

ALTER COLUMN classroom TYPE VARCHAR(30);

ALTER TABLE class\_schedule

ADD COLUMN equipment\_needed TEXT;

ALTER TABLE student\_records

ADD COLUMN extra\_credit\_points DECIMAL(3,1) DEFAULT 0.0;

ALTER TABLE student\_records

ALTER COLUMN grade TYPE VARCHAR(5);

ALTER TABLE student\_records

ADD COLUMN final\_exam\_date DATE;

4.1

CREATE TABLE departments (

department\_id SERIAL PRIMARY KEY,

department\_name VARCHAR(100),

department\_code CHAR(5),

building VARCHAR(50),

phone VARCHAR(15),

budget DECIMAL(15,2),

established\_year INT

);

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 DECIMAL(10,2),

is\_available BOOLEAN,

acquisition\_timestamp TIMESTAMP

);

CREATE TABLE student\_book\_loans (

loan\_id SERIAL PRIMARY KEY,

student\_id INT,

book\_id INT,

loan\_date DATE,

due\_date DATE,

return\_date DATE,

fine\_amount DECIMAL(10,2),

loan\_status VARCHAR(20)

);

4.2

ALTER TABLE professors

ADD COLUMN department\_id INT;

ALTER TABLE students

ADD COLUMN advisor\_id INT;

ALTER TABLE courses

ADD COLUMN department\_id INT;

CREATE TABLE grade\_scale (

grade\_id SERIAL PRIMARY KEY,

letter\_grade CHAR(2),

min\_percentage DECIMAL(4,1),

max\_percentage DECIMAL(4,1),

gpa\_points DECIMAL(3,2)

);

CREATE TABLE semester\_calendar (

semester\_id SERIAL PRIMARY KEY,

semester\_name VARCHAR(20),

academic\_year INT,

start\_date DATE,

end\_date DATE,

registration\_deadline TIMESTAMPTZ,

is\_current BOOLEAN

);

5.1

DROP TABLE IF EXISTS student\_book\_loans;

DROP TABLE IF EXISTS library\_books;

DROP TABLE IF EXISTS grade\_scale;

CREATE TABLE grade\_scale (

grade\_id SERIAL PRIMARY KEY,

letter\_grade CHAR(2),

min\_percentage DECIMAL(4,1),

max\_percentage DECIMAL(4,1),

gpa\_points DECIMAL(3,2),

description TEXT

);

DROP TABLE IF EXISTS semester\_calendar CASCADE;

CREATE TABLE semester\_calendar (

semester\_id SERIAL PRIMARY KEY,

semester\_name VARCHAR(20),

academic\_year INT,

start\_date DATE,

end\_date DATE,

registration\_deadline TIMESTAMPTZ,

is\_current BOOLEAN

);

5.2

DROP DATABASE IF EXISTS university\_test;

DROP DATABASE IF EXISTS university\_distributed;

CREATE DATABASE university\_backup

WITH TEMPLATE = university\_main;