Task 1.1

CREATE DATABASE university\_main

WITH OWNER = postgres

TEMPLATE = template0

ENCODING = 'UTF8';

CREATE DATABASE university\_archive

WITH OWNER = postgres

TEMPLATE = template0

CONNECTION LIMIT = 50;

CREATE DATABASE university\_test

WITH OWNER = postgres

TEMPLATE = template0

CONNECTION LIMIT = 10;

CREATE TABLESPACE student\_data

LOCATION '/Users/arsenijkansin/data/students';

1.2

CREATE TABLESPACE course\_data

OWNER arsenijkansin

LOCATION '/Users/arsenijkansin/data/courses';

CREATE TABLESPACE student\_data

LOCATION '/Users/arsenijkansin/data/students';

CREATE DATABASE university\_distributed\_latin9

WITH OWNER arsenijkansin

TABLESPACE student\_data

ENCODING = 'LATIN9'

LC\_COLLATE = 'C'

LC\_CTYPE = 'C'

TEMPLATE template0;

Task 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 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 INT

);

CREATE TABLE courses(

course\_id SERIAL PRIMARY KEY ,

course\_code CHAR(8),

course\_title VARCHAR(100),

description TEXT,

credits SMALLINT,

max\_enrollment INT,

course\_fee NUMERIC(10,2),

is\_online BOOLEAN,

duration INTERVAL

);

Task 2.2

CREATE TABLE class\_schedule(

schedule\_id SERIAL PRIMARY KEY ,

course\_id INT,

professor\_id INT,

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 INT,

course\_id INT,

semester VARCHAR(20),

year INT,

grade CHAR(2),

attendance\_percentage NUMERIC(4,1),

submission\_timestamp TIMESTAMP WITH TIME ZONE,

last\_updated TIMESTAMP WITH TIME ZONE

);

3.1

ALTER TABLE students

ADD COLUMN middle\_name VARCHAR(30),

ADD COLUMN student\_status VARCHAR(20);

ALTER TABLE students

ALTER COLUMN phone TYPE VARCHAR(20);

ALTER TABLE students

ALTER COLUMN student\_status SET DEFAULT 'ACTIVE';

ALTER TABLE students

ALTER COLUMN gpa SET DEFAULT 0.00;

ALTER TABLE professors

ADD COLUMN department\_code CHAR(5),

ADD COLUMN student\_status 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;

ALTER TABLE courses

ADD COLUMN prerequisite\_course\_id INT,

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 INT,

DROP COLUMN duration,

ADD COLUMN session\_type VARCHAR(15),

ALTER COLUMN classroom TYPE VARCHAR(30),

ADD COLUMN equipment\_needed TEXT;

ALTER TABLE student\_records

ADD COLUMN extra\_credit\_points NUMERIC(4,1),

ALTER COLUMN grade TYPE VARCHAR(5),

ALTER COLUMN extra\_credit\_points SET DEFAULT 0.0,

ADD COLUMN final\_exam\_date DATE,

DROP COLUMN last\_updated;

4.1

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) NOT NULL,

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) NOT NULL,

academic\_year INT NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

registration\_deadline TIMESTAMPTZ NOT NULL,

is\_current BOOLEAN DEFAULT FALSE

);

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 TIMESTAMP WITH TIME ZONE,

is\_current BOOLEAN

);

UPDATE pg\_database

SET datistemplate = false

WHERE datname = 'university\_test';

UPDATE pg\_database

SET datistemplate = false

WHERE datname = 'university\_distributed';

DROP DATABASE IF EXISTS university\_distributed;

CREATE DATABASE university\_backup

WITH TEMPLATE = university\_main

OWNER postgres;

CREATE DATABASE university\_backup

WITH TEMPLATE = university\_main

OWNER postgres;