Part 1

Task 1.1

```
1.  CREATE DATABASE university_main
            WITH
            OWNER       =           postgres
            TEMPLATE  =           template0
            ENCODING =            'UTF8';
2.  CREATE DATABASE university_archive
            WITH
            CONNECTION LIMIT =        50
            TEMPLATE                 =        template0;
3.  CREATE DATABASE university_test
            WITH
            IS_TEMPLATE            =        true
            CONNECTION LIMIT =        10;
```

Task 1.2

```
1.  CREATE TABLESPACE student_data
            LOCATION                'C:/data/students';
2.  CREATE TABLESPACE course_data
            OWNER                   ‘postgres’
            LOCATION                ‘C:/data/courses’;
3.  CREATE DATABASE university_distributed
            WITH
            TABLESPACE           student_data
            ENCODING =            'LATIN9';
```

Part 2

Task 2.1

```
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
    );
2.  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(10, 2),
            is_tenured            BOOLEAN,
            years_experience      INT
    );
3.  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,
            created_at            TIMESTAMP WITHOUT TIME ZONE
    );
```

Task 2.2

1.
   ```
   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 GENERATED ALWAYS AS (end_time - start_time) STORED
   );
   ```

2.
   ```
   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
   );
   ```

Part 3

Task 3.1

```
1.  ALTER TABLE students ADD        middle_name             VARCHAR(30);
    ALTER TABLE students ADD        student_status          VARCHAR(20);
    ALTER TABLE students ALTER      phone                   TYPE VARCHAR(20);
    ALTER TABLE students ALTER      student_status          SET DEFAULT 'ACTIVE';
    ALTER TABLE students ALTER      gpa                     SET DEFAULT 0.00;

2.  ALTER TABLE professors ADD      department_code         CHAR(5);
    ALTER TABLE professors ADD      research_area           TEXT;
    ALTER TABLE professors ALTER    years_experience        TYPE SMALLINT;
    ALTER TABLE professors ALTER    is_tenured              SET DEFAULT false;
    ALTER TABLE professors ADD      last_promotion_date     DATE;

3.  ALTER TABLE courses ADD         prerequisite_course_id  INT;
    ALTER TABLE courses ADD         difficulty_level        SMALLINT;
    ALTER TABLE courses ALTER       course_code             TYPE VARCHAR(10);
    ALTER TABLE courses ALTER       credits                 SET DEFAULT 3;
    ALTER TABLE courses ADD         lab_required            BOOLEAN DEFAULT false;
```

Task 3.2

```
1.  ALTER TABLE class_schedule ADD        room_capacity       INT;
    ALTER TABLE class_schedule DROP       duration;
    ALTER TABLE class_schedule ADD        session_type        VARCHAR(15);
    ALTER TABLE class_schedule ALTER      classroom           TYPE VARCHAR(30);
    ALTER TABLE class_schedule ADD        equipment_needed    TEXT;

2.  ALTER TABLE student_records ADD       extra_credit_points NUMERIC(5, 1);
    ALTER TABLE student_records ALTER     grade               TYPE VARCHAR(5);
    ALTER TABLE student_records ALTER     extra_credit_points SET DEFAULT 0.0;
    ALTER TABLE student_records ADD       final_exam_date     DATE;
    ALTER TABLE student_records DROP      last_updated;
```

Part 4

Task 4.1

1.
```
CREATE TABLE departments (
        department_id               SERIAL PRIMARY KEY,
        department_name             VARCHAR(100),
        department_code             CHAR(5),
        building                    VARCHAR(50),
        phone                       VARCHAR(15),
        budget                      NUMERIC(34, 2),
        established_year            INT
);
```

2.
```
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(32, 2),
        is_available                BOOLEAN,
        acquisition_timestamp       TIMESTAMP WITHOUT TIME ZONE
);
```

3.
```
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                 NUMERIC(32, 2),
        loan_status                 VARCHAR(20)
);
```

Task 4.2

1.
```
ALTER TABLE professors ADD department_id      INT;
ALTER TABLE students   ADD advisor_id         INT;
ALTER TABLE courses    ADD department_id      INT;
```

2.
```
CREATE TABLE grade_scale (
        grade_id             SERIAL PRIMARY KEY,
        letter_grade         CHAR(2),
        min_percentage       NUMERIC(32, 1),
        max_percentage       NUMERIC(32, 1),
        gpa_points           NUMERIC(32, 2)
);
```

3.
```
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
);
```

Part 5

Task 5.1

1.
   ```
   DROP TABLE IF EXISTS student_book_loans;
   DROP TABLE IF EXISTS library_books;
   DROP TABLE IF EXISTS grade_scale;
   ```

2.
   ```
   CREATE TABLE grade_scale (
           grade_id            SERIAL PRIMARY KEY,
           letter_grade        CHAR(2),
           min_percentage      NUMERIC(32, 1),
           max_percentage      NUMERIC(32, 1),
           gpa_points          NUMERIC(32, 2),
           description         TEXT
   );
   ```

3.
   ```
   DROP TABLE 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
   );
   ```

Task 5.2

1.
   ```
   DROP DATABASE IF EXISTS university_test;
   ```

2.
   ```
   DROP DATABASE IF EXISTS university_distributed;
   ```

3.
   ```
   CREATE DATABASE university_backup
           WITH
           TEMPLATE = university_main;
   ```