UNIVERSITY DATABASE DESIGN

Malulului Geoffrey

Let's connect:

Website

Email: malului227@gmail.com

Database Design Documentation

Introduction

This document presents the database design for a comprehensive **University Management System**. The system is intended to support the efficient handling of key academic and administrative operations within an educational institution. Its main objective is to manage data related to students, academic courses, departments, schools, lecturers, exams, and financial transactions.

The database schema is structured to ensure data integrity, scalability, and consistency across multiple functional areas. It leverages a combination of normalized tables, enumerated types, and well-defined relationships to facilitate efficient data storage and retrieval.

NOTE: These design can be adjusted and added more functionality to meet desired database objectives.

Enums

The following enumerations **examples** are used to standardize and restrict the values allowed in specific table fields:

- CoursCode: BSCS, BSIT, BTIT

- CourseLevel: CERTIFICATE, DIPLOMA, DEGREE, MASTERS, PHD

- DepartmentCode: DCST, DOE

- SchoolCode: ICI, SOE, SOB

- UnitCode: CCS4401, BSA2203

- Session: 1.1, 1.2, 1.3, 2.1

- RegistrationStatus: UNREGISTERED, REGISTERED, DEFFERRED

- AdminRole: VC, DVC, REGISTRAR, FINANCE, SUPERADMIN

- UnitGrade: A, B, C, D, E

Tables

The database consists of the following primary tables:

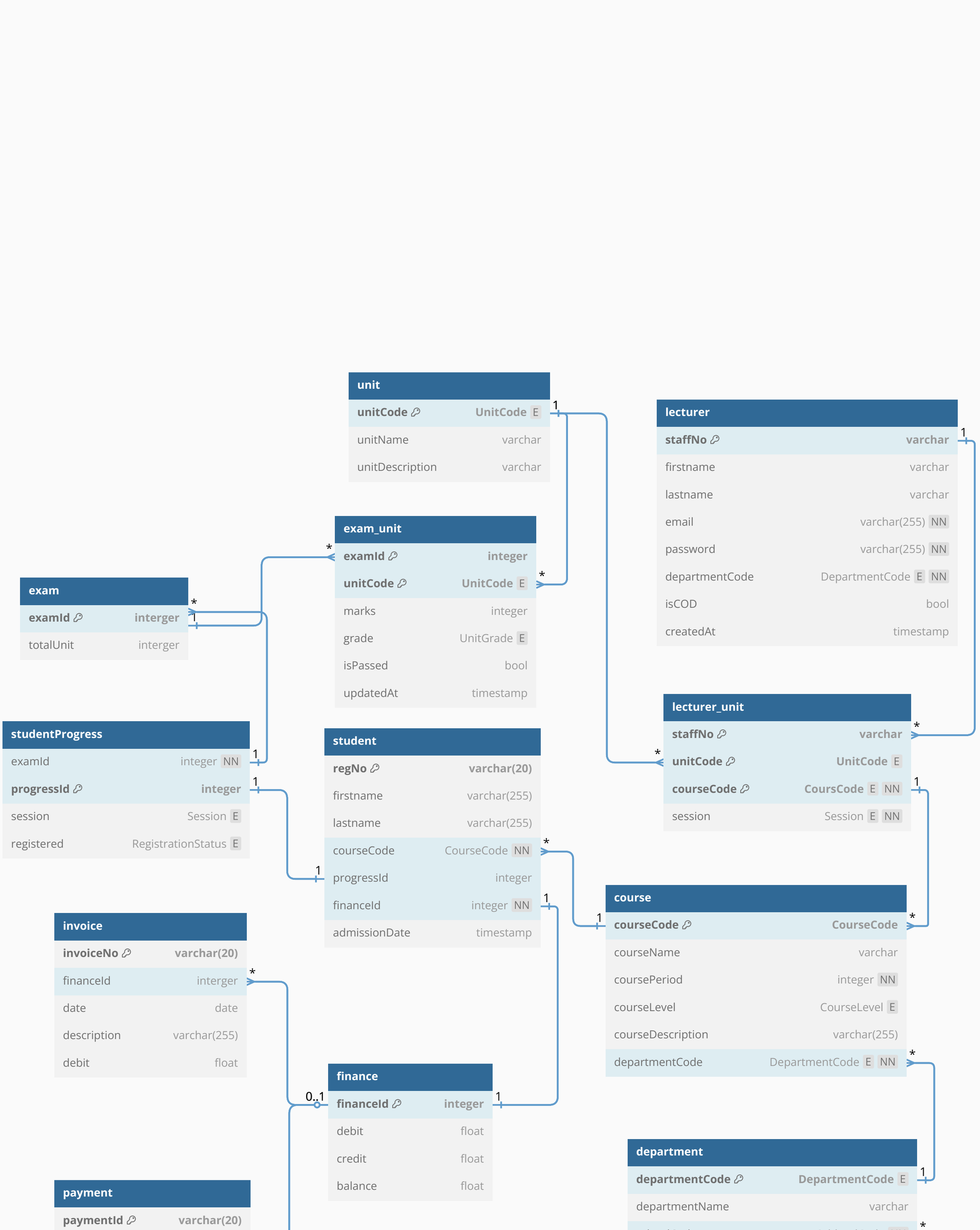
1. **student**: Stores student personal and academic details.
2. **course**: Contains course information and associated department.
3. **department**: Represents academic departments and associated schools.
4. **school**: Holds information about various schools within the institution.
5. **unit**: Academic units taught within courses.
6. **lecturer**: Records lecturer personal and departmental information.
7. **lecturer**_unit: Mapping between lecturers, the units they teach, and course sessions.
8. **finance**: Financial records associated with students.
9. **invoice**: Invoices related to a student's financial account.
10. **payment**: Payments made by students.
11. **studentProgress**: Tracks student's academic progress and exam registration.
12. **exam**: Exam records including number of units taken.
13. **exam_unit**: Performance of a student in a specific unit.
14. **Admin**: Administrative users and their roles within the system.

Relationships

Relationships defined among the tables:

- **Student ↔ Finance**: One-to-one link via `financeId`
- **Student ↔ StudentProgress**: One-to-one link via `progressId`
- **Student → Course**: Many students belong to one course (`courseCode`)
- **Course → Department**: Each course is managed by one department (`departmentCode`)
- **Department → School**: Each department belongs to one school (`schoolCode`)
- **LecturerUnit → Lecturer**: Each unit assignment references a lecturer (`staffNo`)
- **LecturerUnit → Unit**: Each assignment links to one unit (`unitCode`)
- **LecturerUnit → Course**: Unit assignments are course-specific (`courseCode`)
- **ExamUnit → Exam**: Unit exam results are tied to a specific exam (`examId`)

- **StudentProgress → Exam**: Each progress record is linked to one exam (`examId`)
- **ExamUnit → Unit**: Exam results reference the academic unit (`unitCode`)
- **Invoice → Finance**: Invoices are tied to a student's finance record (`financeId`)
- **Payment → Finance**: Payments are recorded against a finance record (`financeId`)



```
Enum CoursCode {
    BSCS
    BSIT
    BTIT
}


Enum CourseLevel {
    CERTIFICATE
    DIPLOMA
    DEGREE
    MASTERS
    PHD
}


Enum DepartmentCode {
    DCST
    DOE
}


Enum SchoolCode {
    ICI
    SOE
    SOB
}


Enum UnitCode {
    CCS4401
    BSA2203
}


Enum Session {
    1.1
    1.2
    1.3
    2.1
}


Enum RegistrationStatus {
    UNREGISTERED
    REGISTERED
    DEFFERRED
}


Enum AdminRole {
    VC
    DVC
    REGISTRAR
    FINANCE
    SUPERADMIN
}


Enum UnitGrade {
    A
    B
    C
    D
    E
}
```