

**University Management System**

**This document is submitted for Database Systems project, by:**

**Ebrahim Ramadan – 320220029 (Team Leader)**

**Momen Hesham – 320220060**

**Ahmed Hamed– 320230024**

**Renada Tamer – 320230143**

**Youssef Kamal - 320220025**

**Under supervision of /**

**Business and System Requirements Specification**

**1. Key Stakeholders**

**1.1 Students Requirements:**

**- Personal profile management (NID, Name, Email, Address)**

**- Academic record access and management**

**- Course registration and tracking**

**- Access to transcripts and GPA information**

**- Library loan management**

**- Research assistant role management**

**- Attendance tracking**

**- Academic advisory services**

**- Club membership management**

**1.2 Academic Staff**

**Professors Requirements:**

**- Profile management (Name, NationalID, Email)**

**- Department affiliation management**

**- Course teaching assignments**

**- Research project supervision**

**- Faculty relationship management**

**- Professional ID management**

**Instructors Requirements:**

**- Profile management (NationalID, Email)**

**- Course teaching assignments**

**- Department affiliation**

**- Office hour management**

**1.3 Administrative Entities**

**Department Management Requirements:**

**- Department information management**

**- Staff assignment management**

**- Building and room allocation**

**- Course offering management**

**- Faculty relationship management**

**Faculty Administration Requirements:**

**- Faculty profile management**

**- Department oversight**

**- Research project management**

**- Staff allocation**

**- Resource management**

**1.4 Facility Management**

**Building Management Requirements:**

**- Building information management**

**- Room allocation and tracking**

**- Capacity management**

**- Equipment tracking**

**2. System Requirements**

**2.1 Academic Management**

**Course Management Requirements:**

**- Course creation and updates**

**- Prerequisites management**

**- Grade tracking**

**- Course Code management**

**- Credit hour tracking**

**- Course material management**

**Examination System Requirements:**

**- Exam scheduling**

**- Score recording**

**- Weight management**

**- Exam type classification**

**- Duration tracking**

**Transcript Management Requirements:**

**- GPA calculation**

**- Course history tracking**

**- Academic performance recording**

**- Semester-wise grade management**

**2.2 Administrative Functions**

**Attendance System Requirements:**

**- Student attendance tracking**

**- Attendance date recording**

**- Attendance status management**

**Library System Requirements:**

**- Loan management**

**- Due date tracking**

**- Book return status**

**- Fine management**

**Club Management Requirements:**

**- Club membership tracking**

**- Student participation records**

**- Club activity management**

**2.3 Research Management**

**Research Projects Requirements:**

**- Project tracking**

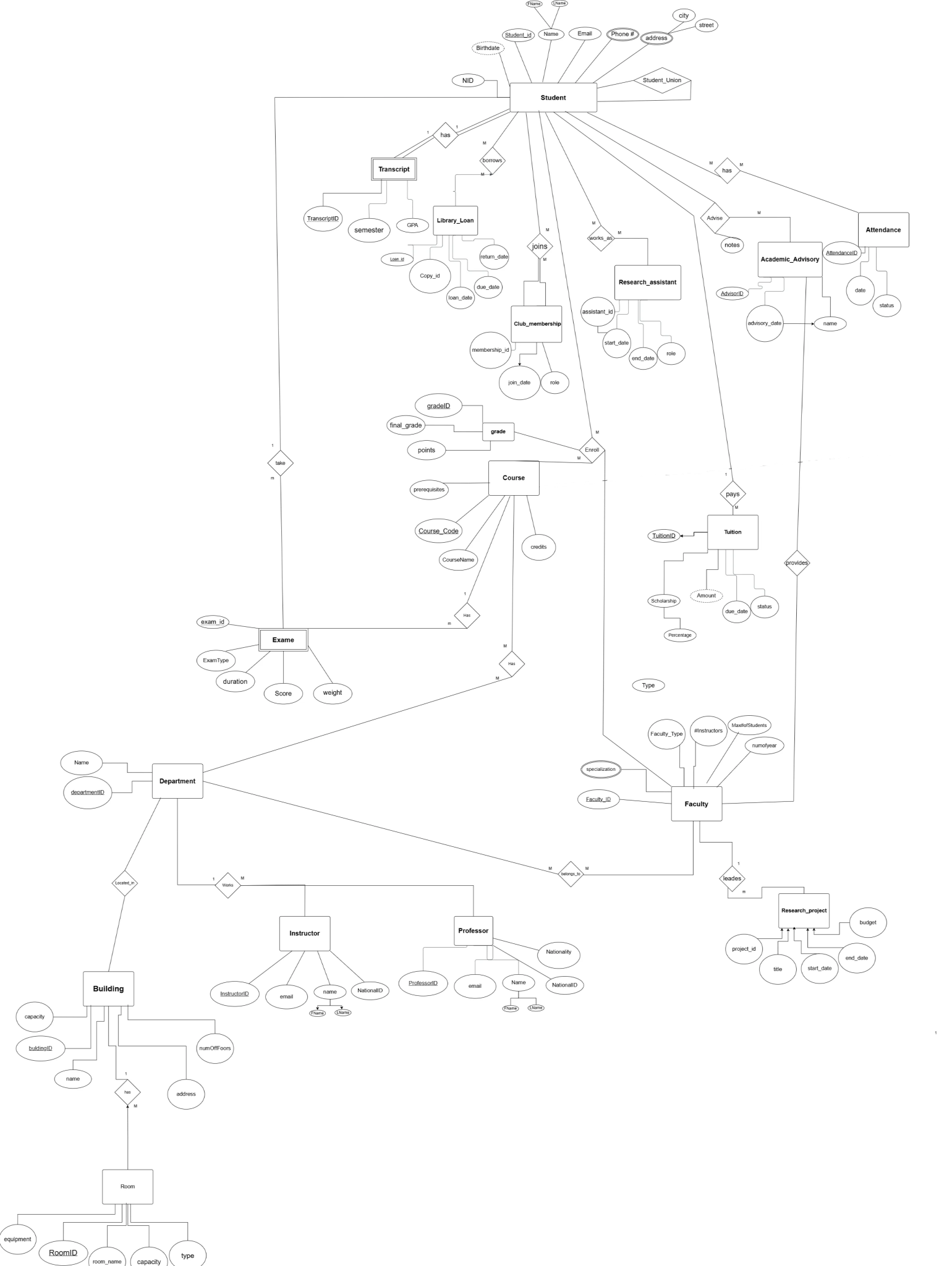
**- Student assistant assignment**

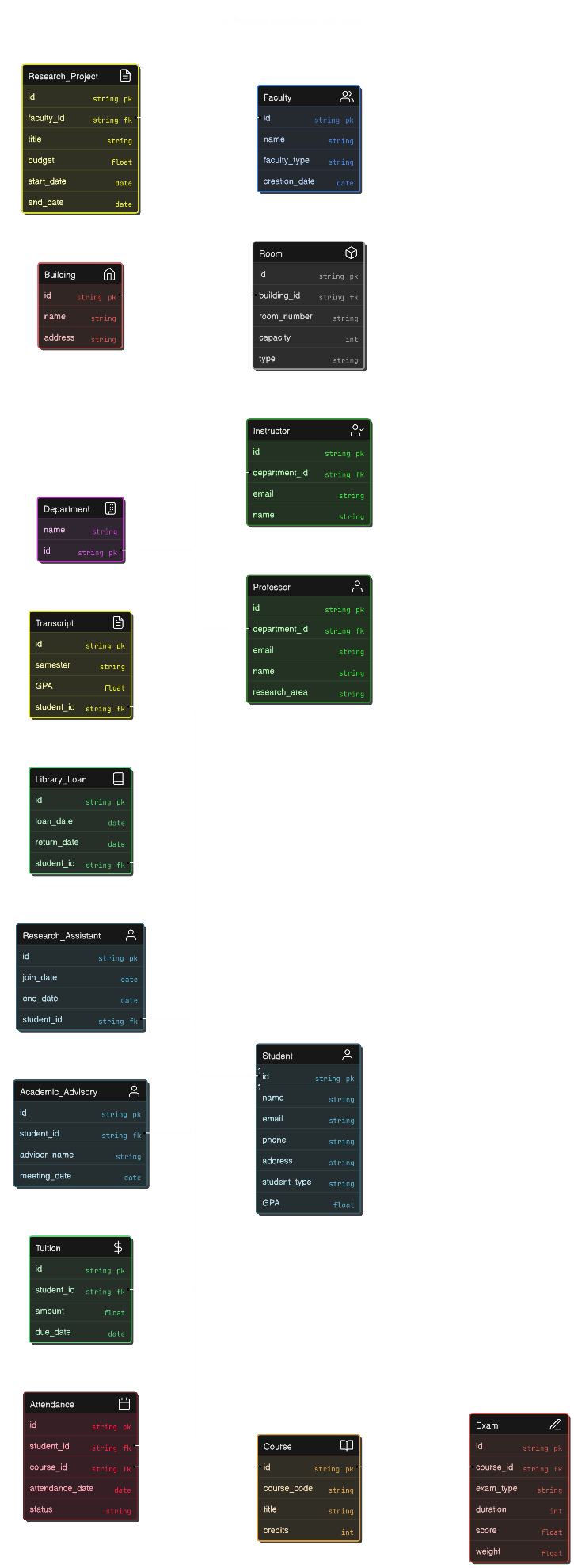
**- Faculty supervision**

**- Project status management**

**- Timeline management**

ERD

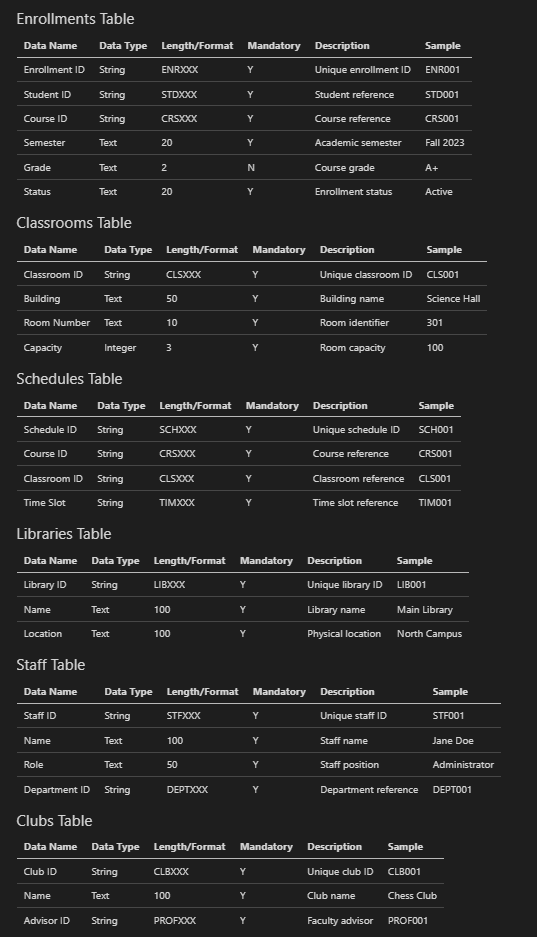




Schema

Data Dictionary







DQL / RA

SQL queries and Relational Algebra

A screenshot of a computer

Description automatically generated1- Advisor Information Retrieval

ADVISOR\_INFO ← π[academic\_advisor\_id, name, student\_id, email](academic\_advisors)

A screenshot of a computer

Description automatically generated2-Student Information Retrieval

STUDENT\_INFO ← π[student\_id, enrollment\_year, name, major, age, gpa, email, faculty\_id](students)

A screenshot of a computer

Description automatically generated3-Students Information by Specific Faculty (Faculty ID 2)

STUDENT\_FACULTY ← (students ⋈students.faculty\_id=faculty.faculty\_id faculty)

RESULT\_3 ← π[student\_id, enrollment\_year, name, major, age, gpa, email, faculty.name](σ[faculty\_id=2](STUDENT\_FACULTY))

4-Professors Working in Specific Faculty (Faculty ID 1)A screenshot of a computer program

Description automatically generated

FACULTY\_PROFESSORS ← (professor ⋈professor.professor\_id=professor\_faculty.professor\_id professor\_faculty)

RESULT\_4 ← π[professor\_id, name, email, nationalID](σ[faculty\_id=1](FACULTY\_PROFESSORS))

5-Single Student Record Lookup (Student ID 10)A screenshot of a computer

Description automatically generated

SINGLE\_STUDENT ← π[student\_id, enrollment\_year, name, major, age, gpa, email](σ[student\_id=10](students))

6-Faculty Name Lookup for Specific Professor (Professor ID 5)A screenshot of a computer

Description automatically generated

PROF\_FACULTIES ← (professor\_faculty ⋈professor\_faculty.faculty\_id=faculty.faculty\_id faculty)

RESULT\_6 ← π[faculty.name](σ[professor\_id=5](PROF\_FACULTIES))

7-Professor Teaching Load Analysis (Faculty Count per Professor)A screenshot of a computer

Description automatically generated

PROF\_COUNT ← G[professor\_id]COUNT(faculty\_id)->faculty\_count(σ[COUNT(faculty\_id)>0](professor\_faculty))

8-Unassigned Professor-Faculty CombinationsA screenshot of a computer

Description automatically generated

ALL\_COMBINATIONS ← (professor ⋈p.professor\_id=pf.professor\_id AND f.faculty\_id=pf.faculty\_id faculty)

RESULT\_8 ← π[professor.name, faculty.name](σ[pf.professor\_id IS NULL](ALL\_COMBINATIONS))

9-Complete Professor-Faculty Relationship OverviewA screenshot of a computer

Description automatically generated

PROF\_FAC\_FULL ← ((professor ⋈professor.professor\_id=professor\_faculty.professor\_id professor\_faculty)⋈professor\_faculty.faculty\_id=faculty.faculty\_id faculty)

RESULT\_9 ← π[professor.name, faculty.name](PROF\_FAC\_FULL)

10-Advisor-Student Assignment ListA computer screen shot of a program

Description automatically generated

ADVISOR\_STUDENTS ← (academic\_advisors ⋈aa.academic\_advisor\_id=students.academic\_advisor\_id students)

RESULT\_10 ← π[aa.academic\_advisor\_id, aa.name->advisor\_name, s.student\_id, s.name->student\_name](ADVISOR\_STUDENTS)

11-Advisors Without Assigned StudentsA screenshot of a computer

Description automatically generated

ADVISORS\_NO\_STUDENTS ← (academic\_advisors ⋈aa.academic\_advisor\_id=students.academic\_advisor\_id students)

RESULT\_11 ← π[aa.academic\_advisor\_id, aa.name->advisor\_name](σ[s.student\_id IS NULL](ADVISORS\_NO\_STUDENTS))

12-Student Load per Advisor (Count)A screenshot of a computer program

Description automatically generated

ADVISOR\_COUNT ← (academic\_advisors ⋈aa.academic\_advisor\_id=students.academic\_advisor\_id students)

RESULT\_12 ← G[aa.academic\_advisor\_id, aa.name]COUNT(s.student\_id)->student\_count(ADVISOR\_COUNT)