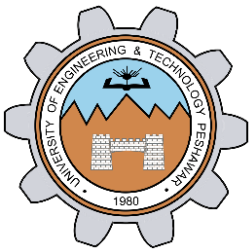
Project Milestone 2



Spring 2025

Group Members

Name	Registration Number
Muhammad Okasha	22PWCSE2146
Muhammad Umar	22PWCSE2148
Sajjad Ahmed	22PWCSE2140

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Summeyea Salahuddin

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Normalized Relational Schema

➢ Step 1: Converted Relational Schema (Unnormalized)

STUDENT(StudentID, Name, Email, Program, Level)
INSTRUCTOR(InstructorID, Name, Email, Department)
COURSE(CourseID, Title, Credits, CoordinatorID → InstructorID)
COURSE_INSTRUCTOR(CourseID, InstructorID)
EXAM(ExamID, Type, Date, CourseID → COURSE.CourseID, ConductedBy → InstructorID)
RESULT(ResultID, StudentID → STUDENT.StudentID, ExamID → EXAM.ExamID, Marks, Grade)
REGISTRATION(StudentID, CourseID)

➢ Step 2: Normalization to 3NF (Step-wise)

1NF - First Normal Form

All attributes in all tables contain atomic values only. No repeating groups or multi-valued attributes exist. Therefore, the schema is in 1NF.

2NF - Second Normal Form

All non-key attributes are fully functionally dependent on the whole primary key. Composite keys appear only in junction tables like REGISTRATION and COURSE_INSTRUCTOR. Hence, the schema is in 2NF.

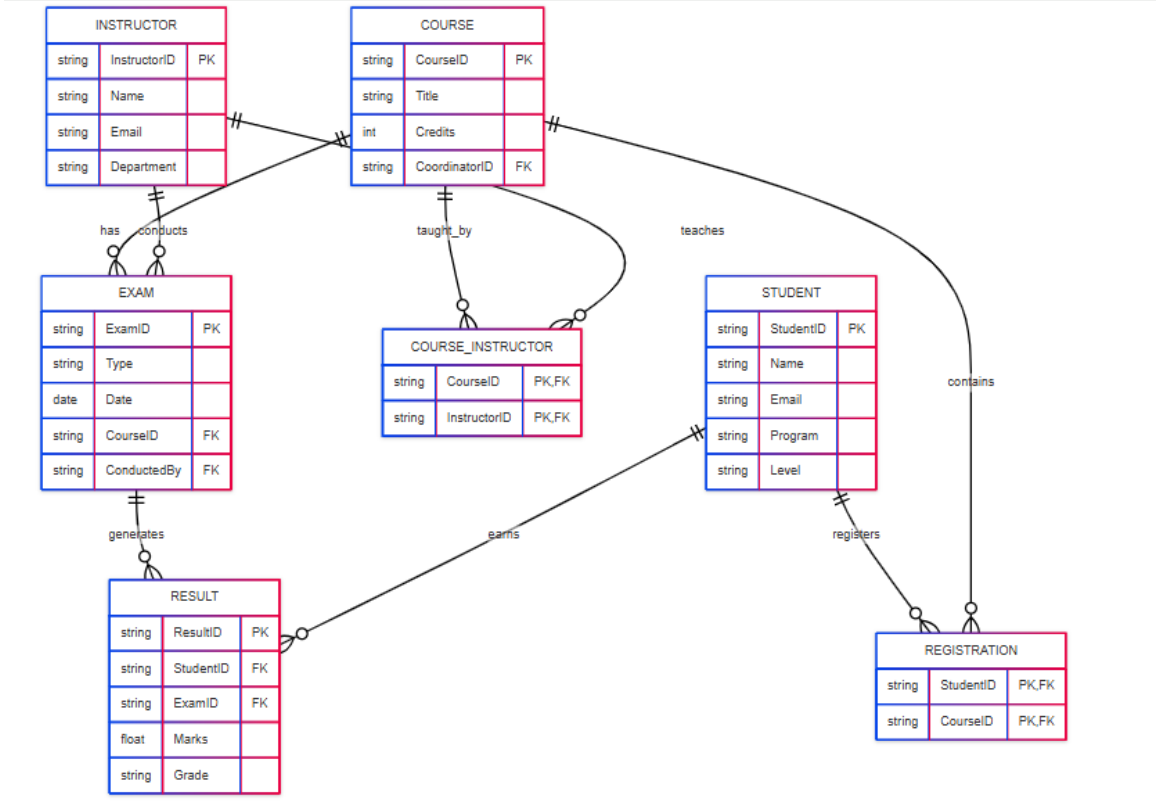
3NF - Third Normal Form

There are no transitive dependencies between non-key attributes and primary keys in any of the tables. Each non-key attribute is dependent only on primary key. Therefore, schema is in 3NF.

➢ Step 3: Final Normalized Relational Schema (3NF)

STUDENT(StudentID, Name, Email, Program, Level)
INSTRUCTOR(InstructorID, Name, Email, Department)
COURSE(CourseID, Title, Credits, CoordinatorID → INSTRUCTOR.InstructorID)
COURSE_INSTRUCTOR(CourseID, InstructorID)
EXAM(ExamID, Type, Date, CourseID → COURSE.CourseID, ConductedBy → INSTRUCTOR.InstructorID)
RESULT(ResultID, StudentID → STUDENT.StudentID, ExamID → EXAM.ExamID, Marks, Grade)
REGISTRATION(StudentID, CourseID)

Figure:


INSTRUCTOR
string InstructorID PK
string Name
string Email
string Department
COURSE
string CourseID PK
string Title
int Credits
string CoordinatorID FK
has
conducts
taught_by
teaches
EXAM
string ExamID PK
string Type
date Date
string CourseID FK
string ConductedBy FK
COURSE_INSTRUCTOR
string CourseID PK,FK
string InstructorID PK,FK
STUDENT
string StudentID PK
string Name
string Email
string Program
string Level
contains
generates
earns
registers
RESULT
string ResultID PK
string StudentID FK
string ExamID FK
float Marks
string Grade
REGISTRATION
string StudentID PK,FK
string CourseID PK,FK