

User_Account(

User_Id PK,
User_Name,
Password,
Role, -- e.g. 'student' or 'instructor'
Is_Active

)**Student(**

User_Id PK, FK -> User_Account(User_Id),
First_Name,
Last_Name,
Std_PhoneNo,
Std_Email,
Std_Gender,
Std_Age,
Std_City,
Std_Street,
Std_ZipCode,
Track_Id FK -> Track(Track_Id)

)**Instructor(**

User_Id PK, FK -> User_Account(User_Id),
Ins_Name,
Ins_Gender,
Ins_PhoneNum,
Ins_Email,
Ins_Salary,
Ins_DOB,
Ins_City,
Ins_Street

)**Track(**

Track_Id PK,
Track_Name

)**Course(**

Crs_Id PK,
Crs_Name,
Instructor_UserId FK -> Instructor(User_Id) -- the instructor who teaches the course

)**Track_Course(**

Track_Id PK, FK -> Track(Track_Id),
Crs_Id PK, FK -> Course(Crs_Id)

)**Topic(**

Topic_Id PK,
Topic_Name

)

Crs_Topic(

Crs_Id PK, FK -> Course(Crs_Id),
Topic_Id PK, FK -> Topic(Topic_Id)

)

Student_Course(

User_Id PK, FK -> Student(User_Id), -- student (from User_Account)
Crs_Id PK, FK -> Course(Crs_Id),
Attend_Status -- e.g. 'enrolled', 'dropped', 'completed'

)

Exam(

Ex_No PK,
Ex_Duration,
Ex_Date,
Crs_Id FK -> Course(Crs_Id)

)

Question(

Qus_No PK,
Qus_Text,
Qus_Type, -- e.g. 'MCQ', 'TF', 'Essay'
Correct_Answer -- (for MCQ may be a choice id or canonical text)

)

Choice(

Choice_Id PK,
Qus_No FK -> Question(Qus_No),
Choice_Text

)

Exam_Question(

Ex_No PK, FK -> Exam(Ex_No),
Qus_No PK, FK -> Question(Qus_No)

)

Student_Exam_Grade(

Ex_No PK, FK -> Exam(Ex_No),
User_Id PK, FK -> Student(User_Id),
Grade

)

Student_Exam_Ans(

ID PK, -- surrogate key for each answer record
Ex_No FK -> Exam(Ex_No),
User_Id FK -> Student(User_Id),
Qus_No FK -> Question(Qus_No),
Choice_Id FK -> Choice(Choice_Id),
Answer_Text

)

