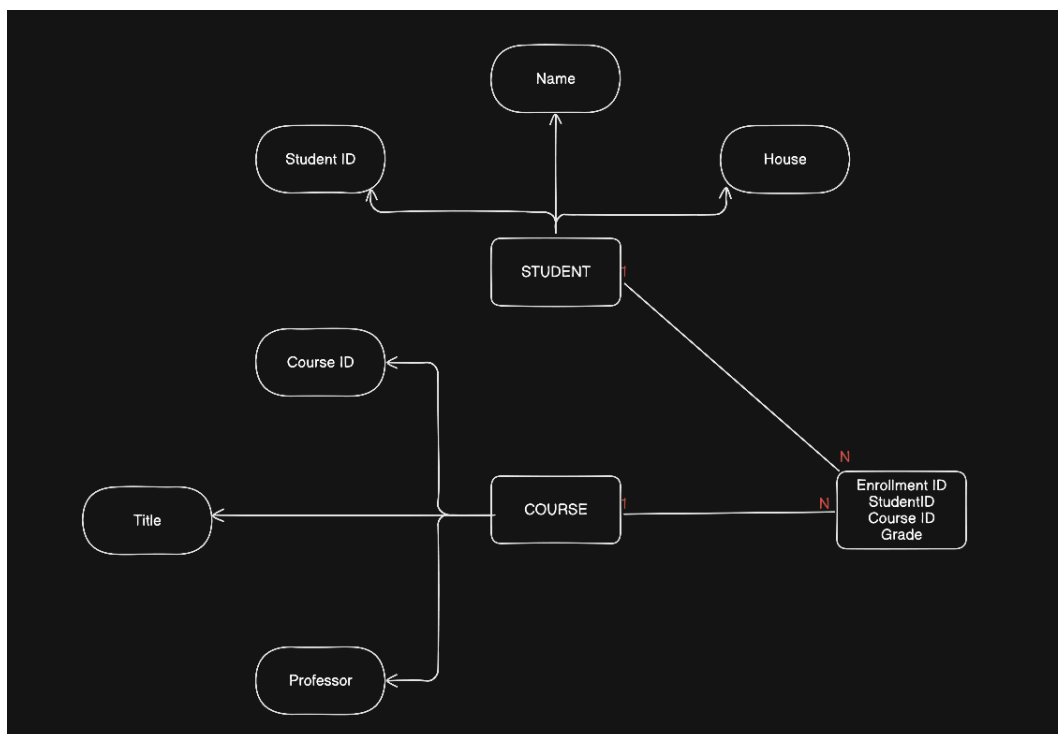


1. ER Diagram



2. Förklaringar och motiveringar

Databasen består av tre primära tabeller. Student, Course och kopplingstabellen Enrollment. Student lagrar all information om elever och Course lagrar info om de kurser som erbjuds. Relationen mellan dessa två är många till många (n-n), eftersom att en student kan läsa flera kurser och en kurs kan läsas av flera studenter. Denna n-n relation löses genom kopplingstabellen Enrollment som fungerar som en länk och etablerar två En-till-många relationer. Enrollment är dessutom utökad för att lagra data om Grades också.

Vi valde INTEGER för alla unika nycklar på grund av dess effektivitet, och VARCHAR för textfält som namn, men det kritiska valet var att använda FOREIGN KEY-samband i kopplingstabellen för att säkerställa dataintegritet. Alla identifierande fält definierades som NOT NULL för att garantera fullständiga data, samtidigt som fältet för Betyg tilläts vara NULL då det sätts vid ett senare tillfälle.

3. Säkerhet

Säker åtkomst till databaser är avgörande eftersom den hanterar känsliga uppgifter som studentbetyg, och ett dataintrång skulle leda till stor skada. För att skydda datan bygger man in två separata spärrar: Authentication och Authorization. Authentication verifierar vem användaren är, men det är Authorization som bestämmer vad den verifierade användaren faktiskt får göra. Detta är kritiskt för att säkerställa att en student får *läsa* sina egna betyg, men absolut inte får *ändra* dem. För att skydda mot de vanligaste attackerna, som SQL Injection, är det nödvändigt att alla frågor som skickas från applikationen är parametriserade. Dessutom måste alla hemligheter, som databaslösenord, hanteras via säkra miljövariabler istället för att lagras direkt i källkoden.

4. Reflektion

Det gick bra att bygga ihop databasen med alla tabeller, och jag lyckades enkelt hantera alla kommandon för att skapa, lägga till och läsa data. Särskilt bra var det att bygga kopplingstabellen (Enrollment) för att hantera studenter och kurser. Det svåraste var att få programmet (SSMS) att förstå vilken databas jag använde, vilket gav röda fel tills jag ändrade inställningen, och jag behövde lägga extra tid på att räkna och gruppera data rätt med GROUP BY. Nästa gång ska jag lägga in en regel i databasen som kollar att betygen bara kan vara godkända bokstäver, och jag ska se till att alla ändringar i databasen sker som en enda säker sak (transaktion) så att ingen data försvinner om något går fel.