**Documentation: Mapping EERD to Actual Database Tables**

This document outlines the process of converting the Enhanced Entity-Relationship Diagram (EERD) into physical database tables using DBML. The goal is to ensure alignment with the EERD structure while adhering to normalization principles and resolving relationships, hierarchies, and constraints.

**1. Super/Sub-Entity Resolution**

**Users (Super Entity)**

* **Table**: AspNetUsers
* **Attributes**:
  + Inherits all shared attributes (e.g., Id, FirstName, LastName, Email, PasswordHash, CountryName, City, State, Age, Gender).
  + Includes fields for ASP.NET Identity (UserName, NormalizedEmail, SecurityStamp, etc.).
  + **Sub-Entities**:
    - **Students**: Mapped to Students table with a 1:1 relationship via Students.Id = AspNetUsers.Id.
    - **Instructors**: Mapped to Instructors table with a 1:1 relationship via Instructors.Id = AspNetUsers.Id.
    - **Admins**: No separate table; uses AspNetUsers directly with role-based permissions.
* **Overlap Handling**: A user can exist in both Students and Instructors tables (e.g., a student who becomes an instructor).

**2. Entity Mappings**

**Roles**

* **Table**: AspNetRoles
* **Attributes**: Id, Name, NormalizedName, ConcurrencyStamp.
* **Relationship**: M:N with AspNetUsers via junction table AspNetUserRoles.

**Notifications**

* **Table**: Notifications
* **Attributes**: Id, Content, audit fields (CreatedDate, IsDeleted).
* **Relationship**: M:N with AspNetUsers via junction table ApplicationUserNotification.

**Questions & Answers**

* **Questions**:
  + **Table**: Asks
  + **Attributes**: Id, Title, Content, CourseId, UserId, audit fields.
  + **Relationships**:
    - 1:M with Answers (Answers.AskId → Asks.Id).
    - M:1 with AspNetUsers (Asks.UserId → AspNetUsers.Id).
    - M:1 with Courses (Asks.CourseId → Courses.Id).
* **Answers**:
  + **Table**: Answers
  + **Attributes**: Id, Content, AskId, UserId, audit fields.
  + **Relationships**: M:1 with Asks and AspNetUsers.

**Students**

* **Table**: Students
* **Attributes**: Id (PK and FK to AspNetUsers.Id), Title, Bio, Wallet.
* **Relationships**:
  + 1:M with Carts (Carts.StudentId → Students.Id).
  + M:N with Courses via Enrollments (with additional attributes like ProgressPercentage).
  + M:N with Quizzes via StudentGrades (with Grade attribute).

**Instructors**

* **Table**: Instructors
* **Attributes**: Id (PK and FK to AspNetUsers.Id), Title, Bio, Wallet, TotalCourses, TotalStudents, TotalReviews.
* **Relationships**: 1:M with Courses (Courses.InstructorId → Instructors.Id).

**Courses**

* **Table**: Courses
* **Attributes**:
  + Core: Id, Title, Description, Status, CourseLevel, Price, Discount, CurrentPrice, Duration, Language, ImageUrl, VideoUrl, Rating, BestSeller.
  + Multi-valued attributes:
    - CourseRequirements: Stored in CourseRequirements (composite PK: Requirement, CourseId).
    - CourseGoals: Stored in CourseGoals (composite PK: Goal, CourseId).
  + Derived: NoSubscribers, IsApproved, IsFree.
* **Relationships**:
  + M:1 with Subcategories (Courses.SubCategoryId → Subcategories.Id).
  + M:N with Orders via CourseOrder (with OrderPrice attribute).
  + M:N with Carts via CartCourse.

**Sections & Lessons**

* **Sections**:
  + **Table**: Sections
  + **Attributes**: Id, Title, Duration, NoLessons, CourseId, audit fields.
  + **Relationships**: 1:M with Lessons (Lessons.SectionId → Sections.Id).
* **Lessons**:
  + **Table**: Lessons
  + **Attributes**: Id, Title, Duration, Type, VideoUrl, ArticleContent, SectionId, audit fields.
  + **Relationships**: M:N with Students via Progresses (tracking lesson completion).

**Quizzes**

* **Table**: Quizzes
* **Attributes**: Id, CourseId, audit fields.
* **Relationships**:
  + 1:M with QuizQuestions (QuizQuestions.QuizId → Quizzes.Id).
  + M:1 with Courses (Quizzes.CourseId → Courses.Id).

**Orders & Carts**

* **Orders**:
  + **Table**: Orders
  + **Attributes**: Id, PaymentMethod, Status, TotalAmount, Discount, audit fields.
  + **Relationships**: M:N with Courses via CourseOrder (with OrderPrice).
* **Carts**:
  + **Table**: Carts
  + **Attributes**: Id, StudentId, Amount, audit fields.
  + **Relationships**: M:N with Courses via CartCourse.

**Categories & Subcategories**

* **Categories**:
  + **Table**: Categories
  + **Attributes**: Id, Name, audit fields.
* **Subcategories**:
  + **Table**: Subcategories
  + **Attributes**: Id, Name, CategoryId, audit fields.
  + **Relationships**: M:1 with Categories (Subcategories.CategoryId → Categories.Id).

**3. Relationship Mappings**

**M:N Relationships**

* **User ↔ Role**: AspNetUserRoles (composite PK: UserId, RoleId).
* **User ↔ Notification**: ApplicationUserNotification (composite PK: NotificationsId, UsersId).
* **Student ↔ Course**: Enrollments (composite PK: StudentId, CourseId with attributes like ProgressPercentage).
* **Order ↔ Course**: CourseOrder (composite PK: OrderId, CourseId with OrderPrice).
* **Cart ↔ Course**: CartCourse (composite PK: CartId, CourseId).

**Hierarchical Relationships**

* **User → Student/Instructor**: Resolved via 1:1 relationships using shared Id.
* **Category → Subcategory**: Subcategories.CategoryId → Categories.Id.

**Composite/Multi-Valued Attributes**

* **Course Requirements**: CourseRequirements table (composite PK).
* **Course Goals**: CourseGoals table (composite PK).
* **Social Media**: SocialMedias table (UserId, Name, Link).

**4. Special Cases & Constraints**

**Overlap (Students ↔ Instructors)**

* Allowed by sharing the same Id in Students and Instructors, both referencing AspNetUsers.Id.

**Derived Attributes**

* **Cart.Amount**: Calculated as the count of courses in CartCourse.
* **Enrollments.ProgressPercentage**: Computed from completed lessons.
* **Courses.CurrentPrice**: Derived from Price and Discount.

**Soft Deletion**

* All tables include IsDeleted to enable soft deletion.

**Audit Fields**

* CreatedDate, ModifiedDate added to all tables for tracking.

**5. Key Design Decisions**

1. **ASP.NET Identity Integration**:
   * Used AspNetUsers, AspNetRoles, and related tables to leverage built-in authentication/authorization.
2. **Composite Keys**:
   * Used for multi-valued attributes (e.g., CourseRequirements, CourseGoals) and M:N relationships (e.g., CourseOrder).
3. **Inheritance via Shared Primary Key**:
   * Students.Id and Instructors.Id = AspNetUsers.Id to enforce subtype relationships.
4. **Progress Tracking**:
   * Progresses table tracks lesson completion status for students.

**6. Deviations from EERD**

1. **Admin Entity**:
   * No separate table; admins are AspNetUsers with a role (AspNetRoles.Name = "Admin").
2. **Quiz Date**:
   * Quizzes.CreatedDate replaces QuizDate for audit purposes.
3. **BestSeller**:
   * Stored as nvarchar(20) instead of a derived attribute (to allow manual curation).