ERD - Cardinality Kirkyagami

Lecture Notes: Relationships in MySQL

Introduction

In database design, relationships between tables are crucial for organizing and structuring data efficiently. We'll explore three types of relationships: one-to-one (1:1), one-to-many (1:many), and many-to-many (many-to-many).

1. One-to-One (1:1) Relationship

A one-to-one relationship exists when each record in Table A corresponds to exactly one record in Table B, and vice versa.

Example: A person and their passport

Person
person_id
name
date_of_birth

Passport
passport_id
person_id
issue_date
expiry_date

CREATE statements:

```
CREATE TABLE Person (
    person_id INT PRIMARY KEY,
    name VARCHAR(100),
    date_of_birth DATE
);
```

```
CREATE TABLE Passport (
    passport_id INT PRIMARY KEY,
    person_id INT UNIQUE,
    issue_date DATE,
    expiry_date DATE,
    FOREIGN KEY (person_id) REFERENCES Person(person_id)
);
```

Note: The `UNIQUE` constraint on `person_id` in the Passport table ensures the one-to-one relationship.

2. One-to-Many (1:many) Relationship

A one-to-many relationship exists when a record in Table A can be associated with multiple records in Table B, but each record in Table B is associated with only one record in Table A.

Example: An author and their books

Author
author_id
name
nationality

Book
book_id
title
author_id
publish_date

CREATE statements:

```
CREATE TABLE Author (
    author_id INT PRIMARY KEY,
    name VARCHAR(100),
    nationality VARCHAR(50)
);

CREATE TABLE Book (
    book_id INT PRIMARY KEY,
    title VARCHAR(200),
    author_id INT,
    publish_date DATE,
    FOREIGN KEY (author_id) REFERENCES Author(author_id)
);
```

3. Many-to-Many (many-to-many) Relationship

A many-to-many relationship exists when multiple records in Table A can be associated with multiple records in Table B, and vice versa. This relationship typically requires a junction table.

Example: Students and courses

Student
student_id
name
email

Course
course_id
title
credits

Enrollment
student_id
course_id
semester

CREATE statements:

```sql
CREATE TABLE Student (
    student_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100)
);

CREATE TABLE Course (
    course_id INT PRIMARY KEY,
    title VARCHAR(200),
    credits INT
);

CREATE TABLE Enrollment (
    student_id INT,
    course_id INT,
    semester VARCHAR(20),
    PRIMARY KEY (student_id, course_id),
    FOREIGN KEY (student_id) REFERENCES Student(student_id),
    FOREIGN KEY (course_id) REFERENCES Course(course_id)
);
```

Note: The Enrollment table serves as a junction table, connecting students and courses. The primary key is a combination of `student_id` and `course_id` to ensure unique enrollment records.

Conclusion

Understanding these relationships is essential for effective database design:

- One-to-One (1:1) relationships connect two tables with a unique relationship between records.
- One-to-Many (1:many) relationships allow a record in one table to be associated with multiple records in another table.
- Many-to-Many (many-to-many) relationships require a junction table to connect multiple records from both tables.

Proper implementation of these relationships ensures data integrity, reduces redundancy, and allows for efficient querying and data management in MySQL databases.