Strong and Weak Relationships and Entities

by Sophia

1. Relationship Strength

Relationship strength refers to the degree of association or dependency between entities. It's important to understand the concept of relationship strength so that you can structure and optimize your databases to accurately represent real-life relationships. An entity's relationship strength determines how it is connected, which has significant implications for data integrity, retrieval efficiency, and database performance.

⭐ BIG IDEA

Data integrity relies on how relationships are formed inside the database. The strength of relationships between entities determines the type of cardinality constraints that need to be applied between entities. You can avoid inconsistencies or data anomalies in your database by setting the right cardinality (e.g., one-to-one, one-to-many, many-to-many). Choosing the wrong cardinality for a relationship, such as using a many-to-many relationship instead of one-to-many, may lead to duplicate records or orphaned data, compromising data integrity.

Relationship strength determines the efficiency of data retrieval. One-to-many relationships, for example, can be leveraged to reduce the number of joins required by optimizing queries. When queries are efficient, they are able to retrieve data faster, which is imperative for applications dealing with large volumes of data and requiring real-time responses.

Understanding relationship strength can help optimize storage space. Using foreign keys in one-to-many relationships can minimize data duplication, leading to a more compact and efficient database schema.

The strength of relationships also influences the implementation of business logic in the database. Depending on the type of relationship, triggers, constraints, or referential integrity rules may be required to enforce business rules and maintain data consistency.

In order to perform complex data analysis and generate meaningful reports, it is important to understand the relationship's strength. Analyzing data in this way provides valuable insights into patterns, trends, and dependencies among entities.

> 🖌 **KEY CONCEPT**
>
> There are two types of relationships between tables: strong and weak. Chen notation does not distinguish the difference between a weak and strong relationship in a diagram. In crow's foot notation, a dashed relationship line is used to signify a weak relationship. This is not always used in diagramming ERDs. However, it is useful to know this if you do see it in a diagram.

> 📄 **TERM TO KNOW**
>
> **Relationship Strength**
> The degree of association or dependency between entities in a relationship.

2. Strong and Weak Entities and Relationships

As you learned in Unit 1, a **strong entity** is an entity that can be uniquely identified by its own attributes, meaning it has a primary key. It doesn't rely on another entity for identification. For example, in our movie ratings database example, the Movie table and the User table are both strong entities. Each movie and each user exist independently of any relationship.

A **weak entity** is an entity that cannot be uniquely identified by only its own attributes. It does not have a primary key that uniquely identifies it. It is dependent on one or more strong entities. It typically has a set of attributes that, in combination with the attributes of its parent entity, forms a unique identifier. Those attributes are referred to as its partial key. In the movie ratings database, the Rating table is a weak entity because it relies on the Movie and User tables. Each of those tables has a one-to-many relationship with Ratings, with the Ratings table on the "many" side of each. In Chen notation, a weak entity is indicated by a double line around the entity's rectangle.

In relationships, a **strong relationship** is one between two strong entities. A **weak relationship** is one between a strong entity and a weak entity.

> **Strong relationships meet these criteria:**   +

- Both participating entities in the relationship must have their own primary keys.

- Each participating entity can be identified independently of the other entity based on its own attributes and its primary key.

- Both entities participate equally. Neither entity plays a dependent role.

- The relationship does not rely on the existence of another entity to establish or maintain its integrity.

- The relationship must enforce referential integrity between related entities. Referential integrity ensures that any foreign key value in the child entity must have a corresponding primary key value in the parent entity. This prevents orphaned records and ensures consistent and valid data.

Weak relationships have these characteristics: +

- They consist of one strong and one weak entity. The strong entity is the parent entity, and the weak entity is the child entity.

- The weak entity is dependent on the strong entity. In the absence of the strong entity, the weak entity's data is not complete.

- The weak entity in the relationship does not have a primary key that uniquely identifies it on its own.

- The weak entity plays a subsidiary or dependent role in the relationship, relying on the strong entity in the relationship for context and significance.

⇗ EXAMPLE Look at the Rating entity in your current design (below). The Rating entity is existence-dependent. It is a bridge entity between User and Movie, which are both strong entities. It has dependencies on both the user and the movie. A rating cannot exist without having a user that submitted the rating. A rating also cannot exist without being connected to a movie. You have not normalized the database yet to create a bridge entity between movie/actor and movie/genre, but those bridge entities would also be weak entities.
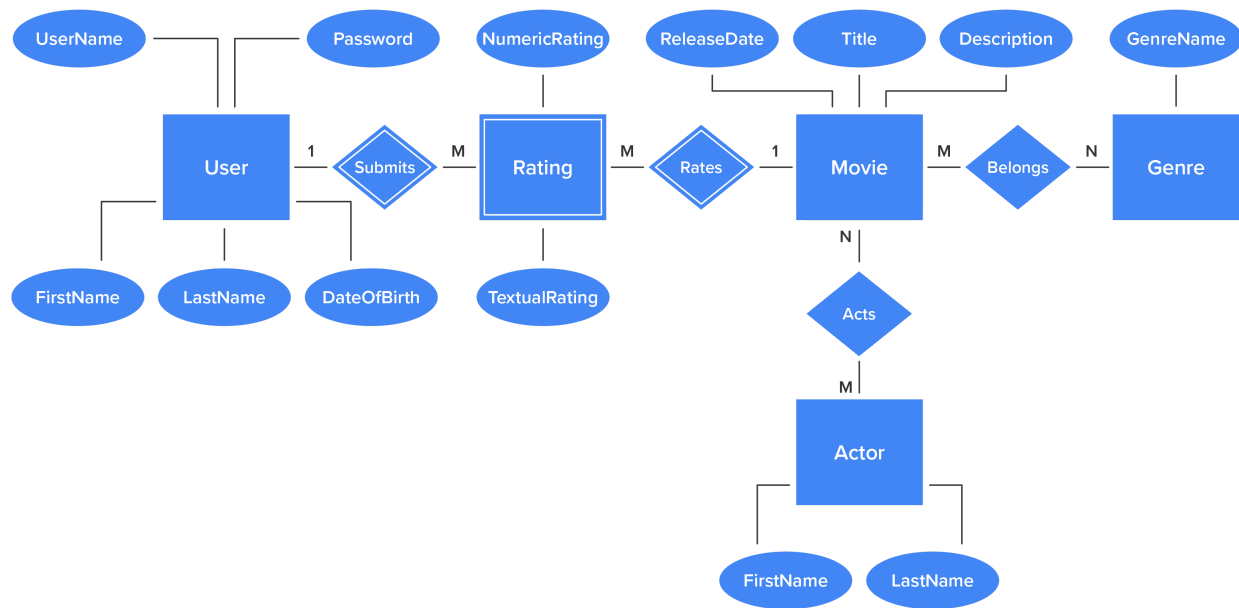
Strong Relationship

    A relationship between two strong entities.

Weak Relationship

    A relationship between a strong entity and a weak entity.

Strong Entity

    An entity that can be uniquely identified by its own attributes alone.

Weak Entity

    An entity that cannot be uniquely identified by only its own attributes.

📋 SUMMARY

In this lesson, you learned about **relationship strength,** which refers to the degree of association or dependency between entities. It is important to understand **strong and weak entities and relationships** because those classifications affect the database's design, which in turn affects its performance, integrity, and usability.

You reviewed the properties of strong and weak entities. A strong entity is one that can be uniquely identified by its own attributes and has its own primary key. An example is the Movie table in the movie ratings database. A weak entity is one that cannot be uniquely identified by only its own attributes. An example is the Rating table, which is dependent on both the User and Movie tables.

You learned that relationships can also be categorized as weak or strong. A strong relationship is one between two strong entities. Both must have their own primary key and must participate equally with

neither playing a dependent role. The relationship must enforce referential integrity.

A weak relationship is one between a strong entity and a weak entity. The weak entity is dependent on the strong one and has no primary key of its own. In the movie ratings database, the relationships between the Rating table and the User and Movie tables are weak relationships.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND Faithe Wempen (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR TERMS OF USE.

📄 TERMS TO KNOW

Relationship Strength
The degree of association or dependency between entities in a relationship.

Strong Entity
An entity that can be uniquely identified by its own attributes alone.

Strong Relationship
A relationship between two strong entities.

Weak Entity
An entity that cannot be uniquely identified by only its own attributes.

Weak Relationship
A relationship between a strong entity and a weak entity.