**Existence dependence** refers to the relationship between entities where the existence of one entity depends on the existence of another entity. In database terms, this often involves a foreign key constraint where the existence of a child record depends on the existence of a parent record.

Employee table: The existence of an employee depends on the existence of a corresponding record in the Person table, as indicated by the foreign key constraint between Employee.User\_id and Person.id.

GroupEmployee table: The existence of a group employee record depends on the existence of both a group (in the Groups table) and an employee (in the Employee table). This is indicated by the foreign key constraints between GroupEmployee.Group\_id and Groups.id, as well as between GroupEmployee.Employee\_id and Employee.Empl\_id.

Evaluation table: The existence of an evaluation record depends on the existence of both an employee (in the Employee table) and a metric (in the Metrics table). This is indicated by the foreign key constraints between Evaluation.Employee\_id and Employee.Empl\_id, as well as between Evaluation.Metric\_id and Metrics.Met\_id.

GroupEvaluation table: The existence of a group evaluation record depends on the existence of both a group (in the Groups table) and a metric (in the Metrics table). This is indicated by the foreign key constraints between GroupEvaluation.Group\_id and Groups.id, as well as between GroupEvaluation.Metric\_id and Metrics.Met\_id.

**Strong relationships** in a database refer to relationships where the existence of a dependent entity (child table) is directly tied to the existence of a related entity (parent table). This is typically enforced through a foreign key constraint in the database schema.

Employee and Person: An employee record depends on a corresponding person record.

GroupEmployee and Groups, Employee: A group employee record depends on both a group and an employee record.

Evaluation and Employee, Metrics: An evaluation record depends on both an employee and a metric record.

GroupEvaluation and Groups, Metrics: A group evaluation record depends on both a group and a metric record.

**Weak relationships** in a database refer to relationships where the existence of a dependent entity (child table) is not directly tied to the existence of a related entity (parent table). This is often implemented using optional foreign keys, where the foreign key column in the child table allows NULL values.

A **weak entity** is an entity in a database that does not have a primary key of its own. Instead, it relies on the existence of another entity, called the identifying entity, and is identified by a partial key, which is a set of attributes that uniquely identifies the weak entity within the context of the identifying entity.

In the Employee Performance Evaluation system schema, there are no weak entities.

A **strong entity** is an entity in a database that has a primary key and does not depend on another entity for its existence or identity. In the Employee Performance Evaluation system schema, all the tables can be considered strong entities because each table has its own primary key and does not rely on another table for its identity.

**Relationship participati**on refers to the involvement of entities in a relationship. It can be total (mandatory) or partial (optional) for both entities in a relationship.

**Total Participation:** In the relationship between Employee and Evaluation, and Groups and GroupEvaluation, total participation exists. This means that every employee must have at least one evaluation, and every group must have at least one group evaluation. This is enforced by the foreign key constraints in the Evaluation and GroupEvaluation tables, which require references to existing Employee and Groups records respectively.

**Partial Participation**: In the relationship between Groups and Supervisor, partial participation exists. This means that a group may or may not have a supervisor. The foreign key constraint in the Groups table referencing the Supervisor table allows for NULL values, indicating that a group can exist without a supervisor.

there are no explicit **recursive relationships**, where an entity is related to itself through a foreign key. Each table relates to other tables in the schema, but there are no direct self-referencing relationships within a single table.

**Associative entitie**s are entities that are used to represent a many-to-many relationship between two or more other entities. They are typically created to resolve a many-to-many relationship into two one-to-many relationships by introducing a new entity.

In the Employee Performance Evaluation system schema, the GroupEmployee table can be considered an associative entity. It resolves the many-to-many relationship between Groups and Employee by creating two one-to-many relationships: Groups to GroupEmployee and Employee to GroupEmployee. This table stores additional information about the relationship between groups and employees, such as the date an employee joined a group.

**Normalization** is the process of organizing data in a database to reduce redundancy and dependency by dividing large tables into smaller ones and defining relationships between them. There are several normal forms, each building on the previous one to ensure data integrity and eliminate data anomalies.

First Normal Form (1NF): All tables have a primary key, and each column contains atomic values. For example, the Person table has a primary key (id) and columns like Email, PhoneNumber, and Username contain atomic values.

Second Normal Form (2NF): All non-key attributes are fully functional dependent on the primary key. For example, in the Employee table, Full\_name is dependent on Empl\_id (primary key) and not on any subset of the primary key.

Third Normal Form (3NF): All attributes are functionally dependent only on the primary key, and no transitive dependencies exist. For example, in the Employee table, Department\_id is dependent on Empl\_id (primary key) and not on any other non-key attribute.