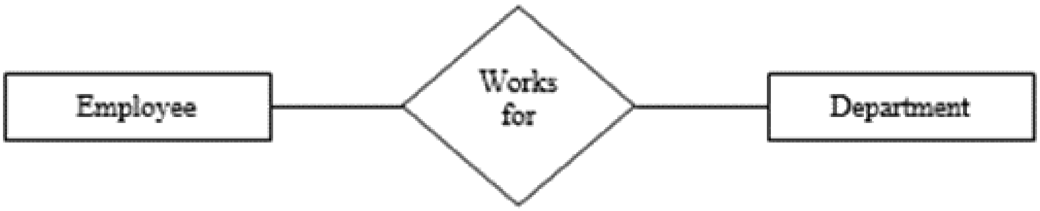
Relationship Review in OOP and Database Design: Constraints, Roles, Weak and Strong Entities, Notation for Competitive Exams 2021

Relationship Review

- Each time an attribute of one entity type refers to another entity type, some relationship exists.
  - In ER diagrams, these references should be represented as relationships, rather than attributes.
  - For example, in the Company database schema, an attribute of employee is the department they work for, rather than representing this information as an attribute of the Employee entity type, it should be represented on a diagram as a relationship between the two entities.
  - Relationships between entities are represented using a diamond shape.
  - Relationships are usually given a verb name, which specifies the relationship between two entities.
  - If we look at the relationship between Employee and Department, an employee works for a department, therefore the relationship would be represented

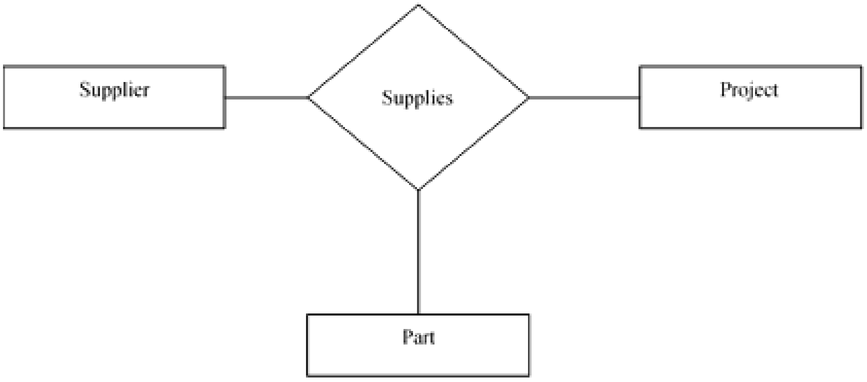


©Examrace. Report ©violations @https://tips.fbi.gov/

Degree of Relationship Type

- The degree of a relationship type is the number of participating entity types. Meaning if the relationship is between two entity types (Employee and Department) , then the relationship is binary, or has a degree of two.

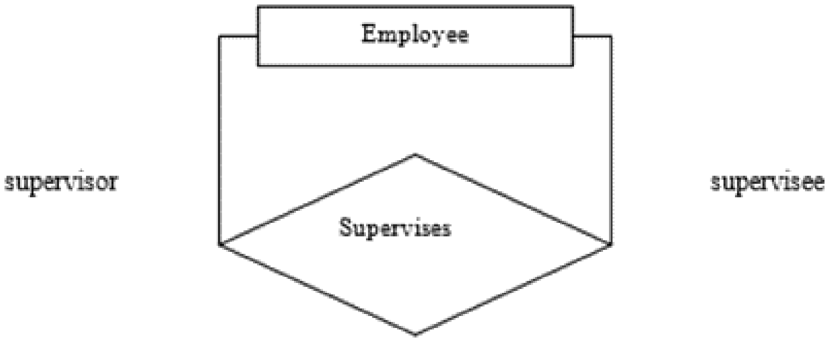
- If the relationship is between three participating entities, it has a degree of three, and therefore is a ternary relationship.
- For example, if we have three entities, Supplier, Project and Part. Each part is supplied by a unique supplier, and is used for a given project within a company; the relationship “Supplies” is a ternary (degree of three) between Supplier, Project and Part, meaning all three participate in the supplies relationship.



©Examrace. Report ©violations @https://tips.fbi.gov/

Role Names and Recursive Relationships

- Each entity type in a relationship plays a particular role. The role name specifies the role that a participating entity type plays in the relationship and explains what the relationship means.
    - For example, in the relationship between Employee and Department, the Employee entity type plays the employee role, and the Department entity type plays the department or employer role.
    - In most cases the role names do not have to be specified, but in cases where the same entity participates more than once in a relationship type in different roles.
    - For example, in the Company schema, each employee has a supervisor, we need to include the relationship “Supervises” , however a supervisor is also an employee, therefore the employee entity type participates twice in the relationship, once as an employee and once as a supervisor, therefore we can specify two roles, employee and supervisor.



©Examrace. Report ©violations @https://tips.fbi.gov/

Constraints on Relationship Types

- Relationship types have certain constraints that limit the possible combination of entities that may participate in relationship.
  - An example of a constraint is that if we have the entities Doctor and Patient, the organization may have a rule that a patient cannot be seen by more than one doctor. This constraint needs to be described in the schema.
  - There are two main types of relationship constraints, cardinality ratio, and participation.

Cardinality for Binary Relationship

- Binary relationships are relationships between exactly two entities.
  - The cardinality ratio specifies the maximum number of relationship instances that an entity can participate in.
  - The possible cardinality ratios for binary relationship types are: 1: 1,1: N, N: 1, M: N.
  - Cardinality ratios are shown on ER diagrams by displaying 1, M and N on the diamonds.
  - The ratio shown closest to an entity, represents the ratio the other entity has to that entity.

Participation Constraints and Existence Dependencies

- The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type.
  - The constraint specifies the minimum number of relationship instances that each entity can participate in.
  - There are two types of participation constraints:
    - **Total**:

        - If an entity can exist, only if it participates in at least one relationship instance, then that is called total participation, meaning that every entity in one set, must be related to at least one entity in a designated entity set.
        - An example would be the Employee and Department relationship. If company policy states that every employee must work for a department, then an employee can exist only if it participates in at lest one relationship instance (i.e.. … an employee can't exist without a department)
        - It is also sometimes called an existence dependency.
        - Total participation is represented by a double line, going from the relationship to the dependent entity.
    - **Partial**:
        - If only a part of the set of entities participate in a relationship, then it is called partial participation.
        - Using the Company example, every employee will not be a manager of a department, so the participation of an employee in the "Manages" relationship is partial.
        - Partial participation is represented by a single line.

Attributes of Relationship Types

- Relationships can have attributes similar to entity types.
    - For example, in the relationship Works_On, between the Employee entity and the Department entity we would like to keep track of the number of hours an employee works on a project. Therefore we can include Number of Hours as an attribute of the relationship.
    - Another example is for the "manages" relationship between employee and department, we can add Start Date as an attribute of the Manages relationship.
    - For some relationships (1: 1, or 1: N) , the attribute can be placed on one of the participating entity types. For example the "Manages" relationship is 1: 1, StartDate can either be migrated to Employee or Department.

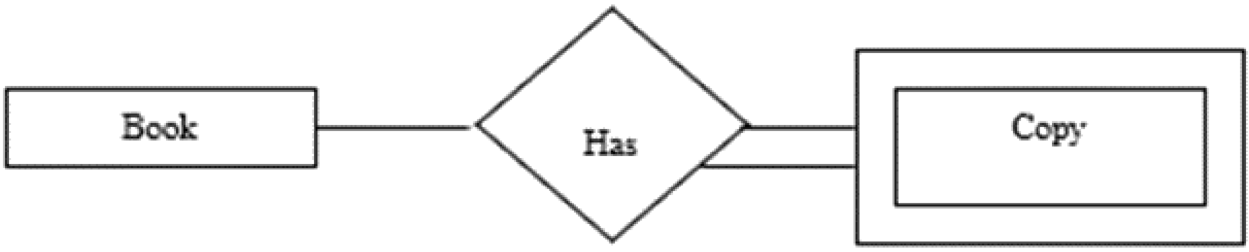
Weak Entity Types

- Entity types that do not have key attributes are called weak entity types.
    - Entities that belong to a weak entity type are identified by being related to specific entities from another entity type in combination with one of their attribute values.
    - This entity type is called an identifying or owner entity type.
    - The relationship that relates the identifying entity type with the weak entity type is called an identifying relationship.

  - A weak entity type always has a total participation constraint with respect to the identifying relationship, because a weak entity cannot exist without its owner.
  - Not all existence dependencies result in a weak entity type; if an entity has a key attribute then it is not a weak entity.
  - A weak entity type usually has a partial key, which is the set of attributes that can uniquely identify weak entities that are **related to the same owner entity**.

Weak Entity Example

- For example, lets assume in a library database, we have an entity type Book. For each book, we keep track of the author, ISBN, and title. The library may own several copies of the same book, and for each copy, it keeps track of the copy number (a different copy number for each copy of a given book) and price of each copy.



©Examrace. Report ©violations @https://tips.fbi.gov/

  - Because the copy number is only unique for each book (meaning Book 123 may have copy 1, copy 2, copy 3, and book 456 may also have copy 1, copy 2 and copy 3) and not for all copies of all books, it cannot be considered unique for each copy.
  - Therefore because the Copy entity does not have a key attribute, it is considered a weak entity type, an is identified by being related to the Book entity. The book entity is the identifying entity, and the relationship is the identifying relationship.
  - Because a copy cannot exist without the owner (Book) the Copy entity type has a total participation constraint with respect to the identifying relationship.
  - The partial key of the Copy entity is Copy Number, for each owner entity Book, the Copy Number uniquely identifies the copy.

Min-Max Notation

- Before we saw that to specify structural constraints (cardinality) we used the M: N notation.
  - An alternate notation involves specifying a pair of integers, which are used to specify the minimum and maximum participation of each entity type in the form of (min, max)

  - A minimum participation of 0 indicates partial participation (meaning that there may be some entities that do not participate in the relationship)
  - A minimum participation of 1 or more indicates total participation, meaning that each entity must participate in exactly/at least one-relationship type.
  - See PowerPoint "ERD_Examples. ppt" , Slide 1 to demonstrate.

Notation Summary

- See Slide 2 and Slide 3 of PowerPoint "ERD_Examples. ppt" to demonstrate.
  - To demonstrate using Min-Max notation, see slide 4.

Examples

University Example – Exercise 3.16

- See PowerPoint "ERD_Examples. ppt" Slide 5 to demonstrate.
  - Consider the following set of requirements for a university database that is used to keep track of student's transcripts:
  - The university keeps track of each student's name, student number, social security number, current address and phone number, permanent address and phone number, birthdate, sex, class (freshman, graduate) , major department, minor department (if any) , degree program (B. A. , B. S. , ... Ph. D.) . Some user applications need to refer to the city, state, and zip code of the student's permanent address and to the student's last name. Both social security number and student number are unique for each student. All students will have at least a major department.
  - Each department is described by a name, department code, office number, office phone, and college. Both the name and code have unique values for each department.
  - Each course has a course name, description, course number, number of credits, level and offering department. The course number is unique for each course.
  - Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its value is an integer (1,2, 3, ... up to the number of sections taught during each semester) .
  - A grade report must be generated for each student that lists the section, letter grade, and numeric grade (0, 1,2, 3, or 4) for each student and calculates his or her average GPA.

Bank Example - Exercise 3.23

- See PowerPoint "ERD_Examples. ppt" to demonstrate.
- Based on the ER diagram shown, answer the following questions.
  - List the non-weak entity types in the diagram.
    - Bank

        - Loan
        - Account
        - Customer
    - For any weak entities, list the name, partial key, and identifying relationship.
        - Name: Bank-Branch
        - Partial Key: Branch No
        - Identifying Relationship: Branches
    - What constraints do the partial key and the identifying relationship of the weak entity type specify in the diagram?
        - Every bank must have at least one branch, and each bank branch belongs to one and only one bank. Together the branch number and bank code uniquely identify a branch.
    - List the names of all relationship types, and specify the (min, max) constraint on each participation of an entity type in a relationship type.
        - Branches, Accts, Loans, A-C, L-C.
        - See diagram for constraints.
- List the user requirements that lead to the diagram.
- Suppose that each customer must have at least one account, but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1000 loans. How does this show up as min-max constraints.
    - Customer-Loan (0,2)
    - Customer-Account (1, N)
    - Bank-Branch – Loan (0,1000)

Developed by: Mindsprite Solutions