

Entity – Relationship Model

Saraswathy

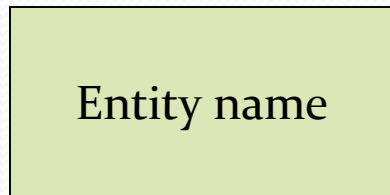


Objectives

- How to use ER modeling in database design.
- Basic concepts about ER model.
- A diagrammatic technique for displaying an ER model using the Unified Modeling Language (UML).
- How to identify and resolve problems with ER models.
- How to build an ER model from a requirements specification.

Entity Types

- **Entity Type**: A group of objects with the same properties, which are identified by the enterprises as having an independent existence.
- **Entity Occurrence**: A uniquely identifiable object of a entity type.



UML representation of a entity

- Rectangle labeled with the name of the entity.
- In UML representation, the first letter of the entity name is a capital letter.

Relationship Types

- **Relationship Type**: A set of meaningful associations among entity types.
- **Relationship Occurrence**: A uniquely identifiable association, which includes one occurrence from each participating entity type.

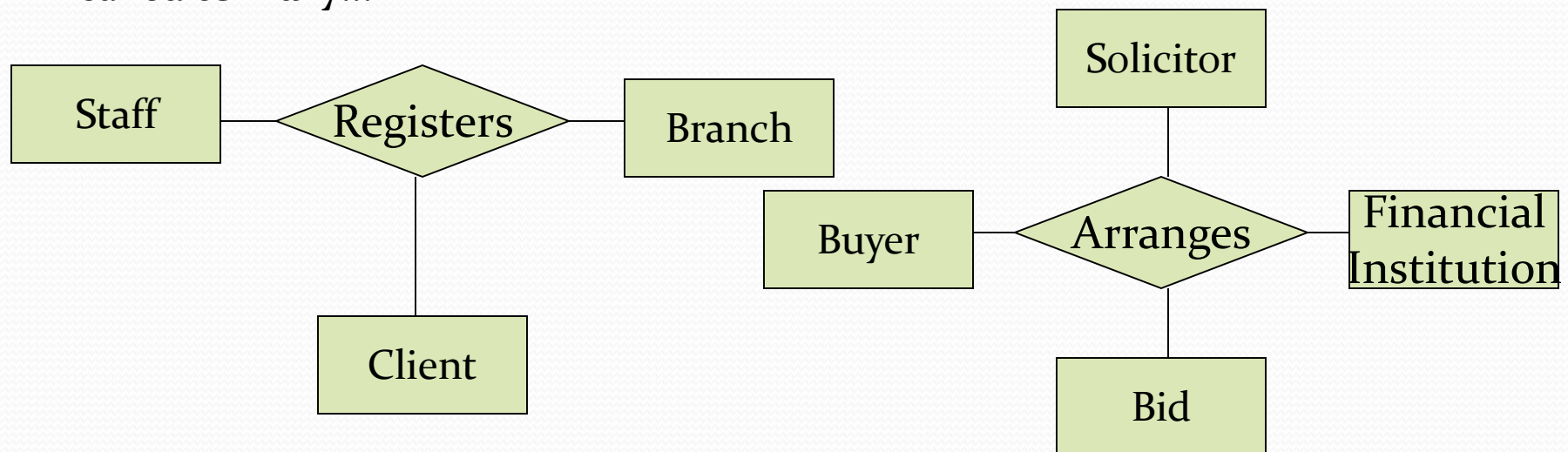


Branch has a Staff

- First capital letter.
- Only labeled in one direction.
- Arrow symbol

Degree of Relationship Type

- **Degree of Relationship type**: the number of participating entity types in a relationship.
- A relationship of degree two is called **binary**, a relationship of degree three is called **ternary**...

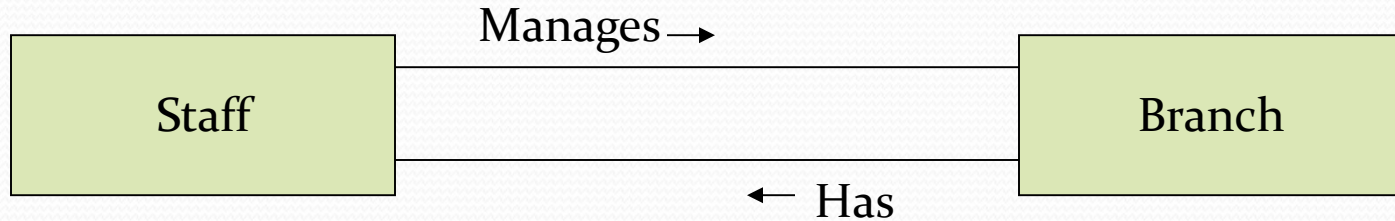


“Staff registers a client at a branch”

“A solicitor arranges a bid on behalf of a buyer supported by a financial institution”

Recursive Relationship

- **Recursive Relationship**: A relationship type where the same entity type participates more than once in different roles.





Attributes

- **Attribute**: A property of an entity or a relationship type. For example: staffNo, name, position... To describe the entity Staff.
- **Attribute Domain**: The set of allowable values for one or more attributes.
- Attributes can be classified as being
 - *simple or composite*
 - *single-valued or multi-valued*
 - *derived*.

Simple and Composite Attributes

- **Simple Attribute:** An attribute composed of a single component with an independent existence.
 - E.g position and salary of the Staff entity.
- **Composite Attribute:** An attribute composed of multiple components, each with an independent existence.
 - E.g address attribute of the branch entity that can be subdivided into street, city and postcode attributes.

Single-Valued and Multi-Valued Attributes

- **Single-Valued Attribute:** An attribute that holds a single value for each occurrence of an entity type.
 - E.g branchNo.
- **Multi-Valued Attributes:** An attribute that holds multiple values for each occurrence of an entity type.
 - E.g telephoneNo.



Derived Attributes

- **Derived Attributes:**

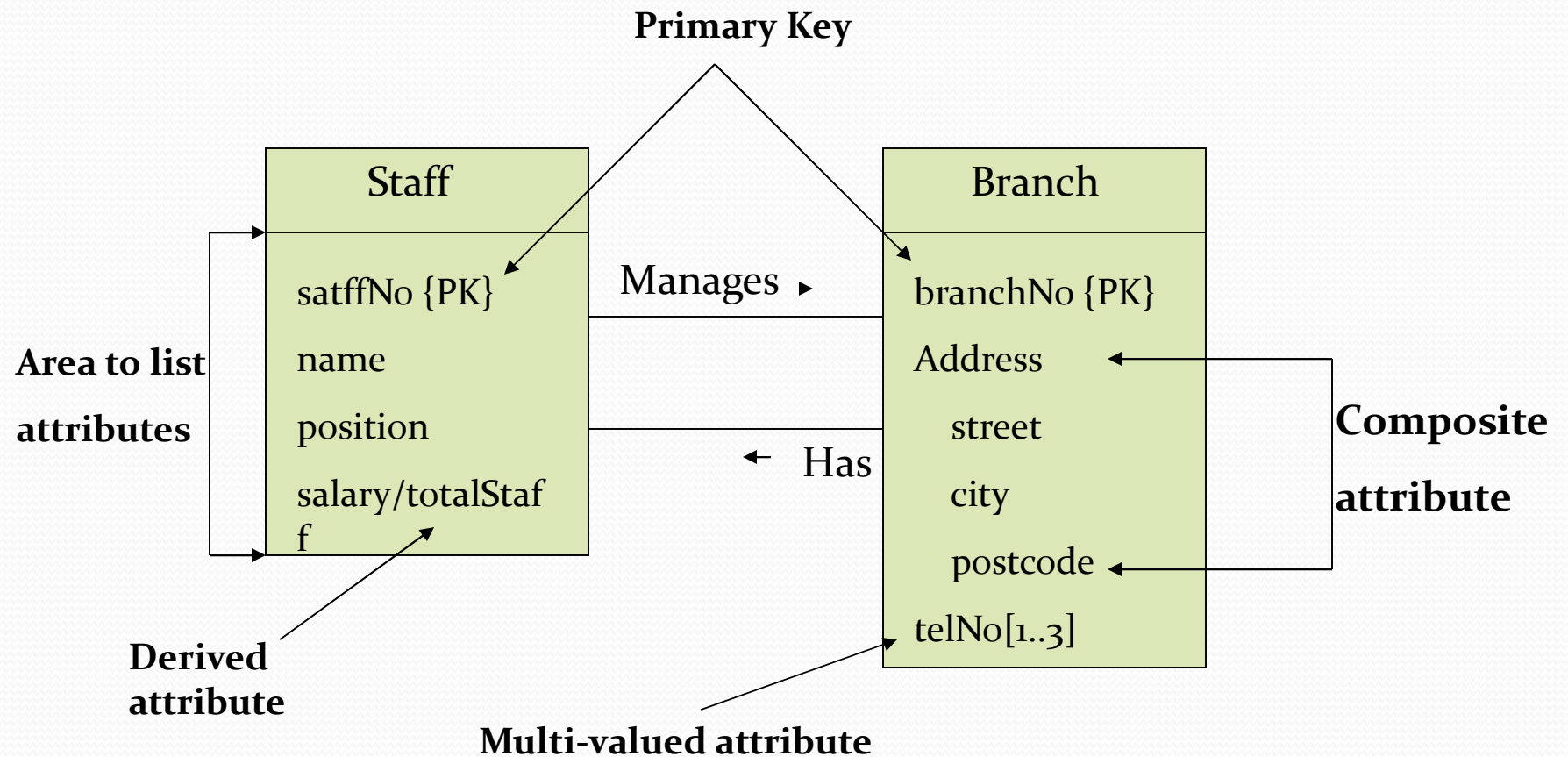
- An attribute that represents a value that is derivable from the value of a related attribute or set of attributes.
- Not necessarily in the same entity type.
- E.g attribute duration which value is derived from the rentStart and rentFinish attributes.



Keys

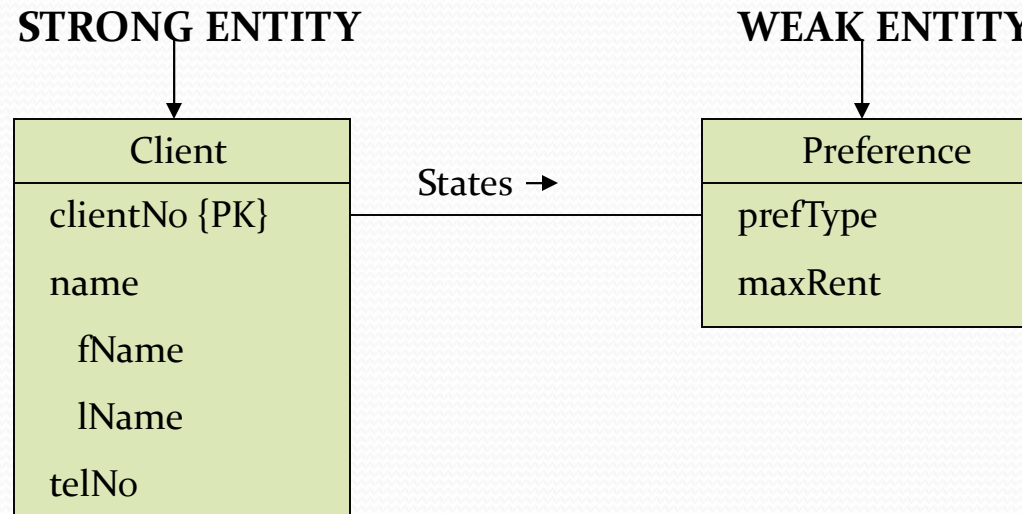
- **Candidate Key** (never NULL): The minimal *set* of attributes that uniquely identifies each occurrence of an entity type. E.g: branchNo in entity Branch.
- **Primary Key**: The candidate key that is selected to uniquely identify each occurrence of an entity type.
 - E.g: National Insurance Number.
- **Composite Key**: A candidate key that consist of two or more attributes.

Diagrammatic Representation of attributes



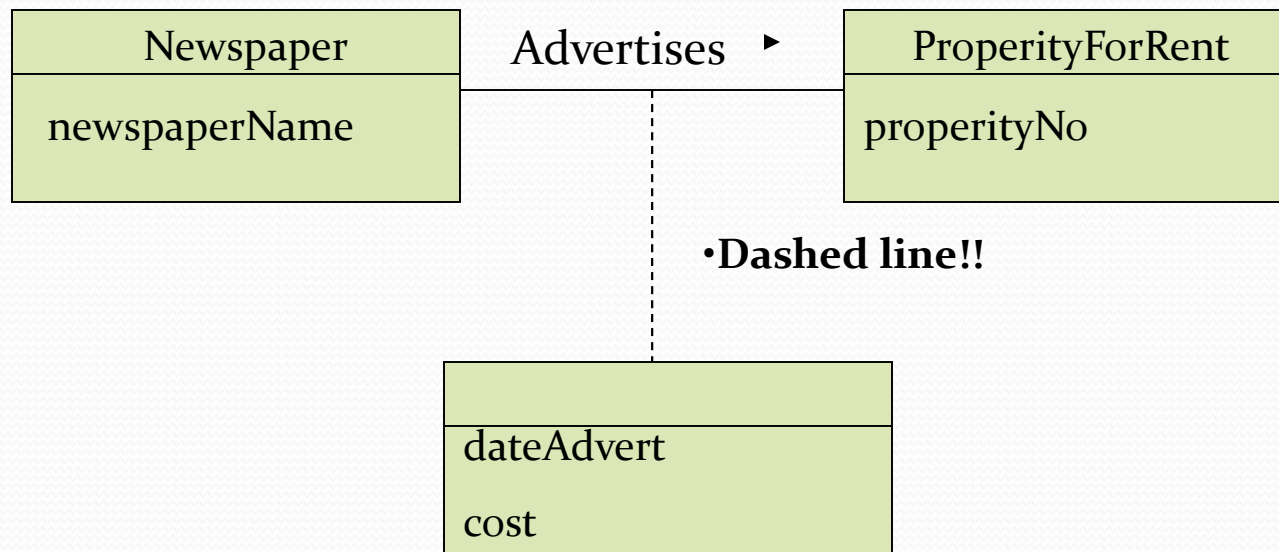
Strong and Weak Entity Types

- Strong Entity Type
 - An entity type that exists irrespective of any other entity type.
- Weak Entity Type
 - An entity type that is existence-dependent on some other type.



Attributes on Relationships

- Attributes can also be assigned to relationships.





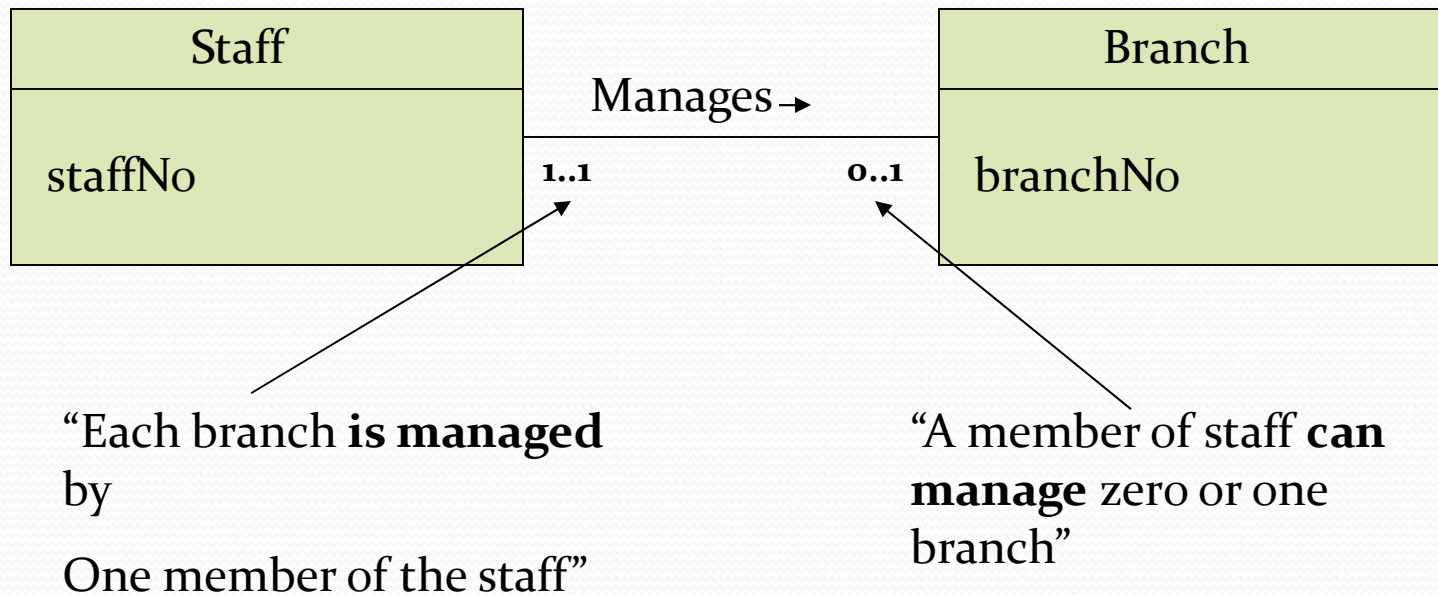
Structural Constraints

- Multiplicity

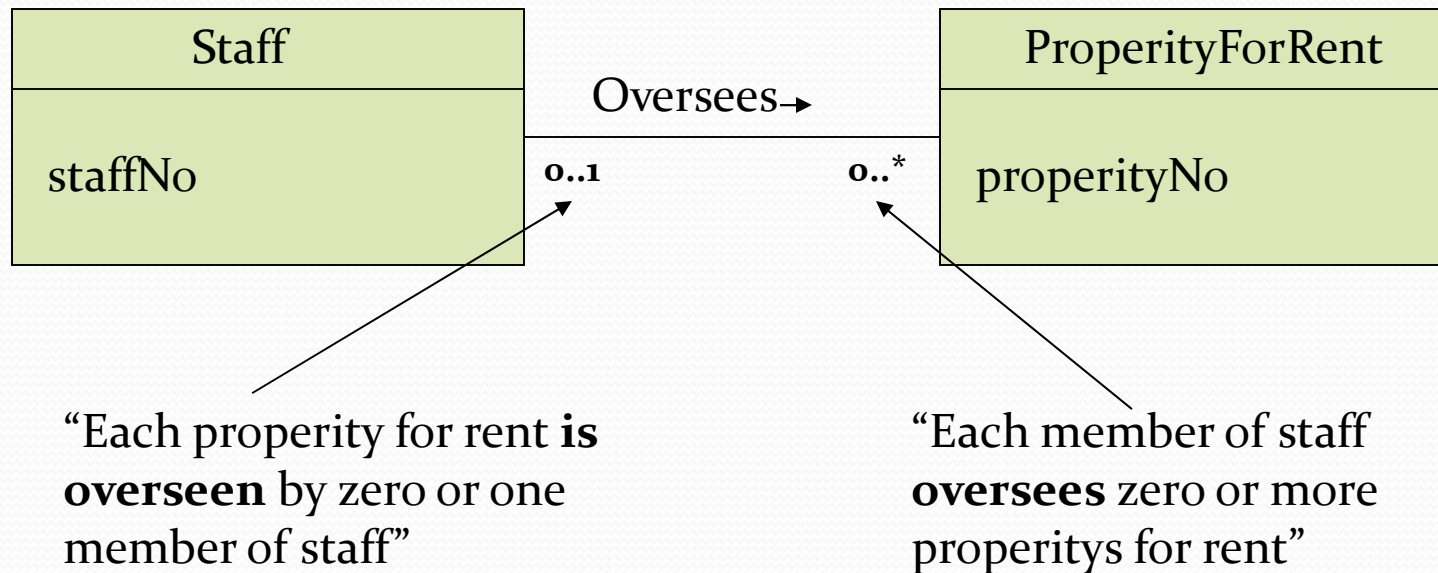
The number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.

- **One-to-one (1:1)**
- **One-to-many (1:*)**
- **Many-to-many (*:*)**

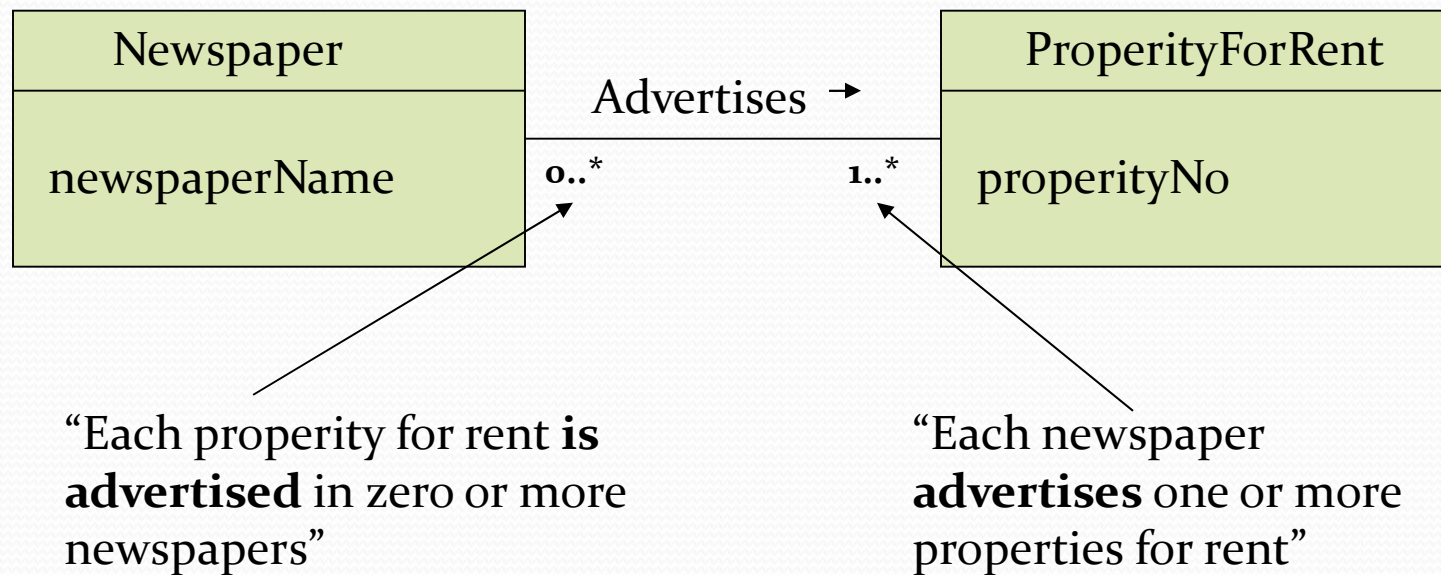
One-to-One (1:1)



One-to-Many (1:*)

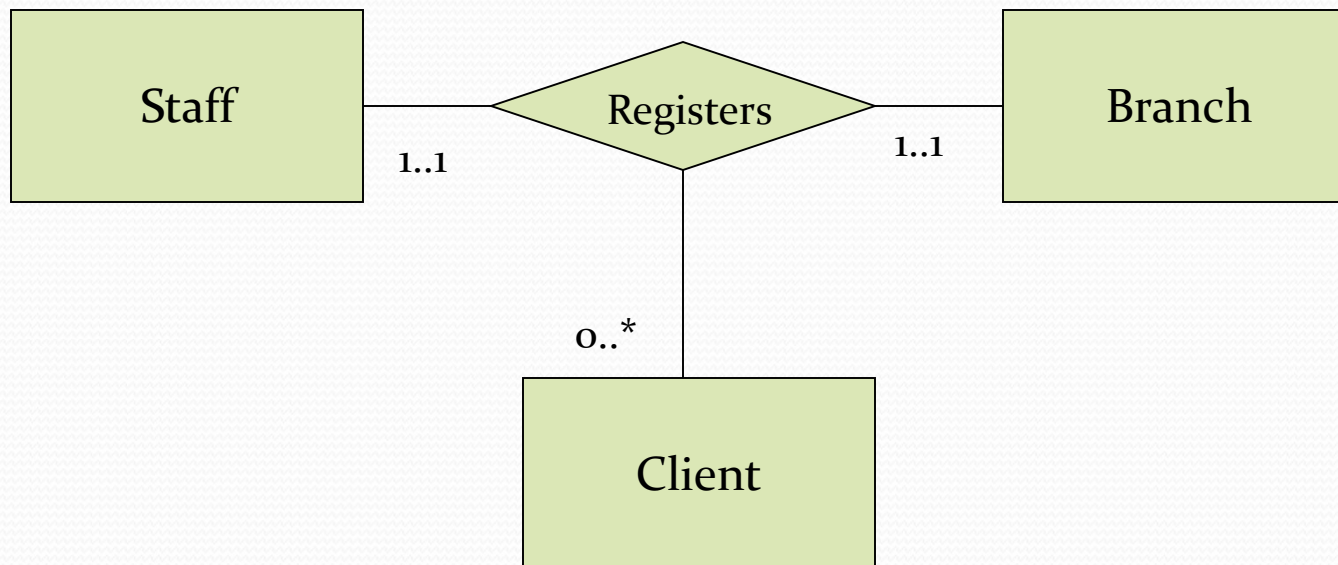


Many-to-Many (*:*)



Multiplicity for Complex Relationships

- **Multiplicity** (complex relationships): The number (or range) of possible occurrences of any entity type in any n-ary relationship when the other (n-1) values are fixed.



Cardinality and Participation Constraints

- **Cardinality**: Describes the maximum number of possible relationship occurrences for an entity participating in a given relationship type. E.g: 1..1, 1..*, *..*, etc,etc.
- **Participation**: Determines whether all or only some entity occurrences participate in a relationship (mandatory or optional).



Constructing an ER model - Entities

- Before beginning to draw the ER model, read the requirements specification carefully.
 - Document any assumptions you need to make.
- 1. Identify entities –
 - list all potential entity types.
 - These are the object of interest in the system.
 - It is better to put too many entities in at this stage and then discard them later if necessary.

Constructing an ER model

- Entities

2. Remove duplicate entities

- Ensure that they really separate entity types or just two names for the same thing.
- Also do not include the system as an entity type
 - e.g. if modeling a library, the entity types might be books, borrowers, etc.
 - The library is the system, thus should not be an entity type.

Constructing an ER model - Attributes

3. List the attributes of each entity (all properties to describe the entity which are relevant to the application).
 - Ensure that the entity types are really needed.
 - are any of them just attributes of another entity type?
 - if so keep them as attributes and cross them off the entity list.
 - Do not have attributes of one entity as attributes of another entity!
4. Mark the primary keys.
 - Which attributes uniquely identify instances of that entity type?
 - This may not be possible for some weak entities.

Constructing an ER model - Relationships

5. Define the relationships
 - Examine each entity type to see its relationship to the others.
 6. Describe the cardinality and optionality of the relationships
 - Examine the constraints between participating entities.
 7. Remove redundant relationships
 - Examine the ER model for redundant relationships.
- ER modeling is iterative, so expect to draw several versions.
 - There is no one right answer to the problem, but some solutions are better than others!

Mapping ER Model – Relational Model

- For each strong (regular) entity type E in the ER schema:
 - Create a relation R that includes all the simple attributes of E.
 - Choose one of the key attributes of E as primary key for R.
 - If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Example – Strong Entity





Weak Entity

- For each weak entity type W in the ERD with owner entity type E :
 - Create a relation R , include all the atomic attributes of W as attributes of R .
 - Include the primary key attributes of the owner entity as foreign key attributes of R .

Example – Weak Entity





1:1 Relationship

- For each binary 1:1 relationship type R in the ER schema
 - Identify the relations S and T that correspond to the entity types participating in R.
 - Choose one of the relations—S, say—and include the primary key of T as foreign key in S.

Example 1:1 Relationship





1:N Relationship

- For each binary 1:N relationship type R:
 - Identify the relation S that represents the participation entity type at the N-side of the relationship type.
 - Include the primary key of the relation T that represents the entity type participating in R as foreign key in S.

Example 1:N Relationship





M:N Relationship

- For each binary M:N relationship type R:
 - Create a new relation S to represent the participation entity types.
 - Include the primary keys of the relations that represent the participating entity types as foreign key attributes in S. This combination forms the primary key of S.
 - Include any atomic attributes of the M:N relationship type as attributes of S.

Example M:N Relationship





Multi-valued Attribute

- For each multi-valued attribute A
 - Create a new relation R.
 - This relation R will include an attribute corresponding to A, plus the primary key attribute of the entity in which attribute A is present - as a foreign key in R.
 - The primary key of R is the combination of A and K.

Example – Multi-valued Attribute





Composite Attribute

- For the composite attribute A
 - Create a new relation R.
 - This relation R will include all attributes corresponding to A, plus the primary key attribute of the entity in which attribute A is present K- as a foreign key in R.
 - The primary key of R is the K.

Example – Composite Attribute





Summary

- Conceptual design follows requirements analysis,
 - Yields a high-level description of data to be stored
- ER model popular for conceptual design
 - Constructs are expressive, close to the way people think about their applications.
- Basic constructs: entities, relationships, and attributes (of entities and relationships).
- Some additional constructs: weak entities, ISA hierarchies, and aggregation.
- Note: There are many variations on ER model.