DATA 300
DATA MANAGEMENT

Aye Hninn Khine, Ph.D.,
Visiting Faculty

PARAMI
UNIVERSITY

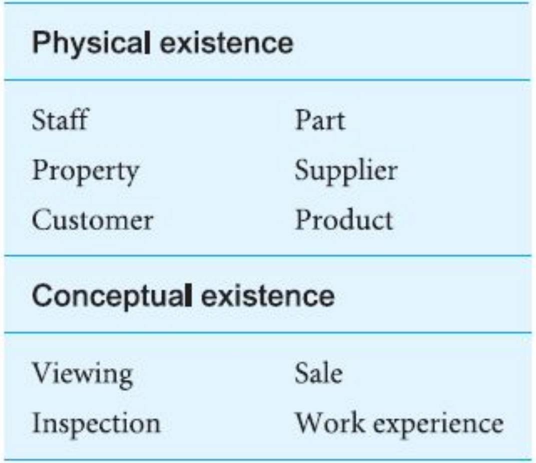
Module 5: Entity-Relationship Modeling

- How to use Entity–Relationship (ER) modeling in database design.
- The basic concepts associated with the ER model: entities, relationships, and attributes.
- A diagrammatic technique for displaying an ER model using the Unified Modeling Language (UML).

Entity

- A group of objects with the same properties, which are identified by the enterprise as having an independent existence.
- A uniquely identifiable object of an entity type.

Figure 12.2 Example of entities with a physical or conceptual existence.



Physical existence	
Staff	Part
Property	Supplier
Customer	Product
Conceptual existence	
Viewing	Sale
Inspection	Work experience

Relationship

- A set of meaningful associations among entity types.
- A **relationship type** is a set of associations between one or more participating entity types.
- Each relationship type is given a name that describes its function.
- An example of a relationship type shown in **Figure 12.1 (textbook)** is the relationship called **POwns**, which associates the **PrivateOwner** and **PropertyForRent** entities.

Semantic Net

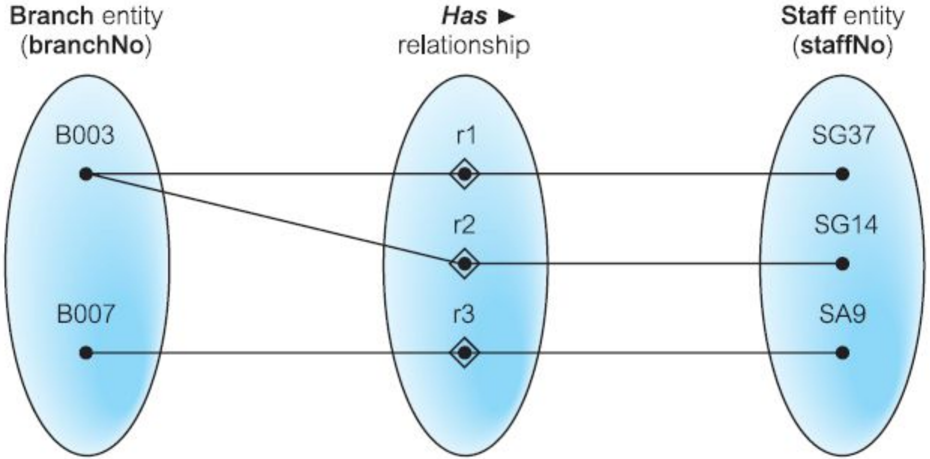
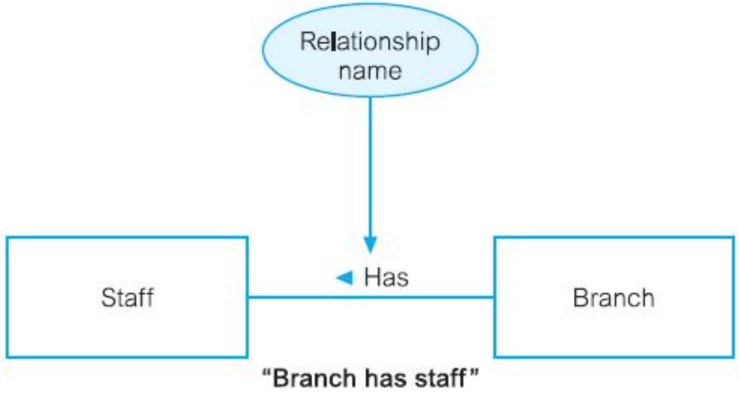


Figure 12.4 A semantic net showing individual occurrences of the *Has* relationship type.

ER Diagram for Staff and Branch Relationship

Figure 12.5
A diagrammatic representation of Branch *Has* Staff relationship type.

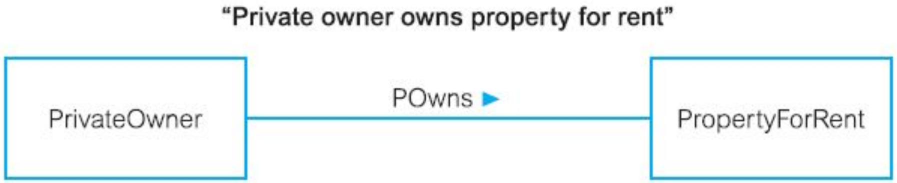


Relationship

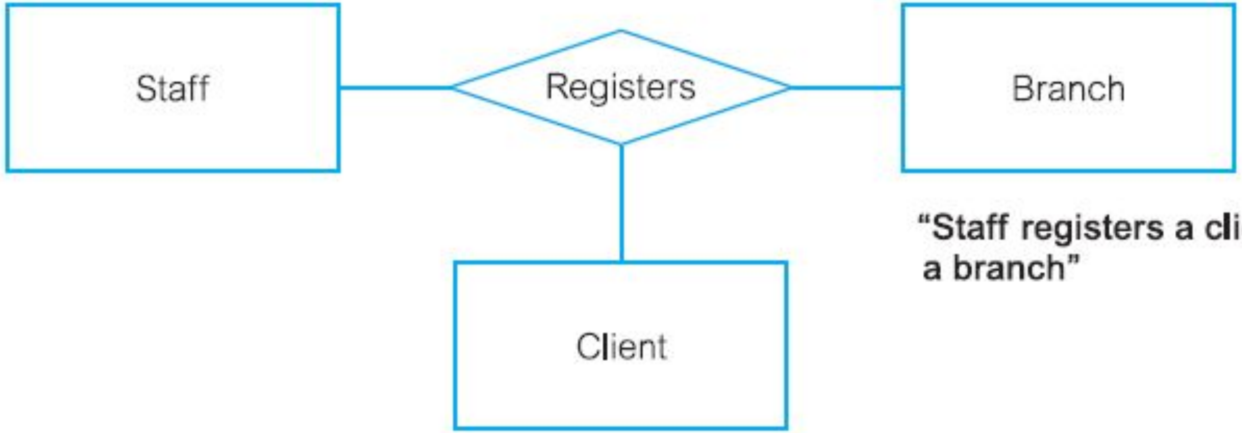
- Each relationship type is shown as a line connecting the associated entity types and labeled with the name of the relationship.
- Normally, a relationship is named using a verb (for example, Supervises or Manages**)** or a short phrase including a verb (for example, LeasedBy).
- Again, the first letter of each word in the relationship name is shown in uppercase.
- Whenever possible, a relationship name should be unique for a given ER model.
- Degree of Relationship
  - The number of participating entity types in a relationship.

Binary Relationship

Figure 12.6 An example of a binary relationship called *POwns.*



Ternary Relationship



“Staff registers a client at a branch”

Figure 12.7 An example of a ternary relationship called *Registers*.

Recursive Relationship

- A relationship type in which the *same* entity type participates more than once in *different roles.*
- Consider a recursive relationship called Supervises, which represents an association of staff with a Supervisor where the Supervisor is also a member of staff.
- In other words, the Staff entity type participates twice in the Supervises relationship; the first participation as a Supervisor, and the second participation as a member of staff who is supervised (Supervisee).
- Relationships may be given **role names** to indicate the purpose that each participating entity type plays in a relationship.

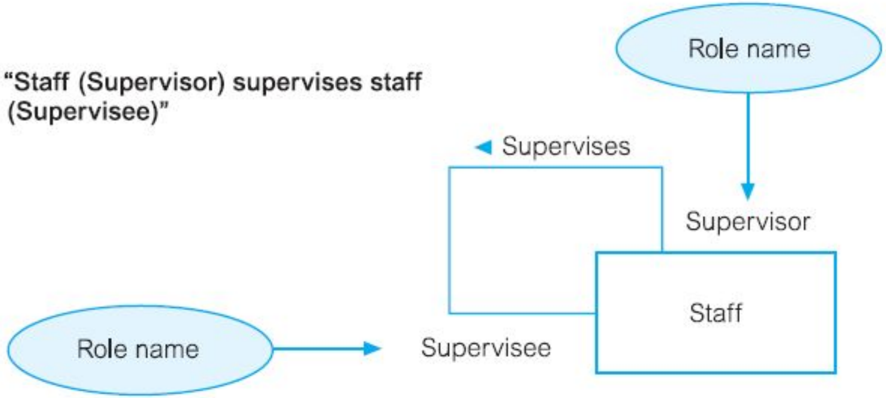
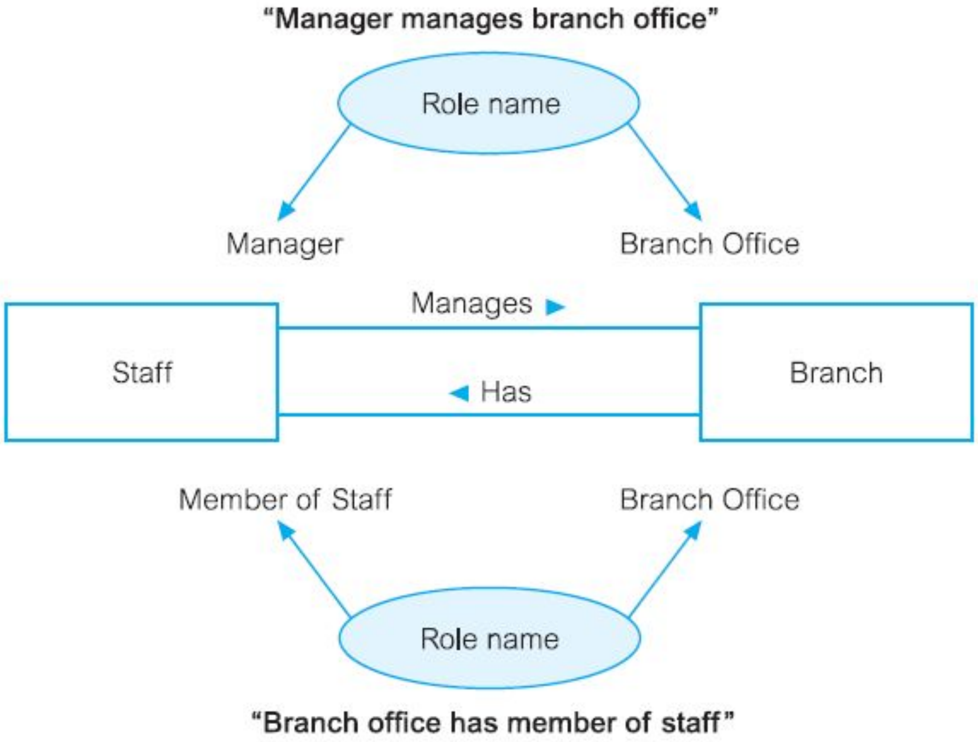


Figure 12.9 An example of a recursive relationship called *Supervises* with role names Supervisor and Supervisee.

Figure 12.10 An example of entities associated through two distinct relationships called *Manages* and *Has* with role names.



Attributes

- A property of an entity or a relationship type. (Staff entity type may be described by the staffNo, name, position, and salary attributes)
- Attribute domain – allowable values for an attribute (age >= 18)
- Simple attribute - An attribute composed of a single component with an independent existence.
- Composite attribute - An attribute composed of multiple components, each with an independent existence. (Address – room no; street; city;)
- Single-Value attribute - An attribute that holds a single value for each occurrence of an entity type. each occurrence of the Branch entity type has a single value for the branch number (**branchNo**) attribute (for example, B003)
- Multi-value attribute - An attribute that holds **multiple values** for each occurrence of an entity type. for example, branch number B003 has telephone numbers 0141-339-2178 and 0141-339-4439

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.
- The values held by some attributes may be derived. For example, the value for the **duration** attribute of the Lease entity is calculated from the **rentStart** and **rentFinish** attributes, also of the Lease entity type.

Keys

- Candidate Key - The minimal set of attributes that uniquely identifies each occurrence of an entity type.
    - Example: Imagine a student database where each student has:
    - **Student ID** (e.g., "S123")
    - **Email** (e.g., "john@example.com")
    - Both **Student ID** and **Email** are unique for each student, meaning either one can be used to find a specific student. So, both are **candidate keys**.
- Primary Key - The candidate key that is selected to uniquely identify each occurrence of an entity type.
    - Example: From the candidate keys (**Student ID** and **Email**), we choose **Student ID** as the **primary key**, because it's short and stable (emails can change).
    - ✅ **Primary Key: Student ID**
      🚫 **Email remains a candidate key but is not the primary key.**

ER Diagram for STAFF and BRANCH

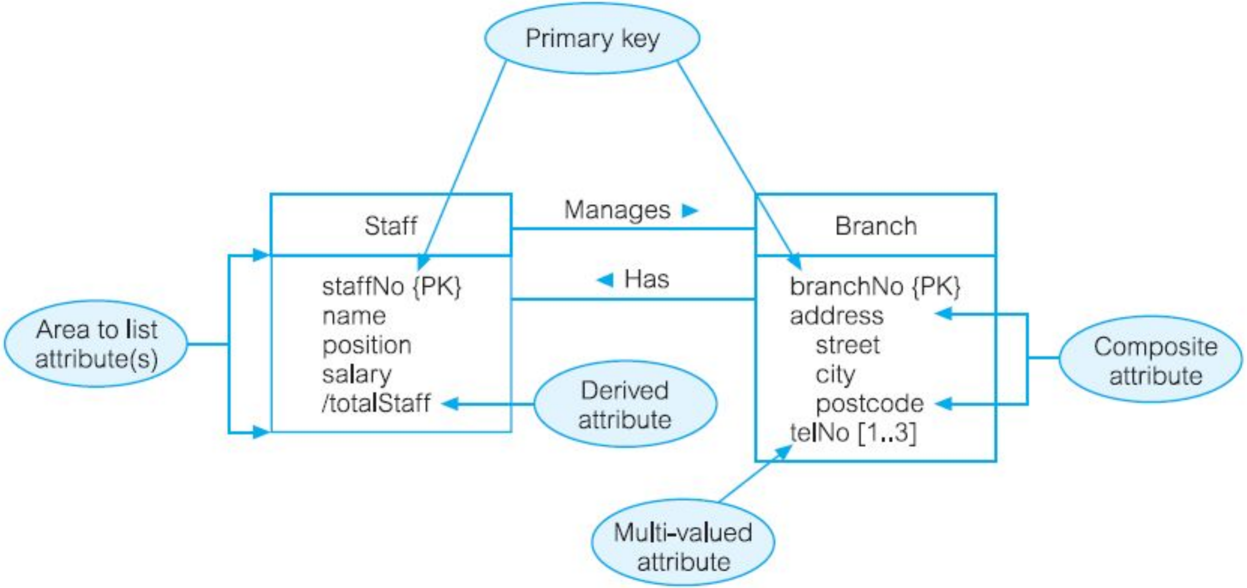


Figure 12.11 Diagrammatic representation of Staff and Branch entities and their attributes.

Multiplicity

Structural Constraints

- 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.
- the most common degree for relationships is binary.
- Binary relationships are generally referred to as being one-to-one (1:1), one-to many (1:*), or many-to-many (*:*).
- a member of staff manages a branch (1:1);
- a member of staff oversees properties for rent (1:*);

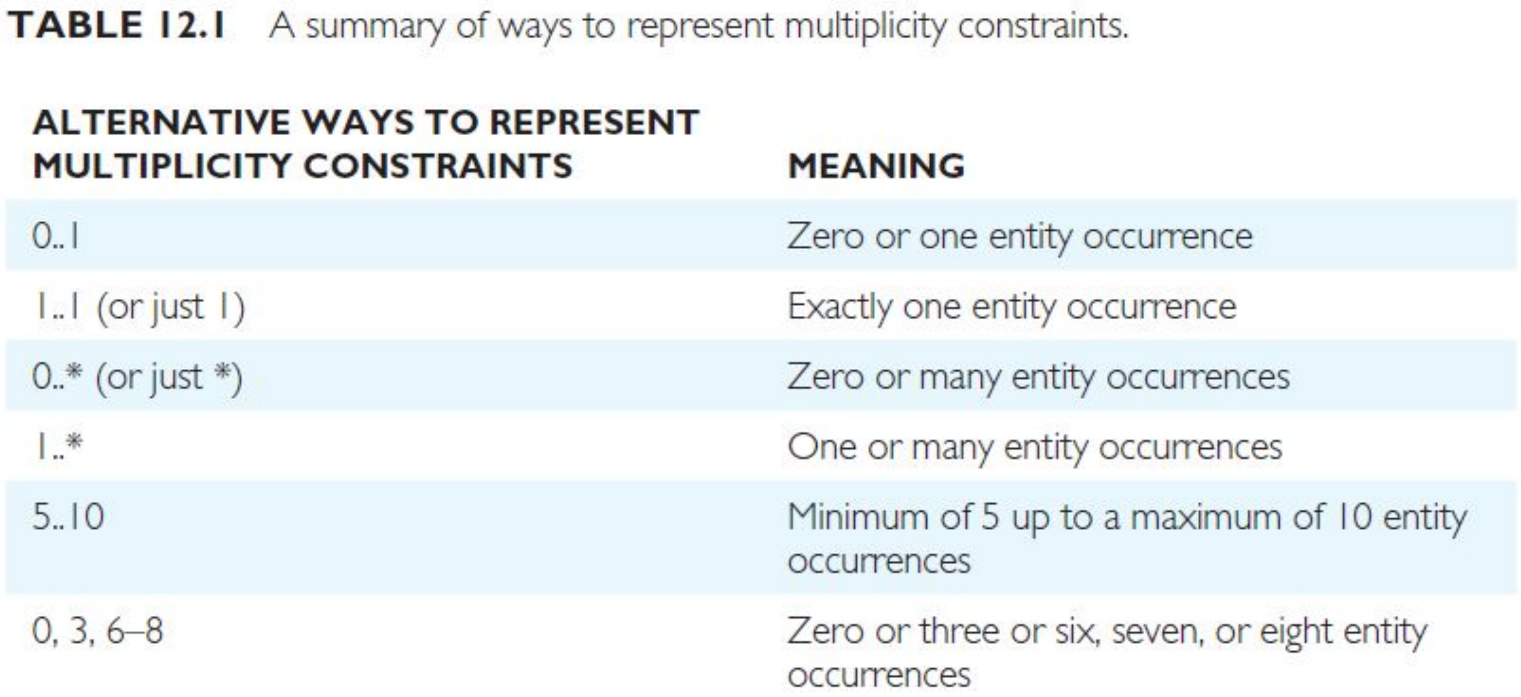


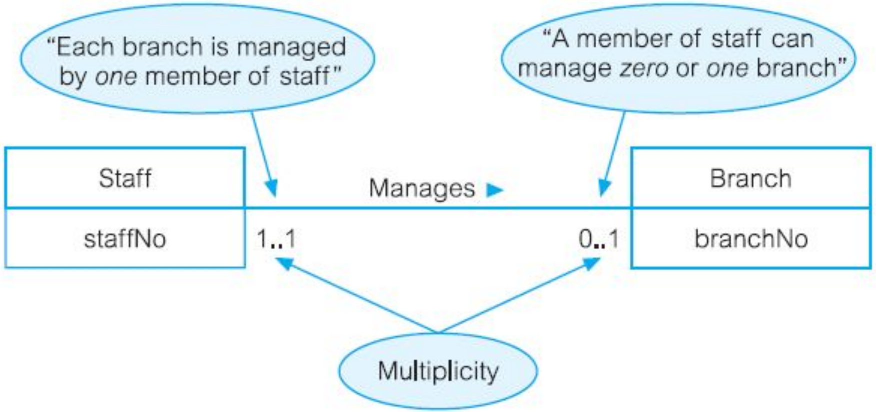
TABLE 12.1 A summary of ways to represent multiplicity constraints.

ALTERNATIVE WAYS TO REPRESENT MULTIPLICITY CONSTRAINTS	MEANING
0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6–8	Zero or three or six, seven, or eight entity occurrences

1:1 One to One Relationship

- Consider the relationship Manages, which relates the Staff and Branch entity types.
- staffNo **SG5** manages branchNo **B003** and staffNo **SL21** manages branchNo **BOO5**, but staffNo **SG37** does not manage any branch.
- In other words, a member of staff can manage *zero or one* branch and each branch is managed by *one* member of staff.

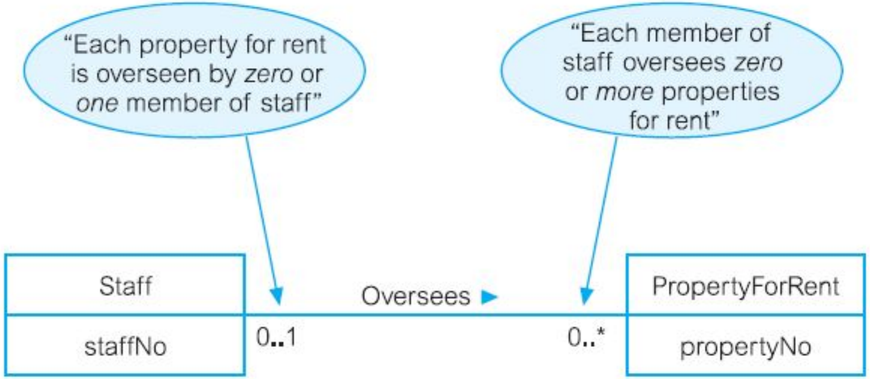
Figure 12.14(b) The multiplicity of the Staff *Manages* Branch one-to-one (1:1) relationship.


"Each branch is managed by *one* member of staff"
"A member of staff can manage *zero* or *one* branch"
Staff
staffNo
Manages ►
Branch
branchNo
1..1
0..1
Multiplicity

One to Many Relationships

- Relationship – “Oversees”
- staffNo **SG37** oversees propertyNos **PG21** and **PG36**, and staffNo **SA9** oversees propertyNo **PA14** but staffNo **SG5** does not oversee any properties for rent and propertyNo **PG4** is not overseen by any member of staff.

Figure 12.15(b) The multiplicity of the Staff *Oversees* PropertyForRent one-to-many (1:*) relationship type.


“Each property for rent is overseen by *zero* or *one* member of staff”
“Each member of staff oversees *zero* or *more* properties for rent”
Staff
staffNo
Oversees ▶
0..1
0..*
PropertyForRent
propertyNo

Exercise – Work in a group of two

Create an ER model for each of the following descriptions:

- (a) Each company operates four departments, and each department belongs to one company.
- (b) Each department in part (a) employs one or more employees, and each employee works for one department.
- (c) Each of the employees in part (b) may or may not have one or more dependents, and each dependent
- belongs to one employee.
- (d) Each employee in part (c) may or may not have an employment history.
- (e) Represent all the ER models described in (a), (b), (c), and (d) as a single ER model.

Assignment

- Create an ER model for each of the following descriptions:
- (a) A large organization has several parking lots, which are used by staff.
- (b) Each parking lot has a unique name, location, capacity, and number of floors (where appropriate).
- (c) Each parking lot has parking spaces, which are uniquely identified using a space number.
- (d) Members of staff can request the sole use of a single parking space. Each member of staff has a unique number, name, telephone extension number, and vehicle license number.
- (e) Represent all the ER models described in parts (a), (b), (c), and (d) as a single ER model. Provide any assumptions necessary to support your model.

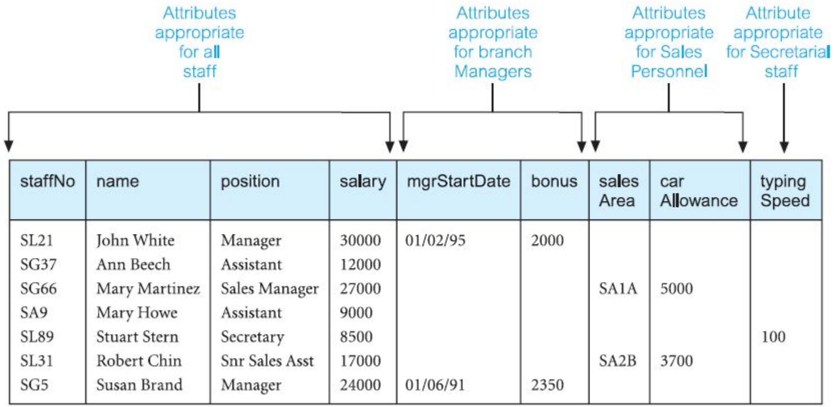
Specialization/Generalization

Inheritance

- The concept of specialization/generalization is associated with special types of entities known as **superclasses** and **subclasses**, and the process of **attribute inheritance**.
- a member of the SalesPersonnel subclass *inherits* all the attributes of the Staff superclass, such as staffNo, name, position, and salary together with those specifically associated with the SalesPersonnel subclass, such as salesArea and carAllowance

Superclass/Subclass

- Superclass
  - An entity type that includes one or more distinct subgroupings of its occurrences, which must be represented in a data model.
- Subclass
  - A distinct subgrouping of occurrences of an entity type, which must be represented in a data model.


Attributes appropriate for all staff
Attributes appropriate for branch Managers
Attributes appropriate for Sales Personnel
Attribute appropriate for Secretarial staff

staffNo	name	position	salary	mgrStartDate	bonus	sales Area	car Allowance	typing Speed
SL21	John White	Manager	30000	01/02/95	2000			
SG37	Ann Beech	Assistant	12000					
SG66	Mary Martinez	Sales Manager	27000			SA1A	5000	
SA9	Mary Howe	Assistant	9000					
SL89	Stuart Stern	Secretary	8500					100
SL31	Robert Chin	Snr Sales Asst	17000			SA2B	3700	
SG5	Susan Brand	Manager	24000	01/06/91	2350			

Figure 13.1 The AllStaff relation holding details of all staff.

Generalization

- The process of minimizing the differences between entities by identifying their common characteristics.
- The process of generalization is a bottom-up approach, that results in the identification of a generalized superclass from the original entity types.
- For example, consider a model where Manager, SalesPersonnel, and Secretary are represented as distinct entity types.
- If we apply the process of generalization on these entities, we attempt to identify similarities between them, such as common attributes and relationships.
- As stated earlier, these entities share attributes common to all staff, and therefore we identify Manager, SalesPersonnel, and Secretary as subclasses of a generalized Staff superclass.

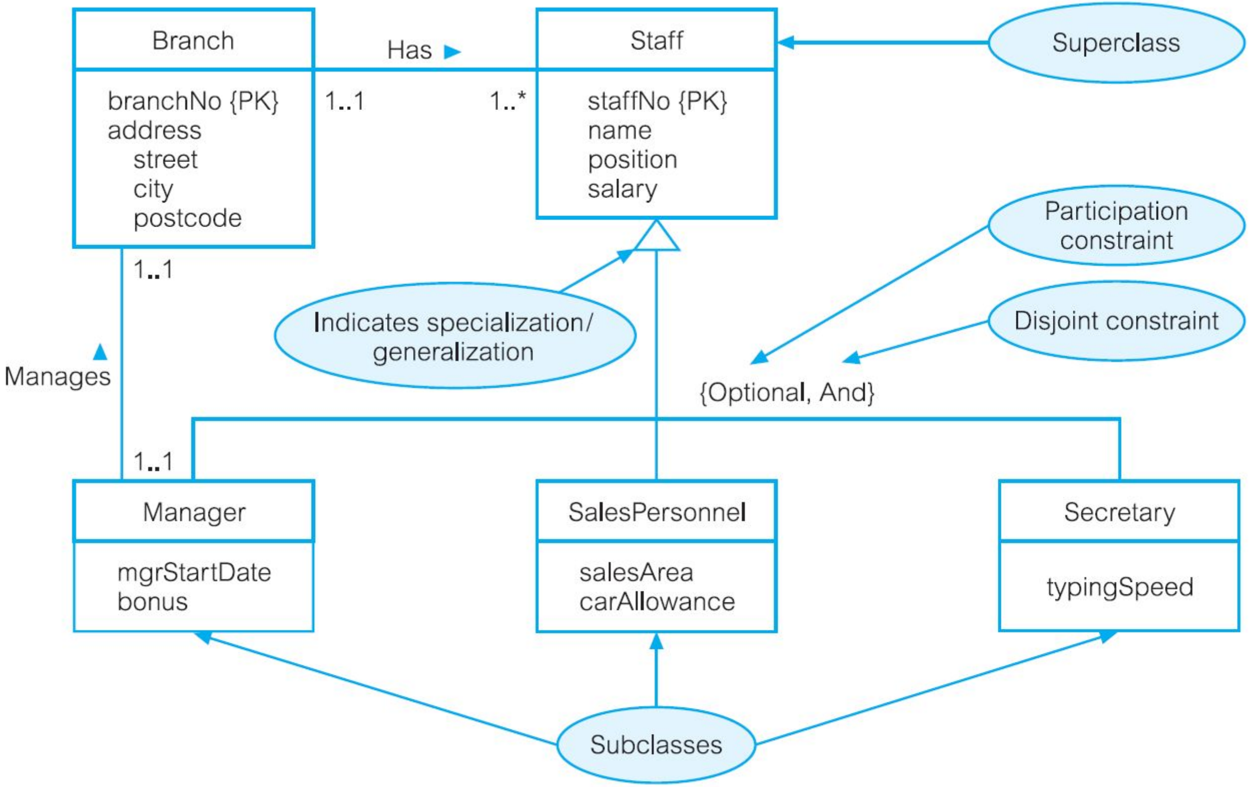


Figure 13.2 Specialization/generalization of the Staff entity into subclasses representing job roles.

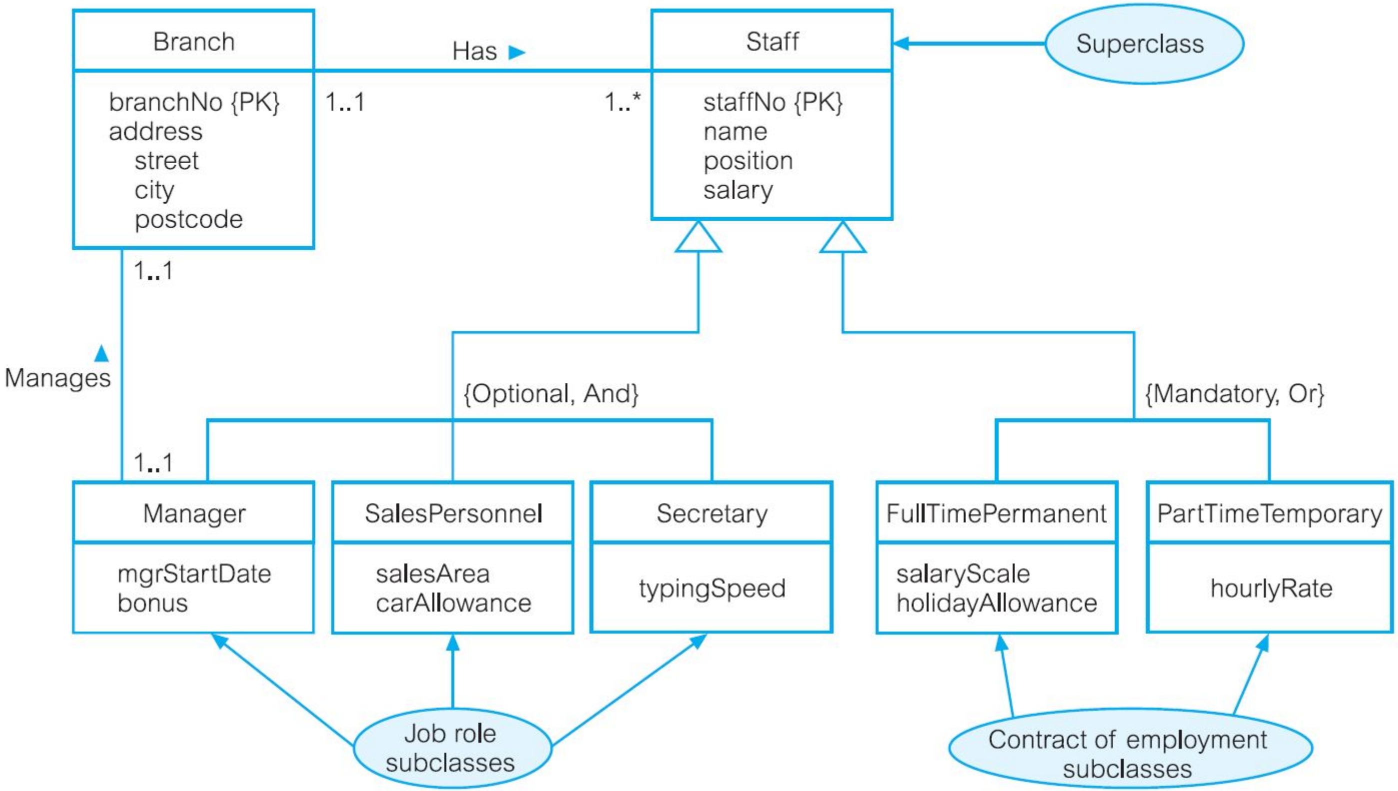
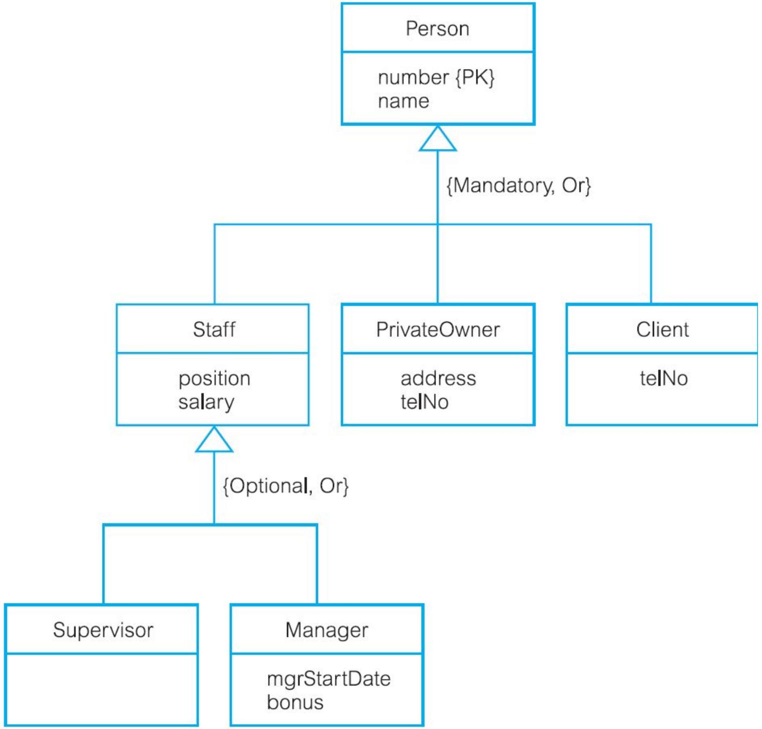


Figure 13.3 Specialization/generalization of the Staff entity into subclasses representing job roles and contracts of employment.

Participation Constraint

- Determines whether every member in the superclass must participate as a member of a subclass.
- Mandatory/Optional



Reading Materials

- Database Systems: Textbook, Chapter 12, and 13

Thank You

ayehninnkhine@parami.edu.mm