SECD2523 DATABASE

TOPIC 4 | ENTITY RELATIONSHIP MODELLING

Content adapted from Connolly, T., Begg, C., 2015. Database Systems: A Practical Approach to Design, Implementation, and Management, Global Edition. Pearson Education.

LECTURE LEARNING OUTCOME

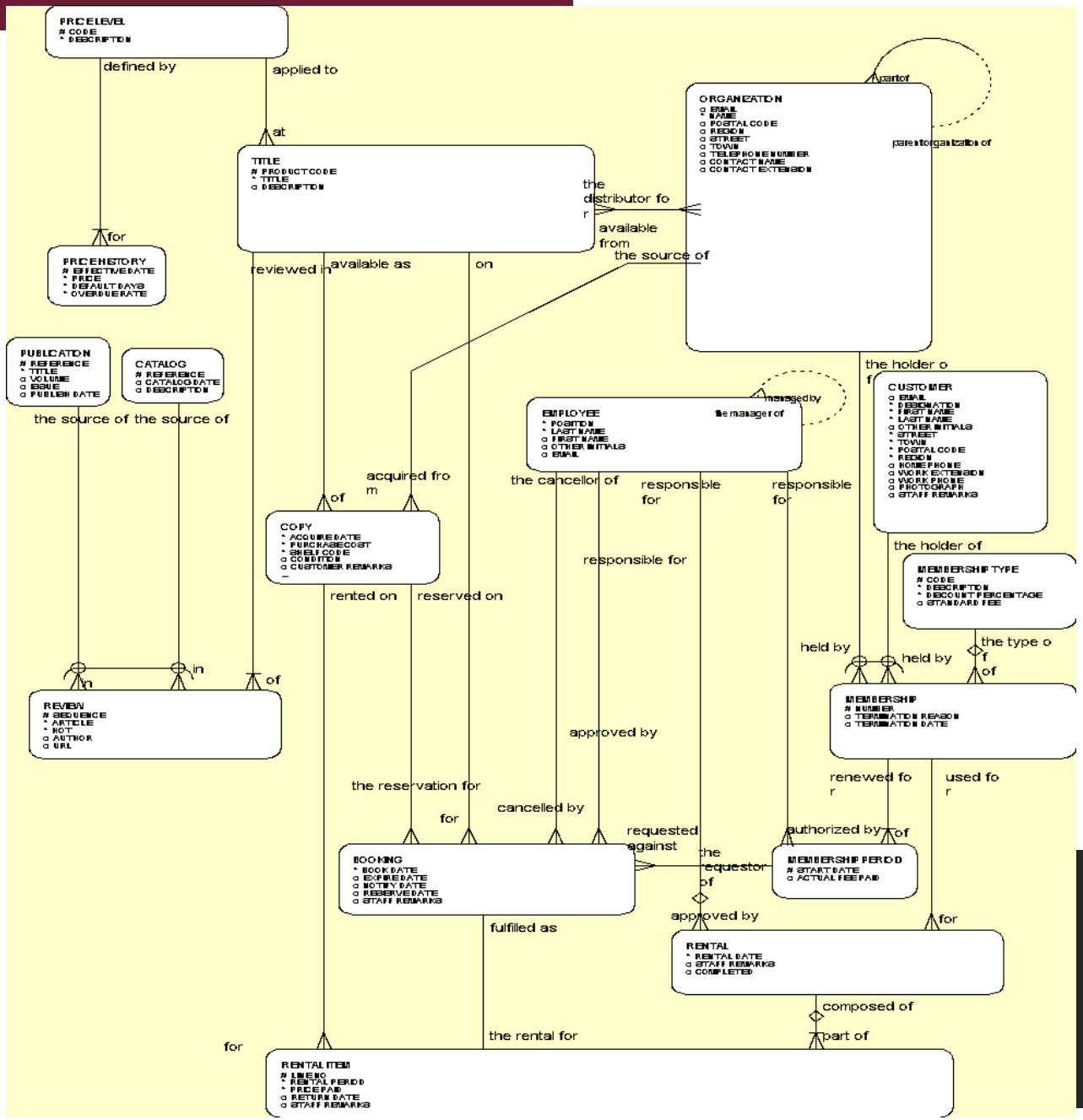
By the end of this lecture, students should be able to:

01 Define basic concepts associated with ER diagram

02 Produce ER model to represent information to application system

Entity Relationship Modeling

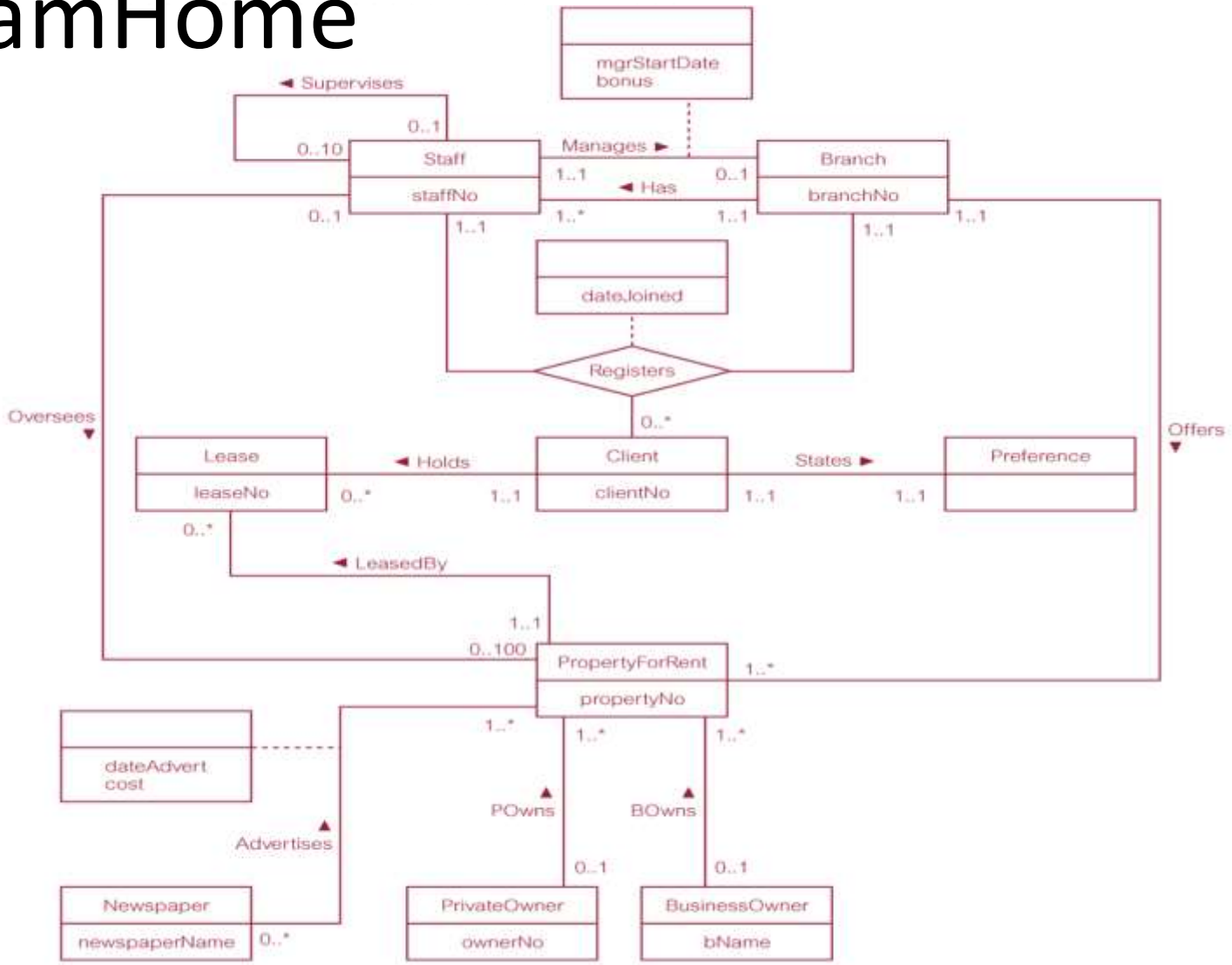
- Models business, not implementation
- Is a well-established technique
- Has a robust syntax
- Results in easy-to-read diagrams…
- …although they may look rather complex at first sight



Goals of Entity Relationship Modeling

- Capture ***all*** required information
- Information appears ***only once***
- Model ***no*** information that is derivable from other information already modelled
- Information is in a predictable, logical place

ER Diagram of Branch View of DreamHome



ER Modelling Notations

UML Notation

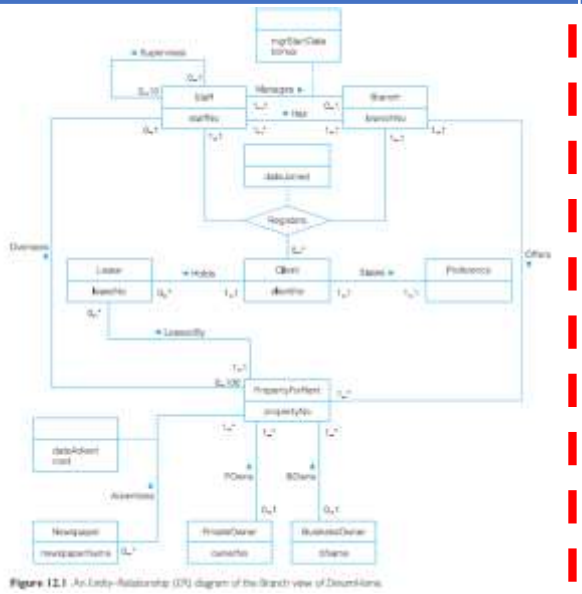


Figure 12.1 An Entity-Relationship (ER) diagram of the Branch view of DreamHome.

We are using this in this course

Chen's Notation

Figure C.1 Part of the ER diagram shown in Figure 12.1 redrawn using the Chen notation.

Crow's Feet Notation

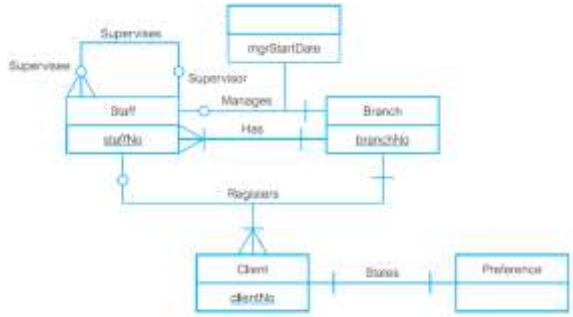


Figure C.2 Part of the ER diagram shown in Figure 12.1 redrawn using the Crow's Feet notation.

Concepts of the ER Model

- Entity types
- Relationship types
- Attributes

Entity Types

- **Entity types**
  - Group of objects with same properties, identified by enterprise as having an independent existence.
  - "Something" of significance to the business about which data must be known
  - A name for the things that you can list
  - Usually a noun
- **Entity occurrence/instances**
  - Uniquely identifiable object of an entity type.

Example of Entities and Instances

ENTITY	INSTANCES
PERSON	Mahatma Gandhi
PRODUCT	2.5 x 35 mm copper nail
PRODUCT TYPE	nail
EMPLOYMENT CONTRACT	my previous contract
JOB	violinist
SKILL LEVEL	fluent
TICKET RESERVATION	tonight: Hamlet in the Royal
PURCHASE	the CD I bought yesterday
ELECTION	for parliament next fall
PRINTER PREFERENCE	…
DOCUMENT VERSION	

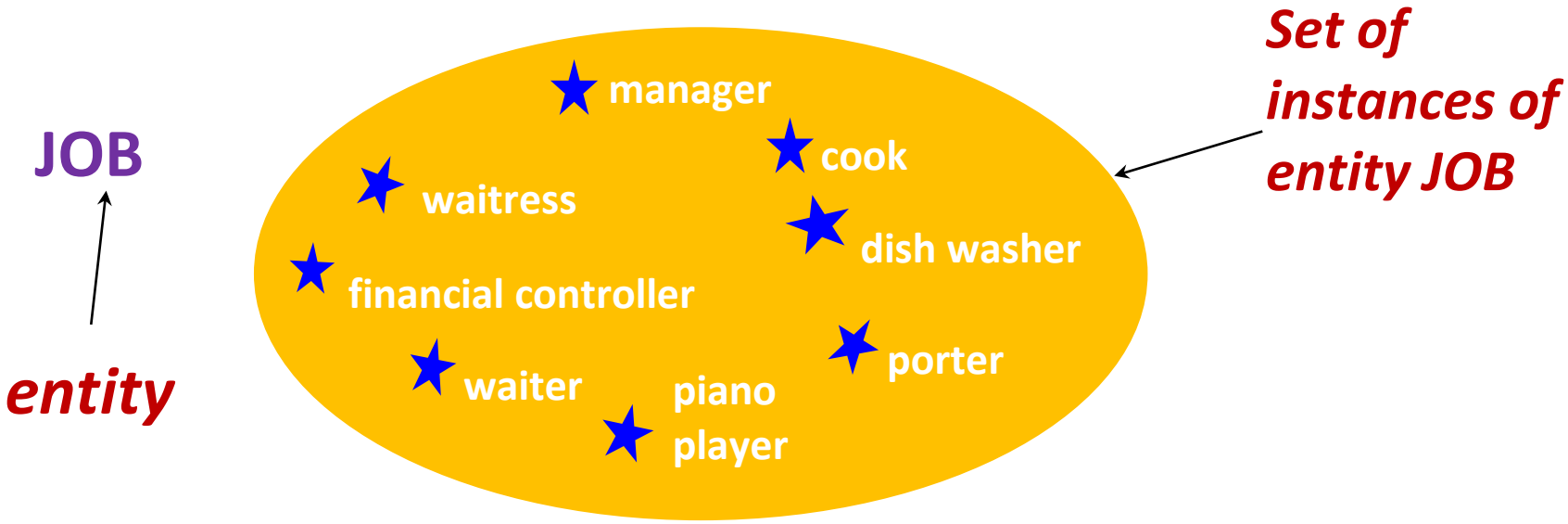
Examples of Entity Types

- Something that exists physically or conceptually

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

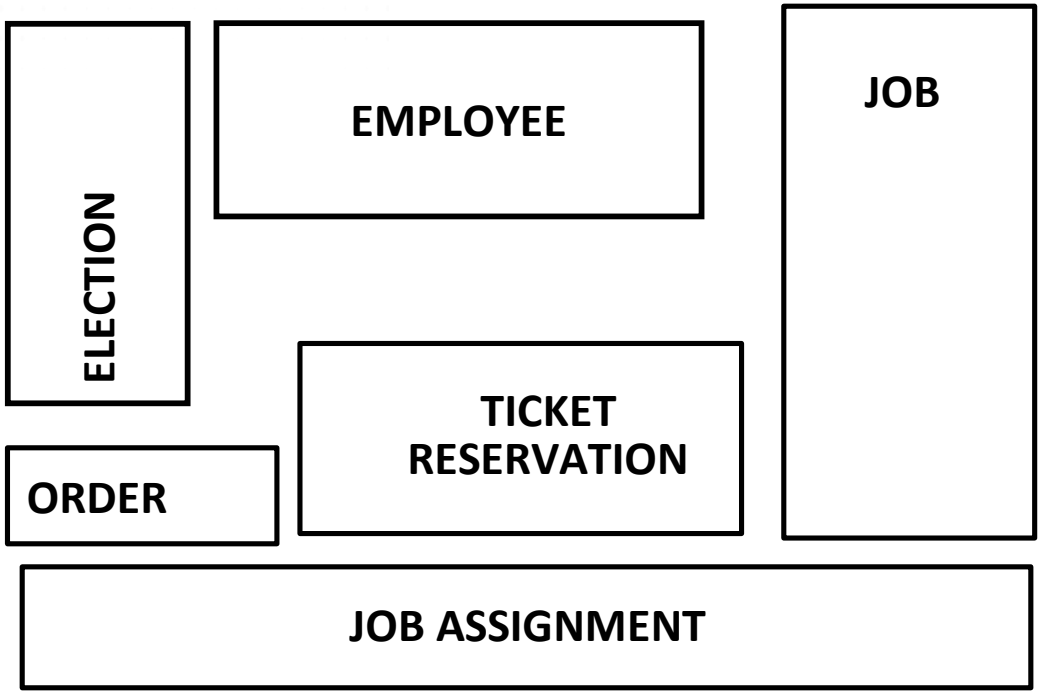
Entities and Sets

- An entity represents a set of instances that are of interest to a particular business.



Entity Representation in Diagram

- Drawn as a box
- Name singular
- Name inside
- Neither size, nor position has a special meaning



During design, entities usually lead to tables.

Remarks on creating Entities

- Give the entity a unique name
- Create a formal description of the entity
- Add a few attributes, if possible
- Be aware of homonyms (words with same pronunciation but different meaning)
- Check entity names and descriptions regularly
- Avoid use of reserved words
- Remove relationship name from entity name

Relationship

- **Relationship type**
  - Set of meaningful associations among entity types.
  - Express how entities are mutually related
  - Always exist between two entities (or one entity twice)
- **Relationship occurrence**
  - Uniquely identifiable association, which includes one occurrence from each participating entity type.

Relationship Examples

- **BRANCH** *has* **STAFF**
  STAFF *belongs to* **BRANCH**
- **EMPLOYEES** *have* **JOBS**
  JOBS *are held by* **EMPLOYEES**
- **PEOPLE** *make* **TICKET RESERVATIONS**
  TICKET RESERVATIONS *are made by* **PEOPLE**

Semantic Net of *Has* Relationship Type

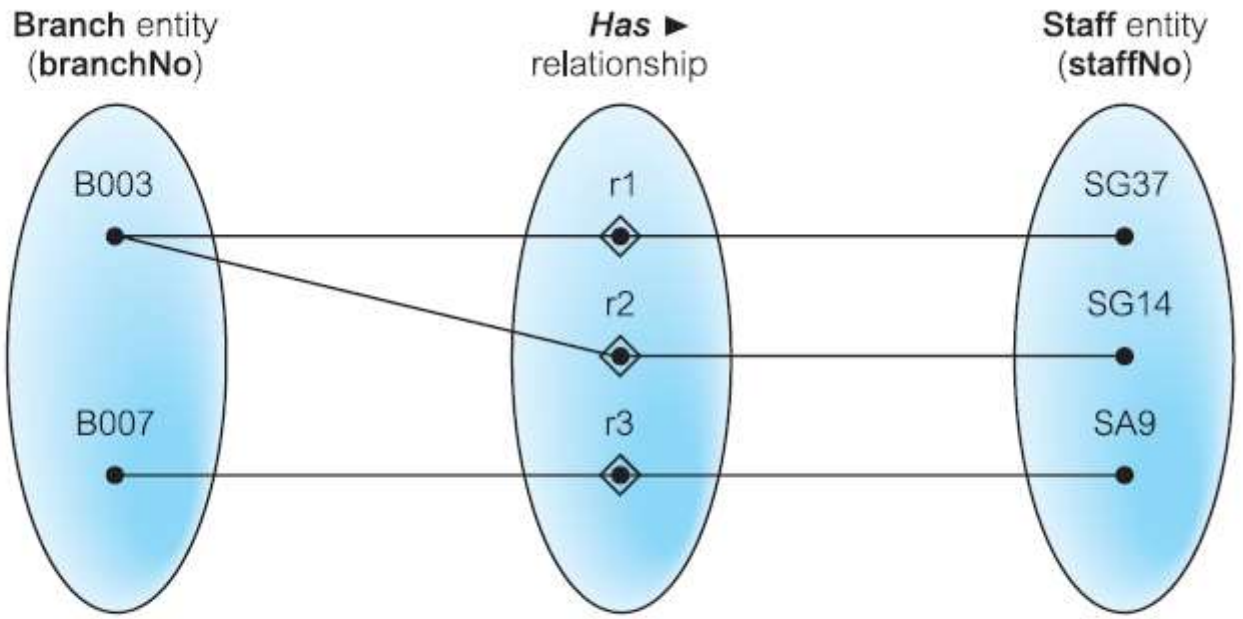
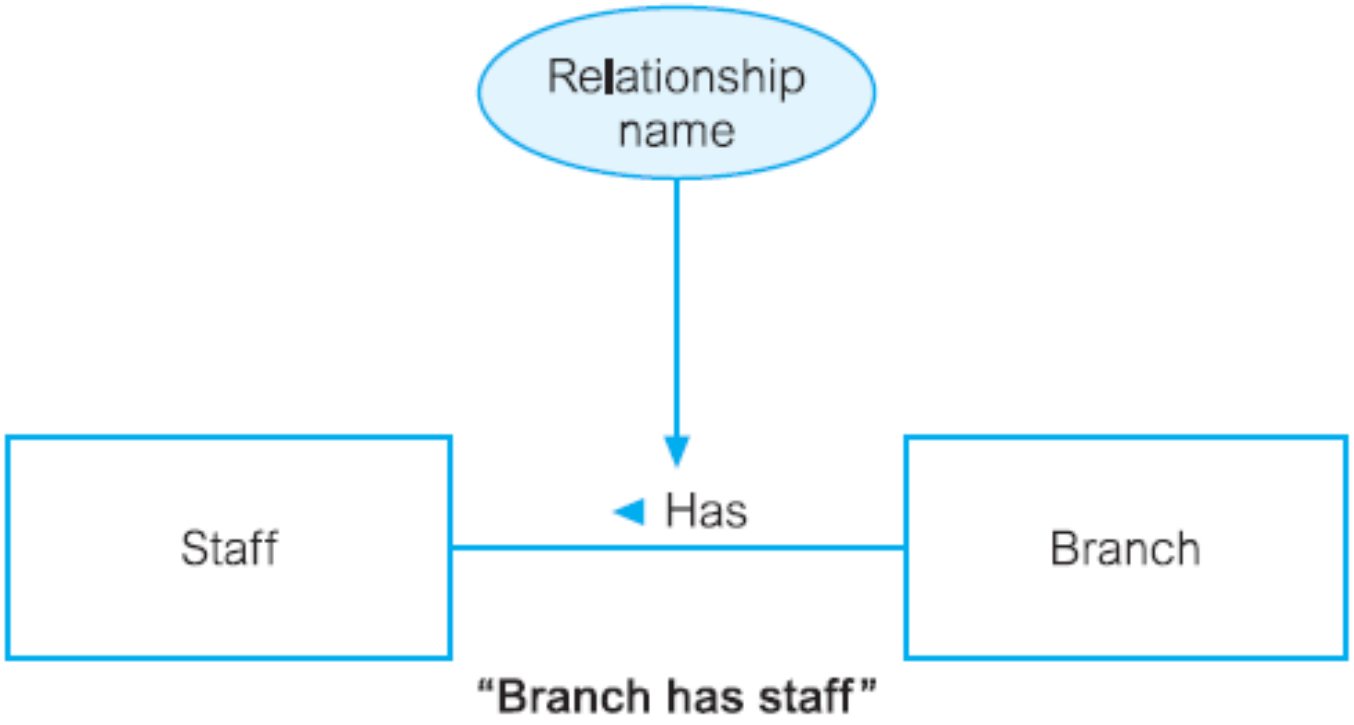
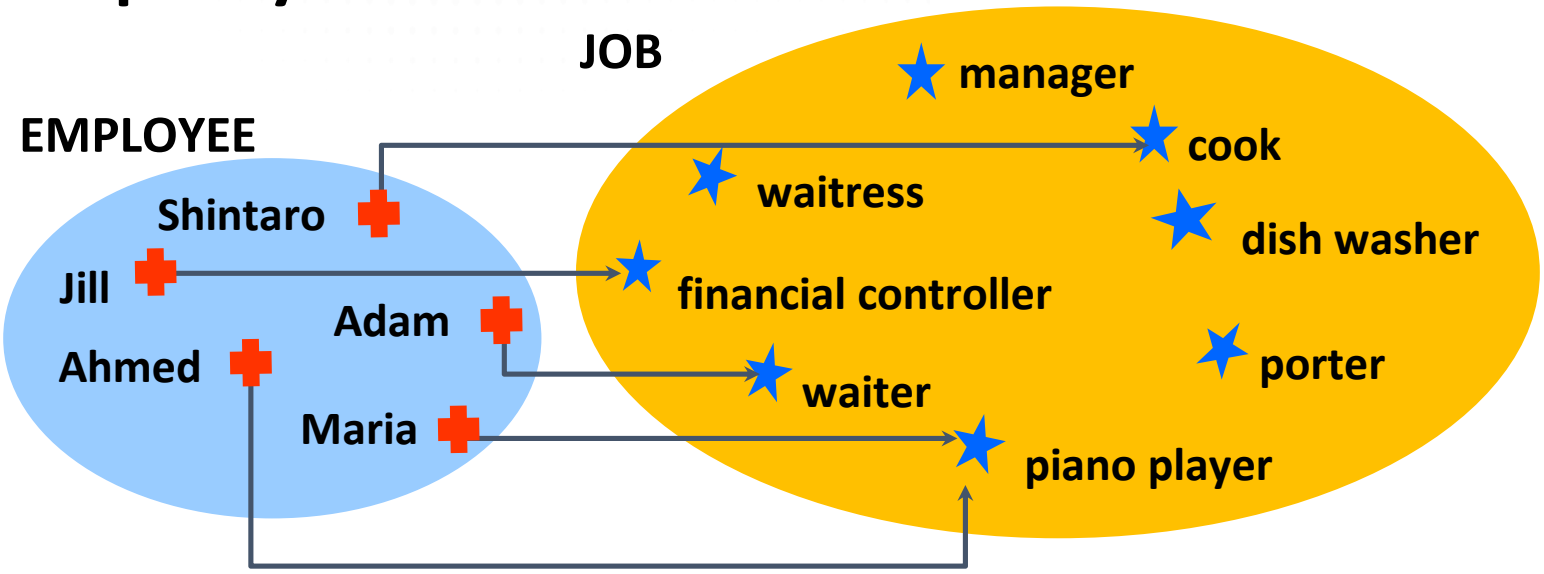


Figure 12.4 A semantic net showing individual occurrences of the *Has* relationship type.

Relationship *Has* Between Staff and Branch



Employees *have* Jobs

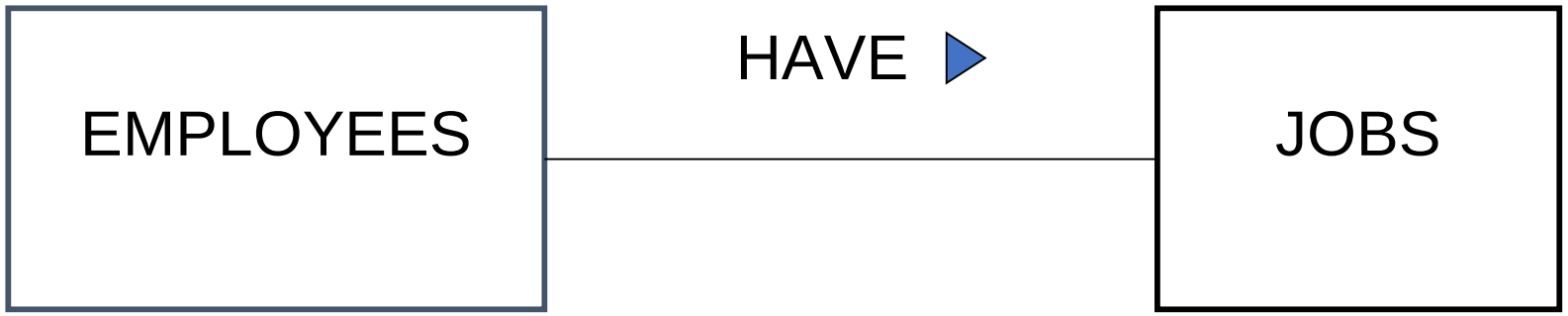


Numerical observation:

- *All* EMPLOYEES have a JOB
- No EMPLOYEE has *more than one* JOB
- *Not all* JOBS are held by an EMPLOYEE
- Some JOBS are held by *more than one* EMPLOYEE

Whether all these are true, it all depends on business rules

Employees *Have* Jobs



Relationship Types

- **Degree of a Relationship**
  - Number of participating entities in relationship.

- Degree of relationship:
  - two entities **binary**;
  - three entities **ternary**;
  - four entities **quaternary**.

Example a binary relationship called: *POwns*

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

Example of a ternary relationship called: *Registers*

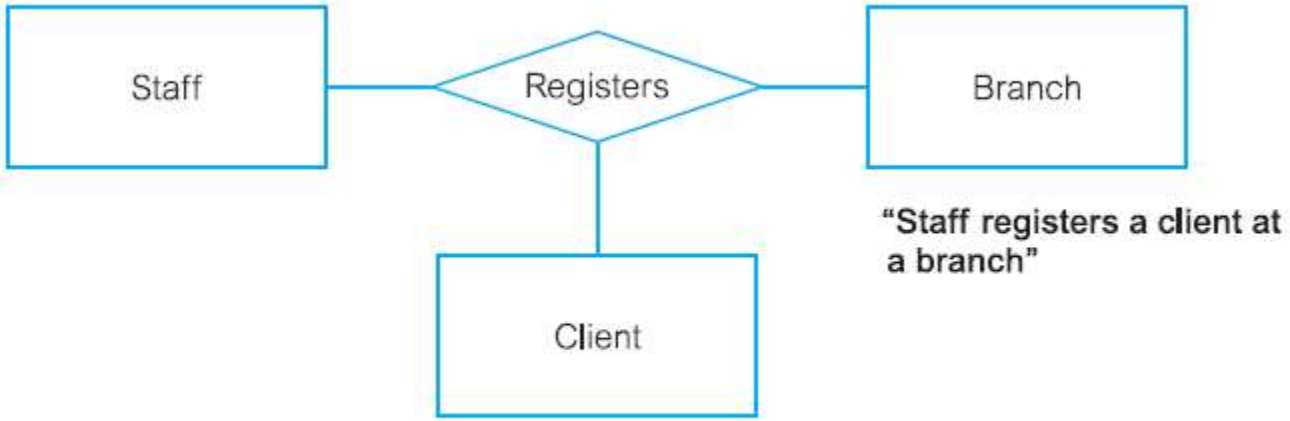


Figure 12.7 An example of a ternary relationship called *Registers*.

Example of a quaternary relationship called: *Arranges*

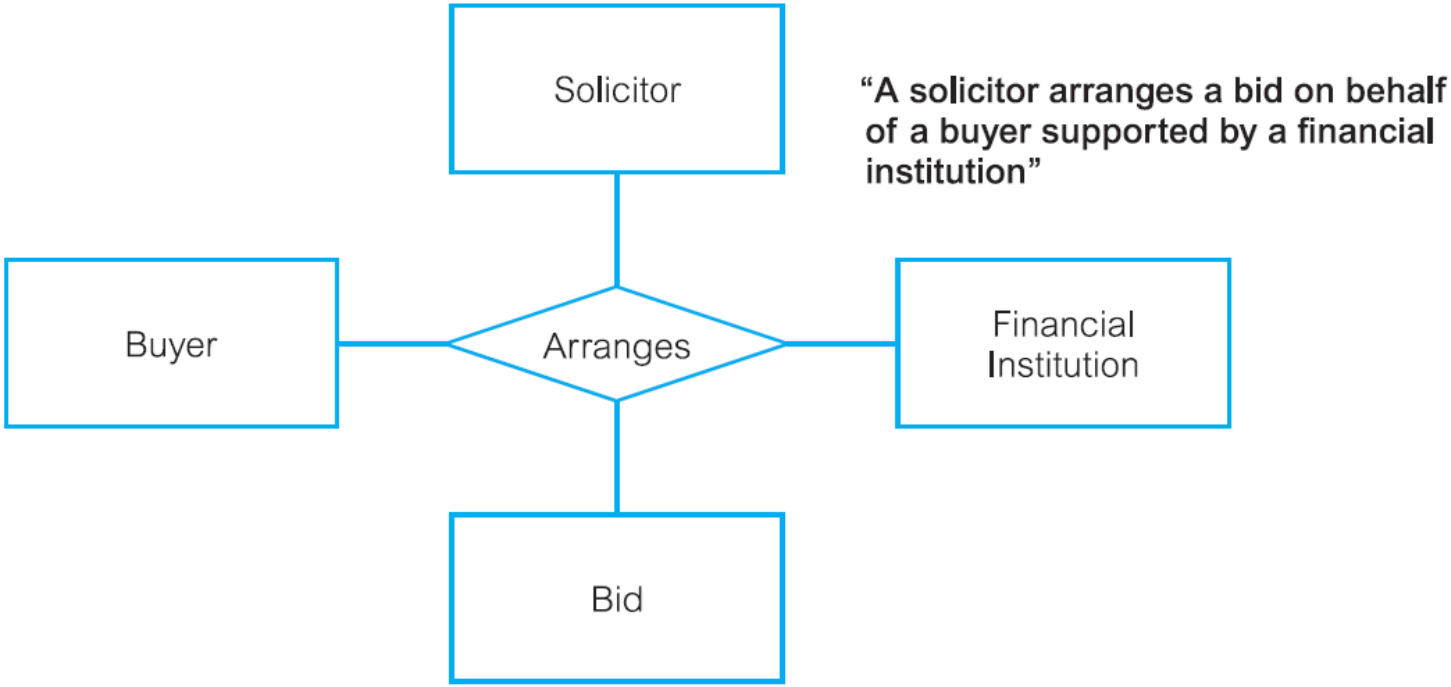


Figure 12.8 An example of a quaternary relationship called *Arranges*.

Relationship Types

- **Recursive Relationship**
  - Relationship type where same entity type participates more than once in different roles.
- Relationships may be given role names to indicate purpose that each participating entity type plays in a relationship.

Example of recursive relationship called: *Supervises* with role names

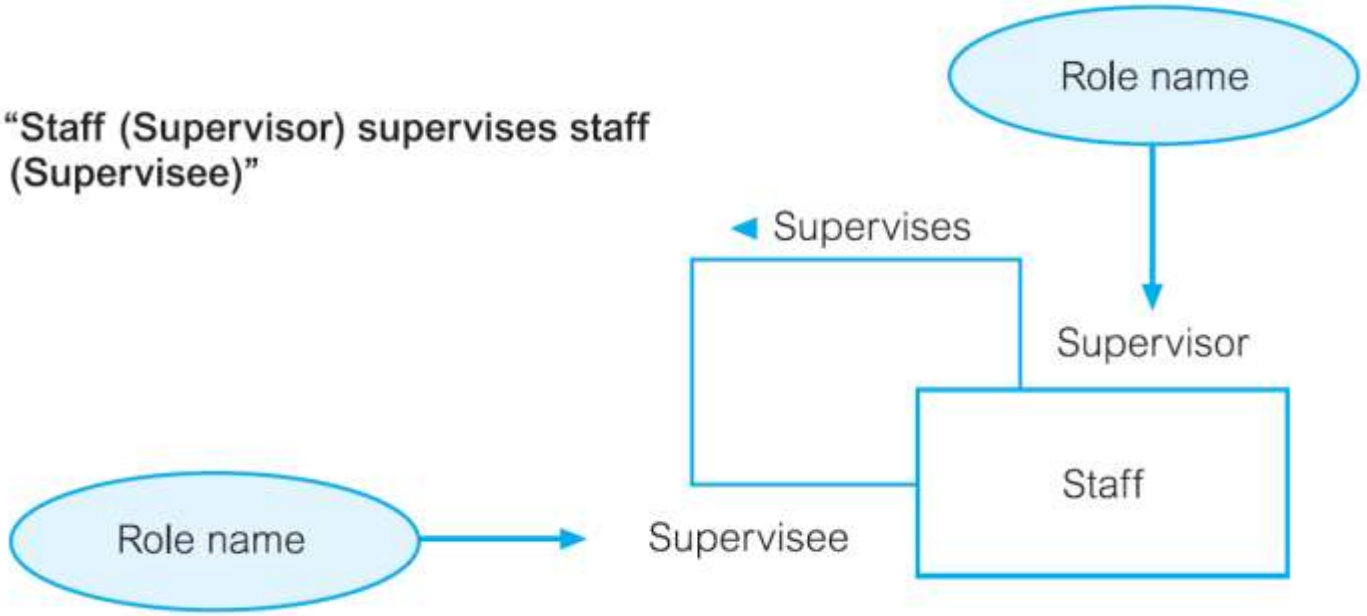
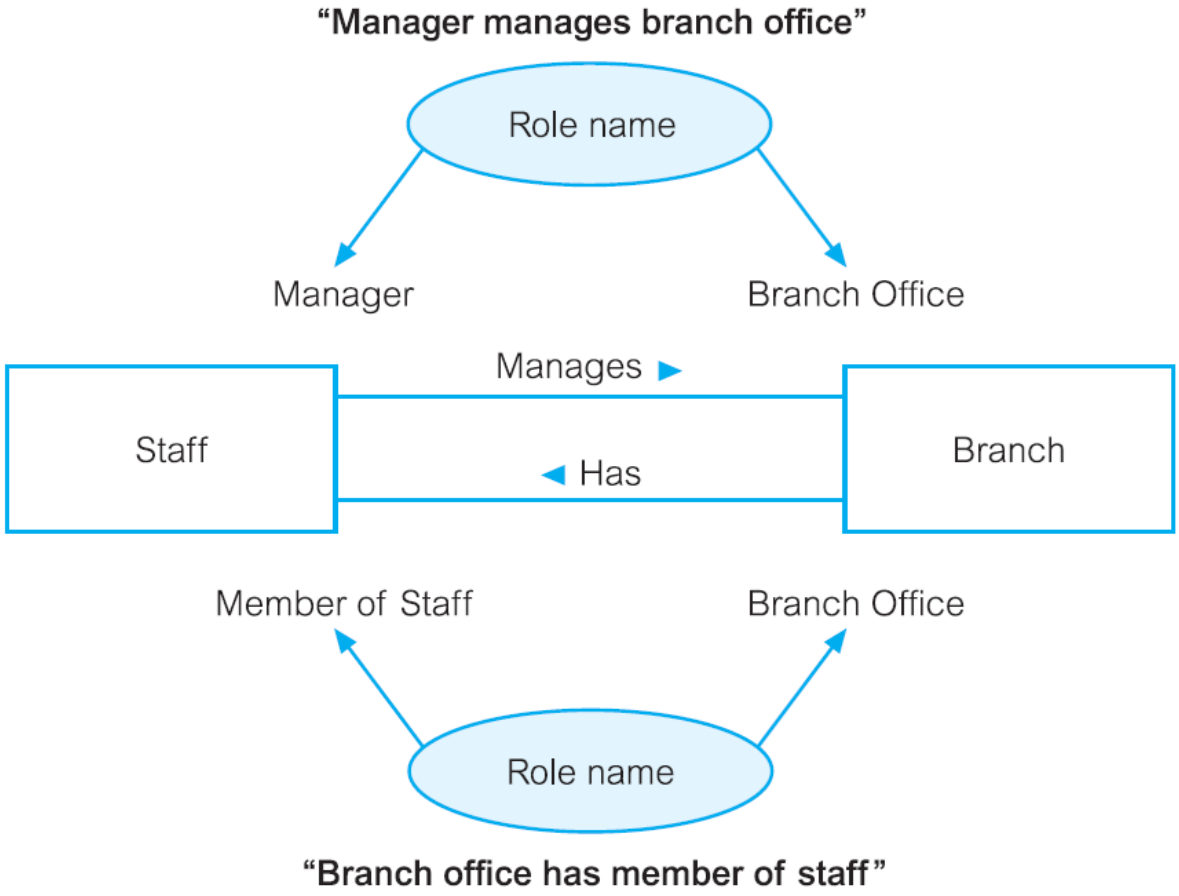


Figure 12.9
An example of a recursive relationship called *Supervises* with role names Supervisor and Supervisee.

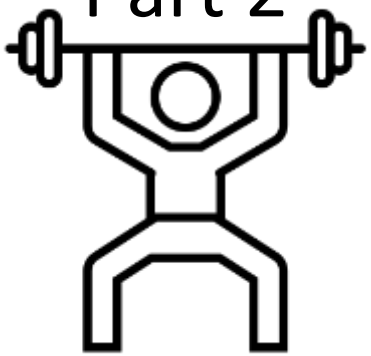
Entities associated through two distinct Relationships with Role Names



Practice 1 (Problem 1): Identify all entities and relationship between entities.

- I am the manager of a training company that provides instructor-led courses in management techniques. We teach many courses, each of which has a code, a name and a fee. Introduction to UNIX and C Programming are two of our more popular courses. Courses vary in length from one day to four days. An instructor can teach several courses. Paul Rogers and Maria Gonzales are two of our best teachers. We track each instructor's name and phone number. Each course is taught by only one instructor. We create a course and then line up an instructor. The students can take several courses over time, and many of them do this. Jamie Brown from AT&T took every course we offer! We track each student's name and phone number. Some of our students and instructors do not give us their phone numbers.

Part 2



Attributes

- **Attribute**
  - Also represents something of significance to the business
  - Is a specific piece of information that:
    - Describes
    - Quantifies
    - Qualifies
    - Classifies
    - Specifies an entity
  - Is a property of an entity or a relationship type.
- **Attribute Domain**
  - Set of allowable values for one or more attributes.

Attribute examples

Entity	Attribute
EMPLOYEE	Family Name, Age, Shoe Size, Town of Residence, Email, ...
CAR	Model, Weight, Catalog Price, …
ORDER	Order Date, Ship Date, …
JOB	Title, Description, ...
TRANSACTION	Amount, Transaction Date, …
EMPLOYMENT CONTRACT	Start Date, Salary, ...

Attribute (Simple vs Composite)

- **Simple Attribute**
  - Attribute composed of a single component with an independent existence.
  - Cannot be further subdivided into smaller components.
- **Composite Attribute**
  - Attribute composed of multiple components, each with an independent existence.
  - Example: address (simple attribute) can be subdivided into attributes of street, city, postcode

Attribute (Single-valued vs Multi-valued)

- **Single-valued Attribute**
  - Attribute that holds a single value for each occurrence of an entity type.
  - Majority of attributes are single-valued.
- **Multi-valued Attribute**
  - Attribute that holds multiple values for each occurrence of an entity type.
  - E.g.: Each branch have multiple telephone numbers. Thus, in this case, the attribute *telNo* is multi-valued.

Attribute (Derived)

- **Derived Attribute**
  - Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.
  - E.g.: the value for the ***duration*** attribute of the **Lease** entity is calculated from the ***rentStart*** and ***rentFinish*** attributes, also of the Lease entity type.

Practice 2 (Problem 1)

- Identify all attributes for each entity identified in Practice 1.
- Can you determine the type of each attribute?

Attribute: Keys

- **Candidate Key**
  - Minimal set of attributes that uniquely identifies each occurrence of an entity type.
- **Composite Key**
  - A candidate key that consists of two or more attributes.
- **Primary Key**
  - Candidate key selected to uniquely identify each occurrence of an entity type.

ER Diagram of Staff and Branch Entities and their Attributes

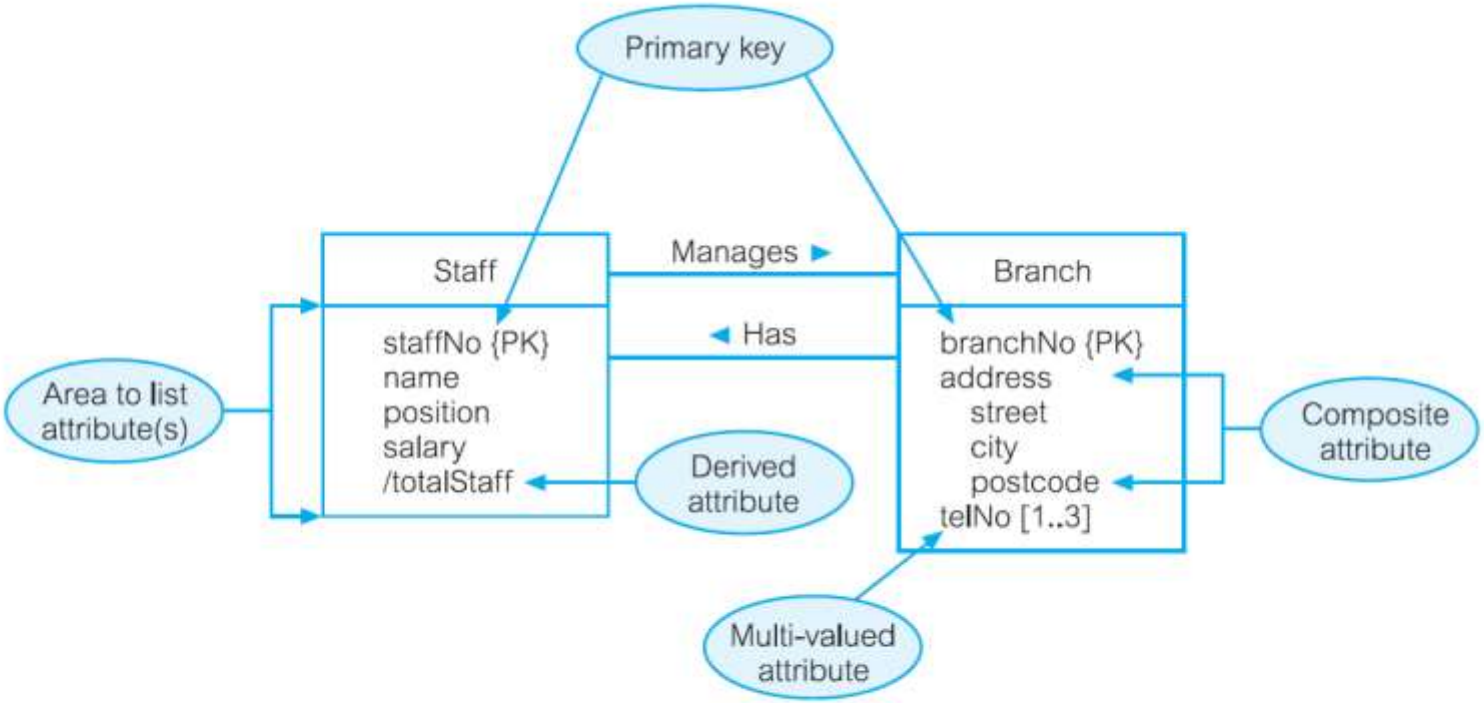


Figure 12.11 Diagrammatic representation of Staff and Branch entities and their attributes.

Practice 3 (Problem 1)

- Identify the primary key for each entity.

Entity Type

- **Strong Entity Type**
  - Entity type that is not existence-dependent on some other entity type.
  - Characteristic:
    - Each entity occurrence is uniquely identifiable using PK attribute of the entity type
- **Weak Entity Type**
  - Entity type that is existence-dependent on some other entity type.
  - Characteristic:
    - Each entity occurrence cannot be uniquely identified using attributes associated with the entity type

Strong vs Weak Entity

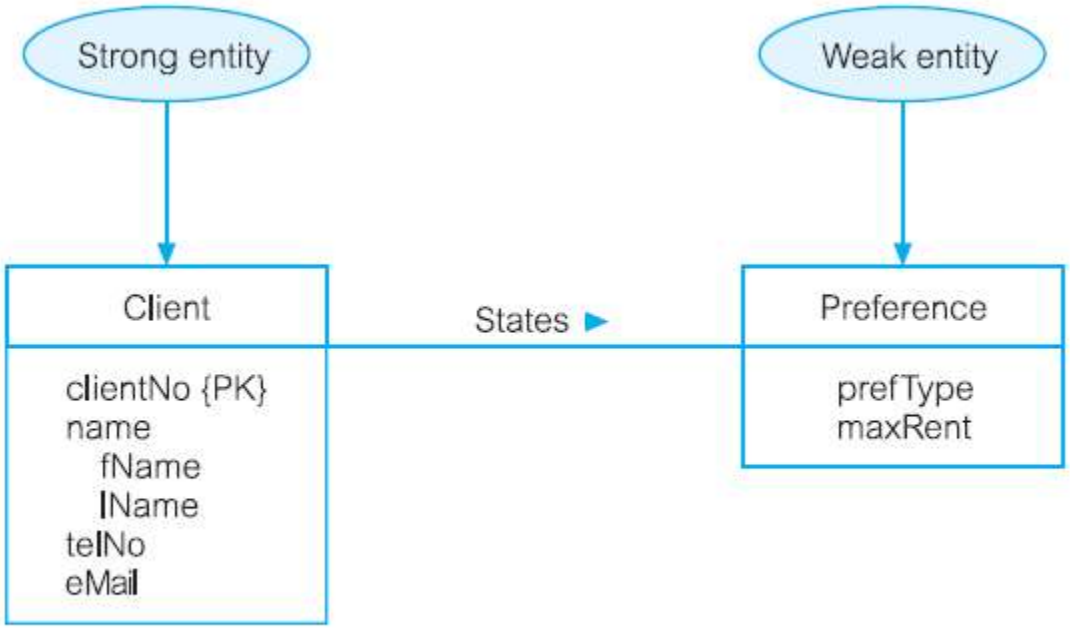
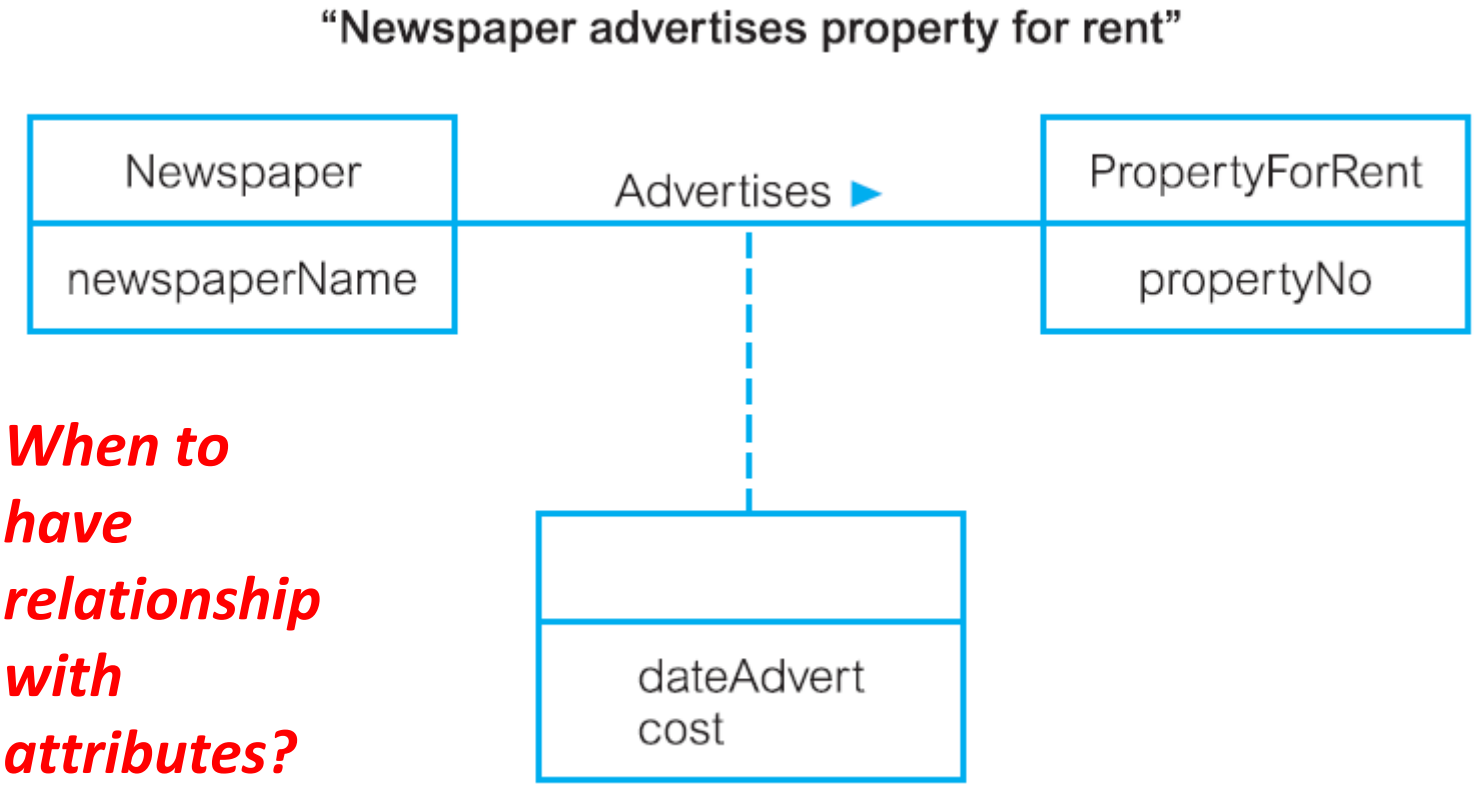
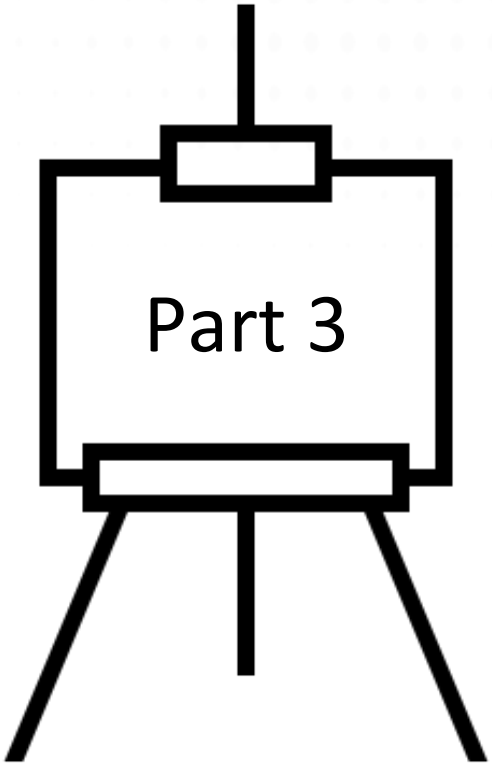


Figure 12.12 A strong entity type called Client and a weak entity type called Preference.

Relationship with Attributes



When to have relationship with attributes?


Part 3

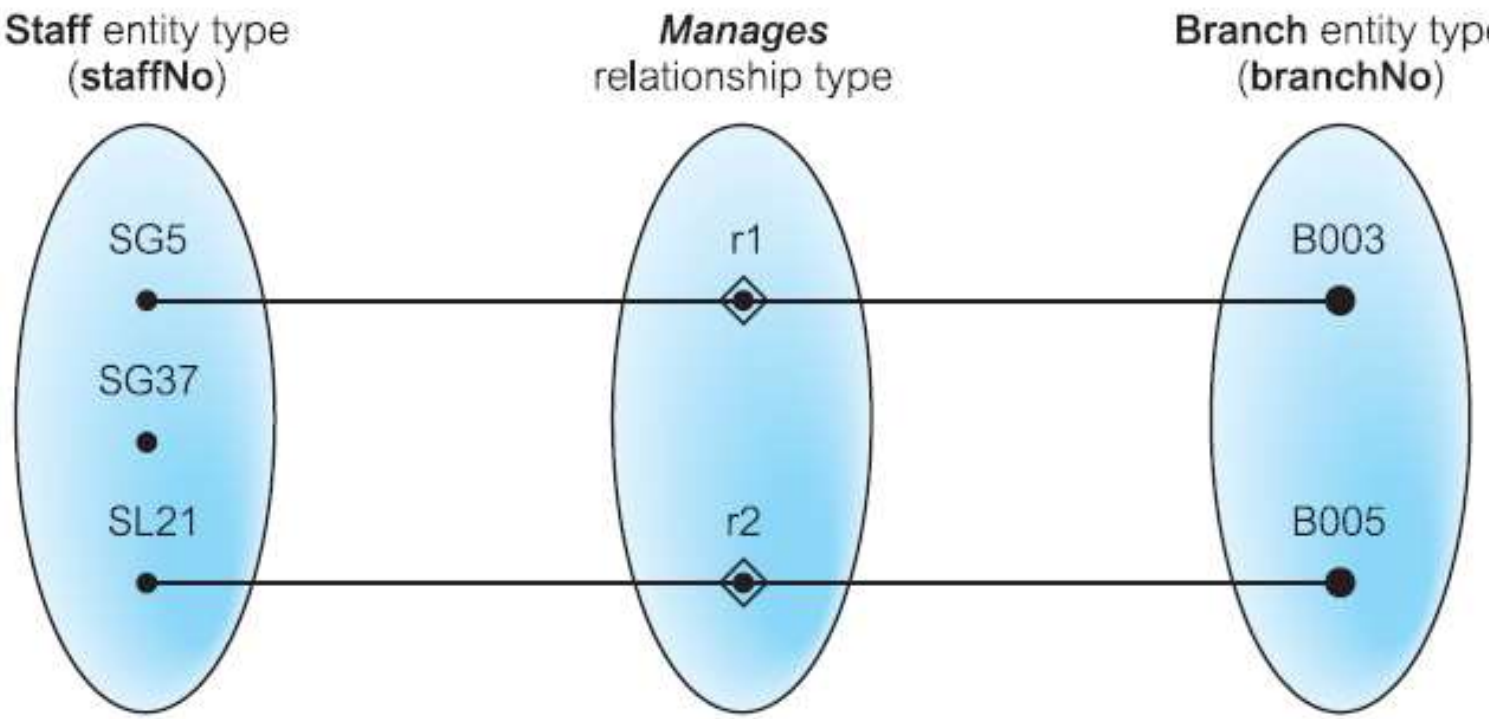
Structural Constraints

- Main type of constraint on relationships is called **multiplicity**.
- **Multiplicity**: **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.**
  - Constrains the way the entities are related in the relationship
  - Represents policies (called business rules) established by user or company.

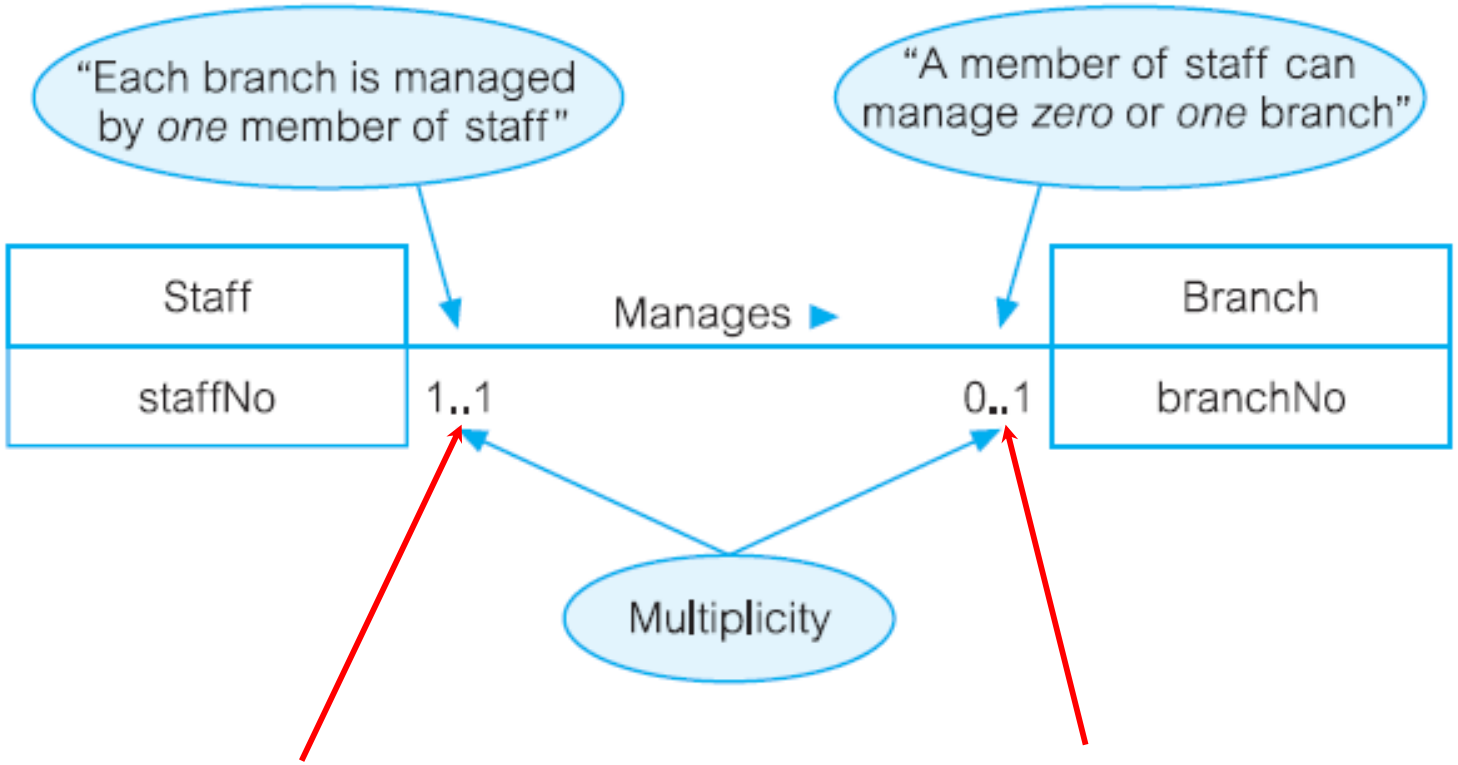
Structural Constraint (Binary Relationship)

- **Binary relationship** (i.e. the most common degree for relationships) is generally referred to as having a multiplicity of:
  - **one-to-one (1:1), or**
  - **one-to-many (1:*), or**
  - **many-to-many (*:*)**
- How to determine multiplicity:
  - Produce a semantic net with sample occurrences that can best represent the relationship of entities
  - Analyze the semantic net

Semantic Net: Staff *manages* Branch

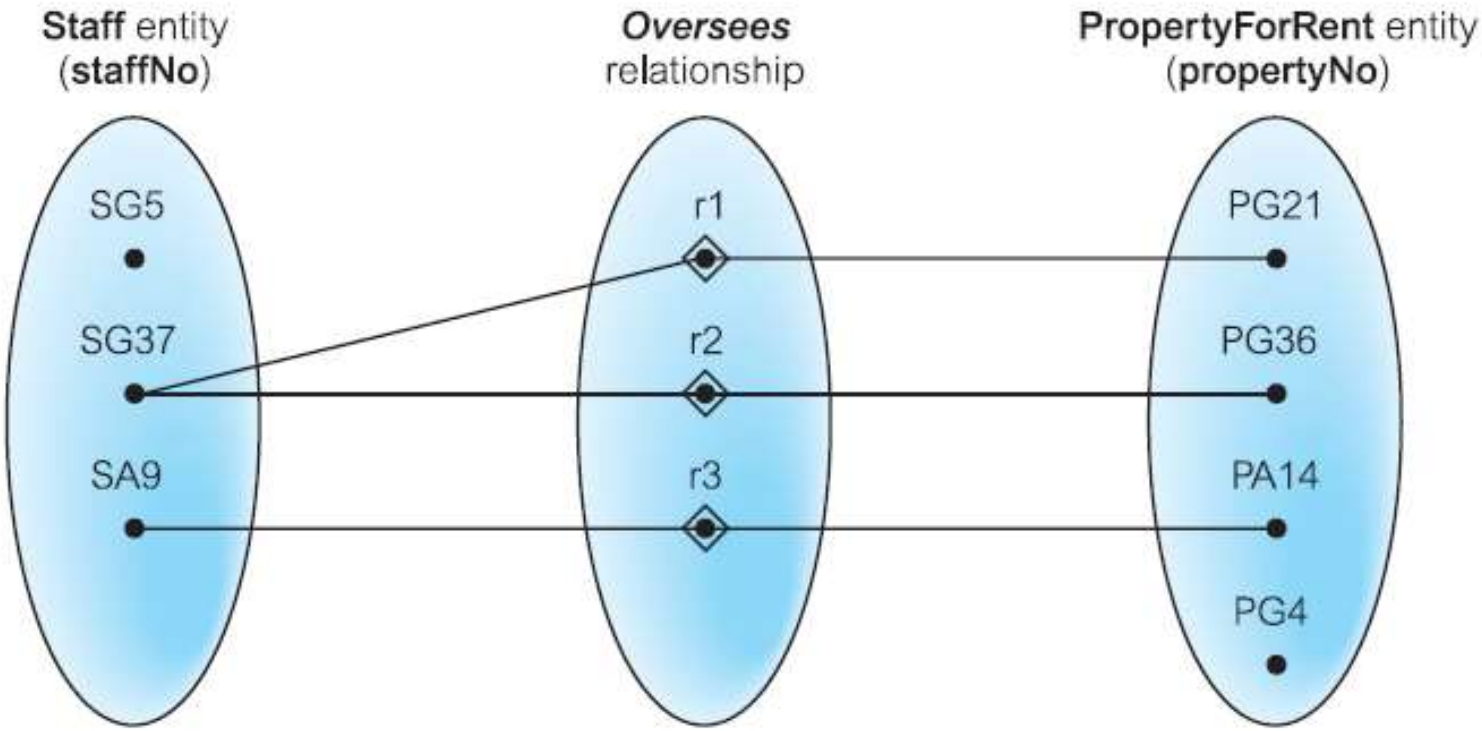


ERD: Staff *manages* Branch

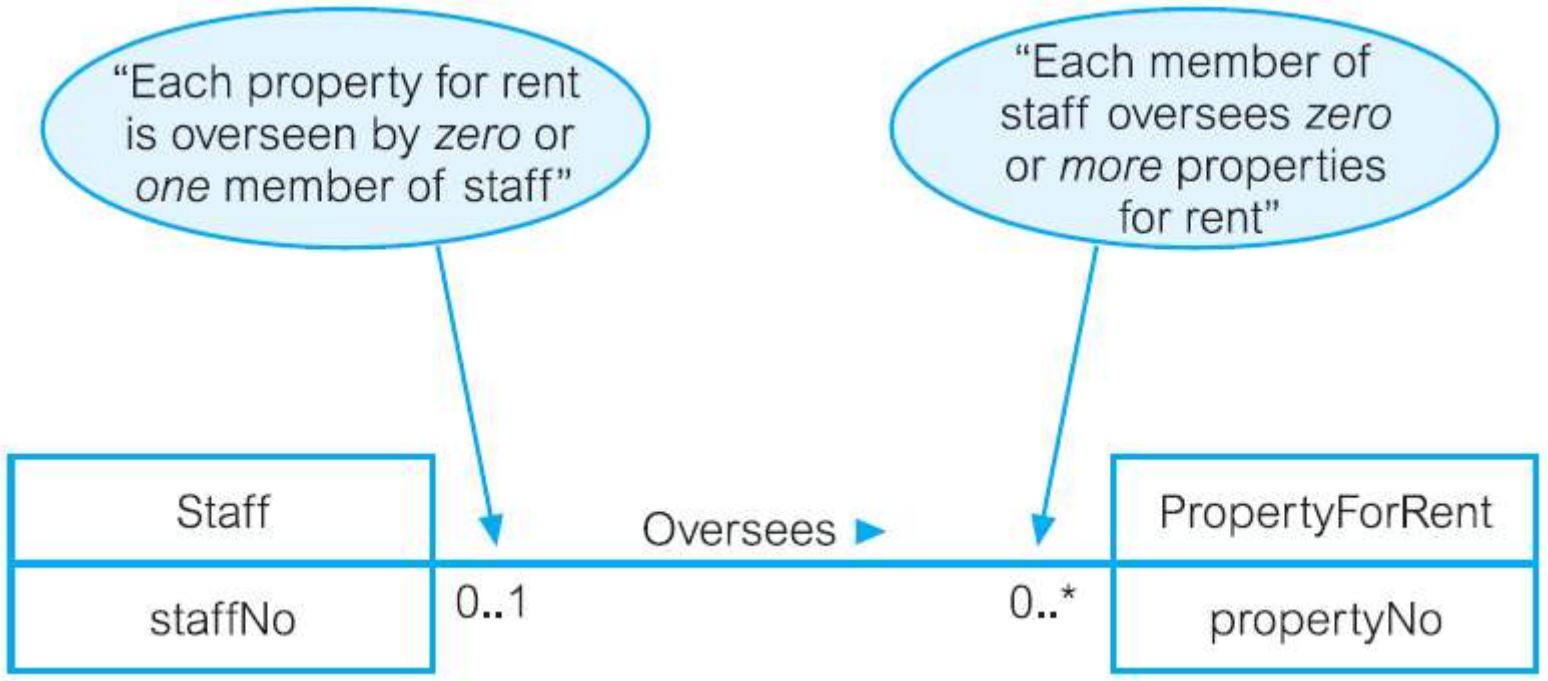


Multiplicity of Manages relationship is one-to-one (1:1) shown through the maximum range value on the multiplicities at both ends of relationship

Semantic Net: Staff *oversees* PropertyForRent

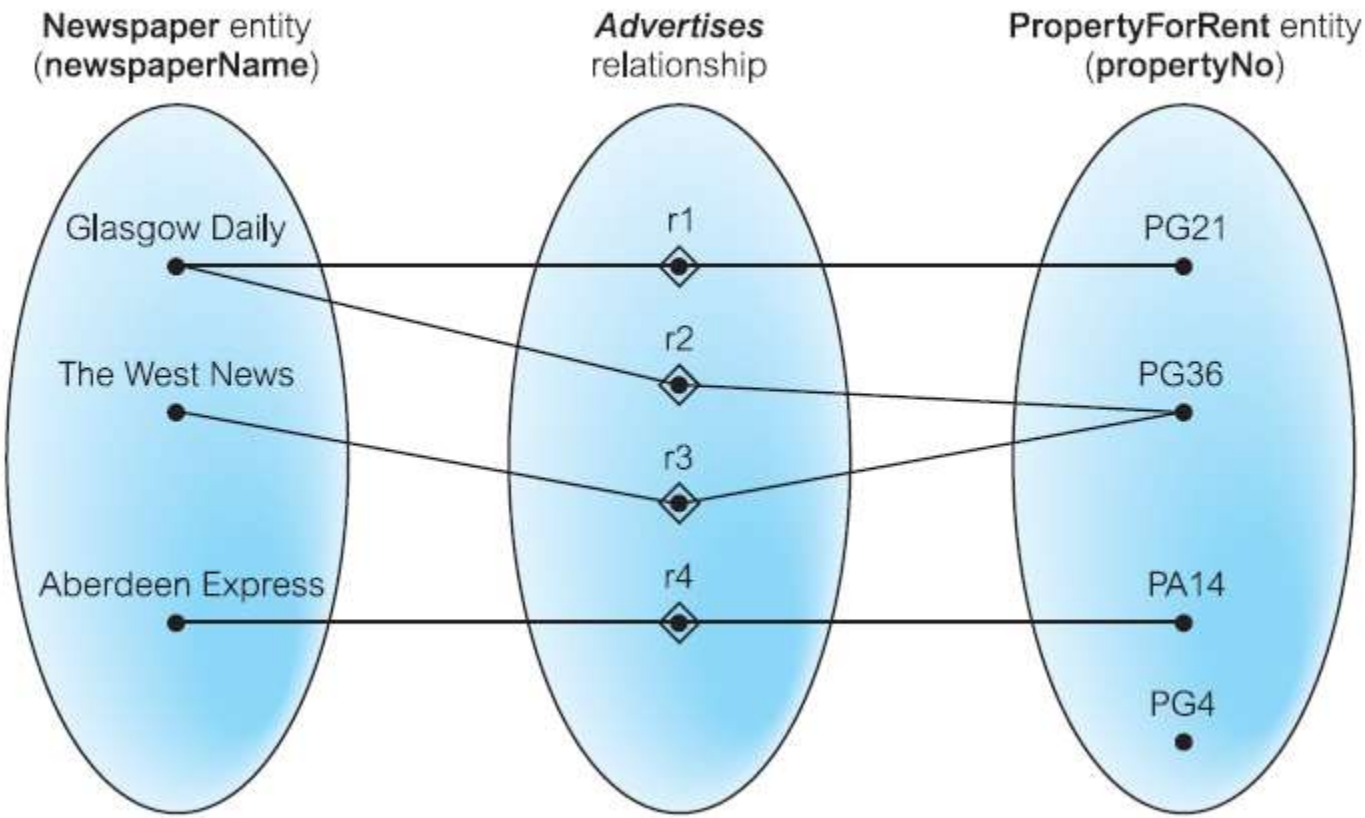


ERD: Staff *oversees* PropertyForRent

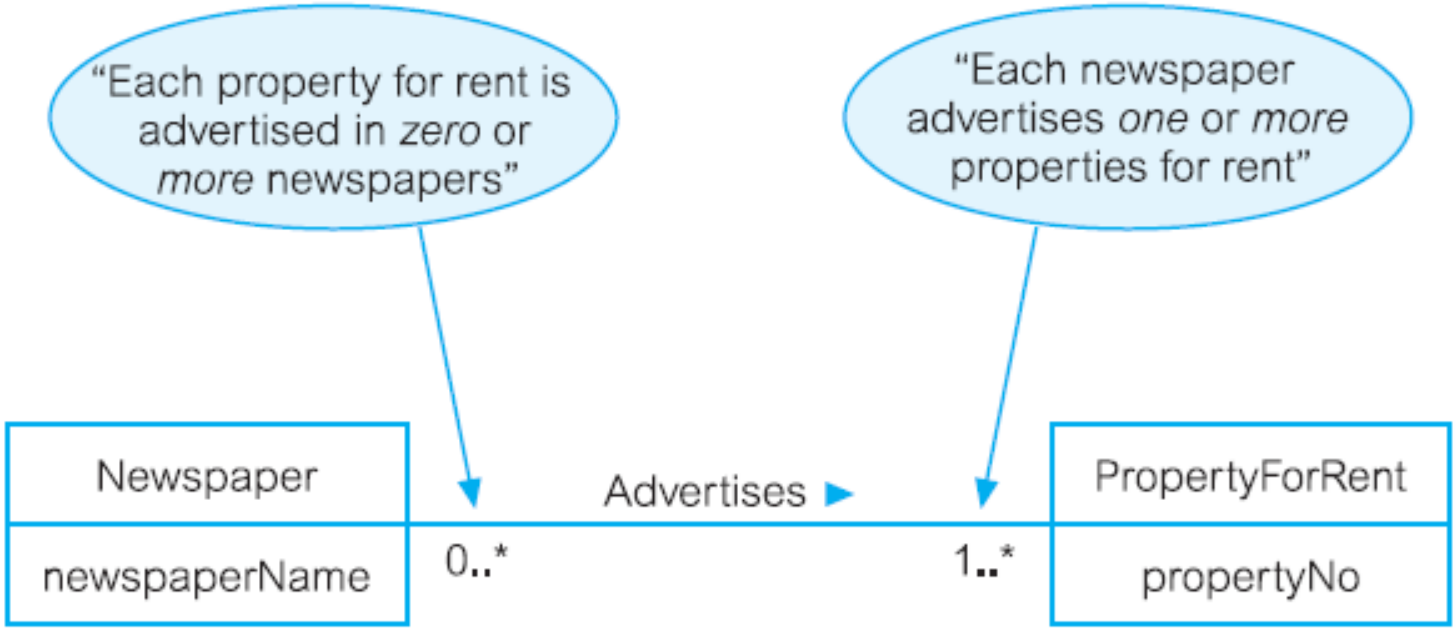


Oversees is a one-to-many (1:*) relationship

Semantic Net: Newspaper *Advertises* PropertyForRent



ERD: Newspaper *Advertises* PropertyForRent

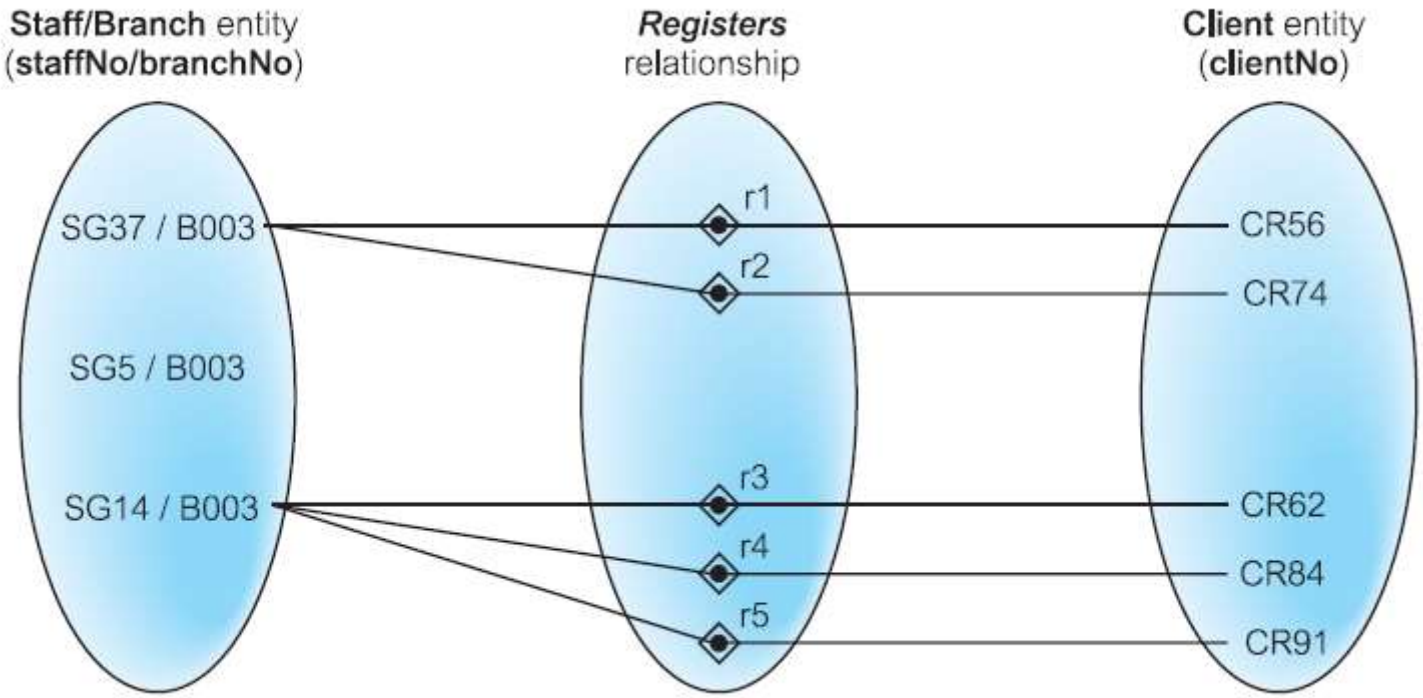


Advertises is a many-to-many (*:*) relationship

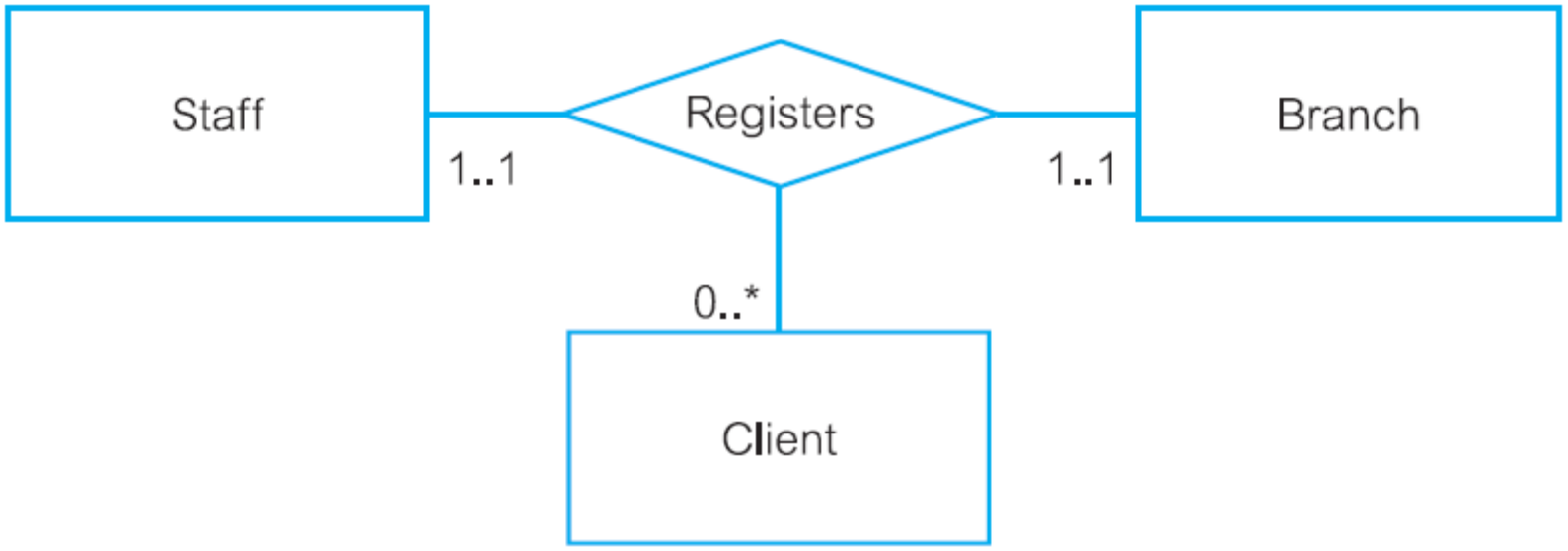
Structural Constraint (Complex Relationship)

- **Multiplicity for Complex Relationships**
  - Complex relationship – relationship higher than binary
  - Number (or range) of possible occurrences of an entity type in an *n*-ary relationship when other (*n*-1) values are fixed.

Semantic Net: Ternary Registers Relationship with Values for Staff and Branch Entities Fixed



Multiplicity of Ternary Registers Relationship



Staff of a branch registers zero client or more

Summary of Multiplicity Constraints

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

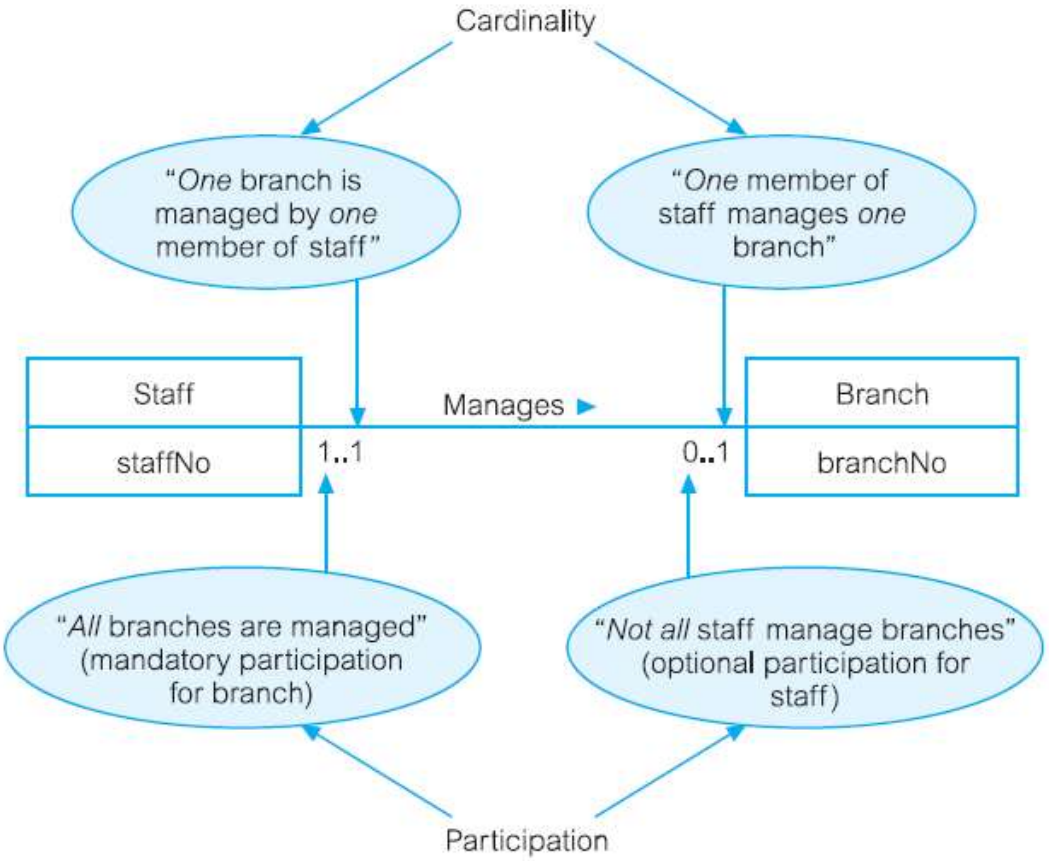
Constraints (Cardinality & Participation)

- From multiplicity we can identify two type of restrictions on relationships: **cardinality** and **participation**.
- **Cardinality**
  - Describes maximum number of possible relationship occurrences for an entity participating in each relationship type.
  - Appears as the maximum values for the multiplicity ranges on either side of the relationship

Constraints (Cardinality & Participation)

- **Participation (mandatory/optional)**
  - determines whether **all** or **only some** entity occurrences are involved (participate) in a relationship.
  - All is **mandatory** (minimum value **1**) ;
  - only some is **optional** (minimum value **0**)
  - How to determine an entity's participation on a diagram?
    - Look at the minimum value of a range at one entity.
    - If value is 0, then the opposite entity (in that relationship) has an **optional participation**
    - If value is 1, then the opposite entity (in that relationship) has a **mandatory participation**

Multiplicity as Cardinality and Participation Constraints



Practice 4: Problem 1

- Complete the ERD for the problem by:
  - Determine whether the entities are strong entity or weak entity;
  - Determine the multiplicity and structural constraints (cardinality and participation) for each relationship in the diagram

Structural Constraints (cont)

- Why need to understand and able to differentiate between the different type of structural constraints in data model (ERD)?
  - Relational schemas will be created from entities-relationship in ERD
  - Different type of constraints will affect the schemas created (will be discussed in logical database design topic)

Problem 2: Produce ERD to represent the following situation

- Model the USA-type geography, where the following is of interest:
  - States of a country including their name, population, area and date established.
  - Cities including their name, population, date established and founder's name.
  - Rivers including their name and length.
  - The capital city of each state, including the date declared.
  - The source of each river, i.e. the state in which it begins (which may be none)
  - The fact that cities may be located on particular rivers.
  - The fact that cities are located in states.
  - The length of each river within each state.
  - Information about states that adjoin each another, and the length of their common border.
  - Assume that the names of states and rivers are unique. City names are only unique within a state.

Problem 3: Produce ERD to represent the following situation

- Consider the following set of requirements for a university database that is used to keep track of students' transcripts.
  - The university keeps track of each student's name, student number, current address, permanent address and phone, birthdate, gender, year of study (1st, 2nd, 3rd, 4th), major department, minor department (if any), and degree program (B.A., B.Sc., ..., Ph.D.). Some user applications need to refer to the city, county, and post code of the student's permanent address and to the student's last name.
  - Each department is described by name, department code, office number and office phones. Both name and code have unique values for each department.
  - Each course has a course name, description, code number, number of semester hours and offering department. The value of the code number is unique for each course.
  - Each module has an instructor, semester, year, course and module number. The module number distinguishes different modules of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of taught during each semester.
  - A grade report has a student, module and grade. modules

Innovating Solutions
Menginovasi Penyelesaian