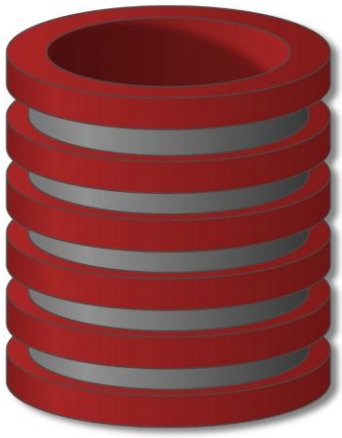


Entity Relationship Modeling

Chapter # 11

Imran Khan
FCS, IBA

Objectives

- How to use Entity–Relationship (ER) modeling in database design.
- Basic concepts associated with ER model.
- How to identify and resolve problems with ER models called connection traps.
- How to build an ER model from a requirements specification.

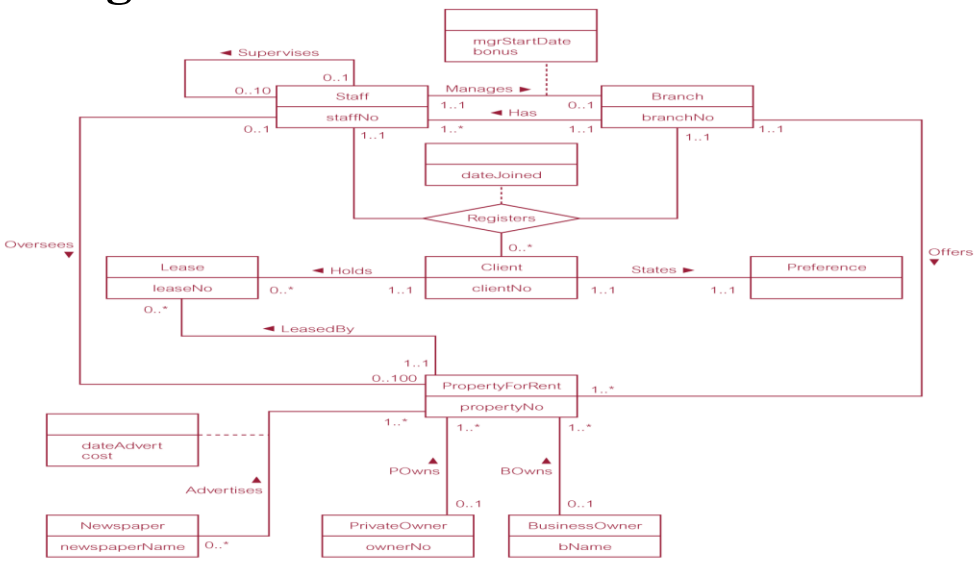
2

In this chapter, you will learn:

- What normalization is and what role it plays in database design
- About the normal forms 1NF, 2NF, 3NF, BCNF, and 4NF
- How normal forms can be transformed from lower normal forms to higher normal forms
- That normalization and E-R modeling are used concurrently to produce a good database design
- That some situations require denormalization to generate information efficiently

3

ER diagram of Branch view of *DreamHome*



4

Concepts of the ER Model

- Entity types
- Relationship types
- Attributes

5

Entity Type

- **Entity type**
  - Group of objects with same properties, identified by enterprise as having an independent existence.
- **Entity occurrence**
  - Uniquely identifiable object of an entity type.

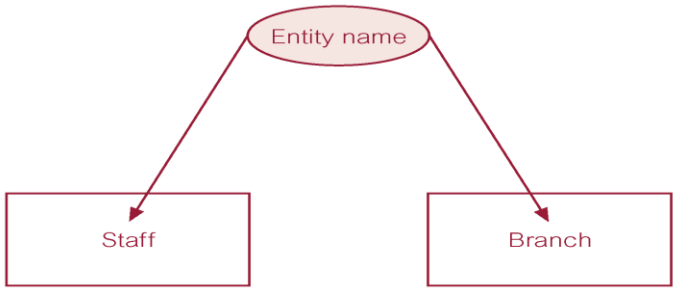
6

Examples of Entity Types

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

7

ER diagram of Staff and Branch entity types



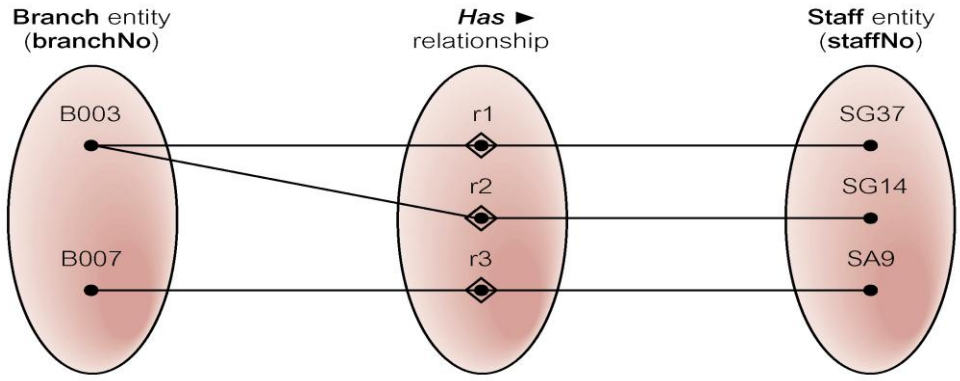
8

Relationship Types

- **Relationship type**
  - Set of meaningful associations among entity types.
- **Relationship occurrence**
  - Uniquely identifiable association, which includes one occurrence from each participating entity type.

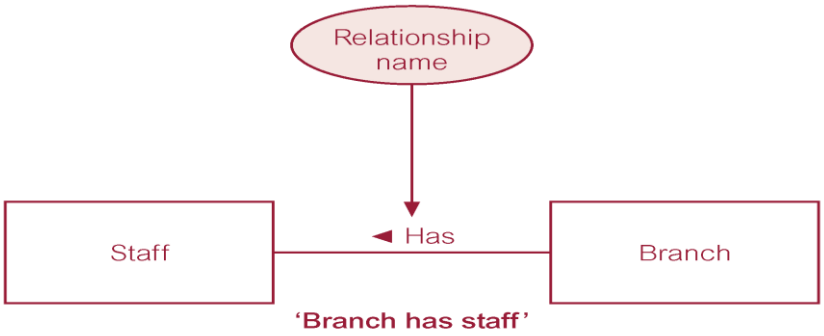
9

Semantic net of *Has* relationship type



10

ER diagram of Branch *Has* Staff relationship



11

Relationship Types

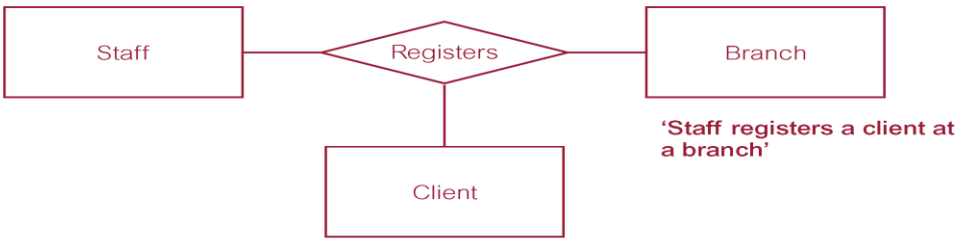
- **Degree of a Relationship**
  - Number of participating entities in relationship.
- **Relationship of degree :**
  - two is binary
  - three is ternary
  - four is quaternary

12

Binary relationship called *POwns*

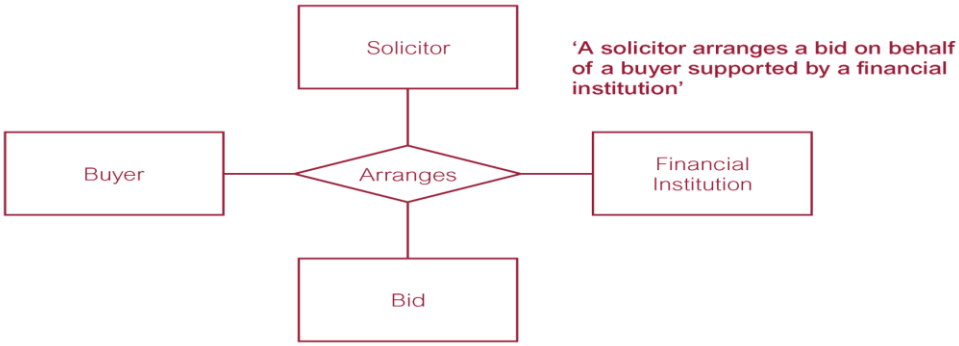
13

Ternary relationship called *Registers*



14

Quaternary relationship called *Arranges*



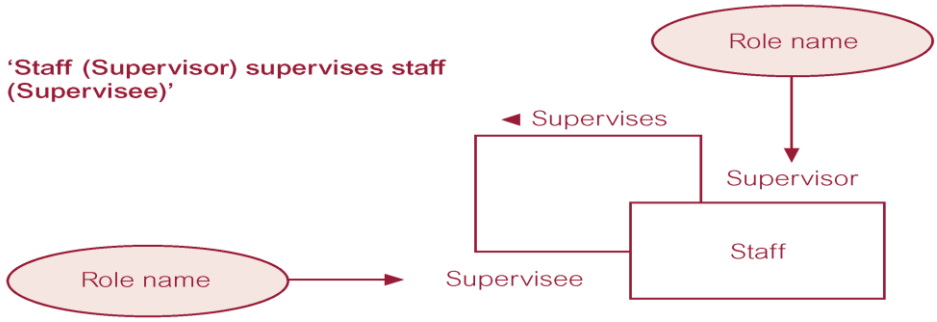
15

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.

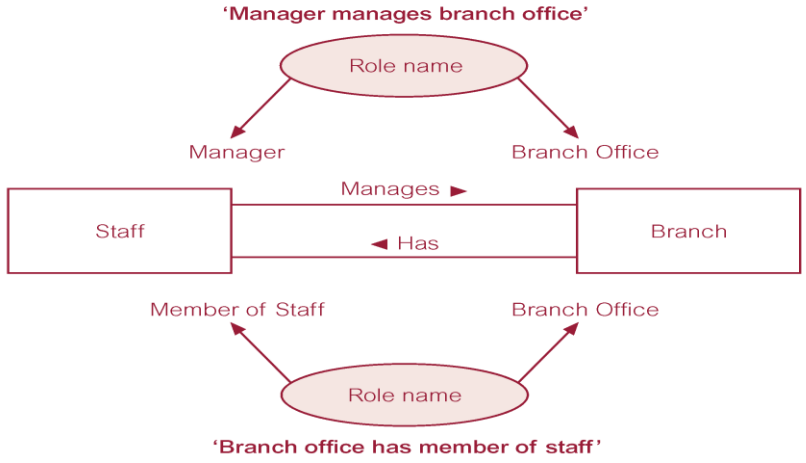
16

Recursive relationship called *Supervises* with role names



17

Entities associated through two distinct relationships with role names



18

Attributes

- **Attribute**
  - Property of an entity or a relationship type.
- **Attribute Domain**
  - Set of allowable values for one or more attributes.

19

Attributes

- **Simple Attribute**
  - Attribute composed of a single component with an independent existence.
- **Composite Attribute**
  - Attribute composed of multiple components, each with an independent existence.

20

Attributes

- **Single-valued Attribute**
  - Attribute that holds a single value for each occurrence of an entity type.
- **Multi-valued Attribute**
  - Attribute that holds multiple values for each occurrence of an entity type.

21

Attributes

- **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.

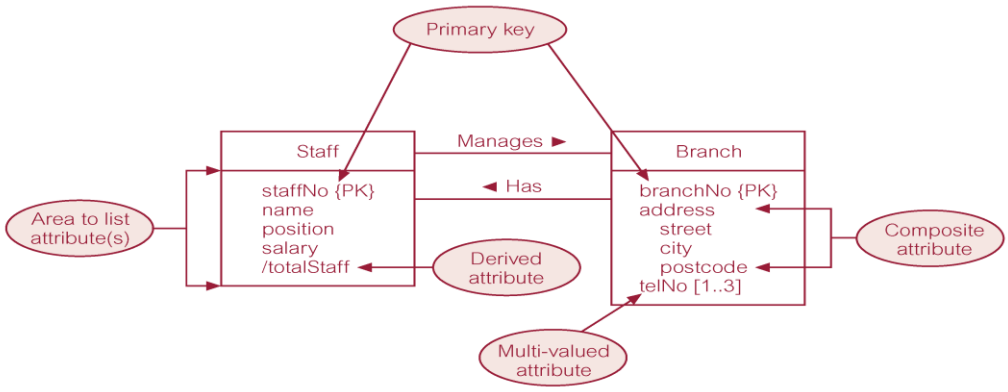
22

Keys

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

23

ER diagram of Staff and Branch entities and their attributes



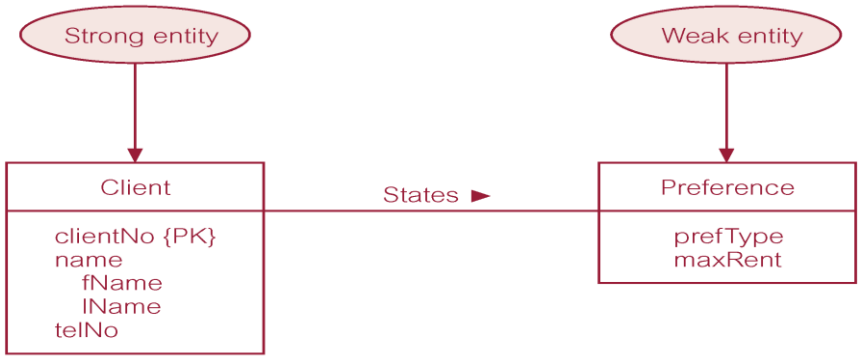
24

Entity Type

- **Strong Entity Type**
  - Entity type that is not existence-dependent on some other entity type.
- **Weak Entity Type**
  - Entity type that is existence-dependent on some other entity type.

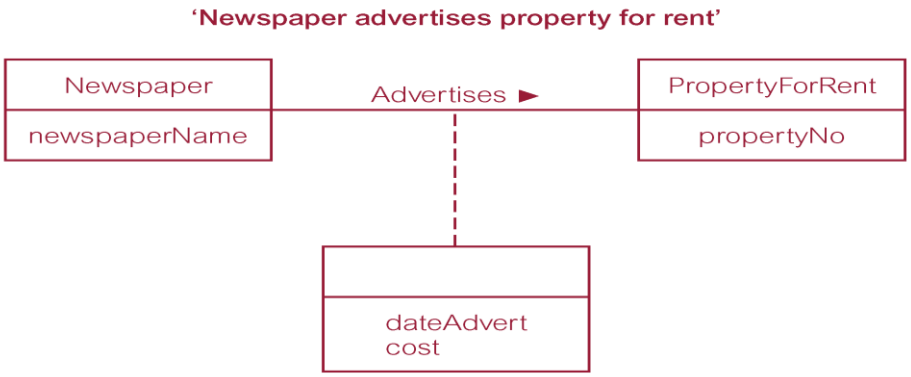
25

Strong entity type called Client and weak entity type called Preference



26

Relationship called *Advertises* with attributes



27

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.
- Represents policies (called *business rules*) established by user or company.

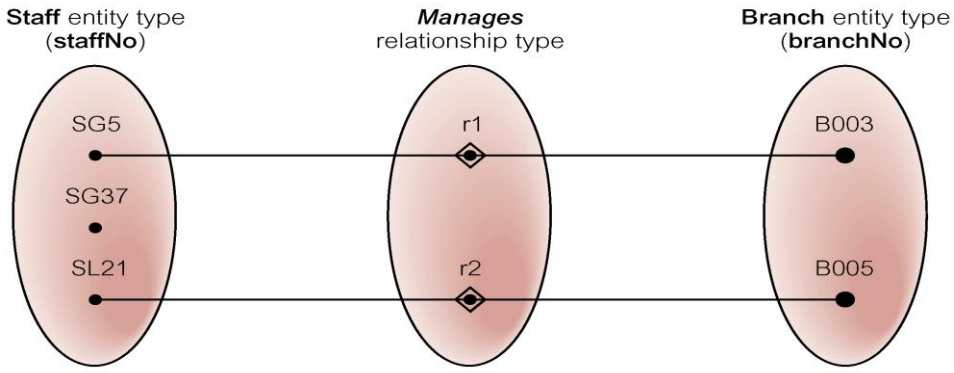
28

Structural Constraints

- 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:*)
  - many-to-many (*:*)

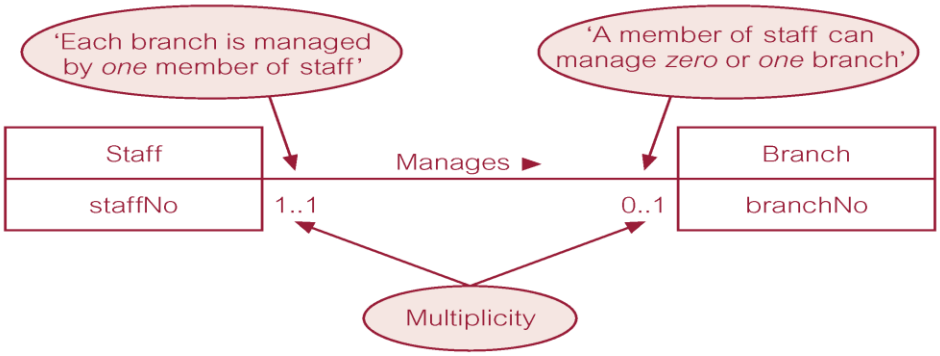
29

Semantic net of Staff *Manages* Branch relationship type



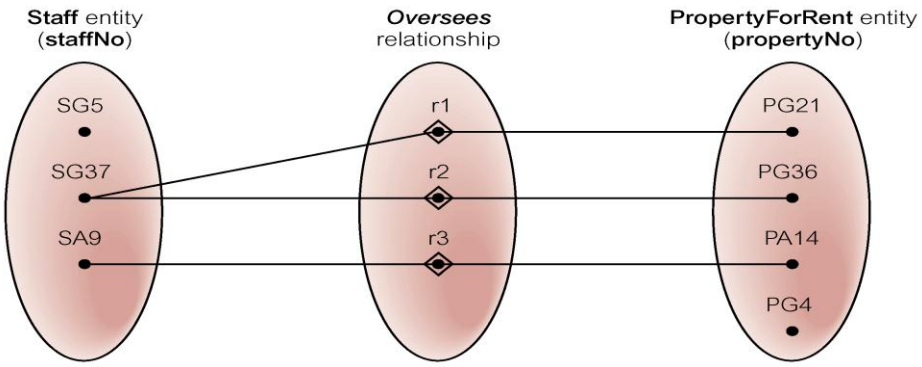
30

Multiplicity of Staff *Manages* Branch (1:1) relationship



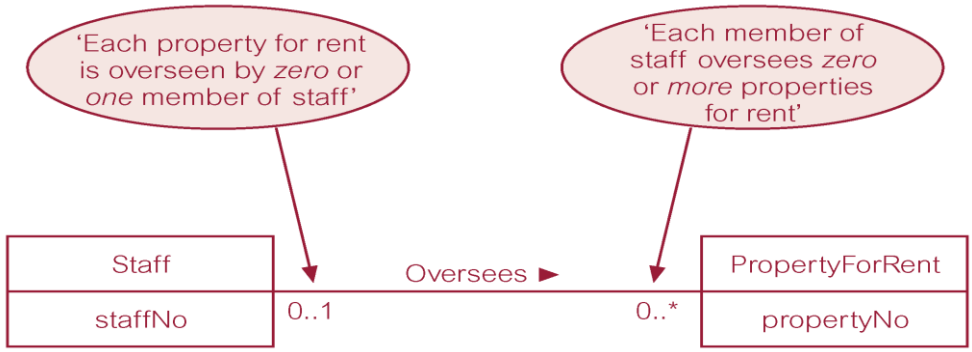
31

Semantic net of Staff *Oversees* PropertyForRent relationship type



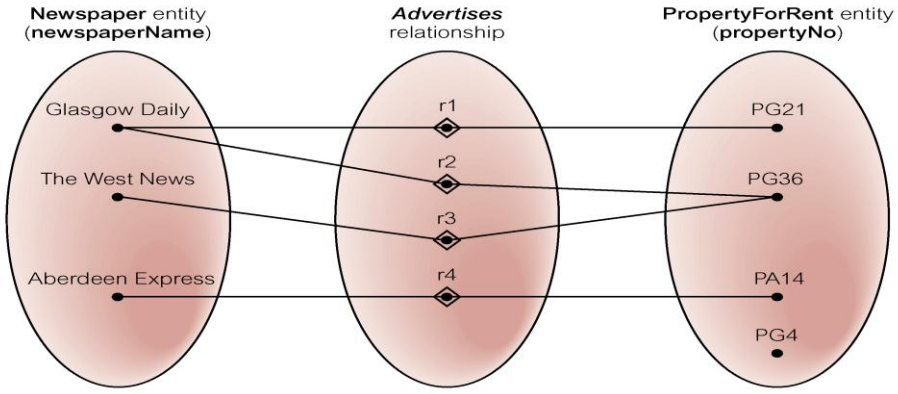
32

Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type



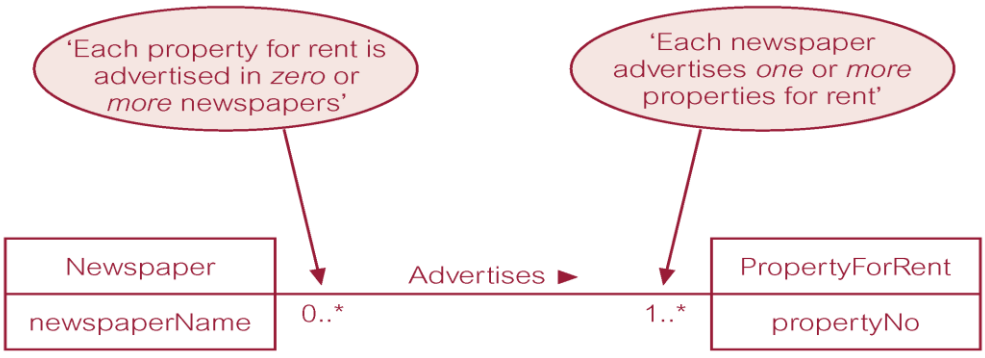
33

Semantic net of Newspaper *Advertises* PropertyForRent relationship type



34

Multiplicity of Newspaper *Advertises* PropertyForRent (*:*) relationship



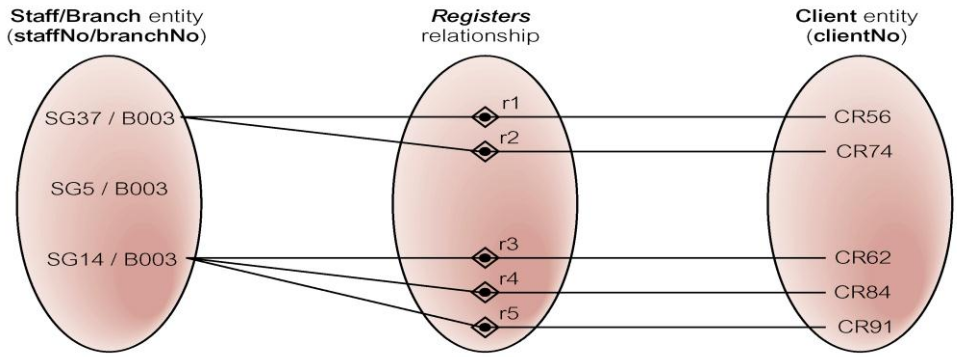
35

Structural Constraints

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

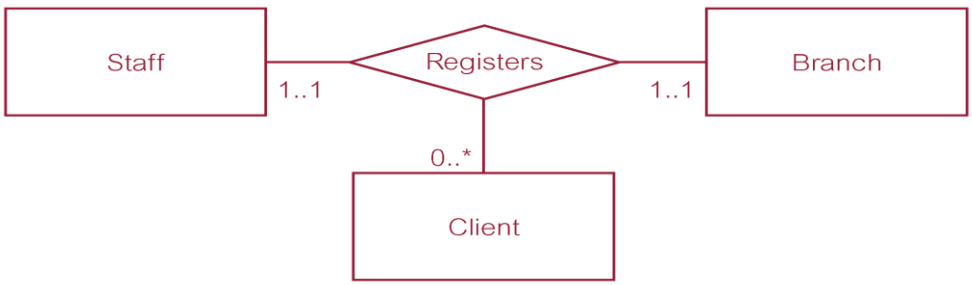
36

Semantic net of ternary *Registers* relationship with values for Staff and Branch entities fixed



37

Multiplicity of ternary *Registers* relationship



38

Summary of multiplicity constraints

Table 11.1 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

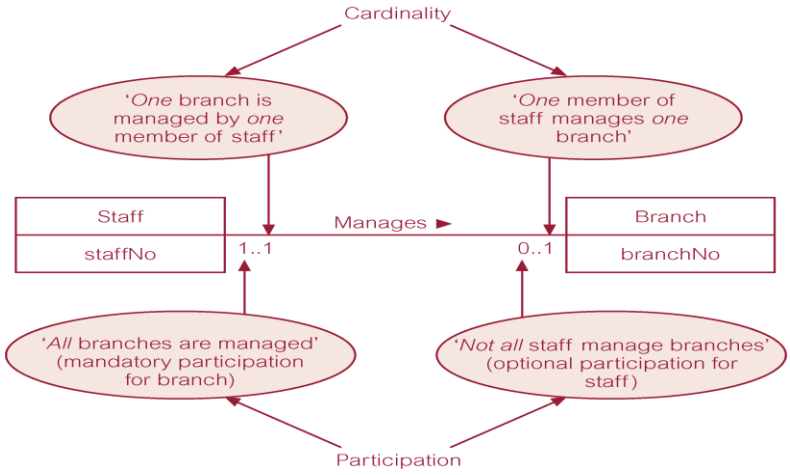
39

Structural Constraints

- Multiplicity is made up of two types of restrictions on relationships: cardinality and participation.
- **Cardinality**
  - Describes maximum number of possible relationship occurrences for an entity participating in a given relationship type.
- **Participation**
  - Determines whether all or only some entity occurrences participate in a relationship.

40

Multiplicity as cardinality and participation constraints



41

Further Objectives

- How to identify and resolve problems with ER models called connection traps.

42

Problems with ER Models

- Problems may arise when designing a conceptual data model called **connection traps**.
- Often due to a misinterpretation of the meaning of certain relationships.
- Two main types of connection traps are called ***fan traps*** and ***chasm traps*****.**

43

Problems with ER Models

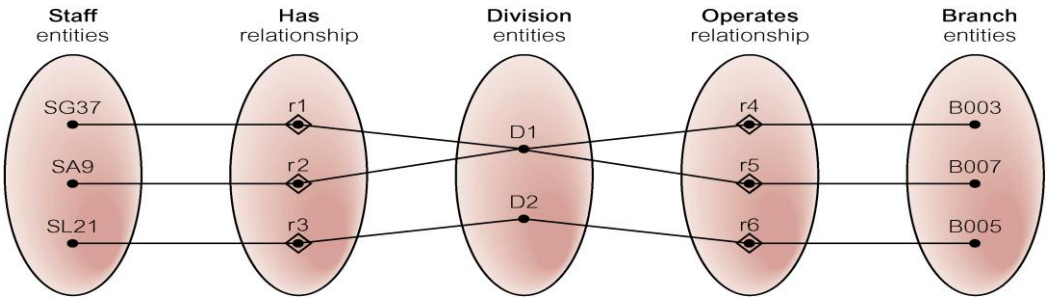
- **Fan Trap**
  - Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.
- **Chasm Trap**
  - Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.

44

An Example of a Fan Trap

45

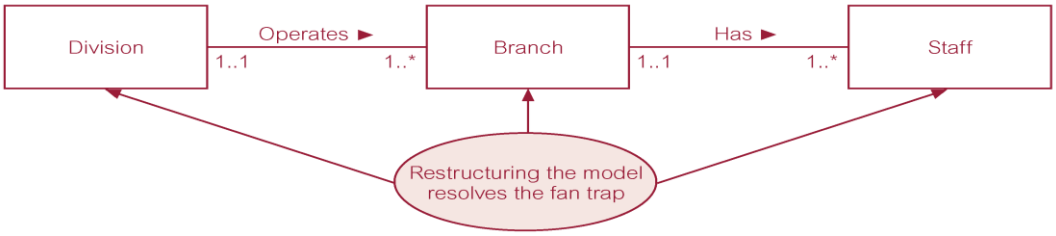
Semantic Net of ER Model with Fan Trap



- At which branch office does staff number SG37 work?

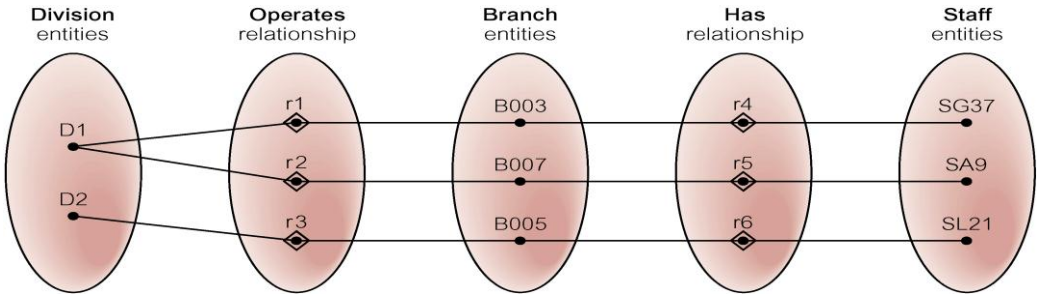
46

Restructuring ER model to remove Fan Trap



47

Semantic Net of Restructured ER Model with Fan Trap Removed



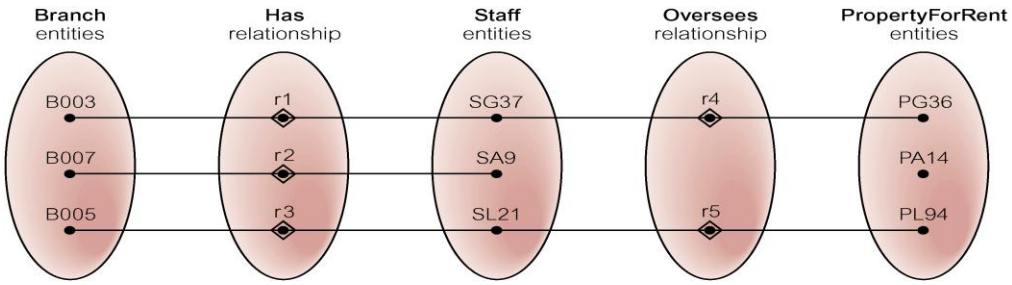
- SG37 works at branch B003.

48

An Example of a Chasm Trap

49

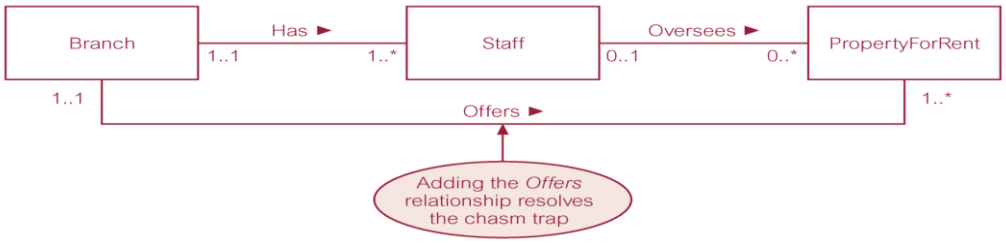
Semantic Net of ER Model with Chasm Trap



- At which branch office is property PA14 available?

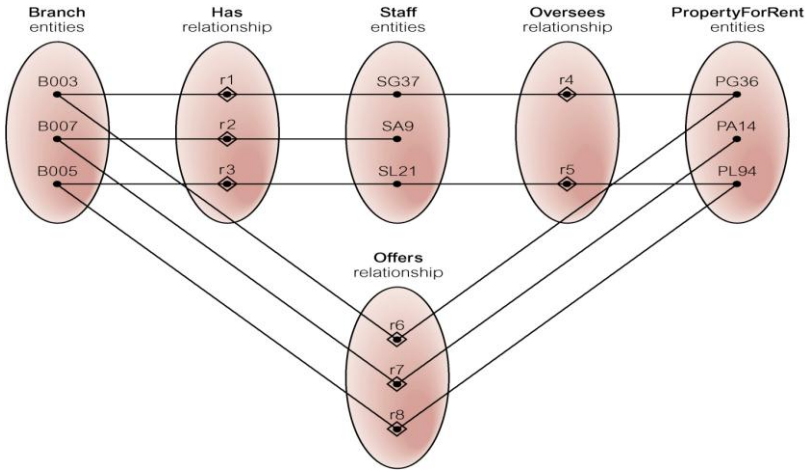
50

ER Model restructured to remove Chasm Trap



51

Semantic Net of Restructured ER Model with Chasm Trap Removed



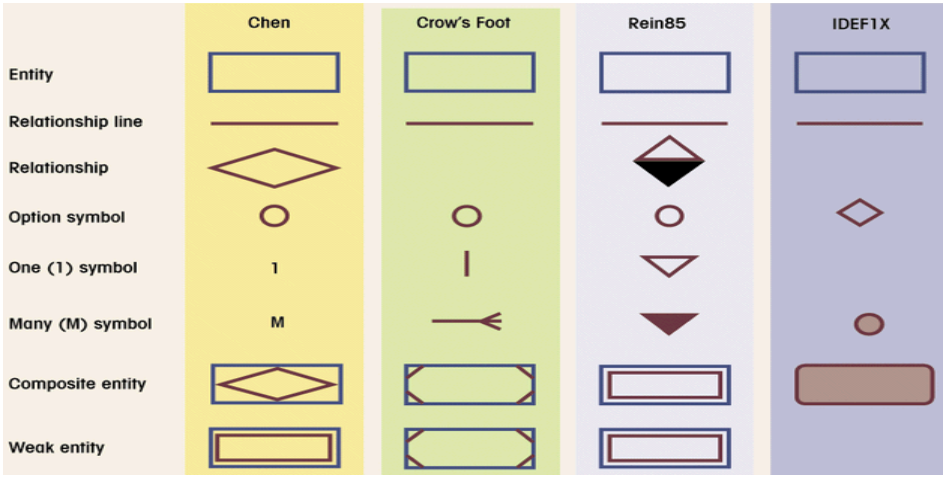
52

Comparison of E-R Modeling Symbols

- Alternate styles developed to enable easier use of CASE tools
- **Chen**
  - Moved conceptual design into practical database design arena
- **Crow's Foot**
  - Cannot detail all cardinalities
- **Rein85**
  - Similar to Crow's Foot
  - Operates at higher level of abstraction
- **IDEF1X**
  - Derivative of ICAM studies in the late 1970's
  - Uses fewer symbols

53

Comparison of E-R Modeling Symbols



54