



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022

E-R Model

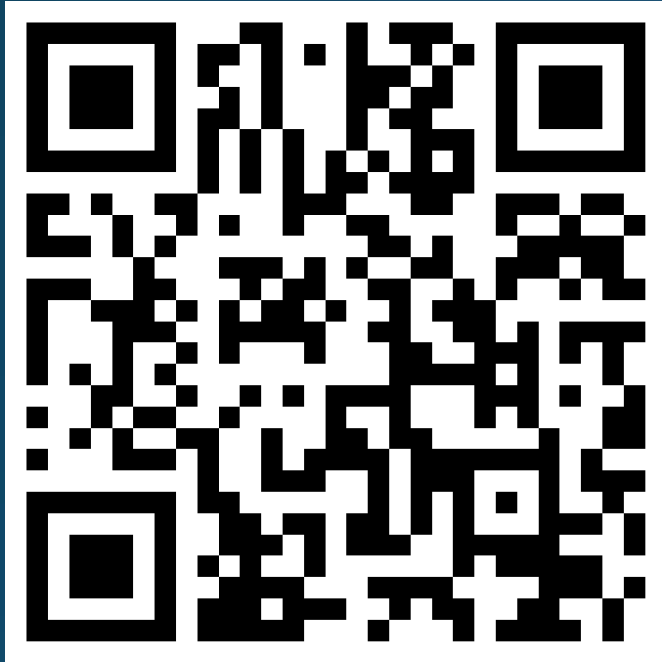
Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

www.uwtsd.ac.uk



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022



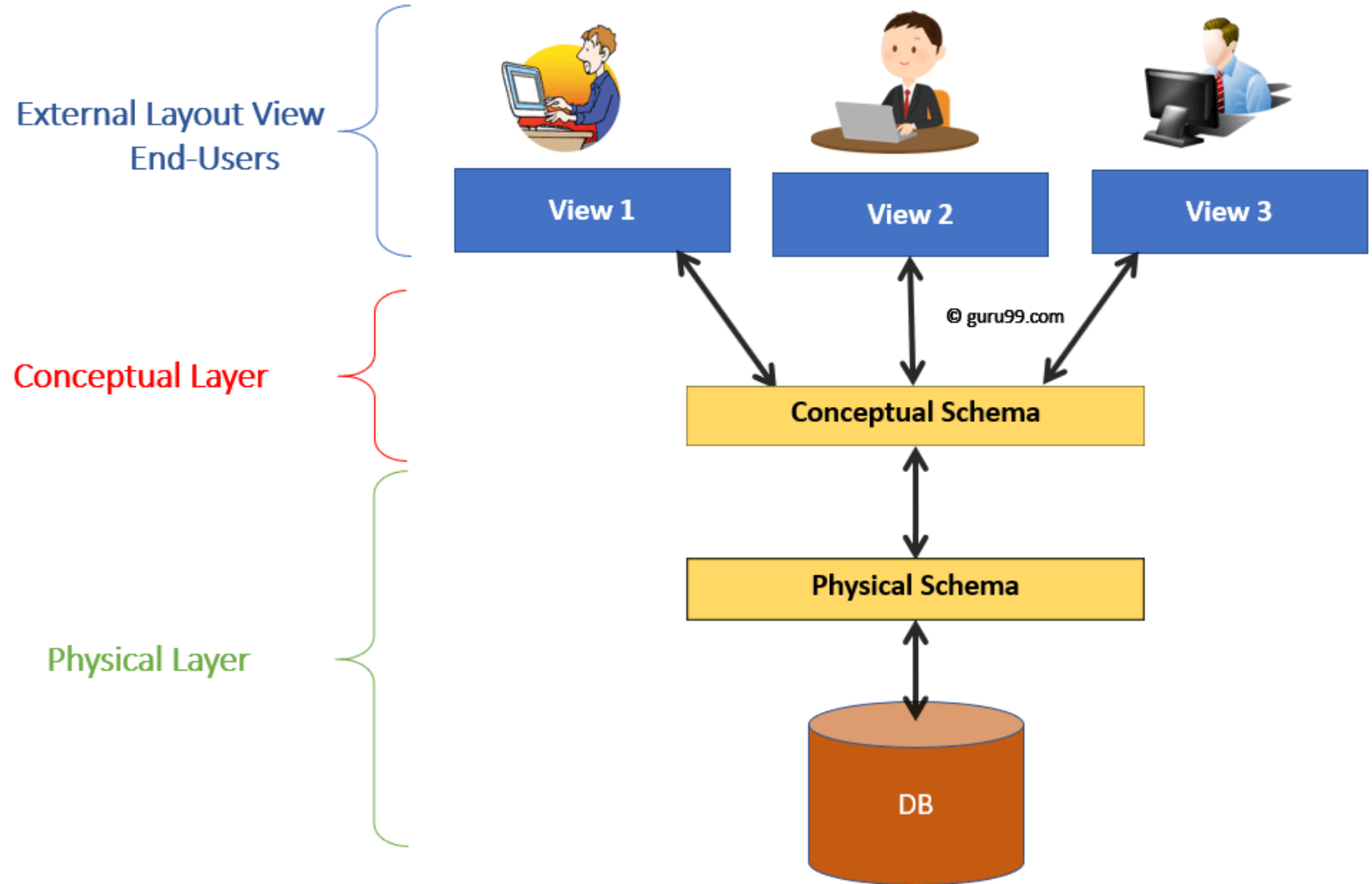
Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

www.uwtsd.ac.uk

<https://forms.office.com/e/9hRmmBaT3r>

Recap:

Three Level ANSI-SPARC Architecture



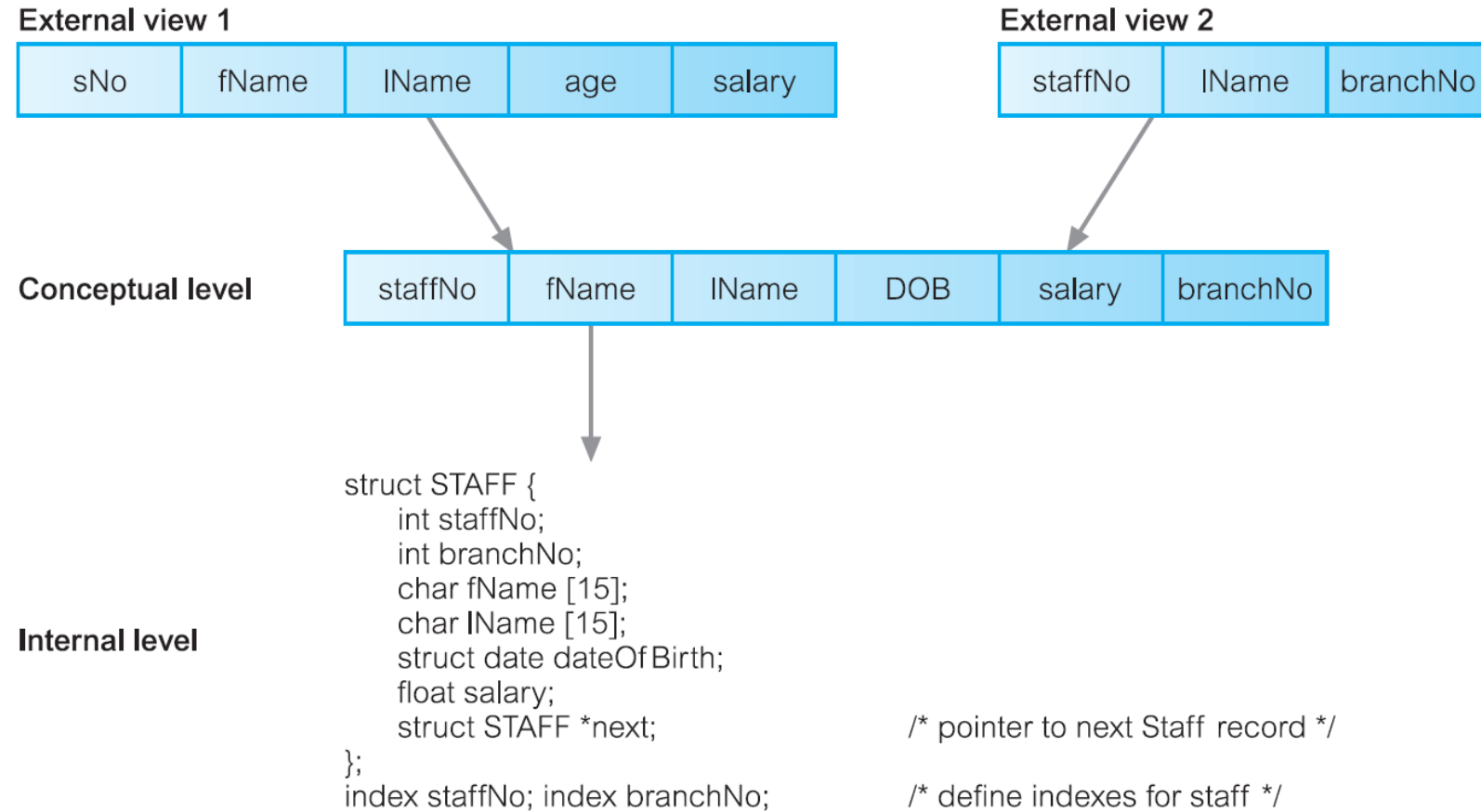
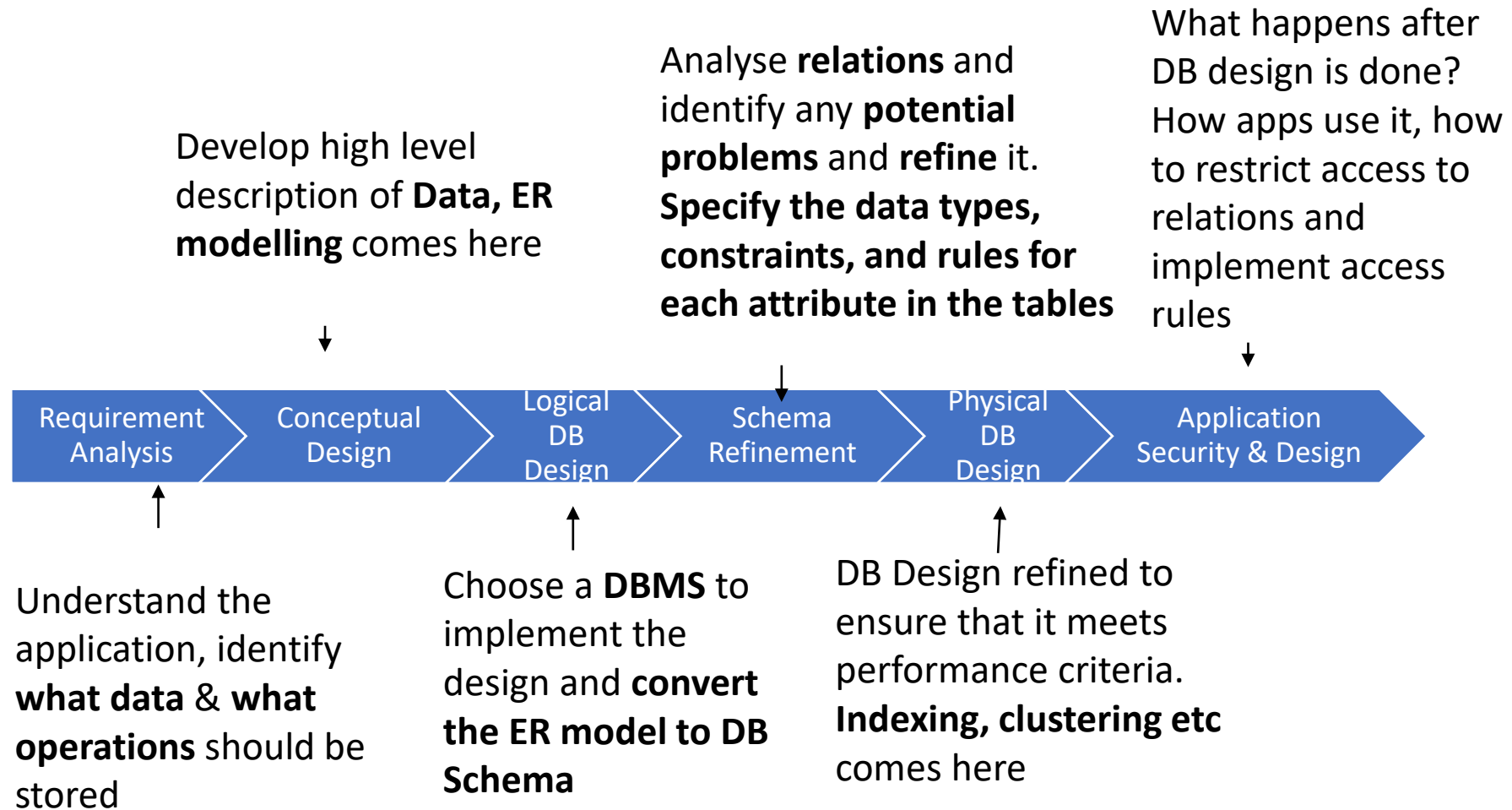


Figure 2.2 Differences between the three levels.

Learning Objectives

1. Data model- Entity-Relationship Model (ER Model)
2. Entity Relationship diagram
3. Representation of Entities, Attributes and Relationships

Database Design Process



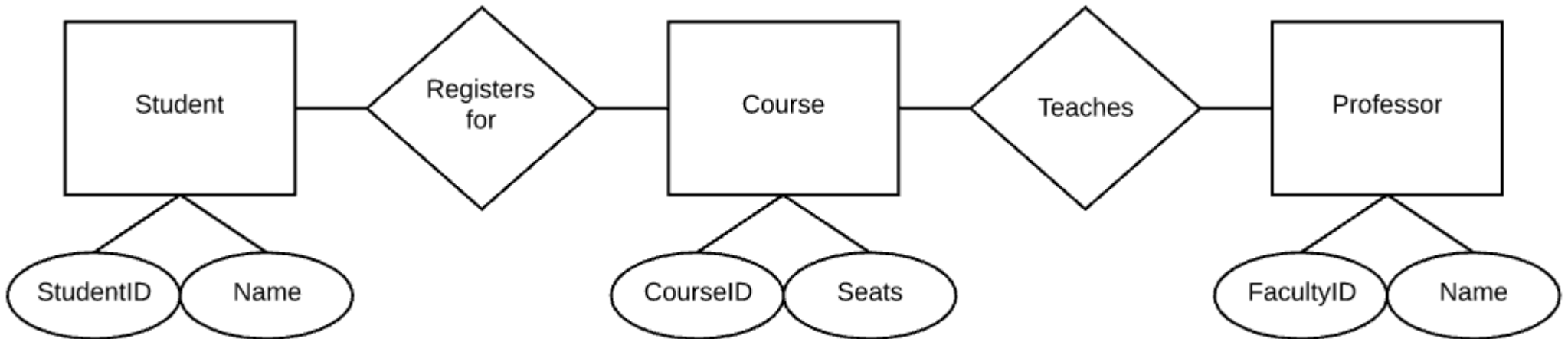
Entity-Relationship Model (ER Model):

- The ER model is a conceptual and abstract representation of the data structure within a database.
- The ER model is usually documented using natural language, text, or a combination of text and diagrams
- Example:
 - Customer entity has attributes: CustomerID (Primary Key), FirstName, LastName, Email.
 - Product entity has attributes: ProductID (Primary Key), Name, Price, and QuantityInStock.
 - The Customer entity has a FirstName attribute of data type Text, and it is required (NOT NULL).
 - A one-to-many relationship exists between Customer and Order, where each customer can place multiple orders, but each order is placed by one customer



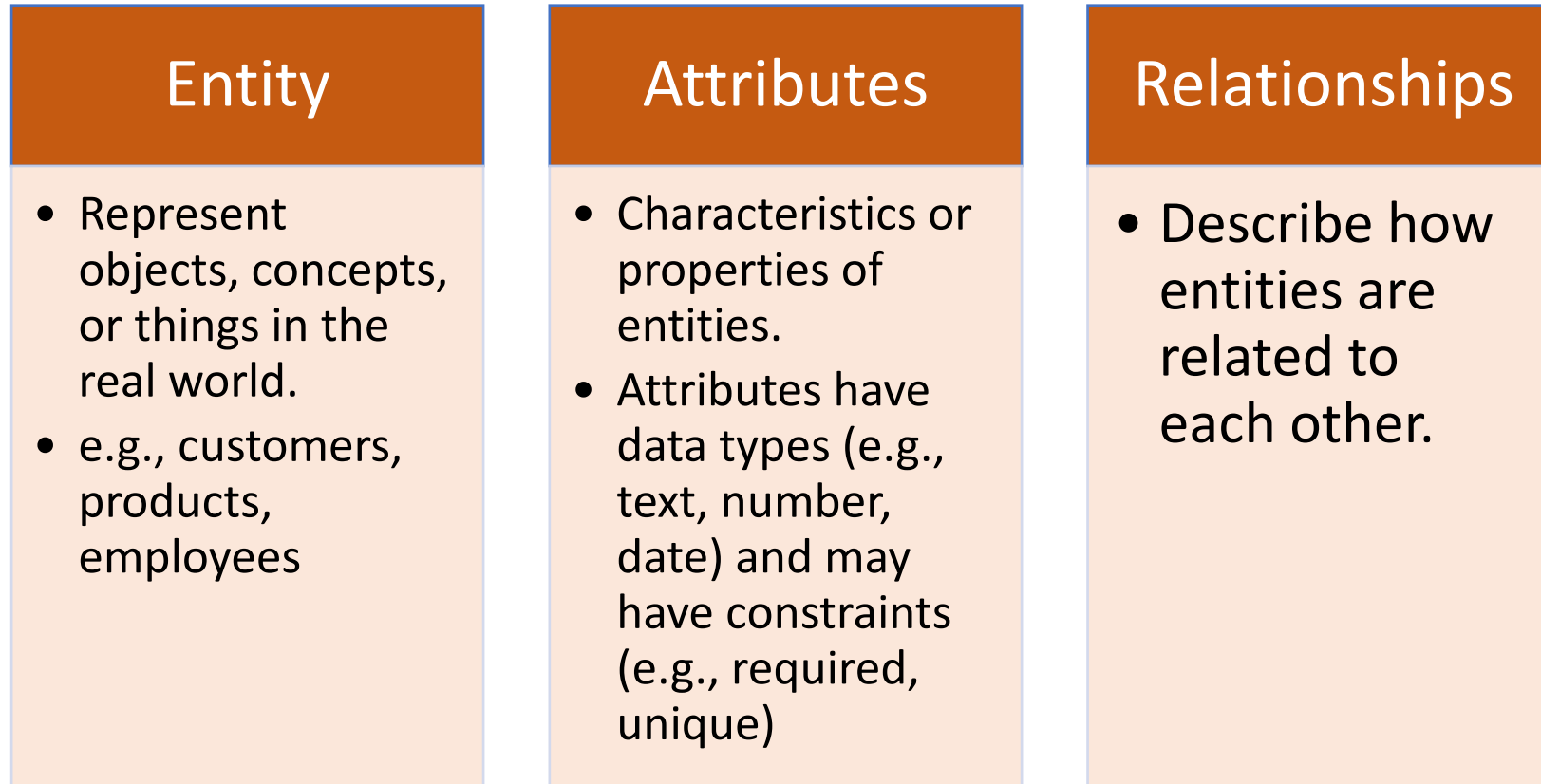
Entity-Relationship Model (ER Model):

- The ER model is usually documented using natural language, text, or a combination of text and diagrams
- Example:



Entity Relationship Diagram(ER diagram)

- **Three** principal element types:



Entity

- **Objects in real life** that are distinguishable from other objects.
- **Usually nouns** in a scenario description but not every noun becomes an entity
- E.g. If you are building a data model for a small retail store, then the potential entities may be
 - Customer
 - Sales Rep
 - Order
 - Item
 - Inventory list

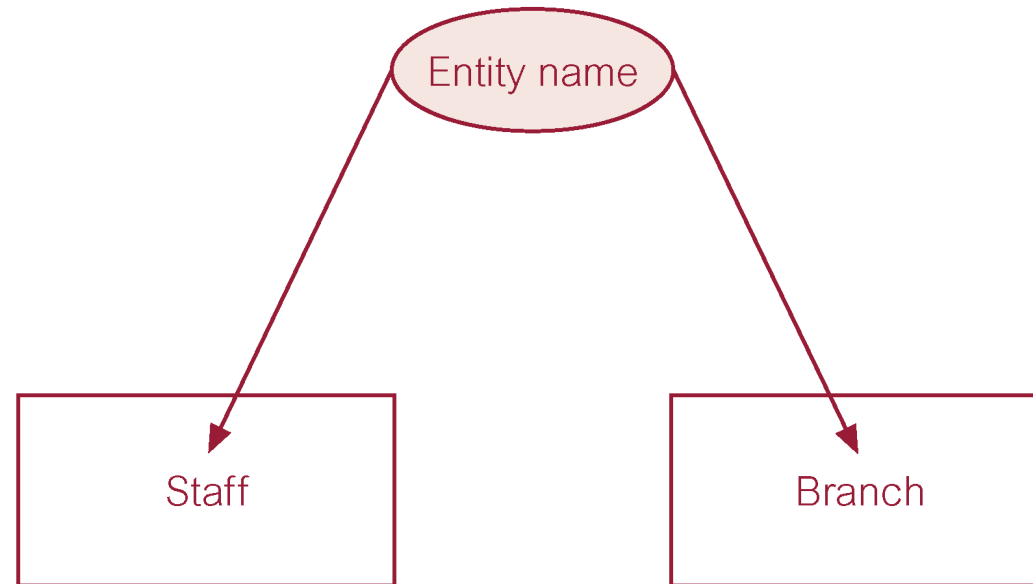
Examples of Entity Types

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

Tangible Entity

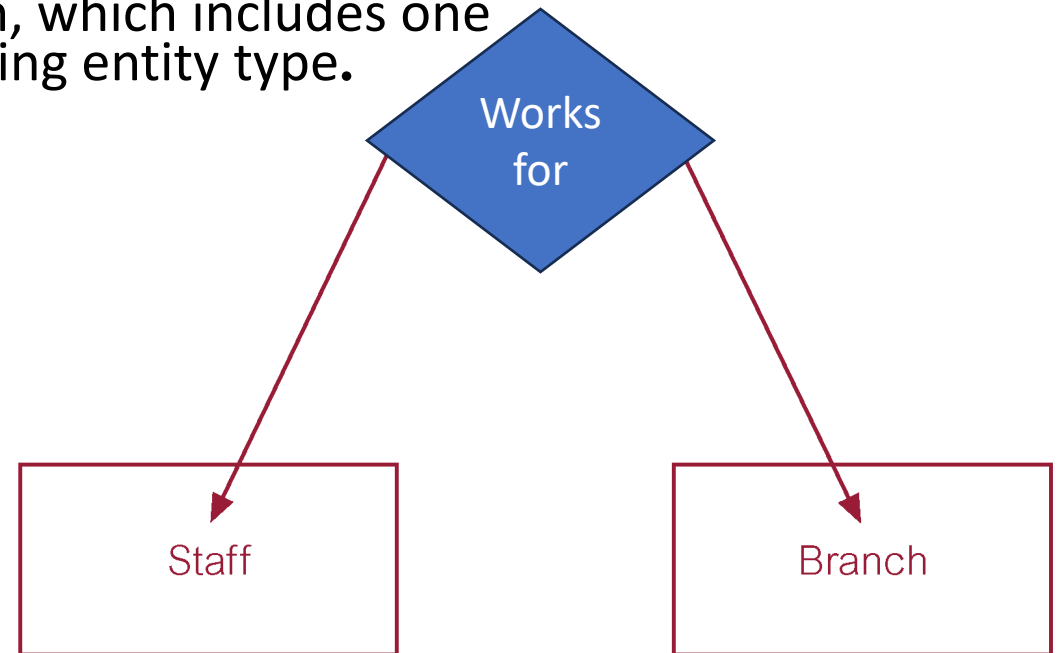
Intangible Entity

Notation of Entity



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.



Types of Entities

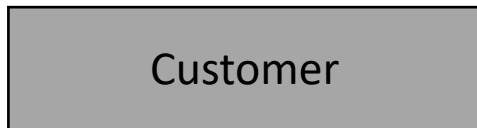
Regular Entities

- Called as Strong Entities.
- Independent Entities
- Ex: customer

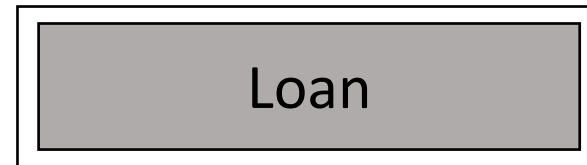
Weak Entities

- It's existence depends on strong entity
- Ex: Loan
 - If an customer Instance is deleted, then the dependants too are deleted. They do not exist without the customer.

Notation for a Regular Entity



Notation for a Weak Entity



Strong and Weak entity types

Strong Entity Type

Customer

Parent

Building

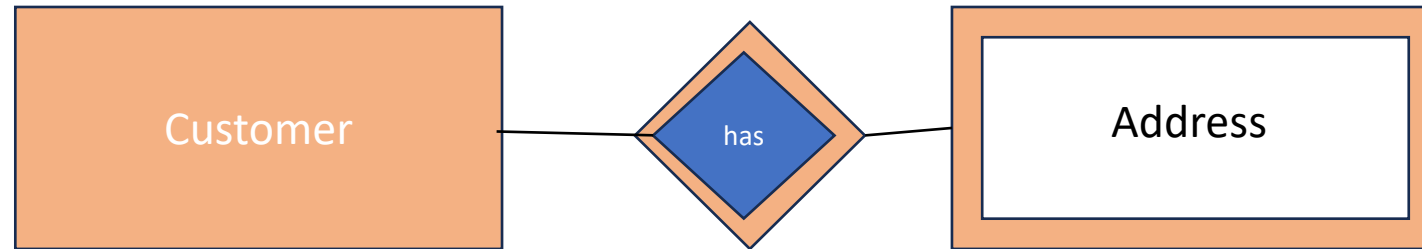
Weak Entity Type

Address

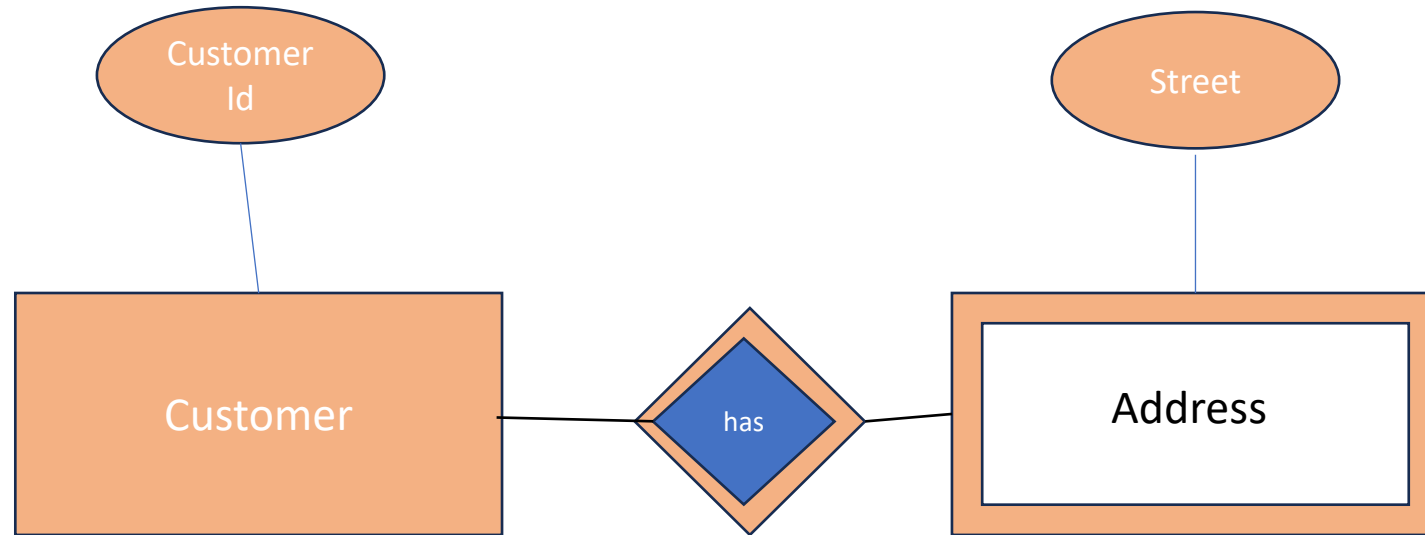
Children

Room

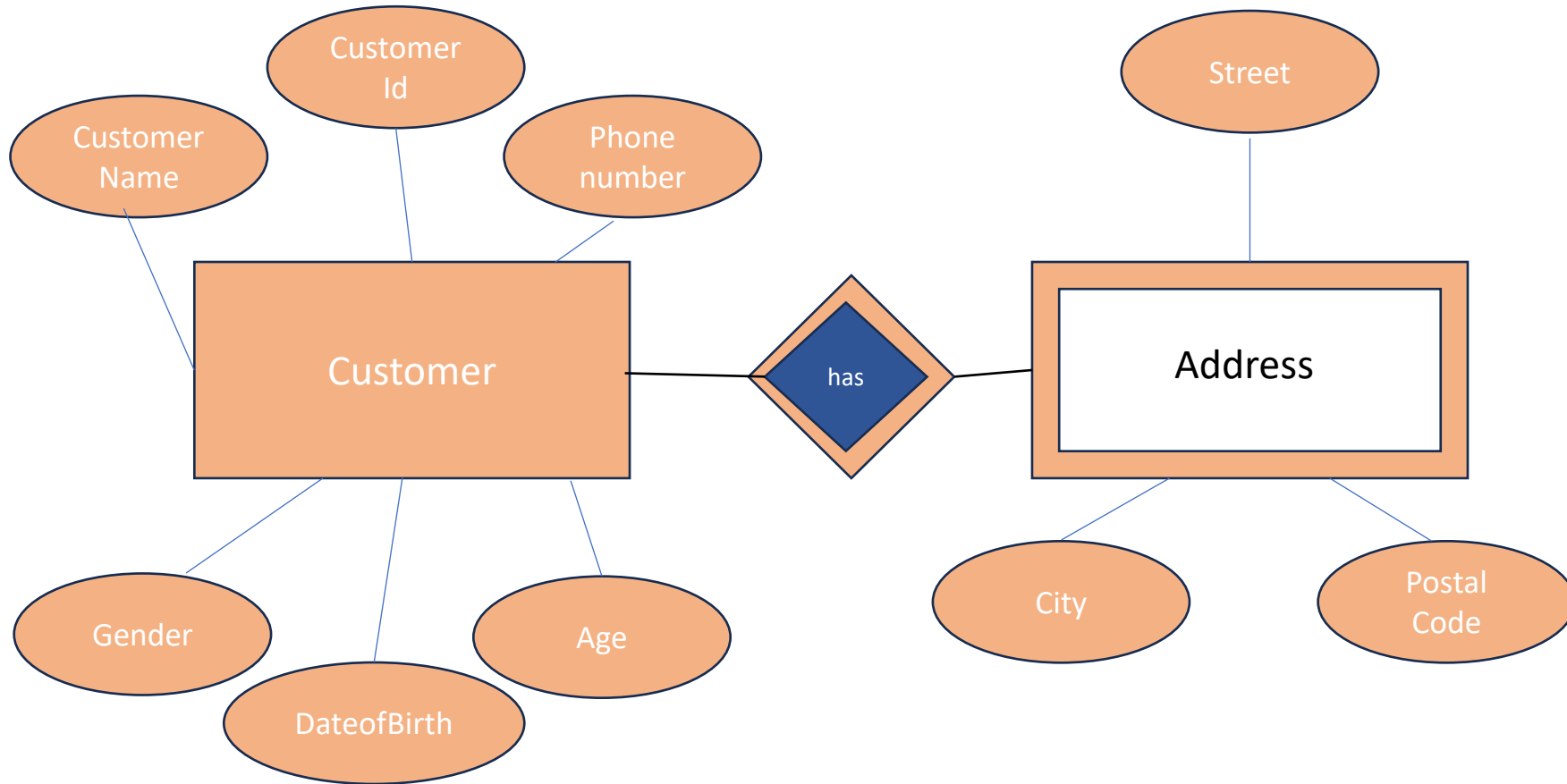
Relationship between Strong and Weak entity types



Attributes



Relationship between Strong and Weak entity types



Attribute Types

- Each entity has attributes – **properties that describe it.**
- A particular entity will have values for its attributes.
- Ex: an employee entity may have name, age, home address, home phone etc as their attributes.
- **Keys:** Unique value in the context.

Attribute Types			
Simple Vs Composite	Single Vs Multivalued	Stored Vs Derived	Null Vs Complex

Attributes-Type 1

- **Simple Attribute**

- Attribute composed of a single component with an independent existence.

- **Composite Attribute**

- Attribute composed of multiple components, each with an independent existence.

Entity:

Person(Firstname,Lastname,DateOfBirth,Gender,Address)

Simple Attributes:

- 1.First Name
- 2.Last Name
- 3.DateOfBirth
- 4.Gender

Composite Attribute:

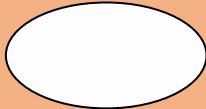
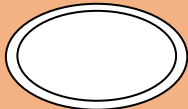
5. Address

- Street address
- State
- Postal Code

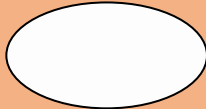
- Street address
 - House Number
 - Street Name

sub attributes

Attribute Types - 2

Type	Description	Example
Single Valued Attribute 	An attribute that holds a single value for each occurrence of an entity type.	Employee entity: Salary → can be only one value. Cannot draw more than one salary.
Multi Valued Attribute 	An attribute that holds multiple values for each occurrence of an entity type.	Employee entity: PhoneNo → A person can have one or more mobile phone number, landline etc.

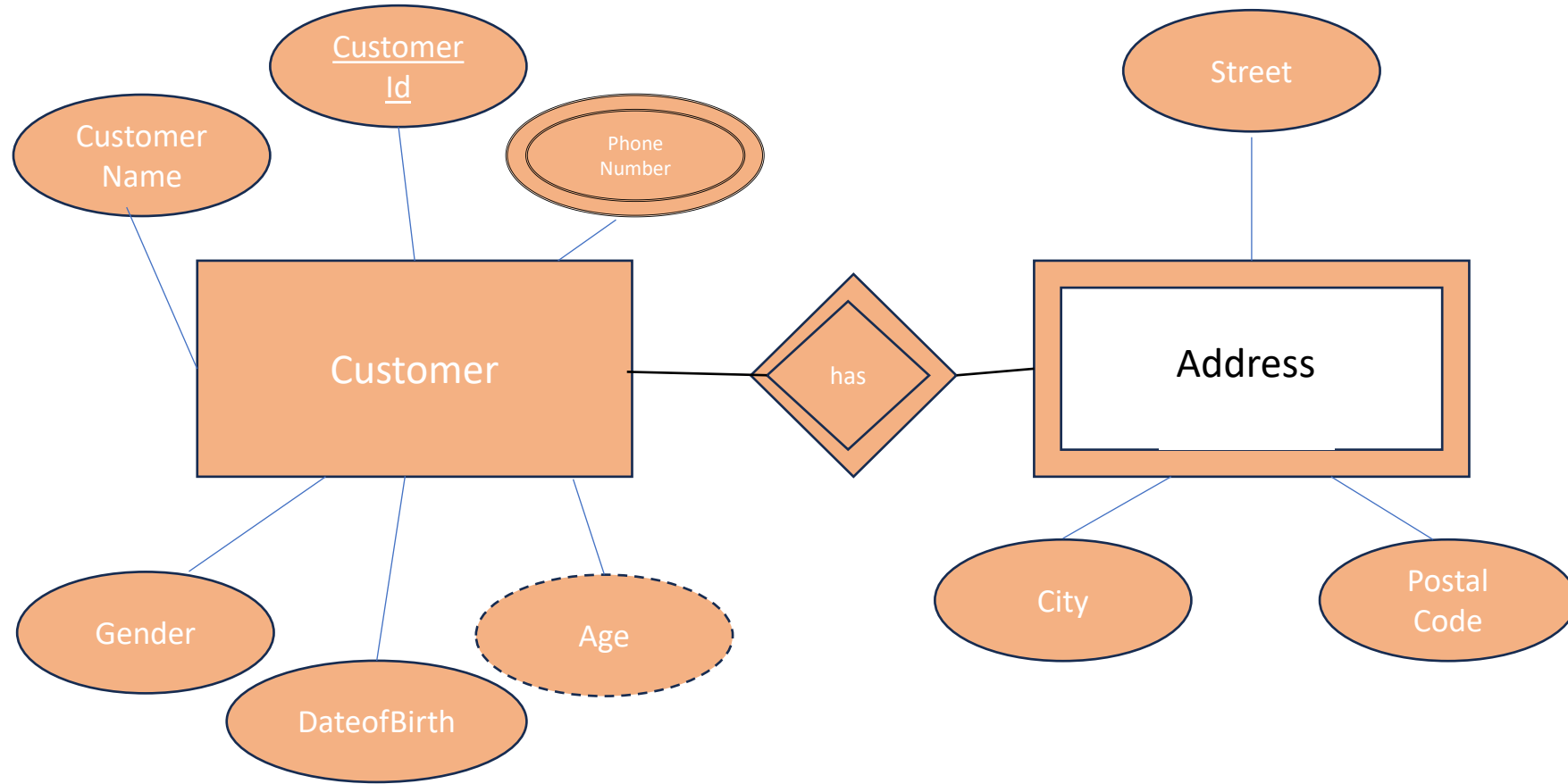
Attribute Types - 3

Type	Description	Example
Stored Attribute 	An attribute that holds a value provided as is.	Employee entity: Name → can be set by the user as used as is.
Derived Attribute 	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.	Employee entity: Age → can be derived from dateOfBirth and currentDate.

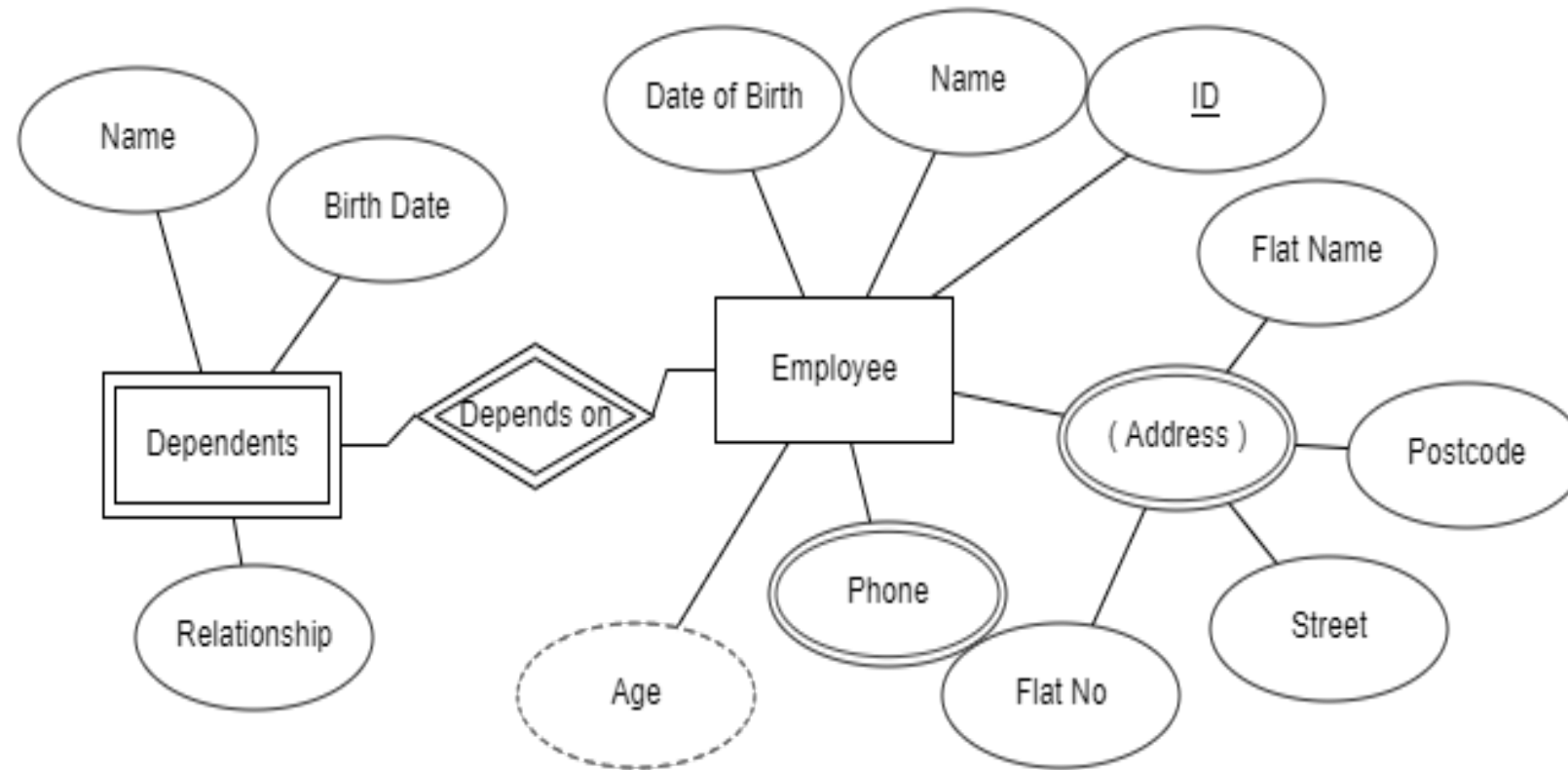
Attribute Types - 4

Type	Description	Example
NULL Attribute	An attribute that means either the value is unknown or inapplicable. It does not mean zero or blank.	Employee entity: Commission → can be NULL
Complex Attribute 	A complex attribute is a data attribute that can be divided into smaller, more elementary components or parts.	Employee entity: A person can have many phone numbers, many e-mail addresses, home addresses etc. So if there is an attribute in the name of 'Contact detail', it can be a complex attribute.

Relationship between Strong and Weak entity types



Entity & Attributes



Exercise- ERD

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

Exercise- Possible Solution

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

Entities:

1.Customer:

Attributes: CustomerID (Primary Key), First Name, Last Name, Address, Phone Number, Email, etc.

2.Car:

Attributes: CarID (Primary Key), Make, Model, Year, License Plate, VIN (Vehicle Identification Number), etc.

3.Accident:

Attributes: ReportId (Primary Key), Date, Location, Description, Damage Amount, etc.

Exercise- Possible Solution

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

Relationships:

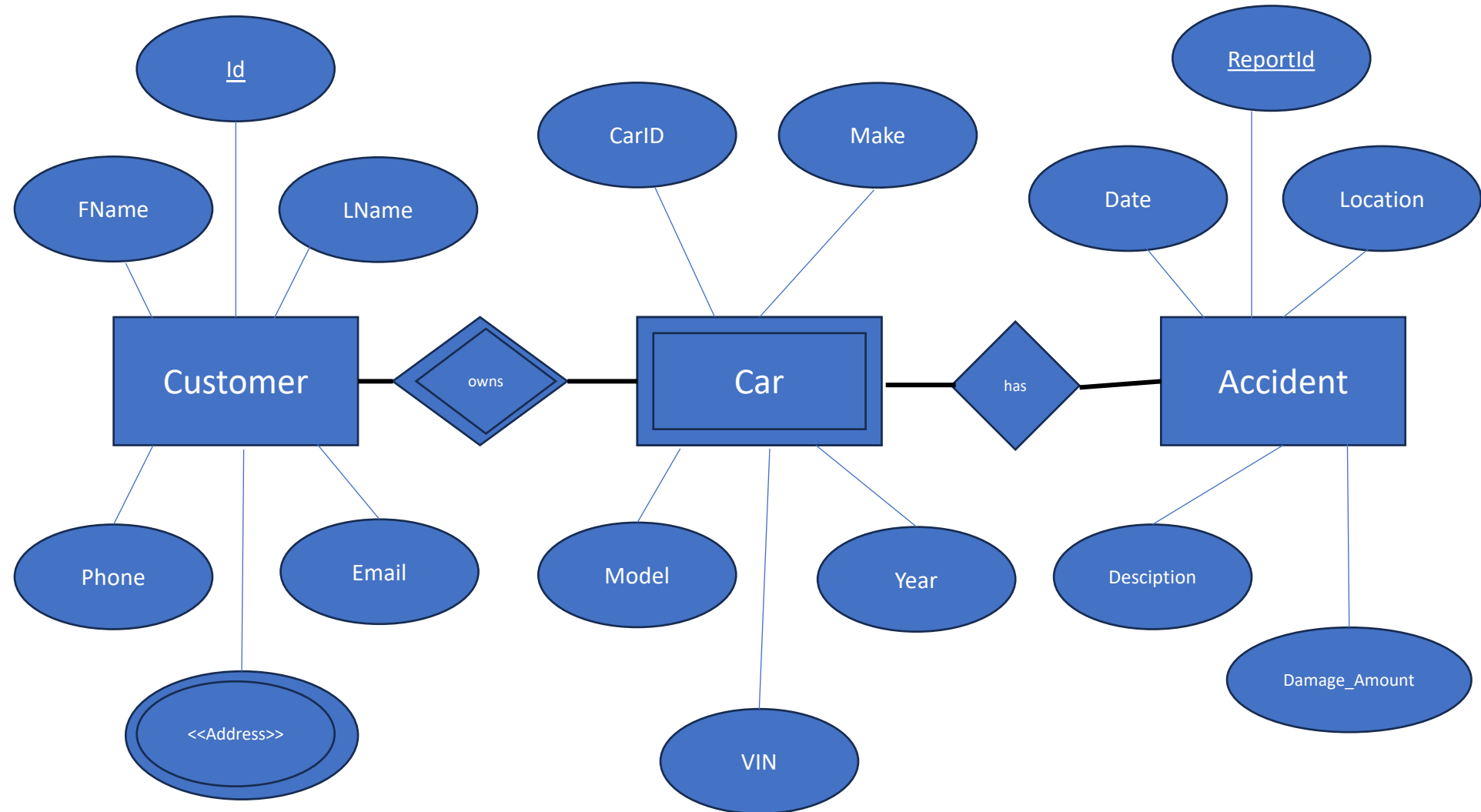
1.Owns:

1. Relationship between Customer and Car.
2. One customer can own one or more cars.
3. One car is owned by one and only one customer.
4. Cardinality: One (Customer) to Many (Car).

2. Has:

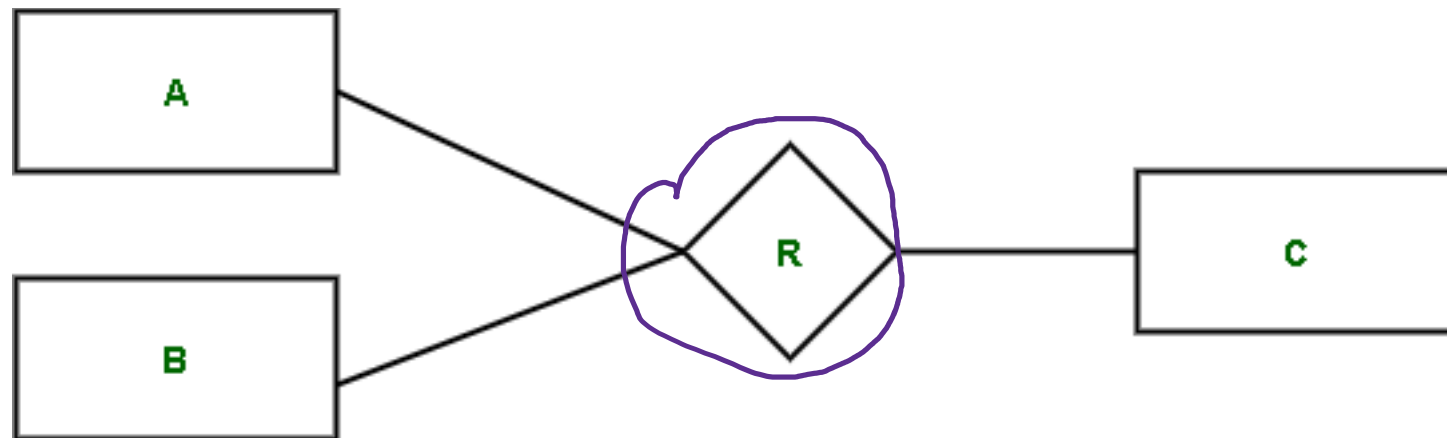
1. Relationship between Car and Accident.
2. One car can be associated with zero to many accidents.
3. One accident is associated with one and only one car.
4. Cardinality: Zero or Many (Car) to One (Accident).

Exercise- Possible Solution



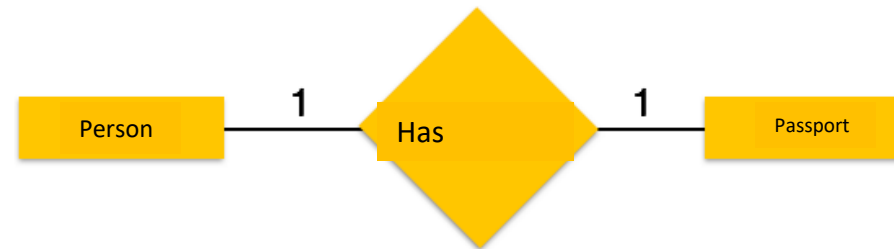
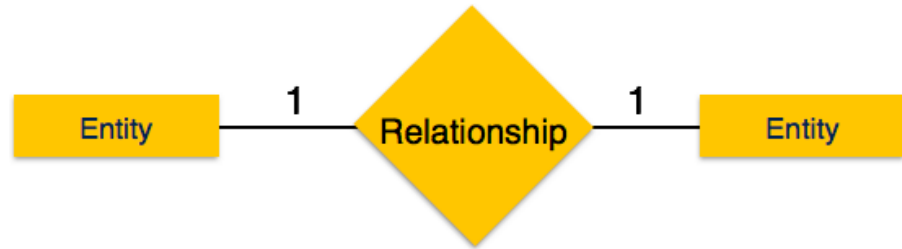
Relationships

Diagrammatic Notation for Relationships



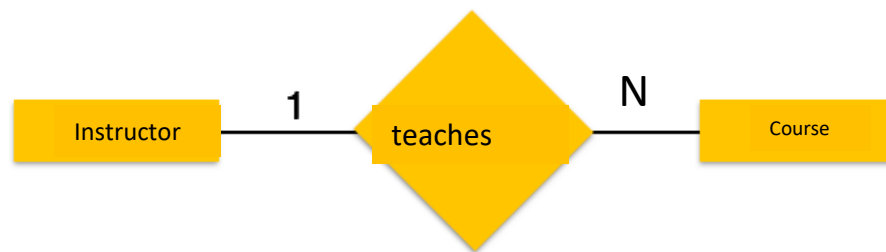
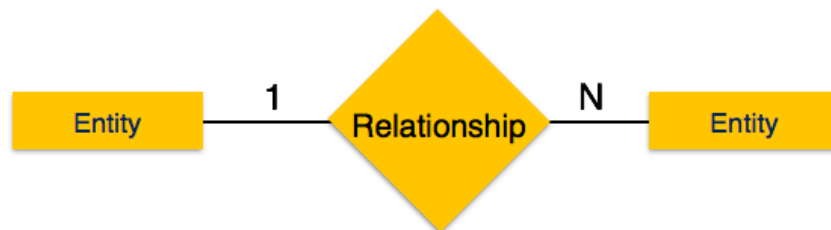
Binary Relationship and Cardinality

One-to-one



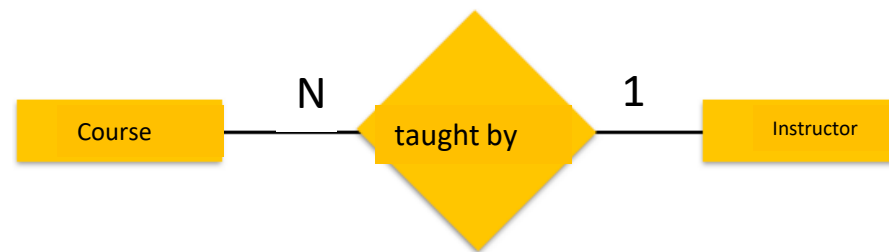
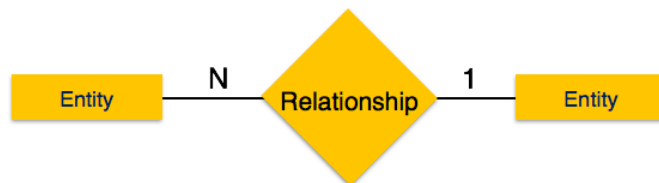
Binary Relationship and Cardinality

One-to-many



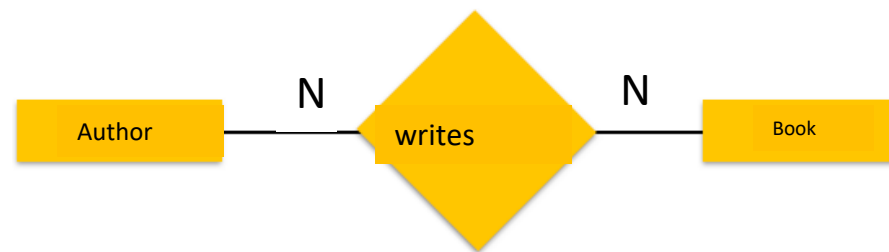
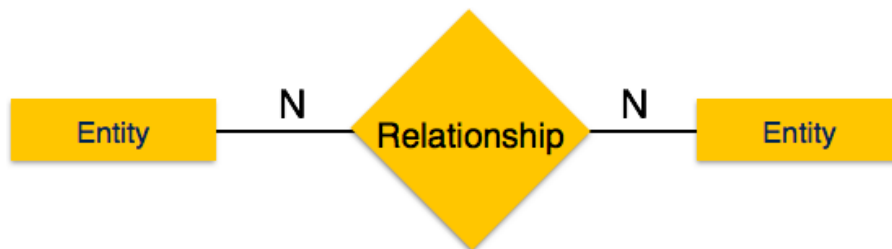
Binary Relationship and Cardinality

Many-to-one

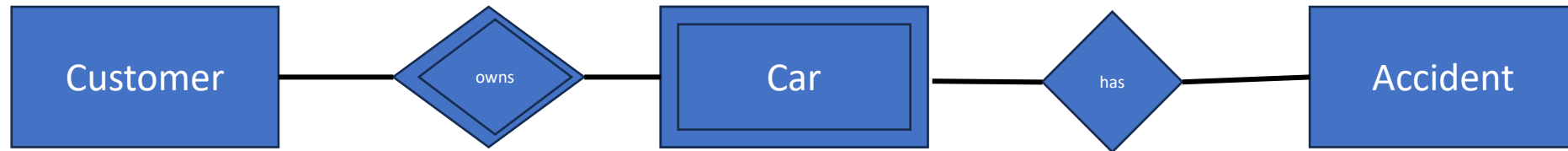


Binary Relationship and Cardinality

Many-to-many

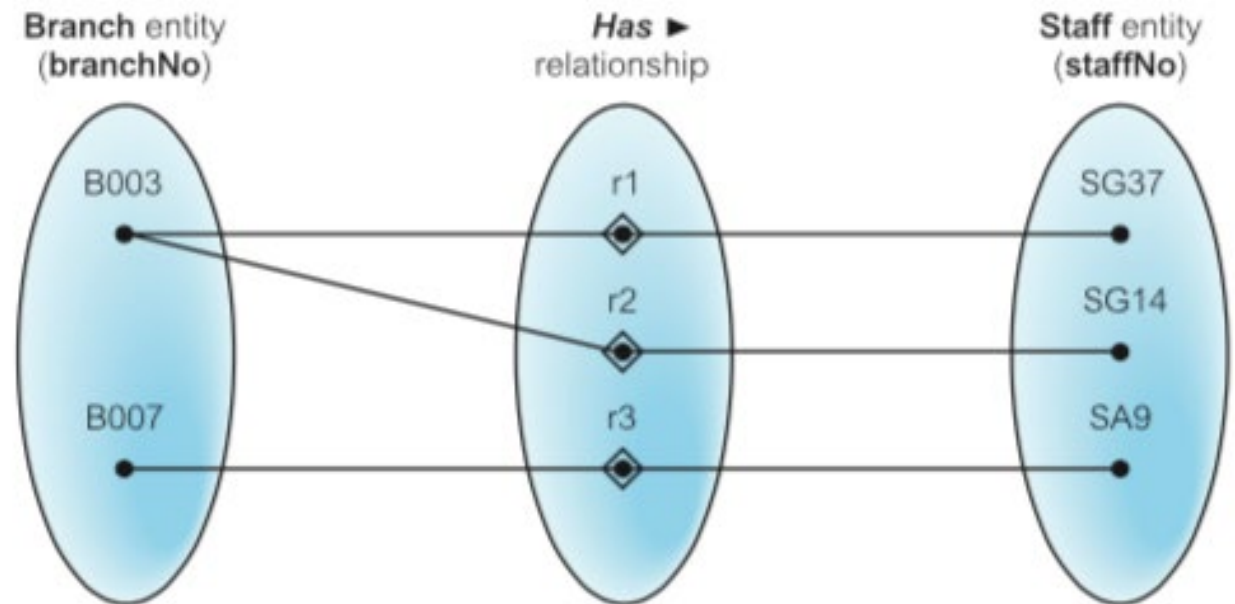


Relationship



Relationship & Relationship Sets

- Relationship → An **association among the entities**.
- Entities involved in a relationship are said to be the **participants** in that relationship.



Relationship Types

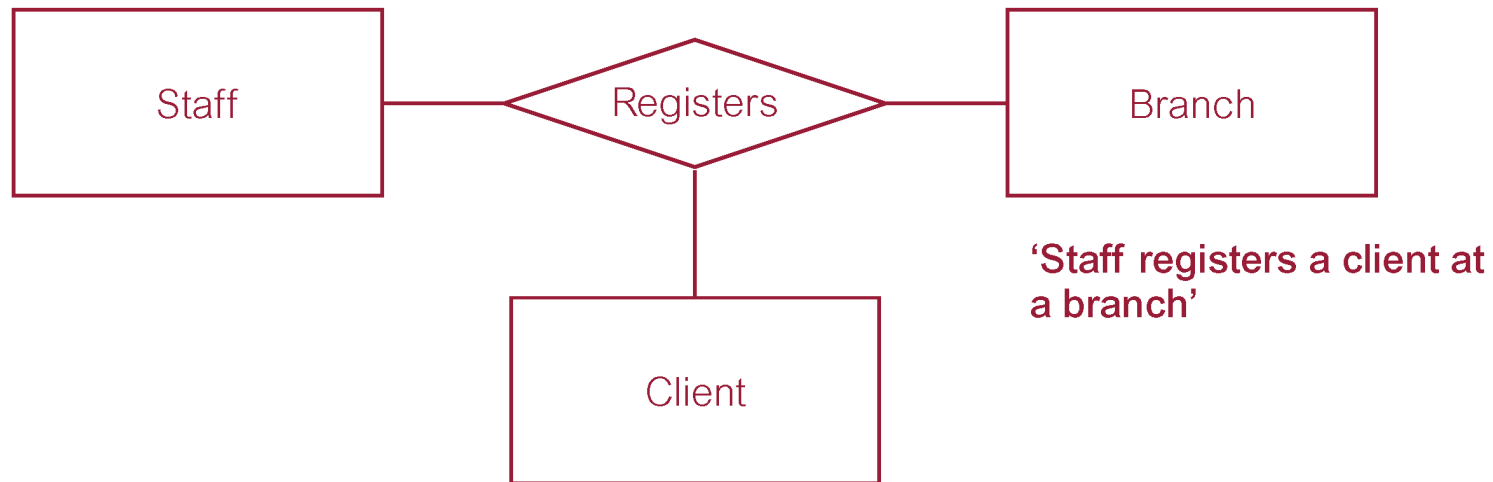
- Degree of a Relationship
 - Number of participating entities in relationship.
- Relationship of degree :
 - two is binary
 - three is ternary
 - four is quaternary.

Binary relationship called *POwns*

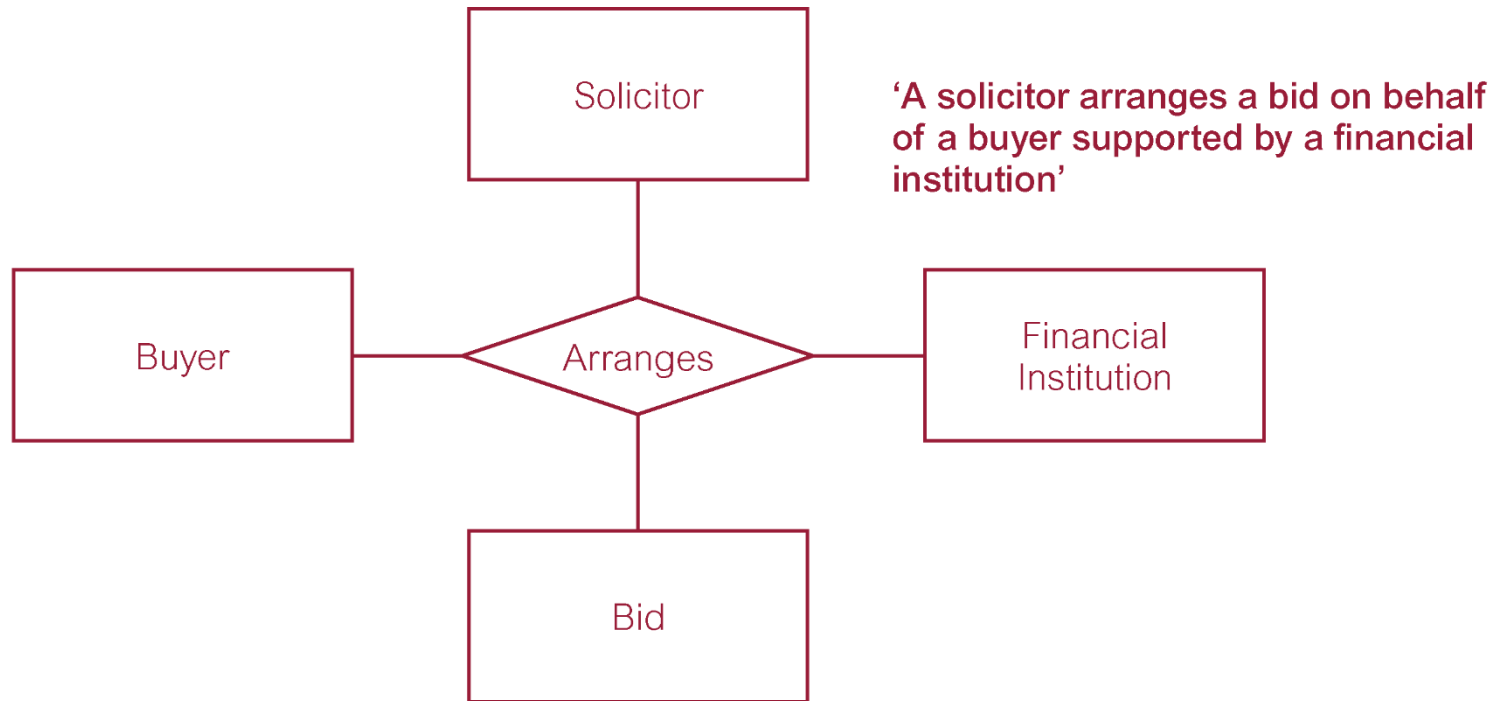
‘Private owner owns property for rent’



Ternary relationship called *Registers*



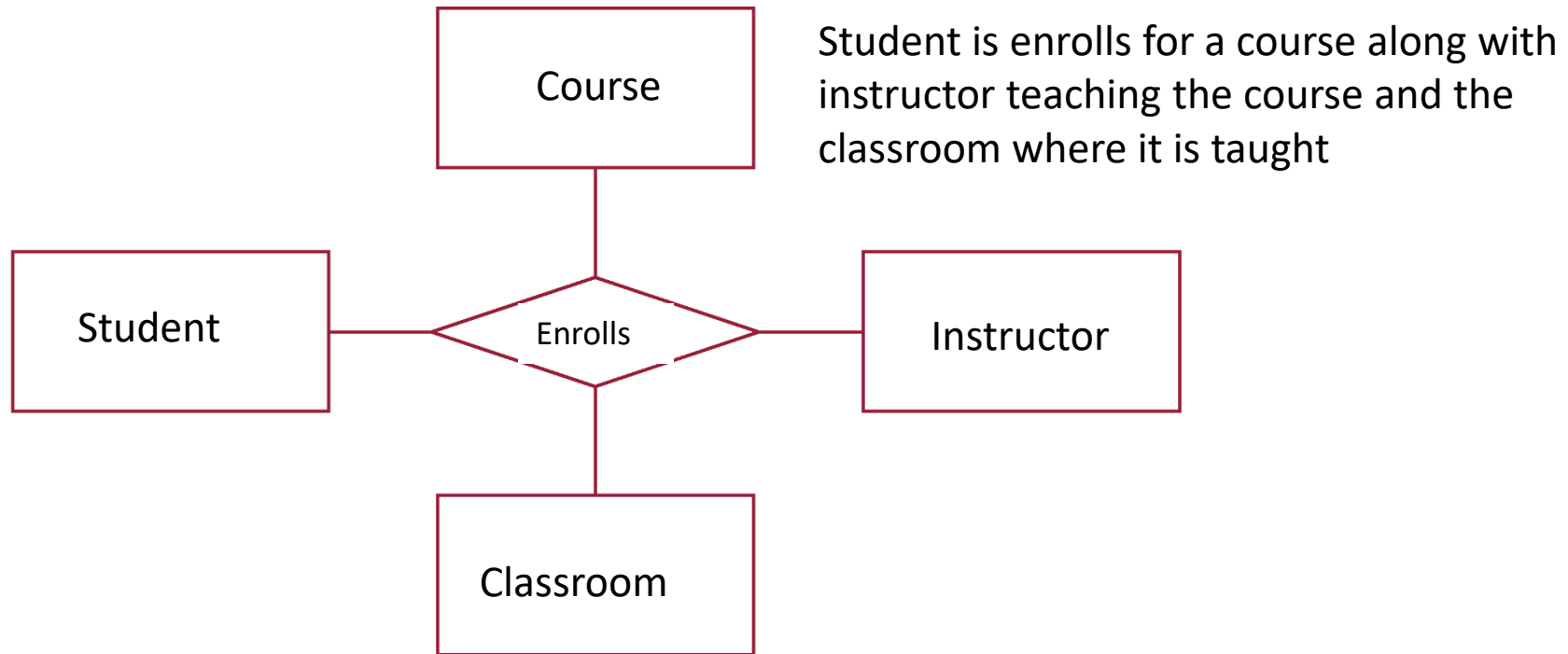
Quaternary relationship called *Arranges*



Try this

Student enrolls for a course along with
instructor teaching the course and the
classroom where it is taught

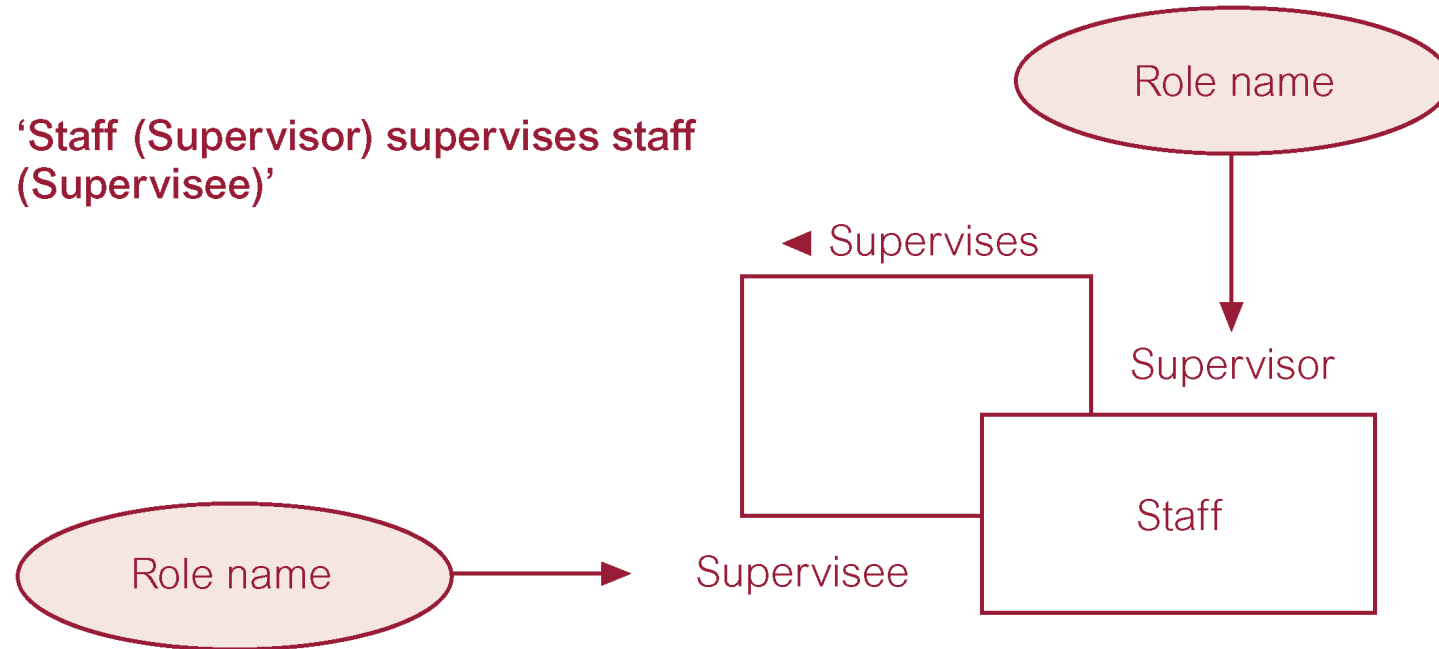
Try this



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.

Recursive relationship called *Supervises* with role names



Recursive relationship – How will the table(s) look like?

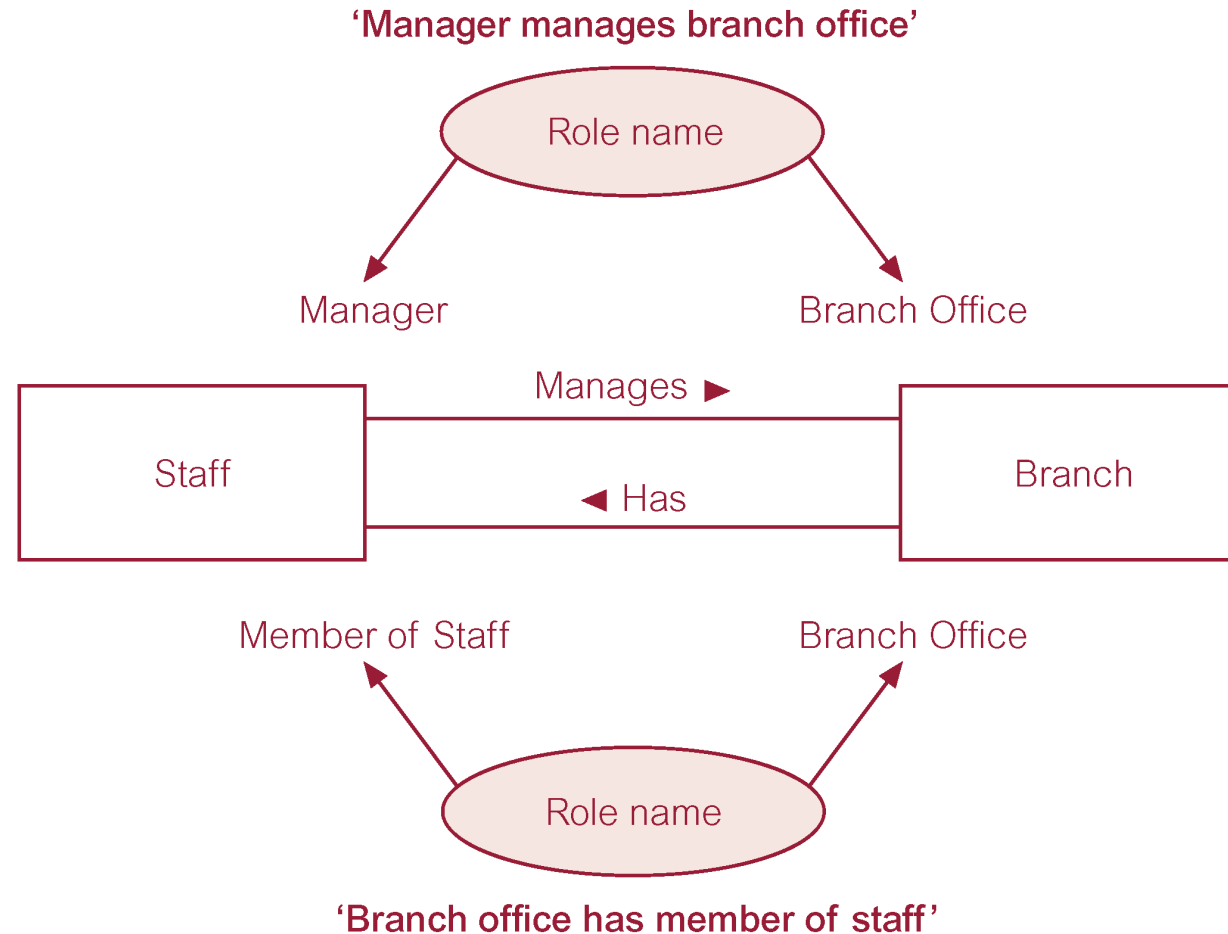
Staff Table Data:

StaffID	Name	Position	SupervisorID
1	John Doe	Manager	NULL
2	Alice Smith	Team Lead	1
3	Bob Johnson	Staff.	1
4	Eve Davis	Staff.	2

Supervisor Table Data:

RelationshipID	StaffID.	SupervisorID	StartDate	EndDate
1	2	1	2023-01-01	NULL
2	3	1	2022-12-15	NULL
3	4	2	2023-02-01	NULL

Entities associated through two distinct relationships with role names



Recap: Attributes

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

Recap: 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.

Recap: 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.

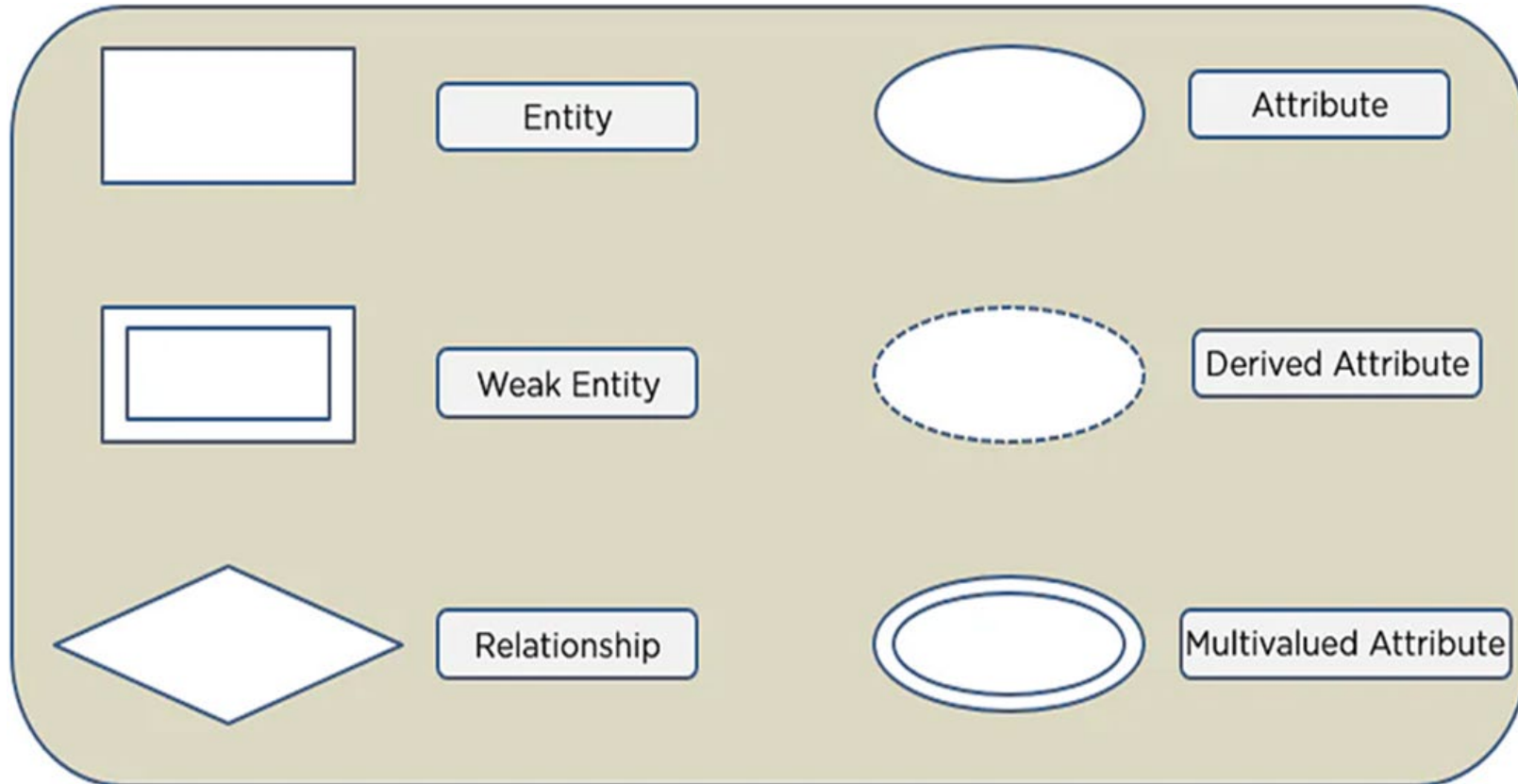
Recap: Attributes

- **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.

Recap: 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.

Recap:E-R Diagram Notations



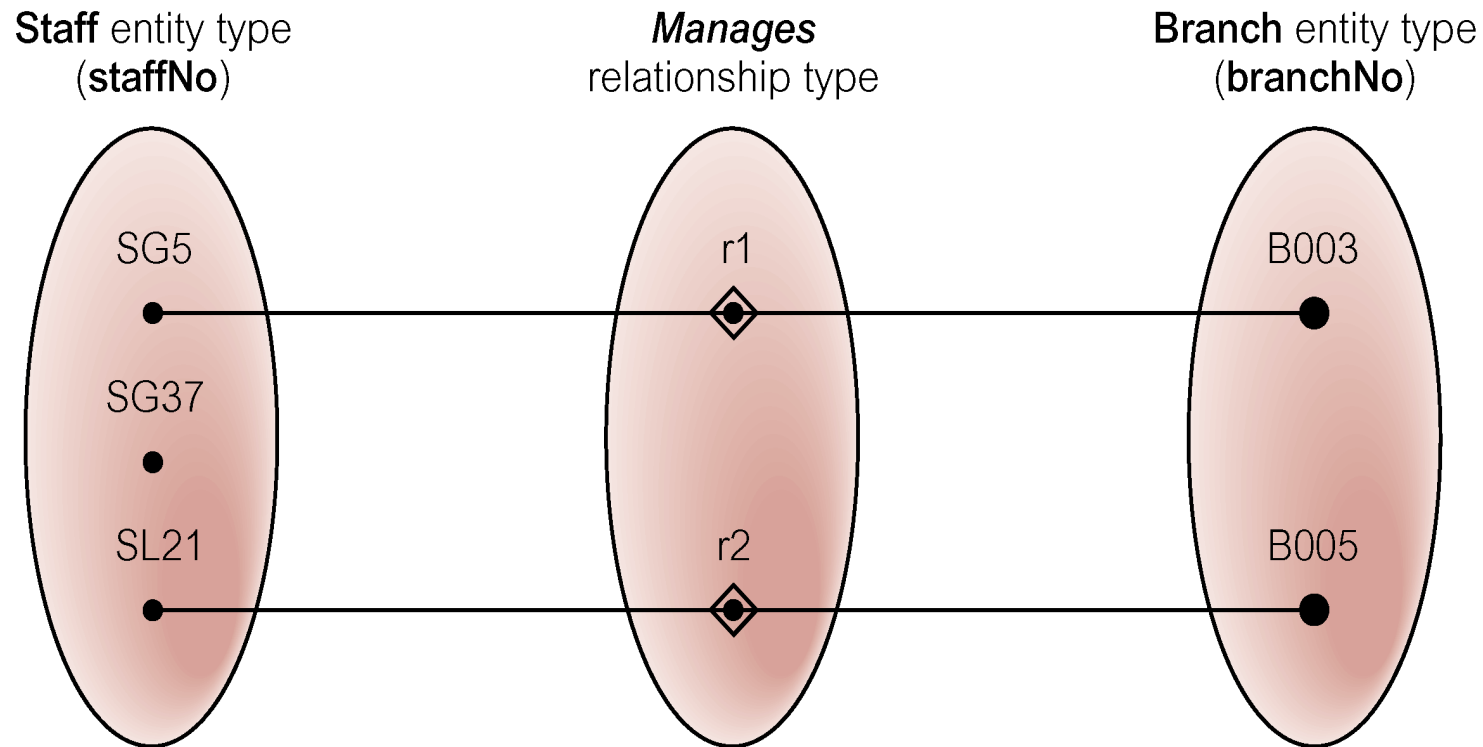
Structural Constraints

- 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.

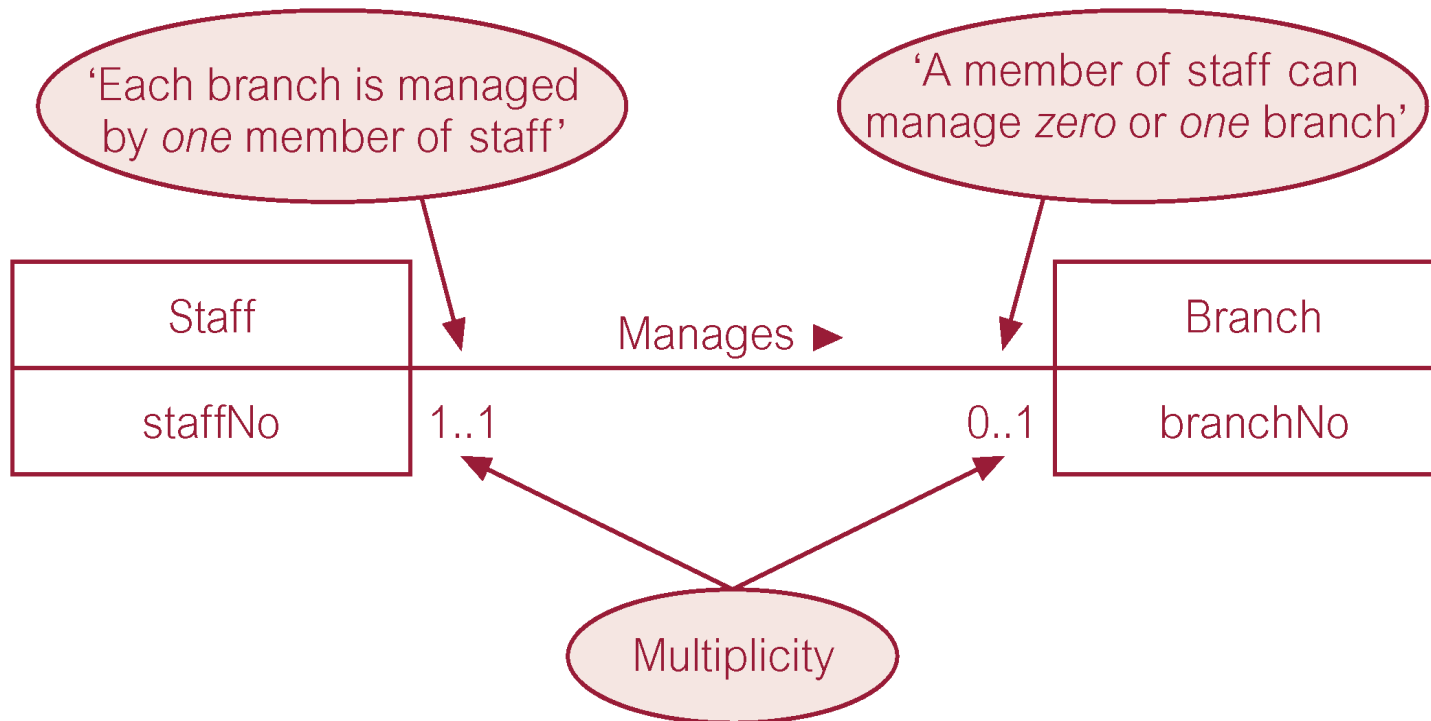
Structural Constraints

- 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 (*:*)

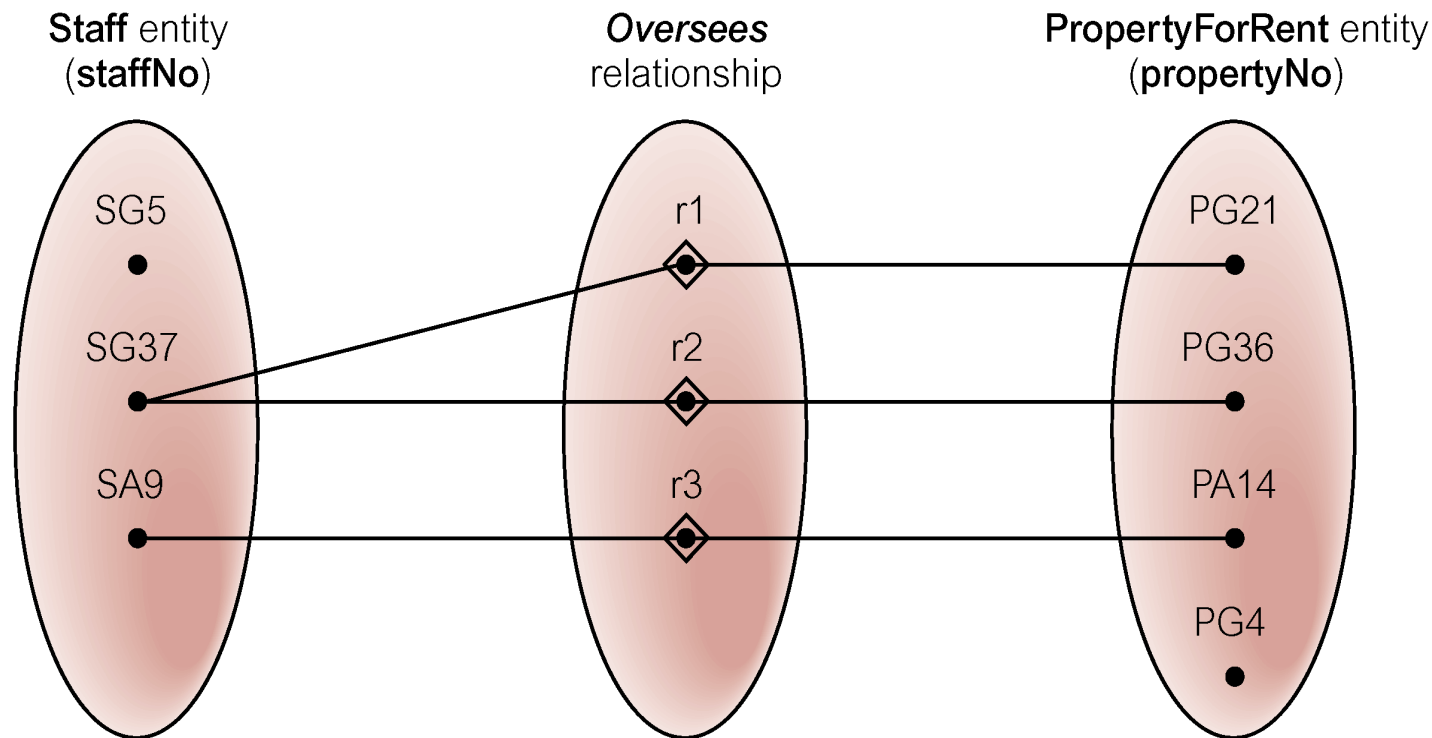
Semantic net of Staff *Manages* Branch relationship type



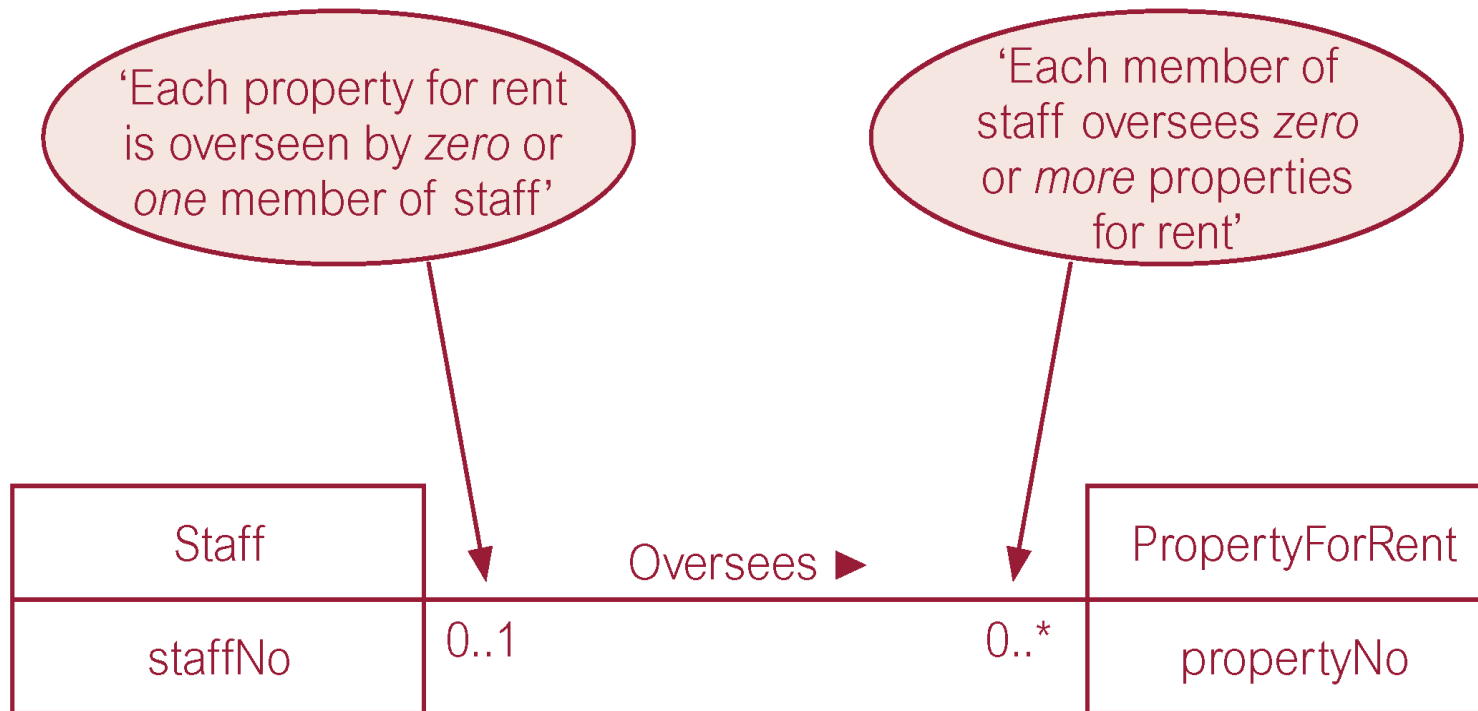
Multiplicity of Staff *Manages* Branch (1:1) relationship



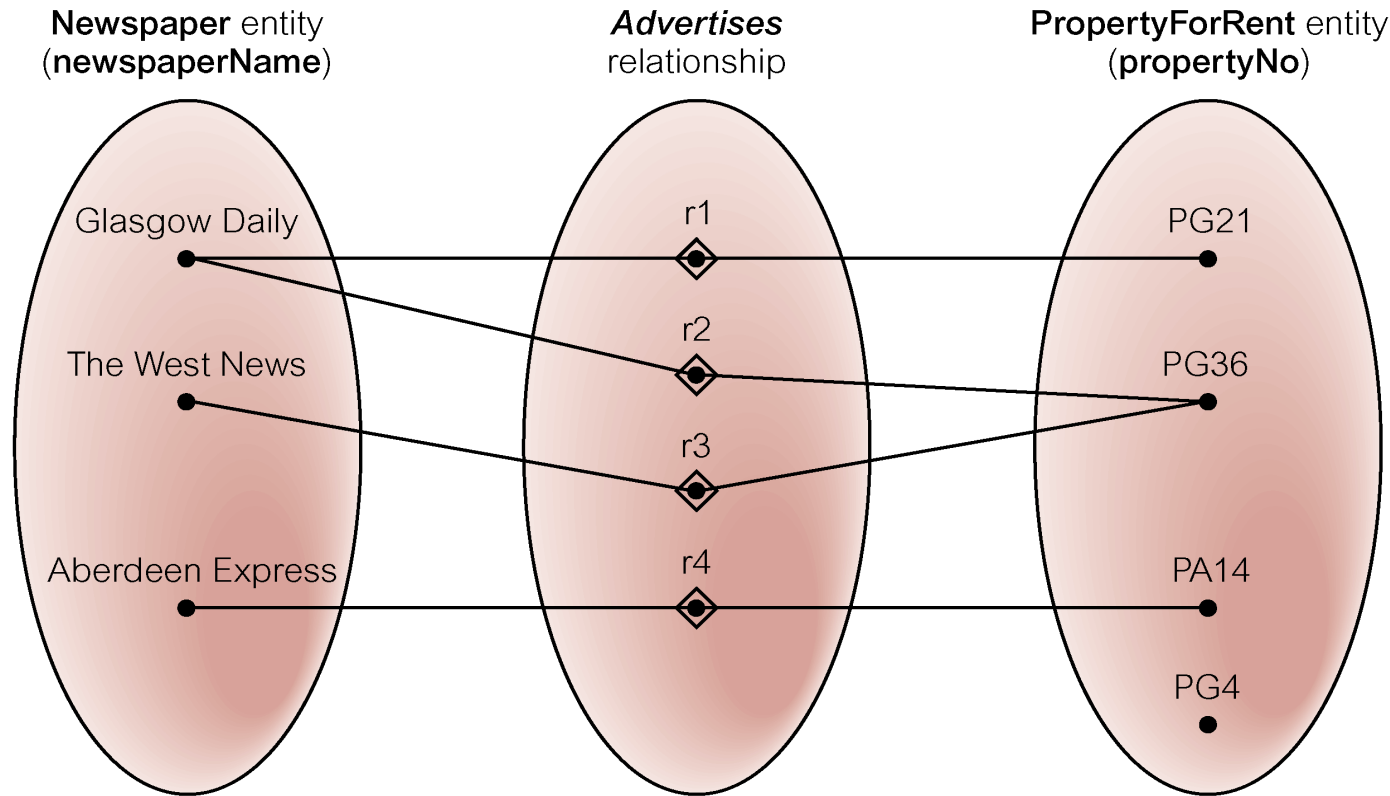
Semantic net of Staff *Oversees* PropertyForRent relationship type



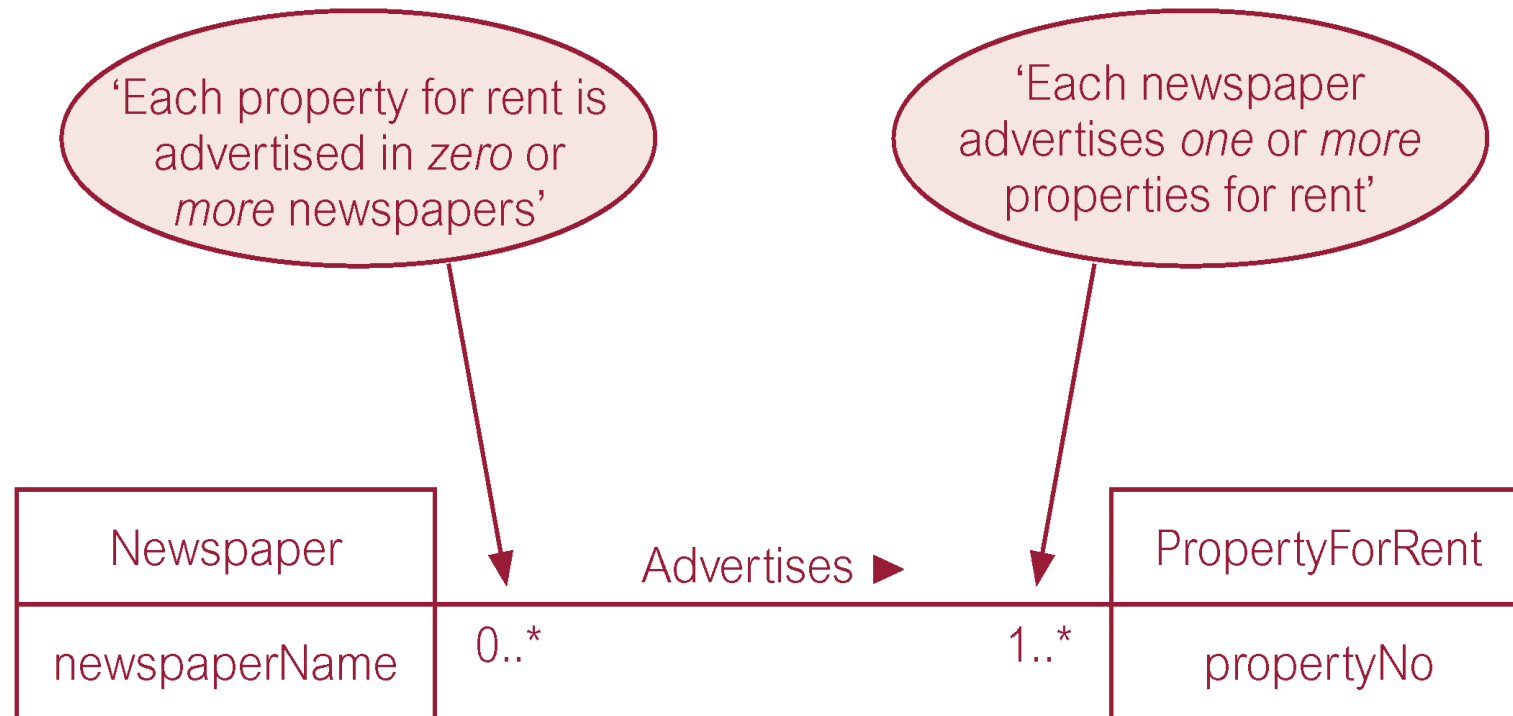
Multiplicity of Staff *Oversees* PropertyForRent (1:*)relationship type



Semantic net of Newspaper *Advertises* PropertyForRent relationship type



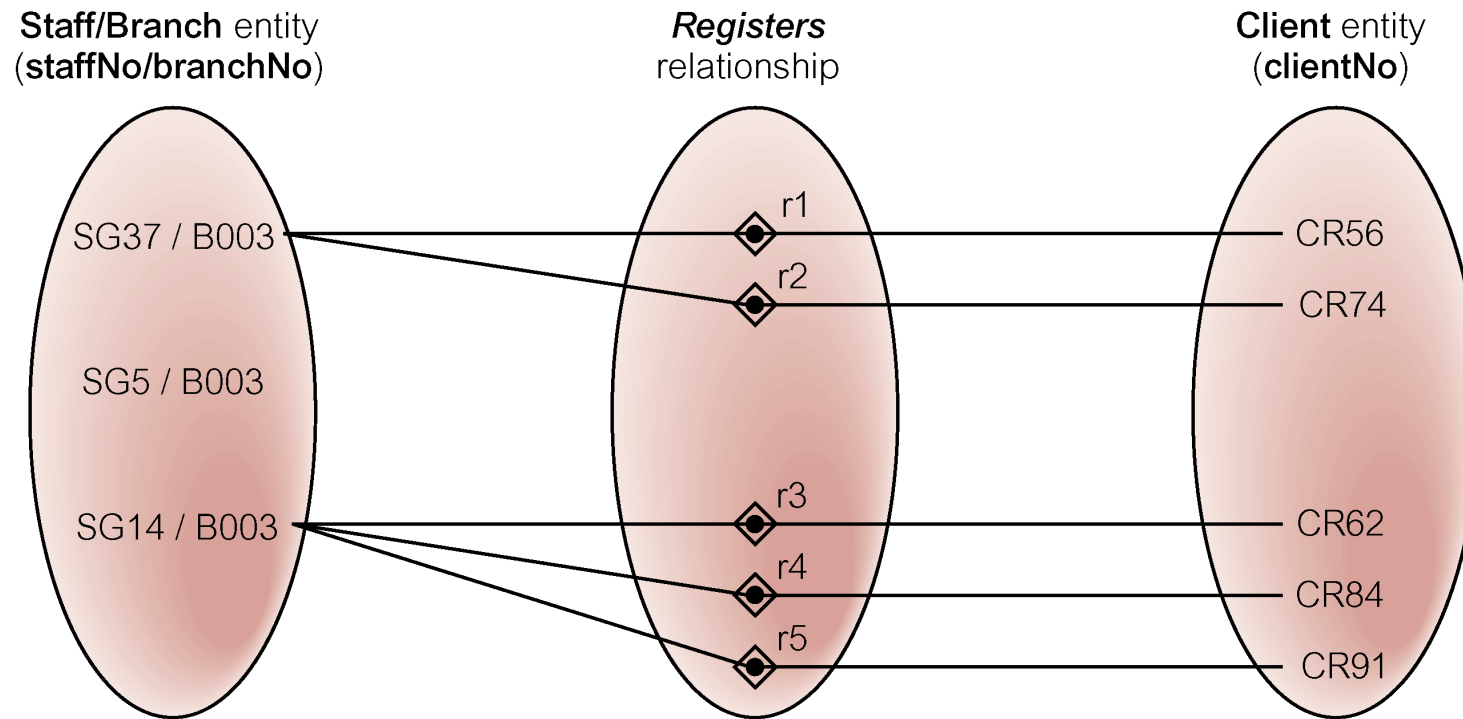
Multiplicity of Newspaper *Advertises* PropertyForRent (*:*) relationship



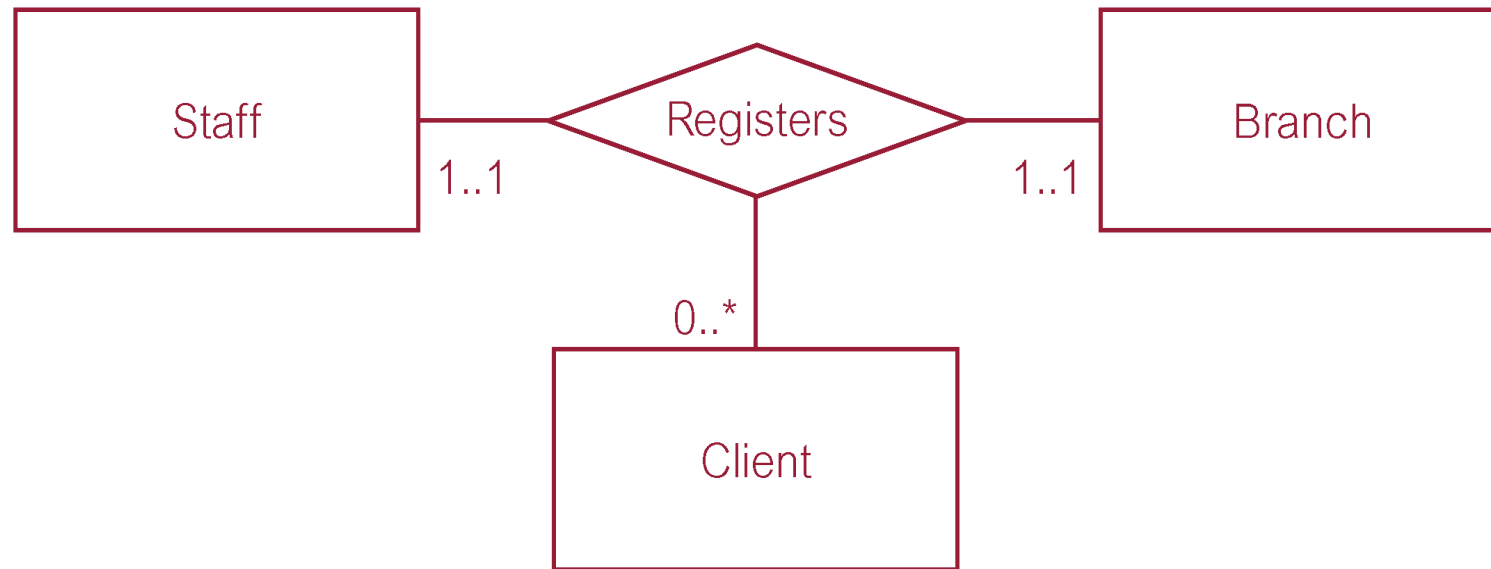
Structural Constraints

- **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.

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



Multiplicity of ternary *Registers* relationship



Summary of 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

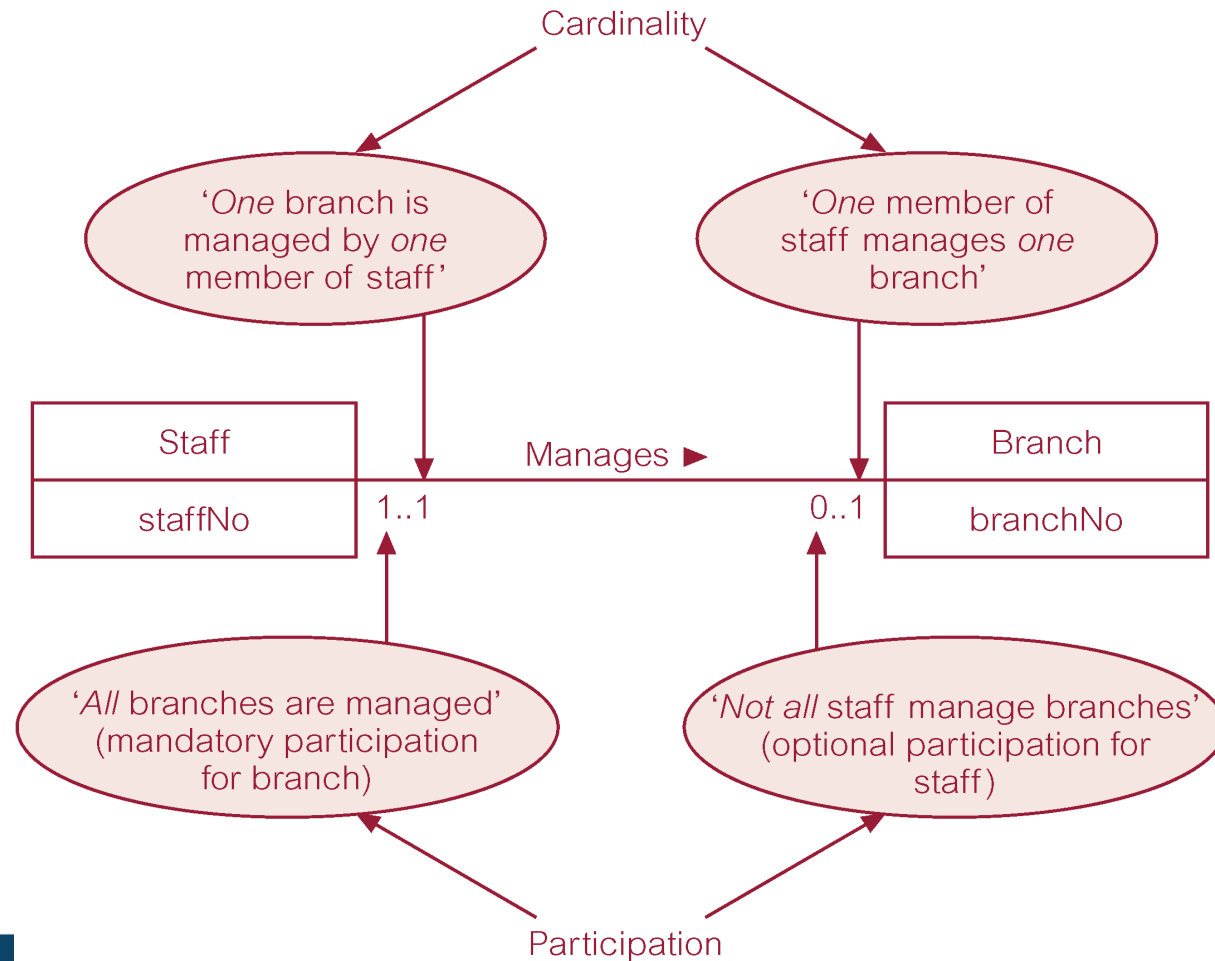
Structural Constraints

- Multiplicity is made up of two types of restrictions on relationships: *cardinality* and *participation*.

Structural Constraints

- **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.

Multiplicity as cardinality and participation constraints



Summary

a. ERD Elements

i. Entities

- i. Weak,
- ii. Strong

ii. Attributes

- i. Simple vs Composite
- ii. Single Value vs Multi value
- iii. Stored vs Derived
- iv. Null vs Complex

iii. Relationships - Degree

- i. 1-ary, Bi-nary, N-ary,

Next session.....

Key and it's type