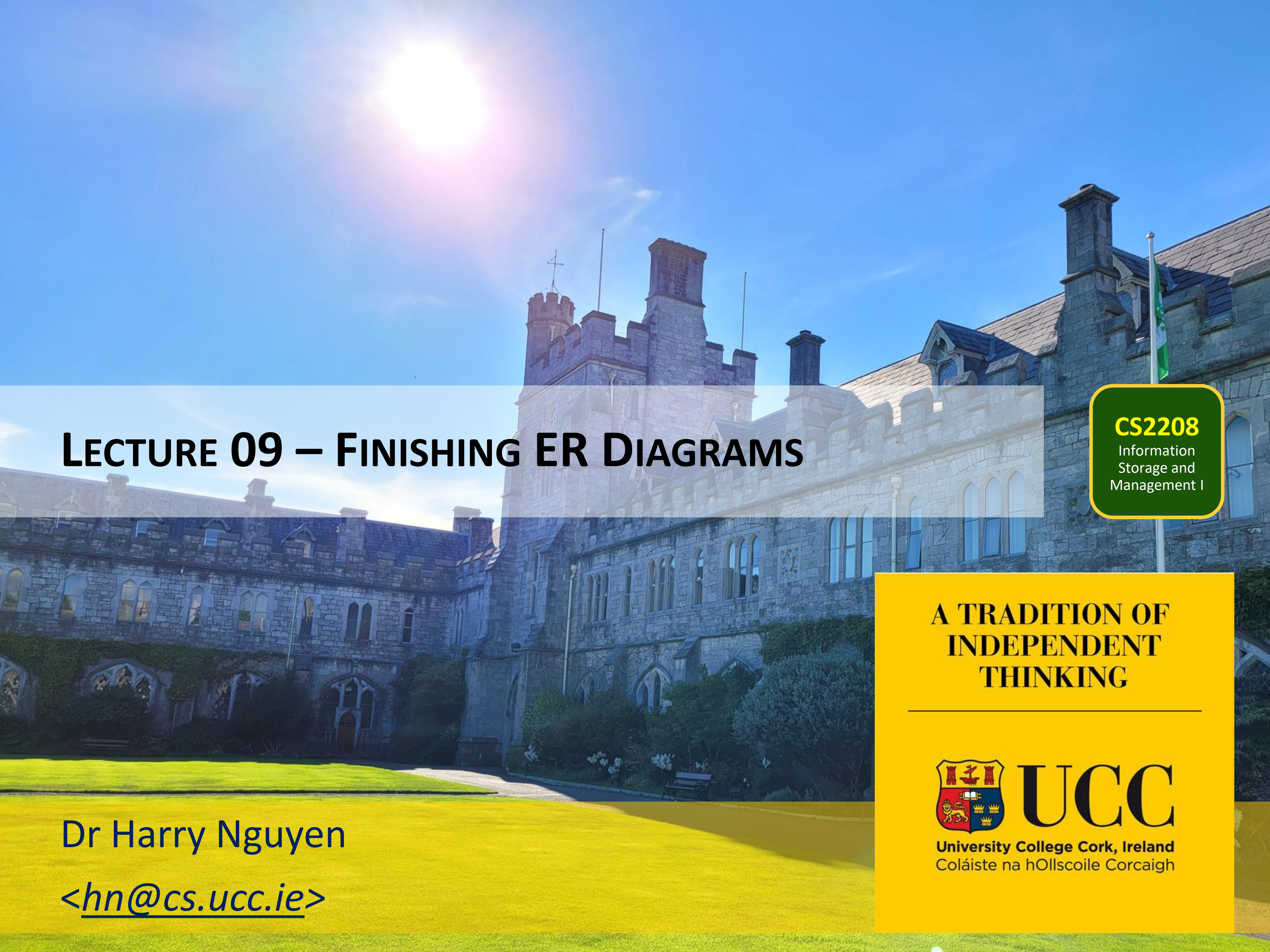


Lecture 09 – Finishing ER Diagrams

CS2208
Information
Storage and
Management I

A TRADITION OF
INDEPENDENT
THINKING

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Dr Harry Nguyen

<hn@cs.ucc.ie>

What We Saw Last Lecture

More ER Models

UML Concepts

Diagram Drawing

IDENTIFYING ENTITLES

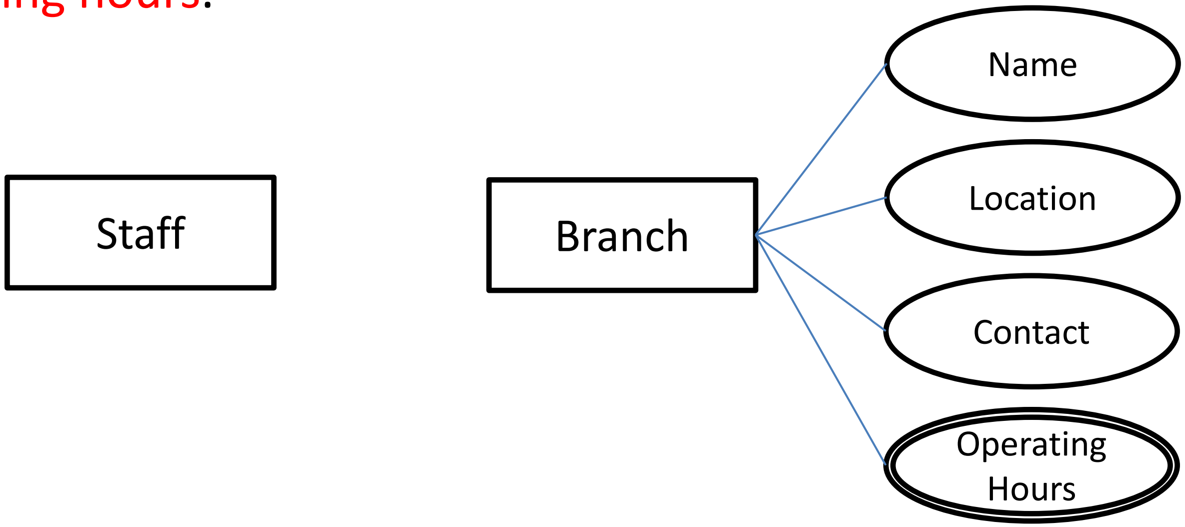
- “FutureLeases has many branches around the world. Every branch has staff. Some staff manage a branch; this includes supervising staff.”

Staff
Branch

  - There are two entities, Branch and Staff.
  - We represent these in our diagram with two rectangles containing a name.
  - Note the name starts with a capital letter.

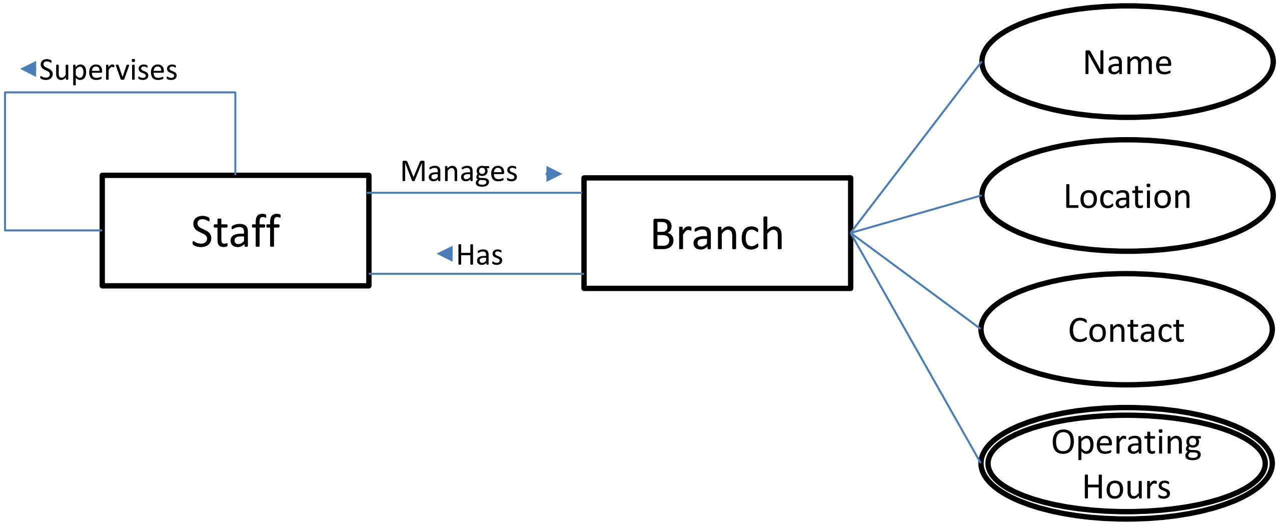
IDENTIFYING ATTRIBUTES

- Entities have properties, which are known as attributes
- For example:
  - “Every FutureLeases branch has name, location, contact information and operating hours.”



Identifying Relationships

- “FutureLeases has many branches around the world. Every branch has staff. Some staff manage a branch; this includes supervising staff.”



Extending our example

Description

Referring to our property leasing company example. Properties are managed by staff. Staff also deal with clients. A client can take a lease on a property. Clients can also view properties. Staff are described by a unique ID, a name, a position and a date of birth. Clients have an id, a name, a telephone number and an email for contact. Properties have an ID as well, along with an address, type of property (house or apartment), number of rooms and a rent amount. Finally, a lease has an ID, specifies a rent amount and a start and end date.

EXTENDING OUR EXAMPLE

Identifying Entities

Referring to our property leasing company example. Properties are managed by staff. Staff also deal with clients. A client can take a lease on a property. Clients can also view properties. Staff are described by a unique ID, a name, a position and a date of birth. Clients have an id, a name, a telephone number and an email for contact. Properties have an ID as well, along with an address, type of property (house or apartment), number of rooms and a rent amount. Finally, a lease has an ID, specifies a rent amount and a start and end date.

Extending our example

Identifying Attributes

Referring to our property leasing company example. Properties are managed by staff. Staff also deal with clients. A client can take a lease on a property. Clients can also view properties. Staff are described by a unique ID, a name, a position and a date of birth. Clients have an id, a name, a telephone number and an email for contact. Properties have an ID as well, along with an address, type of property (house or apartment), number of rooms and a rent amount. Finally, a lease has an ID, specifies a rent amount and a start and end date.

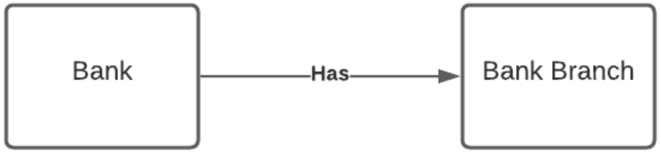
Extending our example

Identifying Relationships

Referring to our property leasing company example. Properties are managed by staff. Staff also deal with clients. A client can take a lease on a property. Clients can also view properties. Staff are described by a unique ID, a name, a position and a date of birth. Clients have an id, a name, a telephone number and an email for contact. Properties have an ID as well, along with an address, type of property (house or apartment), number of rooms and a rent amount. Finally, a lease has an ID, specifies a rent amount and a start and end date.

Strong and Weak Entities

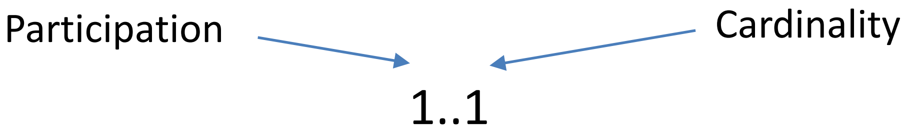
- A weak entity is an entity whose existence is dependent on some other entity type
- A strong entity is an entity whose existence doesn't depend on some other entity type
- For example – if we have a database representing a Bank, then Bank Branches can be considered to be weak as they cannot exist without a Bank existing



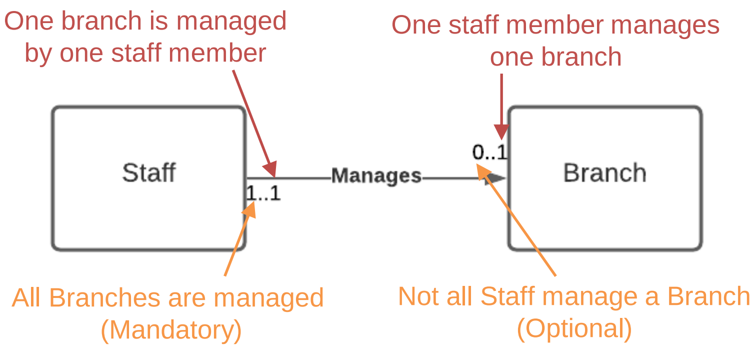
- Note: strong entities and weak entities aren't represented in some ER formats – in this case we would add a note to describe the entity as strong or weak

MULTIPLICITY IN DETAIL

- Multiplicity is made up of two parts, cardinality and participation



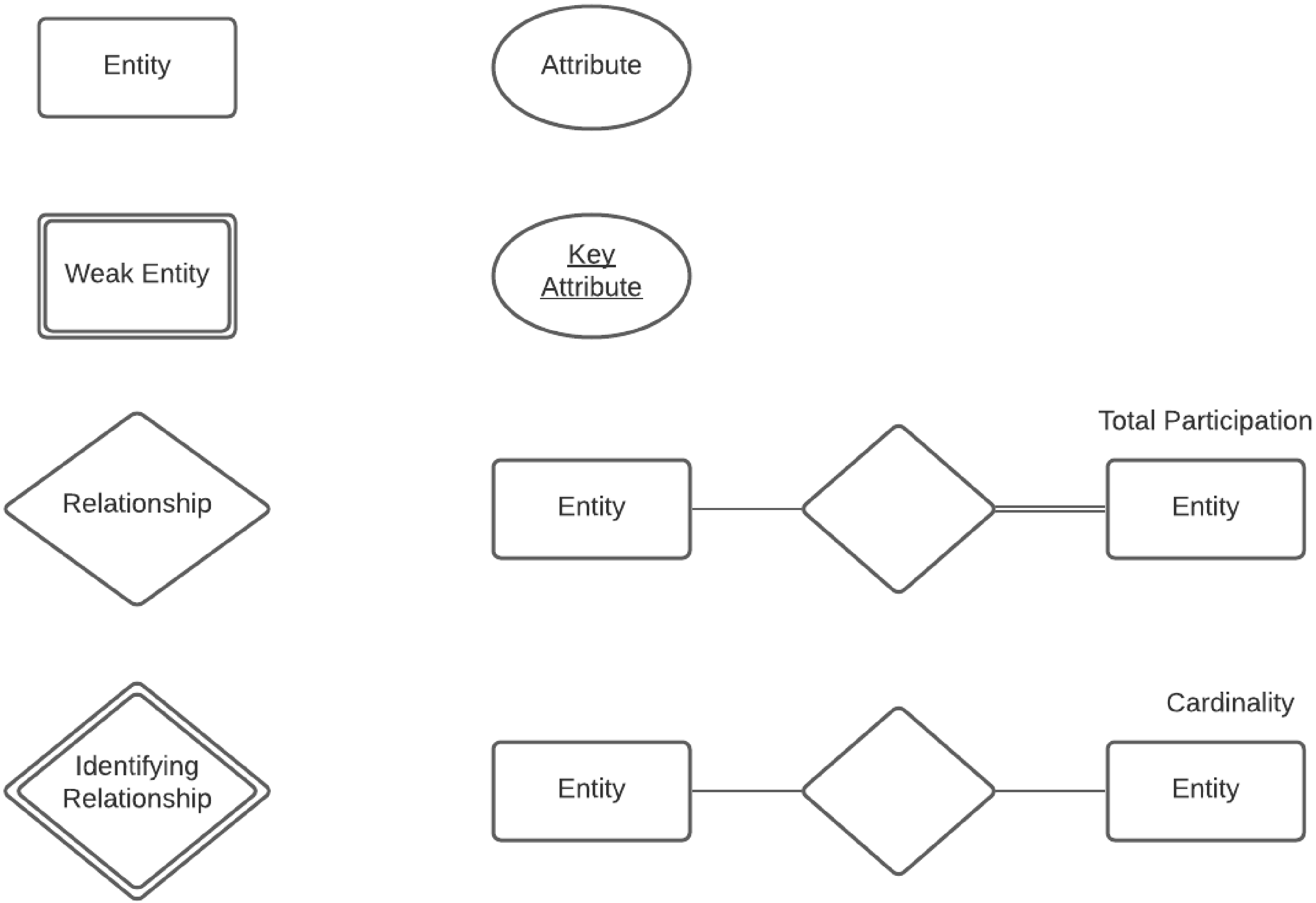
- Cardinality describes the maximum number of possible occurrences of participating in a relationship
- Participation determines whether all or only some entity occurrences participate in a relationship
  - If the value is none zero then participation is mandatory, if it can be zero, participation is optional


One branch is managed by one staff member
One staff member manages one branch
Staff
Manages
0..1
Branch
1..1
All Branches are managed (Mandatory)
Not all Staff manage a Branch (Optional)

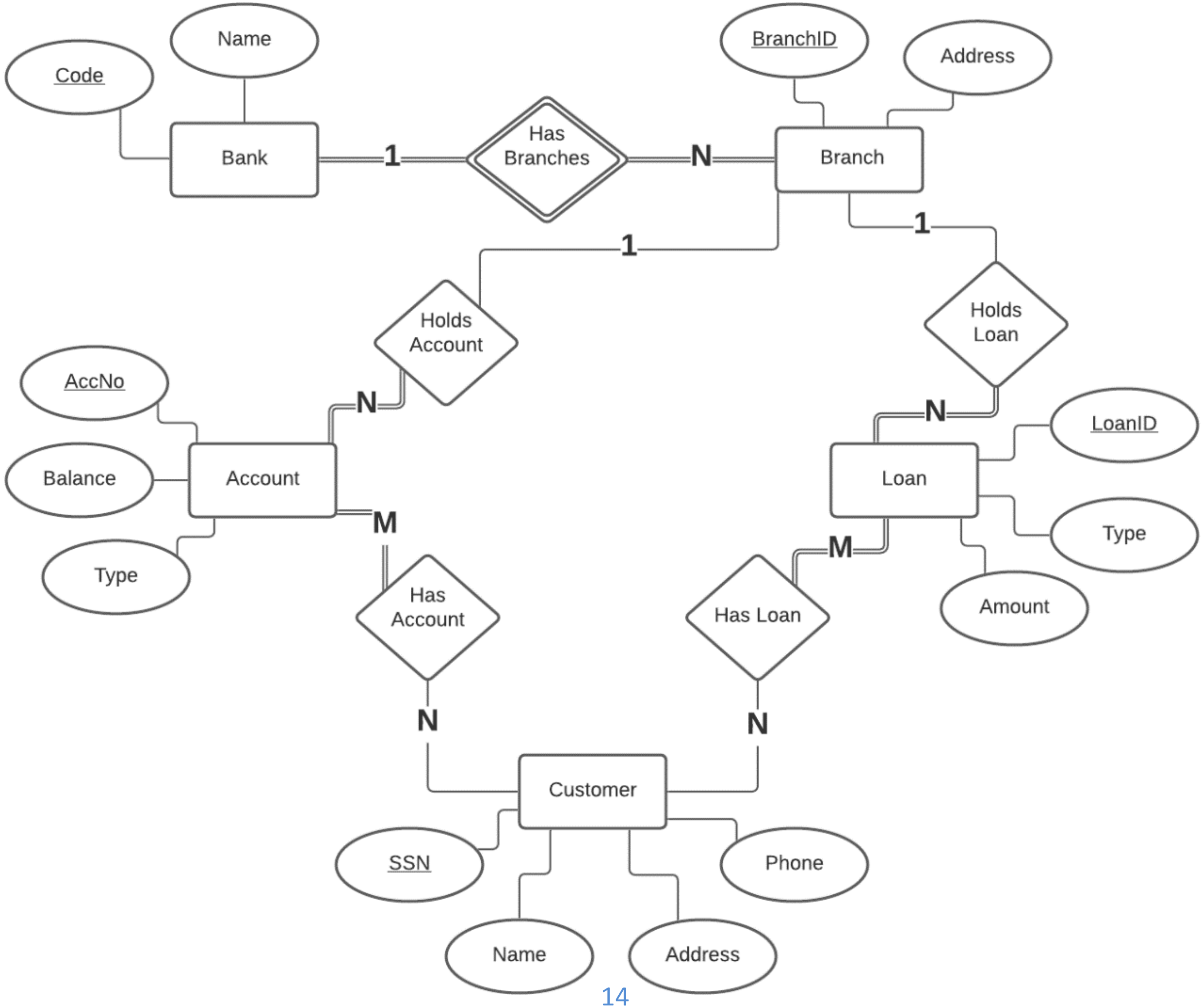
Another Example

Let’s imagine that we have to implement a database to manage a bank, its branches, its customers and their accounts and loans held at the bank. The bank has branches – offices where business can be transacted. Customers deal with a branch. Branches hold accounts for customers while a customer can have one or more accounts at a branch. Branches also manage loans to customers, and customers can have several loans with a branch. Both loans and accounts can be jointly held. Banks have an ID and a name. Branches have an ID and an address. Accounts have an ID, a balance and a type. Loans have an amount, a type and an ID. Customers have an account number, a name, an address and a phone number.

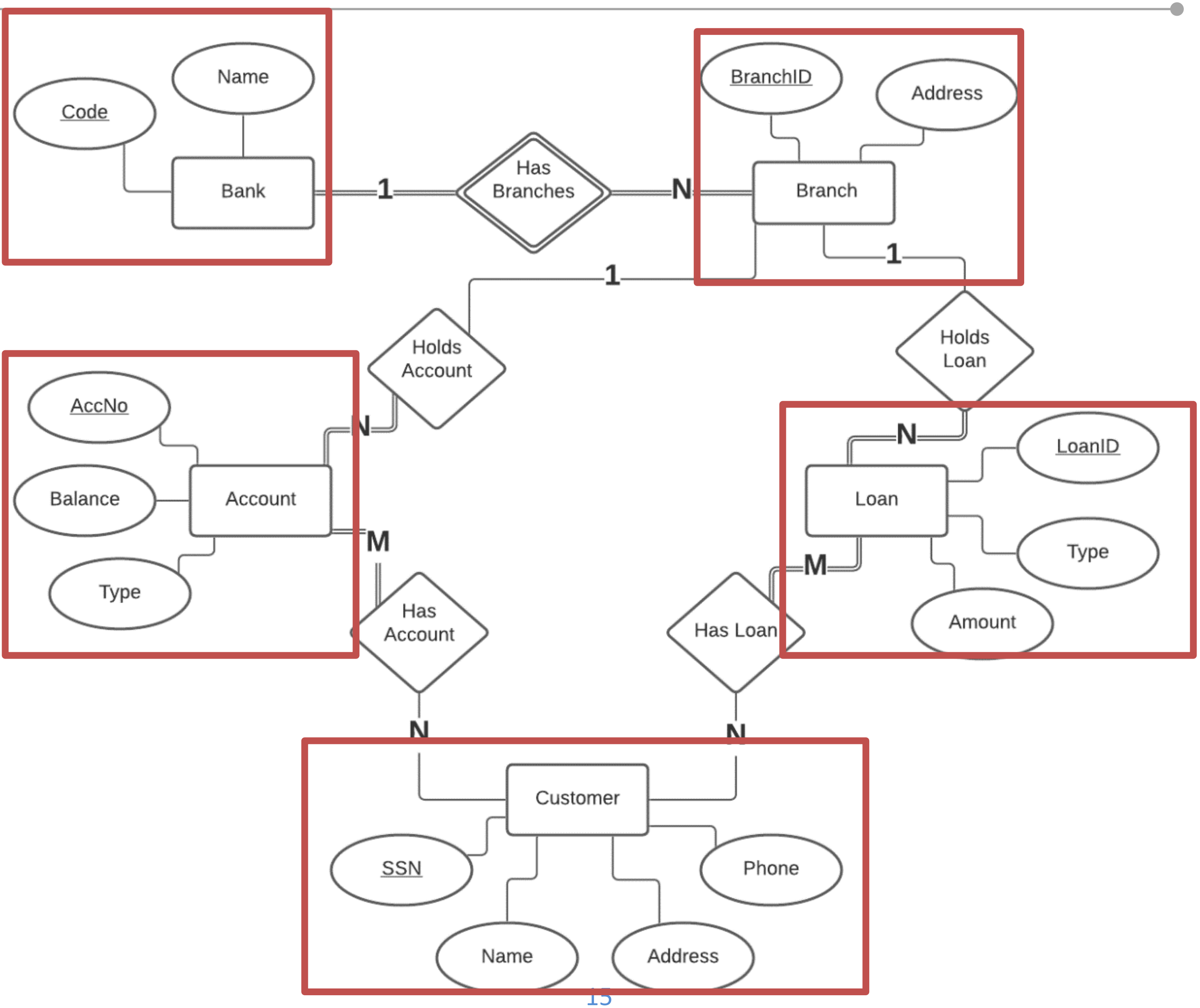
More Symbols



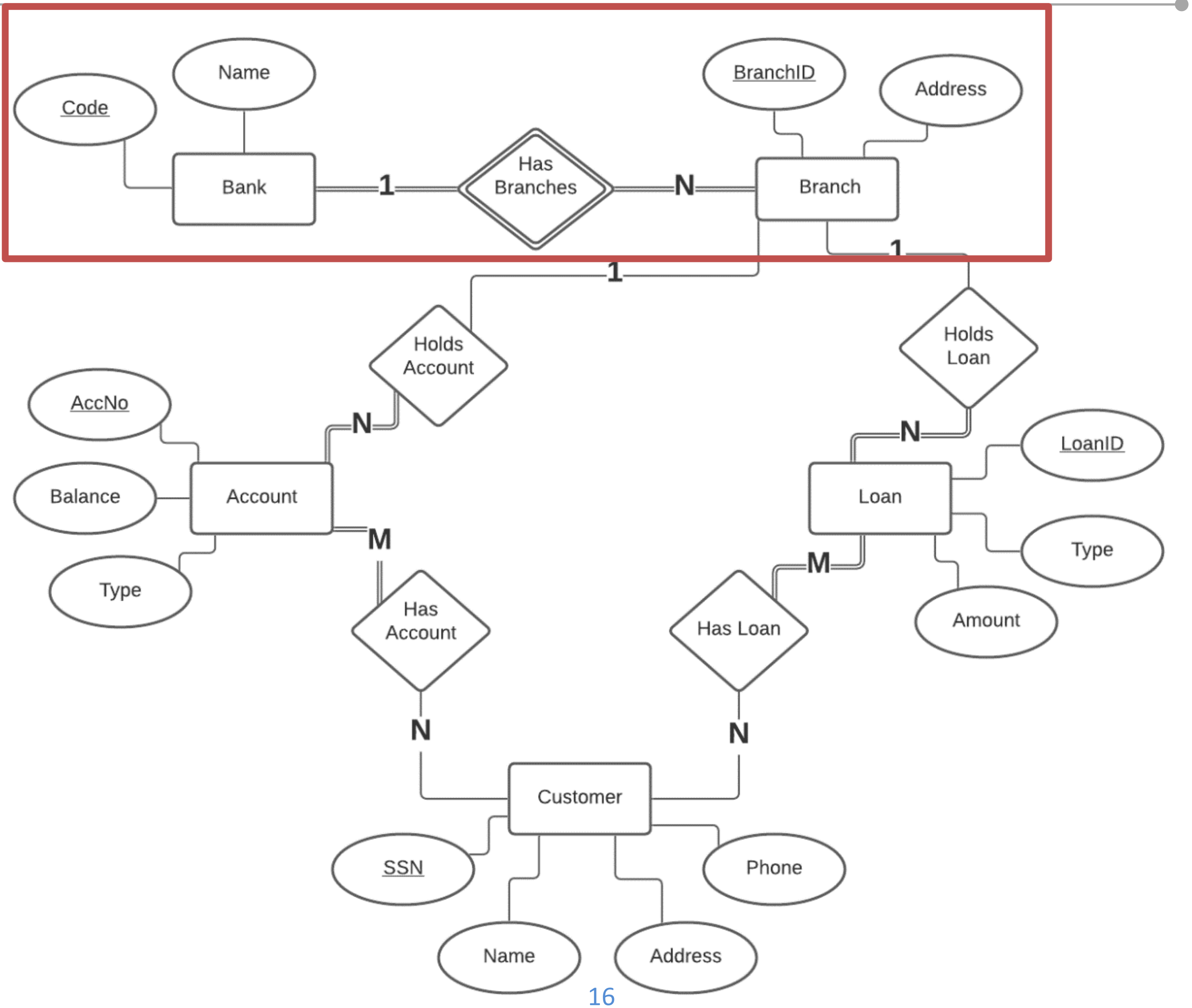
Another Diagram



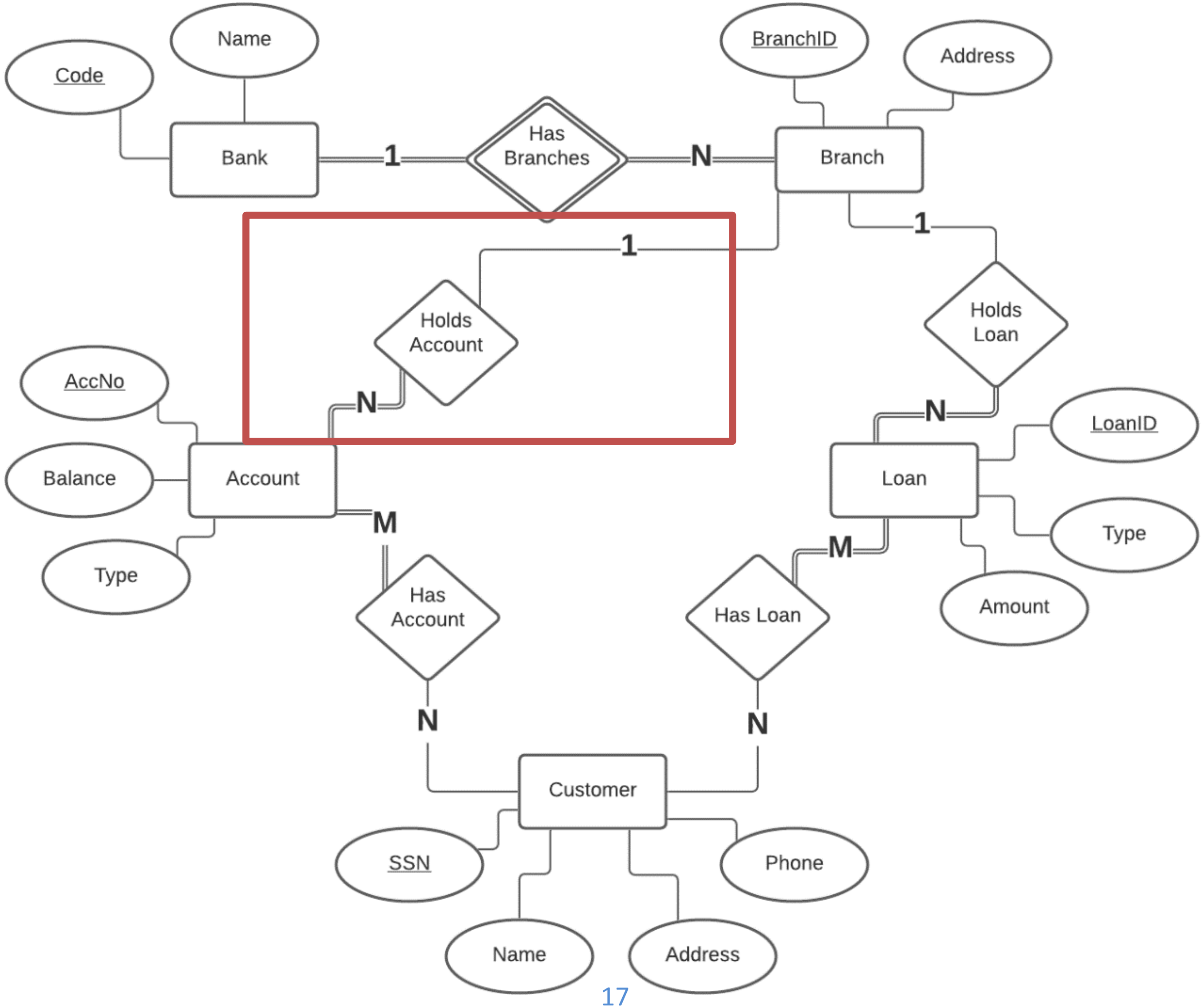
ANOTHER DIAGRAM - ENTITIES



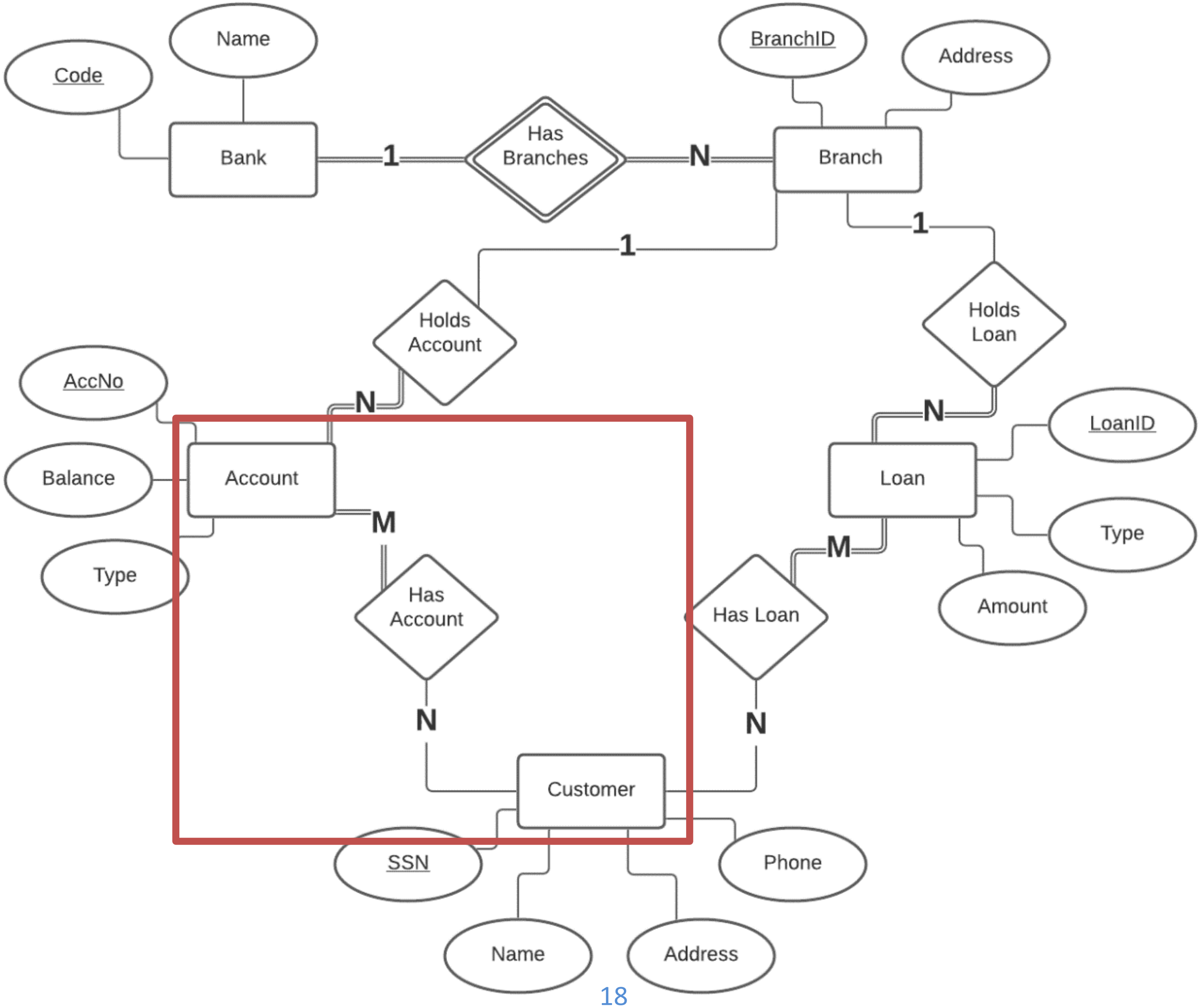
ANOTHER DIAGRAM - RELATIONSHIPS



Another Diagram - Relationships



ANOTHER DIAGRAM - RELATIONSHIPS



EER Diagrams – Why?

ER diagrams, executed correctly are very unambiguous expressions of a set of requirements and can be translated into a database schema

However, different ways of creating databases arose – for example, driven by code. This brought about a need to add more expressivity to ER diagrams

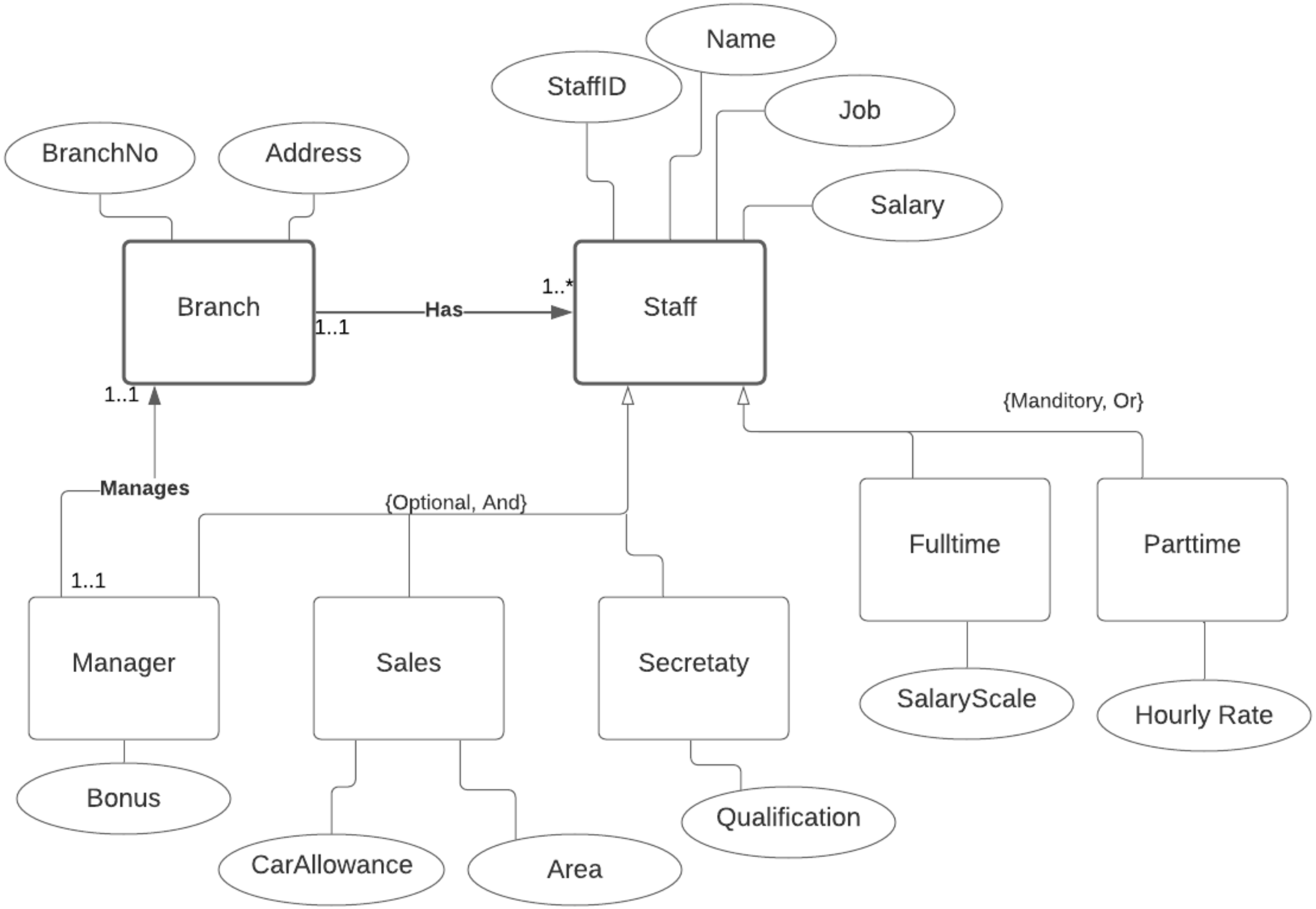
- This gives rise to Extended ER (EER) diagrams

Many of these extras relate to concepts from software engineering and Object Oriented Programming (OOP). For example:

- Specialization/Generalization is similar to inheritance and define has-a relationships
- Aggregation and Composition capture has a relationships

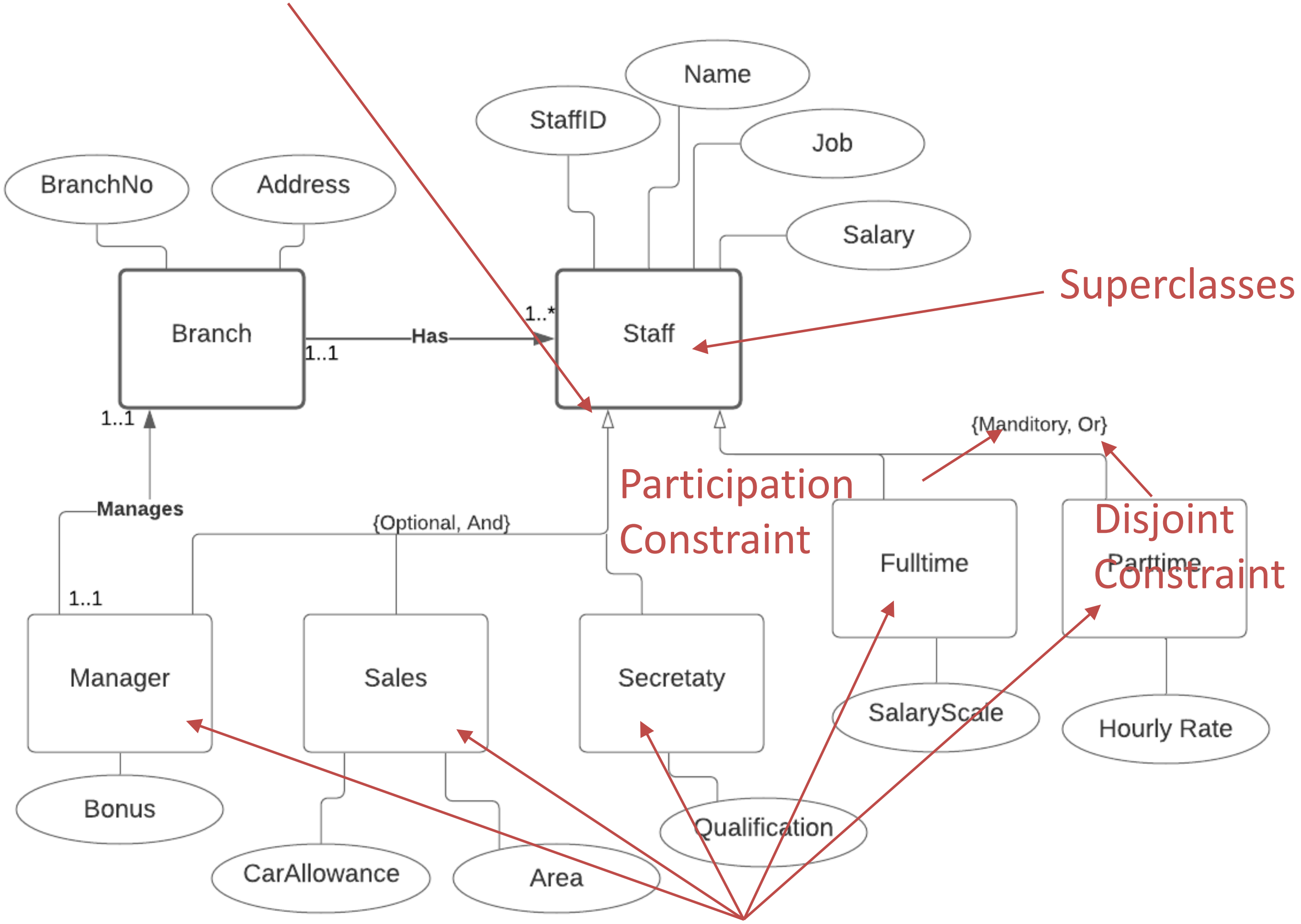
We only briefly touch on EER diagrams to give a flavor – we will use ER diagrams going forward.

EER DIAGRAMS



EER DIAGRAMS

Generalization/Specialization indicator



Superclasses

Participation Constraint

Disjoint Constraint

Subclasses

EER Diagrams – Concepts

- Taking Generalization and Specialization as our focus.
  - We can express which class is a parent class and which is a subclass by drawing an arrow from the subclass to the superclass with the arrow pointing at the superclass. For example, ”Sales People are Staff” is indicated by an arrow from Sales to Staff
  - Participation constraints dictate whether every member of the superclass must participate as a member of a subclass.

EER Diagrams – Concepts

- Disjoint constraints describe the relationship between superclass and subclasses and indicates if a superclass can be more than one type of subclass
  - For example, a staff entity might be both a manager and a sales person – this is indicated by an ‘AND’ constraint.
  - However, staff can only be either fulltime or parttime. In this case the disjoint constraint is ‘OR’

SUMMARY

More ER Examples

ERD Elements

Extended ER