



Institut Informatika & Bisnis
DARMAJAYA
Yayasan Alfian Husin



**Kampus
Merdeka**
INDONESIA JAYA

**MERDEKA
BELAJAR**

DATA SCIENCE DARMAJAYA
“YOUR BEST FUTURE IN DATA”

MEETING: [9 & 10]

ENTITY RELATIONAL DIAGRAM

BY: HENDRA KURNIAWAN



Entity Relationship Diagram

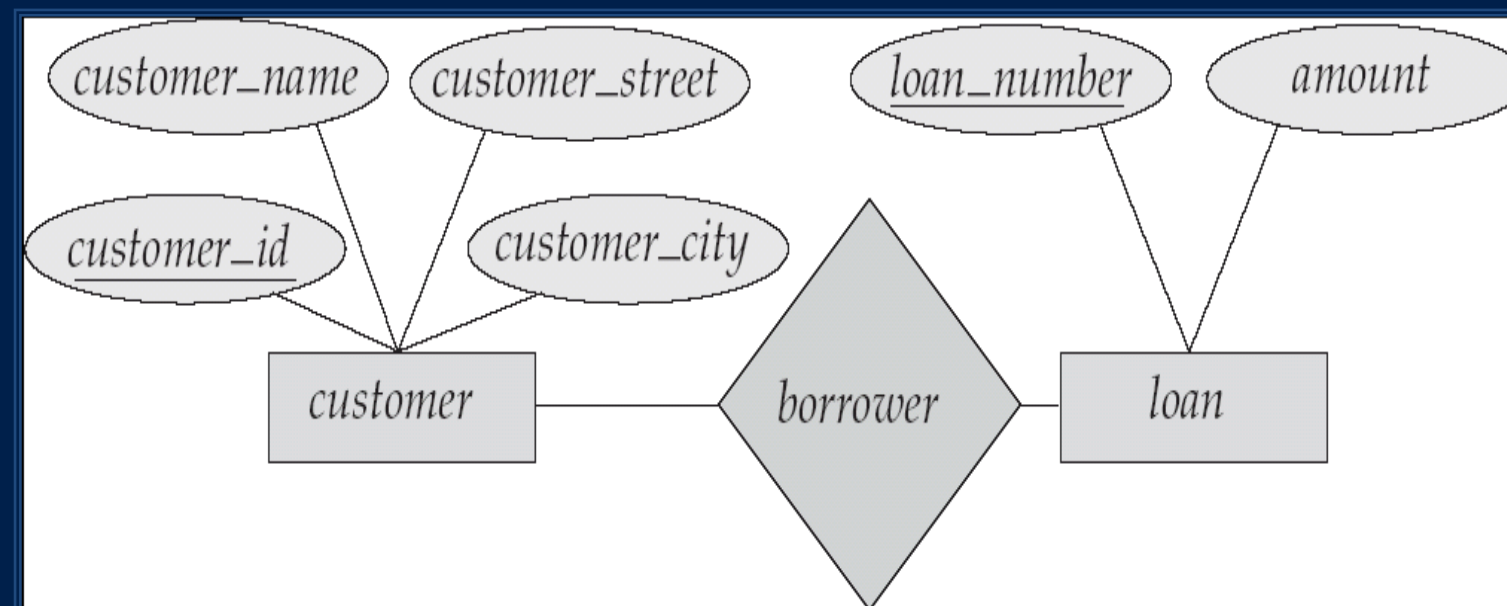
1. ER-Diagram components and their explanations (Entities, Attributes, Relations and Symbols)
2. Techniques for Building Database Designs with ER-Diagrams
3. Advantages and Disadvantages of ER-Diagram



Learning Objectives

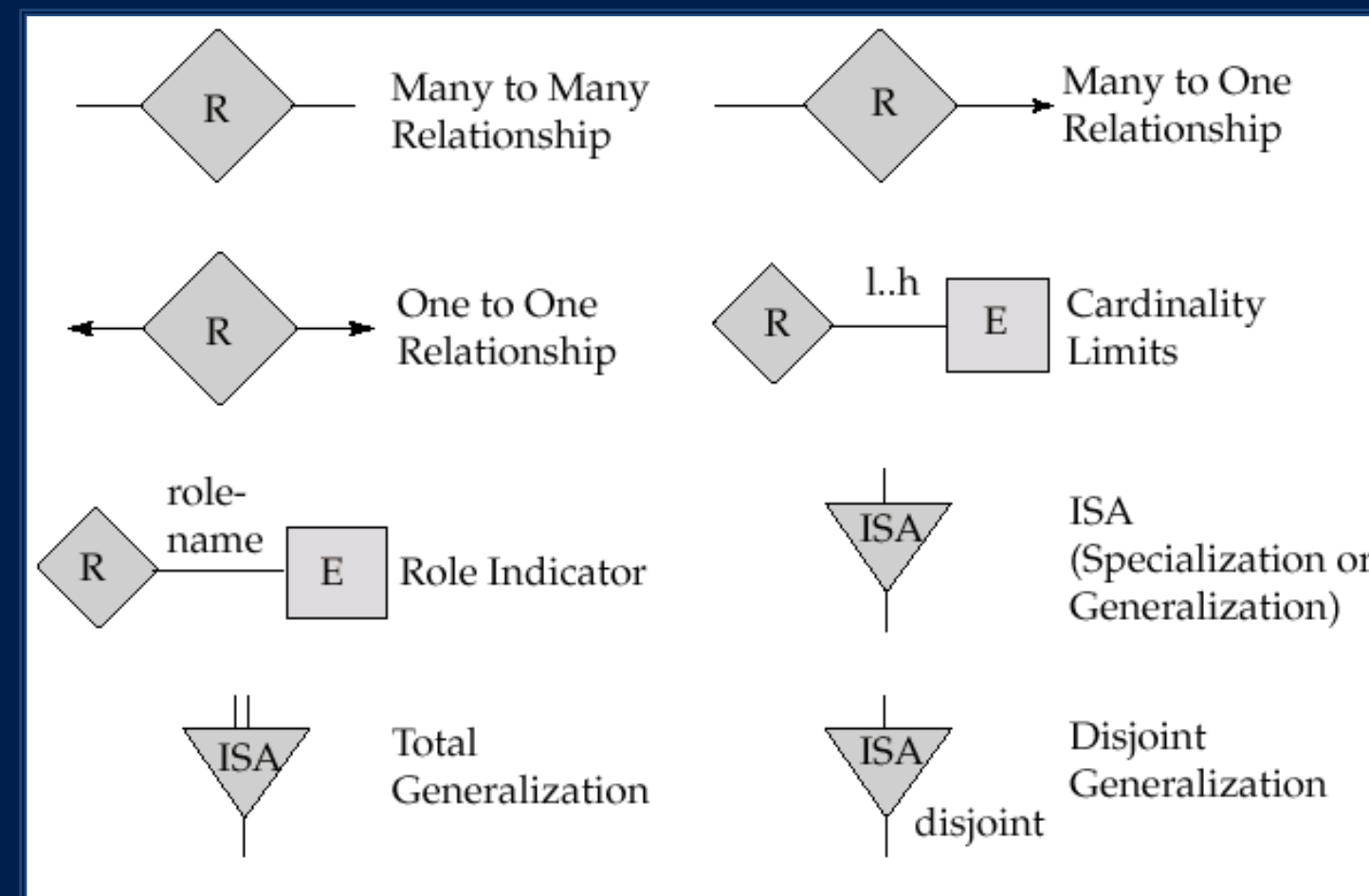
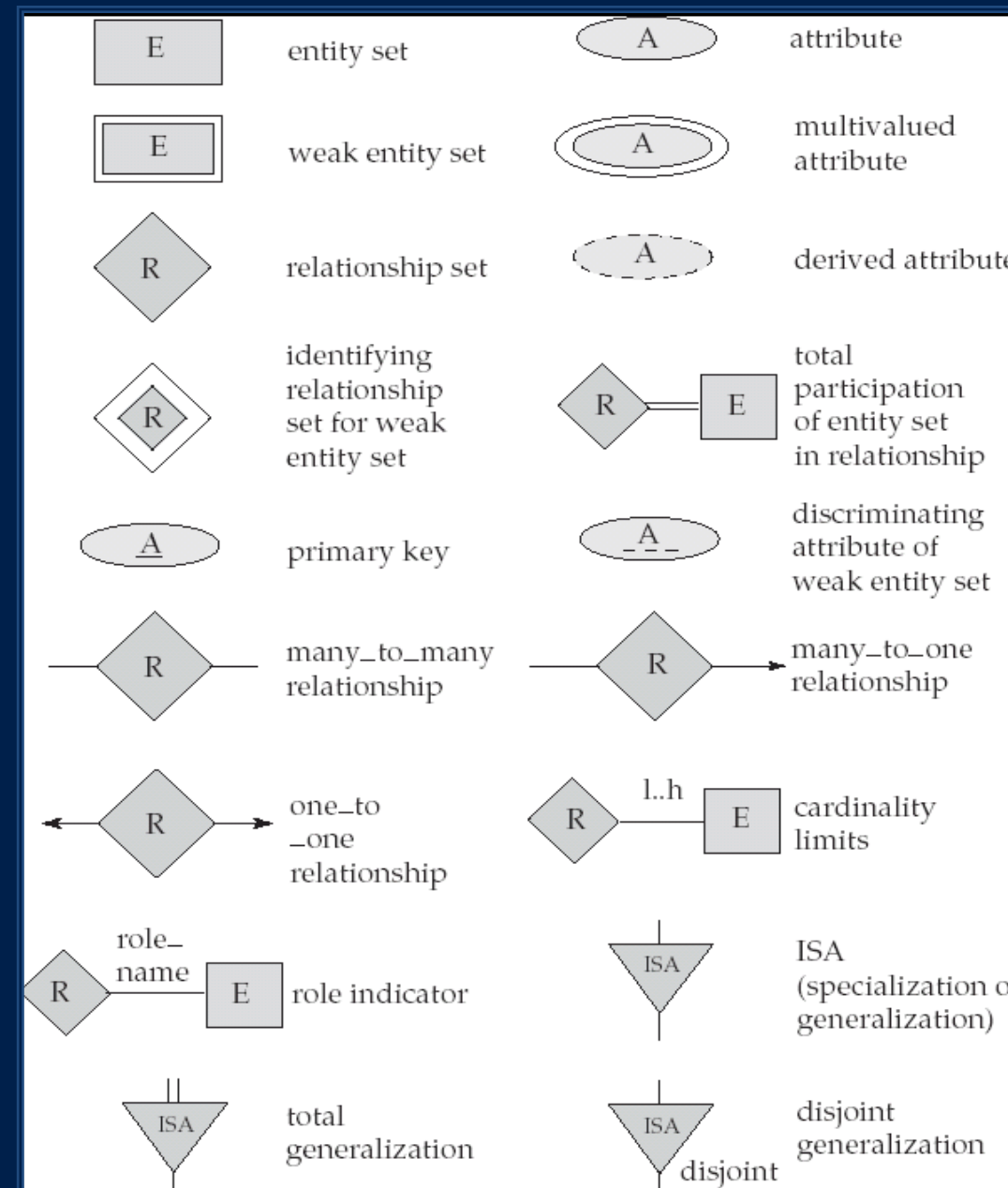
1. Understand the fundamental components of an Entity-Relationship (ER) diagram, including entities, attributes, relationships, and symbols, and their respective roles in representing database structures.
2. Understand various techniques and methodologies for creating effective database designs using ER diagrams
3. Explore the benefits and limitations of using ER diagrams as a tool for designing and visualizing database systems, and gain insights into when and how to effectively employ them in the database design process.

Entity Relationship Diagram

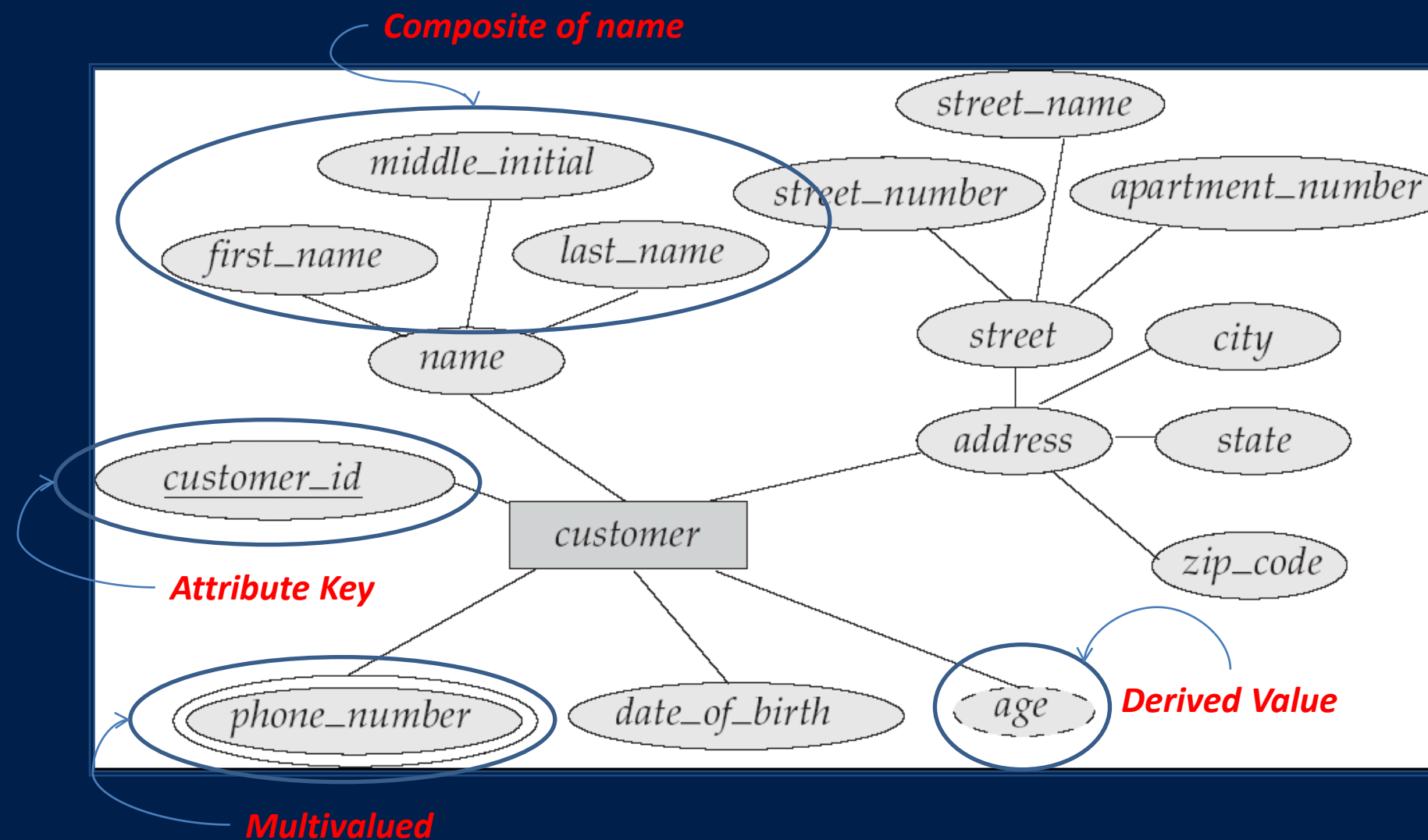


- ❑ **Rectangles** represent entity sets.
- ❑ **Diamonds** represent relationship sets.
- ❑ **Lines link** attributes to entity sets and entity sets to relationship sets.
- ❑ **Ellipses** represent attributes
 - ❑ **Double ellipses** represent multivalued attributes.
 - ❑ **Dashed ellipses** denote derived attributes.
- ❑ **Underline** indicates primary key attributes

ER-Diagram Notation

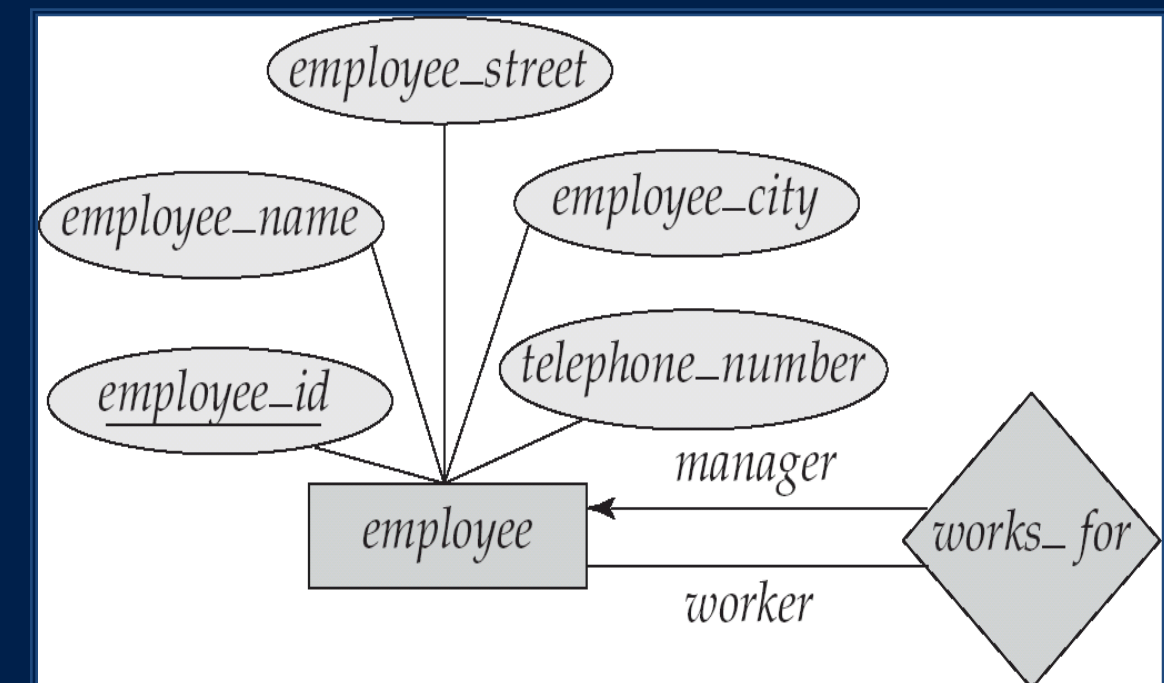


E-R Diagram With Composite, Multivalued, and Derived Attributes



Roles

- Entity sets of a relationship need *not be distinct*
- The labels “manager” and “worker” are called *roles*; they specify how employee entities interact via the works_for relationship set.
- Roles are indicated in E-R diagrams by *labeling the lines that connect diamonds to rectangles*.
- Role labels are optional, and are used to clarify semantics of the relationship



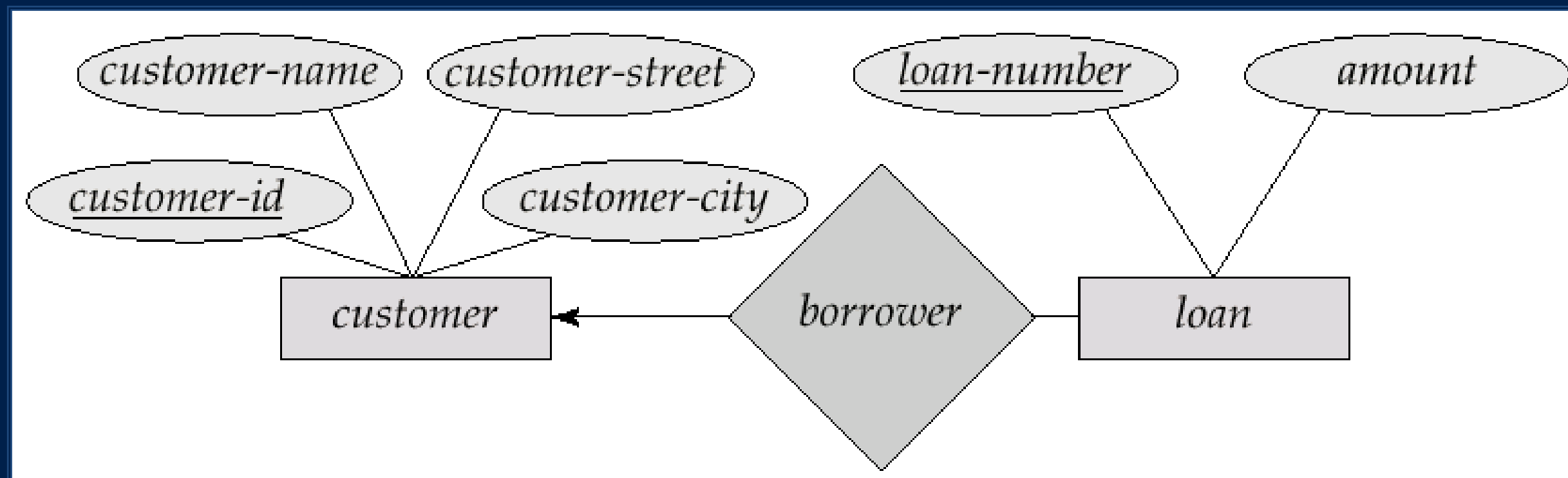


Cardinality Constraints

- We express cardinality constraints by drawing either a directed line (\rightarrow), signifying “one,” or an undirected line ($—$), signifying “many,” between the relationship set and the entity set.
- One-to-one relationship:
 - A customer is associated with at most one loan via the relationship *borrower*
 - A loan is associated with at most one customer via *borrower*

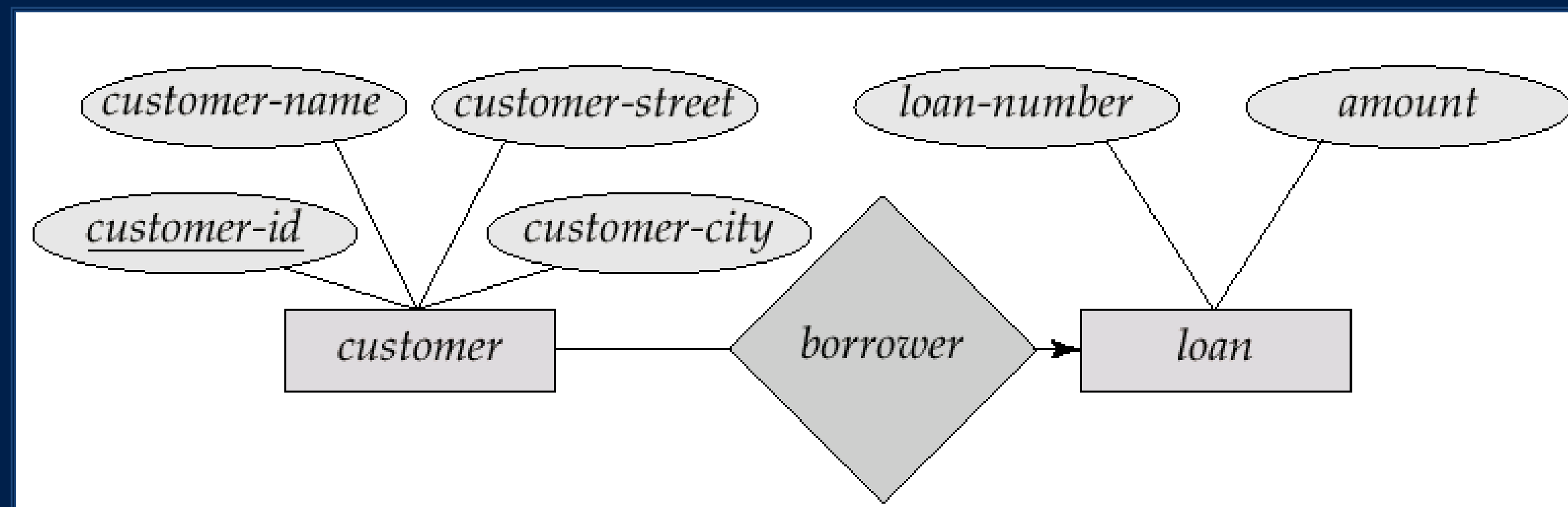
One-To-Many Relationship

- In the one-to-many relationship a loan is associated with at most one customer via *borrower*, a customer is associated with several (including 0) loans via *borrower*



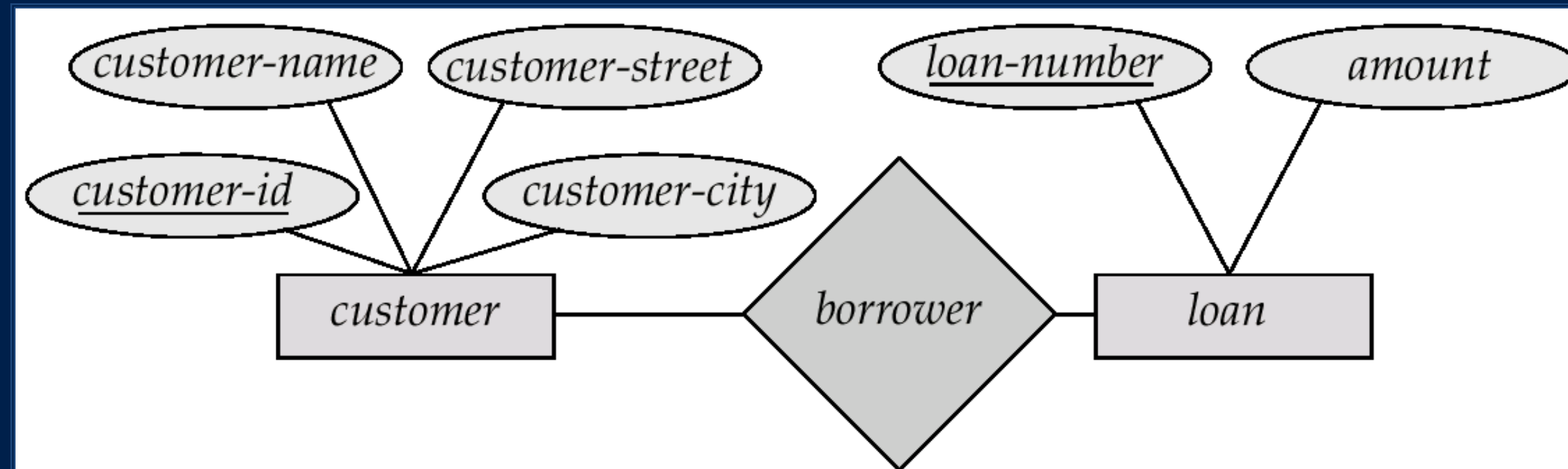
Many-To-One Relationships

- In a many-to-one relationship a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



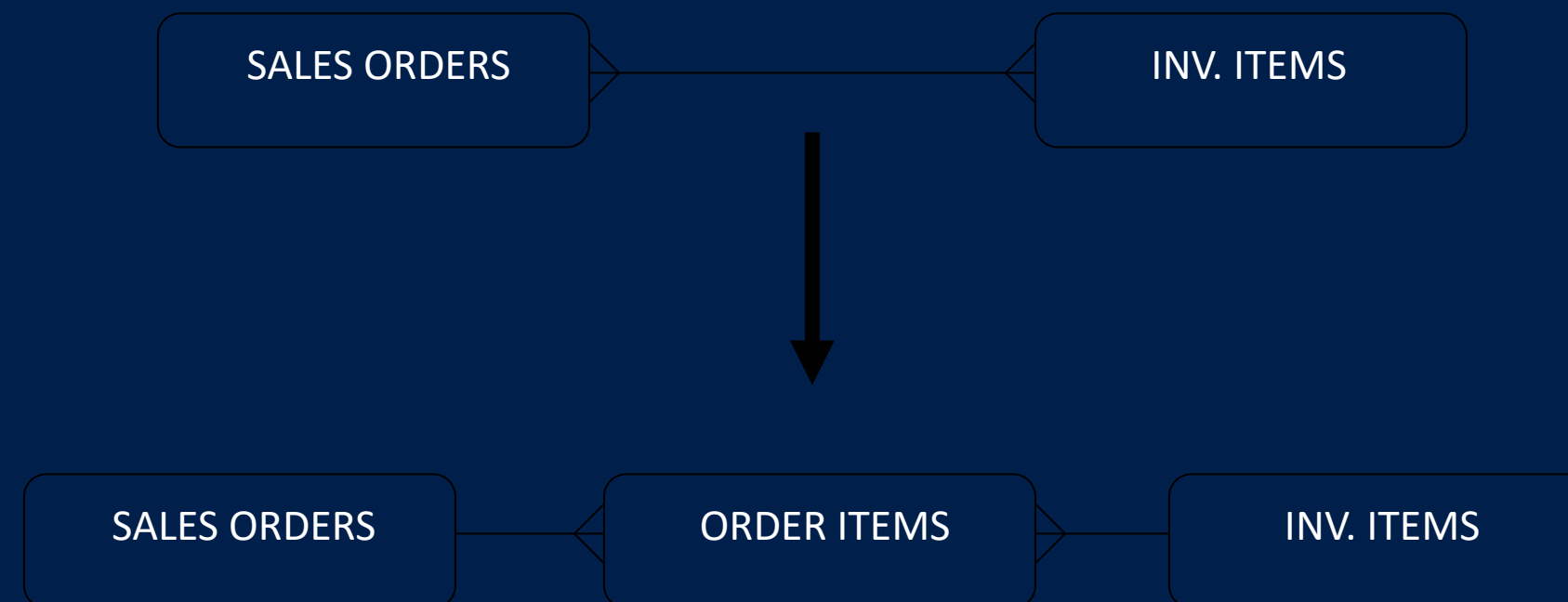
Many-To-Many Relationship

- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower



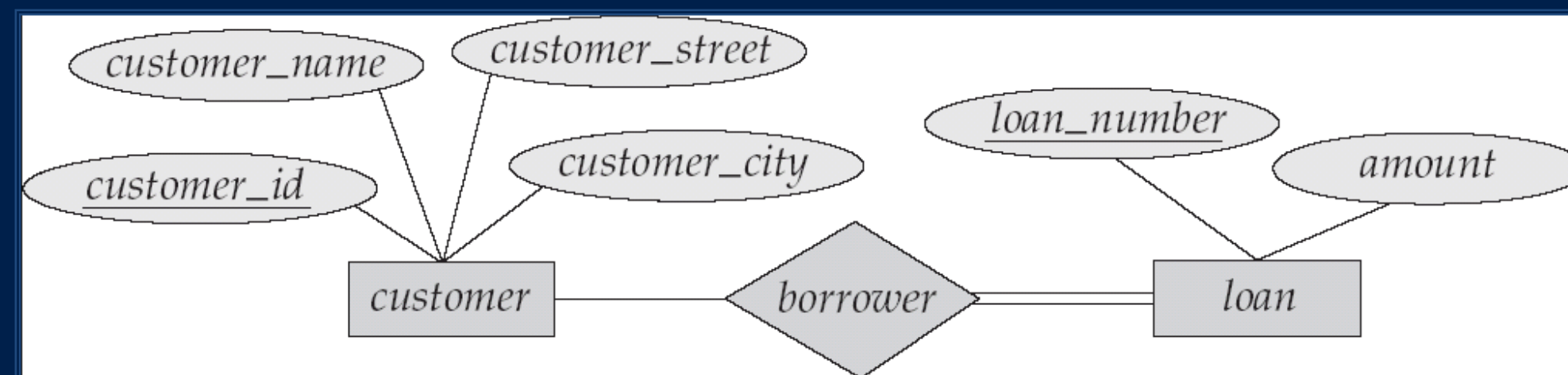
Resolving Many-to-Many Relationships

Many-to-many relationships should be avoided. We can resolve a many-to-many relationship by dividing it into two one-to-many relationships.



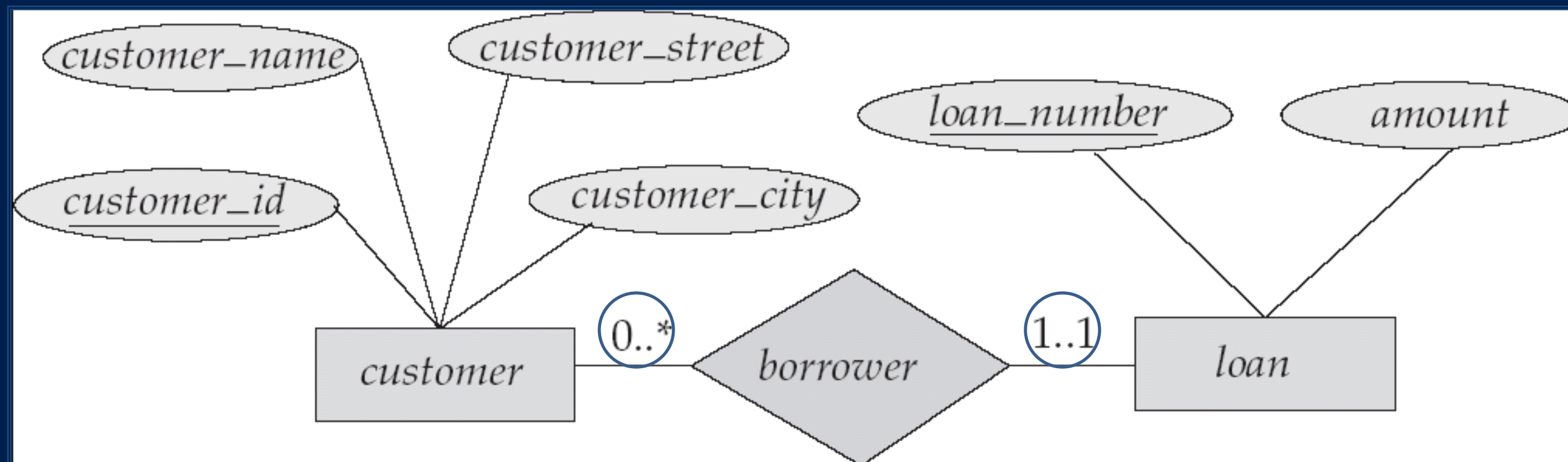
Participation of an Entity Set in a Relationship Set

- **Total participation** (indicated by **double line**): every entity in the entity set participates in at least one relationship in the relationship set
 - E.g. participation of loan in borrower is total
 - ▶ every loan must have a customer associated to it via borrower
- **Partial participation**: some entities may not participate in any relationship in the relationship set
 - Example: participation of customer in borrower is partial



Alternative Notation for Cardinality Limits

- Cardinality limits can also express participation constraints





Weak Entity Sets

- An entity set that does not have a primary key is referred to as a **weak entity set**.
- The existence of a weak entity set depends on the existence of a **identifying entity set**
 - it must relate to the identifying entity set via a total, one-to-many relationship set from the identifying to the weak entity set
 - Identifying relationship depicted using a double diamond

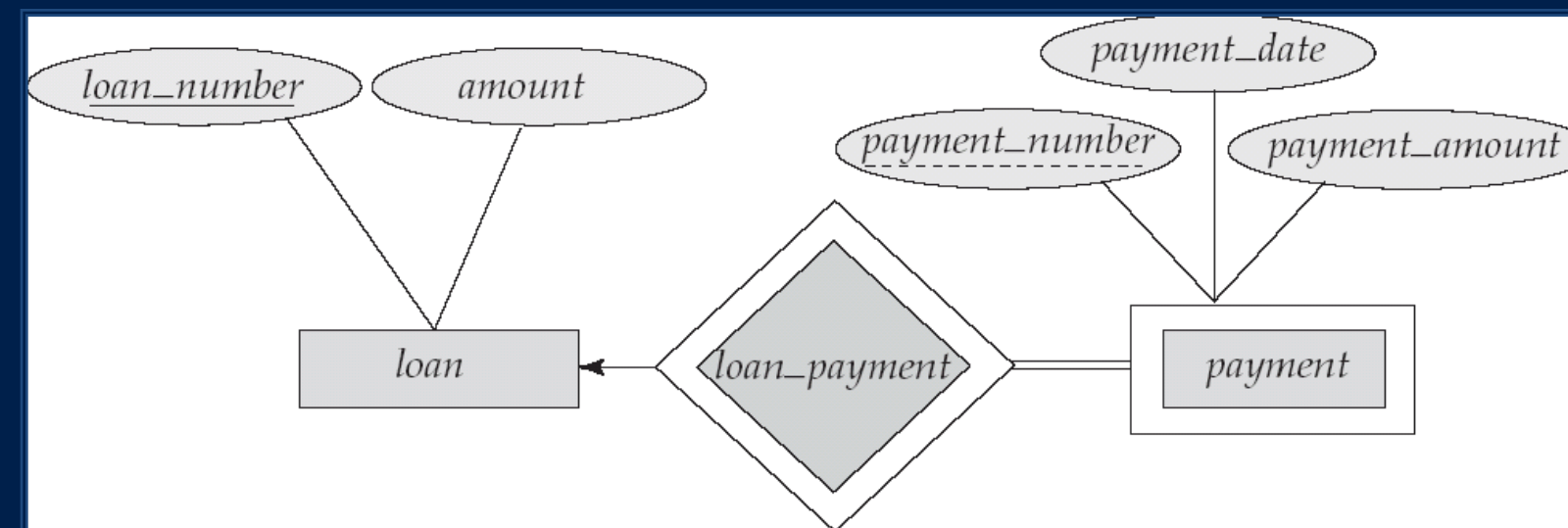


Weak Entity Sets

- The **discriminator** (or *partial key*) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.

Weak Entity Sets

- We depict a weak entity set by double rectangles.
- We underline the discriminator of a weak entity set with a dashed line.
- *payment_number* – discriminator of the *payment* entity set
- Primary key for *payment* – (*loan_number*, *payment_number*)





Weak Entity Sets

- **Note:** the primary key of the strong entity set is not explicitly stored with the weak entity set, since it is implicit in the identifying relationship.
- If *loan_number* were explicitly stored, *payment* could be made a strong entity, but then the relationship between *payment* and *loan* would be duplicated by an implicit relationship defined by the attribute *loan_number* common to *payment* and *loan*

More Weak Entity Sets Example

- In a university, a *course* is a **strong entity** and a *course_offering* can be modeled as a **weak entity**
- The discriminator of *course_offering* would be *semester* (including year) and *section_number* (if there is more than one section)
- If we model *course_offering* as a strong entity we would model *course_number* as an attribute.

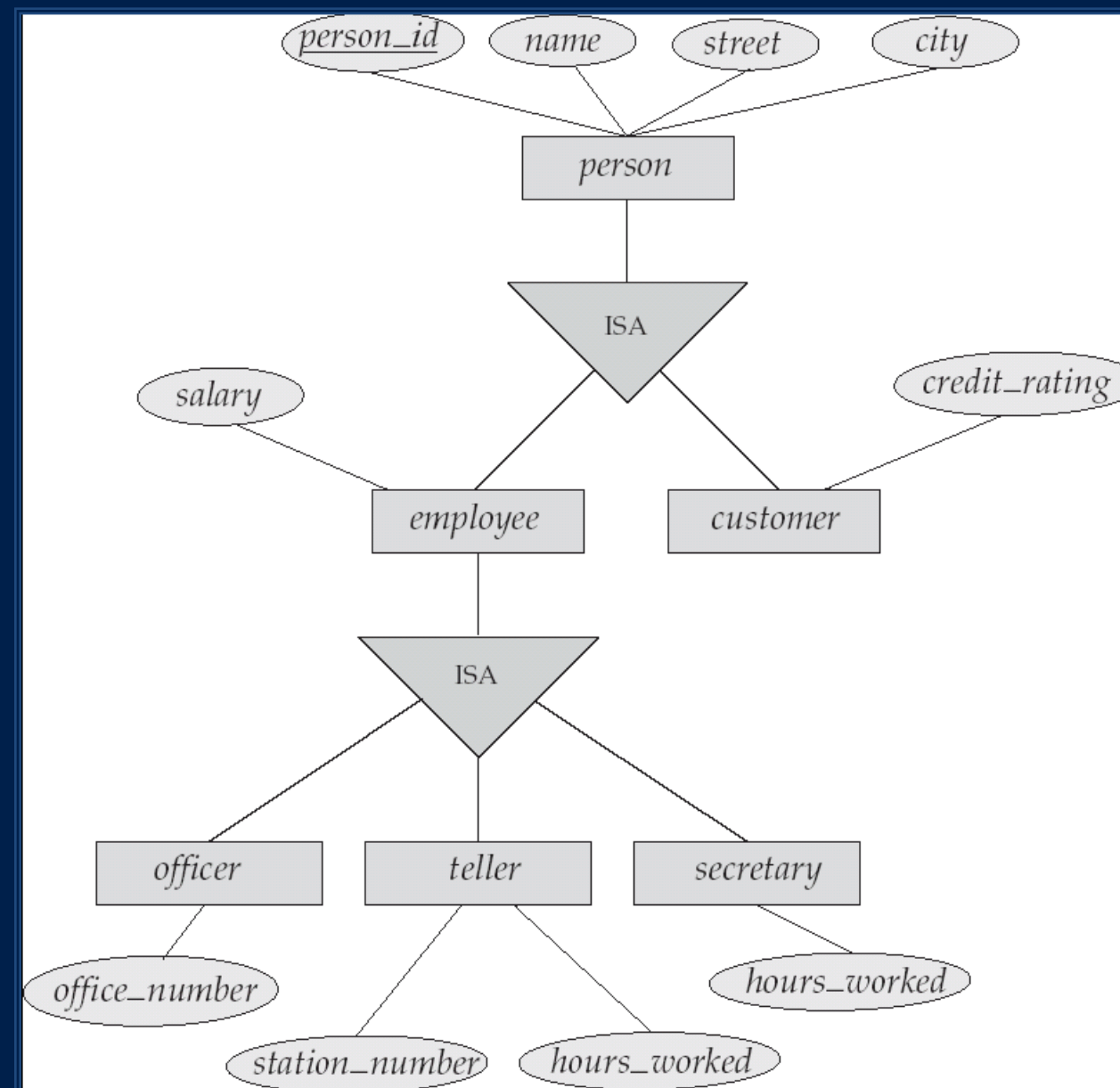
Then the relationship with *course* would be implicit in the *course_number* attribute



Extended E-R Features: Specialization

- Top-down design process; we designate subgroupings within an entity set that are distinctive from other entities in the set.
- These subgroupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set.
- Depicted by a *triangle* component labeled ISA (E.g. *customer* “is a” *person*).
- ***Attribute inheritance*** – a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.

Specialization Example





Extended E-R Features: Generalization

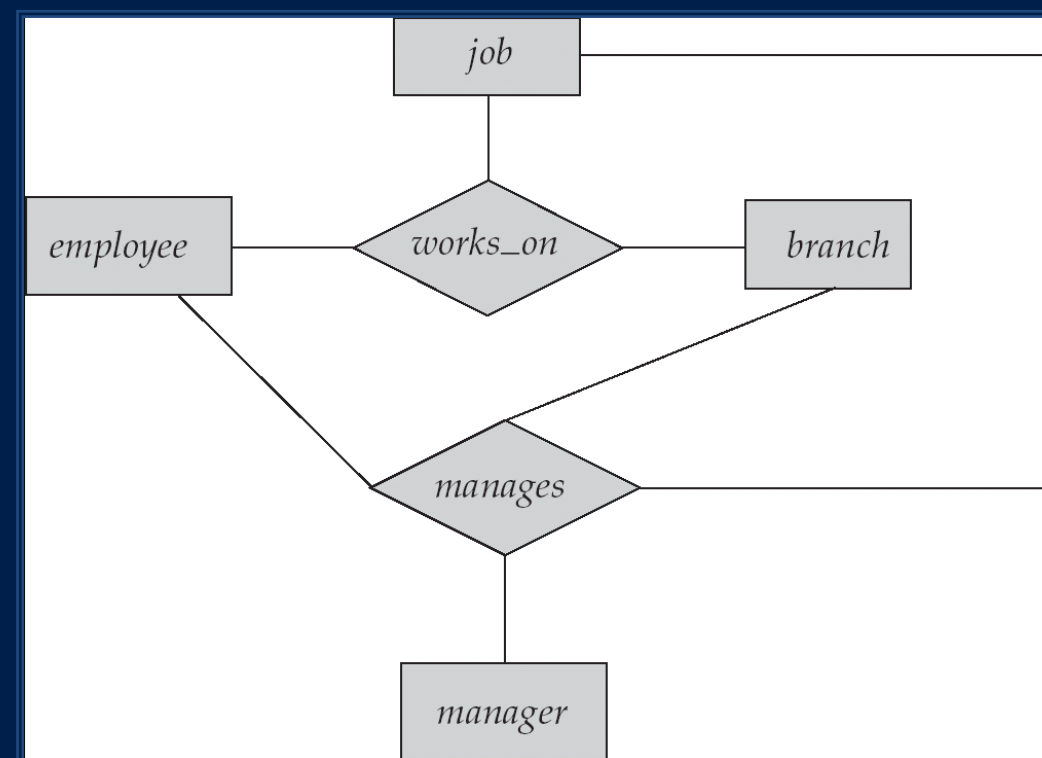
- A **bottom-up design process** – combine a number of entity sets that share the same features into a higher-level entity set.
- **Specialization and generalization** are simple inversions of each other; they are represented in an E-R diagram in the same way.
- The terms specialization and generalization are used interchangeably.

Specialization and Generalization (Cont.)

- Can have multiple specializations of an entity set based on different features.
- E.g. *permanent_employee* vs *temporary_employee*, in addition to *officer* vs. *secretary* vs. *teller*
- Each particular employee would be
 - a member of one of *permanent_employee* or *temporary_employee*,
 - and also a member of one of *officer*, *secretary*, or *teller*
- The *ISA relationship* also referred to as *superclass - subclass* relationship

Aggregation

- Consider the ternary relationship *works_on*, which we saw earlier
- Suppose we want to record managers for tasks performed by an employee at a branch

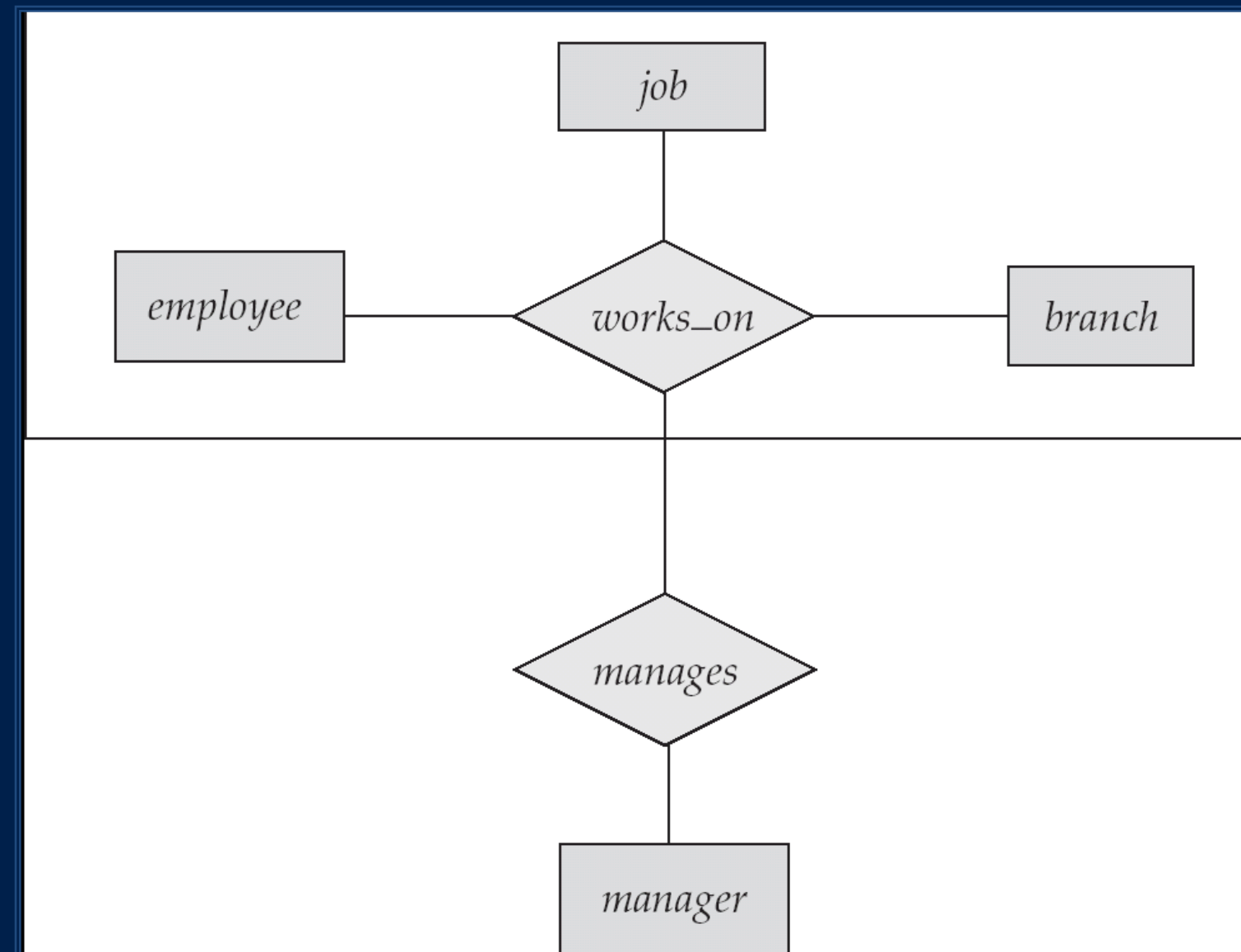




Aggregation

- Relationship sets *works_on* and *manages* represent overlapping information
 - Every *manages* relationship corresponds to a *works_on* relationship
 - However, some *works_on* relationships may not correspond to any *manages* relationships
 - So we can't discard the *works_on* relationship
- Eliminate this redundancy via *aggregation*
 - Treat relationship as an abstract entity
 - Allows relationships between relationships
 - Abstraction of relationship into new entity
- Without introducing redundancy, the following diagram represents:
 - An employee works on a particular job at a particular branch
 - An employee, branch, job combination may have an associated manager

ER-Diagram With Aggregation





CONCLUSION

- An Entity-Relationship (ER) Diagram is a powerful tool in the field of database design. It allows us to visually represent the structure and relationships within a database system. The key components of an ER Diagram include entities, attributes, relationships, and symbols. Entities represent real-world objects or concepts, while attributes define the properties or characteristics of these entities. Relationships depict how entities are connected and interact with each other. Symbols like diamonds and lines help illustrate the nature of these relationships.
- To design effective database structures with ER Diagrams, one must understand various techniques and methodologies. This includes best practices, normalization, and the translation of conceptual models into logical schemas. By employing these techniques, we can create efficient, organized, and reliable database systems that meet our data storage and retrieval needs.



CONCLUSION

- ER Diagrams offer several advantages. They provide a clear and intuitive visual representation of the database structure, aiding communication between stakeholders. They help in identifying potential issues and anomalies during the design phase, allowing for better database quality. However, ER Diagrams also have their disadvantages, such as the potential for complexity in large systems and the need for specialized software for design and maintenance.
- In summary, ER Diagrams are a valuable tool for database designers, allowing them to represent data models effectively, communicate ideas clearly, and create robust database systems. Understanding the components, techniques, advantages, and disadvantages of ER Diagrams is crucial for successful database design and management.



REFERENCES

1. Connolly. T., Begg. Carolyn. 2015. Database System: A Pratical Approach to Design, Implementation, and Management. Sixth Edition. Global Edition. Pearson.



Institut Informatika & Bisnis
DARMAJAYA
Yayasan Alfian Husin



**Kampus
Merdeka**
INDONESIA JAYA

**MERDEKA
BELAJAR**

THANK YOU!!

DATA SCIENCE DARMAJAYA "YOUR BEST FUTURE IN DATA"