

Database

Introduction To SQL Programming

Day1

- DB Life cycle
- File Based System & its Disadvantages and Limitations
- DBMS Advantages & Disadvantages
- ERD Notations
- Entities & Attributes & relations
- Keys & Constraints
- Case Study

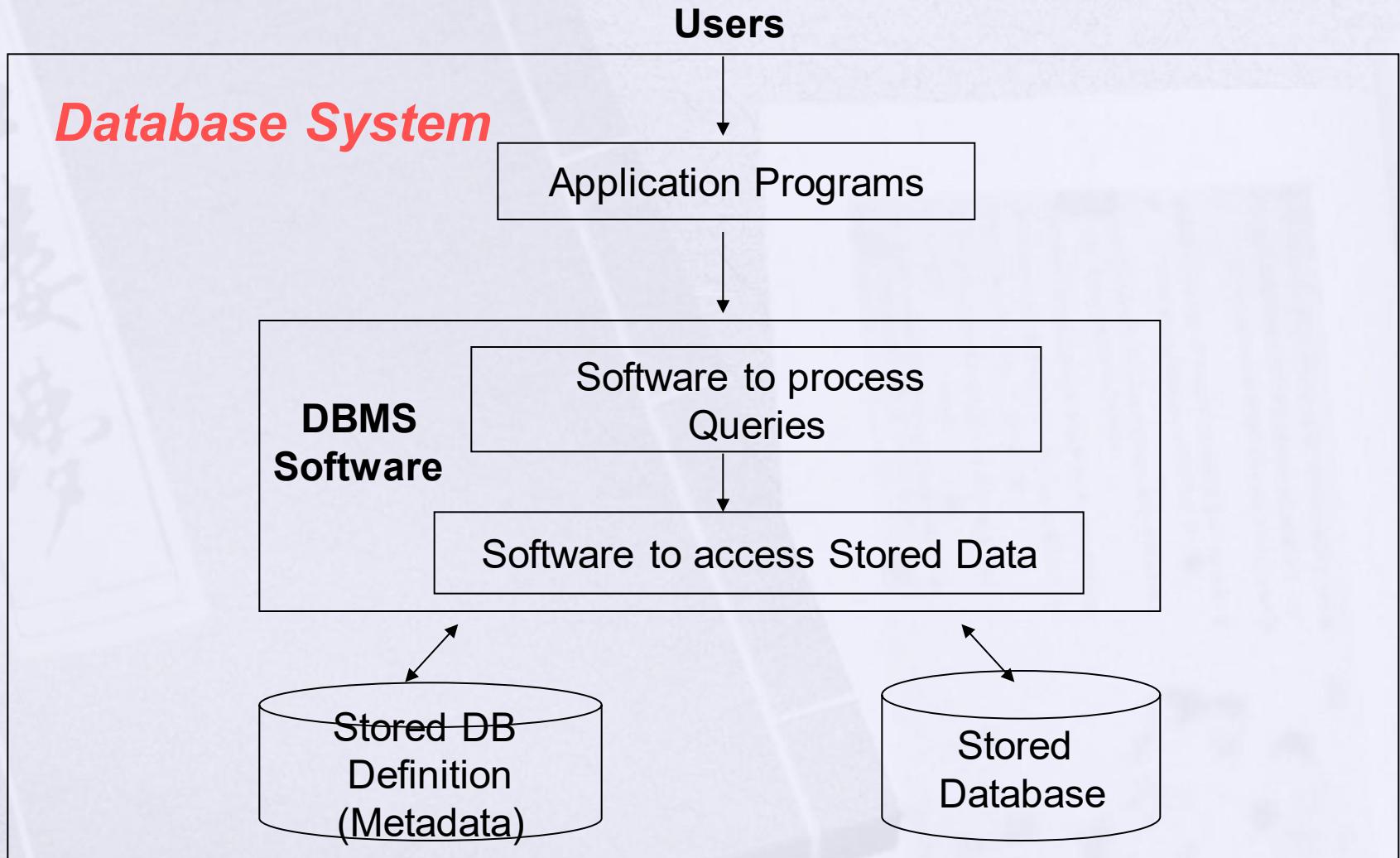
File Based System

- **Separation & Isolation Of data (each user has a copy)** cause **inconsistencies**
- **Incompatible File Formats**
- **Program-Data Dependence**
 - All programs maintain **metadata** for each file they use
 - **Each application** program needs to **include** code for the **metadata** of each file
 - **Non-standard** file formats
- **Lengthy Development Times**
 - Programmers must design their own file formats (**Metadata**)
- **Data Redundancy (Duplication of data)**
 - Different systems/programs have separate copies of the same data
 - When data changes in one file, could cause **inconsistencies**
 - **No Database integrity**
- **Limited Data Sharing**
 - **No centralized** control of data

Basic Definitions

- **Database:** A collection of related data.
- **Database Management System (DBMS):** A software package/ system to facilitate the creation and maintenance of a computerized database.(model introduced in 1970 IBM but RDBMS appears in 1980)
- **Database System:** The DBMS software together with the data itself. Sometimes, the applications are also included. (**Software + Database**)

Database System



DBMS Advantages

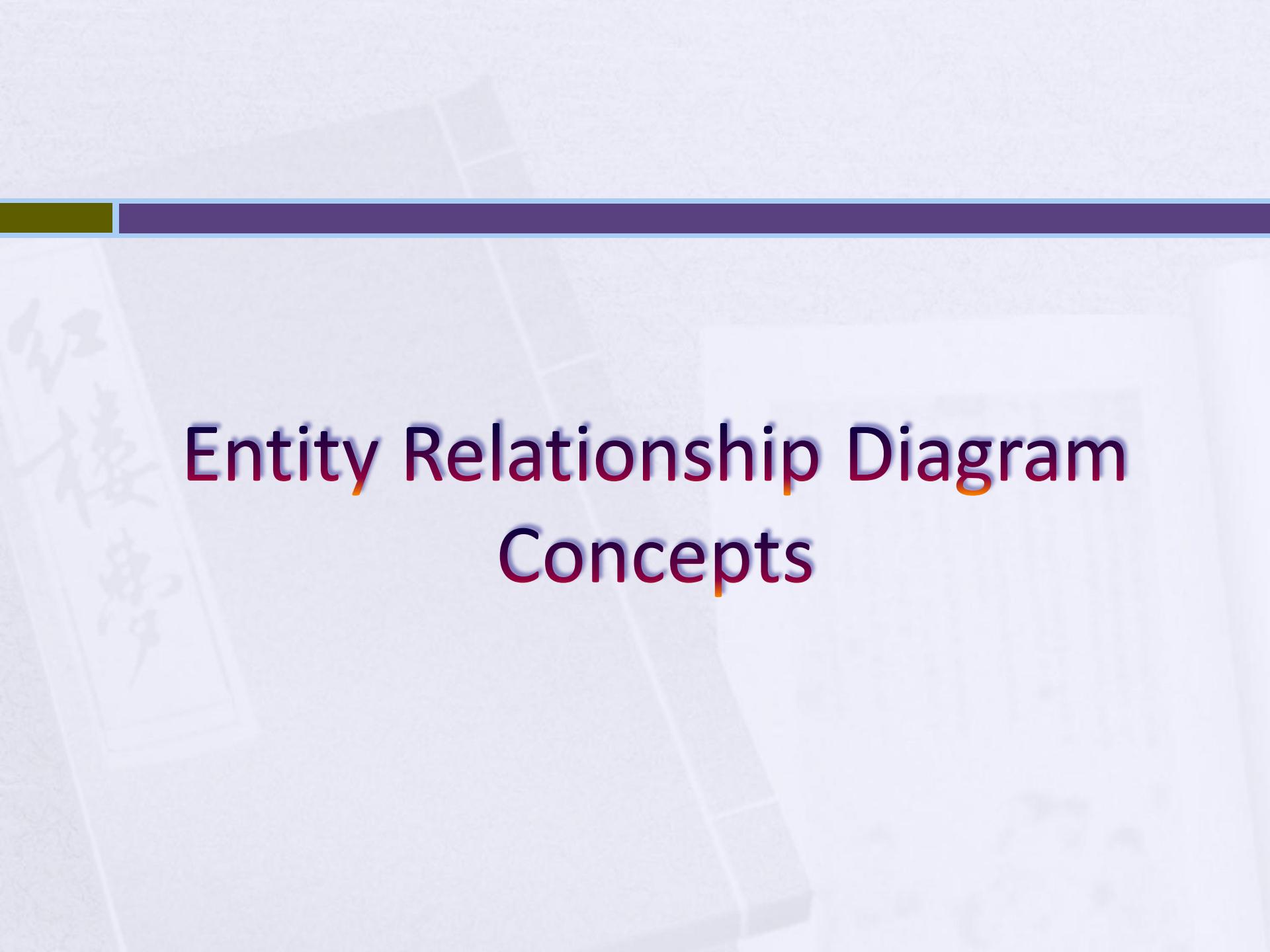
- **Standardization** and better Data accessibility and response (SQL)
- **Sharing data.**
 - Different users get different views of the data
- **Enforcing Integrity Constraints**
- **Improved Data Quality**
 - Constraints, data validation rules
- **Inconsistency can be avoided because of data sharing.**
- **Restricting Unauthorized Access.**
- **Providing Backup and Recovery.**
 - Disaster recovery is easier
- **Minimal Data Redundancy**
 - Leads to increased data integrity/consistency
- **Program-Data Independence**
 - Metadata stored in DBMS, so applications don't worry about data formats
 - Data queries/updates managed by DBMS

DBMS Disadvantages

- It needs **expertise** to use
- DBMS itself is **expensive**
- The DBMS may be **incompatible** with any other available **DBMS**

Database Users

- Database Administrator (DBA)
- System Analysts
- Database Designer
- Database Developer
- Application programmers
- BI & BigData Specialist (Data Scientist)
- End users



The slide features a decorative background with faint, light-colored geometric shapes like circles and triangles. A solid blue horizontal bar runs across the top of the slide. In the top-left corner, there is a small vertical bar composed of three horizontal segments: a dark green segment at the top, a dark purple segment in the middle, and a light blue segment at the bottom.

Entity Relationship Diagram Concepts

Entity Relationship Modeling

Entity-Relationship Diagram (ERD)

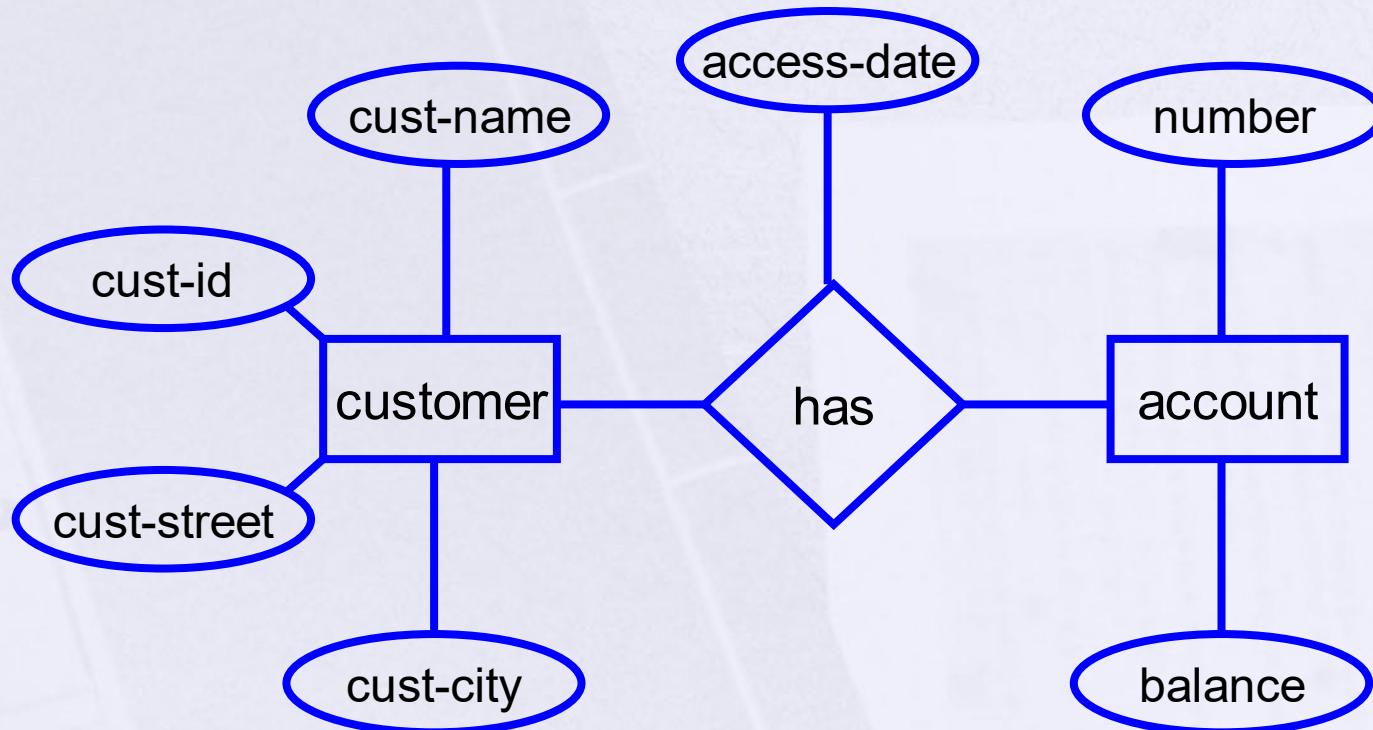
identifies information required by the business by displaying the relevant entities and the relationships between them.

The ER Model

Basic constructs of the E-R model:

1. **Entities** - person, place, object, event, concept (often corresponds to a real time object that is **distinguishable** from any other object)
2. **Attributes** - property or **characteristic** of an entity type (often corresponds to a field in a table)
3. **Relationships** - **link** between entities (corresponds to primary **key-foreign key** equivalencies in related tables)

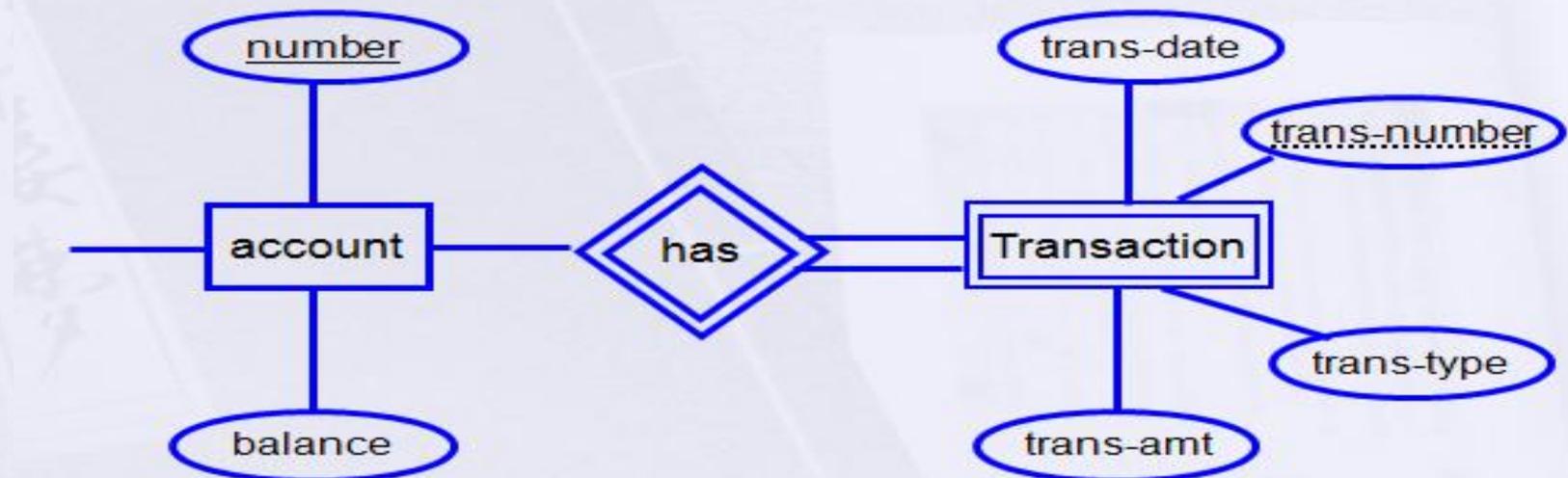
ER Diagram: Starting Example



- ▶ Rectangles: entity sets
- ▶ Diamonds: relationship sets
- ▶ Ellipses: attributes

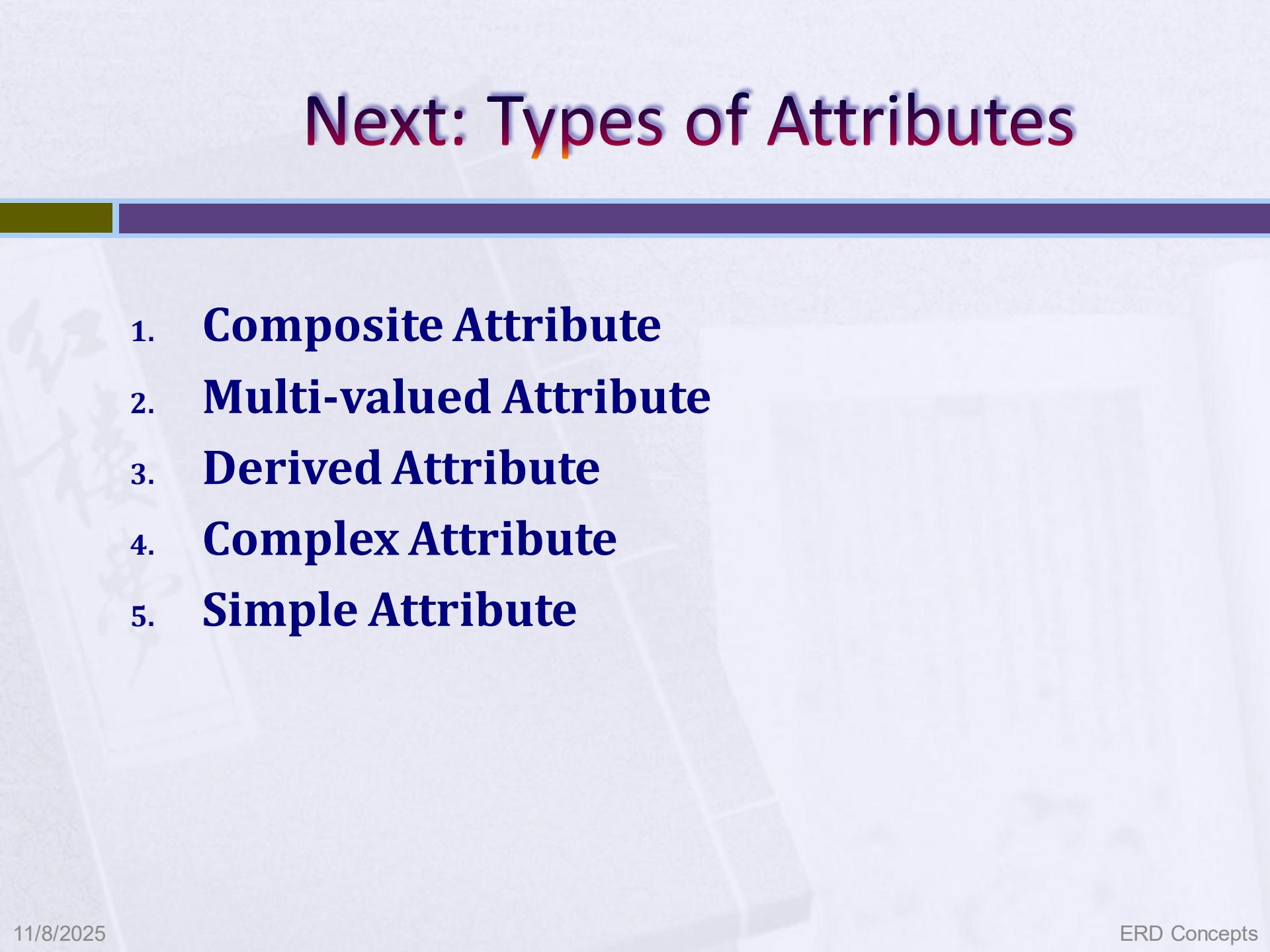
Strong Entity Vs Weak Entity

- A **Strong Entity**- An Entity set that has a primary key.
- A **Weak Entity**- An entity set that do not have sufficient attributes to form a primary key.

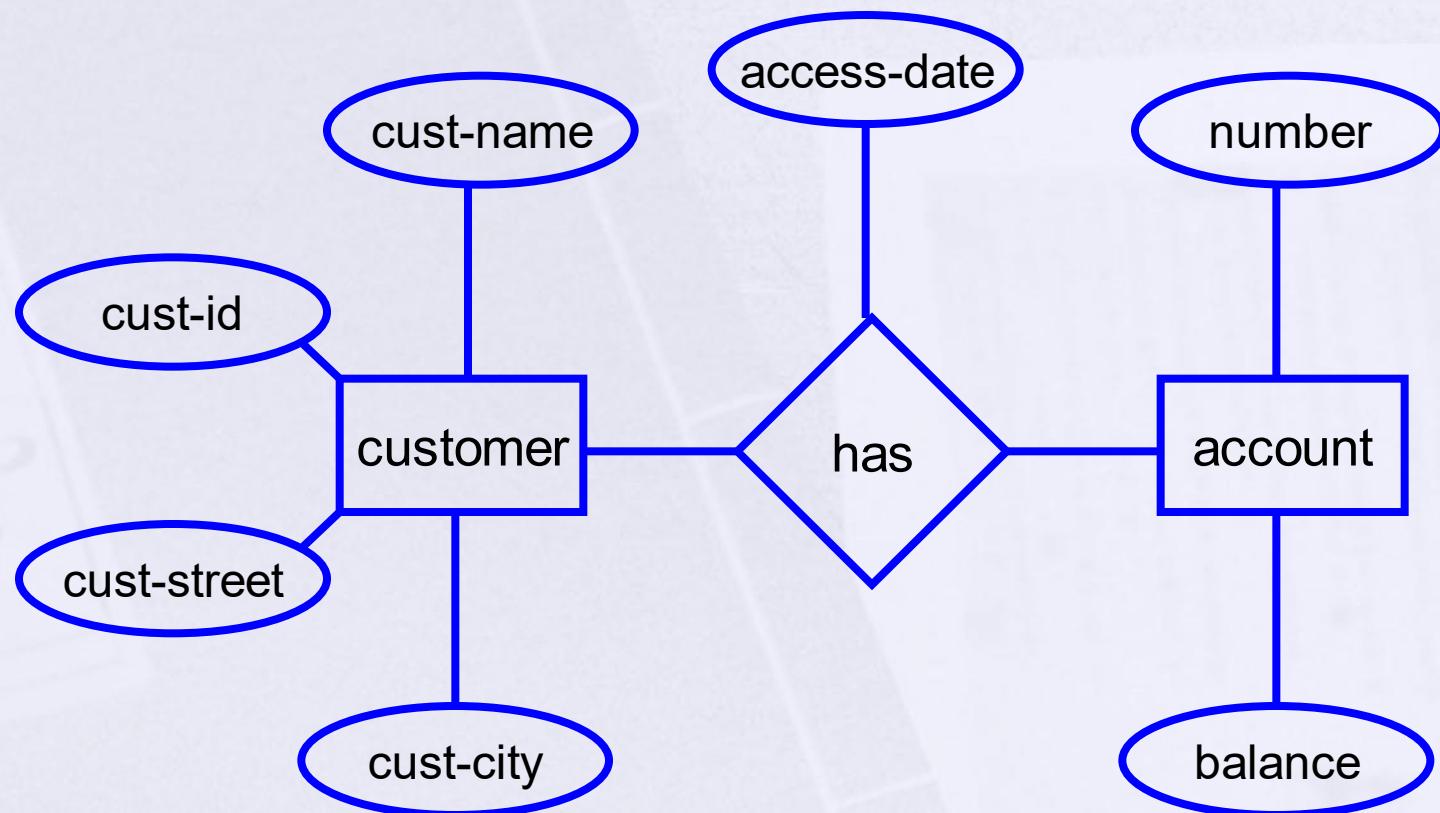


Partial key: A set of attributes that can be associated with P.K of an owner entity set to distinguish a weak entity.

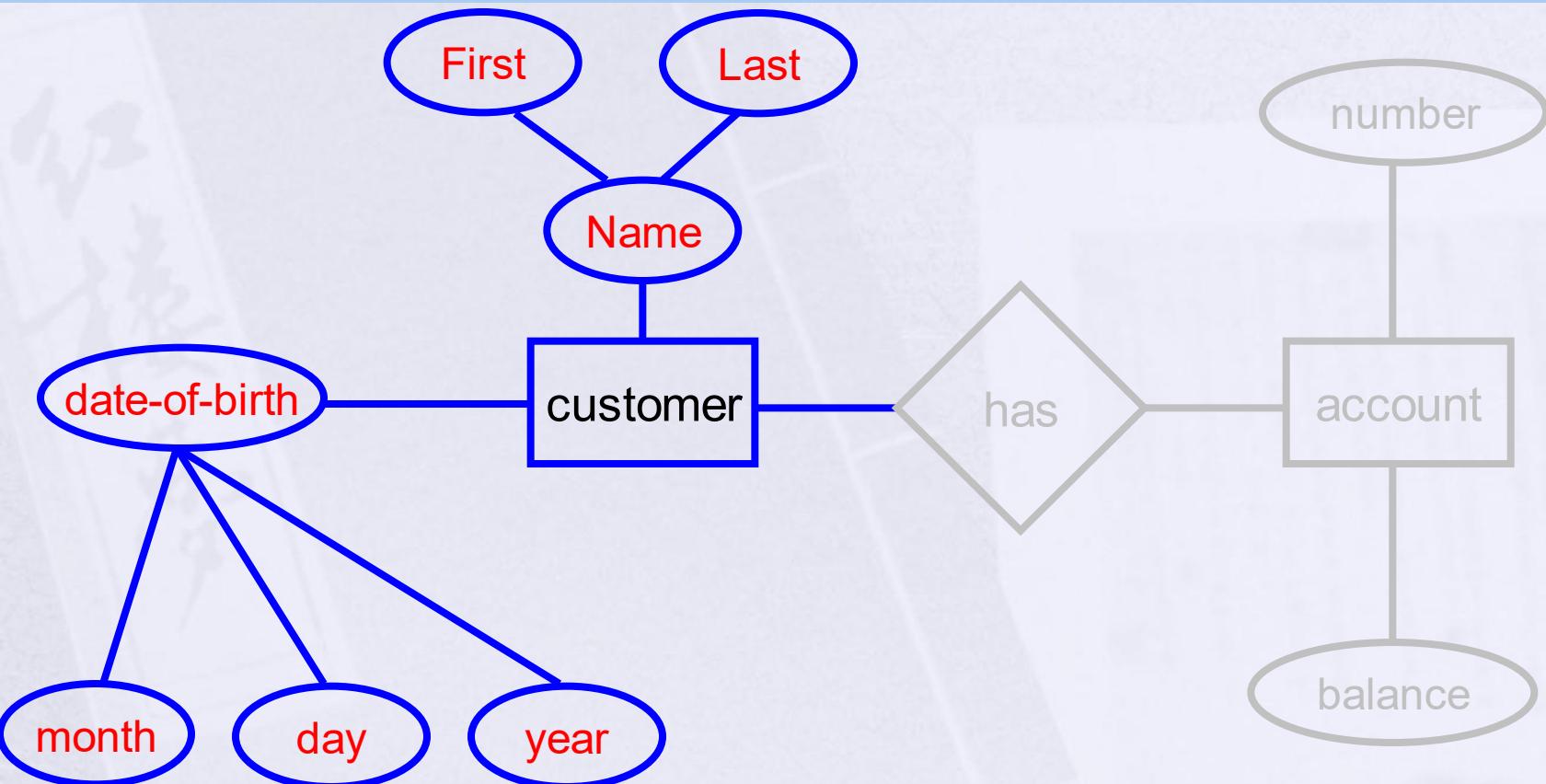
Next: Types of Attributes

- 
1. **Composite Attribute**
 2. **Multi-valued Attribute**
 3. **Derived Attribute**
 4. **Complex Attribute**
 5. **Simple Attribute**

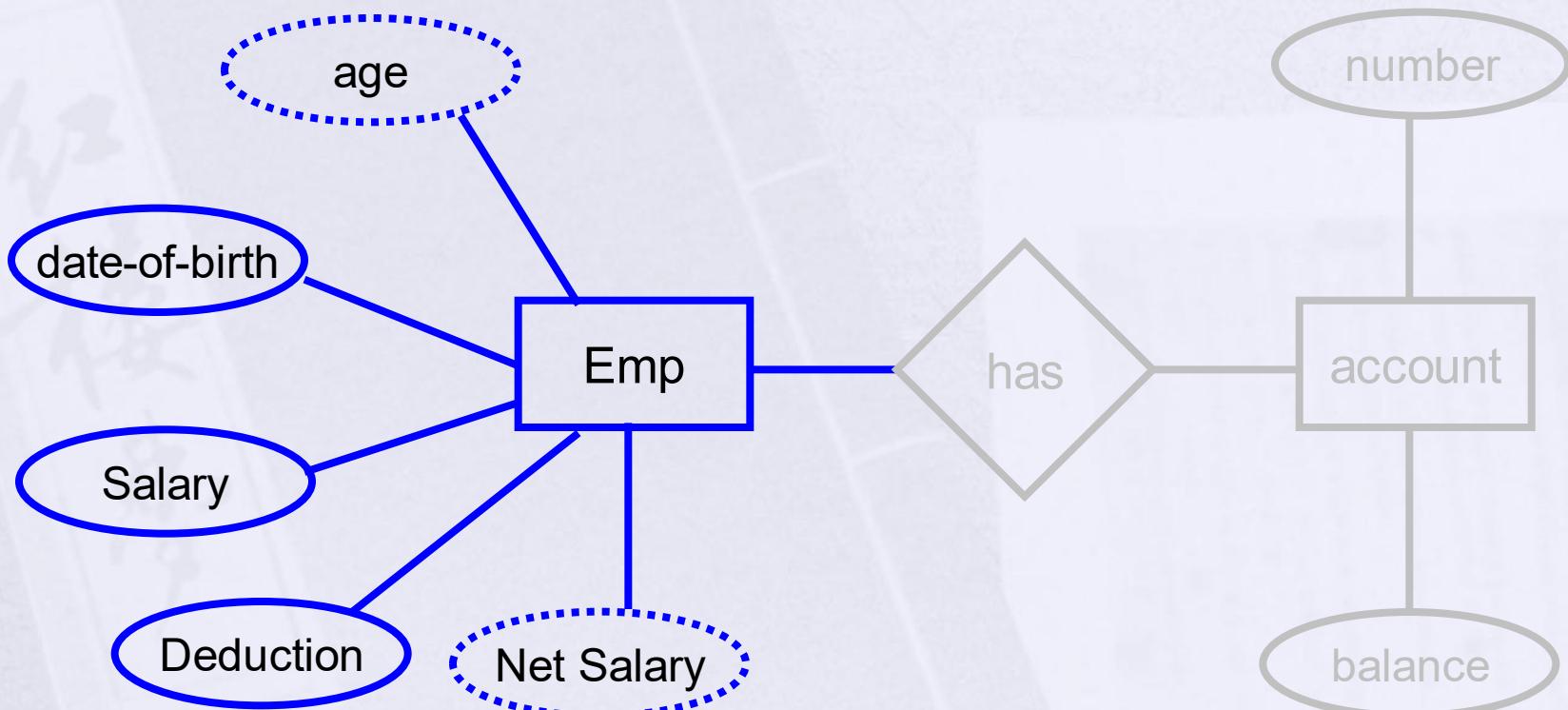
Simple Attribute



Composite Attribute

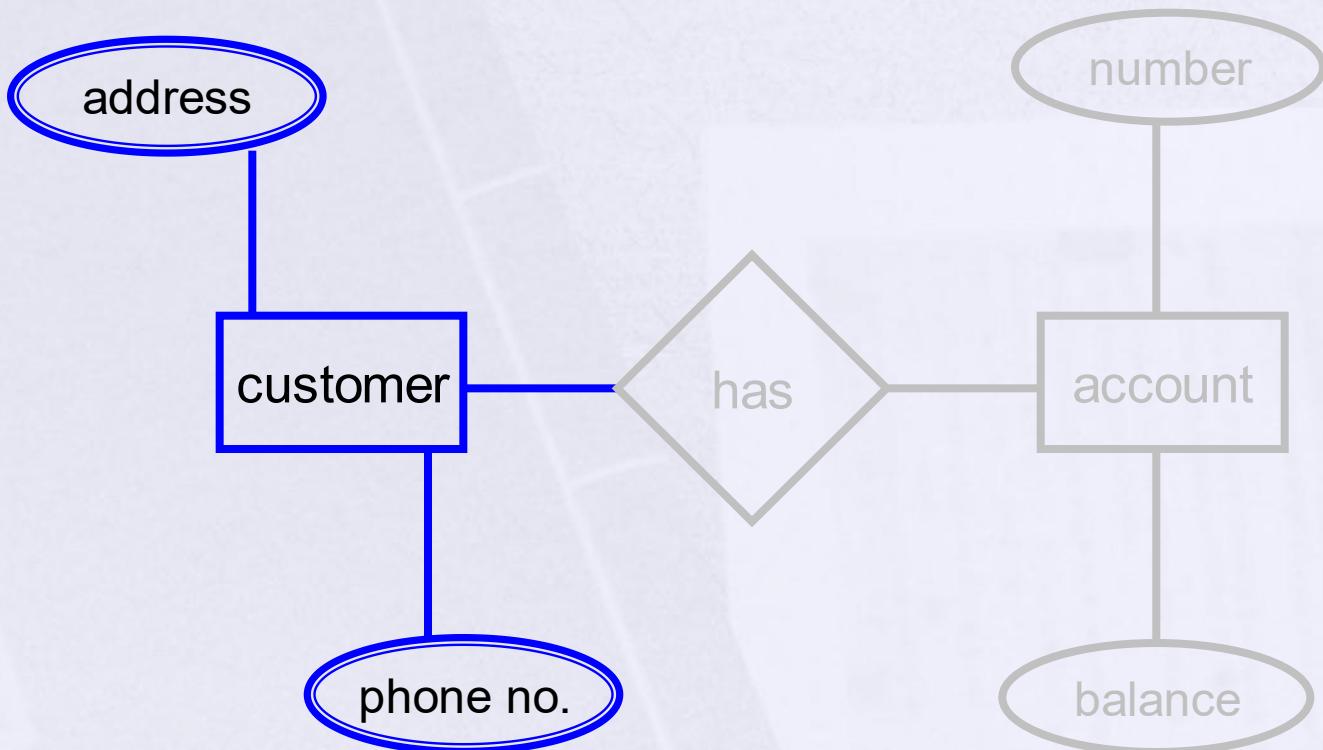


Derived Attribute



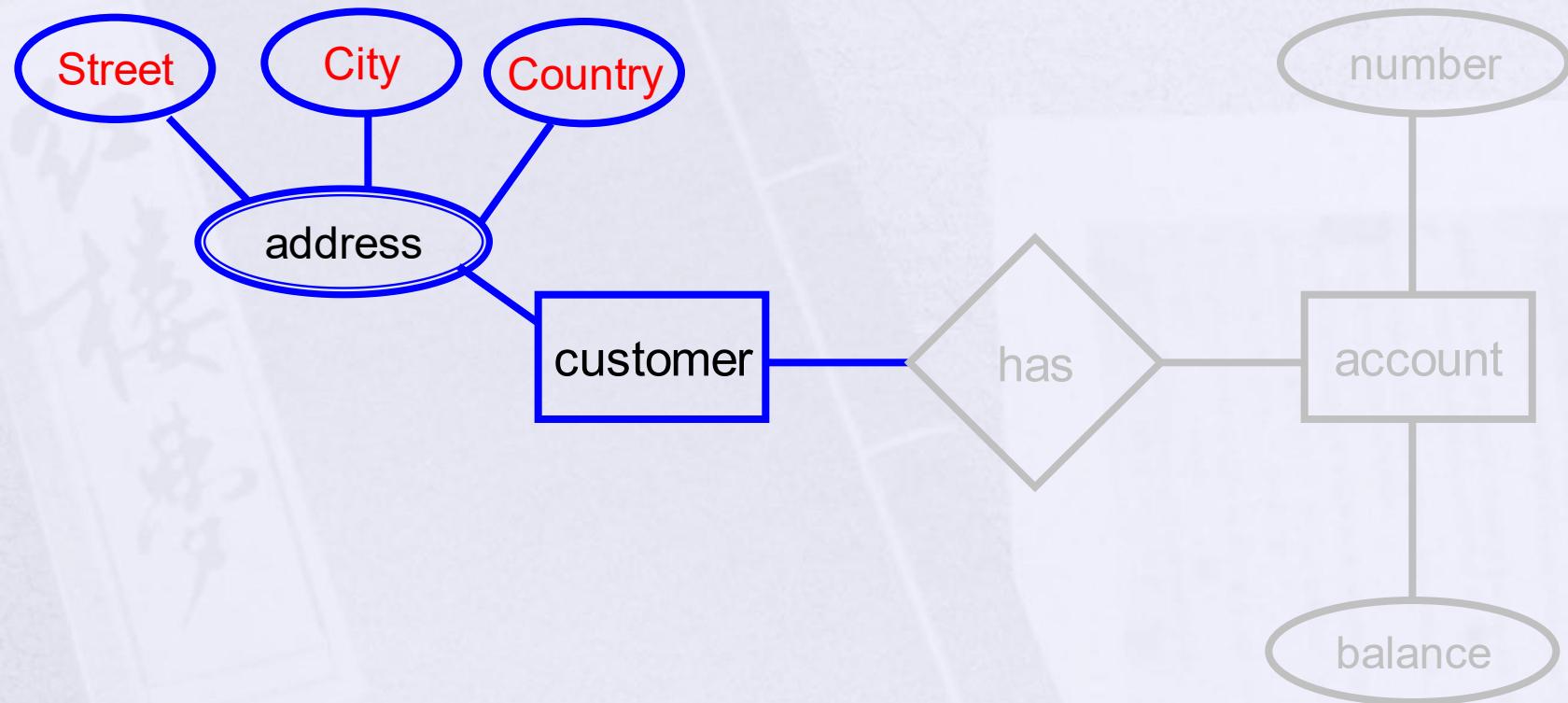
➤ derived (dashed ellipse)

Multi-valued



➤ **multi-valued (double ellipse)**

Complex Attribute



➤ multi-valued + Composite

Relationship

- A Relationship is an association among several entities.
- A relationship may also have attributes

For example, consider the entity sets customer and loan and the relationship set borrower. We could associate the attribute **date-issued** to that relationship to specify the date when the loan was issued.

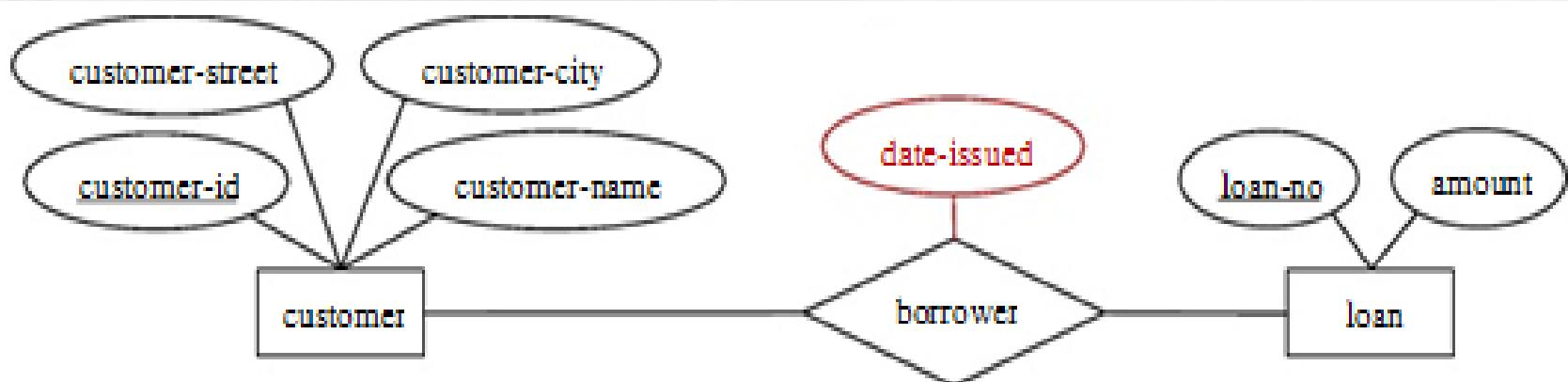


Figure: Descriptive attribute *date-issued*.

Relation

Relation has three Properties:

- Degree of Relationships
- Cardinality Constraint
- Participation Constraint

Degree of Relationships

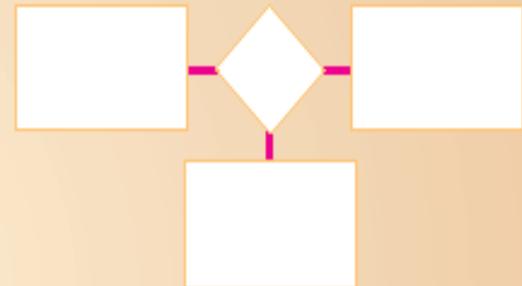
- Degree: number of entity types that participate in a relationship
- Three cases
 - **Unary:** between two instances of one entity type
 - **Binary:** between the instances of two entity types
 - **Ternary:** among the instances of three entity types



Unary



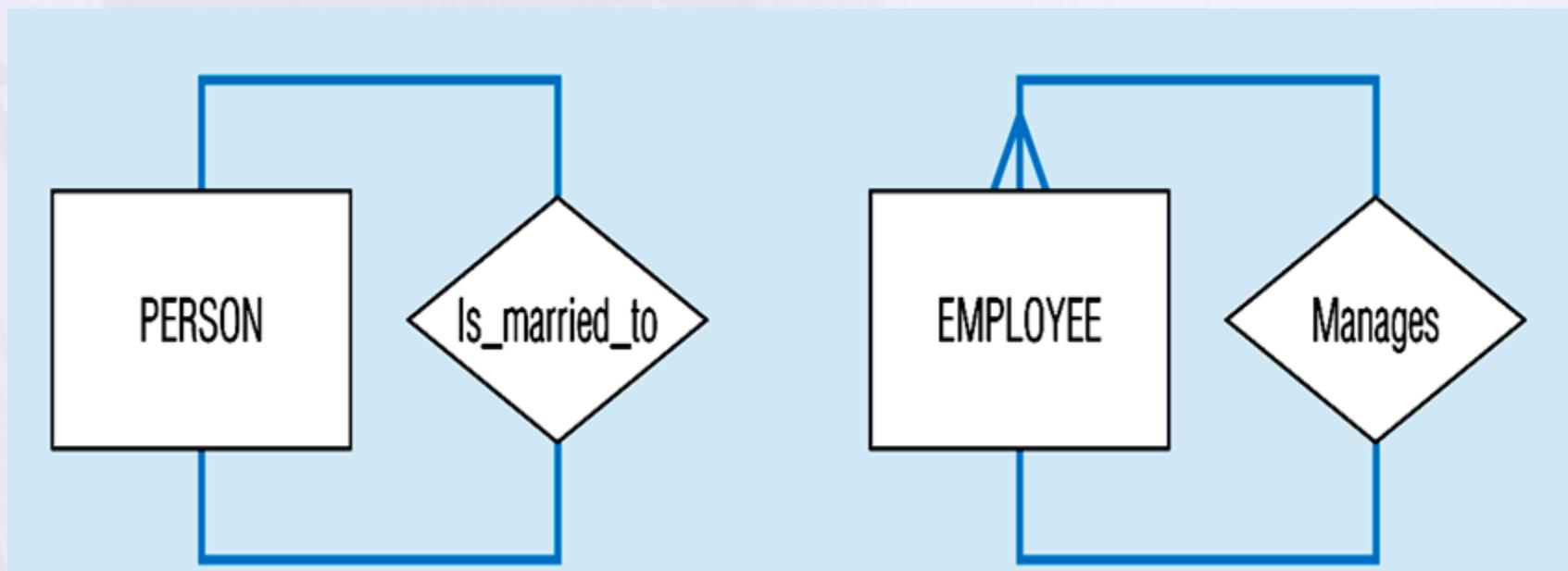
Binary



Ternary

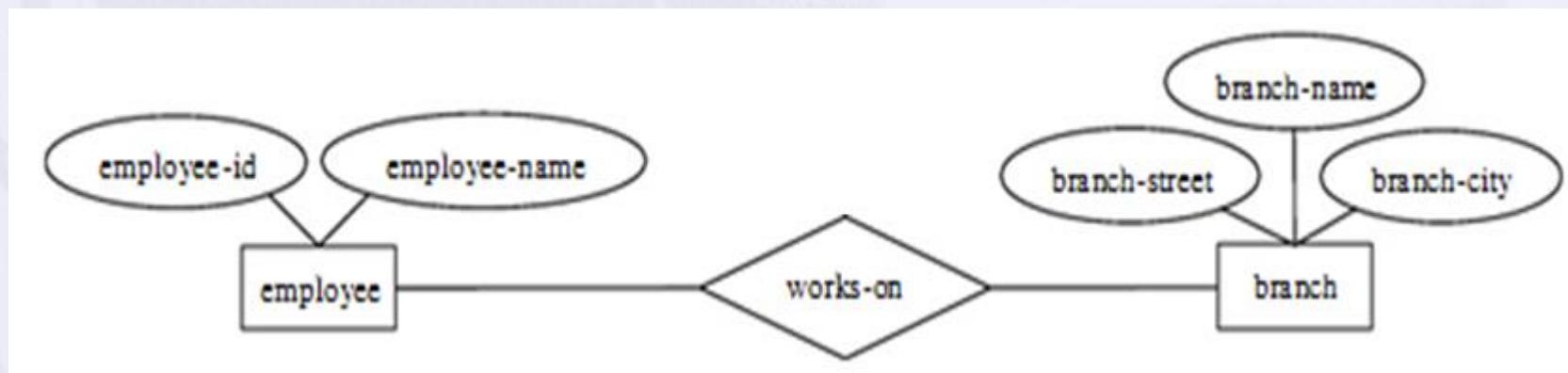
Recursive Relationship (Unary)

- **Recursive Relationships** - A relationship in which the same entity participates more than once.



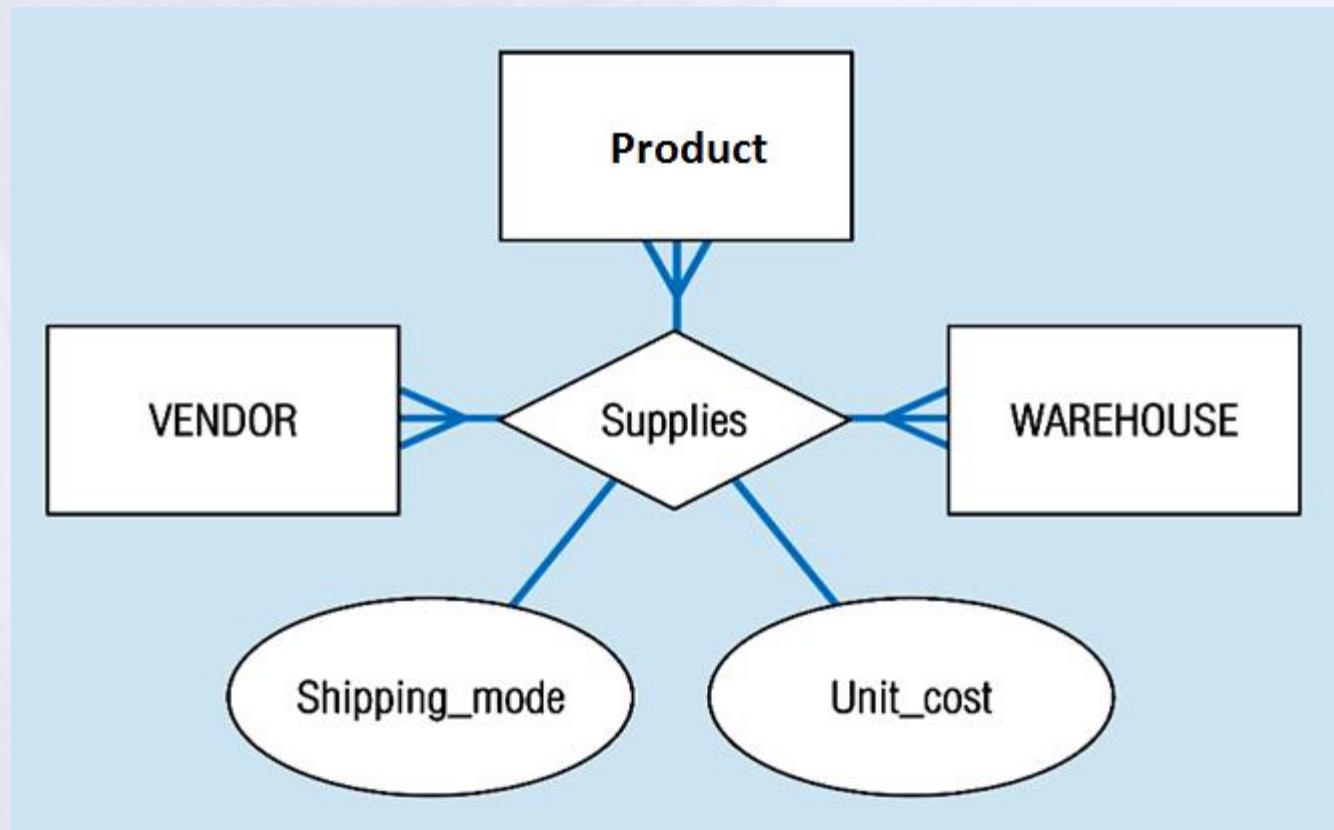
Binary Relationship

- A binary relationship set is of degree 2.



Ternary Relationship

- ▶ ternary relationship set is of degree 3.

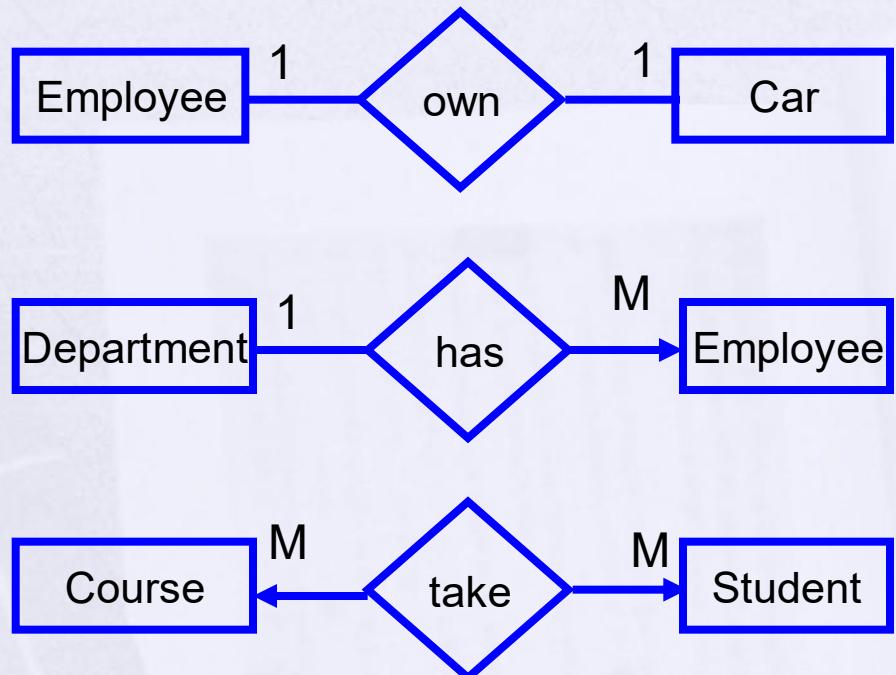


Cardinality

- How many instances of one entity will or must be connected to a single instance from the other entities.
 - **One-One Relationship**
 - **One-Many Relationship**
 - **Many- Many Relationship**

Mapping Cardinalities

- One-to-One
- One-to-Many
- Many-to-Many

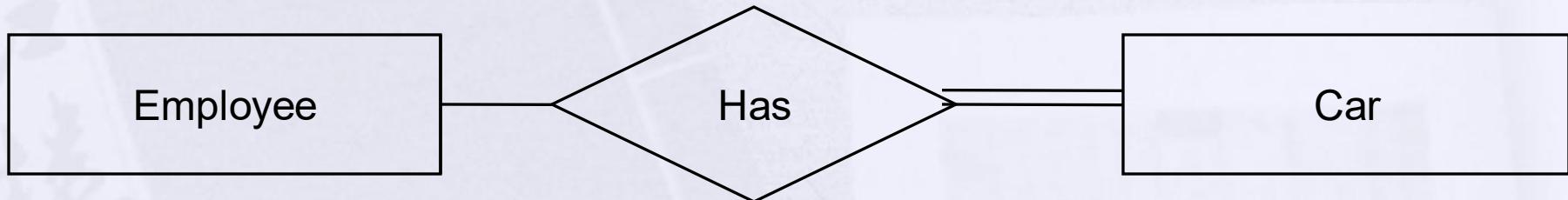


PARTICIPATION CONSTRAINT

- An employee MUST work for a department
An employee entity can exist only if it participates in a
WORKS_FOR relationship instance
So this participation is TOTAL

Only some employees manage departments
The participation is PARTIAL

PARTICIPATION CONSTRAINT



- An Employee **may** have a car.
- A Car **must** be assigned to particular employee

PARTICIPATION CONSTRAINT



- A department **may** hire many employees (Zero or more)
 - An employee **must** be employed by a department
- (Department membership is **Optional**, Employee membership is **Mandatory**)

Keys

- Different Types of Keys:
 1. Candidate Key
 2. Primary Key
 3. Foreign Key
 4. Composite Key
 5. Partial Key
 6. Alternate key
 7. Super Key

Candidate Key

Candidate key: is a set of one or more attributes whose value can uniquely identify an entity in the entity set

- Any attribute in the candidate key cannot be omitted without destroying the uniqueness property of the candidate key.

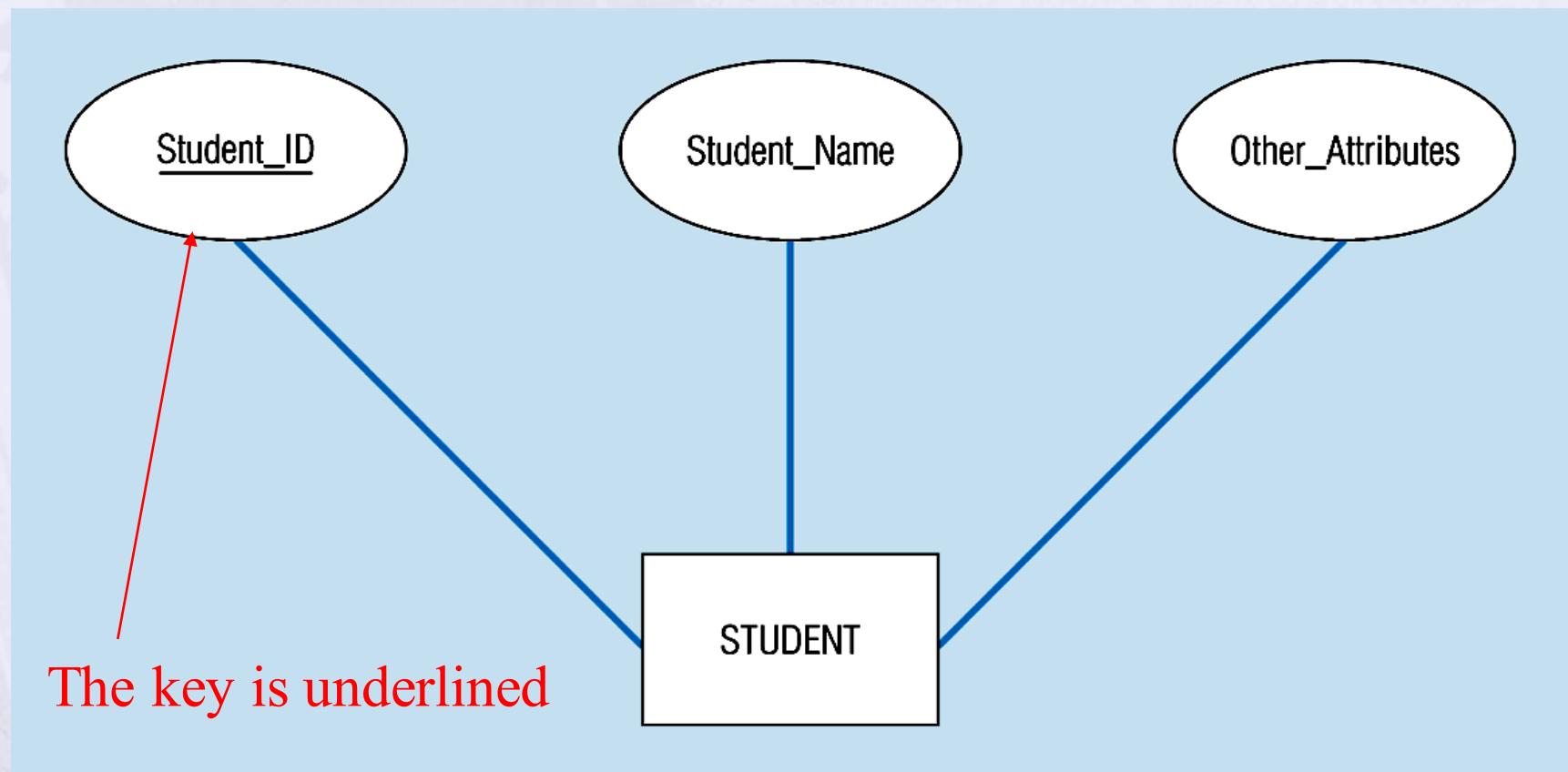
Example:

- $(SSN, Name)$ is NOT a candidate key .
- “ SSN ” is a candidate key of ***customer***.
- Candidate key could have more than one attributes.

Primary Key

- **Example:** Both “SSN” and “License #” are candidate keys of *Driver* entity set.
- **Primary Key:** is the candidate key that is chosen by the database designer as the unique identifier of an entity.
[Unique & Not Null]
- **Primary key May be Composite**

Primary Key



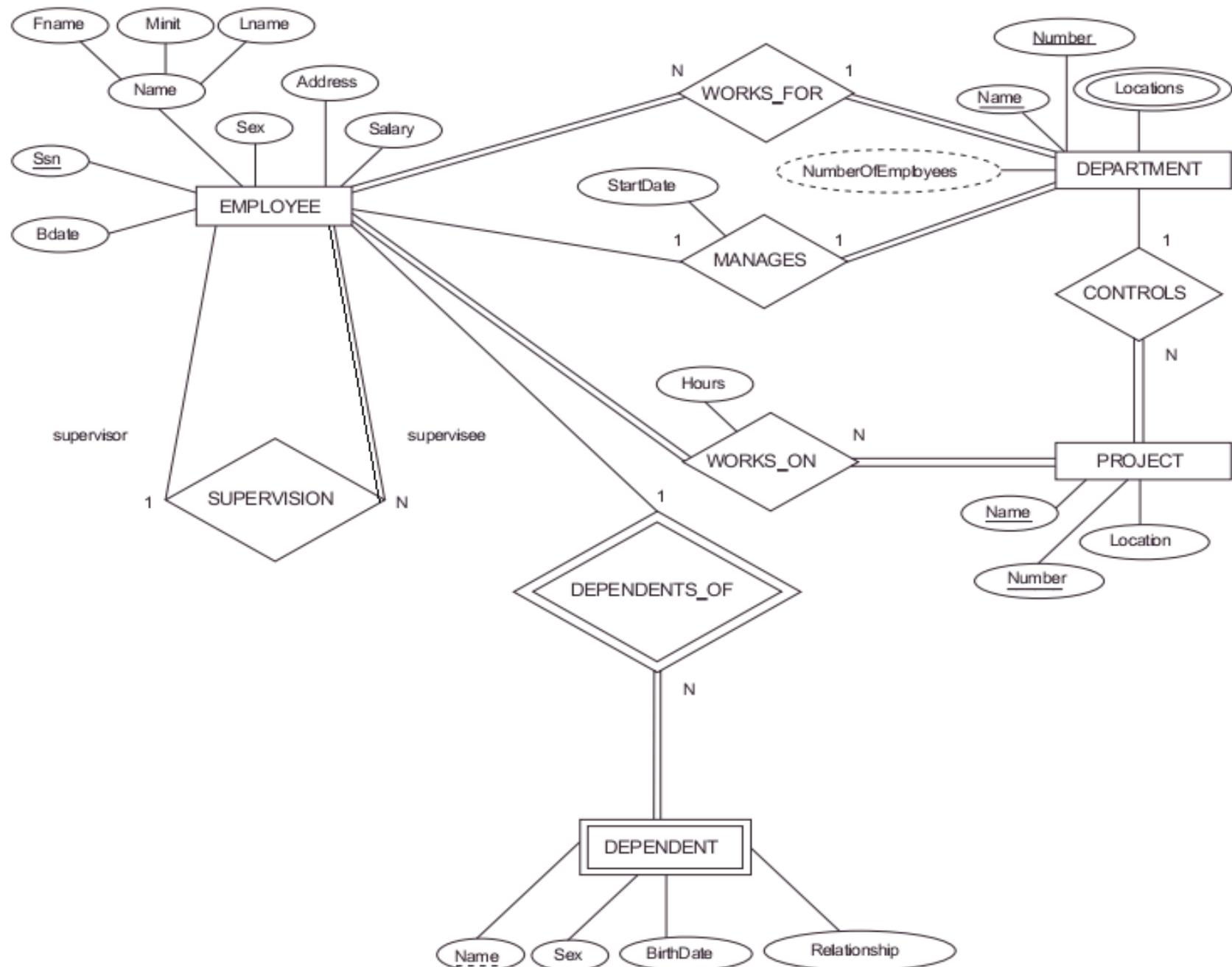
Summary of notation for ER diagrams

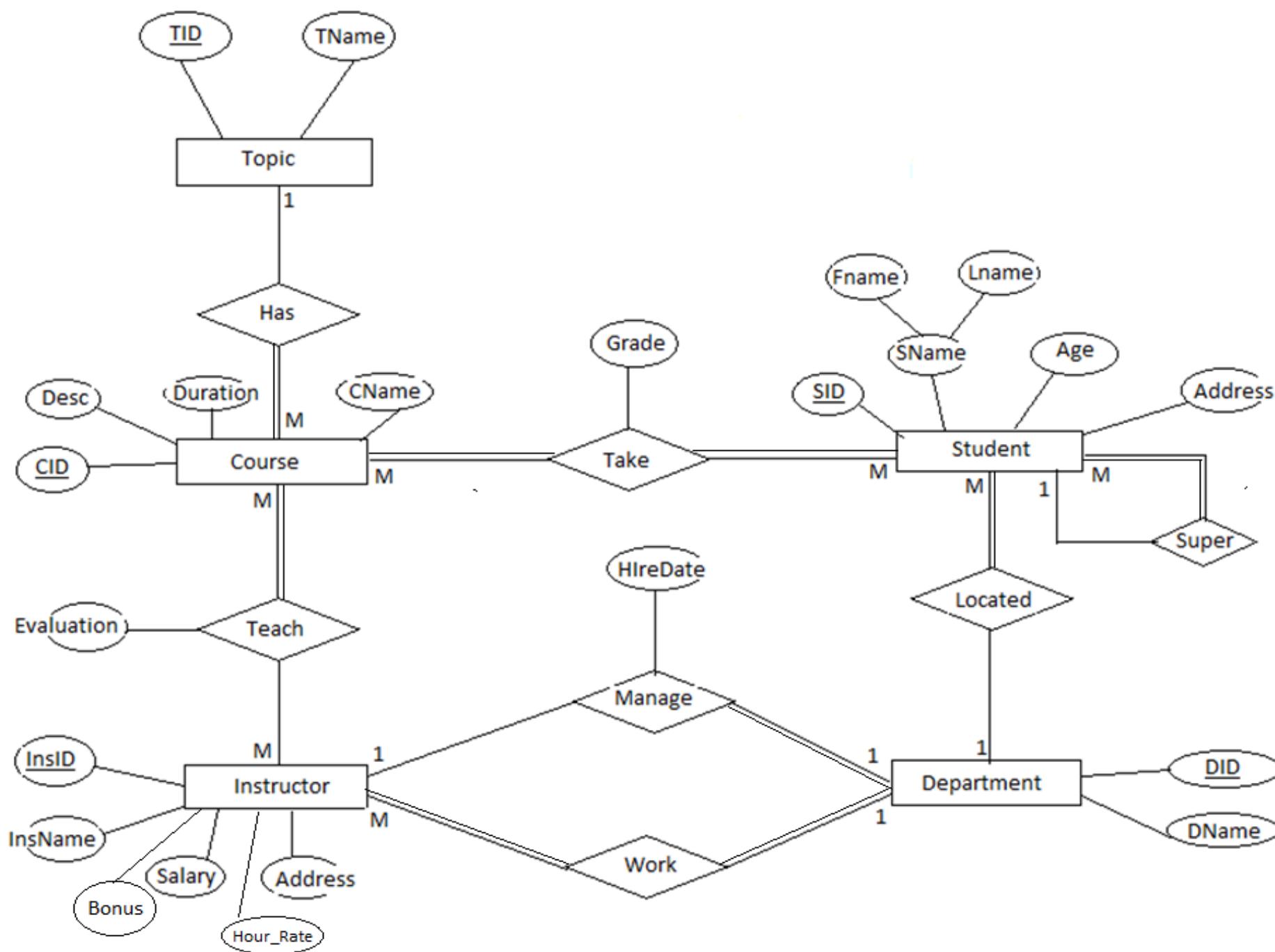
Figure 3.14
Summary of the
notation for ER
diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1 : N for $E_1:E_2$ in R
	Structural Constraint (min, max) on Participation of E in R

Identifying relationship is links strong entities to weak entities and represented with double line diamond

Case Study





Thank You !!!