Data Modeling

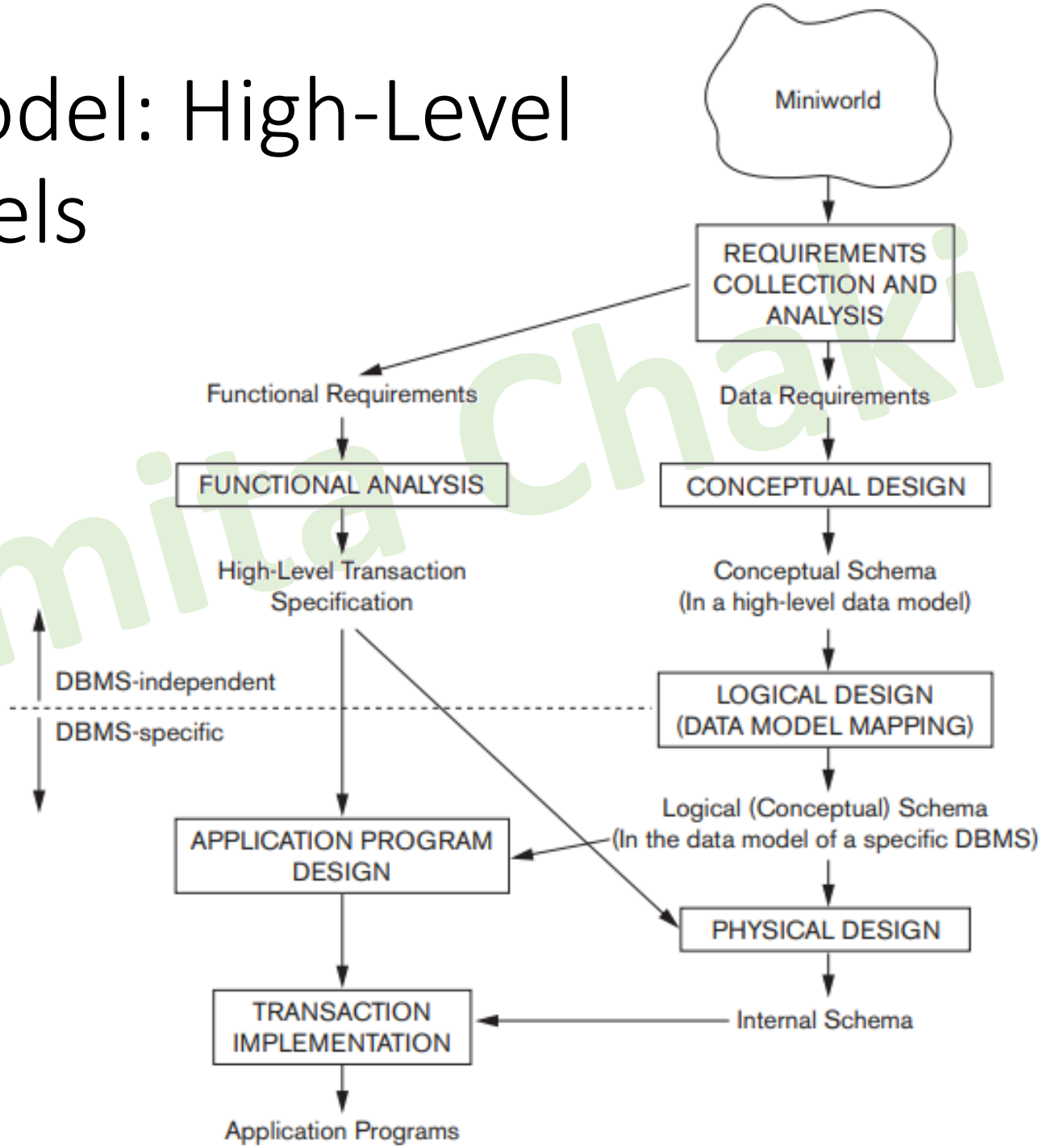
Dr. Jyotismita Chaki

Entity Relationship Model

- **Entity–relationship (ER) model** is a popular high-level conceptual data model.
  - Conceptual modeling is a very important phase in designing a successful database application.
  - Generally, the term database application refers to a particular database and the associated programs that implement the database queries and updates.
- ER model and its variations are frequently used for the conceptual design of database applications, and many database design tools employ its concepts.
- The diagrammatic notation associated with the ER model, known as ER diagrams

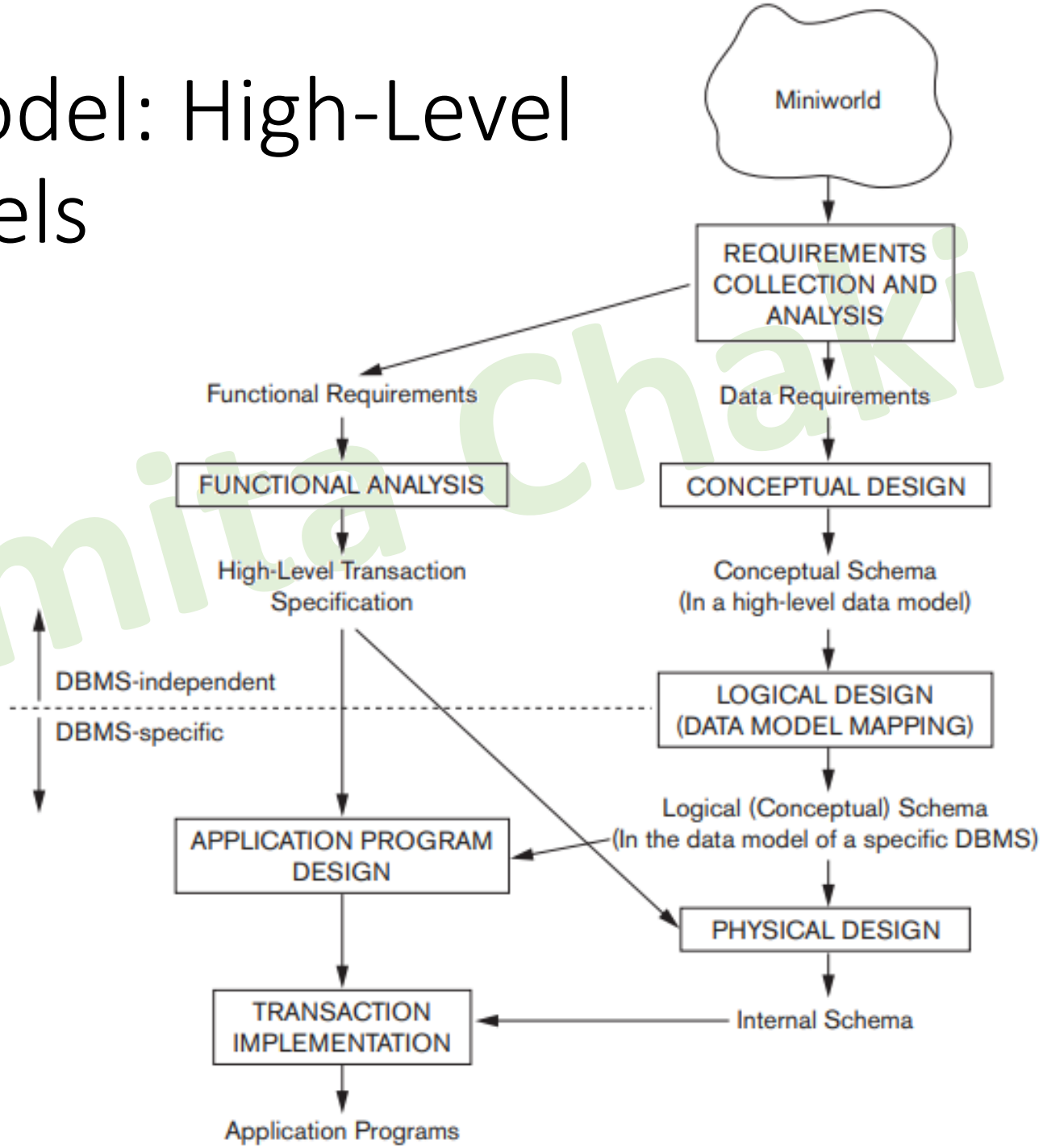
Entity Relationship Model: High-Level Conceptual Data Models

- Requirements collection and analysis: the database designers interview prospective database users to understand and document their **data requirements**
- **Functional requirements**: user defined operations (or transactions) that will be applied to the database
- Conceptual design: create a conceptual schema is a concise description of the data requirements of the users



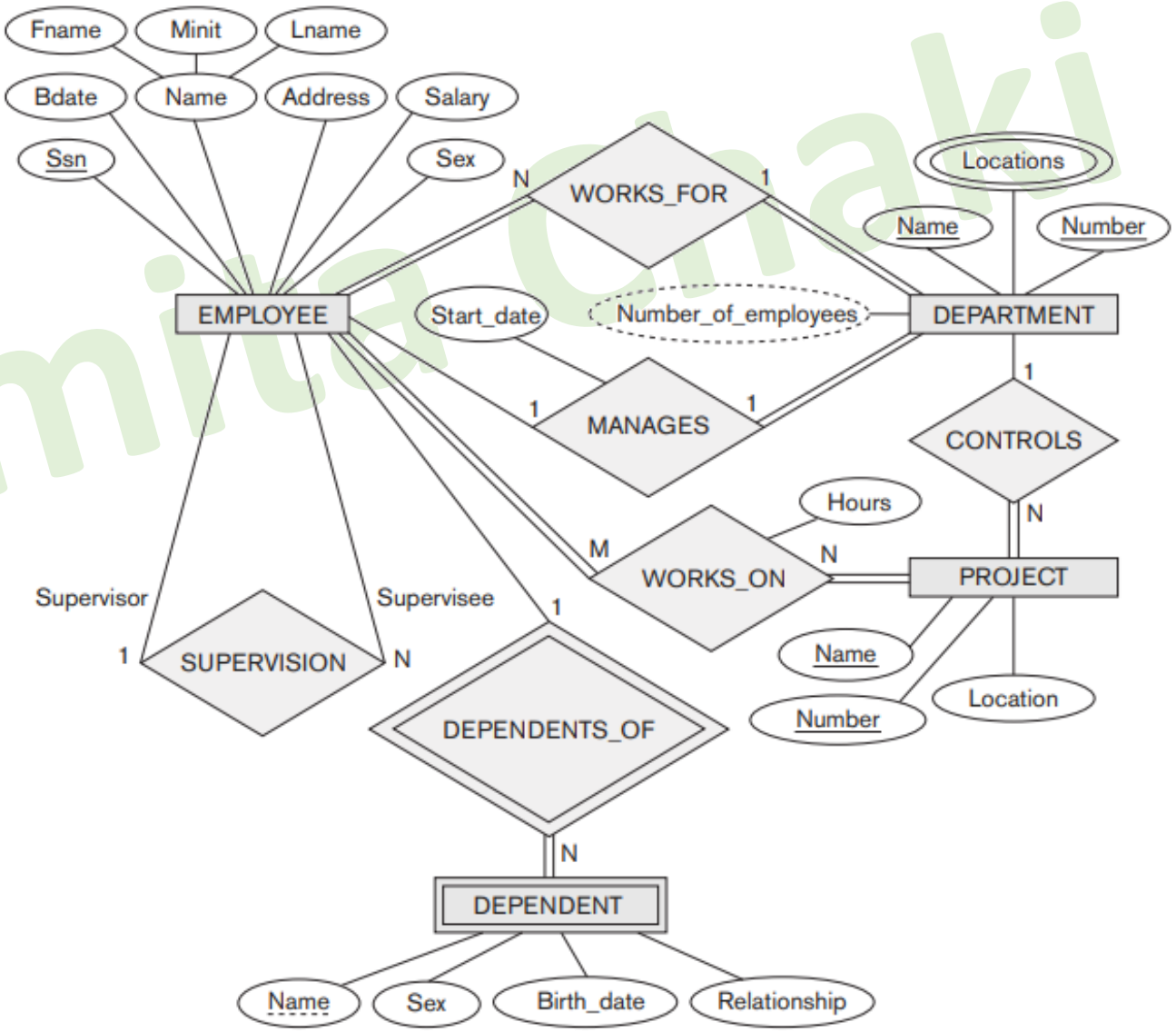
Entity Relationship Model: High-Level Conceptual Data Models

- **Logical design or data model mapping**: the conceptual schema is transformed from the high-level data model into the implementation data model.
- **Physical design**: the internal storage structures, file organizations, indexes, access paths, and physical design parameters for the database files are specified.



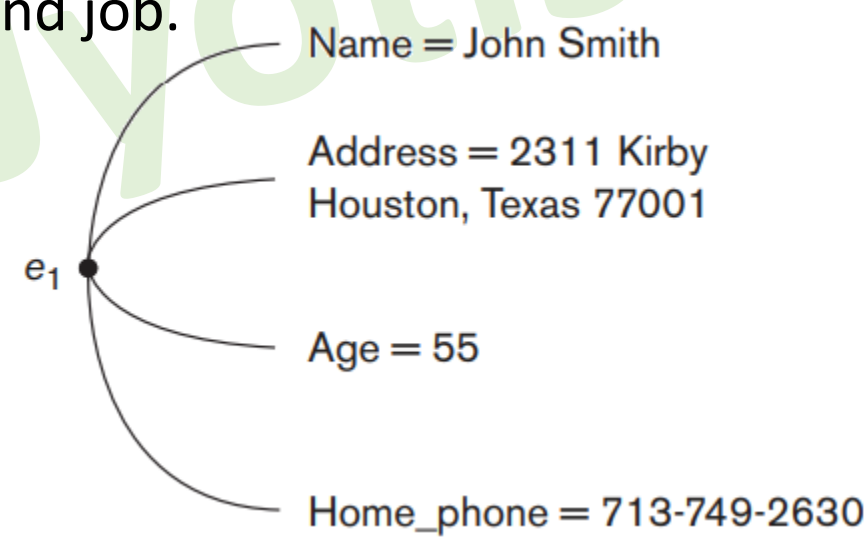
Entity Relationship Model: Example

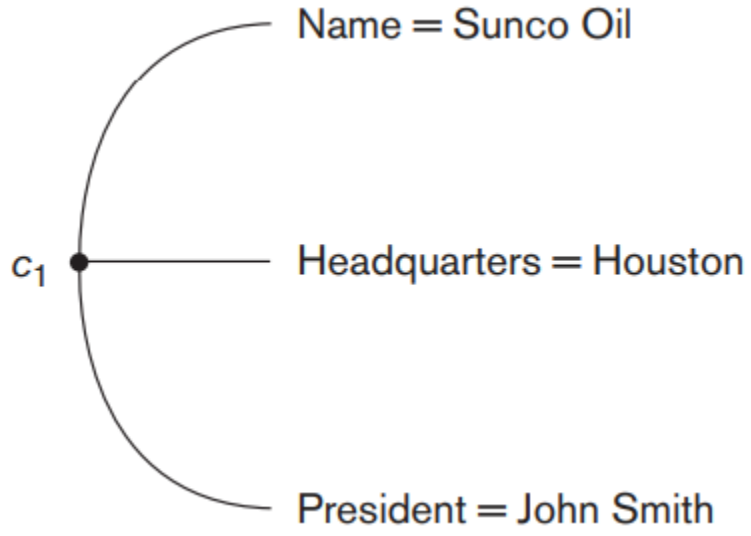
- COMPANY database keeps track of a company's employees, departments, and projects.
- The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department.
- A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
- The database will store each employee's name, Social Security number,2 address, salary, sex (gender), and birth date.
- An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department.



Entity Relationship Model: Entities and attributes

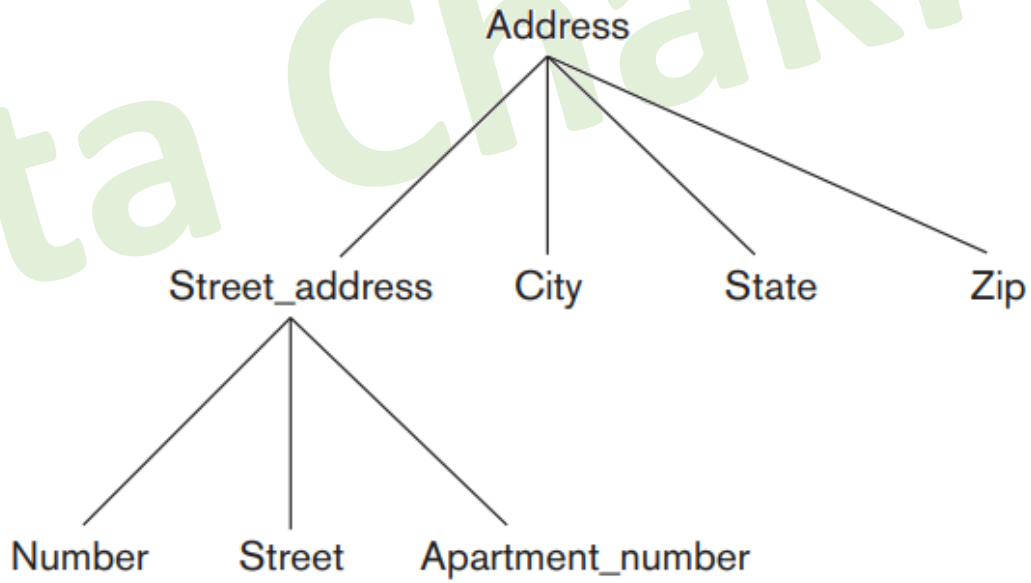
- The ER model describes data as
  - Entities: a thing or object in the real world with an independent existence. For example, a particular person, car, house, or employee.
  - Relationships, and
  - Attributes: the particular properties that describe it. For example, an EMPLOYEE entity may be described by the employee's name, age, address, salary, and job.





Entity Relationship Model: Types of attributes: Simple versus Composite

- **Composite attributes** can be divided into smaller subparts, which represent more basic attributes with independent meanings.
- Attributes that are not divisible are called **simple or atomic attributes**.
- If the composite attribute is referenced only as a whole, there is no need to subdivide it into component attributes.
  - If there is no need to refer to the individual components of an address (Zip Code, street, and so on), then the whole address can be designated as a simple attribute.



Entity Relationship Model: Types of attributes: Single valued versus Multivalued

- Most attributes have a single value for a particular entity; such attributes are called **single-valued**.
  - Age is a single-valued attribute of a person.
- When there are possibilities of multiple values for a particular entity; such attributes are called **multi-valued.**
  - A Colors attribute for a car, or A College_degrees attribute for a person.
- A multivalued attribute may have lower and upper bounds to constrain the number of values allowed for each individual entity.
  - The Colors attribute of a car may be restricted to have between one and two values

Entity Relationship Model: Types of attributes: Stored versus Derived

- In some cases, two (or more) attribute values are related—for example, the Age and Birth_date attributes of a person.
- The Age attribute is called a **derived attribute** and is said to be derivable from the Birth_date attribute, which is called a **stored attribute**.

Entity Relationship Model: Types of attributes: Complex

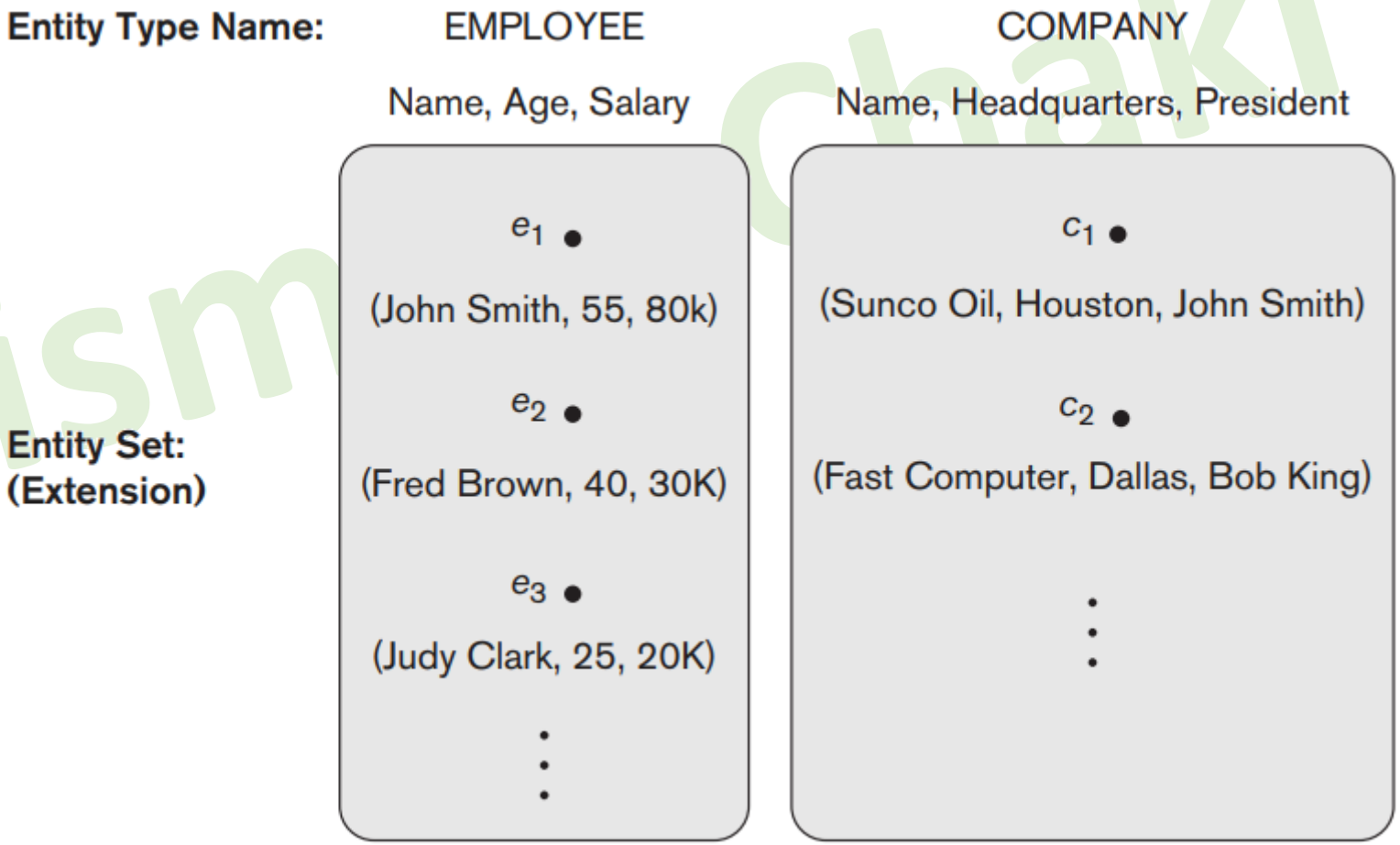
- Group of components of a composite attribute between parentheses ( ) and separating the components with commas, and by displaying multivalued attributes between braces { }.
  - {Address_phone({Phone(Area_code,Phone_number)},Address(Street_address (Number,Street,Apartment_number),City,State,Zip))}

Entity Relationship Model: Types of attributes: NULL

- In some cases, a particular entity may not have an applicable value for an attribute.
- For such situations, a special value called NULL is created.
  - A College_degrees attribute applies only to people with college degrees. A person with no college degree would have NULL for College_degrees
- NULL can also be used if we do not know the value of an attribute for a particular entity: Unknown
  - When it is known that the attribute value exists but is **missing**
  - When it is **not known** whether the attribute value exists

Entity Relationship Model: Entity Types, Entity Sets, Keys, and Value Sets

- An **entity type** defines a collection (or set) of entities that have the same attributes.
- Each entity type in the database is described by its name and attributes.
- The collection of all entities of a particular entity type in the database at any point in time is called an **entity set** or **entity collection**

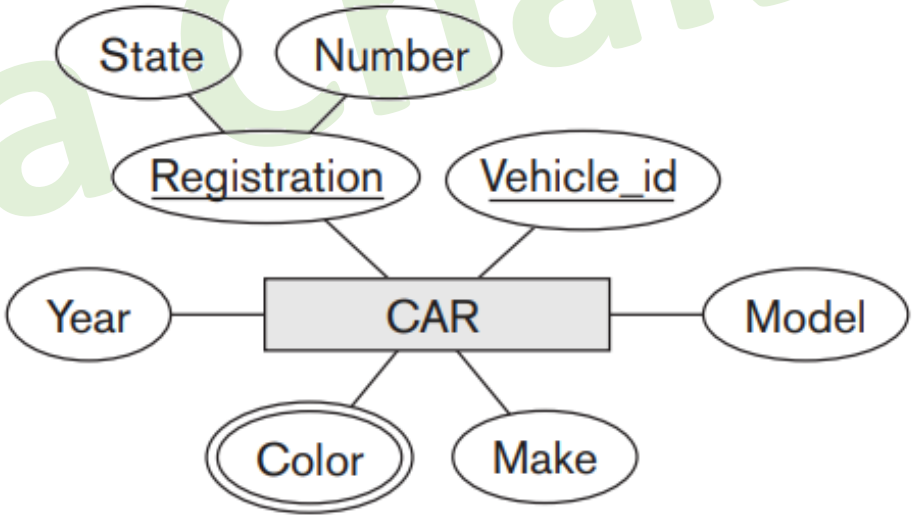


Entity Relationship Model: Entity Types, Entity Sets, Keys, and Value Sets

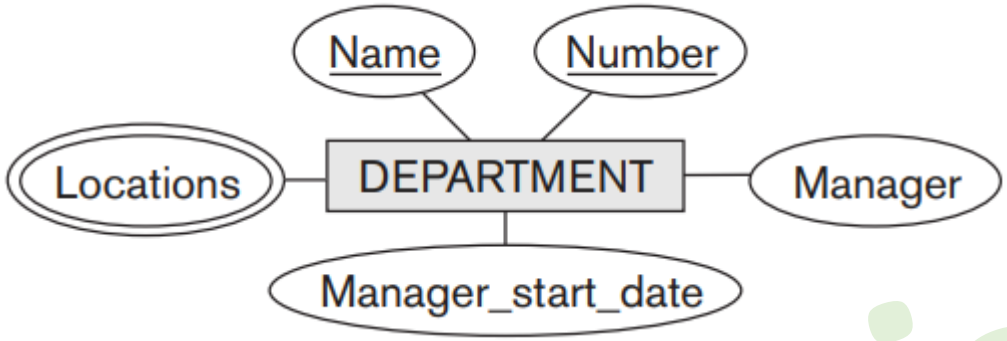
- An entity type usually has one or more attributes whose values are distinct for each individual entity in the entity set: **key attribute**
- Specifying that an attribute is a key of an entity type means that the preceding uniqueness property must hold for every entity set of the entity type.
- It is not the property of a particular entity set; rather, it is a constraint on any entity set of the entity type at any point in time.
- An entity type may also have no key, in which case it is called a weak entity type.
- Each simple attribute of an entity type is associated with a value set (or domain of values), which specifies the set of values that may be assigned to that attribute for each individual entity.

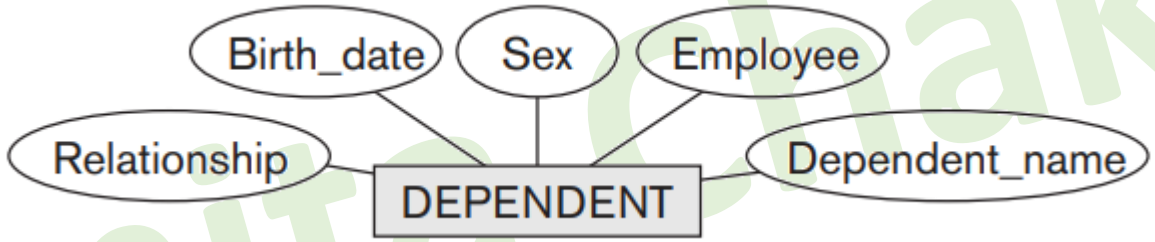
Entity Relationship Model: Notations

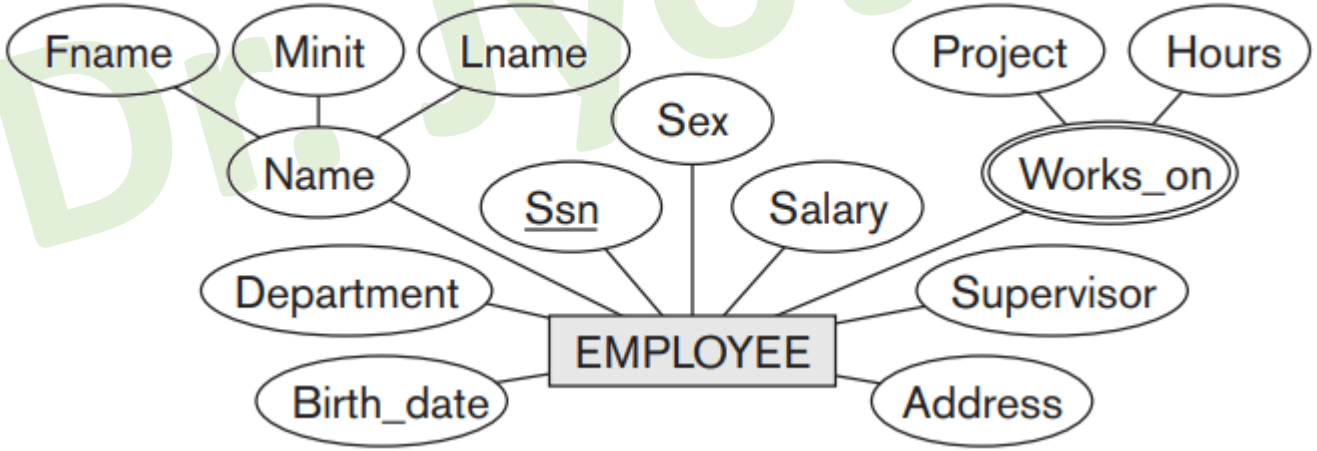
- An **entity type** is represented in ER diagrams as a rectangular box enclosing the entity type name.
- **Attribute names** are enclosed in ovals and are attached to their entity type by straight lines.
- **Composite attributes** are attached to their component attributes by straight lines.
- **Multivalued attributes** are displayed in double ovals.
- **Key attribute** has its name underlined inside the oval
- The Registration attribute is an example of a composite key formed from two simple component attributes, State and Number, neither of which is a key on its own.

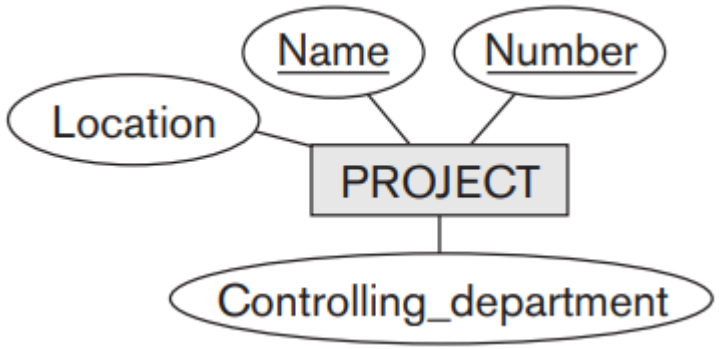


Entity Relationship Model: Example





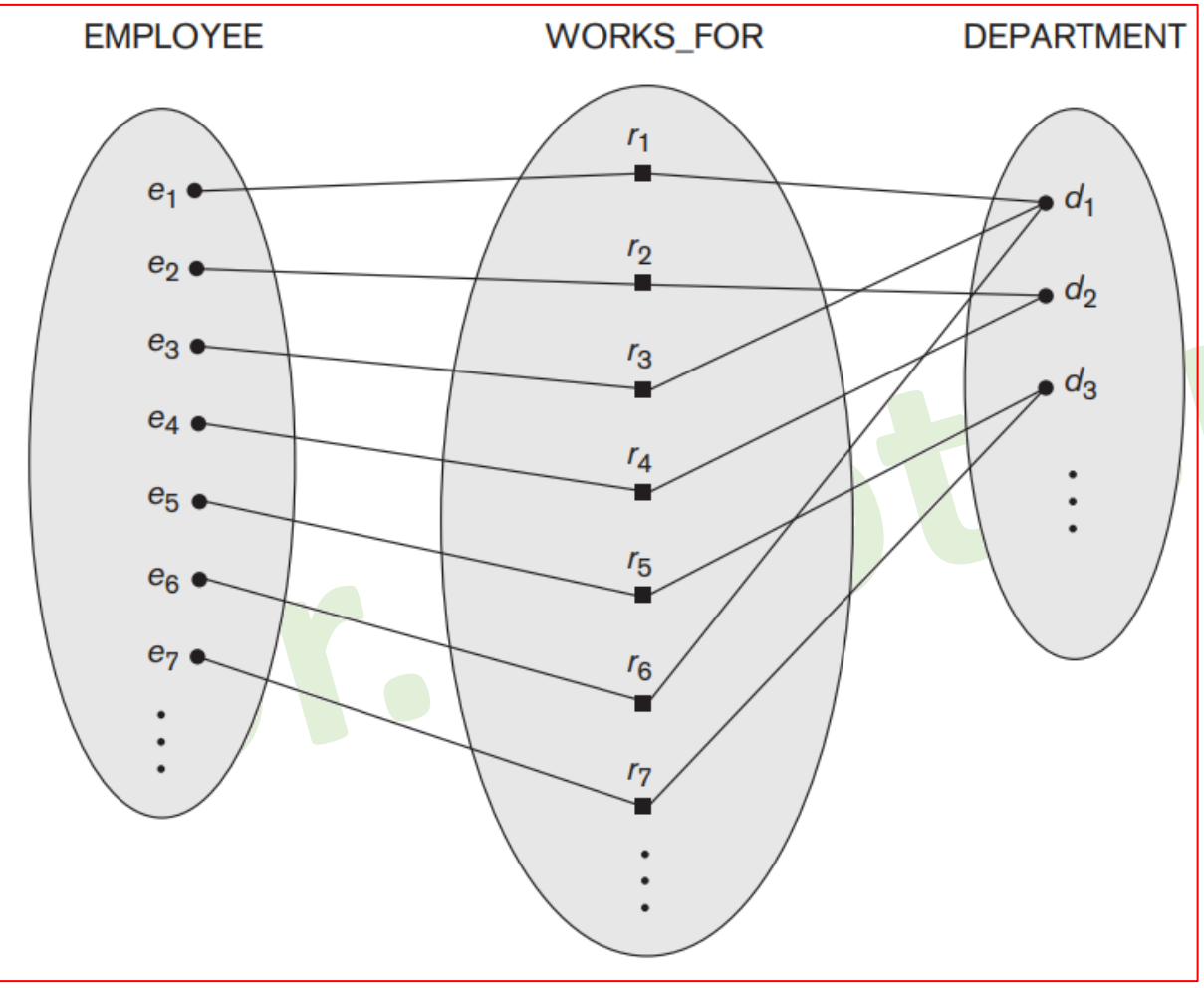


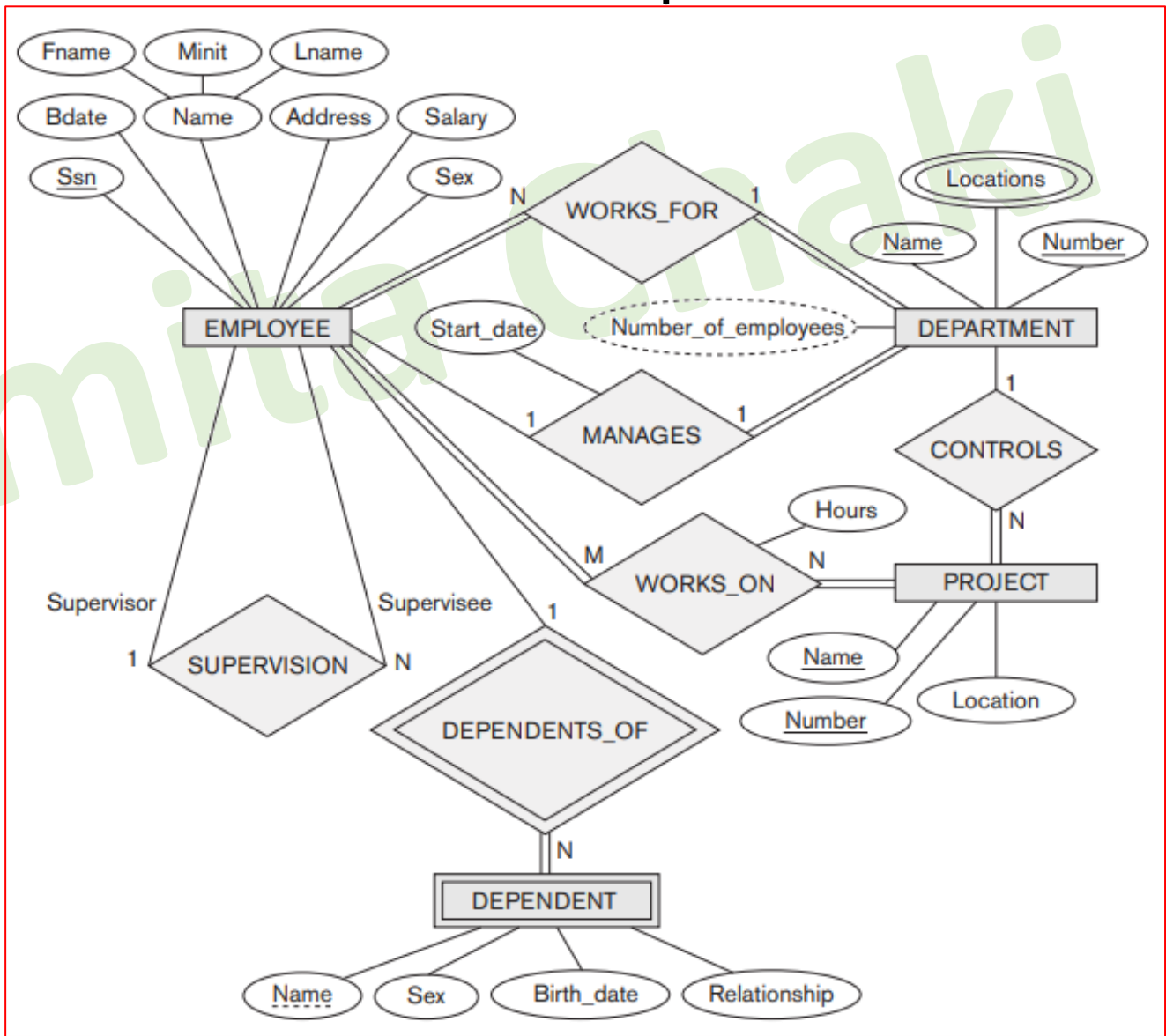


Entity Relationship Model: Relationship

- A **relationship type** R among n entity types E1, E2, . . . , En defines a set of associations—or a **relationship set**—among entities from these entity types.
- The relationship set R is a set of relationship instances ri, where each ri associates n individual entities (e1, e2, . . . , en), and each entity ej in ri is a member of entity set Ej , $1 \leq j \leq n$.
- Each of the entity types E1, E2, . . . , En is said to participate in the relationship type R; similarly, each of the individual entities e1, e2, . . . , en is said to participate in the relationship instance ri = (e1, e2, . . . , en).

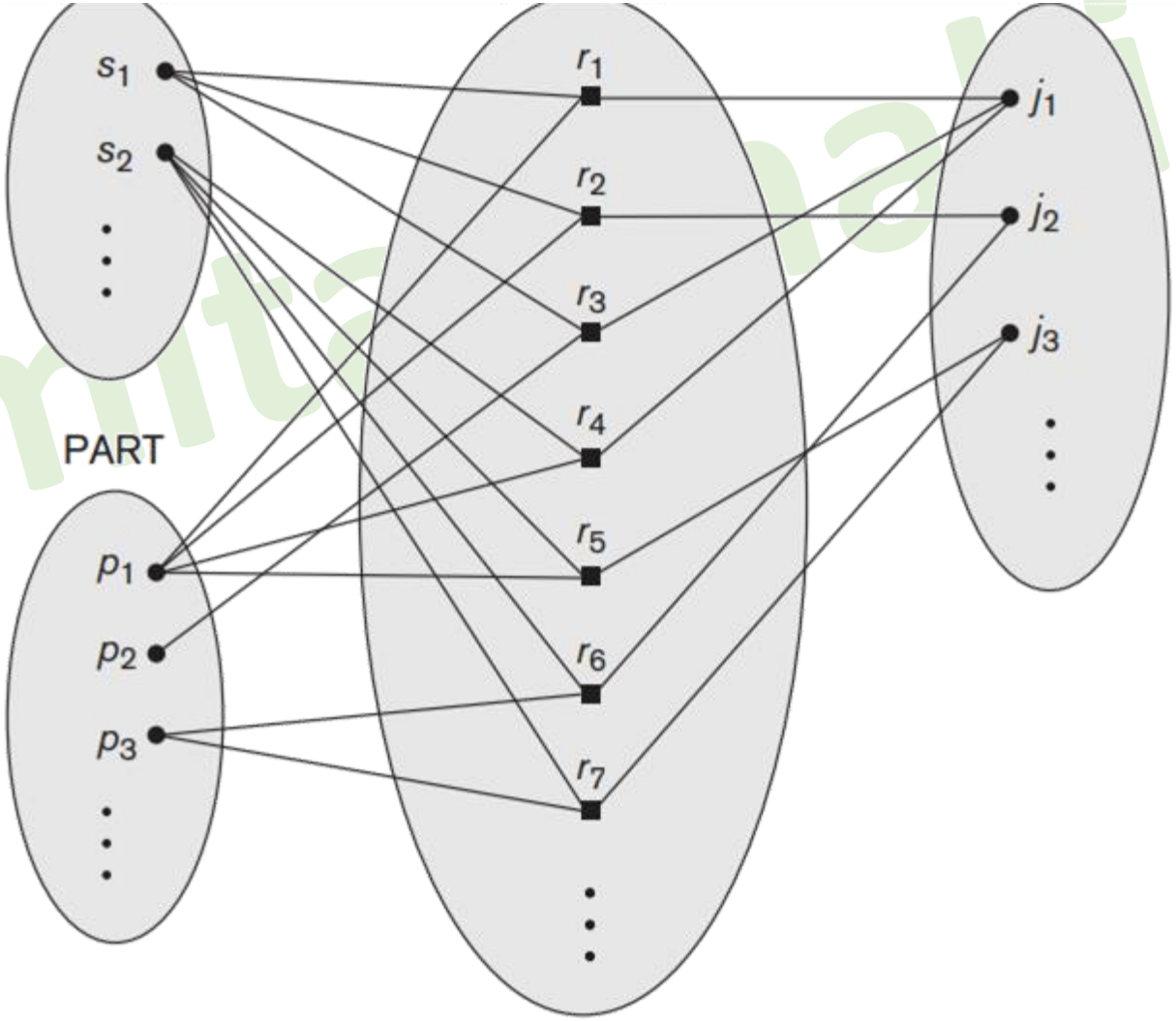
Entity Relationship Model: Relationship





Entity Relationship Model: Relationship

- The degree of a relationship type is the number of participating entity types.
- A relationship type of degree two is called binary, and one of degree three is called ternary.

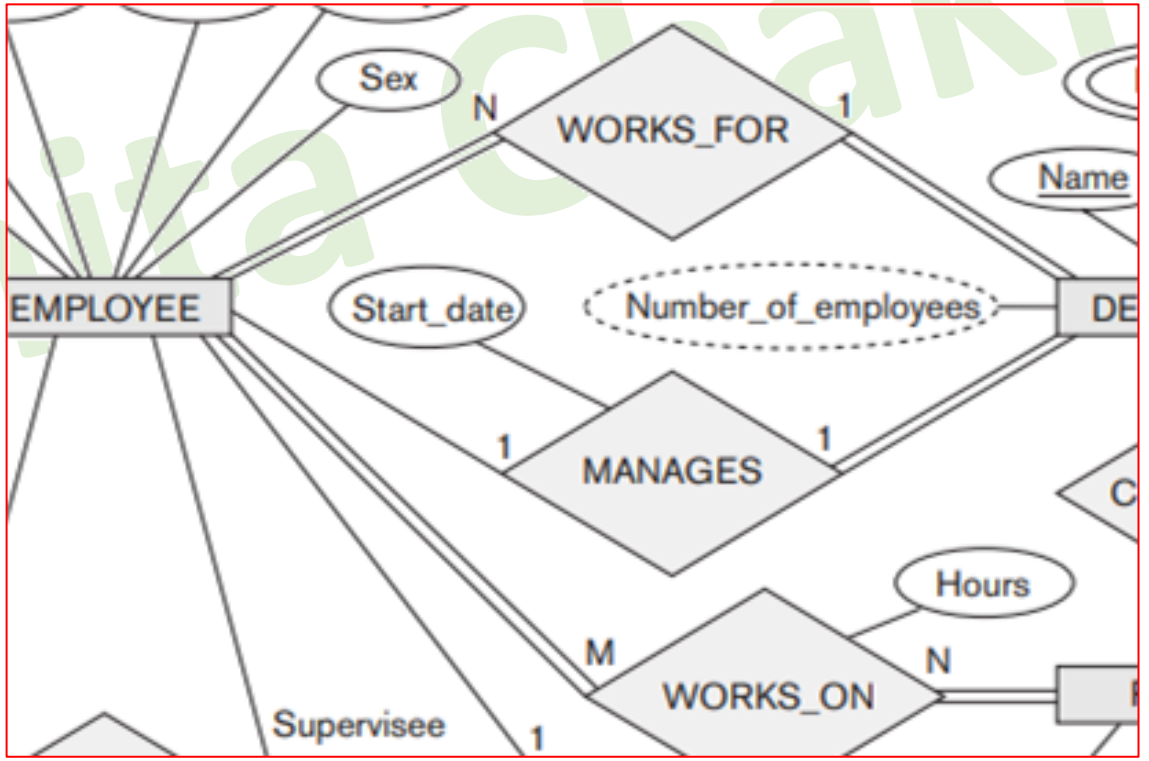


Entity Relationship Model: Binary Relationship Constraints

- Relationship types usually have certain constraints that limit the possible combinations of entities that may participate in the corresponding relationship set.
- Two main types of binary relationship constraints:
  - **Cardinality ratio:** maximum number of relationship instances that an entity can participate in. The possible cardinality ratios for binary relationship types are 1:1 (Employee [Manages] Department), 1:N (Department:Employee), N:1 (Student:Project), and M:N (Employee [Works on] Project). Cardinality ratios for binary relationships are represented on ER diagrams by displaying 1, M, and N on the diamonds.
  - **Participation:** This constraint specifies the minimum number of relationship instances that each entity can participate in: **minimum cardinality constraint**.

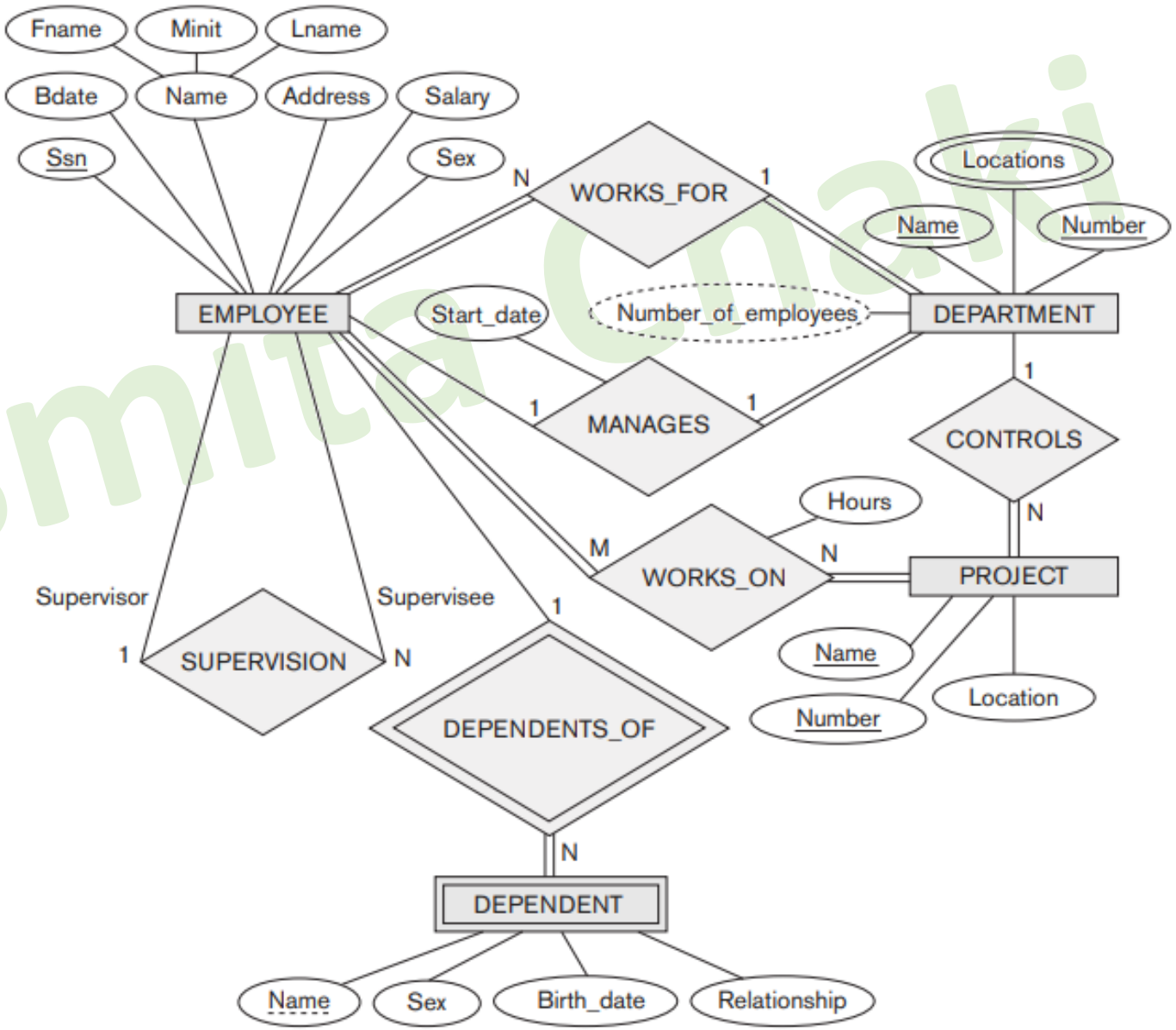
Entity Relationship Model: Binary Relationship Constraints

- Two types of participation constraints—
  - Total: If a company policy states that every employee must work for a department, then an employee entity can exist only if it participates in at least one WORKS_FOR relationship instance: **existence dependency**
  - Partial: we do not expect every employee to manage a department, so the participation of EMPLOYEE in the MANAGES relationship type is partial
- In ER diagrams, total participation (or existence dependency) is displayed as a double line connecting the participating entity type to the relationship, whereas partial participation is represented by a single line



Entity Relationship Model: Structural constraint

- The cardinality ratio and participation constraints, taken together, as the **structural constraints** of a relationship type

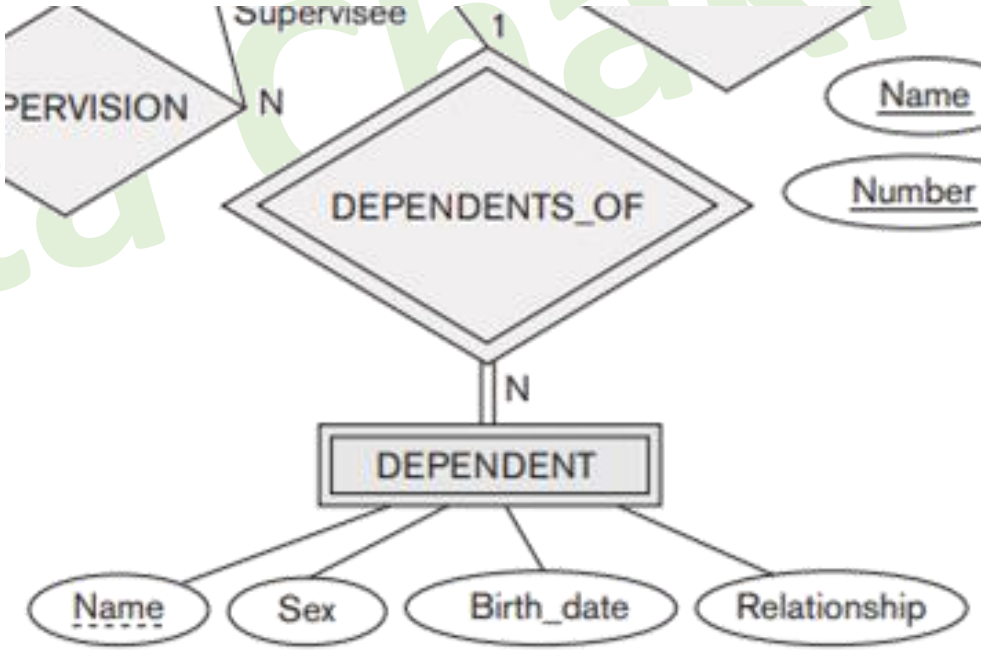


Entity Relationship Model: Weak Entity

- Entity types that do not have key attributes of their own are called weak entity types.
- Regular entity types that do have a key attribute—are called strong entity types.
- The relationship type that relates a weak entity type to its owner the identifying relationship of the weak entity type
- A weak entity type always has a total participation constraint (existence dependency) with respect to its identifying relationship because a weak entity cannot be identified without an owner entity.
- Not every existence dependency results in a weak entity type.
  - For example, a DRIVER_LICENSE entity cannot exist unless it is related to a PERSON entity, even though it has its own key (License_number) and hence is not a weak entity

Entity Relationship Model: Weak Entity

- A weak entity type normally has a partial key, which is the attribute that can uniquely identify weak entities that are related to the same owner entity.
- For example, if we assume that no two dependents of the same employee ever have the same first name, the attribute Name of DEPENDENT is the partial key.
- In ER diagrams, both a weak entity type and its identifying relationship are distinguished by surrounding their boxes and diamonds with double lines. The partial key attribute is underlined with a dashed or dotted line.



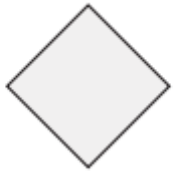
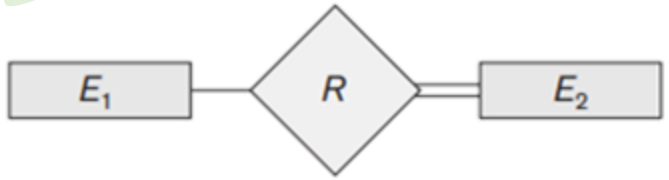
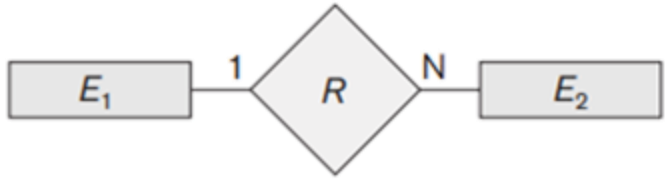
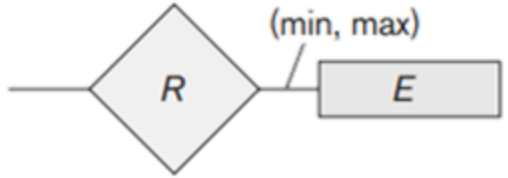
Entity Relationship Model: Summary of Notation for ER Diagrams

- Regular (strong) entity types such as EMPLOYEE, DEPARTMENT, and PROJECT are shown in rectangular boxes.
- Relationship types such as WORKS_FOR, MANAGES, CONTROLS, and WORKS_ON are shown in diamond-shaped boxes attached to the participating entity types with straight lines.
- Attributes are shown in ovals, and each attribute is attached by a straight line to its entity type or relationship type.
- Component attributes of a composite attribute are attached to the oval representing the composite attribute, as illustrated by the Name attribute of EMPLOYEE.
- Multivalued attributes are shown in double ovals, as illustrated by the Locations attribute of DEPARTMENT.
- Key attributes have their names underlined. Derived attributes are shown in dotted ovals, as illustrated by the Number_of_employees attribute of DEPARTMENT.

Entity Relationship Model: Summary of Notation for ER Diagrams

- Weak entity types are distinguished by being placed in double rectangles and by having their identifying relationship placed in double diamonds, as illustrated by the DEPENDENT entity type and the DEPENDENTS_OF identifying relationship type.
- The partial key of the weak entity type is underlined with a dotted line.
- The cardinality ratio of each binary relationship type is specified by attaching a 1, M, or N on each participating edge.
- The cardinality ratio of DEPARTMENT:EMPLOYEE in MANAGES is 1:1, whereas it is 1:N for DEPARTMENT: EMPLOYEE in WORKS_FOR, and M:N for WORKS_ON.
- The participation constraint is specified by a single line for partial participation and by double lines for total participation (existence dependency).

Entity Relationship Model: Summary of Notation for ER Diagrams

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Indentifying 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

Entity Relationship Model: Ternary Relationship

- Relationship type of degree three
- The figure includes a relationship instance (i, s, c) whenever INSTRUCTOR i offers COURSE c during SEMESTER s.
- The three binary relationship types:
  - CAN_TEACH relates a course to the instructors who can teach that course,
  - TAUGHT_DURING relates a semester to the instructors who taught some course during that semester, and
  - OFFERED_DURING relates a semester to the courses offered during that semester by any instructor.