Chapter 5 – Week6

Entity Relationship Model Continued

Announcements

- Lab3
  - On Teams on Monday 4:30pm
  - Due Friday October 22nd
- Assignment 1 – due

RECAP

Main Phases of Database Design

1. Requirements Gathering
2. Conceptual Model (high-level)
   - ER Model
3. Logical Model
   - Relational Model
4. Physical Model (not a part of 3530)

Example : University Staff Database

1. Each department of the University has a unique name, a unique number, and a manager (who is also an employee). For each manager, the start date when he / she was hired as a manager is stored. A department may have several locations.
2. A department runs a number of activities to support its staff and its community, each of which has a name, a unique activity number, and a single location where it is hosted.
3. Each employee's name, social insurance number, address, salary, gender and birth date is stored. **An employee** works for a department but may **volunteer for several activities**. We keep track of the number of hours per week that an employee volunteers on each activity. Each employee has a supervisor.
4. An employee may have 0 or more dependents. We keep each dependent's first name, gender, birth date, and relationship to the employee (for insurance purposes).

Structural Constraints on Relationship Types:

- Limit the possible combination of entities
- Represent business rules established by the user

- 2 Structural Constraints :
  - Cardinality Ratio (also known as Relationship Multiplicity)
  - Participation Constraint

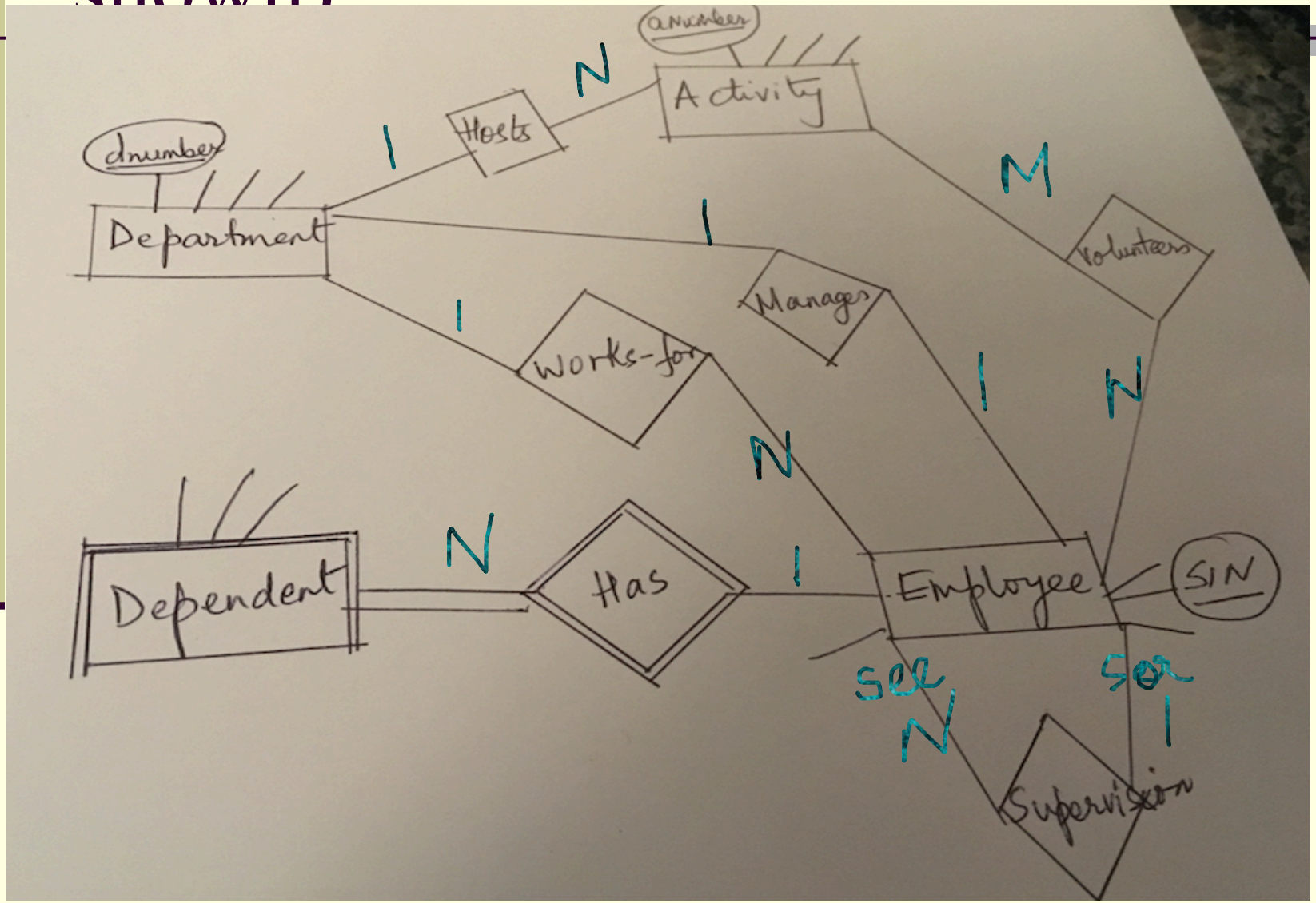
Cardinality Ratio

Cardinality Ratio : number of relationship instances that an entity can participate in.

- Shown By Placing Appropriate Number On The Link.
  - 1:N, 1:1 , M:N

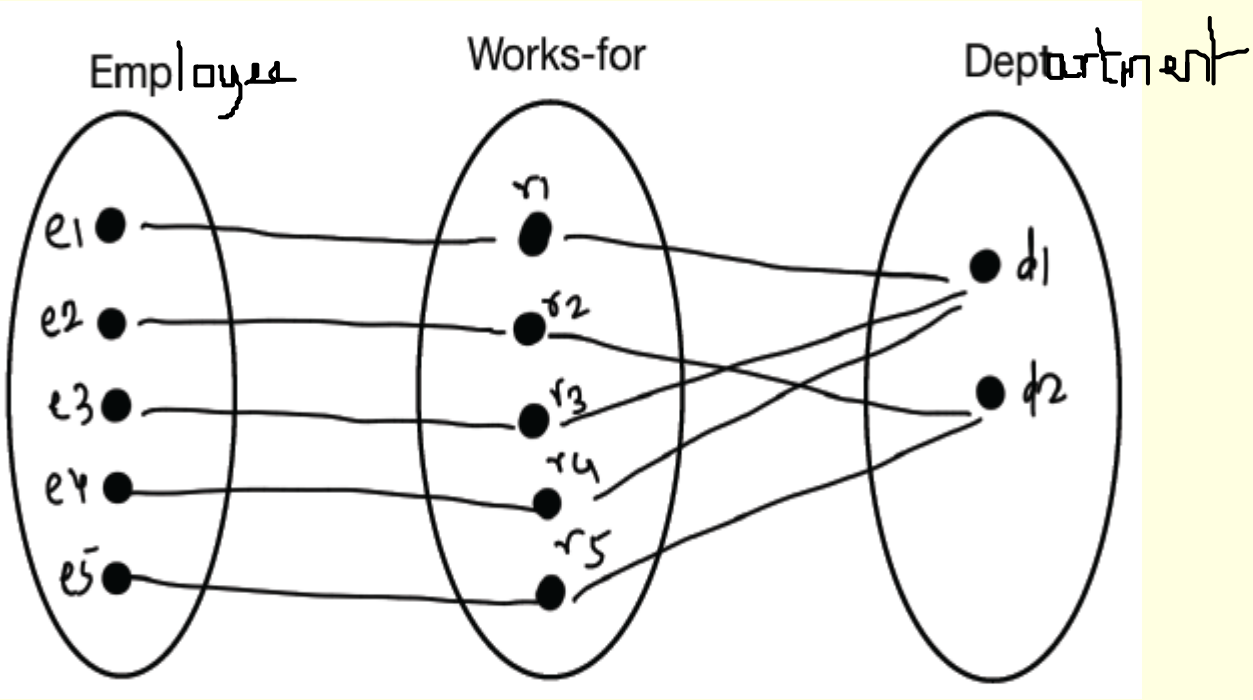
- Note that the textbook uses an arrow in the direction of 1 side of a relationship (an example is shown in the next slide). I will use Chen's notation mostly

Design so far (only key attributes shown)

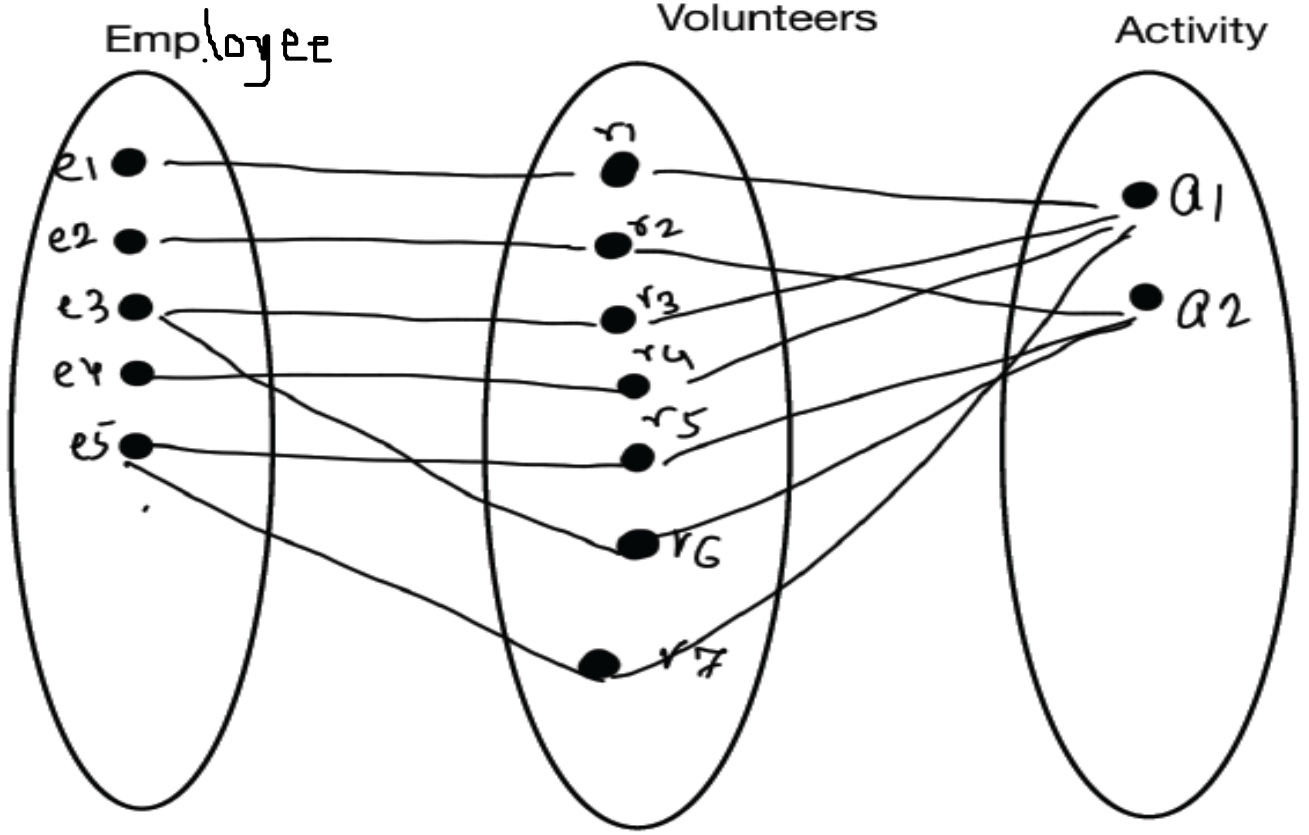


Relationship instances - revisited

- An employee works for a department

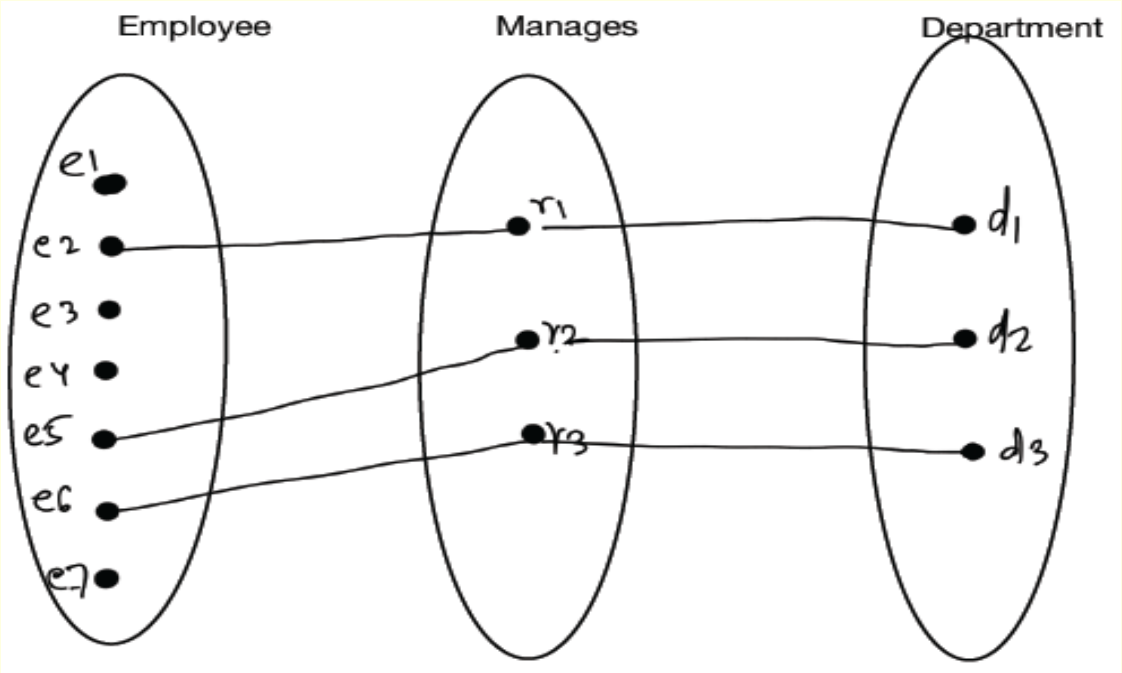


Relationship type Volunteers

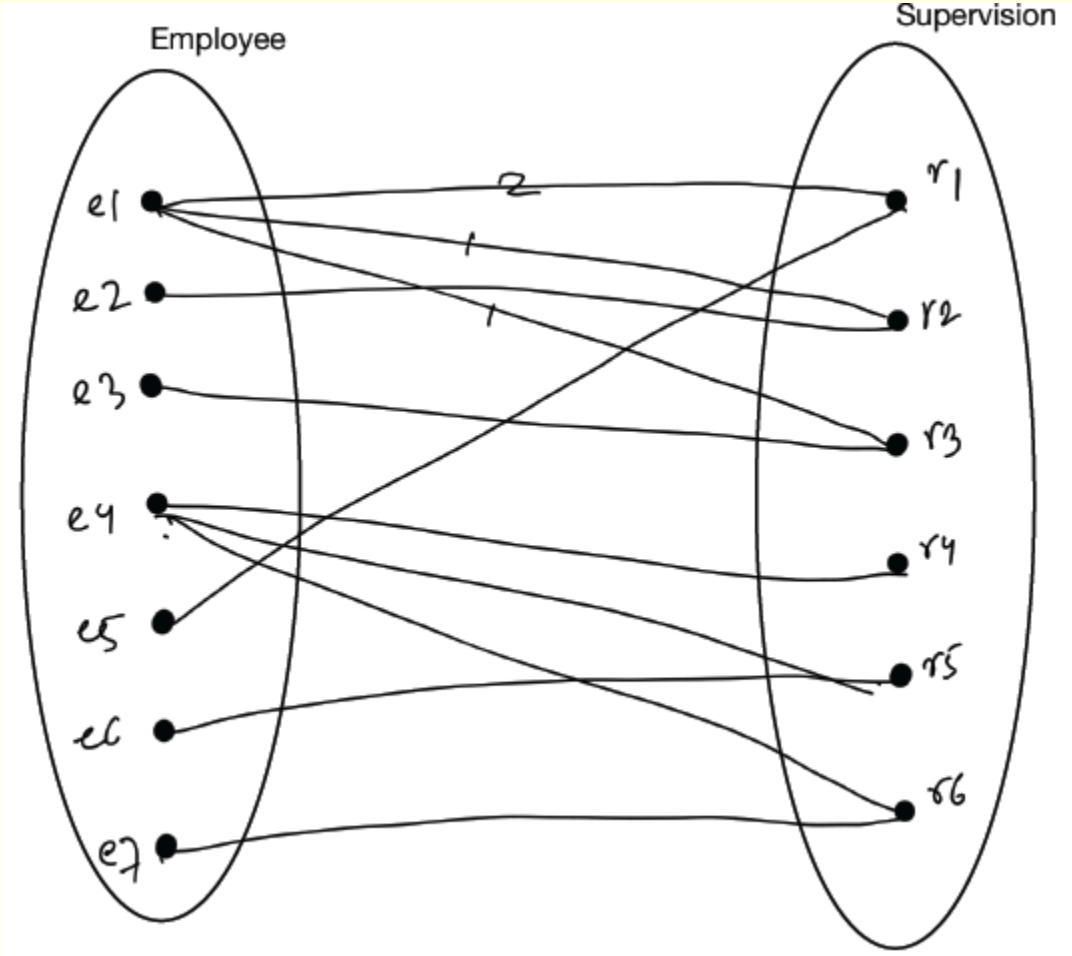


Relationship type MANAGES

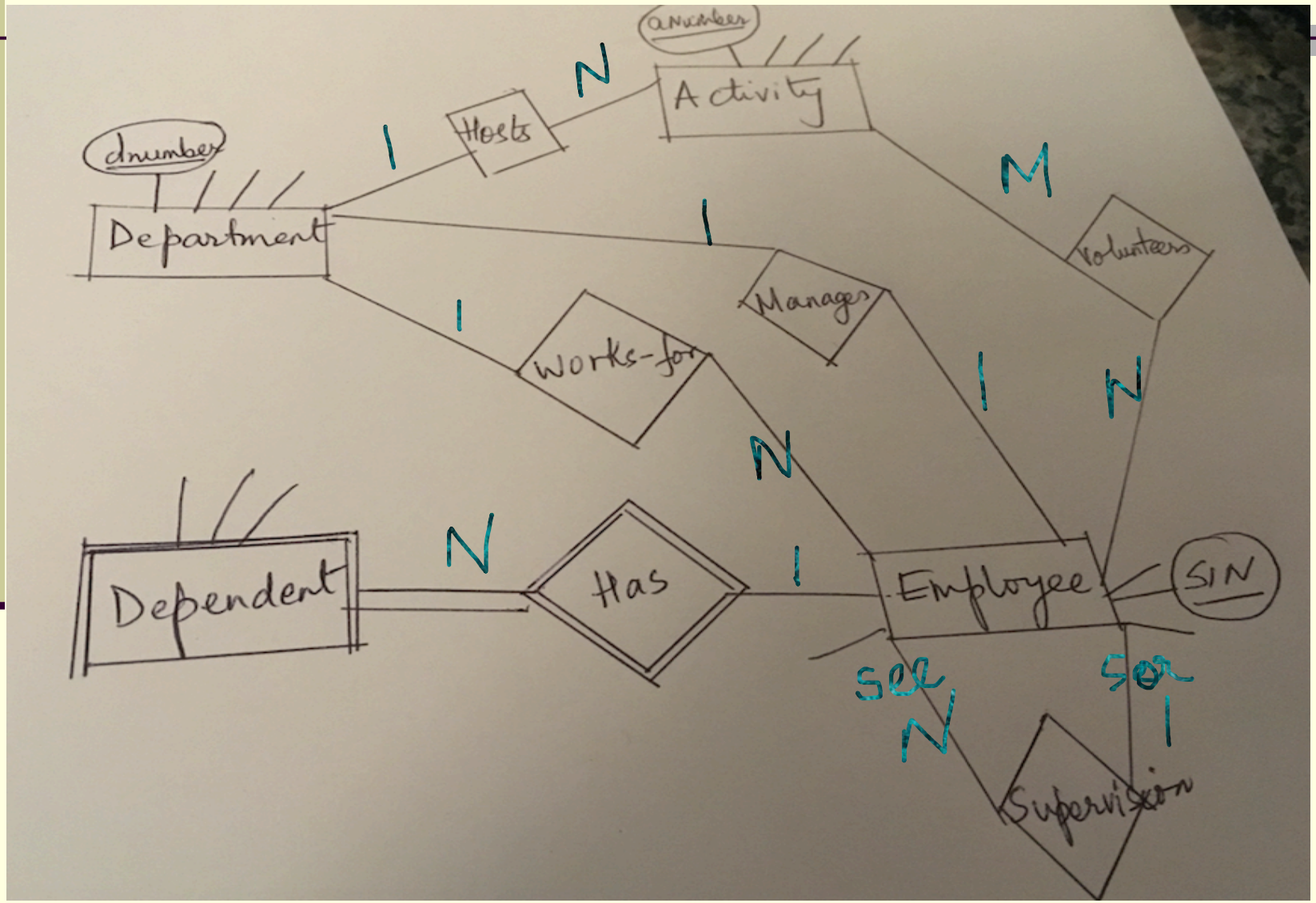
ER Diagram?



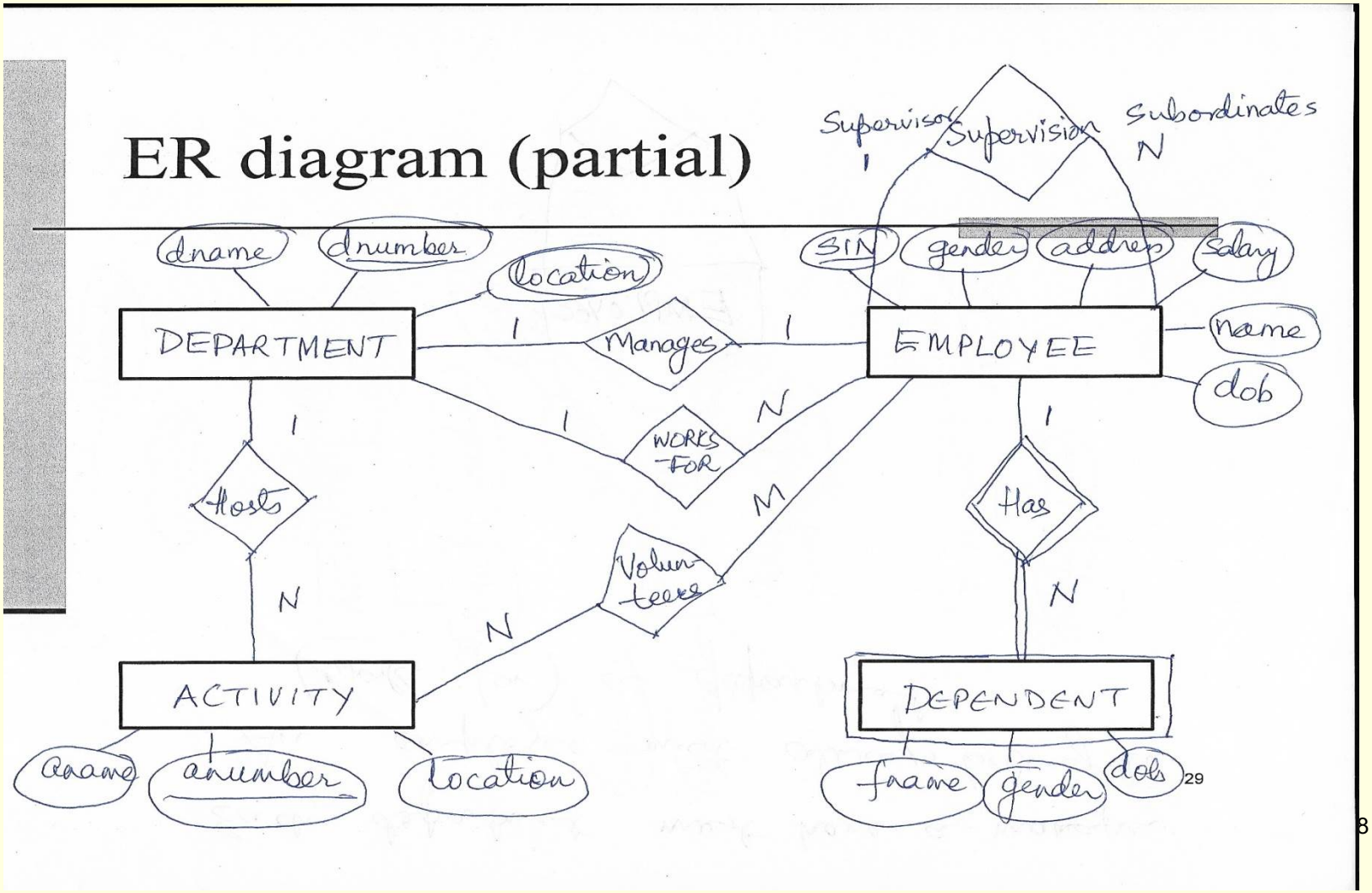
Recursive Relationships



Design so far – repeat slide



University Staff DB ER model – updated with Cardinality ratios



Participation Constraint

Participation Constraint: specifies whether the existence of an entity depends on its being related to another entity via the relationship type.

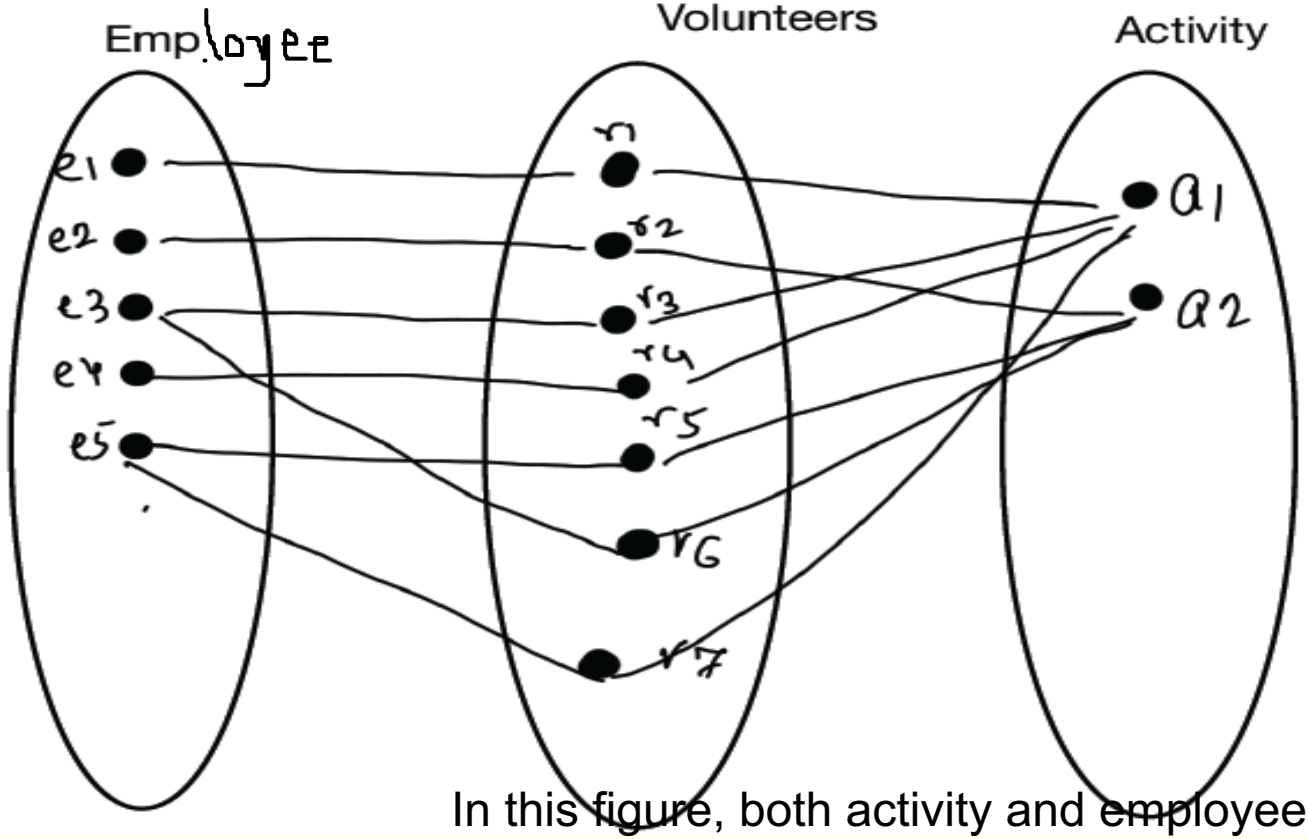
SHOWN BY DOUBLE LINING THE LINK

- Total (Existence dependency)

ER : = (double line)

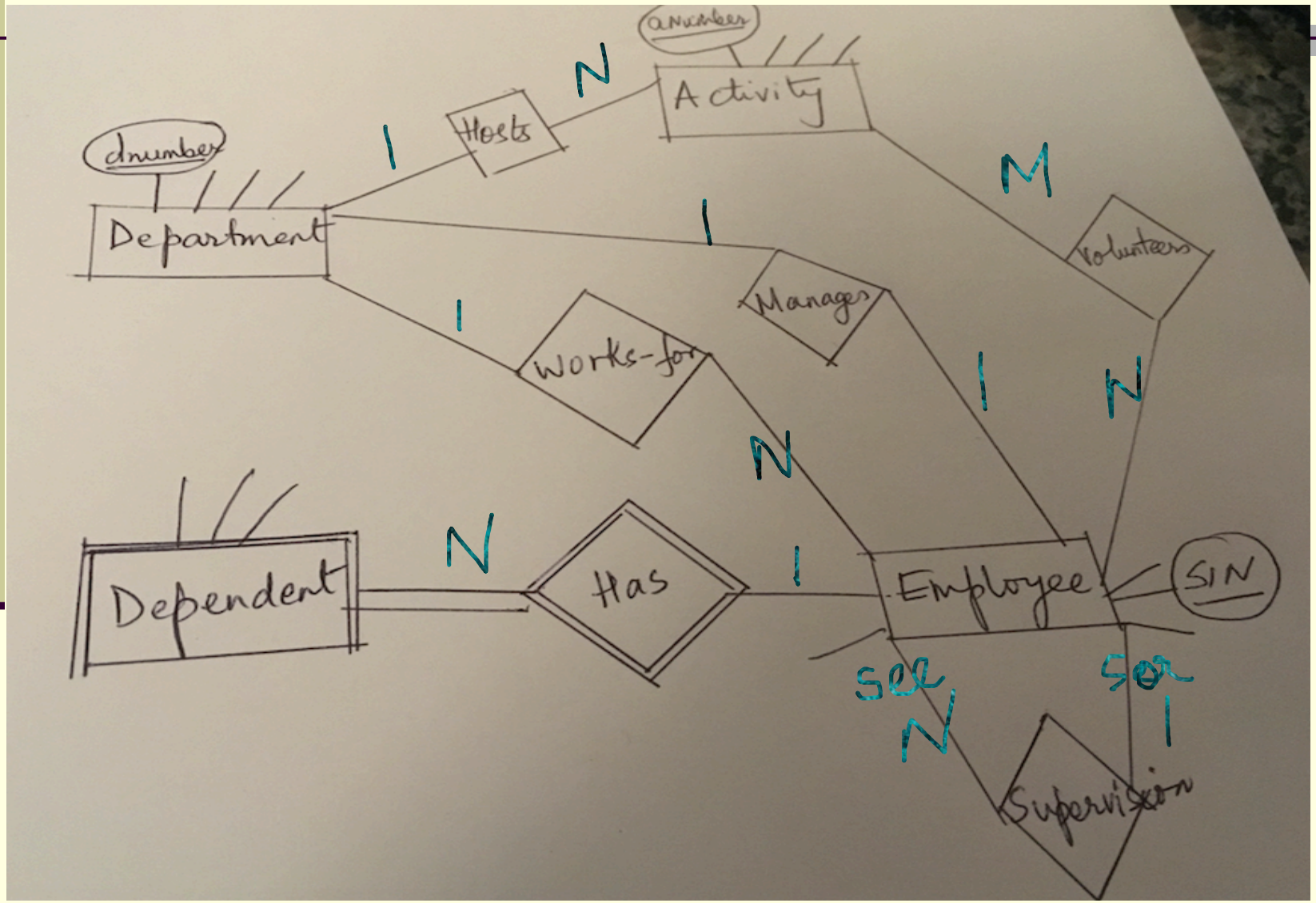
- Partial

Relationship type Volunteers



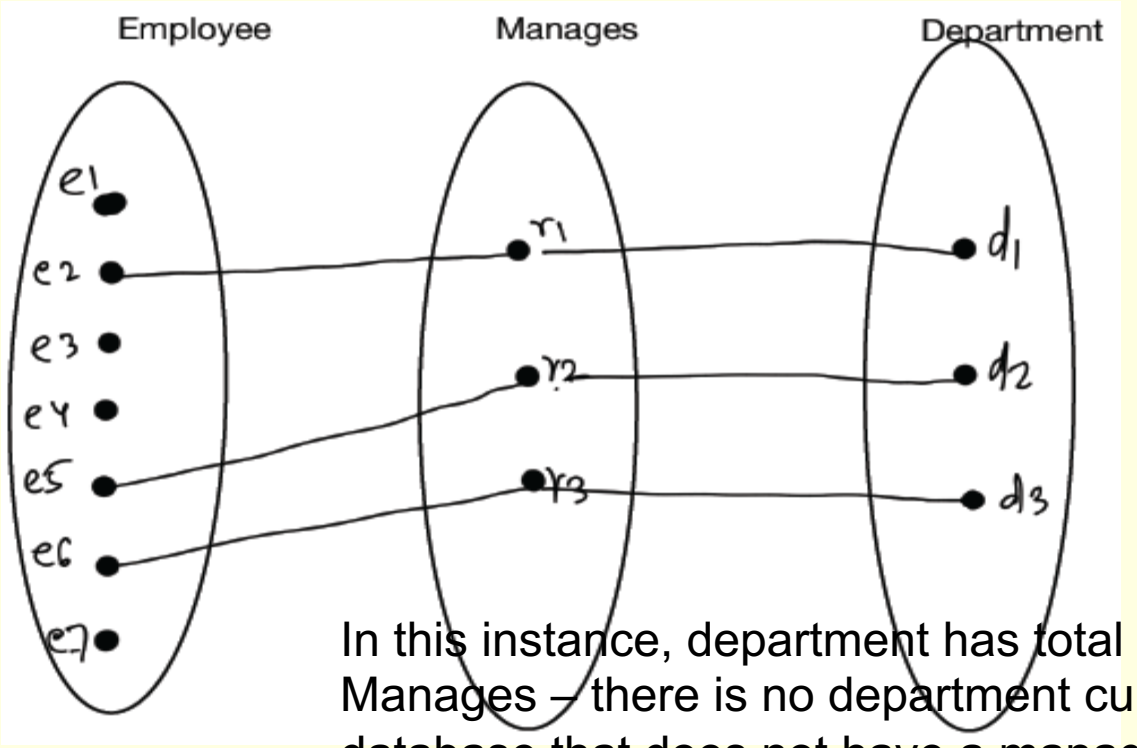
In this figure, both activity and employee have total participation – there is no activity currently in the database that does not have a volunteer. Similarly, there is no employee that doesn't volunteer – all 5 employees volunteer and both activities have at least

Design so far – repeat slide



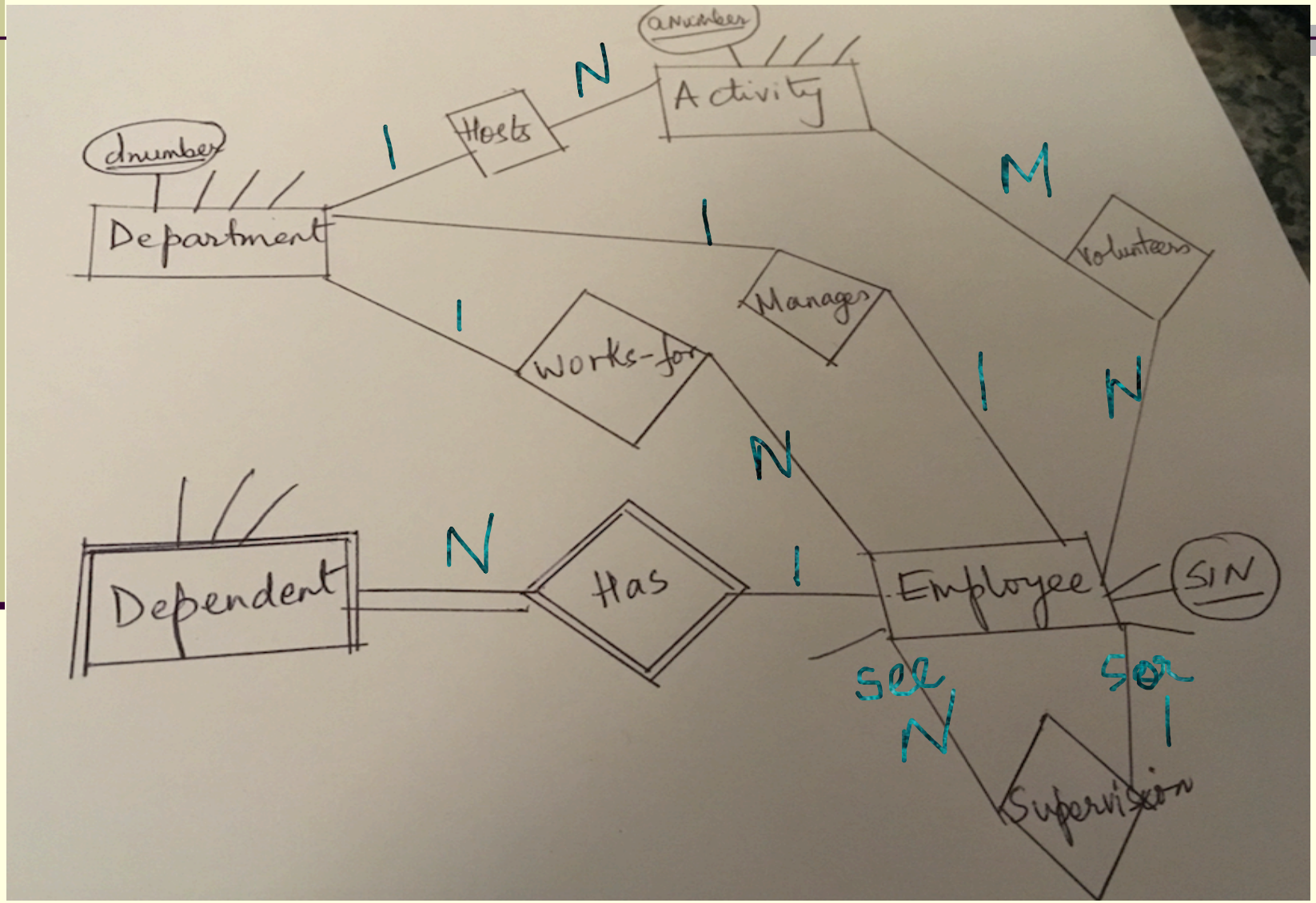
Relationship type MANAGES

ER Diagram?



In this instance, department has total participation in Manages – there is no department currently in the database that does not have a manager. But there are some employees (e1, e3, e4) that do not manage any department – suggesting that employee has partial participation in Manages. .

Design so far – repeat slide



ER model

- Entities
  - Strong
  - Weak
- Attributes
  - Simple, composite, single_valued, multi_valued, key, relationship attributes
- Relationships
  - Binary
  - Recursive
  - Structural constraints
    - Cardinality Ratio (1:1, 1:N, M:N)
    - Participation (total, partial)

Main Phases of Database Design

1. Requirements Gathering
2. Conceptual Model
   - ER Model
3. Logical Model
   - Relational Model
4. Physical Model

Map ER Model to Relational model

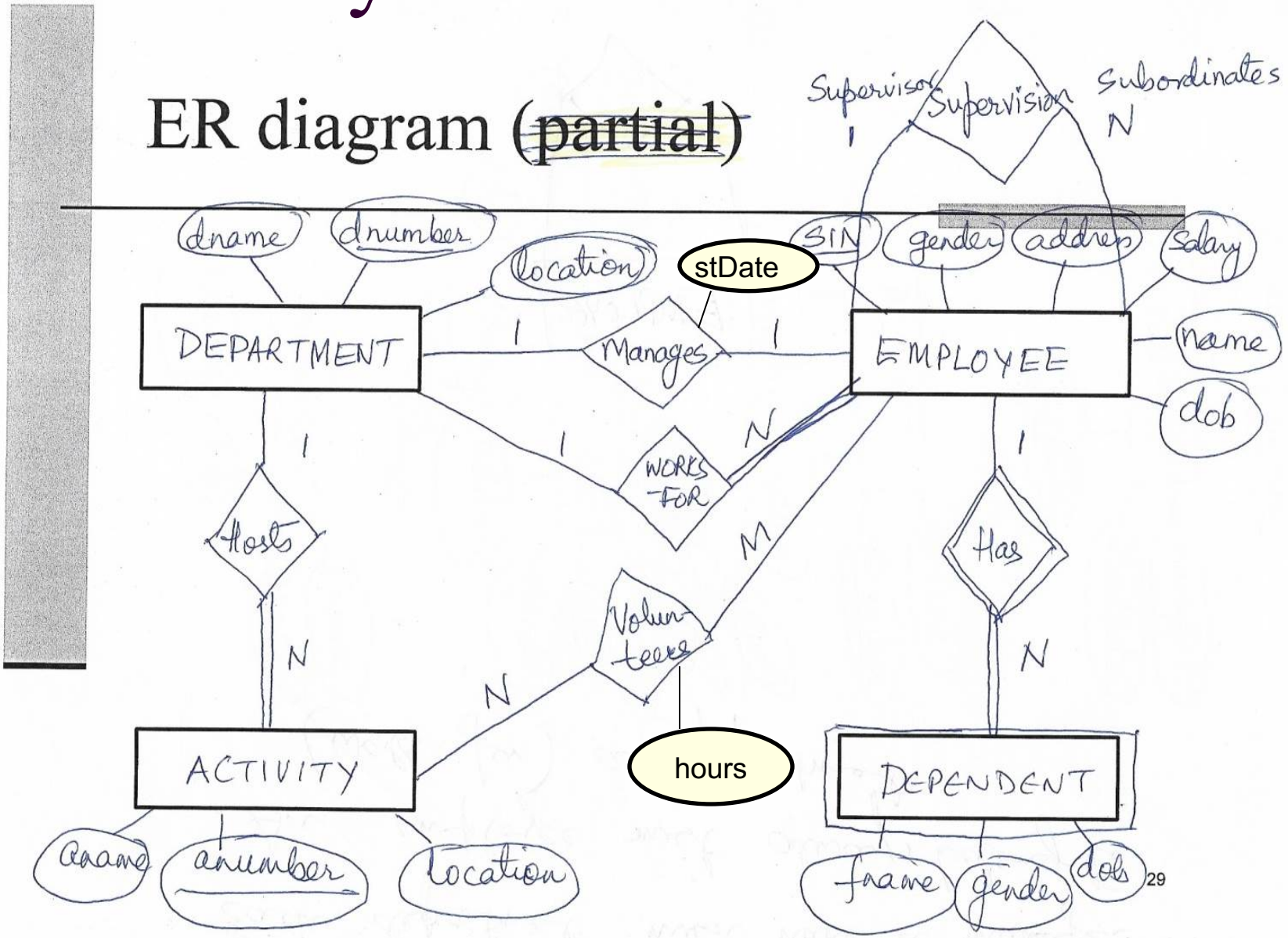
Input: ER Model

Output: Relational Model

General Idea:

- Each entity type (ET) becomes a relation.
- Only the simple components of any composite attribute are taken.
- Each 1:1 and 1:N relationship adds an attribute (as foreign key) to an existing ET.
- Each M:N relationship becomes a new relation
- Each multi-valued attribute becomes a new relation

University Staff DB ER model



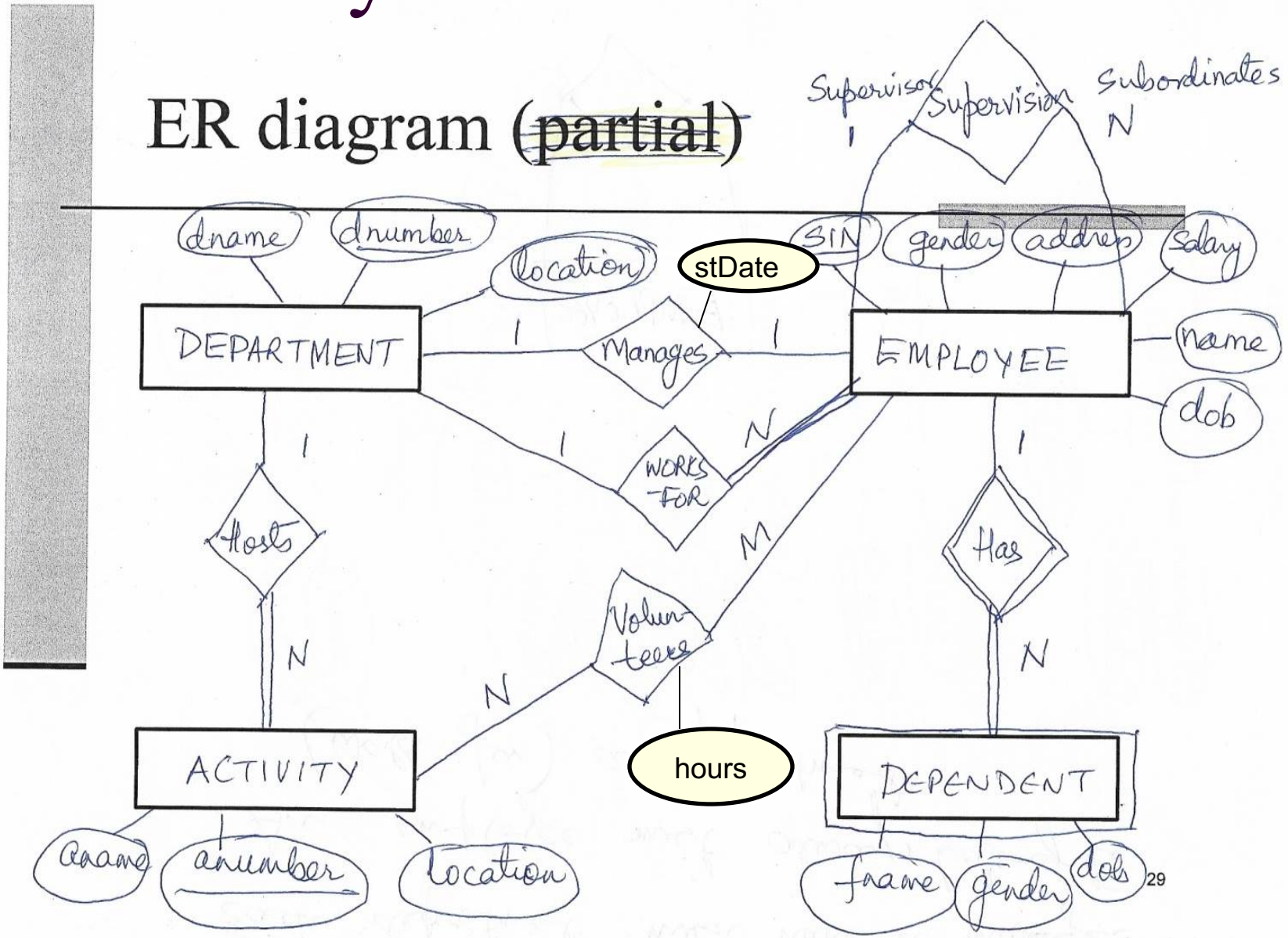
ER-to-Relational Mapping :Step 1

For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as primary key for R. If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

ER-to-Relational Mapping :Step 2

- For each weak entity type W in the ER schema with owner entity type E, create a relation R, and include all simple attributes (or simple components of composite attributes) of W as attributes of R. In addition, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

University Staff DB ER model



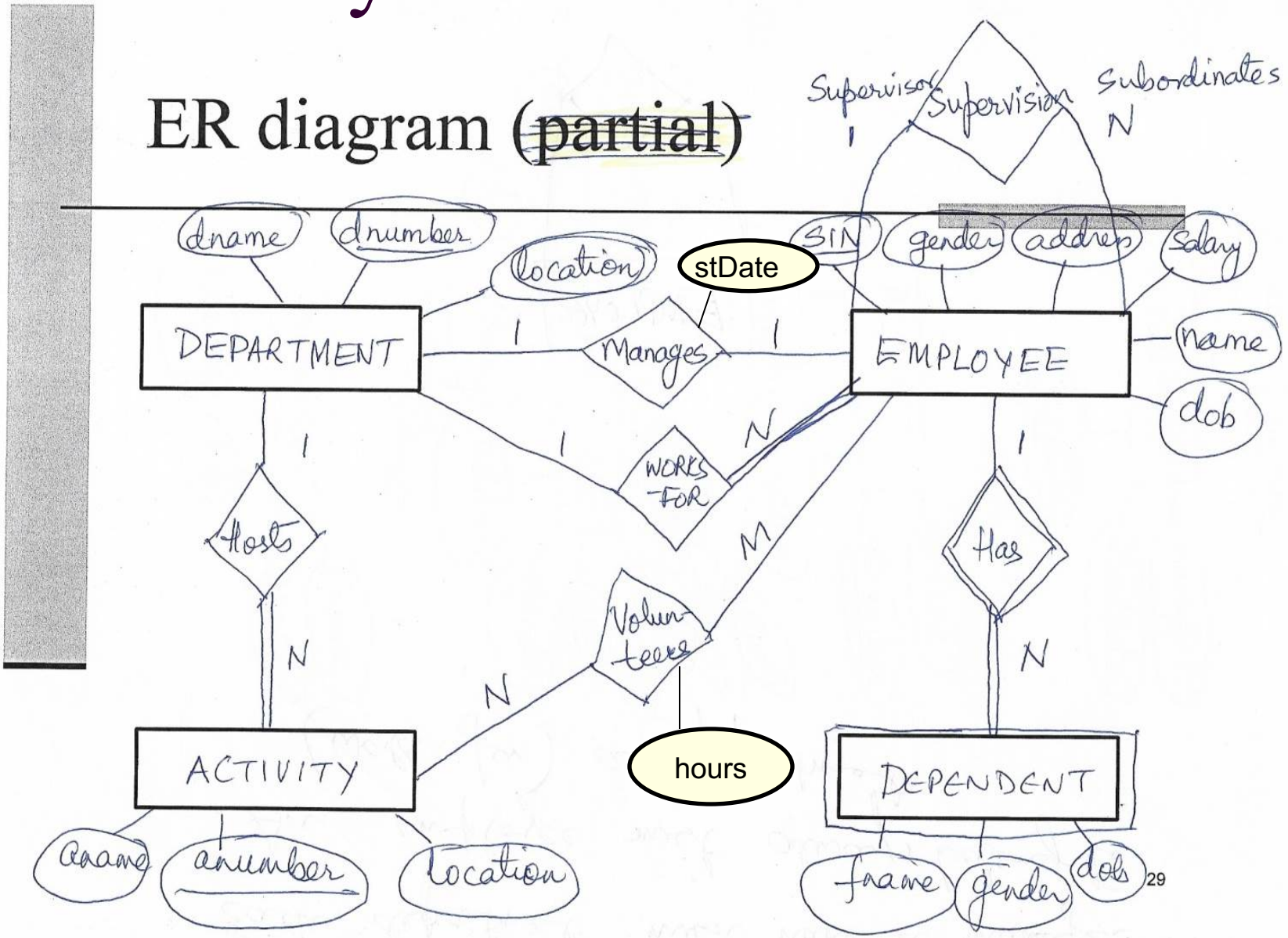
ER-to-Relational Mapping :Step 3

- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. Choose one of the relations—S, say—and include as foreign key in S the primary key of T. It is better to choose an entity type with *total participation* in R in the role of S. Include all the simple attributes (or simple components of composite attributes) of the 1:1 relationship type R as attributes of S.

ER-to-Relational Mapping :Step 4

- For each regular binary 1:N relationship type R, identify the relation S that represents the participating entity type at the *N-side* of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; this is because each entity instance on the N-side is related to at most one entity instance on the 1-side of the relationship type. Include any simple attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S.

University Staff DB ER model



ER-to-Relational Mapping :Step 5

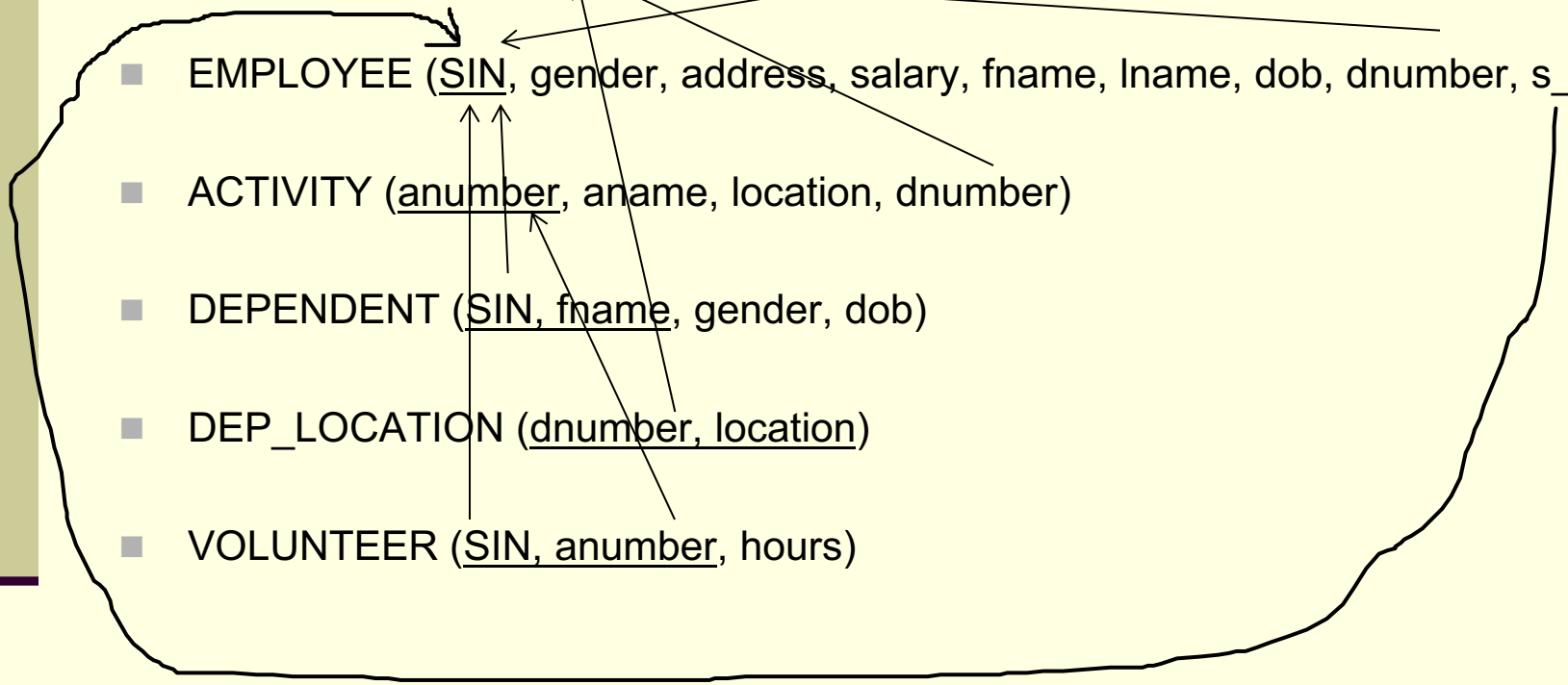
- For each binary M:N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S. Notice that we cannot represent an M:N relationship type by a single foreign key attribute in one of the participating relations—as we did for 1:1 or 1:N relationship types—because of the M:N cardinality ratio.

ER-to-Relational Mapping :Step 6

- For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as an attribute. The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components

Relational model of University staff database

- DEPARTMENT (dnumber, dname, SIN)
- EMPLOYEE (SIN, gender, address, salary, fname, lname, dob, dnumber, s_sin)
- ACTIVITY (anumber, aname, location, dnumber)
- DEPENDENT (SIN, fname, gender, dob)
- DEP_LOCATION (dnumber, location)
- VOLUNTEER (SIN, anumber, hours)



Next class