Entity Relationship

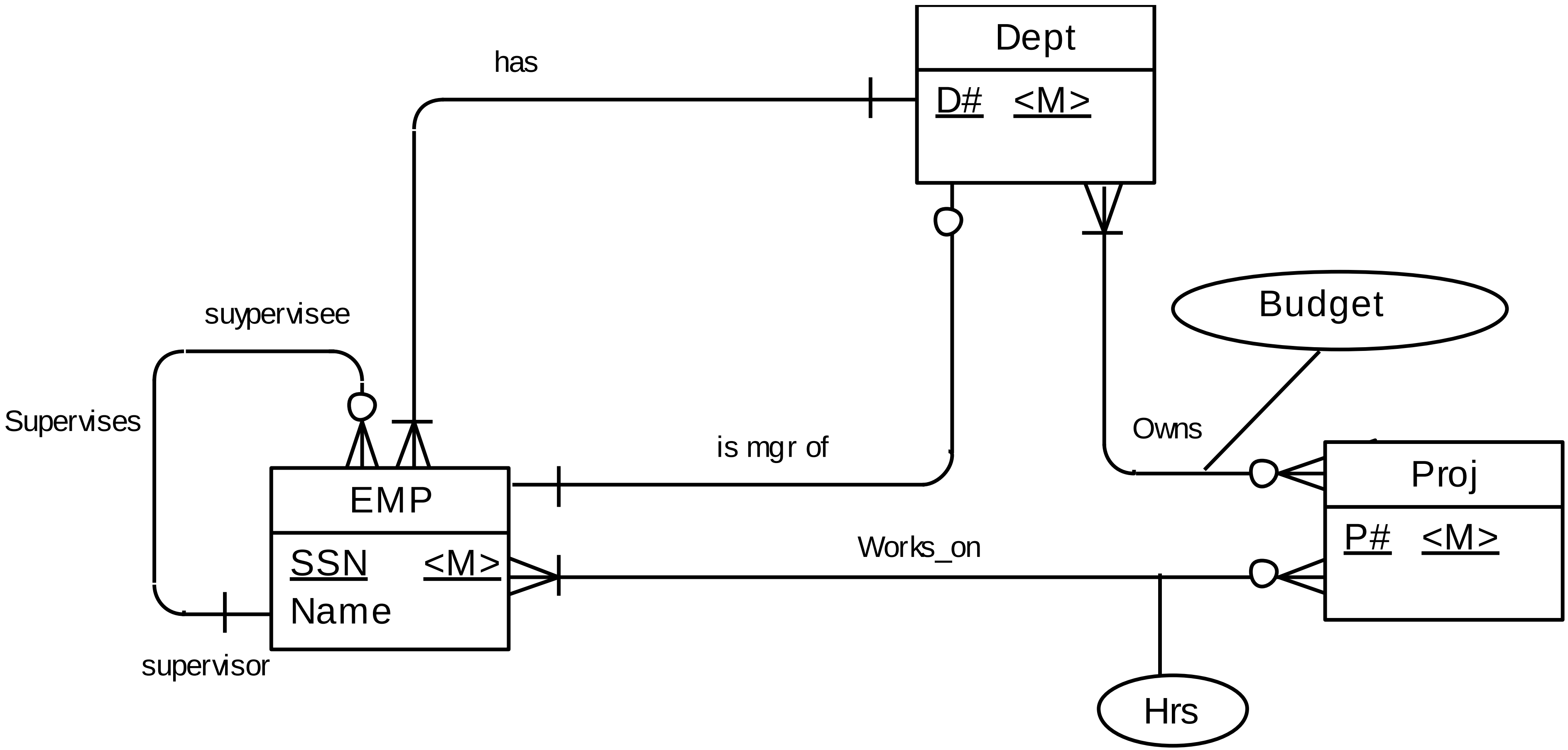
(ER) Model

Higher Level Concepts

Lab

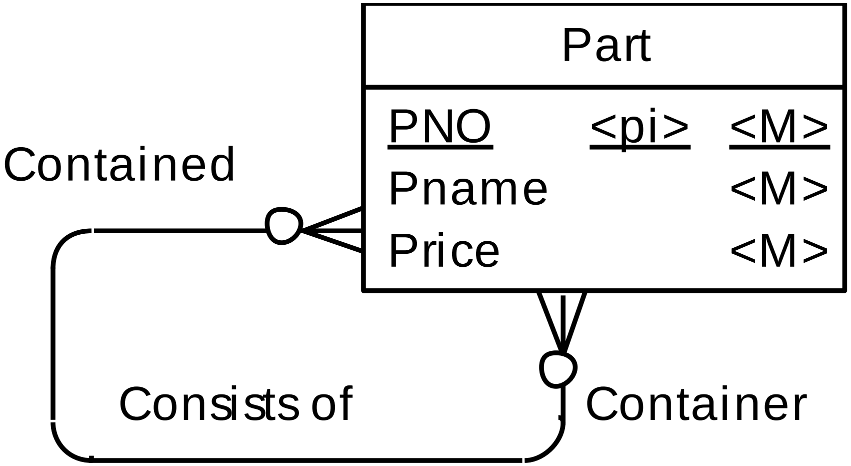
- In your company:
  - An employee must work for one and only one dept
  - A dept must have one or more employees
  - A dept can run one or more projects, but it does not have to.
  - A project must have at least one department that sponsors it but it can belong to more. Each department that participates in a project has a budget for the project.
  - Each dept must have a mgr.
  - Each employee must report to a supervisor
  - An employee may be a supervisor for one or more employees
  - Employees can work on one or more projects for a given number of hours per week
  - A project must have at least one or more employees working on it
- Draw an ER diagram to represent this information

Answer


Dept
D#
<M>
has
suypervisee
Supervises
Budget
Owns
is mgr of
EMP
Proj
SSN
<M>
P#
<M>
Works_on
Name
supervisor
Hrs

Entity Relationship Model

- Question :
  - A part can contain other parts, but it does not have to.
  - A part can be a component of one or more parts, but it does not have to.
  - Draw the ER diagram that shows this.
    - Example:
    - P1 contains P2 and P3
      - P2 is contained in P1
      - P3 is contained in P1



Entity Relationship Model

- Recursive Relationships
  - Roles – entities involved in a recursive relationship play different roles
  - Examples
    - A part plays the container role for the part that plays the contained role
    - A person plays the role of a husband for the person who plays the role of a wife
    - An employee plays the role of supervisor for employee who plays the role of supervisee

Entity Relationship Model

- Question:
  - In a bank, there are many different types of accounts, such as Checking, Savings, CD, Money Market, Trust, etc.
  - All accounts have an account number that is unique, a balance, and a type designation.
  - Customers have a customer ID that is unique, name and address.
  - An account belongs to one and only one customer.
  - A customer must have at least one account but can have more.
  - Draw an entity relationship model for this environment.

Entity Relationship Model

- Specialization vs. Generalization
  - Specialization
    - Top down design approach
    - Less detail to more detail
    - Good when the complete picture is not known
  - Generalization
    - Bottom up design approach
    - More detail to less detail
    - Grouping

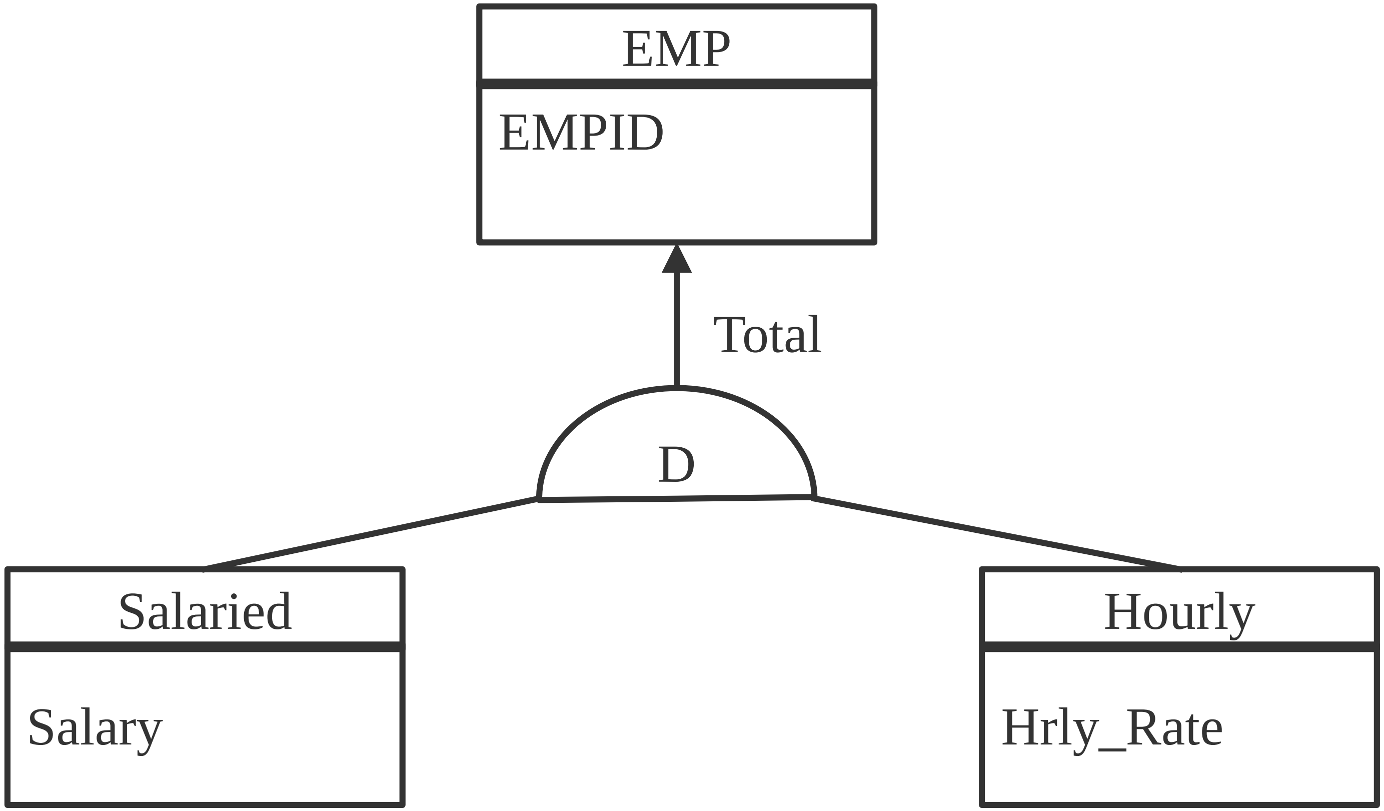
Specialization Types

- Specialization can be:
  - Total – all sub-types have been identified
  - Non-total – some of the sub-types have been identified
- Specialization can also be:
  - Overlapping –sub-types may overlap
  - Disjoint – sub-types cannot overlap

  - Combining the two aspects, gives us four alternatives

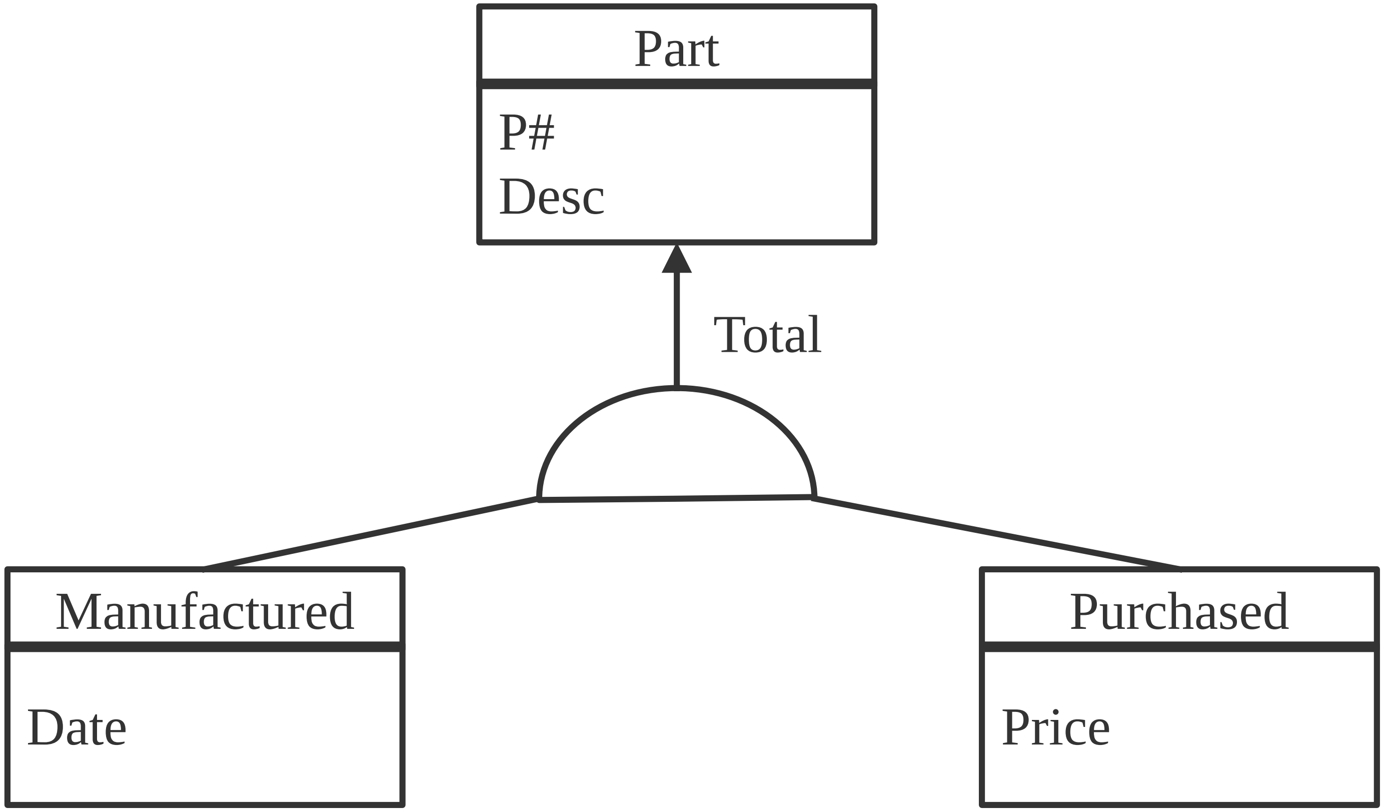
ER Model

- Example
  - Every Emp is either a Salaried employee or an hourly employee.
  - There are no other types of employees
  - This is a **total disjoint** specialization



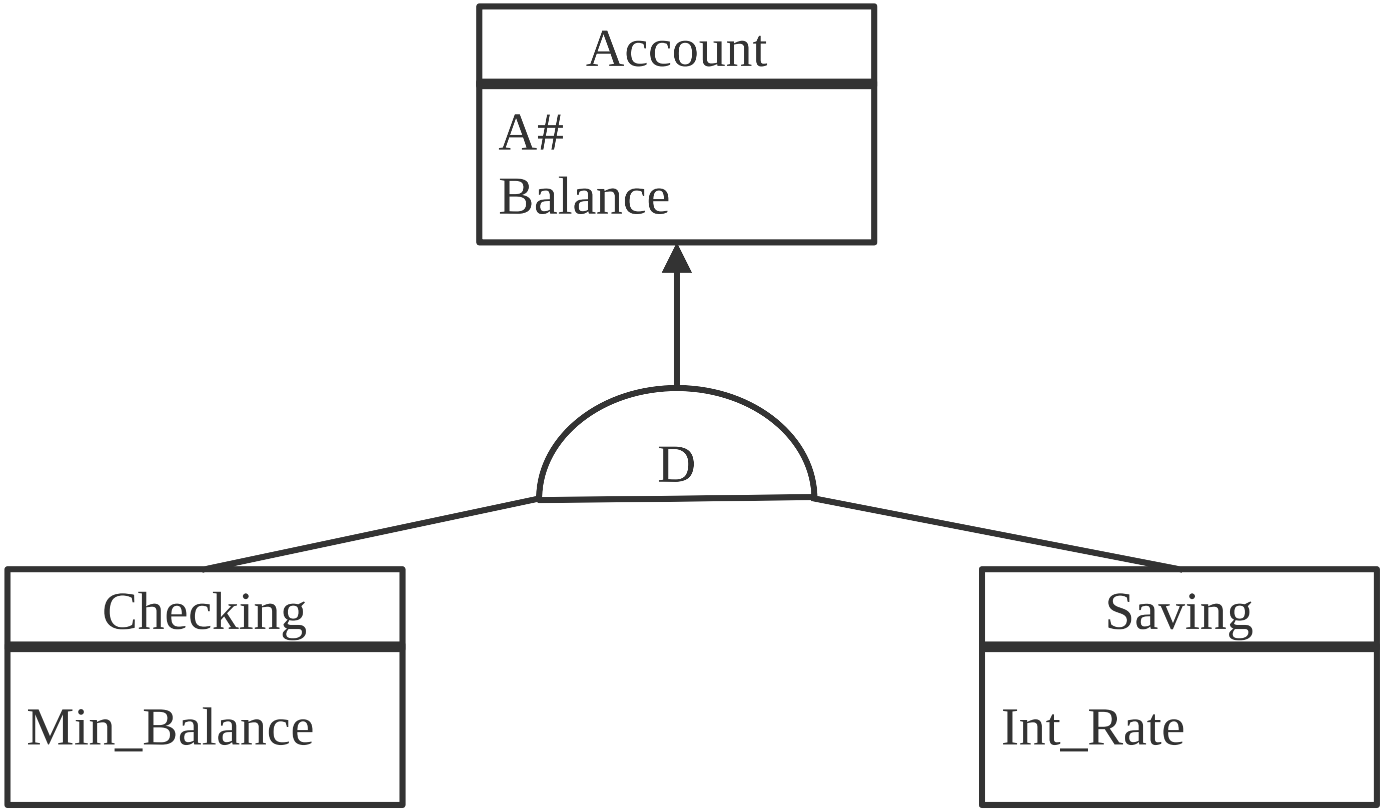
ER Model

- Example:
  - A Part is either a manufactured part or a purchased part
  - The same part could be a purchased part or a manufactured part
  - This is a **total overlapping** specialization



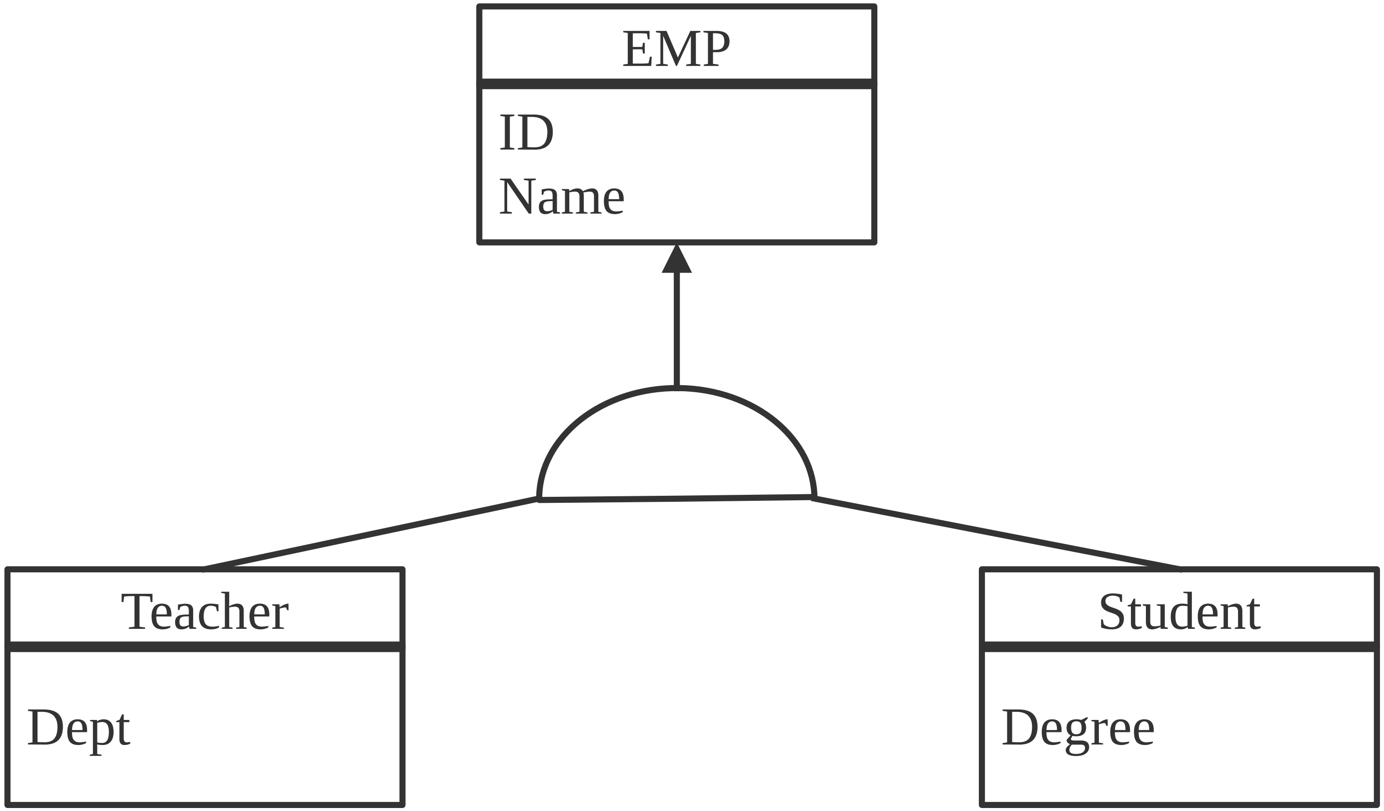
ER Model

- Example
  - An Account is either a checking account or a savings account
  - But there other types of accounts (Money Market, Loans, Etc.)
  - We do not care about other accounts' special attributes. Therefore we do not have them as sub-types
  - This is a **non-total disjoint** specialization



ER Model

- Example:
  - An employee is either a teacher or a student. There are other employees in the organization (non-total)
  - Some of the students in the university also teach (overlapping)
  - This is a **non-total overlapping** specialization



Relational Model
and
Mapping from ER

Relational Model

Conceptual Data Model

Objects

Relationships between Objects

Entity Relationship Diagram (models your business)

Logical Data Model

Relational Schemas

Constraints

Dependencies

Relational Model (a logical model of your business)

Applies to this layer

Physical Data Model

Physical Database Objects

Create Table, Index, etc

Physical Model (Database Specific)

Relational Model Concepts

- ***The Relational Data Model***
  - The model was first introduced by Dr. E.F. Codd of IBM in 1970 in the following article
    - ***"A Relational Model for Large Shared Data Banks,"*** *Communications of the ACM*, June, 1970.
  - It is based on the ***set*** concept (from ***mathematics***)
  - It supports simple and powerful data querying or retrieval languages (Relational Algebra, SQL)
  - The model also supports statements for manipulation of the contents of the database

Relational Model Concepts

- ***Relational model*** is the de facto standard for representation of the logical model
  - In this model, a database is represented as a set of ***Relations (or tables)*** **–** *that is why the name relational model*
  - Each relation/table has a ***unique name***
  - Each relation/table has a set of one or more ***Attributes/columns***
  - The collection of all attributes of a relation is called a ***Tuple/row***
  - The collection of all relations for a database is called the ***Schema*** of the database
  - Textually, we show a table as

    Employee (<u>EID</u>, Name, Sal, DOB, Dno)

Relational Model Concepts

Columns

Shipment

Rows

s#	status	city	p#	qty
s1	20	London	p1	300
s1	20	London	p2	200
s1	20	London	p3	400
s1	20	London	p4	200
s1	20	London	p5	100
s1	20	London	p6	100
s2	10	Paris	p1	300
s2	10	Paris	p2	400
s3	10	Paris	p2	200
s4	20	London	p2	200
s4	20	London	p4	300
s4	20	London	p5	400

Number of rows = 12
Number of columns = 5

Relational Model Concepts

Definitions

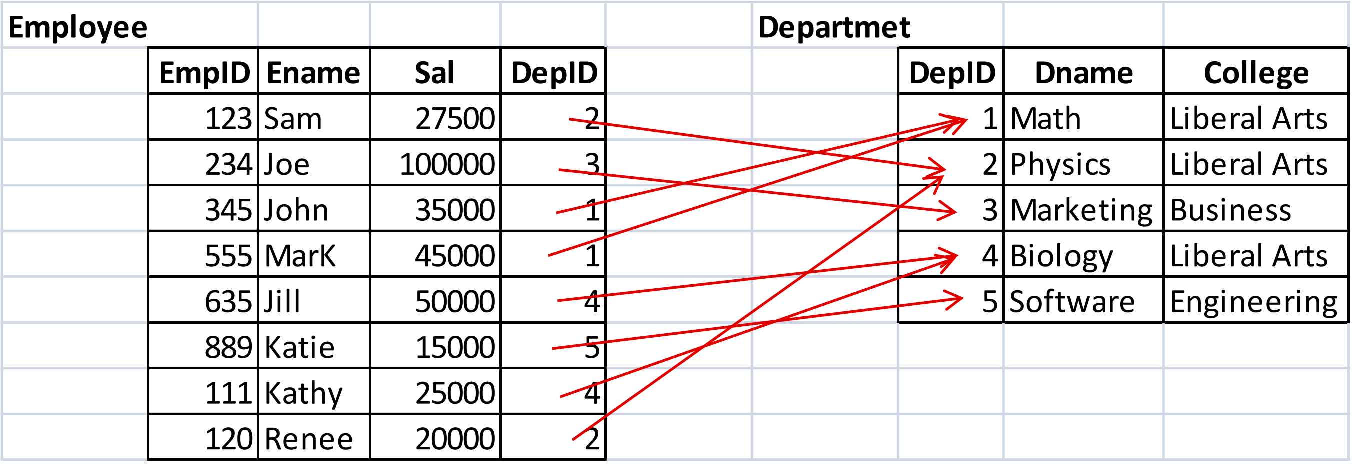
Candidate Key A minimal superkey that does not contain a subset of attributes that is itself a superkey

Primary Key A chosen candidate key that uniquely identifies all other attribute values in any given row but may NOT contain null entities

Foreign Key An attribute or combination of attributes in one table whose values must either match the primary key in another table or be null

Relational Model Concepts

- ***Referential Integrity***
  - DeptID in Employee table is a foreign key pointing to the DeptID primary key in Department table
  - Department table is called the parent table
  - Employee table is called the child table
  - Referential Integrity rule says, “there are no orphaned children rows in the child table.



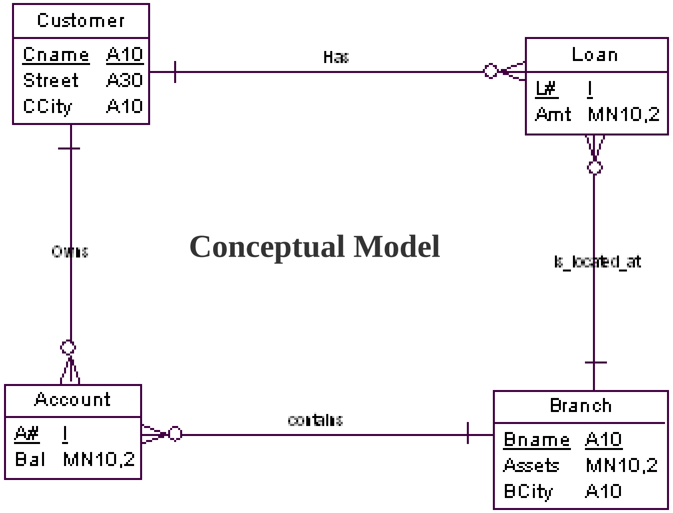
Employee

EmpID	Ename	Sal	DepID
123	Sam	27500	2
234	Joe	100000	3
345	John	35000	1
555	MarK	45000	1
635	Jill	50000	4
889	Katie	15000	5
111	Kathy	25000	4
120	Renee	20000	2

Departmet

DepID	Dname	College
1	Math	Liberal Arts
2	Physics	Liberal Arts
3	Marketing	Business
4	Biology	Liberal Arts
5	Software	Engineering

Three Model View



Conceptual Model

Logical (Relational)

```
Customer(Cname, Street, Ccity)
Branch(Bname, Assets, Bcity)
Account(A#, Bal, Cname, Bname)
Loan(L#, Amt, Cname, Bname)
```

CNAME	STREET	CCITY
Baba	1234 Ireland	Minnetonka
Cook	2222 Main	MPLS
Erickson	1234 Second	Eagan
Ireland	1234 Third	Eagan
James	2222 Fourth	Edina
Janzen	1234 Fifth	Edina
Johnson	3333 Main	MPLS
Jones	1111 Main	MPLS
Love	1234 Ninth	Minnetonka
Rahimi	1111 Sixth	Eden Prairie
Tomcat	1234 Seventh	MPLS
Tones	1234 Eighth	Minnetonka
Melcher	1234 Nineth	MPLS
Woods	1234 Old	Minnetonka
Tomczak	1111 newest	Minnetonka

L#	CNAME	BNAME	AMT
1001	Cook	France	10000
2002	Rahimi	Southdale	5000
3003	Cook	York	1000
4004	Rahimi	France	20000
5005	Rahimi	York	11000
6006	Cook	Main	30000
7007	Jones	France	4500
8008	Jones	Southdale	1200
9009	Jones	York	9000
1004	Tomczak	France	34560
4009	James	France	12300
9008	Melcher	France	4450
8888	Woods	France	4450

A#	CNAME	BNAME	BAL
1111	Rahimi	France	123.5
2222	Rahimi	Southdale	1111
3333	Rahimi	York	45
4444	Cook	France	6690
5555	Cook	York	10000
6666	Cook	Main	3338
7777	Jones	France	5678
8888	Jones	Southdale	999
9999	Jones	York	456
1234	Tomczak	France	123.5
4567	Love	France	300
9800	Love	Main	333
7765	James	France	100
7756	Woods	Eight Street	1000

BNAME	ASSETS	BCITY
3rd Street	100000	Eagan
Airport	446000	Eagan
Eight Street	450000	MPLS
France	125000	Edina
Main	100000	Minnetonka
Northtown	347000	MPLS
Ridgedale	150000	Minnetonka
Second	123000	Minnetonka
Southdale	750000	Edina
York	300000	Edina

Physical (Relational)

Entity Relationship Diagram

Mapping to

Relational Model

Mapping

- How do we realize an ER model by a relational model (Schema)?
- Automated process (using a tool)
- Non-automated process (paper and pencil)
- Key Rules:
  - Each entity maps to a table
  - Each attribute maps to a column – the data type of the column is the date type of the attribute
  - A relationship maps either
    - To a table, or
    - As a Fkey column

Mapping – Examples

Single Entity Mapping

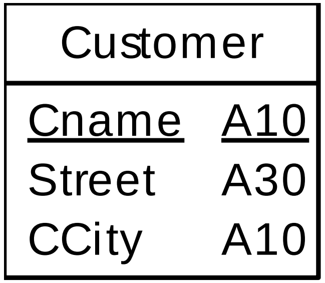
Conceptual

Customer	
Cname	A10
Street	A30
CCity	A10

Mapping – Examples

<u>Single Entity Mapping</u>

<u>Conceptual</u>



Customer

<u>Cname</u> <u>A10</u>
Street A30
CCity A10

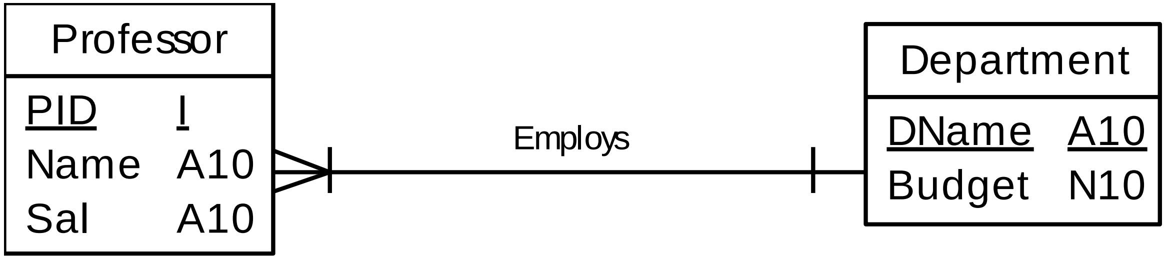
<u>Logical</u>

Customer (<u>Cname</u>, Street, CCity)

Mapping – Examples

Relationship with no Attributes

Conceptual



Professor	
PID	I
Name	A10
Sal	A10

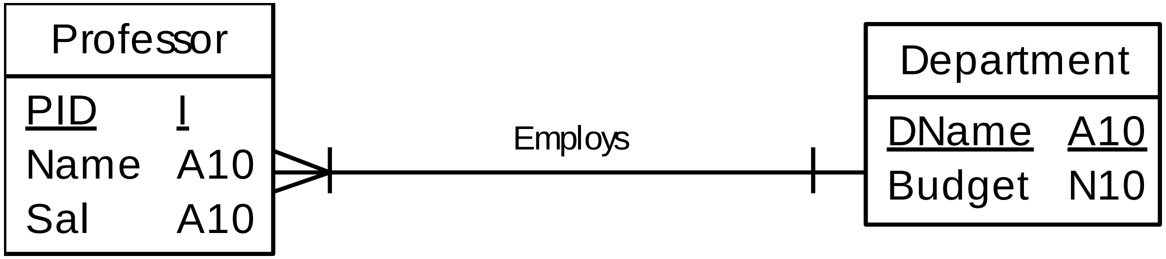
Employs

Department	
DName	A10
Budget	N10

Mapping – Examples

Relationship with no Attributes

Conceptual



Professor

PID I
Name A10
Sal A10

Employs

Department

DName A10
Budget N10

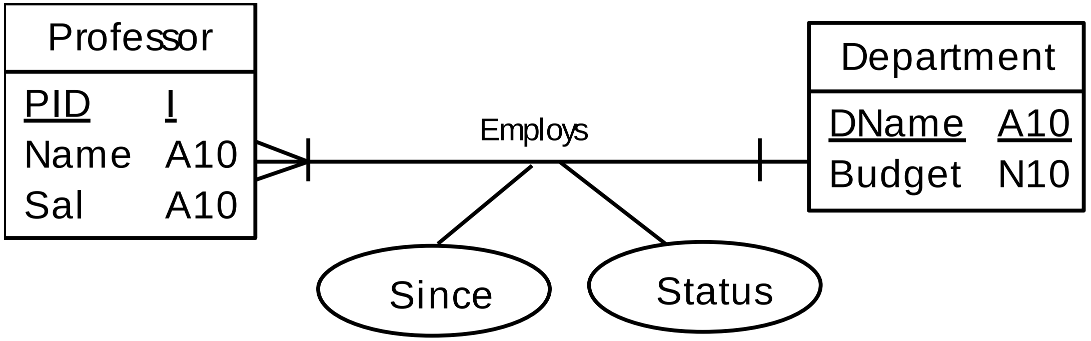
Logical

Department (*DName, Budget*)
Professor *(PID, Name, Sal, DName)*

Mapping – Examples

Relationship with Attributes (1 to M)

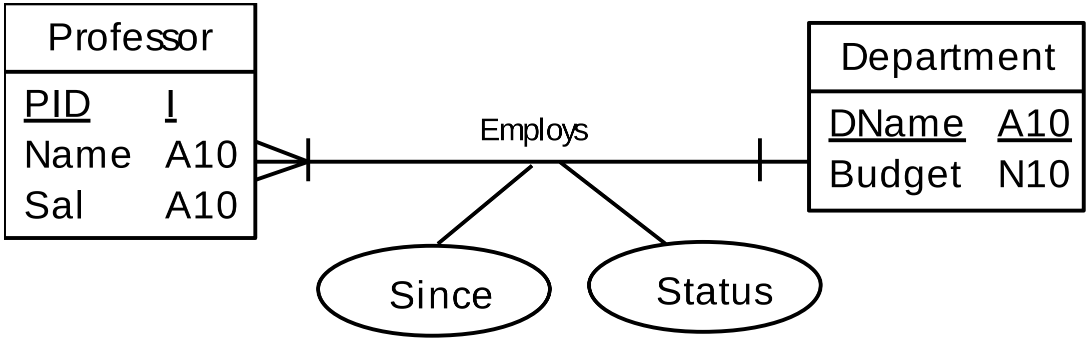
Conceptual



Mapping – Examples

Relationship with Attributes (1 to M)

Conceptual



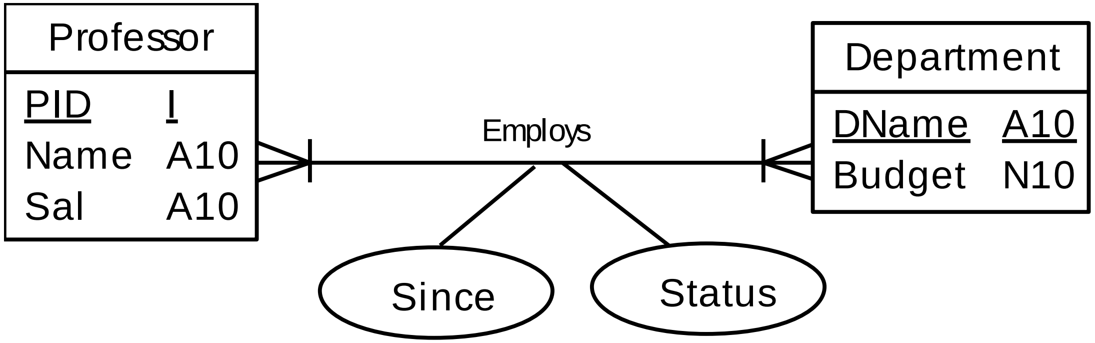
Logical

Department (*DName, Budget*)
Professor *(PID, Name, Sal, Dname, Since, Status)*

Mapping – Examples

Relationship with Attributes (M to M)

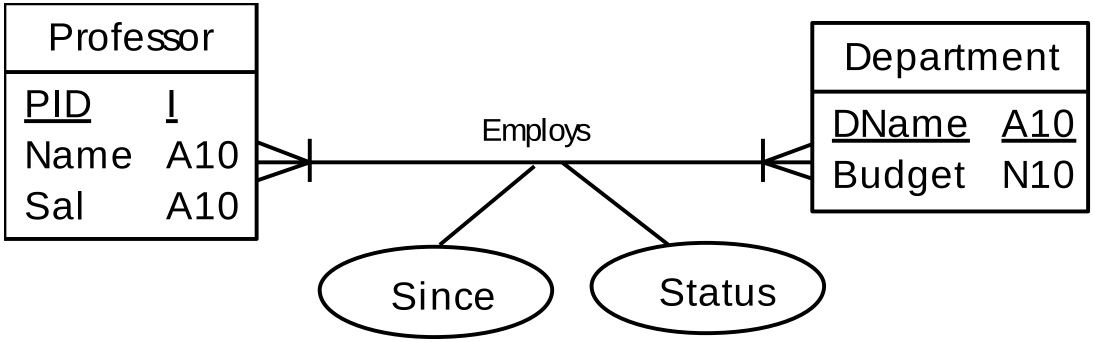
Conceptual



Mapping – Examples

Relationship with Attributes (M to M)

Conceptual



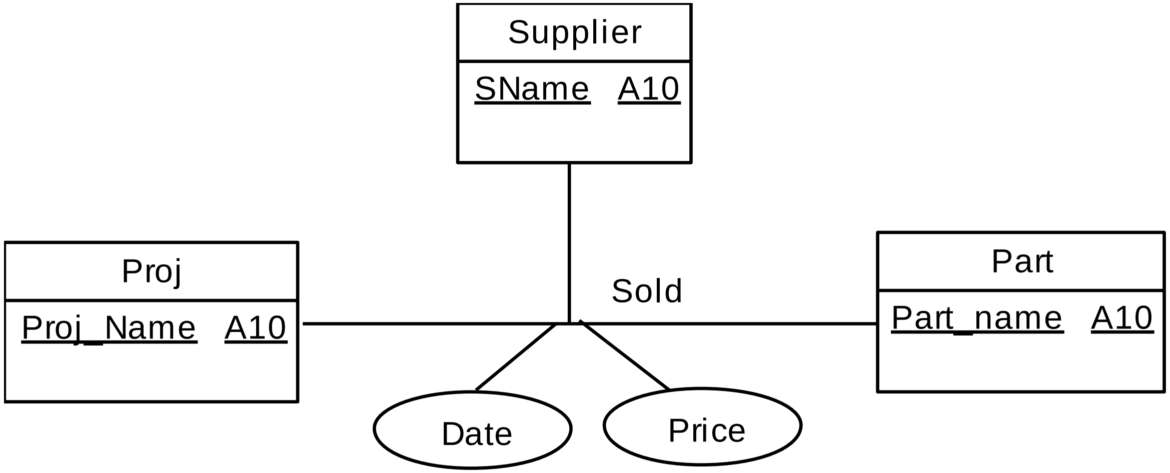
Logical

Department (*DName, Budget*)
Professor *(PID, Name, Sal)*
Employs (DName, PID, Since, Status)

Mapping

- ***Ternary Relationship with Attributes***

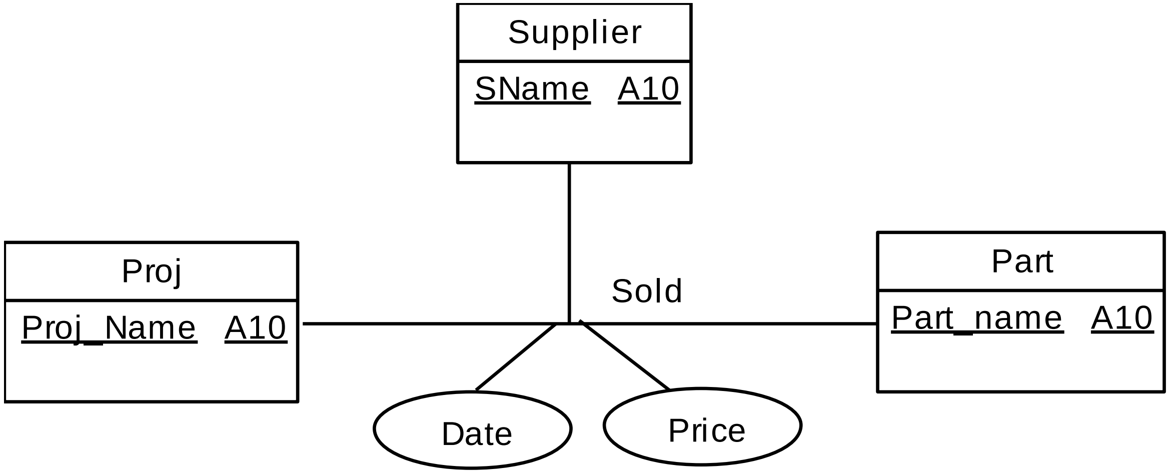
Conceptual



Mapping

- ***Ternary Relationship with Attributes***

Conceptual

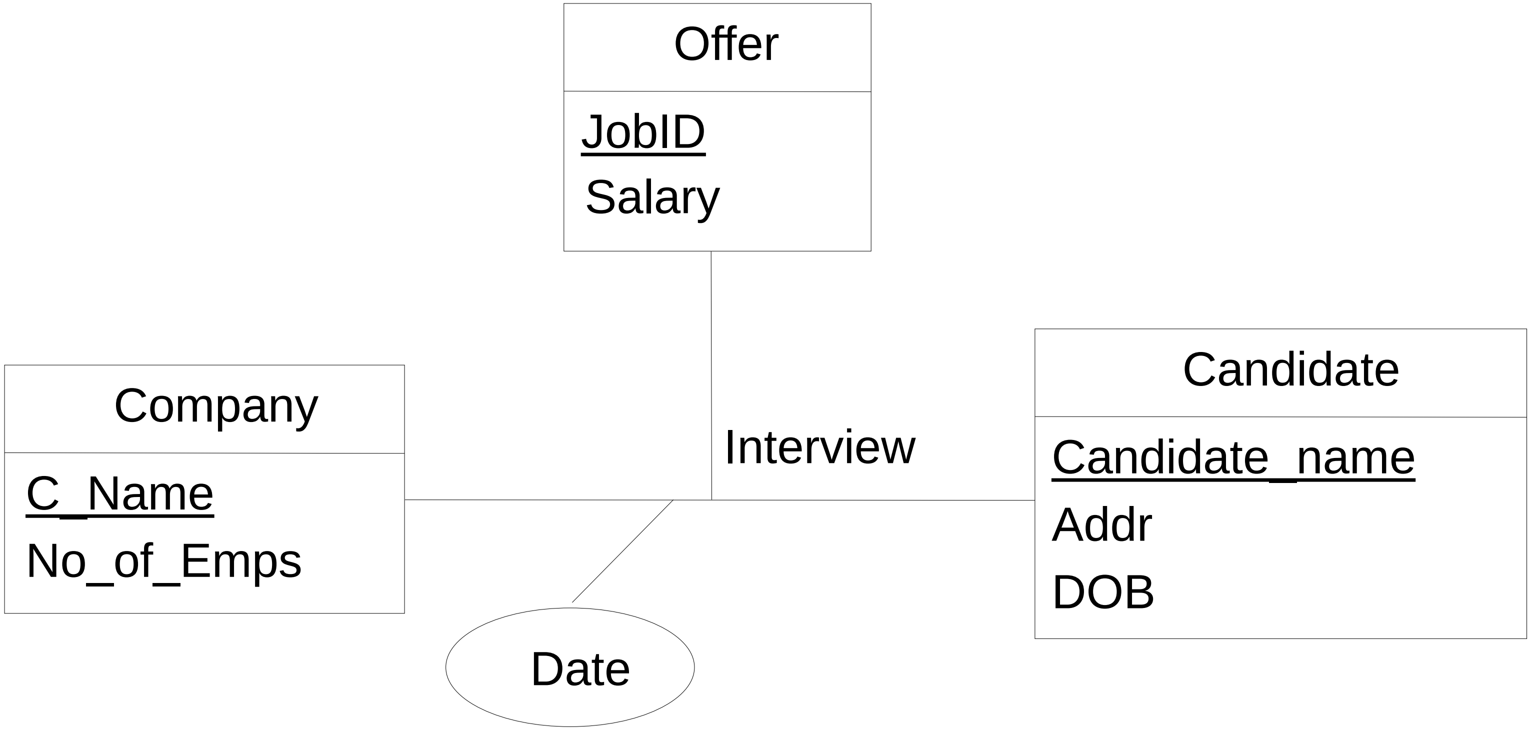


Logical

Proj (*Proj_Name*)
Supplier (*SName*)
Part (*Part_Name*)
Sold (*Proj_Name*, *SName*, *Part_name*, *Date, Price*)

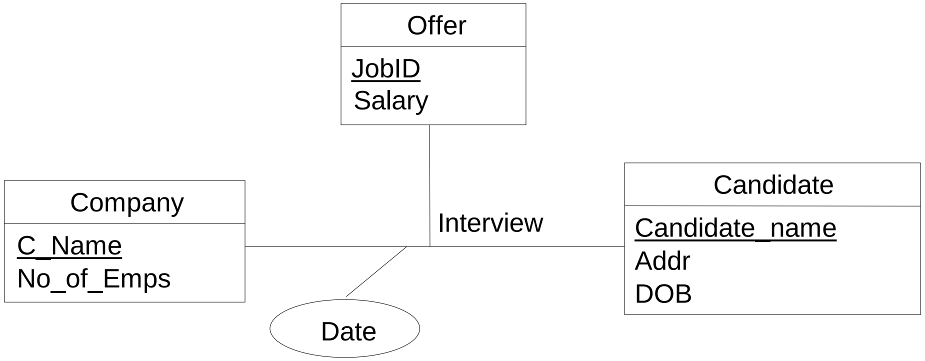
Mapping

- Question:
  - A company interviews some candidates and *may* make some offers for the job opening the company has
  - Map the ERD to set of tables.



Mapping

- Question:
  - A company interviews some candidates and *may* make some offers
  - Map the ERD to set of tables.



Logical

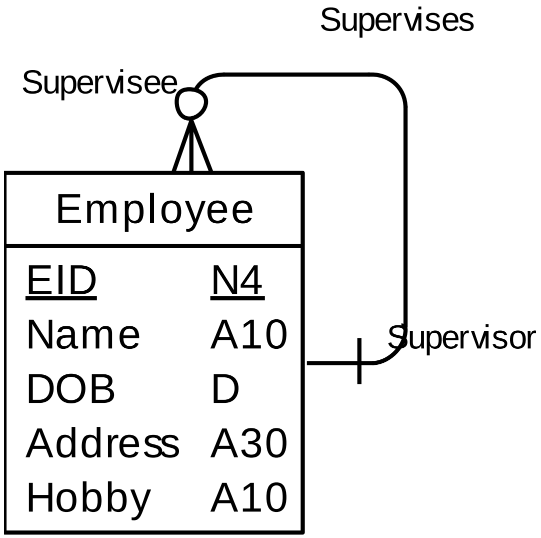
Company (*C_Name*, *No_of_Emps*)
Candidate(*CandidateName*, *Addr*, *DOB*)
Offer (JobID, Salary)
Interview(*C_Name*, *Candidate_Name*, JobID, Date)

Not part of the key

Mapping

- ***Recursive Relationship***

Conceptual

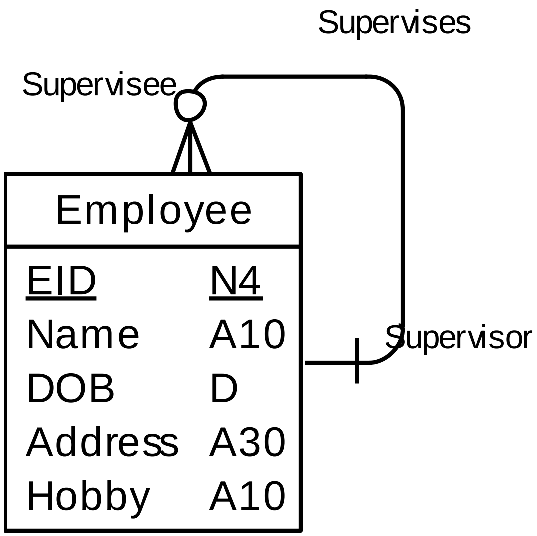


EID	N4
Name	A10
DOB	D
Address	A30
Hobby	A10

Mapping

- ***Recursive Relationship***

Conceptual



Supervises

Supervisee

Supervisor

Employee	
EID	N4
Name	A10
DOB	D
Address	A30
Hobby	A10

Logical

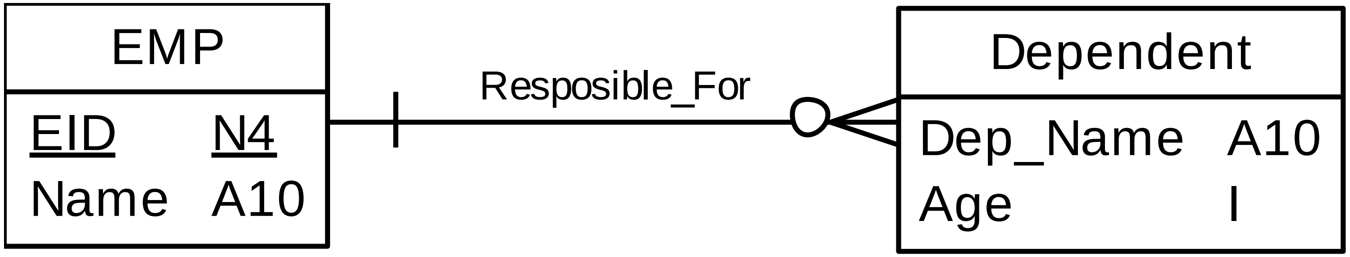
Employee (*EID*, *Name, DOB, Address, Hobby, Supervisor_EID*)

Foreign Key

Mapping

- ***Weak Entity Relationships***

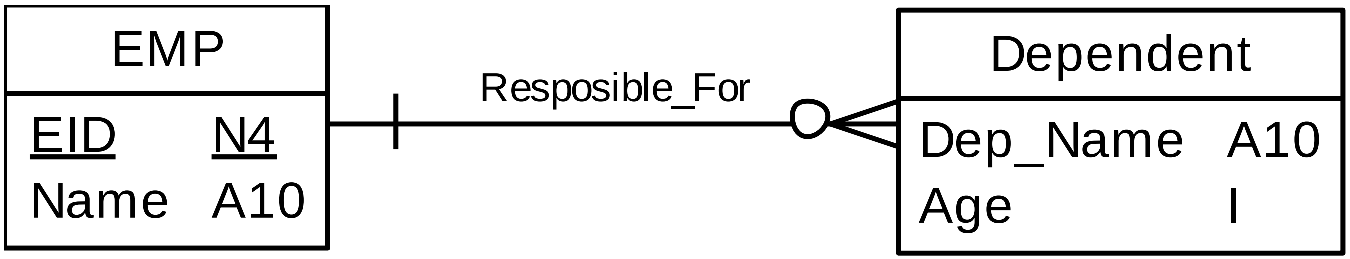
Conceptual



Mapping

- ***Weak Entity Relationships***

Conceptual



EMP

EID N4

Name A10

Resposible_For

Dependent

Dep_Name A10

Age I

Logical

EMP(EID, Name)

DEPENDENT(Dep_Name, EID, Age)

Note: The EID is a Fkey in the Dependent table, but since the Dependent is a Weak entity depending on The Employee entity, we have to make EID of the Pkey of the Dependent

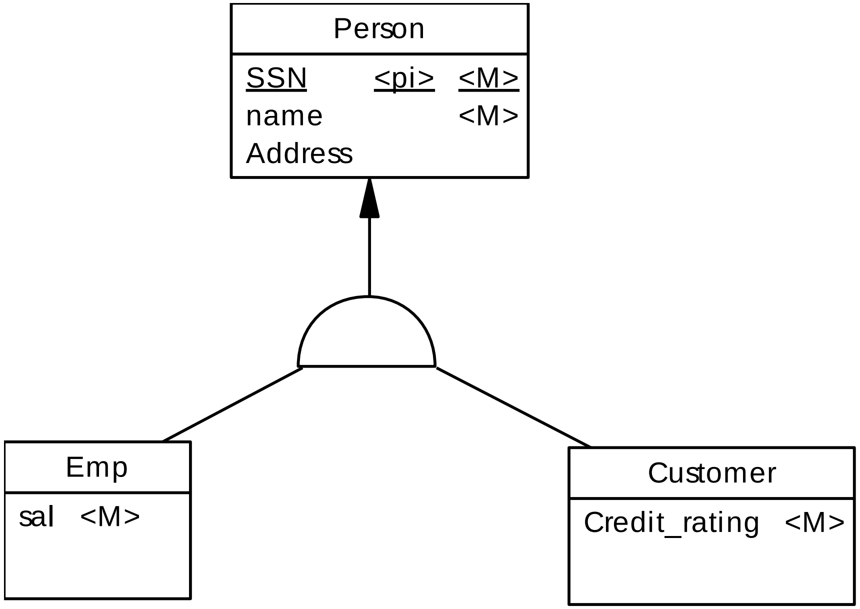
Mapping

Inheritance

Alternative 1:

Map All

Conceptual



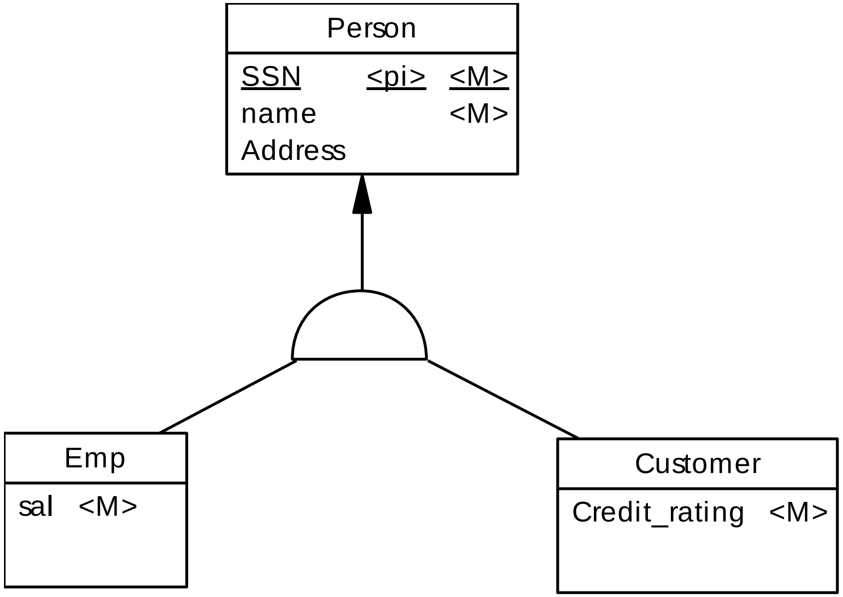
Mapping

Inheritance

Alternative 1:

Map All

Conceptual



Logical

Person (*SSN*, Name, Address)

Customer (*SSN*, Credit_Rating)

Emp (*SSN*, Sal)

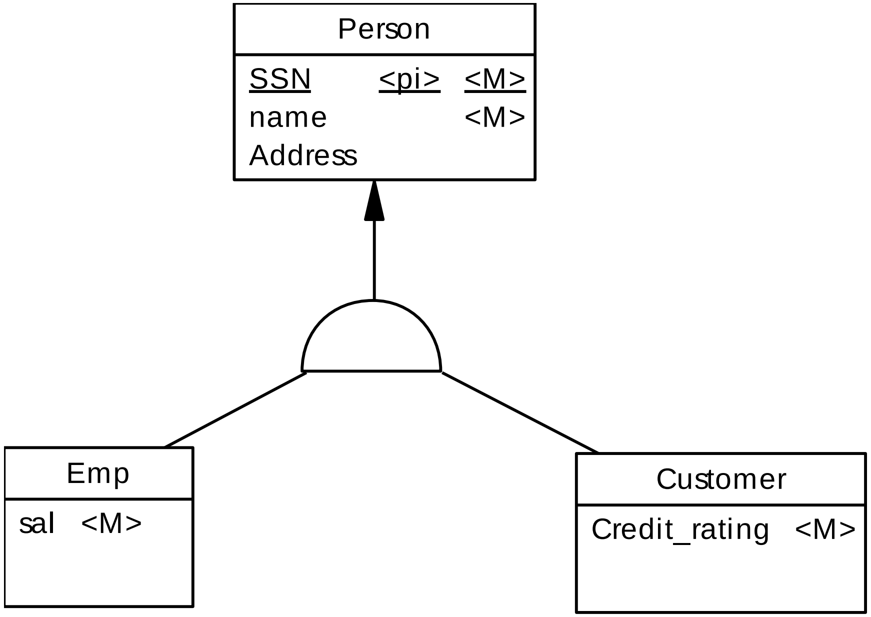
Mapping

Inheritance

Alternative 2:

Map Children

Conceptual



Person

SSN <pi> <M>

name <M>

Address

Emp

sal <M>

Customer

Credit_rating <M>

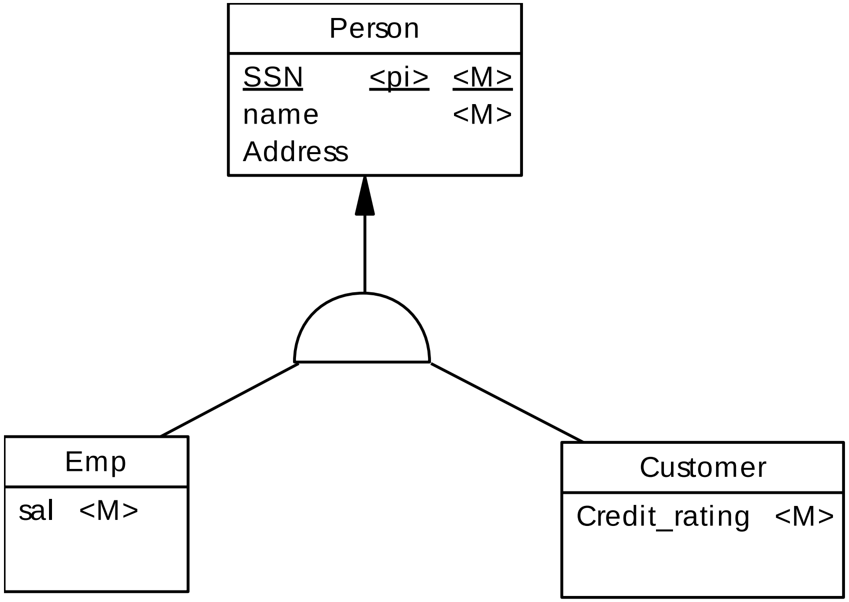
Mapping

Inheritance

Alternative 2:

Map Children

Conceptual



Logical

Customer(*SSN*, Name, Address, Credit_Rating)

Emp(*SSN*, Name, Address, Sal)