GEO 874 I HS25
Universität Zürich

4. Lecture Introduction to Databases

Conceptual Design (Part II)

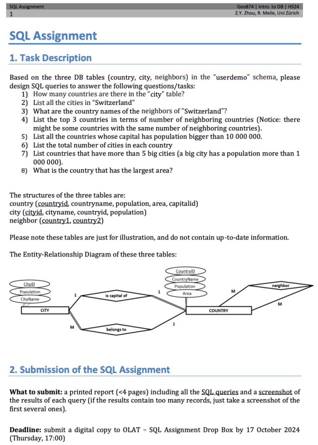
Esra Suel
Dept. of Geography, University of Zürich

Rolf Meile
Eidg. Forschungsanstalt für Wald, Schnee und Landschaft (WSL)

kudos to Dr. Zhiyong Zhou, Dr. Cheng Fu & Dr. Haosheng Huang for providing this document

SQL Assignment DL 16.10.2025 (next Thursday)

- Based on the three DB tables in the „userdemo“ schema, design SQL queries to answer tasks.
  - More details on: https://www.geo.uzh.ch/microsite/geo874/scripte/geo874_HS25_SQL_Assignment.pdf



SQL Assignment | Geo874 | Intro. to DB | HS24
1 | Z.Y. Zhou, R. Meile, Uni Zürich

SQL Assignment

1. Task Description

Based on the three DB tables (country, city, neighbors) in the "userdemo" schema, please design SQL queries to answer the following questions/tasks:

1) How many countries are there in the "city" table?
2) List all the cities in "Switzerland"
3) What are the country names of the neighbors of "Switzerland"?
4) List the top 3 countries in terms of number of neighboring countries (Notice: there might be some countries with the same number of neighboring countries).
5) List all the countries whose capital has population bigger than 10 000 000.
6) List the total number of cities in each country
7) List countries that have more than 5 big cities (a big city has a population more than 1 000 000).
8) What is the country that has the largest area?

The structures of the three tables are:
country (countryid, countryname, population, area, capitalid)
city (cityid, cityname, countryid, population)
neighbor (country1, country2)

Please note these tables are just for illustration, and do not contain up-to-date information.

The Entity-Relationship Diagram of these three tables:

2. Submission of the SQL Assignment

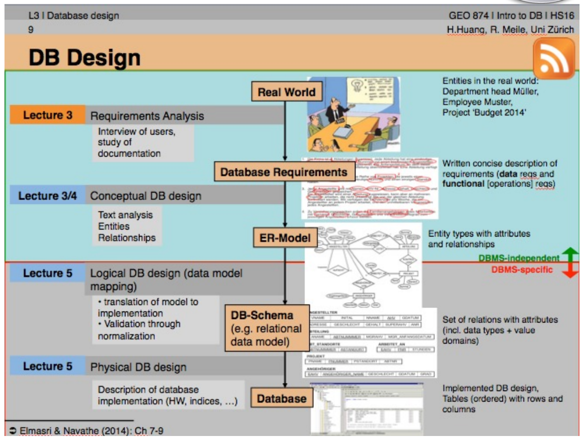
What to submit: a printed report (<4 pages) including all the SQL queries and a screenshot of the results of each query (if the results contain too many records, just take a screenshot of the first several ones).

Deadline: submit a digital copy to OLAT – SQL Assignment Drop Box by 17 October 2024 (Thursday, 17:00)

- 30% of the final grade
- Individual work

- **What to submit:** a report (<4 pages) including all the SQL queries and a screenshot of each query.
- **How to submit:** submit to OLAT
- **Deadline:** by **16 October 2025** (Thursday, 18:00)

Last week



Phases of DB design

- Requirements analysis
- Conceptual DB design (e.g, ER-Model)
- Logical DB design (e.g., relational Model)
- Physical DB design (physical implementation)

Requirements analysis

- Identification of user groups and application area
- Information gathering and structuring (use cases, interviews, documentation…)
- Leads to conceptual design: identification of objects and attributes

Intro to ER modelling

- Entities, Entity types, Entity sets
- Attributes and Attribute types

Exercise 3.1: Conceptual design of the DB "COMPANY"

1. The company is organized into departments. 1 Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager. A department can be spread over multiple locations.
2. Each department controls a number of projects. 2 Each project has a name, number and is located at a single location.
3. We store each employee's 3 social security number (AHV-Nr), name address, salary, sex, and birthdate. Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.
4. Each employee may have a number of 4 dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Entity type

Attribute

Exercise 3.1: Conceptual design of the DB "Company"

Perform a coarse conceptual design based on *Entity types and Attributes (ignoring relationships)*

Department
- Name, <u>Number</u>, {Location}, Manager, ManagerStartDate

Project
- Name, <u>Number</u>, Location, ManagingDepartment

Employee
- Name (FName, LName), <u>AHV</u>, Sex, Address, BDate, Dept, Supervisor, {Works on(Project, Hours)}

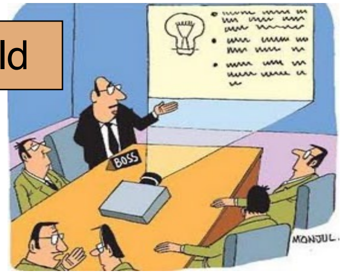
Dependent
- Employee, DependentName, Sex, BDate, Relationship

{} – multivalued attribute

() – composite attribute

DB Design

Real World

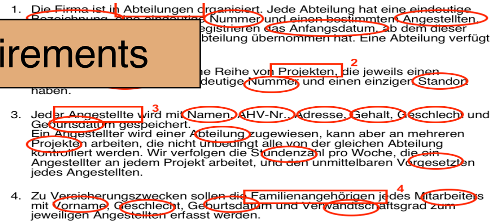


Entities in the real world:
Department head Müller,
Employee Muster,
Project 'Budget 2014'

Lecture 3 – Requirements Analysis

Interview of users, study of documentation

Database Requirements


1. Die Firma ist in Abteilungen organisiert. Jede Abteilung hat eine eindeutige
Bezeichnung, eine eindeutige Nummer und einen bestimmten Angestellten
egistrieren das Anfangsdatum, ab dem dieser
bteilung übernommen hat. Eine Abteilung verfügt

e Reihe von Projekten, die jeweils einen
deutige Nummer und einen einzigen Standort
haben.

3. Jeder Angestellte wird mit Namen, AHV-Nr., Adresse, Gehalt, Geschlecht und Geburtsdatum gespeichert.
Ein Angestellter wird einer Abteilung zugewiesen, kann aber an mehreren Projekten arbeiten, die nicht unbedingt alle von der gleichen Abteilung kontrolliert werden. Wir verfolgen die Stundenzahl pro Woche, die ein Angestellter an jedem Projekt arbeitet, und den unmittelbaren Vorgesetzten jedes Angestellten.

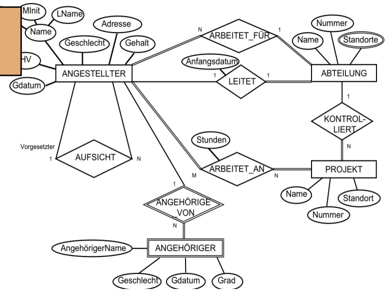
4. Zu Versicherungszwecken sollen die Familienangehörigen jedes Mitarbeiters mit Vorname, Geschlecht, Geburtsdatum und Verwandtschaftsgrad zum jeweiligen Angestellten erfasst werden.

Written concise description of requirements (data reqs and functional [operations] reqs)

Lecture 4 – Conceptual DB design

Text analysis, Entities, Relationships

ER-Model



Entity types with attributes and relationships

DBMS-independent
DBMS-specific

Lecture 5 – Logical DB design (data model mapping)

- translation of model to implementation
- Validation through normalisation

DB-Schema (e.g. relational data model)

NGESTELLTER

VNAME	INITAL	NNAME	AHV	GDATUM

DRESSE	GESCHLECHT	GEHALT	SUPERAHV	ANR

BTEILUNG

ANAME	ABTNUMMMER	MGRAHV	MGR_ANFANGSDATUM

BT_STANDORTE

BTNUMMMER	ASTANDORT

ARBEITET_AN

EAHV	PNR	STUNDEN

PROJEKT

PNAME	PNUMMER	PSTANDORT	ABTNR

ANGEHÖRIGER

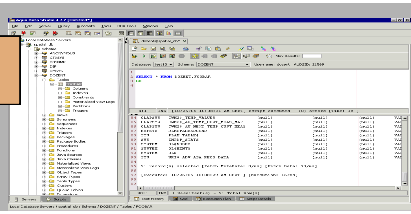
EAHV	ANGEHÖRIGER_NAME	GESCHLECHT	GDATUM	GRAD

Set of relations with attributes (incl. data types + value domains)

Lecture 5 – Physical DB design

Description of database implementation (HW, indices, …)

Database



Implemented DB design, Tables (ordered) with rows and columns

➲ Elmasri & Navathe (2014): Ch 7-9

Learning objectives 4

- ✓ You will be able to define constraints and relationship types
- ✓ You will learn what weak entity types are
- ✓ You will be able to create your own ER models

Contents

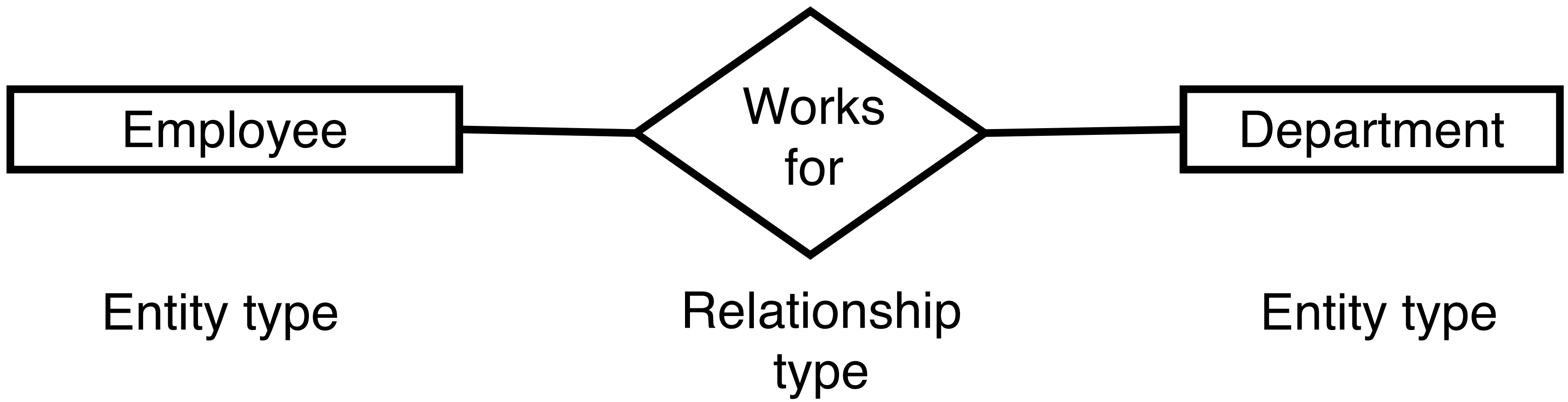
Conceptual DB design (ER-Model cont.)

a. Relationships and their constraints
b. Relationship attributes
c. Weak entity types and partial keys
d. ER Diagram notation
e. Refinement of the design of an ER-Diagram for DB “Company”

Relationship Types

Definition:

- A **Relationship Type** *R* among entity types E_1, E_2, …, E_n *defines* a set of associations among entities from these entity types (**relationship set**)
- Individual entities are said to **participate** in a relationship.
- A relationship type links one or more entity types together
- In ER-Diagram relationship types are represented as diamonds



GEO 874 I Intro to DB I HS25
E. Suel, R. Meile, Uni Zürich

Relationship Types

In Exercise 3.1, we have only covered implicit relationships between entities (e.g., Company) – as attributes of entities that relate to attributes of other entities

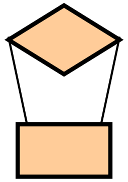
DEPARTMENT
Name, Number, {Locations}, Manager, ManagerStartDate
PROJECT
Name, Number, Location, ManagingDepartment
EMPLOYEE
Name (Fname, LName), AHV, …, BDate, Department, Supervisor, {Works_on(Project, Hours)}
DEPENDENT
Employee, DependentName, Sex, BDate, Relationship

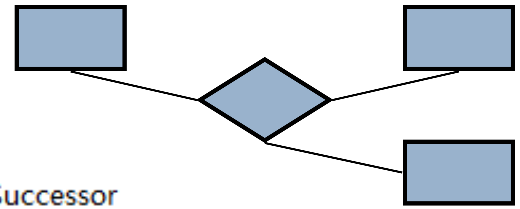
Attribute Manager from DEPARTMENT relates to EMPLOYEE, which leads the department
Attribute ManagingDepartment from PROJECT relates to DEPARTMENT that is reponsible for the project
Attribute Supervisor from EMPLOYEE relates to another employee from EMPLOYEE
Attribute Department from EMPLOYEE relates to DEPARTMENT, for which the employee works
….

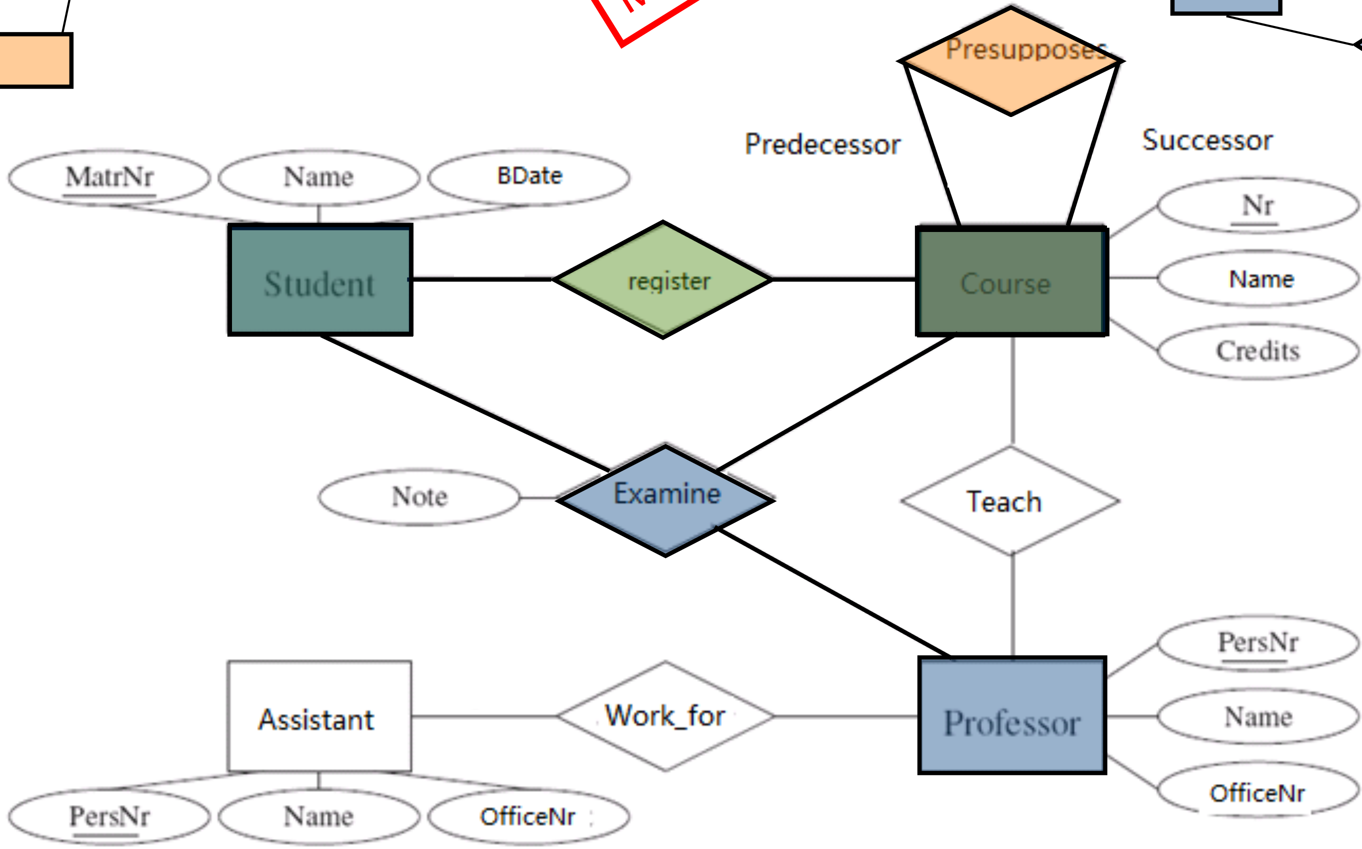
Degree of relationship

Degree of relationship: the number of participating entity types

- recursive (Degree 1): 1 entity type
- Binary (Degree 2): 2 Entity types
- ternary (Degree 3): 3 Entity types





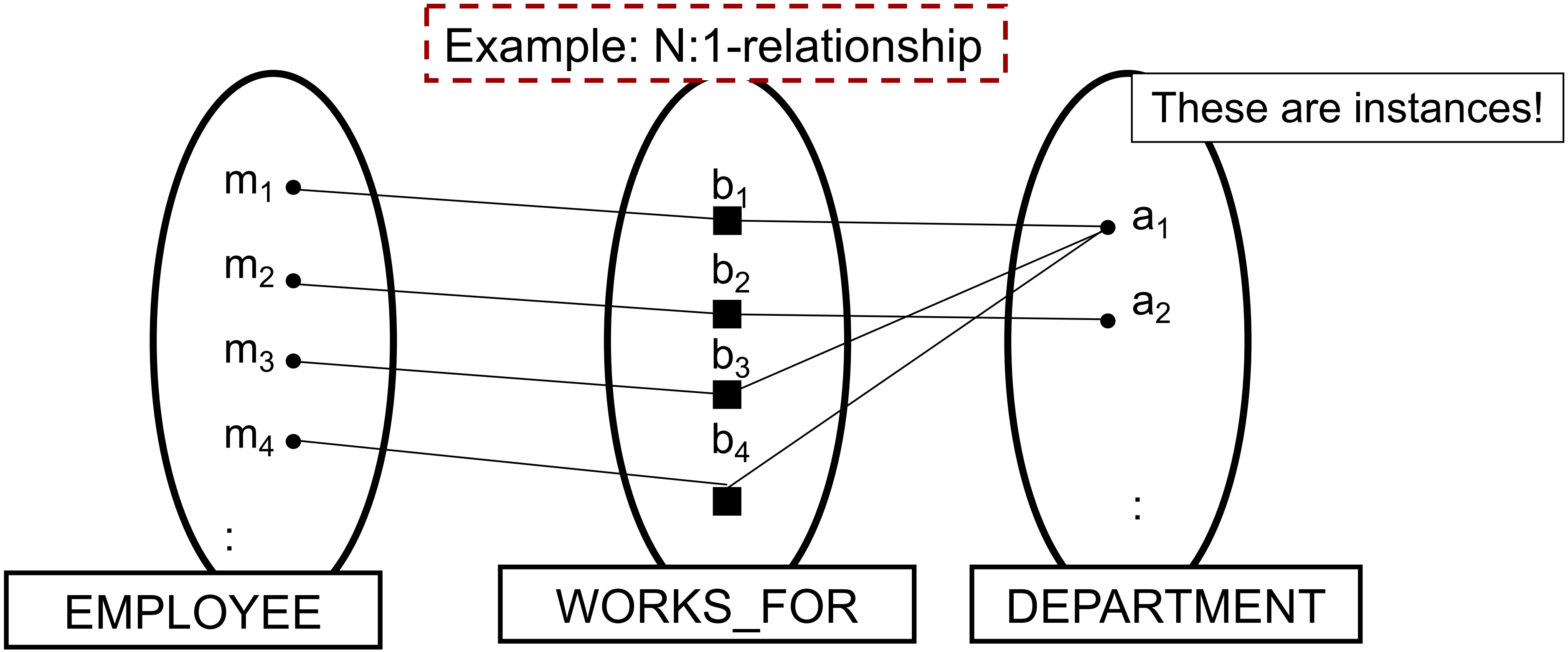


Constraints on relationships

- Relationships may have constraints that restrict the possible combinations of Entities that participate in a relationship set.
- Example - *Company*:
  A rule might state that an Employee can only work for one department
  → this constraint must be represented in the ER-Diagram.
- Two main types of relationship constraints
  1. Cardinality constraint
  2. Participation constraint

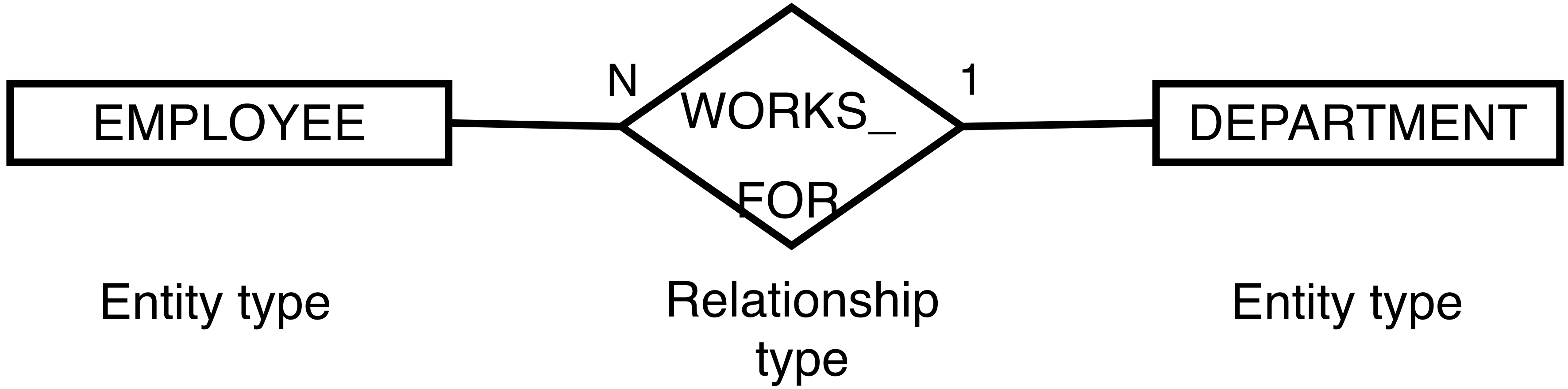
1. Cardinality constraints

The cardinality of a relationship specifies the *maximum* number of relationship instances that an entity can participate in.



1. Cardinality constraints

- In the binary relationship type WORKS_FOR, the cardinality EMPLOYEE:DEPARTMENT is N:1



Each employee works for a single department, but
each department might have multiple employees.

1. Cardinality constraints

Cardinality Example

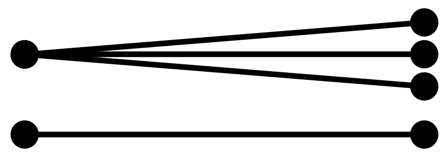
1:1
(one to one)

Employee PC

Each employee has exactly one PC and none else can use it

1:N
(one to many)

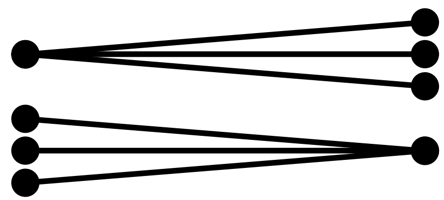
Department Employee



There are multiple employees in a department, but each employee works for a single department

N:M
(many to many)

Projects Employee

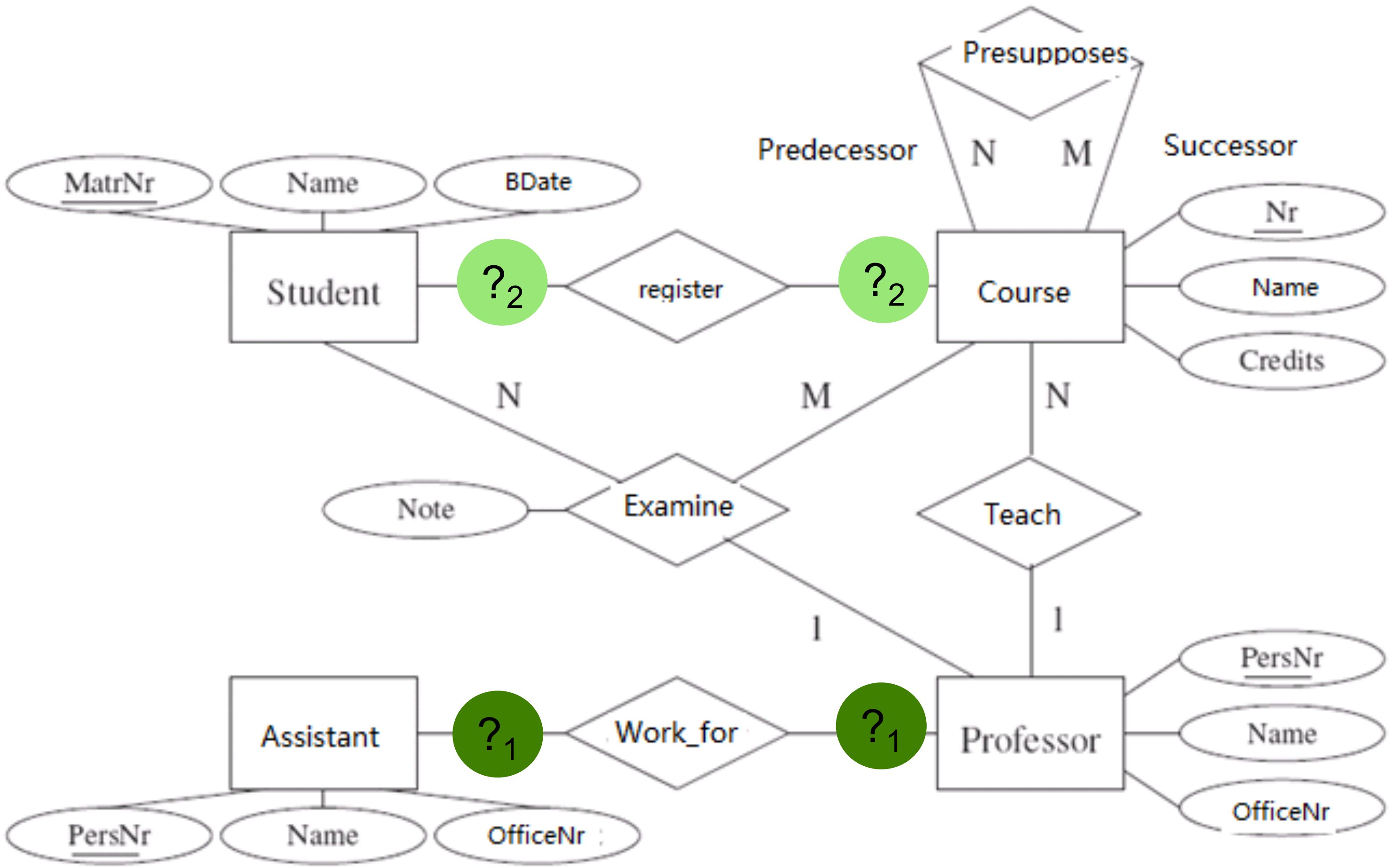


A project involves multiple employees, and each employee can work on multiple projects

Exercise: Cardinalities

Each assistant works for a single professor, but each professor might have multiple assistants.

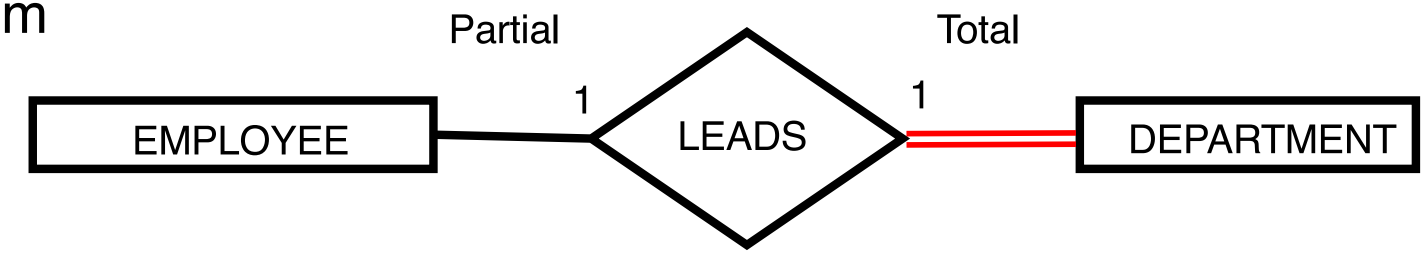
Each students can register for multiple courses, and each course might have multiple students.



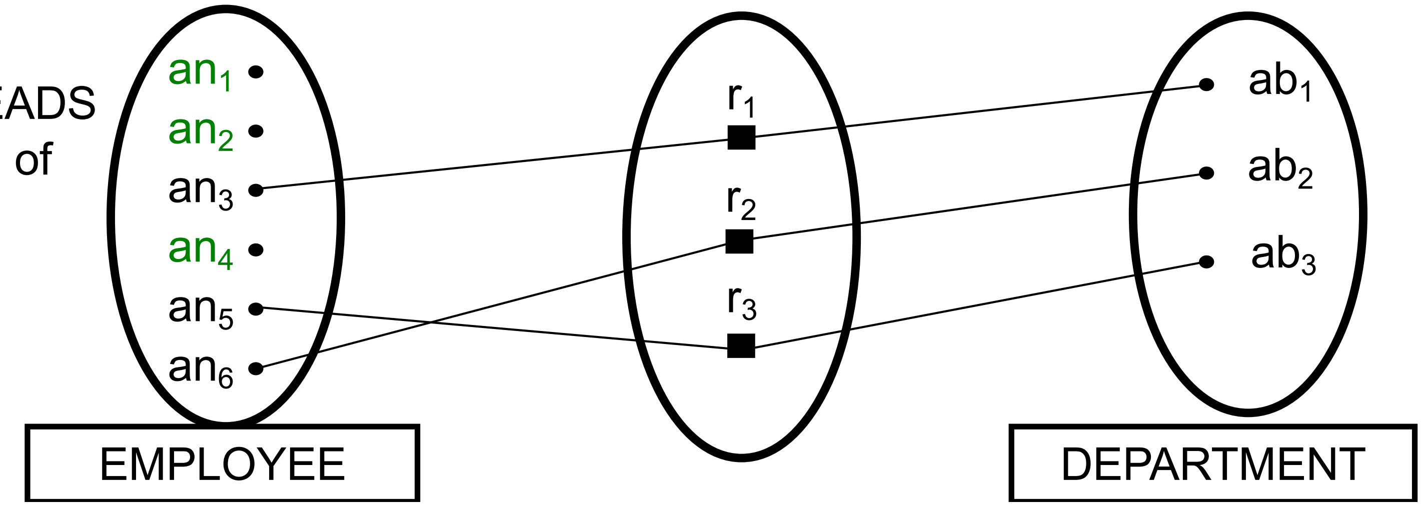
2. Participation constraints

- Specifies, whether all the entity instances of an entity type should participate in the given relationship type.
- Two types of participation constraints
  - **total** (each DEPARTMENT *must be* led by an EMPLOYEE)
  - **partial** (<u>**not**</u> every EMPLOYEE leads a DEPARMTENT)

- In an ER-Diagram the total participation is depicted with double lines



E.g. 1:1 Relationship LEADS
With partial participation of EMPLOYEE and total participation
of DEPARTMENT



Decoding requirements text

Participation constraint

Cardinality

Each employee can lead one department.

Each Department must be led by one Employee

Exercise

- Each Person must live in a single Place
- Each Place can be lived in by one or multiple Persons

Each Person must live in a single Place.

Each Place can be lived in by one or multiple Persons

Exercise

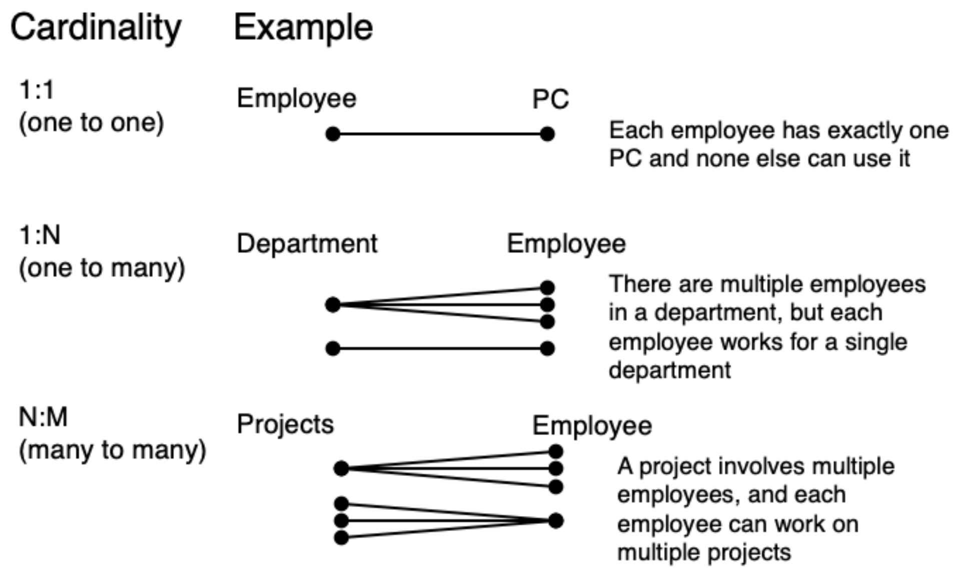
- Each Employee can work on one or multiple Projects
- Each Project can be worked on by one or mulitple Employees

Each Employee can work one one or multiple Projects.

Each Project can be worked on by one or multiple Employees

Summary: Cardinality and Participation constraints

- Cardinality constraints
  - specifies the *maximum* number of relationship instances that an entity can participate in (e.g., each employee maximally works for one dept.).



- Participation constraints
  - Specifies, whether all the instances of an entity type should participate in the given relationship type.
  - Two types of participation constraints
    - **total** (each DEPARTMENT *must be* led by an EMPLOYEE)
    - **partial** (**<u>not</u>** every EMPLOYEE leads a DEPARMTENT)

Contents

Conceptual DB design (ER-Model cont.)

a. Relationships and their constraints
b. Relationship attributes
c. Weak entity types and partial keys
d. ER Diagram notation
e. Refinement of the design of an ER-Diagram for DB “Company”

Relationship attributes

- You have a **rental agreement** for your flat – that is a relationship!
- This rental **starts** at some date, and is backed by some CHF (**rent**);
- ...

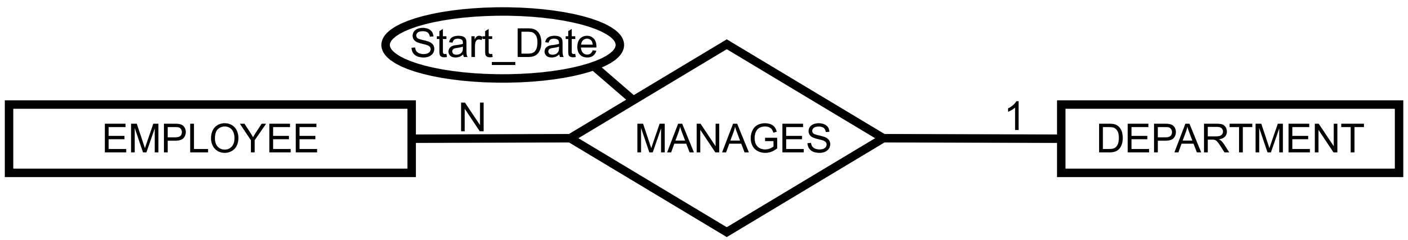
- These facts conceptually relate to the relationship, not to you as a person (you can have multiple tenancies, or none) or to your landlord (he has many clients).

- How do we deal with this in a conceptual DB design?

Relationship attributes

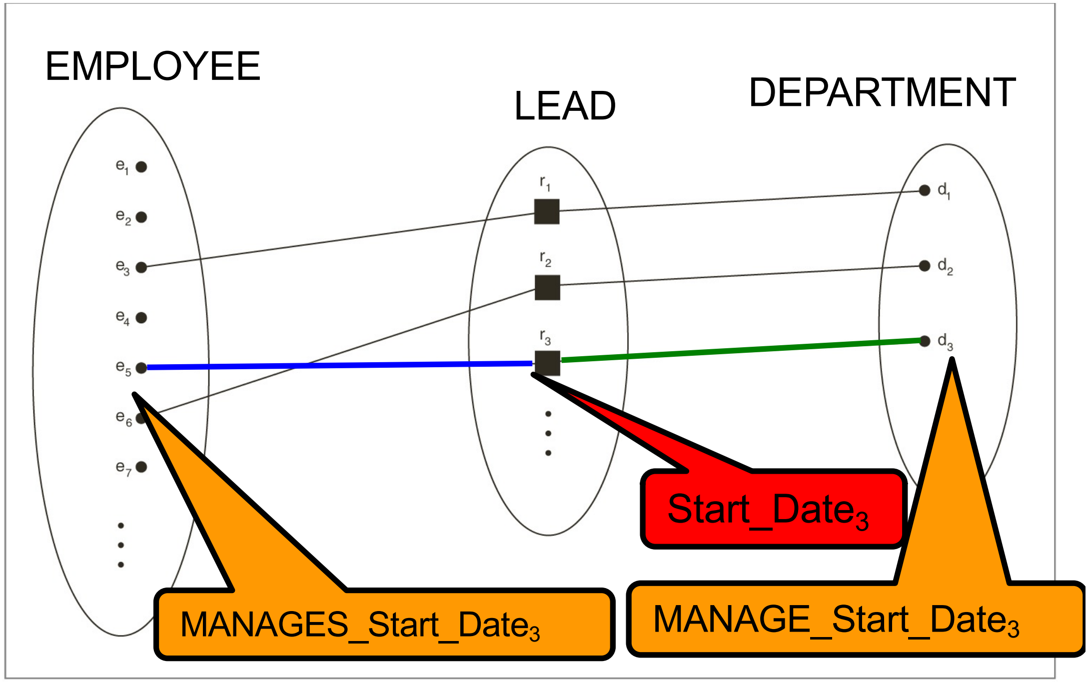
- Modelling attributes of relationships
  - As attributes directly on their relationship types
  - Attributes of 1:1 (and 1:N relationships?) can be migrated to one of the participating entity types.
  - Attributes of M:N relationships must be specified as relationship attributes

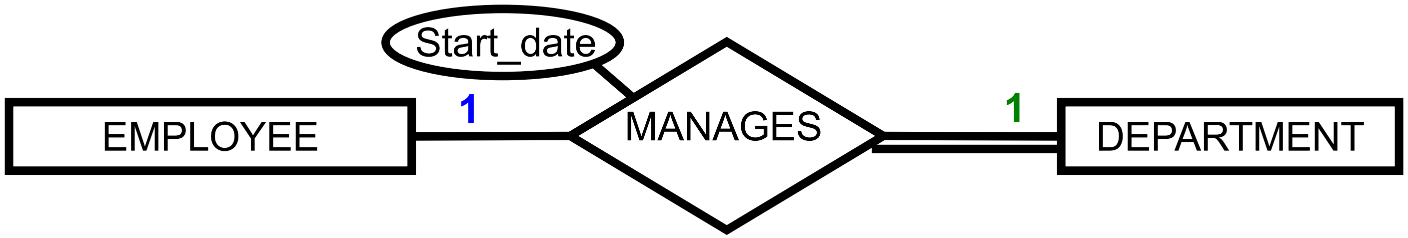
- Could you move Start_Date to any entity below?



Relationship attributes (1:1)

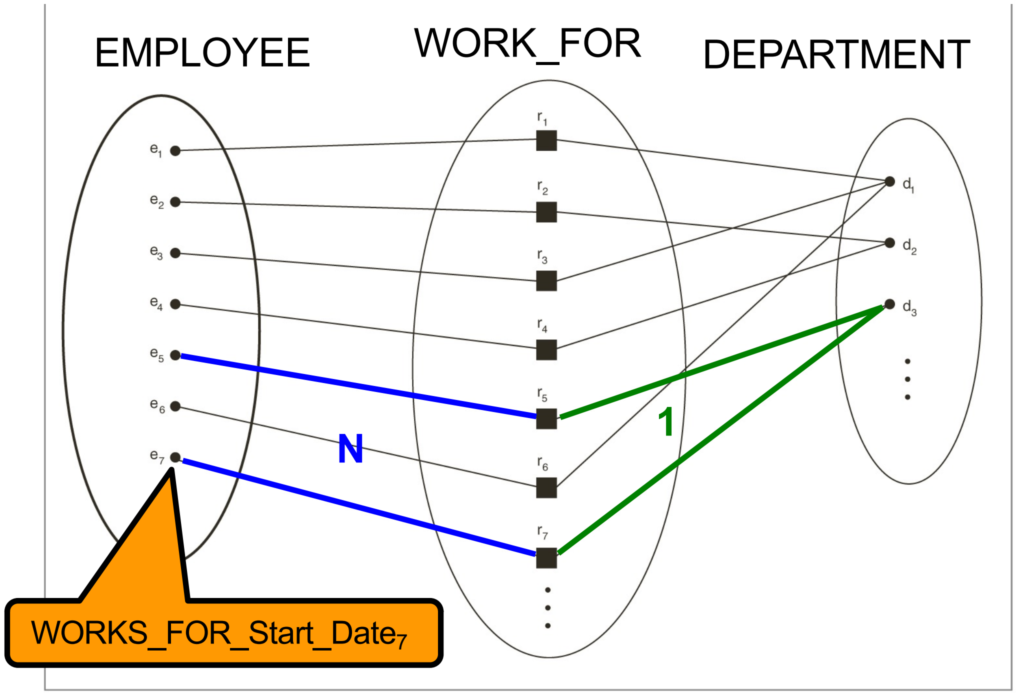
- Attributes of 1:1 relationships– can be migrated to any entity.
  - E.g., Start date attribute for relationship MANAGES can be moved to either EMPLOYEE or DEPARTMENT.

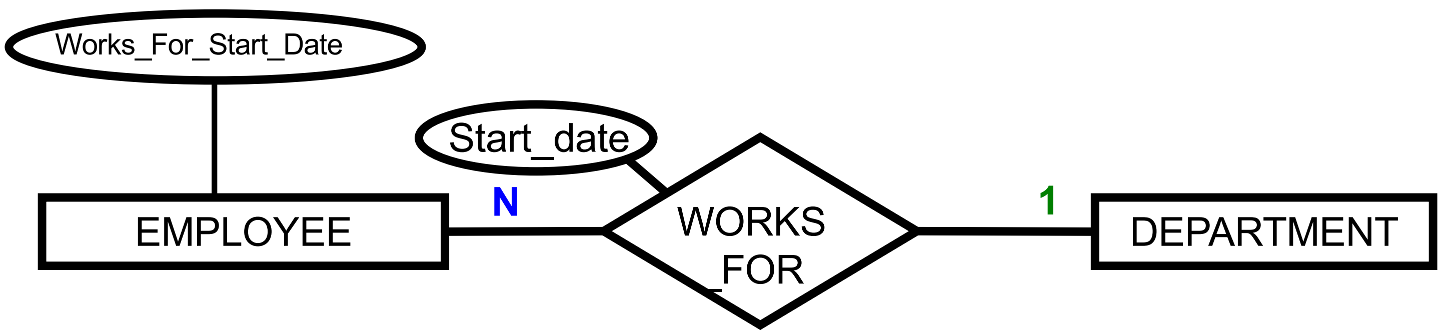




Relationship attributes (1:N)

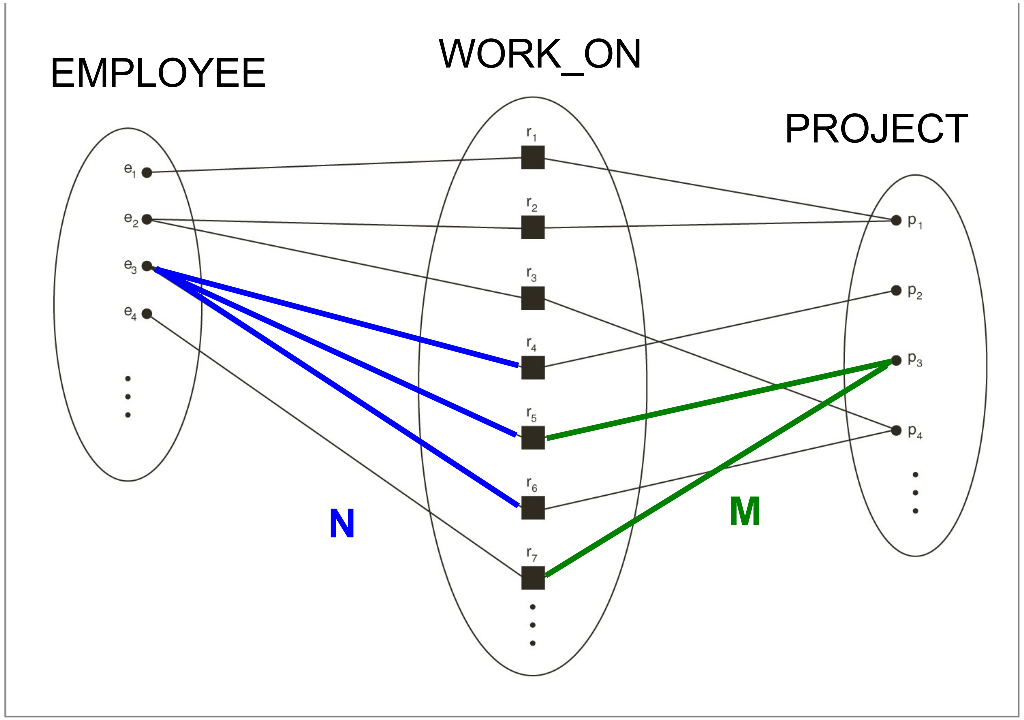
- In case of 1:N relationships the attribute can **only** be moved to the entity on the **N side**.
  - Start_Date for WORKS_FOR can be moved to EMPLOYEE (N side)
  - Each employee instance can WORKS_FOR max one DEPARTMENT.

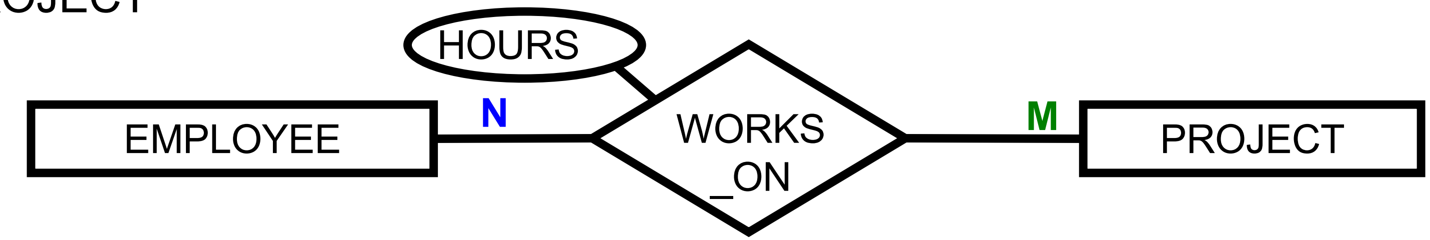




Relationship attributes (N:M)

- Attributes of M:N relationships types must be specified as such (but not to any of their participating entity types).
  - E.g.,: Attr. HOURS for relationship WORKS_ON can not be moved to either EMPLOYEE or PROJECT.
  - Number of hours that an EMPLOYEE works on a given PROJECT is determined through an EMPLOYEE/PROJECT combination
  - Note: Hours that an EMPLOYEE worked on a PROJECT ≠ hours that this EMPLOYEE worked in total ≠ total man-hours of the PROJECT



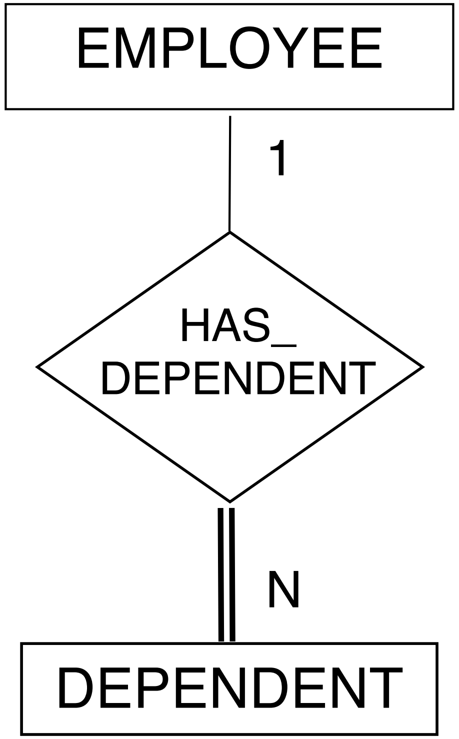


Contents

Conceptual DB design (ER-Model cont.)

a. Relationships and their constraints
b. Relationship attributes
c. Weak entity types and partial keys
d. ER Diagram notation
e. Refinement of the design of an ER-Diagram for DB "Company"

An employee leaves the company



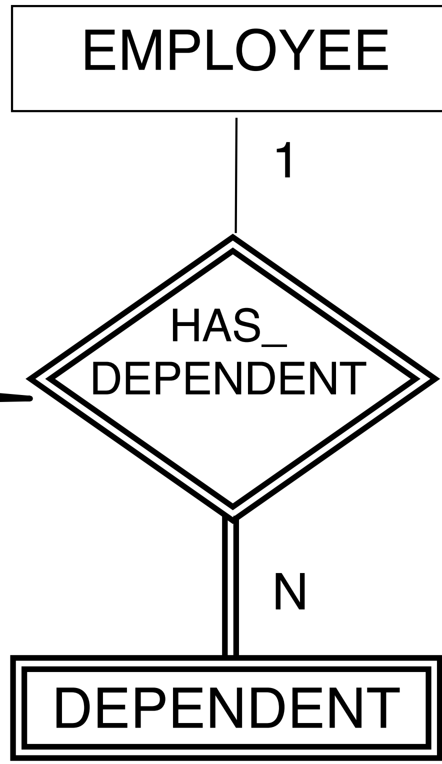
http://www.omniagroup.com/wp-content/uploads/2011/11/employee-exit.jpg

<u>Example in a requirement analysis report</u>

If an employee leaves the company, all his/her dependents should be deleted from the DB.

Weak entity types

- **Weak entity types** normally do not have any **own** key attributes (The existence of a weak entity instance depends on another entity)
- Entities with own key attributes = **strong entity types (their existence doesn't depend on others)**.
- For identifying a weak entity, we **must** refer to the key of another entity → **identifying** or **owner entity type (**they have an **identifying relationship)**
- Weak entities **depend** on the existence of the owner entity (**existence dependency**)
  → total participation constraint is a prerequisite
  (i.e., each dependent must be linked to one employee)

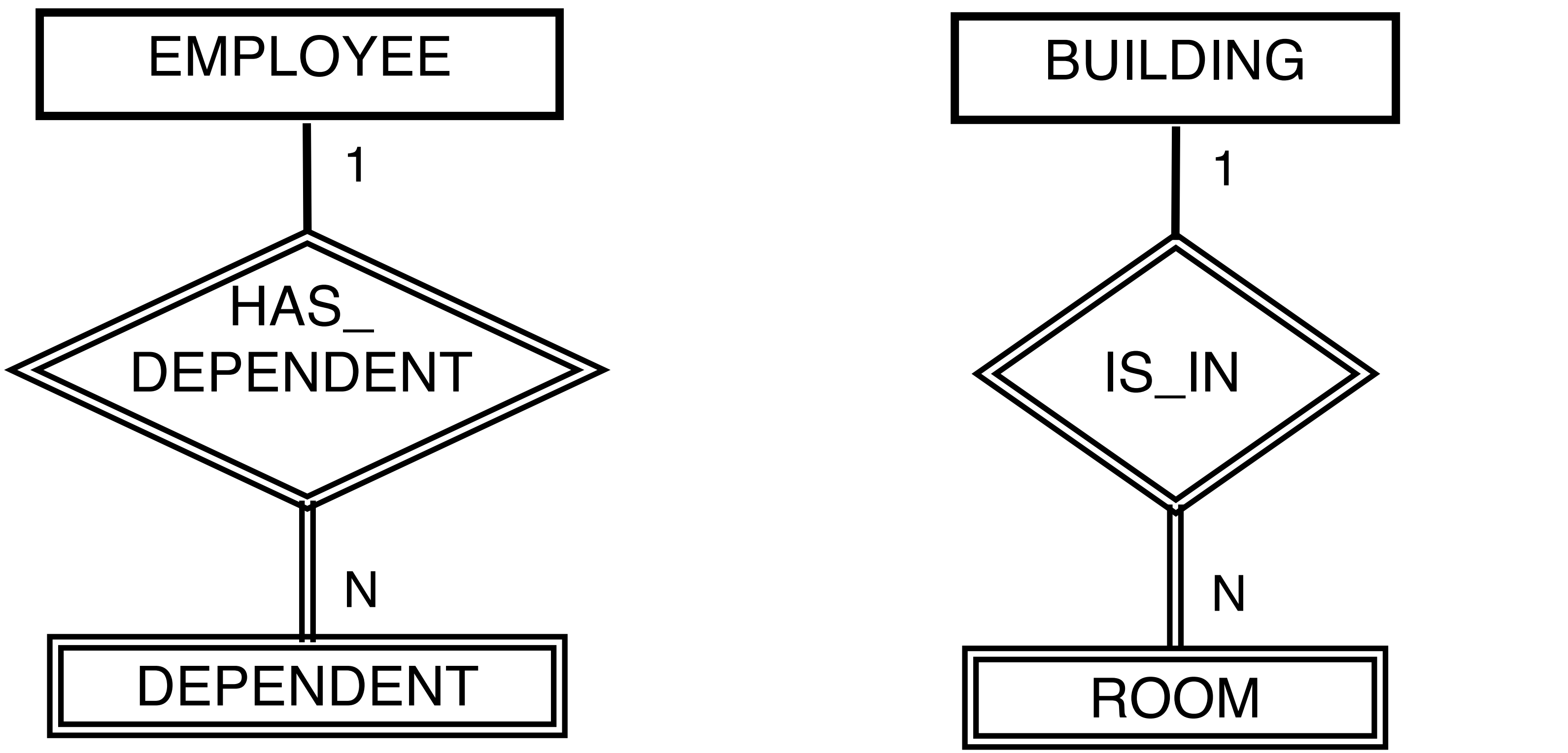


Identifying relationship notation

Weak entity type notation

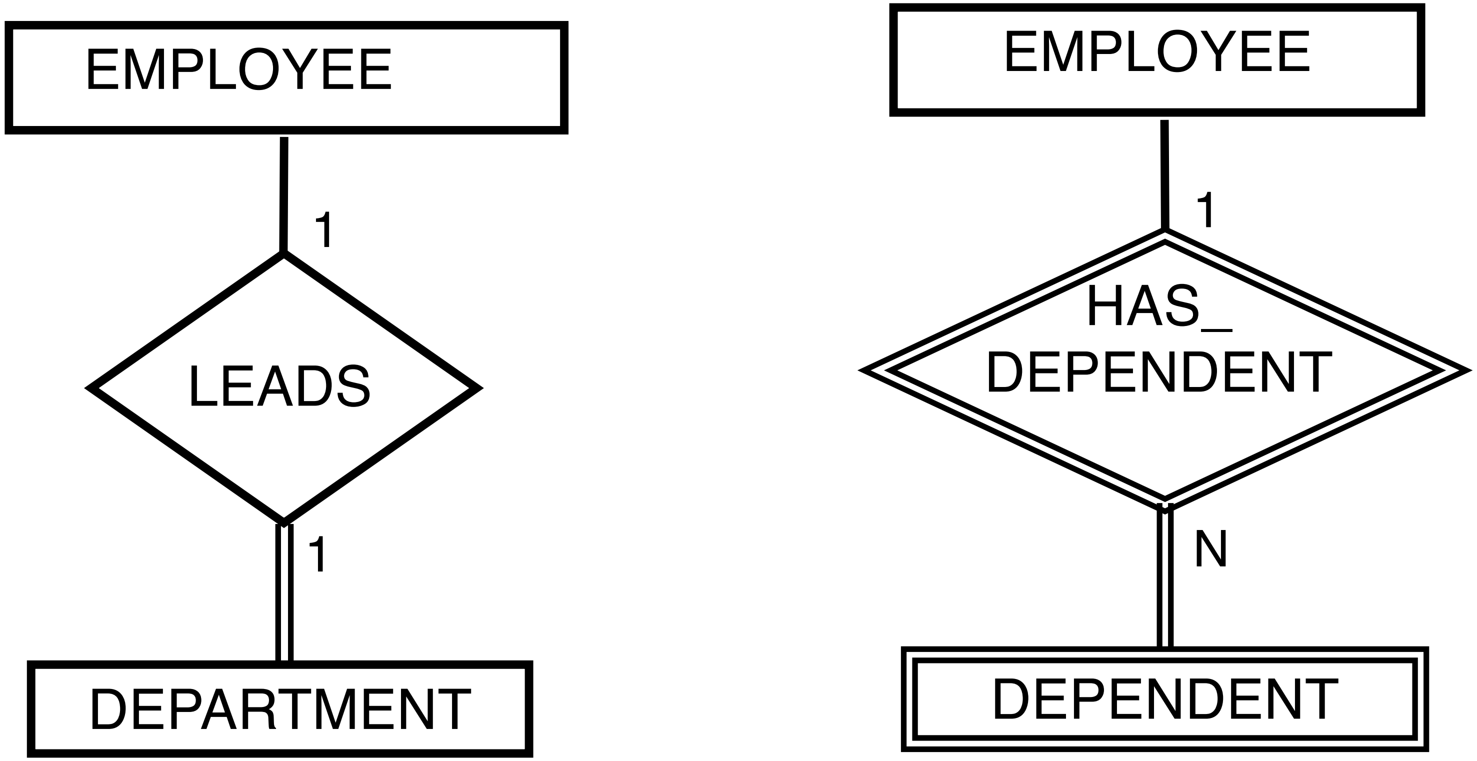
Weak entity types

Examples of existence dependency of entity types:



Total participation vs. weak entity type

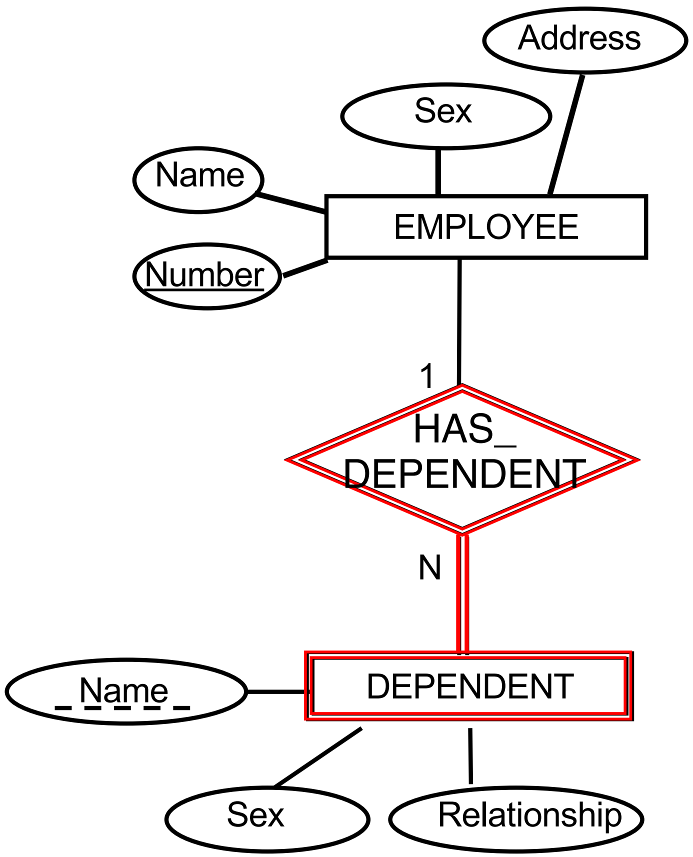
Quiz: Can you tell the difference between total participation and a weak entity type?



Aw: The main difference is on the **existence dependency**

- The leaving of a department manager does not mean the closure of the department
- Weak entity type: The leaving of an employee requires the deletion of his/her dependents (e.g., children) from the DB.

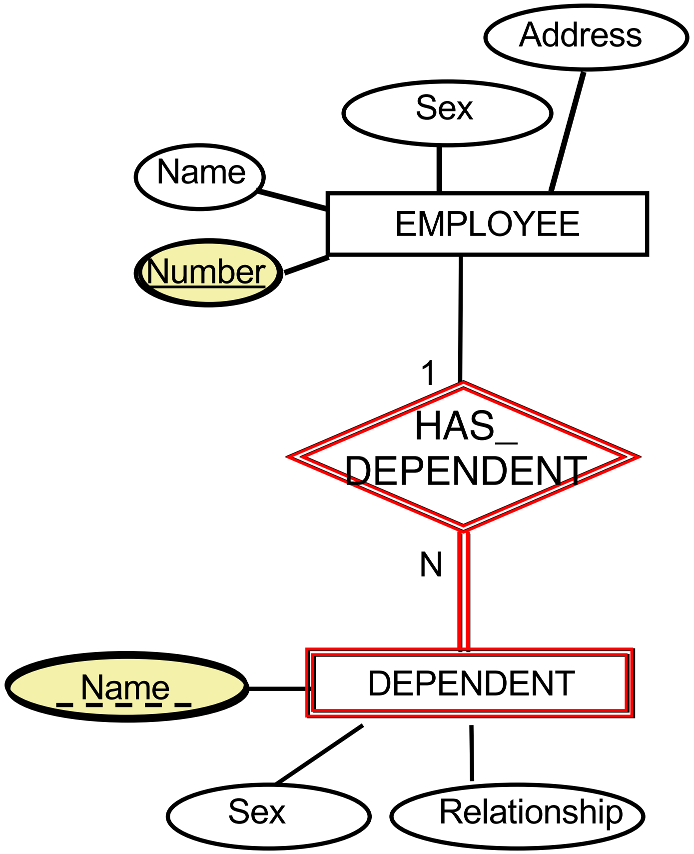
Weak entity types



Example – Entity type DEPENDENT with relationship to EMPLOYEE

- Each DEPENDENT **depends** on an EMPLOYEE with whom he/she has a relationship.
- DEPENDENT can only exist when the employee works for the company.
- Weak entity types and their identifying relationship types are represented through a double line

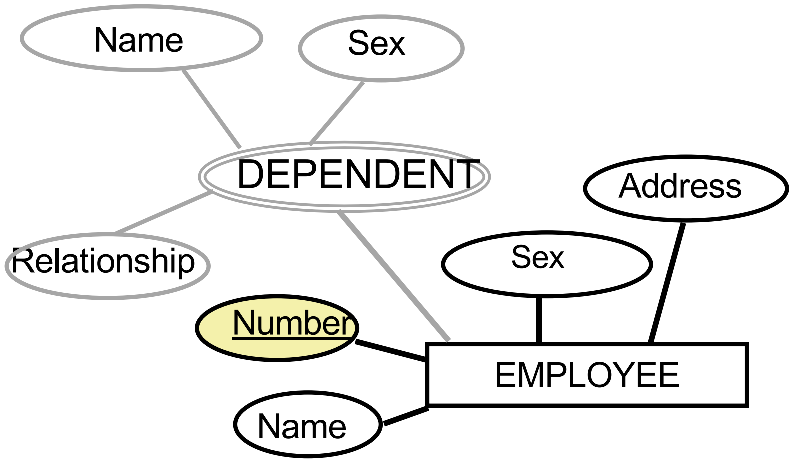
Weak entity types



Example – Entity type DEPENDENT with relationship to EMPLOYEE

- A weak entity type has a **partial key**:
  - the attribute that can uniquely identify weak entities that are related to *the same owner entity*.
    - Example: Name can be used to uniquely identify employee Martin‘s dependents.
- Partial keys are highlighted by underscoring with a dashed line
- Weak entities can **own** other weak entities!

Weak entity types

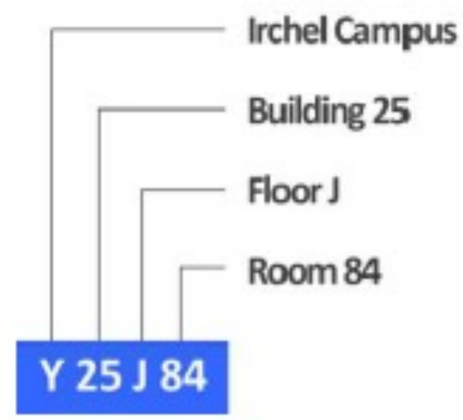


Example – Entity type EMPLOYEE with relationship to DEPENDENT

- Weak entity types are sometimes intuitively modelled as complex (multi-valued, composite) attributes

Exercise: Multiple weak entity types

Room numbering system at campus Irchel: An example of a "room number" is Y25J84. Which would be the entity types in an ER-model? Which partial keys (as a consequence of weak entity types) do you see?

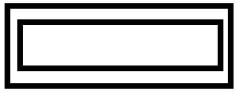


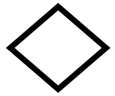
Contents

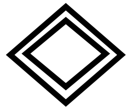
Conceptual DB design (ER-Model cont.)

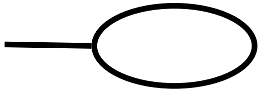
ER Diagram notations

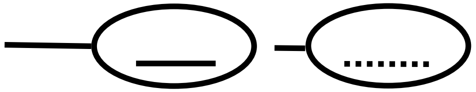
Entity type

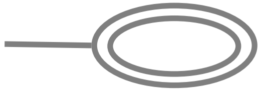

Weak Entity type


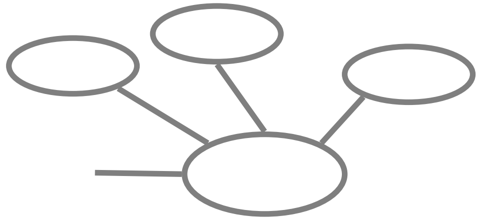
Relationship type


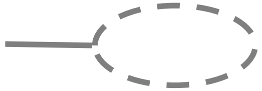
Identifying relationship


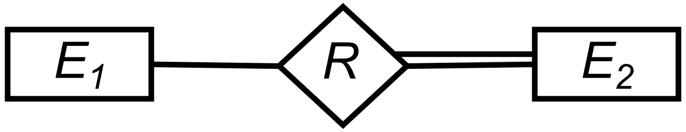
Attribute


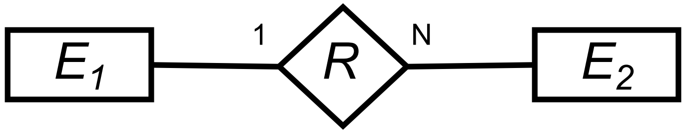
Key attribute (partial)


Multi-valued Attribute


Composite attribute


Derived attribute


Total participation of E_2 in R


Cardinality ratio 1:N for E_1:E_2 in R


Contents

Conceptual DB design (ER-Model cont.)

Exercise: ER design of the DB „Company“ (cont.)

Draw the ER-Diagram for the DB „Company“ with 4 *Entity types* and *Attributes* (still without relationships!) on your sheets

- With pencil...

DEPARTMENT

Name, Number, {Locations}, Manager, ManagerBeginningDate

PROJECT

Name, Number, Location, ControllingDepartment

EMPLOYEE

Name (FName, MInit, LName), SSNo, Sex, Address, Salary, Birthdate, Department, Supervisor, {WorksOn(Project, Hours)}

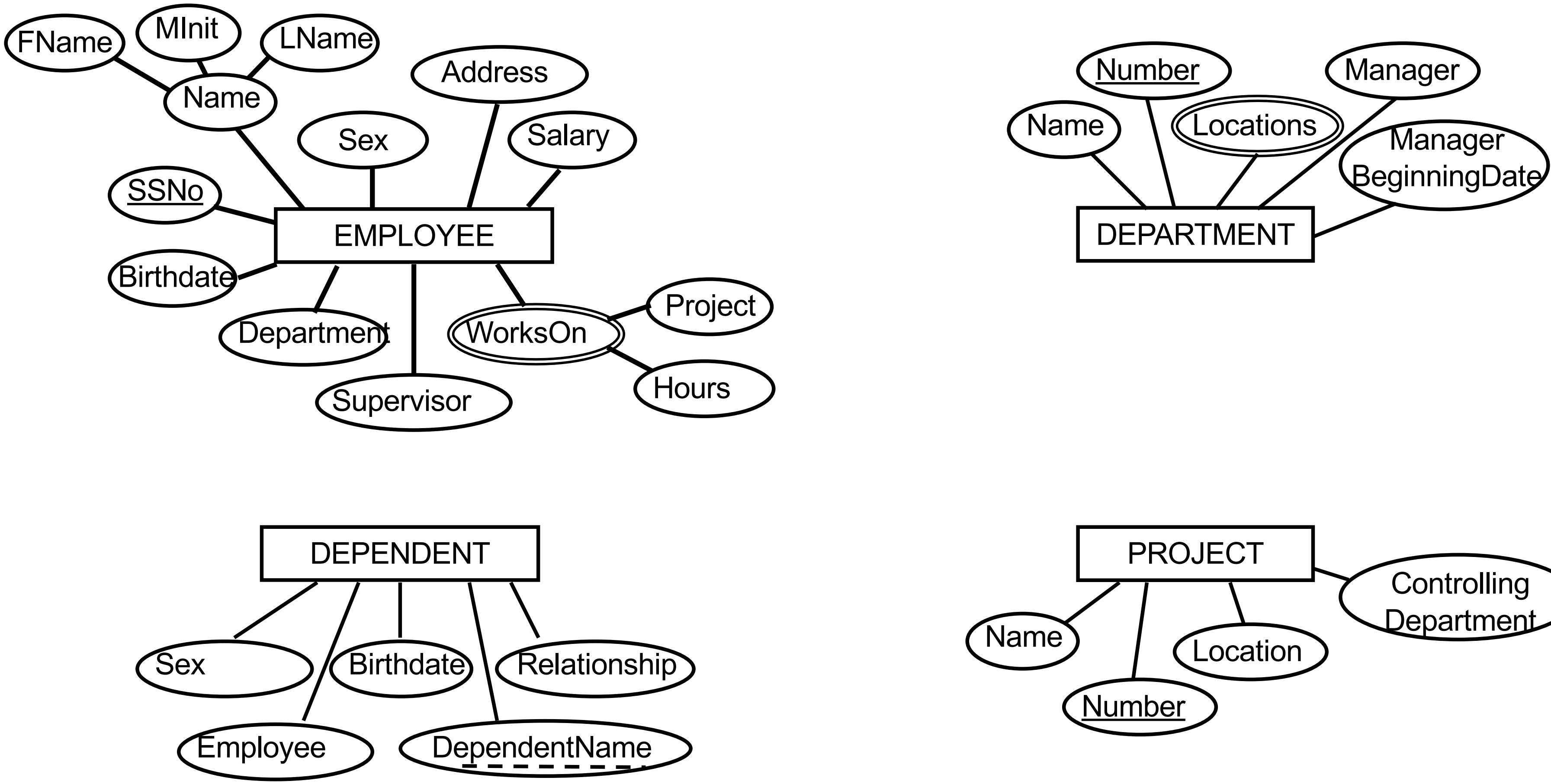
DEPENDENT

Employee, DependentName, Sex, Birthdate, Relationship

{} – multivalued attribute

() – composite attribute

Refinement of the DB „Company“ ER design



Exercise: ER design of the Company DB (relationships)

We will analyse and design the relationships between the entities...

- With pencil...

Refinement of the DB „Company“ ER design

1. The company is organized into departments. Each department has a unique name, a unique number and an employee who manages the department. We keep track of the start date of the department manager. A department is located on multiple locations.
2. Each department controls a number of projects. Each project has a unique name, a unique number and is located at a single location!
3. We store each employee’s social security number (SSNo), name, address, salary, sex, and birthdate.
   Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.
4. Each employee may have a number of dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Refinement of the DB „Company“ ER design

1. The company is organized into departments. Each department has a unique name, a unique number and an employee who manages the department. We keep track of the start date of the department manager. A department is located on multiple locations.

2. Each department controls a number of projects. Each project has a unique name, a unique number and is located at a single location!

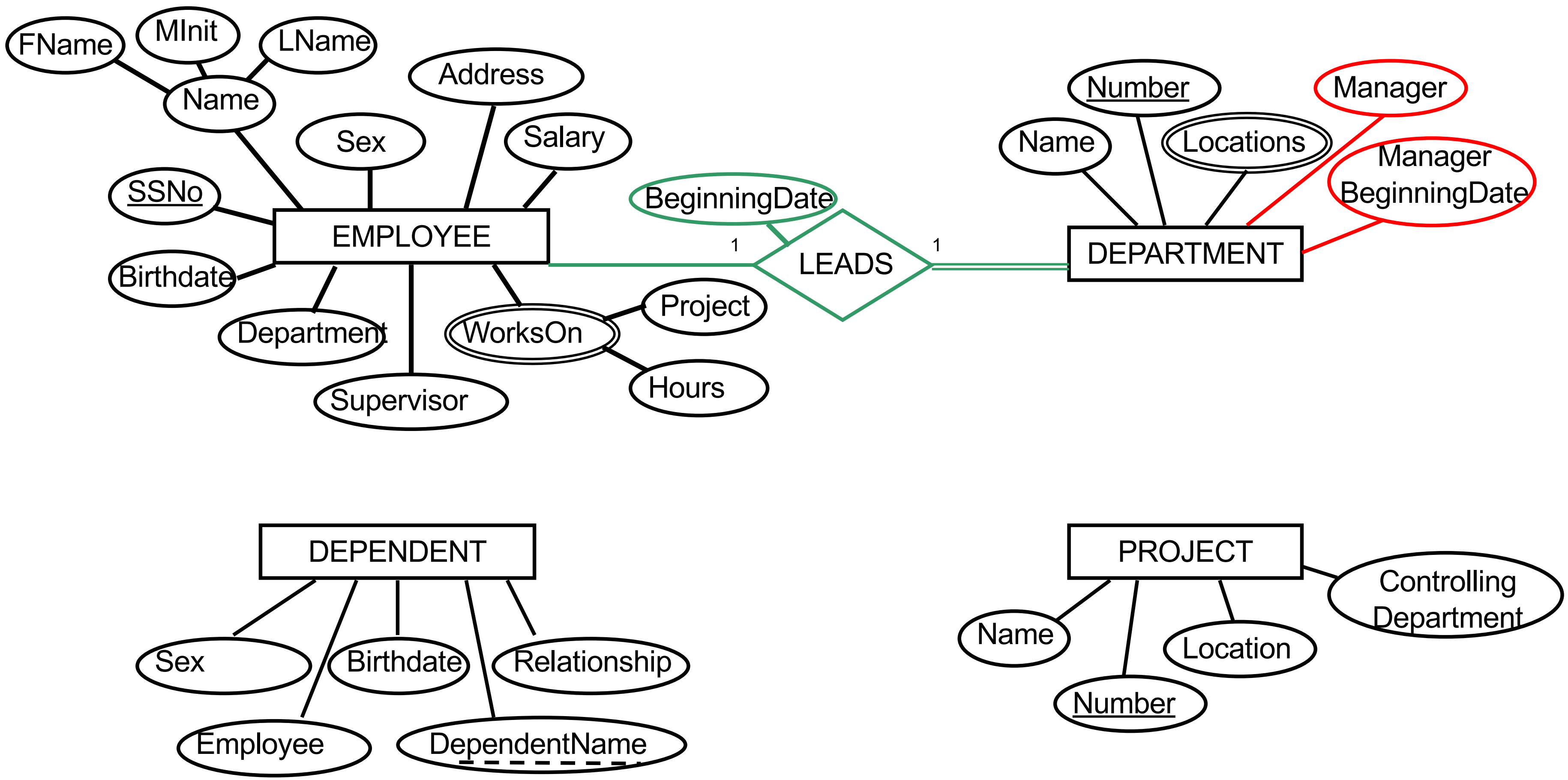
3. We store each employee’s social security number (SSNo), name, address, salary, sex, and birthdate.
   Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.

4. Each employee may have a number of dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Refinement of the DB „Company“ ER design

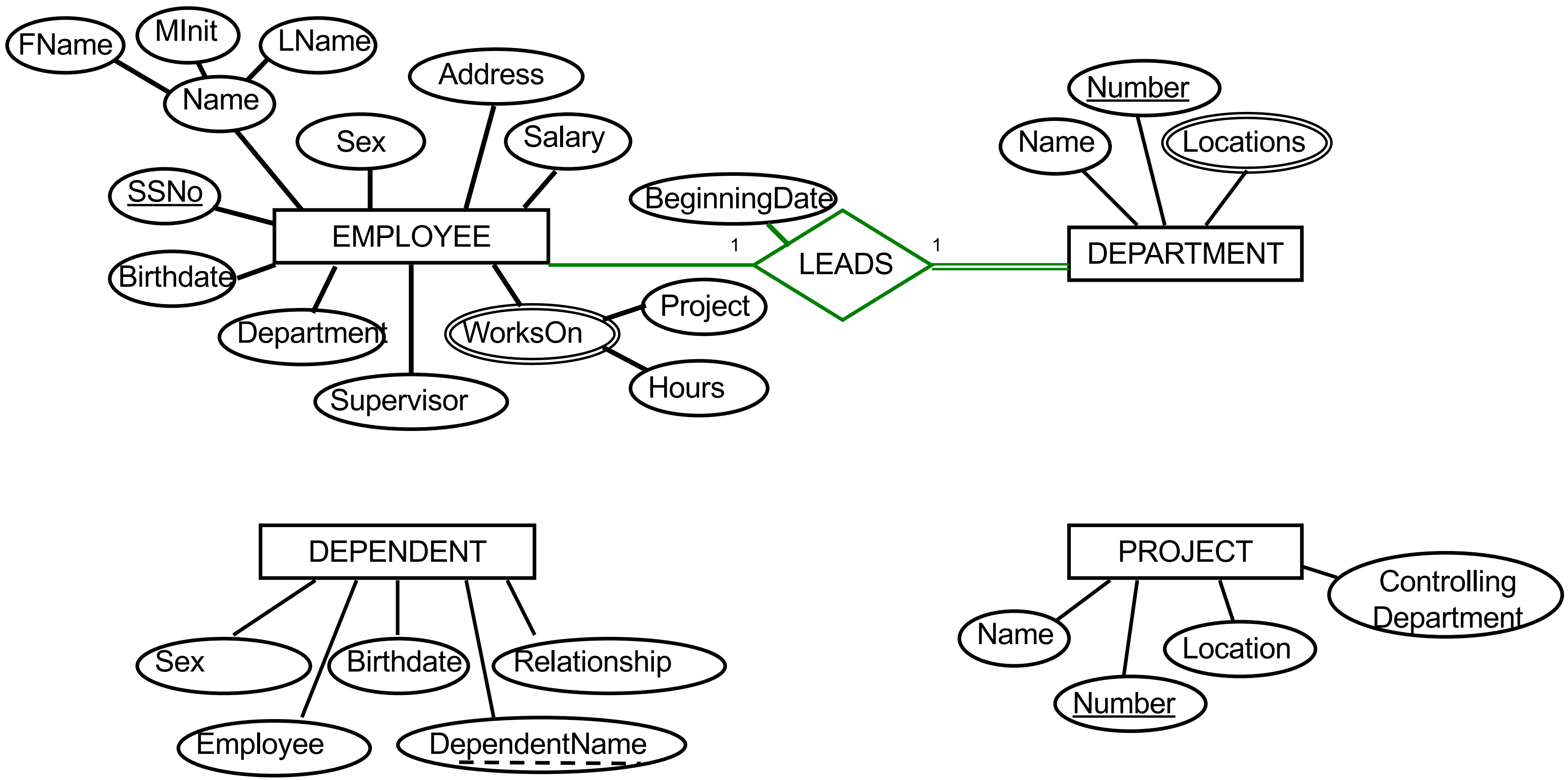
employee [...] manages the department 1

Each department must be led by one employee → DEPARTMENT participation is total
Each employee can lead one department → EMPOLYEE participation is partial



Refinement of the DB „Company“ ER design

employee [...] manages the department **1**



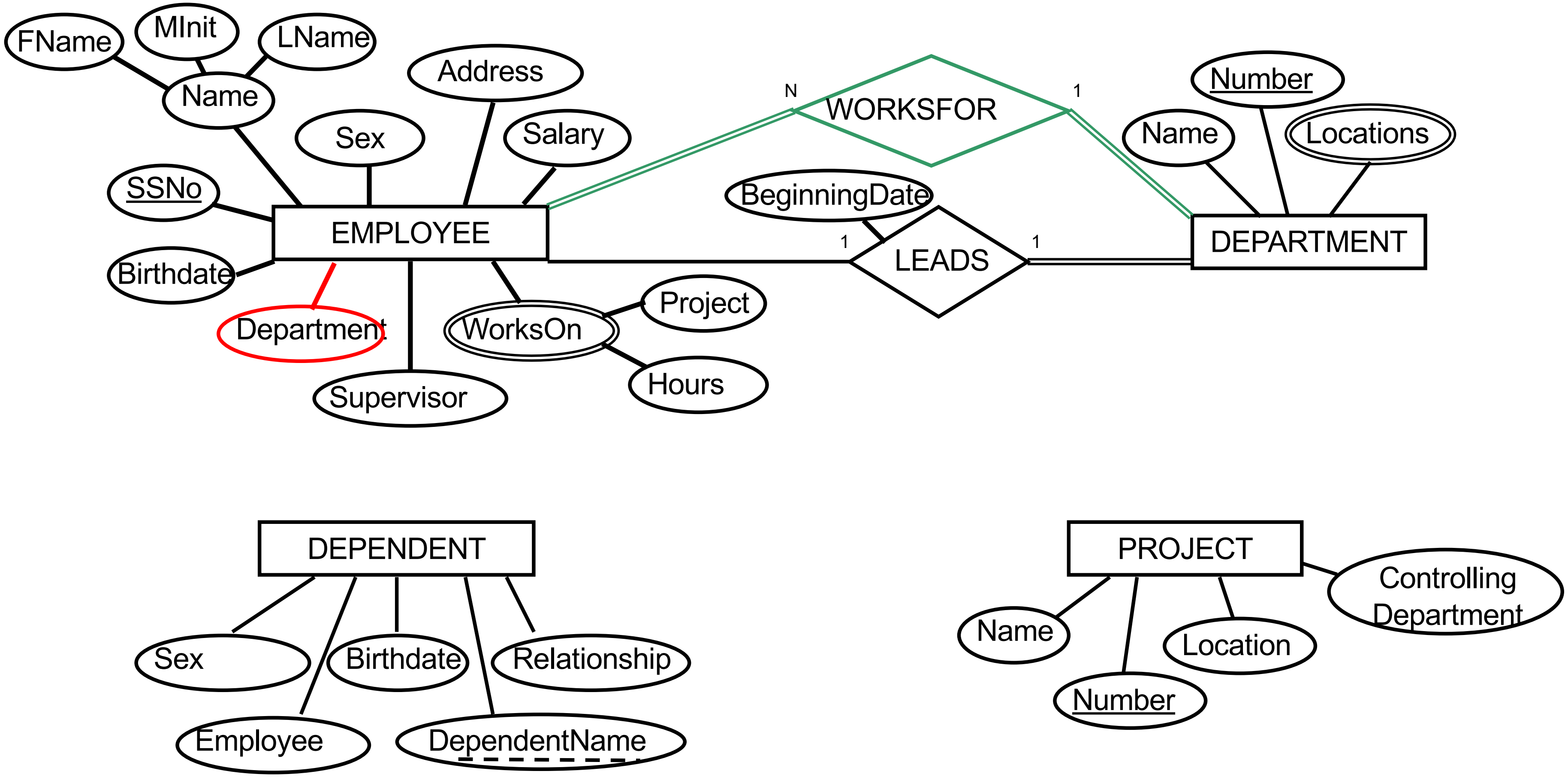
Refinement of the DB „Company“ ER design

1. The company is organized into departments. Each department has a unique name, a unique number and an employee who manages the department. We keep track of the start date of the department manager. A department is located on multiple locations.
2. Each department controls a number of projects. Each project has a unique name, a unique number and is located at a single location!
3. We store each employee's social security number (SSNo), name, address, salary, sex, and birthdate.
   Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.
4. Each employee may have a number of dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Refinement of the DB „Company“ ER design

employee works for one department 2

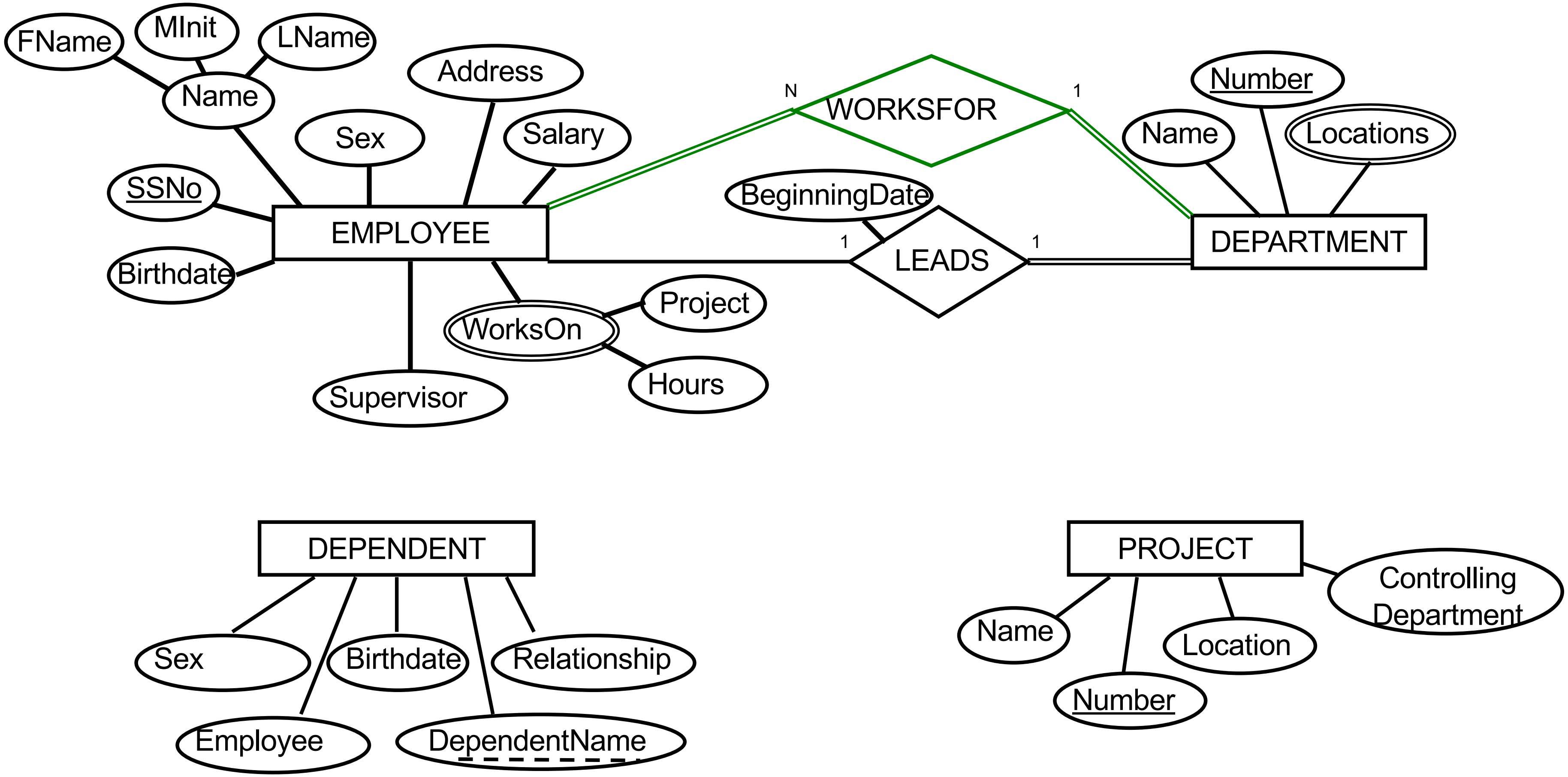
Each employee must work for one department → EMPOLYEE participation is total
Each department must have one or multiple employees → DEPARTMENT participation is total



Refinement of the DB „Company" ER design

employee works for one department

2



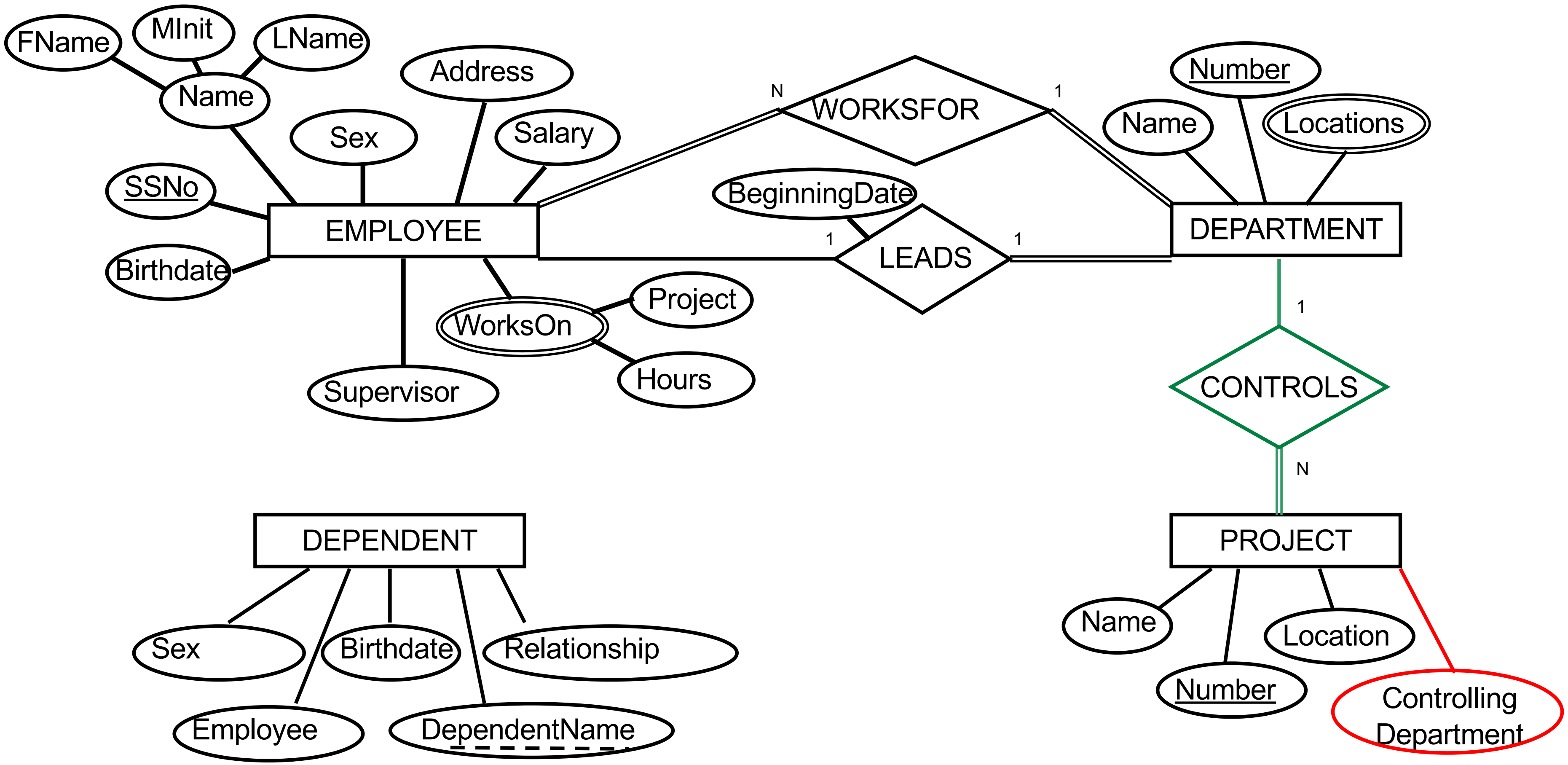
Refinement of the DB „Company“ ER design

1. The company is organized into departments. Each department has a unique name, a unique number and an employee who manages the department. We keep track of the start date of the department manager. A department is located on multiple locations.
2. Each department controls a number of projects. Each project has a unique name, a unique number and is located at a single location!
3. We store each employee’s social security number (SSNo), name, address, salary, sex, and birthdate.
   Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.
4. Each employee may have a number of dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Refinement of the DB „Company“ ER design

Each department controls a number of projects **3**

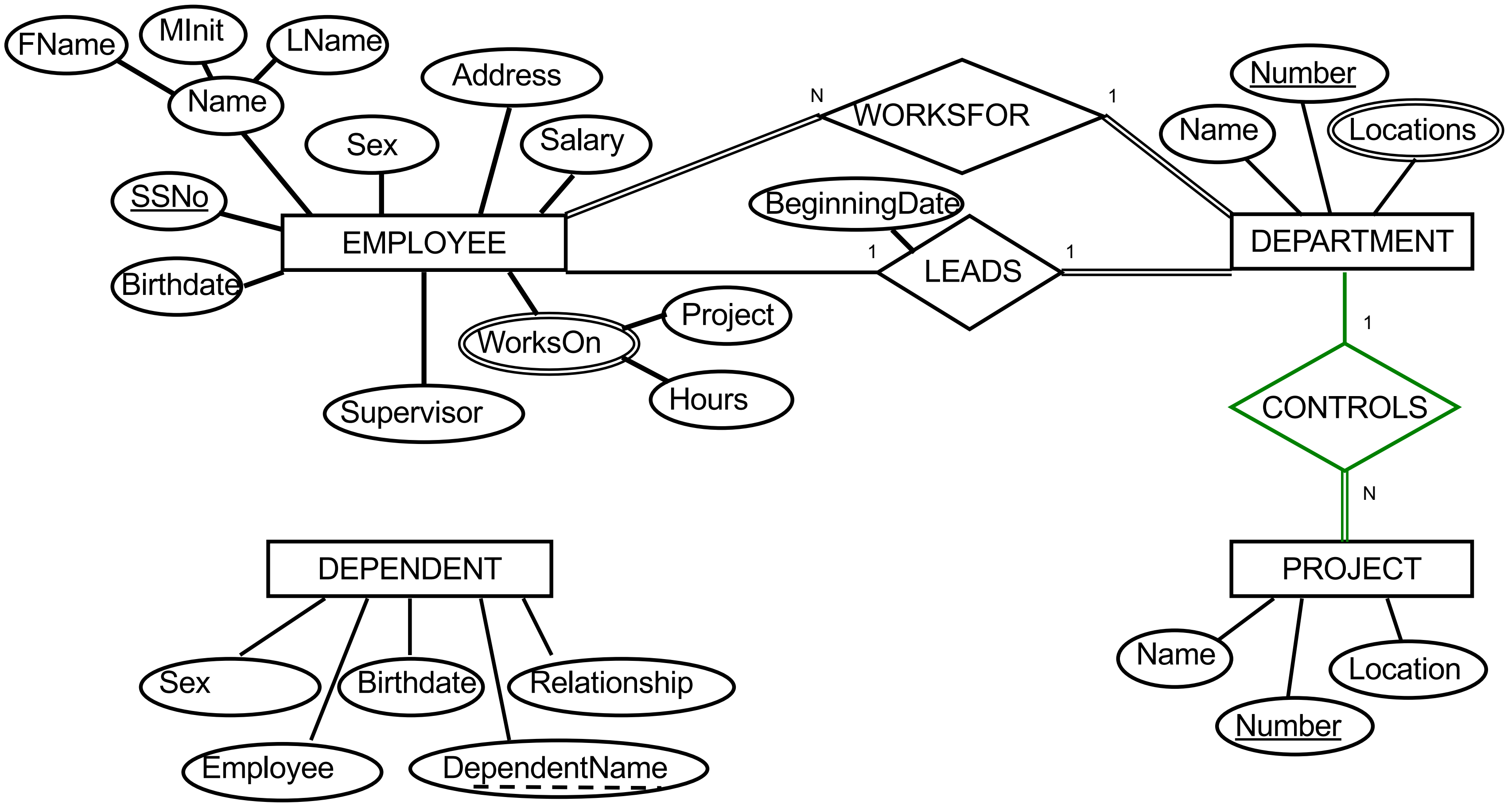
Each project must be controlled by one department → PROJECT participation is total
Each department can control one or multiple projects → DEPARTMENT participation is partial



Refinement of the DB „Company“ ER design

Each department controls a number of projects

3



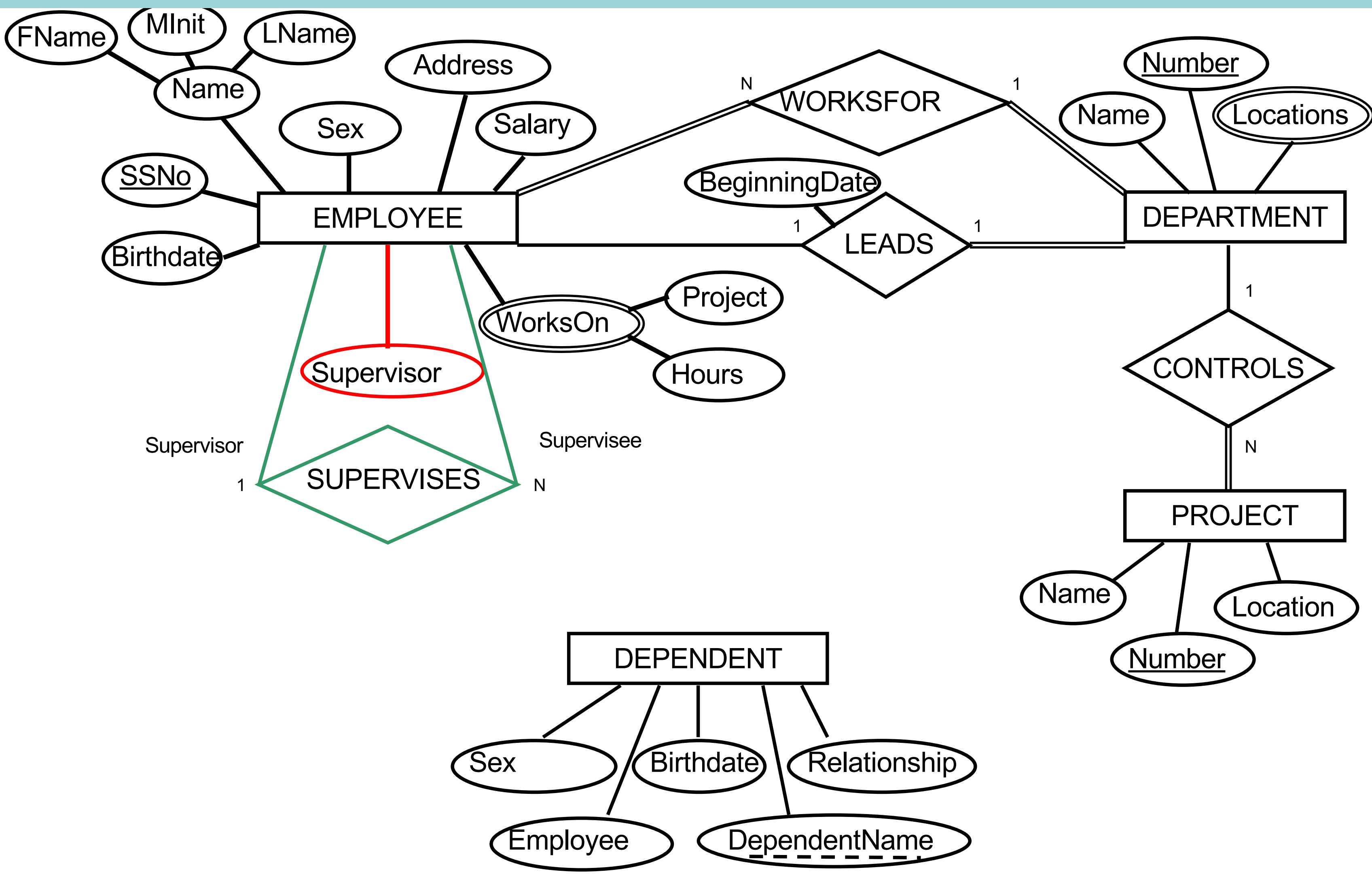
Refinement of the DB „Company" ER design

1. The company is organized into departments. Each department has a unique name, a unique number and an employee who manages the department. We keep track of the start date of the department manager. A department is located on multiple locations.
2. Each department controls a number of projects. Each project has a unique name, a unique number and is located at a single location!
3. We store each employee's social security number (SSNo), name, address, salary, sex, and birthdate.
   Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.
4. Each employee may have a number of dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Refinement of the DB „Company“ ER design

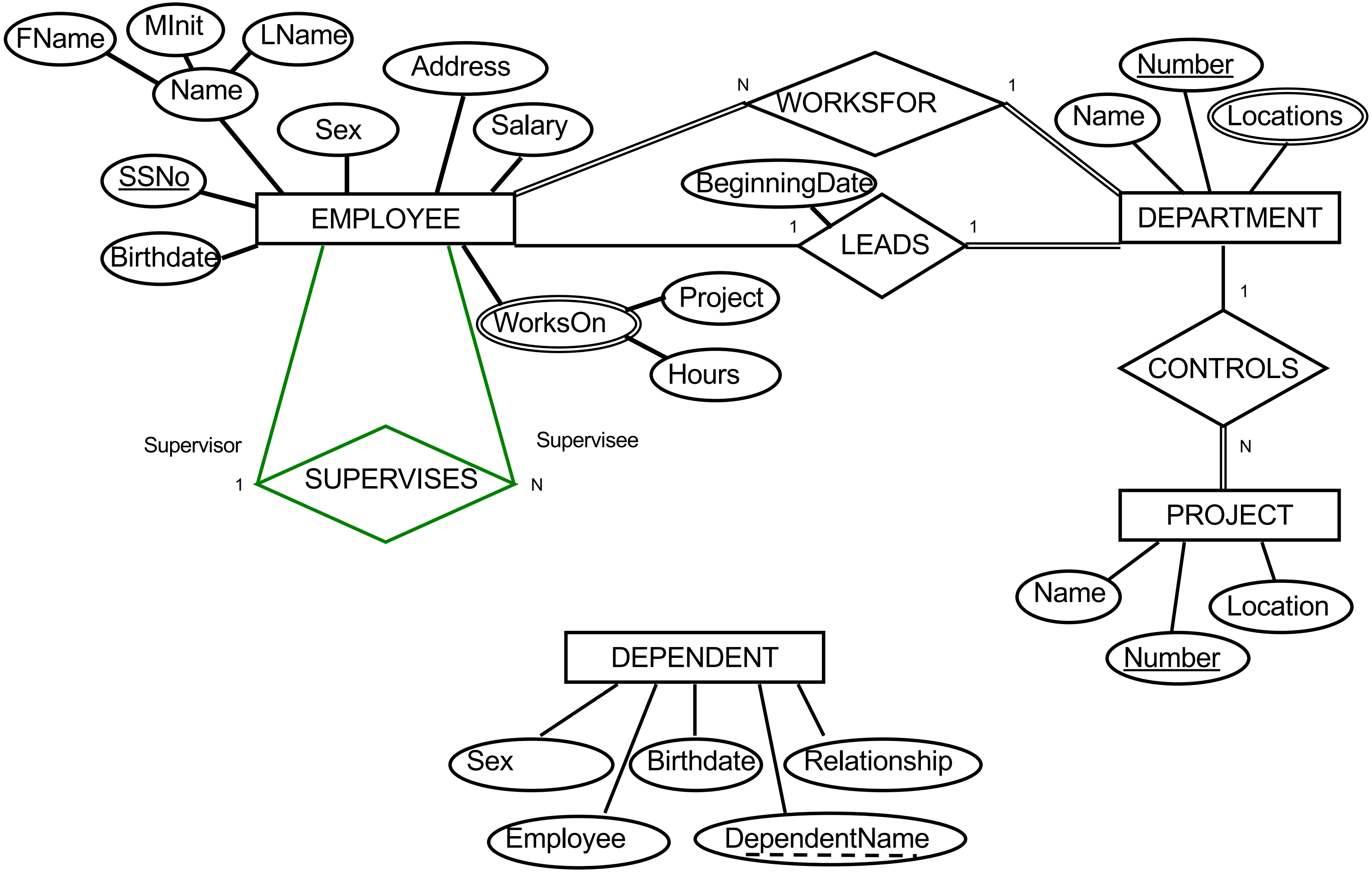
supervisor of each employee 4

Not every employee is a supervisor, and not every employee has a supervisor.
→The participations of EMPLOYEE (supervisor role) and EMPLOYEE (supervisee role) are partial



Refinement of the DB „Company“ ER design

supervisor of each employee 4



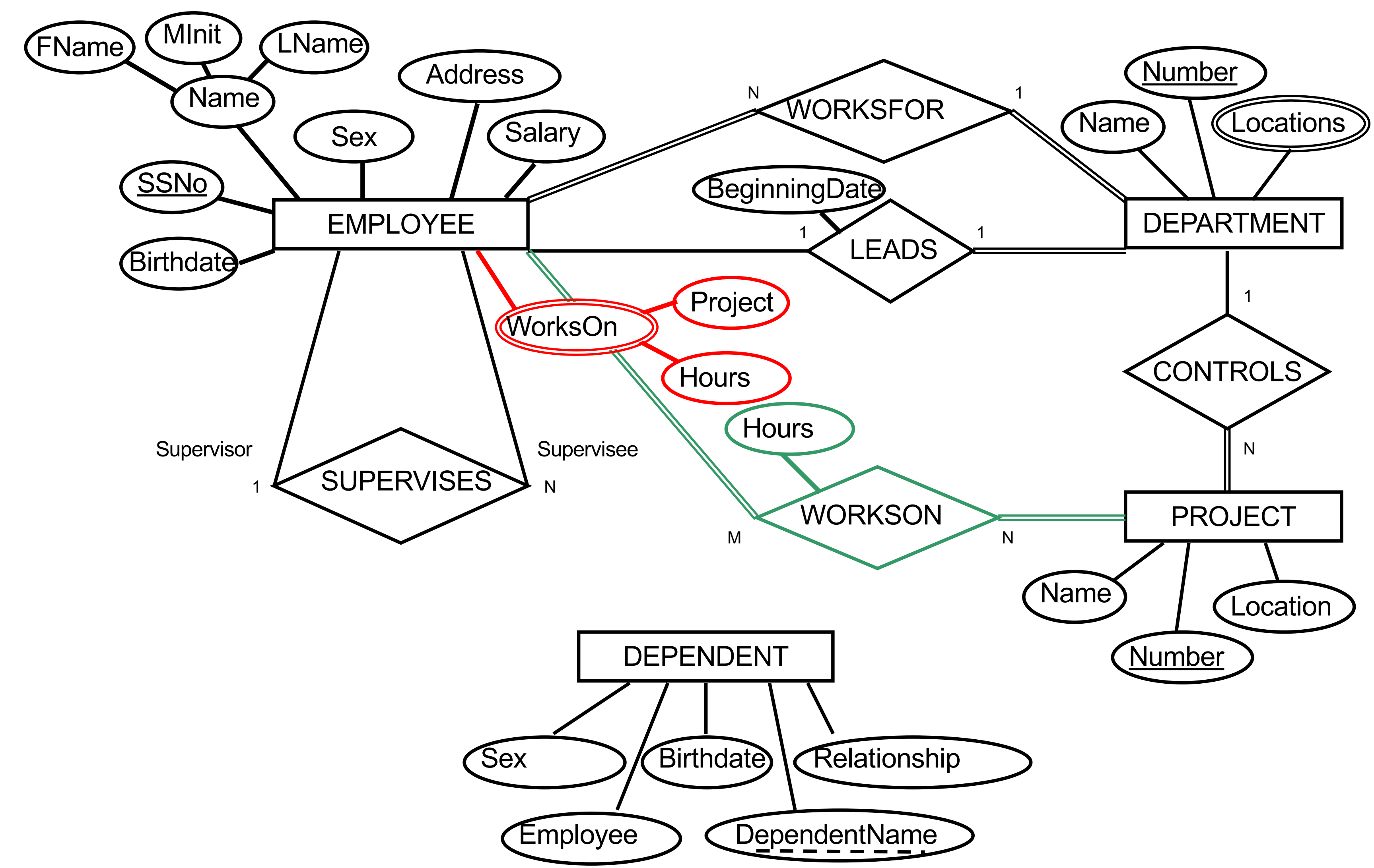
Refinement of the DB Company ER design

1. The company is organized into departments. Each department has a unique name, a unique number and an employee who manages the department. We keep track of the start date of the department manager. A department is located on multiple locations.
2. Each department controls a number of projects. Each project has a unique name, a unique number and is located at a single location!
3. We store each employee's social security number (SSNo), name, address, salary, sex, and birthdate.
   Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.
4. Each employee may have a number of dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Refinement of the DB Company ER design

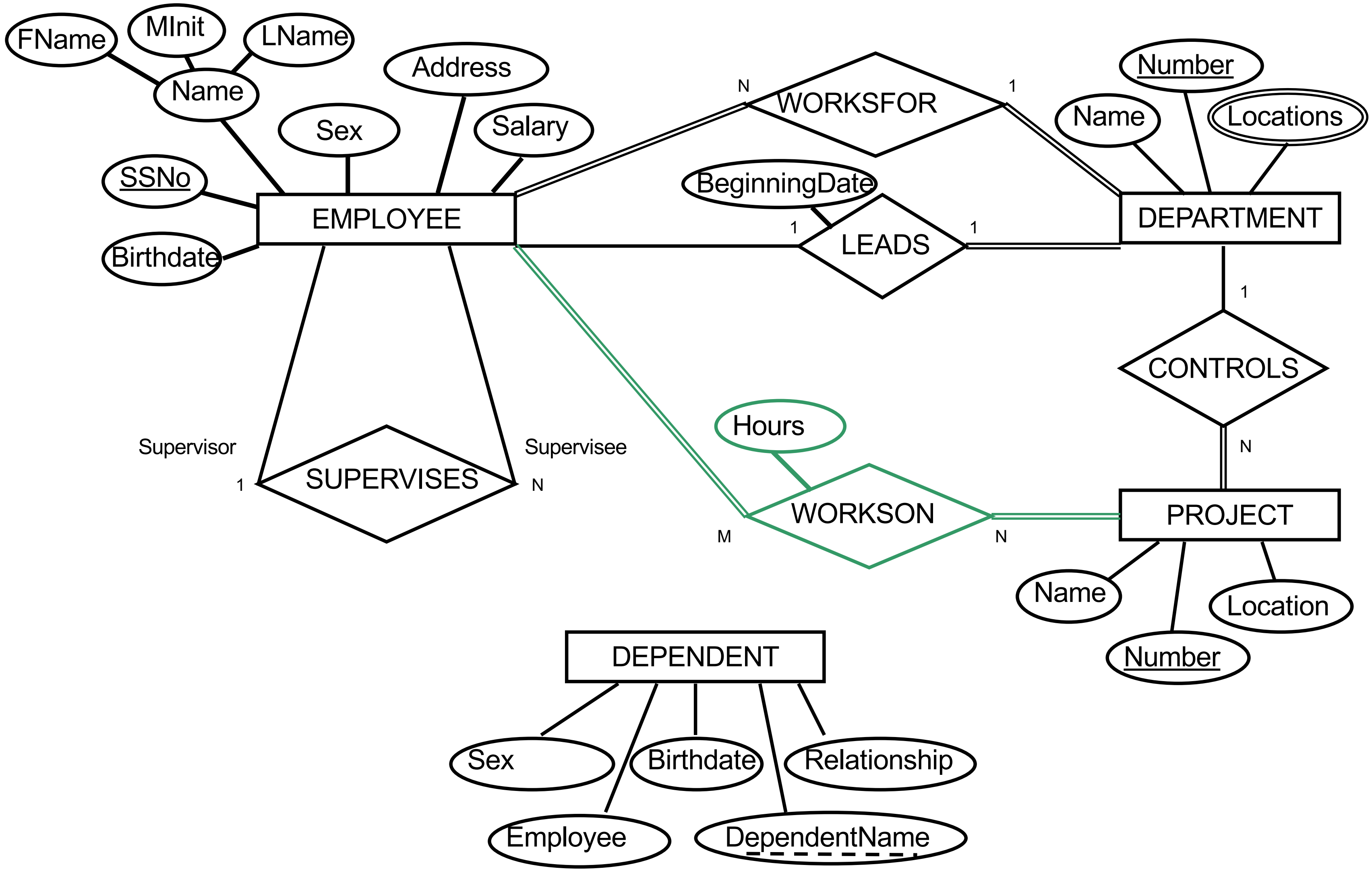
Each employee [...] may work on several projects 5

Both PROJECT participation and EMPLOYEE participation are total



Refinement of the DB Company ER design

Each employee [...] may work on several projects 5



Refinement of the DB Company ER design

1. The company is organized into departments. Each department has a unique name, a unique number and an employee who manages the department. We keep track of the start date of the department manager. A department is located on multiple locations.

2. Each department controls a number of projects. Each project has a unique name, a unique number and is located at a single location!

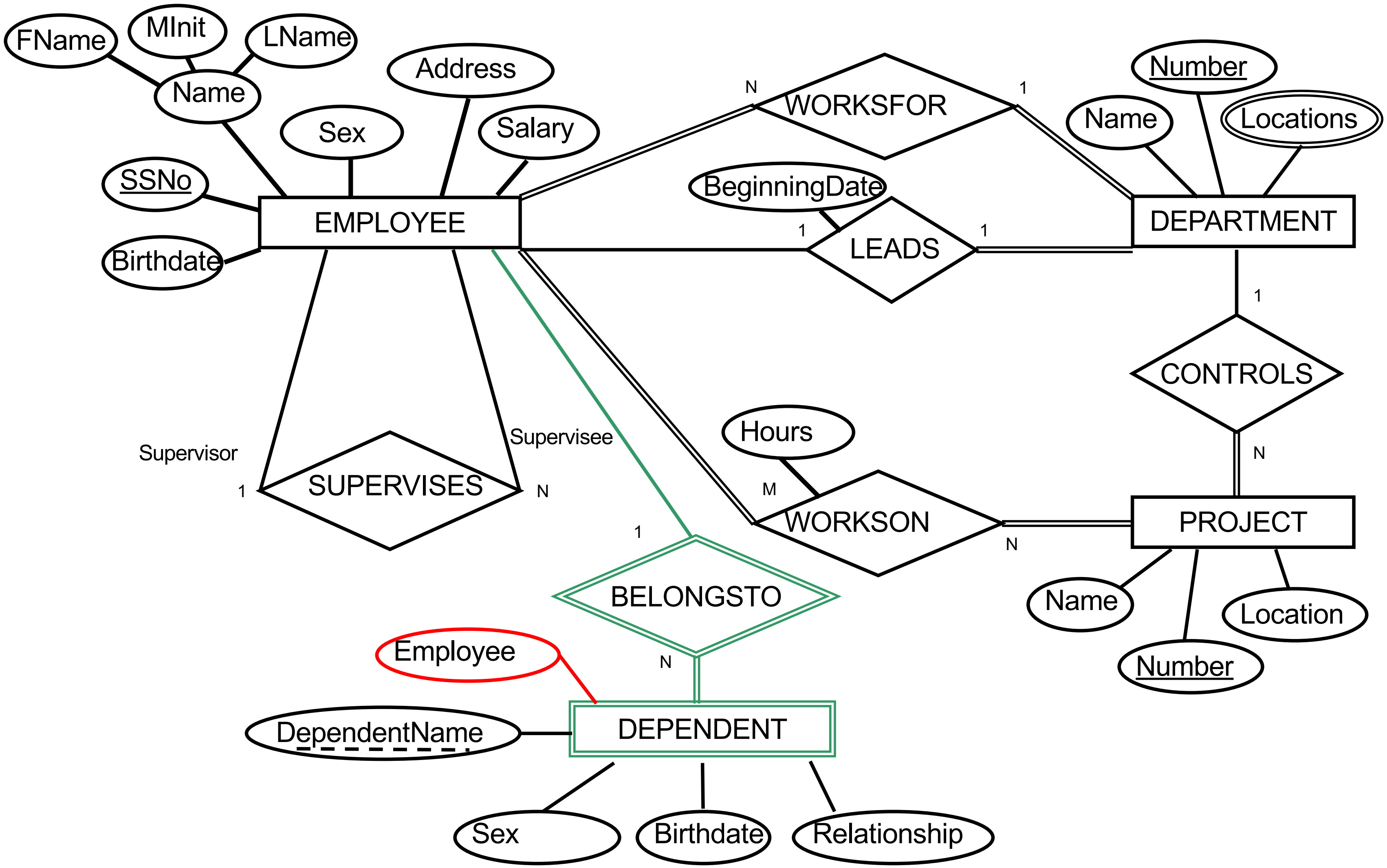
3. We store each employee’s social security number (SSNo), name, address, salary, sex, and birthdate.
   Each employee works for one department but may work on several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct supervisor of each employee.

4. Each employee may have a number of dependents. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.

Refinement of the DB Company ER design

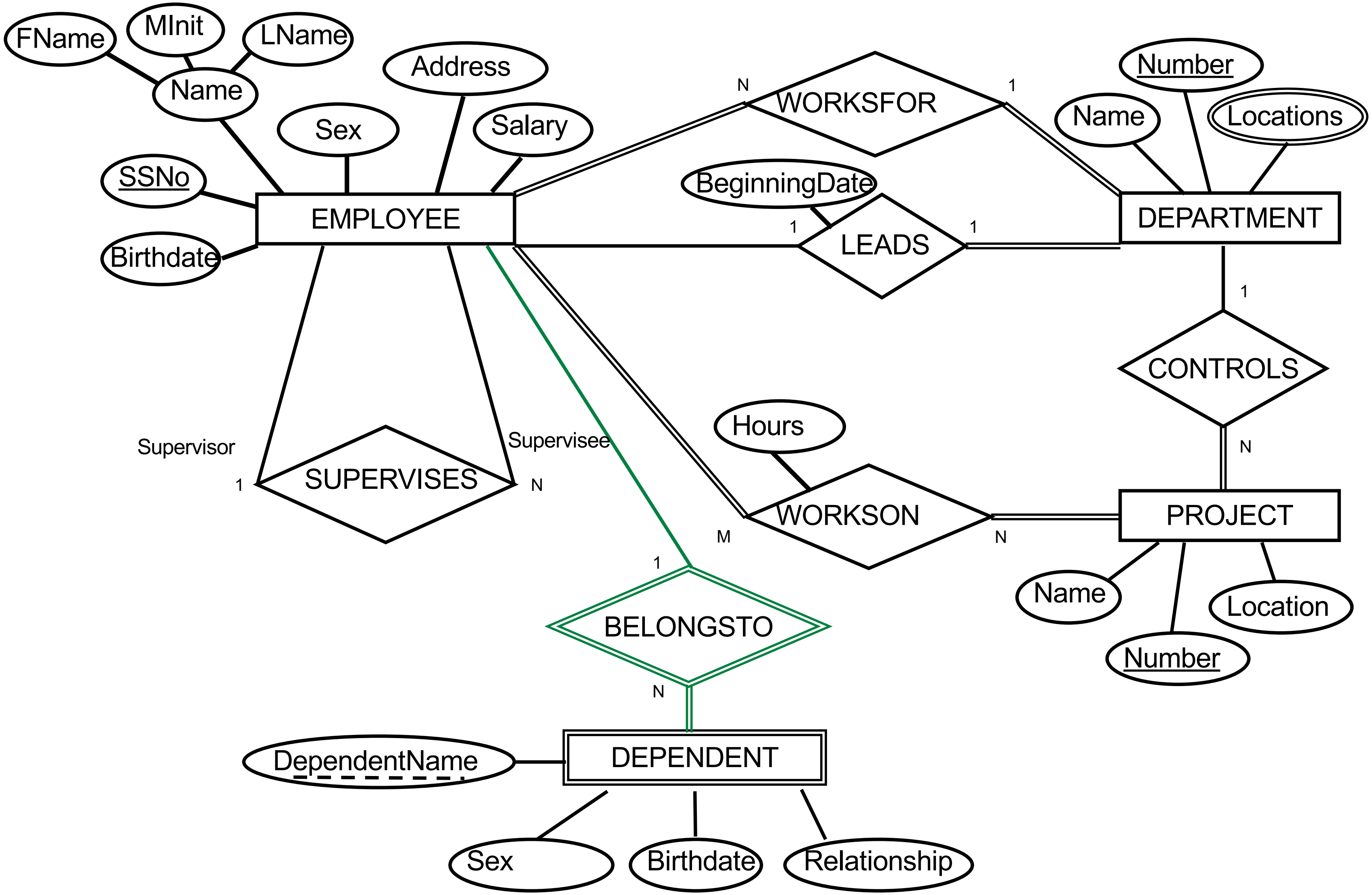
employee may have a number of dependents 6

EMPLOYEE participation is partial, and DEPENDENT participation is total.
DEPENDENT is a weak entity type: the existence of a DEPENDENT depends on EMPOLYEE.



Refinement of the DB Company ER design

employee may have a number of dependents **6**



Guidance for ER modeling

1. In general, the ER modeling is an iterative process (refine, refine, refine)
2. The decision whether something is an entity, attribute, or relationship type is a matter of choice and can change during the process
3. Concepts can first be modeled as an attribute and then be refined as relationship if the attribute is later referred to another entity type
4. An attribute that is referred by multiple entity types can be turned into an entity type of its own
   – Example: Department
5. ..and conversely, if an entity type ends up with a single attribute and a single relationship, it can probably be modeled as an attribute

Summary: ER Modeling

Conceptual ER models are represented through ER Diagrams.
ER modeling concerns the identification of

- Entities, entity types and entity sets
  - Key attributes of the entity types
- Attributes are characterized by name, data type and value domain
- Attribute types
  - Simple, composite, multi-valued, complex
- Relationship types
  - Degree (recursive, **binary**, ternary, …)
  - Constraints on relationship types
    - Cardinality constraint (1:1, 1:N, M:N for binary relationships)
    - Participation constraint (total, partial)
- Weak entity types
  - An identifying entity must be referenced to identify a weak entity (the existence dependency)
  - The primary key of an identifying entity together with a partial key together uniquely identify a weak entity.

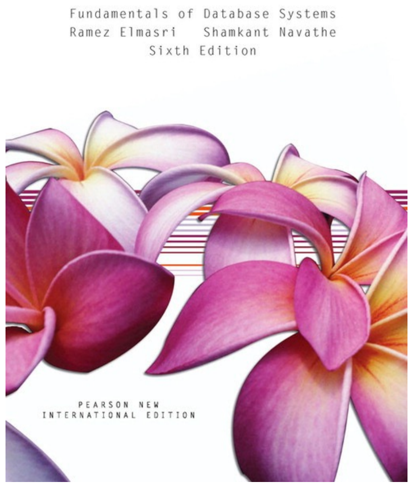
Resources

- Ramez A. Elmasri, Shamkant B. Navathe (2014). Fundamentals of Database Systems, 6th edition, ISBN: 978-1-292-02560-5, Pearson, 1081 pages.

  Ch 7: Data Modelling Using the Entity-Relationship (ER) Model

  Ch 8: The EER Model (optional)

  Ch 9: ER-to-Relational Mapping

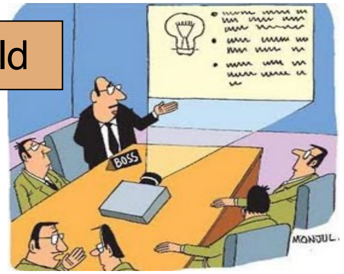


Later, in lab

- 10:15-12:00
- Y25-J-09, Y25-J-10
- **ER Modelling (with pencil and paper)**

Next week: Logical and physical DB design

Real World

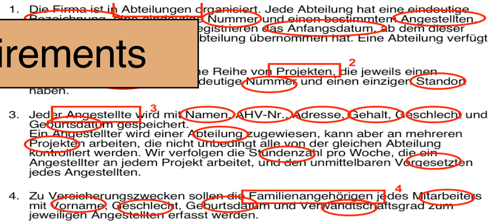


Entities in the real world:
Department head Müller,
Employee Muster,
Project 'Budget 2014'

Lecture 3 — Requirements Analysis

Interview of users, study of documentation

Database Requirements


1. Die Firma ist in Abteilungen organisiert. Jede Abteilung hat eine eindeutige Bezeichnung, eine eindeutige Nummer und einen bestimmten Angestellten. … egistrieren das Anfangsdatum, ab dem dieser … bteilung übernommen hat. Eine Abteilung verfügt

… ne Reihe von Projekten, die jeweils einen … deutige Nummer und einen einzigen Standort haben.

3. Jeder Angestellte wird mit Namen, AHV-Nr., Adresse, Gehalt, Geschlecht und Geburtsdatum gespeichert. Ein Angestellter wird einer Abteilung zugewiesen, kann aber an mehreren Projekten arbeiten, die nicht unbedingt alle von der gleichen Abteilung kontrolliert werden. Wir verfolgen die Stundenzahl pro Woche, die ein Angestellter an jedem Projekt arbeitet, und den unmittelbaren Vorgesetzten jedes Angestellten.

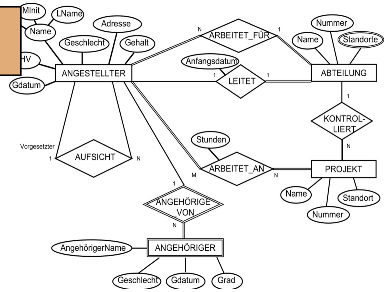
4. Zu Versicherungszwecken sollen die Familienangehörigen jedes Mitarbeiters mit Vorname, Geschlecht, Geburtsdatum und Verwandtschaftsgrad zum jeweiligen Angestellten erfasst werden.

Written concise description of requirements (data reqs and functional [operations] reqs)

Lecture 4 — Conceptual DB design

Text analysis, Entities, Relationships

ER-Model



Entity types with attributes and relationships

DBMS-independent
DBMS-specific

Lecture 5 — Logical DB design (data model mapping)

- translation of model to implementation
- Validation through normalisation

DB-Schema (e.g. relational data model)

NGESTELLTER

VNAME	INITAL	NNAME	AHV	GDATUM

ADRESSE	GESCHLECHT	GEHALT	SUPERAHV	ANR

BTEILUNG

ANAME	ABTNUMMMER	MGRAHV	MGR_ANFANGSDATUM

BT_STANDORTE

BTNUMMMER	ASTANDORT

ARBEITET_AN

EAHV	PNR	STUNDEN

PROJEKT

PNAME	PNUMMER	PSTANDORT	ABTNR

ANGEHÖRIGER

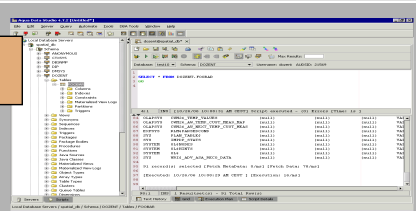
EAHV	ANGEHÖRIGER_NAME	GESCHLECHT	GDATUM	GRAD

Set of relations with attributes (incl. data types + value domains)

Lecture 5 — Physical DB design

Description of database implementation (HW, indices, …)

Database



Implemented DB design, Tables (ordered) with rows and columns

➲ Elmasri & Navathe (2014): Ch 7-9

DB–English English–DB

Strong Entity Type	Entity type with its own key attribute
Weak Entity Type	Entity type without its own key attribute , for identifying an entity it needs to use the key attribute(s) of a different entity type
Partial Key	Key of a weak entity type which is only unique together with the key of a different entity type.
Primary Key	Attribute (column) of a relation (table) which identifies a single record of this relation; never NULL
Foreign Key	Attribute(s) of a relation (table) which build a relationship to a primary key of a different relation (table)
Relation	“Table” in the relational model; consists of unordered tuples (rows), attributes (columns) and domains
Table	Table in the database; consists of rows and columns