Lecture 5 – Real-world Problems and Normalization

CPE241 – Database Systems

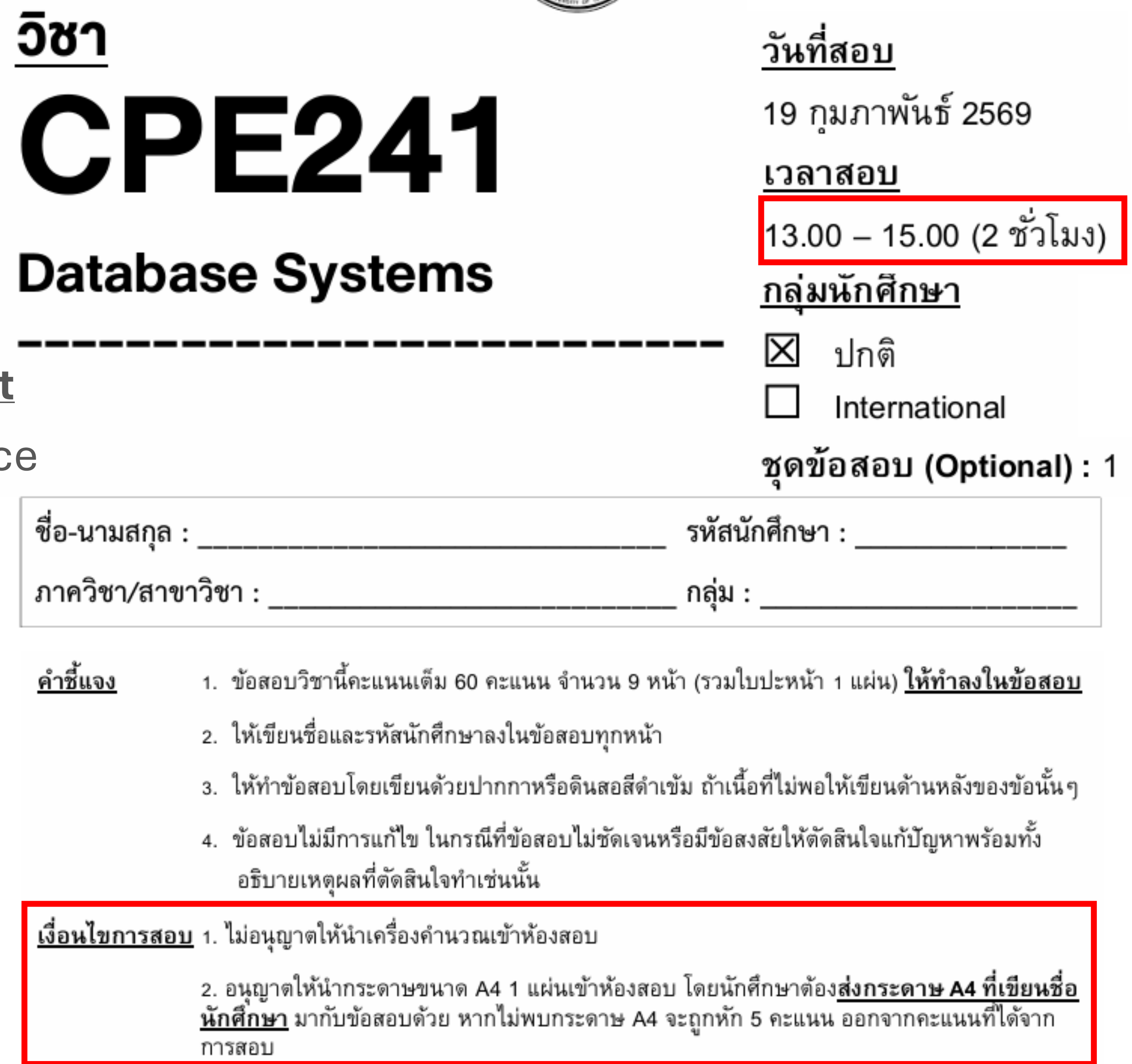
10 February 2026

Dr. Jaturon Harnsomburana
Dr. Piyanit Ua-areemitr

Department of Computer Engineering
King Mongkut's University of Technology Thonburi

About Exam 1

- 2-hour exam
- One A4 & you need to **submit**
- 20 marks from multiple choice questions
- 20 marks from short answer questions
- 10 marks from EERD
- 10 marks from SQL



วิชา

CPE241

Database Systems

วันที่สอบ

19 กุมภาพันธ์ 2569

เวลาสอบ

13.00 – 15.00 (2 ชั่วโมง)

กลุ่มนักศึกษา

☒ ปกติ

☐ International

ชุดข้อสอบ (Optional) : 1

ชื่อ-นามสกุล : ______________________________ รหัสนักศึกษา : ______________

ภาควิชา/สาขาวิชา : __________________________ กลุ่ม : ____________________

คำชี้แจง

1. ข้อสอบวิชานี้คะแนนเต็ม 60 คะแนน จำนวน 9 หน้า (รวมใบปะหน้า 1 แผ่น) **ให้ทำลงในข้อสอบ**
2. ให้เขียนชื่อและรหัสนักศึกษาลงในข้อสอบทุกหน้า
3. ให้ทำข้อสอบโดยเขียนด้วยปากกาหรือดินสอสีดำเข้ม ถ้าเนื้อที่ไม่พอให้เขียนด้านหลังของข้อนั้นๆ
4. ข้อสอบไม่มีการแก้ไข ในกรณีที่ข้อสอบไม่ชัดเจนหรือมีข้อสงสัยให้ตัดสินใจแก้ปัญหาพร้อมทั้ง อธิบายเหตุผลที่ตัดสินใจทำเช่นนั้น

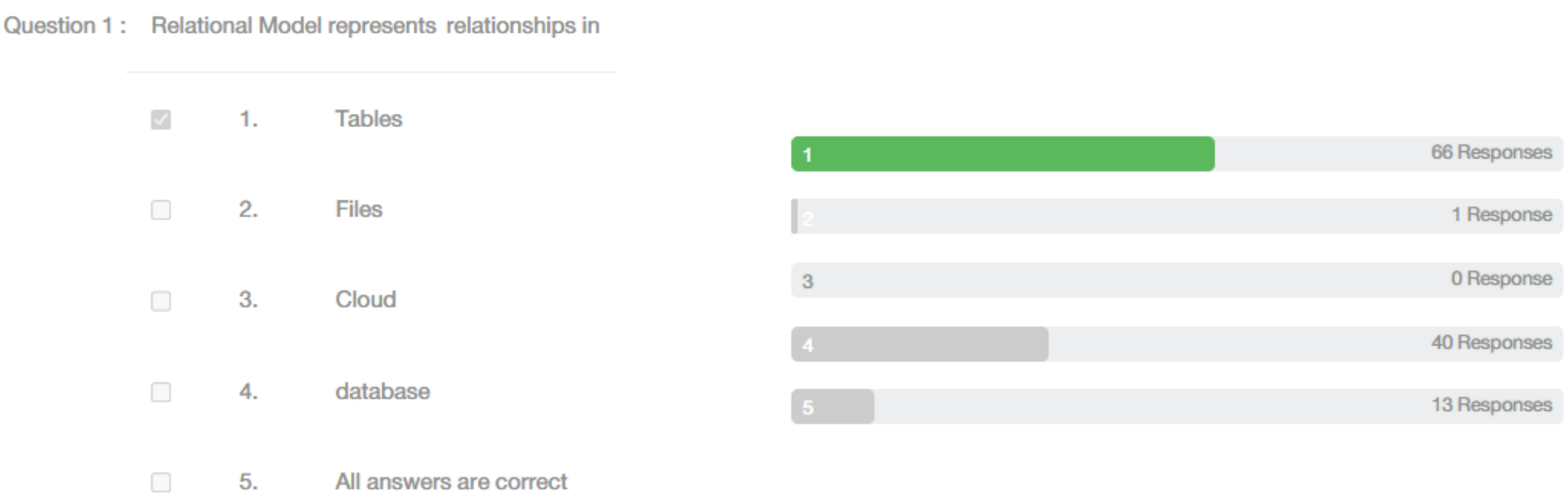
เงื่อนไขการสอบ

1. ไม่อนุญาตให้นำเครื่องคำนวณเข้าห้องสอบ
2. อนุญาตให้นำกระดาษขนาด A4 1 แผ่นเข้าห้องสอบ โดยนักศึกษาต้อง**ส่งกระดาษ A4 ที่เขียนชื่อนักศึกษา** มากับข้อสอบด้วย หากไม่พบกระดาษ A4 จะถูกหัก 5 คะแนน ออกจากคะแนนที่ได้จากการสอบ

Recap

- Data models: conceptual, logical, and physical models
- Database schemas
- The basic relational model: data, relationship & constraints
- ERD: Chen, Crow’s Foot & UML
- Advanced ERD
  - Weak ER
  - Superclass/subclass
  - Generalization/specialization
  - Disjoint/overlapping
  - Total/partial
- Basic SQL

Recap



Question 1 : Relational Model represents relationships in

☑ 1. Tables

□ 2. Files

□ 3. Cloud

□ 4. database

□ 5. All answers are correct

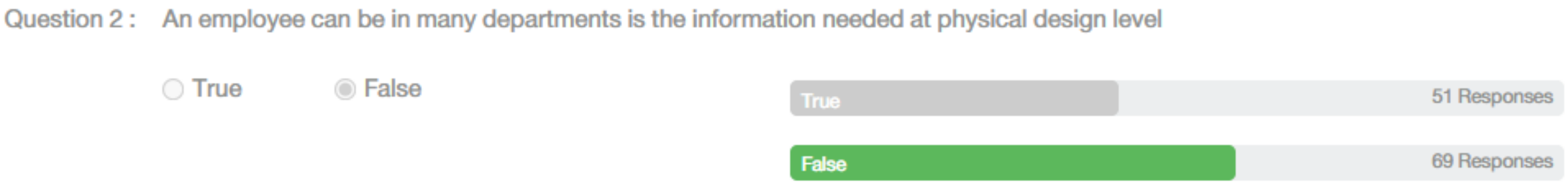
1 66 Responses

2 1 Response

3 0 Response

4 40 Responses

5 13 Responses



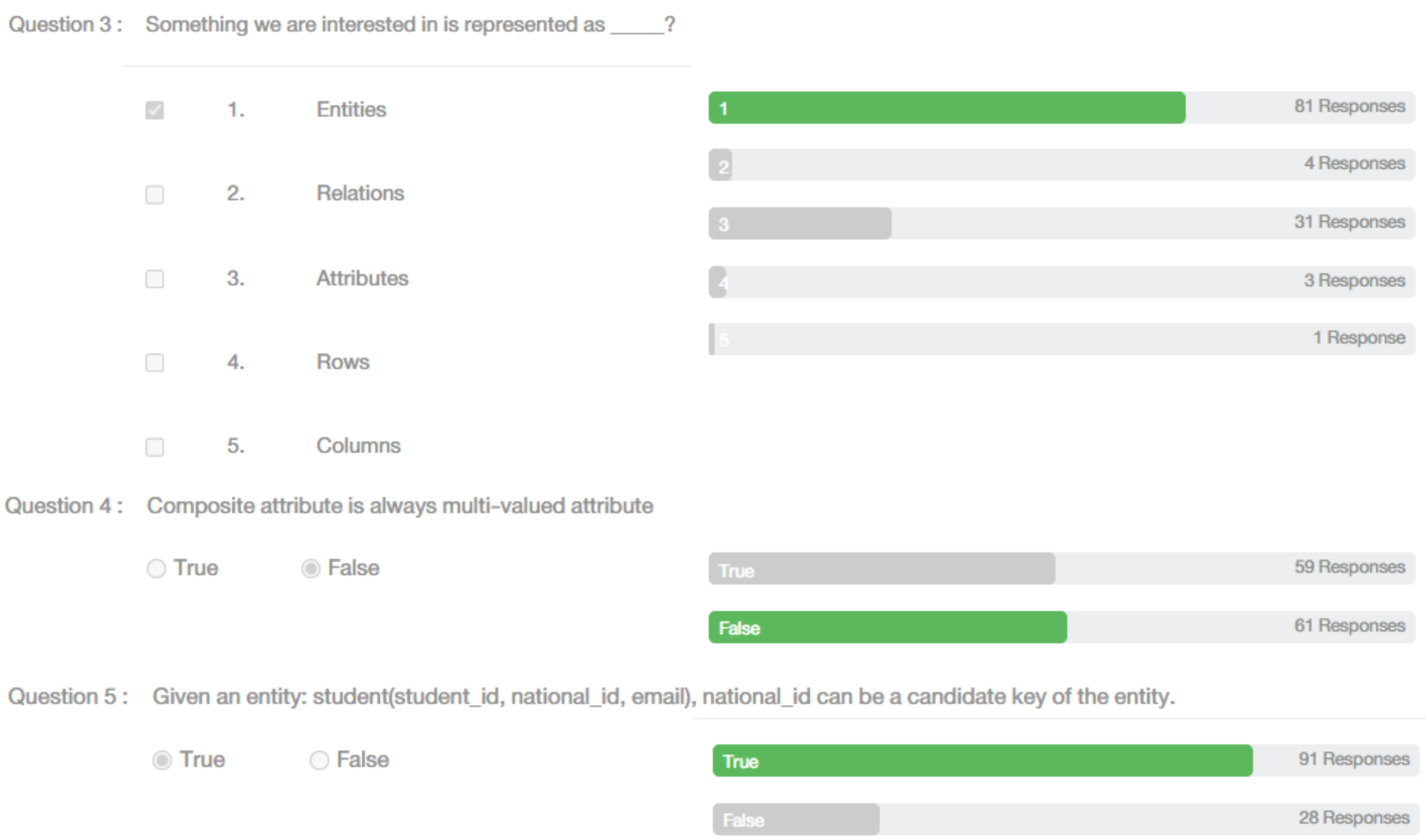
Question 2 : An employee can be in many departments is the information needed at physical design level

○ True ● False

True 51 Responses

False 69 Responses

Recap



Question 3 : Something we are interested in is represented as _____?

☑ 1. Entities

□ 2. Relations

□ 3. Attributes

□ 4. Rows

□ 5. Columns

1 — 81 Responses

2 — 4 Responses

3 — 31 Responses

4 — 3 Responses

5 — 1 Response

Question 4 : Composite attribute is always multi-valued attribute

○ True ● False

True — 59 Responses

False — 61 Responses

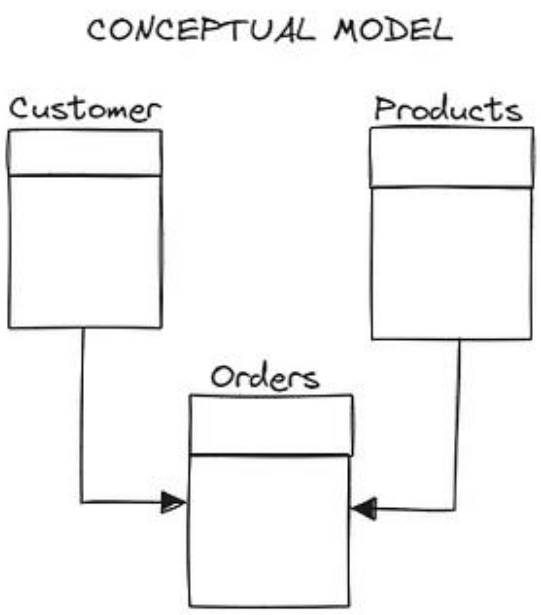
Question 5 : Given an entity: student(student_id, national_id, email), national_id can be a candidate key of the entity.

● True ○ False

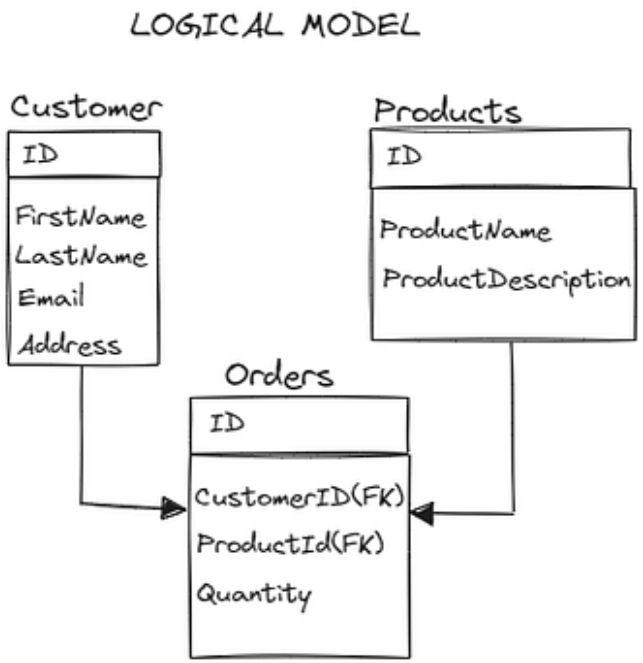
True — 91 Responses

False — 28 Responses

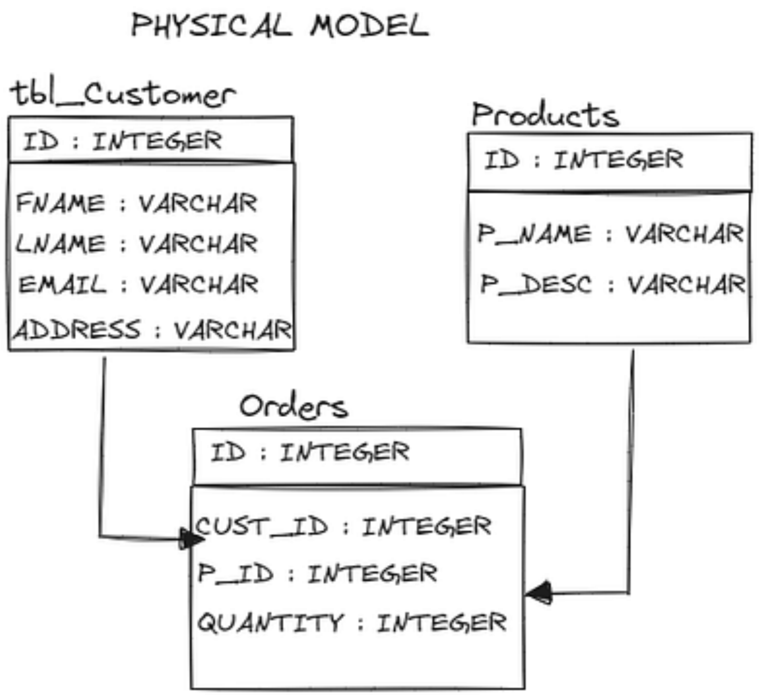
Recap



Entities
Relation



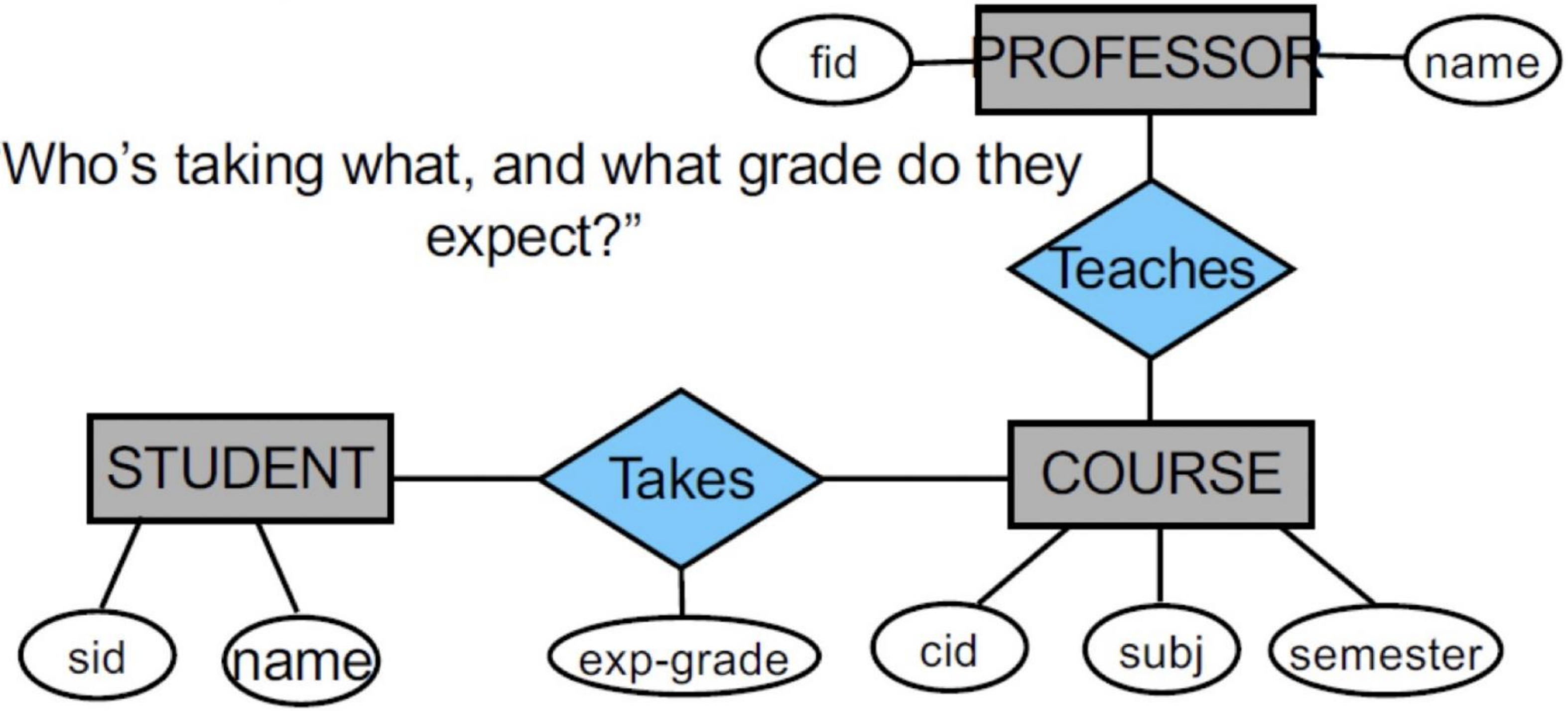
Attributes
Keys



Data Types
Constraints

Source: https://medium.com/@factor-bytes/explained-conceptual-logical-and-physical-data-models-0786b0e90216

Recap



“Who’s taking what, and what grade do they expect?”

Recap

- Constraints determine which values are permissible (or not) in the database.
  - **Inherent or Implicit Constraints**
    Based on the data model itself (Ex. a relation cannot have duplicate tuples)
  - **Schema-based or Explicit Constraints, Integrity Constraints**
    Rules or restrictions that apply to the database and limit the data values it may store
  - **Application based or Semantic Constraints**
    Beyond the expressive power of the model and must be specified and enforced by application programs

Recap

- **Integrity Constraints (ICs)**
  - Conditions that must be true for *any instance of the database* e.g. domain constraints, key constraints
  - ICs are **specified** when schema is defined and **checked** when relations are modified.
  - A **legal instance** of a relation is one that satisfies all specified ICs.
  - DBMS should not allow illegal instances.
  - Integrity rules are used to ensure the data is accurate.

Recap

- Constraint types
  - **Domain constraint** - every value for an attribute must be an element of the attribute's domain or be null.
  - **Entity integrity constraint** - in a base relation, no attribute of a primary key can be null.
  - **Key constraint** - every relation must have a key.
  - **Referential integrity constraint** - if a foreign key exists in a relation, then the foreign key value must match a primary key value of a tuple in the referenced relation or be null.

Today’s Goals

- Real-world problem to an ERD
- An ERD to relational database
- Normalization

Case Study

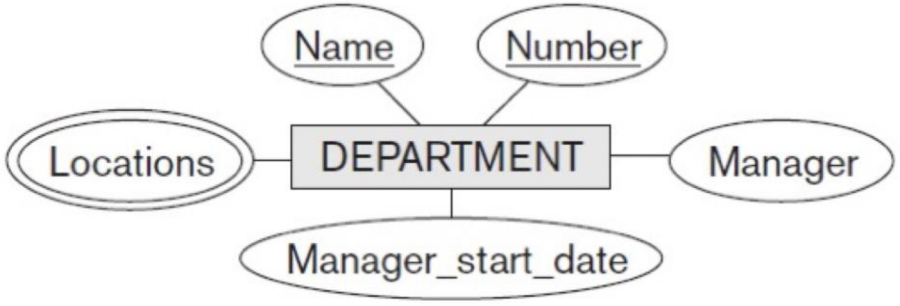
- Requirements of a COMPANY database (1)
  - The company is organized into DEPARTMENTs. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date of the department manager. A department may have several locations.
  - Each department controls a number of PROJECTs. Each project has a unique name, unique number, and is located at a single location.

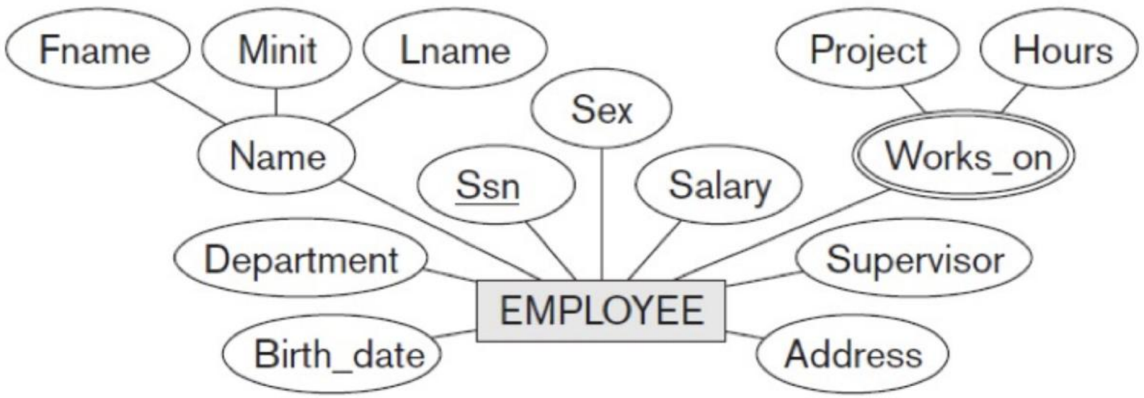
Case Study

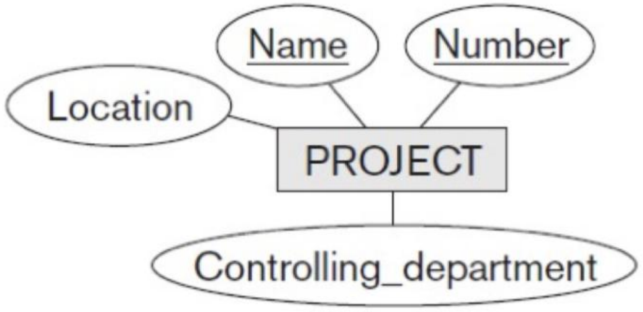
- Requirements of a COMPANY database (2)
  - The database will store each EMPLOYEE's social security number, name, address, salary, sex, and birthdate.
    - ✓ Each employee works for one department but may work on several projects.
    - ✓ The DB will keep track of the number of hours per week that an employee currently works on each project.
    - ✓ It is required to keep track of the direct supervisor of each employee.
  - Each employee may have a number of DEPENDENTs.
    - ✓ For each dependent, the DB keeps a record of name, sex, birthdate, and relationship to the employee.

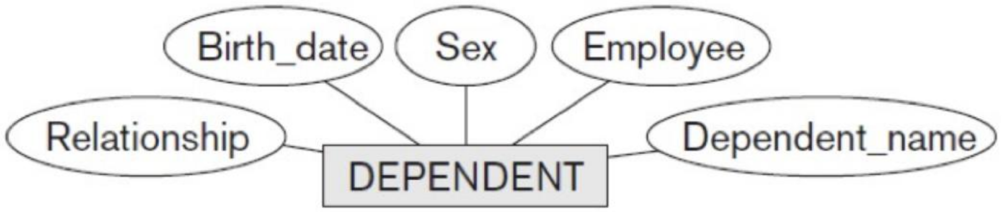
Initial Conceptual Design

- 4 Entity types









Relationship

- The fact that an employee can work on several projects.

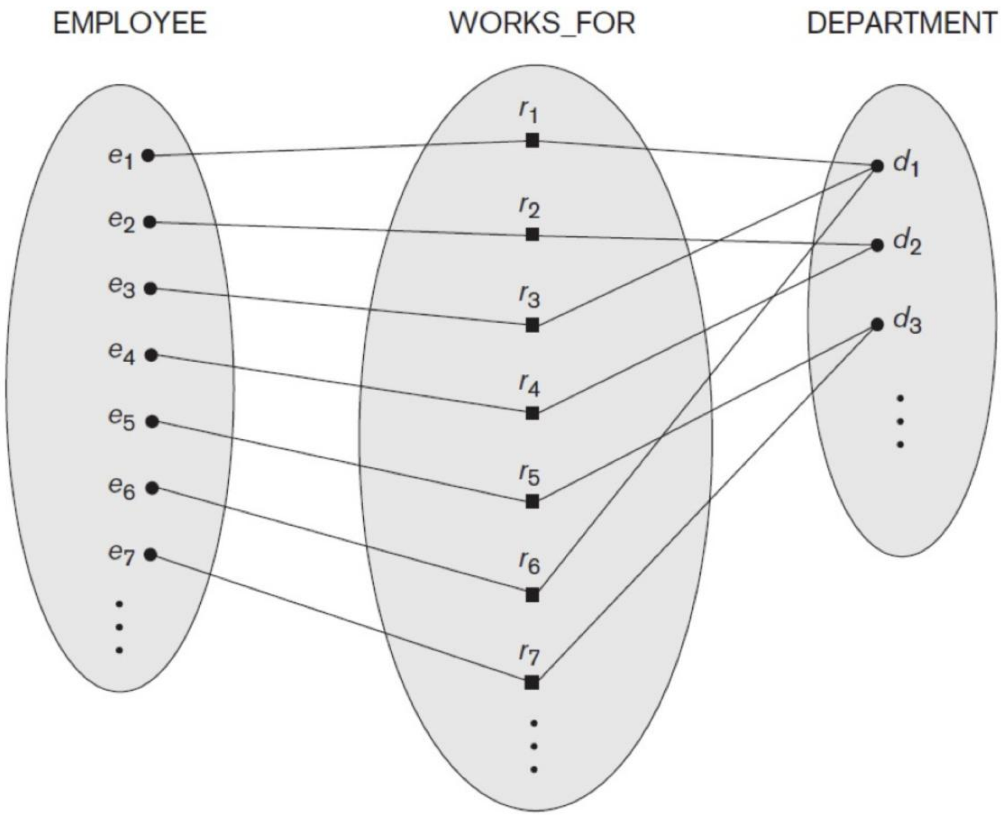
- Implicit relationships among various entity types.
  - Manager of DEPARTMENT -> An EMPLOYEE who manages the department
  - Controlling_department of PROJECT -> The DEPARTMENT that controls the project.
  - Supervisor of EMPLOYEE -> An EMPLOYEE

- Whenever an attribute of one entity type refers to another entity type, some relationship exists.

Relationship

- **Relationship type** - a schema description of a relationship
  - o Identify a relationship name and the participating entity types
  - o Identify relationship constraints

- **Relationship set** - the current set of relationship instances represented in a database
  - o The current state of a relationship type



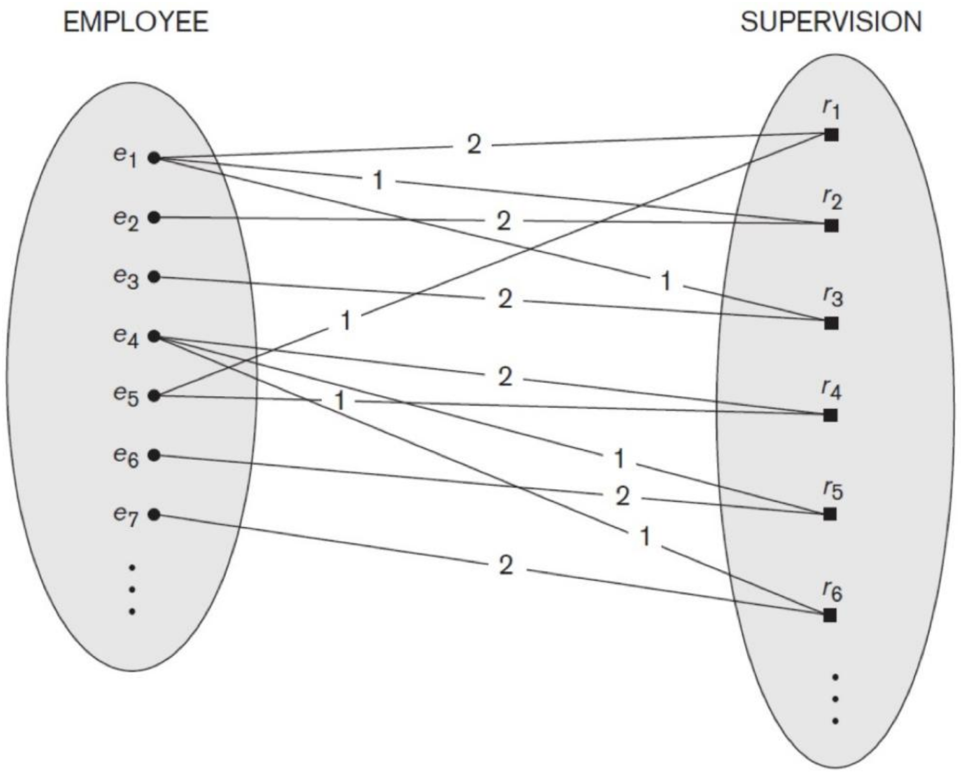
Recursive Relationships

- A self-referencing relationship type
- A relationship type between the same participating entity type in distinct roles

The EMPLOYEE entity type participates twice in the SUPERVISION relationship as
(1) The supervisor role (boss), and
(2) The supervisee role (subordinate).

Recursive Relationships

- Each relationship instance relates two distinct EMPLOYEE entities: one employee in the supervisor role and one employee in the supervisee role.



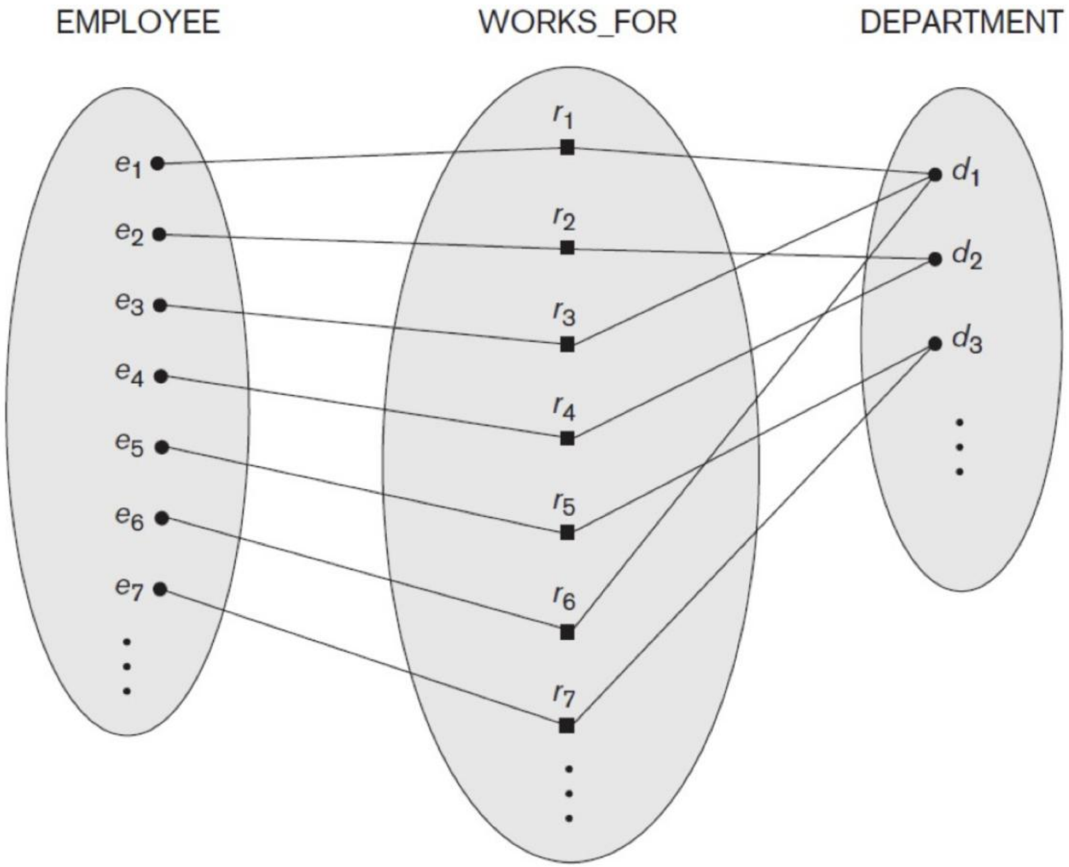
Let the lines marked ‘1’ represent the supervisor role, and those marked ‘2’ represent the supervisee role.

Constraints on Relationship

- **Cardinality Ratios** for Binary Relationships
  - Specify the ***maximum*** number of relationship instances that an entity can participate in.
  - Types: one-to-one (1:1), one-to-many (1:N, N:1), many-to-many (M:N)
- **Participation Constraints**, Existence Dependency Constraints
  - **Modality** specifies the ***minimum*** number of relationship instances that an entity can participate in.
  - Total participation: 1 or more (mandatory participation)
  - Partial participation: 0 (optional participation)

Cardinality Ratios

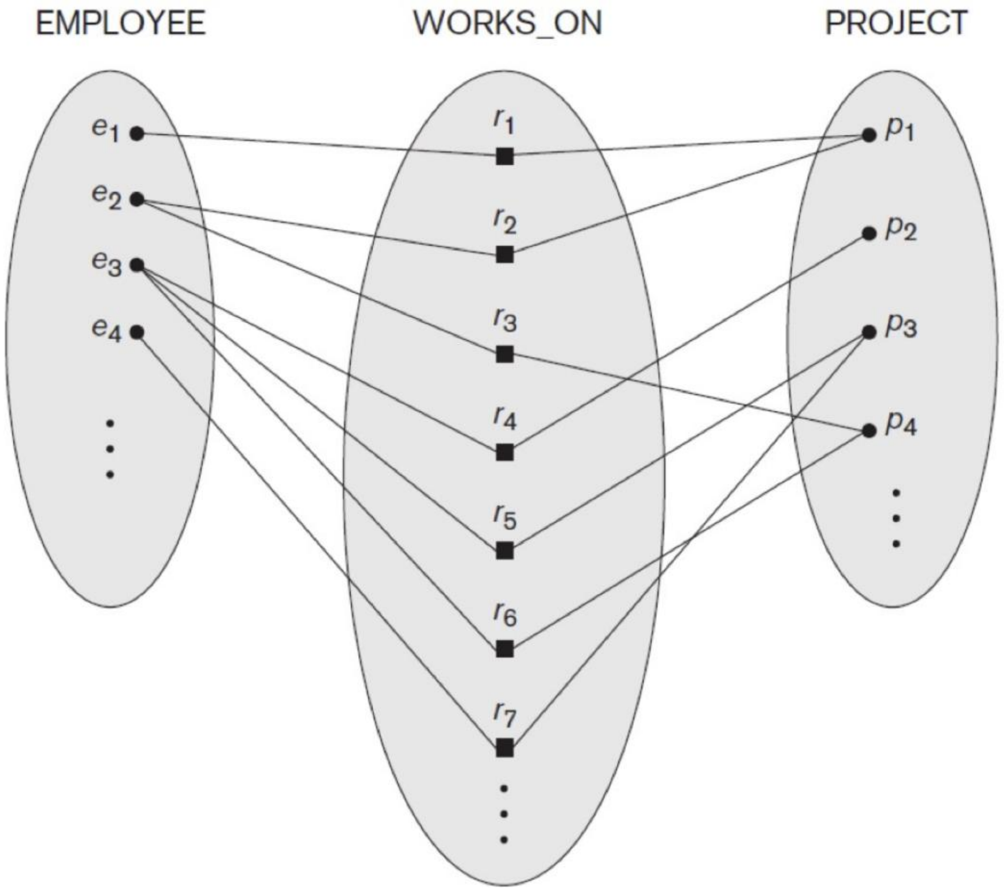
- N:1



Each department can be related to any number of employees, but an employee can be related to only one department.

Cardinality Ratios

- M:N



An employee can work on several projects and a project can have several employees.

Weak Entity Types

- An entity that *does not have a key attribute* and that is *identification dependent on another entity type*.

- An identifying relationship - relates a weak entity type to its owner (or identifying) entity type.

- Entities are identified by the combination of:
    - A partial key of the weak entity type (the attribute that can uniquely identify weak entities.
    - The particular entity they are related to in the identifying relationship type.

Weak Entity Types

- DEPENDENT is a weak entity type.
- Name of DEPENDENT a partial key assuming that no two dependents of the same employee have the same first name.
- A DEPENDENT entity is identified by the dependent’s first name and the specific EMPLOYEE whom the dependent is related.
- Employee is its identifying entity type via the identifying relationship type DEPENDENT_OF

Refined ER Diagram

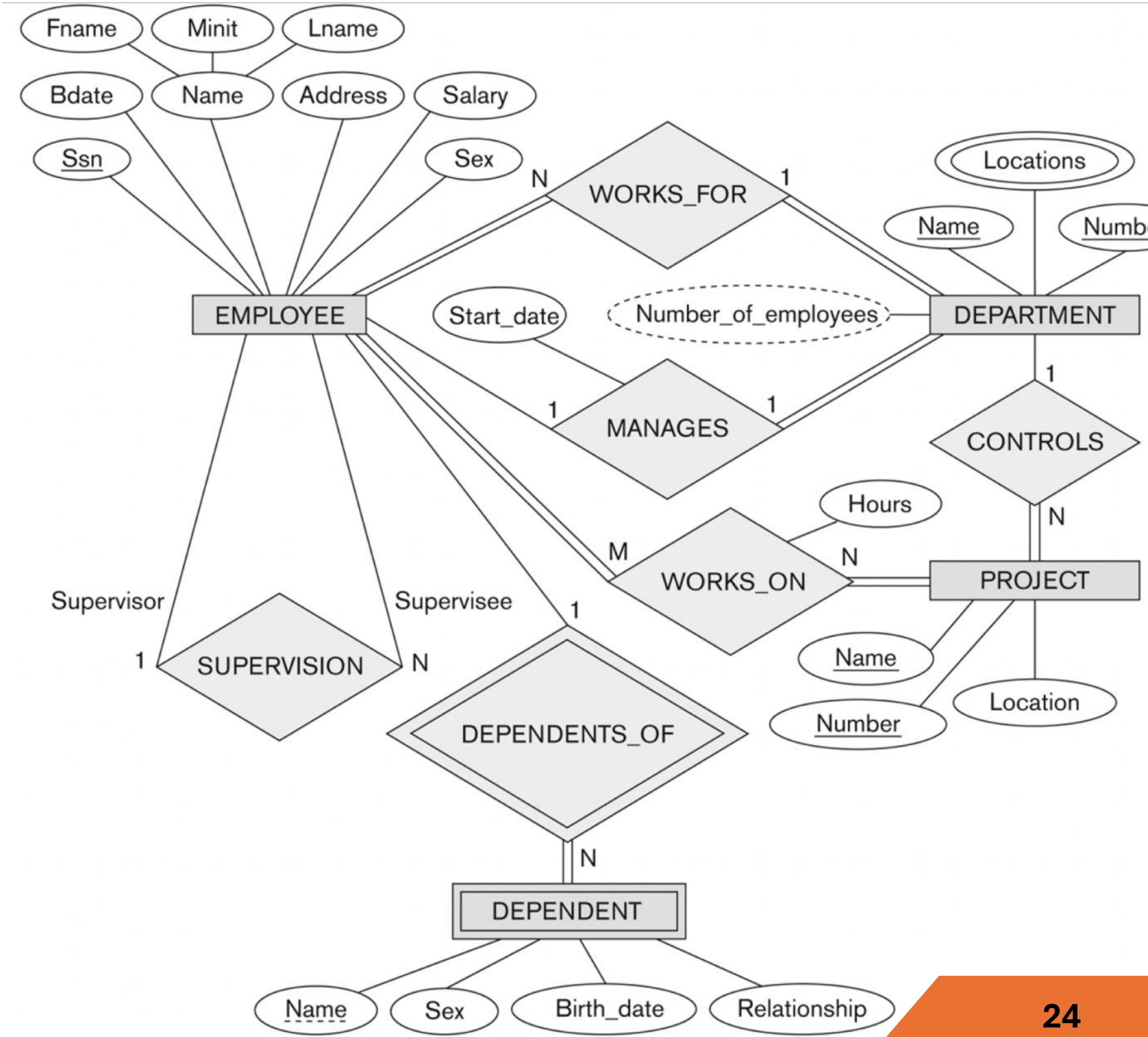
- Use cardinality constraints

- The company is organized into DEPARTMENTs. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date of the department manager. A department may have several locations.

- Each department controls a number of PROJECTs. Each project has a unique name, unique number, and is located at a single location.

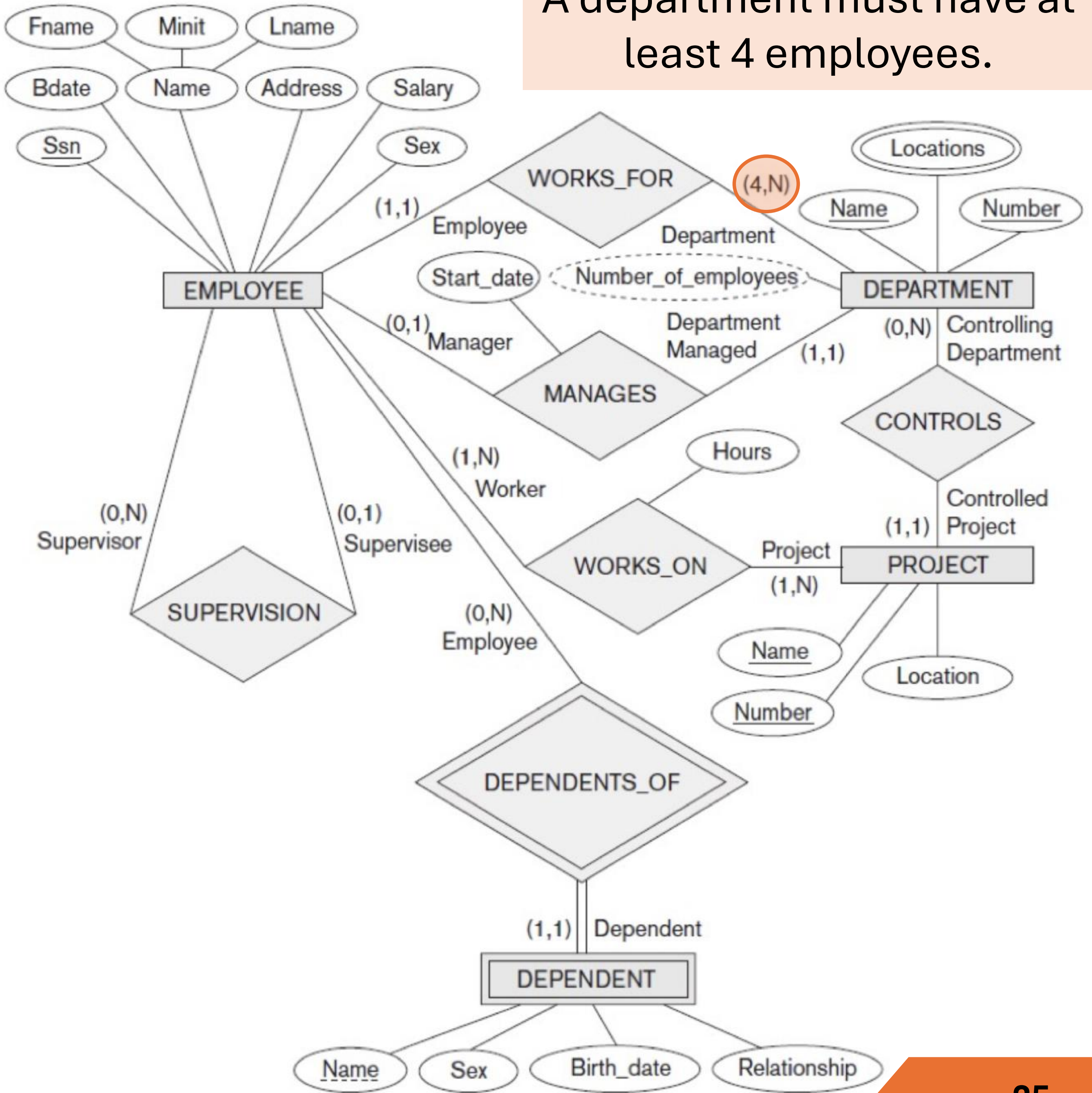
- The database will store each EMPLOYEE's social security number, name, address, salary, sex, and birthdate.
  - ✓ Each employee works for one department but may work on several projects.
  - ✓ The DB will keep track of the number of hours per week that an employee currently works on each project.
  - ✓ It is required to keep track of the direct supervisor of each employee.

- Each employee may have a number of DEPENDENTs.
  - ✓ For each dependent, the DB keeps a record of name, sex, birthdate, and relationship to the employee.



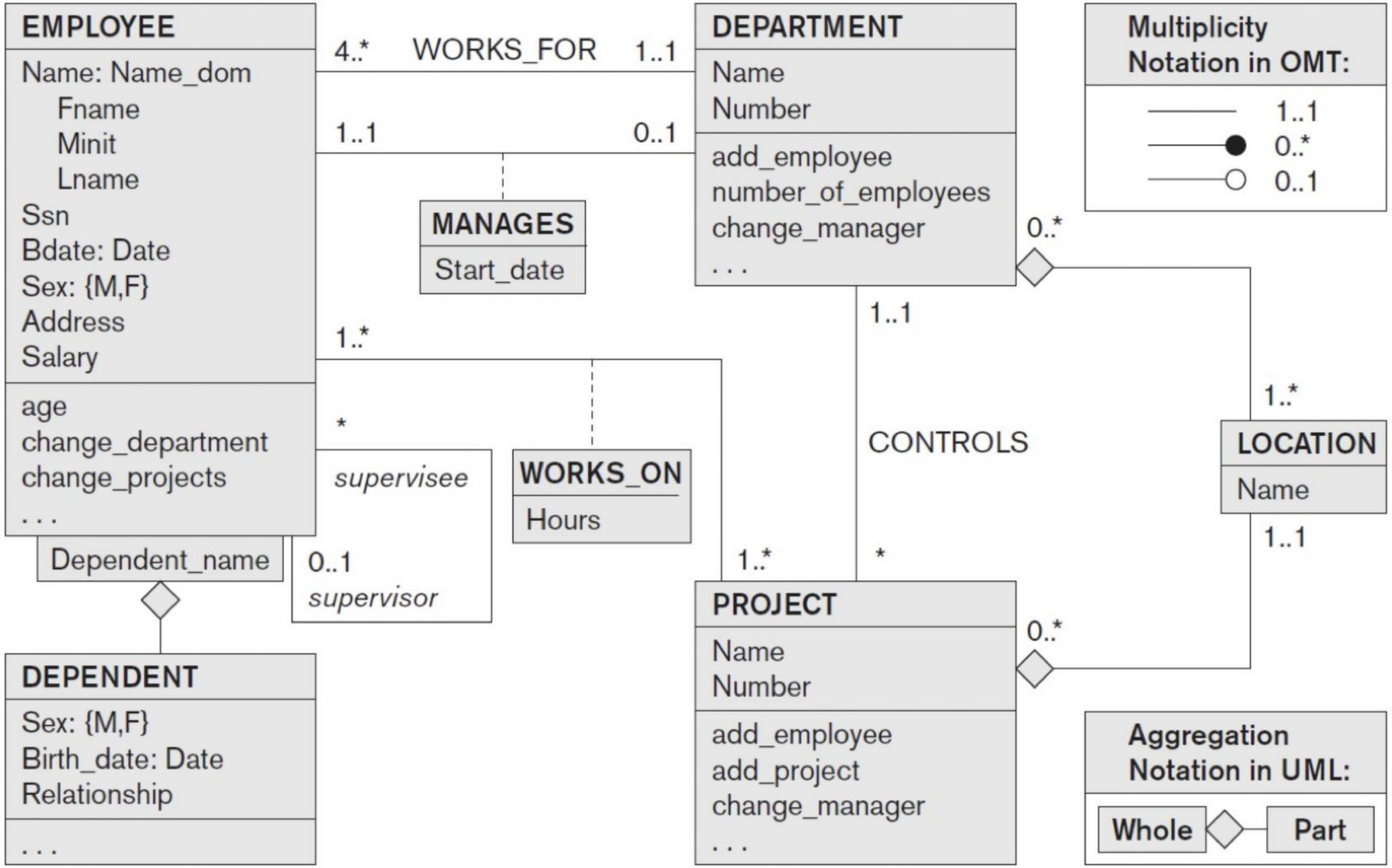
Refined ER Diagram

- Use participating constraints as a pair of integer number (min, max) with each participation of an entity type E in a relationship type R.
- 0 <= min <= max, and max >= 1
- For each entity e must participate in at least min and at most max relationship instance in R at any point in time.
- min = 0 implies partial participation.
- min > 0 implies total participation.



A department must have at least 4 employees.

UML Class Diagram



Wrap up

- Create ERD from real-world examples.
- Self-exercise (for reviewing)
- PS04 ERD of your term project

Coming Next Time

- ER-to-relational mapping
- Normalization