COMP-3150 CHAPTERS 1 TO 3: LECTURE NOTES (Chs 2 and 3)

~~Chapter 1: Databases and Database Users~~ (see the previous posting on Ch. 1 lectures)
Chapter 2: Database System Concepts and Architecture
Chapter 3: Data Modeling Using the Entity-Relationship (ER) Model.

Fall 2021 Comp 3150 online recordings can be downloaded from the black board virtual classroom for the class of the day. The links below are the recordings of similar classes in Fall 2020 that I uploaded to one drive in case you prefer to review these.

Links to any posted recorded in-class lectures for Chapters 2 and 3 classes done on Sept. 22, 2020 (similar to that of Sep. 21, 2021); Sept. 24, 2020 (similar to that of Sep. 23, 2021) and Sept. 29, 2020 (similar to that of Sep. 28, 2021) will be found below:

1. Links on one drive to recorded lecture of Tuesday, Sept. 22, 2020 (similar to Sep. 21, 2021) (saved in two recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EU-nO0KqewFCjpld2DELPm8BxAGXE0PD4fMyQHbzFTPQww

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/ERaOJlsHixFDueqeW-EYHM8Bgxpb8kaL9kOqOLaGZK_ZVA

2. Links on one drive to recorded lecture of Thursday, Sept. 24, 2020 (similar to Sep. 23, 2021) (saved in two recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EdaxENNyy_9Dv_vSc2eHsw4B_10ve0i_b6iiUEoe9OtY1g

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EYixmCcqs6VEqQkIqiTtuPMB31dd3vu0q6NVHbvS0arhDg

3. Links on one drive to recorded lecture of Tuesday, Sept. 29, 2020 (similar to Sep. 28, 2021) (saved in two recorded sessions to break down the files) is provided below in sequences 1, 2:

Frist Recording 1 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EUWRFAK5Mb1CmEt8LJPKqQsBmajvYqtYQi7FS68NCTa4AA

Second Recording 2 of 2:

https://uwin365-my.sharepoint.com/:v:/g/personal/cezeife_uwindsor_ca/EfPTx7E3O5lHmhwyk_340HsBMsTdff23_qgu1FTiVcsSmw

***

The following questions on database systems concept and architecture discussed in Chapter 2 of Comp 3150 text book, Chapter 2, Comp 3150 posted course slide notes, are in-class questions for students to ponder and answer as I teach.

- The answers to the questions are found also by reviewing the Comp 3150, posted power point slide notes for Chapter 2 and being in class.

- Students are advised to review Chapters 1 to 3 of course book and Comp 3150 posted slide notes before and after each class.

- I will also go over the Slide notes in class with examples and integrate them into the class lectures, which are also posted in the More course material link on black board with any links to recorded live lectures.

1. WHAT ARE DATA MODELS, DATABASE SCHEMAS AND INSTANCES?

2. WHAT ARE THREE-SCHEMA ARCHITECTURE OF THE DATABASE?

3. WHAT DOES PROVISION OF DATA INDEPENDENCE MEAN?

4. WHAT ARE DBMS LANGUAGES AND INTERFACES?

5. WHAT ARE DATABASE SYSTEM UTILITIES AND TOOLS?

6. WHAT ARE CENTRALIZED AND CLIENT-SERVER DBMS ARCHITECTURES

7. WHAT IS THE HISTORY OF DATABASE DATA MODELS?

**

Try to answer the 7 questions above with an example with reference to a simple 3-table database of student-takes-course given in Chapter 1 lecture in class.

An Example simple database Problem.
You are asked to define a simple student record

database for keeping track of student grades in courses they take.

Solution

1. An example data model is relational data model where data representing entities and their relationships are presented as relations (or tables)

   An example database schema (showing the structure of each of its entities and relationships) for the Student-takes-course database is:

   Student (Studid, Name, Class, Major)
   Course (Cid, Ctitle, Credithr, Dept)
   Takes (Studid, Cid, grade)

An instance of the database showing the contents of the database at a particular point in time is:

Student

Studid	Name	Class	Major
1	John Smith	1	CS
2	Patty Moore	2	Math

Course

Cid	Ctitle	Credithr	Dept
Comp1400	Intro to C	3	CS
Comp3150	Database	3	CS
Math1720	Calculus	3	Math

Takes

Studid	Cid	Grade
1	Comp1400	80
1	Math1720	70
2	Comp1400	60

2. Three-Schema Architecture

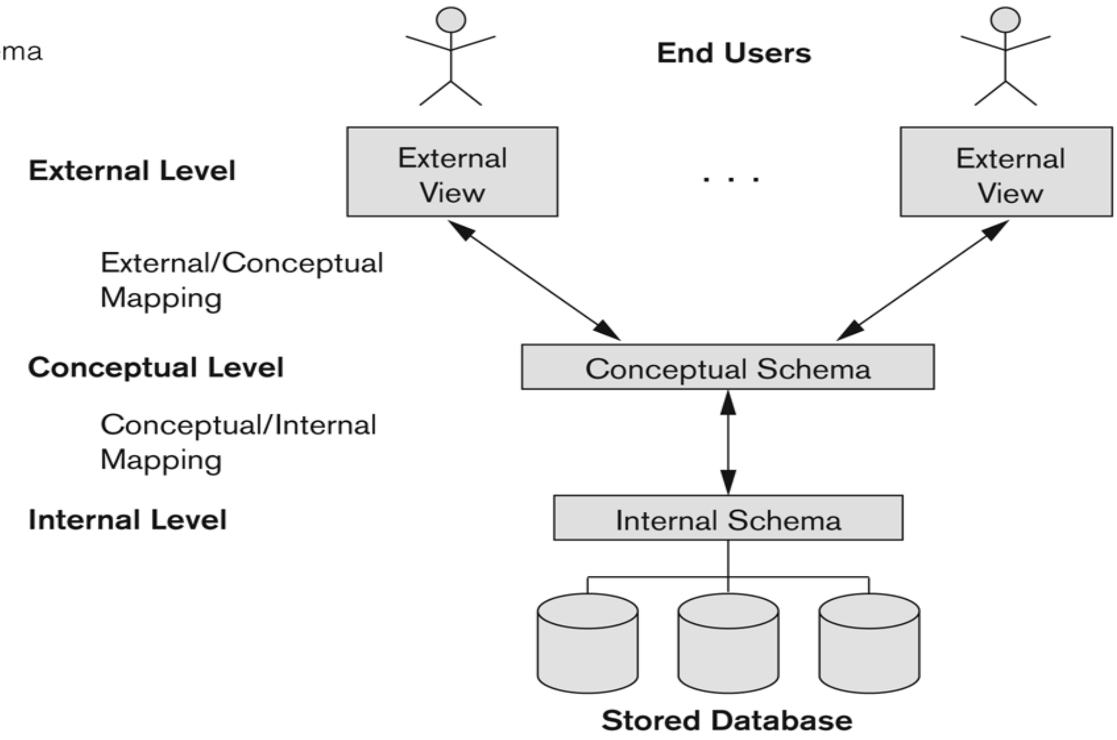


Figure 2.2
The three-schema architecture.

An example external level schema for user queries like "give me the name and studid of all CS students" is: Student (Name, Studid).

Conceptual level Schema for Student in this example is: Student (Studid: string (4), Name: string (20), Class: Number (1), Major: string (4))

An example internal /physical storage format for a student record (that is the actual physical structure for storing on disk the student table defined in the conceptual schema above (as given in Fig. 13, p13 of book) is:

Data item Name	record Start position	length(bytes)
Studid	1	4
Name	5	20
Class	25	1
Major	26	4

- 3. What are data independence provided by the DBMS?
  Slide 15 of Chapter 2 notes describes this as: This is the ability of the DBMS to change the schema at a lower level of DB system without having to change the schema at the next higher level.

- DBMS provides the following two types of data independence.

- 1. Logical data independence
  The ability to change the conceptual schema (e.g. modify the schema of Student by adding address as an additional column) without having to change

the external schema or application program (which gets the name and studid)

- 2. Physical data independence: the ability to change the internal/physical schema without having to change the conceptual schema. For example, an access path (eg. Index) to improve retrieval speed of STUDENT records should not require a query like “list all students’ names and studid” to be changed.

1. DBMS LANGUAGES AND INTERFACES
   Discussed on slides 18 to 22 of chapter 2 slide notes.

2. DATABASE SYSTEM UTILITIES AND TOOLS
   Discussed on slides 23 to 25.

3. CENTRALIZED AND CLIENT-SERVER ARCHITECTURES.
   Discussed on slides 26 and 27.

4. HISTORY OF DATA MODELS
   Discussed on slide 28.

**

Chapter 3 Continues with Database Design through the Entity-Relationship model.

Slides 7 to 8 of Chapter 3 slides presents you with a requirements description of a COMPANY database and your job is to design the entity-relationship model database for this database.

- ■ We need to create a database schema design based on the (simplified) **requirements** of the COMPANY Database presented on slides 7 and 8.

- Extract all the entities and their attributes. Also, extract all the relationship between entities and their attributes.

- Note that initially, some attributes of an entity may be an implicit relationship that can later be refined to be an explicit relationship in the final design. For example, in Department, Manager is an implicit relationship which is later refined to a “Manages” explicit relationship.

- Identified entities are 4 with all initial attributes:

  1. DEPARTMENT (Name, Number, Manager, Manager_start_date, {Location})

  2. PROJECT (Name, Number, Location, Controlling_department)

  3. EMPLOYEE (SSN, Name (Fname, Minit, Lname), Address, Department, Birthdate, Sex, Supervisor, Salary, {Works_on(Project, Hours)})

  4. DEPENDENT (Dependent_name, Employee, Relationship, Birth_date, Sex)

- Relationships are 5 (now described in words but should also be clearly presented in Schema):

  1. An Employee manages Department
  written in schema form as:
  Manages(SSN, DNumber, Start_date)
  *** Note that this relationship Manages is an example of refining an implicit relationship attributes Manager, Manager_start_date in the Entity (1) schema Department above.

  2. Each Department controls some PROJECTS.
  This is written in schema form as:
  Controls(DNumber, PNumber)

  3. Each Employee works for one Department but

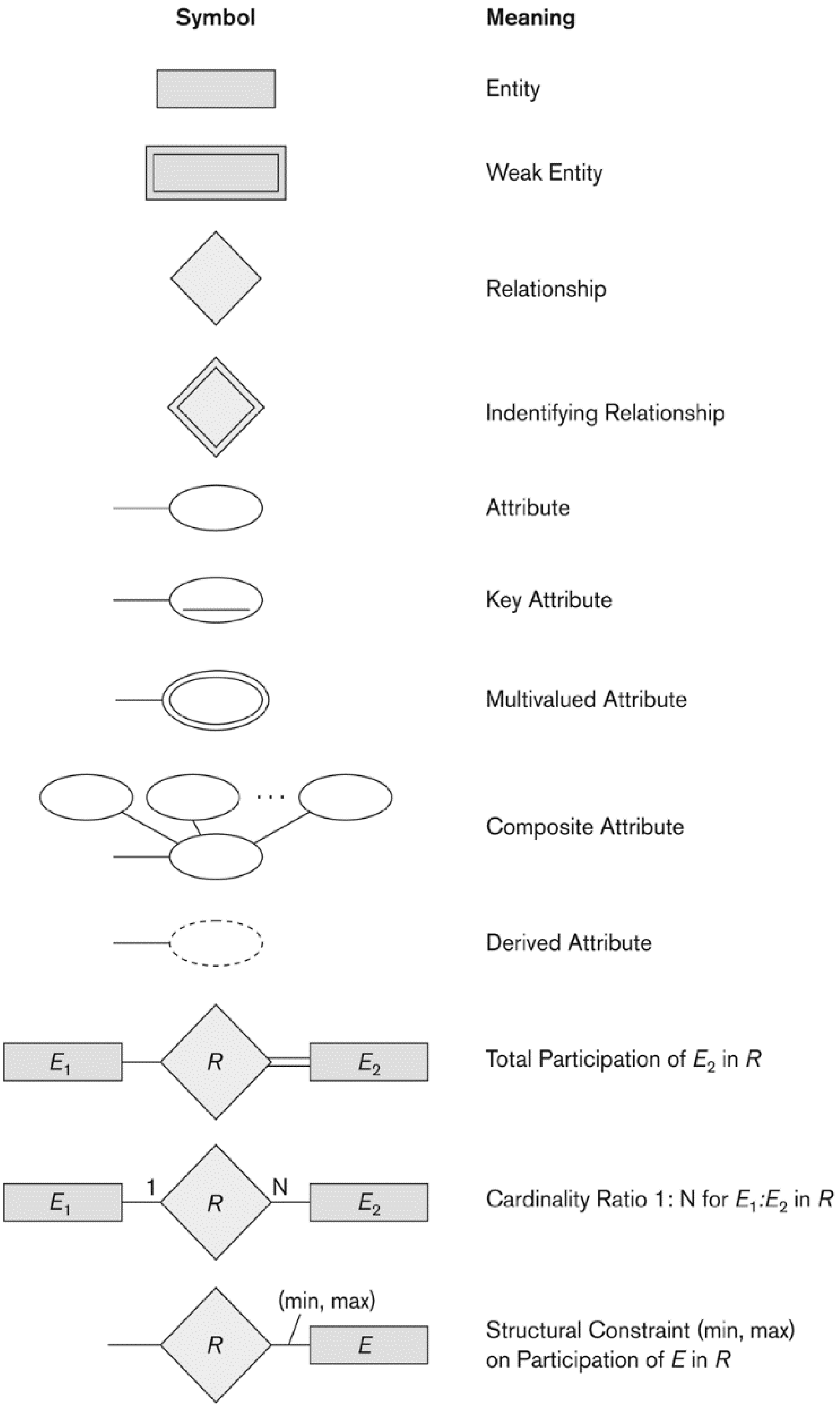
may work on several PROJECTS. This is used in the explicit relationship Works_For(Ssn, Dnumber).

4. Each Employee has a supervisor. This is a recursive relationship as: Supervision(SSsn, ESsn).

5. Each Employee may have some DEPENDENTs. This is an identifying relationship for a weak entity type as: Dependents_of(Ssn, Name)

- Slide 19 of Chapter 3 presents a summary of symbols for representing different aspects of the model.

Figure 3.14
Summary of the notation for ER diagrams.


Symbol
Meaning
Entity
Weak Entity
Relationship
Indentifying Relationship
Attribute
Key Attribute
Multivalued Attribute
. . .
Composite Attribute
Derived Attribute
E_1
R
E_2
Total Participation of E_2 in R
E_1
1
R
N
E_2
Cardinality Ratio 1: N for E_1:E_2 in R
(min, max)
R
E
Structural Constraint (min, max) on Participation of E in R

More Descriptions of meaning of symbols of Fig. 3.14 above

1. Each E2 record must relate to
an E1 record for total participation but not every E1 has to relate to an E2 (for 3rd symbol (double line for total participation) from bottom of page 14).

2. Each E1 record can relate to up to
N records but each E2 record must relate to only 1 E1 record. (for 2nd symbol from bottom of page 14. Read from left to right, and then read again from right to left to improve on clarity for constraint that applies to each relation (the right or left relation)).

3. Each E1 record can relate to a minimum of 0 and a maximum of N E2 records. Each E2 record is related to 0 to 1 E1 records. Assume we are interpreting the E1--1-- R -- N—E2 relation above using the structural constraint of (min, max). (for last symbol from bottom of page 14 but using the 2nd but last E1/E2 symbol to describe the structural constraint (min, max). Read from left to right, and then read again from right to left to improve on clarity for constraint that applies to each relation (the right or left relation)). Note the (min, max) constraint that applies to the left relation is written on the opposite side of the left relation and same applies to the right relation.

The full ER design for the COMPANY database is given

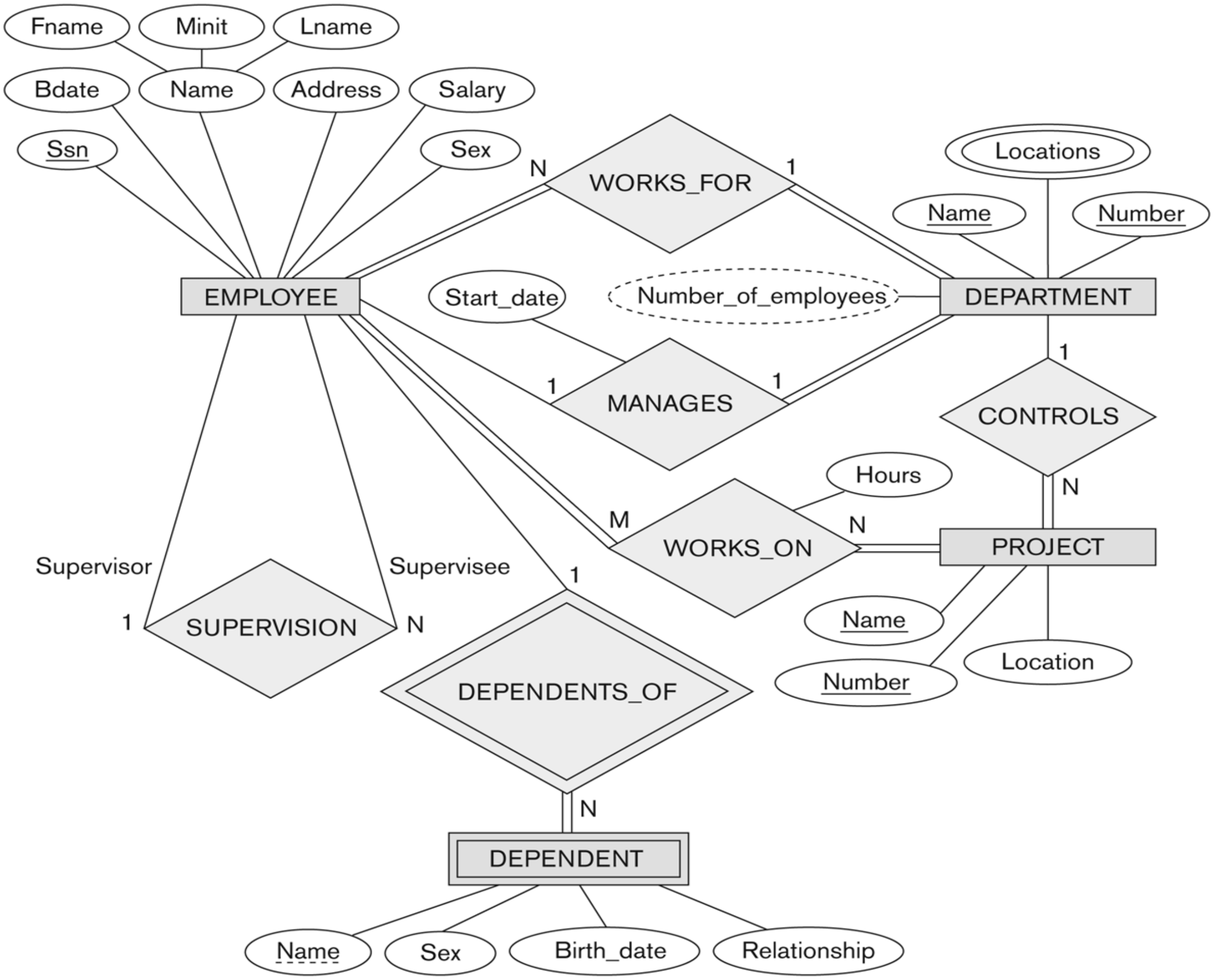
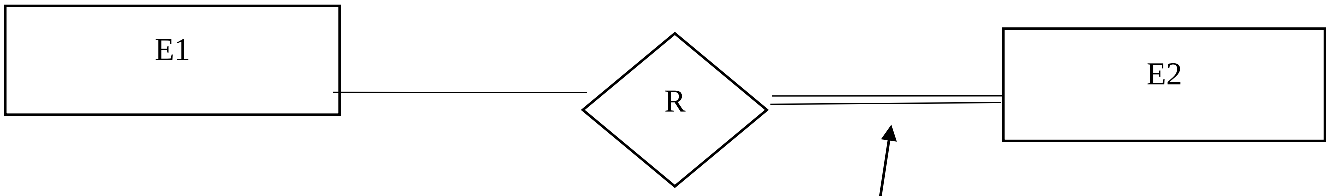


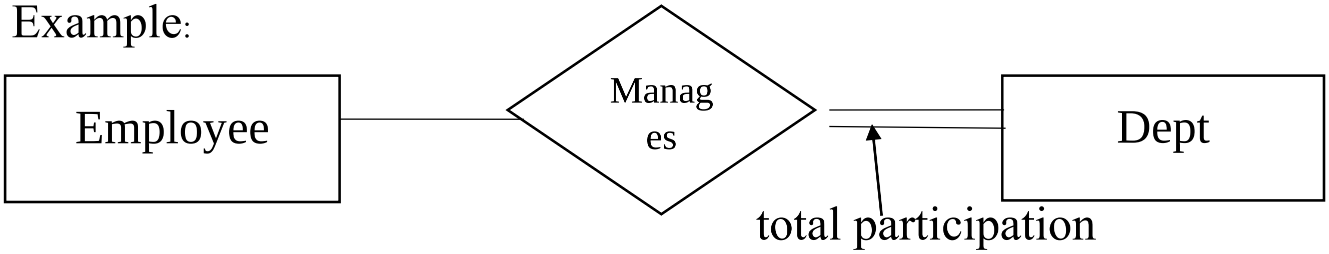
Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

More on Discussion of Ch 3
ER Diagrams on slide 19

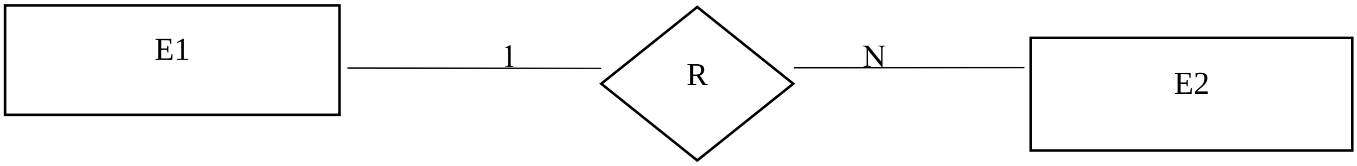


Means that every E2 record must relate to an E1 record through R
Shows total participation of entity E2 in relationship R.

Example:



1. For Total participation, Every dept is managed by an employee. (reading from right side entity to the left). (total)
2. However, not every employee is a manager of a dept. (reading from left to right side).

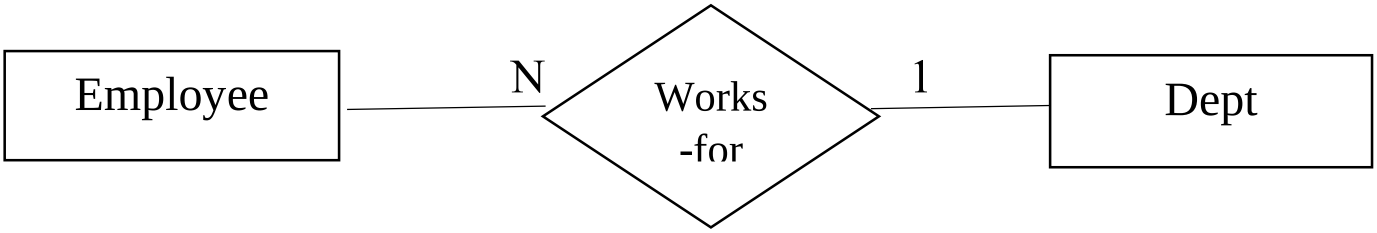


Shows Cardinality Ratio 1:N between entities E1:E2 through relationship R.
Meaning that:

1. Each E1 record is related to up to N records of E2. (reading from left side entity to right side entity).
2. Each E2 record is related to only 1 record of E1 through R. (reading from right side entity to left side).

ER Diagrams Slide 19 of Chapter 3 Notes.

Example. N:1 Relationship



Shows Cardinality Ratio

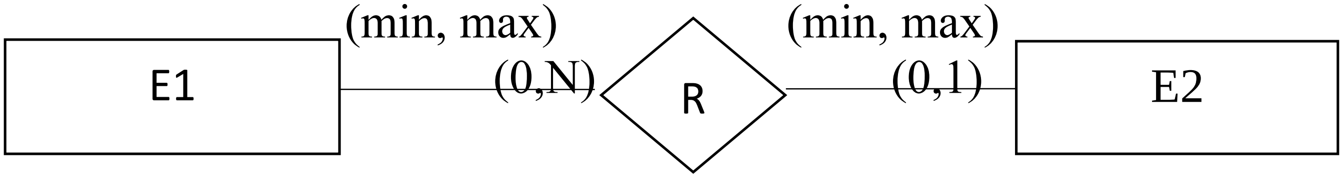
Meaning that :

1. Each employee record works for one dept. (reading from left side entity to right side entity).
2. However, each dept record has several employees (N) working for it. (reading from right side entity to left side).

An example database instance for this Employee-Works_for-Dept database relationship is (note there are 3 tables here):

Employee	Works_for		Dept
Ssn	Ssn	Deptid	Deptid
101	101	CS	CS
102	102	CS	Math
103	103	Math	Admin
104			

ER Diagrams on Slide 19. (Also, these alternative notations for structural constraints discussed on page 84 of course book).

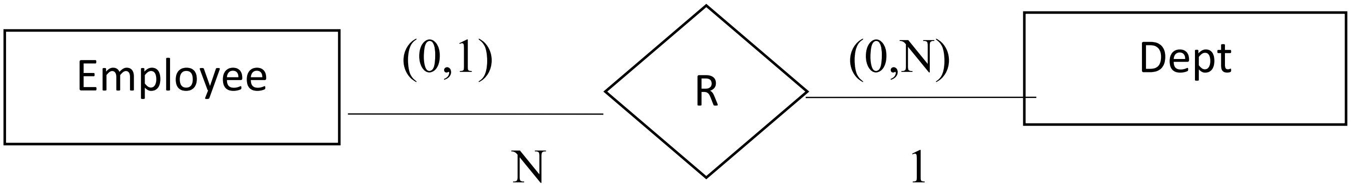


Shows structural constraints (min, max) on participation of E1 and E2 on R.

This means:

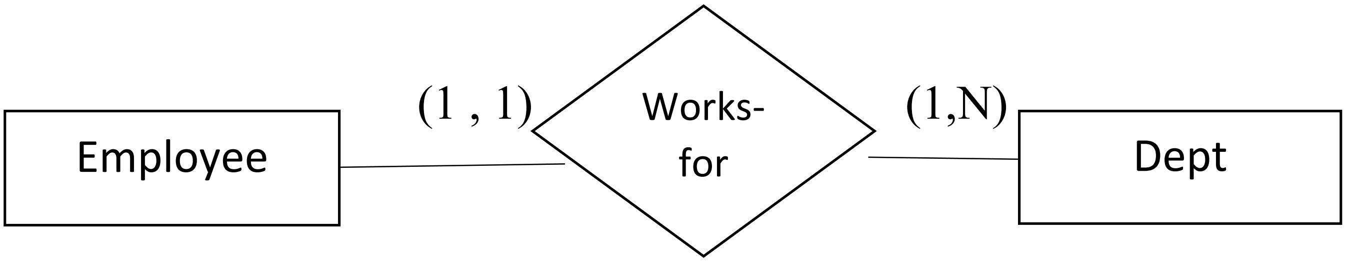
1. Each E1 record can relate to a minimum of 0 and a maximum of N of E2 records through R. (reading from left to right).
2. But each E2 record can relate to a minimum of 0 and a maximum of 1 records of E1 entity through R.

E.g.,



This means that each employee can work for 0 to 1 depts. However, each dept has 0 to N employees working for it.

To have total participation with both of these entities, we set the minimum to 1 as in:



Building the ER Diagram for the company DB.

From the requirements analysis, we define the entities with their attributes and relationships between the entities with their attributes.

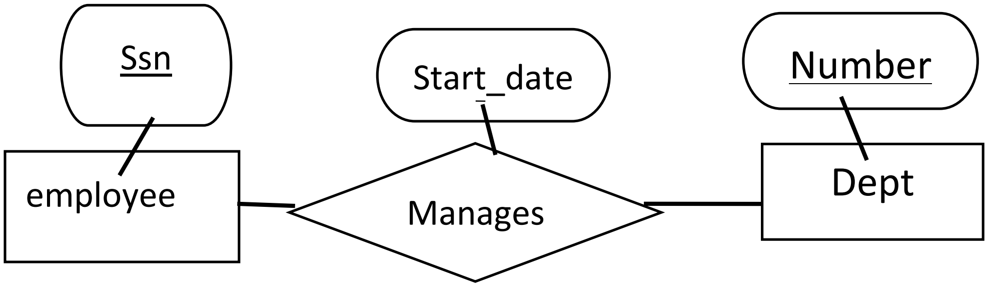
Entities:

1. DEPARTMENT (Name, Number, Locations, Manager, Manager_start_date)
   Note: Location is a multivalued attribute.
2. PROJECT (Name, Number, Location, Controlling_dept)
3. EMPLOYEE (Name (Fname, Minit, Lname), Ssn, Sex, Address, Salary, Birth_date, Dept, Supervisor, Works_on)
4. DEPENDENT (Employee, Dependent_name, Sex, Birth_date,

   Relationship)

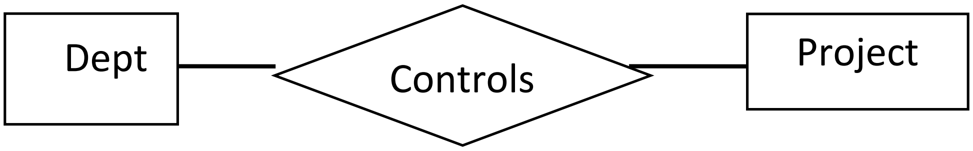
Relationships:

1. An Employee manages department.



Manages (Ssn, Dnumber, start_date)

2. Controls: Each Department controls some projects.
   Controls (Dnumber, PNumber)



3. (a) Works_for
   (b) Works_on

Each employee work for one Department but many work on several projects
Work_for(Ssn, Dnumber)
Work_on(Ssn, Pnumber, Hours)

4. Supervises:
Each employee has a supervisor.
Supervises(Ssn, eSsn)

5. Dependent_of:
Each employee may have some dependents.
Dependent_of (Ssn, Dependent_name)

Company Database Entity Relationship Model (Discussion on Refinement)

- Figure 3.8 on Ch. 3 slide notes, shows the preliminary design of the four entities types for company DB.
- In figure 3.8, some implicit relationship among entity types exist.
- This is when an attribute of one entity type refers to another entity type
- E.g. Attribute manager of Dept. refers to an employee who manages the Dept.
- In the ER model, these references should not be represented as attributes but relationships
- Thus, the initial company DB schema from fig. 3.8 is refined so that relationships are represented explicitly as relationships.
- The implicit relationships in Fig 3.8 refined into explicit relationships are:

1. Employee: Works_on Project (leads to removing the Works_on attribute previously in Employee during the initial design (see page 11 of this document, so it is now in a relationship type called Works_on).
2. Department: is managed by Employee.
3. Project : there is no implicit attribute to remove from Project
4. Dependent: is of Employee

- After removing the implicit relationships from the entity type listed on the left above, we replace them with explicit relationship to connect the left entity type to the right hand entity type.
- The final ER Diagram is Fig 3.2.
- On ER, Cardinality ratios for binary relationship type are 1:1, 1:N, N:1 and M:N.
- Two main types of binary relationship constraints are cardinality ration (e.g. 1:1) and participation (total or partial)
- An alternative notation for specifying structural constraints shown in Fig. 3.15 (Slide 39) can be used to both the cardinality ratio and single/double line notation for participation constraint. (see slides 37, 38, 39, 19 for details).