

Conceptual Database Design Using the Entity-Relationship (ER) Model

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 1
Lecture 7

Overview of Database Design

- **Conceptual design:** (ER Model is used for this.)
 - What are the **entities** and **relationships** we need?
- **Logical design:**
 - Transform ER design to Relational Schema
- **Schema Refinement:** (Normalization)
 - Check relational schema for redundancies and related anomalies.
- **Physical Database Design and Tuning:** *We'll do this later ...*
 - Consider typical workloads; (sometimes) modify the database design; select file types and indexes.

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 2
Lecture 7

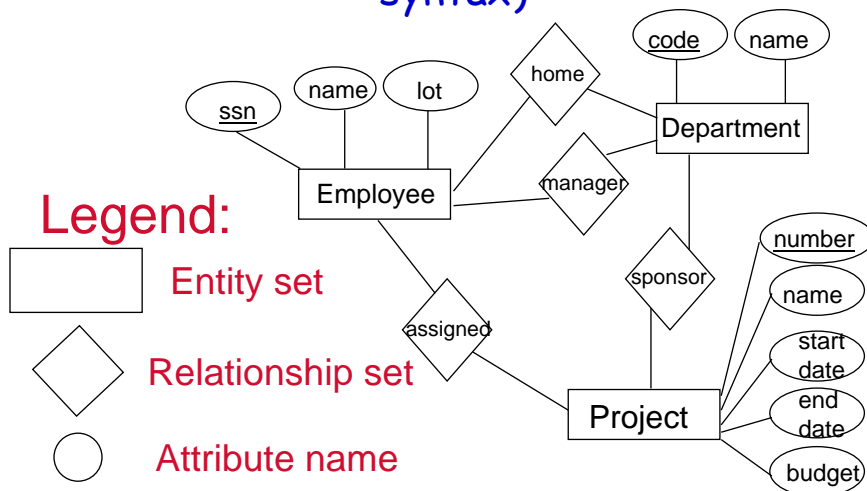
Entity-Relationship Model is a different model than the Relational Model

- **Relation model** has:
 - **tables** (relations) with attributes, keys, foreign keys, domain definitions for attributes
- **Entity-Relationship model** has:
 - **Entities and entity sets** with attributes, keys, and domain definitions for attributes
 - **relationships among entities and relationship sets** with cardinality constraints (in the book they refer to key constraints and participation constraints)

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 3
Lecture 7

Entity-Relationship Diagram (original syntax)



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 4
Lecture 7

Definitions

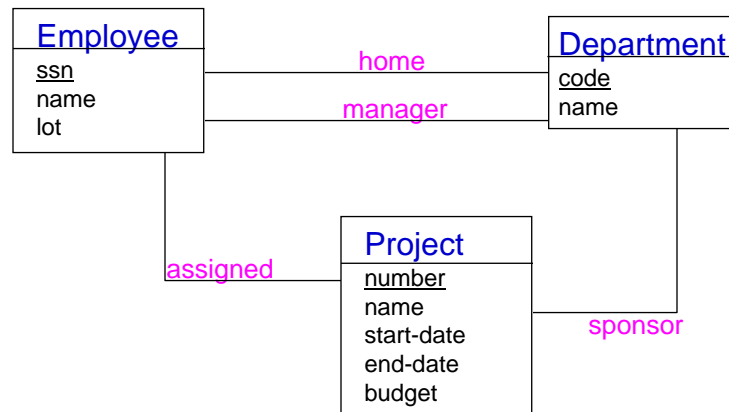
- *Entity*: Real-world object distinguishable from other objects.
 - An entity is described using a set of *attributes*.
- *Entity Set*: A collection of similar entities. E.g., all employees. (often referred to as just entity)

Definitions

- *Relationship*: Association among 2 or more entities. E.g., Attishoo's home department is Pharmacy.
- *Relationship Set*: Collection of similar relationships. E.g., Home. (often referred to as just relationship).

UML version of the same E-R Diagram

UML: Unified Modeling Language - for OO Design



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 7
Lecture 7

Example

- Professors have a SSN, a name and an age and their SSNs uniquely identify them.
- Professors teach courses. Each course is supervised by one professor.
- Courses are uniquely identified by their courseID, and they have a name.
- Draw the ERD.

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 8
Lecture 7

Practice

- Draw the UML version of the previous ERD.

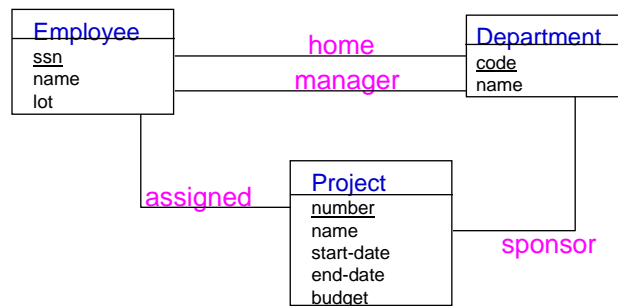
Equivalent Relational Schema

Employee (ssn, name, lot, home-dept)

Project-team(ssn, number)

Department (id, name, manager)

Project (number, name, start-date, end-date, budget, sponsor)



Equivalent Relational Schema - with foreign keys shown

Employee (ssn, name, lot, home-dept)

Project-team(ssn, number)

Department (id, name, manager)

Project (number, name, start-date, end-date, budget, sponsor)

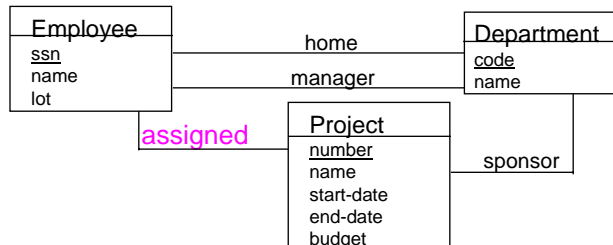
Notice that the many-to-many relationship set must be represented in a (new) table.

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

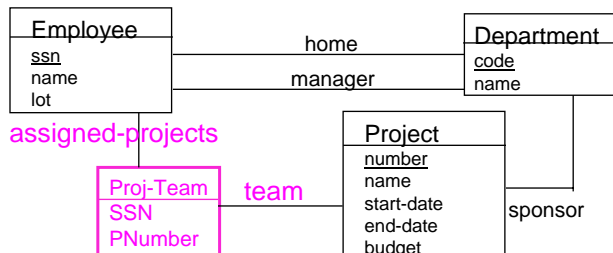
Slide 11
Lecture 7

Many-to-many relationship sets

ERD



Relational
DB Diagram



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 12
Lecture 7

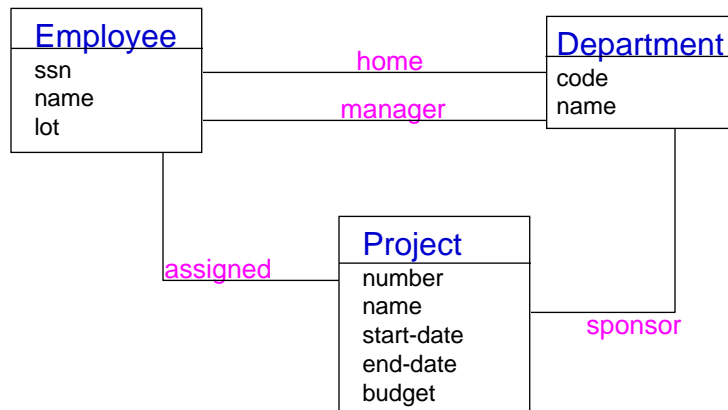
What data do we need to record a relationship?



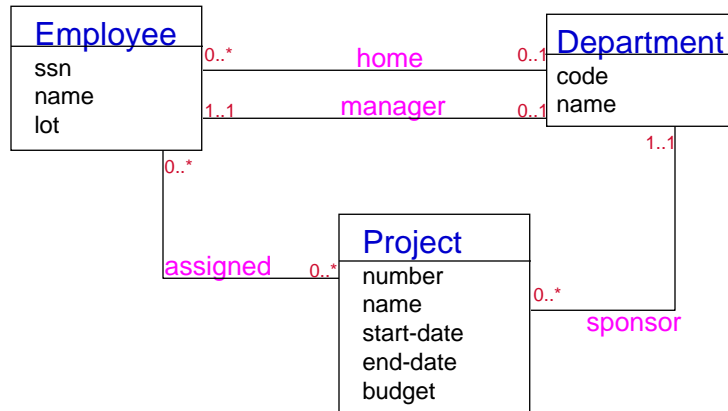
we must indicate which employee and which department we want to have connected (for each relationship).

we need the key value for an employee and the key value for the department – stored together – to represent the relationship.

Cardinality Constraints on Relationship sets: How many entities can participate?

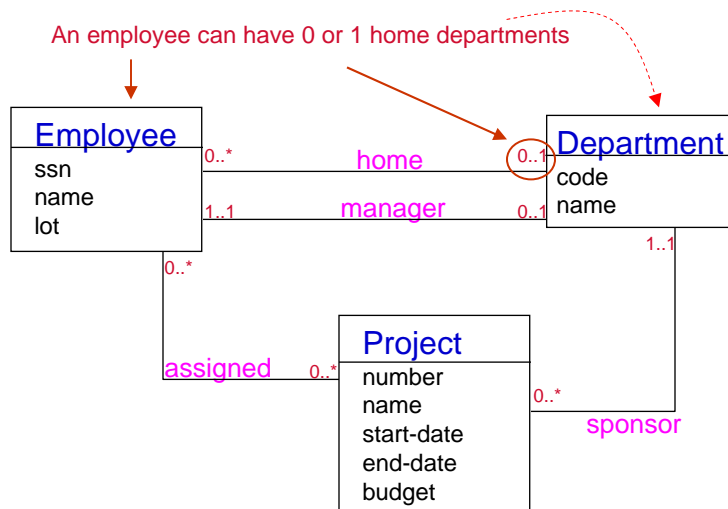


Cardinality Constraints on Relationship sets How many entities can participate?



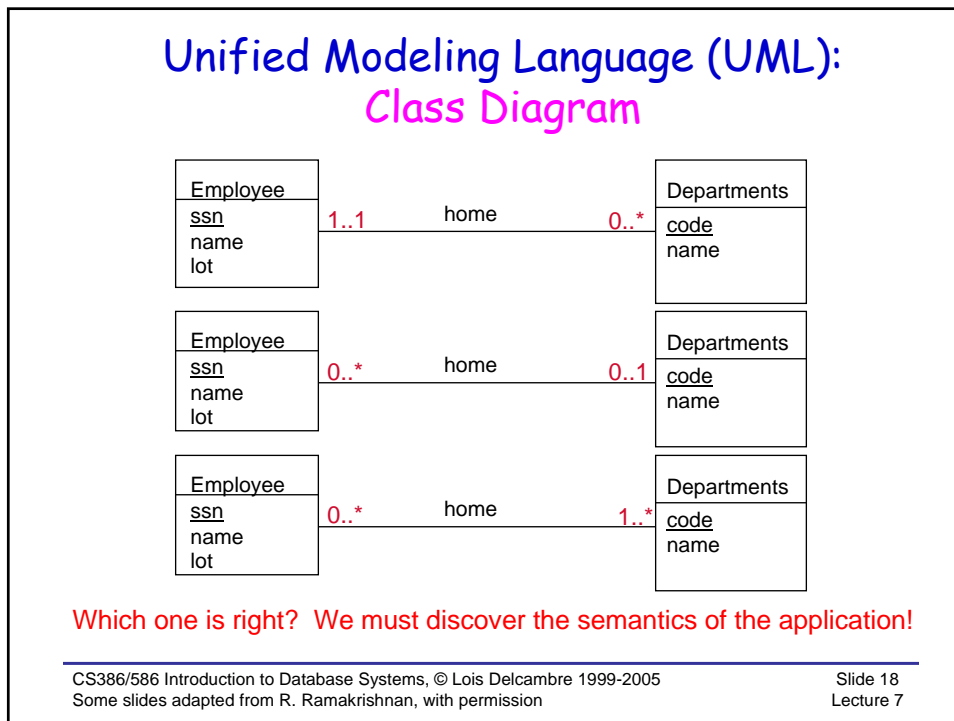
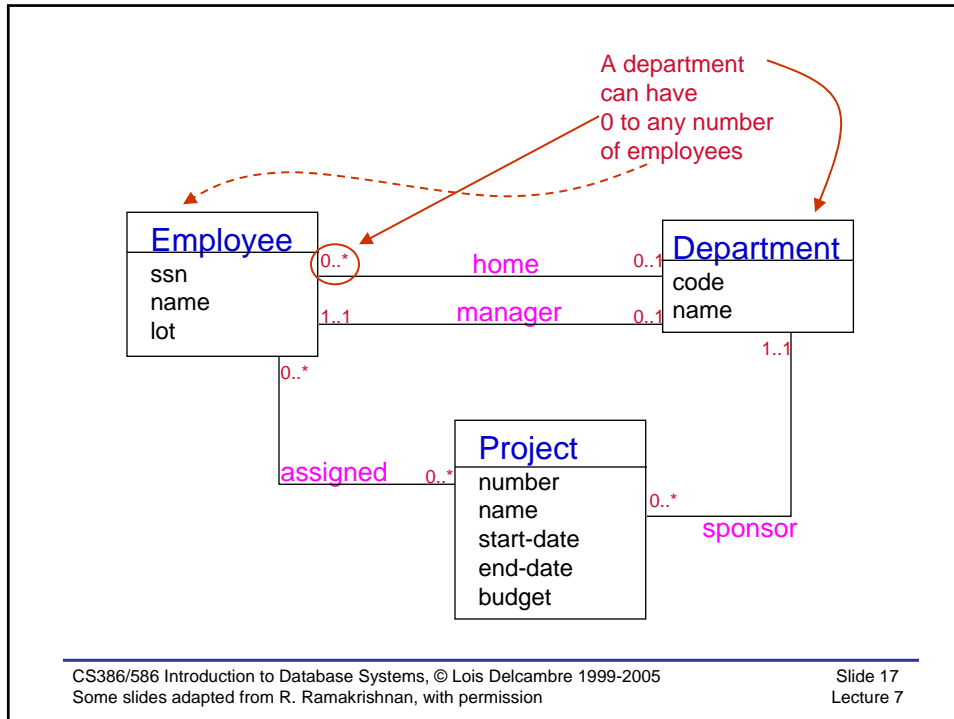
CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 15
Lecture 7

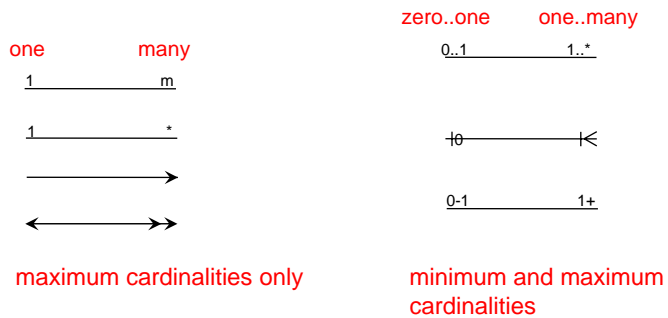


CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

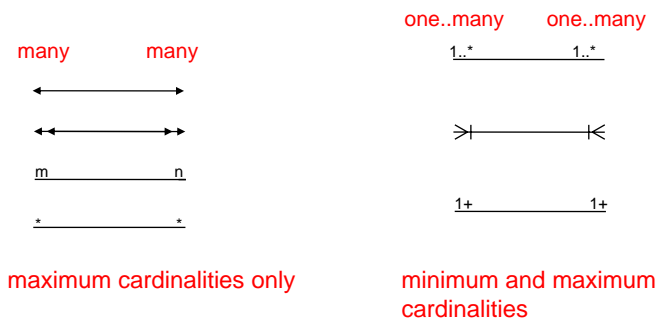
Slide 16
Lecture 7



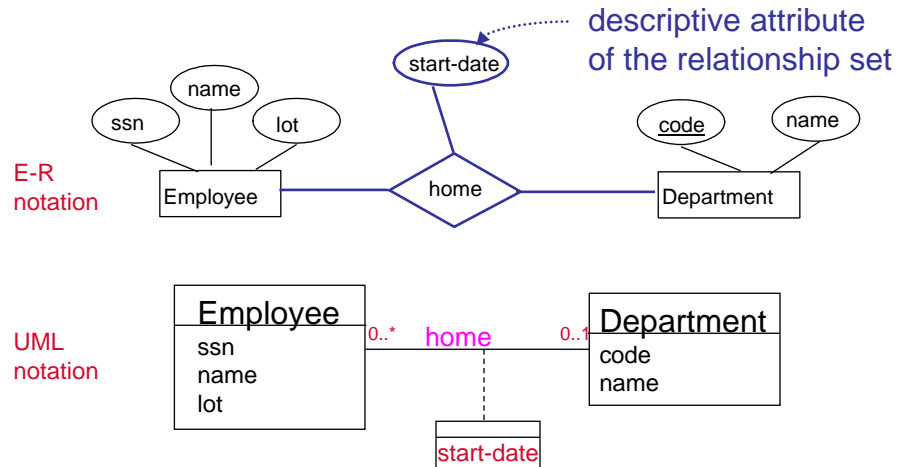
Various notation for "one-to-many"



Various notations for "many-to-many"



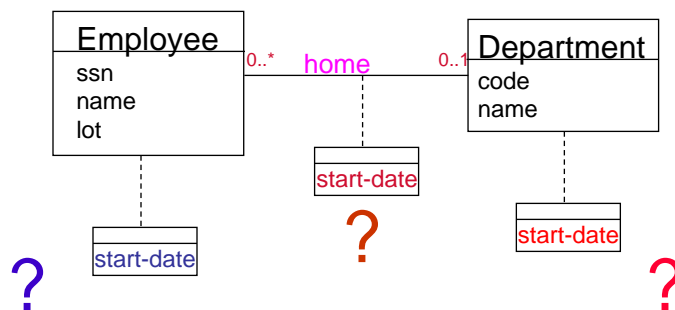
Relationship sets can have attributes



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 21
Lecture 7

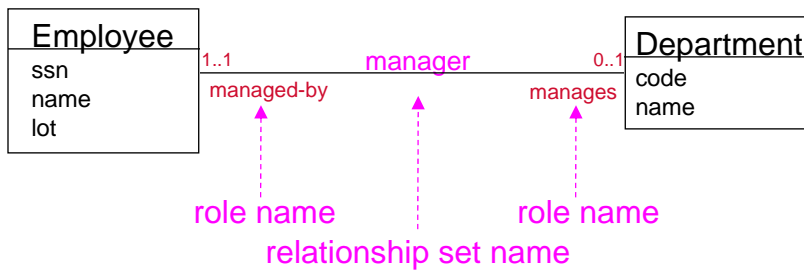
Try all three locations for the attributes:
which one makes sense?



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 22
Lecture 7

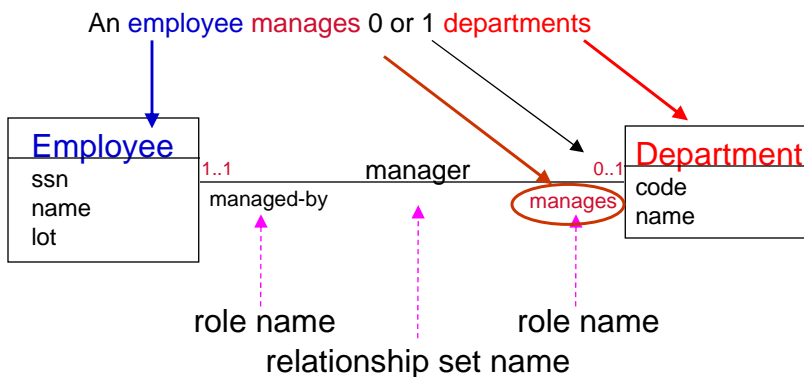
Relationship sets can have **role** names (in addition to the name of the relationship set)



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 23
Lecture 7

Example: reading **role** names

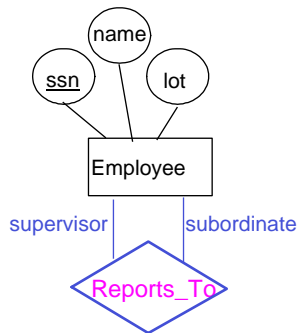


CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

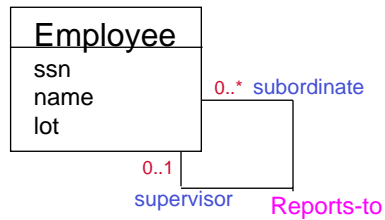
Slide 24
Lecture 7

Same entity sets can participate in different "roles" for the same relationship set

E-R notation



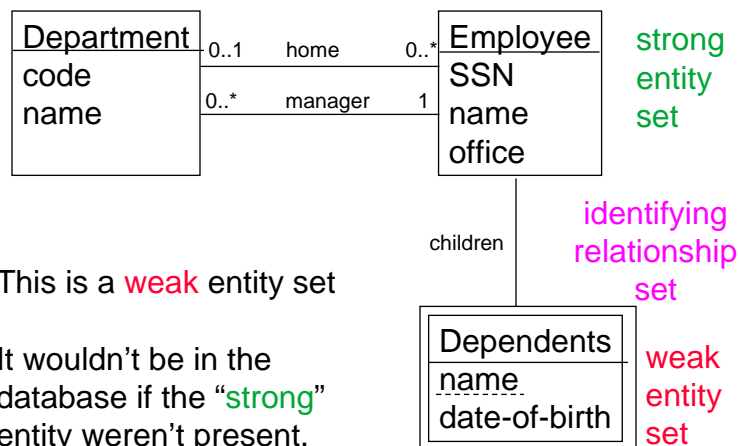
UML notation



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 25
Lecture 7

Weak Entity Sets (and Identifying Relationship sets)



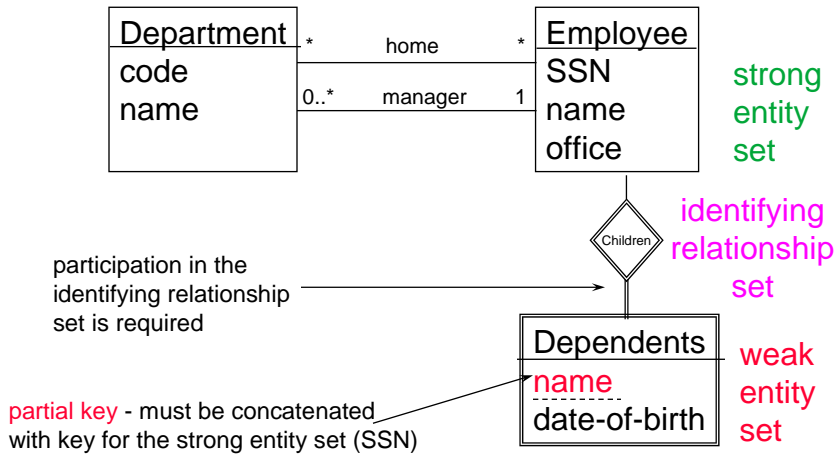
This is a **weak** entity set

It wouldn't be in the database if the "strong" entity weren't present.

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 26
Lecture 7

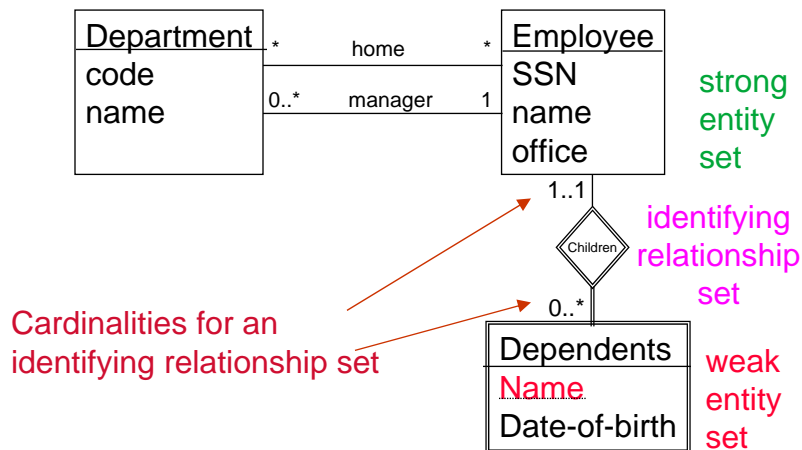
Weak Entity Sets and Identifying Relationship sets: Alternative Notation



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 27
Lecture 7

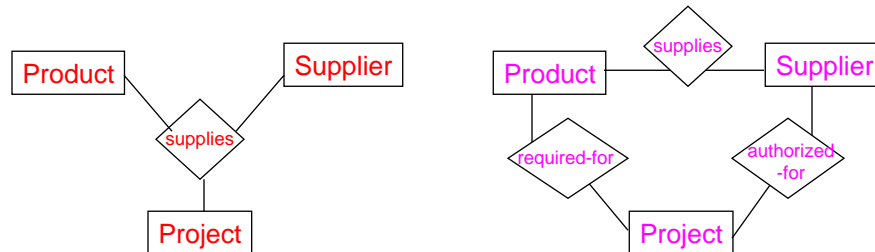
Weak Entity Sets and Identifying Relationship sets: Alternative Notation (cont.)



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 28
Lecture 7

Ternary vs. Binary Relationship sets

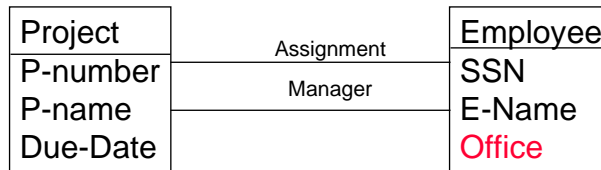


These two schemas are not equivalent!
When would we use a ternary relationship set?
When would we use three binary relationship sets?

Binary vs. Ternary Relationship sets (Cont.)

- The ternary relationship set means that a Supplier must be authorized to supply a particular part to a particular project. e.g., **Office-Depot can supply laser printer paper to project 112.** **Office-Max can supply paper clips to Project 112.** **Office-Max can supply pencils to project 115.** (But based on that much information, **Office-Max can't supply pencils to 112.**)
- The three binary relationship sets each represent something distinct. A Supplier can be authorized to supply certain products (Office-Max can supply pencils). A Project can require certain products (112 needs pencils). And a Supplier can be authorized to supply a certain project. (Office-Max supplies 112)
Therefore: **Office-Max can supply pencils to 112.**

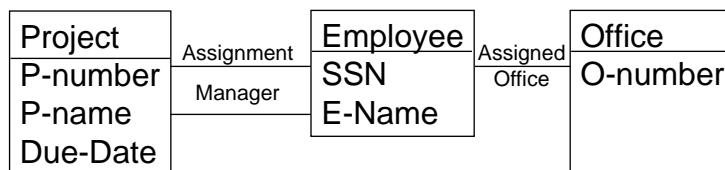
Duality: **entity** ↔ **value**
 and **attribute** ↔ **relationship**



Should **Office** be an **attribute of Employee**? or a **separate entity set**? Most attributes can be “promoted” to an entity set and some entities can be “demoted” to an attribute value.

This explains why there are so many different ways to design a schema.

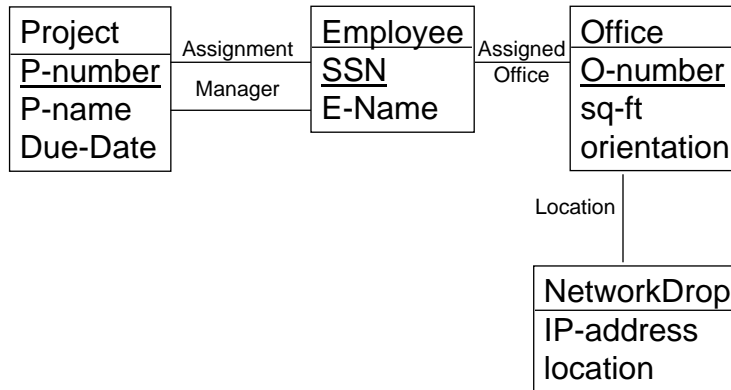
Entity vs. Value of an Attribute



What are some reasons to model Office as an entity set?

- an employee can have more than one office
- there are other attributes of Office
- Office needs to participate in other relationship sets such as a relationship set connecting to furniture or telephones or network drops (located in the office)

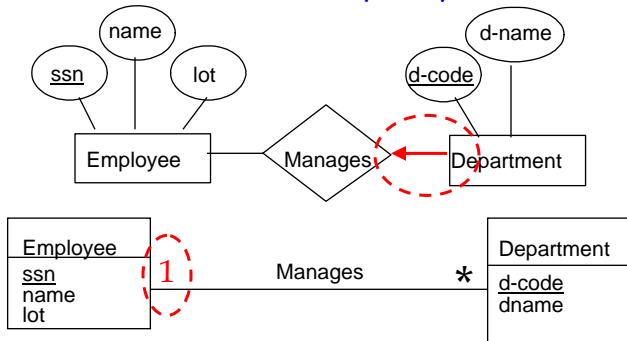
Entity vs. Value of an Attribute



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 33
Lecture 7

Key Constraints - as described in the text
(limiting participation in relationship set to
at most 1 entity)
same as maximum multiplicity of 1 in UML



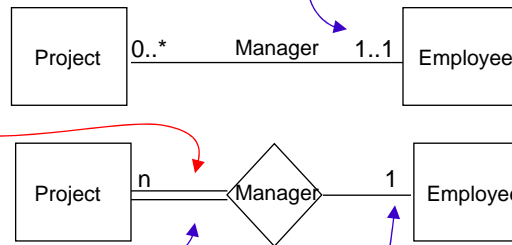
Each dept has at most one manager, according to the
key constraint on Manages.

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 34
Lecture 7

Participation Constraint - as in text: when every entity **MUST** participate in a relationship set

a Project has exactly one manager



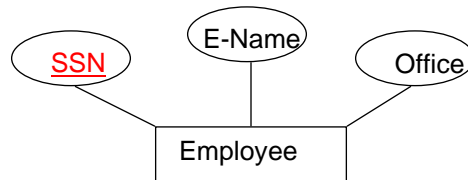
**a Project MUST have a manager
and there is at most 1 employee who is manager**

Translating an ER Diagram to a Relational Schema

1. Translate each entity set into a table, with keys.

- **Entity set:**

- can be represented as a table in the relational model
- has a **key** ... which becomes a key for the table



```
CREATE TABLE Employee  
(SSN CHAR(11),  
E-Name CHAR(20),  
Office INTEGER,  
PRIMARY KEY (SSN))
```

A relational DBMS does not allow multi-valued attributes. If yours doesn't,

2. Create a table for the multi-valued attribute.

How many offices can one employee have?

Just one

Project (P-number, P-name, Due-Date)
Employee (SSN, E-Name, Office)

vs.

More than one

Project (P-number, P-name, Due-Date)
Employee (SSN, E-Name)
Office-Assignment (SSN, Office)

Sample Data

Project (P-number, P-name, Due-Date)

Employee (SSN, E-Name, Office)

Just one

12 Smith O-105

15 Wei O-110

20 Jones O-112

Project (P-number, P-name, Due-Date)

Employee (SSN, E-Name)

More than one

12 Smith

15 Wei

Office-Assignment (SSN, Office)

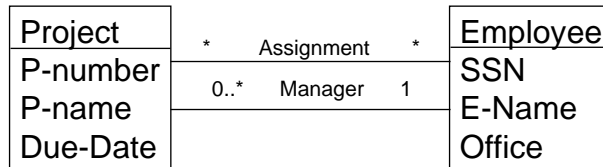
12 O-105

12 O-106

15 O-110

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 39
Lecture 7



2. Translate each **many to many** relationship set into a table

What are the attributes and what is the key for Assignment?

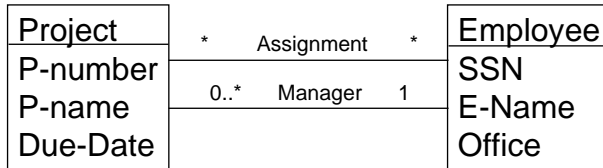
Assignment (?)

Project (P-number, P-name, Due-Date)

Employee (SSN, E-Name, Office)

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 40
Lecture 7



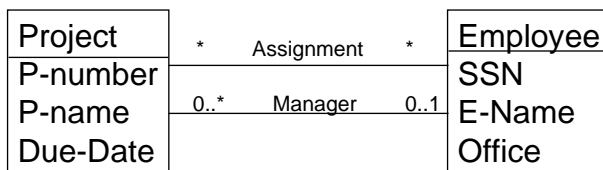
Answer: Assignment (P-Number, SSN)

P-Number is a foreign key for Project

SSN is a foreign key for Employee

Project (P-Number, P-Name, Due-Date)

Employee (SSN, E-Name, Office)

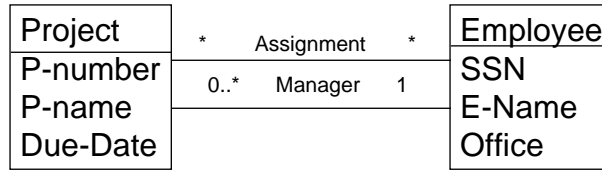


What should we do with each **one to many** relationship set?

Manager (?)

Project (P-number, P-name, Due-Date)

Employee (SSN, E-Name, Office)

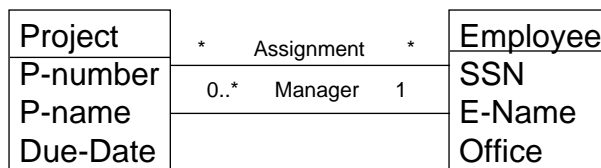


Project (P-number, P-name, Due-Date, **Manager**)
 Employee (SSN, E-Name, Office)

3. Create a foreign key for a 1-to-many relationship set.

Manager is a foreign key (referencing the Employee relation)

value of Manager must match an SSN



Project (P-number, P-name, Due-Date, **Manager**)
 Employee (SSN, E-Name, Office)

vs.

3. Or...Create a table for a 1-many relationship set.

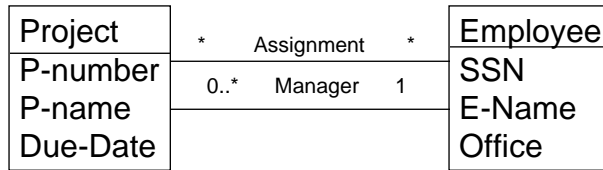
Project (P-number, P-name, Due-Date)

Employee (SSN, E-Name, Office)

Manager (P-number, SSN)

What are the tradeoffs between these two?

Note:
 P-number
 is the key
 for Manager

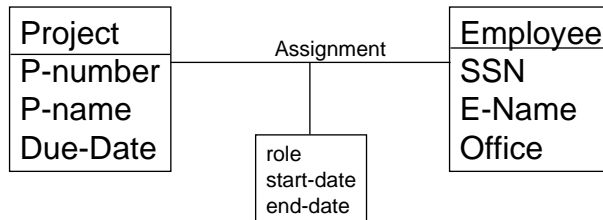


Project (P-number, P-name, Due-Date)
 Employee (SSN, E-Name, Office, Managed-project)

vs.

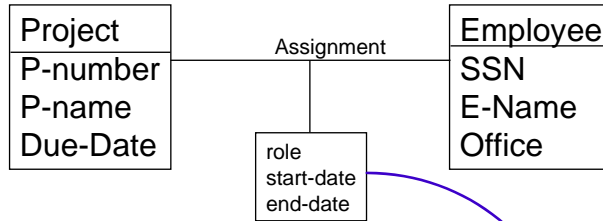
Project (P-number, P-name, Due-Date)
 Employee (SSN, E-Name, Office)
 Manager (P-number, SSN)

What if SSN is the key for Manager?



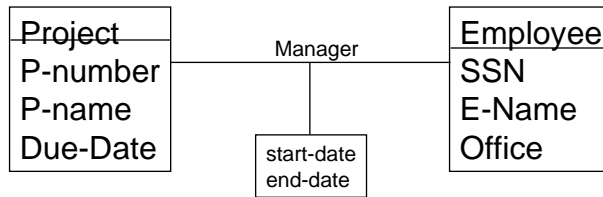
What do we do when a many-to-many relationship set has an attribute?

Assignment (A-project, A-SSN)
 Project (P-number, P-name, Due-Date)
 Employee (SSN, E-Name, Office)



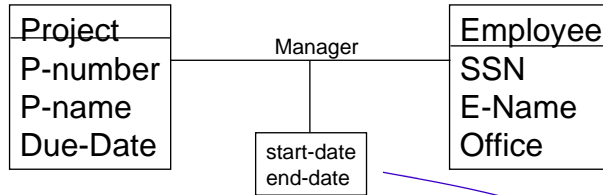
What do we do when a many-to-many relationship set has an attribute?

Assignment (A-project, A-SSN, role, start-date, end-date)
 Project (P-number, P-name, Due-Date)
 Employee (SSN, E-Name, Office)



What do we do when a 1-to-many relationship set has an attribute?

Project (P-number, P-name, Due-Date, Manager)
 Employee (SSN, E-Name, Office)



What do we do when a 1-to-many relationship set has an attribute?

Project (P-number, P-name, Due-Date, Manager, start-date, end-date)
 Employee (SSN, E-Name, Office)

Participation Constraints in SQL

- We can require any table to be in a binary relationship using a foreign key which is required to be NOT NULL (but little else without resorting to CHECK constraints)

```

CREATE TABLE Department (
  d-code          INTEGER,
  d-name          CHAR(20),
  manager-ssn    CHAR(9) NOT NULL,
  since          DATE,
  PRIMARY KEY (d-code),
  FOREIGN KEY (manager-ssn) REFERENCES Employee,
  ON DELETE NO ACTION)
  
```

Translating Weak Entity Sets

- Weak entity sets and identifying relationship sets are translated into a single table. Must include key of strong entity set, as a foreign key.
- When the owner entity is deleted, all owned weak entities must also be deleted.

```
CREATE TABLE Insurance_Policy (  
  dep-name CHAR(20),  
  age INTEGER,  
  cost REAL,  
  ssn CHAR(11) NOT NULL,  
  PRIMARY KEY (dep-name, ssn),  
  
  FOREIGN KEY (ssn) REFERENCES Employee,  
  ON DELETE CASCADE)
```

Note ERDs and UML Diagrams can be at two levels:

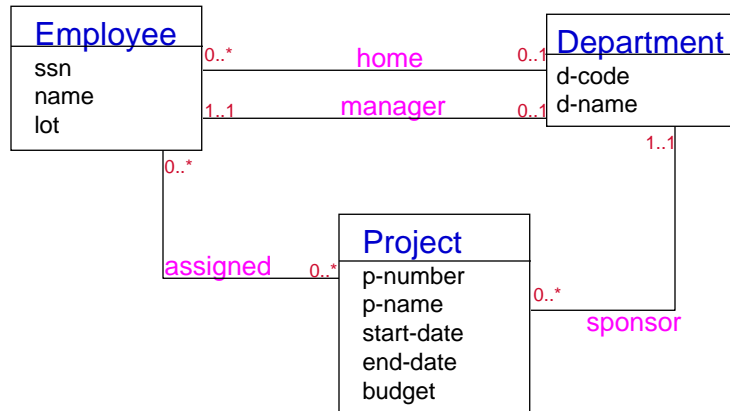
the ERD/UML level

and

the Relational Table level.

The difference is primarily with the many-to-many relationship sets.

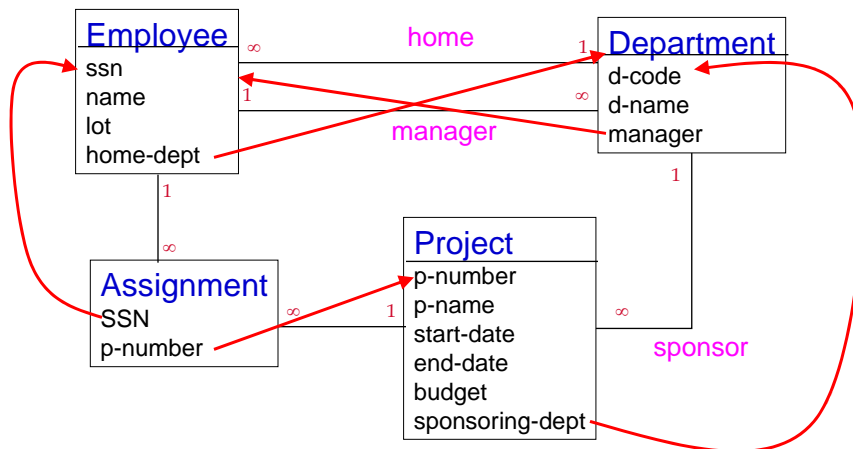
Entity-Relationship Diagram



CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 53
Lecture 7

Equivalent Relational Schema



Notice that the relationship sets shown in this diagram aren't really needed. *foreign keys* reference other tables.

CS386/586 Introduction to Database Systems, © Lois Delcambre 1999-2005
Some slides adapted from R. Ramakrishnan, with permission

Slide 54
Lecture 7

Summary of Translation Steps: ER to Tables

1. Create table and choose key for each entity set; include single-valued attributes.
2. Create table for each weak entity set; include single-valued attributes. Include key of owner as a foreign key in the weak entity. Set key as foreign key of owner plus local, partial key.
3. For each 1:1 relationship set, add a foreign key to one of the entity sets involved in the relationship (a foreign key to the other entity in the relationship)*.
4. For each 1:N relationship set, add a foreign key to the entity set on the N-side of the relationship (to reference the entity set on the 1-side of the relationship)*.
5. For each M:N relationship set, create a new table. Include a foreign key for each participant entity set, in the relationship set. The key for the new table is the set of all such foreign keys.
6. For each multi-valued attribute, construct a separate table. Repeat the key for the entity in this new table. It will serve as both the key for this table as well as a foreign key to the original table for the entity.

* Unless relationship set has attributes. If it does, create a new table for the relationship set.

This algorithm from Elmasri/Navathe, p. 174.