Introduction to Data Modeling (ER)

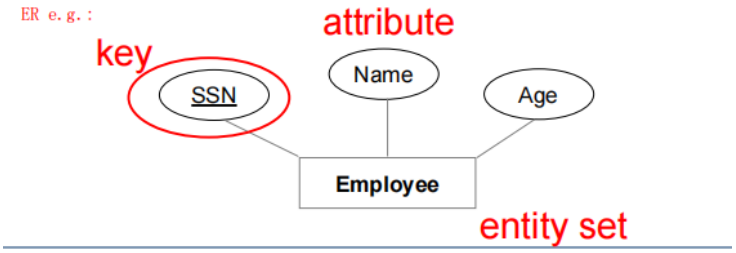
Basic ER modeling concepts

Entity

real-world object; described with a set of attributes.

Entity Set

A collection of entities of the same type, or same set of attributes (all employees); each entity has a key; Often used interchangeably with Entity.

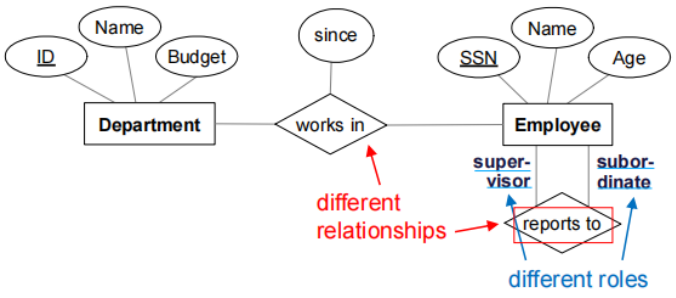


Relationship

association among two or more entities; can have their own attributes.

Relationship Set

Collection of relationships of the same type; same entity can participate in different relationship sets or even different roles in the same set.

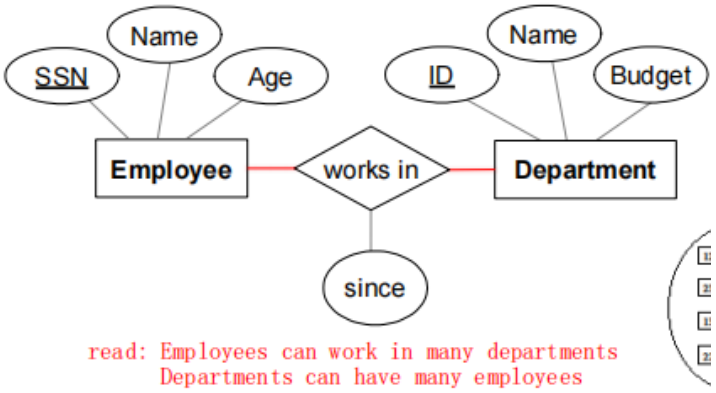


Constraints

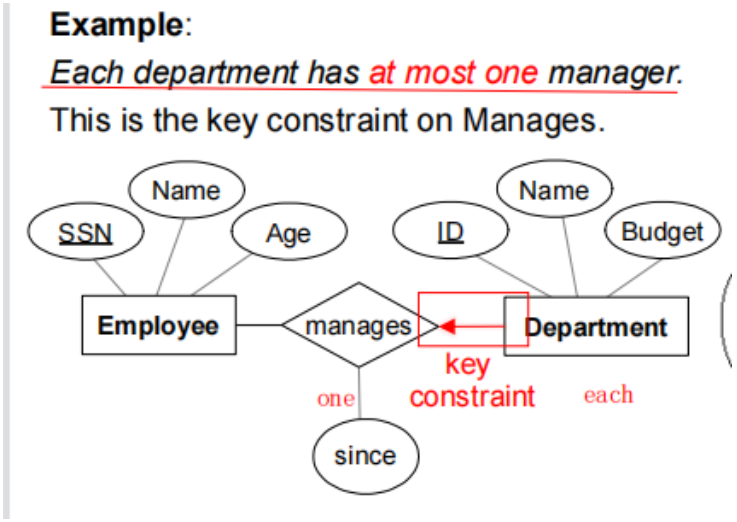
Key constraints

determine the number of objects taking part in the relationship set. (也就是映射关系)

1. Many-to-many (Line-Line)



1. One-to-Many (Line-Arrow) (Many-to-one; Arrow-Line): 其中某个set的entity只能和一个entity对应（单射）; **each -> at most one rl**



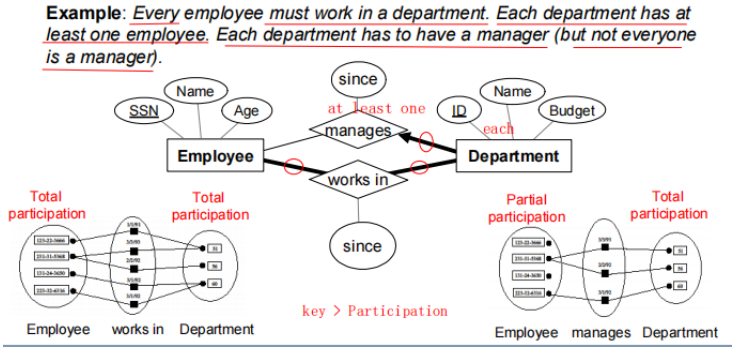
1. One-to-one (Arrow-Arrow) 两边都是箭头

Participation Constraints

Total Participation: all entities of one set take part in at least one relationship (well-defined function); **each = at least one rl**

Partial Participation: exsist instances have no relationship

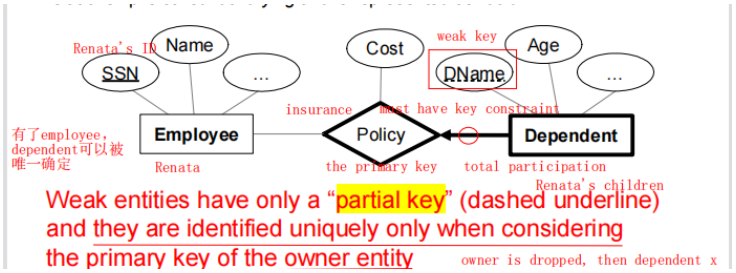
在SQL里是not null与否的设置



Weak Entities

can be identified uniquely only by considering the primary key of another owner entity. 用粗正方形表示; 也就是一个entity，只有一个relation，并且划了粗箭头；**each => at least and at most one rl**

Partial key: the key of weak entity; dashed underline ---

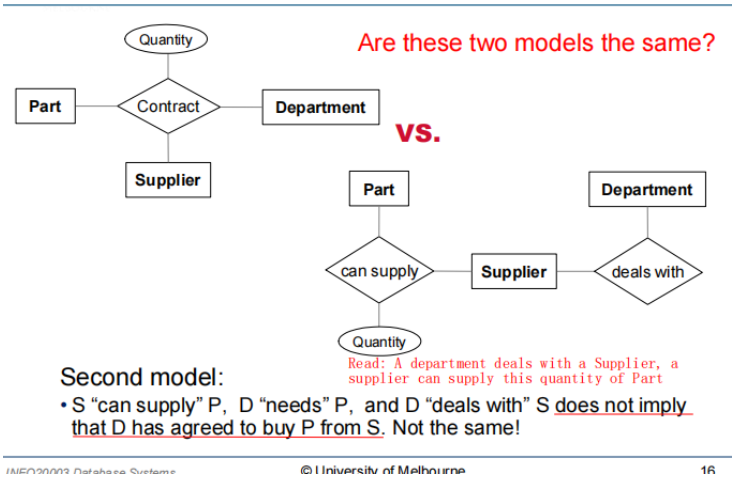


在ER图中，weak entity 中存在一个foreign key.

Ternary Relationships

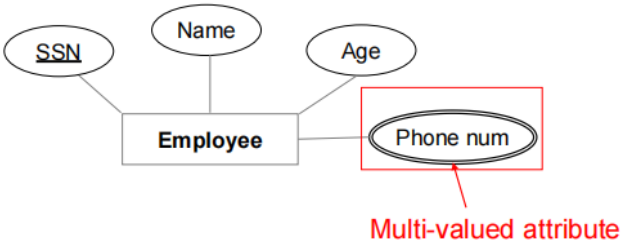
generalization: n-ary relationships; larger n-ary relationships can't be represented by smaller n-ary.

e.g

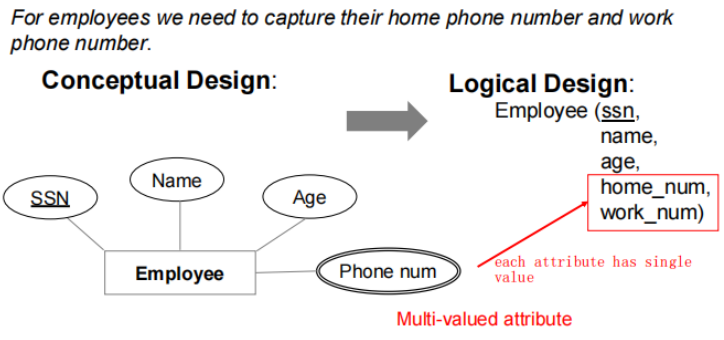


Special attribute type: Multi-valued attributes

These attributes can have multiple values of the same type. in double circle

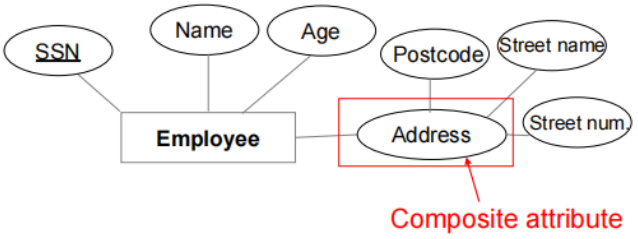


在logical design时需要把重复项全部展开写 / create a lookup table

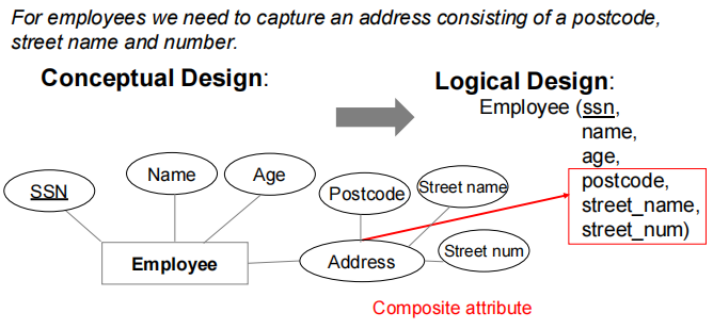


Special attribute type: Composite attributes

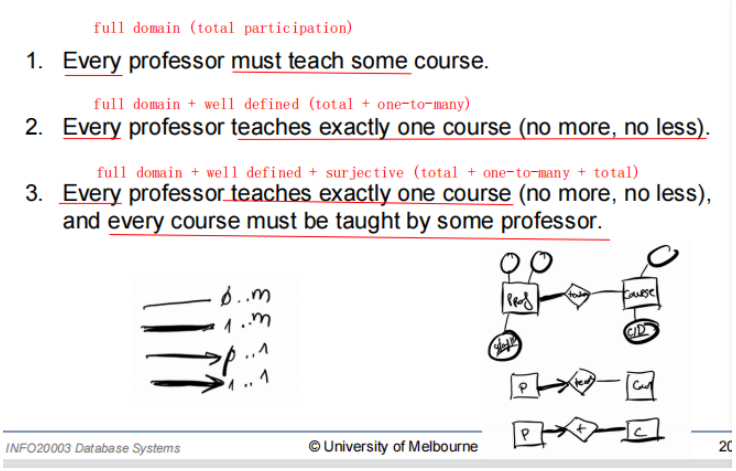
Attributes that have a structure hidden inside, each element can be of different type.



在logical design时需要把子项全部展开（即将address单独列为一个entity set; 或者单纯地展开为几个attribute）



An expample



Conceptual Design

follows requirements analysis

ER model is popular, subjective; no standard notation; above is Chen's notation

Basic constructs: entities, relationships, attributes, key & participation constraints, weak entities

Translating ER diagrams

Relational Model

Data Model

allows us to translate real world things into structures that a computer can store.

e.g.: Relational, ER, Network, Hierarchical, etc

Relational Model (Table)

Rows (Tuples/records) & Columns (Attributes/fields)

Keys & Foreign Keys to link relations

Relational Database (a set of relations)

Relation

1. Schema: specifies name of relation, plus name and type of each column (attribute)

e.g.: Students (sid:string, name:string, login:sting, age:integer, gpa:real)

1. Instance: a table, with rows and columns

No. of rows = cardinality; No. of fields = degree (arity)

relation <=> a set of rows or tuples; rows are **distinct**, no order among rows.

Keys & Integrity Constraints

Keys

a way to associate tuples in different relations; one form of **Integrity constraint (IC)**

Primary Keys

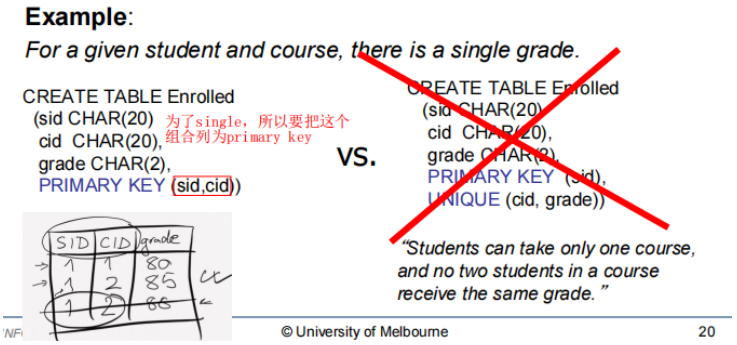
Superkey: no two distinct tuples can have same values in all key fields (数据库中一组不重复的atribute集合)

key: minimal subset of a superkey (最小无重复fields集合)

primary key: one of the keys that were chosen

Candidate keys: key (minimal subset of a superkey), except for primary key.

Each relation has a primary key.



Foreign key

a set of fields in one relation that is used to refer to a tuple in another relation; must correspond to the primary key of the other relation.

Example: *Only students listed in the Students relation should be allowed to enroll in courses.*

- *sid* is a foreign key referring to Students

```
CREATE TABLE Enrolled
(sid CHAR(20),
 cid CHAR(20),
 grade CHAR(2),
 PRIMARY KEY (sid,cid),
 FOREIGN KEY (sid) REFERENCES Students )
```

Enrolled

sid	cid	grade
53666	15-101	C
53666	18-203	B
53650	15-112	A
53666	15-105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@cs	18	3.2
53650	Smith	smith@math	19	3.8

Referential integrity

all foreign key constraints are enforced in a DBMS

e.g.: what should be done if a student tuple is deleted? or primary key of Students tuple is updated?

1. delete all enrolled tuples that refer to it?
2. disallow deletion of a student tuple that is referred to?
3. set sid in enrolled tuples that refer to it to a default sid? (often "null", or unknown, inapplicable in SQL)

Integrity Constraints (ICs)

IC: condition that must be true for any instance of the database (e.g. domain constraints)

specified when define schema; checked when relations are modified.

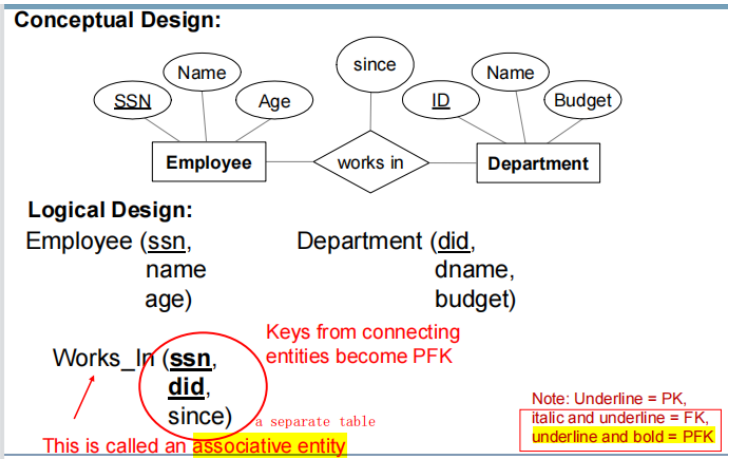
legal instance: a relation satisfying all specified ICs.

Translating ER to Logical and Physical Model

ER -> Logical: Many to Many (key constraint)

attributes of a new relation must include:

1. keys for each participating entity set (as foreign keys), which will form a superkey of the relation
2. all descriptive attributes



to physical design

Logical Design:

Employee (ssn, name, age)
Department (did, dname, budget)
Works_In (**ssn**, **did**, since)

Note: Underline = PK, italic and underline = FK, underline and bold = PFK

Physical Design:

```
Employee
(ssn CHAR(11),
name VARCHAR(20),
age INTEGER)

Department
(did INTEGER,
 dname VARCHAR(20),
 budget FLOAT)

Works_In(
ssn CHAR(11),
did INTEGER,
since DATE)
```

to implementation (create table)

Logical Design:

Employee (ssn, name, age)
Department (did, dname, budget)
Works_In (**ssn**, **did**, since)

Note: Underline = PK, italic and underline = FK, underline and bold = PFK

Implementation:

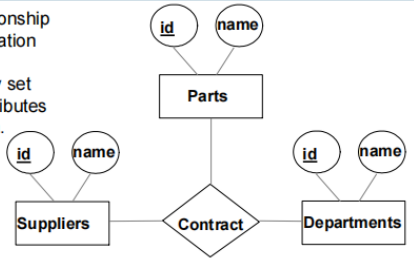
```
CREATE TABLE Employee
        (ssn CHAR(11),
         name VARCHAR(20),
         age INTEGER,
         PRIMARY KEY  (ssn))

CREATE TABLE Department
        (did INTEGER,
         dname VARCHAR(20),
         budget FLOAT,
         PRIMARY KEY  (did))

CREATE TABLE Works_In
    (ssn  CHAR(11),
     did  INTEGER,
     since  DATE,
     PRIMARY KEY (ssn, did),
     FOREIGN KEY (ssn) REFERENCES Employee,
     FOREIGN KEY (did) REFERENCES Department)
```

ER -> Logical: Ternary relationship

same as binary, 仅仅是多加一个foreign key; n-ary就加n个 foreign keys

In translating a many-to-many relationship set to a relation, attributes of the relation must include:

- Keys for each participating entity set (as foreign keys). This set of attributes forms a *superkey* for the relation.
- All descriptive attributes.



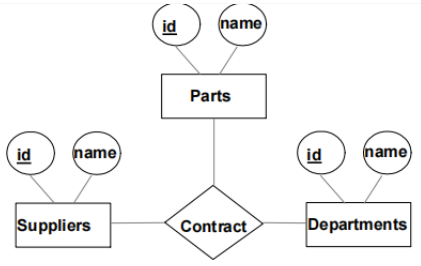
Logical Design:

Contracts (
supplier_id,
part_id,
department_id)

Note: Underline = PK, italic and underline = FK, underline and bold = PFK

Logical Design:

Contracts (
supplier_id,
part_id,
department_id)



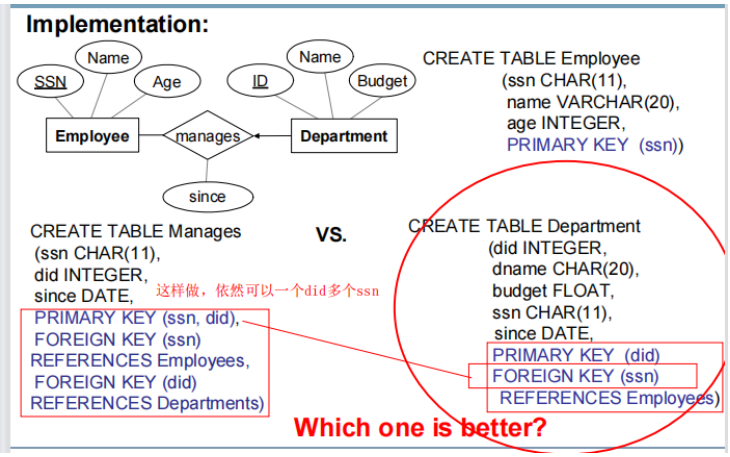
Implementation:

```
CREATE TABLE Contracts (
    supplier_id  INTEGER,
    part_id  INTEGER,
    department_id  INTEGER,
    PRIMARY KEY (supplier_id, part_id, department_id),
    FOREIGN KEY (supplier_id) REFERENCES Suppliers,
    FOREIGN KEY (part_id) REFERENCES Parts,
    FOREIGN KEY (department_id) REFERENCES Departments)
```

和binary没啥区别

ER -> Logical: one to Many (key constraint)

Rule: Primary key from the many side becomes a foreign key on the one side, to ensure "at most one"



Participation Constraints in SQL

- We specify total participation with key words NOT NULL
  - **NOT NULL = this field cannot be empty**

```
CREATE TABLE Department (
  did  INTEGER NOT NULL,
  dname  CHAR(20) NOT NULL,
  budget  FLOAT NULL,
  ssn  CHAR(11) NOT NULL,        Total
  since  DATE NULL,
  PRIMARY KEY  (did),
  FOREIGN KEY  (ssn) REFERENCES Employee
    ON DELETE NO ACTION)
```

ER: Weak Entities

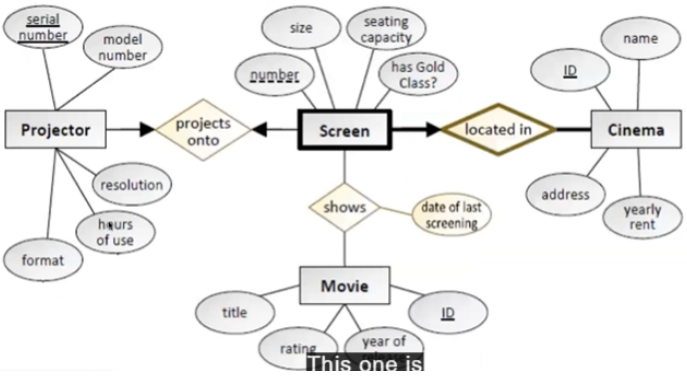
are translated into a single table; when the owner entity is deleted, all owned weak entities must also be deleted.

在SQL中即在PRIMARY KEY里面加上owner entity的FOREIGN KEY，同时FOREIGN KEY里面也要保持

Logical Design Summary

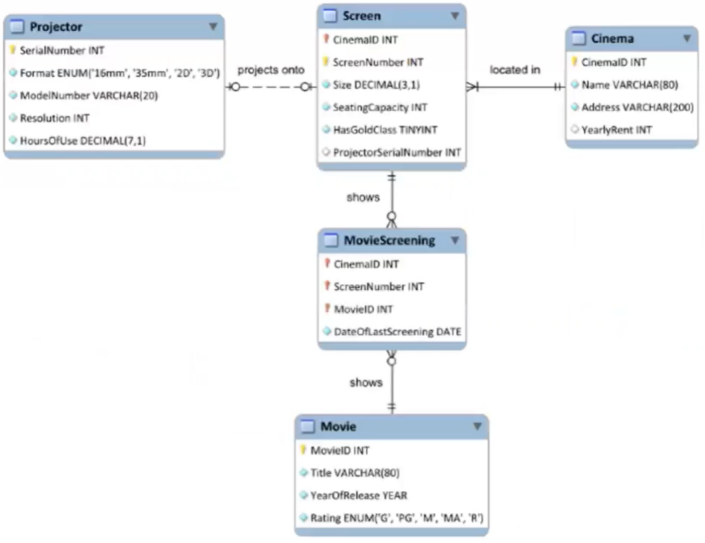
在logical design过程中，需要用foreign key的形式表示relationships；

1. one-one: 任意一个中拿另外一个做foreign即可；
2. one-to-many: 在受限的那一侧，加入主的PK做FK；
3. many-to-many: 加入一个新的association entity定义, 把两侧entity的PK全部加进来作为FK（记得加上attribute）



This one is

- Cinema (Cinema ID, name, address, yearly rent)
- Screen (Screen Number, Cinema ID (FK), Size, Seating Capacity, IsGoldClass, *Projector Serial Number(FK)*)
- MovieScreening(Cinema ID (FK), Screen Number (FK), Movie ID (FK), DateOfLastScreening)
- Projector (Format, Serial Number, Model Number, Resolution, Hours of Use)
- Movie (Movie ID, Title, Year of Release, Rating)



Relational Model: Summary

a tabular representation of data

integrity constraints can be specified based on IC: primary and foreign keys + domain constraints

Examinable

1. translating conceptual (ER) into logical & physical design
2. understand integrity constraints
3. use ddl of SQL to create tables with constraints

Lecture 2

how to decide the weak entity: bold rectangle & bold the relationshiop & dashed underline

Besides, in the DB system, remember to construct a composite key (with owner's PK as PK as well)