Database Concepts (IV)

Database Analysis and Design

Chaokun Wang

School of Software, Tsinghua University

chaokun@tsinghua.edu.cn

April 12, 2021

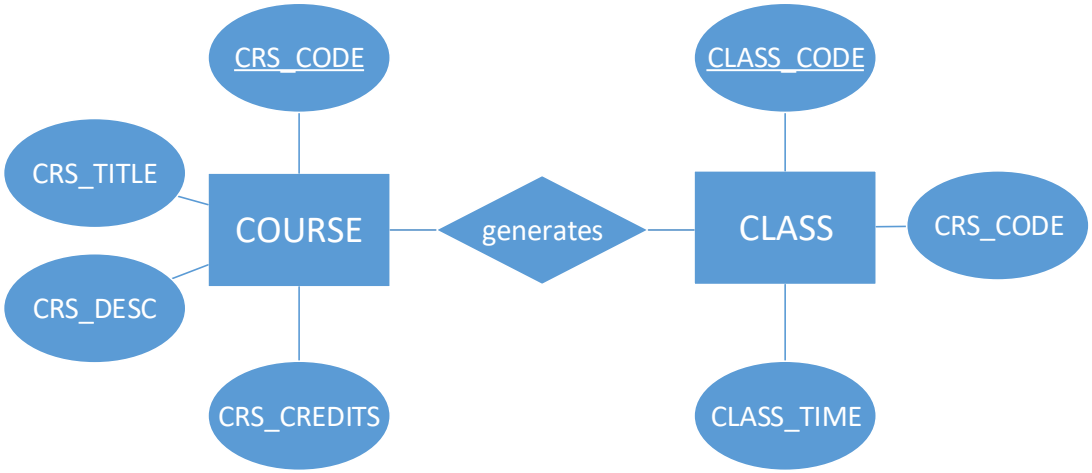
Outline

- Entity-relationship Modeling*
- Database Normalization*

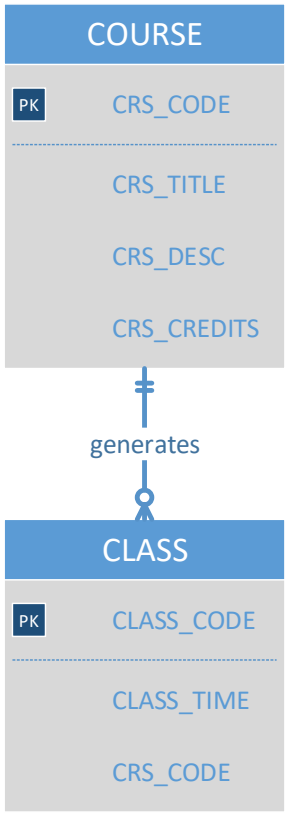
Entity-Relationship Modeling

- Depict the database's main components
  - Entity types
  - Attributes
  - Relationship

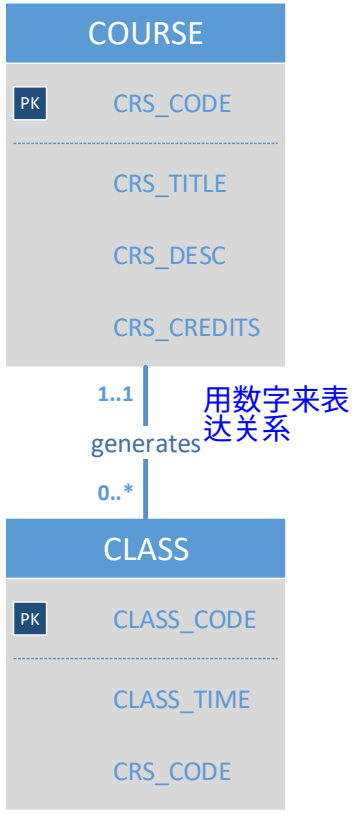
https://blog.csdn.net/aopstudio/article/details/122772545?utm_medium=distribute.pc_aggpage_search_result.none-task-blog-2~aggregatepage~first_rank_ecpm_v1~rank_v31_ecpm-2-122772545.pc_agg_new_rank&utm_term=Crows+foot&spm=1000.2123.3001.4430
【这是有关Craw'sFoot的一个很好懂的解说】



Chen notation



Crow's foot notation



UML notation

Entity Type

- Entity type
  - Group of objects with same properties, identified by enterprise as having an independent existence.

可以是客观存在，还可以是概念
属性可以是物体的属性，也可以是物体之间联系的属性

Physical existence	
Staff	Part
Property	Supplier
Customer	Product

Conceptual existence	
Viewing	Sale
Inspection	Work experience

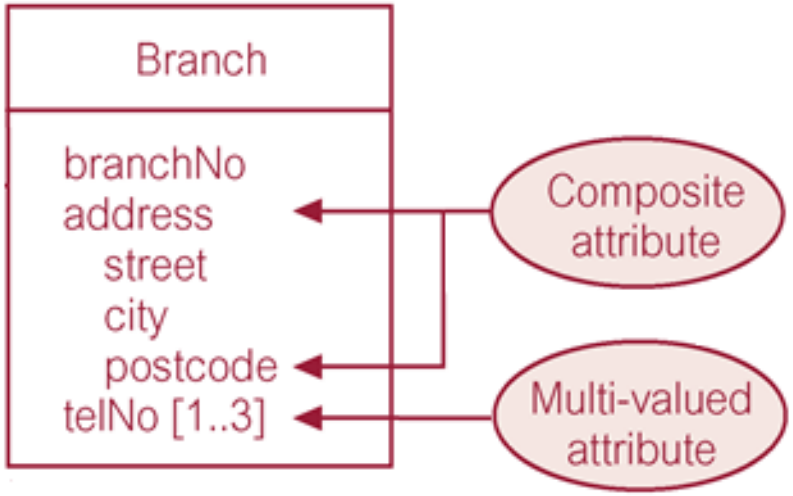
Attributes

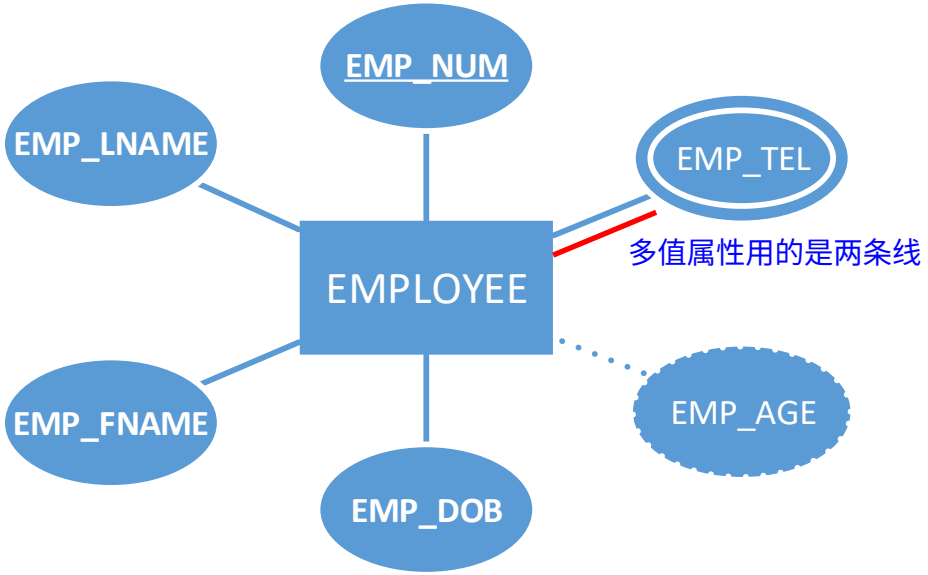
- Property of an entity or a relationship type.
- Identifiers (Primary Keys)
  - Composite Identifiers
- Required & Optional Attributes
  - Required: an attribute that must have a value
  - Optional: it can be left empty
- Simple & Composite Attribute
  - Simple: composed of a single component with an independent existence. 不可分的
  - Composite: composed of multiple components, each with an independent existence.

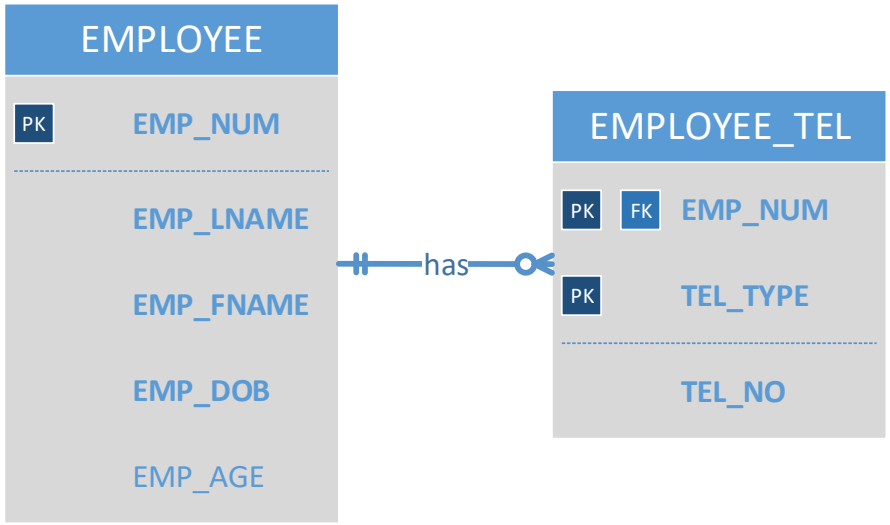
Attributes

- Single-valued Attribute 单值
  - Attribute that holds a single value for each occurrence of an entity type.
- Multi-valued Attribute 多值
  - Attribute that holds multiple values for each occurrence of an entity type.
- Derived Attribute 导出属性
  - Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.

Attributes


Branch
branchNo
address
street
city
postcode
telNo [1..3]
Composite attribute
Multi-valued attribute

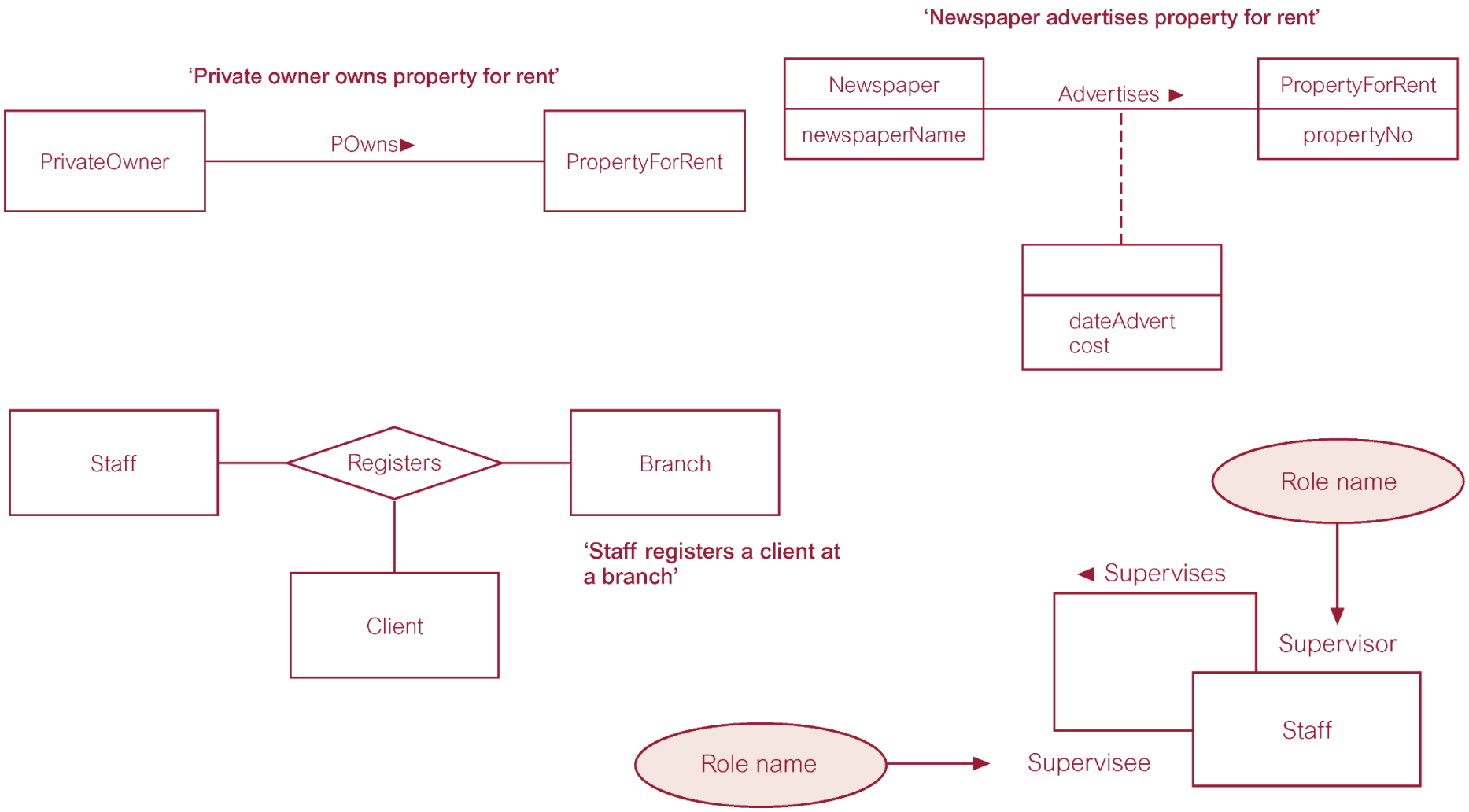

EMP_NUM
EMP_LNAME
EMP_TEL
多值属性用的是两条线
EMPLOYEE
EMP_FNAME
EMP_AGE
EMP_DOB


EMPLOYEE
PK
EMP_NUM
EMP_LNAME
EMP_FNAME
EMP_DOB
EMP_AGE
has
EMPLOYEE_TEL
PK
FK
EMP_NUM
PK
TEL_TYPE
TEL_NO

Relationship

- Relationship type
  - Set of meaningful associations among entity types.
- Degree 度 of a Relationship
  - Number of participating entities in relationship.
- Recursive Relationship 递归联系
  - Relationship type where same entity type participates more than once in different roles.

Relationship

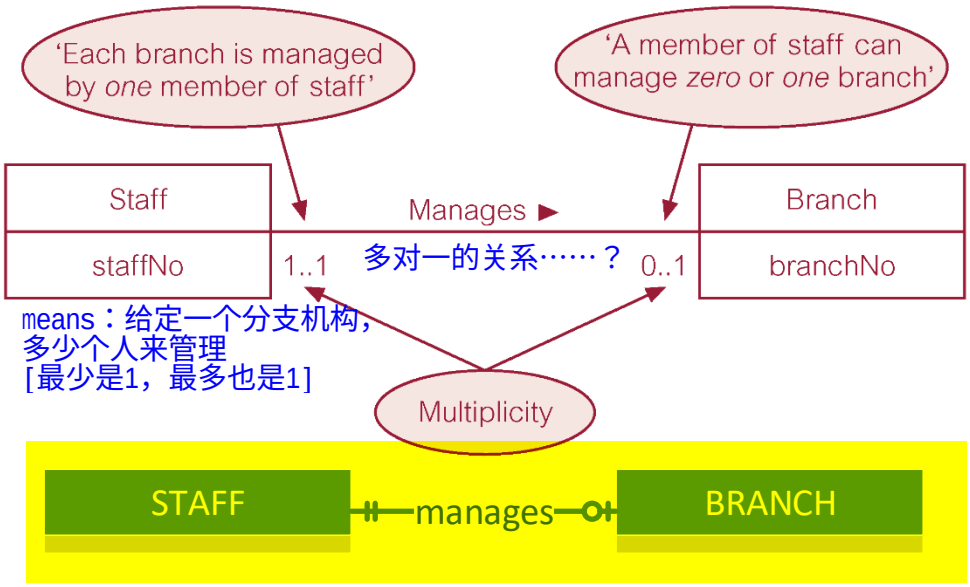


Multiplicity Constraints

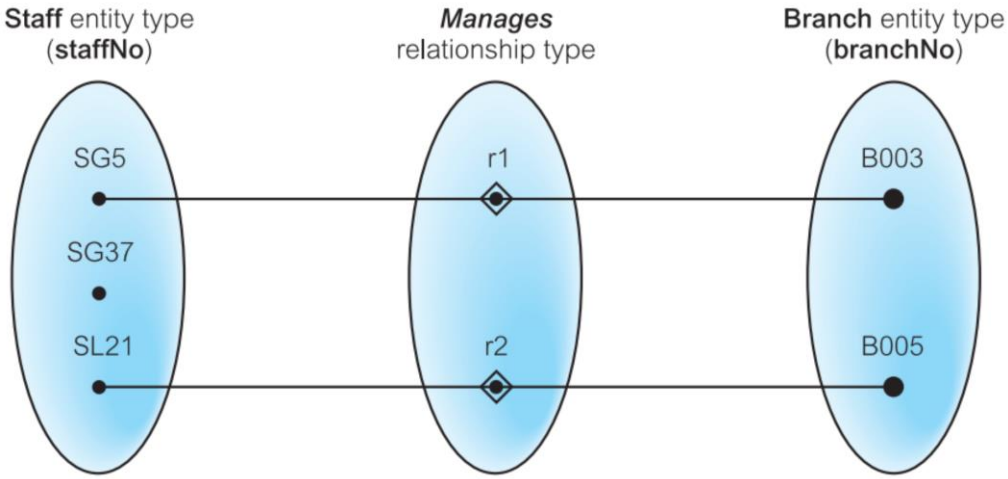
- Multiplicity 多重性：量化上的约束
  - Number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.

- Binary relationships are generally referred to as being:
  - One-to-one (1:1)
  - One-to-many (1:*)
  - Many-to-many (*:*)

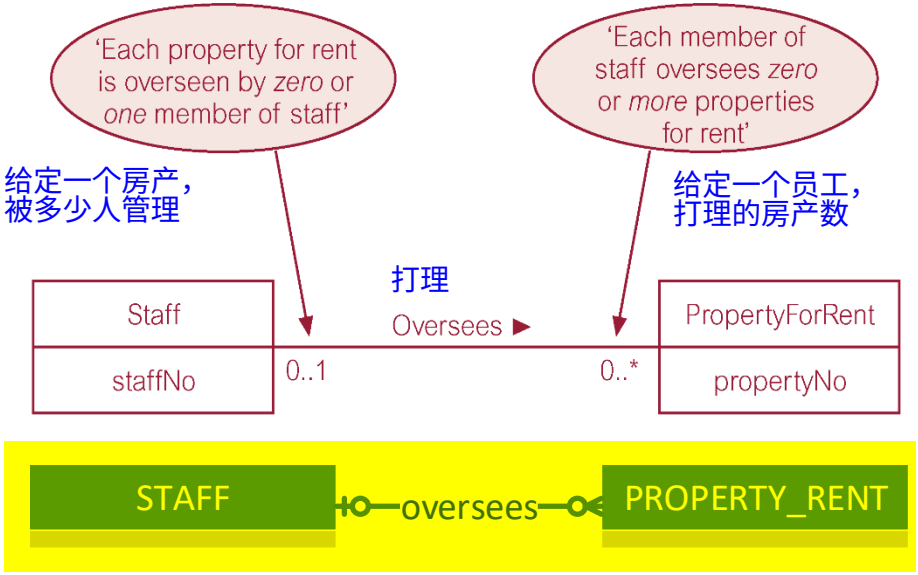
Multiplicity Constraints

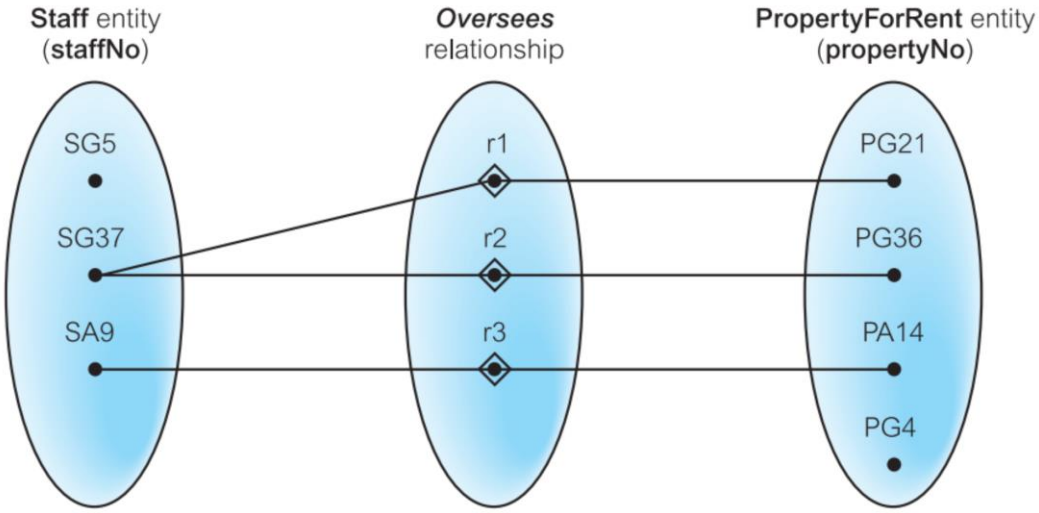


means：给定一个staff，他管理的branch的数量[最少是0，最多是1]

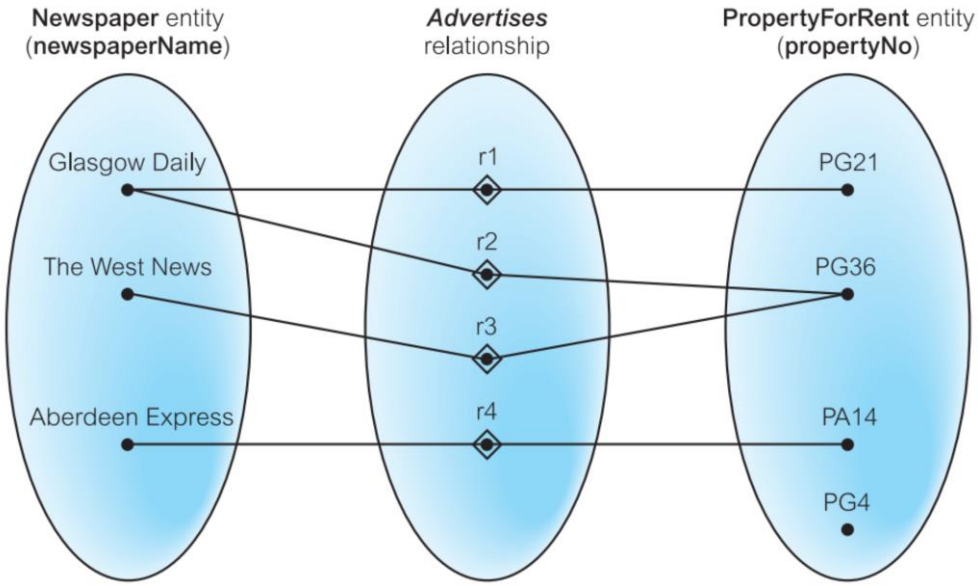


Multiplicity Constraints

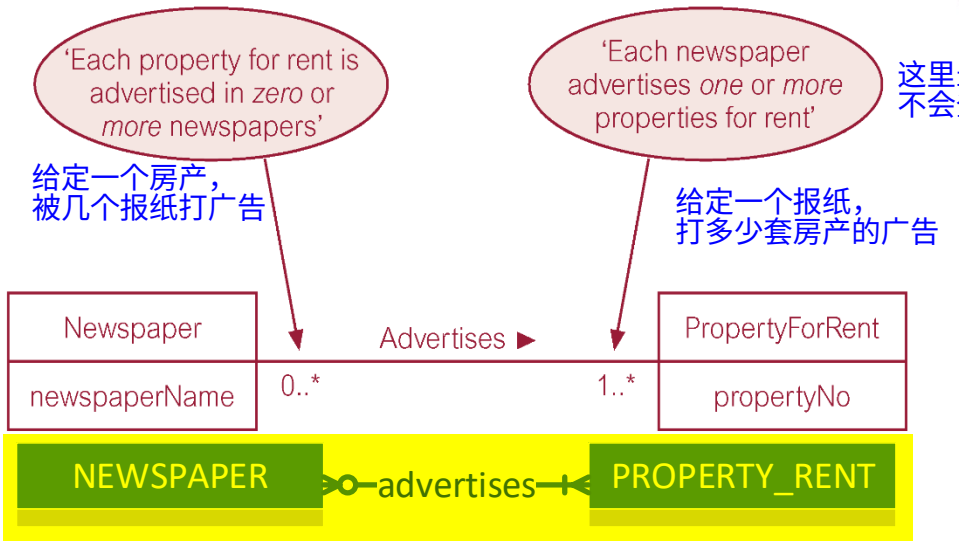




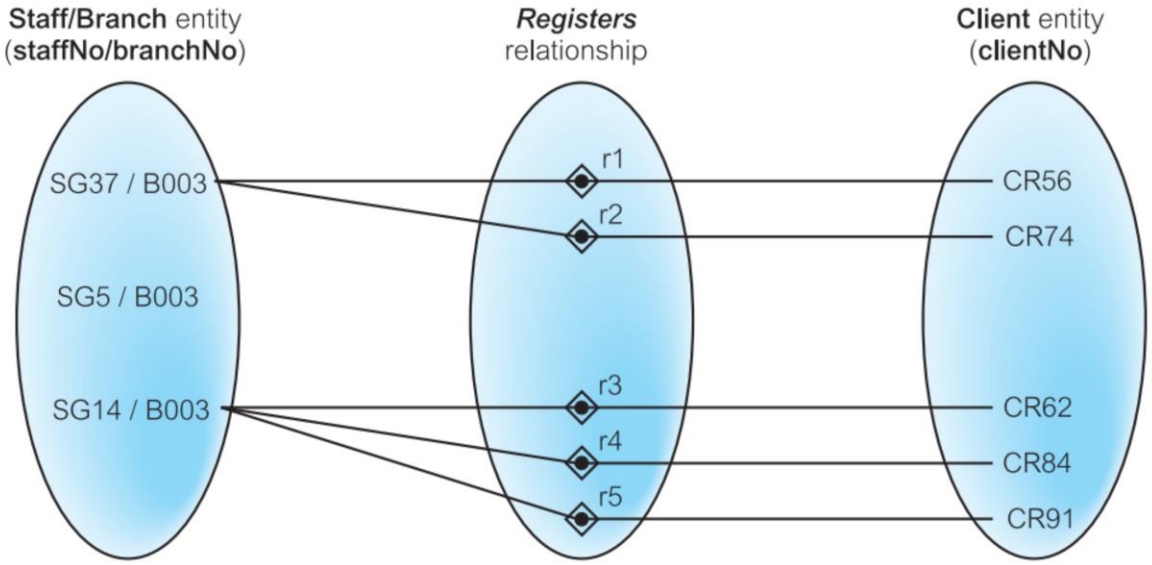
Multiplicity Constraints



这里最少是1是因为如果不打广告的话
不会登记在这个数据库里



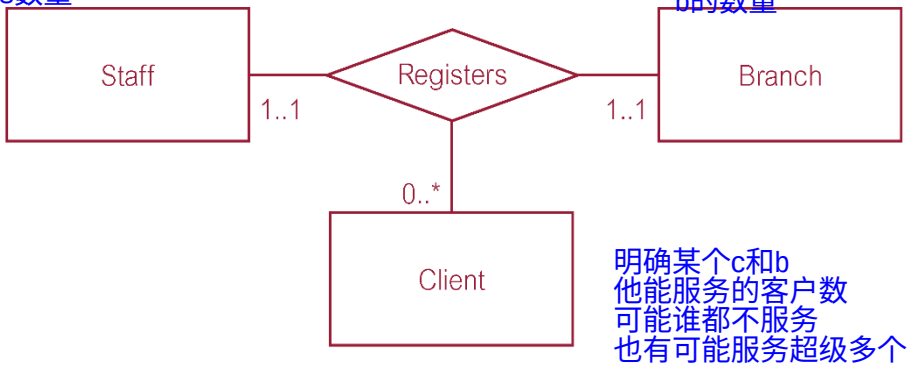
Multiplicity Constraints



原则大概是，
把除了他之外的entity都当做给定

给定某个b和c，接待的
s数量

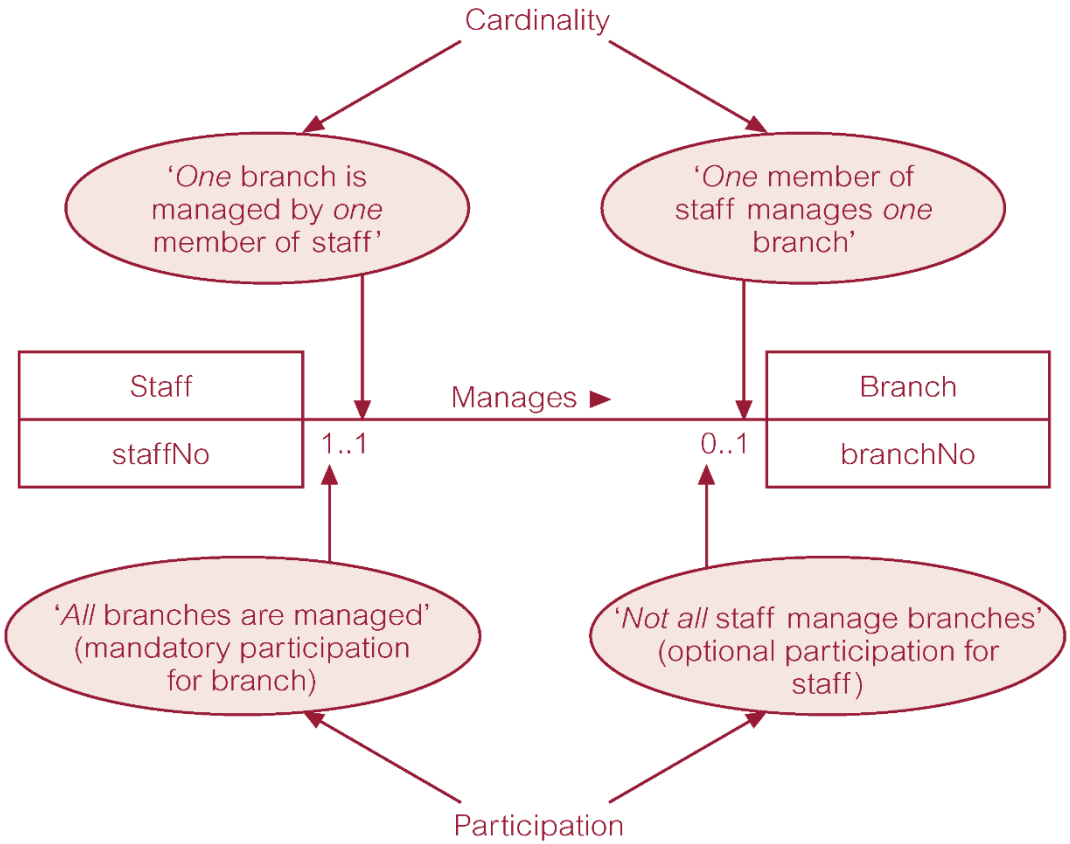
给定某个s和c
b的数量



明确某个c和b
他能服务的客户数
可能谁都不服务
也有可能服务超级多个

Multiplicity Constraints

- Cardinality
  - Describes maximum number of possible relationship occurrences for an entity participating in a given relationship type.
- Participation
  - Determines whether all or only some entity occurrences participate in a relationship.

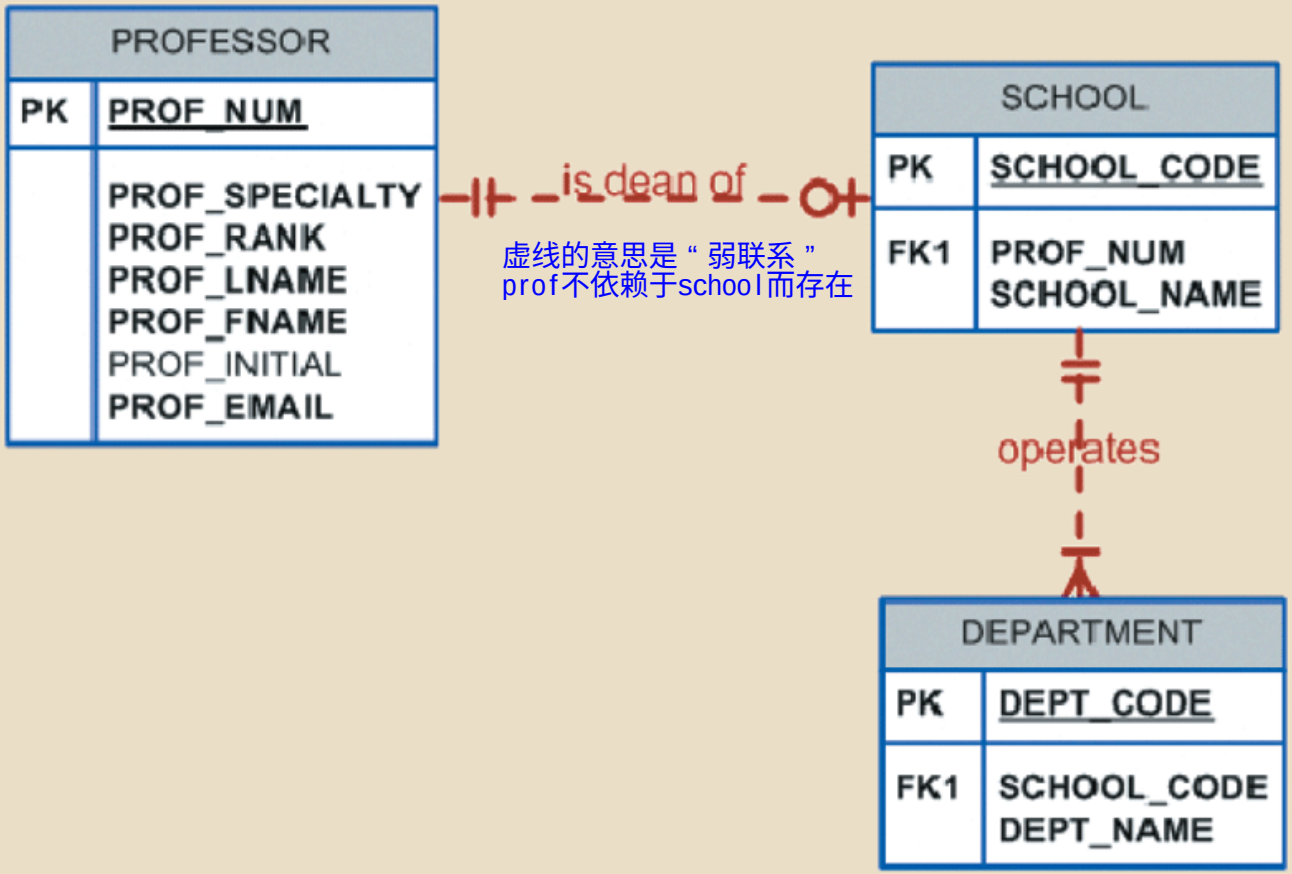


Developing an ER Diagram

- Activities involved in building an ERD
  - Create a detailed narrative of the organization's description of operations
  - Identify business rules based on the descriptions
  - Identify main entities and relationships from the business rules
  - Develop the initial ERD
  - Identify the attributes and primary keys that adequately describe entities
  - Revise and review ERD

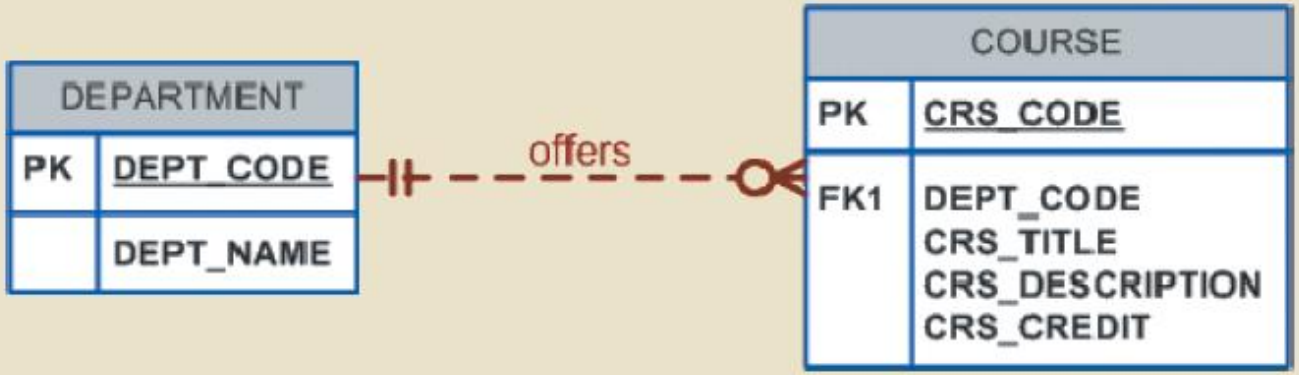
Developing an ER Diagram

FIGURE 4.26 THE FIRST TINY COLLEGE ERD SEGMENT



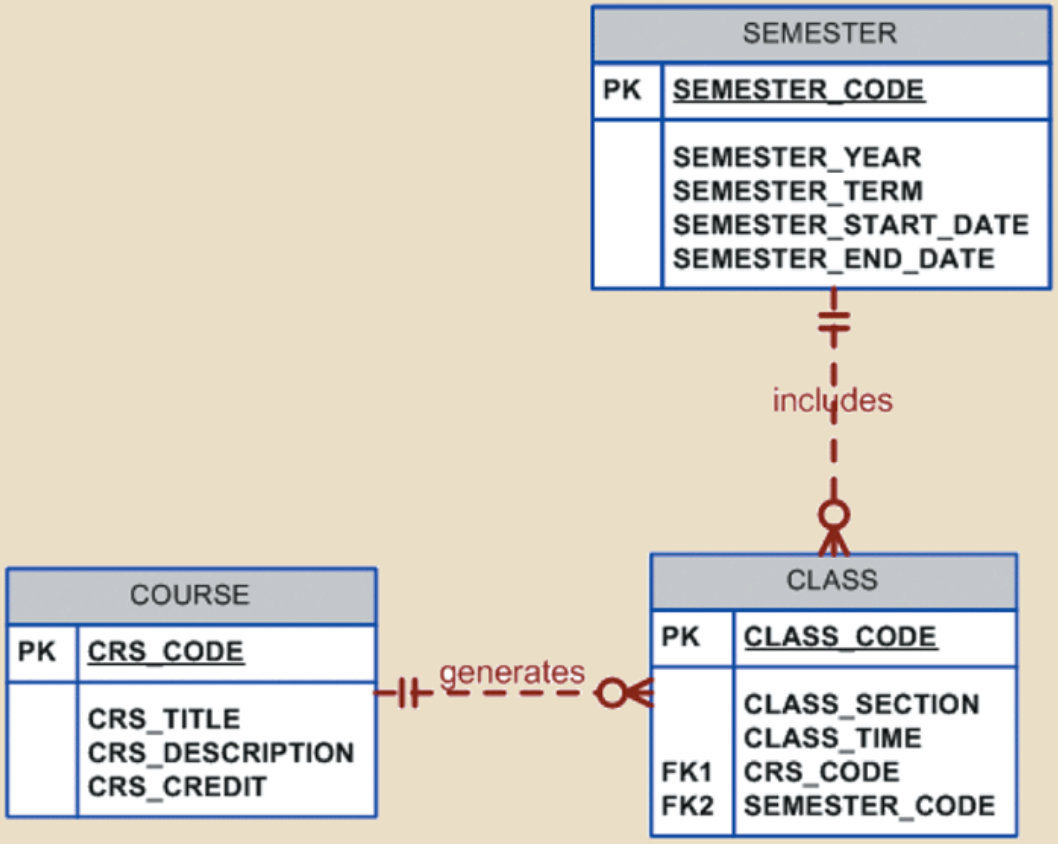
Developing an ER Diagram

FIGURE 4.27 THE SECOND TINY COLLEGE ERD SEGMENT



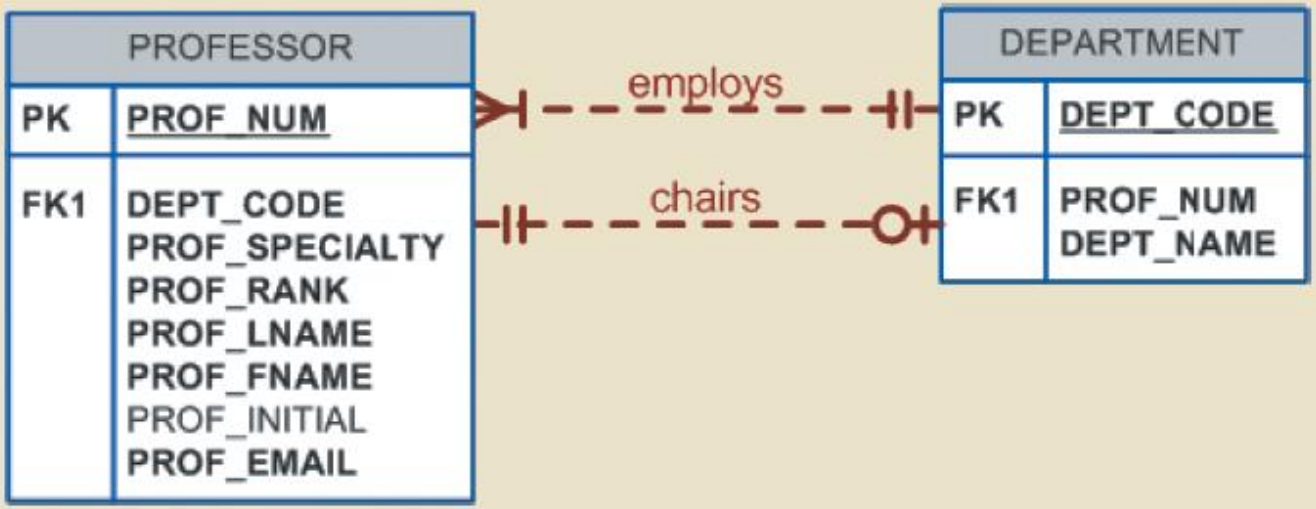
Developing an ER Diagram

FIGURE 4.28 THE THIRD TINY COLLEGE ERD SEGMENT



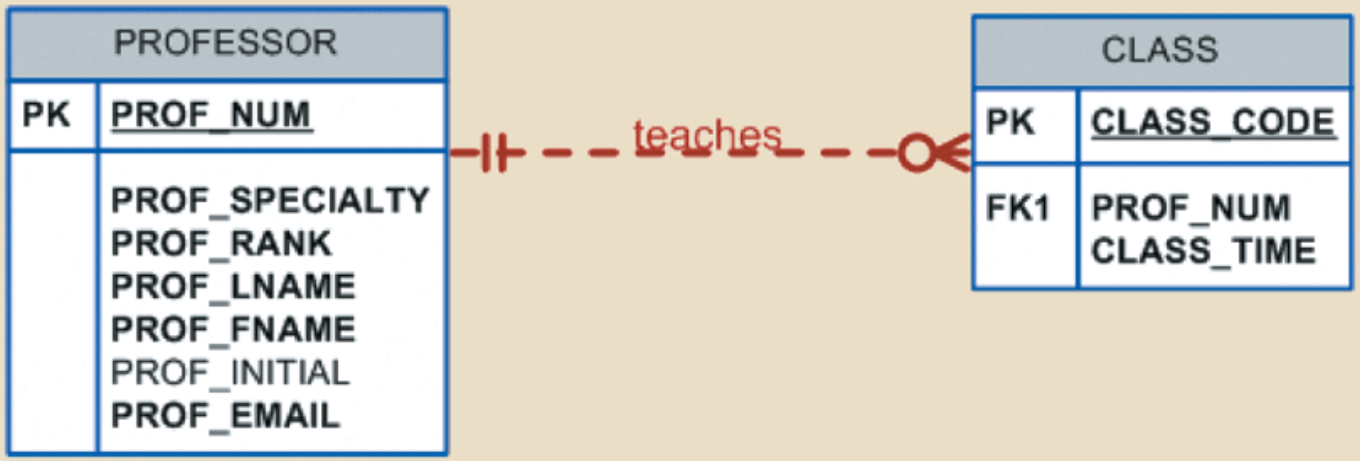
Developing an ER Diagram

FIGURE 4.29 THE FOURTH TINY COLLEGE ERD SEGMENT



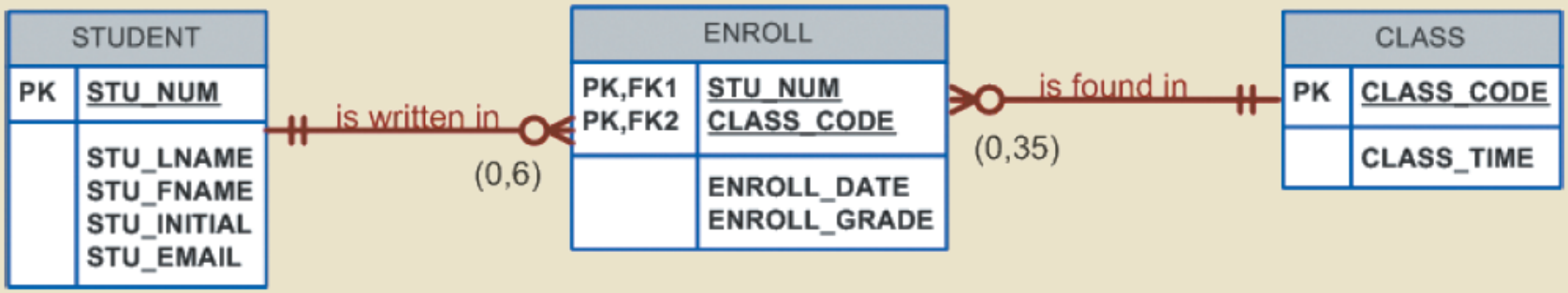
Developing an ER Diagram

FIGURE 4.30 THE FIFTH TINY COLLEGE ERD SEGMENT



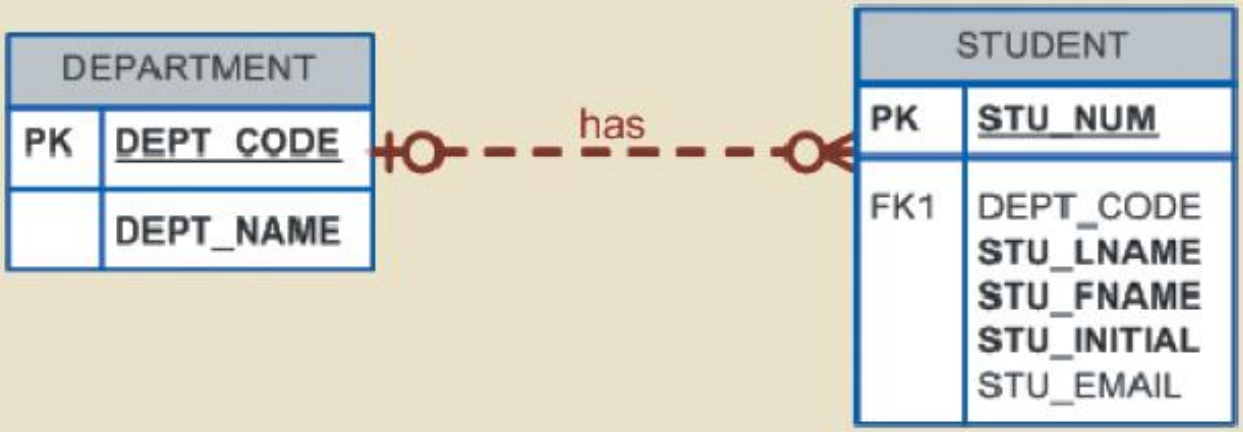
Developing an ER Diagram

FIGURE 4.31 THE SIXTH TINY COLLEGE ERD SEGMENT



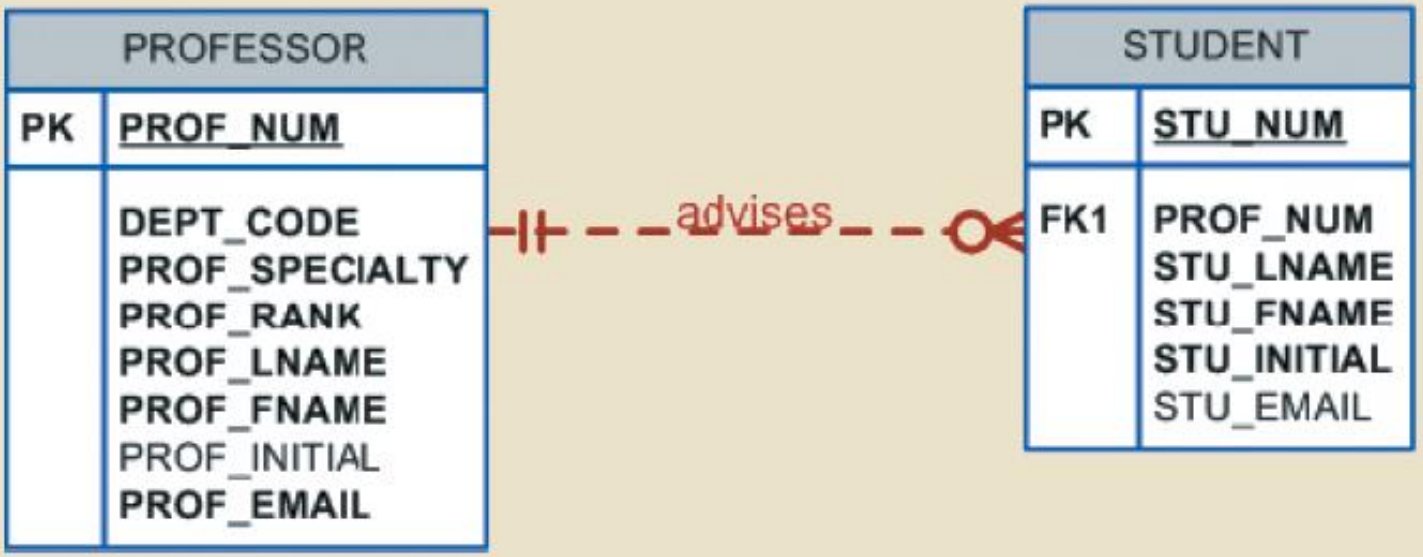
Developing an ER Diagram

FIGURE 4.32 THE SEVENTH TINY COLLEGE ERD SEGMENT



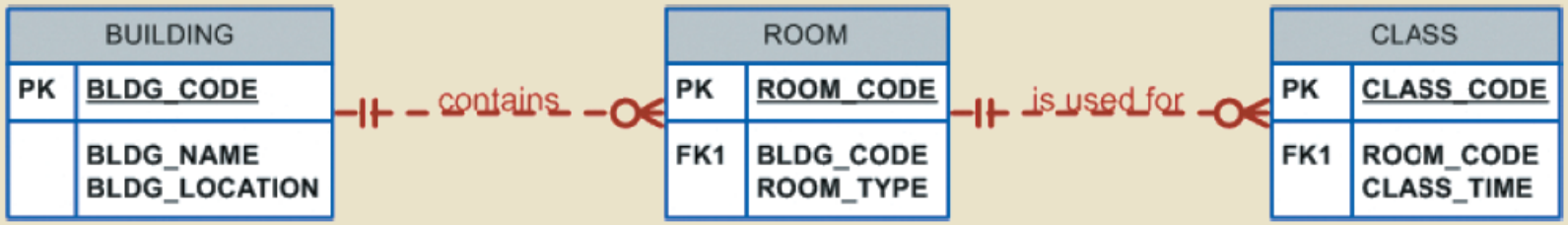
Developing an ER Diagram

FIGURE 4.33 THE EIGHT TINY COLLEGE ERD SEGMENT



Developing an ER Diagram

FIGURE 4.34 THE NINTH TINY COLLEGE ERD SEGMENT



Developing an ER Diagram

Table 4.4	Components of the ERM		
Entity	**Relationship**	**Connectivity**	**Entity**
SCHOOL	operates	1:M	DEPARTMENT
DEPARTMENT	has	1:M	STUDENT
DEPARTMENT	employs	1:M	PROFESSOR
DEPARTMENT	offers	1:M	COURSE
COURSE	generates	1:M	CLASS
SEMESTER	includes	1:M	CLASS
PROFESSOR	is dean of	1:1	SCHOOL
PROFESSOR	chairs	1:1	DEPARTMENT
PROFESSOR	teaches	1:M	CLASS
PROFESSOR	advises	1:M	STUDENT
STUDENT	enrolls in	M:N	CLASS
BUILDING	contains	1:M	ROOM
ROOM	is used for	1:M	CLASS
	Note: ENROLL is the composite entity that implements the M:N relationship "STUDENT enrolls in CLASS."		

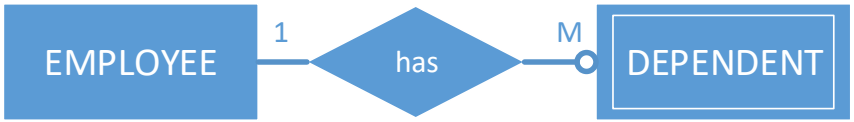
Strong & Weak Entity

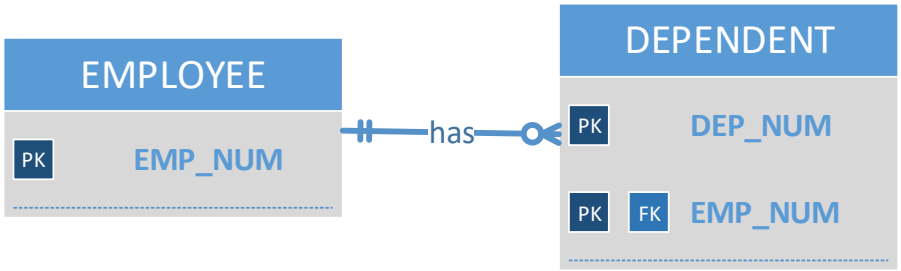
- Strong Entity Type 强实体 e.g.选课信息依赖于学生信息
  - Entity type that is not existence-dependent on some other entity type.

强弱实体的例子给反了

- Weak Entity Type 弱实体 e.g. 教授不依赖于学院而存在
  - Entity type that is existence-dependent on some other entity type.

较为显然的一个特征就是主键关系

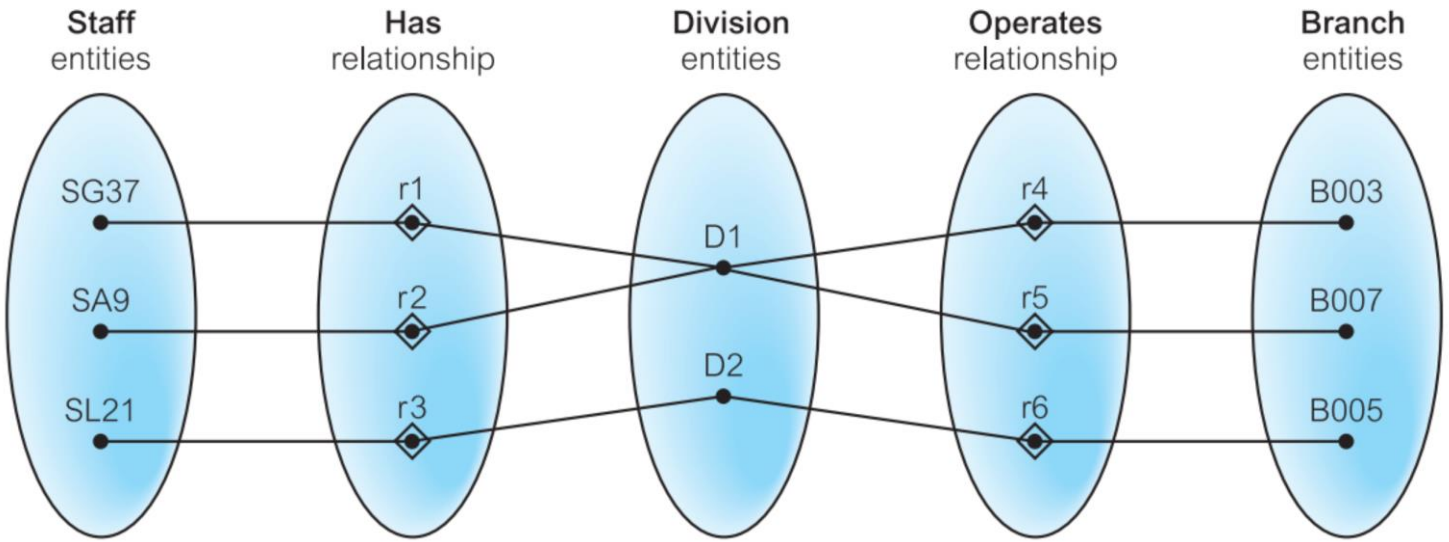




Problems with ER Models

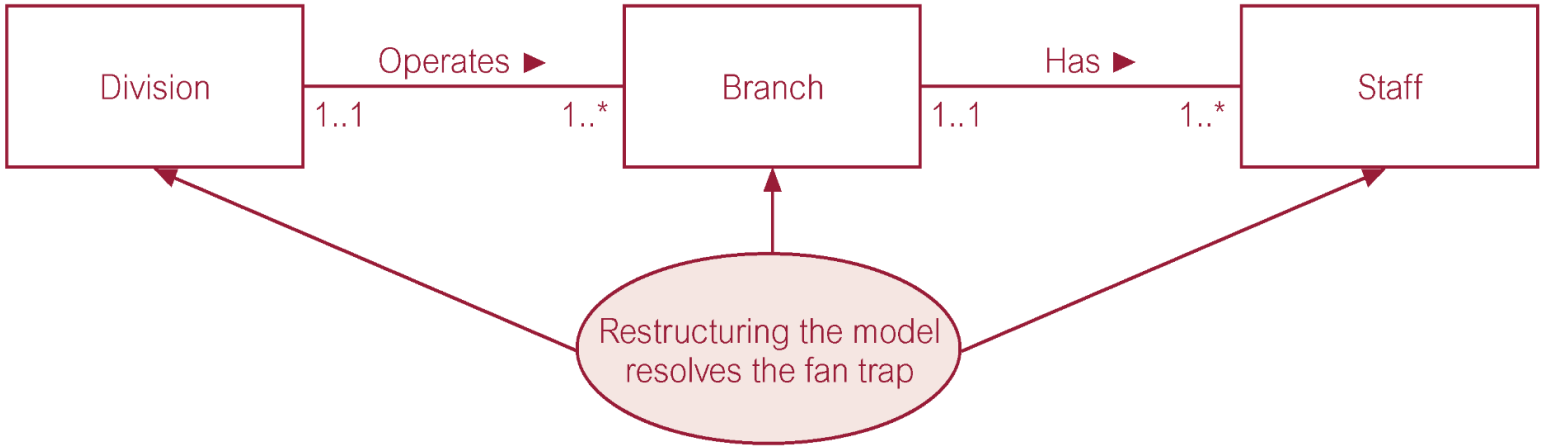
- Fan Trap 扇形陷阱
  - Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.

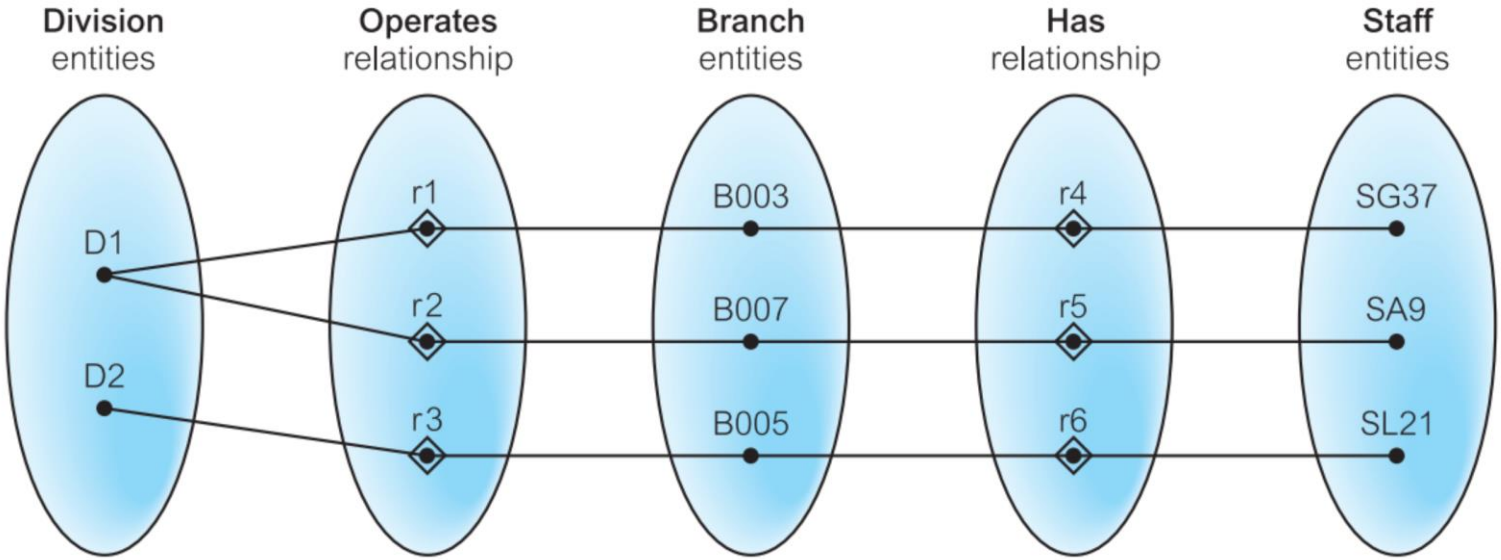
这样子没办法回答“ s在哪个b里 ”



Problems with ER Models

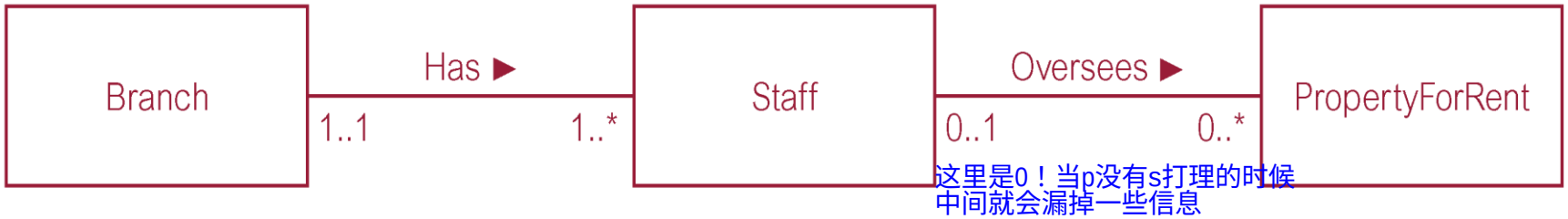
- Fan Trap



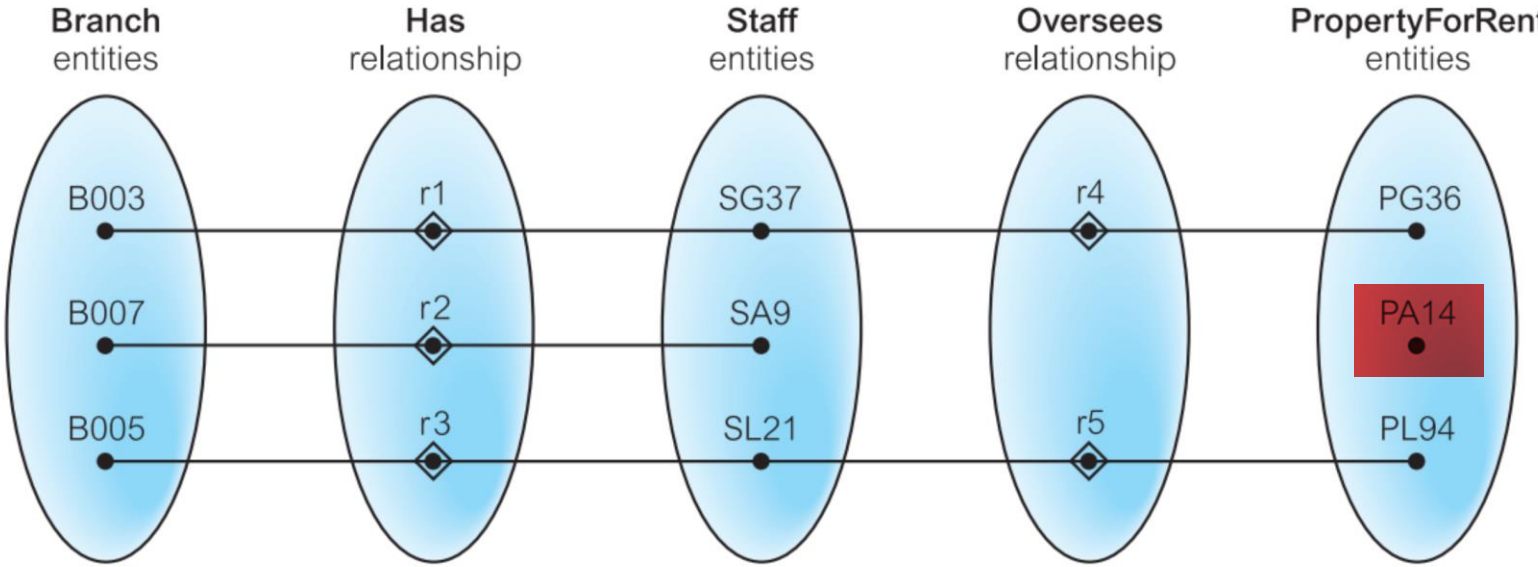


Problems with ER Models

- Chasm Trap
  - Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.



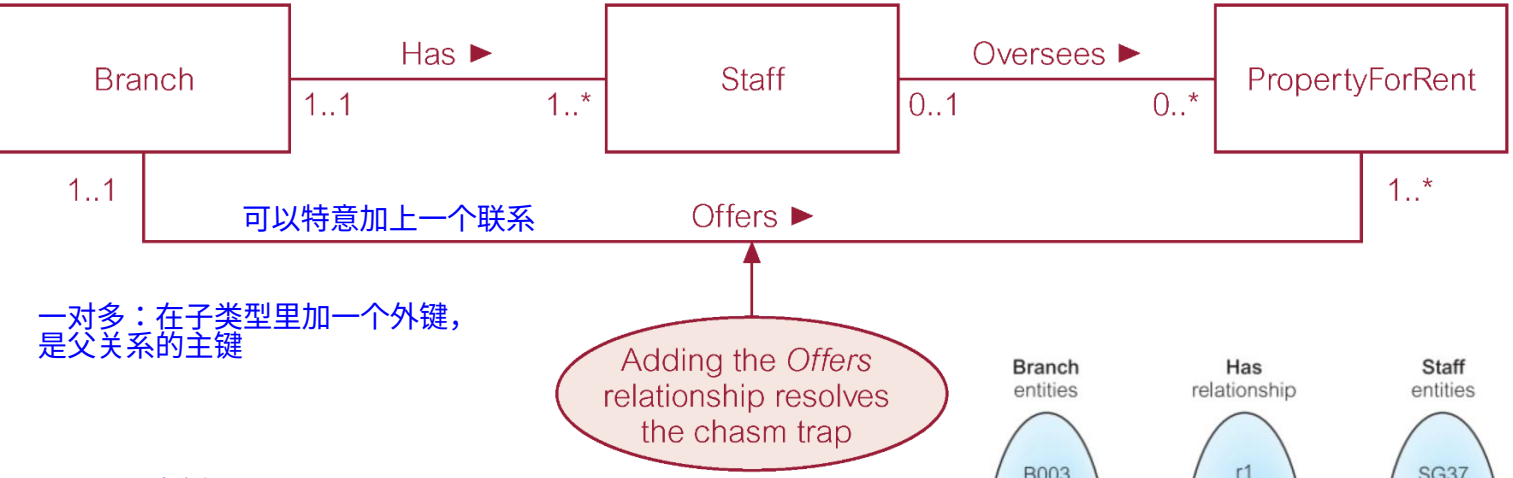
这里是0！当p没有s打理的时候
中间就会漏掉一些信息



咦！！！

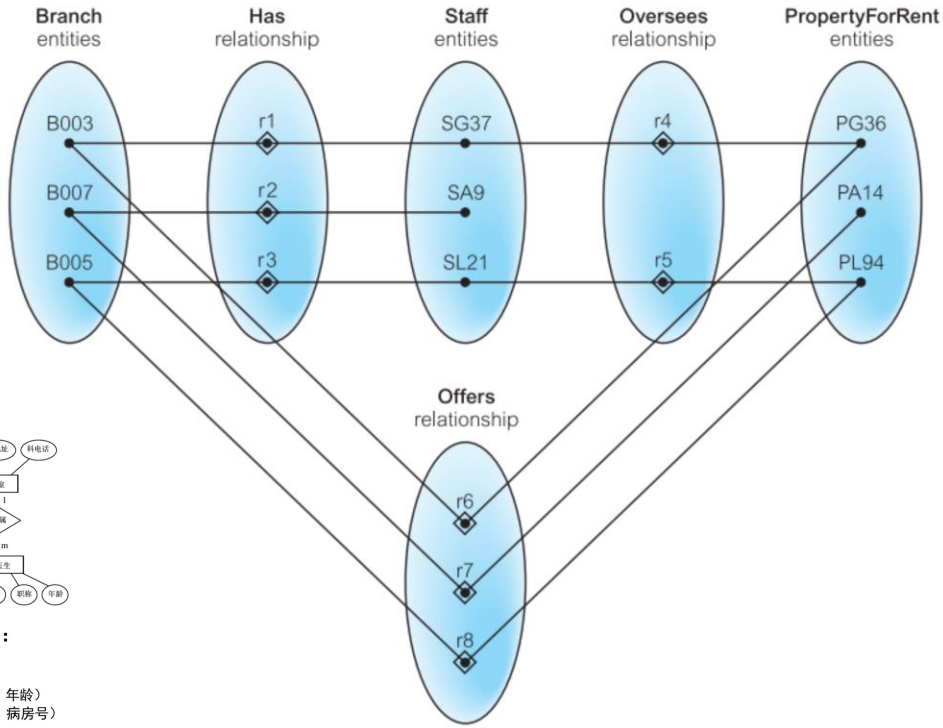
Problems with ER Models

- Chasm Trap



一对多：在子类型里加一个外键，是父关系的主键

Adding the *Offers* relationship resolves the chasm trap



一个例题！

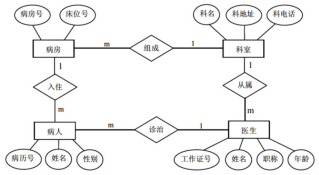
某医院病房计算机管理中心需要如下信息：

科室：科名、科地址、科电话、医生姓名
病房：病房号、床位号、所属科室名
医生：姓名、职称、所属科室名、年龄、工作证号
病人：病历号、姓名、性别、主管医生、病房号

其中，一个科室有多个病房、多个医生，
一个病房只能属于一个科室，
一个医生只属于一个科室，但可负责多个病人的诊治，
一个病人的主管医生只有一个。

完成如下设计：

(1) 设计该计算机管理系统的E-R图。
(2) 将该E-R图转换为关系模式结构。
(3) 指出转换结果中每个关系模式的主键。



- 对应的关系模式结构如下：
  - ✓ 科室（科名，科地址，科电话）
  - ✓ 病房（科室名，病房号，床位号）
  - ✓ 医生（工作证号，姓名，职称，科室名，年龄）
  - ✓ 病人（病历号，姓名，性别，主管医生，病房号）

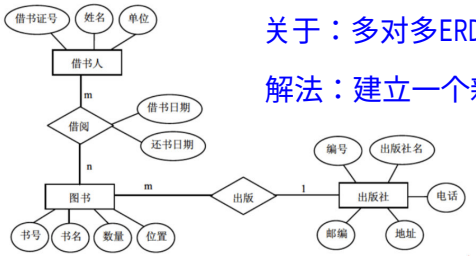
Outline

- Entity-relationship Modeling*
- Database Normalization*

又一道例题：

一个图书借阅管理数据库要求提供下述服务：
（1）可随时查询书库中现有书籍的品种、数量与存放位置。所有各类书籍均可由书号唯一标识。
（2）可随时查询书籍借还情况，包括借书人单位、姓名、借书证号、借书日期和还书日期。
约定：任何人可借多种书，任何一种书可为多个人所借，借书证号具有惟一性。
（3）当需要时，可通过数据库中保存的出版社的电报编号、电话、邮编及地址等信息向相应出版社增购有关书籍。
约定：一个出版社可出版多种书籍，同一本书仅为一个出版社出版，出版社名具有惟一性。
根据以上情况和假设，试作如下设计：
（1）构造满足需求的E-R图。
（2）转换为等价的关系模式结构。

ERD



关于：多对多ERD的范例

解法：建立一个新关系

- 转换为等价的关系模式结构如下：
  - ✓ 借书人（借书证号，姓名，单位）
  - ✓ 图书（书号，书名，数量，位置，出版社名）
  - ✓ 出版社（出版社名，电报编号，电话，邮编，地址）
  - ✓ 借阅（借书证号，书号，借书日期，还书日期）

Data Redundancy

假设SA9离职，删掉记录的时候007branch的信息就没了

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Staff

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Branch

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

Database Tables and Normalization

- Normalization: evaluating and correcting table structures to minimize data redundancies
  - Reduces data anomalies
  - Assigns attributes to tables based on determination
- Normal forms
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)

The Need for Normalization

- Used while designing a new database structure
  - Analyzes the relationship among the attributes within each entity
  - Determines if the structure can be improved through normalization
  - Improves the existing data structure and creates an appropriate database design
- Structural point of view of normal forms
  - Higher normal forms are better than lower normal forms

The Normalization Process

- Objective is to ensure that each table conforms to the concept of well-formed relations
  - Each table represents a single subject
  - Each row/column intersection contains only one value and not a group of values
  - No data item will be unnecessarily stored in more than one table
  - All nonprime attributes in a table are dependent on the primary key
  - Each table has no insertion, update, or deletion anomalies 没有增删改

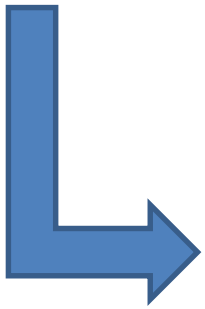
The Normalization Process (cont)

- Ensures that all tables are in at least 3NF
  - Higher forms are not likely to be encountered in business environment
- Works one relation at a time
  - Identifies the dependencies of a relation (table)
  - Progressively breaks the relation up into a new set of relations

Basic Idea

- The domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain

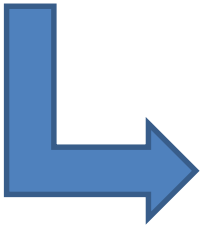
Employee	Skill	Branch	Work Location
Brown	Simulation, Database	B1	73 Industrial Way, London, UK
Harrison	Database	B1	73 Industrial Way, London, UK
Jones	Simulation, Programme	B2	114 Main Street, Dublin, Ireland



Employee	Skill	Branch	Country	City	Address Line
Brown	Simulation	B1	UK	London	73 Industrial Way
Brown	Database	B1	UK	London	73 Industrial Way
Harrison	Database	B1	UK	London	73 Industrial Way
Jones	Simulation	B2	Ireland	Dublin	114 Main Street
Jones	Programme	B2	Ireland	Dublin	114 Main Street

Basic Idea (cont)

Employee	Skill	Branch	Country	City	Address Line
Brown	Simulation	B1	UK	London	73 Industrial Way
Brown	Database	B1	UK	London	73 Industrial Way
Harrison	Database	B1	UK	London	73 Industrial Way
Jones	Simulation	B2	Ireland	Dublin	114 Main Street
Jones	Programme	B2	Ireland	Dublin	114 Main Street



Employee	Branch
Brown	B1
Harrison	B1
Jones	B2

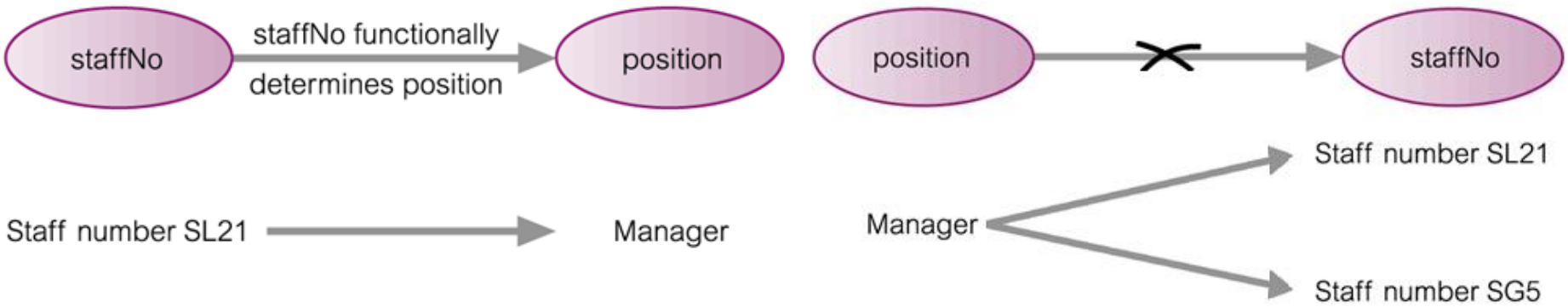
Branch	Country	City	Address Line
B1	UK	London	73 Industrial Way
B2	Ireland	Dublin	114 Main Street

Employee	Skill
Brown	Simulation
Brown	Database
Harrison	Database
Jones	Simulation
Jones	Programme

Functional Dependencies

函数依赖

- Functional dependency describes relationship between attributes.
  - If A and B are attributes of relation R, B is functionally dependent on A (denoted A → B), if each value of A in R is associated with exactly one value of B in R.



Full Functional Dependencies

- Full functional dependency 完全函数依赖
  - A and B are attributes of a relation. B is fully functionally dependent on A, if B is functionally dependent on A, but not on any proper subset of A.

e.g.

Staff

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Transitive Dependencies

- Transitive dependency describes a condition where A, B, and C are attributes of a relation such that if A → B and B → C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

StaffBranch　　baddress是传递依赖于staffno

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

The Process of Normalization

具体的case可以看雨课堂 4月2日的ppt -> 同一文件夹下的note7

Users' Requirements Specification

Transfer attributes into table format

Unnormalized Form

，就是元组

消除了重复的项

Remove repeating groups

First Normal Form (1NF)

消除了部分依赖

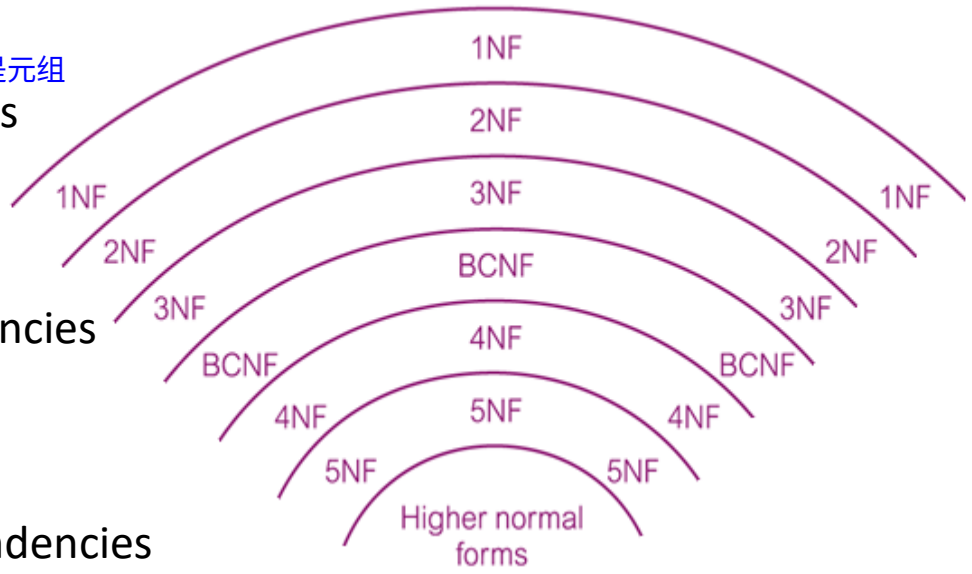
Remove partial dependencies

Second Normal Form (2NF)

消除了传递依赖

Remove transitive dependencies

Third Normal Form (3NF)



Conversion to First Normal Form (1NF)

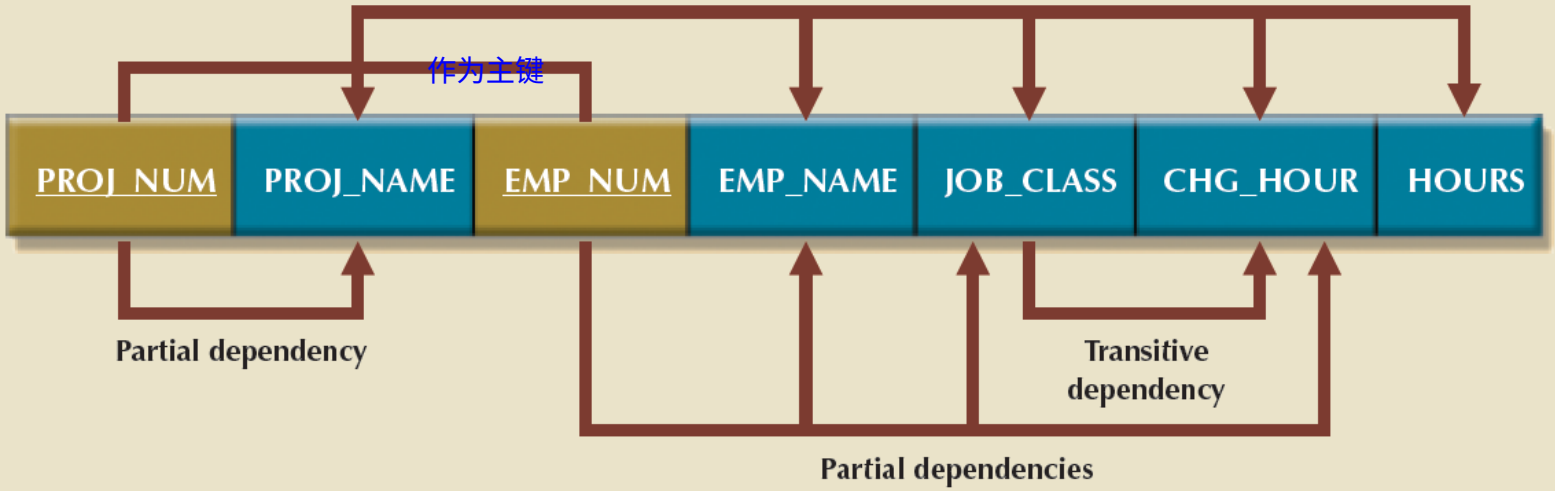
- Repeating group: group of multiple entries of same type can exist for any single key attribute occurrence
  - Reduces data redundancies
- Three step procedure
  - Eliminate the repeating groups
  - Identify the primary key
  - Identify all dependencies
- Dependency diagram: depicts all dependencies found within given table structure
  - Helps to get an overview of all relationships among table's attributes
  - Makes it less likely that an important dependency will be overlooked

Conversion to First Normal Form (1NF)

- 1NF describes tabular format in which:
  - All key attributes are defined
  - There are no repeating groups in the table
  - All attributes are dependent on the primary key
- All relational tables satisfy 1NF requirements
- Some tables contain partial dependencies
  - Update, insertion, or deletion

Conversion to First Normal Form (1NF)

FIGURE 6.3 FIRST NORMAL FORM (1NF) DEPENDENCY DIAGRAM



1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

PARTIAL DEPENDENCIES:
(PROJ_NUM → PROJ_NAME)
(EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR)

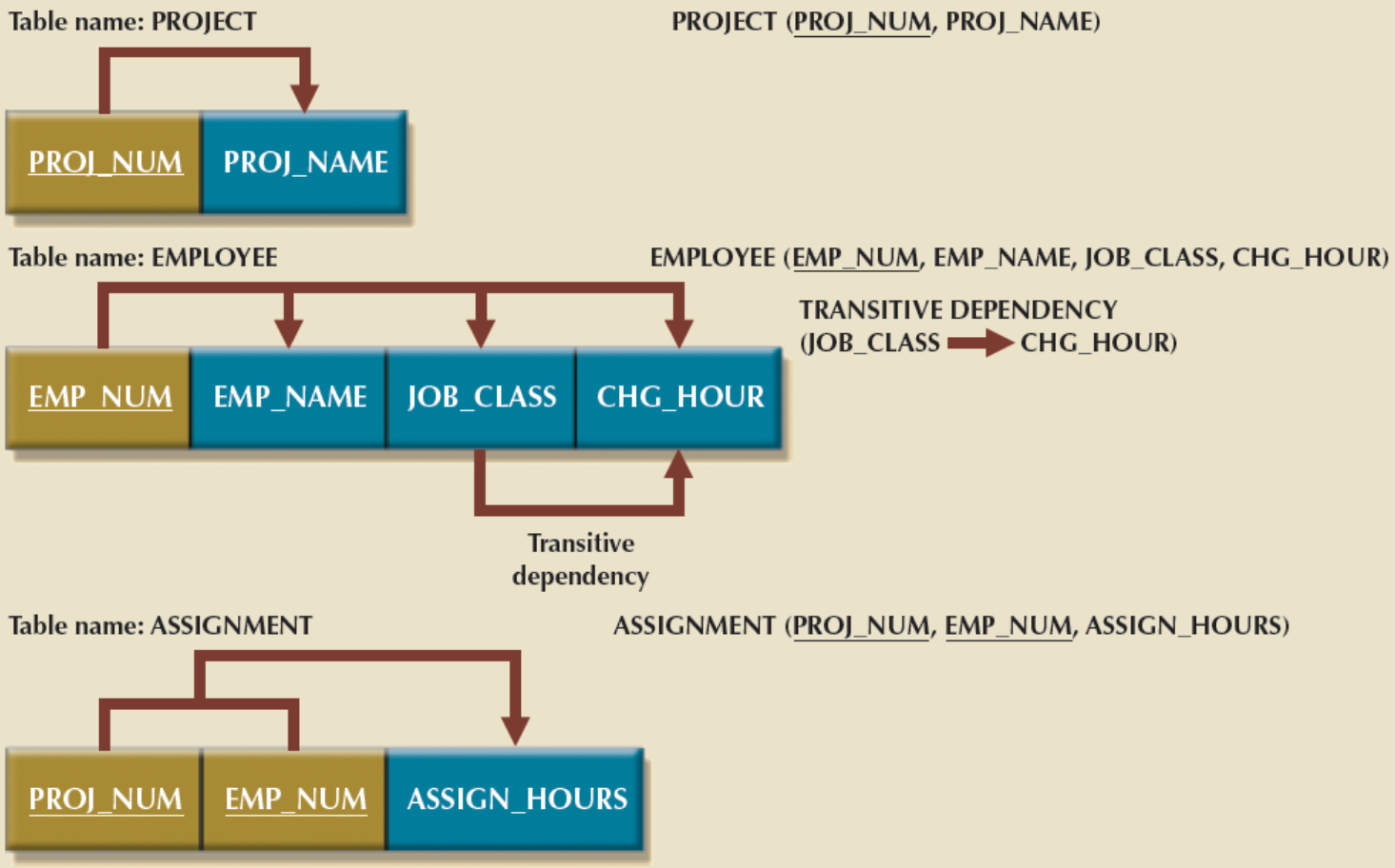
TRANSITIVE DEPENDENCY:
(JOB_CLASS → CHG_HOUR)

Conversion to Second Normal Form (2NF)

- Conversion to 2NF occurs only when the 1NF has a composite primary key
  - If the 1NF has a single-attribute primary key, then the table is automatically in 2NF
- The 1NF-to-2NF conversion is simple
  - Make new tables to eliminate partial dependencies
  - Reassign corresponding dependent attributes
- Table is in 2NF when it:
  - Is in 1NF
  - Includes no partial dependencies

Conversion to Second Normal Form (2NF)

FIGURE 6.4 SECOND NORMAL FORM (2NF) CONVERSION RESULTS



Conversion to Third Normal Form (3NF)

- The data anomalies created by the database organization shown in Figure 6.4 are easily eliminated
  - Make new tables to eliminate transitive dependencies
  - Reassign corresponding dependent attributes
- Table is in 3NF when it:
  - Is in 2NF
  - Contains no transitive dependencies

Conversion to Third Normal Form (3NF)

FIGURE 6.5 THIRD NORMAL FORM (3NF) CONVERSION RESULTS

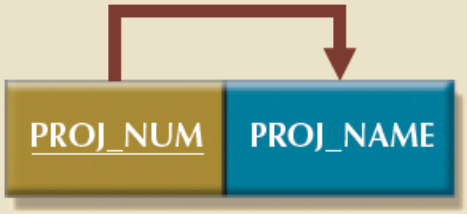


Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

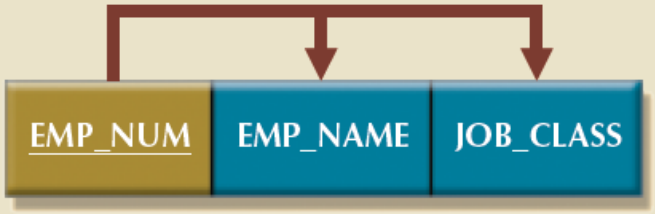


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)

Table name: JOB

JOB (JOB_CLASS, CHG_HOUR)

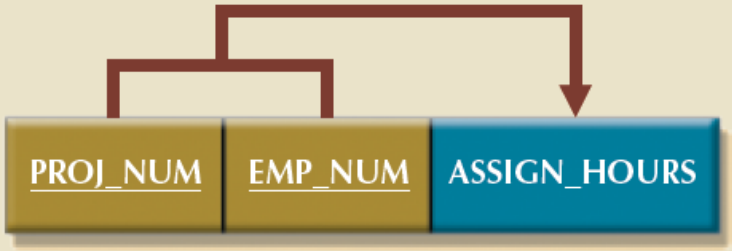


Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

Tutorial: Normalization Data Table

User ID	Customer Name	Order Product	Total Price	Card Type	Card No.	Billing Address	Shipment Address	Purchase Time
1	Sheldon Cooper	Agents of Atlas #1, $3.99 Alien Legion #19, $2.99	6.98	Visa	0622 1234 5678 4321	350 Fifth Avenue, New York, NY 10118-3299	350 Fifth Avenue, New York, NY 10118-3299	2019-11-11 09:36:45
1	Sheldon Cooper	Agents of Atlas #2, $3.99 Alien Legion #20, $2.99	6.98	Master Card	0543 1234 4321 9876	350 Fifth Avenue, New York, NY 10118-3299	1145 17th Street NW, Washington, D.C. 20090-8199	2019-11-12 19:45:32
2	Howard Wolowitz	Avengers A.I. #1, $0.99 Alien Legion #20, $1.99	2.98	Master Card	4321 7777 5332 1986	1145 17th Street NW, Washington, D.C. 20090-8199	1145 17th Street NW, Washington, D.C. 20090-8199	2019-11-14 14:32:12
3	Leonard Hofstadter	Dark Angel #16, $5.99	5.99	Visa	1735 8973 4578 2975	1200 West Harrison Street, Chicago, Illinois 60607-7161	1200 West Harrison Street, Chicago, Illinois 60607-7161	2019-11-15 15:04:03

Tutorial: Normalization Data Table

User ID
Last Name
First Name
Series
Issue
Price
Total Price
Card Type
Card No
Street (Billing)
City (Billing)
State (Billing)
Postcode (Billing)
Street (Shipping)
City (Shipping)
State (Shipping)
Postcode (Shipping)
Purchase Time

Order ID
User ID
Last Name
First Name
Prod ID
Series
Issue
Price
Total Price
Card Type
Card No
Street (Billing)
……
Street (Shipping)
……
Purchase Time

Order ID
User ID
Last Name
First Name
Total Price
Card Type
Card No
Street (Billing)
……
Street (Shipping)
……
Purchase Time

Prod ID
Series
Issue
Price

Order_Prod ID
Order ID
Prod ID

建立一个表吧他俩联系起来

Tutorial: Normalization Data Table

就是一个分解+拆表格的过程

Order ID
User ID
Last Name
First Name
Total Price
Card Type
Card No
Street (Billing)
City (Billing)
State (Billing)
Postcode (Billing)
Street (Shipping)
City (Shipping)
State (Shipping)
Postcode (Shipping)
Purchase Time

Order ID
User ID
Total Price
Card Type
Card No
Street (Billing)
City (Billing)
State (Billing)
Postcode (Billing)
Street (Shipping)
City (Shipping)
State (Shipping)
Postcode (Shipping)
Purchase Time

User ID
Last Name
First Name

Address ID
User ID
Street
City
State
Postcode

Pay Method ID
User ID
Card Type
Card No

Prod ID
Series
Issue
Price

Order_Prod_ID
Order ID
Prod ID

Order ID
User ID
Total Price
Pay Method ID
Billing Addr ID
Shipping Addr ID
Purchase Time

Tutorial: Normalization Data Table

User ID
Last Name
First Name

Prod ID
Series
Issue
Price

City ID
City
State ID

Address ID
User ID
Street
City ID
Postcode

Order_Prod_ID
Order ID
Prod ID

State ID
State

Order ID
User ID
Total Price
Pay Method ID
Billing Addr ID
Shipping Addr ID
Purchase Time

Pay Method ID
User ID
Card Type
Card No

Codd's Relational Database Rules (1 of 2)

Table 13.8	Dr. Codd's 12 Relational Database Rules	
Rule	Rule Name	Description
1	Information	All information in a relational database must be logically represented as column values in rows within tables.
2	Guaranteed access	Every value in a table is guaranteed to be accessible through a combination of table name, primary key value, and column name.
3	Systematic treatment of nulls	Nulls must be represented and treated in a systematic way, independent of data type.
4	Dynamic online catalog based on the relational model	The metadata must be stored and managed as ordinary data—that is, in tables within the database; such data must be available to authorized users using the standard database relational language.
5	Comprehensive data sublanguage	The relational database may support many languages; however, it must support one well-defined, declarative language as well as data definition, view definition, data manipulation (interactive and by program), integrity constraints, authorization, and transaction management (begin, commit, and rollback).
6	View updating	Any view that is theoretically updatable must be updatable through the system.
7	High-level insert, update, and delete	The database must support set-level inserts, updates, and deletes.

Codd's Relational Database Rules (2 of 2)

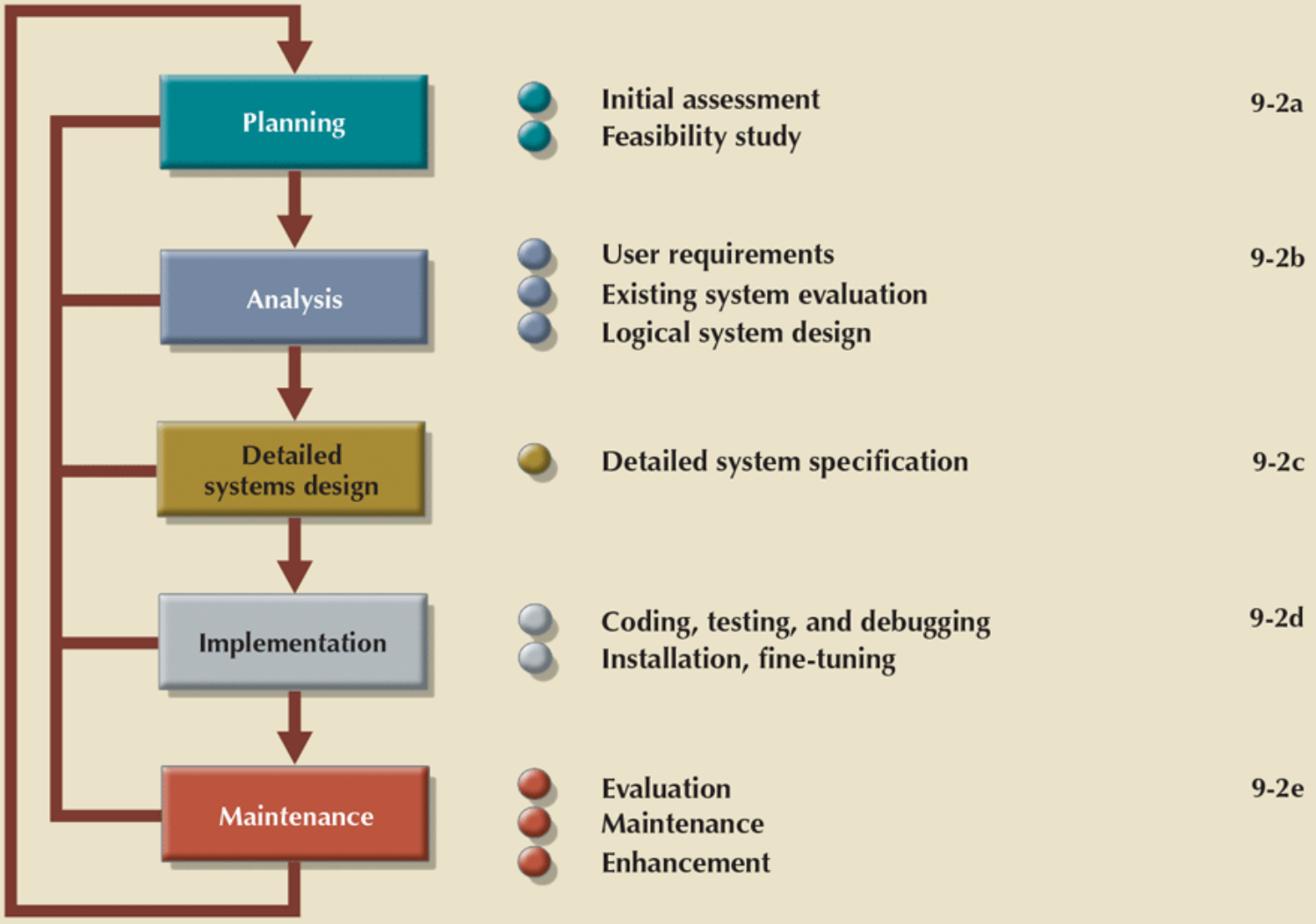
Table 13.8	Dr. Codd's 12 Relational Database Rules	
Rule	Rule Name	Description
8	Physical data independence	Application programs and ad hoc facilities are logically unaffected when physical access methods or storage structures are changed.
9	Logical data independence	Application programs and ad hoc facilities are logically unaffected when changes are made to the table structures that preserve the original table values (changing order of columns or inserting columns).
10	Integrity independence	All relational integrity constraints must be definable in the relational language and stored in the system catalog, not at the application level.
11	Distribution independence	The end users and application programs are unaware of and unaffected by the data location (distributed vs. local databases).
12	Nonsubversion	If the system supports low-level access to the data, users must not be allowed to bypass the integrity rules of the database.
13	Rule zero	All preceding rules are based on the notion that to be considered relational, a database must use its relational facilities exclusively for management.

Note on Database Design

- The database is part of a larger whole known as an information system (IS)
  - Provides for data collection, storage, and retrieval
    - People, hardware, and software
    - Database(s), application programs, and procedures
- Systems analysis: establishes need for and extent of information system
  - Systems development: process of creating information system

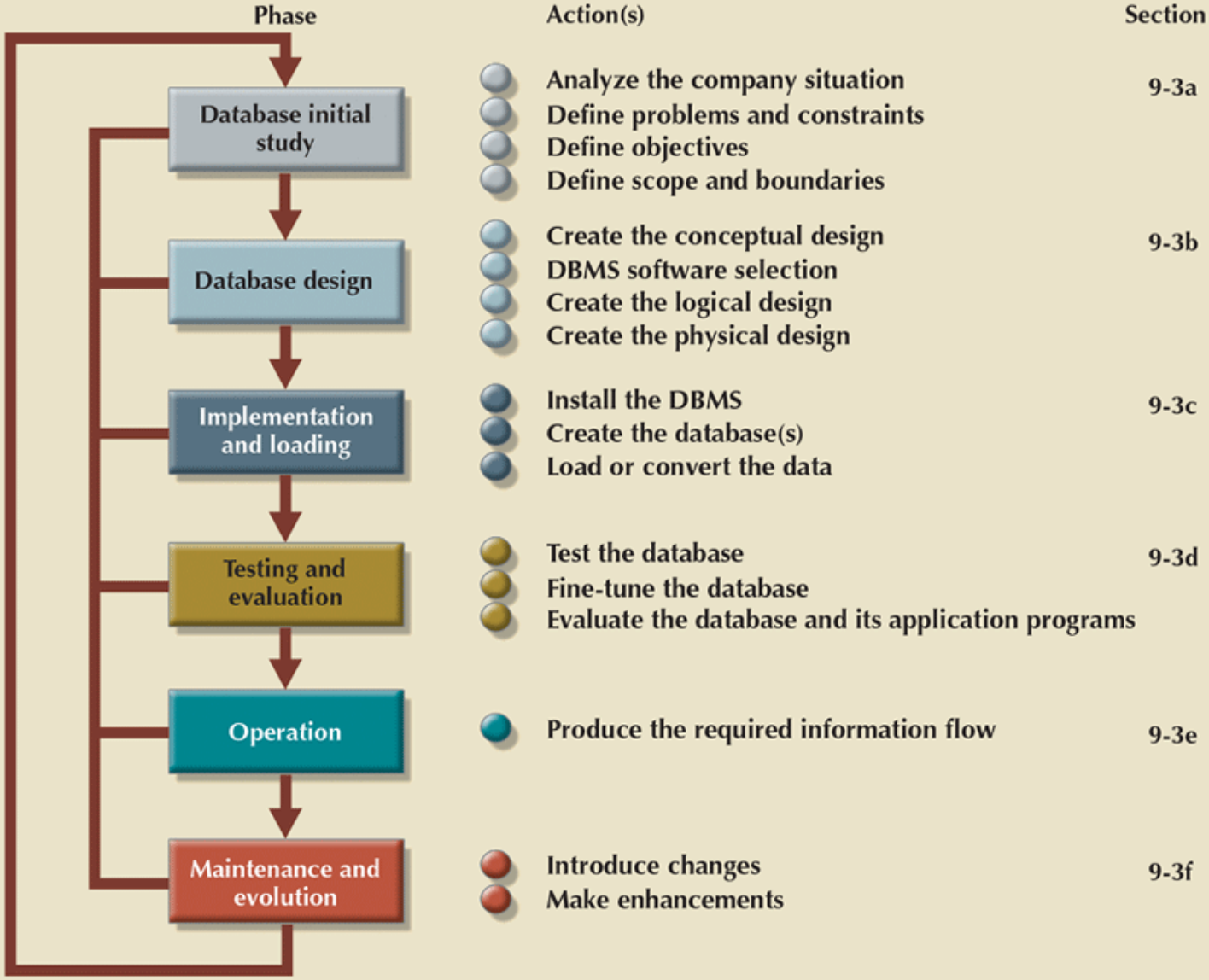
Systems Development Life Cycle (SDLC)

FIGURE 9.2 THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)



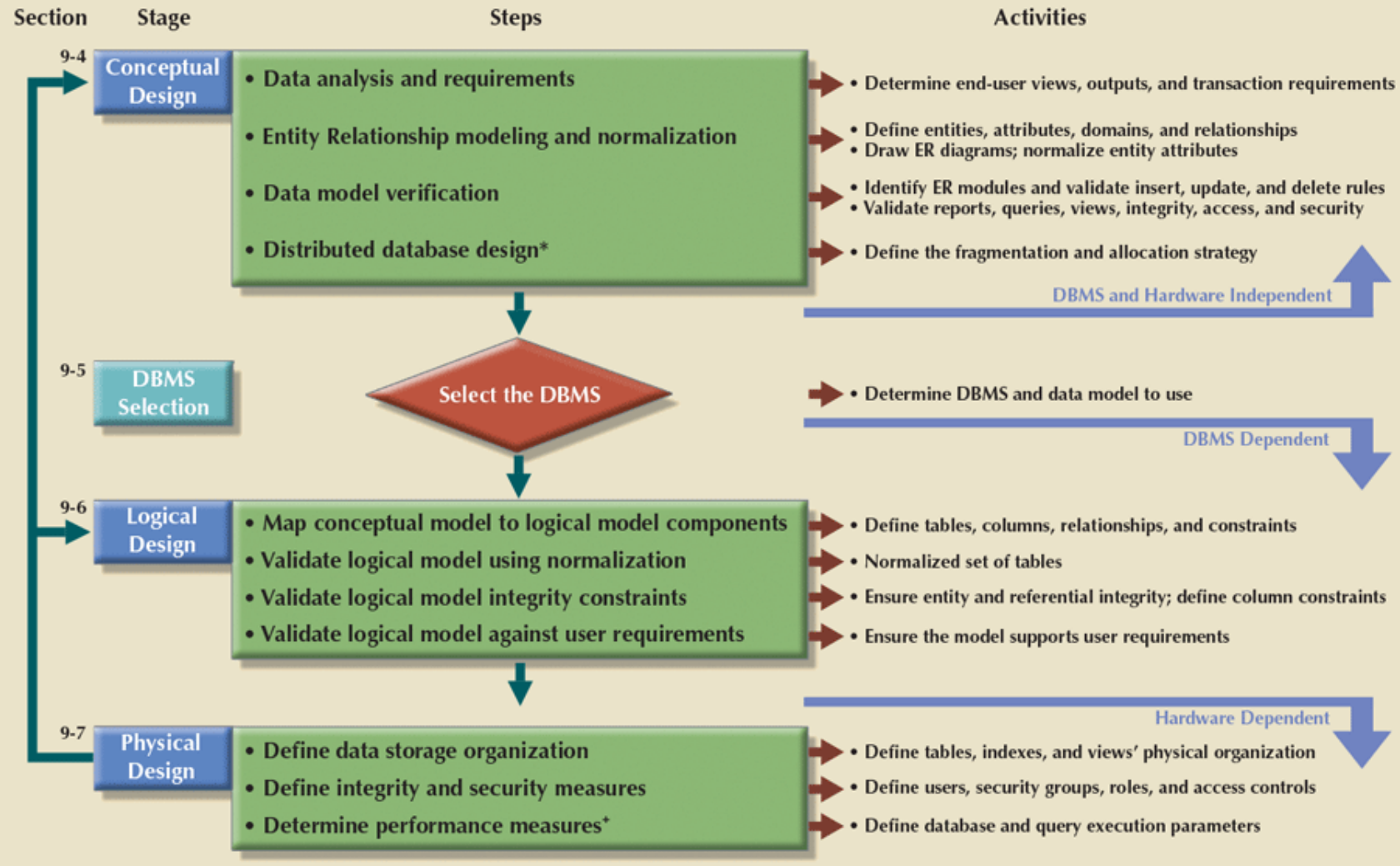
The Database Life Cycle

FIGURE 9.3 THE DATABASE LIFE CYCLE (DBLC)



Database Design Process

FIGURE 9.6 DATABASE DESIGN PROCESS



* See Chapter 12, Distributed Database Management Systems
+ See Chapter 11, Database Performance Tuning and Query Optimization

PARALLEL ACTIVITIES IN

FIGURE 9.8 PARALLEL ACTIVITIES IN THE DBLC AND THE SDLC