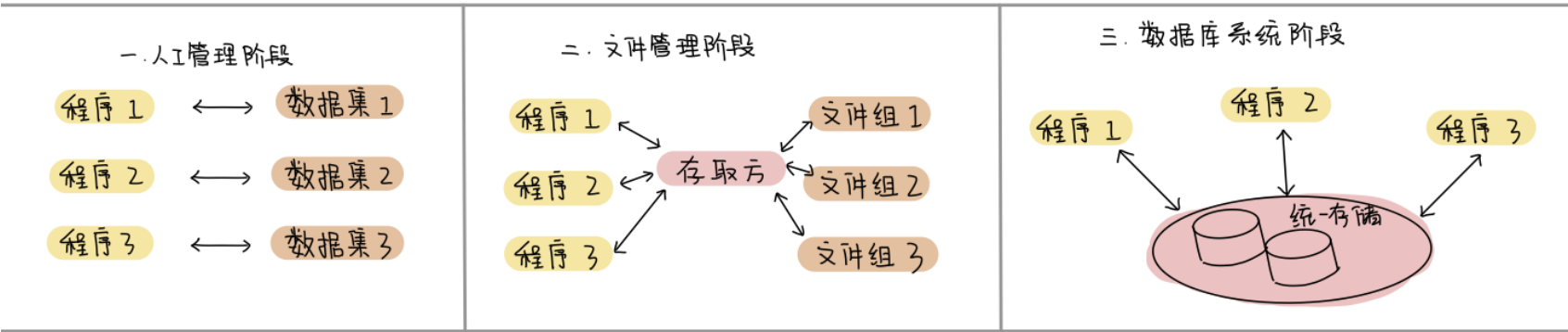


数据库系统

▼ 第一章：数据库基本概念

▼ 数据库系统概述

- 数据库是数据管理的有效技术，是计算机科学的重要分支。**数据库技术是信息系统的核心和基础。**
- 数据库的4个基本概念
 - 数据data：描述事物的符号记录。数据的含义成为语义，数据与其语义密不可分
 - 数据库database：数据是长期存储在计算机内、有组织、可共享的大量数据的集合
 - 数据库管理系统database management system：数据库管理系统是位于用户和操作系统之间的一层数据管理软件，其任务是科学、高效地管理数据库中的数据
 - 数据库管理系统的主要功能：
 - 数据定义功能：DBMS提供数据定义语言（DDL），允许用户对数据库中的数据对象进行定义
 - 数据组织、存储和管理：DBMS要分类组织、存储和管理各种数据。包括数据字典、用户数据、数据的存储路径等。
 - 数据组织和存储的基本目标
 - 提高存储空间利用率
 - 方便存取
 - 提高存储效率，提供多种存取方法，如索引查找、hash查找、顺序查找等
 - 数据操纵功能：DBMS提供数据操纵语言（DML），允许用户对数据库中的数据对象进行基本操作，如查询、插入删除、修改等
 - 数据库的事务管理和运行管理：数据库在建立、运用和维护时由DBMS对数据库进行统一管理和控制，以保证数据的安全性、完整性、以及处理多用户并发控制和发生故障后的系统恢复
 - 数据库的建立和维护功能：如数据库原始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重组、性能监视、分析功能等。这些功能通常由一些程序或管理工具来完成
 - 其他功能：比如DBMS与网络中其他软件系统的通信功能、异构数据库之间的互访和互操作功能、多个DBMS之间的数据转换功能等。
 - 数据库系统database system：计算机系统中引入数据库后的系统，由数据库、数据库管理系统、应用系统、数据库管理员组成。
- 数据管理技术的发展

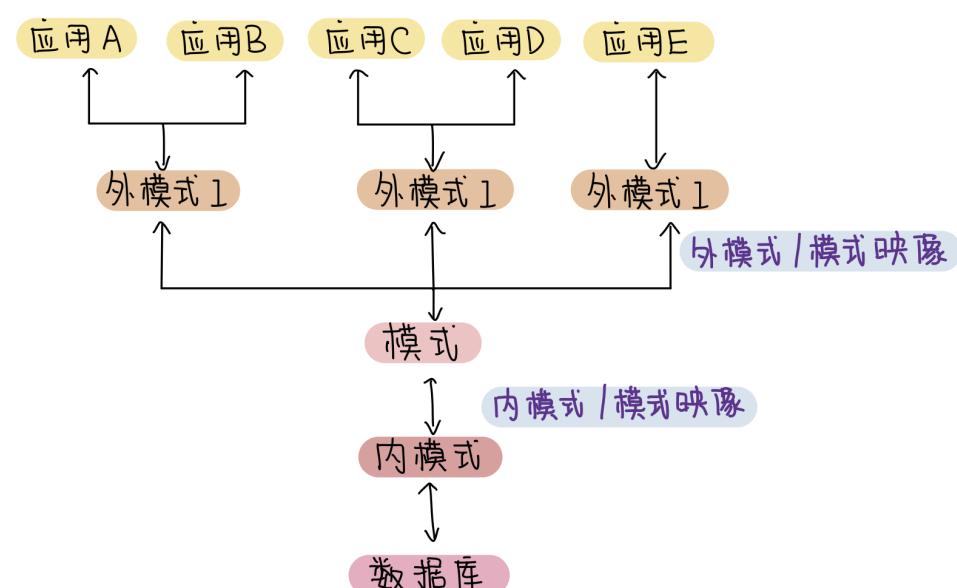


- 第一阶段：人工管理阶段，20世纪50年代中期以前。**“应用程序和数据一一对应”**
 - 数据处理方式为批处理
 - 数据不保存：当时计算机主要用于科学计算，一般不对数据进行长期保存
 - 应用程序管理数据：此阶段没有操作系统，更没有管理数据的专用系统（DBMS）。需要自己设计、管理应用程序，加重了程序员的负担
 - 数据不共享：数据不能被多个应用程序共享。数据是面向应用程序的，一组数据只能被一个应用程序使用。多个程序使用相同数据时，需要各自定义。因此存在大量冗余数据。

- 数据不具有独立性：当数据的逻辑结构或物理结构发生改变，应用程序需要被修改。
2. 第二阶段：文件系统阶段，20世纪50年代后期至60年代中期
- 数据可以长期保存：此阶段计算机大量用于数据处理，数据需要长期保存在外存上
 - 由专门的软件系统（文件系统）管理数据：此阶段出现了操作系统和数据管理软件（文件系统）
 - 文件系统：实现了记录内的结构性，但整体无结构 → 文件内部有结构，但文件和文件之间无结构
 - 数据共享性差、冗余度大：在文件系统中，文件仍然是面对应用程序的。当多个应用程序使用相同文件时，必须建立各自文件。由于数据的重复存储，导致了数据冗余、数据不一致，使数据修改和处理难度加大。（数据不一致指同一数据不同拷贝的值不一样）
 - 数据独立性差：文件系统使应用程序和数据之间有了一定的独立性，但仍然不足。由于文件是面向应用程序的，当文件系统逻辑结构发生改变后，必须修改应用系统对文件结构的定义。
3. 第三阶段：数据库系统阶段，20世纪90年代至今
- 数据结构化：实现了整体数据的结构化，即文件之间也是有结构的，这种结构可以通过参照完整性来表述。
 - 第二阶段和第三阶段的本质区别：整体是否结构化（文件之间是否有结构）
 - 数据的共享性高，冗余度低，易扩充：数据面向整个系统，而不是某一应用程序，实现了不同程序对数据的共享。数据共享可以大大减少数据冗余，节约存储空间，还能够避免数据之间的不相容性和不一致性，方便了数据的修改和维护。
 - 数据的独立性高：由于数据不再面向某一应用程序，使得数据的定义从程序中分离出来，大大减小了程序员的负担。在DBMS中，数据的独立性是由二级映像来保证的。
 - 数据由DBMS统一管理和控制：数据不再由应用程序管理，而是由DBMS统一管理。DBMS通过4个方面实现数据控制功能：数据的安全性保护、数据的完整性检查、并发控制、数据库恢复。

▼ 数据库系统结构

- 数据库系统的三级模式：外模式、模式、内模式
 - 模式schema：也叫逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个模式。
 - 外模式external schema：也叫子模式，用户模式等。他是数据库用户（程序员、最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。是保证数据库安全的有力措施。一个数据库可以有多个外模式。
 - 内模式internal schema：也叫存储模式，是数据物理结构和存储方式的描述，是数据在数据库内部的组织方式。一个数据库只有一个内模式。
- 数据库系统的二级映像



- 外模式/模式映像：定义了外模式与模式之间的对应关系。当模式改变时，由数据库管理员对各个外模式/模式的映像做相应改变，可以使外模式保持不变。应用程序是根据外模式编写的，从而使应用程序不必被修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。
- 模式/内模式映像：定义了数据全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变时，由数据库管理员对各个模式/内模式的映像做相应改变，可以使模式保持不变，从而应用程序也不必变，保证了数据与程序的物

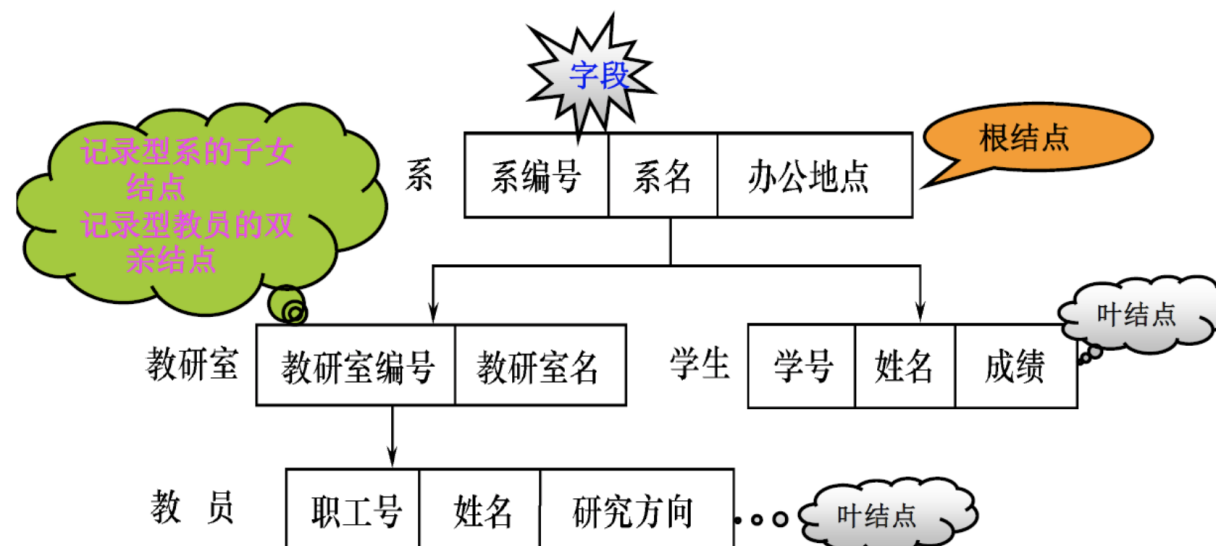
理独立性，简称数据的物理独立性。

- 二级映像的意义：保证了数据库外模式的稳定性，从而从底层保证了应用程序的稳定性，数据与程序之间的独立性，使得数据定义和描述可以从应用程序中分离出去。存取路径由DBMS管理，简化了应用程序的编制，大大减少了维护和修改的负担。

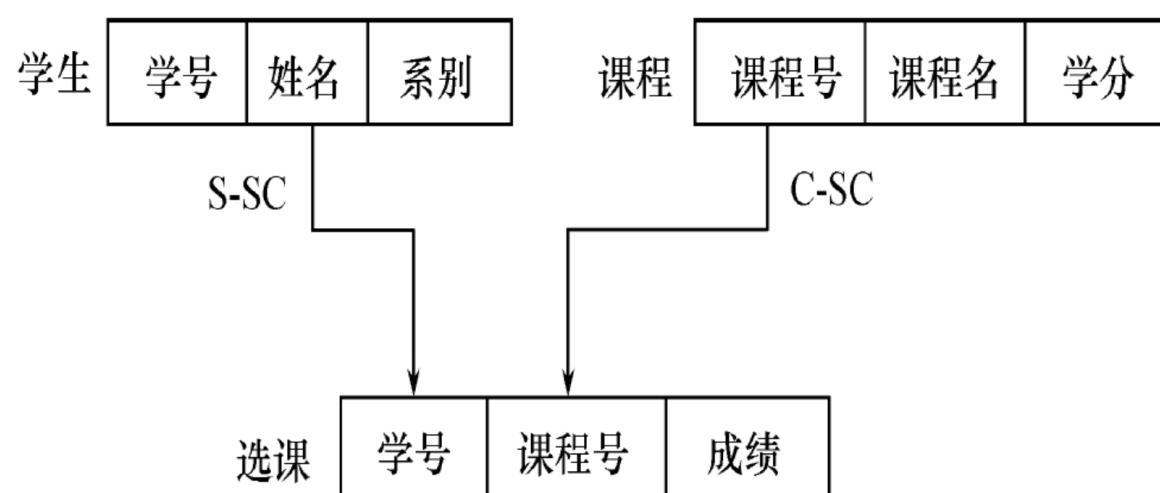
▼ 第二章：数据模型

▼ 数据模型的基本理论

- 数据模型data model是一种模型，是对现实世界数据特征的抽象。**数据模型是数据库系统的核心和基础。**
- 两类数据模型：1. 概念模型，2. 逻辑模型和物理模型
 1. 概念模型：也叫信息模型、E-R模型，它是脱离计算机的、按用户的观点进行数据建模，主要用于数据库设计，表示方法为E-R图。它是现实世界 到信息世界的第一层抽象。
 2. 逻辑模型：按计算机系统的观点进行数据建模，用于DBMS的实现。逻辑模型有五种，分别为层次模型、网状模型、关系模型，面向对象模型等。[CSDN讲解](#)
 3. 物理模型：是对数据最底层的抽象，描述数据在系统内部的表示方法和存取方法。
- 数据模型的三大组成要素
 1. 数据结构：描述数据库的组成对象以及对象之间的联系，是数据模型最重要的方面。如层次结构、网状结构、关系结构等。
 2. 数据操纵：对数据库中各种对象的实例（值）可执行的操作的集合，包括操作及有关的操作规则。数据操纵分查询和更新（插入、删除、修改）两大类。
 3. 数据完整性的约束条件：指的是为了防止不符合规范的数据进入数据库，在用户对数据进行插入、修改、删除等操作时，DBMS自动按照一定的约束条件对数据进行监测，使不符合规范的数据不能进入数据库，以确保数据库中存储的数据正确、有效、相容。
 - 数据完整性(Data Integrity)是指数据的精确(Accuracy)和可靠性(Reliability)。
 - 分为以下四类：
 1. 实体完整性：规定表的每一行在表中是惟一的实体。如职工表将编号作为主关键字，那么，该列不得有空值，否则无法对应某个具体的职工，这样的表格不完整，对应关系不符合实体完整性规则的约束条件。
 2. 域完整性：是指表中的列必须满足某种特定的数据类型约束，其中约束又包括取值范围、精度等规定。
 3. 参照完整性：是指两个表的主关键字和外关键字的数据应一致，保证了表之间的数据的一致性，防止了数据丢失或无意义的数据在数据库中扩散。如在学生管理数据库中，如果将选课表作为参照关系，学生表作为被参照关系，以“学号”作为两个关系进行关联的属性，则“学号”是学生关系的主关键字，是选课关系的外部关键字。选课关系通过外部关键字“学号”参照学生关系。
 4. 用户定义的完整性：不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义的完整性即是针对某个特定关系数据库的约束条件，它反映某一具体应用必须满足的语义要求。例如要求“考查”课的分数以60分或40分计，在用户输入“考查”课的成绩时，要进行检查，以确保满足特定的约束要求。
- 常用的数据模型
 - 层次模型、网状模型、关系模型，面向对象模型是常用的逻辑数据模型。其中，层次模型、网状模型属于格式化模型，现基本已被关系模型取代。
 - 在格式化模型中，实体用记录表示，实体的属性对应记录的数据项，实体之间的联系在格式化模型中转化成记录之间的两两联系。
 - 格式化模型中数据结构的基本单位是基本层次联系（指两个记录以及它们之间的一对一，一对多联系）
 - 关系数据模型是近三四十年来最主流的数据模型，后面课程会细讲。
 - 面向对象模型，是20世纪80年代以来随着面向对象的方法和技术在计算机领域的应用和影响，促进了面向对象模型的发展。
 - 层次模型（典型代表：IBM公司的IMS系统）
 - 数据结构：有且只有一个结点，没有双亲结点，这个结点称为根结点。根以外的其他结点有且只有一个双亲结点。因此，层次模型数据库只能处理一对多联系。每个记录类型可以定义一个排序字段，也称为码字段。任何记录值只有按其路径查看时，才能显出它的全部意义。



- 数据操纵和完整性约束条件：层次模型中的数据操纵主要有插入、删除、更新。在插入时，如果没有对应的双亲结点值，就不能插入它的子女结点值；在删除时，如果删除双亲结点值，则相应的子女结点值也被同时删除。更新操作时，应更新所有相应记录，以保证数据的一致性。
- 优点
 - 层次模型数据结构比较简单清晰
 - 层次数据库的查询效率高
 - 提供了良好的完整性支持
- 缺点
 - 真实世界中很多联系是非层次的，如多对多联系，此时用层次模型表达很困难
 - 当一个结具有多个双亲时，层次模型只能通过引入冗余数据（产生数据不一致），或创建非自然的数据结构（虚拟结点）来解决，对插入和删除操作的限制很多，应用程序的编写困难
 - 查询子女结点必须通过双亲结点
 - 由于结构严密，层次命令趋于程序化
- 网状模型
 - 数据结构：允许两个或者多个结点无双亲，一个结点可以有多个双亲结点。节点之间的对应关系不再是1:n，而是一种m:n的关系克服了层次状数据模型的缺点。



- 数据操纵和完整性约束条件：网状数据模型可以看做是放松层次数据模型的约束性的一种扩展：码唯一标识记录项；保证一个联系中双亲记录和子女记录之间是一对多的联系；
- 优点
 - 网状数据模型可以很方便的表示现实世界中的很多复杂的关系（多对多，多对一）
 - 修改网状数据模型时，没有层次状数据模型的那么多的严格限制，可以删除一个节点的父节点而依旧保留该节点；也允许插入一个没有任何父节点的节点，这样的插入在层次状数据模型中是不被允许的，除非是首先插入的是根节点
 - 实体之间的关系在底层中可以借由指针实现，因此这种模型具有良好的性能，存取效率高

- 缺点
 - 结构复杂，使用不易，随着应用环境的扩大，数据结构越来越复杂，数据的插入、删除牵动的相关数据太多，不利于数据库的维护和重建
 - 网状数据模型数据之间的彼此关联比较大，该模型其实一种导航式的数据模型结构，不仅要说明要对数据做些什么，还说明操作的记录的路径
 - 网状模型的DDL、DML语言复杂，用户不容易使用
 - 记录之间联系是通过存取路径实现的，用户必须了解系统结构的细节，加重了编写程序的负担
- 关系模型
 - 数据结构：
 - 关系数据模型中，无论是是实体、还是实体之间的联系都是被映射成统一的关系 - 一张二维表，在关系模型中，操作的对象和结果都是一张二维表，它由行和列组成
 - 关系型数据库可用于表示实体之间的多对多的关系，只是此时要借助第三个关系 - 表，来实现多对多的关系
 - 关系必须是规范化的关系，即每个属性是不可分割的实体，不允许表中表的存在



| 学 号 | 姓 名 | 年 龄 | 性 别 | 系 名 | 年 级 |
|---------|----------|-----|-----|-----|------|
| 2013004 | 王 未命名_副本 | 19 | 女 | 社会学 | 2013 |
| 2013006 | 黄大鹏 | 20 | 男 | 商品学 | 2013 |
| 2013008 | 张文斌 | 18 | 女 | 法律 | 2013 |
| ... | ... | ... | ... | ... | ... |

- 数据操纵和完整性约束条件：数据操纵包括增删改查，完整性约束条件包括三大类：实体完整性、参照完整性、用户完整性
- 优点
 - 结构简单，关系数据模型是一些表格的框架，实体的属性是表格中列的条目，实体之间的关系也是通过表格的公共属性表示，结构简单明了
 - 关系数据模型中的存取路径对用户而言是完全隐蔽的，是程序和数据具有高度的独立性和保密性，其数据语言的非过程化程度较高
 - 操作方便，在关系数据模型中操作的基本对象是集合而不是某一个元祖
 - 有坚实的数学理论做基础，包括逻辑计算、数学计算等
- 缺点
 - 查询效率低，关系数据模型提供了较高的数据独立性和非过程化的查询功能（查询的时候只需指明数据存在的表和需要的数据所在的列，不用指明具体的查找路径），因此加大了系统的负担
 - 由于查询效率较低，因此需要数据库管理系统对查询进行优化，加大了DBMS的负担；

▼ 相关术语

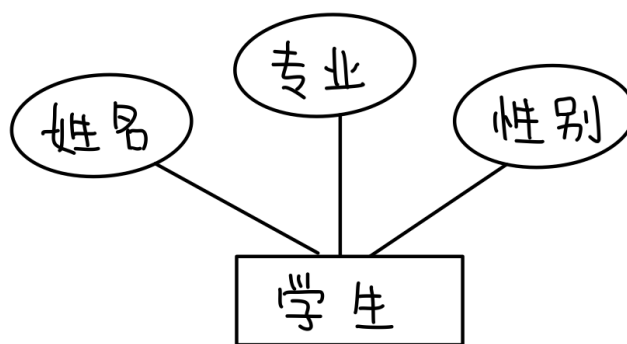
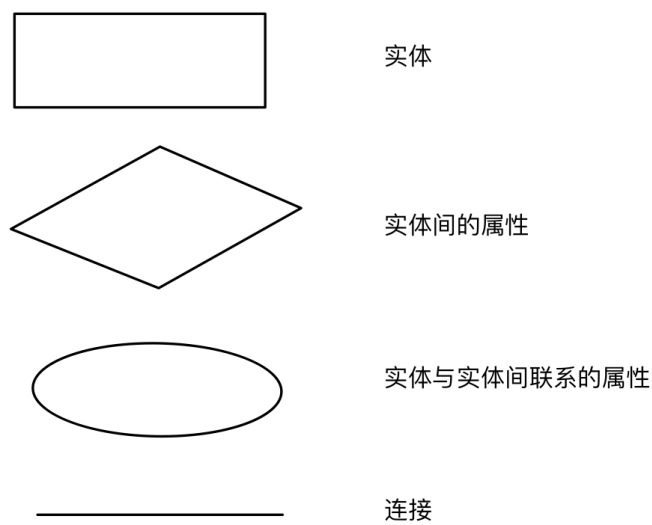
- 关系（Relation）：一个关系对应通常说的一张表
- 元组（Tuple）：表中的一行即为一个元组
- 属性（Attribute）：表中的一列即为一个属性，给每一个属性起一个名称即属性名
- 主码（Key）：也称码键。表中的某个属性组，它可以唯一确定一个元组
- 域（Domain）：是一组具有相同数据类型的值的集合。属性的取值范围来自某个域。
- 分量：元组中的一个属性值。

- 关系模式：对关系的描述，关系名（属性1，属性2，...，属性n），如：学生（学号，姓名，年龄，性别，系名，年级）
- 面向对象数据模型（Object Oriented Data Model）
 - 用面向对象观点来描述现实世界实体（对象）的逻辑组织、对象间限制和联系
 - 数据操纵：增删改查，也包括并发控制、故障恢复、存储管理等完整功能

▼ 概念模型的表示方法：E-R图

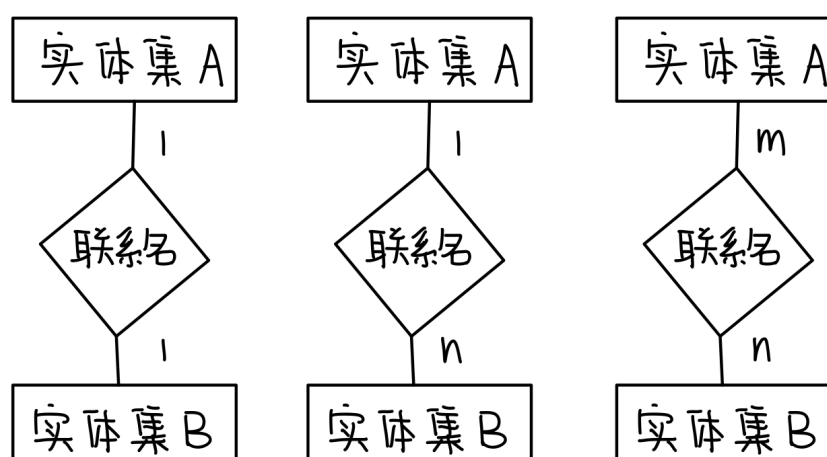
- 相关概念
 - 实体entity：客观存在并可相互区别的事物。在E-R图中，实体用码来区分。
 - 属性attribute：实体所具有的某一特性。一个实体可由若干属性来刻画。
 - 码key：唯一标识实体的属性集。例如学生的学号。
 - 域domain：一组具有相同数据类型的值的集合。如性别的域为（男、女）。
 - 实体型entity type：用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。如学生（学号、姓名、性别）就是一个实体型。
 - 实体集entity set：同一类型实体的集合。如全体学生。
 - 联系relationship：实体（型）内部的联系和实体（型）之间的联系。联系包括1:1联系、1:n联系、m:n联系（多对多）。

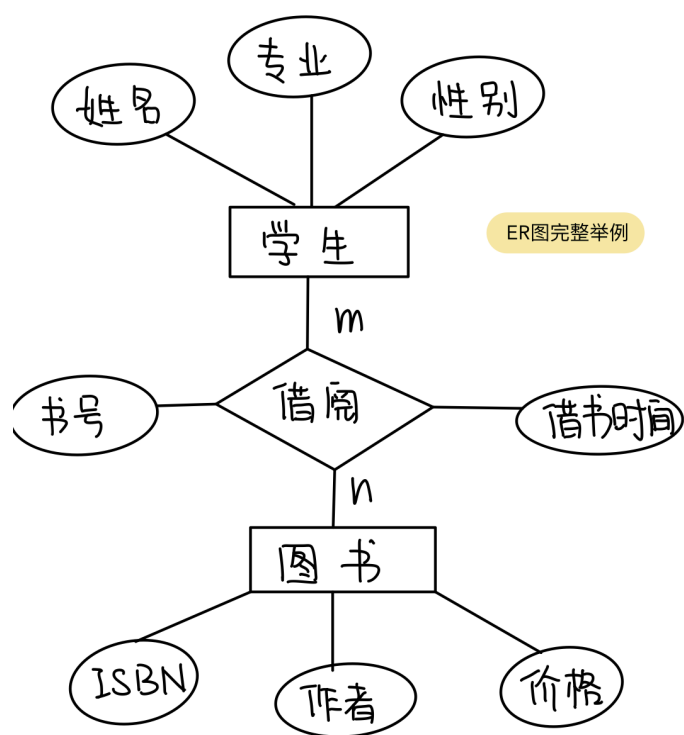
▼ E-R图的表示方法



实体属性举例说明

实体间的联系表示





- ▼ 第三章：关系数据库基本理论
- ▼ 第四章：关系数据库标准语言SQL
 - ▼ SQL语言的特点
 - ▼ 数据定义，数据操纵
 - ▼ 数据更新
 - ▼ 数据查询
 - ▼ 视图
- ▼ 第五章：关系查询处理和查询优化
 - ▼ 查询处理
 - ▼ 查询优化
 - ▼ 代数优化
 - ▼ 物理优化
- ▼ 第六章：数据库安全
- ▼ 第七章：数据库完整性
- ▼ 第八章：数据库恢复技术
- ▼ 第九章：并发控制
- ▼ 第十章：关系数据库设计理论
 - ▼ 规范化理论
 - ▼ 数据库设计