

# 第七章 数据库设计

- 数据库设计概述

- 数据库设计：

- 指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）。它是数据库在应用领域的主要研究课题。

- 数据库应用系统：

- 使用数据库的各类系统统称为数据库应用系统。

- 信息系统和数据库：

- 数据库设计方法

- 数据库设计基本步骤

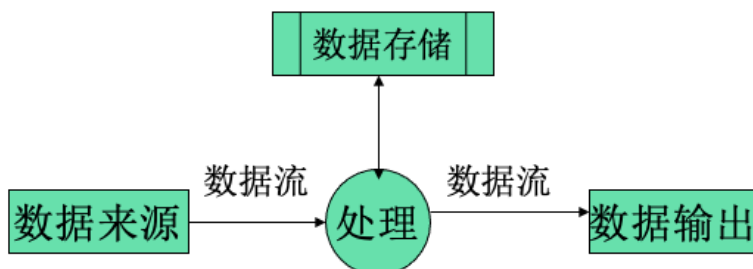
- 数据库设计是在DBMS支持下设计数据库应用系统（如管理信息系统）的过程
    - 是以概念结构设计、逻辑数据库设计和物理数据库设计为核心的规范化设计
    - 六个阶段
      - 需求分析
      - 概念结构设计（构造ER模型）
      - 逻辑结构设计（生成一组关系模式）
      - 物理结构设计（设计具体DBMS下的表、视图、索引）
      - 数据库实施
      - 数据库运行和维护

- 需求分析

- 步骤

- 数据流图和数据字典

- 数据流图：表达了数据和处理过程的关系



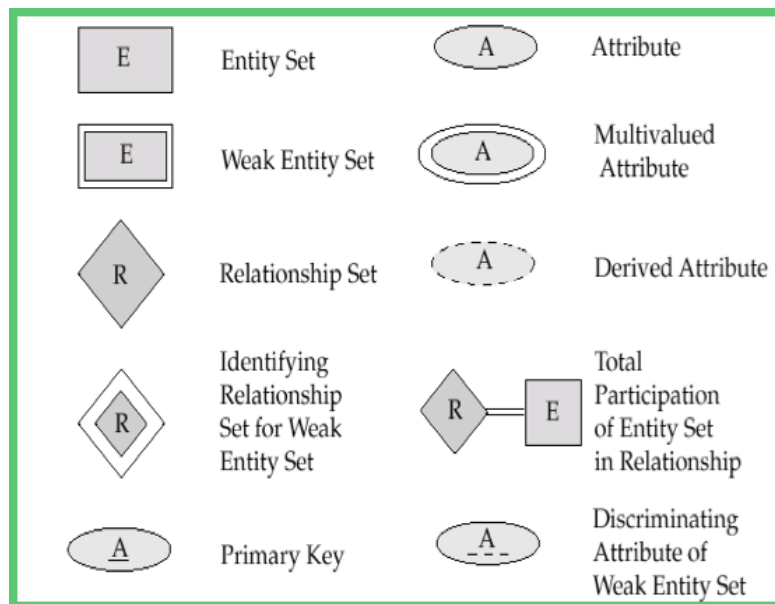
- 数据字典DD

- 数据字典，即描述数据的数据，是各类数据描述的集合，是进行详细的数据收集和数据分析所得的主要结果。
    - DD将数据流图中的每个对象进行具体详尽的描述。

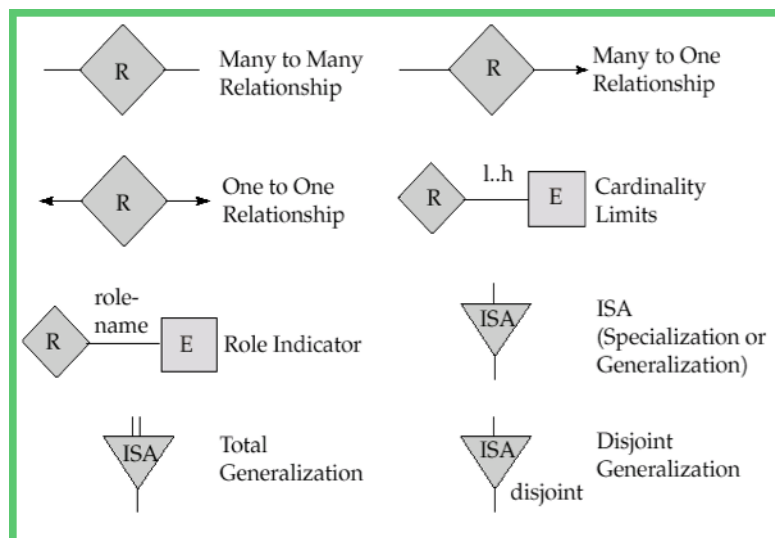
- 包括数据结构，数据项，数据流，数据存储和处理过程描述。它为后续阶段提供了数据和处理的依据。

## • 概念结构设计

- 产生整体数据库的概念结构（概念模式），它是整个组织的各个用户关心的信息结构，也是数据库设计的关键。
- 概念结构是各种数据模型的共同基础，它比数据模型更独立于机器、更抽象，从而更加稳定
- 特点
  - 1、是对现实世界抽象的一个真实模型。
  - 2、易于理解。
  - 3、易于更改和扩充。
  - 4、易于向关系等各种数据模型转换。
- 概念模型的基本概念
  - 实体（强实体、弱实体）
  - 属性（简单属性/复合属性，单值属性/多值属性，导出属性）
  - 码
  - 域
  - 实体型
  - 实体集
  - 联系
    - 基数约束（一对一，一对多，多对多）
    - 参与性约束（全参与，部分参与）
- E-R方法
  - **E-R方法**：Entity-Relationship**实体-联系方法**
  - 该方法用E-R图来描述现实世界的概念模型，是建立概念模型的有用工具。
  - E-R图提供了表示**实体型、属性和联系**的方法：
  - E-R模型记号
    - 1



• 2



## • 数据库概念设计步骤

- Step 1 确定实体集
- Step 2 确定联系集
- Step 3 确定实体和联系的属性
- Step 4 确定属性的域
- Step 5 如果需要，使用特化/泛化等EER表示
- Step 6 检查、消除冗余

## • 逻辑结构设计

### • 任务：

- 把概念结构设计阶段设计好的基本E-R图转换为与选用DBMS产品所支持的数据模型相符合的逻辑结构。
- 完成后应得到系统的关系模型和各个关系的模式结构和各种完整性约束条件（主要反映为主码、外码约束）。

### • 设计步骤：

- E-R图向关系模型的转换原则：

- ER模型到关系模型的9步映射算法

- S1:

- 每一**强实体**用一个**新表**表示
  - 对ER模型中的每一实体E创建对应的新表T
  - 确定新表T的属性：
    - E的所有简单属性转换为对应的表T的属性
    - 对于E的复合属性，直接从其简单成分转换
  - 确定T的主码和Unique码
    - E的码转换为候选码，选定一个（意义最明显的）为主码，其余作Unique码

- S2:

- 处理ER模型中所有参与1:1标识联系的弱实体W
  - 找出W所依赖的强实体对应的表T。
  - 在T表中加入W的所有简单属性和复合属性的简单成分。

- S3:

- 处理ER模型中所有参与1:N或M:N标识联系的弱实体W
  - 为W创建对应的新表T
  - 确定新表T的属性：
    - W的所有简单属性转换为对应的表T的属性
    - 对于W的复合属性，直接从其简单成分转换
  - 确定T的主码
    - 对于1: N联系，T中增加W所依赖的强实体对应表的主码为外码，T的主码为该外码+W的鉴别器（部分码）。
    - 如果是M: N联系，构建新列（具有惟一值）为主码

- S4:

- 处理ER模型中每一二元1:1联系R
  - 确定参与该联系的实体型转换的表S和T
  - 选定S（最好是全参与的一方，推广到参与程度高的一方）
    - 将T的主码作为外码加入S
  - 将联系R的所有简单属性和复合属性成分作为列加入S。

- S5:

- 处理ER模型中每一二元1:N联系R
  - 确定处于N端的实体表S和1端的实体表T
  - 将T的主码作为外码加入S
  - 将R的所有简单属性和复合属性成分作为列加入S

- S6:
  - 处理ER模型中的每一多元联系和二元 (M:N联系) R
    - 为R创建对应的新表T
    - 将R的所有简单属性和复合属性成分作为列加入T
    - 将参与联系的 (强、弱) 实体型的主码作为外码加入T
    - 将 (上步得到的) 所有外码组合, 共同构成T的主码
- S7:
  - 处理ER模型中的每一多值属性A
    - 为A创建对应的新表T
    - 将A的所有简单属性和复合属性成分作为列加入T
    - 将A所属的实体或联系转换得到的表的主码作为外码加入T
    - 将 (上步得到的) 外码和A对应的全部属性确定为T的主码
- S8:
- S9:
  - 处理ER模型中每一相交子类的特化
    - 为超类对应表T
    - 超类的简单属性和复合属性简单成分加入T
    - 确定T的主 (备选) 码
    - 为子类Si创建对应的新表Ti
    - Si的简单属性和复合属性简单成分加入Ti
    - 表T的主码作为外码加入Ti, 并作为Ti主码的一部分
- 关系模式的优化(通常以规范化理论为指导)
  - 并不是规范程度越高的关系就越优, 应根据需要权衡而定(规范化与反规范化达到平衡)。
  - 优化方法:
    - 确定数据依赖
      - (每个关系模式内部各属性之间的数据依赖以及不同关系模式间属性之间的数据依赖 (参照) 。)
    - 对模式之间的数据依赖极小化处理, 消除冗余的联系。
    - 按照数据依赖的理论对每一个关系模式逐一进行分析, 考察是否存在部分函数依赖、传递函数依赖、多值依赖等。确定各关系模式分别属于第几范式。
    - 按照需求分析阶段得到的处理要求, 分析这些模式是否适合, 确定是否要对某些模式进行合并或分解。
    - 对需要处理的模式进行必要的分解, 提高数据操作的效率和存储空间利用率。
      - 横向分解
      - 纵向分解

- 二八率的应用

- 数据库的物理设计

- 任务：
  - 为一给定的逻辑数据模型选取一个最适合应用要求的物理结构，包括数据库在物理设备上的存储结构和存取方法。
- 内容：
  - 为关系模式**选择存取方法**（索引、聚簇或HASH方法）
  - 设计关系、索引等数据文件的**物理存储结构**
    - 确定数据库对象（关系、索引、日志和备份等）的存放位置和存储结构。
    - 确定系统配置参数，如用户数，内存、缓冲区分参数等。
- 评价物理结构：
  - 设计过程中需要对时间效率、空间效率、维护代价和各种用户要求进行权衡，其结果可以产生多种方案，数据库设计人员必须对这些方案进行细致的评价，从中选择一个较优的方案

- 数据库的实施

- 数据库实施包括以下工作：
  - 用DDL定义数据库结构
    - 用选用的DBMS提供的数据库定义语言来严格描述数据库结构，如：SQL。
  - 组织数据入库
    - 是该阶段的主要工作。可以人工方法组织数据入库，可以设计一个数据输入子系统，由计算机辅助数据的入库工作
  - 编制与调试应用程序
    - 应用程序的设计应该与数据设计并行进行，当数据库结构建立好后，就可以开始编制并调试数据库的应用程序。当数据未入库前可以使用模拟数据。
  - 数据库试运行
    - 应用程序编写并调试完毕，并且有一部分数据入库后，就可以开始数据库的试运行，也称作联合调试，目的是实际测量系统的各种性能指标，及时修正前面阶段中的错误。同时必须做好数据库的转储和恢复工作，尽量减少对数据库的破坏

- 数据库的运行和维护

- 运行阶段，数据库经常性的维护工作主要是由DBA完成的，主要工作是调整、修改数据库以及应用程序修改。包括：
  - 数据库的转储和恢复
  - 数据库的安全性、完整性控制
  - 数据库性能的监督、分析和改进
    - 目前许多DBMS产品都提供了监测系统性能参数的工具，DBA可以使用这些工具获得性能参数的值，从而分析并调整某些参数以改进数据库性能。
  - 数据库重组织

- 记录的不断增、删、改，会使数据库的物理存储变差，从而降低空间的利用率和数据的存储效率，使数据库性能下降，这时DBA要对数据库进行重组。它只是按照原设计要求重新安排存储位置，回收垃圾，减少指针，不会改变设计的数据逻辑结构和物理结构。DBMS一般都提供了实用程序帮助重新组织数据库。
- 数据库重构造
  - 当增加新的应用或新的实体，取消某些已有应用，不得不调整数据库的模式和内模式时，如：增加新的数据项，改变数据库的容量，增加或删除索引，修改完整性约束条件等。这就是数据库的重构造。DBMS都提供了修改数据库结构的功能。

以上内容整理于 [幕布文档](#)