

数据库设计

1、数据库设计概述

1) 数据库应用系统的生命周期

数据库应用系统的生命周期分为一下6个阶段：

- 数据库规划

数据库规划的任务是：确定软件的开发目标及可行性。该阶段应该给出问题定义、可行性分析和项目开发计划。

- 需求描述与分析

需求描述与分析是以用户的角度，从系统中的数据和业务规则入手，收集和整理用户的信息，以特定的方式加以描述，是下一步工作的基础。

- 数据库与应用程序设计

数据库的设计是对用户用户数据的组织和存储设计；应用程序设计是在数据库设计基础上对数据操作及业务实现的设计，包括事务设计和用户界面设计。

- 数据库设计实现

数据库设计实现是依照设计，使用DBMS支持的数据定义语言实现数据库的建立，用高级语言编写应用程序。

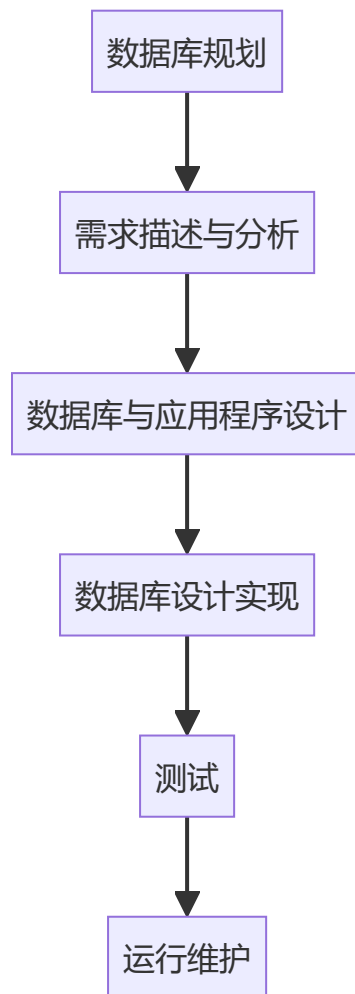
- 测试

测试是在数据库系统投入使用之前，通过精心定制的测试计划和测试数据数据来测试系统的性能是否满足设计要求，以便发现问题。

- 运行维护

数据库应用系统经过测试、试运行后即可正式投入运行。运行维护是投入使用后，必须不断地对其进行评价、调整和修改，直至系统消亡。

数据库应用系统的生命周期六个阶段顺序



2、数据库设计一般的策略

数据库设计的一般策略有两种：自顶向下 和 自底向上。

3、数据库设计的基本步骤

一般将数据库设计分为如下6个步骤：

- 1) 用户需求分析
- 2) 概念结构设计
- 3) 逻辑结构设计
- 4) 物理结构设计
- 5) 数据库实施阶段
- 6) 数据库运行与维护

1、系统需求分析/(用户需求分析):

- 需求分析阶段的任务、方法和目标

需求分析阶段的任务：

综合各个用户的应用需求，对现实世界要处理的对象（组织、部门和企业等）进行详细调查，在了解现行系统的概况，确定新系统功能的过程中，手机支持系统目标的基础数据及方法。

（1）信息要求：用户需要在系统中保存哪些信息，由这些保存的信息要得到什么样的信息，这些信息以及信息间应满足的完整性要求。

（2）处理要求：用户在系统中要实现什么样的操作功能，对保存信息的处理过程和方式，各种操作处理的频度、响应的时间要求、处理方式等以及处理过程中的安全性要求和完整性要求。

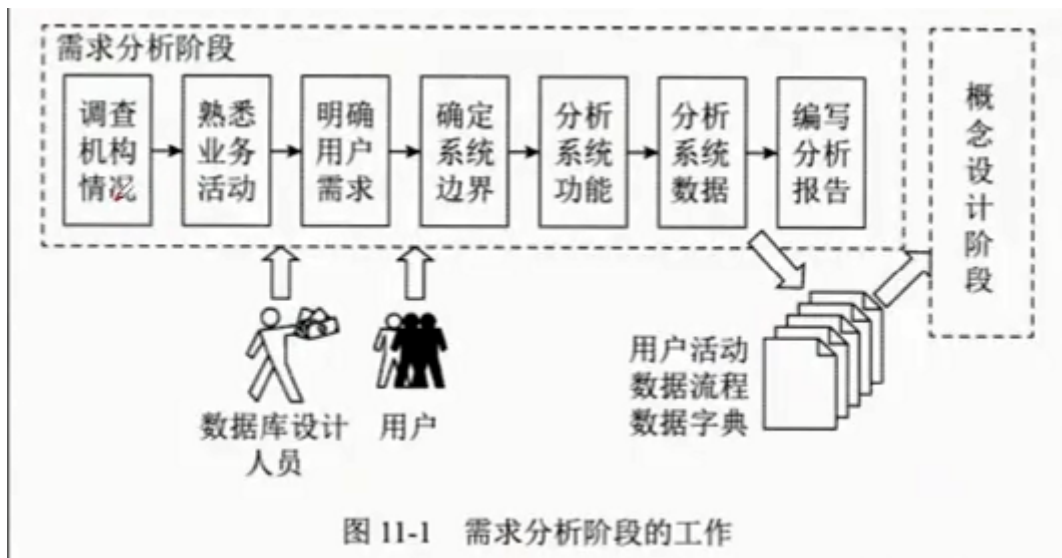
（3）系统要求：包括安全性要求、使用方式要求和可扩充性要求。

（安全性要求：系统有几种用户使用，每一种用户的使用权限如何）

（使用方式要求：用户的使用环境是什么，平均有多少用户同时使用，最高巅峰有多少用户同时使用，有无查询相应的时间要求等）

（可扩充性要求：对未来功能、性能和应用访问的可扩充性要求。）

• 需求分析阶段的文档



需求分析阶段的成果是：**系统需求说明书**，主要包括：

数据流图、数据字典、各种说明性表格、统计输入表、系统功能结构图等

数据字典（Data Dictionary,DD）：是各类数据描述的集合，它是关于数据库中数据的描述，即元数据，而不是数据本身。如用户向数据库输入了说明信息，从数据库中得到什么信息，各类信息的内容和结构，信息之间的联系等。

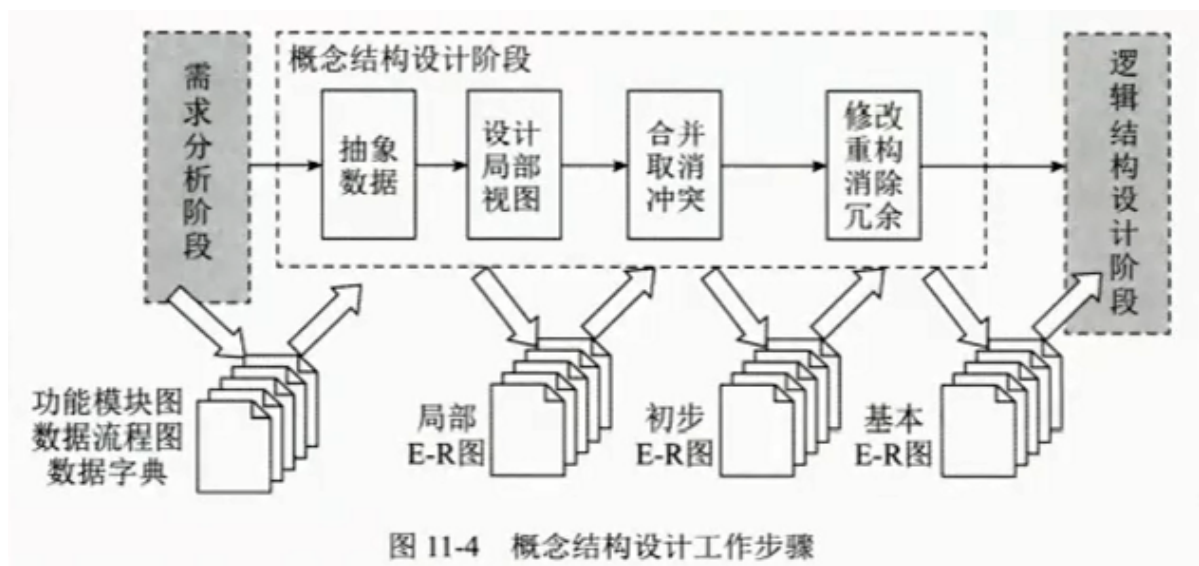
数据字典包括如下5个部分：

数据项、数据结构、数据流、数据存储、处理过程。

2、概念结构设计

- 概念结构设计是在需求分析的基础上，依照需求分析中的信息要求，对用户信息加以分类、聚集和概况，建立信息模型
- 概念结构设计最著名最常用的方法是**实体 - 联系方法**，简称：**E-R方法**。它将现实世界的信息结构统一实体、属性以及实体之间的联系来描述。
- 概念结构设计工作步骤包括：**选择局部应用、逐一设计分E-R图和E-R图合并。**

画**E-R图**是在概念结构设计阶段



1. 选择局部应用：

选择适当层次的数据流程图，让这一层的每一个部分对应一个局部应用，实现某一项功能，从这一层入手，就能很好地设计分E-R图。

2. 逐一设计分E-R图：

划分好各个局部应用后，要对每个局部应用逐一设计分E-R图，又称为局部E-R图。

现实生活中许多事物作为实体还是属性是没有明确界定的，需要根据具体情况而定，一般遵循以下两条准则：

- 1) 属性不可再分，即属性不再具有需要描述的性质，不能有属性的属性。
- 2) 属性不能与其他实体发生联系，联系是实体与实体之间的联系。

3. E-R图合并：

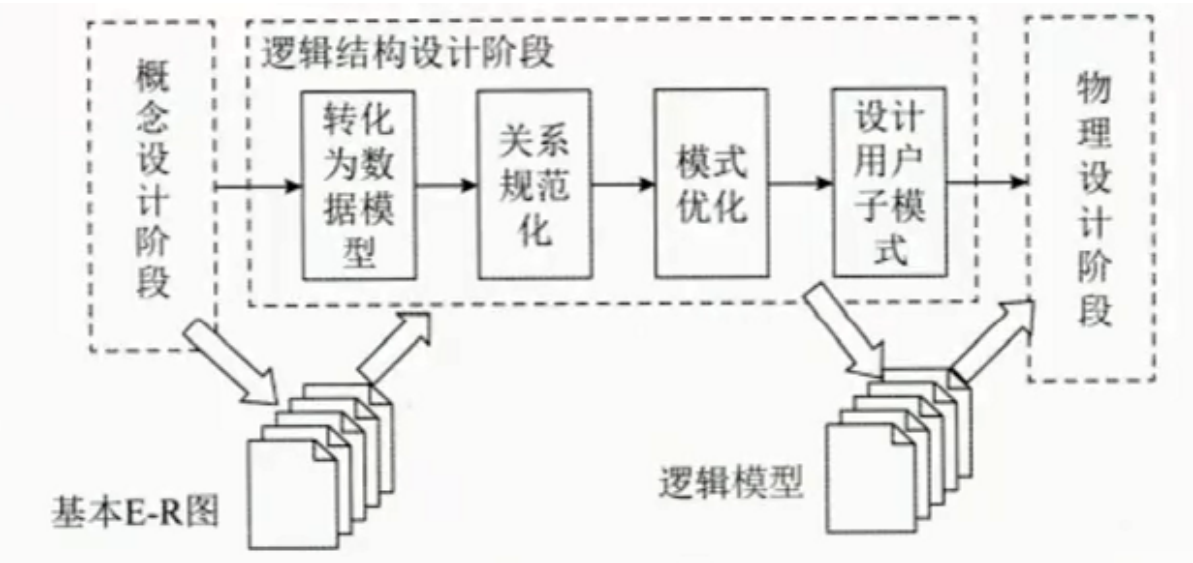
根据局部应用设计好局部E-R图之后，就可以对各分E-R图进行合并。合并的目的在与解决冲突，消除冗余

分E-R图合并时，它们之间存在的冲突主要有一下三类：

- 1) 属性冲突：同一属性可能存在与不同的分E-R图上有着不同的命名，由于设计人员不同或是出发点不同，对属性的类型、取值范围、数据单位等可能会不一致。
- 2) 命名冲突：相同意义的属性，在不同的分E-R图上有着不同的命名，或是名称相同的属性在不同的分E-R图中代表着不同的意义，这也需要进行统一。
- 3) 结构冲突：同一实体在不同的分E-R图中有不同的属性，同一对象在某一分E-R图中被抽象为实体而在另一份E-R图中又被抽象为属性。

3、逻辑结构设计

- 逻辑结构设计就是在概念结构设计的基础上进行数据模型设计，可以是层次模型、网状模型和关系模型。
- 逻辑结构设计阶段主要工作步骤包括：**确定数据模型、将E-R图转换成指定的数据模型、确定完整性约束、确定用户视图**



E-R图向关系模式的转换

1、实体向关系模式的转换

E-R图中的每一个实体单独转换成一个关系模式，实体名对应关系模式的名称，实体的属性转换为关系模式的属性，实体标识符就是关系的码

2、联系向关系模式的转换

一对一的联系转换（1：1）：

通常一对一联系不需要将其转换为一个独立的关系模式，只需要将联系归并到关联的两个实体的任意一方，给待归并的一方实体属性集中增加另一方实体的码和该联系的属性即可，归并后的实体码保持不变。

一对多联系的转换（1：*）：

通常一对多联系也不需要将其转换为一个独立的关系模式，只需要将联系归并到关联的两个实体的多方，给待归并的多方实体属性集中增加一方实体的码和该联系的属性即可，归并后的多方实体码保持不变

多对多的联系转换（*：*）：

多对多联系只能转换成一个独立的关系模式，关系模式的名称取联系的名称，关系模式的属性取该联系所关联的两个多方实体的码及联系的属性，关系的码是多方实体的码构成的属性组。

多对多对多联系转换（*：*：*）：

三方联系的多对多对多（*：*：*）也是一样，只能转换成一个独立的关系模式，关系模式的属性取三方实体的码及联系的属性，关系模式的码为三方实体的码组成的属性组。如：一个供应商可以给多个项目供应多种零件，一个项目可以使用多个供应商供应的多种零件。

关系模式的规范化

由E-R图转换得来的初始关系模式并不能完全符合要求，还会由数据冗余、更新异常存在，这就需要经过进一步的规范化处理。

- (1)根据语义确定各关系模式的数据依赖
- (2)根据数据依赖确定关系模式的范式，判断关系模式是否达到了3NF或4NF
- (3)如果关系模式不符合要求，要进行分解，达到3NF、BCNF或4NF
- (4)关系模式的评价及修正，有时根据处理要求，可能还需要增加部分冗余以满足处理要求，这就需要做部分关系模式的处理，分解、合并或增加冗余属性，提高存储效率和处理效率。

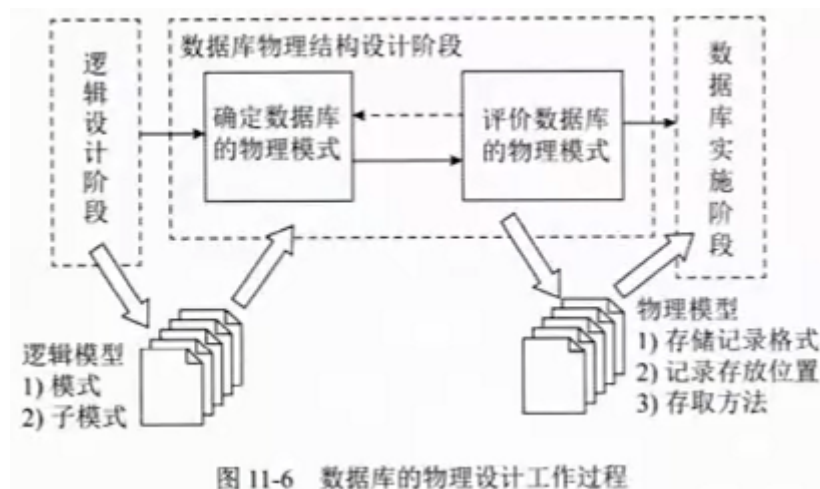
确定完整性约束

用户视图的确定

- 根据数据流图确定处理过程使用的视图。
- 根据用户类别确定不同用户使用的视图。

4、数据库的物理设计

1、数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，它依赖于给定的计算机系统。为一个给定的逻辑数据模型设计一个最合适应用要求的物理结构的过程，就说数据库的物理设计。



2、数据库物理设计工作步骤：

物理设计的主要工作步骤包括：

确定数据分布、存储结构、访问方式

3、确定数据分布

1. 根据不同应用分布数据
2. 根据处理要求确定数据的分布
3. 对数据的分布存储必然会导致数据的逻辑结构的变化，要对关系模式做新的调整，回到数据库逻辑设计阶段做必要的修改。

4、确定数据的存储结构

存储结构具体指数据文件中记录之间的物理结构。

为提高数据的访问速度，通常会采用索引技术。在物理设计阶段，要根据数据处理和修改要求，确定数据库文件的索引字段和索引类型。

5、确定数据的访问方式

1. 存储记录结构设计
2. 存储记录布局
3. 存取方法的设计

5、数据库运行维护与管理

1. 数据库重组和重构

数据库重组是指在不改变数据库逻辑和物理结构的情况下，去除数据库存储文件中的废弃空间以及碎片空间中的指针链，使数据库记录在物理上紧连。

数据库系统运行过程中，会因为一下原因而对数据库的结构做修改，称为数据库重构。重构包括表结构的修改和视图的修改。

2. 数据库系统的审计

审计是一种DBMS工具，它记录数据库资源和权限的使用情况。启用审计功能，可以产生审计跟踪信息，包括哪些数据库对象受到影响，谁在什么时候执行了这些操作。

审计是被动的，它只能跟踪对数据库的修改而不能防止，但作为一个安全性手段，起到对非法入侵的威慑作用，可以据此追究非法入侵者的法律责任。

审计功能的开启会影响系统的性能。

3. 数据库的存储管理

在数据库系统运行过程中，随着数据的不断变更，会影响到系统的响应效率。通过以下手段进行存储管理，可有效的提高系统性能。

- （1）索引文件和数据文件分开存储，日志文件存储在高速设备上。
- （2）适时修改数据文件和索引文件的页面大小。
- （3）定期对数据进行排序。
- （4）增加必要的索引项。

4. 数据安全性管理

- （1）建立网络安全，主要是防火墙的设置
- （2）操作系统级安全，进行登录用户的管理
- （3）DBMS级安全，对访问数据库的用户进行密码验证。
- （4）角色和用户的授权管理
- （5）建立视图和存储过程加强安全性
- （6）使用审计功能，为追究非法入侵者法律责任提供证据，发现安全漏洞

5. SQL语句的编码检验

- （1）尽可能的减少多表查询或建立物化视图。
- （2）以不相关子查询替代相关子查询
- （3）只检索需要的列
- （4）用带IN的条件子句等价替换OR子句
- （5）经常提交COMMIT，以尽早释放锁

6. 数据安全性管理

- (1) 如果频繁的访问是对两个相关的表进行连接操作，则考虑将其合并。
- (2) 如果频繁的访问只是在表中的某一部分字段上进行，则考虑分解表，将该部分单独作为一个表
- (3) 对于更新很少的表、引入物化视图。

7. 索引维护和改进

- (1) 如果查询是瓶颈，则在关系上建立适应的索引，通常在作为查询条件的属性上建立索引，可以提高查询效率。
- (2) 如果更新是瓶颈，每次更新都会重建表上的索引，引起效率的降低，则考虑删除某些索引。
- (3) 选择适当的索引类型，如果是经常使用范围查询，则B树索引比散列索引更高效。
- (4) 将有利于大多数数据查询和更新的索引设为聚簇索引。

8.