

数据库

数据库

绪论

关系数据库

关系数据库标准语言SQL

数据库安全性

数据库完整性

关系数据库理论

数据库设计

数据库编程

关系查询处理和查询优化

数据库恢复技术

并发控制

绪论

1. 1、数据库的 4 个基本概念

- ① 数据 data：描述事物的符号，数据库中存储的基本对象
- ② 数据库 DataBase, DB：长期存储在计算机内的、有组织的、可共享的大量数据的集合
- ③ 数据库管理系统 DataBase Management System, DBMS：位于用户与操作系统之间的一层数据管理软件
- ④ 数据库系统 DataBase System, DBS：由数据库、数据库管理系统、应用程序和数据库管理员（DBA）组成

2. 2、数据库系统的特点

- ① 数据结构化
- ② 数据的共享性高，冗余度低而且容易扩充
- ③ 数据独立性高：物理独立性（应用程序与物理存储相互独立，数据的物理存储改变，应用程序不改变），逻辑独立性（应用程序与逻辑结构相互独立，数据的逻辑结构改变，应用程序不改变）
- ④ 数据由数据库管理系统统一管理和控制

3. 3、数据模型

数据模型包括概念模型（信息模型）和逻辑模型/物理模型

概念模型按用户的观点建模，用于数据库设计，表示方法有 E-R 模型

逻辑模型按计算机的观点建模，用于 DBMS 实现，包括网状模型、层次模型、关系模型等等

物理模型是对数据最底层的抽象

4. 4、数据模型的组成要素：数据结构、数据操作、数据的完整性约束条件

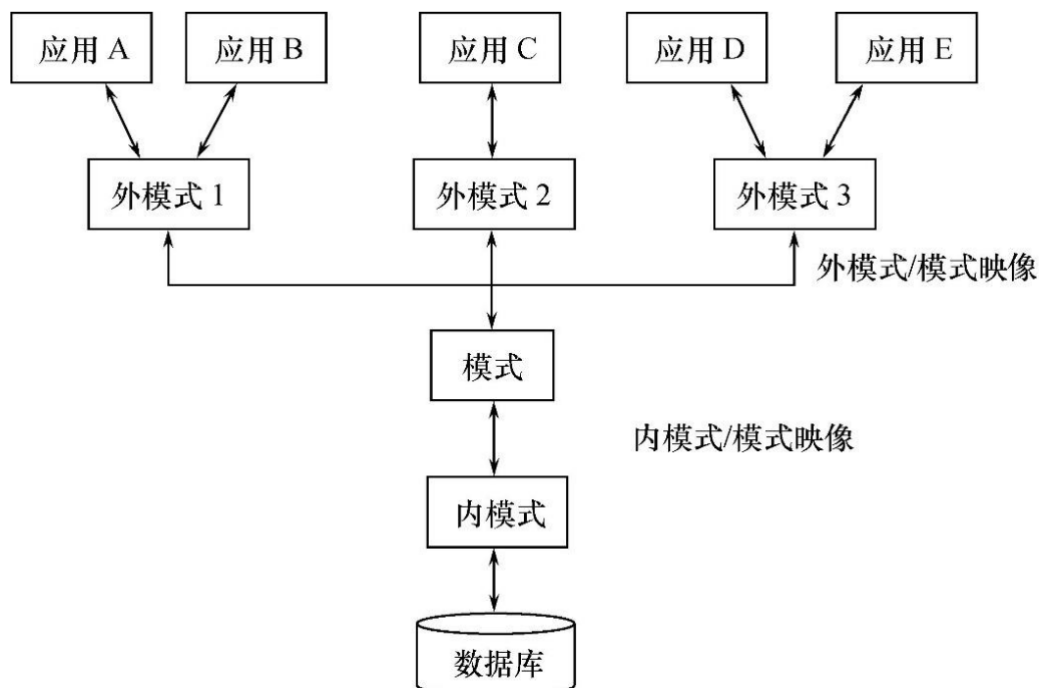
数据模型是严格定义的一组概念的集合，需要精确描述系统的静态特性、动态特性和完整性约束条件

因此数据模型由以下三个部分组成：

- ① 数据结构：描述系统的静态特性→描述数据库的组成对象以及对象之间的联系
- ② 数据操作：描述系统的动态特性→是对数据库中各种对象的实例所允许的操作的集合，其类型有查询和更新（增删改）
- ③ 数据的完整性约束条件：用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效与相容，完整性约束条件是一组完整性规则的集合

5.

5、数据库系统的三级模式结构：外模式、模式（逻辑模式）、内模式



- ① 外模式（子模式/用户模式）：数据库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，数据库用户的数据视图，是与某应用有关的数据的逻辑表示
- ② 模式（逻辑模式）：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
- ③ 内模式（存储模式）：数据物理结构和存储方法的描述，是数据在数据库内部的组织方式

6、数据库的二级映像功能与数据独立性

- ① 外模式/模式映像：保证了数据的逻辑独立性
当模式改变时，对外模式/模式的映像做出改变，保证外模式不变，应用程序由外模式编写，从而应用程序不变
- ② 模式/内模式映像：保证了数据的物理独立性
当数据库的存储结构改变时，对模式/内模式映像做出改变，使模式不变，进而应用程序不变

7.

关系数据库

关系数据库标准语言SQL

数据库安全性

1. 1、不安全因素

- ① 非授权用户对数据库的恶意存取和破坏
- ② 数据库中重要数据的泄露
- ③ 安全环境的脆弱性

2. 2、数据库安全性控制

- ① 用户身份鉴别：静态口令鉴别、动态口令鉴别、生物特征鉴别、智能卡鉴别
- ② 存取控制
- ③ 自主存取控制方法

3. 3、为什么要授权

授权是指授予（GRANT）和收回（REVOKE），是数据库安全性控制中的自主存取控制方法。是为了保护数据库，防止不合法使用所造成的数据泄露、更改或破坏

4. 4、如何授权：授予 GRANT

发出如下语句的可以是数据库管理员、也可以是数据库对象创建者、也可以是已经拥有该权限的用户

GRANT 权限 ON 对象类型 对象名 TO 用户名 [WITH GRANT OPTION];

权限：查询权限 SELECT，全部操作权限 ALL PRIVILEGES

对象类型&对象名：对象类型可以是 TABLE 也可以是 VIEW，对象名为对应的表名或者视图名

用户名：可以是指定用户，也可以是全体用户 PUBLIC

如果没有指定 WITH GRANT OPTION 子句，则获得某种权限的用户只能使用该权限，不能传播该权限

注意：SQL 不允许循环授权，即被授权者不能把权限再授回给授权者或其祖先

eg. 把查询权限授给用户 U1

GRANT SELECT ON TABLE Student TO U1;

5. 5、收回授权：收回 REVOKE

REVOKE 权限 ON 对象类型 对象名 FROM 用户名 [CASCADE|RESTRICT];

CASCADE：级联回收。将用户传播出去的权限一并收回

RESTRICT：受限回收。若用户传播过该权限，回收将会失败

eg. 把用户 U4 修改学生学号的权限收回

REVOKE UPDATE(Sno) ON TABLE Student FROM U4;

6. 6、创建数据库模式的权限

对创建数据库模式一类的数据库对象的授权再数据库管理员创建用户时实现：

CREATE USER username [WITH DBA|RESOURCE|CONNECT];

只有系统的超级用户才有权创建一个新的数据库用户，新创建的数据库用户有三种权限：DBA、RESOURCE、CONNECT（此为默认）

DBA：可以创建新用户、创建模式、创建基本表和视图等，拥有对所有数据库对象的存取权限，还可以把这些权限授予一般用户

RESOURCE：可以创建基本表和视图，但不能创建模式和新用户。数据库对象的属主可以用 GRANT 语句把该对象上的存取权限授予其他用户

CONNECT：只能登陆数据库，根据其他用户或者数据库管理员授予的权限的情况对数据库对象进行权限范围内的操作

7. 7、数据库角色

角色是权限的集合。可以为一组具有相同权限的用户创建一个角色，使用角色来管理数据库权限可以简化授权的过程

创建角色：CREATE ROLE 角色名;

在一个角色中添加角色或者用户：GRANT 角色 TO 某角色/某用户 [WITH ADMIN OPTION];

指定 WITH ADMIN OPTION 子句，则获得某种权限的角色或者用户还可以把这种权限再授予其他角色

给角色授权：GRANT 权限 ON 对象类型 对象名 TO 角色;

角色权限收回：REVOKE 权限 ON 对象类型 对象名 FROM 角色;

8. 8、视图机制

为不同的用户定义不同的视图，把不需要的数据隐藏，这样用户就不会误操作

9. 9、审计

把对数据库的所有操作都记录到审计日志中，然后就可以通过日志审查这个里面是否有一些非法的行为

10. 10、数据加密

通过一些加密算法，把明文变成密文。这样别人就无法查看

数据库完整性

1. 1、数据库的完整性是指数据的正确性和相容性

正确性：符合现实世界语义

相容性：同一对象在不同表中的数据符合逻辑

2. 维护完整性需要实现的功能

- ① 提供定义完整性约束条件的机制
- ② 提供完整性检查的方法
- ③ 进行违约处理

3. 三大完整性

① 实体完整性：主码唯一且非空

检查与违约处理：检查主码是否唯一，如果不唯一则拒绝插入或修改；检查主码各个属性是否为空，只要有一个为空就拒绝插入或修改

② 参照完整性：外码的约束

检查：对被参照表和参照表进行增删改操作时可能会破坏参照完整性，必须检查以保证两个表的相容性

违约处理：外码约束可以在创建参照表时说明不同的违约策略

可能破坏参照完整性的情况及违约处理

被参照表（例如Student）	参照表（例如SC）	违约处理
可能破坏参照完整性 ←	插入元组	拒绝
可能破坏参照完整性 ←	修改外码值	拒绝
删除元组 →	可能破坏参照完整性	拒绝/级联删除/设置为空值
修改主码值 →	可能破坏参照完整性	拒绝/级联修改/设置为空值

③ 用户定义完整性：属性上约束条件的定义

1. 属性上的约束条件

非空 NOT NULL

列值唯一 UNIQUE

检查列值是否满足一个表达式 CHECK

检查和违约处理：当向表中插入元组或者修改属性值时，关系数据库管理系统将检查属性上的约束条件是否被满足，如果不满足则操作被拒绝

2. 元组上的约束条件

在 CREATE TABLE 语句中可以用 CHECK 短语定义元组上的约束条件，元组级的限制可以设置不同属性之间的取值的相互约束条件

检查和违约处理：当向表中插入元组或者修改属性值时，关系数据库管理系统将检查元组上的约束条件是否被满足，如果不满足则操作被拒绝

关系数据库理论

数据库设计

1、需求分析

1. 调查机构情况与熟悉业务活动，明确用户的需求，确定系统的边界，生成数据字典和用户需求规格说明书

2、概念结构设计

2. 将需求分析得到的用户需求抽象为概念模型，绘制 E-R 图

3、逻辑结构设计

3. 将 E-R 图转换为与 DBMS 相符合的逻辑结构（包括数据库模式和外模式），例如 E-R 图向关系模型转换，再根据规范化理论对数据模型进行优化，设计用户子模式

4、物理结构设计

4. 通常关系数据库物理设计的内容包括关系模式选择存取方法、以及设计关系、索引等数据库文件的物理存储结构

5、数据库实施

5. 建立实际数据库结构、试运行、装入数据

6. 数据库运行和维护

6. 维护数据库的安全性、完整性控制以及系统的转储和恢复；性能的监督、分析和改进；增加新功能；发现错误和修改错误

数据库编程

- 1.

关系查询处理和查询优化

数据库恢复技术

1、事务的概念

1. ① 事务是数据库操作序列，这些操作要么全做，要么全不做，是不可分割的工作单位。一个事务可以是一个 sql 语句，一组 sql 语句，或者整个程序。一般来说，一个程序包含多个事务。
② 事务是恢复的基本单位，也是并发控制的基本单位。

2、定义事务的 SQL 语句

开始语句：BEGIN TRANSACTION;

2. 结束语句：COMMIT; //表示提交事务的所有操作
或者 ROLLBACK; //表示回滚，即事务运行中发生了某种故障，需要对已完成的操作进行撤销，回滚到事务开始的状态。

3、事务的 4 个特性 (ACID)：原子性，一致性，隔离性，持续性

- ① 原子性：一个事务要么全做，要么全不做
② 一致性：事务执行的结果必须一致。如果一个事务被迫中断，即事务中的一些操作做了，剩下的还没做，且做了
3. 的对数据库造成的修改已经存入物理数据库的话，就会造成数据库处于不一致的状态。因此事务要么全做，要么全不做，一致性和原子性是密切相关的
③ 隔离性：一个事务的执行不能被其他事务干扰
④ 持续性（也称为永久性）：一个事务一旦提交，它对数据库中数据的改变应该是永久性的

4、数据库系统的故障种类：事物内部的故障、系统故障、介质故障、计算机病毒

- ① 事务内部的故障
事务内部的故障有的是可以通过事务程序本身发现的，有的是非预期的、不能由事务程序处理的
事务故障的情况下，数据库可能处于不正确的状态。恢复程序要在不影响其他事务运行的情况下，强行回滚该事务，即撤销该事务造成的修改，这类恢复操作称为 UNDO（事务撤销）
② 系统故障（软故障）
4. 系统故障是指造成系统停止运转的任何事件，使得系统要重新启动（例如硬件错误、操作系统故障、DBMS 代码错误、系统断电等等）
系统故障的情况下，在系统重新启动后，恢复子系统除了需要撤销未完成的事务以外，还需要 REDO（事务重做）所有已提交的事务
③ 介质故障（硬故障）
外存故障，如磁盘损坏、磁头碰撞、瞬时强磁场干扰等
④ 计算机病毒

并发控制