

数据库系统概论

教师: 梁敬东

01章 绪论

日期 2017.6

数据库定义: ①长期储存在计算机内 ②有组织的 ③可共享的 ④大量的数据的集合

数据库4个基本概念: ① Data, ② DB, ③ DBMS, ④ DBS

数据库系统

DataBase Management System 数据库管理系统

DataBase System

DBMS的主要功能: ①数据定义功能 ②数据组织、存储和管理 ③数据操纵功能

④数据库的事务管理和运行管理 ⑤数据库的建立和维护功能(实用程序)

DBS { DB
DBMS
DBA (DataBase Administrator 数据库管理员)
应用系统

DBMS的定义: ①位于用户与操作系统之间的一层数据管理软件

②是基础软件, 是一个大型复杂的软件系统

人工管理阶段 → 文件系统阶段 → 数据库系统阶段

数据库系统的特点: ①整体数据的结构化

②数据的共享性高, 冗余度低且易扩充

③数据独立性高 → { 物理独立性
逻辑独立性

④数据由数据库管理系统统一管理和控制

两类数据模型: ①概念模型

②逻辑模型和物理模型

三个世界: 现实世界, 信息世界, 机器世界

两类数据操作: 查询, 更新(插入, 删除, 修改)

逻辑数据模型: 层次模型, 网状模型, 关系模型

数据库系统的核心和基础: 数据模型

数据的完整性约束条件(关系模型) ~~①~~ 实体完整性

实体: 客观存在并可相互区别的事物 ~~②~~ 参照完整性

属性: 实体所具有的某一特性 ~~③~~ 用户自定义的完整性

码: 唯一标识实体的属性称为码

实体之间的联系 = ① 一对一 ② 一对多 ③ 多对多

关系数据模型

优点: ① 建立在严格的数学概念上

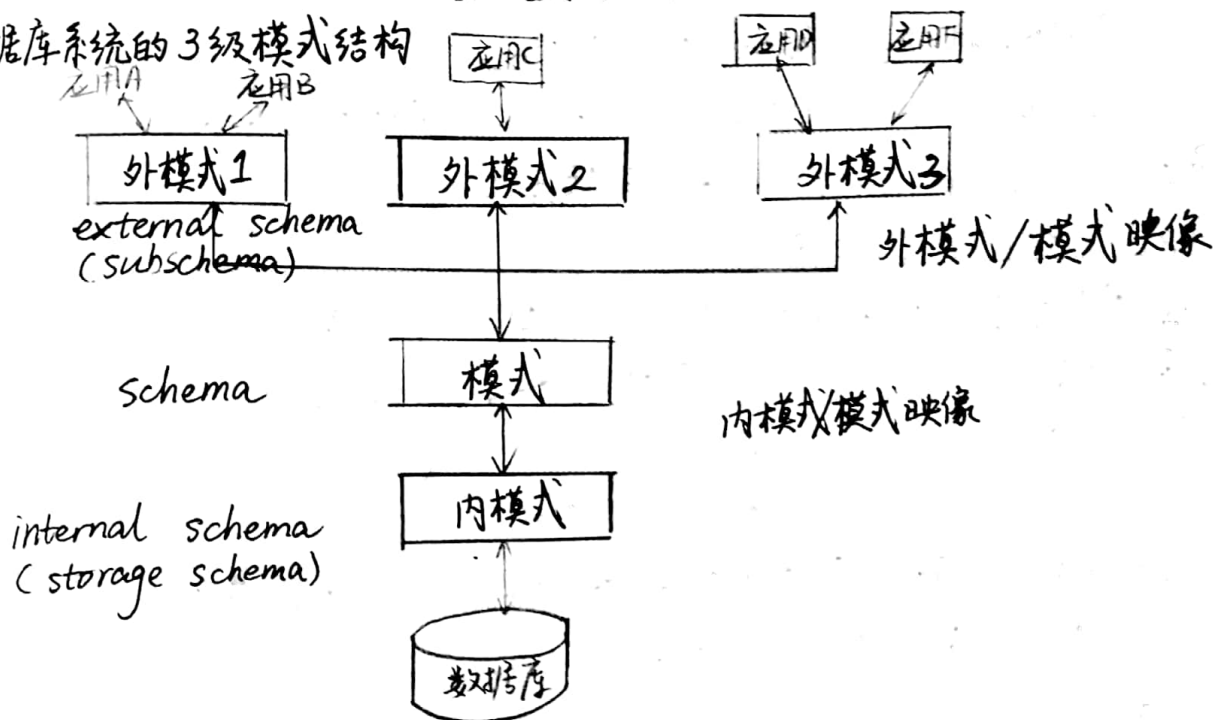
② 概念单一

③ 关系模型的存储路径对用户透明

缺点: ① 查询效率不如非关系模型

② 要对查询请求优化以提高性能, 增加DBMS开发难度

数据库系统的3级模式结构



数据库的三级模式结构 { 外模式 (用户模式/子模式)
模式 (一个数据库只有一个模式)
内模式 (存储模式)

数据库的二级映像 { 外模式/模式映像
内模式/模式映像

数据库三级模式结构的优点:

① 数据与程序独立, 数据的定义和描述可以从应用程序中分离

② 数据存取由DBMS管理, 简化了应用程序的编写, 减少其维护和修改

DBA的职责: ① 决定数据库内容 ② 决定DB的存储结构和存取策略

Administrator ③ 定义数据的安全性要求和完整性约束条件 ④ 运行维护 ⑤ 改进重构

02章 关系数据库

数据模型的3个要素：①关系数据结构②关系操作③集合和关系完整性约束

域(domain)：一组具有相同数据类型的值的集合 (把所有情况列出)

笛卡尔积(Cartesian product) $D_1 \times D_2 \times D_3 = \{(x, x, x), (\dots) \dots\}$

关系(relation)： $D_1 \times D_2 \times \dots \times D_n$ 的子集叫做在域 D_1, D_2, \dots, D_n 上的关系，记为 $R(D_1, D_2, \dots, D_n)$

候选码(Candidate key)：若关系中的某一属性组的值能唯一地标识一个元组，而其子集不能，则称该属性组为候选码。

主码(primary key)：若一关系有多个候选码，选定其中一个为主码

主属性(Prime attribute)：候选码的诸属性

关系模式 $R(U, D, DOM, F)$

5种基本关系操作：①选择②投影③并④差⑤笛卡尔积

关系模型中的
三类完整性约束

{ ①实体完整性
②参照完整性
③用户自定义的完整性 } > 两个不变性

外码：设F是关系R的一个或一组属性，但不是关系R的码， K_s 是关系S的主码，若F与 K_s 相对应，称F是R的外码(foreign key)

F是R的外码

$R(K_r, \textcircled{F}, \dots)$

参照关系

$S(K_s, \dots)$

被参照关系(目标关系)

参照完整性规则：R中每个元组在R的外码F上的值必须：

要么为空值，要么等于S中某个元组的主码值

传统的集合运算
(4种)

{ ①并(union) $R \cup S = \{t \mid t \in R \vee t \in S\}$
②差(except) $R - S = \{t \mid t \in R \wedge t \notin S\}$
③交(intersection) $R \cap S = \{t \mid t \in R \wedge t \in S\}$
④笛卡尔积(Cartesian product) (广义的笛卡尔积)
 $R \times S = \{tr ts \mid tr \in R \wedge ts \in S\}$

专门的关系运算：选择 σ ，投影 π ，连接 \bowtie ，除 \div

选择 selection (限制 restriction)

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t) = \text{'真'}\}$$

σ 要查的满足条件 (要查的东西)

eg $\sigma_{\text{Sage} < 20}(\text{Student})$

投影 projection

$$\pi_A(R) = \{t[A] \mid t \in R\}$$

$$\pi_{A_1, A_2}(R)$$

eg $\pi_{\text{Sname}, \text{Sdept}}(\text{Student})$

连接 (join)

$$R \bowtie_{A \theta B} S = \{tr \mid tr \in R \wedge ts \in S \wedge tr[A] \theta ts[B]\}$$

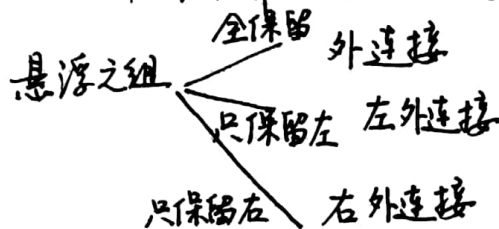
θ

等值连接 $R \bowtie S$

自然连接是 $R.B=S.B$ 一种特殊的等值连接

(会在结果中把重复的属性列去掉)

* 连接过程中) 被舍弃的之组称为悬浮之组



除法 (division)

通过一个东西找另外一个东西 (看到“至少”, 多用除法)

$R \div S = T$. 则 T 包含所有在 R 但不在 S 中的属性及其值且 T 的之组与 S 的之组的所有组合都在 R 中。

eg. 查询至少选修 1 号课程和 3 号课程的学生号码

$$\pi_{\text{Sno}, \text{Cno}}(SC) \div K$$

$$= \{201215121\}$$

K
Cno
1
3

数据字典 P213

数据字典是进行详细的数据收集和数据分析所获得的主要成果

数据字典通常包括: ① 数据项 ② 数据结构 ③ 数据流

④ 数据存储 ⑤ 处理过程

B+树索引存取方法选择的一般规则 P236

(或组合索引)

- ① 若一个(或一组)属性经常在查询条件中出现,则考虑在这个(或这组)属性上建立索引
- ② 若一个属性经常作为最大值等聚集函数的参数,则考虑在这个属性上建索引
- ③ 若一个(或一组)属性经常在连接操作的连接条件中出现,考虑在这个属性上建索引

查询优化中代数优化的基本原则 P284

- ① 选择运算尽可能先做 (最重要最基本)
- ② 投影运算和选择运算同时进行
- ③ 投影同其前或其后的双目运算结合起来
- ④ 把某些选择同在它前面要执行的笛卡尔积结合起来成为一个连接运算
- ⑤ 找出公共表达式

数据库设计的基本步骤 P207 (6个阶段)

- 概念
- ① 需求分析阶段
 - ② ~~数据库~~结构设计阶段
 - ③ 逻辑结构设计阶段
 - ④ 物理结构设计阶段
 - ⑤ 数据库实施阶段
 - ⑥ 数据库运行和维护阶段

ODBC的工作原理(4个部分) P260

- ① 用户应用程序
- ② ODBC驱动程序管理器
- ③ 数据库驱动程序
- ④ ODBC数据源管理

游标的概念 P247

游标是系统为用户开设的一个数据缓冲区,存放SQL语句的执行结果

数据库系统的概念

数据库系统是指在计算机系统中引入数据库后的系统,一般由数据库,数据库管理系统,应用系统,数据库管理员组成

数据库的完整性 P45

① 数据库的正确性与相容性 包括完整性约束机制, 完整性检查机制和违背完整性约束条件应采取的预防措施

② 有实体完整性, 参照完整性和用户自定义的完整性

实体完整性 P45

基本关系的主属性不能取空值

外码的概念 P46

关系R中的一个属性组, 它不是R的主属性, 但它与另一个关系S中的主码相对应, 则这个属性组为R的外码。

用户自定义的完整性 P48

用户自定义的完整性是用户为反映某一具体应用所涉及的数据必须满足的语义要求, 设置的某一具体关系数据库的约束条件, 由关系数据库模型提供的定义机制和检查机制支持。

参照完整性 P48

关系R中的每个元组在R的外码F上的值必须:

要么为空值, 要么等于R的被参照关系S中某个元组的值

死锁 P315

① 事务T₁封锁了数据R₁, T₂封锁了数据R₂ ② T₁又请求封锁R₂, 因T₂已封锁R₂, 于是T₁等待T₂释放R₂上的锁 ③ 接着T₂又申请封锁R₁, 因T₁已封锁了R₁, T₂也只能等待T₁释放R₁上的锁
这样T₁在等待T₂, 而T₂又在等待T₁, T₁和T₂两个事务都不能结束形成死锁。

模式 P29

是数据库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

外模式 (用户模式) P29

是数据库用户能够看见和使用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

内模式 P29 (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。

主码: 表中的某个属性组, 它可以唯一的确定一个元组

数据库阶段数据管理的特色 P11

- ① 数据结构化
- ② 数据共享性高, 冗余度低且易扩充
- ③ 数据独立性高
- ④ 数据由数据库管理系统统一管理和控制

函数依赖 P180

设 $R(U)$ 是一个属性集 U 上的关系模式, X 和 Y 是 U 的子集, 若对于 $R(U)$ 的任意一个可能的关系 r , r 中不可能存在两个元组在 X 上的属性值相等, 而在 Y 上的属性值不等, 则称 " X 函数确定 Y " 或 " Y 函数依赖于 X ", 记作 $X \rightarrow Y$.

视图

视图是从一个或几个基本表导出的表, 视图本身不独立存储在数据库中, 是一个虚表, 可以在视图上再定义视图.

触发器 P168

触发器 (trigger) 是用户定义在关系表上的一类由事件驱动的特殊过程, 它的执行由事件来触发, 是保证数据完整性的一种方法.

数据抽象的三种方法

- ① 分类
- ② 聚集
- ③ 概括

不好的关系模式存在的问题 P179

- ① 数据冗余
- ② 更新异常
- ③ 插入异常
- ④ 删除异常

X 锁 (写锁) exclusive locks : 加了 X 锁后, 不能加任何锁

S 锁 (读锁) share locks : 加了 S 锁后, 只能再加 S 锁

并发控制的主要技术: ① 封锁 ② 时间戳 ③ 乐观控制法 ④ 多版本并发控制

并发操作带来的数据不一致性: ① 丢失修改 ② 不可重复读

③ 读脏数据

—— 读后再读
—— rollback

封锁对象的大小称为封锁粒度

故障的种类 P295

- ① 事务内部的故障
- ② 系统故障
- ③ 介质故障
- ④ 计算机病毒

定义外码的格式

foreign key (Sno) references Student(Sno)
建视图

create view "" as select 语句

建立数据字典的时机：需求分析阶段

数据库恢复要涉及的技术：① 数据转储 ② 登记日志文件 (logging)

where 子句中不能用聚集函数作为条件表达式，只能用 having

求各个课程号及相应的选课人数

select Cno, count(Sno) from SC Group by Cno.