

数据库系统 期末速通教程

1. 前言

1.1 文件系统的问题

[ACID] ① 原子性(atomicity); ② 一致性(consistency); ③ 隔离性(isolation); ④ 持久性(durability).

[文件系统的问题]

- (1) 不保证不备份时数据不丢失.
- (2) 不支持查询语言.
- (3) 尽管数据位置已知, 也无高效的查找数据的方法.
- (4) 应用依赖于数据的定义(结构), 改变数据的定义会影响应用程序.
- (5) 数据的视图单一, 只有按物理存放的视图.
- (6) 不同应用有不同的文件, 难支持新应用.
- (7) 对多访问的控制有限.
- (8) 无对数据的集中控制.
- (9) 文件常为一个特定的应用而创建.
- (10) 文件独立创建和管理.
- (11) 常有数据冗余(redundancy)和不一致(inconsistency).
- (12) 缺少并发控制, 不能防止并发请求间相互干扰.

[DB 的优点]

- (1) 控制数据冗余.
- (2) 多用户共享数据.
- (3) 提供数据的访问控制.
- (4) 对不同用户提供不同接口.
- (5) 表示数据间的复杂关系.
- (6) 保证数据完整性.
- (7) 提供备份和还原操作.
- (8) 有通用标准.
- (9) 能适应数据的结构的变化.
- (10) 减少应用开发时间.
- (11) 保证最新信息的可用性.

(12) 因经营规模扩大而得到的经济节约.

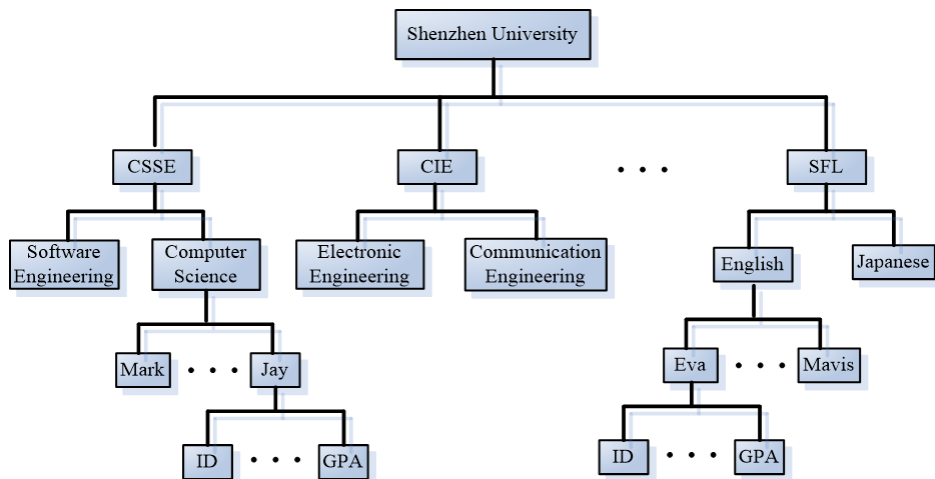
(13) 提供并发控制.

1.2 不同类型的 DB

[分层数据库系统, Hierarchical DBS, HDBS]

(1) 数据组织为树, 数据间用指针连接, 父子数据间存在一对多的关系.

(2) 示意图:



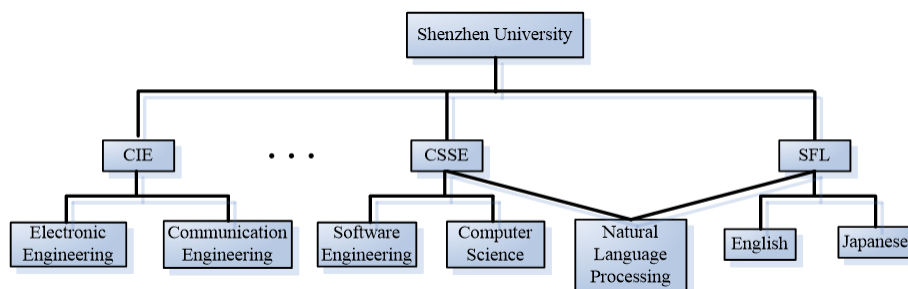
(3) 缺点:

- ① 查询复杂: 需要复杂的查询语言或递归查询.
- ② 数据冗余: 子树中的节点信息在多个地方使用时, 更新操作复杂, 存储空间大.
- ③ 难保证数据一致性: 修改一个节点的数据时, 需保证其父子节点的数据也被正确更新.
- ④ 难扩展: 引入新的关系或数据模型时, 可能修改整个 DB 的结构.
- ⑤ 缺乏标准: 无 RDBS 的标准查询语言, 可能不兼容不同系统.
- ⑥ 访问控制困难: 需保证只有授权用户能访问特定的节点.

[网状数据库系统, Network DBS, NDBS]

(1) 数据组织为一个无环图.

(2) 示意图:



[关系型数据库系统, Relational DBS, RDBS]

- (1) 产生原因: HDBS 和 NDBS 太难用.
- (2) 数据组织为关系表.
- (3) 示意图:

Student:						
ID	Name	Age	Dept-Name	GPA		
130	Mark	22	CSSE	3.2		
522	Jay	21	CSSE	3.8		
282	Eva	19	SFL	3.5		

Course:					
CourseID	Course Title	Professor	ROOM		
CSSE20	DBS Principle	Dr.Peng	A305		
CSSE30	COBOL	Obama	B409		
CSSE40	Database Mgt	Ronald	C606		
CSSE50	Sys.Analysis	Kobe	B313		

Student-Course:	
StudentID	CourseID
522	CSSE20
522	CSSE40
522	CSSE50
131	CSSE30
131	CSSE50
282	CSSE20
282	CSSE30
282	CSSE50

1.3 数据库系统

[数据库系统, DBS]

- (1) 产生原因: 数据逐渐增多, 需要存储和处理技术.
- (2) 组成:

① **[Database]** 有如下三条性质的数据的集合:

- (i) 对应现实中的某些方面.
- (ii) 因特殊要求而创建.
- (iii) 数据间相关.

② **[Data Model]** DBS 中对数据的描述.

- (i) 数据的结构.
- (ii) 数据的操作.
- (iii) 数据的限制.

③ **[DBMS]** DBS 中用于定义(define)、存储(store)、操作(manipulate)和控制(control)数据的软件.

1.4 DB 语言

[DB 语言]

- (1) **[Data Definition Language, DDL]** 定义 DB 的模式(schema).
- (2) **[Data Manipulation Language, DML]** 对数据进行增(insert)、删(delete)、改(update)、查(retrieve).

① **[Query Language]** 用于检索数据的 DML .

② 分类:

(i) **[Procedural DML]** 用户需同时知道需要什么数据和如何得到它.

难用, 但是效率高.

(ii) **[Non-procedural DML]** 用户只需知道需要什么数据. 例: SQL.

好用, 但是效率低.

③ 解决办法: Non-procedural DML + query optimization.

(3) **[Data Control Language, DCL]** 对 DB 进行权限控制.

1.5 DB Schema

[DB Schema]

(1) 定义: DB 中对用户数据的描述.

(2) 在关系模型中, 一个 DB Schema 由数据表的集合构成.

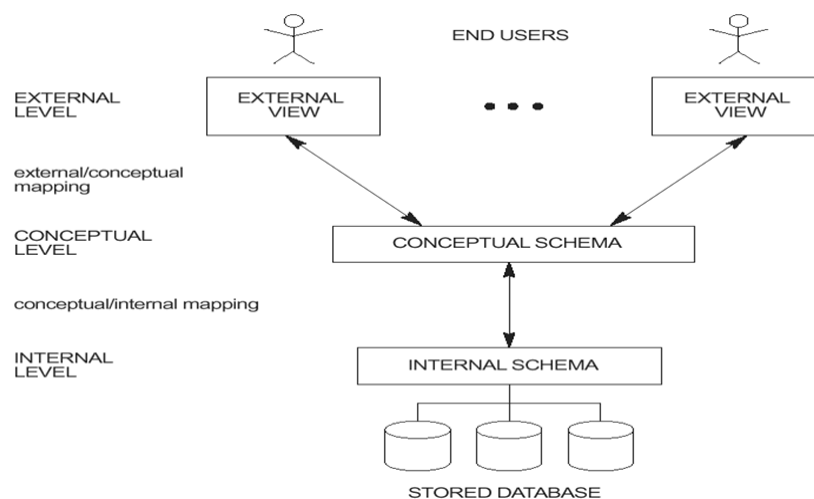
(3) 分类:

[Conceptual Schema] 逻辑上地定义 DB 中的所有数据.

[Internal Schema, Physical Schema] 描述数据在 DB 中如何存储.

[External Schema, User View] 每个 External Schema 描述特定用户组感兴趣或可访问的一部分数据.

[DBMS 3-Tier Architecture]



(1) 3 层:

① **[External Level]** 关注个体用户看见数据的方式.

② **[Conceptual Level]** 对组织感兴趣的数据的正式描述, 独立于存储.

③ **[Internal Level]** 关注数据如何存放.

(2) 有点:

① 用户容易理解和使用 DB .

② 提供访问控制机制.

