

第一章 数据库简介

数据库系统组成

数据库系统是指由用户、数据库应用程序、数据库管理系统和数据库组成的系统。

数据库组成（对象）

用户数据、元数据、索引或其它
开销数据、应用元数据。

数据库定义

关联表的自描述集合。（书）
长期存储在计算机上内,有组织,可共享的数据集合。

元数据

是一种专门描述数据库结构的数据。

数据库管理系统（DBMS）

用于创建、处理和管理数据库的计算机程序。

SQL 的使用方式

交互和嵌入。它嵌入到编程语言(如 JAVA、C#等)中执行或在 DBMS 工具中执行。

DBMS 接收以 SQL 编码的请求,并将这些请求转化为数据库中的操作。应用程序通过向 DBMS 发送 SQL 语句来读取或修改数据库数据。

DBMS 主要功能

- 创建数据库和表
- 创建支持结构,如索引等
- 读取数据库数据
- 修改数据库数据
- 维护数据库结构
- 执行规则
- 并发控制
- 提供安全性
- 执行备份和恢复

数据库应用程序

根据用户需要在 DBMS 支持下运行的一类计算机应用程序。

功能

- 创建并处理表单
- 处理用户查询
- 创建并处理报表
- 执行应用逻辑
- 控制应用

个人数据库系统与企业数据库系统比较

个人数据库主要运用在少量数据处理、

单用户的场景。企业级数据库主要运用在大量数据处理、多用户使用的场景。

数据模型

描述数据、数据间关系及其约束的形式描述。

关系模型

是一种基于关系表的数据模型

小结

关系模型是目前主流数据库技术

SQL 语言是数据库操作的标准语言

数据库应用结构: 集中式结构、客户/服务器结构、分布式结构

第二章 关系模型

关系模型

是一种基于关系表的数据模型。

关系的特征

表的每行存储了某个实体或实体某部分的数据;表的每列包含了用于表示实体某个属性的数据;单元格不允许有重复元素;任意一列中所有条目类型一致;每列必须有唯一名称;行顺序任意;任意两行不能有完全相同的数据。

键

是指表中用来标志行的以列或多列。

复合键

包含两个或多个属性的键。

候选键

唯一标志表中每一行的键

主键

唯一标志表中每一行的候选键

外键

表中用于表示关联的属性。

主键与候选键关系:

主键也是候选键之一,关系表可以有多个候选键,但只能定义一个主键

主键的作用

- 唯一标识关系的每行
- 作为关联表的外键,链接两个表
- 使用主键值来组织关系的存储
- 使用主键索引快速检索数据

代理键

采用 DBMS 自动生成的数字序列作为关系表的主键。

参照完整性

是指外键取值必须与现有主键值对应。

良好结构关系的设计原则：

- 1、在关系中，每个函数依赖关系的决定因子都必须是候选键
- 2、对于任何违反上述原则的关系，应当将其分解为两个或多个符合该原则的关系。

规范化过程

- 1、确认候选键
- 2、确认函数依赖关系
- 3、检查是否有决定因子不是候选键有的话，存在问题。

第三章 结构化查询语言

数据操纵语言（DML）

是 SQL 语言中用于查询和增删修改数据的语句

Select、update、delete、insert

数据定义语言（DDL）

是 SQL 语言中用于创建或删除表。也可以定义索引（键），规定表之间的链接，以及施加表间的约束。

Create database、alter database、create table、alter table、drop table、create index、drop index

Create Table 见书

INSERT

Insert into TableName Values ('列值', '列值', , '列值');

若缺少某些列

Insert into TableName ('属性', '属性') Values ('列值', '列值');

SELECT

为了使读取的数据删除重复行,可以在查询语句中加入 DISTINCT 关键字:

OR:

where.....or

e.g WHERE Department = 'Accounting'

OR Phone= '360-285-8410'; Ch03-SQLQuery11

IN、NOT IN:

指定列值必须是值集中的一个

e.g WHERE Department IN ('Accounting', 'Finance', 'Marketing')

通配符

BETWEEN....AND...

e.g WHERE EmployeeNumber BETWEEN 2 AND 5 等价于

WHERE EmployeeNumber >=2 AND EmployeeNumber<= 5;

LIKE

和下划线（代表一个未制定字符）和百分号（代表一个或多个未指定字符）连用

e.g WHERE Phone LIKE '360-287-%';

IS NULL

ORDER BY

默认为升序，降序为 DESC

内置函数

Count、sum、avg、max、min

列名不能和内置函数一起用，group by 除外

COUNT

AS 关键字可以将列名赋值给该结果。

e.g Select count (*) as NumberOfPro From project

内置函数和分组

使用 group by 后，分组列的名称和内置函数可以同时出现在 select 语句中。只是唯一允许列和内置函数同时出现的情况。

使用 Having 可以应用某些条件以进一步限制结果。不能用 AND。

子查询

e.g 想知道在某个任务中，工作时间超过 50 小时的所有雇员姓名。雇员名称存储在 EMPLOYEE 表中，但他们的工作时间则存储在 ASSIGNMENT 表中。

SELECT FirstName, LastName

FROM EMPLOYEE

WHERE EmployeeNumber IN

(SELECT DISTINCT EmployeeNumber FROM ASSIGNMENT

WHERE HoursWorked> 50);

连接查询

在处理多个表时，子查询只有在结果来自一个表的情况下才有用。但如果需要显示两个或多个表中的数据，就不能使用子查询，而需要采用连接操作。

连接本身其实也是一个表，同样可以采用 SELECT 语句操作，还可使用 AS 关键字将别名赋予表名，简化语句。

e.g 关联查询 EMPLOYEE 表和 ASSIGNMENT 表，得到雇员任务工时列表结果集；然后对结果集按雇员分组，统计他们的工作时间。

```
SELECT FirstName, LastName,
SUM(HoursWorked) AS TotalHoursWorked
FROM
```

```
EMPLOYEE AS E, ASSIGNMENT AS A
WHERE E.EmployeeNumber
=A.EmployeeNumber
```

```
GROUP BY FirstName, LastName;
```

修改数据

```
UPDATE TableName
SET Column=列值
Where someCondition;
```

删除

```
Delete from TableName
Where someCondition;
```

视图

是一种通过其它表或视图构建的虚拟表。它本身没有自己的数据，而是使用了存储在表或其它视图中的数据。

e.g 定义一个关于 EMPLOYEE 表的名为 EmployeePhoneView 视图。

```
CREATE VIEW EmployeePhoneView AS
SELECT FirstName,LastName,Phone AS
EmployeePhone
FROM EMPLOYEE;
```

第四章 数据建模与实体-关系模型

数据库开发的步骤

数据库需求分析、数据库设计、数据库实现

实体:用户希望跟踪的对象

属性:描述实体的特征

关系:实体间通过关系来关联。

标识符:指标识不同实体例的属性。

基数

实体之间的关系反映了一个给定实体有多少实例与另一实体实例存在的数量对应关系

软件开发阶段与数据模型的关系

需求分析阶段: 设计者与用户交流,记录系统需求,建立数据模型

系统设计阶段: 数据模型转换成关系数据库设计。

实现阶段: 创建数据库，填充数据，创建查询，表单，报表和应用程序。

弱实体

逻辑上依赖另一个实体的实体而存在。所有 ID 依赖实体都是弱实体。不是关系中所有最小基数为 1 的实体都是弱实体，只要逻辑上依赖其他实体的实体才称为弱实体。

弱实体最小基数都是 1，但最小基数为 1 的实体不一定是弱实体。

子实体:是另一种父型实体的特例。

第五章 数据库设计

数据模型转到数据库的设计方案

- 1、为数据模型中的每个实体创建一个表
- 2、正确规范化所有的表
- 3、创建表之间的关系（主外键约束）

关系规范化

第一范式（1NF）

如果关系中的属性不可再细分，该关系满足第一范式。

第二范式（2NF）

如果关系满足第一范式、并消除了关系中的属性部分依赖，则满足第二范式。

第三范式（3NF）

如果关系满足第二范式，并切断了关系中的属性传递依赖，该关系满足第三范式。

第六章 数据库管理

数据库管理的必要性

- ◆ 数据库系统随规模增大,系统会变得异常复杂
- ◆ 多用户数据库应用带来数据库访问复杂性
- ◆ 数据安全和数据隐私对机构和用户都非常重要
- ◆ 数据库性能随系统使用发生变化,需要定期维护

- ◆ 最大程度地发挥数据库对其所属机构的作用

数据库管理员（DBA）职责

- ◆ 负责数据库系统开发与运行
- ◆ 数据库用户管理
- ◆ 数据备份与数据恢复管理
- ◆ 数据库性能调优管理

数据库三个重要的管理功能

并发控制、安全性、备份和恢复

并发控制

保证一个用户的工作不对另一个用户造成不合理的影响。

事务:数据库采取的一系列操作.

- ◆ 也称为逻辑工作单元、原子。
- ◆ 完成单一逻辑工作单元的操作集合
- ◆ 操作序列要么完整执行,或者任何操作都不执行,以保证数据完整性。

事务全局变量使用方法

begin transaction

..... /* A 组语句序列* /

save transaction 保存点 1

..... /* B 组语句序列* /

if @@error <> 0

rollback transaction 保存点 1 /* 回滚到

保存点 1 * /

else

commit transaction

/*提交 A 和 B 组语句* /

事务特性

原子性（Atomicity）:

事务所有操作在数据库中要么全部执行,要么全部不执行。

一致性（Consistency）:

事务多次执行,其结果应一致。

隔离性（Isolation）:

事务间隔离,并发执行透明。

持续性（Durability）:

事务完成后,数据改变必须是永久的

并发问题

脏读取

当一个事务读取一个未提交到数据库但已修改的记录时,就会发生脏读取。

比如:如果一个事务读取一行数据,而这行数据被另一个事务修改过,后来第二个事务又取消了修改,就发生了脏读取。

不可重复读取

事务 T1 按一定条件从数据库中读取了某些数据记录后,事务 T2 删除了其中部分记录,当 T1 再次按相同条件读取数据时,发现某些记录消失了。

幻想读取

事务 T1 按一定条件从数据库中读取某些数据记录后,事务 T2 插入了一些记录,当 T1 再次按相同条件读取数据时,发现多了一些记录。称为幻象读取。

事务并发执行带来问题原因:

当多个事务同时存取同一数据时,由于不同操作时序,可能出现存取不正确的数据,破坏数据库的一致性。

并发控制的主要方法

资源锁定

在将要修改某些数据行或表时,通过加锁禁止多个事务同时获取这些行或表的数据,这种方法称为“资源锁定”



串行事务

在并发处理两个或多个事务时,如果数据库操作结果在逻辑上同事务以串行方式处理结果保持一致。这种处理并发事务的模式称为“串行模式”。

并发事务正确调度准则:

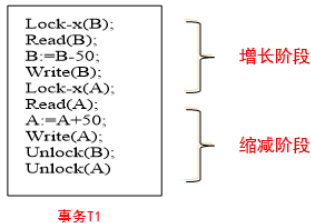
一个给定的并发调度,当且仅当它是可串行化的,才能保证正确调度。

保证可串行化的一个协议是:二阶段锁定协议

二阶段锁定协议

规定每个事务必须分两个阶段提出加锁和解锁申请:

- ♦ 增长阶段, 事务可以获得锁, 但不能释放锁。
- ♦ 缩减阶段, 事务可以释放锁, 但不能获得新锁。



死锁

多个事务由于因争夺资源而造成的一种相互等待而不能执行的一种状态。

死锁防范策略

允许用户一次发出当前所需全部资源的锁定, 使用完成后, 再释放给其它用户访问。

规定所有应用程序锁定资源的顺序必须完全相同。

游标: 指向一组记录的指针。

数据库的安全: 身份认证和授权。

数据库备份和恢复: 前滚和后滚. 日志.

数据库管理的目的是管理数据库实现他的最大价值。

原子事务的含义

前滚和后滚的含义.

前滚是指通过已保存的数据库文件恢复数据库, 并且重新应用保存后的所有有效事务。

回滚是指通过取消错误地执行或者没有全部完成的事务对数据库的更改以纠正错误。

第七章

存储过程: 存储过程类似于计算机程序的子例程或函数, 但它存储在执行数据库操作的数据库中。

触发器: 触发器是一个存储在数据库中的程序, 并在特定时间发生时可以由 DBMS 执行。

7.1 指出三种 ODBC 数据源之间的区别.

- A. 文件数据来源是一个文件, 数据库用户可以共享。唯一的要求是让用户用相同的数据库驱动器并拥有相同的数据库访问权限。其数据来源文件可以通过电邮或以其他方式分发给可能的用户。
- B. 系统数据源对应于一台电脑。操作系统和任何用户在该系统(拥有适当的权限)都可以使用系统数据源。
- C. 用户数据来源只对其创建者可用。