# 00 关系数据库概述

为什么需要数据库?

因为应用程序需要保存用户的数据,比如Word需要把用户文档保存起来,以便 下次继续编辑或者拷贝到另一台电脑。

要保存用户的数据,一个最简单的方法是把用户数据写入文件。例如,要保存一个班级所有学生的信息,可以向文件中写入一个CSV文件:

```
id,name,gender,score
1,小明,M,90
2,小红,F,95
3,小军,M,88
4,小丽,F,88
```

如果要保存学校所有班级的信息,可以写入另一个CSV文件。

但是,随着应用程序的功能越来越复杂,数据量越来越大,如何管理这些数据就 成了大问题:

- 读写文件并解析出数据需要大量重复代码;
- 从成千上万的数据中快速查询出指定数据需要复杂的逻辑。

如果每个应用程序都各自写自己的读写数据的代码,一方面效率低,容易出错, 另一方面,每个应用程序访问数据的接口都不相同,数据难以复用。

所以,数据库作为一种专门管理数据的软件就出现了。应用程序不需要自己管理数据,而是通过数据库软件提供的接口来读写数据。至于数据本身如何存储到文件,那是数据库软件的事情,应用程序自己并不关心:

这样一来,编写应用程序的时候,数据读写的功能就被大大地简化了。

## 数据模型

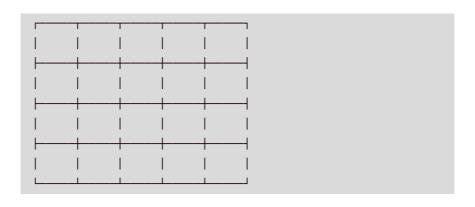
数据库按照数据结构来组织、存储和管理数据,实际上,数据库一共有三种模型:

- 层次模型
- 网状模型
- 关系模型

层次模型就是以"上下级"的层次关系来组织数据的一种方式,层次模型的数据结构看起来就像一颗树:

网状模型把每个数据节点和其他很多节点都连接起来,它的数据结构看起来就像 很多城市之间的路网:

关系模型把数据看作是一个二维表格,任何数据都可以通过行号+列号来唯一确定,它的数据模型看起来就是一个Excel表:



随着时间的推移和市场竞争,最终,基于关系模型的关系数据库获得了绝对市场份额。

为什么关系数据库获得了最广泛的应用?

因为相比层次模型和网状模型,关系模型理解和使用起来最简单。

关系数据库的关系模型是基于数学理论建立的。我们把域(Domain)定义为一组具有相同数据类型的值的集合,给定一组域D1,D2,...,Dn,它们的笛卡尔集定义为D1×D2×......×Dn={(d1,d2,...,dn)|di  $\in$  Di,i=1,2,...,n},而D1×D2×.....×Dn的子集叫作在域D1,D2,...,Dn上的关系,表示为R(D1,D2,...,Dn),这里的R表示You can't use 'macro parameter character #' in math mode #;!~%¥%:(......算了,根本讲不明白,大家也不用理解。

基于数学理论的关系模型虽然讲起来挺复杂,但是,基于日常生活的关系模型却十分容易理解。我们以学校班级为例,一个班级的学生就可以用一个表格存起来,并且定义如下:

ID	姓名	班级ID	性别	年龄
1	小明	201	M	9
2	小红	202	F	8
3	小军	202	M	8
4	小白	201	F	9

其中, 班级ID对应着另一个班级表:

ID	名称	班主任
201	二年级一班	王老师
202	二年级二班	李老师

通过给定一个班级名称,可以查到一条班级记录,根据班级ID,又可以查到多条学生记录,这样,二维表之间就通过ID映射建立了"一对多"关系。

## 数据类型

对于一个关系表,除了定义每一列的名称外,还需要定义每一列的数据类型。关系数据库支持的标准数据类型包括数值、字符串、时间等:

名称 类型 说明

名称	类型	说明
INT	整型	4字节整数类型,范围约+/-21亿
BIGINT	长整型	8字节整数类型,范围约+/-922亿亿
REAL	浮点型	4字节浮点数,范围约+/-1038
DOUBLE	浮点型	8字节浮点数, 范围约+/-10308
DECIMAL(M,N)	高精度 小数	由用户指定精度的小数,例如,DECIMAL(20,10)表示一共20位,其中小数10位,通常用于财务计算
CHAR(N)	定长字 符串	
VARCHAR(N)	变长字 符串	存储可变长度的字符串,例如,VARCHAR(100)可以存储0~100个字符的字符串
BOOLEAN	布尔类 型	存储True或者False
DATE	日期类型	存储日期,例如,2018-06-22
TIME	时间类型	存储时间,例如,12:20:59
DATETIME	日期和 时间类 型	存储日期+时间,例如,2018-06-22 12:20:59

上面的表中列举了最常用的数据类型。很多数据类型还有别名,例如,REAL 又可以写成 FLOAT (24)。还有一些不常用的数据类型,例如,TINYINT(范围在 0~255)。各数据库厂商还会支持特定的数据类型,例如 JSON。

选择数据类型的时候,要根据业务规则选择合适的类型。通常来说,BIGINT能满足整数存储的需求,VARCHAR(N)能满足字符串存储的需求,这两种类型是使用最广泛的。

# 主流关系数据库

目前,主流的关系数据库主要分为以下几类:

- 1. 商用数据库,例如: Oracle, SQL Server, DB2等;
- 2. 开源数据库,例如: MySQL, PostgreSQL等;
- 3. 桌面数据库,以微软Access为代表,适合桌面应用程序使用;
- 4. 嵌入式数据库,以Sqlite为代表,适合手机应用和桌面程序。

## SQL

什么是SQL? SQL是结构化查询语言的缩写,用来访问和操作数据库系统。SQL语句既可以查询数据库中的数据,也可以添加、更新和删除数据库中的数据,还可以对数据库进行管理和维护操作。不同的数据库,都支持SQL,这样,我们通过学习SQL这一种语言,就可以操作各种不同的数据库。

虽然SQL已经被ANSI组织定义为标准,不幸地是,各个不同的数据库对标准的 SQL支持不太一致。并且,大部分数据库都在标准的SQL上做了扩展。也就是说,如果只使用标准SQL,理论上所有数据库都可以支持,但如果使用某个特定数据库的扩展SQL,换一个数据库就不能执行了。例如,Oracle把自己扩展的

SQL称为PL/SQL,Microsoft把自己扩展的SQL称为T-SQL。

现实情况是,如果我们只使用标准SQL的核心功能,那么所有数据库通常都可以执行。不常用的SQL功能,不同的数据库支持的程度都不一样。而各个数据库支持的各自扩展的功能,通常我们把它们称之为"方言"。

总的来说,SQL语言定义了这么几种操作数据库的能力:

#### **DDL: Data Definition Language**

DDL允许用户定义数据,也就是创建表、删除表、修改表结构这些操作。通常, DDL由数据库管理员执行。

#### **DML: Data Manipulation Language**

DML为用户提供添加、删除、更新数据的能力,这些是应用程序对数据库的日常操作。

#### **DQL: Data Query Language**

DOL允许用户查询数据,这也是通常最频繁的数据库日常操作。

# 语法特点

SQL语言关键字不区分大小写!!!但是,针对不同的数据库,对于表名和列名,有的数据库区分大小写,有的数据库不区分大小写。同一个数据库,有的在Linux上区分大小写,有的在Windows上不区分大小写。

所以,本教程约定: SQL关键字总是大写,以示突出,表名和列名均使用小写。