

数据库应用设计的研究

苏顺开 (中山大学计算中心)

摘要: 本文根据笔者在 VAX Rdb / VMS 上从事系统开发的经验, 以系统论的观点探讨了数据库应用设计的思路, 并讨论了文件、索引以及数据规范化等关键技术。

一、前言

我们知道, 管理信息系统(MIS)是一个由人、计算机硬件、软件等资源组成的能进行对信息的收集、传递、存贮、加工、维护以及使用的计算机应用系统。MIS 又是建立在数据库管理系统(DBMS)之上的, 它们两者之间互相联系, 又互相作用。MIS 对信息的存取 / 删除、以及检索等操作都是由 DBMS 来实现的。

对 MIS 的设计, 最重要的一项任务就是对数据库应用的设计, 其设计质量的好坏直接影响到 MIS 的性能和效率。特别是对大型 MIS 的开发更应当重视对数据库的设计。

本文以笔者在 VAX Rdb / VMS 关系数据库系统支持下所开发的长途电话业务管理系统为背景, 总结并讨论对数据库进行应用设计的思想方法。

二、数据库应用设计的依据

1. 以软件需求为基础

在软件需求说明书中, 数据流图, 数据字典, 以及有关的输入输出资料都是设计数据库的依据。

特别是, 以数据流图为主导, 更有利于开展数据库的设计。因为数据流图贯穿于系统的业务流程, 反映了业务的需求。在这个环节上需要输入什么资料, 这在流图中都有所表达, 这些输入输出资料将作为数据库设计的主要对象。

2. 根据 DBMS 的特点进行考虑

不同数据模型的 DBMS 有着不同的特点, 就算同样是关系模型的数据库, 它们之间也有差别。例如, VAX 上的 Rdb 和 ORACLE 数据库, 它们虽然都是以 SQL

语言为基础, 在 Rdb 中有多文件(multifile)和单文件(single file)这两个概念, 多文件指根文件和存储区数据文件是分离的, 而单文件则是结合的, 但在 ORACLE 中就没有这种分别。

三、数据库应用设计的思路

哲学上认为: 任何事物都是作为系统而存在的, 系统性是事物的根本属性。信息作为事物的联系、变化、差异的一种表现, 这就启示我们, 对信息系统的设计, 也就是对数据库的应用设计, 可以根据系统论的观点进行研究。

系统有哪些基本特性呢? 通常, 它具有整体性、结构性、层次性以及动态性等特性。

1. 数据库的整体性设计

系统的整体性, 揭示了系统整体与其组成要素之间的关系。信息系统通常是由各种用途的资料组成的, 在这些资料中, 有的是临时性使用的, 有的是长期性使用的, 还有的是为统计而用的。

同样, 一个数据库应用系统应该包括哪些方面的文件呢? 这要根据具体情况而定, 正如上述, 数据库的设计主要依赖于与数据流程有关的输入输出文件, 此外, 还有一些用于构成一个 MIS 所必须的辅助文件。概括地说, 从应用的角度来看, 数据库应用主要包括下列五类文件。

(1) 临时文件。这类文件在业务处理过程中属于过渡性的, 它用于在 MIS 中收集到的资料临时存放起来, 有待进一步的处理, 这些临时性的记录一经处理, 就被分类传递到有关的主体文件中。例如长途电话业务管理系统, 在营业受理中的申请文件, 它用于记录每天的用户电话申请情况, 一旦核算室对该用户电话审核之后, 就把该申

请记录分别写进用户资料文件和电话业务文件中,而该申请记录就被转移到历史文件中了。

(2)主体文件。这类文件属于 MIS 需要长期使用的核心资料,记录着 MIS 主要处理的业务信息。如上述的用户资料文件和电话业务文件。

(3)统计文件。这类文件用于记录各年、或各季度、或各月份的业务统计数据,用于打印年报、季报或月报表。

(4)历史文件。这类文件用于保存已经过期的,但还不宜删去的资料,以便日后可以追查。

(5)对照表文件。它是 MIS 用于记录代码与名称之间的对应关系的辅助性文件。

数据库的整体性设计就是根据软件需求把系统应包括哪些数据库文件确定下来。把握了整体性,就可进一步讨论其它系统特性。

2.数据库的结构式设计

系统的结构性,揭示的是系统中诸要素之间的关系。在数据库系统中,它所包括的文件之间通常都有一定的联系。我们的目地,就是要找出联接这些文件的连接字段,特别是要确定主体文件与临时文件、历史文件之间的连接字段。连接字段通常是相关文件中共有的具有唯一性的数据项。

3.层次性的信息关系

系统的层次性,指系统中各组成部分之间的依次隶属关系。层次性的表现形式是多方面的,在信息实体中它反映了一对多的关系。例如,一个用户可以申请使用多个长途电话,显然,这就存在一对多的关系,于是,可以把用户电话的申请资料设计为二个文件;一个是用户资料文件,它记录着用户编码、用户名称、用户地址以及银行帐号等信息;另一个是电话业务文件,它记录着用户编码、电话号码、业务代码以及申请日期等多项信息。不论用户何时来申请这些电话,都写一个记录到电话业务文件中,但是,在用户资料文件中,始终只有一个该用户的记录,这是因为对于这多个电话来说,它的主人(即电话户主)的个人基本资料是相同的。并且,我们通过用户编码这个字段就可以把这两个文件联接起来。

因此,在数据库设计时,要进一步把具有这类层次性的、一对多关系的资料区分出来。

4.动态性的客观变化

系统有一个从孕育、产生、发展、成熟到衰退、消亡的过程。系统的这种运动、发展、变化过程就是它的动态性。

信息系统充分体现了这种动态性。有些信息刚输入到系统时是很有用的,但经过一段时期后,由于某种原因,该信息不起作用了,就该从系统中予以清理。那么,如何去反映这种信息变化的动态性呢?

我们认为,可以在相应文件中设计反映事物发生及结束的日期字段,同时,设计反映事物变化情况的状态字段。有了这两类字段,就可以记录事物的动态变化情况。它将为数据维护工作提供依据。

以上是运用系统论的观点论述数据库的设计思路,有助于我们拓宽视野。

四、数据库文件的要素

前面讨论了数据库的整体设计,下面讨论单个文件的组成要素。一个文件除了文件名之外,组成记录结构的数据项应包括哪些方面的内容呢?这个问题应按具体情况而定,本文认为下列六类不同作用的字段是值得考虑的。

1.业务字段

它用于记录与该文件有关业务内容的数据,如电话号码、电话业务代码、申请日期等。

2.连接字段

每个文件都应有一个与其它文件发生关系的连接字段,这个字段也可以是业务字段。如用户编码、电话号码。

3.日期字段

用于记录该项业务前后发展的经过时间。如开始日期、结束日期、或反映其中某种变化情况的日期。

4.状态字段

用于记录该项业务动态变化过程中的当前状态。作为一个标志,它通常以 0、1、2、...来代表不同的情况。

5.操作员字段

用于记录该项业务是谁写入的。在多用户环境下,对同一文件进行操作时,把此项信息保存下来是有好处的,它便于区别是哪位操作员处理该项业务数据。例如,在业务收费文件中我们就设置了营业员工号这个字段,也就是每个营业员在 VAX/VMS 上对应的帐号。

6.备用字段

这是为了适应将来业务发展的需要而预先定义好的字段,以免在系统使用期间,由于要求增加输入一项新的数据而需要修改数据库。

上述所列的六类字段,在实际应用中要根据具体情况灵活掌握。

五、索引设计的问题

索引设计是数据库应用设计的关键问题之一,因为它将直接影响系统的使用性能。

1.MIS的使用现象

一个大型 MIS 投入使用一段时期后,由于数据量不断增加,常常会出现下列现象:

在输入一张申请表中,当输入了某项数据后,由于系统需求对该项数据作唯一性检查,导致光标停在那个字段足有 20 秒以上,令现场营业员怨声四起。

当业务主管想对本部门某段时期内的业务增减情况作统计查询时,在十多万记录的文件中,竟然花了半小时还没有显示出统计结果,令人误以为死机了。

还有,在电话业务营业厅,为了能把当天的各类统计报表打印出来,在下班前半小时就停止了对用户申请的办理,然而,由于计算机进行统计的时间太长了,直到下班时间到了还没有把报表打印完。

诸如此类的现象,其原因是什么呢?

2.索引的优缺点

造成上述现象的主要原因之一,就是索引问题。或者是对有关的字段没有建立索引,或者即使建立了索引,但是采取的是匹配查询,所以其检索速度自然就慢了。

索引的主要作用是为了提高查询的性能。如果一个程序是按某个条件从数据库中读出记录,而该条件又是对应文件记录中的字段,且已对该字段建立了索引,则检索的速度就快。

索引的另一优点是可以保证数据的唯一性。在 Rdb/VMS 中,Create Unique Index 语句是用于对指定文件指定字段定义唯一索引。如果定义了唯一索引,当存入记录时,若那个字段值已经在该数据库中存在,则 Rdb/VMS 返回错误信息,该记录将不被写入数据库中。可见,定义唯一索引,不但提高查询速度,而且起到记录主键的作用。

然而,索引也会带来一些不利问题。当建立索引后,每当对数据库进行插入,更改或删除记录时,不但不会提高速度,还会降低处理速度,因为此时 Rdb/VMS 必须对索引文件按特定的要求作相应的调整,以保证索引文件与数据实体联系的有效性,这就会增加系统的 I/O 操作,多耗费 CPU 时间,也就是增加了系统的开销。

3.索引的设计原则

一个数据库文件可以建立多个索引结构,而一个索引键又可以由一个或多个字段组成。那么,对哪个字段或哪一组字段建立索引才合理呢?这不仅要 MIS 进行综合性的考虑,有些还要在实际使用中才能确定。

笔者在实践中的体会认为:

首先,对 MIS 进行整体分析,找出关键环节,找出在关键环节中使用最频繁的、经常对数据库进行增删改记录操作的程序,通常这些程序就是那么几个,然后,分析这些程序,进而找出对某些输入数据需要进行唯一性检查的字段,找出需要按某个条件从数据库中取记录进行删改的字段,那么,对于这些字段就应该建立索引。

其次,对于经常使用的和要求速度较快的查询程序和打印程序,在相应的数据库文件中找出与查询或打印条件有对应关系的字段,对于这些字段也应该建立索引。如日期字段、单位代码字段。

再次,对一些数据量比较多的文件才需要建立索引,而对一些数据比较少的文件(估计记录个数稳定在 2 千之内),则不一定建立索引。因为记录个数不多,就算是采用顺序查找的方法也不显得很慢,但是,若记录个数在几千个以上,或者在几万个以上,其查询速度将有明显的差别;我们在实践中感到,若记录个数在五万个以上,就非有索引不可,否则,速度之慢将令用户无法忍受。

至于一个文件定义多少个索引才合适呢?显然,越少越好,应该抓住主要矛盾,一般情况下以 3 个左右为宜。

六、数据规范化问题

规范化是关系数据库设计的理论,它对数据库的应用设计同样具有指导意义。

1.规范化的目的

在现实世界中,往往存在一些比较复杂的数据结构,存在大量的重复字段,这不但产生大量的数据冗余,还使得修改困难,容易造成数据不一致的情况。

因此,在数据库的逻辑设计中,数据规范化的目的就是减少数据冗余和保证数据的一致性。解决方法就是将复杂的数据结构逐步分解为简单的数据结构。

通常认为:如果一个数据结构没有重复出现的数据项,就称该数据结构是规范化的。数据结构的第一规范形式(简称第一范式)表示消除了重复的数据项,第二范式表示消除多余的数据相关性,第三范式表示消除传递相关性。

2.规范化带来的问题

我们应当以一分为二的观点来看待数据库的规范化

设计。虽然规范化可以减少数据冗余和保证一致性,但是,由于规范化程度越高,就会增加文件的个数,当需要对多个文件进行复杂的联接操作时,就不但增加了系统的复杂性,而且增加了 I/O 操作,使数据库的性能下降,影响了处理速度。也就是理论上最优的数据库在实践中不一定最优。

所以,进行数据库设计时,应根据实际需要,以规范化思想作为指导,在保证数据的一致性和系统性能的前提下,允许有少量冗余。