

一个历史关系数据库管理系统 —HXDB

许云涛 翁新平 朱欣焰

摘 要

历史关系数据库管理系统HXDB是在关系数据库管理系统XDB基础上扩充的,它在提供查询历史数据机制的同时,保证对当前数据的快速查找及最大限度地减少数据冗余。

【关键词】 历史数据库; 当前数据; 有效时间

1 引 言

数据库技术发展异常迅速,现在发展的一个分支是历史数据库技术。历史数据库所要解决的问题是使数据库能够存贮与查询历史数据。现在,历史数据库面临的一个主要难题是如何在提供存贮与查询历史数据的机制的同时,保证对当前数据的查询快速且数据冗余较少。作者在关系数据库管理系统XDB基础上扩展的HXDB系统比较圆满地解决了这个问题。

2 历史性信息的表示

传统的关系数据库模型属第一代数据库模型,它不适于表示历史性数据。历史性数据是随时间变化的,例如,在处理银行帐户事务时,一笔帐用一个元组表示,当前事务中的余额要依赖于上一笔帐(实体的历史)的余额以及当前事务中的收入支出情况(当前的输入值)。因此,这样的数据就是历史性数据。

为表示历史性数据,将平面(二维)结构扩展成三维立体结构,即再加上时间这一维。这种三维立体模型按时间变化不断重复存贮数据库的状态的完整地保留历史性数据。但显然冗余太多,存贮开销太大且难以保证数据库的一致性。改进的办法是在二维关系中增加有效时间(Valid-time)。所谓有效时间是指信息在现实世界中产生的时间。

收稿日期:1989-09-09

*国家测绘局“七.五”重点科研项目“全图测绘资料档案管理系统”研究课题。

将有效时间不作为关系模型式的一部分，并用双竖线将它与关系模式区分开来。此时，只把有效时间看成是每个元组的附加信息。

这种表示方法减少了冗余，但因Valid-time不是关系模式的一部分，因而传统的查询语言不再适用。这种表示方法仍存在较多的冗余，比如象noms、sex等这些不变化的属性也被重复存贮。同时，新旧数据同在一张表格之中，其地位没有任何区别，这势必影响数据的处理效率，因为经常使用的往往是新数据，对旧数据的访问毕竟是次要的，少量的。

历史性信息的表示仍然是现在人们继续研究的问题。我们提出了一个新策略，试图进一步减少冗余，并使新旧数据分开以保证对最新数据的处理快速。按照这个新策略，设计了HXDB系统。

3 HXDB系统的设计策略

HXDB系统是对关系数据库系统XDB的扩展，而扩展关系数据库的关键是扩展其关系模型。因此，关系模型的扩展是设计HXDB的出发点。扩展关系模型的策略如下：

(1) 数据分为当前数据和历史数据。当前数据是那些在时间上最新的数据；

(2) 属性分为静态属性和动态属性。动态属性的值是随时间变化的，而静态属性是与时间无关的；

(3) 关系分为动态关系和静态关系。动态关系由当前关系和历史区这两部份组成。当前关系存贮当前数据，历史区存贮历史数据。每个动态属性在每个元组中可以有一个历史区。

(4) 当前关系不含有效时间，有效时间仅在历史区中出现。

为说明这个策略，设有如下一个关系faculty1，如表1所示：

表 1 一个关系

faculty1

name	slx	birthplace	nationality	professlon	rank	salaoy	Ualid-time
wang	male	guangdong	han	teacher	assistant	67	80~84
wang	male	guangdong	han	teacher	leiturer	97	85~
li	female	hunan	han	teacher	associate	97	82~82
li	female	hunan	han	teacher	associate	121	83~84
li	female	hunan	han	teacher	full	121	85~

faculty1关系是个带有 valid-time 附加信息的扩展关系。按照上面提出的策略，该关系变为图 1 所示的结构：

faculty1							01/01/85	01/01/80
name	sex	birthplace	nationality	profession	rank	salary	valid time	
wang	male	guangdong	han	teacher	lecturer	97	97	67
li	female	hunan	han	teacher	full	121		

图 1 当前关系与历史区

图 2 所示的结构是个立体结构，由当前关系与历史区所助成。faculty2关系应有四个历史区：wang的rank历史区，wang的salary 历史区，li的rank 历史区，以及li的salary 历史区。图 1 中只画了wany的salary历史区。历史区的结构如图 2 所示：

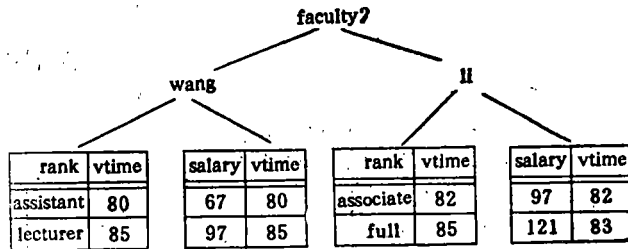


图 2 历史区的结构

在历史区中，时间是连续的。从图 1、图 2 可看出，表 1 中的冗余大部分已经消除，现在唯一存在的冗余是动态属性的当前值（当前数据）分别在当前关系和历史关系和历史区中存贮。但这种冗余并不大，尤其当历史区比较大时。

从图 1 中可知，当前数据存贮在当前关系中。如果对数据库的操作没有显式给出时间信息，则被认为是对当前数据的操作，这便保证了对当前数据的快速处理。如果给出时间信息，则各静态属性的值从当前关系中查询，动态属性的值则在历史区中查询。这意味着不仅要访问当前关系还要访问历史区，显然速度受到了影响。

如果关系不含动态属性，则便无历史区，所有数据全部存贮在当前关系中，这种含有时间属性但不含有动态属性的关系也叫静态关系，这简化了历史库的复杂性。

由于当前数据（动态属性的）既在历史区中又在当前关系中，这不仅增加了冗余，同时也使得操作、维护数据变得复杂，更新、删除操作可能导致不一致性。为避免这种不一致性，凡是更新、删除当前数据，均必须同时更新或删除当前关系及历史区中的值。这显然使系统变得复杂，降低了速度。

4 HXDB系统构成与查询语言HXQL

HXDB系统是在长城0520-CH和DOS系统的支持下，用C语言实现的。HXDB 系统采用多级菜单方式为用户服务，全部菜单都是经过精心安排的，主要包括系统主菜单、查询菜单和查询结果处理菜单。

表 2 HXDB 系统多级菜单

系统文 革单	HXQL 查 询 革单	结 果 处 理 革单
to HXQL	execute commane	to Report writer
initialiye	point commane	to Graph writer
help	list commane	print result
to DOS	save commane	help
	remove commane	to HXQL
	clear commane	
	HXDB directory	
	tator	
	insert	
	delete line	
	help line	
	to DOS	

HXDB 系统结构如下：

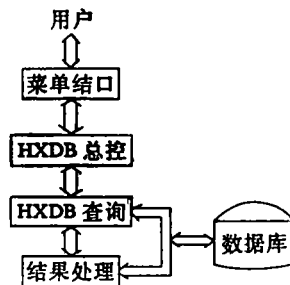
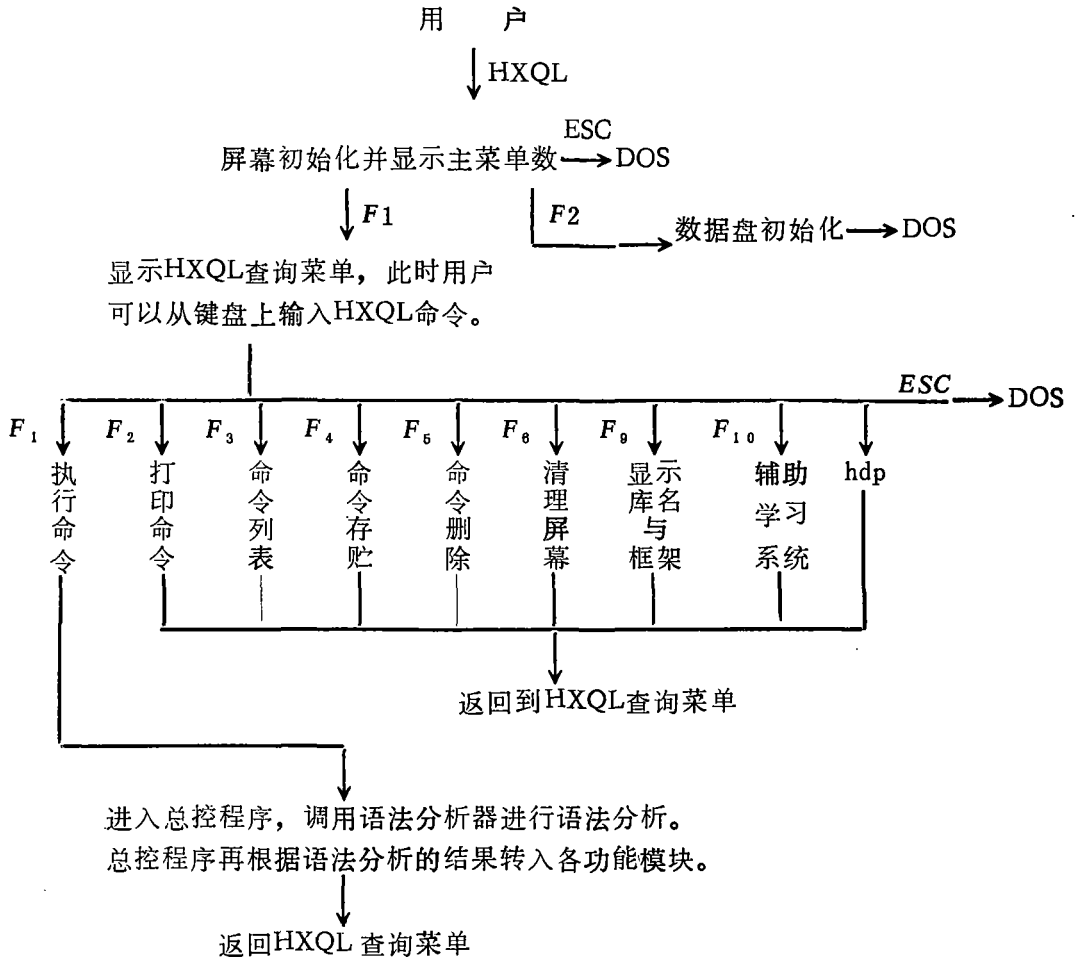


图 3 HXDB系统结构

HXDB系统的关键部份是其查询模块，它具有建表、查询、更新等多种功能。其工作流程如下：



HXDB 的查询语言叫HXQL，是XQL 语言的扩展，即The Historical XQL。HXQL 语言包含了XQL的所有功能，同时增加了处理历史性数据的能力。历史数据库能够处理历史性数据是因为其能够处理有效时间，因而其查询语言必然应有处理有效时间的能力。当用户使用HXQL语言时，必须显式地给出有效时间信息，否则便退化为XQL语言。因此，用户可以用HXQL语言建立、使用历史数据库，也可以用它建立、使用一般的传统关系数据库（只要不给出有效时间信息即可）。

HXQL语言主要包括数据定义语句、数据查询与操纵语句。HXQL语言的基本查询命令采种如下：

```

select A1, A2, ..., An
from T1, T2, ..., Tm
[where 条件表达式]
when 时间表达式]
  
```

可见，它仅比XQL语言增加了when子句。when子句是HXQL语言的关键，其后的时间

表达式是由有效时间 valid time、时间常数、比较符、时间属性等组成的逻辑表达式，它显式地位给出有效时间信息。when子句可有可无，当无when子句时，则表明仅查询当前数据。

HXQL语言支持下六种数据类型：

表 3 HXQL 支持的数据类型

名 字	字 节 数	最 大 值
smallintz 整数	2 bytes	32: 767
int 整数	4 bytes	2, 146, 483, 647
money 金额	4 bytes	\$21, 474, 836, 47
float 浮点数	8 bytes	1.0E + 300
date 日期	6 bytes	12/32/9999
char 字符	1~100 bytes	1000 char

5 实 例

历史数据库系统的特点在于它能够处理历史性数据，这种处理历史性数据的能力主要体现在其查询功能上。本节通过一个实例进一步分析HXDB中的HXQL语言的查询功能。

设有关系 faaulty1 (name, seo, birehplace, nationlity, profession, rank, salary) 及关系 submitted (name, paper, time)。关系faculty1如表 1 所示，其在HXDB中的结构如图 1、图 2 所示。

【例1】 如果要查找li的整个rank历史，则when后面不加具体时间便可，如下：

```
select rank
from faculty1
where name = "li"
when
```

查询结果为：

```
associate 01/01/82
full      01/01/85
```

【例2】 如果要同时查询li的rank、salary历史，则查询语句如下：

```
select rank, salsry
from faculty1
where name = "li"
when
```

查询结果为“

```
associate 97 01/01/82
associate 121 01/01/83
full      121. 01/01/58
```

此查询分二步执行：先分别查出li的rank历史和salary历史，再将它们重新以时间顺序排列整理。

【例3】 当li为associate时，谁为lecturer?

```
select name, rank
from faculty1
where rank = "lecturer"

when vtime = :
.....
: select
:
: from faculty
:
: where name = "li"
:
: and rank = "associate"
:
: when
:
: .....
```

该查询中，虚线部分为求li为associate的时间。如果select后面不带任何属性，则表明是求时间。这种select后不跟域名的查询语句只能在when后面嵌套使用。

when后面的时间表达式是HXQL语言的关键，它既可以是简单的时间，也可以是复杂的结构。另外，进一步地，它还应该可以包含有“to”、“start of”、“end of”等一些时间关键字，如Vtime = 84 to 86。这将是以后应进一步完善的工作。

6 结 论

扩展关系数据库系统使之能够处理历史性信息是人们一直努力的目标，目前成功的例子当推美国 R. Snodgrass 等人的时序数据库[3]和法国 M. Adiba 等人的历史多介质数据库。历史关系数据库管理系统HXDB 在提供处理历史性信息机制的同时，成功地减少了数据冗余，保证了对当前数据的快速处理，系统简单灵活，便于推广应用。

目前，系统本身还有一些不足之处，主要表现在：

(1) 与现实世界的不一致性：HXDB 使用的是有效时间，有效时间只反映信息产生的时间，不能反映信息进入数据库的时间。比如，王某于82年12月1日被提升为正教授，此信息则在82年12月15日进入数据库，则12月1日到12月14日这段时间内，数据库不能准确地反映王某的职称，出现了不一致。

(2) 不能保留数据库本身的活动历史：数据库反映现实世界，而数据库本身也是现实世界的一部分。由于HXDB 系统只有有效时间，所以它不能记载其本身的活动历史。

克服以上两个不足的办法是使系统不仅有有效时间，同时也有事务时间 (transaction time) [4]。这两种时间互相结合，取长补短，使系统性能进一步提高。这种系统实际上便是所谓的时序数据库系统 (Temporal Database system) [4]。

参 考 文 献

- [1] Adiba M. etc. HISTORICAL MUTIL-MEDIA DATABASES. Pro. of the 12th International Conference or VLDB, Aug, 1986.
- [2] Clifford J. and Tomsel A. On An Algebra For Historical Database: Two Views. Pro. of ACM-SIGMOD, 1985.
- [3] Snodgrass R. The Temporal Query Language TQuel. Pro. of ACM SIGACT-SIGMOD Symp. on Prin. of Database Systems, 1984.
- [4] Snodgrass R. etc. A Taxonomy of Time in Database. Pro. of the ACM-SIGMOD, 1985.

A Historional Database Management System — HXDB

Xu Rentao Wong Xiping Zhu Xingyan

Abstract

The historical relational database management system HXDB is extended from the relational database management system XDB. The system HXDB tries to keep the queries of the clerrent data fase, to process historical information easily, to make the abundance minimum, and as well as to by simple flexible.

[Key words] historical database; current data; valid time.