

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 200410068290.X

[45] 授权公告日 2008 年 11 月 12 日

[11] 授权公告号 CN 100432995C

[22] 申请日 2004.8.27

[21] 申请号 200410068290.X

[73] 专利权人 国际商业机器公司

地址 美国纽约州

[72] 发明人 沈华品 魏 恺 胡 炯

[56] 参考文献

CN1271906A 2000.11.1

US6018742A 2000.1.25

JP2002-259803A 2002.9.13

US2002/0184308A1 2002.12.5

审查员 郑守志

[74] 专利代理机构 北京市柳沈律师事务所

代理人 黄小临 王志森

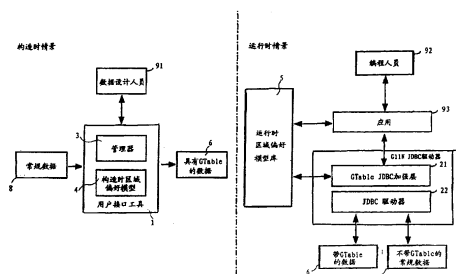
权利要求书 5 页 说明书 13 页 附图 6 页

[54] 发明名称

全球化数据库系统及其访问方法

[57] 摘要

本发明提出一种全球化数据库系统和用于访问全球化数据库的方法，该全球化数据库系统包括：区域偏好确定装置，用于从用户的应用确定区域偏好标识；数据库，其中存储全球化数据库表，其至少包括对应于用户查询的全球化列，其中每个全球化列中包括多个与不同的区域偏好相关的数据值；以及数据库访问驱动器，用于截取用户的数据库查询命令，根据从区域偏好确定装置取得的区域偏好标识，从所述全球化数据库表中的全球化列取得对应该区域偏好标识的与区域偏好相关的数据值。通过本发明的全球化数据库系统及其访问方法，可以在数据库中提供透明的区域偏好数据存取机制。



1. 一种全球化数据库系统，包括：

区域偏好确定装置，用于根据用户的应用确定区域偏好标识；

数据库，其中存储全球化数据库表，该全球化数据库表至少包括对应于用户查询的全球化列，其中每个全球化列中包括多个与不同的区域偏好相关的数据值；以及

数据库访问驱动器，用于截取用户的数据库查询命令，根据从区域偏好确定装置取得的区域偏好标识，从所述全球化数据库表中的全球化列取得对应应该区域偏好标识的与区域偏好相关的数据值。

2. 根据权利要求1所述的全球化数据库系统，其中所述全球化数据库表包括：

第一表，包括全球化数据库表标识列、全球化列和非全球化列，其中全球化数据库表标识列是第一表的主键；以及

第二表，包括全球化数据库表标识列、区域偏好标识列和全球化列，其中全球化数据库表标识列和区域偏好标识列是第二表的主键，全球化数据库表标识列是第一表的外键。

3. 根据权利要求2所述的全球化数据库系统，其中所述区域偏好标识列具有默认区域偏好标识，以便能够从第一表中的各全球化列取得对应应该默认区域偏好标识的与区域偏好相关的数据值。

4. 根据权利要求1所述的全球化数据库系统，其中：

所述全球化数据库表包括全球化数据库表标识列、全球化列和非全球化列，其中全球化数据库表标识列是全球化数据库表的主键；并且

所述数据库还包括视图，每个视图对应一个区域偏好标识，并且维护与全球化数据库表相同的数据库模式。

5. 根据权利要求1所述的全球化数据库系统，其中：

所述全球化数据库表包括全球化数据库表标识列、全球化列和非全球化列，

其中，全球化数据库表标识列是全球化数据库表的主键；全球化列采用结构列，用于维护数据结构，使得在一列中能够维护多个与不同的区域偏好相关的数据值。

6. 根据权利要求1所述的全球化数据库系统,其中数据库访问驱动器还包括查询命令转换装置,用于截取用户的查询命令,并且结合用户的区域偏好标识将用户的查询命令转换为能够访问全球化数据库表的形式。

7. 根据权利要求1所述的全球化数据库系统,其中所述区域偏好确定装置在运行时使用:

区域偏好登记接口装置,用于根据应用情况登记区域偏好标识;以及

区域偏好取得接口装置,用于由数据库访问驱动器取得所述登记的区域偏好标识。

8. 根据权利要求7所述的全球化数据库系统,其中所述区域偏好确定装置在运行时使用的区域偏好登记接口装置和区域偏好取得接口装置适合用于单机应用环境或分布式应用环境。

9. 根据权利要求2所述的全球化数据库系统,其中所述区域偏好确定装置在运行时使用标记调用级别接口装置,用于将访问全球化列的操作状态调整为以下状态之一:

状态1,访问操作与区域偏好相关,如果全球化数据库表没有对应当前区域偏好标识的数据值,在该区域偏好标识下访问全球化列的操作取得值为null(空)的记录;

状态2,访问操作与区域偏好相关,如果全球化数据库表没有对应当前区域偏好标识的数据值,在该区域偏好标识下访问全球化列的操作不能取得记录;以及

状态3,访问操作与区域偏好无关,仅仅对第一表进行,而对第二表没有影响。

10. 一种访问全球化数据库的方法,包括步骤:

在数据库中提供全球化数据库表,其至少包括对应于用户查询的全球化列,其中每个全球化列中包括多个与不同的区域偏好相关的数据值;

截取用户的数据库查询命令;

取得用户的区域偏好标识;

根据取得的区域偏好标识,从所述全球化数据库表中的全球化列取得对应应该区域偏好标识的与区域偏好相关的数据值。

11. 根据权利要求10所述的方法,其中所述全球化数据库表包括:

第一表,包括全球化数据库表标识列、全球化列和非全球化列,其中全

全球化数据库表标识列是第一表的主键；以及

第二表，包括全球化数据库表标识列、区域偏好标识列和全球化列，其中全球化数据库表标识列和区域偏好标识列是第二表的主键，全球化数据库表标识列是第一表的外键。

12. 根据权利要求 11 所述的方法，其中所述区域偏好标识列具有默认区域偏好标识，以便能够从第一表中的各全球化列能够取得对应该默认区域偏好标识的与区域偏好相关的数据值。

13. 根据权利要求 10 所述的方法，其中：

所述全球化数据库表包括全球化数据库表标识列、全球化列和非全球化列，其中全球化数据库表标识列是全球化数据库表的主键；并且

所述数据库还包括视图，每个视图对应一个区域偏好标识，并且维护与全球化数据库表相同的数据库模式。

14. 根据权利要求 10 所述的方法，其中：

所述全球化数据库表包括全球化数据库表标识列、全球化列和非全球化列，

其中，全球化数据库表标识列是全球化数据库表的主键；全球化列采用结构列，用于维护数据结构，使得在一列中能够维护多个与不同的区域偏好相关的数据值。

15. 根据权利要求 10 所述的方法，还包括：

在取得用户的区域偏好标识之后，结合用户的区域偏好标识，将用户的查询命令转换为能够访问全球化数据库表的形式。

16. 根据权利要求 10 所述的方法，还包括：

在截取用户的数据查询命令之前，登记用户的区域偏好标识，并且其中通过取得用户的区域偏好标识而获得的区域偏好标识是所述登记的区域偏好标识。

17. 根据权利要求 16 所述的方法，其中所述的登记区域偏好标识步骤和所述的取得登记的区域偏好标识步骤适合用于单机应用环境或分布式应用环境。

18. 根据权利要求 11 所述的方法，还包括将访问全球化列的操作状态调整为以下状态之一：

状态 1，访问操作与区域偏好相关，如果全球化数据库表没有对应当前

区域偏好标识的数据值,在该区域偏好标识下访问全球化列的操作取得值为null(空)的记录;

状态2,访问操作与区域偏好相关,如果全球化数据库表没有对应当前区域偏好标识的数据值,在该区域偏好标识下访问全球化列的操作不能取得记录;以及

状态3,访问操作与区域偏好无关,仅仅对第一表进行,而对第二表没有影响。

19. 一种全球化数据库系统,包括:

输入装置,用于输入构成数据库的常规数据;以及

全球化数据库表生成装置,用于从输入的常规数据提取与区域偏好相关的数据,并生成全球化数据库表,其包括至少一个全球化列,并将对应不同的区域偏好的多个同类数据值放入同一全球化列中,用于以后的数据库查询。

20. 根据权利要求19所述的全球化数据库系统,其中所述全球化数据库表包括:

第一表,包括全球化数据库表标识列、全球化列和非全球化列,其中全球化数据库表标识列是第一表的主键;以及

第二表,包括全球化数据库表标识列、区域偏好标识列和全球化列,其中全球化数据库表标识列和区域偏好标识列是第二表的主键,全球化数据库表标识列是第一表的外键。

21. 根据权利要求20所述的全球化数据库系统,其中所述区域偏好标识列具有默认区域偏好标识,以便能够从第一表中的各全球化列能够取得对应该默认区域偏好标识的与区域偏好相关的数据值。

22. 根据权利要求19所述的全球化数据库系统,其中:

所述全球化数据库表包括全球化数据库表标识列、全球化列和非全球化列,其中全球化数据库表标识列是全球化数据库表的主键;并且

所述数据库还包括视图,每个视图对应一个区域偏好标识,并且维护与全球化数据库表相同的数据库模式。

23. 根据权利要求19所述的全球化数据库系统,其中:

所述全球化数据库表包括全球化数据库表标识列、全球化列和非全球化列,

其中,全球化数据库表标识列是全球化数据库表的主键;全球化列采用

结构列，用于维护数据结构，使得在一列中能够维护多个与不同的区域偏好相关的数据值。

全球化数据库系统及其访问方法

技术领域

本发明涉及一种全球化数据库系统和访问全球化数据库（globalized database）的方法。

背景技术

在开发全球化的应用、特别是网络（Web）应用时，一些与区域偏好相关的数据（locale sensitive data）被存储在数据库而非属性文件中，这样的数据库称为全球化数据库。开发人员需要设计额外的表，以存储这些与区域偏好相关的数据，并且实现特定的数据库访问功能来处理这些数据。由于全球化的考虑与核心业务考虑如此混杂在一起，因而使用 JDBC（Java 数据库连接，Java DataBase Connectivity）或如 Hibernate（一种软件产品）的其它构架的数据库访问代码更加复杂，而且比不具有全球化考虑的常规数据库访问代码需要更多的维护工作量。区域偏好（locale）是作为用户对某一地域语言的偏好的标识，可以由地域标识和语言标识组成。

目前，开发人员对此问题使用应用专用的解决方案，而他们会早在设计阶段就捕捉全球化需求。对于新开发的系统，开发人员会按照下面的步骤在数据库访问层实现全球化特征：

- a) 对数据库中的多语言数据设计额外的表。
- b) 处理和维持用户的区域偏好（locale）信息，通常在多数接口中使用区域偏好参数。
- c) 编写 DB（数据库）访问代码，并且处理各特定语言的表以取得多语言数据。

开发人员会花费很多工作量来处理此全球化考虑。但实际上，这些工作量在多数情形是类似的。而且在维护阶段，全球化数据访问层还要大量成本来应付任何变化。

另外，在现有系统想赋予全球化特征的能力时，在数据存储层需要庞大的工作量。数据库中新增加的区域偏好数据导致数据库表模式（schema）及

其所有访问代码的许多变化。

一般地，开发人员应当通过下面的步骤修改 DB 访问层：

- a) 对数据库中的多语言数据设计额外的表。
- b) 得到用户的区域偏好信息，并且修改多数数据库访问接口。
- c) 修改 DB 访问代码，并且引入各特定语言的表以处理多语言数据。

在许多情形，这些工作量几乎等于重新生成该应用。

此外，如果系统的源代码不可得到，那么迁移将更难于进行。实际的方法是分别重写视图、控制器和模型：

- a) 对数据库中的多语言数据设计额外的表。
- b) 得到用户的区域偏好信息，并且编写一些代码以从视图部分转移到 DB 访问代码。

c) 重写 DB 访问代码，其可结合以前的数据库访问操作和新增加的数据库表上的操作。

这正像制作现有系统的壳(shell)，意味着大量的工作以及明显差的性能。

因此，应当在数据存储层引入将全球化考虑与核心业务考虑分离的新方法。这可以给开发人员提供透明的全球化数据存储特征，并且减轻他们的开发和维护工作。

发明内容

本发明的目的在于提供一种全球化数据库系统和访问全球化数据库的方法，以便提供一种管理和访问数据库中的多语言数据的通用和可重用的解决方案，使全球化考虑与核心业务考虑相分离，从而使开发人员从实现数据库访问层中的全球化特征所必需的重复工作中解脱出来。

为了实现上述目的，本发明提供一种全球化数据库系统，包括：区域偏好确定装置，用于根据用户的应用确定区域偏好标识；数据库，其中存储全球化数据库表，该全球化数据库表至少包括对应于用户查询的全球化列，其中每个全球化列中包括多个与不同的区域偏好相关的数据值；以及数据库访问驱动器，用于截取用户的数据库查询命令，根据从区域偏好确定装置取得的区域偏好标识，从所述全球化数据库表中的全球化列取得对应该区域偏好标识的与区域偏好相关的数据值。

本发明还提供一种访问全球化数据库的方法，包括步骤：在数据库中提

供全球化数据库表，其至少包括对应于用户查询的全球化列，其中每个全球化列中包括多个与不同的区域偏好相关的数据值；截取用户的数据库查询命令；取得用户的区域偏好标识；根据取得的区域偏好标识，从所述全球化数据库表中的全球化列取得对应该区域偏好标识的与区域偏好相关的数据值。

本发明还提供一种全球化数据库系统，包括：输入装置，用于输入构成数据库的常规数据；以及全球化数据库表生成装置，用于从输入的常规数据提取与区域偏好相关的数据，并生成全球化数据库表，其包括至少一个全球化列，并将对应不同的区域偏好的多个同类数据值放入同一全球化列中，用于以后的数据库查询。

通过使用全球化数据库表（GTable）以及构造时和运行时区域偏好模型，可以在数据库提供透明的区域偏好数据存取机制。其主要优势在于：

a) 在开发新的赋予全球化特征的能力的系统时，开发人员可将注意力集中在其核心业务功能上，并且使用通常的数据库访问功能取得这些多语言数据。

b) 对 GTable 的支持对客户代码或现有的数据库访问代码几乎是透明的。

c) 需要赋予全球化特征的能力的现有系统不需要改变数据库访问代码以容纳多语言数据。

d) 作为 c) 的结果，降低了访问和维护数据库中的区域偏好数据的成本，开发人员可以用比较小的工作量并且更容易地在数据库存储层获得全球化特征。

附图说明

图 1 显示了在构造时和运行时两个情景下本发明的概览。

图 2 说明了根据本发明的一个实施方式的 GTable 的内部结构。

图 3 是在图 2 所示的各表的基础上说明本发明的实现的一个实例。

图 4 说明单一 JVM（Java 虚拟机）环境中的运行时区域偏好模型。

图 5 说明分布式环境中的运行时区域偏好模型的一个实例。

图 6 是根据本发明的一个实施方式的访问全球化数据库的方法的流程图。

具体实施方式

实施本发明的前提是使用关系数据库和在数据库中支持如 UTF-8 的 unicode（全球统一编码）的存储。不满足这些前提的现有系统，在实施本发明前需要首先转换以满足这些要求。在多数情形，特别是在近期的系统中，这两个前提是很容易满足的。

首先来简单说明全球化数据库表（简称 GTable）的概念。

本发明引入新类型的数据库表，称为全球化数据库表。在 GTable 中，可以有許多全球化列（简称 GColumn）。对每个 GColumn，在一个 GTable 记录中会存在不同的对应区域偏好的值。常规数据库表可转换为全球化数据库表类型，而全球化数据库表的常规非键列（non-key column）也可转换为全球化列以存储与区域偏好相关的数据。在实现时，本发明提供 G11N 数据库驱动器，其实际上在现有数据库驱动器上增加了一薄层以支持此全球化数据库表概念。下文会结合附图来详细说明。

其次，简单说明构造时（build time）和运行时（run time）区域偏好模型的概念。区域偏好模型对此全球化数据库存取机制非常重要，本发明从构造时和运行时的观点分别将其引入。

在构造时，使用新引入的系统表记录该数据库中所有支持的区域偏好，而 GTable 看起来正像常规的数据库表。然后，开发人员可使用相同的代码访问这些常规数据库表和 GTable。为了实现这点，本发明还需要获得 GTable 访问操作的运行时区域偏好，而不是直接从访问代码中的参数获得。本发明使用全局性的区域偏好库（locale repository）管理每个请求的当前区域偏好。开发人员只需要在适当地方（通常在一些服务开始时）登记其区域偏好。这样，区域偏好是一个过程中的环境变量。对于分布式系统，下文中将引入其它方法来在不同过程之间传播区域偏好。在 GTable 访问操作中，运行时区域偏好和构造时区域偏好模型都将有助于确定与区域偏好相关的数据的正确目标。

以下，参照附图来详细说明本发明的上述概念和实施方式。

用于数据库中的全球化数据库表、构造时和运行时区域偏好模型的概念可在不同平台上使用不同技术实现。在本发明的优选实施方式中，使用了 Java 技术和 IBM DB2（IBM 公司的一种数据库管理系统软件产品）数据库管理系统。图 1 以构造时和运行时两个情景显示了本发明的概览。

参照图 1，在构造时情景，数据设计人员 91 可利用 GTable UI（用户接

口) 工具 1 生成 GTable, 从而设计与其业务应用模式相关的数据库模式。GTable UI 工具 1 可以视为一种全球化数据库表生成装置, 其作用是: 在本发明提供的 GTable 管理器 3 的平台上使用构造时区域偏好模型 4 按照特定应用进行配置, 从而将需要整合全球化特征的常规数据 8 自动转换为利用 GTable 组织的全球化数据 6, 以支持多语言应用。常规数据 8 通常由适当的输入装置 (未显示) 输入数据库中, 形成常规的数据库表。因此, 构造时情景反映了设计时的观点。

而运行时情景反映实现时的观点。在运行时情景, 编程人员 (开发人员) 92 编写特定应用 93, 编程人员 92 的注意力集中在特定应用 93 的核心业务功能上, 并且使用通常的数据库访问功能取得多语言数据, 而不需要考虑全球化数据库访问的细节, 即全球化数据库访问对编程人员 92 是透明的。全球化数据库访问由 G11N JDBC 驱动器 2 完成。G11N JDBC 驱动器 2 是在常规的 JDBC 驱动器 22 上增加了 GTable JDBC 加强层 (GTable 核心) 21, 其截取来自应用 93 的查询命令, 通常是 SQL 查询语句, 然后将其转换为可以访问 GTable 的形式。在转换时, 作为查询命令转换装置的 GTable JDBC 加强层 21 将从运行时区域偏好模型库 5 获得的当前区域偏好包含在转换后的查询命令 (包含更多参数的 SQL 查询语句) 中; 而当前区域偏好是在特定应用 93 在适当时 (通常在启动时) 登记在运行时区域偏好模型库 5 中的。然后, GTable JDBC 加强层 21 将转换后的查询命令委托给常规的 JDBC 驱动器 22, 由此根据转换后的查询命令直接访问下层的数据库, 从 GTable 6 中取得要查询的对应当前区域偏好的多语言数据。对于查询非全球化数据库表的查询命令, GTable JDBC 加强层 21 不进行转换, 直接将其委托给 JDBC 驱动器 22, 以访问非 GTable 的数据库表 7。在图 1 中, 虽然 GTable (带 GTable 的数据) 6 和非 GTable (不带 GTable 的常规数据) 7 显示为分开的数据库模块, 但这不应被视为限制, 这两类数据也可以存储在同一数据库中。另外, 图 1 中的运行时区域偏好模型库 5 是区域偏好确定装置, 用于从各种各样的应用确定区域偏好, 并且可以存储区域偏好, 以便在 G11N JDBC 驱动器 2 转换 SQL 查询语句时可以获得正确的区域偏好。运行时区域偏好模型库 5 的细节将在下文进行描述。

下面将从 GTable 核心、构造时和运行时区域偏好模型以及 GTable 管理工具三部分来详细说明本发明。

1. GTable 核心

GTable 是本发明的关键。本发明从开发人员的观点提供 G11N JDBC 驱动器，开发人员可获得由 GTable 透明地提供的所有特征。实际上，本发明仅仅提供了现有 JDBC 驱动器上微弱的一层，以整合 GTable 概念，例如，本实施方式使用 DB2 JDBC 驱动器作为基础。在本实施方式中，一个 GTable 在下层数据库中由两个常规表表示，第一表是包括用户可以访问的所有列的主 GTable，第二表是仅仅用于全球化列的从 GTable。从 GTable 对开发人员是隐藏而不可直接访问的，并且具有区域偏好 ID（标识）列，该区域偏好 ID 列是系统区域偏好表（将在下文说明）的外键。此外，每个表具有默认的区域偏好，其可对表特别设置或从数据库的默认区域偏好属性继承。主 GTable 中的各 GColumn 具有对应表的默认区域偏好的各值，而 GColumn 的其它多语言数据保存在隐藏的从 GTable 中。如果 GTable 的全球化特征关闭（将在下文说明），GTable 上的操作只能影响主 GTable 中的对应表的默认区域偏好的各值。主 GTable 和从 GTable 之间的连接由主键和外键对来维护。

图 2 显示了根据本发明的一个实施方式的 GTable 的内部结构，说明以上概念。图 2 左边部分的表 61 表示构造时在开发人员观点下的 GTable 的结构，它体现的是核心业务考虑。表 61 显示了一个 ScenicSpot 表的例子，其主键是全球化表 ID（标识）SCENICSPOT_ID，该表具有两个全球化列 SCENICSPOT_NAME 和 SPOT_INTRODUCTION，还有一个非全球化列 SCENICSPOT_TYPE。全球化列 SCENICSPOT_NAME 和 SPOT_INTRODUCTION 分别具有多个语言的数据，对应各区域偏好，如 SCENICSPOT_NAME 在 SCENICSPOT_ID=1 时，SCENICSPOT_NAME 可能是“西湖”或“west lake”等对应各种语言的名称。

图 2 右边部分表示运行时实现表 61 的主 GTable 62 和从 GTable 63 的结构，其中主 GTable 62 包含了表 61 的所有列，而从 GTable 63 仅仅包含并且完全包含必需的全球化列（这里是 SCENICSPOT_NAME 和 SPOT_INTRODUCTION）。此外，除了主键 SCENICSPOT_ID，从 GTable 63 还包含识别区域偏好的主键 LOCALE_ID（区域偏好 ID），它是系统区域偏好表的外键，由此可唯一识别对应每个区域偏好的全球化记录。也就是说，在从 GTable 中，一个记录中的每个列唯一对应一个多语言数据值。而非全球化列 SCENICSPOT_TYPE 不出现在从 GTable 中。从图 2 可见，对于主 GTable 62，

SCENICSPOT_ID 也是其外键，主 GTable 62 和从 GTable 63 之间的连接由其各自的主键 SCENICSPOT_ID 形成的主键和外键对来维护。

与数据库管理系统相互作用的应用在底层使用 SQL 语言，不管开发人员使用什么构架。GTable JDBC 加强层截取与 GTable 有关的 SQL 查询，并且根据当前线程环境区域偏好，将该 SQL 查询转换以访问对应的隐藏表，而且还可以增加区域偏好条件。在改变（转换）SQL 命令后，GTable JDBC 加强层通过将改变的 SQL 命令传递给下层 JDBC 驱动器，从而将各功能委托给下层 JDBC 驱动器。对于客户代码，所有这些是透明的。

可配置 GTable JDBC 加强层的特性以适合多种多样的现有要求。在实际情况中，在 GTable 上开发人员可既需要与区域偏好相关的操作，也需要与区域偏好不相关的操作。因此，本发明提供了 GTable 上的开关以允许和禁止全球化特征。有三个开关状态：ON、STRICT-ON 和 OFF。在表 1 中说明了这些开关状态各自表示的特性。

表 1 GTable 的不同特性

开关状态	特性描述
ON (默认)	在该状态，数据库访问操作与区域偏好相关；GColumn 中的多语言数据将使用当前区域偏好正确处理。并且如果 GTable 没有当前区域偏好的多语言数据，选择该区域偏好下的操作将在其 GColumn 中仍然能获得具有 null（空）的记录。
STRICT-ON	在该状态，数据库访问操作仍然与区域偏好相关；GColumn 中的多语言数据将使用当前区域偏好正确处理。但如果 GTable 没有当前区域偏好的多语言数据，选择该区域偏好下的操作将不能获得记录。
OFF	在该状态，数据库访问操作与区域偏好无关；所有数据库访问操作是对主 GTable 进行，并且对从 GTable 没有影响。

提供这样的开关允许开发人员对 GTable JDBC 加强层进行定制。例如，在转换来自客户代码的 SQL 命令时，ON 状态表示 SQL 命令中的 FROM 语句的结合类型是左联接（left join）或右联接（right join）；STRICT-ON 状态表示 SQL 命令中的 FROM 语句的结合类型是内联接（inner join）；而 OFF 状态表示 SQL 命令中的所有操作仅仅针对主 GTable 中的对应默认区域偏好的各值。

由此,这些开关状态可在运行时经由本发明提供的 API (应用编程接口) 改变。这样,开发人员还可对 GTable 进行一些与区域偏好无关的操作。对于现有系统,在将一些常规表改变为 GTable 后,开发人员可发现对一些 GTable 的一些操作仍然应当是与区域偏好无关的,那么他们可增加代码在数据库访问操作前关闭这个开关,使得数据库访问操作在应用 GTable 前后没有变化。

在引入 GTable 和 GColumn 后,SQL 的语法 (semantics) 是相同的,而对 GColumn 的下层操作比对常规列的操作改变了一点。GTable 记录在其 GColumn 中可具有许多可能值。但在环境区域偏好确定后,可将 GTable 记录仅仅视为常规记录并对其执行 SQL 命令。此方法对客户代码是相当透明的,并且将全球化考虑与核心业务逻辑分离。在本实施方式中,使用配置文件来保存对数据库和 GTable 的所有设置。而且这些配置将由 GTable JDBC 加强层使用。

引入该方法后的性能影响不大。GTable 元数据 (meta data) 仅仅在开始时装入一次。对于 GTable,要由开发人员处理的操作现在由 GTable JDBC 加强层处理:涉及 GColumn 时,就进行 SQL 转换,然后委托给下层 JDBC 驱动器;不涉及 GColumn 时,就直接委托给下层 JDBC 驱动器。而对于非 GTable 上的操作,仅仅将其委托给下层 JDBC 驱动器,没有显著的性能影响。因此,引入该方法只是引入一些检查和 SQL 转换工作,这对性能将不会有大的影响。

图 3 是在图 2 所示的各表的基础上说明本发明的实现的一个实例。特定应用请求访问全球化数据库以查询多语言数据。该特定应用在访问代码中包含 SQL 查询命令来完成所述查询,如图 3 中的标号 93 所示的 SQL 语句:

```
Select * from ScenicSpot where SCENICSPOT_ID=1
```

G11N JDBC 驱动器 2 通过其 GTable JDBC 加强层 21 截取该 SQL 语句,准备进行 SQL 语句的转换。GTable JDBC 加强层 21 从区域偏好库 51 获得当前区域偏好,然后将上述 SQL 语句转换为:

```
Select    ScenicSpot.SCENICSPOT_ID  as  SCENICSPOT_ID,  
ScenicSpot_EXTRA.SCENICSPOT_NAME  as  SCENICSPOT_NAME,  
ScenicSpot.SCENICSPOT_TYPE  as  SCENICSPOT_TYPE,  
ScenicSpot_EXTRA.SPOT_INTRODUCTION as SPOT_INTRODUCTION  
from ScenicSpot, ScenicSpot_EXTRA where SCENICSPOT_ID=1 AND  
ScenicSpot_EXTRA.LOCALE_ID='zh_CN'
```

其中已经将当前区域偏好 zh_CN 包含在转换后的 SQL 语句中。GTable JDBC 加强层 21 将转换后的 SQL 语句委托给下层 JDBC 驱动器 22, 由其访问下层的数据库, 根据从图 2 所示的主 GTable 和从 GTable 取得查询结果 (数据库存储的多语言数据) 6, 其中从主 GTable 中取得 SCENICSPOT_ID=1 的非全球化数据 SCENICSPOT_TYPE = ST101, 根据从 GTable 中的区域偏好 LOCALE_ID=zh_CN, 从从 GTable 取得 SCENICSPOT_ID=1 的全球化数据 SCENICSPOT_NAME=西湖和 SPOT_INTRODUCTION=最忆是江南。

因此, GTable 的概念是形成本发明提出的通用和可重用的解决方案的基础。这样的概念使开发人员从实现数据库访问层中的全球化特征所必需的重复工作中解脱出来。从开发人员的观点, GTable 是能够具有多个 GColumn 和默认区域偏好属性的表。

当然, 图 2 所示的 GTable 的内部结构仅仅是本发明的一个实施方式的示例, 实现本发明的 GTable 也能够采用其它形式。例如, 在图 2 中, 保留主 GTable 结构, 而创建附加视图来维护多语言数据, 一种语言对应一个视图。所有视图通过查询主 GTable 而创建, 并且维护与主 GTable 相同的 DB 模式 (schema)。

另外一种 GTable 的实现可以采用结构列, 在列中采用维护一个数据结构来代替简单数据类型。可以使用结构列的这一特征使其在一列中维护所有多语言信息, 从而只有主 GTable 而不需要附加的表来支持 GTable。一些如 DB2 和 Oracle 的数据库能支持结构列。

2. 构造时和运行时区域偏好模型

本发明提出的区域偏好模型提供透明的 GTable 支持, 将区域偏好设计为环境属性而不是函数接口参数。在构造时, 本发明引入另一个新表, 即系统区域偏好表来管理数据库中支持的区域偏好。将在后面说明的管理工具将处理这些区域偏好设置。对于数据库和表, 存在一个与其关联的默认区域偏好属性。而且, 如果表的默认区域偏好没有特别设置, 那么它可从数据库的默认区域偏好继承。表的默认区域偏好决定哪些与区域偏好相关的值存储在主 GTable 的 GColumn 中。

在运行时, 本发明提供图 1 中所示的区域偏好模型库 5, 以使开发人员登记/取得当前区域偏好, 并且标记调用级别 (改变在表 1 中引入的开关状态)。这个库可能使用不同技术在不同环境实现定义的公共接口。开发人员也可定

制其自己的应用来实现给定的接口。接口包括三个方法：

1. 登记区域偏好
2. 标记调用级别（可选择的）
3. 取得区域偏好

在典型过程中，一旦接收到用户的请求后，开发人员将调用“登记区域偏好”接口。之后 GTable JDBC 加强层将调用“取得区域偏好”接口，以自动获得运行时区域偏好。所有用户的现有代码不需要改变来传送区域偏好信息。“标记调用级别”接口是可选择的，用于在运行时由开发人员根据应用的实际需要调整表 1 中引入的 GTable 的开关状态。例如在前面说明地，一些特定应用中 GTable 的一些操作仍然应当是与区域偏好无关的，那么可增加代码在数据库访问操作前关闭 GTable 的开关（OFF），使得特定数据库访问操作对从 GTable 没有影响。

图 4 说明单一 JVM（Java 虚拟机）环境中的运行时区域偏好模型。

在单机应用环境中，客户应用的代码应该在某个适当时间使用运行时区域偏好库设置环境区域偏好，并且如果对一个线程没有定义区域偏好，就将使用系统的默认区域偏好。区域偏好信息被置于线程本地存储中，因此确保 G11N JDBC 驱动器将获得作为登记的区域偏好的同一区域偏好标识。

例如在图 4 中，在单一 JVM 中，应用程序（实际应用）的线程 94 包含了对全球化数据库 61 的查询命令。首先线程 94 取得用户的区域偏好设置，在 ST41 调用“登记区域偏好”接口将有关用户的区域偏好的这个信息（区域偏好句柄或区域偏好 ID）放到线程本地存储，即区域偏好库 51 中。G11N JDBC 驱动器 2 在 ST43 调用“取得区域偏好”接口，自动抽取这个信息，并且整合入线程 94 的数据库查询（SQL 命令）当中，即包含在 G11N JDBC 驱动器 2 转换的 SQL 命令中，从而可以从下层数据库 61 中取得需要的对应该区域偏好 ID 的全球化数据值。图 4 中的线程 95 执行与线程 94 相同的处理，只是使用线程 95 自身的线程本地存储，即区域偏好库 52，并且从下层数据库 62 中取得需要的全球化数据值。由图 4 可见，区域偏好库 51 和 52 分别是线程 94 与线程 95 的线程本地存储，两个线程本地存储是分开的，也就是说，本发明涉及的区域偏好库是一种逻辑库概念。如上所述，图 4 中的数据库 61 和数据库 62 并不限制于分离的数据库，而是可以存在于同一数据库中。图 4 中的 ST42 所示的调用“标记调用级别”接口是可选择的，其作用是将作为对 GTable

JDBC 加强层的定制的如上所述的开关状态作为配置参数存储在作为线程本地存储的区域偏好库 51 或 52 中。而 G11N JDBC 驱动器 2 在 ST43 调用“取得区域偏好”接口时,也自动抽取开关状态参数信息,并且整合入 G11N JDBC 驱动器 2 转换的 SQL 命令中。

下面考虑分布式环境。在此情形,运行时区域偏好模型需要不同的实现。例如,图 5 说明分布式环境中的运行时区域偏好模型的一个实例。图 5 所示的环境是 WebShpere 应用服务器 V5 企业版(WebShpere Application Server V5 Enterprise Edition,简称 WAS V5)。实际上,WAS 具有其自身的国际化服务来维护多个 WAS 服务器之间的区域偏好信息,那么可基于 WAS V5 中的国际化服务来实现运行时区域偏好模型。于是,开发人员只需要在适当时间登记区域偏好设置值,而不管复杂的系统结构。

参照图 5,采用 WAS V5 的请求处理器 96 与采用 WAS V5 的数据对象 97 在分布式环境中相互分开。请求处理器 96 处理来自用户的请求,特别是查询全球化数据的请求。然后请求处理器 96 调用远程 EJB (Enterprise Java Beans) 而不考虑全球化特征,即不带区域偏好句柄;远程 EJB 驱动数据对象 97 完成数据库访问,该数据库访问委托给本地的 G11N JDBC 驱动器 2。请求处理器 96 从如带有区域偏好句柄的 HTTP (超链接文本协议) 请求的用户请求中取得用户的区域偏好设置(区域偏好 ID),在 ST51 调用“登记区域偏好”接口将有关用户的区域偏好的这个信息(区域偏好句柄或 ID)放到本地区域偏好库 51 中。WAS V5 的 i18N 服务可以自动同步请求处理器 96 的本地区域偏好库 51 和数据对象 97 (或者说与数据对象 97 直接耦合的 G11N JDBC 驱动器 2)的本地区域偏好库 52,使得 G11N JDBC 驱动器 2 在 ST54 调用“取得区域偏好”接口,自动抽取在 ST51 登记的区域偏好 ID,并且整合入应用程序的数据库查询(SQL 命令)当中,即包含在 G11N JDBC 驱动器 2 转换的 SQL 命令中,从而可以从下层数据库 6 中取得需要的对应该区域偏好 ID 的全球化数据值。图 5 中的 ST52 和 ST53 所示的调用“标记调用级别”接口是可选的,其作用是将如上所述的开关状态作为配置参数存储在本地区域偏好库 51 或 52 中。而 G11N JDBC 驱动器 2 在 ST54 调用“取得区域偏好”接口时,也自动抽取该开关状态参数信息,并且整合入 G11N JDBC 驱动器 2 转换的 SQL 命令中。

3. GTable 管理工具

现有数据库管理系统中没有 GTable 的概念，因此，有必要给开发人员提供 GTable 管理工具，即图 1 中所示的 GTable 管理器 3。GTable 管理工具也维护构造时和运行时之间的 GTable 元数据的一致性，而所有 GTable 元数据可能存储在文件系统或数据库中，使得运行时 G11N JDBC 驱动器可以取得这些元数据处理 SQL 转换。GTable 元数据包括例如 GTable 的表名称列表、GColumn 的名称列表、GTable 和 GColumn 的关系列表。该管理工具主要包括以下功能：

- a) 将常规表转换为 GTable 类型，或者进行反向转换。
- b) 将常规列转换为 GColumn 类型，或者进行反向转换。
- c) 管理数据库中构造时支持的区域偏好，这实际上是通过操作系统区域偏好表完成的。
- d) 为数据库和 GTable 设置默认区域偏好。
- e) 帮助导入和导出 GTable 的多语言数据。

下面，参照图 6 说明根据本发明的一个实施方式的访问全球化数据库的方法，图 6 显示了所述访问方法的流程。在步骤 601，在数据库中提供全球化数据库表，其结构可以是如图 2 所示的包括主 GTable 和从 GTable，其特征如上所述。在步骤 602，用户应用在适当时间（通常是开始时）登记其区域偏好标识，登记过程可采用图 4 或图 5 所示的用于不同应用环境的适当方式。在步骤 603，G11N JDBC 驱动器 2 截取用户应用中的数据查询命令（SQL 命令）。在步骤 604，采用图 4 或图 5 所示的用于不同应用环境的适当方式，G11N JDBC 驱动器 2 取得用户应用登记的区域偏好标识。在步骤 605，用户应用可选择是否改变 GTable 的开关状态。如果在步骤 605 选择改变 GTable 的开关状态（是），就在步骤 606 将开关状态参数传递给 G11N JDBC 驱动器 2，具体的传递方式可采用在图 4 或图 5 的说明中描述的方式。如果在步骤 605 选择不改变 GTable 的开关状态（否）或者完成步骤 606 之后，就转到主流程，即步骤 607。虽然图 6 中显示步骤 605 和 606 是在步骤 604 之后，但也可以在步骤 604 之前与步骤 602 登记区域偏好标识的同时，此时完成步骤 606 之后就转到步骤 603。也就是说，可以根据特定应用来选择步骤 605 和 606 的适当位置。在步骤 607，G11N JDBC 驱动器 2 将用户应用中的 SQL 命令转换为可以直接访问主 GTable 和从 GTable 的形式，其中包含了在步骤 604 取得的用户应用登记的区域偏好标识和在步骤 606 传递的开关状态参数。在步骤

608, 通过转换的 SQL 命令, 从主 GTable 和从 GTable 的全球化列取得对应所述登记的区域偏好标识的与区域偏好相关的数据值, 然后流程结束。

值得注意的是, 本发明并不限于上述实施方式, 在不脱离本发明范围的情况下, 可以进行形式和细节上的各种变形和修改。

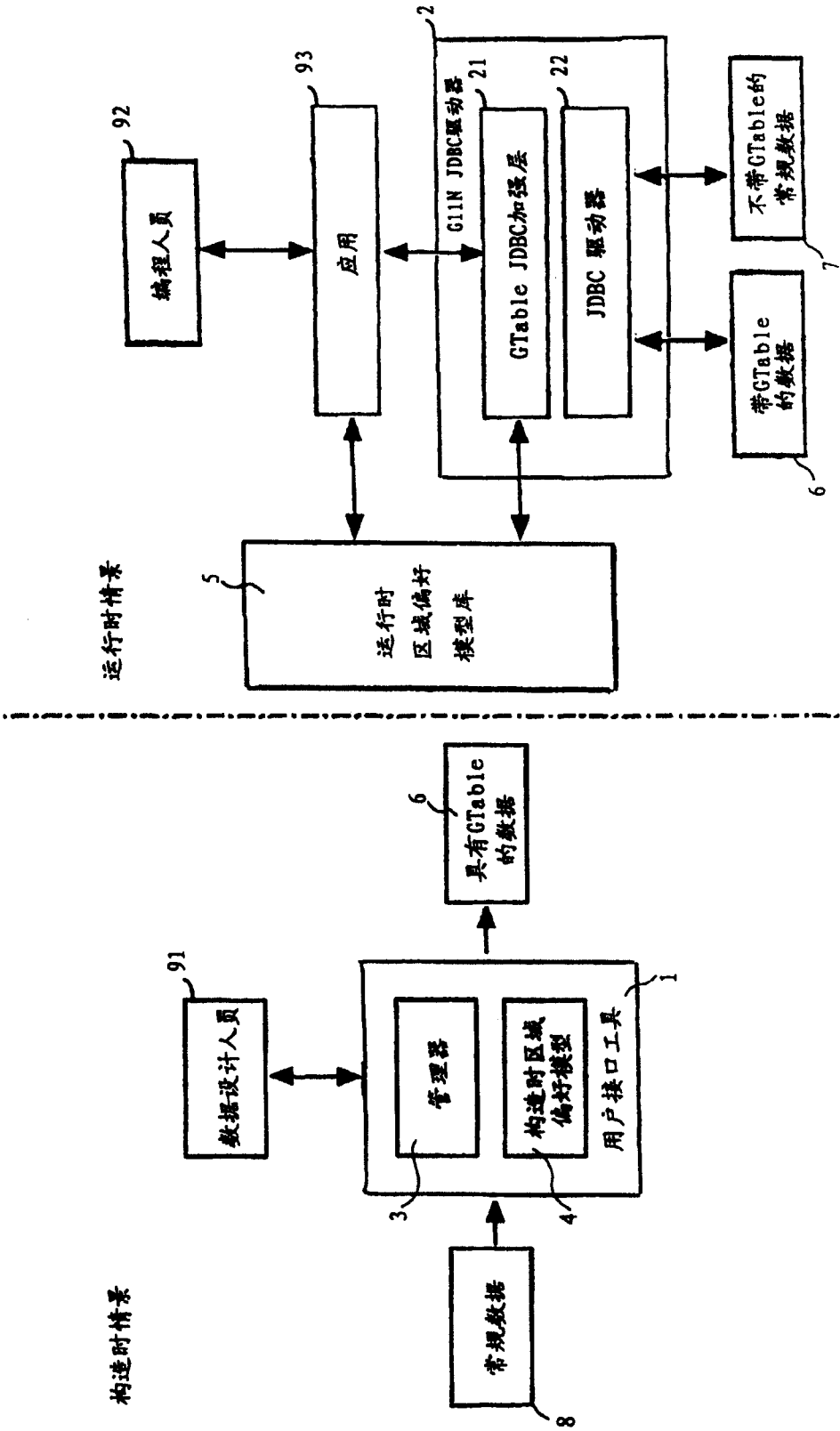


图 1

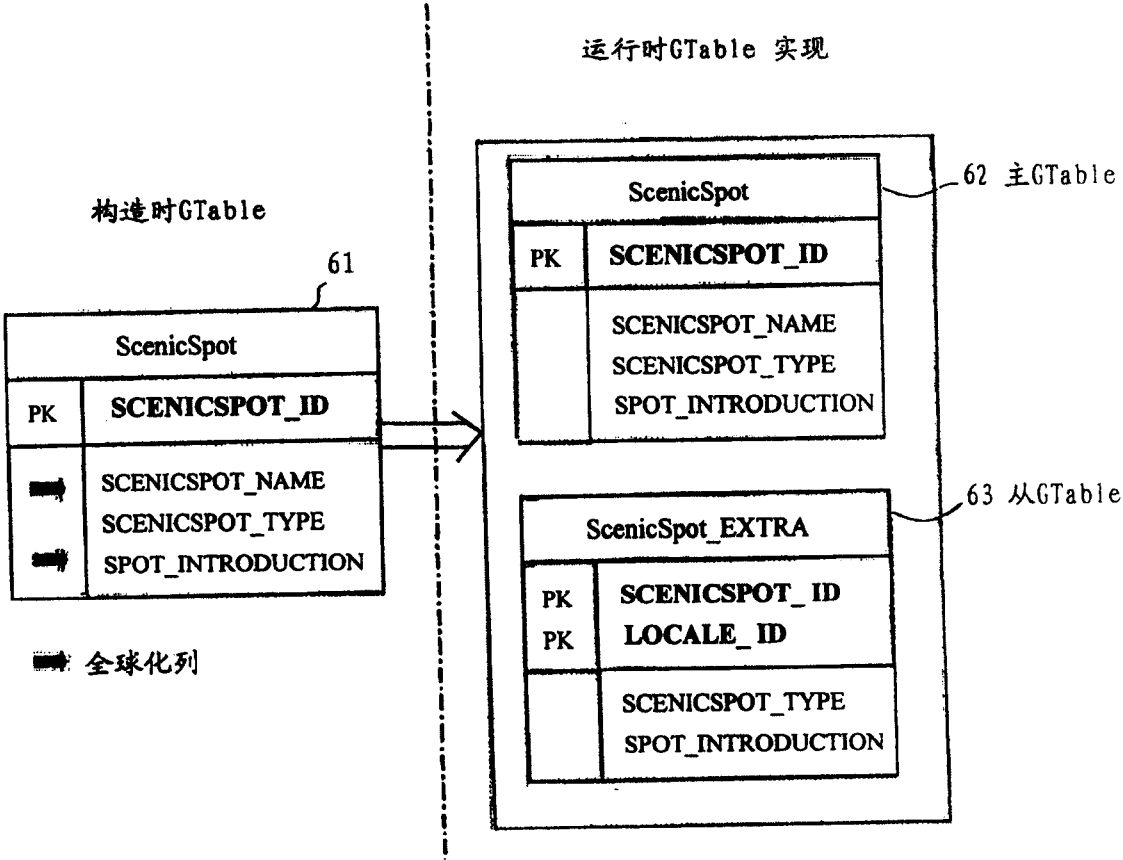


图 2

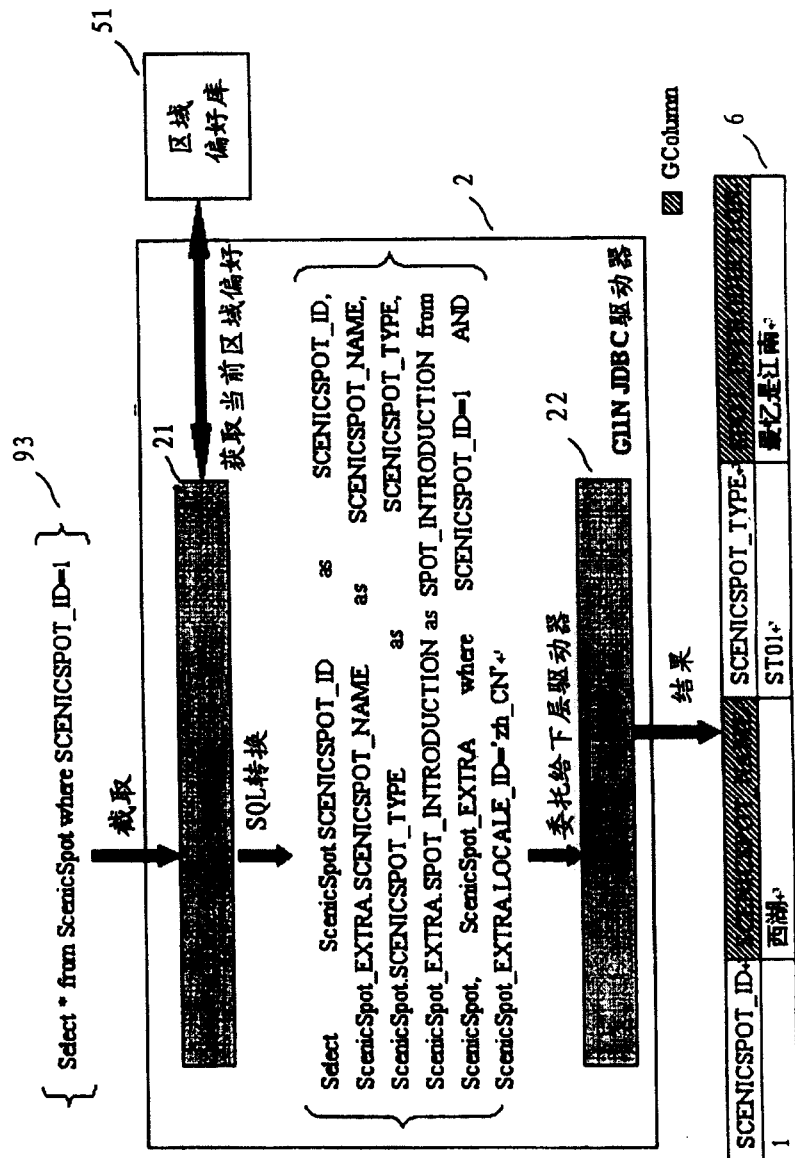


图 3

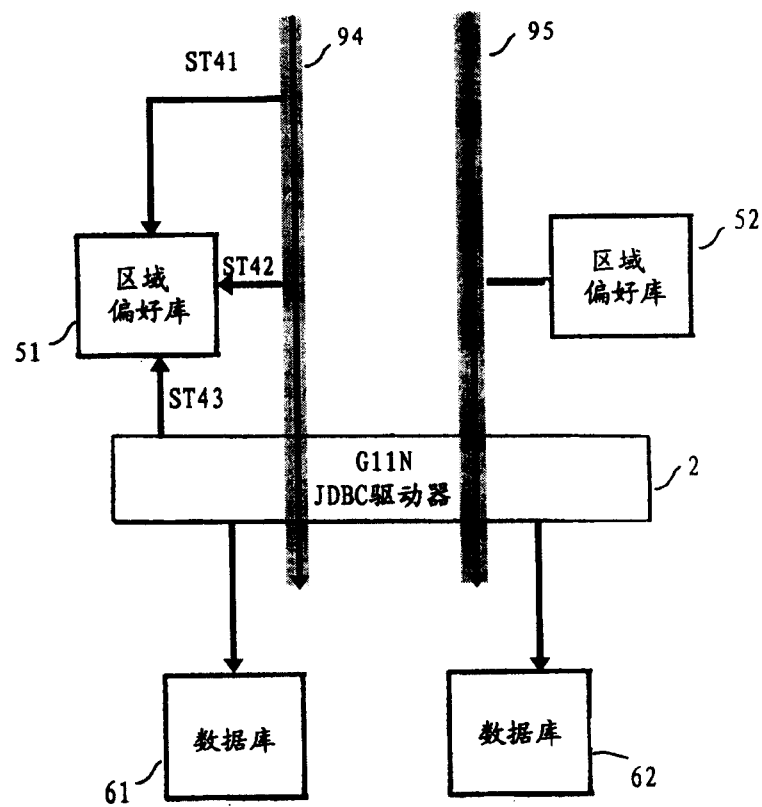


图 4

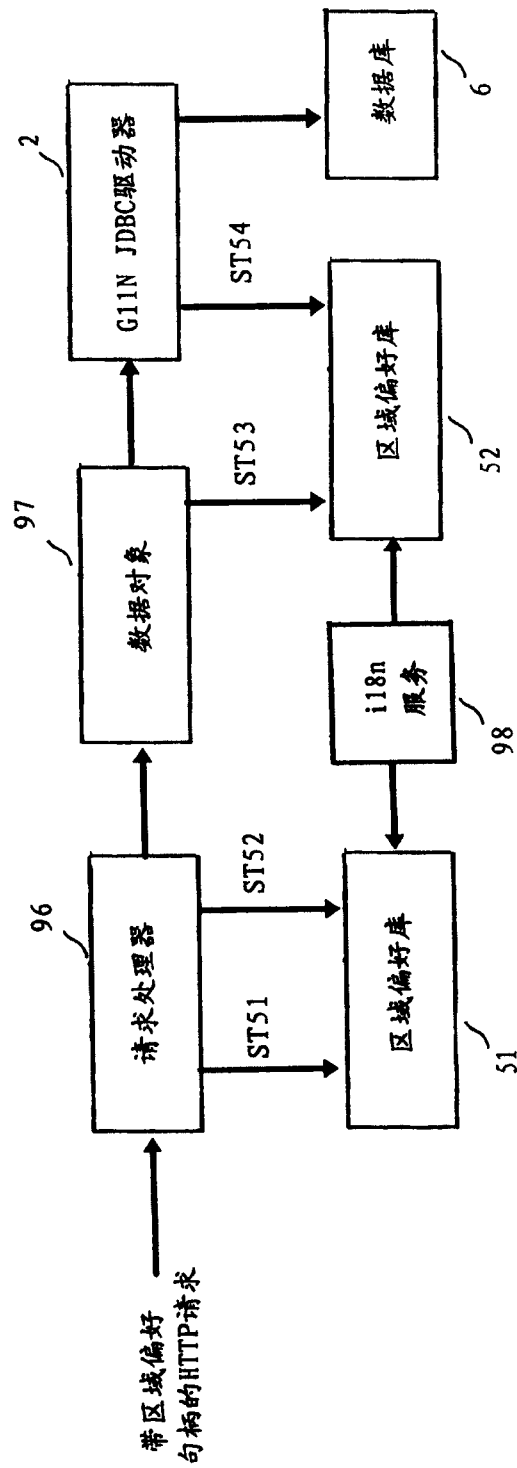


图 5

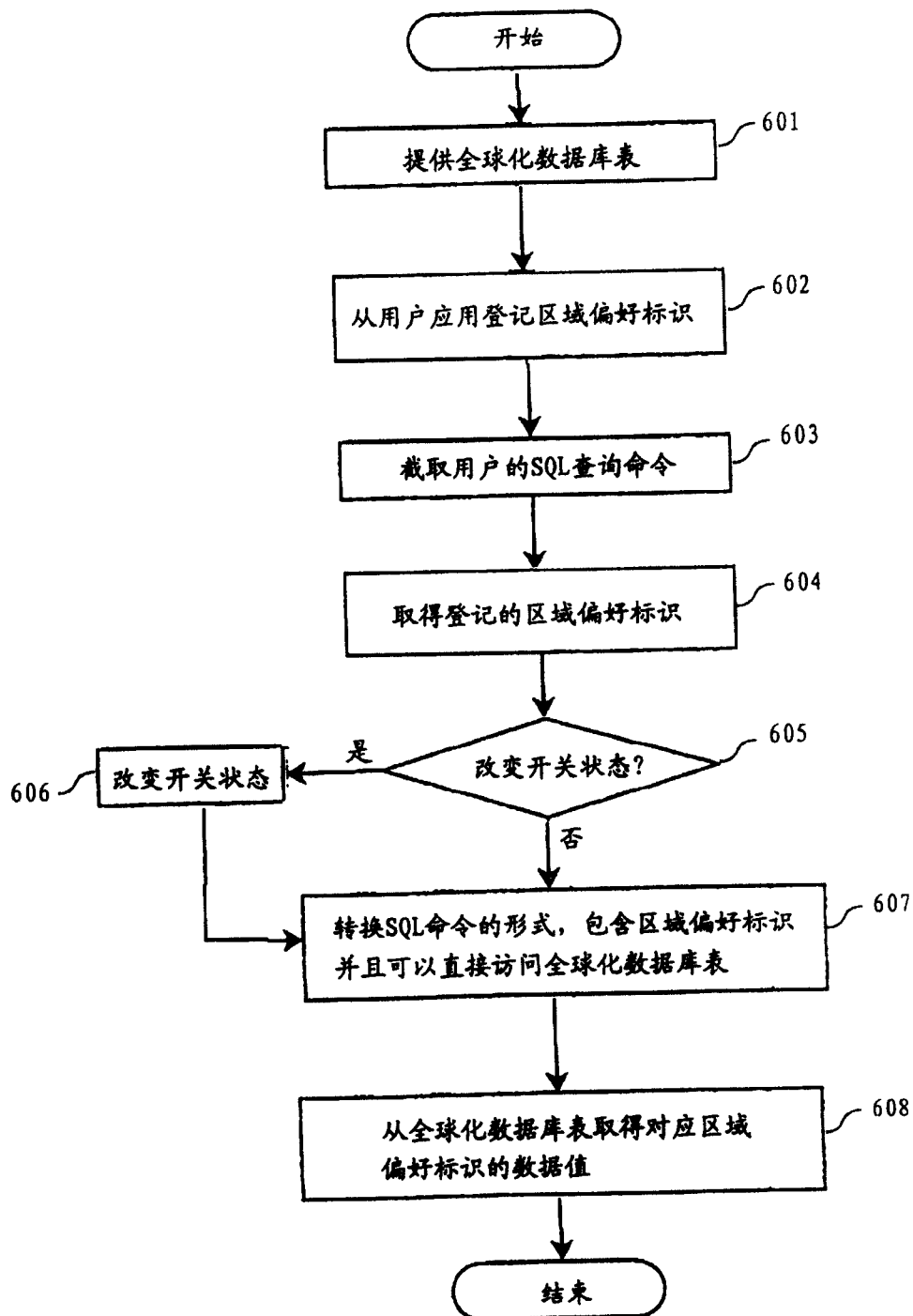


图 6