计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
43(6): 1097~1103, 2006

增量 ETL 过程自动化产生方法的研究

张旭峰　孙未未　汪　卫　冯雅慧　施伯乐
(复旦大学计算机与信息技术系　上海　200433)
(011021380@fudan.edu.cn)

Research on Generating Incremental ETL Processes Automatically

Zhang Xufeng, Sun Weiwei, Wang Wei, Feng Yahui, and Shi Baile
(*Department of Computing and Information Technology, Fudan University, Shanghai* 200433)

Abstract　ETL processes are used for collecting data from data sources to data warehouse. ETL processes can be separated into two portions: full ETL processes and increment ETL processes. A full ETL process can be designed easily but it can only deal full data. An incremental ETL process is used for loading only those data which are newly created in the data sources, but it is difficult to design manually. In this paper, using existing methods of incremental maintenance of materialized views for reference, an approach to generate an incremental ETL process automatically from a full ETL process is put forward. Existing researches are focused on the incremental maintenance of materialized views in such circumstances which involve the operators of selection, projection, join and aggregation but not the difference operators. Since difference operators are used frequently in an ETL process, incremental maintenance of materialized views defined with difference operators is also discussed in detail.

Key words　ETL; data warehouse; incremental maintenance; materialized views; self-maintenance

摘　要　ETL 过程用于将数据从数据源装载到数据仓库中,它可以被划分为两种类型:全量 ETL 过程和增量 ETL 过程. 全量 ETL 过程只能处理全量数据,但易于设计. 而增量 ETL 过程设计起来比较复杂,但适用于处理增量数据. 主要对增量 ETL 过程的自动化产生方法进行了研究,根据已有的全量 ETL 过程,可以自动产生增量 ETL 过程,从而降低设计增量 ETL 过程的代价. 利用已有的物化视图增量维护的方法,给出了根据全量 ETL 过程自动产生增量 ETL 过程的方法. 但是已有的研究集中在包含选择、投影、联接和聚合运算情况下物化视图的增量维护,未见对包括差运算情况下的讨论. 作为研究工作的基础,还详细讨论了包含差运算情况下物化视图的增量维护问题.

关键词　ETL;数据仓库;增量维护;物化视图;自维护

中图法分类号　TP311.13

1　引　　言

数据仓库将多个数据源的数据集中起来用于分析和处理,将数据从数据源装载到数据仓库的过程称为 ETL(extract, transform, and load)过程.

ETL 过程可以被划分为两种类型:全量 ETL 过程和增量 ETL 过程. 全量 ETL 过程一般用于数据仓库的初始化,而增量 ETL 过程则用于数据仓库的增量维护. 相对于全量 ETL 过程而言,增量 ETL

收稿日期:2005-01-14;修回日期:2005-08-25
基金项目:国家"八六三"高技术研究发展计划软件重大专项基金项目(2002AA4Z3430)

过程设计更复杂.

以往的文献中未见有增量 ETL 过程设计方面的研究,但文献[1]中提出"数据仓库可以看成是数据源的物化视图",对应地,我们可以把全量 ETL 过程看成是视图定义,增量 ETL 过程看成是物化视图的增量维护,则增量 ETL 过程完全可以根据全量 ETL 过程推导得到. 物化视图的增量维护方面已经有大量的研究,这些成果有助于实现从已有的全量 ETL 过程自动产生增量 ETL 过程.

目前有不少商业化的 ETL 工具用于实现 ETL 过程[2],但这些工具并未考虑全量 ETL 过程与增量 ETL 过程间的联系,无法提供全量 ETL 过程到增量 ETL 过程的自动转换,需要用户同时手工设计全量和增量 ETL 过程. 而本文主要对增量 ETL 过程的自动化产生方法进行了研究,并给出一种实现算法,根据已有的全量 ETL 过程,可以自动产生增量 ETL 过程,从而降低用户设计增量 ETL 过程的代价.

2 相关工作

在关系数据库中,不同类型的运算有不同的增量维护方法.

在 5 种基本关系运算(选择、投影、联接、并、差)和聚合运算中,选择运算的增量维护是最简单的,一般只在与其他运算组合时才被考虑.

投影运算和并运算的增量维护可以用相同方法实现,这两种运算都可能有多条输入元组对应一条输出元组,为了解决这个问题引入了包语义的概念[3],即允许在输出关系中包含多条相同的元组.另一种解决的方法是借用聚合运算中的计数方法[4]. 利用这两种方法只需要增量信息就可以实现对输出关系的增量维护(这称为自维护[5]).

联接运算的增量维护一般要用到基本关系的全量数据,但是当基本关系增量只包含删除和更新时可以实现自维护[2]. 文献[6]讨论了外联接的增量维护问题.

聚合运算的情况最复杂,其增量维护的实现方式与具体的聚合公式有关. 对于要实现求平均值运算的自维护,需要增加一个用于计数的属性[7]. 对于求最大值和最小值运算,为了实现增量维护,对基本关系的全量查询是不可避免的,但是可以通过一些手段来减少对基本关系的访问次数[8].

文献[5]研究如何实现 SPJ(选择、投影与联接)视图的增量维护,文献[9]研究如何实现 ASPJ(聚合、选择、投影与联接)视图的增量维护. 文献[10]并未考虑特定的关系运算组合,而是考虑了在给定存储空间的情况下,如何选择最优物化视图集,使得视图的查询和增量维护总代价最低.

文献[11]中讨论了如何计算各种基本关系运算(包括差运算)的净增量,但并未涉及增量维护问题.

文献[12]给出了普通视图定义(即视图定义包含选择、投影、联接、并、差和聚合运算)的规范化方法,但文献本身考虑的不是增量维护问题,我们使用这一方法来对全量 ETL 过程进行规范化.

3 基本概念

3.1 增量 ETL 过程自动化产生的基本原理

要自动化产生合理的增量 ETL 过程,首先要将全量 ETL 过程进行规范化,我们使用文献[12]中的规范化方法来实现对全量 ETL 过程的规范化.

因此整个增量 ETL 过程的自动化产生方法主要包括两个步骤:

① 根据文献[12]中的方法对全量 ETL 过程进行规范化(见第 4 节);

② 根据得到的规范化 ETL 过程,对每一个 AUSPJ 片断或 D 片断得到相应的增量维护方法(见第 5,6 节).

3.2 数据源增量的获得

要获得数据源的增量,可以有如下方法:使用时间戳、使用 trigger、使用日志文件和使用全量数据,其中前两种方法需要对数据源进行修改,而后两种方法则不需要.

3.3 自维护物化视图与自维护净增量

在以往的文献中,对于各种运算情况下物化视图的自维护特性[3]做了详细的讨论,用它可以判断是否需要将输入关系物化,但它默认输出关系必然是物化视图.

为了同时判断输入关系和输出关系是否需要物化,我们给出如下定义.

定义 1. 自维护净增量:只需要基本关系的增量信息就可以求得视图或物化视图的净增量,这种净增量称为自维护净增量.

对于 ETL 过程中的某个运算片断,如果输出净增量是自维护净增量,则在实现增量 ETL 过程时,不需要物化输入关系和输出关系. 这有助于降低增量 ETL 过程的实现代价. 显然,当一个视图的净增量是自维护净增量,该视图必然是自维护视图,反之则不成立.

4 规范化 ETL 过程

对于一个包含有 5 种基本关系运算(选择、投影、联接、并、差)和聚合运算的 ETL 过程, 可以规范化为 AUSPJ 片断和 D 片断的组合, 具体方法在文献[12] 中有详细说明, 下面只给出结果.

ETL 过程的规范化结果[12]:

一个 D 片断只包含单个差运算.

一个 AUSPJ 片断包含其他 5 种运算, 并以 α—∪—π—σ—⋈ 顺序排列, 一个 AUSPJ 片断中并不一定需要包含所有这 5 种运算, 但必须满足下面 4 种情况之一:

① 如果有聚合运算, 则聚合运算必须在 AUSPJ 片断的最上面;

② AUSPJ 片断跟在 D 片断的后面;

③ AUSPJ 片断跟在联接运算后面, 而并运算则在 AUSPJ 片断的最上面;

④ AUSPJ 片断是 ETL 过程中最上面的片断, 即在它之上没有任何其他片断.

例 1. ETL 过程的规范化.

以下是一个 ETL 例子, 数据源中包括 4 张表, *Customer*, *VIP*, *Order-A* 和 *Order-B*, 而目标表是 *Total-Consume*, 5 张表分别包括如下属性:

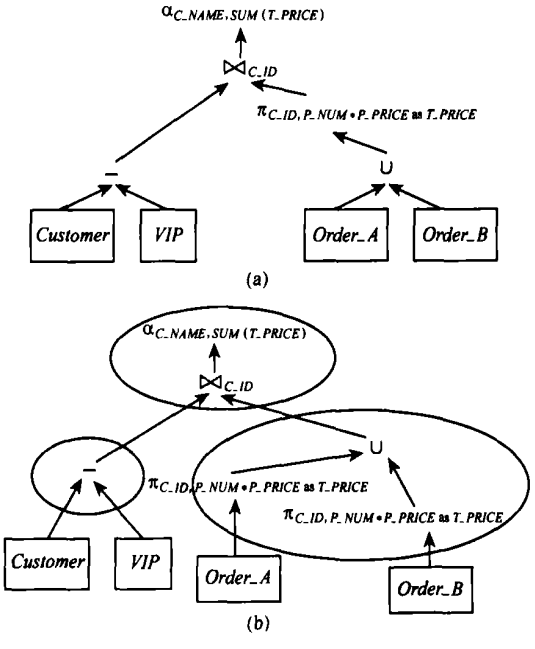

Fig. 1 Standardization of an ETL process. (a) A common ETL process and (b) The canonical form for the ETL process.

图 1 ETL 过程的规范化. (a) 未规范化的 ETL 过程; (b) 规范化后的 ETL 过程.

Customer & *VIP*: *C-ID*, *C-NAME*

Order-A & *Order-B*: *ORDER-ID*, *C-ID*, *PRODUCT-ID*, *P-NUM*, *P-PRICE*

Total-Consume: *C-NAME*, *T-CONSUME*

其未规范化的 ETL 过程定义见图 1(a), 根据文献[12] 中方法规范化的 ETL 过程见图 1(b).

5 实现单个片段的增量维护

要实现整体的增量 ETL 过程, 须先实现单个片断的增量维护. 在此我们假设片断所有的输入关系都不是基本关系.

5.1 AUSPJ 片断的增量维护

对于一个 AUSPJ 片断, 其增量维护的方法可以利用已有的物化视图维护方法[2,4~8] 实现, 在此只给出总结后的结果.

对于一个完整的 AUSPJ 片断, 要实现增量维护, 必须要维护如下的物化视图:

① 联接运算的所有输入关系;

② 投影运算和并运算的输出关系, 并且此关系要增加计数属性或允许保存重复数据, 若输出关系与输入关系有相同主键且计数属性取值必为 1, 则无需物化;

③ 聚合运算的输出关系, 而对于求均值的运算, 输出关系还需要增加计数属性; 对于求最大或最小值的运算, 还要物化输入关系.

根据 AUSPJ 片断的定义, 一个 AUSPJ 片断中可能会缺失某些运算, 则根据缺失的不同运算, 减少相应的物化视图.

5.2 D 片断的增量维护

在数据仓库环境中, 定义物化视图时很少使用差运算, 因此以往的文献并未对差运算物化视图的增量维护方法进行研究. 本小节中我们将对其进行详细讨论.

根据定义, D 片断中只包含单个差运算. 不失一般性地, 考虑差运算 $A-B$. 设其输出关系为 C, 根据文献[11] 中的公式, 关系 C 的净增量计算公式为

$$\triangle C = (\triangle A - (B \cup \triangle B)) \cup ((A \cup \triangle A) \cap \nabla B) - \nabla A, \quad (1)$$

$$\nabla C = (\nabla A - B) \cup (A \cap \triangle B). \quad (2)$$

从上面的公式可以看出, 对于一个 D 片断, 其输入关系必须是物化视图. 我们称其为 BRA(base relations access)方法.

令 $D=A\cap B$, $E=B-D$, 则可得到

$$\triangle C=(\triangle A-(E\cup\triangle B))\cup((D\cup\triangle A)\cap\nabla B)-\nabla A, \quad (3)$$

$$\nabla C=(\nabla A-D)\cup(C\cap\triangle B). \quad (4)$$

因此当关系 C, D, E 是物化视图时, 可以实现对D 片断的增量维护, 我们将其称为 SRA (split relations access)方法.

与 BRA 方法相比, SRA 方法求取关系 C 净增量的效率比较高, 但是为了维护关系 D, E, 增加了求 D, E 增量的步骤. D, E 的增量求取公式分别为

$$\triangle D=(\triangle A-\triangle C)\cup(\nabla C-\nabla A), \quad (5)$$

$$\nabla D=(\triangle C-\triangle A)\cup(\nabla A-\nabla C), \quad (6)$$

$$\triangle E=(\triangle B-\triangle D)\cup(\nabla D-\nabla B), \quad (7)$$

$$\nabla E=(\triangle D-\triangle B)\cup(\nabla B-\nabla D). \quad (8)$$

当增量数据与全量数据相比是一个小量时, 求 D, E 增量的代价可以忽略, 此时 SRA 方法占优. 在数据仓库环境下, 增量一般是小量, 因此大部分情况下应选择 SRA 方法实现增量维护.

一般情况下差运算物化视图是无法实现自维护的, 但是在某些条件下它可以实现自维护, 如下定理所述.

定理 **1**. 设物化视图 C 的定义为 $C=A-B$, 当满足下面 3 个条件之一时, C 是自维护的.

① A 只有删除增量且 B 只有插入增量;

② A 只有删除增量且 B 没有增量;

③ A 没有增量且 B 只有插入增量.

利用式(1)(2), 容易证明定理 1 成立.

另外在实际应用中, 对于差运算 A-B, 有可能 A, B 满足 $B\subset A$. 针对这种情况有如下定理.

定理 **2**. 设物化视图 C 的定义为 $C=A-B$, 当 A, B 满足 $B\subset A$, C 的净增量满足自维护净增量定义.

证明. 根据条件 $B\subset A$, 式(3)(4)可化为

$$\triangle C=(\triangle A-\triangle B)\cup(\nabla B-\nabla A), \quad (9)$$

$$\nabla C=(\nabla A-\nabla B)\cup(\triangle B-\triangle A). \quad (10)$$

从式(9)(10)看出, C 的净增量可以通过 A, B 的净增量计算得到, 满足对自维护净增量的定义.

证毕.

综上所述, 一个 D 片断要实现增量维护, 一般情况下根据增量大小选择 SRA 方法或 BRA 方法来实现, 若输入数据满足定理 1 或定理 2, 则前者只需物化输出关系而后者无需物化.

6 自动产生增量 ETL 过程

增量 ETL 过程的实现依赖于全量 ETL 过程执行时建立的辅助视图, 增加辅助视图可以降低求增量的代价, 但辅助视图的维护代价会增量, 同时也会增加存储空间.

给定全量ETL 过程为 G 和存储空间 S, 并选择一组辅助视图 M, 则增量 ETL 过程的执行代价为[10]

$$\tau(G, M)=\sum_{i=1}^{k}D(D_i, M)+\sum_{j=1}^{m}U(V_j, M)+C, \quad (11)$$

同时在存储空间 S 上应该满足

$$\sum_{j=1}^{m}V_j\leqslant S. \quad (12)$$

原有的辅助视图选择问题就转化为在满足式(12)的前提下, 选择辅助视图集 M 使得式(11)的值最小, 这是一个 NP 难的问题[10].

但是由于数据仓库本身是海量存储, 因此不需考虑式(12)的限制.

同时我们将增量 ETL 过程分为两个步骤: 第 1 步中求出目标关系的净增量后, 仅更新主视图; 第 1 步则更新所有辅助视图, 我们将之称为 TSU (two step update)方法.

由于 TSU 方法中仅第 1 步执行时需要锁住数据仓库, 因此只需考虑降低 TSU 方法中第 1 步的执行代价, 则式(11)简化为

$$\tau(G, M)=\sum_{i=1}^{k}D(D_i, M)+C. \quad (13)$$

因此辅助视图选择问题简化为选择辅助视图集 M 使得式(13)的值最小.

与文献[12]中对视图定义的规范化相似, 一个规范化的 ETL 过程也用一棵查询树来描述, 称为 ETL 过程树. 下面给出根据 ETL 过程树自动产生增量 ETL 过程的算法 MCCI (minimal cost of calculating increment).

算法 **1**. MCCI 算法.

CreateIncrementalETL (t_0, V_0)

输入: 规范化的全量 ETL 过程树 t_0, 一组视图定义的集合 V_0(每个视图定义对应 t_0 中的一个片断);

输出: 使用物化视图重新定义的一组视图定义的集合 V, 以及用于实现增量 ETL 过程的 SQL 语句的集合 S.

方法:

```
begin
  V←∅, S←∅;
  do{从 t0 中取出位于最顶层且未被处理的片断si;
    从 V0 中取出与 si 相应的视图定义 vi;
  if(vi 是AUSPJ 片断)
    对 vi 中的每个关系运算, 在 vi 中修改相应的视图定义, 并增加物化视图①;
  else
    对 vi 中的差运算, 在 vi 中修改相应的视图定义, 并增加物化视图②;
    产生对 vi 进行增量维护的SQL 语句集合 Si;
    V← V∪ vi; S←S∪ Si;}
  until(t0 中不再有未被处理的片断);
  return(V, S);
end
```

这一算法在产生增量 ETL 过程时, 只需对 ETL 过程树扫描一次, 设片段节点数为 N, 则其计算复杂度为 $O(N)$. 根据式(13), 如果保证每个片断的增量计算代价最小, 则 MCCI 算法输出的增量 ETL 过程在执行 TSU 方法第 1 步时, 其执行代价最小. 但实际上增量维护代价是否最优取决于增量的具体数据[2], 因此我们只能保证算法产生的增量 ETL 过程在普遍情况下具有较好执行效率.

7 实验与分析

我们在自己的 ETL 工具上实现了产生自动增量的 MCCI 算法, 并用例 1 来比较增量与全量的执行效率, 因为在这个例子中除了没有选择运算外, 包含了其他 4 种基本运算以及聚合运算, 较有代表性. 测试用数据为 *Customer*, *Order_A*, *Order_B*, 都为 100000 条记录, *VIP* 是 10000 条记录.

在实验中, 我们除了测试增量与全量执行时间外, 也测试了对增量 ETL 过程进行优化后的执行时间, 例子中考虑了如下优化: 由于 *Order_A* 和 *Order_B* 主键(*ORDER_ID*, *PRODUCT_ID*)的值不重复, 因此在投影运算中保留了该主键, 则并运算增量维护无需使用计数方法; *VIP* 是 *Customer* 的子集, 根据定理 2 用式(9)(10)实现差运算的增量维护. 实验结果见图 2 所示:

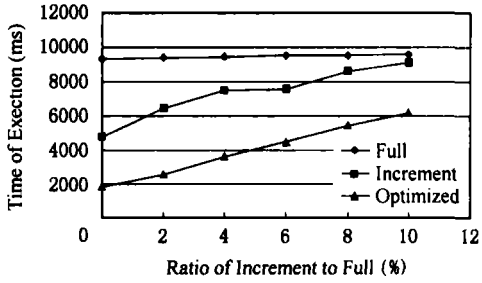

Fig. 2 Time of executing full or incremental ETL processes.

图 2 全量与增量执行时间的比较

从实验结果来看, 在增量远小于全量的情况下, 我们的 MCCI 算法得到的增量 ETL 过程确实比全量的执行时间要少, 而优化后的增量 ETL 过程执行时间更进一步减少.

我们同时也在上海期货交易所用于统计信息共享的实际数据仓库项目上进行了测试. 该项目需要每天执行一次 ETL, 而每次增加的数据大约是 20MB, 不到全量数据的百分之一, 从上面的实验可以知道, 在这种情况下, 增量 ETL 有很好的执行效率. 我们在该项目上选择了 6 个比较复杂的 ETL 过程, 分别比较了自动产生的增量 ETL 过程和原有手工设计增量 ETL 过程执行时所耗的时间, 结果见表 1 所示(单位 s). 从表 1 中可以看出两者的执行时间大致相同, 惟一有较大差距的是第 5 个增量 ETL 过程, 经比较后发现由于该 ETL 过程中一个联接运算可以保证增量数据不会与以前的数据相关且无删除, 因此在手工设计的增量 ETL 过程中增量运算公式简化为 $\triangle R_1 \bowtie \triangle R_2$, 这提高了执行效率.

Table 1 Time of Executing Manual or Automatically-Designed Incremental ETL Processes

表 1 手工和自动设计增量 ETL 过程执行时间 s

ETL Process	Manual	Auto.
ETL1	329.7	331.3
ETL2	153.1	153.4
ETL3	87.5	86
ETL4	132.8	134.4
ETL5	348.5	501.5
ETL6	490.6	493.7

① AUSPJ 片断中物化视图的选择方法见第 5.1 节.

② D 片断的增量维护选择方法见第 5.2 节.

实验结果说明,根据全量 ETL 过程自动产生的增量 ETL 过程在大部分情况下完全可以代替手工设计的增量 ETL 过程,只是在某些情况下,考虑特定数据分布情况时手工设计的增量 ETL 过程会有更好表现,但完全可以在自动产生的增量 ETL 过程上进行修改后得到,而设计代价显然比完全用手工设计要低.

8 总 结

本文主要讨论了如何根据已有的全量 ETL 过程自动产生增量 ETL 过程. 由于 ETL 过程与数据库中的视图定义相似,我们借鉴了已有物化视图增量维护的方法,来帮助实现增量 ETL 过程的自动产生.

我们首先用文献[12]中的方法将全量 ETL 过程进行规范化. 之后利用已有的物化视图增量维护方法,给出了 AUSPJ 片断的增量维护方法;同时详细讨论了 D 片断的增量维护问题.

最后我们讨论了如何自动产生增量 ETL. 为了实现增量 ETL 过程,必须选择必要的辅助视图集,而在给定存储空间下求得最优辅助视图集是 NP 难问题. 考虑到数据仓库的海量存储特性,我们忽略存储空间的限制,同时提出了 TSU 方法减少了在执行时间上的约束条件,最终简化了辅助视图的选择公式,并根据这一公式给出了自动产生增量 ETL 过程的 MCCI 算法.

在一个 ETL 工具中,我们实现了 MCCI 算法并分别在随机数据和实际数据上进行了实验,取得了预期的结果. 相较于以往增量 ETL 过程需要手工设计,我们的工作使得这一步骤可以自动完成,并且输出的增量 ETL 过程具有很好的通用性. 但是不可否认,在自动产生的增量 ETL 过程的基础上,根据数据分布特性进行优化,将获得更高的效率.

参 考 文 献

1 Divyakant Agrawal, Amr El Abbadi, *et al*. Efficient view maintenance at data warehouses. ACM SIGMOD Conf. Management of Data, Tucson, USA, 1997

2 Ashish Gupta, Inderpal Singh Mumick. Maintenance of materialized views: Problems, techniques, and applications. IEEE Data Eng. Bull., 1995, 18(2): 3~18

3 Amr El Abbadi, Michael L. Brodie, *et al*. Performance issues in incremental warehouse maintenance. The 26th Int'l. Conf. on Very Large Data Bases, Cairo, Egypt, 2000

4 I. S. Mumick, O. Shmueli. Finiteness properties of database queries. The 4th Australian Database Conf., Brisbane, Australia, 1993

5 José A. Blakeley, Per-Åke Larson, Frank Wm. Tompa. Efficiently updating materialized views. ACM SIGMOD Conf. Management of Data, Washington, USA, 1986

6 Ashish Gupta, H. V. Jagadish, Inderpal Singh Mumick. Data integration using self-maintainable views. The 5th Int'l. Conf. Extending Database Technology, Avignon, France, 1996

7 T. Palpanas, R. Sidle, R. Cochrane, *et al*. Incremental maintenance for non-distributive aggregate functions. The 28th Int'l Conf. on Very Large Data Bases, Hong Kong, 2002

8 Ke Yi, Hai Yu, Jun Yang, *et al*. Efficient maintenance of materialized top-k views. The 19th Int'l. Conf. Data Engineering, Bangalore, India, 2003

9 Yingwei Cui, Jennifer Widom. Storing auxiliary data for efficient maintenance and lineage tracing of complex views. The 2nd Int'l Workshop on Design and Management of Data Warehouses, Stockholm, Sweden, 2000

10 Himanshu Gupta. Selection of views to materialize in a data warehouse. The 6th Int'l Conf. Database Theory, Delphi, Greece, 1997

11 Timothy Griffin, Leonid Libkin, Howard Trickey. An improved algorithm for the incremental recomputation of active relational expressions. IEEE Trans. Knowl. Data Eng., 1997, 9(3): 508~511

12 Y. Cui, J. Widom, J. L. Wiener. Tracing the lineage of view data in a warehousing environment. ACM Trans. Database Systems, 2000, 25(2): 179~227

Zhang Xufeng, born in 1974. Ph. D. candidate of Fudan University. His current research interests are data warehouse and data stream.

张旭峰, 1974 年生, 博士研究生, 主要研究方向为数据仓库、数据流.

Sun Weiwei, born in 1973. Associate professor of Fudan University. His current research interests are data warehouse, portable database, etc.

孙未未, 1973 年生, 副教授, 主要研究方向为数据仓库、移动数据库.

Wang Wei, born in 1970. Received his Ph. D. degree in 1998 and now is professor of Fudan University. His current research interests are data warehouse, data mining, and management of complex data.

汪卫, 1970 年生, 博士, 教授, 主要研究方向为数据仓库与数据挖掘、复杂数据管理技术.

Feng Yahui, born in 1983. She now is a senior student of Fudan University. Her current research interests include ad hoc network.

冯雅慧, 1983 年生, 大学本科四年级, 主要研究方向为 ad hoc 网络

Shi Baile, born in 1936. Professor and Ph. D. supervisor of Fudan University. His main research interests are database, knowledge base, etc.

施伯乐, 1936 年生, 教授, 博士生导师, 主要研究方向为数据库、知识库

Research Background

ETL processes are used for collecting data from data sources to data warehouse, and they are very important for founding data warehouse. ETL processes can be separated into two portions: full ETL processes and increment ETL processes. Full ETL processes are used to initialize data warehouse and increment ETL processes are used to maintain data warehouse. Normally increment ETL processes are more complex than full ETL processes, so designer have to spend more outlay to design increment ETL processes manually. But in fact an increment ETL process can be generated automatically from a full ETL process which has been designed. Now there are a large variety of ETL tools available on the market, but they can't support generating increment ETL processes automatically. In this paper, using the existing methods of incremental maintenance of materialized views for reference, an approach to generate an incremental ETL process automatically from a full ETL process is put forward. Our work is supported by the National High Technology Research and Development Program of China (2002AA4Z3430).

2006 年全国高性能计算学术会议(HPC China 2006)会议征文通知

会议由中国计算机学会高性能计算专业委员会主办, 中国科学院网络信息中心承办. 会议将于 2006 年 10 月 27~29 日在北京友谊宾馆召开.

论文涉及的领域如下: 高性能计算机体系结构、高性能计算机软件、并行算法、高性能计算机应用、网格技术及应用.

会议网址: http://www.sccas.cn/hpcchina2006

投稿地址: hpcchina2006@sccas.cn 或 chi@sccas.cn

论文截至日期: 2006 年 8 月 30 日

论文通知日期: 2006 年 9 月 30 日

论文交印日期: 2006 年 10 月 20 日