Oracle Discoverer's™ Fan Trap Resolution - Correct Results Every Time

An Oracle Technical White Paper

April 2000

ORACLE
Enabling the Information Age™

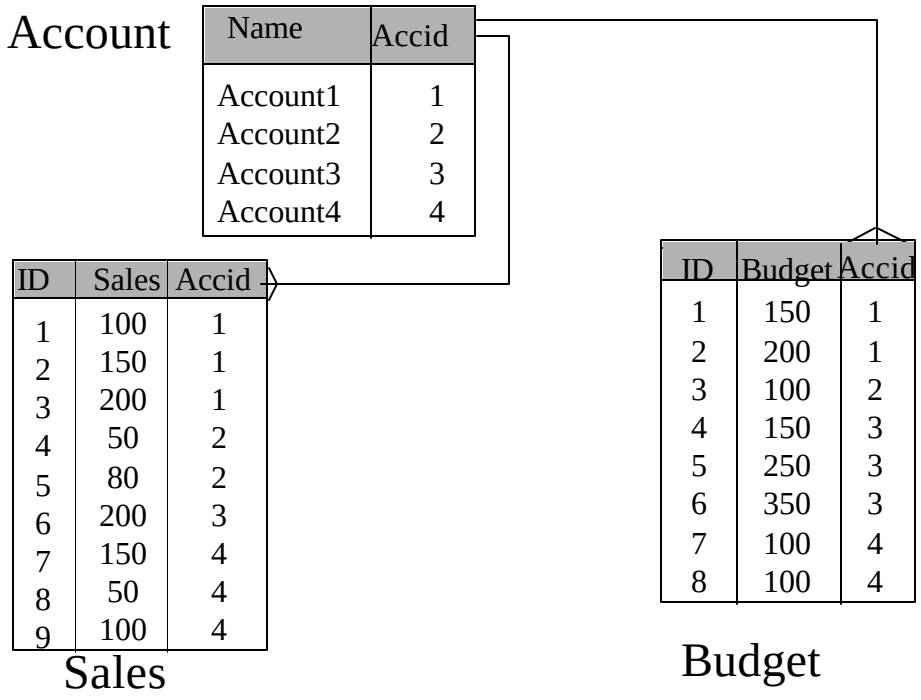
Oracle Discoverer's™ Fan Trap Resolution - Correct Results Every Time

Introduction

Administrators have been struggling with business intelligence solutions to guarantee the correct ad-hoc query results. Many of these solutions have required very complicated or maintenance intensive workarounds, until now. The latest Discoverer release includes a scaleable and, most importantly, a maintenance free and robust query rewrite algorithm that resolves the "fan trap." This elegant solution is presented to you in this white paper.

What is a Fan Trap?

A "fan trap" is a SQL query that generates unexpected results. The most common manifestation occurs when a master table is joined to two detail tables separately and results in incorrectly aggregated measures. To the end user the definition is meaningless, yet the consequences can be catastrophic. The following example illustrates this point.



Simple data model example: Multiple detail tables linked to single master table

Question: "What is the total sales and total budget by account?"

In this scenario an account can have both sales and budget information. The aggregates (Sum) from the 2 detail tables (SALES and BUDGET) are coming off the same master table (ACCOUNT). The result we expect is:

Name	Sales Sum	Budget Sum
Account1	$450	$350
Account2	$130	$100
Account3	$200	$750
Account4	$300	$200

Example 1: Expected results

The result using straightforward SQL is:

Name	Sales Sum	Budget Sum
Account1	$900	$1050
Account2	$130	$ 200
Account3	$600	$ 750
Account4	$600	$ 600

Example 2: Unexpected results

While the results are relationally correct, the results are unexpected.

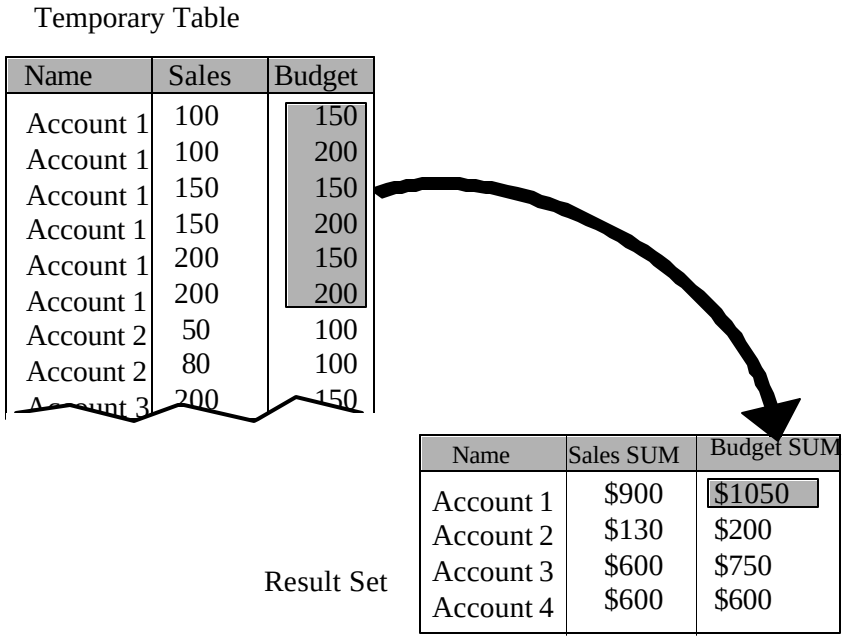
How Unexpected Results are Generated

Example 2 is based on a single query in which the tables are joined together first in a temporary table and then the aggregation is done. This causes the aggregates to be summed multiple times. The straightforward SQL used in example 2 is:

```
SELECT
 Account.name,
 SUM(sales),
 SUM(budget)
FROM
 Account,
 Sales,
 Budget
WHERE
 Account.id=Sales.accid
AND
 Account.id = Budget.accid
GROUP BY
Account.name
```

Straight forward SQL resulting in unexpected results

The result is unexpected because each sale and each budget has been summed several times in a temporary table. The temporary table and result set are shown below.



Straight forward SQL Illustration

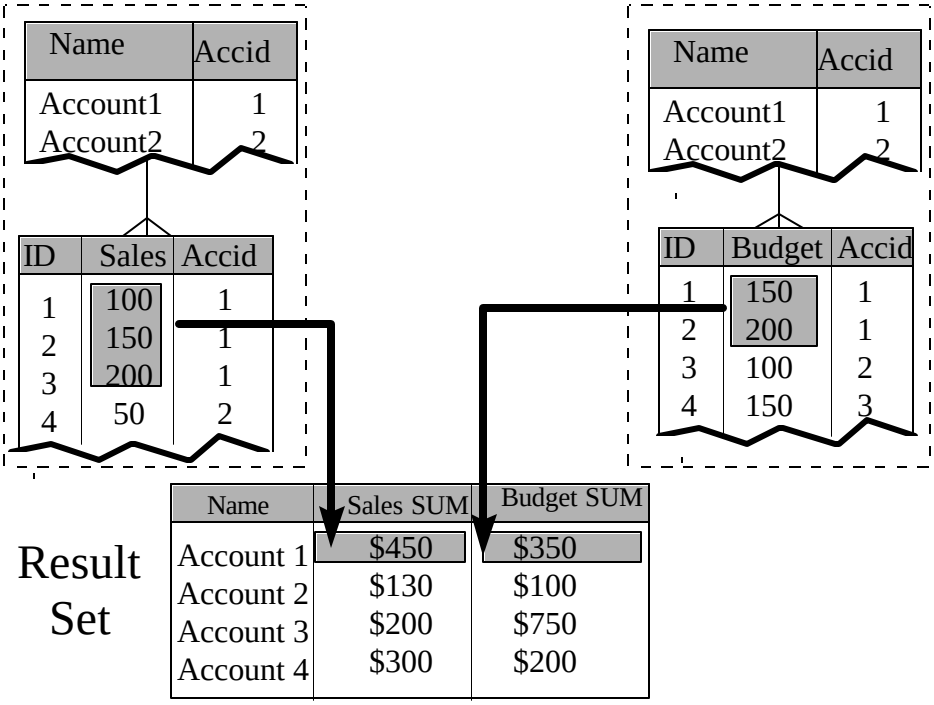
Guaranteeing the correct results

Every query generated by Discoverer is interrogated. Discoverer will detect a fan trap and rewrite the query to ensure that the aggregation is done at the correct level. Discoverer rewrites the query using inline views, one for each master-detail aggregation, and then combines the results in the outer query.

```
SELECT inACC as NAME ,SUM(inSalesSum) as SALES_SUM , SUM(inBudgetSum) as BUDGET_SUM
 FROM
   ( SELECT masterID AS OutMasterIDSales, SUM(SalesDetailSales) AS inSalesSum
    FROM
         ( SELECT ID AS masterID, NAME AS masterName FROM ACCOUNT ) INLineAccount,
         ( SELECT ID AS SalesDetailId, ACCID AS SalesDetailAccID, SALES AS SalesDetailSales FROM SALES )INLineSales
   WHERE (masterID = SalesDetailAccID(+))
   GROUP BY masterID ) inner1,
    ( SELECT masterID AS OutMasterIDBudget, SUM(BudgetDetailBudget) AS inBudgetSum, masterName AS inACC
   FROM
         ( SELECT ID AS masterID, NAME AS masterName FROM ACCOUNT ) INLineAccount,
         ( SELECT ID AS BudgetDetailId, ACCID AS BudgetDetailAccID, BUDGET AS BudgetDetailBudget FROM BUDGET ) INLineBudget
   WHERE (masterID = BudgetDetailAccID(+))
   GROUP BY masterName ,masterID ) inner2
WHERE ( (OutMasterIDBudget = OutMasterIDSales))
GROUP BY inACC;
```

Inline View SQL resulting in correct results

The result is correct because each sale and each budget has been summed individually (one for each master-detail aggregation) and then combined with a join based on the master key(s). The inline views and result set are shown below.



Inline views illustration

Robust Server side implementation

Discoverer computes the result set entirely within the database engine. No joining is required on either the client or middle tier server. Not only does it leverage the database engine to completely derive the result set but Discoverer's fan trap solution extends to other fan trap scenarios.

The second type of fan trap involves aggregation at different levels within a hierarchy that spans multiple tables. For example, Customer-Order-Order Detail. A customer's order can contain many orderlines - but the order table may hold its own totals and a query may try to aggregate both from the order table and the orderline table. Discoverer rewrites the query similar to the example above by aggregating first and then joining the results together through the use of inline views.

Zero Administration and Zero training required

Discoverer's solution requires no data model changes or semantic layer modifications to implement. Once installed, user's queries will *automatically* be rewritten. Even saved queries containing a fan trap will automatically be rewritten.

Scaleable Solution

Discoverer is the only product on the market that detects a fan trap, rewrites the query and executes the entire query on the server. The intelligence of recognizing a fan trap situation resided in the Discoverer semantic layer, the End User Layer (EUL). The Discoverer EUL resides in the database and is used for both client/server and Web implementation.

Availability

The fan trap solution is available on Windows with release 3.1.41 of Discoverer. The fan trap solution will be available on the Web shortly for both the Discoverer User Edition and Discoverer Viewer.

Summary

Business Users need confidence in their data and reporting results. To do so however, they should not have to be experts in in-line views, cartesian products, temporary tables, or other nuances of SQL. Discoverer's elegant solution delivers the correct results business users expect, within the same easy to use framework they have come to expect.

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
+1.650.506.7000
Fax +1.650.506.7200
http://www.oracle.com/