Transbase®

WHITE PAPER

Handling of Duplicates in Transbase®

1. Introduction

In the relational data model a table is defined as a set of n-ary tuples (or rows). Each tuple is said to consist of n fields (or columns), each field having identical data type. In the following we denote a tuples as t = (f1, f2, …, fn).

Differences exist on how duplicates tuples are handeled. Two tuples t1=(f11, f12, …, f1n) and t2=(f21, f22, …, f2n) are said to be duplicates iff f11=f21 & f12=f22 & … & f1n=f2n.

Some database systems (like Oracle) allow duplicate tuples to exist within the same table, some other systems (like Transbase) guarantee that duplicates are eliminated, i.e. tables are – mathematically – true sets. The SQL standard permits both implementations. C. Date who laid the theoretical fundament for the relational data model in his book "A Guide to the SQL Standard", however, truly favors that no duplicates should be permitted ever:

> "First, note that in the relational model no table is ever permitted to contain any duplicate rows. That is, the body of every table is guaranteed to be a true mathematical set of rows (sets in mathematics do not contain duplicate elements). In SQL, however, this discipline is (very unfortunately, in this writer's opinion) not enforced, and tables are indeed permitted to contain duplicate rows."

The following paper describes in detail how duplicates are handled in Transbase®.

2. Indexed Tables

The standard table in Transbase® is an indexed table (implemented as B* tree). For indexed tables a primary key is mandatory. If no primary key is specified, Transbase® assumes the whole tuple to be key. Tuples in an indexed table are stored in order of their primary key values.

Contact

Transaction Software GmbH
Willy-Brandt-Allee 2
81829 Muenchen, Germany

Tel.: +49 89 / 627 09 - 0
Fax: +49 89 / 627 09 - 11

info@transaction.de
www.transaction.de
www.transbase.de

The tuples of an indexed table consist of key fields and non-key fields, denoted in the following as t = (k1, …, ks, f1, …, ft) with s+t=n, where the tuple consists of s 1 key fields and t 0 non-key fields.

Following the definition above two tuples t1 and t2 are said to be “full duplicates” iff all key fields are identical and all non-key fields are identical. They are said to be “key duplicates” if all key fields are identical and some non-key fields are not identical.

From the semantics of a primary key it is obviously prohibited that key duplicates can exist within one table. Full duplicates are considered to be undistinguishable and therefore are ignored.

Consequently, the INSERT of a key duplicate Transbase® throws an error, while the INSERT of a full duplicate is ignored (but reported as such). In particular, Transbase® reports for INSERT statement two counters, namely the number of tuples provided for INSERT and the number of tuples actually INSERTed. The latter count may be less than the former count indicating that some tuples (namely the difference between the two counters) have been ignored.

As an example, assume a table to consist only of a INTEGER valued primary key and a CHARACTER valued non-key field. After INSERTing the tuples:

(1, 'Transbase')

(2, 'Oracle')

(3, 'DB/2')

the table consists of exactly these three tuples.

If these tuples are INSERTed once more, the table still consists of these three tuples, i.e. the INSERT reports the counters 3 and 0, meaning that 3 tuples were provided but 0 tuples were actually INSERTed (hence 3 tuples ignored). Note that no error is thrown (which is in many situations very practical).

If tuples

(1, 'Transbase')

(2, 'MySQL')

(4, 'Postgres')

are INSERTed, an error is reported as the tuple (2, 'MySQL') is a key duplicate and would violate the primary key constraint. As the statement is erroneous, no tuples are INSERTed at all, because the semantics of a SQL statement requires that all or no changes are made. In particular, the tuples (1, 'Transbase') and (4, 'Postgres') could have been INSERTed without constraint violation.

If the next INSERT statement would provide the tuples

(1, 'Transbase')

(4, 'Postgres')

the statement would be accepted and reported as 1 tuple INSERTed and 1 tuple ignored.

3. Flat Tables

From version 6.7 Transbase® supports a further table type, so-called flat tables. These tables do not require a primary key be specified. Tuples are not stored in a particular order.

Flat tables show a different behavior on INSERT, as duplicates are simply INSERTed multiply, i.e. flat tables show the property of a so-called "multi-set".

Following the example above, with the same table definition (as flat table) and the same INSERT statements, the table would consist of the following tuples (in this order):

(1, 'Transbase')

(2, 'Oracle')

(3, 'DB/2')

(1, 'Transbase')

(2, 'Oracle')

(3, 'DB/2')

(1, 'Transbase')

(2, 'MySQL')

(4, 'Postgres')

(1, 'Transbase')

(4, 'Postgres')

Eliminating full duplicates in this table would be expensive afterwards because typically all tuples have to be sorted to identify those duplicates.

Contact

Transaction Software GmbH
Willy-Brandt-Allee 2
81829 Muenchen, Germany

Tel.: +49 89 / 627 09 - 0
Fax: +49 89 / 627 09 - 11

info@transaction.de
www.transaction.de
www.transbase.de