Rdb Journal
technical corner

Automatic Columns

A feature of Oracle Rdb

By Ian Smith
Oracle Rdb Relational Technology Group
Oracle Corporation

The examples in this article use SQL language from Oracle Rdb V7.1 and later versions.

COMPUTED BY Columns

Rdb has supported a special read-only COMPUTED BY column since its first release. This special column computes a value when selected or when referenced in a WHERE clause.

```
SQL> create table PERSON (
cont>     employee_id         integer,
cont>     employee_id_disp
cont>     computed by
cont>     SUBSTRING (CAST(employee_id + 100000
cont>          as VARCHAR (6)) from 2)
cont>     …);
```

In this example the column EMPLOYEE_ID_DISP is used to display the employee id with leading zeros. The expression can be simple such as CURRENT_USER, or very complex using the full power of the SQL subquery syntax.

Since values are computed at select time there is no space required within the row, and the expression need not be evaluated during INSERT or UPDATE of the referenced columns. Neither the INSERT nor the UPDATE statement may assign values to this type of column.

A COMPUTED BY column can reference other computed columns in the same table. Rdb expands each expression when referenced so that the correct value is returned. If many COMPUTED BY columns use subquery syntax in the computed by expression, then Oracle recommends that these calculations be replaced with calls to SQL functions to limit the tables contexts used by these computed columns.

AUTOMATIC Columns

Oracle Rdb V7.1 now includes a new type of read-only column called AUTOMATIC columns. Automatic columns are closely related to COMPUTED BY columns; however, they are stored in the database and are only evaluated at INSERT and UPDATE time.

The database designer can define an AUTOMATIC column to be computed and stored during INSERT, UPDATE or during both these statements. These columns can also be used as part of an index key, and referenced by constraints.

```
SQL> create table PERSON
cont>     person_key
cont>     automatic insert as GET_NEW_ID () primary key,
cont>     …);
SQL> create unique index PERSON_INDEX on PERSON(person_key);
```

This partial example shows the use a SQL function to calculate a unique value for the primary key field. The column PERSON_KEY will inherit its data type from the value expression. Use the CAST operator to choose the data type for yourself.

As read-only columns they may not be targets for an UPDATE or INSERT statement and are therefore ideal for calculating auditing information that you do not want modified by unprivileged users.

The SQL syntax allow for three types of AUTOMATIC columns:

1. AUTOMATIC INSERT AS
   that is calculated only during an INSERT statement.
2. AUTOMATIC UPDATE AS
   that is calculated only during an UPDATE statement.
3. AUTOMATIC AS
   that is calculated at both UPDATE and INSERT time.

Consider this simple example:

- When a row is inserted track who executed the statement (CURRENT_USER) and when this action occurred (CURRENT_TIMESTAMP).

- When a row is inserted or updated record the change timestamp.

```
SQL> create table PRODUCTS
cont>     product_id     integer primary key,
cont>     entered_by     automatic insert as current_user,
cont>     change_dt      automatic as current_timestamp,
cont>     …);
```

Here the ENTERED_BY column will contain the user name of the user who inserted the row, and the current timestamp will be written to CHANGE_DT. Rdb will revise the CHANGE_DT column automatically during subsequent UPDATE statements.

Note: in prior versions read-only columns were included in the default column list for INSERT, even though they could not be modified. Starting with Rdb V7.1 read-only columns (COMPUTED BY and AUTOMATIC) are excluded from the default column list for INSERT. This simplifies programming by allowing new COMPUTED BY and AUTOMATIC columns to be added without requiring changes to existing code.

Frequently Asked Questions on AUTOMATIC columns

What if the updated or inserted data is wrong, how can I fix it? A privileged user who has the database privilege DBADM can use SET FLAGS 'AUTO_OVERRIDE' statement to disable the AUTOMATIC column for new queries in this session. They are then treated as read-write columns and can be updated.

This is also a common requirement when reloading data during database restructuring. Use the RDMS$SET_FLAGS logical to define AUTO_OVERRIDE prior to running RMU/LOAD or similar load program.

The SQL IMPORT statement enables this flag automatically when importing tables with AUTOMATIC columns so that previously recorded values are not replaced when loading the new database.

Can I modify an existing column to an AUTOMATIC column? Currently the ALTER TABLE … ALTER COLUMN statement does not support this type of conversion. This may be possible in a future release of Rdb.

Can I use sequences in an automatic column? Yes, this is an ideal use of AUTOMATIC columns. Use the expression seq.NEXTVAL as the expression. The default data type will be BIGINT but you can add a CAST expression to use a numeric type acceptable to your application.

The following example shows the use of a sequence to provide PRIMARY KEY values to the department table.

```
SQL> set dialect 'sql99';
SQL> set display no row counter;
SQL> create sequence dept_id;
SQL> create table departments
cont> (dept_id
cont>     automatic insert as dept_id.nextval
cont>     primary key,
cont>  dept_name char(20));
SQL> insert into departments values ('Admin');
SQL> insert into departments values ('Engineering');
SQL> insert into departments values ('Accounting');
SQL> insert into departments values ('Marketing');
SQL> select * from departments;
     DEPT_ID   DEPT_NAME
          1    Admin
          2    Engineering
          3    Accounting
          4    Marketing
4 rows selected
```

In Rdb V7.1.0.2 the IDENTITY clause implicitly creates an AUTOMATIC column to deliver the results from the tables private sequence.

Can I provide a DEFAULT for an AUTOMATIC column? A DEFAULT clause is only required when AUTOMATIC UPDATE is used. In this case during INSERT the default value will be used, or possibly NULL if none were provided.

What type of constraints can be defined for AUTOMATIC columns? In V7.1.0.2 all types of constraints are permitted. Prior to this release only NOT NULL, and CHECK constraints were permitted.

Oracle Rdb
Automatic Columns
May 2002

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle Corporation provides the software that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2002 Oracle Corporation
All rights reserved.