Nuevamente necesito tu cerebro.

Desde hace tiempo que me molesta cómo HotRod define las columnas autogeneradas. O sea, funciona bien para el 99% de los casos y hasta la fecha ha sido suficiente. Sin embargo, creo que debería cubrir todos los casos.

Después de revisar como 12 bases de datos distintas operan llegué a la conclusión que hay tres tipos de columnas autogeneradas:

* Identity Always,
* Identity By Default, y
* DEFAULT Constraint.

A estas tres variantes se agrega las columns obtenidas desde una secuencia, que sería el cuarto seudo tipo. "Seudo" porque la base de datos no dispone de metadata al respecto. Todo esto es para que cuando se inserta una fila en una tabla, no es necesario indicar los valores para las columnas autogeneradas.

**EJEMPLO**

Por ejemplo, en Oracle 12c R1:

**create** sequence id\_externo\_seq;

**create** **table** persona (

id\_persona number(9) generated **by** **default** **as** identity **not** **null**, -- identity

id\_sucursal number(6) **default** 106 **not** **null**, -- DEFAULT constraint

id\_externo number(9) **not** **null**, -- generado por sequencia

nombre varchar2(40) **not** **null**

);

En este caso tan sólo la propiedad "nombre" es realmente requerida. Un insert mínimo podría ser:

**insert** **into** persona (id\_externo, nombre)

**values** (id\_externo\_seq.nextval, 'Pedro');

En este caso la columna "id\_persona" tomaría el siguiente valor de la secuencia, y el "id\_sucursal" sería 106 (valor por defecto).

Por otro lado, un insert con todos los valores podría ser:

**insert** **into** persona (id\_persona, id\_sucursal, id\_externo, nombre)

**values** (1001, 500, id\_externo\_seq.nextval, 'Juan');

Hoy en día HotRod reconoce las columnas IDENTITY automáticamente y también las columnas con DEFAULT constraints, así que teóricamente toda la metadata está disponible para implementar esto.

**RECUPERACIÓN DE VALORES AUTOGENERADOS**

Durante un insert los valores autogenerados son insertados en la tabla, pero no necesariamente son informados de vuelta a la aplicación cliente. Al estudiar estas bases de datos, encontré tres casos distintos:

**Grupo 1 - Integrado**

En estas bases de datos un INSERT puede recuperar los valores autogenerados si así lo indica. Este grupo incluye:

* Oracle
* DB2
* PostgreSQL
* SQL Server

Una única llamada SQL inserta la fila y recupera los valores generados por la base de datos para las secuencias, identity y columnas DEFAULT.

**Grupo 2 - Semi Integrado**

En estas bases de datos un INSERT sólo recupera los valores identity. Este grupo incluye:

* MySQL
* MariaDB
* SAP ASE
* HyperSQL
* H2
* Apache Derby
* Teradata

Dado que sólo la(s) columnas identity pueden ser recuperadas como parte del insert, se necesita llamadas SQL adicionales para obtener los valores de las sequencias y columnas DEFAULT.

**Grupo 3 - Rudimentatio**

En estas bases de datos un INSERT no recupera valor autogenerado alguno. Este grupo incluye:

* SQLite

Se necesita múltiples llamadas SQL adicionales para recuperar los valores de las identity, secuencias y columnas DEFAULT.

Tal vez HotRod debería hacer estos casos transparentes. Es decir, para mantener una API única, HotRod podría ejecutar tras bambalinas las operaciones adicionales en caso de ser necesario.

**VARIANTES DE AUTOGENERACIÓN**

Ahora bien, creo que existen los siguientes casos para las tablas:

**1. Tabla sin columnas autogeneradas**

Por ejemplo:

**create** **table** bitacora (

fecha **date** **not** **null**,

comentario varchar2(1000) **not** **null**

);

En este caso se configura en HotRod como:

<table name=*"bitacora"* />

En donde las dos columnas (fecha y comentario) siempre deben ser especificadas durante un insert.

**2. Generación por secuencia - Caso general**

Por ejemplo:

**create** sequence sec\_folio;

**create** **table** vehiculo (

vin varchar2(20) **not** **null** **primary** **key**,

folio number(12) **not** **null** -- no es PK

);

**create** sequence sec\_serial\_dia;

**create** **table** factura (

fecha **date** **not** **null**,

serial\_dia number(12) **not** **null**, -- es parte de la PK compuesta

monto number(10, 2) **not** **null**,

**primary** **key** (fecha, serial\_dia)

);

Se configura en HotRod como:

<table name=*"vehiculo"*>

<auto-generated-column name=*"folio"* sequence=*"sec\_folio"* />

</table>

<table name=*"factura"*>

<auto-generated-column name=*"serial\_dia"* sequence=*"sec\_serial\_dia"* />

</table>

**3. Generación por secuencia - Caso simplificado**

Aunque el caso anterior resuelve todas las combinaciones posibles, en el 99% de los casos las tablas que usan secuencias sí tienen PK, esta PK tiene una sola columna, que es generada por la secuencia. Si estas condiciones se cumplen entonces se puede usar la variante simplificada en HotRod.

**create** sequence seq\_sucursal;

**create** **table** sucursal (

id number(12) **not** **null** **primary** **key**,

nombre varchar2(100) **not** **null**

);

Se configura en HotRod como:

<table name=*"sucursal"* sequence=*"seq\_sucursal"* />

**4. Generación por identity**

Por ejemplo:

**create** **table** cliente (

id number(12) generated always **as** identity **not** **null** **primary** **key**,

nombre varchar2(60) **not** **null**,

identificador\_bancario number(12) generated **by** **default** **as** identity **not** **null**

);

Dado que HotRod identifica las columnas identity, se configura en HotRod como:

<table name=*"cliente"* />

Nótese que hay dos columnas identity (id, identificador\_bancario) y que son de distinto tipo. HotRod recupera la metadata e identifica ambos casos automáticamente.

**5. Generación por restricción DEFAULT**

Por ejemplo:

**create** **table** taller (

id\_taller number(9) **not** **null** **primary** **key**,

tipo number(4) **default** 3 **not** **null**, -- DEFAULT

direccion varchar2(200) **not** **null**

);

Dado que HotRod identifica las columnas con restricción DEFAULT, se configura en HotRod como:

<table name=*"taller"* />

Al insertar, si no se especifica el "tipo" de taller, entonces la base de datos inserta un valor "3".

**6. Todo combinado**

Por ejemplo:

**create** sequence sec\_contabilidad;

**create** sequence sec\_id\_externo;

**create** **table** vendedor (

id number(9) generated **by** **default** **as** identity **not** **null** **primary** **key**, -- identity

nombre varchar2(40) **not** **null**,

codigo\_contabilidad number(12) **not** **null**, -- generado por secuencia

id\_externo number(12) **not** **null**, -- generado por sequencia

id\_sucursal number(6) **default** 106 **not** **null**, -- DEFAULT constraint

tipo\_vendedor varchar2(20) **default** 'JUNIOR' **not** **null** -- DEFAULT constraint

);

Se configura en HotRod como:

<table name=*"vendedor"*>

<auto-generated-column name=*"codigo\_contabilidad"* sequence=*"sec\_contabilidad"* />

<auto-generated-column name=*"id\_externo"* sequence=*"sec\_id\_externo"* />

</table>

Aunque hay 5 columnas autogeneradas (id, codigo\_contabilidad, id\_externo, id\_sucursal, y tipo\_vendedor) sólo las columnas generadas por secuencias deben ser declaradas explícitamente. HotRod obtiene la metadata de las otras automáticamente.

**Nota**: El ejemplo está hecho para Oracle 12c. Oracle soporta un máximo de una columna identity por tabla. Si este ejemplo fuera en PostgreSQL, se podría definir múltiples columnas identity.

**QUÉ COLUMNAS RECUPERAR**

No estoy seguro de qué valores recuperar automáticamente durante un insert. Se me ocurre que debería recuperar siempre las columnas IDENTITY y las columnas generadas por secuencias. Creo que las columnas DEFAULT podrían ser recuperadas también, pero eso depende de cada llamada (paramétricamente?), porque podríamos estar hablando de columnas pesadas. En cualquier caso, yo las dejaría afuera por defecto.

Yo creo que esta discusión es teórica, y no necesariamente urgente de implementar. Hemos sobrevidido harto bien tal como estamos. Eso sí hay casos (más raros) que no estamos cubriendo.

Bueno, eso sería.

Vladimir