

Implementación Bases de Datos Relacionales

Cuestiones a Resolver:

- Modelo de Implementación (A / B)
- Atributos derivados / calculados
- Integridad Semántica (Disparadores)

Tipos de Implementación

Tipo de Implementación “A”

El usuario y/o las aplicaciones tienen *acceso directo* a las tablas de la BD

Tipo de Implementación “B”

El usuario y/o las aplicaciones *NO tienen acceso directo* a las tablas de la BD, sino que interactúan con una ***API de stored procedures***

Tipo Implementación “A”

Problema: implementación de reglas semánticas.

¿Cómo proteger los datos de control?: Tablas Auxiliares, más código, más complejidad

Gestión de Permisos

Implementación de atributos derivados o calculados

No hay aislación entre usuario/aplicación y la BD real

Tipo Implementación “B”

Datos de control protegidos

Facilidad en Gestión de Permisos

Facilidad en implementación atributos derivados o calculados

Mayor cantidad de código a implementar

Aislación entre usuario/aplicación y la BD real

Atributos derivados o calculados

Implementación como atributo físico

Problema: protección en impl. Tipo A
+Performance Consultas

Implementación como atributo derivado

Simple: clausula COMPUTED BY
Complejo: utilización de vistas,
select procedures
-Performance Consultas

Integridad Semántica

Disparadores o “Triggers”

Integridad Semántica:

Reglas de negocio

Cardinalidad Máxima

Tipo Sub-Tipo Mutuamente Excluyente

Generators

Entidad Fuerte – Entidad Debil con discriminante
correlativo ascendente

...

Programación de BD

BD “Boba” vs BD “Inteligente”

Integridad Semántica

Disparadores o “Triggers”

Código que se ejecuta ante evento:
Insert, Update, Delete sobre tablas/views
Alias NEW, OLD:

EVENTO	NEW	OLD
INSERT	Nueva tupla a insertar	NULL
UPDATE	Tupla modificada	Tupla con valores antes del cambio
DELETE	NULL	Tupla a borrar

Disparadores o “Triggers”

Secuencia de Eventos

1) Eventos Before

Insert, Update, Delete

2) Constraints: Unique Key, Primary Key, Foreign Key, Check, Domain, etc

3) Eventos After

Insert, Update, Delete

Disparadores o “Triggers”

- Dentro del contexto de una transacción
- Cuanto más avance.. más costoso será hacer un rollback
- No codificar ni controlar constraints, la BD lo hará de todas formas
- Codificar teniendo en cuenta la secuencia 1) 2) 3) antes descripta

Disparadores o “Triggers”

- Restricciones:

No usar DDL No hacer COMMIT

No leer la tabla sobre la cual se disparó el trigger

Logica: detectar lo malo y lanzar excepción

Considerar tipo de implementación A o B para saber donde poner los datos de control

Alias NEW modificable en evento BEFORE

Alias NEW no modificable en evento AFTER

Disparadores o “Triggers”

- Restricciones:

No son “invocables” o “ejecutables”

Su ejecución es indirecta, como respuesta a un evento de inserción, update o delete

Pueden invocar un select procedure o execute procedure (forma de reuso de código)

Pueden ejecutar funciones externas e internas

Utilizan cursores y código PL/SQL no interactivo

Disparadores o “Triggers”

- Restricciones:

Debería ser código simple, corto, bien diseñado, apoyado en datos de control

Se puede usar un mismo objeto trigger para N eventos, no recomendable

Usar nomenclatura para nombres de objetos trigger para tener claro el evento asociado

Se puede otorgar permiso al objeto trigger sobre otro objeto, por ejemplo la tabla que debe modificar

Disparadores o “Triggers”

Una metodología posible

Metodología posible:

accion describe la tarea
a realizar

tabla1 .. N tablas que
intervienen en el control
a implementar

Determinar datos de control
a utilizar y dónde ubicarlos

Evento	tabla1	tabla2	tabla3	...
Before insert	accion	accion	accion	accion
After insert	accion	accion	accion	accion
Before Delete	accion	accion	accion	accion
After Delete	accion	accion	accion	accion
Before Update	accion	accion	accion	accion
After Update	accion	accion	accion	accion

Disparadores o “Triggers” Vs Tipo Implementación A o B

- Implementación tipo A:

Usar tablas auxiliares para datos de control y vistas que derivan de tablas auxiliares

Otorgar permisos a trigger sobre tablas auxiliares, el usuario no tiene permisos a tablas auxiliares

- Implementación de tipo B:

No usar tablas auxiliares, datos de control en tablas del sistema

Ejemplos de código

Para Implementaciones de tipo A, B

- Generator
- Tipo Sub-Tipo mutuamente excluyente
- Cardinalidad máxima
- Entidad Fuerte-Debil con discriminante ascendente consecutivo
- Log de actividad

Bibliografía

- [1] Cherencio, G., “Implementación de Triggers en Firebird/Interbase SQL Server”,
<http://www.grch.com.ar/docs/bdd/apuntes/unidad.i/11078-programacion-triggers.pdf>