



**TRABAJO PRACTICO VII – Secuencia de ejecución de Triggers y Constraints**

El objetivo de este trabajo es determinar cual es la secuencia de ejecución de los triggers con respecto a los constraints en Firebird. ¿Qué se ejecuta primero? ¿Un constraint o un trigger de before o un trigger de after? ¿En todos los tipos de eventos sucede lo mismo? ¿Qué sucede con los constraints de integridad referencial que controlan las relaciones padre – hijo? Determinar este tipo de cuestiones es de vital importancia para la correcta implementación de controles y reglas de negocio.

1) En primer lugar, cree la siguiente tabla (elija los tipos de datos que desee):

PRUEBA(PK,DESCR,CANTIDAD)

Donde PK es la clave primaria, todos los atributos son obligatorios. CANTIDAD debe ser mayor que cero. DESCR es una clave candidata o aspirante. Cuando cree la tabla, indique cada constraint de la siguiente forma (para permitir luego su fácil identificación cuando Firebird emita sus mensajes de error):

CONSTRAINT	REGLA DE INTEGRIDAD
c_prueba_pk	Primary key constraint
c_prueba_ak	Unique key constraint (control clave candidata DESCR)
c_prueba_ck	Check constraint (control CANTIDAD > 0 )

2) Cree 6 objetos exception y 6 objetos trigger para controlar cada uno de los eventos posibles en esta tabla, de manera tal que cada trigger lance una determinada excepción, de la siguiente forma:

EVENTO	TIPO	TRIGGER	EXCEPTION	ERROR
INSERT	BEFORE	trg_biprueba	ex_biprueba	'ejecuto trg_biprueba'
INSERT	AFTER	trg_aiprueba	ex_aiprueba	'ejecuto trg_aiprueba'
UPDATE	BEFORE	trg_buprueba	ex_buprueba	'ejecuto trg_buprueba'
UPDATE	AFTER	trg_auprueba	ex_auprueba	'ejecuto trg_auprueba'
DELETE	BEFORE	trg_bdprueba	ex_bdprueba	'ejecuto trg_bdprueba'
DELETE	AFTER	trg_adprueba	ex_adprueba	'ejecuto trg_adprueba'

3) Ahora comience a interactuar “metódicamente” con la tabla prueba, de forma de determinar el orden de ejecución y poder responder las preguntas que se planteaban al comienzo. Deberá ingresar datos que no violen ninguna regla de integridad y datos que pretendan hacer lo contrario. Deberá habilitar y deshabilitar triggers (usar ALTER TRIGGER ) para comprobar el cambio de los mensajes de error de Firebird y poder completar el ingreso de datos. Sea cuidadoso en esta tarea.

Indicaremos con las siglas: CK (check constraint), PK (primary key constraint), FK (foreign key constraint), AK (alternative/candidate/unique key constraint), TBI (trigger before insert), TAI (trigger after insert), TBD (trigger before delete), TAD (trigger after delete), TBU (trigger before update), TAU (trigger after update).

Complete el siguiente cuadro, utilizando las siglas indicadas:



SE VIOLAR	INTENTO	EVENTO	SECUENCIA	COMENTARIOS
c_prueba_pk		INSERT	Ej: TBI, CK, PK, TAI	
c_prueba_pk		UPDATE		
c_prueba_pk		DELETE		
c_prueba_ak		INSERT		
c_prueba_ak		UPDATE		
c_prueba_ak		DELETE		
c_prueba_ck		INSERT		
c_prueba_ck		UPDATE		
c_prueba_ck		DELETE		
Ningún constraint		INSERT		
Ningún constraint		UPDATE		
Ningún constraint		DELETE		

4) ESCRIBA SUS CONCLUSIONES EN CUANTO A LA SECUENCIA DE EJECUCION DE TRIGGERS Y CONSTRAINTS SOBRE UNA TABLA SIMPLE

5) Ahora agregue la siguiente tabla (elija los tipos de datos que desee):

PRUEBAH(PK,FK,DESCR,CANTIDAD)

Donde PK es la clave primaria, todos los atributos son obligatorios. CANTIDAD debe ser mayor que cero. DESCR es una clave candidata o aspirante. FK es la foreign key con respecto a la tabla PRUEBA (no active delete, update en cascada, ni set null, ni set default). Cuando cree la tabla, indique cada constraint de la siguiente forma (para permitir luego su fácil identificación cuando Firebird emita sus mensajes de error):

CONSTRAINT	REGLA DE INTEGRIDAD
c_pruebah_pk	Primary Key constraint
c_pruebah_fk	Foreign Key constraint
c_pruebah_ak	Unique key constraint (control clave candidata DESCR)
c_pruebah_ck	Check constraint (control CANTIDAD > 0)

6) Ingrese en PRUEBAH los mismos datos que actualmente tiene en la tabla PRUEBA, vinculando tuplas de PRUEBAH con tuplas de PRUEBA.

7) Cree 6 objetos exception y 6 objetos trigger para controlar cada uno de los eventos posibles en esta tabla, de manera tal que cada trigger lance una determinada excepción, de la siguiente forma:

EVENTO	TIPO	TRIGGER	EXCEPTION	ERROR
INSERT	BEFORE	trg_bipruebah	ex_bipruebah	'ejecuto trg_bipruebah'
INSERT	AFTER	trg_aipruebah	ex_aipruebah	'ejecuto trg_aipruebah'
UPDATE	BEFORE	trg_bupruebah	ex_bupruebah	'ejecuto trg_bupruebah'
UPDATE	AFTER	trg_aupruebah	ex_aupruebah	'ejecuto trg_aupruebah'
DELETE	BEFORE	trg_bdpruebah	ex_bdpruebah	'ejecuto trg_bdpruebah'
DELETE	AFTER	trg_adpruebah	ex_adpruebah	'ejecuto trg_adpruebah'

8) Ya sabemos la secuencia de ejecución sobre una tabla simple, ahora debemos investigar cuál es la secuencia de ejecución cuando hay operaciones que involucran a una tabla padre y una tabla hija, vinculadas a través de una foreign key.

De la misma forma que lo hicimos con anterioridad, complete el siguiente cuadro:



SE INTENTO VIOLAR	EVENTO	SECUENCIA	COMENTARIOS
c_pruebah_fk	INSERT		
c_pruebah_fk	UPDATE		

9) Ahora, intentemos hacer cambios en PRUEBA que afectan a PRUEBAH debido a la existencia del constraint c\_pruebah\_fk, indicando la secuencia de la misma forma que lo hicimos con anterioridad, anteponiendo el nombre de la tabla delante, Ejemplo: PRUEBA.TBU, PRUEBAH.FK, etc.

ACCION	SECUENCIA	COMENTARIOS
Realizar un update sobre una sola tupla de la tabla PRUEBA cambiando el valor del atributo PRUEBA.PK por otro valor no existente, sobre una tupla que esté relacionada con otra(s) tupla(s) de la tabla PRUEBAH. Esto provocará un intento de violación del constraint c_pruebah_fk en PRUEBAH (puesto que no está declarado un update en cascada).		
Realizar un delete sobre una sola tupla de la tabla PRUEBA que esté relacionada con otra(s) tupla(s) de la tabla PRUEBAH. Esto provocará un intento de violación del constraint c_pruebah_fk en PRUEBAH (puesto que no está declarado un delete en cascada).		

**10) ESCRIBA SUS CONCLUSIONES EN CUANTO A LA SECUENCIA DE EJECUCION DE TRIGGERS Y CONSTRAINTS SOBRE UNA RELACION PADRE - HIJO**