Bases de Datos

Anexo a la Tarea 16:

- **Objetivos:**
  - **Ilustrar con un ejemplo qué implicaciones tiene cuándo se comprueban las restricciones.**
  - **Ilustrar cómo la definición de cuándo se comprueba una restricción permite actualizaciones de la base de datos que de otra forma no serían posibles.**

1 COMPROBACIÓN DE LAS RESTRICCIONES DE INTEGRIDAD

Consideremos sólo dos de las tablas que hemos definido en la Tarea 16:

```
PAIS(cod_pais:char(5),nombre:char(20))
  CP:{cod_pais}
  VNN:{nombre}

ACTOR(cod_act:char(5),nombre:char(70),fecha_nac:date,cod_pais:char(5))
  CP:{cod_act}
  VNN:{nombre,fecha_nac,cod_pais}
  CAj:{cod_pais} → Pais(cod_pais)
```

Y supongamos que en un momento dado el contenido es el siguiente:

COD_PAIS	NOMBRE
p1	España
p2	Francia

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p1
a2	ana	01/12/1958	p1
a3	eva	18/07/1989	p1

Las instrucciones que hemos ejecutado para crearlas son:

```
CREATE TABLE PAIS_T16
(COD_PAIS CHAR(5) CONSTRAINT PK_PAIS PRIMARY KEY,
 NOMBRE VARCHAR2(20) NOT NULL);

CREATE TABLE ACTOR_T16
(COD_ACT CHAR(5) CONSTRAINT ACTOR_PK PRIMARY KEY,
 NOMBRE VARCHAR2(70) NOT NULL,
 FECHA_NAC DATE NOT NULL,
 COD_PAIS CHAR(5) NOT NULL
          CONSTRAINT FK_ACTOR_PAIS REFERENCES PAIS_T16 (COD_PAIS));
```

Como para ninguna de las restricciones definidas se ha especificado la cláusula `cuándo_comprobar` que define el modo de comprobación, el modo de todas ellas será `NOT DEFERRABLE INITIALLY IMMEDIATE`, es decir, que las instrucciones anteriores son equivalente a:

```
CREATE TABLE PAIS_T16
(COD_PAIS CHAR(5) CONSTRAINT PK_PAIS PRIMARY KEY NOT DEFERRABLE INITIALLY IMMEDIATE,
 NOMBRE VARCHAR2(20) NOT NULL NOT DEFERRABLE INITIALLY IMMEDIATE);

CREATE TABLE ACTOR_T16
(COD_ACT CHAR(5) CONSTRAINT ACTOR_PK PRIMARY KEY NOT DEFERRABLE INITIALLY IMMEDIATE,
 NOMBRE VARCHAR2(70) NOT NULL NOT DEFERRABLE INITIALLY IMMEDIATE,
 FECHA_NAC DATE NOT NULL NOT DEFERRABLE INITIALLY IMMEDIATE,
 COD_PAIS CHAR(5) NOT NULL NOT DEFERRABLE INITIALLY IMMEDIATE
         CONSTRAINT FK_ACTOR_PAIS REFERENCES PAIS_T16 (COD_PAIS)
                NOT DEFERRABLE INITIALLY IMMEDIATE);
```

Os recuerdo que el modo de comprobación `NOT DEFERRABLE INITIALLY IMMEDIATE` significa que:

- la restricción se comprueba tras cada instrucción que pueda violarla, en caso de violación la instrucción se anula y la transacción continúa.
- el modo de comprobación no se puede cambiar.

Así, si ejecutamos la siguiente transacción:

```
Insert into actor_t16 (cod_act,nombre,fecha_nac, cod_pais)
    Values ('a4','rubén', '05/08/1985','p1');
Insert into actor_t16 (cod_act,nombre,fecha_nac, cod_pais)
    Values ('a1','alba', '07/07/1977','p1');
Insert into actor_t16 (cod_act,nombre,fecha_nac, cod_pais)
    Values ('a5','paz', '15/09/1995','p1');
commit;
```

El sistema dice lo siguiente:

```
1 fila insertadas.


Error que empieza en la línea: 3 del comando :
Insert into actor_t16 (cod_act,nombre,fecha_nac, cod_pais)
       Values ('a1','alba', '07/07/1977','p1')
Informe de error -
ORA-00001: restricción única (LMOTA.ACTOR_PK) violada


1 fila insertadas.

Confirmación terminada.
```

Es decir ha insertado al actor 'a4' (1 fila insertada), al intentar insertar a 'alba' con cod_act ='a1' como ya existe un actor con ese código ('pepe) y la restiction de clave primaria está en modo inmediato, se comprueba la restricción y, como se viola, el sistema rechaza esa inserción pero la transacción continúa insertando a 'paz'. La tabla queda como se muestra después de la transacción:

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p1
a2	ana	01/12/1958	p1
a3	eva	18/07/1989	p1
a4	rubén	05/08/1985	p1
a5	paz	15/09/1995	p1

Supongamos ahora que la definición de la tabla actor_t16 la realizamos con la siguiente instrucción:

```
CREATE TABLE ACTOR_T16
(COD_ACT CHAR(5) CONSTRAINT ACTOR_PK PRIMARY KEY
         DEFERRABLE INITIALLY IMMEDIATE,
 NOMBRE VARCHAR2(70) NOT NULL,
 FECHA_NAC DATE NOT NULL,
 COD_PAIS CHAR(5) NOT NULL
         CONSTRAINT FK_ACTOR_PAIS REFERENCES PAIS_T16 (COD_PAIS));
```

El modo de comprobación `DEFERRABLE INITIALLY IMMEDIATE` significa que:

- la restricción se comprueba tras cada instrucción que pueda violarla, en caso de violación la instrucción se anula y la transacción continúa.
- el modo de comprobación se puede cambiar.

Así, si ejecutamos la misma transacción que antes, el efecto sería el mismo, pero si ejecutamos la siguiente transacción en la que la primera instrucción cambia la comprobación de la clave primaria de la tabla a modo diferido la situación es diferente:

```
Set constraint actor_pk deferred;
Insert into actor_t16 (cod_act,nombre,fecha_nac, cod_pais)
    Values ('a4','rubén', '05/08/1985','p1');
Insert into actor_t16 (cod_act,nombre,fecha_nac, cod_pais)
    Values ('a1','alba', '07/07/1977','p1');
Insert into actor_t16 (cod_act,nombre,fecha_nac, cod_pais)
    Values ('a5','paz', '15/09/1995','p1');
commit;
```

El sistema dice lo siguiente:

```
Constraint ACTOR_PK correcto.


1 fila insertadas.


1 fila insertadas.


1 fila insertadas.


Error que empieza en la línea: 8 del comando :
commit
Informe de error -
ORA-02091: transacción con rollback
ORA-00001: restricción única (LMOTA.ACTOR_PK) violada
02091. 00000 -  "transaction rolled back"
*Cause:    Also see error 2092. If the transaction is aborted at a remote
           site then you will only see 2091; if aborted at host then you will
           see 2092 and 2091.
*Action:   Add rollback segment and retry the transaction.
```

Es decir, ha insertado las tres filas, justo antes del commit la tabla tiene el siguiente contenido:

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p1
a2	ana	01/12/1958	p1
a3	eva	18/07/1989	p1
a4	rubén	05/08/1985	p1
a1	alba	07/07/1977	p1
a5	paz	15/09/1995	p1

Como es fácil observar, esa filas violan la integridad de la clave primaria ya que hay dos actores con el mismo código. Cuando se ejecuta el commit, el sistema comprueba las restricciones que están diferidas (la clave primaria

de la tabla) y como no se cumple, anula la transacción entera. Así al finalizar la transacción la tabla es:

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p1
a2	ana	01/12/1958	p1
a3	eva	18/07/1989	p1

Es decir que no se ha insertado ninguna de las tres filas.

Tiene que quedar claro que el cambio que se ha hecho en la comprobación de la integridad de clave primaria en esa transacción sólo es aplicable a la transacción que ejecuta la instrucción.

2 APLICACIONES DE LA COMPROBACIÓN DIFERIDA DE LA INTEGRIDAD

Como se dijo en la solución de la tarea T16, Oracle no tienen la directriz ON UPDATE CASCADE[1], de manera que, en caso de que en una tabla cambiara el valor de un atributo al que hacen referencia otras tablas mediante claves ajenas, la transmisión de ese cambio, no la hará el sistema automáticamente. Veámoslo con un ejemplo.

Supongamos de nuevo las dos tablas país_t16 y actor_t16 con la definición inicial (todas las restricciones se comprueban en modo inmediato no diferible) y el contenido inicial (3 actores y 2 países). Dado que en Oracle no existe la posibilidad de indicar en una clave ajena ON UPDATE CASCADE, la comprobación de la integridad de la clave ajena de actor_t16 a país_t16 se hará en modo restrictivo, es decir cuando se detecte que se ha violado hay que anular la modificación.

Supongamos ahora que el código de España está equivocado y es p0 en lugar de p1. La base de datos después de corregir el error debería quedar como sigue:

COD_PAIS	NOMBRE
p0	España
p2	Francia

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p0
a2	ana	01/12/1958	p0
a3	eva	18/07/1989	p0

Para conseguir llegar a ese contenido, hay dos posibles transacciones, la primera cambia el código de país en país_t16 y luego lo cambia en actor_t16

```
update pais_t16 set cod_pais='p0' where cod_pais='p1';
update actor_t16 set cod_pais='p0' where cod_pais='p1';
commit;
```

Al ejecutar esa transacción, el sistema da dos errores:

```
Error que empieza en la línea: 1 del comando :
update pais_t16 set cod_pais='p0' where cod_pais='p1'
Informe de error -
ORA-02292: restricción de integridad (LMOTA.FK_ACTOR_PAIS) violada - registro secundario encontrado


Error que empieza en la línea: 2 del comando :
update actor_t16 set cod_pais='p0' where cod_pais='p1'
Informe de error -
ORA-02291: restricción de integridad (LMOTA.FK_ACTOR_PAIS) violada - clave principal no encontrada

Confirmación terminada.
```

El primer error se produce en el primer update, indicando que no se puede cambiar en pais_t16 el código del

[1] Este concepto está explicado en el apartado 3.4.4.3 de la UD1.2.

país a p0 porque hay filas en otra tabla que tienen el valor que va a desaparecer con lo que se violaría la integridad referencial, este update es anulado y la transacción continúa.

El segundo error se produce en el segundo update, indicando que no se puede cambiar en actor_t16 el código del país a p0 porque no hay una fila en la tabla referenciada que tenga el valor p0 en el código (recordad que el primer update ha fallado).

Ejecutemos las instrucciones en orden inverso:

```
update actor_t16 set cod_pais='p0' where cod_pais='p1';
update pais_t16 set cod_pais='p0' where cod_pais='p1';
commit;
```

En este caso los errores son:

```
Error que empieza en la línea: 1 del comando :
update actor_t16 set cod_pais='p0' where cod_pais='p1'
Informe de error -
ORA-02291: restricción de integridad (LMOTA.FK_ACTOR_PAIS) violada - clave principal no encontrada


Error que empieza en la línea: 2 del comando :
update pais_t16 set cod_pais='p0' where cod_pais='p1'
Informe de error -
ORA-02292: restricción de integridad (LMOTA.FK_ACTOR_PAIS) violada - registro secundario encontrado

Confirmación terminada.
```

En conclusión, con esta definición de las tablas, el código de un país nunca podría modificarse a no ser que no tuviera actores (p.e. se puede vambiar el código de Francia sin problemas).

Para resolver este problema, podemos diferir la comprobación de la integridad referencial al final de la transacción, para ello podríamos definir la tabla actor de la forma siguiente:

```
CREATE TABLE ACTOR_T16
(COD_ACT CHAR(5) CONSTRAINT ACTOR_PK PRIMARY KEY,
 NOMBRE VARCHAR2(70) NOT NULL,
 FECHA_NAC DATE NOT NULL,
 COD_PAIS CHAR(5) NOT NULL
          CONSTRAINT FK_ACTOR_PAIS REFERENCES PAIS_T16 (COD_PAIS)
                     DEFERRABLE INITIALLY DEFERRED);
```

El modo de comprobación `DEFERRABLE INITIALLY DEFERRED` significa que:

- la restricción se comprueba tras cada transacción que contenga una instrucción que pueda violarla, en caso de violación la transacción se anula completamente.
- el modo de comprobación se puede cambiar.

Suponiendo de nuevo que tenemos los 2 países y los 3 actores iniciales:

COD_PAIS	NOMBRE
p1	España
p2	Francia

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p1
a2	ana	01/12/1958	p1
a3	eva	18/07/1989	p1

Veamos ahora cuál sería el resultado de ejecutar las dos transacciones anteriores:

```
update pais_t16 set cod_pais='p0' where cod_pais='p1';
update actor_t16 set cod_pais='p0' where cod_pais='p1';
commit;
```

En este caso el mensaje del siguiente es el siguiente:

```
1 fila actualizadas.


3 filas actualizadas.

Confirmación terminada.
```

Es decir que hemos conseguido cambiar el código erróneo.

En este caso, el contenido de la base de datos entre el primer y el segundo update sería:

COD_PAIS	NOMBRE
p0	España
p2	Francia

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p1
a2	ana	01/12/1958	p1
a3	eva	18/07/1989	p1

Ese contenido viola la integridad referencial (hay tres actores con un código de país que no aparece en la tabla país_t16), pero en ese instante, no se comprueba la integridad referencial ya que su comprobación está diferida. El segundo update arregla el problema y cuando se comprueba la integridad referencial en el commit, ya no se viola.

La situación sería similar si se ejecutara la segunad transacción antes vista:

```
update actor_t16 set cod_pais='p0' where cod_pais='p1';
update pais_t16 set cod_pais='p0' where cod_pais='p1';
commit;
```

La base de datos entre las dos instrucciones sería:

COD_PAIS	NOMBRE
p1	España
p2	Francia

COD_ACT	NOMBRE	FECHA_NAC	COD_PAIS
a1	pepe	05/04/1978	p0
a2	ana	01/12/1958	p0
a3	eva	18/07/1989	p0

Violando la integridad referencia, pero de nuevo el segundo update resuelve el problema.