Ficheros y bases de datos

Práctica de claves ajenas y vistas

20 de mayo de 2002

1. Introducción

El objetivo de esta práctica es que el estudiante asimile el concepto de clave ajena y sus reglas de comportamiento ante el borrado o la modificación de la clave primaria a la que se hace referencia. Además, trataremos el apartado de las vistas, que forma parte del tema 4 de teoría. Se dará a conocer qué son las vistas, cuál es su utilidad y cuándo una vista es o no actualizable.

Para todo ello, utilizaremos el SGBD Access de Microsoft ya que tiene un entorno gráfico fácil de utilizar y existe documentación disponible para los estudiantes a través de la web en <http://yan.act.uji.es/E44/libro_access.pdf> y en reprografía: "Access en la práctica", de Gregorio Quintana (publicación número 133).

2. Claves ajenas

Una *clave ajena* es un atributo o un conjunto de atributos de una relación cuyos valores coinciden con los valores de la clave primaria de alguna otra relación (puede ser la misma relación). Las claves ajenas representan *relaciones entre datos*. Se dice que un valor de clave ajena representa una *referencia* a la tupla que contiene el mismo valor en su clave primaria; a esta tupla se le llama *tupla referenciada*.

Cuando se diseña una base de datos relacional, además de establecer cuáles son las tablas que van a formar parte ella y qué atributos forman la clave primaria de cada tabla, se debe establecer qué atributos son claves ajenas, a qué tabla hace referencia cada clave ajena y sus reglas de comportamiento. En este apartado intentaremos comprender bien todos estos conceptos con un caso práctico.

En primer lugar, deberemos crear, y cargar con datos, la base de datos de los proveedores y las piezas. Crearemos las tablas desde la vista diseño y sus nombres y los de sus columas deberán ser los siguientes:

S(sn, snombre, estado, ciudad)
P(pn, pnombre, color, peso, ciudad)
SP(sn, pn, cant)

Las columnas S.estado, P.peso y SP.cant son de tipo entero. Nótese que el valor por defecto en las columnas de tipo numérico es el valor cero ¿no sería más apropiado que fuera nulo? El resto de columnas son cadenas de caracteres. Para fijar la longitud máxima de cada una, os podéis fijar en los datos que vamos a introducir en la base de datos (se muestran más adelante). La clave primaria de cada tabla es la que aparece subrayada. Al crear las tablas, especificar también la clave primaria de cada una. Nótese que automáticamente se crea un índice sin duplicados sobre la clave primaria cuando ésta es simple. ¿Qué sucede cuando la clave primaria es compuesta, como en el caso de la tabla SP?

En Access utilizamos un entorno gráfico para la creación de las tablas, en lugar de hacerlo en SQL (donde utilizaríamos la sentencia `CREATE TABLE`). En este entorno la creación de las claves ajenas se realiza después de crear y guardar las tablas. A las claves ajenas Access las llama *relaciones*.

Hay dos claves ajenas en nuestra base de datos: SP.sn es clave ajena a S y SP.pn es clave ajena a P. Al crear estas claves ajenas, debes especificar lo siguiente:

- Si quieres que el SGBD se encargue de hacer respetar la integridad referencial.
- Cuando se escoge respetar la integridad referencial, hay que especificar las reglas de comportamiento de cada clave ajena, tanto ante la modificación como ante el borrado de la clave primaria a la que referencian. La opción por defecto es **restringir** y en su lugar se puede escoger **propagar**. La opción de **anular** no está disponible. Estas reglas de comportamiento se fijan mediante las casillas:
  - Actualizar en cascada los campos relacionados (**regla de modificación**).
  - Eliminar en cascada los campos relacionados (**regla de borrado**).
- El tipo de relación es detectado automáticamente. ¿En alguna ocasión crees que se detectarán relaciones de muchos a muchos? ¿Y relaciones de uno a uno?
- También se puede especificar el tipo de *join* que se quiere hacer por defecto cuando se utilicen ambas tablas en una consulta de SQL. Esto se hace mediante el botón de tipo de combinación. ¿Qué tipo de *join* se hará en cada caso de los que podemos escoger?

Para nuestras claves ajenas escogeremos, en principio, la opción de restringir el borrado y la modificación (las opciones por defecto). Más adelante cambiaremos estas reglas y veremos cómo cambia el comportamiento de las claves ajenas.

A continuación rellenaremos las tablas de la base de datos. ¿Por qué tabla deberíamos empezar primero: S, P o SP? ¿Tiene esto alguna importancia o no importa por cuál se empiece?

Los datos a introducir en la tabla S son:

sn	snombre	estado	ciudad
S1	Salazar	20	Londres
S2	Jaimes	10	París
S3	Bernal	30	París
S4	Corona	20	Londres
S5	Aldana	30	Atenas

Los datos a introducir en la tabla P son:

pn	pnombre	color	peso	ciudad
P1	tuerca	verde	12	París
P2	perno	rojo	17	Londres
P3	birlo	azul	17	Roma
P4	birlo	rojo	14	Londres
P5	leva	azul	12	París
P6	engrane	rojo	19	París

Los datos a introducir en la tabla SP son:

sn	pn	cant
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

A continuación se harán una serie de ejercicios que permitirán comprender mejor el uso y funcionamiento de las reglas de comportamiento de las claves ajenas. Verás que se te plantean una serie de preguntas. Intenta responder a cada pregunta antes de hacer cada apartado y luego comprueba si estabas o no en lo cierto.

1. En este momento, las reglas de borrado y modificación de ambas claves ajenasso son **restringir**.

   a) ¿Se puede borrar la fila de S que corresponde al proveedor S1?

   b) ¿Se puede borrar la fila de S que corresponde al proveedor S5?

   c) En general ¿qué proveedores se pueden borrar?

d) ¿Se puede borrar de SP el envío del proveedor S1 y la pieza P2?

e) En general ¿qué envíos se pueden borrar?

f) ¿Se puede insertar en SP un envío del proveedor S1 y de la pieza P7 con una cantidad de 300?

g) ¿Se puede insertar en SP un envío del proveedor S1 y de la pieza P6 con una cantidad de 200?

h) ¿Se puede insertar en SP un envío del proveedor S2 y de la pieza P3 con una cantidad de 400?

i) En general ¿cuándo se puede insertar envíos en la tabla SP?

j) ¿Se puede borrar la fila de P que corresponde a la pieza P6?

k) En general ¿qué piezas se pueden borrar?

2. Vuelve a dejar las tablas en su estado original. Cambia las reglas de comportamiento de la clave ajena SP.sn (a S) de **restringir** a **propagar**.

   a) ¿Se puede borrar la fila de S que corresponde al proveedor S1?

   b) ¿Qué ha pasado con los envíos del proveedor S1?

   c) ¿Se puede borrar la fila de S que corresponde al proveedor S5?

   d) En general ¿qué proveedores se pueden borrar?

   e) ¿Se puede borrar de SP el envío del proveedor S2 y la pieza P2?

   f) En general ¿qué envíos se pueden borrar?

   g) ¿Se puede borrar la fila de P que corresponde a la pieza P6?

   h) ¿Se puede borrar la fila de P que corresponde a la pieza P2?

   i) En general ¿qué piezas se pueden borrar?

3. Vuelve a insertar en las tablas aquellas filas que hayas borrado. Cambia las reglas de comportamiento de la clave ajena SP.pn (a P) de **restringir** a **propagar**.

   a) ¿Se puede borrar la fila de S que corresponde al proveedor S1?

   b) ¿Qué ha pasado con los envíos del proveedor S1?

   c) ¿Se puede borrar la fila de S que corresponde al proveedor S5?

   d) En general ¿qué proveedores se pueden borrar?

   e) ¿Se puede borrar de SP el envío del proveedor S2 y la pieza P2?

   f) En general ¿qué envíos se pueden borrar?

   g) ¿Se puede borrar la fila de P que corresponde a la pieza P6?

   h) ¿Se puede borrar la fila de P que corresponde a la pieza P2?

   i) ¿Qué ha pasado con los envíos de la pieza P2?

   j) En general ¿qué piezas se pueden borrar?

4. Vuelve a insertar en las tablas aquellas filas que hayas borrado. Cambia las reglas de comportamiento de ambas claves ajenas, de modo que para el borrado escojas **restringir** y para la modificación escojas **propagar**.

   a) ¿Se puede borrar la fila de S que corresponde al proveedor S1?

   b) ¿Se puede borrar la fila de S que corresponde al proveedor S5?

   c) En general ¿qué proveedores se pueden borrar?

   d) ¿Se puede modificar en S el código del proveedor S1 y cambiarlo por S6?

   e) ¿Qué ha pasado con los envíos del proveedor S1?

   f) ¿Se puede modificar en SP el envío del proveedor S3, poniendo que es del proveedor S4?

   g) ¿Se puede modificar en SP el envío anterior, poniendo que es del proveedor S5?

   h) ¿Se puede modificar en SP el envío anterior, poniendo que es del proveedor S7?

   i) ¿Cuándo se puede modificar el código de un proveedor en S?

   j) ¿Cuándo se puede modificar el código de un proveedor en un envío de SP?

   k) En general ¿qué piezas se pueden borrar?

   l) ¿Cuándo se puede modificar el código de una pieza en P?

   m) En general ¿qué envíos se pueden borrar?

   n) ¿Cuándo se puede modificar el código de una pieza en un envío de SP?

5. Si quieres averiguar qué pasa cuando no se pide respetar la integridad referencial, quita esta opción en las dos claves ajenas y elimina todas las filas de S y todas las de P. Después consulta la tabla SP ¿puedes saber cómo se llaman los proveedores y las piezas correspondientes a cada envío?

   Tenemos una base de datos que viola la integrida referencial: hay envíos de proveedores y de piezas que no existen en la base de datos. En una base de datos relacional esto nunca debe suceder.

3. Vistas

En el modelo relacional, una vista es una relación virtual (una relación que en realidad no existe como tal) y que se construye realizando operaciones como las del álgebra relacional: restricciones, proyecciones, concatenaciones, etc. a partir de las relaciones base de la base de datos. Una vista es una relación virtual que se produce cuando un usuario la consulta.

El esquema externo que los usuarios ven de la base de datos puede tener tanto relaciones base como vistas derivadas de las relaciones base de la base de datos. Cuando un usuario realiza un cambio sobre la vista (no todo tipo de cambios están permitidos), este cambio se realiza sobre las relaciones de las que se deriva.

Las vistas son útiles porque:

- Proporcionan seguridad ocultando partes de la base de datos a ciertos usuarios.
- Permiten que los usuarios accedan a los datos en el formato que ellos desean o necesitan.
  - Se pueden simplificar operaciones sobre las relaciones base que son complejas.
  - Se puede utilizar una vista para ofrecer un esquema externo a un usuario de modo que éste lo encuentre "familiar".
  - Los atributos se pueden renombrar.
  - Se puede cambiar el orden en que se visualizan las columnas.
  - Se pueden hacer restricciones para que sólo se vea un subconjunto horizontal de la relación.
  - Etc.
- Proporcionan independencia de datos a nivel lógico:
  - Si se añade un atributo a una relación, los usuarios no se percatan de su existencia si sus vistas no lo incluyen.
  - Si una relación existente se reorganiza o se divide en varias relaciones, se pueden crear vistas para que los usuarios la sigan viendo como al principio.

Según el estándar de SQL, hay ciertas condiciones bajo las cuales se determina si se permite realizar una actualización sobre una vista:

- Se permiten las actualizaciones de vistas que se definen mediante una consulta simple sobre una sola relación base y que contienen la clave primaria de la relación base.
- No se permiten las actualizaciones de vistas que se definen sobre varias relaciones base.
- No se permiten las actualizaciones de vistas definidas con operaciones de agrupamiento (GROUP BY).
- No se permiten las actualizaciones de vistas definidas con operaciones de conjuntos.

En este apartado vamos a crear algunas vistas con Access y comprobaremos cuáles de ellas son actualizables. En SQL las vistas se crean mediante la sentencia `CREATE VIEW`. En el entorno gráfico de Access a las vistas se las denomina "consultas" y quizá esta no es la mejor denominación, ya que puede crear confusión. En adelante seguiremos haciendo referencia al término correcto (vistas) aunque en Access nos estaremos refiriendo a lo que llama consultas.

Como en el apartado anterior, volveremos a dejar la base de datos en su estado original y se irán planteando una serie de preguntas a las que se debe intentar responder antes de hacer el ejercicio. Para crear las vistas utilizaremos siempre la ventana de trabajo de SQL.

1. Crea una vista que corresponda a la siguiente relación:

   ```
   S[sn,snombre,ciudad]
   ```

   a) ¿Qué sucede si se trata de insertar en la vista el proveedor S6 que se llama Vidal y que está en Atenas? Observa la tabla base sobre la que se ha definido la vista.

   b) ¿La vista es actualizable?

   c) Cambia la definición de la tabla S, especificando que la columna estado no admite nulos (es requerida) y que no tiene valor por defecto. ¿La vista es actualizable?

2. Cambia la definición de la vista del apartado anterior y elimina el código del proveedor.

   a) ¿Qué sucede si se trata de insertar un nuevo proveedor que se llama Gil y que está en París?

   b) ¿Qué sucede si se cambia la ciudad del proveedor Jaimes y pasa de ser París a ser Roma?

   c) ¿La vista es actualizable?

3. Crea una vista que corresponda a la siguiente relación:

   ```
   P[pn,pnombre] JOIN SP JOIN S[sn,snombre]
   ```

   a) ¿Qué sucede si se trata de insertar en la vista un envío de 300 unidades del proveedor S7, que se llama Monfort y la pieza P7 cuyo nombre es tornillo? Observa las tablas base sobre las que se ha definido la vista.

   b) ¿La vista es actualizable?

   c) ¿Debería serlo según el estándar?

   d) ¿Qué sucede si sobre la vista se trata de cambiar el envío de S3 para convertirlo en un envío de S2?

   e) ¿Qué sucede si sobre la vista se cambia el nombre del proveedor S1 por Marcos en el envío de P1?

   f) ¿Qué sucede si se trata de insertar en la vista un envío de 100 unidades del proveedor S2, se pone como nombre Prades, y la pieza es P8, con nombre clavo?

   g) ¿Será actualizable la vista si se elimina de su definición la columna pn?

4. Crea una vista que muestre, para cada proveedor, su código, su nombre, su estado, su ciudad y la cantidad total correspondiente a todos sus envíos.

   a) ¿Se puede insertar en la vista un nuevo proveedor S8, cuyo nombre es Valero, su estado es 30, su ciudad es Roma y que no envía ninguna pieza (su cantidad total será cero ... ¿o nula?)?

   b) ¿La vista es actualizable?

5. Crea una vista que corresponda a la siguiente expresión:

```
(P WHERE color=''rojo'')[pn,pnombre] UNION
(P WHERE color=''azul'')[pn,pnombre]
```

   a) ¿Se puede insertar en la vista una nueva pieza P9 con nombre tornillo?

   b) ¿La vista es actualizable?

6. Crea una vista que obtiene el mismo resultado que la anterior pero que no utiliza operaciones de conjuntos :

```
(P WHERE (color=''rojo'' OR color=''azul''))[pn,pnombre]
```

   a) ¿Se puede insertar en la vista una nueva pieza P9 con nombre tornillo?

   b) Si la vista sólo muestra piezas rojas o azules, y se ha insertado en ella la pieza P9 ¿de qué color es esta nueva pieza? Consulta la tabla de las piezas.

   c) Vuelve a consultar la vista ¿aparece la pieza P9?

   d) ¿La vista es actualizable? ¿Debería serlo?

Si aún tienes fuerzas, puedes descargarte la base de datos de los proveedores y las piezas que hay en el apartado de teoría de la web de la asignatura[1]. Utiliza la versión que tiene resueltos los ejercicios del boletín de álgebra y cálculo relacional, y examina cada una de las vistas para ver si son o no actualizables según su definición.

[1] <http://nuvol.uji.es/~mmarques/f47/teoria/>