

MEMORIA DE LA PRÁCTICA 1

José Ignacio Gómez García, Óscar Gómez Borzdynski

CABECERAS DISEÑO TONTORRÓN

AUTOR (autor_id, nombre, apellido)

ESCRIBE (autor_id, titulo)

LIBRO (título)

TIENE (título, ISBN)

EDICIÓN (ISBN, fecha, nº pág, precio, editor, tapa, idioma, bolsillo S/N)

APLICADO (oferta_id, ISBN)

OFERTA (oferta_id, %, fin, inicio, PFin)

INCLUYE (ISBN, venta_id)

VENTA (venta_id, precio, tarjeta S/N, fecha, oferta S/N)

POSEE (user_id, ccard_num)

CCARD (ccard_num)

REALIZA (ccard_num, venta_id)

USUARIO_FIDELIZADO (user_id)

10% (user_id, venta_id)

CABECERAS DISEÑO OPTIMIZADO

AUTOR (autor_id, nombre, apellido)

ESCRIBE (autor_id, título)

LIBRO (título)

EDICIÓN (ISBN, fecha, nº pág, precio, editor, tapa, idioma, bolsillo S/N, título)

APLICADO (oferta_id, ISBN)

OFERTA (oferta_id, %, fin, inicio, PFin)

INCLUYE (ISBN, venta_id)

VENTA (venta_id, precio, ccard_num, fecha, user_id)

CCARD (ccard_num)

USUARIO_FIDELIZADO (user_id, ccard_num)

ANÁLISIS DE DISEÑO

Se elimina la relación Posee, ya que tiene cardinalidad 1 a 1 y añadimos la clave Ccard_num a la tabla de USUARIO_FIDELIZADO.

Se elimina la relación Tiene añadiendo la clave Título a la tabla EDICIÓN, ya que una edición sólo puede pertenecer a un LIBRO.

También eliminamos la columna oferta S/N de la tabla VENTA, ya que la comprobación de la Oferta se realiza en EDICIÓN, comprobando si el id_oferta es 0 (no se aplica oferta a la edición) o no. De igual manera, se elimina la relación Realiza, ya que se incluye la clave Ccard_num en la tabla VENTA, y se asume que si el pago se realiza en efectivo, Ccard_num = 0.

Eliminamos la relación 10%, ya que se incluye la clave user_id en la tabla VENTA, de modo que si user_id = 0, se asume que la compra la efectúa un usuario no fidelizado y, por lo tanto, no se aplica el descuento.

En cuanto a los posibles inconvenientes del diseño:

- En una venta sólo se incluye un ejemplar de cada edición. Para comprar dos ejemplares, habría que realizar dos ventas.
- Hemos atribuido cada oferta a una única edición, de modo que si una misma oferta se aplica a más de una edición, habría que emplear una nueva oferta, con un nuevo id, pero el mismo porcentaje de descuento.

CONSULTAS SQL

EJERCICIO 1

```
SELECT COUNT *  
  
FROM EDICION AS E  
  
WHERE E.Título = "<Título>"
```

```
SELECT COUNT E.Idioma  
  
FROM EDICION AS E  
  
WHERE E.Título = "<Título>"
```

COMENTARIO: La primera consulta realiza un COUNT sobre una tabla de ediciones con el mismo título, contando el número de ediciones que tiene un libro. Sólo accedemos a la entidad Edición. La segunda consulta realiza un COUNT de los distintos idiomas que se dan en la tabla anterior, de modo que obtenemos el número de traducciones realizadas.

EJERCICIO 2

```
SELECT COUNT *  
  
FROM (SELECT *  
  
      FROM EDICION AS E, VENTA AS V, INCLUYE AS I  
  
      WHERE V.venta_id = I.venta_id AND E.isbn = I.isbn)  
  
      AS L, ESCRIBE AS W, AUTOR AS A  
  
WHERE A.autor_id = E.autor_id AND E.titulo = L.Titulo AND A.nombre = "<nombre>"
```

COMENTARIO: Primero obtenemos una tabla de todas las ediciones vendidas, y después realizamos un COUNT de aquellas ediciones escritas por el autor que nos interesa. Accedemos a las entidades VENTA y EDICIÓN, y pasamos por la relación INCLUYE.

EJERCICIO 3

SELECT COUNT *

FROM (SELECT *

FROM EDICION AS E, VENTA AS V, INCLUYE AS I

WHERE V.venta_id = I.venta_id AND E.isbn = I.isbn AND E.oferta_id != 0) AS L,
ESCRIBE AS W, AUTOR AS A

WHERE A.autor_id = E.autor_id AND E.titulo = L.Titulo AND A.nombre = "<nombre>"

COMENTARIO: Primero obtenemos una tabla de todas las ediciones vendidas sin oferta, y después realizamos un COUNT de aquellas ediciones escritas por el autor que nos interesa. Accedemos a las entidades ESCRIBE y AUTOR.

EJERCICIO 4

SELECT SUM *

FROM (SELECT SUM E.Precio /*Ni user ni oferta*/

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A

WHERE E.isbn = A.isbn AND A.oferta_id = 0 AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id = 0 AND E.editor = "<Editor>")

JOIN

(SELECT SUM E.Precio * 0.9 /*No oferta, sí user */

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A

WHERE E.isbn = A.isbn AND A.oferta_id = 0 AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id != 0 AND E.editor = "<Editor>")

JOIN

(SELECT SUM O.PFin /*con oferta, sin user */

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A, OFERTA AS O

WHERE E.isbn = A.isbn AND A.oferta_id = O.oferta_id AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id = 0 AND E.editor = "<Editor>")

JOIN

(SELECT SUM O.PFin * 0.9 /*con oferta, con user */

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A, OFERTA AS O

WHERE E.isbn = A.isbn AND A.oferta_id = O.oferta_id AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id != 0 AND E.editor = "<Editor>")

COMENTARIO: Realizamos cuatro consultas diferentes, para obtener los precios de venta en las condiciones en las que no haya oferta aplicada ni el comprador esté fidelizado, sí haya oferta pero no esté fidelizado, no haya oferta pero sí esté fidelizado, y no haya oferta ni esté fidelizado. En los casos en que haya oferta, obtenemos el precio almacenado en O.PFin, que hace referencia al precio con la oferta aplicada. En los casos en los que el usuario esté fidelizado, multiplicamos el precio por 0.9, aplicando así un 10% de descuento. Finalmente, sumamos con SUM todos los valores obtenidos de las consultas anteriores. Accedemos a las entidades EDICIÓN, VENTA y OFERTA, y pasamos por las relaciones INCLUYE y APLICADO.

EJERCICIO 5

```
SELECT COUNT *
```

```
FROM USUARIO_FIDELIZADO AS U, VENTA AS V, INCLUYE AS I, EDICIÓN AS E
```

```
WHERE U.user_id = V.user_id AND V.venta_id = I.venta_id AND I.isbn = E.isbn
```

COMENTARIO: Realizamos un COUNT sobre una tabla en la que obtenemos todos los libros comprados por usuarios fidelizados. Accedemos a las entidades USUARIO_FIDELIZADO, VENTA y EDICIÓN, y pasamos por la relación INCLUYE.

EJERCICIO 6

```
SELECT COUNT (DISTINCT U.user_id)
```

```
FROM USUARIO_FIDELIZADO AS U, VENTA AS V, INCLUYE AS I, EDICIÓN AS E
```

```
WHERE U.user_id = V.user_id AND V.venta_id = I.venta_id AND I.isbn = E.isbn AND E.idioma = 'Inglés'
```

COMENTARIO: Obtenemos una tabla con todos los usuarios que han comprado libros en Inglés, y le aplicamos un COUNT DISTINCT, que nos devuelve el número de usuarios distintos que han comprado algún libro. Accedemos a las entidades USUARIO_FIDELIZADO, VENTA y EDICIÓN, y pasamos por la relación INCLUYE.

EJERCICIO 7

SELECT SUM *

FROM (SELECT SUM E.Precio /*Ni user ni oferta*/

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A

WHERE E.isbn = A.isbn AND A.oferta_id = 0 AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id = 0 AND E.idioma = "Francés")

JOIN

(SELECT SUM E.Precio * 0.9 /*No oferta, sí user */

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A

WHERE E.isbn = A.isbn AND A.oferta_id = 0 AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id != 0 AND E.idioma = "Francés")

JOIN

(SELECT SUM O.PFin /*Con oferta, sin user */

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A, OFERTA AS O

WHERE E.isbn = A.isbn AND A.oferta_id = O.oferta_id AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id = 0 AND E.idioma = "Francés")

JOIN

(SELECT SUM O.PFin * 0.9 /*Con oferta, con user */

FROM EDICION AS E, VENTA AS V, INCLUYE AS I, APLICADO AS A, OFERTA AS O

WHERE E.isbn = A.isbn AND A.oferta_id = O.oferta_id AND I.isbn = E.isbn AND i.venta_id = V.venta_id AND V.user_id != 0 AND E.idioma = "Francés")

COMENTARIO: En este caso, hemos seguido la misma dinámica que en el ejercicio 4, solo que en vez de buscar ediciones pertenecientes a un autor, las buscamos en idioma Francés. Accedemos a las mismas entidades y relaciones.

EJERCICIO 8

```
SELECT O.inicio, O.fin
```

```
FROM OFERTA AS O, APLICADO AS A, EDICION AS E
```

```
WHERE E.editor = 'Adelphi' AND A.isbn = E.isbn AND A.oferta_id = O.oferta_id
```

COMENTARIO: Creamos una tabla en la que tenemos las distintas ediciones pertenecientes a la editorial Adelphi, y metemos también las ofertas que se aplican a estas ediciones. De esta tabla sacamos una nueva tabla en la que almacenamos las fechas de inicio y fin de cada oferta. Accedemos a las entidades OFERTA y EDICIÓN, y pasamos por la relación APLICADO.

EJERCICIO 9

```
SELECT U2.user_id
```

```
FROM USUARIO_FIDELIZADO AS U2
```

```
WHERE U2.user_id NOT IN (SELECT U1.user_id
```

```
FROM USUARIO_FIDELIZADO AS U1, VENTA AS V, INCLUYE AS I, EDICIÓN AS E
```

```
WHERE U1.user_id = V.user_id AND V.venta_id = I.venta_id AND I.isbn = E.isbn AND  
E.bolsillo = TRUE)
```

COMENTARIO: Primero hacemos una consulta que nos devuelva una tabla en la que se incluyan los usuarios que han comprado libros de bolsillo. Después, usando un NOT IN, obtenemos los ids de los usuarios que NO han comprado libros de bolsillo. Accedemos a las entidades USUARIO_FIDELIZADO (U1 para los que compran de bolsillo, y U2 para los que no), EDICIÓN y VENTA, y pasamos por la relación INCLUYE.