Anales PANEL'81/12 JAIIO
Sociedad Argentina de informática
e Investigación Operativa. Buenos Aires, 1981

MECANIZACION DE UNA BIBLIOTECA UTILIZANDO UNA BASE DE DATOS RELACIONAL

J. L. Becerril
R. Casajuana
F. Valer
Centro de Investigación UAM-IBM
Paseo de la Castellana 4, Madrid-1, España

J. Muñoz
Facultad de Informática
Universidad Politécnica de Madrid, España

RESUMEN

En este trabajo se describe un sistema diseñado e implementado en el Centro de Investigación UAM-IBM para la mecanización de una biblioteca. El sistema permite realizar las operaciones típicas de mantenimiento y gestión de una biblioteca, así como seleccionar información de modo conversacional.

El sistema esta implementado en PL/I y utiliza como sistema de gestión de base de datos el Sistema R, un prototipo experimental de base de datos relacional desarrollado en el Laboratorio de Investigación de IBM en San José (USA). Los programas de PL/I que realizan las operaciones citadas incluyen sentencias de SQL, lenguaje no procedural que constituye la interfase externa del Sistema R.

INTRODUCCION

En el área de las bases de datos relacionales (BDR), uno de los aspectos más importantes es el de los lenguajes para su tratamiento (DAT75). En ellos podemos distinguir los siguientes aspectos:

- Interrogación (interacción directa del usuario con el sistema de gestión de la BD).
- Manipulación de datos (inserción, eliminación y actualización de tuplas).

- Definición de datos (creación de relaciones y otras estructuras derivadas).
- Control de datos (manejo de transacciones, integridad de los datos y autorizaciones).

Considerando el contexto de la gestión de una biblioteca, y dado el entorno en que se encuentran sus usuarios, es evidente que el lenguaje de tratamiento de la BDR deberá ser en todas sus facetas un lenguaje cómodo de utilizar y que no requiera profundos conocimientos de informática. El modo elegido en nuestro sistema es el de guiar al usuario por medio de menús, evitando así los problemas de utilización de comandos con una sintaxis fija. Otra de las decisiones básicas que se han tomado es que dichos menús engloben todas las funciones necesarias para el tratamiento de la BDR.

El sistema presenta al usuario en la pantalla una serie de menús encadenados donde se le ofrecen las diversas opciones a las que tiene acceso según sea un usuario privilegiado o no. En la zona inferior de la pantalla es donde el usuario escribe los datos exigidos por el sistema según la operación que desee realizar. En la parte superior el usuario tiene presentes los datos que ya ha suministrado al sistema. Cuando el sistema requiera información adicional la solicitará utilizando la zona central de la pantalla. Los errores detectados por el sistema se indican asimismo en la parte central de la pantalla.

Los mensajes de error que da el sistema son en general autoexplicativos. Con el fin de que el sistema sea cómodo de utilizar, los errores no producen ruptura de la comunicación con el usuario.

Los resultados son enviados a la impresora cuando previsiblemente son de gran volumen, como es el caso de listados completos de la biblioteca. En la búsqueda selectiva el usuario puede decidir donde obtener la salida, es decir, por impresora o por la pantalla, disponiendo en este último caso de facilidades de edición y/o copia de la pantalla.

El sistema prepara un fichero, a modo de diario, con la secuencia de operaciones realizadas, que imprime al concluir la sesión, constituyendo una importante ayuda a la hora de una posterior verificación.

Para la realización del sistema se ha utilizado como lenguaje el PL/I y como sistema de gestión de BD el Sistema R (AST76), un prototipo experimental desarrollado en IBM Research (San Jose,USA). Los programas de PL/I que realizan las operaciones citadas incluyen sentencias de SQL (CHA76), lenguaje que constituye la interfase externa del Sistema R. La comunicación con el usuario se realiza a traves de una pantalla IBM 3270.

DISEÑO DE LA BASE DE DATOS

La información considerada en este sistema bibliográfico se han organizado en relaciones según el esquema conceptual que muestra la tabla siguiente:

RELACION	DOMINIO	TIPO
AUTORES	NUM	BIN FIXED(15)
AUTORES	AUTOR	CHAR(40) VAR
AUTORES	TIPO	CHAR(1)
LIBROS	NUM	BIN FIXED(15)
LIBROS	F_EDICION	BIN FIXED(15)
LIBROS	N_PAG	BIN FIXED(15)
LIBROS	N_EJEMPLARES	BIN FIXED(15)
LIBROS	EDITORIAL	CHAR(40) VAR
LIBROS	TITULO	CHAR(130) VAR
LIBROS	I_SIG	BIN FIXED(15)
PRESTAMOS	NUM	BIN FIXED(15)
PRESTAMOS	EJEMPLAR	BIN FIXED(15)
PRESTAMOS	I_USUARIO	BIN FIXED(15)
PRESTAMOS	FECHA	BIN FIXED(31)
SIGNATURAS	I_SIG	BIN FIXED(15)
SIGNATURAS	SIG1	CHAR(5) VAR
SIGNATURAS	SIG2	CHAR(5) VAR
SIGNATURAS	MATERIA	CHAR(50) VAR
USUARIOS	I_USUARIO	BIN FIXED(15)
USUARIOS	NOMBRE	CHAR(30) VAR
USUARIOS	APELLIDO1	CHAR(30) VAR
USUARIOS	APELLIDO2	CHAR(30) VAR
USUARIOS	LOCALIZACION	CHAR(3)
USUARIOS	COMENTARIOS	CHAR(100) VAR

Tabla 1 . Esquema conceptual de la BD.

Se ha tratado de conseguir un equilibrio entre:

- Minimizar el número de relaciones, ya que lo contrario implicaría la necesidad de gran número de "joins" para realizar cualquier operación, lo cual en general es costoso en tiempo.
- Evitar duplicaciones innecesarias de dominios que, además de aumentar el tamaño de la BD, hacen difícil y costoso el mantener la integridad.
- Prever la posibilidad de añadir nueva información sobre los libros, sin que esto suponga la reestructuración de la BD, lo cual equivale a escoger un diseño facilmente extensible.

Observemos que salvo en las relaciones SIGNATURAS y USUARIOS, en todas las restantes hay un dominio común (NUM) correspondiente al número del libro, dominio que permite realizar "joins" entre esas relaciones. Para la relación SIGNATURAS el dominio que permite hacer "joins" con las restantes es I_SIG, el identificador numérico de la signatura, mientras que para la USUARIOS es I_USUARIO, el identificador numérico de usuario. La figura 1 muestra estas conexiones; las flechas indican posibles "joins".

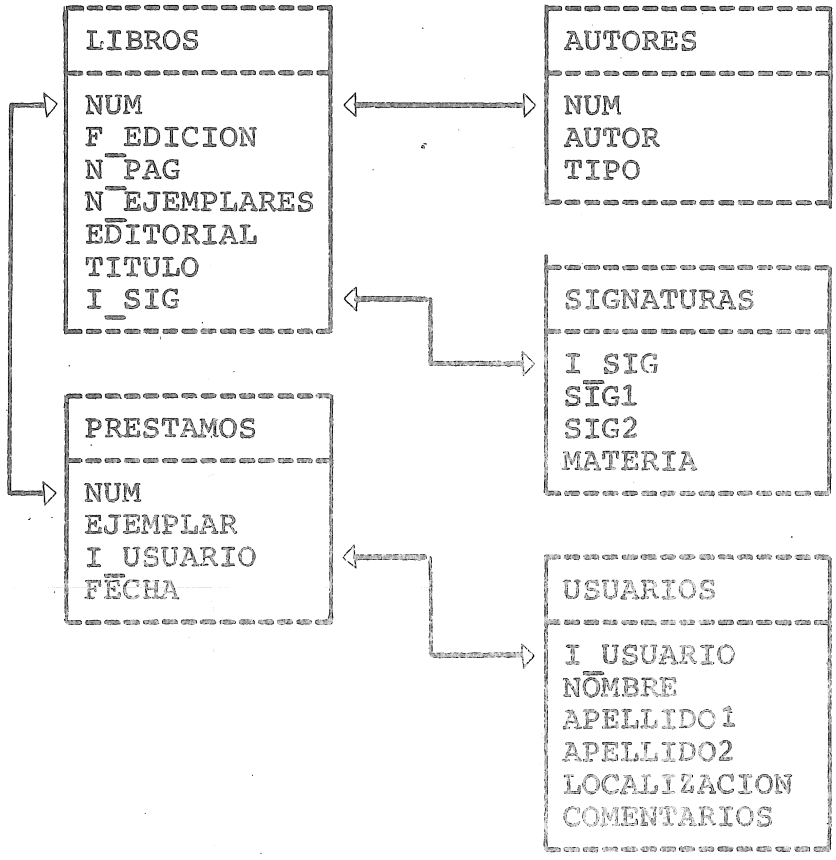

Figura 1 . Posibles conexiones entre las relaciones

Los únicos dominios que no tienen nombres autoexplicativos son TIPO en la relación AUTORES y el par SIG1, SIG2 en SIGNATURAS. TIPO contempla el hecho de que un libro pueda tener uno o varios autores; en el primer caso este dominio contiene una 'U', mientras que en el segundo el primer autor tiene en este dominio una 'P', y los demas el valor ' ' . El par de dominios SIG1, SIG2 contienen la signatura del libro partida en dos campos, entendiendo por signatura un identificador único para cada libro asignado según la materia principal que contenga.

Es de destacar que, dada la flexibilidad de las BDR, la extensión de este esquema, incluyendo nuevos dominios (en el

caso de que se desee almacenar nuevos datos sobre los libros) no provocaría grandes complicaciones en la estructura de los programas y por supuesto, no requeriría una nueva estructuración y carga de la BD. Caso de desear añadir información que sea única por libro (ej. precio) sólo será preciso expandir la relación libros con los dominios necesarios. En el caso de que a cada libro corresponda más de un elemento de información (ej. palabras clave), deberá crearse una nueva relación conectada con LIBROS mediante el dominio NUM que contenga la información deseada, análogamente a lo que ocurre con las relaciones AUTORES o SIGNATURAS.

Con el fin de que las interrogaciones sean mas eficientes, se han creado índices (ordenaciones por valor) sobre los dominios mas significativos en las operaciones mas típicas. Existe la posibilidad adicional de añadir o cambiar los índices, con lo que se consigue "adaptar" la BD a las aplicaciones más usuales o conceder un trato privilegiado a ciertas operaciones.

Se han creado índices en el dominio NUM de las relaciones AUTORES, LIBROS y PRESTAMOS, y en I_SIG de las relaciones LIBROS y SIGNATURAS, en I_USUARIO de las relaciones PRESTAMOS y USUARIOS, con la idea de disminuir el costo de los "joins" entre las relaciones de mayor cardinalidad. Con el fin de que algunas busquedas típicas sean más rápidas, también se han creado índices en el dominio AUTOR de la relación AUTORES y en el par SIG1, SIG2 de la relación SIGNATURAS.

OPCIONES DEL SISTEMA

El sistema permite realizar operaciones que podemos agrupar en:

- Mantenimiento y gestión de la biblioteca: entrada de libros, gestión de préstamos, archivo de usuarios, obtención de listados de la biblioteca completa ordenados por autores, signaturas, usuarios, etc.

- Búsqueda conversacional, seleccionando entre la información almacenada según ciertos criterios elementales o combinados (fecha de edición, autor, etc.).

Los dos modos de operación implican dos tipos de usuarios diferentes; el primero es el administrador de la biblioteca, mientras que el segundo es el auténtico usuario final de un sistema bibliográfico. No obstante, se ha tratado de que la forma de utilización del sistema sea igualmente sencilla en cualquiera de los casos sin requerir apenas aprendizaje, siempre preservando los niveles de seguridad imprescindibles en cuanto a acceso a la actualización y mantenimiento de la integridad de los datos almacenados.

El sistema se invoca tecleando la palabra LIBROS. Aparece el primer menú que ofrece al usuario las posibilidades de:

- Actualización de la biblioteca (ACT).
- Listados de la biblioteca (LIS).
- Busqueda coversacional (BUS).

Existen otras opciones que no aparecen en el menú para uso del administrador de la base de datos, que permiten operaciones privilegiadas como creación de índices, obtención de información sobre el catálogo de la BD, etc.

Las operaciones típicas de mantenimiento y gestión se realizan escogiendo en el primer menú alguna de las dos opciones ACT ó LIS. Desde la opción ACT, las operaciones posibles son:

- Entrada de libros (LIB).
- Actualización del catálogo de signaturas (CAT).
- Actualización de bajas (BAJ).
- Actualización de préstamos (PRE).
- Devolución de libros (DEV).
- Actualización de la lista de usuarios (AUS).

En función de la opción escogida el sistema va solicitando al usuario los datos necesarios para su posterior proceso y almacenamiento en la BD.

Desde la opción LIS, las operaciones posibles son:

- Listado de la biblioteca completa por autores (AUT).
- Listado de la biblioteca completa por signaturas (LSG).
- Catálogo de signaturas (CSG).
- Listado de la biblioteca por números (NUM).
- Listado de préstamos por usuarios (LPU).
- Listado de préstamos por números (LPN).
- Listado de usuarios (USU).
- Todos los anteriores (TODOS).

Las operaciones relativas a búsqueda conversacional se realizan escogiendo en el primer menú la opción BUS. Desde esta opción las operaciones posibles son:

- Obtención de libros que contengan en su título una cierta palabra clave (TITC).
- Obtención de libros con un autor o autores determinados (AUTC).
- Obtención de libros cuyo año de edición sea N, mayor que N, o entre N y M (FEDC).
- Obtención de libros con una cierta signatura (SIGC).
- Obtención de libros que contengan en su materia una cierta palabra clave (MATC).
- Obtención de libros que pertenezcan a una editorial (EDIC).

- Mezcla de las posibilidades anteriores (MEZC).

IMPLEMENTACION DE LAS OPCIONES

En esta sección vamos a describir cómo se ha realizado la implementación de las distintas opciones del sistema. Todas ellas tienen como parte básica sentencias de SQL (interfase externa del Sistema R) incluidas en programas PL/I. En la descripción de las sentencias de SQL utilizaremos nombres que comienzan por el signo "$" cuando hagamos referencia a variables del programa, para no confundirlas con los mismos nombres utilizados como identificadores de dominios.

LIB. Entrada de libros.

Se trata de almacenar en la base de datos la información correspondiente a un libro nuevo. En primer lugar, conocida la signatura (el par $SIG1, $SIG2), se localiza el identificador de signatura ($ISIG) que le corresponde.

```
SELECT I_SIG INTO $ISIG
FROM   SIGNATURAS
WHERE  SIG1=$SIG1 AND SIG2=$SIG2
```

Si no existe la signatura se calcula el nuevo identificador y se insertan en la relación SIGNATURAS los datos correspondientes.

```
SELECT MAX(I_SIG) INTO $ISIG
FROM   SIGNATURAS

INSERT INTO SIGNATURAS :
  (<$ISIG+1,$SIG1,$SIG2,$MAT>)
```

A continuación se inserta el resto de los datos en las tres relaciones involucradas (AUTORES, PRESTAMOS y LIBROS). Siempre se supone que un libro nuevo no está prestado, es decir $IUSU=1.

```
INSERT INTO AUTORES :
  (<$NUM,$AUT,$TIPO>)

INSERT INTO PRESTAMOS :
  (<$NUM,$EJ,$IUSU,$FECHA>)

INSERT INTO LIBROS :
  (<$NUM,$ISIG,$FEDI,$NPAG,$NEJ,$EDI,$TIT>)
```

En el caso de querer añadir un ejemplar de un libro ya existente registrado con el número $NUM, sólo es necesario conocer el número del último ejemplar, actualizar el número de

ejemplares en la relación LIBROS e insertar una nueva tupla en PRESTAMOS indicando que existe un nuevo ejemplar del libro citado.

```
SELECT MAX(EJEMPLAR) INTO $EJE
FROM   PRESTAMOS
WHERE  NUM=$NUM

UPDATE LIBROS
SET    N_EJEMPLARES = N_EJEMPLARES + 1
WHERE  NUM=$NUM

INSERT INTO PRESTAMOS :
  < $NUM,$EJE,$IUSU,$FECHA >
```

CAT. Actualización del catálogo de signaturas.

Para añadir una signatura nueva, en primer lugar se calcula el identificador que le va a corresponder ($ISI + 1) y después se inserta una tupla en la relación SIGNATURAS.

```
SELECT MAX(I_SIG)
INTO   $ISI
FROM   SIGNATURAS

INSERT INTO SIGNATURAS :
< $ISI+1,$SIG1,$SIG2,$MAT >
```

BAJ. Actualización de bajas.

En primer lugar se obtienen los datos correspondientes al libro con número $NUM y ejemplar $EJE.

```
SELECT PRESTAMOS.I_USUARIO,NOMBRE,APELLIDO1,APELLIDO2
INTO   $IUSU,$NOM,$AP1,$AP2
FROM   PRESTAMOS,USUARIOS
WHERE  PRESTAMOS.I_USUARIO = USUARIOS.I_USUARIO
AND    PRESTAMOS.NUM = $NUM
AND    PRESTAMOS.EJEMPLAR = $EJE
```

A continuación se elimina la información correspondiente al libro que causa baja.

```
DELETE PRESTAMOS
WHERE  NUM=$NUM
AND    EJEMPLAR=$EJE

DELETE LIBROS WHERE NUM=$NUM

DELETE AUTORES WHERE NUM=$NUM
```

Seguidamente se actualiza el número de ejemplares.

```
UPDATE LIBROS
SET    N_EJEMPLARES = N_EJEMPLARES-1
WHERE  NUM=$NUM
```

PRE, DEV. Actualización de préstamos o devoluciones.

En ambos casos, tras detectar si el libro a prestar (devolver) está en la biblioteca (prestado a ese usuario), se debe actualizar la relación PRESTAMOS de modo que refleje la nueva situación del libro, es decir, se cambia la fecha y la identificación del usuario.

```
UPDATE PRESTAMOS
SET    I_USUARIO = $I_USU,
       FECHA     = $FECHA
WHERE  NUM       = $NUM_PRE
AND    EJEMPLAR  = $EJE_PRE
```

AUS. Actualización de usuarios.

En este caso, tras calcular la identificación que le va a corresponder al nuevo usuario, se inserta una tupla en la relación USUARIOS con los datos pertinentes.

```
SELECT MAX(I_USUARIO)
INTO   $IUS
FROM   USUARIOS

INSERT INTO USUARIOS:
  <$IUS+1,$NOM,$APL1,$APL2,$LOC,$COM>
```

AUT. Listado por autores

La sentencia siguiente:

```
LET C1 BE
  SELECT AUTOR,TITULO,LIBROS.NUM,N_EJEMPLARES
         EDITORIAL,F_EDICION,N_PAG,SIG1,SIG2
  INTO   $AUT,$TIT,$NUM,$NEJ,$EDI,$FEDI,$NPAG,$SIG1,$SIG2
  FROM   LIBROS,AUTORES,SIGNATURAS
  WHERE  LIBROS.NUM = AUTORES.NUM
  AND    LIBROS.I_SIG = SIGNATURAS.I_SIG
  ORDER BY 1,2
```

define un cursor, que en sucesivos "fetch" pondrá en las variables asociadas los datos necesarios en orden alfabético de autores y títulos.

LSG. Listado por signaturas.

El cursor S1 nos suministrará el identificador I_SIG de cada signatura, además de la materia asociada a cada una de ellas.

```
LET S1 BE
  SELECT I_SIG,SIG1,SIG2,MATERIA
  INTO   $ISIG,$SIG1,$SIG2,$MAT
  FROM   SIGNATURAS
  ORDER BY 2,3
```

Para cada valor de la variable $ISIG se define el cursor C1, mediante el cual se obtienen todos los libros con esa signatura (condición I_SIG=$ISIG). La condición sobre el dominio TIPO es necesaria para limitar el resultado al primer autor de cada libro.

```
LET C1 BE
  SELECT AUTOR,TITULO,LIBROS.NUM,N_EJEMPLARES,
         EDITORIAL,F_EDICION,N_PAG,TIPO
  INTO   $AUT,$TIT,$NUM,$NEJ,$EDI,$FEDI,$NPAG,$TIPO
  FROM   LIBROS,AUTORES
  WHERE  LIBROS.NUM = AUTORES.NUM
  AND    LIBROS.I_SIG = $ISIG
  AND    AUTORES.TIPO  ≠ ' '
  ORDER BY 1,2
```

CSG. Catálogo de signaturas.

El siguiente cursor proporciona los identificadores y la significación para cada signatura:

```
LET C1 BE
  SELECT SIG1,SIG2,MATERIA
  INTO   $SIG1,$SIG2,$MAT
  FROM   SIGNATURAS
  ORDER BY 1,2
```

NUM. Listado por números.

Este caso es muy similar a la segunda parte del LSG:

```
LET C1 BE
  SELECT LIBROS.NUM,TITULO,N_EJEMPLARES,
         SIG1,SIG2,AUTOR,TIPO
  INTO   $NUM,$TIT,$NEJ,$SIG1,$SIG2,$AUT,$TIPO
  FROM   LIBROS,AUTORES,SIGNATURAS
  WHERE  LIBROS.NUM = AUTORES.NUM
  AND    LIBROS.I_SIG = SIGNATURAS.I_SIG
  AND    AUTORES.TIPO  ≠ ' '
  ORDER BY 1
```

LPU. Listado de préstamos por usuarios.

El cursor C1 nos suministra los datos de cada usuario, además de su identificador (I_USUARIO), con excepción del usuario número 1 que corresponde a la propia biblioteca.

```
LET C1 BE
  SELECT I_USUARIO,NOMBRE,APELLIDO1,
         APELLIDO2,LOCALIZACION,COMENTARIOS
  INTO   $IUSU,$NOM,$AP1,$AP2,$LOC,$COM
  FROM   USUARIOS
  WHERE  I_USUARIO  ≠ 1
  ORDER BY 3
```

La sentencia siguiente calcula el número de tuplas de la relación PRESTAMOS que verifican la condición I_USUARIO=$IUSU, es decir, para cada identificador de usuario calcula cuantos libros tiene préstados.

```
  SELECT COUNT(*) INTO $N
  FROM   PRESTAMOS
  WHERE  I_USUARIO = $IUSU
```

LPN. Listado de préstamos por números.

Este cursor selecciona todos los libros que aparecen en la relación PRESTAMOS con un identificador de usuario distinto del de la biblioteca (≠1).

```
LET C1 BE
  SELECT LIBROS.NUM,TITULO,SIG1,SIG2,
         NOMBRE,APELLIDO1,APELLIDO2,EJEMPLAR,FECHA
  INTO   $NUM,$TIT,$SIG1,$SIG2,$NOM,$AP1,$AP2,$EJE,$FECHA
  FROM   LIBROS,SIGNATURAS,USUARIOS,PRESTAMOS
  WHERE  LIBROS.NUM = PRESTAMOS.NUM
  AND    LIBROS.I_SIG = SIGNATURAS.I_SIG
  AND    PRESTAMOS.I_USUARIO = USUARIOS.I_USUARIO
  AND    PRESTAMOS.I_USUARIO  ≠ 1
  ORDER BY 1
```

USU. Listado por usuarios.

En este caso el cursor U1 obtiene los datos de cada uno de los usuarios.

```
LET U1 BE
  SELECT I_USUARIO,NOMBRE,APELLIDO1,APELLIDO2
  INTO   $IUSU,$NOM,$AP1,$AP2
  FROM   USUARIOS
  WHERE  I_USUARIO  ≠ 1
  ORDER BY 3
```

El cursor C1, para cada valor de $IUSU, obtiene todos los datos de los libros prestados a ese usuario.

```
LET C1 BE
  SELECT LIBROS.NUM,TITULO,AUTOR,TIPO,FECHA,EJEMPLAR
  INTO   $NUM,$TIT,$AUT,$TIPO,$FECHA,$EJE
  FROM   LIBROS,AUTORES,PRESTAMOS
  WHERE  LIBROS.NUM = PRESTAMOS.NUM
  AND    LIBROS.NUM = AUTORES.NUM
  AND    PRESTAMOS.I_USUARIO = $IUSU
  AND    AUTORES.TIPO  ≠ ' '
  ORDER BY 5
```

Búsqueda conversacional.

En líneas generales, el proceso a seguir en todos los casos de la búsqueda conversacional consiste en encontrar los números de los libros que verifican las condiciones indicadas por el usuario. Posteriormente, es fácil obtener los datos restantes de los libros involucrados mediante sentencias similares a las anteriormente descritas.

Para encontrar los números de los libros seleccionados hay tres casos muy similares, en los que la sentencia será del tipo:

```
LET C1 BE
  SELECT  NUM     INTO  $NUM
  FROM    LIBROS
  WHERE   .........
```

donde la condición será para la opción:

TITC: TITULO LIKE '%xx...xx%'

FEDC: FECHA = $N (ó <,>)
      FECHA BETWEEN $N AND $M

EDIC: EDITORIAL = $EDI

Para AUTC los números seleccionados se obtienen mediante el cursor:

```
LET C1 BE
  SELECT NUM INTO $NUM
  FROM   AUTORES
  WHERE  AUTOR IN ($A1,$A2, ...,$An)
```

o condiciones similares.

En el caso de SIGC, conocida la signatura ($SIG1,$SIG2) se obtiene el identificador de signatura mediante:

```
LET S1 BE
  SELECT I_SIG INTO $ISIG
  FROM   SIGNATURAS
  WHERE  SIG1=$SIG1  AND SIG2=$SIG2
```

Para el caso de MATC el cursor sería:

```
LET S1 BE
  SELECT I_SIG INTO $ISIG
  FROM   SIGNATURAS
  WHERE  MATERIA LIKE $MAT
```

teniendo $MAT el valor adecuado.

En cualquiera de estos dos últimos casos, una vez conocido el valor o valores de $ISIG, se define el siguiente cursor:

```
LET C1 BE
  SELECT NUM INTO $NUM
  FROM LIBROS
  WHERE I_SIG IN ($ISIG1, ...,$ISIGn)
```

El caso MEZC es una combinación de todos los anteriores.

Citemos como detalle de interés que, salvo el programa asociado a la opción MEZC, todos los demás están preprocesados y preparados para su ejecución, es decir, con su correspondiente módulo de acceso a la base de datos generado, con la consiguiente ventaja en lo referente a tiempo de ejecución.

REFERENCIAS

(AST76) Astrahan M.M. y otros (1976), System R: A Relational Approach to Database Management, ACM Trans. on Database Systems, Vol. 1, No. 2, Junio 1976.

(CHA76) Chamberlin D.D. y otros (1976), SEQUEL 2: A Unified Approach to Data Definition, Manipulation and Control, Research Report RJ1798, IBM Research Lab., San Jose, Cal.,USA. Junio 1976.

(DAT75) Date C.J. (1975), An Introduction to Data Base Systems, Addison-Wesley 1975.