

Tema 1. Introducción

Ficheros y Bases de Datos

17 de abril de 2002

1. Introducción

Este documento contiene preguntas del tema 1 recogidas por estudiantes de la asignatura (Elisa Ortíz López, Laura Pedregosa Valverde y Ernesto Cervera Capdevila) y que han sido contestadas por el profesorado. En ocasiones, las respuestas son bastante extensas y aportan información, sobre todo ejemplos, que no se han utilizado en las clases de teoría y que tampoco aparecen en los apuntes. Esto se ha hecho para intentar responder de un modo más claro a estas preguntas y, por lo tanto, no son conceptos que se añadan al temario de la asignatura (no se podrán preguntar en los exámenes).

2. Preguntas y respuestas

2.1. *¿Qué problemas plantean los sistemas de ficheros?*

- *Duplicidad de datos*, puesto que, distintos departamentos de una institución pueden tener los mismos datos en distintos sistemas de ficheros. Este hecho también conlleva la *Separación y aislamiento de datos* entre unas aplicaciones y otras.
- *Problemas de integridad de datos*, es decir, pueden haber copias de un mismo dato en el mismo sistema de ficheros que no coinciden. Por ejemplo, tenemos la dirección de una misma persona almacenada en dos ficheros distintos y ésta no coincide.

La integridad la deben controlar las aplicaciones puesto que no pueden almacenar reglas de integridad. Debido a la redundancia de datos al realizar actualizaciones (modificaciones o borrados) puede que se actualice un dato en algún fichero y no en otros.

- *Dificultad para procesar consultas*, puesto que se debe diseñar una aplicación específica para cada una que se quiera realizar. Esto conlleva la *Proliferación de programas de aplicación* que acaba desbordando generalmente al departamento encargado de realizarlas.

- *Problema de consistencia de datos*, es decir, datos del mismo tipo almacenados con formatos distintos. Por ejemplo, guardamos varias veces el teléfono unas veces con prefijo y otras sin él, con guiones o sin ellos,
- *Formatos de ficheros incompatibles*, por estar realizadas las aplicaciones en distintos lenguajes de programación, por estar definidos los datos con diferentes formatos tal y como se comenta en el anterior punto,
- *Falta de independencia de datos(lógica-física)*, es decir, que las aplicaciones dependen del tipo de datos que manejan. Una modificación en la definición de los datos obliga a modificar las aplicaciones que trabajan con los datos modificados (e incluso muchas de las que trabajan con otros datos).

2.2. *¿Cuáles son las ventajas de los Sistemas de Bases de Datos (SBD)?*

Las ventajas de los SBD se obtienen por dos motivos: por la *integración de datos* y por la *existencia del Sistema Gestor de Bases de Datos (SGBD)*.

Por la *integración de datos*:

- Existe una menor redundancia de datos, lo cual facilita mantener la integridad de la BD y permite utilizar menos espacio en disco, cosa especialmente importante cuando se está trabajando con BD de cientos de miles o millones de registros.
- Por otro lado, de esta forma se evitan los problemas de consistencia de datos ya que normalmente sólo aparecerán una vez en la BD.
- Se puede extraer información estadística sobre los datos y las relaciones entre ellos.
 - Se pueden compartir los datos entre distintas aplicaciones y departamentos.
 - Se facilita el mantenimiento de los estándares.

Por la *existencia del Sistema Gestor de Bases de Datos*:

- Se reduce el problema de la integridad, puesto que se pueden definir reglas que el SGBD se encargará de mantener.
- Se mejora la seguridad de la Base de Datos, ya que existen diferentes tipos de usuarios que se deben identificar para poder acceder a la Base de Datos y sobre los que se pueden definir distintos tipos de permisos (acceso, modificación, borrado, . . .).
- Se pueden realizar consultas de datos simples sin tener que escribir el programa, por ejemplo mediante sentencias de SQL o QBE (Query By Example).

- Se mejora la productividad de los programadores, puesto que hay que realizar muchas menos comprobaciones al hacer actualizaciones de datos.
- Proporciona independencia de datos lógica-física, lo cual permite realizar modificaciones en la Base de Datos y en las aplicaciones de un modo mucho más sencillo.
- Se aumenta la concurrencia, es decir, el acceso simultáneo de varios usuarios a la Base de Datos.
- Mejora los servicios de copias de seguridad y de recuperación de los datos mediante aplicaciones propias del SGBD.

2.3. *Una de las ventajas de la existencia del SGBD es la reducción del problema de integridad de datos ¿qué es exactamente la integridad de datos?*

Cuando se almacenan datos, bien sea en un sistema de ficheros o en un sistema de bases de datos, se pretende que estos datos siempre reflejen la realidad. Por ejemplo, todos los contratos de alquiler deben ser de inmuebles conocidos (cuyos datos están almacenados en el sistema) y de clientes conocidos; si se almacenan varias copias de los datos de los clientes, todas las copias deben coincidir. En un determinado momento, los datos almacenados pueden ser correctos y que el usuario realice una operación que los lleve a un estado en el que se viola la integridad: por ejemplo, cuando se borran los datos de un inmueble que tiene contratos de alquiler: los datos del inmueble ya no están en la base de datos, y los contratos permanecen, por lo que tenemos contratos de inmuebles que no sabemos en qué calle están ni quién es su propietario. En los sistemas de ficheros son los programas de aplicación quienes se deben encargar de mantener la integridad de los datos. Sin embargo, en los SGBD es el propio sistema el que se encarga de mantener la integridad. Para ello, cuando se crea la base de datos, se dispone de unas directivas que permiten especificar reglas de integridad. El SGBD se encargará de hacer que se respeten.

Por ejemplo, para la inmobiliaria se creará una tabla de inmuebles y una tabla de contratos. En la creación de la tabla de contratos se especifica mediante la cláusula `CONSTRAINT` que `inum` es una clave ajena (`FOREIGN KEY`) a la tabla de inmuebles y que la regla de comportamiento es “propagar el borrado” (`ON DELETE CASCADE`):

```
CREATE TABLE contratos
(
  cnum VARCHAR(5) NOT NULL,
  inum VARCHAR(4) NOT NULL,
  . . . ,
  CONSTRAINT contratos_pk PRIMARY KEY (cnum),
  CONSTRAINT contratos_fk_inum FOREIGN KEY (inum)
    REFERENCES inmuebles ON DELETE CASCADE
```

);

Nótese que a cada restricción (**CONSTRAINT**) se le da un nombre que servirá para registrarla en el diccionario de datos. A partir de este momento, siempre que se borre un inmueble que tenga contratos, se borrarán todos los contratos de dicho inmueble. Otra posibilidad es escoger “restringir el borrado”: de este modo sólo será posible borrar inmuebles que no tengan ningún contrato almacenado. Es siempre el propietario de la información quien debe indicar cuál es el comportamiento deseado.

2.4. *¿Qué es un SGBD?*

El SGBD es una aplicación que permite a los usuarios definir , crear y mantener la base de datos y proporciona un acceso controlado a la misma.

El acceso a los datos se realiza siempre a través de él. De esta forma:

- garantiza independencia lógica-física.
- disminuye la redundancia de datos.
- mantiene la integridad y consistencia de los datos.
- permite el acceso concurrente a los datos
- ...

todo ello de modo transparente a los usuarios y a las aplicaciones, es decir, estos no se han de preocupar de todos esos aspectos.

2.5. *En un SGBD, como por ejemplo, el de la empresa inmobiliaria ¿podrías explicar algún ejemplo de reglas o restricciones de integridad para algunos datos de dicha empresa?*

En el ejemplo de la pregunta anterior se está considerando una regla de integridad general a todas las bases de datos relacionales: la regla de integridad referencial (mantenida gracias a las claves ajenas). Otros ejemplos de reglas de integridad, esta vez de reglas específicas de la base de datos de la inmobiliaria, pueden ser: que el identificador del inmueble debe comenzar por la letra I, que cada inmueble debe tener al menos una habitación y que el tipo del inmueble sólo puede ser piso, casa o bajo. Para ello, en la sentencia de creación de la tabla inmueble especificaríamos estas reglas mediante restricciones **CHECK**:

```

CREATE TABLE inmuebles
(
    inum VARCHAR(4) NOT NULL,
    tipo VARCHAR(10) NOT NULL,
    hab NUMBER(2) NOT NULL,
    . . . ,
    CONSTRAINT inmuebles_pk PRIMARY KEY (inum),
    CONSTRAINT inmuebles_ck_inum CHECK (SUBSTR(inum,1,1)='I'),
    CONSTRAINT inmuebles_ck_tipo CHECK (tipo IN ('casa','piso','bajo')),
    CONSTRAINT inmuebles_ck_hab CHECK (hab >0)
) ;

```

2.6. *¿Cual ha sido la evolución de los SBD?*

Los SBD se pueden dividir en tres generaciones :

- La primera generación en la que se utilizaba los sistema jerárquico y más tarde los sistemas de red. Se basaban en definir los registros y las relaciones entre ellos utilizando punteros físicos. Requerían complejos programas de aplicación.
- En La segunda generación se utiliza el modelo relacional que utiliza tablas para representar los datos. Esta basado en un modelo teórico y ha sido el más utilizado en los últimos años.
- La tercera generación incluye los modelo orientados a objetos y modelos relacionales extendidos. Ambos utilizan las ventajas de la metodología orientada a objetos.

2.7. *¿Cuáles son los problemas de los SBD de la primera generación?*

Cabe destacar tres que son bastante importantes:

- Requieren complejos programas de aplicación, puesto que deben acceder a la estructura de punteros de la Base de Datos.
- La independenciam de datos es mínima, las aplicaciones dependen fuertemente de la estructura de la Base de Datos.
- No tienen un fundamento teórico que avale su comportamiento.

2.8. ¿Qué es el diseño lógico?

El diseño de una base de datos se realiza en tres fases: diseño conceptual, diseño lógico y diseño físico. En el diseño conceptual el objetivo es obtener un esquema (normalmente representado mediante el modelo entidad/relación) en el que aparezcan todas las entidades de interés, con sus propiedades y relaciones. En el diseño lógico se parte de este esquema conceptual para producir el esquema lógico de la base de datos. Si se está diseñando una base de datos relacional, el esquema lógico será un conjunto de tablas, cada una de ellas con una clave primaria y con claves ajenas que permiten relacionar la información almacenada en dichas tablas.

En el siguiente ejemplo se muestra una parte del esquema lógico (que se obtiene de la fase de diseño lógico) de la base de datos de la inmobiliaria. Concretamente se muestra la tabla que almacena los datos de los propietarios, cuya clave primaria es `pnum`, y la tabla que almacena los datos de los inmuebles, cuya clave primaria es `inum`. De la tabla de inmuebles hay una clave ajena a la tabla de propietarios y para ella se definen unas reglas de comportamiento: no acepta nulos, no se puede borrar un propietario que tiene algún inmueble y, si se modifica el valor de `pnum` en la tabla de los propietarios, el cambio se propagará a las filas correspondientes a sus inmuebles.

```
propietario(pnum, nombre, apellido, dirección, teléfono)
inmueble(inum, calle, área, población, tipo, hab, alquiler, pnum)
  inmueble.pnum es CLAVE AJENA a propietario
  nulos: no
  borrado: restringir
  modificación: propagar
```

En la fase del diseño físico se parte del esquema lógico y se produce el esquema físico de la base de datos: sentencias de creación de tablas con especificaciones sobre su almacenamiento, índices que se deben crear, etc. En el siguiente ejemplo se muestra parte de la sentencia que crea la tabla de los contratos. En ella se crea un índice sobre la clave primaria, se especifica el espacio físico (`TABLESPACE`) donde se almacenará el índice y donde se almacenará la tabla, y cómo se va a gestionar el espacio de almacenamiento (`STORAGE`) del índice y también de la tabla (espacio inicial que se toma y cómo va a ir creciendo este espacio):

```
CREATE TABLE contratos
(
  cnum VARCHAR(5) NOT NULL,
  inum VARCHAR(4) NOT NULL,
  ...,
  CONSTRAINT contratos_pk PRIMARY KEY (cnum)
  USING INDEX
  TABLESPACE users
```

```

        STORAGE (INITIAL 10K NEXT 10k PCTINCREASE 0),
        CONSTRAINT contratos_fk_inum FOREIGN KEY (inum)
        REFERENCES inmuebles ON DELETE CASCADES
    )
TABLESPACE users
STORAGE (INITIAL 100K NEXT 100k PCTINCREASE 0);

```

2.9. *Desde el punto de vista lógico, los sistemas de bases de datos almacenan las relaciones entre los datos ¿qué relaciones almacenan?*

Almacenan las relaciones lógicas entre las entidades de interés: cada inmueble “pertenece” a un propietario, cada contrato “es” de un inmueble y “es” de un cliente, cada empleado “pertenece” a una oficina, etc. Estas relaciones, que se dan entre las entidades del mundo real, se pueden representar en los sistemas de bases de datos. Por ejemplo, en una base de datos relacional se representan mediante las claves ajenas (`inmueble.pnum` es una clave ajena que relaciona a cada inmueble con su propietario).

2.10. *¿Por qué la existencia del diccionario de datos proporciona independencia de datos lógico-física?*

En los sistemas de ficheros cada programa de aplicación que accede a un fichero debe conocer de antemano cuál es la estructura de éste: tipo y longitud de sus registros, tipo y longitud de cada uno de los campos, delimitadores de registros, delimitadores de campos, etc. Sin embargo, en los sistemas de bases de datos las definiciones de los datos (tipo, descripción, restricciones) se encuentran en una base de datos denominada “diccionario de datos”. Los programas de aplicación no necesitan conocer la estructura física de los datos y cualquier tipo de información sobre ellos la tienen accesible mediante el diccionario de datos. Por ejemplo, en Oracle podemos escribir un programa en PL/SQL (una extensión procedural de SQL) en donde se permite declarar variables del siguiente modo:

```

alq_inm inmueble.alquiler%TYPE;
fila_inm inmueble%ROWTYPE;

```

La variable escalar `alq_inm` se declara con el mismo tipo que la columna `alquiler` de la tabla `inmueble`. La variable tupla (registro) `fila_inm` se declara del mismo tipo que las filas de la tabla `inmueble`. La información sobre estos tipos se encuentra en el diccionario de datos. Si, por cualquier motivo, el tipo de los datos cambia en el esquema físico de la base de datos (por ejemplo, para pasar el alquiler de pesetas a euros, se pasa de un número entero a un número real con dos decimales), este cambio es transparente para las aplicaciones que han declarado sus variables de este modo (siempre que las operaciones y funciones que se apliquen sobre los datos sigan siendo coherentes con el nuevo tipo).

2.11. *¿Qué es la visión lógica global desde el punto de vista de los diseñadores de BD y del administrador de la BD?*

En los sistemas de bases de datos, los datos se encuentran todos en un mismo lugar, la base de datos, pudiendo ser compartidos por todos los usuarios. Ya que cada usuario no necesita, ni tiene porqué ver, el esquema completo de la base de datos (por ejemplo, el departamento de personal no necesita ver ninguna información sobre los clientes o los inmuebles), los sistemas de bases de datos permiten definir subesquemas de la base de datos, de modo que cada usuario pueda ver la parte de la base de datos que le interesa.

Los diseñadores de la BD y el administrador de la BD son quienes tienen la visión lógica global de la base de datos: tienen a su alcance el esquema lógico completo. En el caso de una base de datos relacional, pueden ver todas las tablas y sus relaciones. Los programadores de aplicaciones y los usuarios finales (usuarios con conocimientos de bases de datos), verán sólo aquella parte de la base de datos que les concierne: su subesquema.

2.12. *¿Qué es la concurrencia?*

Al hablar de concurrencia nos referimos a que se posibilita el acceso simultáneo a los datos por parte de distintos usuarios y de distintas aplicaciones. El que el SGBD proporcione un sistema de control de concurrencia nos asegura que nunca se producirán faltas de integridad a causa del acceso concurrente a la base de datos. Normalmente, los sistemas relacionales implementan el control de la concurrencia mediante protocolos basados en bloqueos.

2.13. *¿Qué es una vista?*

Es la visión lógica que tienen los distintos usuarios de la Base de Datos, es decir, un subconjunto de la Base de datos.

Aunque la Base de Datos contiene todos los datos, las vistas permiten definir a qué parte de la misma permitimos el acceso a cada uno de los usuarios, por motivos de seguridad o por necesidades del propio usuario.

2.14. *¿Cuál de las siguientes afirmaciones es la correcta?*

1. Los sistemas de ficheros pueden guardar información redundante y los sistemas de base de datos no.
2. Los sistemas de ficheros no pueden almacenar reglas de integridad mientras que los sistemas de base de datos si pueden hacerlo.

3. En los sistemas de ficheros existe independencia de datos (lógico-física) mientras que en los sistemas de base de datos no existe.

La 2 es la correcta.

La 1 no lo es puesto que tanto en los sistemas de ficheros como los SBD se pueden almacenar datos redundantes, aunque en estos últimos se pueden reducir bastantes y controlar más fácilmente.

La 3 es incorrecta ya que la dependencia de datos (lógico-física), es decir, la dependencia de los programas respecto a los datos, es un problema de los sistemas de ficheros que se solventa en parte en los SBD.

3. Preguntas sin respuesta

Alguien, que se hace llamar Alumno Aplicado, ha tenido a bien mandarnos una serie de preguntas que no tienen que ver mucho sobre este tema y para las que nosotros tampoco hemos encontrado respuesta. Ponemos aquí alguna de ellas.

¿Por qué venden tabaco en las gasolineras, si está prohibido fumar? Adán y Eva ¿tenían ombligos? Si nada se pega al teflón ¿cómo lo pegaron a la sartén? Si Superman es tan listo entonces ¿por qué lleva los calzoncillos por fuera? Si una palabra estuviese mal escrita en el diccionario ¿cómo lo sabríamos? ¿Por qué Bill Gates llamó a su sistema operativo Windows (“ventanas” en inglés) si lo podría haber llamado Gates (“puertas” en inglés)? ¿Por qué “separado” se escribe todo junto y “todo junto” se escribe separado? ¿Por qué apretamos más fuerte los botones del control remoto cuando tiene poca batería? ¿Por qué corremos rápido bajo la lluvia, si delante también llueve? ¿Por qué cuando llueve levantamos los hombros? ¿Acaso nos mojamos menos? Según las estadísticas, una persona es atropellada por un automóvil cada cinco minutos ¿cómo hace esa persona para sobrevivir a cada uno de esos atropellos? Si hay un más allá ... ¿hay un menos acá? ¿Qué cuentan las ovejas para poder dormir? ¿Por qué las ciruelas negras son rojas cuando están verdes? Si el dinero es la causa de todos los males ¿entonces, por qué tenemos que trabajar? ¿Dónde está la otra mitad del Medio Oriente? ¿A qué árbol pertenece el fruto del trabajo? ¿Qué hay que hacer si uno ve un animal en peligro de extinción comiendo una planta en peligro de extinción? ¿Por qué será que cuando uno habla con Dios, la gente piensa que eres espiritual, pero si Dios habla con uno, la gente piensa que estás loco de remate? ¿Por qué lavamos las toallas? ¿no se supone que estamos limpios cuando las usamos? ¿Cómo puedo saber cuántas vidas le quedan a mi gato? Una vida más larga ... ¿acortaría la muerte? ¿Existe alguna otra palabra para “sinónimo”? Si estás en un callejón sin salida ¿por qué no salir por la entrada? ¿Por qué hay día del padre, día de la madre, pero no hay día del hijo? Si quiero comprar un boomerang nuevo, ¿cómo hago para deshacerme del viejo? Si cárcel y prisión son sinónimos ¿por qué no lo son carcelero y prisionero? La caja negra de los aviones es indestructible ... ¿por qué no

harán todo el avión de ese mismo material? ¿Por qué el fútbol americano se llama así, si no se usa el pie? ¿Por qué la palabra “abreviatura” es tan larga?