

# Conceptos de Bases de Datos

Agustin Murray

March 12, 2024

# Contents

<b>1</b>	<b>Conceptos generales de bases de datos</b>	<b>3</b>
1.1	Conceptos Basicos . . . . .	3
1.2	Origenes de las BD . . . . .	3
1.3	Gestores de Bases de Datos . . . . .	3
<b>2</b>	<b>Archivos, estructuras y operaciones basicas</b>	<b>5</b>
2.1	Definicion . . . . .	5
2.2	Aspectos fisicos . . . . .	5
2.3	Niveles de vision . . . . .	5
2.4	Organizacion interna de los datos . . . . .	6
2.5	Acceso a informacion contenida en los archivos . . . . .	6
2.6	Operaciones basicas sobre archivos . . . . .	6
2.6.1	Definicion de archivos . . . . .	6
2.6.2	Correspondencia archivo logico - archivo fisico . . . . .	7
2.6.3	Apertura y creacion de archivos . . . . .	7
2.6.4	Cierre de archivos . . . . .	7
2.6.5	Lectura y escritura de archivos . . . . .	7
2.6.6	Operaciones adicionales con archivos . . . . .	8

# 1 Conceptos generales de bases de datos

## 1.1 Conceptos Basicos

- Se considera **Base de Datos**, a una coleccion o conjunto de datos interrelacionados con un proposito especifico vinculado a la resolucion de un problema del mundo real.
- Cualquier informacion dispuesta de manera adecuada para su tratamiento por una computadora.
- Una coleccion de archivos diseñados para servir a multiples aplicaciones.

## 1.2 Origenes de las BD

En los origenes de las **BD**, las unidades de almacenamiento de gran volumen eran muy lentas, entonces se buscaba reducir el acceso a los mismos. Para esto, era necesario repetir datos en distintos puestos de trabajo y al final del dia actualizar los mismos en todas las unidades para evitar problemas de perdida/modificacion de informacion.

Con el tiempo, la tecnologia fue avanzando y los sistemas de informacion evolucionaron. Se logra integrar aplicaciones, interrelacionar archivos y eliminar la redundancia de datos.

## 1.3 Gestores de Bases de Datos

Un **Sistema de Gestion de Bases de Datos (SGBD)** consiste en un conjunto de programas necesarios para acceder ya administrar una BD.

Actualmente, cualquier sistema de software necesita interactuar con informacion almacenada en una BD y para ello requiere del soporte de un SGBD.

Un SGBD posee dos tipos diferentes de lenguajes: uno para especificar el esquema de una BD, y el otro para la manipulacion de los datos.

La definicion del esquema de una BD implica:

- Diseño de la estructura que tendra efectivamente la BD.
- Describir los datos, la semantica asociada y las restricciones de consistencia.

Para ello se utiliza un lenguaje especial, llamado **Lenguaje de Definicion de datos (LDD)**. El resultado de compilar lo escrito con el LDD es un archivo llamado **Diccionario de Datos**. Un **Diccionario de Datos** es un archivo con metadatos, es decir, datos acerca de los datos.

Los objetivos mas relevantes de un SGBD son:

- **controlar la concurrencia:** varios usuarios pueden acceder a la misma informacion en un mismo periodo de tiempo. Si el acceso es para consulta, no hay inconvenientes, pero si mas de un usuario quiere actualizar el mismo dato a la vez, se puede llegar a un estado de inconsistencia que, con la supervision del SGBD, se puede evitar.
- **Tener control centralizado:** tanto de los datos como de los programa que acceden a los datos.

- **Facilitar el acceso a los datos:** dado que provee un lenguaje de consulta para recuperacion rapida de informacion.
- **Proveer seguridad para imponer restricciones de acceso:** se debe definir explicitamente quienes son los usuarios autorizados a acceder a la BD.
- **Mantener la integridad de los datos:** esto implica que los datos incluidos en la BD respeten las condiciones establecidas al definir la estructura de la BD y que, ante una falla del sistema, se posea la capacidad de restauracion a la situacion previa.

## 2 Archivos, estructuras y operaciones basicas

### 2.1 Definicion

Un **archivo** es una coleccion de registros semejantes, guardados en dispositivos de almacenamiento secundario de una computadora.

Los archivos se caracterizan por el crecimiento y las modificaciones que se efectuan sobre estos. El crecimiento indica la incorporacion de nuevos elementos, y las modificaciones involucran alterar datos contenidos en el archivo, o quitarlos.

### 2.2 Aspectos fisicos

**Almacenamiento primario (RAM):**

- Capacidad de almacenamiento limitada.
- Volatil.
- Alto costo.
- Acceso rapido (orden de los nanosegundos)

**Almacenamiento secundario (cintas y discos):**

- Alta capacidad de almacenamiento.
- No volatil.
- Menor costo que el primario.
- Acceso lento a comparacion del primario (orden de los milisegundos). Se busca optimizar este aspecto:
  1. Búsqueda de un unico dato: Obtencion en un intento o en pocos.
  2. Búsqueda de varios datos: obtencion de todos de una sola vez.
- **Cintas:** acceso secuencial, economico, estable en diferentes condiciones ambientales, facil de transportar.
- **Discos:** acceso directo, se almacenan datos en sectores.

### 2.3 Niveles de vision

- **Archivo fisico:** es el archivo residente en la memoria secundaria y es administrado (ubicacion, tipo de operaciones disponibles) por el sistema operativo.
- **Archivo logico:** es el archivo utilizado desde el algoritmo. Cuando el algoritmo necesita operar con un archivo, genera una conexion con el sistema operativo, el cual sera el responsable de la administracion. Esta accion se denomina independencia fisica.

## 2.4 Organizacion interna de los datos

- **Secuencia de bytes:** Se determina como unidad mas pequeña de L/E al byte. No se puede determinar facilmente el comienzo y el final de cada dato por lo que generalmente son archivos de texto.
- **Campos:** Se determina como unidad mas pequeña de L/E al campo. El campo es un item de datos elemental y se caracteriza por su tipo de dato y su tamaño.
- **Registros:** Se determina como unidad mas pequeña de L/E al registro. El registro es un conjunto de campos agrupados que definen un elemento del archivo. Los campos internos a un registro deben estar logicamente relacionados, como para ser tratados como una unidad.

## 2.5 Acceso a informacion contenida en los archivos

Basicamente, se pueden definir tres formas de acceder a los datos de un archivo:

- **Secuencial:** el acceso a cada elemento de datos se realiza luego de haber accedido a su inmediato anterior. El recorrido es, entonces, desde el primero hasta el ultimo de los elementos, siguiendo el orden fisico de estos.
- **Secuencial indizado:** el acceso a los elementos de un archivo se realiza teniendo presente algun tipo de organizacion previa, sin tener en cuenta el orden fisico.
- **Directo:** es posible recuperar un elemento de dato de un archivo con un solo acceso, conociendo sus características, mas alla de que exista un orden fisico o logico predeterminado.

## 2.6 Operaciones basicas sobre archivos

Para poder operar con archivos, son necesarias una serie de operaciones elementales disponibles en todos los lenguajes de programacion que utilicen archivos de datos. Estas operaciones incluyen:

- La definicion del archivo logico.
- La definicion de la forma de trabajo del archivo (creacion inicial, utilizacion).
- La administracion de datos (L/E info).

### 2.6.1 Definicion de archivos

Como cualquier otro tipo de datos, los archivos necesitan ser definidos. Se reserva la palabra clave **file** para indicar la definicion del archivo.

```
Var archivo_logico: file of tipo_de_dato;
```

Otra opcion para definir archivos se presenta a continuacion:

```
Type archivo = file of tipo_de_dato;  
Var archivo_logico: archivo;
```

### 2.6.2 Correspondencia archivo logico - archivo fisico

Se debe indicar que el archivo logico utilizado por el algoritmo se corresponde con el archivo fisico adminstrado por el sistema operativo. La sentencia encargada de hacer esta correspondencia es:

```
Assign (nombre_logico , nombre_fisico );
```

### 2.6.3 Apertura y creacion de archivos

Hasta el momento, se ha detallado como definir un archivo y se ha esctablecido la relacion con el nombre fisico. Para operar con un archivos desde un algoritmo, se debe realizar la apertura.

- La operacin **rewrite** indica que el archivo va a ser creado y, por lo tanto, la unica operacion valida sobre el mismo es escribir informacion.
- La operacino **reset** indica que el archivo ya existe y, por lo tanto, las operaciones validas sobre el mismo son L/E de informacion.

```
rewrite(nombre_logico );  
reset(nombre_logico );
```

### 2.6.4 Cierre de archivos

Para mantener valida la marca de end of file (EOF) se cierra el archivo de la siguiente forma:

```
close (nombre_logico );
```

### 2.6.5 Lectura y escritura de archivos

Para leer o escribir informacion en un archivo, las instrucciones son:

```
read(nombre_logico , var_dato );  
write(nombre_logico , var_dato );
```

**Buffers de memoria:** Las lecturas y escrituras desde o hacia un archivo se realizan sobre buffers. Se denomina buffer a una memoria intermedia (ubicada en RAM) entre un archivo y un programa, donde los datos residen provisoriamente hasta ser almacenados definitivamente en la memoria secundaria, o dodnde los datos residen una vez recuperados de dicha memoria secundaria. La razon de esto es la mejora de performance al trabajar con mayor frecuencia en memoria principal.

La operacion read lee desde un buffer y, en caso de no contar con informacion, el sistema operativo realiza automaticamente una operacion input, trayendo mas informacion al buffer. La diferencia radica en que cada operacion input transfiere desde el disco una serie de registros. De esta forma, cada determinada cantidad de instrucciones read, se realiza una operacion input.

De forma similar procede la operacion write; en este caso, se escribe en el buffer, y si no se cuenta con espacio suficiente, se descarga el buffer a disco por medio de una operacion output, dejandolo nuevamente vacio.

### 2.6.6 Operaciones adicionales con archivos

- Control fin de datos:

`eof(nombre_logico);`

La funcion retornara verdadero si el puntero del archivo referencia e EOF, y falso en caso contrario.

- Control de tamaño del archivo:

`filesize(nombre_logico);`

- Control de posicion de trabajo dentro del archivo:

`Filepos(nombre_logico);`

- Ubicacion fisica en alguna posicion del archivo:

`seek(nombre_logico , posicion);`

- Truncamiento de un archivo a partir de la posicion acutal:

`truncate(nombre_logico);`