Ficheros y bases de datos

Álvaro González Sotillo

9 de septiembre de 2023

Índice

1. Introducción

- Se manejan grandes cantidades de datos desde hace mucho tiempo
  - Censos romanos
  - Bancos medievales
  - Información fiscal de cada país
  - Empresas de todo tipo
- Tradicionalmente, se han usado
  - Fichas, informes, expedientes archivadores, carpetas...

1.1. Antes de la informática

- Tradicionalmente
  - Manejados por personas
  - De forma manual
  - Gran componente subjetivo
- Algunos sistemas intentan eliminar el componente subjetivo
  - Sistemas burocráticos

1.2. Informática

- Tratamiento automatizado de la información
- Se elimina el componente subjetivo
- Las operaciones con los datos se vuelven
  - Precisas
  - Rápidas
- Permite un mayor volumen de datos

2. Discos de datos

- Originalmente, los programas de ordenador utilizaban directamente los soportes de memoria (cinta, disco)
  - Ventaja: No se depende de otros sistemas
  - Pero...
- Un programa $\Leftrightarrow$ Un disco de datos
  - Un cambio de datos hacía inútil el programa
  - Un cambio de programa hacía inútiles los datos anteriores
- Cada programa debe aprender a manejar los discos

3. Ficheros (archivos)

- El sistema operativo crea archivos
- Los programas se simplifican
- Los programas pueden compartir los discos
- Más de un programa puede usar los mismos ficheros de datos
  - Es necesaria una coordinación para acceder y modificar ficheros

3.1. ¿Qué es un archivo?

- Un archivo se compone de registros
  - Un registro son los datos agrupados de alguna entidad
- Un registro contiene campos de datos
- Cada campo tiene un nombre y un valor
  - Por simplicidad, supondremos que todos los registros tienen los mismos campos

3.2. Ejemplo de archivo

Identificador	Nombre	Deuda	Dirección
987	juan	87345	10 norte 342
876	pedro	43649	8 oriente 342
123	jorge	03342	av. libertad 23
69	vicente	61560	valencia nº183
18	lorenzo	06490	sol nº18
19	lucía	06480	luna nº8

3.3. Nombres de los campos

Identificador	**Nombre**	**Deuda**	**Dirección**
987	juan	87345	10 norte 342
876	pedro	43649	8 oriente 342
123	jorge	03342	av. libertad 23
69	vicente	61560	valencia nº183
18	lorenzo	06490	sol nº18
19	lucía	06480	luna nº8

3.4. Un registro

Identificador	Nombre	Deuda	Dirección
987	juan	87345	10 norte 342
876	pedro	43649	8 oriente 342
123	**jorge**	**03342**	**av. libertad 23**
69	vicente	61560	valencia nº183
18	lorenzo	06490	sol nº18
19	lucía	06480	luna nº8

3.5. Una columna

Identificador	**Nombre**	Deuda	Dirección
987	**juan**	87345	10 norte 342
876	**pedro**	43649	8 oriente 342
123	**jorge**	03342	av. libertad 23
69	**vicente**	61560	valencia nº183
18	**lorenzo**	06490	sol nº18
19	**lucía**	06480	luna nº8

4. Tipos de archivos

- Según su uso
- Según formato
- Según su organización

4.1. Tipos según su uso

- Permanentes
  - Datos que deben ser guardados
  - Ejemplo: Empleados contratados, nóminas pagadas, declaraciones de impuestos,...
- De movimiento
  - Cambios que deben ser incluidos en archivos permanentes
  - Ejemplo: un puesto de peaje debe guardar todos los pagos con tarjeta, y enviarlos juntos
- De maniobra
  - Se utilizan como extensión a la RAM de un ordenador, se borran cuando el proceso termina
  - Ejemplo: caché de disco de los navegadores

4.2. Según formato

- De texto (o planos, o ASCII, o UNICODE)
  - Pueden editarse con el bloc de notas
  - Son teóricamente legibles directamente por las personas
- Binarios
  - La información se guarda en un formato numérico (binario), no legible directamente

4.2.1. Ficheros binarios

- `exe`, `dll` : Ficheros ejecutables
- `png`, `jpg`, `gif` : Ficheros de imagen
- `zip`, `rar` : Ficheros comprimidos
- `docx`, `pptx`, `xlsx`, `pdf` : Documentos ofimáticos

4.2.2. Ficheros de texto

- `txt`: Texto
- `html`, `rtf`, `ps`: Texto con formato
- `ini`, `inf`, `conf`, `xml`: configuración de programas
- `sql`, `java`, `php`, `c`, `bat`, `sh`: instrucciones de programas informáticos

Variantes:

- Encodig: ASCII, UNICODE (utf-8, utf-16, utf-32), ISO-8859,...
- Fin de línea: Unix, Windows

4.2.3. Texto ¿plano?

- No es fácil/posible deducir en qué variante está guardado un fichero con texto plano
- Los programas utilizan
  - Heurísticas: pruebas en las primeras líneas del fichero
  - BOM

4.2.4. Ficheros de texto como binarios

- Al final, todos los ficheros son solo **números** almacenados en disco
  - Los programas o personas *interpretan* los números
- Un fichero de texto es en el fondo un fichero binario
- La traducción a "humano" es el estándar ASCII (o UNICODE), que asigna a cada byte una letra

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

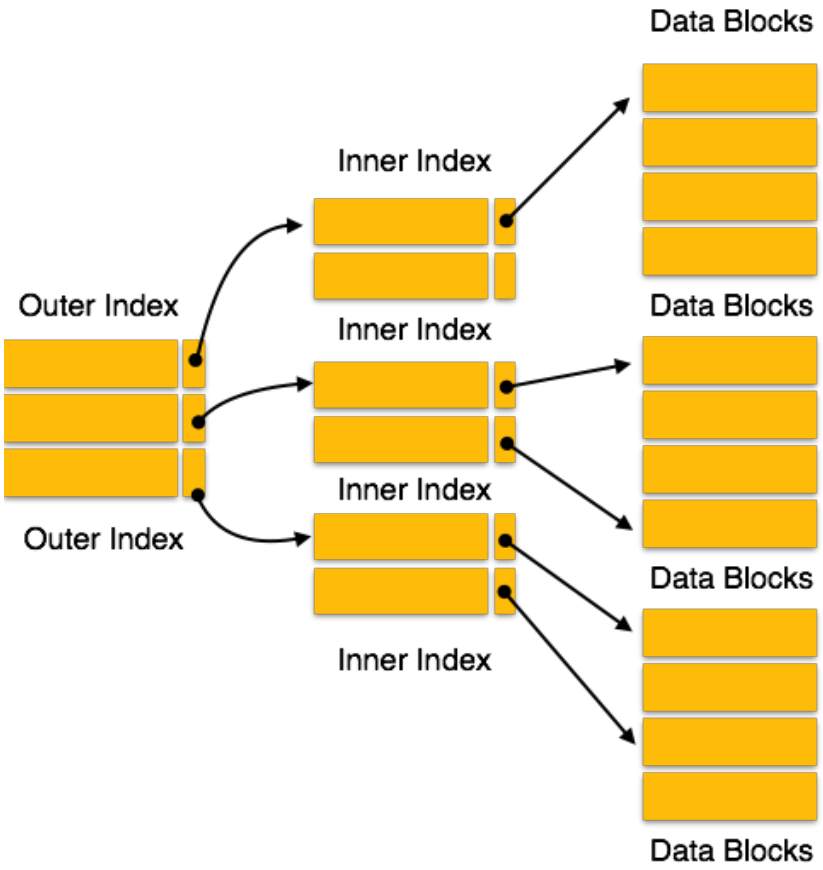
4.3. Tipos de ficheros según organización

- Organización secuencial
  - Los registros se colocan unos detrás de otros
  - Pueden estar ordenados por algún criterio
    - Orden de llegada
    - Alfabético por algún campo
- Organización indexada
  - Cada fichero secuencial puede tener otros ficheros de índice
  - El índice está ordenado por algún criterio
  - En el índice aparece
    - Identificador de cada registro
    - En qué línea (posición) está ese registro

`fichero-indexado.gif`
Crédito: www.dlsweb.rmit.edu.au

4.3.1. Ficheros indexados

- El fichero secuencial con datos es el fichero principal
- Cada fichero principal puede tener otros ficheros de índice
  - Uno por cada criterio que se desee buscar rápidamente
- Cada fichero de índice es a su vez un fichero secuencial
  - Podría indexarse, con un índice de segundo nivel



Créditos: www.tutorialspoint.com

4.3.2. Área de desbordamiento (*overflow*)

- Los criterios de un índice pueden no ser únicos
  - Por ejemplo, código postal en un fichero de alumnos
- Si hay un *conflicto*, los datos se almacenan en un área de *overflow*

Créditos: kpvxy.blogspot.com.es

4.4. Secuencial vs Indexado (escritura)

- Organización secuencial:
  - Si no se ordena, basta con añadir: rápido
  - Si se ordena, se puede necesitar cambiar todo el fichero: *muy* lento
- Organización indexada:
  - Si no hay colisiones, dos escrituras (índice y fichero principal)
  - Si hay colisiones (la clave ya está usada)
    - Usar un fichero de *overflow* (y reorganizar con el fichero principal en un futuro)
    - Reorganizar el fichero principal *muy* lento
- Para lectura, ver acceso vs organización

5. Acceso a ficheros

- Acceso secuencial
  - Para llegar a un registro, es necesario pasar por todos los anteriores
  - *Obligatorio* en
    - cintas
    - ficheros sin indexar con campos de longitud variable (csv, xml,...)
- Acceso directo (aleatorio)
  - Se puede leer directamente un registro sin tener que pasar por los anteriores
  - Se necesita saber su posición (por un índice)

5.1. Acceso vs organización (lectura)

	Acceso secuencial	Acceso directo
Organización secuencial	Fácil y rápido	Deben leerse los registros anteriores, o estar ordenado
Organización indexada	Algo más lento (dos lecturas mínimo)	Más rápido (dos lecturas)

6. Bases de datos

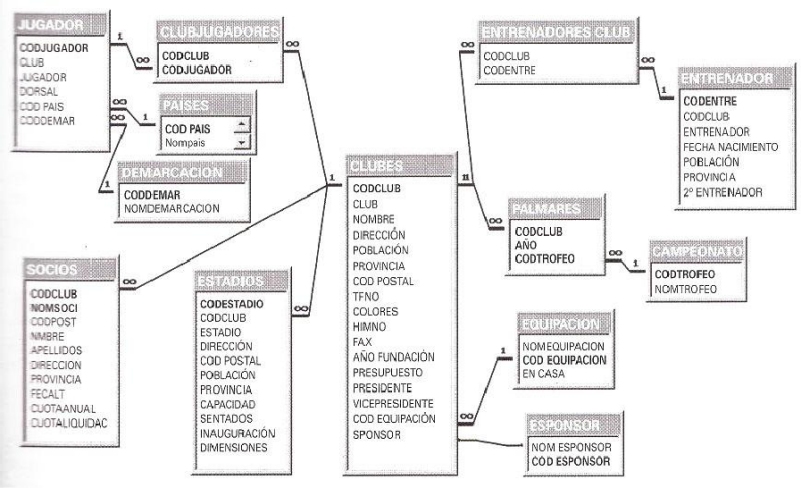
- En una empresa, los datos pueden estar dispersos y duplicados
- Hay que actualizar todas las copias a la vez
  - centralización de los datos
- Puede haber datos confidenciales
  - permisos por fichero
- Se puede necesitar más de un programa accediendo a los mismos registros
- Pero no a los mismos campos
  - permisos por campo,
- Diferentes departamentos pueden tener nombres distintos para los ficheros, o los campos
  - diferentes formas de ver los registros

6.1. Definición (I)

Una colección de datos que están lógicamente relacionados entre sí, que tiene una definición y una descripción comunes y que están estructurados de una forma particular

6.2. Definición (II)

Una base de datos es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ésta, y su definición y descripción han de ser únicas estando almacenados junto a los mismos. Por último, los tratamientos que sufran estos datos tendrán que conservar la integridad y seguridad de éstos



6.3. Ventajas de las bases de datos

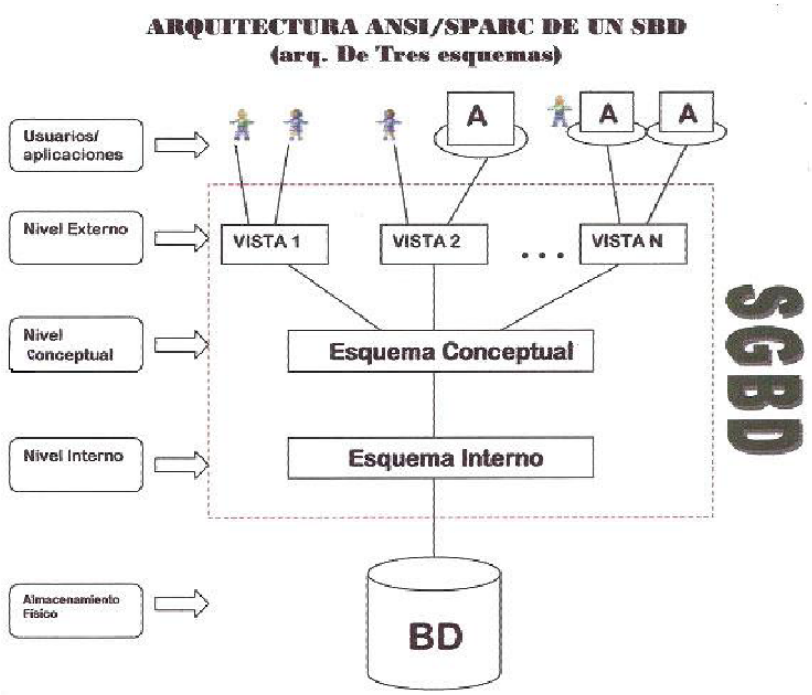
- **Independencia de los datos y los programas y procesos**. Esto permite modificar los datos sin modificar el código de las aplicaciones.
- **Menor redundancia**. Aunque, sólo los buenos diseños de datos tienen poca redundancia.
- **Integridad**. Mayor dificultad de perder los datos o de realizar incoherencias con ellos.
- **Mayor seguridad**. Al limitar el acceso a ciertos usuarios.
- **Datos más documentados**. Gracias a los metadatos que permiten describir la información de la base de datos.
- **Acceso a los datos más eficiente**. La organización de los datos produce un resultado más óptimo en rendimiento.

6.4. Inconvenientes

- Instalación costosa
  - El control y administración de bases de datos requiere de un software y hardware poderoso
- Requiere personal cualificado
  - Debido a la dificultad de manejo de este tipo de sistemas.
- De todas formas, **las ventajas superan ampliamente los inconvenientes**

7. Estándar ANSI/SPARC

- Define tres niveles, para ayudar a conseguir los objetivos de un SGBD
  - **Interno**: es como se almacena la información realmente. Por lo general, en ficheros en disco
  - **Conceptual**: incluye la estructura de la base de datos total
    - Entidades
    - Campos de las entidades
    - Relaciones entre entidades
  - **Externo**: Cada tipo de usuario/aplicación puede operar con una parte del nivel conceptual, a veces con una transformación intermedia



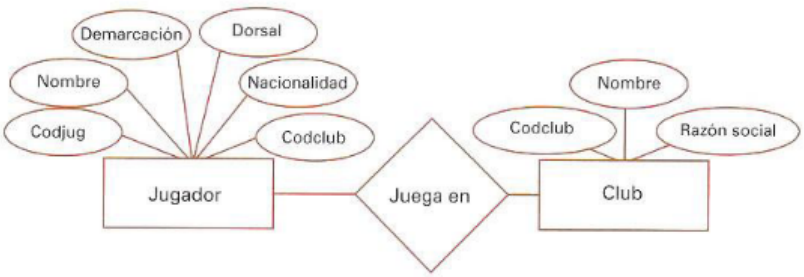
8. Diseño de bases de datos

- No es evidente abstraer, a partir de datos en bruto, la estructura de una base de datos
- Las bases de datos se diseñan en tres pasos
  - Nivel conceptual
  - Nivel lógico
  - Nivel físico

Nota: estos niveles son del **diseño**, no confundir con los niveles de la implementación Ansi/SPARC

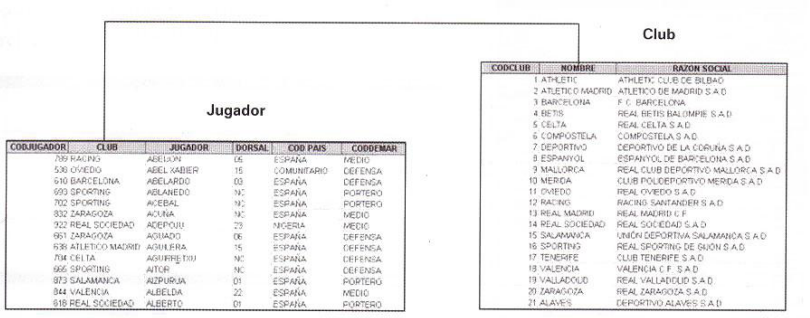
8.1. Nivel conceptual

- Un usuario no informático debe poder entenderlo
- Trata sobre
  - entidades
  - relaciones entre ellas
  - datos a almacenar por cada entidad y relación



8.2. Nivel lógico

- El modelo conceptual debe ser sistematizado y simplificado, para que un ordenador pueda manejarlo
- No se decide cómo se guardarán los datos, pero sí qué forma tendrán
  - Generalmente, en forma de tabla



CODJUGADOR	CLUB	JUGADOR	DORSAL	COD PAIS	CODDEMAR
789	RACING	ABELLON	05	ESPAÑA	MEDIO
530	OVIEDO	ABEL XABIER	15	COMUNITARIO	DEFENSA
610	BARCELONA	ABELARDO	03	ESPAÑA	DEFENSA
699	SPORTING	ABLANEDO	NC	ESPAÑA	PORTERO
702	SPORTING	ACEBAL	NC	ESPAÑA	PORTERO
932	ZARAGOZA	ACUÑA	NC	ESPAÑA	MEDIO
922	REAL SOCIEDAD	ADEPOJU	23	NIGERIA	MEDIO
661	ZARAGOZA	AGUADO	06	ESPAÑA	DEFENSA
638	ATLETICO MADRID	AGUILERA	15	ESPAÑA	DEFENSA
704	CELTA	AGUIRRETXU	NC	ESPAÑA	DEFENSA
666	SPORTING	AITOR	NC	ESPAÑA	DEFENSA
873	SALAMANCA	AIZPURUA	01	ESPAÑA	PORTERO
844	VALENCIA	ALBELDA	22	ESPAÑA	MEDIO
618	REAL SOCIEDAD	ALBERTO	01	ESPAÑA	PORTERO

CODCLUB	NOMBRE	RAZON SOCIAL
1	ATHLETIC	ATHLETIC CLUB DE BILBAO
2	ATLETICO MADRID	ATLETICO DE MADRID S.A.D.
3	BARCELONA	F.C. BARCELONA
4	BETIS	REAL BETIS BALOMPIE S.A.D.
5	CELTA	REAL CELTA S.A.D.
6	COMPOSTELA	COMPOSTELA S.A.D.
7	DEPORTIVO	DEPORTIVO DE LA CORUÑA S.A.D.
8	ESPANYOL	ESPANYOL DE BARCELONA S.A.D.
9	MALLORCA	REAL CLUB DEPORTIVO MALLORCA S.A.D.
10	MERIDA	CLUB POLIDEPORTIVO MERIDA S.A.D.
11	OVIEDO	REAL OVIEDO S.A.D.
12	RACING	RACING SANTANDER S.A.D.
13	REAL MADRID	REAL MADRID C.F.
14	REAL SOCIEDAD	REAL SOCIEDAD S.A.D.
15	SALAMANCA	UNIÓN DEPORTIVA SALAMANCA S.A.D.
16	SPORTING	REAL SPORTING DE GIJÓN S.A.D.
17	TENERIFE	CLUB TENERIFE S.A.D.
18	VALENCIA	VALENCIA C.F. S.A.D.
19	VALLADOLID	REAL VALLADOLID S.A.D.
20	ZARAGOZA	REAL ZARAGOZA S.A.D.
21	ALAVES	DEPORTIVO ALAVES S.A.D.

8.3. Nivel físico

- Se describe de qué forma el nivel lógico será almacenado en ficheros
  - CSV
  - Excel
  - XML
  - Utilizando un Sistema Gestor de Bases de Datos

9. SGBD

9.1. SGBD: Componentes

- Hardware: Servidores, discos, componentes de red,...
- Software: Incluye un software de base de datos y las aplicaciones que los manejan
- Datos: Tanto los datos originales como los metadatos

9.2. SGBD: Funciones

- Almacenar datos en la base de datos, acceder a ellos y actualizarlos
- Mantener descripciones de los datos accesibles por los usuarios (metadatos)
- Integridad: una transacción debe realizarse en su totalidad o no realizarse
- Integridad: los cambios deben poder ser realizados por varios usuarios a la vez
- Integridad: Se deben poder recuperar los datos si se pierden (backup)
- Integridad y confidencialidad: sólo usuarios autorizados pueden ver/modificar datos
- Integridad: sólo los datos que sigan el diseño lógico pueden ser almacenados
- Comunicación: Datos y operaciones están disponibles para usuarios y aplicaciones

9.3. SGBD: Objetivos

- Independencia física de datos
  - Un programa debería poder seguir funcionando aunque el diseño físico (cómo se almacenan los datos en disco) cambie
  - Basta con que el SGBD ofrezca sólo un nivel conceptual que pueda usar diferentes niveles físicos
- Independencia lógica de datos
  - Un programa debería poder seguir funcionando aunque el diseño lógico (cómo se relacionan los datos) cambie
  - Es más difícil, pero teóricamente son suficientes las vistas (niveles externos)
- Estos objetivos se ven facilitados por los niveles definidos en la arquitectura ANSI-SPARC

10. Referencias

- Formatos:
  - Transparencias
  - PDF
  - EPUB
- Creado con:
  - Emacs
  - org-re-reveal
  - Latex
- Alojado en Github