Sistemas de gestión
de bases de datos
tema 6

ANSI/X3/SPARC

SGBD

Optimización

Conclusión

ANSI/SPARC

- ANSI/X3/SPARC Study Group on Data Base Management Systems: (1975)
  - *Interim Report. FDT*, ACM SIGMOD bulletin. Volume 7, No. 2
  - ANSI
    - American National Science Institute
  - X3
    - Accredited Standards Committee on Information Processing Systems
  - SPARC
    - Standards Planning and Requirements Commitee

Un comité de la organización ANSI (American National Standards Institute) aborda la problemática del almacenamiento de datos para su procesamiento en aplicaciones informáticas en 1975.

ANSI/SPARC: antes

- Problemas detectados con los sistemas de archivo convencional
  - redundancia de datos
    - “Este señor puede que se llame Jonathan del Amor o Vanesssa, o vete a saber”
  - dependencia de los programas respecto de los datos
    - ...que no usan
  - insuficientes medidas de seguridad
    - concurrencia, recuperación, permisos...

Como ya se comentó en el primer tema, los sistemas de archivo convencional, por aquel entonces, mostraban ciertas carencias que desembocaron en la definición de la "crisis crónica del software".

ANSI/SPARC: antes

- Consecuencias
  - pobre control de los datos
  - capacidades de manipulación de los datos limitadas o inadecuadas
  - excesivo esfuerzo de programación

Estas carencias se pueden resumir, mucho, en estos aspectos.

ANSI/SPARC: la solución

- ¡**SGBD**!
  - Descripción centralizada de los datos
  - Vistas parciales

El principal cambio en la forma de implementar sistemas de información mecanizados consiste en la centralización de los datos que permite un mayor control sobre los mismos, al tiempo que es necesario permitir vistas parciales puesto que aplicaciones de muy distinta naturaleza necesitan sus propios datos, y todas acceden a esa descripción centralizada. Estamos creando un interfaz entre las aplicaciones y el sistema operativo. Al tiempo, estamos mejorando la seguridad de los datos, tendremos herramientas de análisis y presentación de los datos, etc.

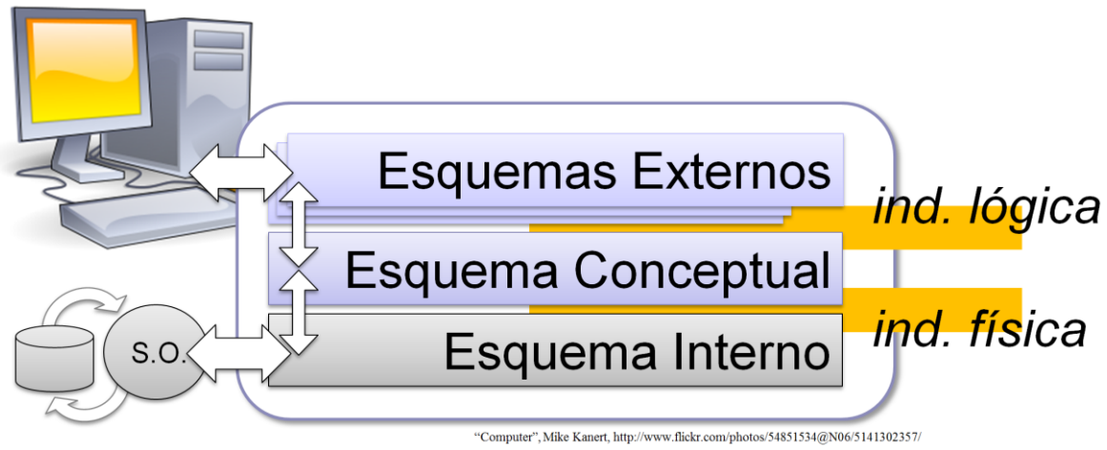
ANSI/SPARC: objetivos

- Integridad
  - los datos deben ser correctos
- Seguridad
  - acceso restringido a quien y como esté autorizado
- Independencia
  - modificar la definición de un dato **NO** implica reprogramar / recompilar una aplicación que **NO** lo usa

Al centralizar los datos (y estructurarlos según los distintos niveles de representación, los esquemas que vienen en las siguientes diapositivas) se consigue mejorar la independencia, integridad y seguridad de los datos.

ANSI/SPARC: ¿cómo?

- Arquitectura SGBD estándar
  - 3 niveles de representación de datos
    - difícil encontrar SGBD que lo implemente por completo



"Computer", Mike Kanert, http://www.flickr.com/photos/54851534@N06/5141302357/

En un principio, el estándar definió 3 niveles de representación que tenían distintas finalidades según "quién" debía usarlos.

•El esquema conceptual sirve para que los humanos nos comuniquemos con el motor de la base de datos. Es, entre otras cosas, el lenguaje que debemos usar para diseñar la base de datos.

•El interno es el medio que tiene el motor de base de datos para comunicarse con el sistema operativo y, por ende, con el almacenamiento primario y secundario.

•Los esquemas externos son los interfaces que conectan con los distintos lenguajes de programación y, más concretamente, con las aplicaciones que acceden a la base de datos.

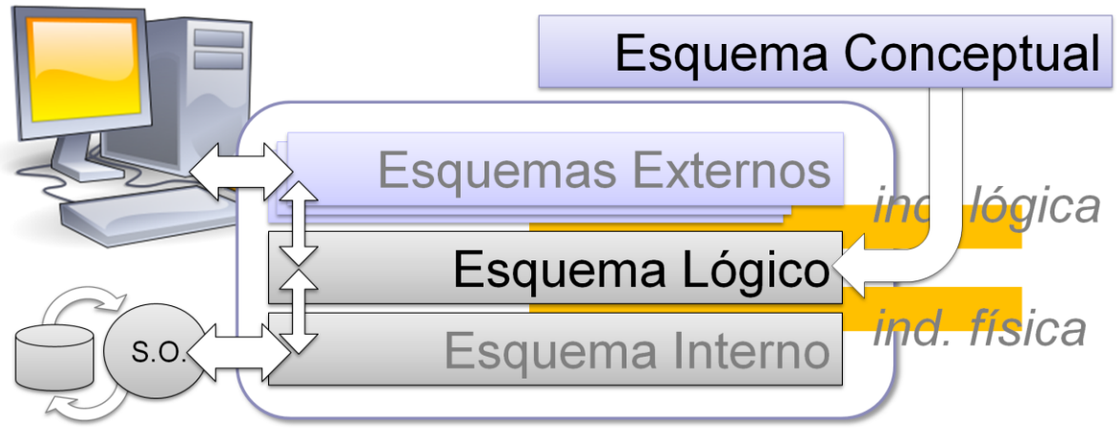
Esta arquitectura lógica (que no tiene por qué traducirse en una arquitectura física literalmente igual) persigue conseguir dos tipos de independencia de datos: lógica y física.

•La independencia lógica hace que cambios en el esquema conceptual NO afecte a esquemas externos que NO usan esos datos modificados. Es el ejemplo ya visto de cambiar la longitud de una columna T.c. Si T.c se utiliza en un programa "nóminas" (que se comunica con el SGBD mediante un esquema externo, más o menos ese subconjunto del esquema conceptual que necesita para generar las nóminas) "nóminas" debe recodificarse y, si fuera el caso, recompilarse. Sin embargo "correo", supongamos, no necesita T.c: "correo" no se toca y funciona perfectamente aún a pesar de haber cambiado el esquema conceptual.

•La independencia física viene a ser lo mismo pero entre el sistema operativo y el esquema conceptual: si cambiamos el tamaño de un índice o hacemos que un fichero directo pase a tener una organización *hash*, el esquema conceptual no necesita ser cambiado puesto que estos detalles "físicos" no le afectan, yo solo veo tablas, no sé ni me interesa cómo la guarda dentro de un archivo del SO.

ANSI/SPARC: ¿cómo?

- Un esquema más
  - se puede ver como la división del interno en dos y la separación del conceptual



No obstante, pronto se dieron cuenta de que era muy recomendable establecer un nivel de representación adicional que fuera realmente independiente de la máquina y del SGBD. La idea es que uno puede diseñar un sistema de información, describirlo, sin depender de una máquina concreta o incluso de un SGBD. Es decir, primero nos centramos en el problema y sus soluciones y, después, decidimos si lo traducimos a otro lenguaje que pueda entender un ordenador. Evidentemente, eso no quiere decir que no utilicemos ordenadores para todos ellos; por ejemplo, podemos diseñar en E-R y utilizando una herramienta informática pero, una vez que lo tenemos dibujado, ese esquema nos vale tanto para obtener las tablas correspondientes como para hacer una presentación a nuestro jefe, explicándole cómo va a funcionar el sistema.

En realidad, cada esquema identifica una fase de diseño e implementación de un sistema de información mecanizado.

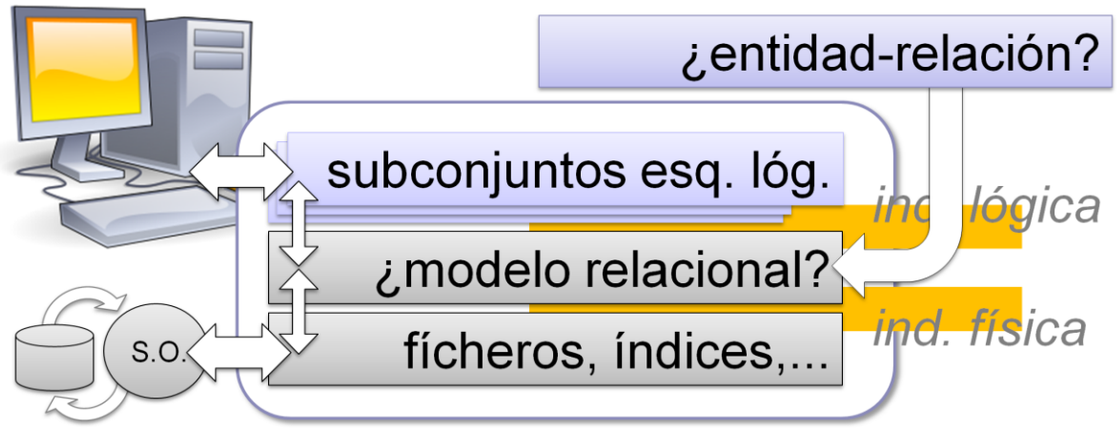
ANSI/SPARC: esquemas

- conceptual
  - descripción de los datos (y los procesos)
  - independiente del software o hardware
- lógico
  - según el modelo subyacente en el SGBD
- interno
  - comunicación con el sistema operativo
- externos
  - vistas parciales de aplicación/usuario

Por tanto, este es el resumen de la función de cada esquema.

ANSI/SPARC: modelos de datos

- cada esquema puede utilizar un modelo de datos diferente



Una posible secuencias de decisiones sobre qué modelos de datos utilizar para cada crear cada esquema.

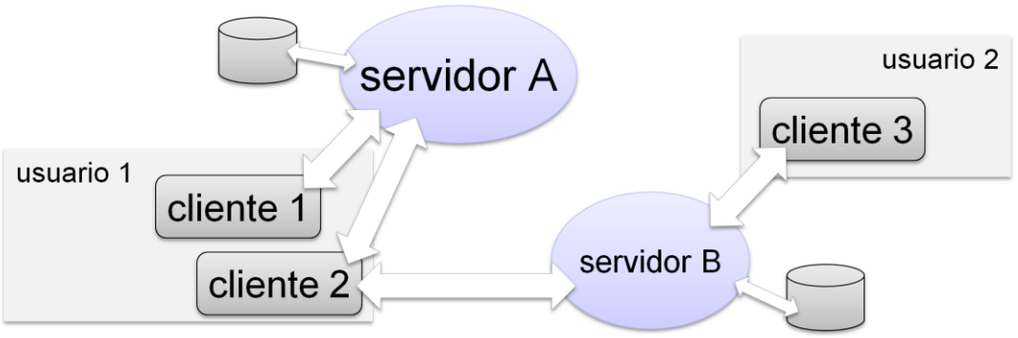
SGBD

- módulos, capas, herramientas...

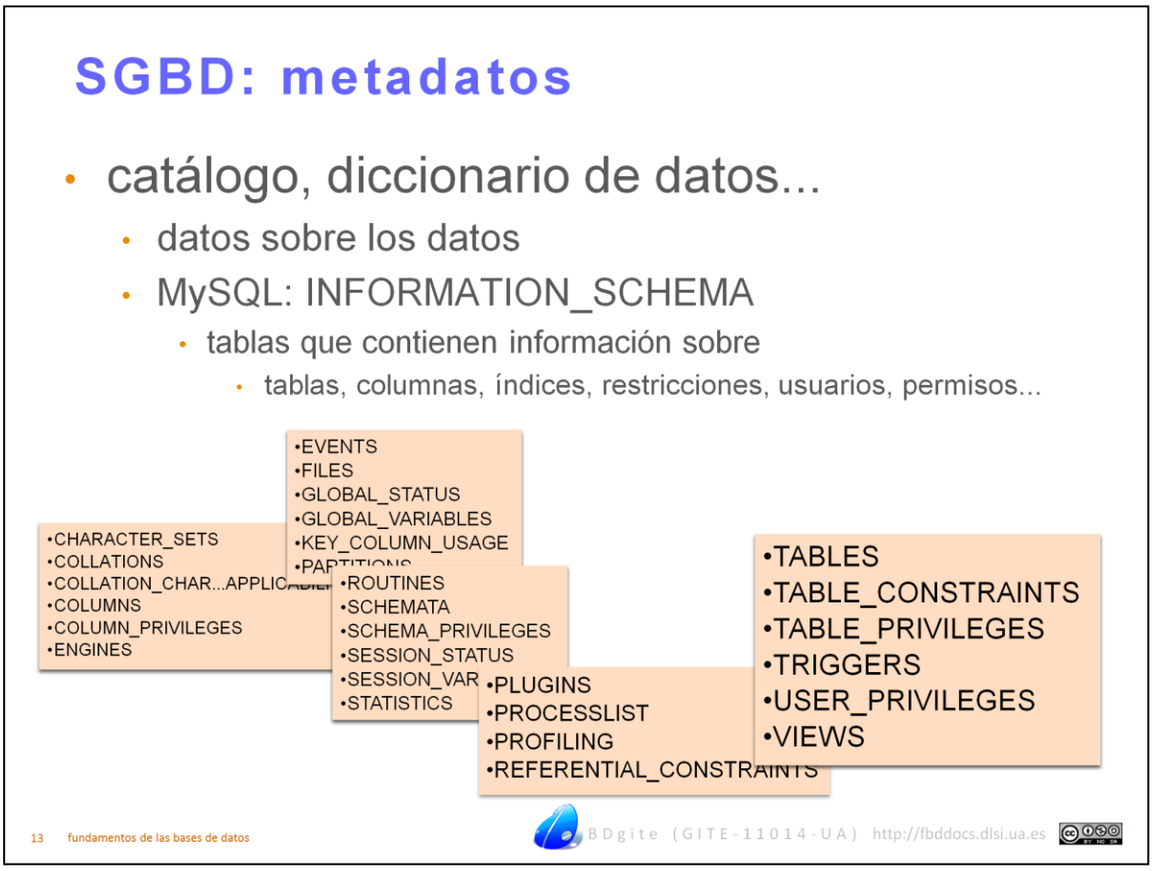
En realidad, un SGBD es un conjunto de software bastante complejo por su extensión, no es solo el módulo de consultas SQL. La idea es que hay ciertos módulos o conceptos indispensables para el SGBD (los de dentro de la elipse) y otros de tipo herramienta auxiliar.

SGBD: cliente-servidor

- consumidor-proveedor servicios
  - descargar al SGBD de labores de presentación
  - aprovechar la red, distribución de datos, transparencia, diferentes SGBD

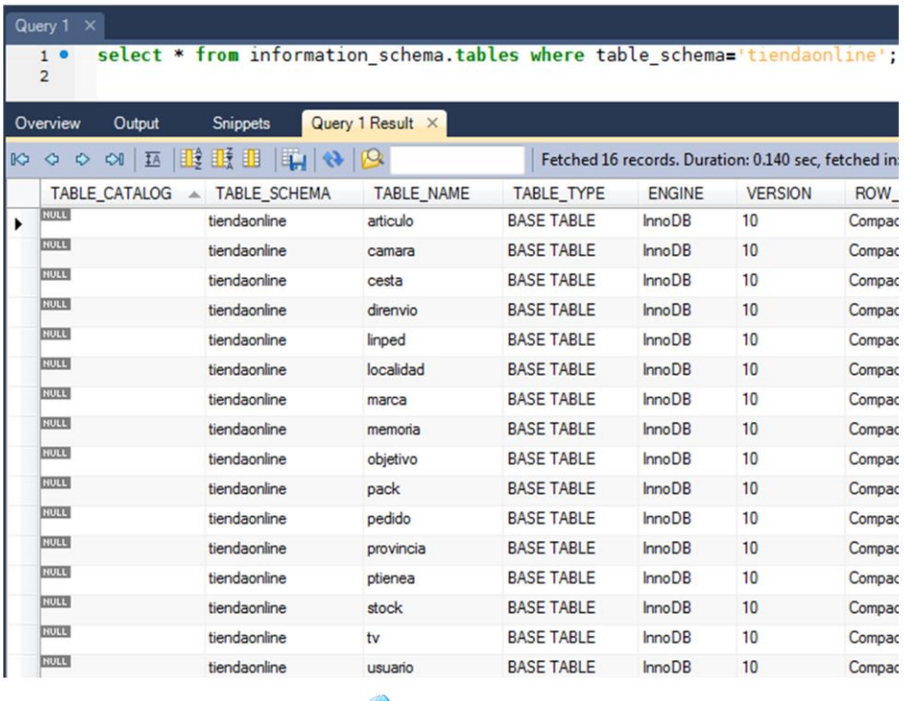


Un concepto que se ha generalizado tanto con Internet que ahora casi ni se nombra por asumido, aprovechar la red: un servidor puede atender varias peticiones, y un usuario puede realizar varias consultas simultáneas a servidores distintos y esperar las respuestas, en vez de hacer una consulta siempre después de haber recibido la respuesta a la anterior. En especialmente muy aprovechado por los SGBD que deben compartir datos entre muchos usuarios concurrentemente.



Un dato curioso sobre la gestión de las bases de datos es que toda la información de funcionamiento (los metadatos, datos sobre los datos) se encuentra almacenada en forma de tablas. Viene a ser algo así como los *.ini de cualquier programa. En general, la lectura de estas tablas, salvo en datos sensibles, está permitida a todos los usuarios y, además, son actualizadas siempre que hacemos algo (insertar, borrar, modificar, consultar). De hecho, el SGBD sabe de qué estamos hablando en una select porque puede consultar aquí cómo se llaman las columnas de una tabla concreta, avisarnos si nos hemos equivocado y esa tabla o columna no existe, ejecutar la consulta, etc.

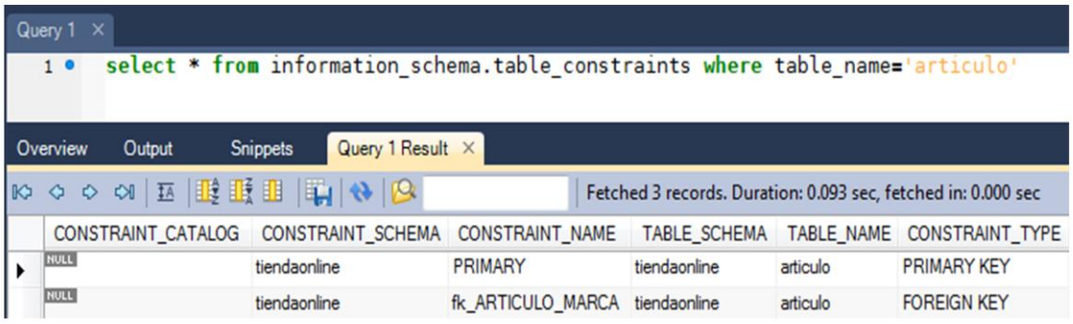
SGBD: metadatos



TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE	ENGINE	VERSION	ROW_
NULL	tiendaonline	articulo	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	camara	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	cesta	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	direnvio	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	linped	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	localidad	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	marca	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	memoria	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	objetivo	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	pack	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	pedido	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	provincia	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	ptienea	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	stock	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	tv	BASE TABLE	InnoDB	10	Compac
NULL	tiendaonline	usuario	BASE TABLE	InnoDB	10	Compac

Una vista del catálogo MySQL desde MySQL Workbench. En realidad, el diccionario de datos se guarda en la base de datos "mysql", information_schema se puede asociar a "vistas" sobre las tablas de "mysql" que restringen la información visible a aquella sobre la que tiene permisos el usuario que está accediendo.

SGBD: metadatos



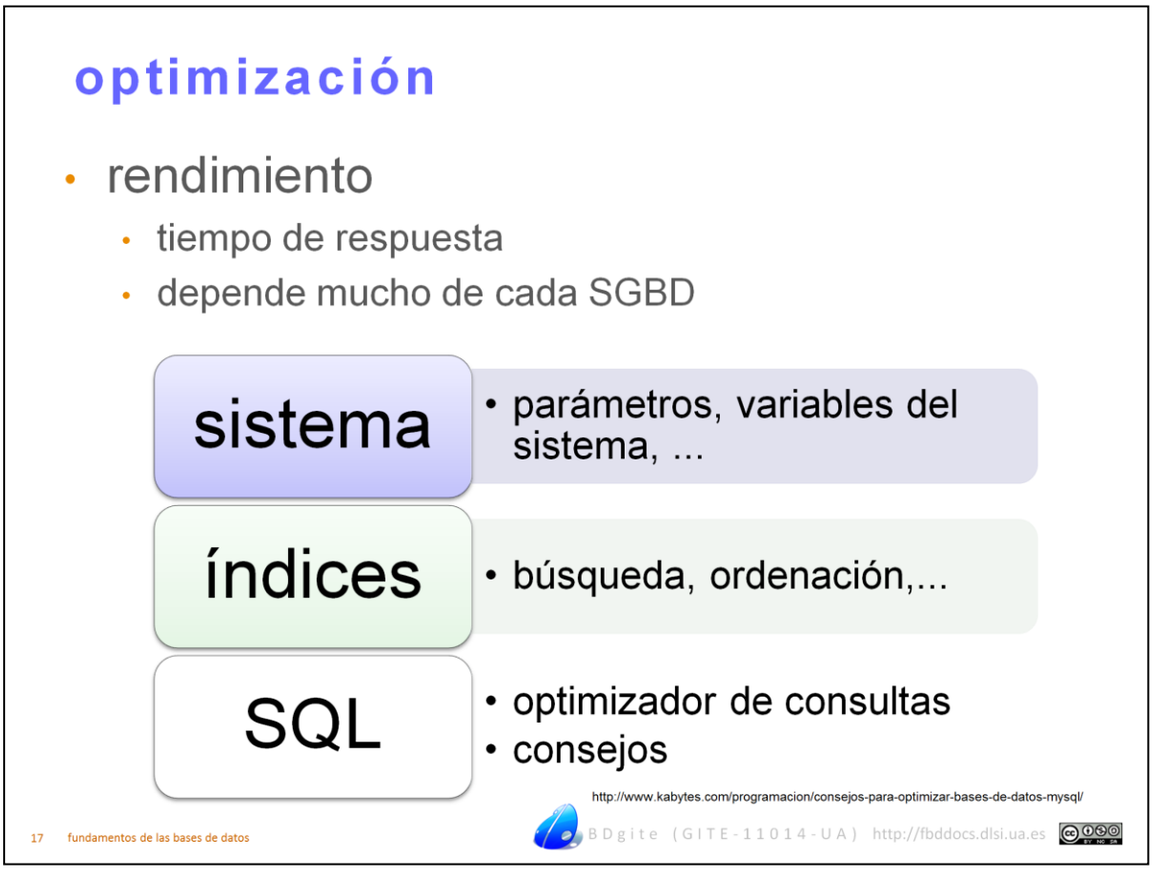
La función del catálogo incluye el proporcionarnos información a nosotros como usuario. En este caso, las restricciones sobre una tabla.

SGBD: personal

- directivos
  - jefes: organizan, coordinan, planifican
- analistas
  - control, diseño, coordinación
- administradores
  - responsable seguridad, integridad, rendimiento...
- desarrolladores/programadores
  - carga, aplicaciones, ...
- equipo de mantenimiento
  - soporte a usuarios

http://es.scribd.com/doc/51654817/19/2-4-5-Modelo-ANSI-X3-SPARC

Como todas las listas de este tipo, la necesidad filosófica de analizar, generalizar y clasificar el mundo, ahora el de las bases de datos. Que la realidad y el día a día son bastante más complejos, nadie lo duda.



Uno de los aspecto primordiales, y objetivo inexcusable de cualquier administrador de base de datos, es garantizar el rendimiento óptimo del motor de base de datos en casi cualquier situación. Esta tarea, compleja donde las haya, se basa en una multitud de parámetros que debemos controlar en los posible. Van desde los propios de configuración tanto de sistema operativo como de sistema de base de datos, hasta la definición de índices (estructuras auxiliares que, en general, se definen para acelerar la búsqueda y recuperación de datos desde las tablas cuando se detecta cierto tipo de consultas como "muy frecuentes") y el análisis y reescritura, automática o no, de consultas SQL.

optimización

- factores
  - búsqueda en disco
  - lectura y escritura en disco
  - ciclos de CPU
  - ancho de banda de memoria

http://dev.mysql.com/doc/refman/5.0/es/optimize-overview.html

Como decíamos, aspectos no directamente bajo la responsabilidad del SGBD afectan extraordinariamente al rendimiento de la base de datos. Estos de aquí casi se puede decir que son de sentido común.

optimización: índices

- estructura auxiliar de localización de registros
- balance
  - aceleran la búsqueda
  - retrasan la modificación e inserción
- consejos
  - claves cortas, sin duplicados,...
  - sopesar bien su necesidad

Los índices son una de las herramientas de rendimiento más potentes de un SGBD. Algunos se definen de forma automática (cuando creamos un clave primaria o ajena, por ejemplo) y otros los podemos crear nosotros para disminuir el tiempo de respuesta. Son estructuras más cercanas al esquema interno (aunque los definimos desde el propio SGBD) y, por eso, tienen un efecto directo sobre cómo se organizan los archivos en almacenamiento secundario.

Por otro lado, hay que ser cuidadoso en cuántos se definen y cómo, la creación de un índice no garantiza de por sí que nuestras consultas vayan a ir mejor. Es evidente que no deja de ser "un paso más" al que obligamos al SGBD a realizar para devolvernos los resultados. Solo que a veces compensa dar ese paso adicional. También afecta el tipo de claves que utilizamos para definir ese índice, por ejemplo.

optimización: índices

- MySQL los usa para
  - *...where*
  - *joins*
  - encontrar min()/max()
  - ordenar
  - agrupar (*group by*)
- otros SGBD
  - tendrán sus propios tipos de índice
    - clustered
    - bitmap
    - ...

Motor	Tipos
MyISAM	BTREE
InnoDB	BTREE
MEMORY/HEAP	HASH, BTREE

Por otro lado, tipos de índice y función de los mismos hay muchos y para cada producto en particular (MySQL no tiene por qué ofrecer los mismos tipos de índice que Oracle, por ejemplo). Aquí vemos que MySQL utiliza árboles B y tablas hash.

optimización: SQL

- el optimizador de consultas
  - módulo presente en casi todos los SGBD
  - las consultas
    - pueden tener expresiones alternativas
    - diferentes algoritmos para cada operación
    - distintos recursos susceptibles de utilización (índices, ...)
  - se evalúa
    - el coste de cada alternativa
  - es transparente al usuario

http://www.slideshare.net/koolkampus/ch14

Desde los albores del modelo relacional, se dieron cuenta de que había que encontrar un modo de mejorar el tiempo que se tarda en lanzar una consulta SQL y obtener el resultado. El optimizador de consultas es un módulo del SGBD indispensable. De hecho, hemos escrito "...en casi todos los SGBD" más por prudencia que por convencimiento.

Resumiendo mucho, lo que hace es analizar la consulta, trocearla en subconsultas, evaluar, por ejemplo, si se ganará algo ejecutándolas en uno u otro orden y, finalmente, decidir que una de estas secuencias es la mejor y ejecutarla. Todo ello, claro, sin que el usuario se entere ni note retardo alguno. Curiosamente, muchos optimizadores de consultas utilizan álgebra relacional en alguna parte de sus algoritmos.

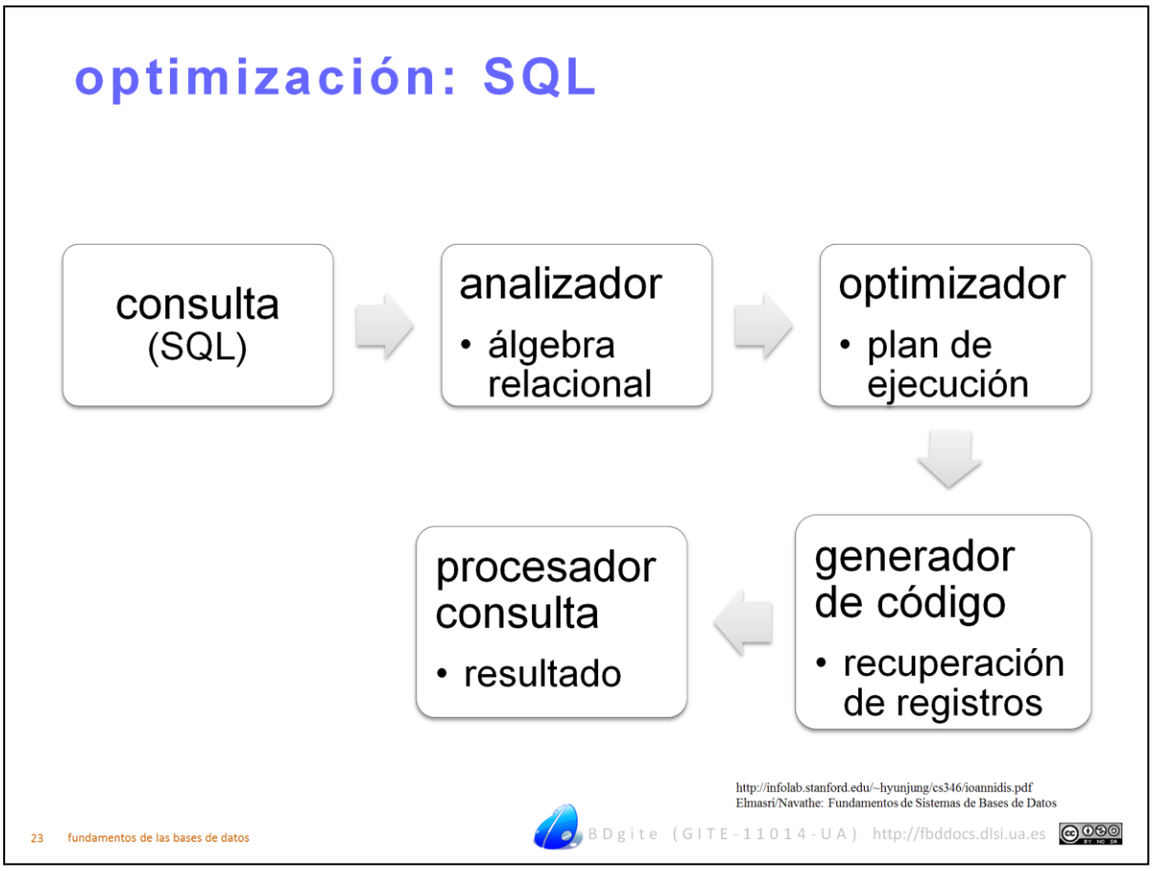
optimización: SQL

- planes de ejecución
  - distintas alternativas de resolución de las que el sistema escoge la menos costosa
  - uso de información estadística
    - tamaño de la tupla
    - número de tuplas en una relación
    - número de bloques (de disco) que contienen esas tuplas
    - tamaño del bloque
    - valores distintos de un atributo
    - cantidad de niveles de índice
    - cercanía física de los registros
    - ...
  - MySQL: *explain* select ...

http://www.slideshare.net/koolkampus/ch14
http://www.sergiquinonero.net/explain-mysql-o-como-optimiza-sql.html
http://dev.mysql.com/doc/refman/5.0/es/explain.html

Esas secuencias alternativas de ejecución de los trozos obtenidos de una consulta a ejecutar se traducen en planes de ejecución, que le sirven al SGBD para calcular, aproximadamente, cuál va a ser el más rápido teniendo en cuenta una serie de variables.

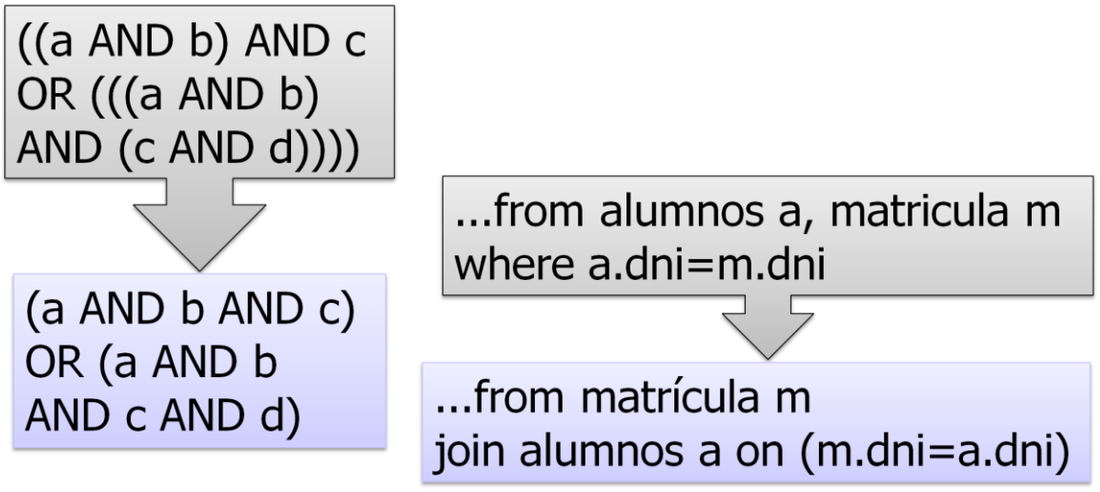
Todos los SGBD tienen comandos para mostrarnos el plan de ejecución elegido por él. El objetivo es que nosotros podamos decidir reescribir la consulta de otra forma o generar un índice para ayudar, por ejemplo.



Esquema básico del optimizador de consultas.

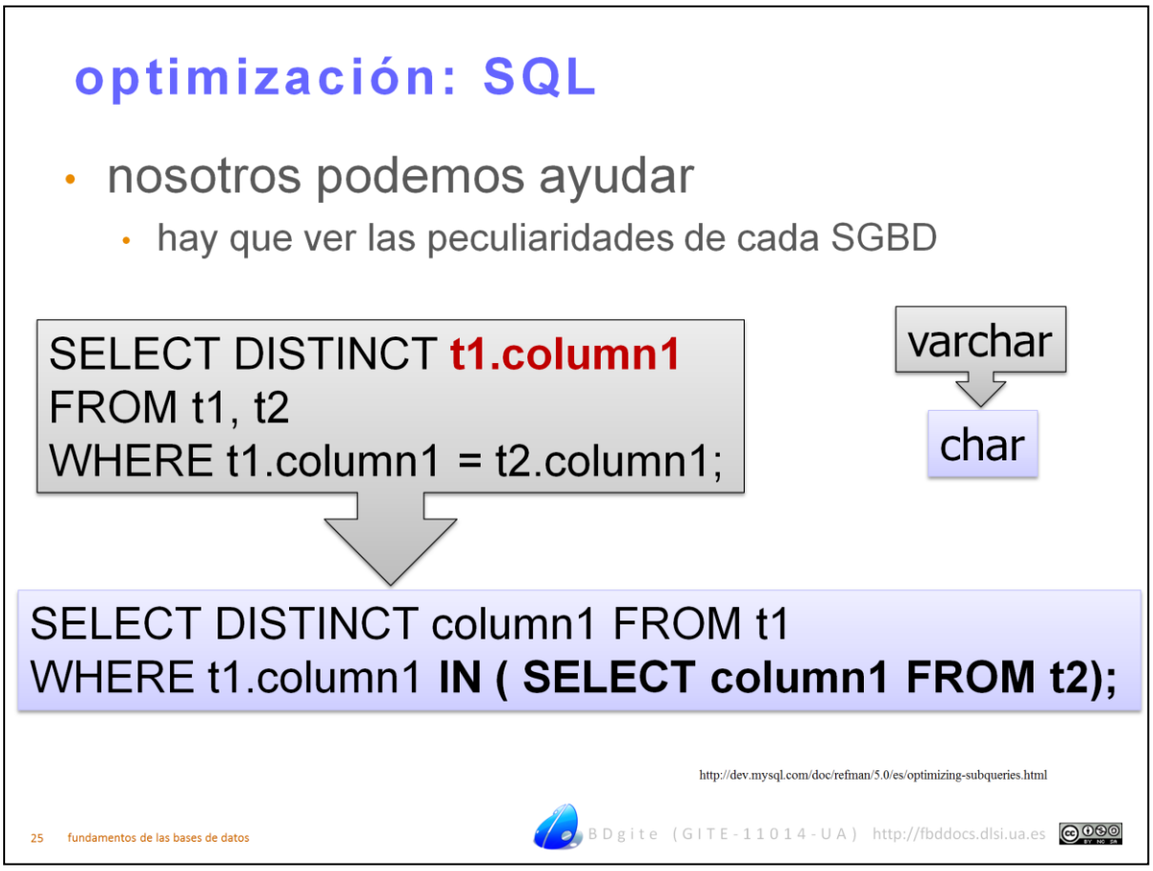
optimización: SQL

- nosotros podemos ayudar
  - hay que ver las peculiaridades de cada SGBD



http://dev.mysql.com/doc/refman/5.0/es/where-optimizations.html
http://mundobi.wordpress.com/2007/06/16/optimizacion-de-consultas-sql-parte-ii/

Ejemplos básicos de acciones que podemos llevar a cabo nosotros mismos para acelerar una consulta: eliminar paréntesis innecesarios, el from-where hace un producto cartesiano, el join ¿no?, el orden de las tablas puede influir, select *, ...



...usar char en vez de varchar en los índices acelera la recuperación de datos, sustituir el join por una subconsulta (cuando sea posible)...

Téngase en cuenta que estamos hablando de situaciones "límite". Si tienes una base de datos con una tabla de 10 filas sería una tontería pensar en estas cosas.

conclusión

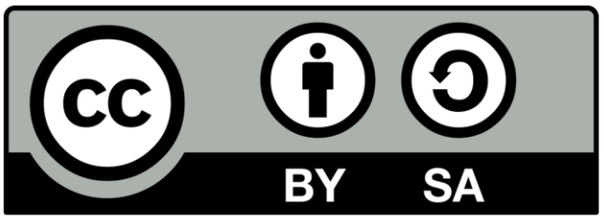
- Sistemas de Gestión de Bases de datos
  - ANSI propuso un estándar centrado en los niveles de representación de datos
  - es un paquete de software complejo
  - cliente-servidor es el modelo más usual
  - necesita muchas horas de afinación y constante supervisión
  - reescribir consultas, utilizar correctamente los índices, …

conclusiones y fin

conclusión: referencias

http://fbddocs.dlsi.ua.es/lecturas

licencias



licencias

imágenes:http://pinterest.com/fundamentosbd/t6-sgbd/
Librería MS Office
commons.wikimedia.org