DISEÑO FÍSICO de Bases de Datos

- Especificación de estructuras de almacenamiento y caminos de acceso específicos para que las diversas aplicaciones que accedan a la BD tengan un buen rendimiento
- Cada SGBD específico ofrece varias opciones de organización de ficheros y caminos de acceso:
  - Diferentes tipos de ÍNDICES
  - Agrupamiento de registros (de distinto tipo) relacionados, en los mismos bloques de disco (CLUSTER de ficheros)
  - Distintos tipos de técnicas de dispersión (HASHING)
  - Diferentes valores para los parámetros físicos (tamaño de bloque, de buffers, ...)
  - ...
- Una vez elegido el SGBD, el Diseño Físico consiste en elegir las estructuras más apropiadas para los ficheros de la BD, entre las opciones que ofrece el SGBD
- El Diseño Físico es muy dependiente del SGBD comercial seleccionado

DISEÑO FÍSICO de Bases de Datos

CRITERIOS PARA ELEGIR OPCIONES DE DISEÑO FÍSICO (~OBJETIVOS)

- TIEMPO DE RESPUESTA (debe minimizarse)

  Tiempo entre la introducción de una transacción T de BD y la obtención de respuesta
  - Depende de (entre otras cosas)...
    - TIEMPO DE ACCESO A LA BASE DE DATOS para obtener los datos que T necesita (bajo el control del SGBD)
    - CARGA DEL SISTEMA, PLANIFICACIÓN DE TAREAS DEL SO, RETRASOS DE COMUNICACIÓN (fuera del control del SGBD)
- PRODUCTIVIDAD DE LAS TRANSACCIONES (debe maximizarse)

  Número promedio de transacciones que el SBD puede procesar por minuto
  - Parámetro crítico de los sistemas de procesamiento masivo de transacciones
  - Debe medirse en "condiciones pico" del sistema
- APROVECHAMIENTO DEL ESPACIO (debe optimizarse)

  Cantidad de espacio ocupado por ficheros de la BD y sus estructuras de acceso

DISEÑO FÍSICO de Bases de Datos

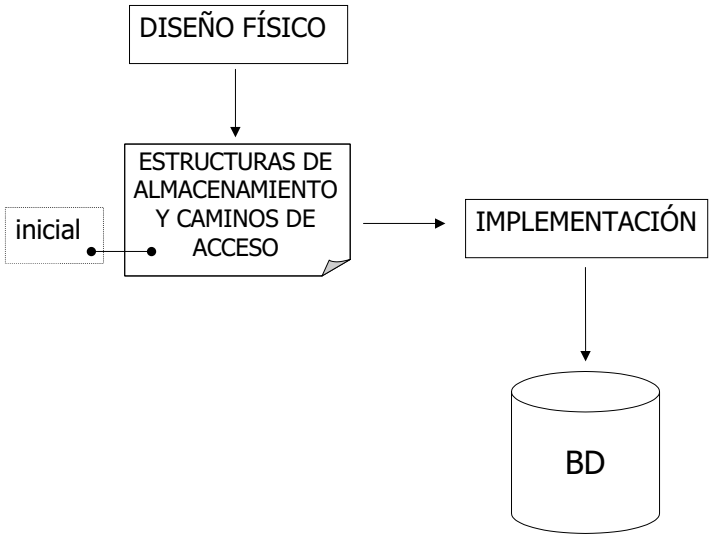
Normalmente se suele...

- Especificar LÍMITES PROMEDIO y del PEOR DE LOS CASOS de cada parámetro como parte de los REQUISITOS de RENDIMIENTO del Sistema
- Puesto que el rendimiento depende del TAMAÑO y NÚMERO de REGISTROS en los ficheros ⇒
  - * estimar parámetros tamaño-registro y nº-registros para cada fichero
  - * estimar también "cómo y cuánto va a crecer" cada fichero (en tamaño de registro, o en número de registros)
- Estimar patrones de actualización y obtención de datos, para cada fichero, considerando todas las transacciones

Y en el proceso de diseño físico...

- Utilizar técnicas analíticas o experimentales (prototipos, simulación...) y suponiendo diferentes decisiones de diseño físico, ESTIMAR valores promedio y del peor de los casos para cada una de ellas, para ver si se satisfacen los requisitos de rendimiento especificados

DISEÑO FÍSICO de Bases de Datos



DISEÑO FÍSICO de Bases de Datos

- Muchos sistemas incluyen UTILIDADES DE SUPERVISIÓN que reúnen **ESTADÍSTICAS DE RENDIMIENTO**
  - Almacenadas en el CATÁLOGO, para su análisis posterior
  - Datos Estadísticos:
    - Nº de INVOCACIONES de TRANSACCIONES y/o CONSULTAS PREDEFINIDAS
    - Actividad de Entrada/Salida, para cada fichero
    - Nº de bloques (páginas) por fichero
    - Nº de entradas por índice
    - Frecuencias de UTILIZACIÓN de índices
    - …
- Los cambios en Requerimientos del Sistema de BD provocan
  - Nuevas Consultas o Transacciones,
  - Nuevos requerimientos temporales, ...

⇒ Reorganizar algunos ficheros:

-- Creación de nuevos índices, o

-- Modificación de métodos primarios de acceso

DISEÑO FÍSICO de Bases de Datos

- El SGBD no permite que el usuario especifique el mecanismo/camino de acceso que utilizar en la ejecución de operaciones SQL
- ¿Quién realiza la elección? El Optimizador (módulo del SGBD)
  - Analiza varias formas de realizar los pasos para ejecutar una sentencia SQL
  - Selecciona un método óptimo:

    Evalúa alternativas de acceso basándose en
    - Tiempo de CPU estimado
    - Nº operaciones de E/S
    - Estadísticas y estimaciones (que saca del catálogo) sobre nº de tuplas, existencia de índices o hashing, etc.
- Pero el diseñador puede especificar los mecanismos/caminos de acceso de los que dispondrá el Optimizador

«El problema de diseño físico para el administrador de la BD consiste en proporcionar un conjunto eficiente de estructuras de acceso, de modo que el Optimizador pueda tomar las mejores decisiones» (Ceri, 83)

FACTORES QUE CONSIDERAR EN EL DISEÑO FÍSICO de Bases de Datos

Factores que afectan al rendimiento de las aplicaciones (transacciones y consultas)

- Objetivos del DISEÑO FÍSICO
  - ESTRUCTURACIÓN ADECUADA de datos en el ALMACENAMIENTO
  - de tal forma que GARANTICE un BUEN RENDIMIENTO de las aplicaciones
- Un Esquema Lógico tiene muchos Esquemas Físicos posibles en cierto SGBD

  ¿Cuál es el más apropiado?

  ¿En qué podemos basarnos para decidirnos por uno u otro?
- Imposible analizar el RENDIMIENTO ni TOMAR DECISIONES DE DISEÑO sin saber QUÉ USO SE LE VA A DAR A LA BASE DE DATOS
  1. Consultas/Transacciones
  2. Frecuencia (esperada) de Consultas y Transacciones
  3. Restricciones de Tiempo (especiales) de Consultas y Transacciones
  4. Frecuencia (esperada) de operaciones de Actualización de la BD

FACTORES que CONSIDERAR en el DISEÑO FÍSICO

1. ANÁLISIS DE CONSULTAS Y TRANSACCIONES

- Definir (alto nivel) transacciones y consultas que se espera ejecutar en la BD

PARA CADA CONSULTA *se debe especificar...*

(a) FICHEROS a los que accede

(b) CAMPOS sobre los que se especifica alguna CONDICIÓN DE SELECCIÓN

(c) CAMPOS sobre los que se especifica alguna CONDICIÓN DE REUNIÓN o de ENLACE de REGISTROS de diferente tipo

(d) CAMPOS cuyos valores obtiene la consulta

(b) y (c) ⇨ Candidatos para DEFINIR ESTRUCTURAS DE ACCESO sobre ellos

PARA CADA TRANSACCIÓN y operación de ACTUALIZACIÓN, *especificar…*

(a) FICHEROS que actualiza

(b) Operación que realiza en cada fichero (INSERCIÓN, MODIFICACIÓN, ELIMINACIÓN)

(c) CAMPOS sobre los que se especifica alguna CONDICIÓN DE SELECCIÓN para las modificaciones y borrados

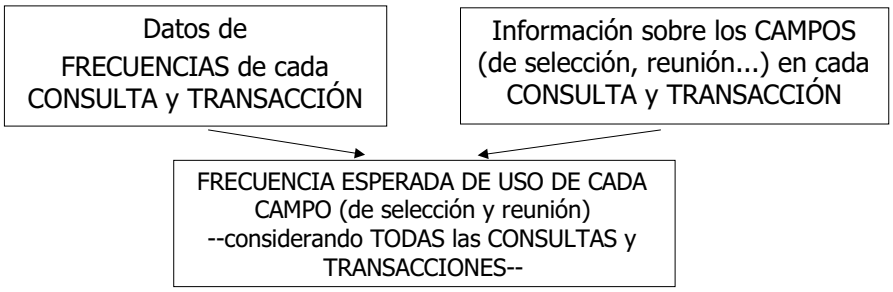
(d) CAMPOS actualizados (por una operación de modificación)

(c) ⇨ Candidatos para DEFINIR ESTRUCTURAS DE ACCESO sobre ellos

(d) ⇨ Candidatos para EVITAR definir ESTRUCTURAS DE ACCESO sobre ellos

FACTORES que CONSIDERAR en el DISEÑO FÍSICO

2. ANÁLISIS DE FRECUENCIA ESPERADA DE INVOCACIÓN DE CONSULTAS Y TRANSACCIONES (tasas o velocidades de invocación)



Si el volumen de procesamiento es elevado, aplicar la **regla informal del 80-20**

⇒ NO es necesario SUPERVISAR TODAS las Consultas y Transacciones para recoger Estadísticas y Tasas de Invocación, sino que BASTA con hacerlo con un 20% de ellas (las que realizan el 80% del procesamiento).

** Además, raras veces se conoce todas las consultas/transacciones en el momento en que se hace el Diseño Físico.

FACTORES que CONSIDERAR en el DISEÑO FÍSICO

3. ANÁLISIS DE RESTRICCIONES DE TIEMPO SOBRE CONSULTAS Y TRANSACCIONES

"T debe TERMINAR ANTES de 6 segundos en el 95% de las veces que sea invocada y NUNCA debe DURAR más de 25 segundos"

- Restricciones de Tiempo ⇒ ASIGNACIÓN DE PRIORIDADES ADICIONALES a los campos candidatos para ESTRUCTURAS DE ACCESO

  Un campo de selección utilizado por T será candidato con mayor prioridad para estructura de acceso que otros

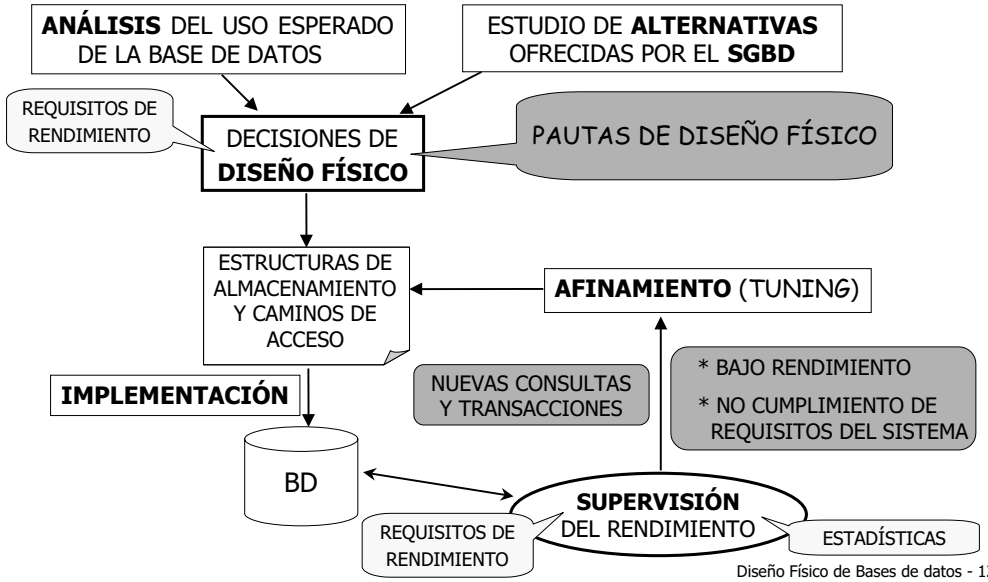
FACTORES que CONSIDERAR en el DISEÑO FÍSICO

4. ANÁLISIS DE FRECUENCIA ESPERADA DE OPERACIONES DE ACTUALIZACIÓN

- Los ficheros que se modifican mucho deben tener el MÍNIMO número posible de estructuras/caminos de acceso

  Actualizar el fichero (inserciones, eliminaciones, modificaciones)
  ⇒ Actualizar los caminos de acceso
  ⇒ más lentas las operaciones de actualización

PROCESO DE DISEÑO FÍSICO



DISEÑO FÍSICO DE BD's RELACIONALES

PAUTAS DE AYUDA EN LA TOMA DE DECISIONES DE DISEÑO FÍSICO DE BASES DE DATOS RELACIONALES

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

- La mayoría de SGBD representa cada RELACIÓN base como un FICHERO
- Es necesario especificar...
  - Tipo de fichero,
  - Atributos sobre los que definir estructuras de acceso
  - Tipo de dichas estructuras de acceso

② TÉCNICAS PARA ACELERAR la operación de EQUIRREUNIÓN y/o de REUNIÓN NATURAL

(a) Archivos Mixtos

(b) Desnormalización, por razones de eficiencia

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

(a) UTILIZACIÓN DE FICHERO NO ORDENADO SIN ÍNDICES

- El acceso siempre es secuencial (*scan*)
- Conviene si...
  - ...la tabla es pequeña (inspeccionada con pocos accesos E/S --a bloque)
  - ...normalmente se seleccionan muchas tuplas (>20%)
  - ...otros caminos de acceso resultan demasiado costosos

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

(b) UTILIZACIÓN DE FICHERO ORDENADO

- Para tablas medio-grandes que son frecuentemente...
  - ...ordenadas según valores de un atributo ***A*** (***A*** aparece en cláusulas ORDER BY, GROUP BY, UNION, DISTINCT)
  - ...accedidas con criterios de selección que incluyen un rango de valores de ***A***
  - ...procesadas secuencialmente en el orden de los valores de ***A*** (acceso ordenado)
- y no frecuentemente actualizadas con INSERT/DELETE, ni UPDATE (*A*)
- se las llama Clustered Tables o Tablas Agrupadas
- Algunos SGBD 'obligan' a construir un índice para la tabla sobre el atributo *A*
  - ÍNDICE PRIMARIO (...UNIQUE...CLUSTER), si *A* es clave
  - ÍNDICE de AGRUPAMIENTO (...CLUSTER), si *A* no es clave

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

(c) UTILIZACIÓN DE ÍNDICES

- Tablas indexadas

OPCIÓN 1.- (preferida)

- Usar un FICHERO **NO ORDENADO**, y definir
- ÍNDICES SECUNDARIOS para atributos muy usados en condiciones de selección y de reunión

OPCIÓN 2.-

- FICHERO **ORDENADO** según valores de un atributo *A*, especificando un
- ÍNDICE sobre *A*
  - PRIMARIO (UNIQUE), si *A* es clave
  - de AGRUPAMIENTO (CLUSTER), si *A* no es clave
- ÍNDICES SECUNDARIOS para otros atributos muy usados en condiciones de selección y de reunión

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

Conviene UTILIZAR índices...

- sobre tablas medianas/grandes (ocupan gran cantidad de espacio)
- para facilitar el acceso a pequeños porcentajes del total de tuplas (≤20%)
- y para...
  - evitar el recorrido (scan) de la tabla completa
  - evitar la ordenación de las tuplas de la tabla (registros del fichero)
    - ORDER BY, GROUP BY, UNION, DISTINCT, JOIN
  - evitar el acceso al fichero para determinadas consultas
    - EXISTS, IN, ...
  - evitar el acceso al fichero para consultas que incluyen un pequeño subconjunto de columnas (gracias a los índices compuestos)

```
PERSONA(ssnum, apellido, nombre, edad, telef)
CREATE INDEX idx_perso ON PERSONA(apellido, nombre)
```


① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

Conviene EVITAR los índices...

- si la tabla es muy pequeña (ocupa poco espacio)
- si se degradan los requisitos de procesamiento crítico
- si el costo de almacenamiento+mantenimiento >>>> beneficio
  - INSERCIÓN, MODIFICACIÓN, ELIMINACIÓN sobre tablas indexadas
    ⇒ mantenimiento del índice (automático, por parte del SGBD)
  - A veces conviene CREAR la tabla, rellenarla, y luego CREAR los ÍNDICES
  - Y a veces conviene
    - ELIMINAR ÍNDICES
    - realizar las MODIFICACIONES sobre la BD y
    - luego CREAR de nuevo los ÍNDICES
- Evitar crear índices sobre columnas...
  - que son modificadas muy a menudo
  - con distribución irregular de valores (pues confunden al Optimizador)

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

Consejos para la utilización de índices (1)

- Máximo de 4 índices por tabla. Más, si rara vez es actualizada la tabla.
  - ↑ indexación ⇒ ↑ necesidad de espacio + ↓ velocidad de inserción/eliminación
- Crearlos sobre columnas ...
  - que suelen aparecer en cláusulas WHERE en comparaciones de =, >, <, BETWEEN
  - que suelen ser atributos de reunión de varias tablas (p.ej. claves externas)
  - con gran variedad de valores (mayor discriminación en las búsquedas)
- Almacenar índices en espacio de almacenamiento distinto al de los datos

  * Aplicaciones con muchas operaciones de INSERT, DELETE *
  - Índices primarios/agrupamiento en el mismo espacio que los datos
    - La mayoría de LECTURAS y todas las INSERCIONES los usan
  - Índices secundarios en un espacio de almacenamiento diferente
    - Mejor balanceo de carga, pues un INSERT en la tabla implica ACTUALIZAR índices

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

Consejos para la utilización de índices (y 2)

El SGBD Oracle ...

- Implementa los índices mediante árboles B^+
- No incluye entrada en el índice para tupla con NULL en la columna de indexación
- El Optimizador NO usa el índice construido sobre una columna, si ...
  - La consulta no incluye WHERE,
  - La consulta usa la columna indexada aplicándole una función (SUBSTR ...) u operador (||...)

  pero el SGBD sí usa el índice si...
  - La consulta contiene un ORDER BY la columna indexada,
  - La consulta aplica MAX o MIN sobre una única columna (y es la indexada)

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

UTILIZACIÓN DE HASHING (Ficheros dispersos)

- Alternativa al uso de tablas indexadas y tablas agrupadas
- Reduce el nº de accesos a bloque para encontrar un registro (fila)

Conviene USAR ficheros dispersos...

- Si los accesos a la tabla son frecuentes, individuales y aleatorios
- y el campo de dispersión será un atributo o conjunto de atributos ***A*** tal que...
  - Es muy utilizado en operaciones de SELECCIÓN POR IGUALDAD
  - Es muy usado en condiciones de REUNIÓN (JOIN)
  - Apenas son modificados sus valores
  - Los valores de ***A*** son muchos y variados (permiten una buena distribución de los registros, evitando colisiones)
- Si se conoce el tamaño del fichero y no se espera crecimiento/reducción considerable (en caso contrario, puede usarse Dispersión Dinámica)

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

Conviene <u>EVITAR</u> ficheros dispersos...

- Si los accesos a la tabla son
  - por rango de valores de ***A*** (salvo si es una lista: *A* IN (v_1, v_2, ...v_n))
  - con criterios de selección que NO incluyen ***A***, o incluyen sólo parte de ***A***
  - en orden, según los valores de ***A***

NOTAS:

- No todos los SGBD soportan hashing
- Sólo puede definirse un hashing por tabla
- No es posible combinar hashing y fichero ordenado (ambos determinan la posición física del registro en el fichero)

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

RESUMEN: CONSEJOS GENERALES para la elección de la organización física para un fichero individual (1)

- Usar **fichero ordenado** y escoger como atributo de ordenación el más usado
  - para obtener o procesar los registros en orden, o
  - en operaciones de reunión con este fichero, o
  - en selecciones que incluyen un rango de valores sobre ese atributo

  creando un **índice** sobre ese atributo (camino de acceso primario)
    - Índice PRIMARIO (atributo clave) -- CREATE UNIQUE INDEX... CLUSTER
    - Índice de AGRUPAMIENTO (no clave) -- CREATE INDEX ... CLUSTER
- Usar un **fichero no ordenado** si
  - la tabla es pequeña, o
  - ningún atributo satisface estos criterios, o
  - el fichero es frecuentemente actualizado ...
    - inserciones/borrados de tuplas (registros)
    - modificaciones del atributo de ordenación

① ELECCIÓN DE ESTRUCTURAS DE DATOS Y CAMINOS DE ACCESO

CONSEJOS GENERALES (y 2)

- Para cada atributo (no de ordenación) que se use mucho en operaciones de selección o reunión, crear un **índice secundario** -- CREATE [UNIQUE] INDEX ... (camino de acceso secundario)
- Si el fichero se va a actualizar mucho (INSERCIÓN / ELIMINACIÓN), reducir al MÍNIMO el número de ÍNDICES del fichero
- Usar un **fichero disperso** si algún atributo ...
  - se usa mucho para selección por igualdad u operaciones de reunión
  - nunca se utiliza para obtener los registros en orden
  - sus valores son muchos y diferentes, y apenas se modifica su valor

  y se puede crear **índices secundarios** sobre otros atributos del fichero

(Elección limitada por el SGBD escogido)

②TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

(a) FICHEROS MIXTOS (cluster de tablas)

- Almacenar DOS RELACIONES
  - con una interrelación 1:N (**clave primaria/clave externa**)
  - en un ÚNICO FICHERO {JERÁRQUICO de DOS NIVELES}
  - según la clave del cluster (correspondiente a los atributos PK y FK)
- Cada registro del lado 1 (clave primaria) está SEGUIDO de los registros del lado N (claves ajenas coincidentes)
- En Oracle es necesario crear un índice sobre la clave del cluster (cluster index) para poder acceder a su contenido

☺

* Consultas MUY EFICIENTES sobre la clave del cluster ...
  - REUNIONES entre ambas tablas
  - Agrupamiento de registros del lado N por el valor de la clave externa

  (los registros relacionados están en el mismo bloque ⇒ menos accesos a bloque)

* Mayor aprovechamiento del espacio

② TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

☹

* Consultas a cada tabla por separado resultan muy poco eficientes (acceso multi-tabla)

* Recorrido completo (scan), de cada tabla por separado, más costoso (demasiados accesos a bloque)

* INSERCIÓN poco eficiente debido a que se debe
  - mantener la ordenación física,
  - al tiempo que desaprovechar el mínimo espacio de almacenamiento

⇒ muchos encadenamientos entre registros (zonas de desborde, etc.)

⇒ menor eficiencia de las búsquedas según la clave del cluster

⇒ reducción del rendimiento de las consultas

② TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

(b) DESNORMALIZACIÓN

«Desnormalizar, es decir, violar la normalización, sólo tiene una excusa: rendimiento … y sólo en algunas situaciones » [Shasha-92]

- La normalización hasta la 3FN, FNBC o superior, facilita la comprensión de...
  - los datos y
  - relaciones entre los datos,

  que deben protegerse y mantenerse al crear las aplicaciones
- En aplicaciones importantes o muy sencillas,

  cuyas tareas no se adaptan a tablas normalizadas,

  una vez terminado el análisis, puede ser necesario desnormalizar algunas tablas,

  para conseguir una aplicación que se adapte a...
  - las tareas de los usuarios
  - los requisitos de tiempo

② TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

- CLAVES SIGNIFICATIVAS

Esquema de base de datos normalizado

EMPLEADO(<u>nombre</u>, edad, alojamiento)

OFICIO_EMPLEADO(<u>nombre, oficio</u>, calificacion)

OFICIO(<u>oficio</u>, descripcion)

ALOJAMIENTO(<u>alojamiento</u>, nombreCompleto, direccion, responsable)

El análisis de requisitos indica que la mayoría de las consultas más frecuentes sobre empleados, necesitan obtener el nombre del oficio y el del alojamiento

Este esquema...

- Usa claves significativas (son 'datos útiles')
- Para la mayoría de las consultas sobre oficios y alojamientos, el SGBD sólo necesita reunir las dos primeras tablas
- Si la consulta se realiza sobre columna indexada, el SGBD no necesita acceso a la tabla: sólo accede al índice

② TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

Diseño típico para aplicaciones con gran cantidad de datos en las tablas

EMPLEADO(codEmpleado, nombre, edad, codAlojamiento)
OFICIO_EMPLEADO(codEmpleado, codOficio, calificacion)
OFICIO(codOficio, oficio, descripcion)
ALOJAMIENTO(codAlojamiento,alojamiento,nombreCompleto,direccion,responsable)

Este esquema...

- Usa claves NO significativas
  - secuenciales,
  - sin significado específico,
  - usadas como 'meros conectores' entre tablas
- para la mayoría de las consultas sobre oficios y alojamientos, necesita reunir las cuatro tablas!!
- si la consulta se realiza sobre columna indexada, el SGBD necesita acceso a índice (búsqueda por clave) y a tabla (recuperación de datos)

② TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

¿Qué claves deben ser significativas?

- Detección en la fase de Análisis de Requisitos
- El proceso de **normalización** (análisis-diseño) puede restar significado a las claves

- Pero usar claves significativas **no es una desnormalización**, pues...
  - La Integridad Referencial se mantiene
  - No se producen Anomalías de Actualización

② TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

- DESNORMALIZACIÓN REAL
  - Consultas o Transacciones frecuentes que necesitan reunir (join)
    - demasiadas tablas (tres o más), o
    - dos tablas con muchas tuplas

    se ejecutarán muy lentamente
  - Solución: Romper la 3FN (o FNBC o superior)
    - Almacenar atributos de una tabla en otra tabla
    - REPETICIÓN, DUPLICACIÓN o REDUNDANCIA de atributos
  - Importante: las transacciones que actualizan el atributo deben MANTENER LA CONSISTENCIA entre los duplicados (evitar anomalías)
    - Procedimientos almacenados, Disparadores (Triggers), Aserciones
    - Redundancia controlada
- DESNORMALIZACIÓN EXTREMA

  Almacenar una TABLA con el RESULTADO de la REUNIÓN de las tablas

  ⇒ Resolver explícitamente las ANOMALÍAS DE ACTUALIZACIÓN

② TÉCNICAS para ACELERAR la REUNIÓN de TABLAS

Ejemplo de esquema de base de datos no normalizado -- desnormalización real

El análisis de requisitos indica que siempre que se necesita el perfil de un trabajador, se requiere el nombre del responsable de donde esté alojado el trabajador.

EMPLEADO(<u>nombre</u>, edad, alojamiento, responsable)
OFICIO_EMPLEADO(<u>nombre, oficio</u>, calificacion)
OFICIO(<u>oficio</u>, descripcion)
ALOJAMIENTO(<u>alojamiento</u>, nombre-completo, direccion, responsable)

Este esquema...

- INSERCIÓN NO DIRECTA en la tabla EMPLEADO: al insertar un empleado ⇒ el nombre del responsable debe 'venir' desde la tabla ALOJAMIENTO, según el valor de "alojamiento" para el nuevo empleado
- MODIFICACIÓN del nombre de un responsable en la tabla ALOJAMIENTO, o cambio de responsable para un alojamiento ⇒ modificación de tuplas EMPLEADO