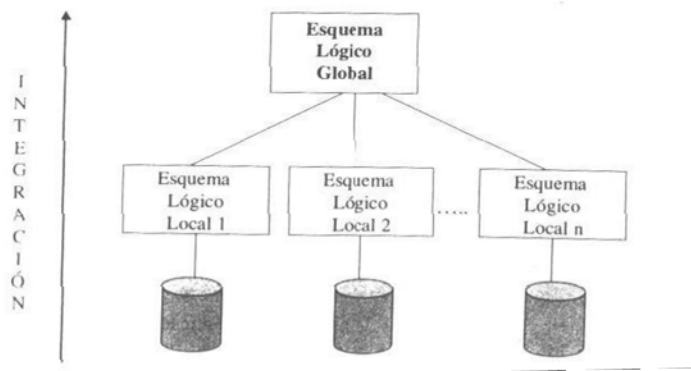


DISEÑO

- Ascendente (Bottom-Up)
- Descendente (Top-Down)

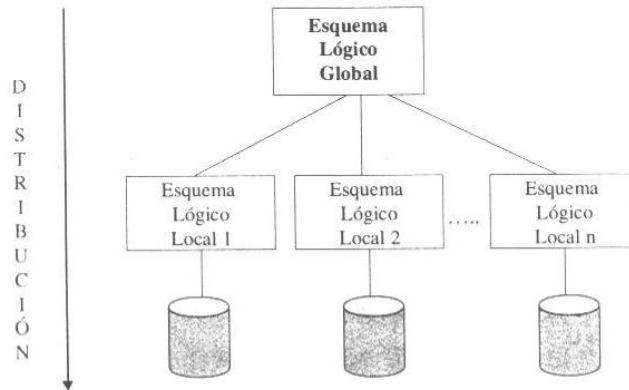
DISEÑO

- M. Ascendente se utiliza, en general, cuando existen varias BD locales y deseamos construir una BDD. Se parte de distintos esquemas lógicos locales (ELL) que se corresponden a BD ubicadas en diferentes nodos de una red y se integran, parte de ellos o todo, en un único esquema lógico global (ELG)
- Puede ser aplicable para unir distintas BD existentes, pero puede ser el caso donde se parte de cero y se avanza en el desarrollo de la BDD.



DISEÑO

- M. Descendente parte de un esquema lógico global (ELG) y construye los distintos esquemas lógicos que se definen a partir de los esquemas de fragmentación, asignación y replicación, los cuales determinan la distribución de los datos a través de los nodos de la red.



Prepared by: Henry Echeverría
Desarrollo de Base de Datos

3.3

Cuadra, Castro, Iglesias, Martínez, Calle, Harith, Moreno

DISEÑO

- Partimos de un esquema lógico global (esquema o grafo relacional) de la BD que queremos distribuir y a partir de éste construimos los esquemas de fragmentación, de asignación y replicación de los datos para distribuirlos en los diferentes nodos de la red.
- Los esquemas de fragmentación se basan en el análisis de los datos utilizados por las distintas aplicaciones que acceden a la BD para crear relaciones más pequeñas y más adaptadas a las operaciones de recuperación y actualización que las aplicaciones utilizan, es decir, tener los datos divididos según la utilización que de ellos se hace.
- En los esquemas de asignación y replicación se fija desde qué nodo se demandan los datos y el tipo de operación que se realiza (de consulta o actualización), para que estas operaciones se puedan llevar a cabo de forma local y minimizar de esta forma el tráfico por la red.

Prepared by: Henry Echeverría
Desarrollo de Base de Datos

3.4

Cuadra, Castro, Iglesias, Martínez, Calle, Harith, Moreno

DISEÑO

- La replicación o duplicación se puede realizar cuando desde distintos nodos se requiere al misma información, si además las operaciones son de consulta no existe ningún problema en duplicar los datos. Ahora bien, si lo que hacemos es actualizarlos el SGBD debe asegurar que todas las copias son consistentes, es decir, que todas mantienen la misma información, por ejemplo, con mecanismos de propagación que actualicen todas las copias de los datos modificados, con el alto costo que esto conlleva. Por tanto, debemos llegar a una solución de compromiso y analizar VENTAJAS y DESVENTAJAS de replicar los datos.

DISEÑO

ESQUEMA DE FRAGMENTACION

El esquema de fragmentación se compone de las distintas relaciones en las que hemos dividido una relación proveniente del Esquema Lógico Global junto con la condición empleada para esta división, expresado todo ello en álgebra relacional.

RAZONES

- Las aplicaciones de BDD normalmente funcionan con vistas, por lo tanto se pueden emplear distintas relaciones en distintos nodos para formar la unidad distributiva
- Permite obtener mayor eficiencia porque los datos se almacenan cerca del nodo donde se utilizan con mayor frecuencia
- Permite aumentar el grado de concurrencia porque la fragmentación de las relaciones permite que una transacción pueda dividirse en subconsultas que operan sobre estos fragmentos
- Proporciona más seguridad porque los datos no requeridos por un nodo local no se almacenen en él, por lo tanto no están disponibles para personas no autorizadas.



DISEÑO

ESQUEMA DE FRAGMENTACION

Semántica de los atributos

- Hay que considerar que la fragmentación podría afectar el rendimiento del SGBDD, especialmente cuando se utilizan distintos fragmentos ubicados en distintos nodos para construir una vista.
 - El control de la semántica de los datos puede ser más complicado por la fragmentación
-
- Cuando se agrupan atributos para formar un esquema de relación, se supone que hay un cierto significado asociado a los atributos
 - Este significado o semántica, especifica cómo se han de interpretar los valores de los atributos almacenados en una tupla de la relación (que relación hay entre los valores de los atributos de una tupla)
 - Cuánto más fácil sea explicar la semántica de la relación, mejor será el diseño del esquema correspondiente

Prepared by: Henry Echeverría
Desarrollo de Base de Datos

3.7

Cuadra, Castro, Iglesias, Martínez, Calle, Harith, Moreno



DISEÑO

ESQUEMA DE FRAGMENTACION

Para asegurar que la BD no sufrirá cambios semánticos durante la fragmentación de los datos, se definen tres normas que determinan la calidad de la fragmentación de una relación:

- 1.- COMPLETITUD. Todos los datos de una relación fragmentada han de encontrarse en al menos un fragmento.
- 2.- DISYUNCION.- Los fragmentos deberán ser disjuntos, es decir, los datos que aparecen en un fragmento no deben aparecer en otro; excepto las claves primarias donde sí pueden aparecer más de una vez como en el caso de la fragmentación vertical
- 3.- RECONSTRUCCION.- Siempre se ha de poder reconstruir la BD global a partir de los fragmentos.

Prepared by: Henry Echeverría
Desarrollo de Base de Datos

3.8

Cuadra, Castro, Iglesias, Martínez, Calle, Harith, Moreno



DISEÑO

ESQUEMA DE FRAGMENTACION

- Fragmentación Horizontal
- Fragmentación Vertical
- Fragmentación Mixta



DISEÑO

FRAGMENTACION VERTICAL

La fragmentación vertical divide la relación R en conjuntos de columnas, así cada fragmento mantiene ciertos atributos de la relación original. La fragmentación vertical se realiza mediante el operador algebraico de proyección y su notación es la siguiente:

$R_i = \Pi_{L_i}(R)$ donde $i = 1..n$ y R_i es el conjunto de fragmentos en que se divide la relación original R

L_i es la condición por la que fragmentamos, en este caso al ser una proyección se trata de un subconjunto de atributos de R. Para que se cumplan las condiciones anteriormente descritas sobre una correcta fragmentación los subconjuntos de atributos L_i han de cumplir:

DISEÑO

FRAGMENTACION VERTICAL

$$R_i = \Pi_{L_i}(R)$$

- La unión de todos los L_i nos da todos los atributos de R .
- La intersección de todos los L_i nos da la clave primaria de la relación R . Esto supone que todos los L_i tienen en común la clave primaria para de esta forma poder reconstruir a partir de los fragmentos, R_i , la relación inicial, R , mediante la operación de combinación.

DISEÑO

FRAGMENTACION VERTICAL

Veamos con un ejemplo la aplicación de este tipo de fragmentación. Supongamos que tenemos una base de datos centralizada de unos grandes almacenes y que una de sus relaciones es:

EMPLEADOS.(apellidos, nombre, dirección, teléfono, función, localización, extensión)

Se quieren distribuir los datos de EMPLEADOS teniendo en cuenta que se tienen dos sedes o nodos enlazados por una red de área local y que en el NODO1 se encuentra el departamento de contabilidad y desde aquí se manejan los datos personales del empleado (nombre, apellidos, dirección, teléfono y función), mientras que en el NODO2 se encuentra centralita y en este nodo se consultan los datos correspondientes a la localización del empleado (localización, y extensión).



DISEÑO

FRAGMENTACION VERTICAL

Esquema de fragmentación:

Fragmentaremos verticalmente la tabla EMPLEADOS creando dos nuevas relaciones:

EMPLEADO_CONTABILIDAD =

$\Pi_{\text{apellidos, nombre, dirección, teléfono, función}}(\text{EMPLEADOS})$

EMPLEADO_CENTRALITA = $\Pi_{\text{apellidos, nombre, localización, extensión}}(\text{EMPLEADOS})$

La combinación de ambas relaciones proporcionará la relación original:

EMPLEADOS =

EMPLEADO_CONTABILIDAD *_{apellidos, nombre} EMPLEADO_CENTRALITA



DISEÑO

FRAGMENTACION HORIZONTAL

Este tipo de fragmentación divide una relación en subconjuntos de tuplas, cada uno de ellos con un significado lógico.

Existen dos tipos de fragmentación horizontal: primaria o derivada.

- La fragmentación horizontal primaria se define como una selección de la relación R y su notación es la siguiente: $R_i = \sigma_{P_i}(R)$ donde P_i es un predicado sobre uno o más atributos de R y es la condición empleada para seleccionar el contenido de los fragmentos.

DISEÑO

FRAGMENTACION HORIZONTAL

- La fragmentación horizontal derivada se realiza en función de predicados definidos sobre atributos de otras relaciones o fragmentos; esto se debe a que la relación R a fragmentar, depende de la relación Q, sobre cuyos atributos está definido el predicado de la fragmentación. Se define empleando el operador relacional de semicombinación, que actúa de la misma forma que el operador de combinación, pero la relación resultante sólo tiene los atributos del primer operando.

$R_i = R \bowtie Q_i$ donde Q_i corresponde al conjunto de fragmentos en los que se ha dividido la relación Q. La semicombinación se realiza por el atributo o atributos que relacionan estas dos tablas.

Para que la fragmentación sea correcta se tiene que cumplir que la unión de todos los R_i sea la relación R y que su intersección sea vacía.

DISEÑO

FRAGMENTACION HORIZONTAL

Supongamos que al ejemplo anterior le añadimos la siguiente especificación:

Cada producto, comercializado por los grandes almacenes, se encuentra en un almacén. No hay productos que se encuentren en más de un almacén. Se sabe que los almacenes están divididos en áreas (norte, sur, este y oeste) y que además de los nodos de centralina y contabilidad hay otros cuatro nodos más, cada uno de ellos se encuentra en una de estas áreas, y en cada una de estas sedes sólo se maneja la información referente a los productos almacenados en dichas áreas.

Por lo que además de la tabla EMPLEADO tendremos las siguientes relaciones:

PRODUCTOS (cód_pród, nombre, descripción, almacén)

ALMACÉN (cód_almacén, dirección, teléfono, área)

DISEÑO

FRAGMENTACION HORIZONTAL

Dado que en cada almacén sólo se requiere información de los productos almacenados en el área en que se encuentra, podemos realizar una fragmentación para repartir la información que se encuentra en la relación PRODUCTOS. Para ello haremos una fragmentación horizontal primaria en la relación ALMACÉN:

$$ALMACÉN_i = \sigma_{\text{área} = i} (ALMACÉN) \text{ donde } i = \{\text{norte, sur, este, oeste}\}$$

Cada uno de estos fragmentos contiene información de los almacenes ubicados en cada una de las áreas.

DISEÑO

FRAGMENTACION HORIZONTAL

Si realizamos una fragmentación horizontal derivada en la relación PRODUCTOS:

$$PRODUCTOS_i = PRODUCTOS \bowtie_{\text{almacen}=\text{cod_almacen}} ALMACÉN_i$$

Conseguiremos la información de los productos almacenados en cada área.

Las relaciones originales se obtendrán mediante la unión de los fragmentos.

$$ALMACÉN = ALMACÉN_{\text{norte}} \cup ALMACÉN_{\text{sur}} \cup ALMACÉN_{\text{este}} \cup ALMACÉN_{\text{oeste}}$$

$$PRODUCTOS =$$

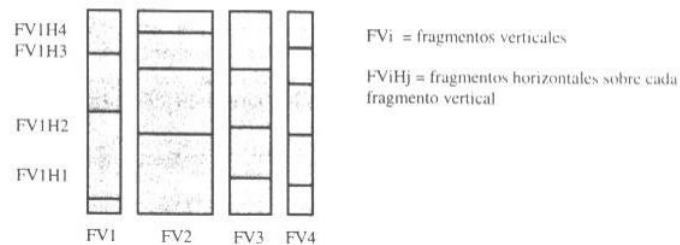
$$PRODUCTOS_{\text{norte}} \cup PRODUCTOS_{\text{sur}} \cup PRODUCTOS_{\text{este}} \cup PRODUCTOS_{\text{oeste}}$$

DISEÑO

FRAGMENTACION MIXTA

En algunos casos la fragmentación horizontal o vertical del esquema de la base de datos no es suficiente para satisfacer los requisitos de las aplicaciones de usuario, por lo que puede necesitarse volver a fragmentar los fragmentos obtenidos. Esto es lo que se denomina fragmentación mixta y se clasifica de la siguiente forma:

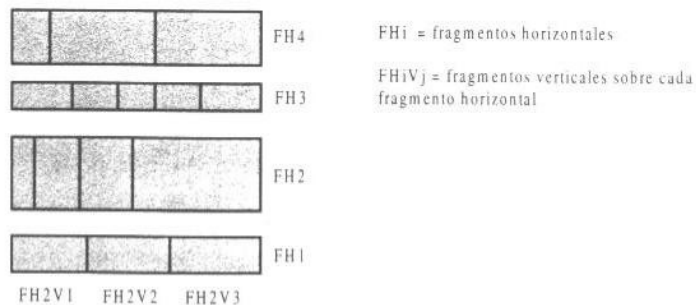
Fragmentación VH. Una fragmentación vertical seguida de una fragmentación horizontal, sobre cada uno de los fragmentos verticales.



DISEÑO

FRAGMENTACION MIXTA

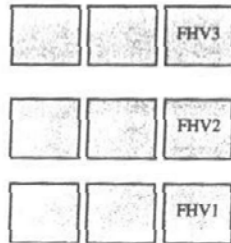
Fragmentación HV. Una fragmentación horizontal seguida de una fragmentación vertical, sobre cada uno de los fragmentos horizontales.



DISEÑO

FRAGMENTACION MIXTA

Red de celdas. Aplicamos simultáneamente una fragmentación vertical y horizontal.



FHVi = fragmento horizontal y vertical

DISEÑO

ESQUEMA DE ASIGNACION Y REPLICACION

El esquema de asignación y replicación consiste en la realización de la correspondencia entre los fragmentos y los nodos que constituyen la red de comunicaciones de la BDD. Dicha correspondencia debe hacerse de la manera más óptima, teniendo en cuenta los dos parámetros siguientes:

Mínimo coste: minimiza el coste de almacenamiento de cada fragmento en el nodo correspondiente, el coste de modificar un fragmento que está replicado en distintos nodos y el coste de la transferencia de datos por la red.

Rendimiento: minimiza los tiempos de respuesta y maximiza la capacidad de procesamiento del sistema en cada nodo.

Aunque existen algoritmos para realizar la asignación teniendo en cuenta estos parámetros, resultan complejos y no proporcionan la solución óptima sino simplemente una solución.

DISEÑO

ESQUEMA DE ASIGNACION Y REPLICACION

La asignación lo realizaremos buscando siempre el procesamiento local de los datos y sólo utilizar replicación cuando la operación que se realice desde el nodo que solicita los datos sea siempre de consulta

DISEÑO

ESQUEMA DE ASIGNACION Y REPLICACION DEL EJEMPLO

