

TEMA 2  
DISEÑO DE BASES DE DATOS DISTRIBUIDAS.

Material para revisar en clase.

1. PROCESO DE DISEÑO TOP-DOWN.

Análisis de requerimientos.

- Se definen las necesidades en términos de:
  - Desempeño
  - Confiabilidad
  - Disponibilidad
  - Costos
  - Escalabilidad.

Diseño de la vista.

- Se definen las interfaces para los usuarios finales

Diseño conceptual

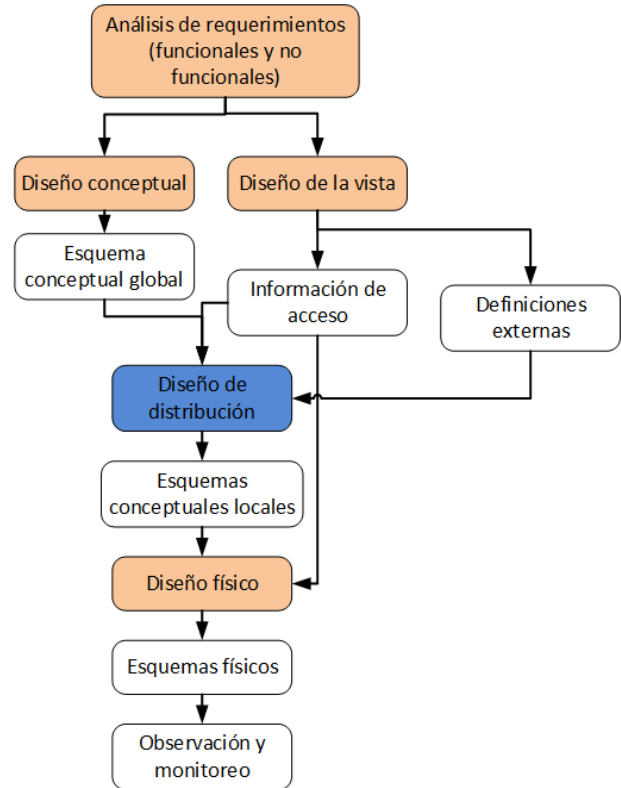
- Similar al diseño conceptual en un DBMS centralizado.

Diseño de distribución:

- En este paso se definen los esquemas conceptuales locales (LCS) en términos de fragmentos.
- De lo anterior, el diseño de distribución se enfoca en 2 actividades:
  - Fragmentación
  - Asignación.

Diseño físico.

- Similar al diseño físico de una BD centralizada, pero enfocado a cada uno de los sitios. En este diseño se definen las estructuras físicas de almacenamiento físico de los datos. P.E. Tablespace, grupos de discos, etc.



2. ALTERNATIVAS PARA DECIDIR LA UBICACIÓN DE CADA FRAGMENTO.

- Replicación. Llenar la siguiente tabla:

	Replicación completa	Replicación parcial	Particionamiento. Se le llama particionamiento cuando no existe replicación. Solo existe una copia por fragmento.
Procesamiento de consultas			
Administración del directorio			
Control de concurrencia.			
Confiabilidad			

- ¿Qué otras alternativas existen?

3. CONSTRUCCIÓN DEL MODELO RELACIONAL PARA EJEMPLOS DE FRAGMENTACIÓN.

Considere las siguientes reglas de negocio, construir un modelo relacional global. Emplear notación Crow's Foot

Una empresa decide distribuir los datos de sus empleados y los datos de los proyectos en los que trabajan. Se almacena nombre, apellido paterno, apellido materno y el puesto asignado al empleado. La empresa cuenta con un catálogo de empleados que contiene el nombre del puesto y su salario mensual.

Para cada uno de los proyectos que se desarrollan en la empresa se almacena el nombre del proyecto, el presupuesto asignado y la ubicación donde se va a desarrollar el proyecto (clave del país, no se requiere crear un catálogo). En un proyecto pueden participar varios empleados.

Para cada uno de los proyectos en los que participa un empleado se almacena la duración en horas que el empleado estará trabajando en el proyecto. Finalmente, para cada uno de los proyectos en los que participa un empleado, se almacena el rol que desempeña (no se requiere crear catálogo).

#### 4. EJEMPLO DE FRAGMENTACIÓN HORIZONTAL PRIMARIA.

Del resultado del análisis, se obtuvo que las consultas a la relación AE se comportan de la siguiente manera:

- La empresa decide distribuir los datos de la tabla en 2 grupos. El primer grupo de se ubicará en su sitio 1 y estará formado por todos aquellos registros que cumplan con todas las siguientes condiciones:
    - Se consultan los registros que tienen rol de 'admin o 'supervisor'.
    - Su duración en el proyecto asignado está entre 1 y 3 años.
  - El segundo grupo lo formarán todos los demás registros y estarán ubicados en el sitio 2.
- A. Generar la lista de predicados simples que son de interés para las condiciones anteriores.  
 B. Determinar los N fragmentos con sus respectivos predicados.  
 C. Determinar la expresión que permita realizar la reconstrucción de la Tabla.

#### 5. EJEMPLO FRAGMENTACIÓN HORIZONTAL DERIVADA

La empresa decide distribuir los datos de los empleados en 2 sitios con base a su sueldo.

- Sitio 1: empleados con sueldo  $\leq 25,000$
- Sitio 2: empleados con sueldo  $> 25,000$

A. Considerando la siguiente muestra de datos, construir 2 nuevas tablas que contengan los datos en cada sitio:

EMPLEADO

EMPLEADO_ID	NOMBRE	CLAVE_PUESTO
1	JUAN	DR
2	EVA	JD
3	MARIO	OP
4	LILI	DR
5	LUIS	JD
6	PEDRO	AT
7	IVAN	OP
8	IRMA	AT

PUESTO

SUELDO	NOMBRE	CLAVE_PUESTO
48000	DIRECTOR	DR
33000	JEFE DEPTO.	JD
15000	OPERATIVO	OP
18000	AUXILIAR TÉCNICO	AT

- B. Generar una expresión en álgebra relacional que determine los registros obtenidos en el punto anterior.  
 C. Revisar la expresión anterior para ver si esta puede ser optimizada de tal forma que se minimice la cantidad de datos a transmitir asumiendo que cada relación se encuentra en un sitio diferente.

#### 6. EJEMPLO DE SEMI-JOIN

A. Considerando las reglas de fragmentación del punto anterior, generar 2 sentencias SQL que obtengan los datos anteriores a través del uso del concepto de Semi – Join.

#### 7. EJEMPLO FRAGMENTACIÓN HORIZONTAL DERIVADA

La empresa ha decidido distribuir su BD en 5 sitios considerando las siguientes reglas:

- Los proyectos se administran con base a su ubicación: MX, BR y JAP. Existe un nodo en cada país. Por otro lado, los empleados se administran con base a su sueldo. Existen 2 aplicaciones, cada una instalada en un sitio (su ubicación es diferente a los 3 anteriores).
  - Aplicación 1: Empleados con sueldo  $\leq 20,000$
  - Aplicación 2: Sueldo  $\geq 20,000$
- A. Determinar el esquema de fragmentación para EMPLEADO (EMP), PUESTO (PU) y PROYECTO (PR)
- B. Determinar las posibles opciones que existen para fragmentar ASIGNACION\_EMPLEADO (AE).
- C. Para cada opción proponer un esquema de asignación empleado para ello un modelo relacional.

## 8. EJEMPLO FRAGMENTACIÓN VERTICAL

Considerar la relación CLIENTE y las características de los siguientes sitios:

- Se cuentan con 3 sitios:
  - Sitio 1: Servidor con amplia capacidad de almacenamiento y software especializado para procesar imágenes.
  - Sitio 2: Servidor principal de captura de datos generales.
  - Sitio 3: Servidor de seguridad empleado para almacenar los datos considerados como “delicados” o “confidenciales”.

CLIENTE		
CLIENTE_ID	NUMERIC(10,0)	NOT NULL
NOMBRE	VARCHAR(40)	NOT NULL
AP_PAT	VARCHAR(40)	NOT NULL
AP_MAT	VARCHAR(40)	NOT NULL
FOTO	VARBINARY/BLOB(max)	NOT NULL
NUM_TARJETA	VARCHAR(16)	NOT NULL
NUM_SEGURIDAD	VARCHAR(3)	NOT NULL

- Generar una propuesta de esquema de fragmentación
- Generar la expresión de reconstrucción.

## 9. EJEMPLO MATRIZ DE USO

Obtener  $uso(q_i, A_j)$  para las siguientes consultas que genera una aplicación.

Q1: select presupuesto from proyecto where proyecto\_id=X

Q2: select nombre, presupuesto from proyecto

Q3: select nombre, from proyecto where ubicación=X

Q4: select sum(presupuesto) from proyecto where ubicación = x

Asumir:

A1: proyecto\_id

A2: nombre

A3: presupuesto

A4: ubicación.

## 10. EJEMPLO MATRIZ DE AFINIDAD.

Obtener la afinidad entre los atributos A1 y A3.

- Considerar el número de accesos a los atributos por cada ejecución del query  $q_k$  y por sitio  $L$  es 1, es decir  $ref_L(q_k) = 1$
- Las frecuencias de acceso por query y por sitio son:

Sitio 1	Sitio 2	Sitio 3
$acc_1(q1) = 15$	$acc_2(q1) = 20$	$acc_3(q1) = 10$
$acc_1(q2) = 5$	$acc_2(q2) = 0$	$acc_3(q2) = 0$
$acc_1(q3) = 25$	$acc_2(q3) = 25$	$acc_3(q3) = 25$
$acc_1(q4) = 3$	$acc_2(q4) = 0$	$acc_3(q4) = 0$

## 11. DEFINICIÓN MATRIZ CLUSTERIZADA.

El siguiente paso para encontrar la afinidad entre atributos agruparlos para formar conjuntos de atributos llamados Clusters. Cada cluster formará un fragmento.

- A partir de la matriz de afinidad (AA), se consideran sus atributos y se reorganizan en un determinado orden para formar clusters.
- La característica que tienen cada cluster es que los valores de sus atributos presentan una afinidad alta entre ellos, lo que permite agruparlos y considerarlos como un fragmento de la tabla.
- Existe un algoritmo llamado Bond Energy Algorithm (BEA) con complejidad  $O(n^2)$  para realizar esta tarea.
- La salida del algoritmo es una nueva matriz llamada Matriz de Afinidad Clusterizada (CA) que se forma a partir de realizar permutaciones entre las columnas y renglones de la matriz AA
- Las permutaciones se realizan de tal manera que sea posible la identificación de este conjunto de clusters maximizando un valor llamado Medida de Afinidad Global (AM)
- AM es un valor que se produce al aplicar operaciones entre los valores de AA. Entre mayor sea el valor de AM mejor será la identificación de estos clusters.
- Para lograr un valor alto de AM se deberán realizar permutaciones entre columnas y renglones.

A nivel general, el valor AM se obtiene aplicando la siguiente fórmula:

$$AM = \sum_{i=1}^n \sum_{j=1}^n aff(A_i, A_j) [aff(A_i, A_{j-1}) + aff(A_i, A_{j+1}) + aff(A_{i-1}, A_j) + aff(A_{i+1}, A_j)]$$

Donde:

$$aff(A_0, A_j) = aff(A_i, 0) = aff(A_{n+1}, A_j) = aff(A_i, A_{j+1}) = 0$$

Estos casos corresponden para los atributos que se encuentran en la primera y última columna, así como en el renglón superior e inferior.

- En general la idea de esta fórmula es verificar afinidad de un valor  $aff(A_i, A_j)$  con sus valores vecinos a la izquierda, derecha (primeros 2 términos), arriba y abajo (últimos 2 términos).
- Debido a que AA es simétrica, AM se puede reducir a los 2 primeros términos:

$$AM = \sum_{i=1}^n \sum_{j=1}^n aff(A_i, A_j) [aff(A_i, A_{j-1}) + aff(A_i, A_{j+1})]$$

## 12. DESARROLLO DEL ALGORITMO BEA:

Para obtener CA se aplican los siguientes pasos:

- Inicializar CA con las 2 primeras columnas de AA
- Iteración: Seleccionar la siguiente columna de AA, tratar de ubicarla entre las columnas existentes de CA de tal manera que su posición aporte la mayor contribución para obtener el valor de AM. Continuar iterando hasta terminar con las columnas de AA.
- Ordenamiento de renglones. Con base al ordenamiento de columnas obtenido, aplicar el mismo orden a los renglones de CA.

Para determinar el valor de contribución de la columna se puede reestructurar la fórmula AM de la siguiente manera:

$$AM = \sum_{i=1}^n \sum_{j=1}^n [aff(A_i, A_{j-1})aff(A_i, A_j) + aff(A_i, A_{j+1})aff(A_i, A_j)]$$

Aplicando distributividad de la sumatoria externa con  $i$ :

$$AM = \sum_{j=1}^n \left[ \sum_{i=1}^n aff(A_i, A_{j-1})aff(A_i, A_j) + \sum_{i=1}^n aff(A_i, A_{j+1})aff(A_i, A_j) \right]$$

El algoritmo BEA define la función  $bond(A_x, A_y)$  que representa la medida de afinidad entre 2 atributos  $A_x$  y  $A_y$

$$bond(A_x, A_y) = \sum_{z=1}^n aff(A_z, A_x)aff(A_z, A_y)$$

Z representa a los renglones de AA.

- Observar que la definición de la función bond corresponde con la definición de AM, por lo que

$$AM = \sum_{j=1}^n [bond(A_j, A_{j-1}) + bond(A_j, A_{j+1})]$$

- ¿Qué sucede con el valor de AM si se mueve una columna  $A_k$  y se inserta entre un par de columnas contiguas  $A_i$  y  $A_j$  ?
  - R: El valor de AM cambiará debido a que dicho movimiento genera un valor de contribución (cont) nuevo, es decir:

$$cont(A_i, A_k, A_j) = bond(A_i, A_k) + bond(A_k, A_j) - bond(A_i, A_j)$$

- Observar que se resta el valor de contribución original que se tenía entre  $A_i$  y  $A_j$