

# CONCEPTOS DE BASES DE DATOS

## CLASE 7



# Dispersión

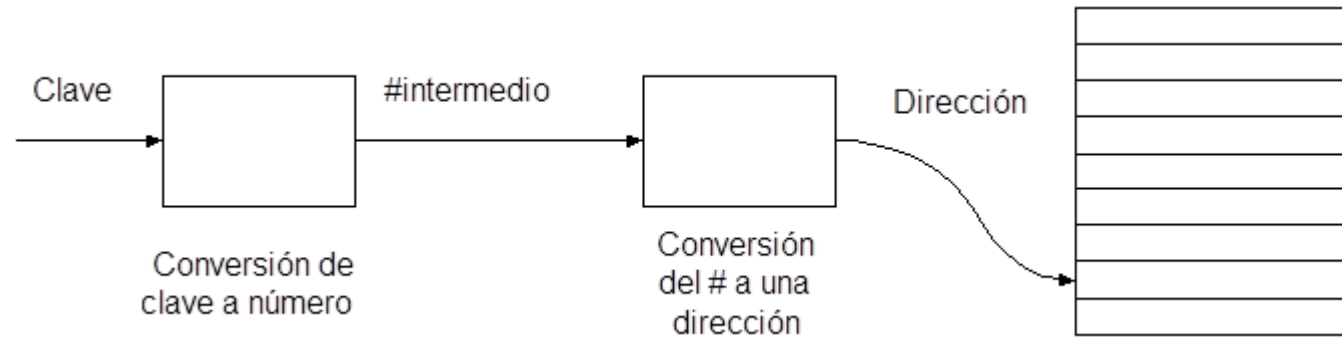
- Hasta el momento
  - Secuencial:  $N/2$  accesos (promedio)
  - Ordenado:  $\log_2 N$  accesos
  - Árboles: 3 o 4 accesos
- Sin embargo, para ciertos casos se necesita un mecanismo de acceso a registros con una única lectura → DISPERSIÓN (HASH)

# Dispersión

## Definiciones

- Técnica para generar una **dirección base única** para una clave dada
- Técnica que convierte la clave del registro en un **número aleatorio**, el que sirve después para determinar donde se almacena el registro
- Técnica de almacenamiento y recuperación que usa una **función de hash** para mapear registros en direcciones de almacenamiento

# Dispersión



- Para determinar la dirección de un registro:
  - La clave se convierte en un **número aleatorio**
  - El número se convierte en una **dirección de memoria**
  - El registro **se debe guardar en esa dirección**
    - Si la dirección está completa: **saturación (overflow)** → **tratamiento especial**

# Dispersión

- Beneficios

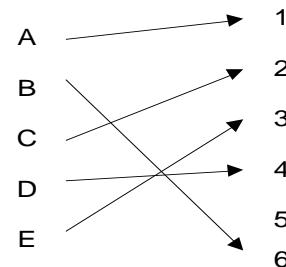
- No requiere almacenamiento adicional (índices)
- Facilita inserción y eliminación rápida de registros
- Encuentra registros con muy pocos accesos al disco en promedio (generalmente menos de 2)

- Costos

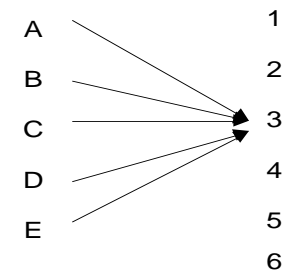
- No es posible el uso de registros de longitud variable
- No existe el orden físico de datos
- No permite claves duplicadas

# Dispersión

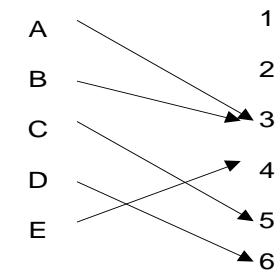
- Algoritmos de dispersión
  - **Uniforme**: reparte los registros en forma uniforme en el espacio de direcciones disponible → **difícil de lograr**
  - **Aleatoria**: las claves son independientes, no influyen una sobre la otra. Cualquier dirección tiene la misma probabilidad de ser elegida para una clave



uniforme



peor



aceptable



# Dispersión

- Tipos de dispersión
  - Con espacio de direccionamiento **estático**
    - El espacio disponible para dispersar los registros esta **prefijado**
  - Con espacio de direccionamiento **dinámico**
    - El espacio disponible para dispersar los registros **aumenta o disminuye según las necesidades** de espacio que en cada momento tiene el archivo

# Dispersión

Estático

- Parámetros que afectan la **eficiencia**
  - Función de dispersión
  - Tamaño de los compartimentos (espacio de almacenamiento)
  - Densidad de empaquetamiento
  - Método de tratamiento de saturación



- **Función de dispersión**
  - **Caja negra:** a partir de una clave se obtiene la dirección donde debe estar el registro
  - **Diferencias con índices**
    - No hay relación aparente entre la clave y la dirección
    - Dos claves distintas pueden transformarse en una misma dirección  
→ **colisión**
  - **Resultado:** retorna un valor aleatorio, que luego debe ser mapeado al rango de valores disponibles
  - **Ejemplo:** función simple usando códigos ASCII

- **Función de dispersión** → **ejemplos**
  - **Centros cuadrados:** la clave se multiplica por si misma y se toma los dígitos centrales, posteriormente se ajusta al espacio disponible.
  - **División:** la clave se divide por un nro aproximadamente igual a la cantidad de direcciones (número primo → tiende a distribuir residuos en forma más eficiente). Luego se obtiene el residuo.
  - **Desplazamiento:** se realiza un desplazamiento del número, se particiona y se suman las partes resultantes. Luego se ajusta al espacio disponible.
  - **Plegado:** se realiza el plegado del número, se suman las partes resultantes y adaptan al espacio de direcciones.
  - **Transformación de base:** la base del número se modifica y luego se toma el resultado del módulo.

# Dispersión

Estático

- Función de dispersión → Centros cuadrados
  - N= 7000 (direcciones)
  - Clave: 6 dígitos

Clave = 172148

Clave ^ 2 = 029634933904 → 3493 \* 0.7 = 2445

Para llevar al espacio asignado al archivo multiplico por 0.7, haciendo el calculo con la mayor clave posible: 7000/9999 = 0.7

# Dispersión

Estático

- Función de dispersión → División
  - N= 7000 (direcciones)
  - Clave: 6 dígitos

Clave = 172148

Clave mod 6997 = 4220 → El resultado siempre estará entre 0 y 6996.

# Dispersión

Estático

- Función de dispersión → Desplazamiento
  - N= 7000 (direcciones)
  - Clave: 8 dígitos

Clave = 20175973

Clave desplazada= 17207359 = 1720 + 7359 → 9079 \* 0.35 = 3177

Para llevar al espacio asignado al archivo multiplico por 0.35, haciendo el calculo con la mayor clave posible:  $7000/19998 = 0.35$

# Dispersión

Estático

- Función de dispersión → Plegado
  - N= 999 (direcciones)
  - Clave: 8 dígitos

Clave = 17207359

Clave plegada en 3 partes = 017 207 359 = 953 + 207 + 710 → 1870 \* 0.33  
= 617

Para llevar al espacio asignado al archivo multiplico por 0.33, haciendo el calculo con la mayor clave posible:  $999/2988 = 0.33$



# Dispersión

Estático

- Función de dispersión → Transformación de base
  - N= 100 (direcciones)
  - Clave: 3 dígitos

Clave = 453

Clave en base 11 = 382 →  $382 * 0.12 = 45$

Para llevar al espacio asignado al archivo multiplico por 0.12, haciendo el calculo con la mayor clave posible:  $100/829 = 0.12$

- Función de dispersión
  - ¿Cuál elegir?
  - Tomar algunas claves del problema y **simular el comportamiento** con algunos métodos, y luego elegir el que **mejor** se comporte
  - En general:
    - División mejor
    - Plegado, para claves muy largas

- Colisiones

- Cuando un registro es asignado a una dirección ya ocupada, se produce una **colisión**
  - A las claves que por dispersión se convierten en la misma dirección se las llama **sinónimos**

- **Alternativas**

- Generar un algoritmo de dispersión perfecto → sin colisiones
- Intentar minimizar la cantidad de colisiones

# Dispersión

Estático

- Colisiones → disminución
  - **Esparcir registros:** buscar métodos que distribuyan los registros de la forma **más aleatoria posible** entre las direcciones disponibles
    - Suma de códigos ASCII → **no es bueno**
    - Otras de las funciones vistas → **mejor**
  - **Usar memoria adicional:** distribuir pocos registros en muchas direcciones (ej: 75 registros en 1000 direcciones)
    - **Disminuye** las colisiones
    - **Desperdicia** espacio

# Dispersión

Estático

- Colisiones → disminución
  - Colocar más de un registro por dirección
    - Direcciones con **N** claves → mejoras notables
    - Las direcciones que pueden almacenar 1 o más registros se denominan **cubetas** o **compartimentos**
    - **Ejemplo:** archivo con direcciones de **512** bytes y cada registro a almacenar es de **80** bytes
      - Se puede almacenar hasta **6 registros por cada dirección** de archivo (cada dirección tolera **hasta 5 sinónimos**)

# Dispersión

Estático

- Tamaño de los compartimentos
  - A **mayor tamaño** del compartimento:
    - **Menor** probabilidad de saturación
    - **Mayor** fragmentación → espacios vacíos
    - Búsqueda **más lenta** dentro del compartimento



- Tamaño de los compartimentos
  - ¿Cuál es el tamaño adecuado del compartimento?
    - **Tamaño muy grande** → la recuperación de un registro es **muy lenta**
    - **Depende del sistema**: el tamaño máximo dependerá de las posibilidades de transferencia en operaciones de E/S → buffers SO

# Dispersión

Estático

- Densidad de empaquetamiento (DE)
  - No es relevante el tamaño total del archivo. La DE es la proporción de espacio asignado al archivo que **en realidad almacena registros**
    - Cantidad de **espacio que es usado** en un archivo
  - $DE = \frac{\text{número de registros del archivo}}{\text{capacidad total de las cubetas}}$
  - Si se cuenta con una DE menor:
    - **Menor** saturación
    - **Mayor** desperdicio de espacio

# Dispersión

## Estimación overflow

- Es posible realizar una **estimación de la saturación** analizando las características de su almacenamiento

- **N**= #cubetas, **C**= capacidad cubeta, **K**= #reg. del archivo

$$DE = \frac{K}{C \times N}$$

- ¿Cuál es la probabilidad de que una cubeta en particular reciba una cantidad determinada de registros?

# Dispersión

## Estimación overflow

- Estimación del N° de claves que recibe una cubeta

- **Dadas:**

- **A** = no utilizar una cubeta en particular
- **B** = utilizar una cubeta en particular

- Para **una clave:**

$$P(\mathbf{B}) = 1/N$$

$$P(\mathbf{A}) = 1 - P(\mathbf{B}) = 1 - 1/N$$

- Para **dos claves:**

$$P(\mathbf{BB}) = P(\mathbf{B}) * P(\mathbf{B}) = (1/N)^2$$

$$P(\mathbf{BA}) = P(\mathbf{B}) * P(\mathbf{A}) = (1/N) * (1 - 1/N)$$

(cada clave es independiente, función de hash uniforme)

# Dispersión

## Estimación overflow

- Estimación del N° de claves que recibe una cubeta

- $P(\text{secuencia}) = P(A)^{\#A} * P(B)^{\#B}$

- Para una secuencia determinada de  $K$  claves, la probabilidad de que  $I$  caigan en una cubeta es:

$$(1/N)^I * (1 - 1/N)^{K-I}$$

- La cantidad de formas de **combinar** esa secuencia es ( $K$  tomadas de a  $I$  combinaciones):

$$\frac{K!}{I! * (K - I)!}$$

# Dispersión

## Estimación overflow

- Estimación del N° de claves que recibe una cubeta
  - La probabilidad de que una cubeta reciba **I** elementos de los **K** disponibles es entonces:

$$P(I) = \underbrace{\frac{K!}{I! * (K-I)!}}_{\text{COMBINATORIO } \binom{K}{I}} * \underbrace{\left(\frac{1}{N}\right)^I * \left(1 - \frac{1}{N}\right)^{K-I}}_{\substack{P(B) * P(A) \\ I \text{ VECES } \quad K-I \text{ VECES}}}$$

- Fórmula equivalente acotada:

$$P(I) = \frac{(K / N)^I * e^{-(K / N)}}{I!}$$



# Dispersión

## Estimación overflow

- En general, si hay **N** direcciones → el nro esperado de direcciones con **I** registros asignados es igual a **N**\*P(**I**)
- Se analizarán archivos con diferentes **DE**, estimando la saturación en cada caso
- Ejemplo 1: **N = 10000** **K = 10000** **DE = 1** → **100%**
  - La proporción de direcciones con 0 registros asignados es:

$$P(0) = \frac{(10.000 / 10.000)^0 * e^{-(10.000/10.000)}}{0!} = 0.3679$$

# Dispersión

## Estimación overflow

- Ejemplo 1: **N** = 10000   **K** = 10000   **DE** = 1
  - Nº de direcciones **sin registros** asignados (**I** = 0):  
**0:**  $10000 * P(0) = 10000 * 0.3679 = 3679$
  - Nº de direcciones **con uno, dos y tres registros**:  
**1:**  $10000 * P(1) = 10000 * 0.3679 = 3679$   
**2:**  $10000 * P(2) = 10000 * 0.1839 = 1839$   
**3:**  $10000 * P(3) = 10000 * 0.0613 = 613$

# Dispersión

## Estimación overflow

- Ejemplo 1: **N = 10000** **K = 10000** **DE = 1**
  - La cantidad de registros en colisión será entonces:
    - 3679 direcciones tienen 0 registros asignados → 0
    - 3679 direcciones tienen 1 registro asignado → 0
    - 1839 direcciones tienen 2 registros asignados → 1839
    - 613 direcciones tienen 3 registros asignados → (613 \* 2)
  - Saturación estimada hasta el momento: 1839 + (613\*2) = 3065
    - Aún no se finalizó la estimación y ya es mayor a un 30%
    - Más del 30% de los registros serán almacenados en algún lugar que no es su dirección base

# Dispersión

## Estimación overflow

- Ejemplo 2: **N** = 1000   **K** = 500   **DE** = 0.5
  - Nº de direcciones **sin registros** asignados (**I** = 0) :  
**0**:  $1000 * P(0) = 1000 * 0.607 = 607$
  - Nº de direcciones **con un sólo registro**:  
**1**:  $1000 * P(1) = 1000 * 0.303 = 303$
  - Nº de direcciones **con más de un registro**:  
**2**:  $1000 * P(2) = 1000 * 0.0758 = 75$   
**3**:  $1000 * P(3) = 1000 * 0.0126 = 12$   
**4**:  $1000 * P(4) = 1000 * 0.0016 = 1$   
**5**:  $1000 * P(5) = 1000 * 0.0002 = 0$

# Dispersión

## Estimación overflow

- Ejemplo 2: **N** = 1000   **K** = 500   **DE** = 0.5
  - ¿Cuántos registros en saturación pueden esperarse?  
→ **N** \* ( 1\* P(2) + 2\* P(3) + 3 \* P(4) + 4 \* P(5) ) =  
= 1000 \* (1\* 0.0758 + 2\* 0.0126 + 3\*0.0016 + 4\*0.0002) =  
= **107**
  - Dado que se cuenta con 500 registros en total, la saturación estimada hasta el momento es: **107/500 = 0.214 = 21.4 %**
    - Más del **21%** de los registros serán almacenados en algún lugar que **no es su dirección base**

# Dispersión

## Estimación overflow

- Valores de **saturación** para diferentes **DE**

**DE**

**Saturación**

**0.10**

**4.8 %**

**0.30**

**13.6 %**

**0.50**

**21.4 %** → Ejemplo 2

**0.70**

**28.1 %**

**0.80**

**31.2 %**

**0.90**

**34.1 %**

**1.00**

**36.8 %** → Ejemplo 1



# Dispersión

## Estimación overflow

- En los ejemplos se utilizó compartimentos con capacidad para un **único registro**
  - En estos casos, los términos de colisión y saturación son **equivalentes**
- Los números bajos de saturación se presentan cuando se dispone de una baja DE
  - Disminuye la cantidad de colisiones
  - Deja muchas direcciones (compartimentos) libres
- **Solución** → usar compartimentos con capacidad para más de un registro

# Dispersión

## Estimación overflow

- Si se consideran compartimentos con capacidad para **más de un registro**:
  - Dependiendo de su **capacidad** y del **lugar libre** que tenga un compartimento en un momento dado, **dos sinónimos se podrán almacenar en la misma dirección**
  - Entonces ahora, al producirse una colisión:
    - Si el compartimento **tiene lugar suficiente** el registro se almacena en dicho lugar → **no hay saturación**
    - Si el compartimento **está completo** (no tiene más lugar) entonces si **se produce saturación**

# Dispersión

## Estimación overflow

**K**: Nº registros    **B**: Tamaño de compartimento

**N**: Nº de direcciones de memoria disponibles

$$DE = K / B * N$$

	Archivo 1	Archivo 2
Nº registros ( <b>K</b> )	750	750
Nº direcciones ( <b>N</b> )	1000	500
Tamaño Comp. ( <b>B</b> )	1	2
<b>DE</b>	0.75	0.75
Proporción regs/dirs	0.75	1.5

# Dispersión

## Estimación overflow

- Con 750 registros, 1000 direcciones de memoria disponibles y compartimentos de tamaño 1, el N° esperado de registros en saturación es:

$$\begin{aligned} \rightarrow N * ( 1 * P(2) + 2 * P(3) + 3 * P(4) + 4 * P(5) ) &= \\ = 1000 * (1 * 0.1328 + 2 * 0.0332 + 3 * 0.0062 + & \\ 4 * 0.0009 + 5 * 0.0001) & \\ = 222 & \end{aligned}$$

- Saturación =  $222 / 750 = 29.6 \%$

# Dispersión

## Estimación overflow

- Con 750 registros, 500 direcciones de memoria disponibles y compartimentos de tamaño 2, el N° esperado de registros en saturación es:

$$\begin{aligned} &\rightarrow N * (1*P(3) + 2*P(4) + 3*P(5) + 4*P(6) + 5*P(7)) = \\ &= 500 * (1*0.1255 + 2*0.0471 + 3*0.0141 + \\ &4*0.0035 + 5*0.0008) = 140 \\ &= \mathbf{140} \end{aligned}$$

- Saturación =  $140 / 750 = 18.7\%$  → mejora

# Dispersión

## Estimación overflow

- Probabilidad de saturación

	TAMAÑO COMPARTIMENTO				
DE	1	2	5	10	100
10%	4.8	0.6	0.0	0.0	0.0
20 %	9.4	2.2	0.1	0.0	0.0
30 %	13.6	4.5	0.4	0.0	0.0
40 %	17.6	7.3	1.1	0.1	0.0
50 %	21.4	10.4	2.5	0.4	0.0
60 %	24.8	13.7	4.5	1.3	0.0
70 %	28.1	17.0	7.1	2.9	0.0
75 %	29.6	18.7	8.6	4.0	0.0
80 %	31.2	20.4	10.3	5.3	0.1
90 %	34.1	23.8	13.8	8.9	0.8
100 %	36.8	27.1	17.6	12.5	4.0