

# **BD2**

# **OPTIMIZACIÓN**

**SERGIO ÁLVAREZ**

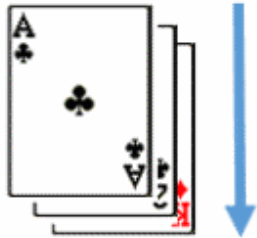
**VERSIÓN 1.0**

# TEORÍA

**BUSCAR  
PRODUCTO NATURAL (JOIN)**

# BUSCAR

## Random Pile of Cards

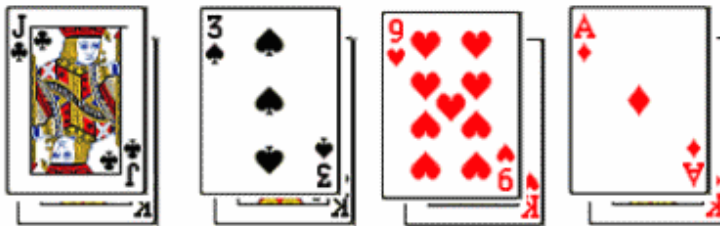


To Find 8 of Hearts...  
Scan 52 Cards (26)  
Average Flips = 26

```
public int busquedaLineal(int x){  
    for(int i=0; i<a.length; i++){  
        if(x==a[i]){  
            return i;  
        }  
    }  
}
```

## Random Pile of Cards By Suit

1. →



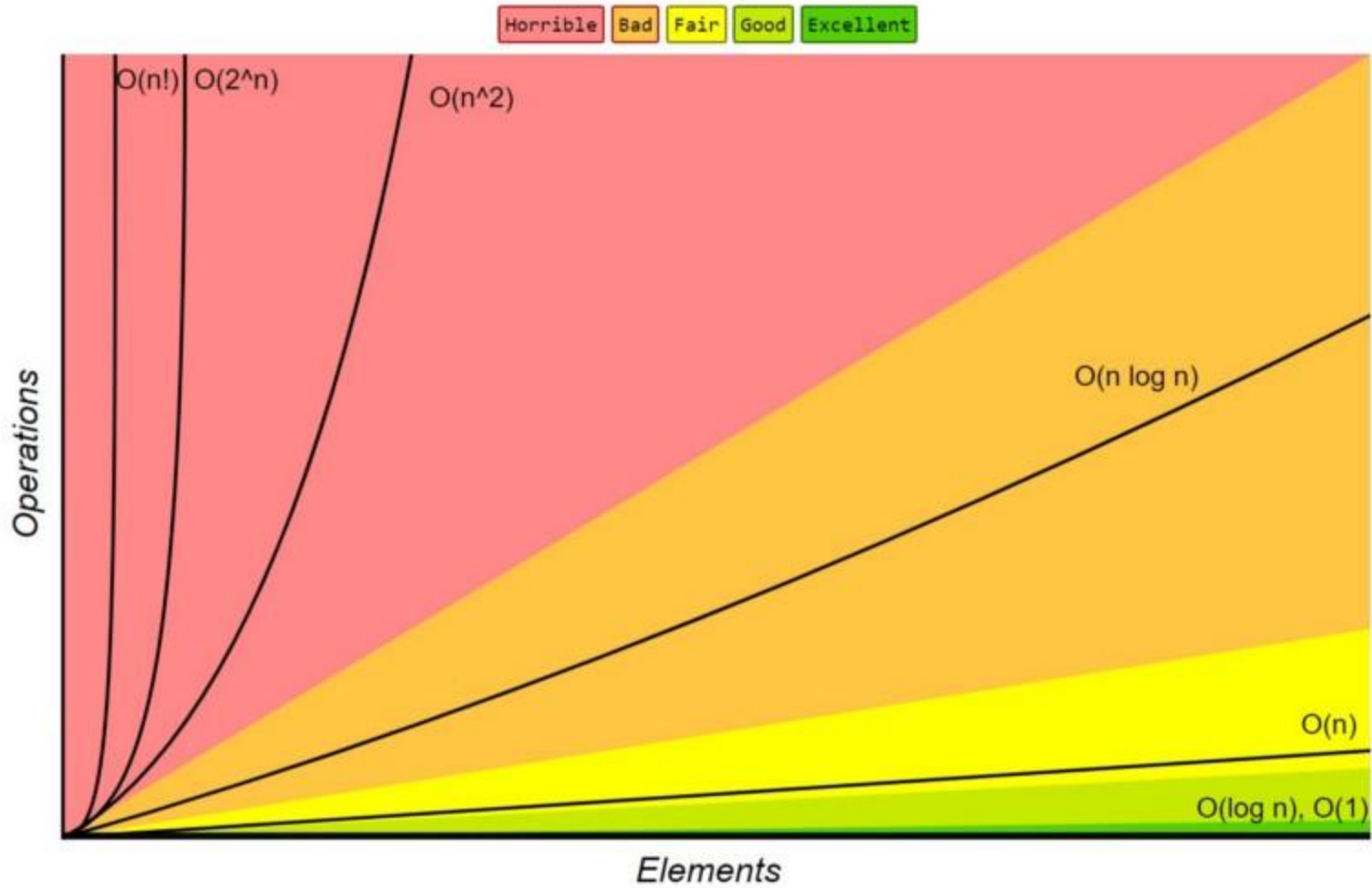
2. ↓

To Find 8 of Hearts...  
1. Pick Suit (2)  
2. Scan 13 Cards (7)  
Average Flips = 9

Creating Piles reduces flip from 26 to 9!

# BIG O

Big-O Complexity Chart



# BIG O

## Tiempo Constante ( $O(1)$ )

```
function exampleConstantFunc(n) {  
  return n*n;  
}
```

## Tiempo Lineal ( $O(n)$ )

```
function exampleLinear(n) {  
  for (var i = 0 ; i < n; i++) {  
    console.log(i)  
  }  
}
```

## Tiempo Logarítmico ( $O(\log(n))$ )

```
function log(n) {  
  for (let i = 1; i < n; i*=2) {  
    const result = i;  
    console.log(result);  
  }  
}
```

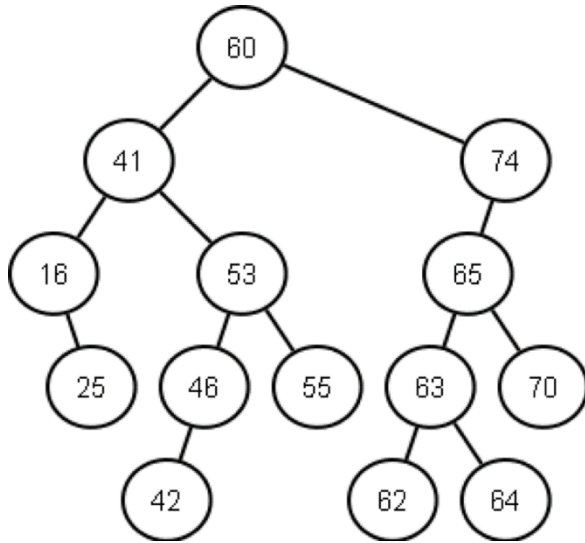
## Tiempo Cuadrático ( $O(n^2)$ )

```
for (int i = 0; i < n; i += c) {  
  for (int j = 0; j < n; j += c) {  
    // some  $O(1)$  expressions  
  }  
}
```

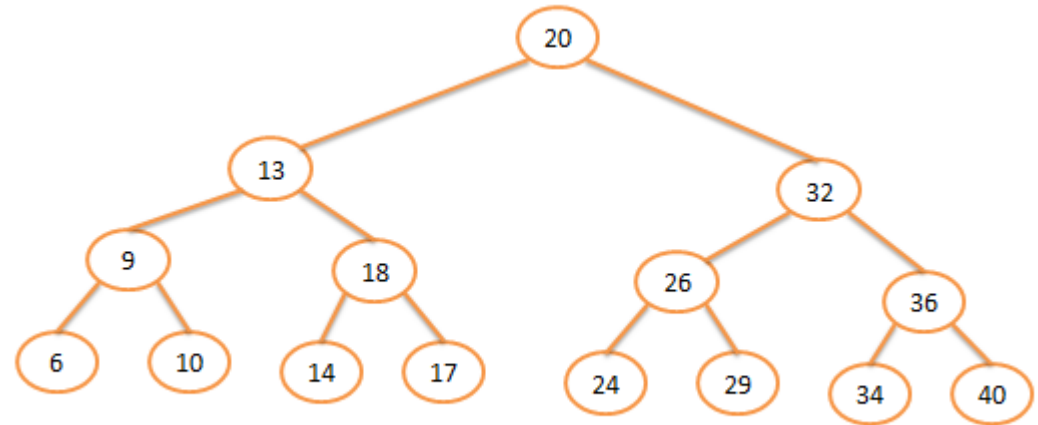
# MÉTODOS DE ORDENAMIENTO

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
<u>Quicksort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(\log(n))$
<u>Mergesort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
<u>Timsort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
<u>Heapsort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(1)$
<u>Bubble Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Insertion Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Selection Sort</u>	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Tree Sort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(n)$
<u>Shell Sort</u>	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$O(n(\log(n))^2)$	$O(1)$
<u>Bucket Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$	$O(n)$
<u>Radix Sort</u>	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$	$O(n+k)$
<u>Counting Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$	$O(k)$
<u>Cubesort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$

# ARBOLES PARA BUSCAR

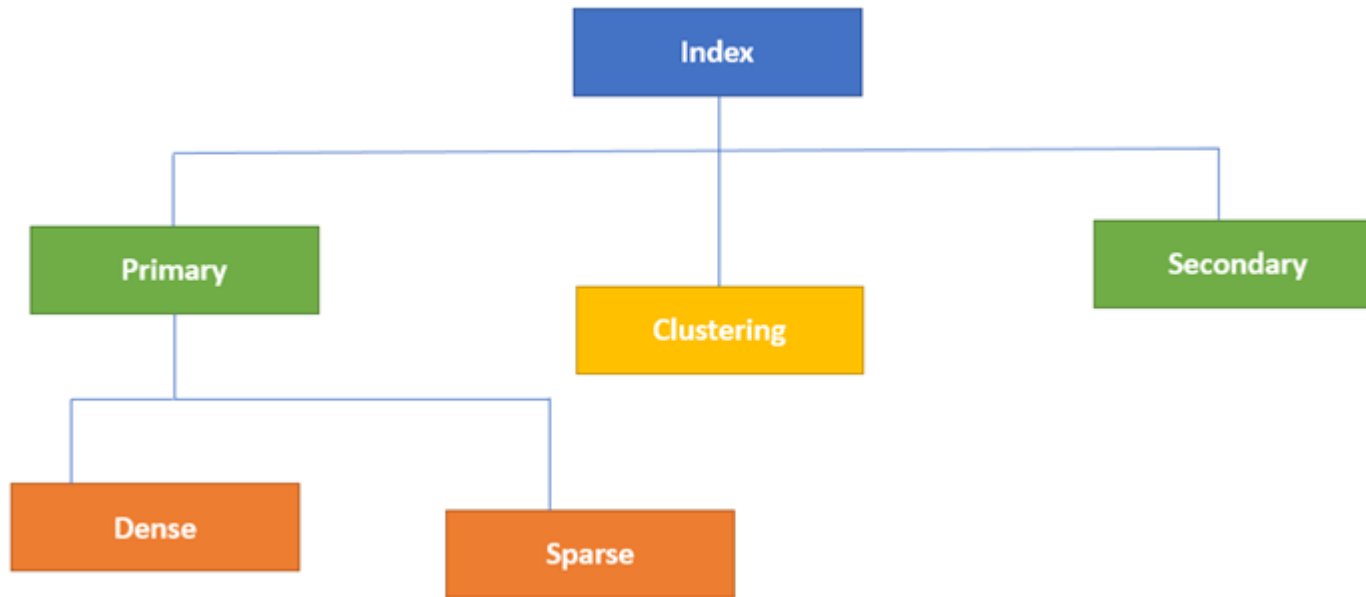


Árbol Binario  
5 Niveles



Árbol Binario Balanceado  
4 Niveles

# TIPOS DE ÍNDICES





# ÍNDICE CLÚSTER

SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice
43659	1	4911-403C-98	1	776	1	2024.994
43659	2	4911-403C-98	3	777	1	2024.994
43659	3	4911-403C-98	1	778	1	2024.994
43659	4	4911-403C-98	1	771	1	2039.994
43659	5	4911-403C-98	1	772	1	2039.994
43659	6	4911-403C-98	2	773	1	2039.994
43659	7	4911-403C-98	1	774	1	2039.994
43659	8	4911-403C-98	3	714	1	28.8404
43659	9	4911-403C-98	1	716	1	28.8404
43659	10	4911-403C-98	6	709	1	5.70
43659	11	4911-403C-98	2	712	1	5.1865
43659	12	4911-403C-98	4	711	1	20.1865
43660	13	6431-4D57-83	1	762	1	419.4589
43660	14	6431-4D57-83	1	758	1	874.794
43661	15	4E0A-4F89-AE	1	745	1	809.76

Non-Clustered

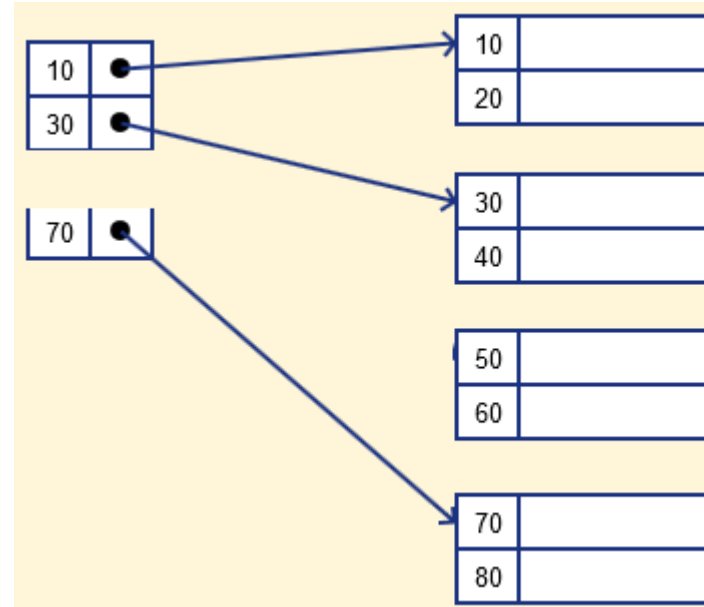
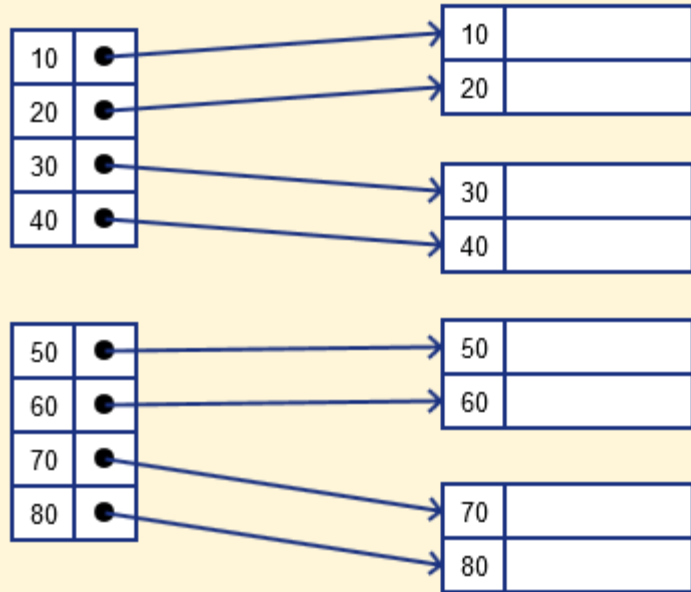
SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice
43659	1	4911-403C-98	1	776	1	2024.994
43659	2	4911-403C-98	3	777	1	2024.994
43659	3	4911-403C-98	1	778	1	2024.994
43659	4	4911-403C-98	1	771	1	2039.994
43659	5	4911-403C-98	1	772	1	2039.994
43659	6	4911-403C-98	2	773	1	2039.994
43659	7	4911-403C-98	1	774	1	2039.994
43659	8	4911-403C-98	3	714	1	28.8404
43659	9	4911-403C-98	1	716	1	28.8404
43659	10	4911-403C-98	6	709	1	5.70
43659	11	4911-403C-98	2	712	1	5.1865
43659	12	4911-403C-98	4	711	1	20.1865
43660	13	6431-4D57-83	1	762	1	419.4589
43660	14	6431-4D57-83	1	758	1	874.794
43661	15	4E0A-4F89-AE	1	745	1	809.76

Clustered Index

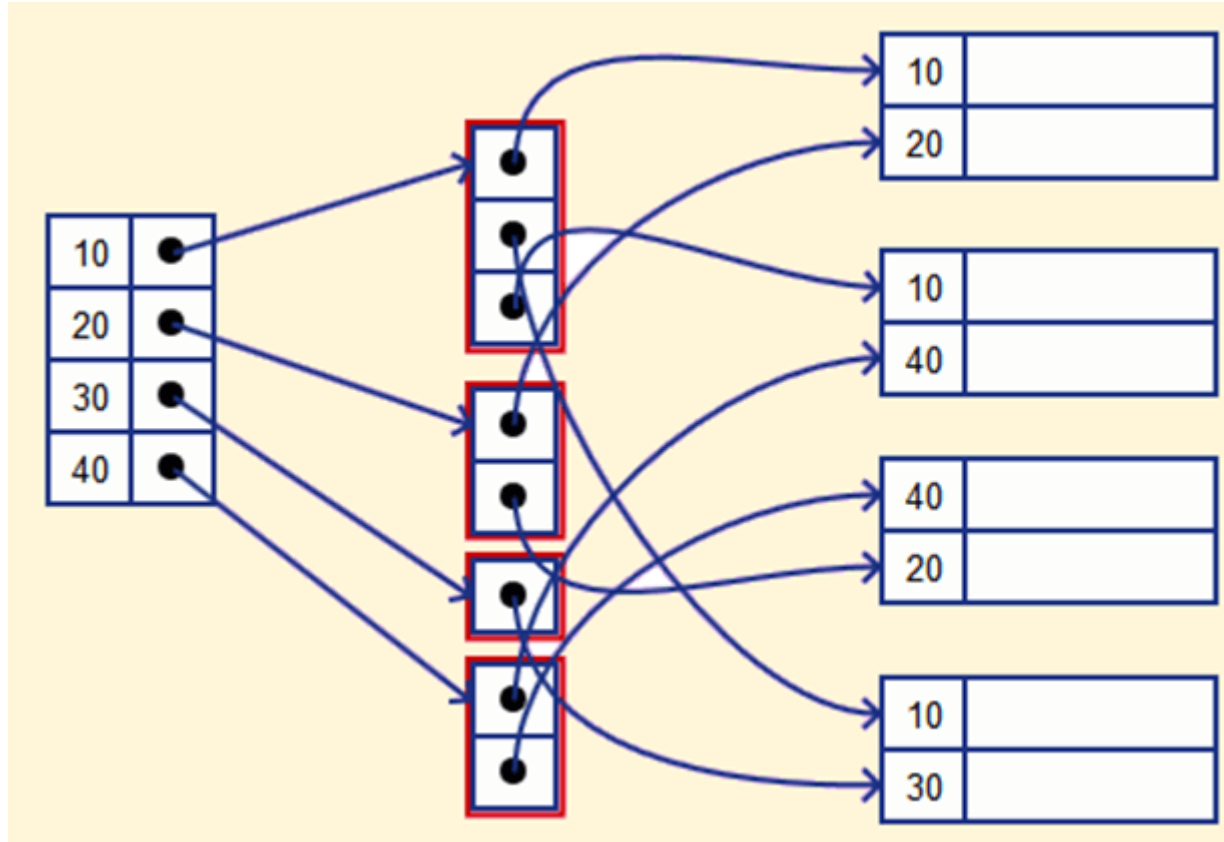
# ÍNDICE CLÚSTER

Parameters	Clustered	Non-clustered
Use for	You can sort the records and store clustered index physically in memory as per the order.	A non-clustered index helps you to creates a logical order for data rows and uses pointers for physical data files.
Data accessing	Faster	Slower compared to the clustered index
Additional disk space	Not Required	Required to store the index separately
Type of key	By Default Primary Keys Of The Table is a Clustered Index.	It can be used with unique constraint on the table which acts as a composite key.

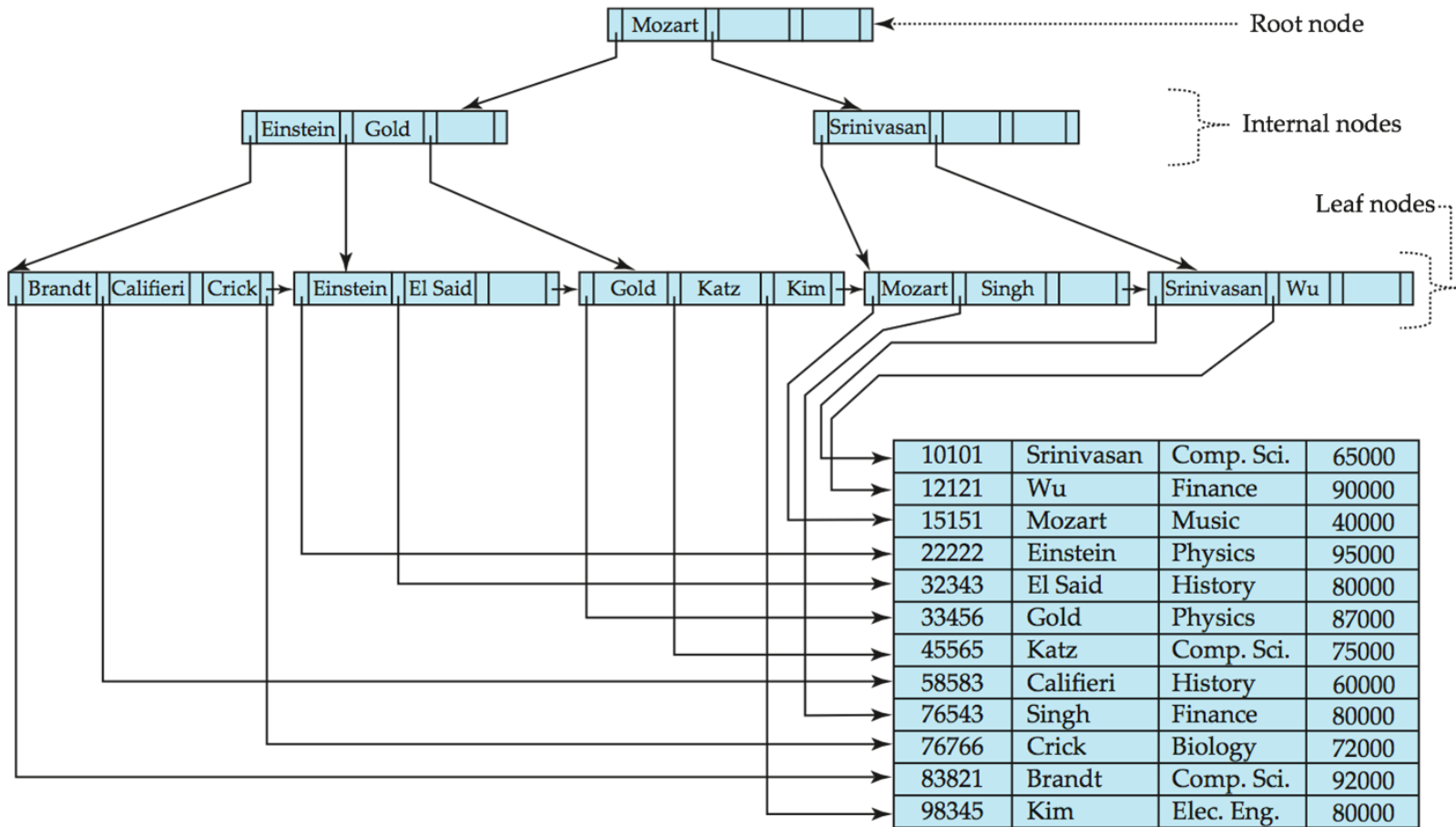
# DENSO VS DISPERSO



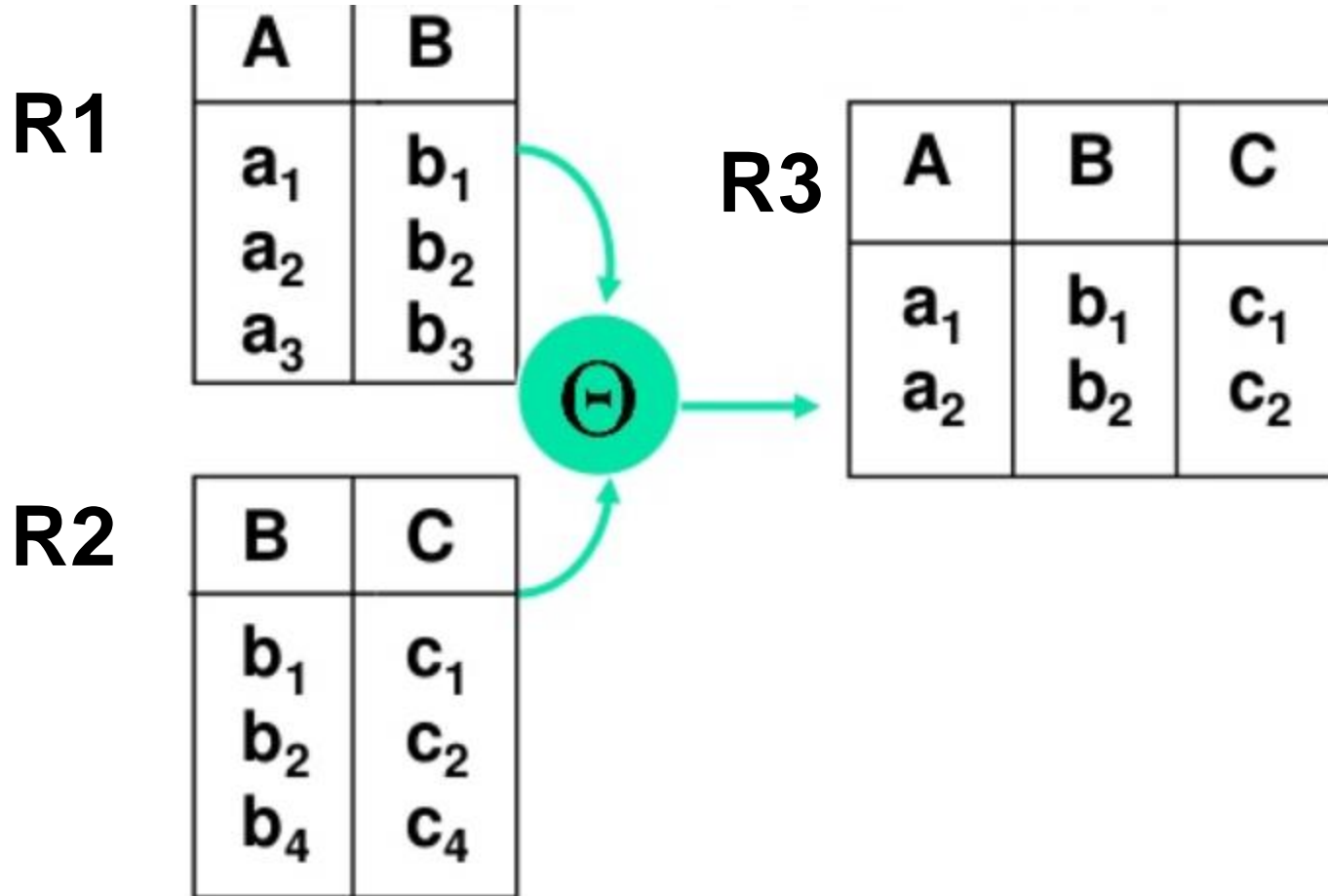
# ÍNDICE SECUNDARIO



# BTREE



# PRODUCTO NATURAL



# NESTED LOOP JOIN (1/3)

1. Scan CUSTOMER table for C\_NAME starting with "B"

000002	ASHLEY
000005	BULLOCK
000007	CAMPBELL
000006	BATES
000001	PRESCOTT
000014	GEESLIN
000008	CARTER
000073	CORLISS

2. Scan ORDERS table for rows with matching CUSTKEYs

569	000005	19980119
570	000004	19980120
571	000005	19980120
572	000006	19980120
573	000002	19980120
574	000005	19980121
573	000001	19980121
574	000006	19980121

3. Keep only if  
ORDERDATE  
= 19980120

000005	BULLOCK	571	19980120
000006	BATES	572	19980120

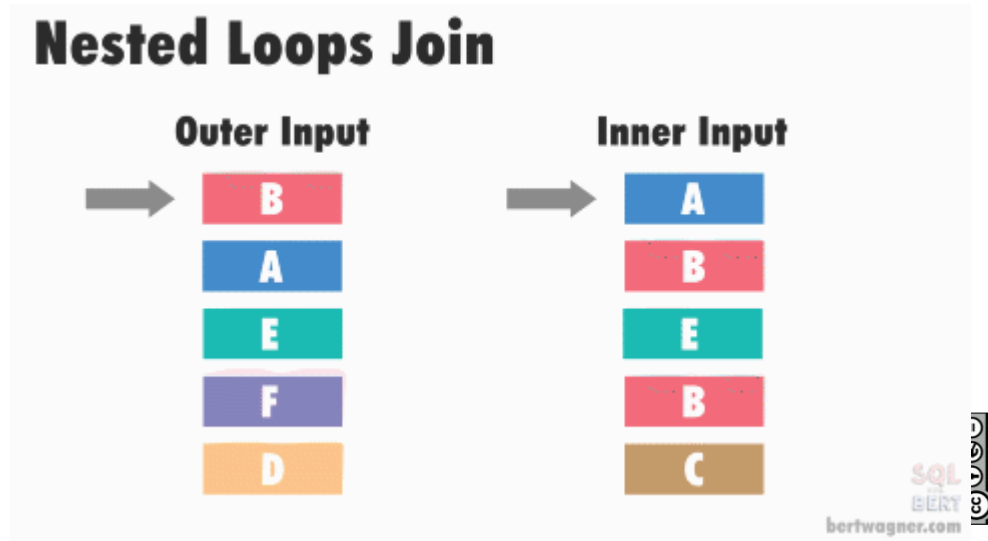
4. Sort on C\_NAME

5. Rows  
Returned  
to User

000006	BATES	572	19980120
000005	BULLOCK	571	19980120

# NESTED LOOP JOIN (2/3)

```
algorithm nested_loop_join is
  for each tuple  $r$  in OuterInput do
    for each tuple  $s$  in InnerInput do
      if  $r$  and  $s$  satisfy the
        join condition then
        yield tuple  $\langle r, s \rangle$ 
```

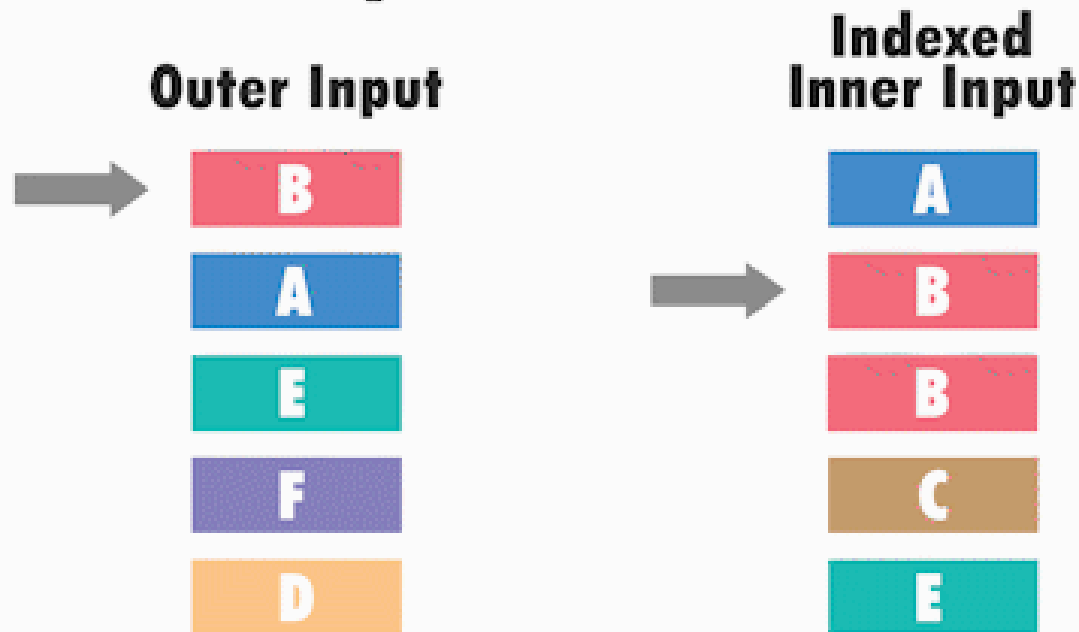




# NESTED LOOP JOIN (3/3)

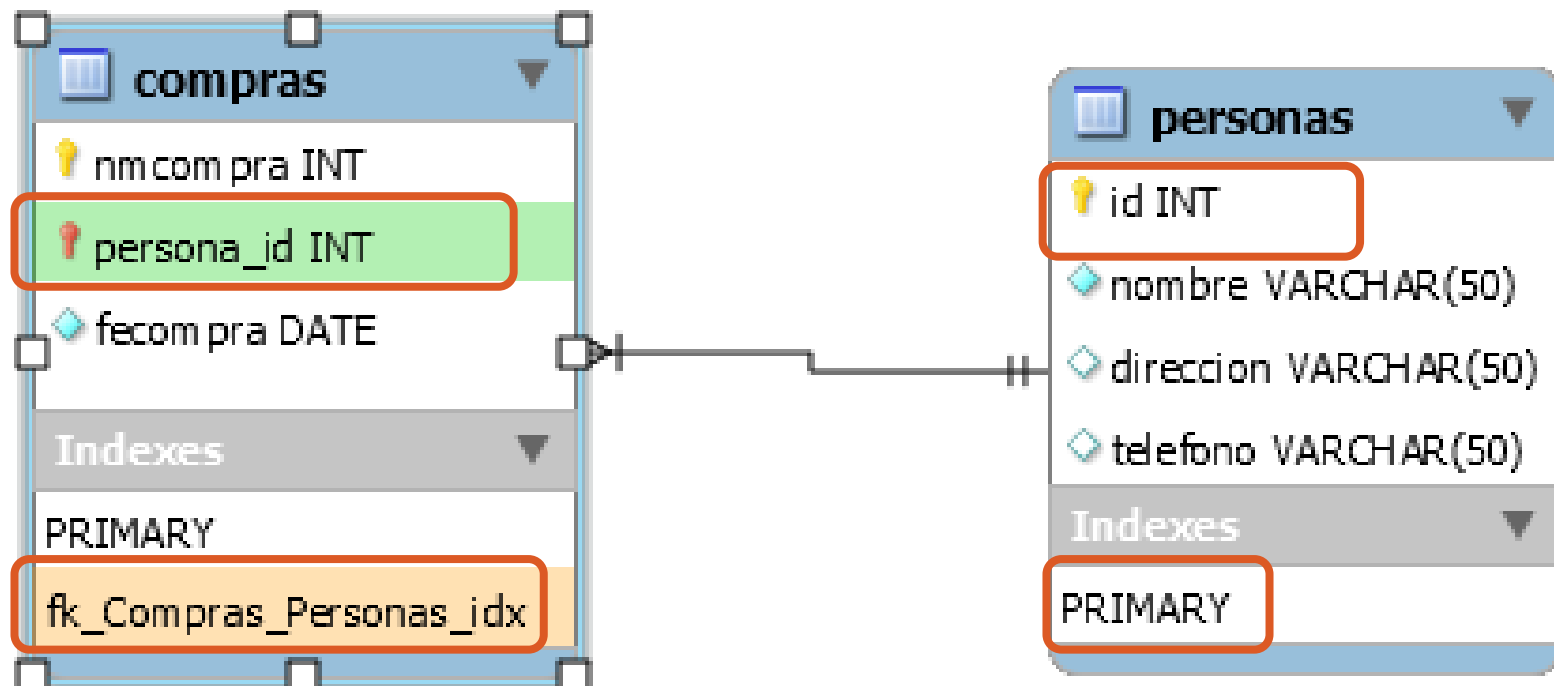
Que pasa si están indexadas las claves foráneas?

## Nested Loops Join

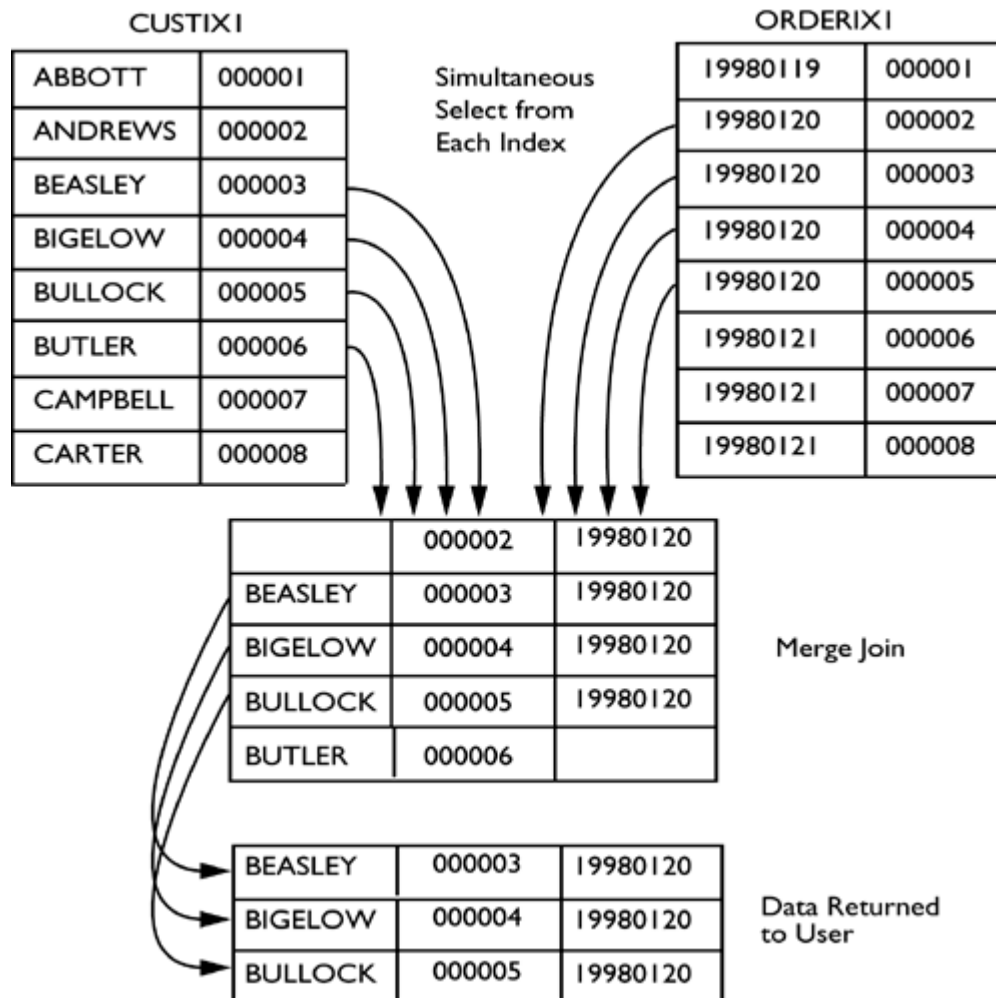


SQL  
BERT  
bertwagner.com

# CLAVES FORANEAS

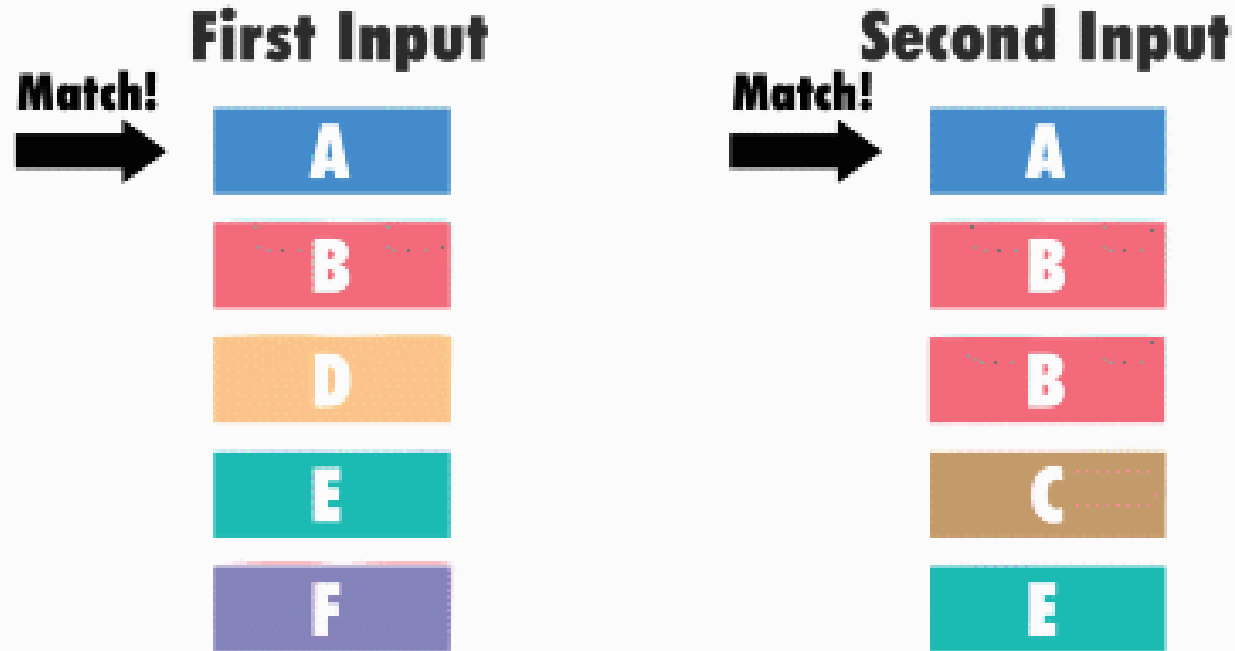


# MERGE JOIN (1/2)



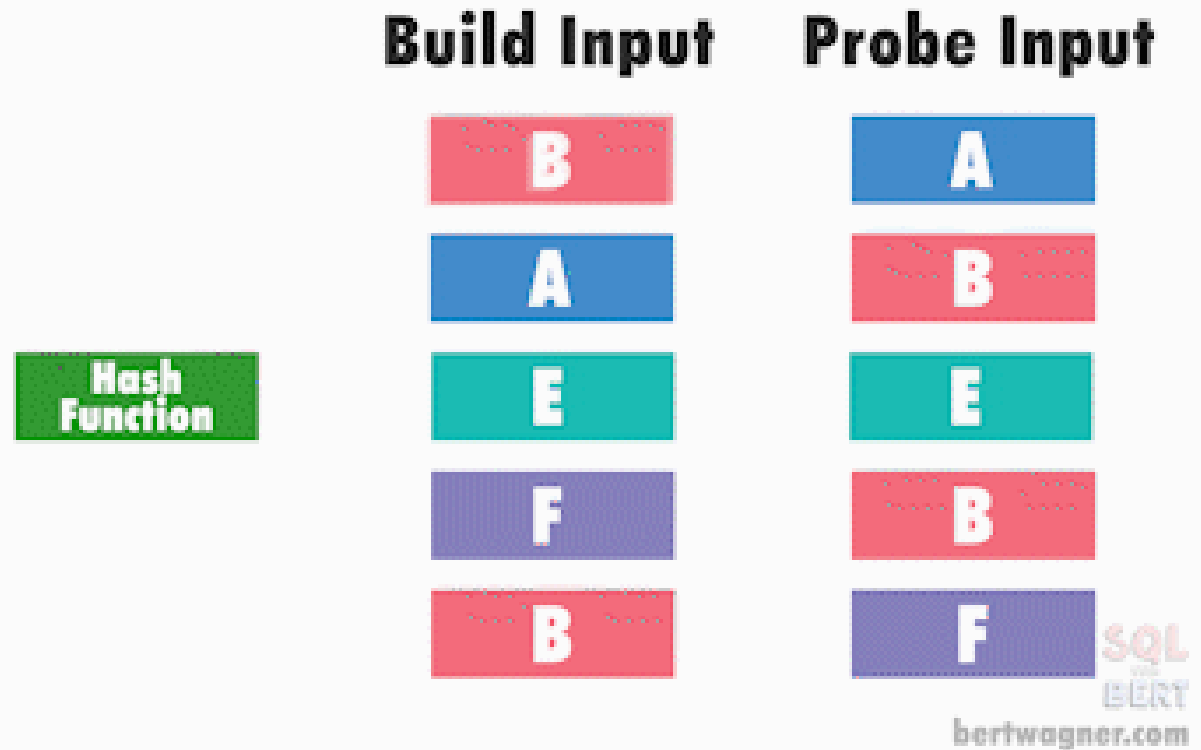
# MERGE JOIN (2/2)

## Merge Join



# HASH MATCH JOIN

## Hash Match Join



# INDICES

# **MÉTODO: “ESCOPETA DE PERDIGONES”**

**“Los índices mejoran el desempeño, entonces Indexo cada columna”**

# **MÉTODO: “ESCOPETA DE PERDIGONES”**

- **Muchos índices nunca son usados.**
- **Los índices ocupan espacio en disco y los buffers de memoria.**
- **Cada índice necesita ser modificado durante un Insert, Update y Delete**
- **El optimizador de consulta analiza que índices usar por cada consulta, mas índices hace que se necesite mas trabajo.**



# **MÉTODO: “ODIO A LOS ÍNDICES”**

**“Los índices requieren extra esfuerzo, entonces nunca creo un índice”**

# **MÉTODO: “ODIO A LOS ÍNDICES”**

- **Los índices correctos son esenciales para mejorar el desempeño.**
- **La mayoría de la carga de trabajo son consultas (Select), por lo que el beneficio neto de los índices supera la sobrecarga de la actualización.**
- **Los índices son compactos, por lo que son un uso más eficiente de la memoria intermedia.**

## **MÉTODO: “ODIO A SQL”**

**“Solo uso bases de datos  
NoSQL porque estas  
funcionan bien”**

# MÉTODO: “ODIO A SQL”

- Muchas bases de datos no relacionales requieren que usted defina los índices de forma explícita.
- Otras requieren definir de forma especial las tablas para poder tener un desempeño ideal.

# PREGUNTA BÁSICA

**La elección de un índice depende de:**

- Que tablas y columnas se necesitan en los queries.
- Que “JOINS” va a usar.
- Que “GROUP BYs” y “ORDER BYs” va a usar.

**El diseño de los índices no es implícito en el diseño del modelo.**

**El diseño de los índices depende de los queries.**

**Es decir se especifican a medida que se desarrolla la aplicación.**

# EJEMPLO

**IR A PÁGINA PRINCIPAL**

**100- LABORATORIOS CLINICOS (cont.) ... LAVANDERIAS DE ALFOMBRAS, CORTINAS, TAPETES Y TAPICERIA AUTOMOTRIZ**

**LABORATORIO CLINICO MARTINEZ**

Patricia Martínez Ariza. U. Javeriana  
Daisy Martínez Ariza. U. Metropolitana

**Tels: 421 27 96  
431 36 22**

Cels: 300 8163390 - 300 8154701  
301 3895338

Cl. 22 No. 14-70 C. Méd. Perla del Caribe  
labpat50@hotmail.com  
Santa Marta - Colombia

**ARNALDO VIVES PALMEZANO**  
Laboratorio Dental

**Francisco Melo Redondo**  
Criminología Dental - Prótesis Removable - Prótesis Total  
Prótesis Fija - Ortopedia Maxilar

Cel. 310 3681352 - 311 6665687  
Cl. 15 No. 7-89 P-2  
E-mail: arvident.lab@hotmail.com

**LABORATORIO DENTAL DEL CARIBE**  
Servicio a Domicilio

PROCESOS Removibles, Fijas  
Carillas Estéticas, Aperturación Ortodóncica  
Acidulados en General

Tel. 431 33 01 - Cels: 311 6918977 - 300 7045255  
Cl. 21 No. 4-54

**LABORATORIO CLINICO MLC**

Dra. MARBEL LÓPEZ CORTINA  
Bacterióloga - U. Metropolitana

Tel. 433 18 86 - 434 09 15  
Cel. 316 8086552 - 312 6209428

Av. del Libertador No. 27-128  
E-mail: labclinicomlc@hotmail.com

**SARAVIA GÓMEZ PATRICIA**  
Bacterióloga - U. de Antioquia  
Servicio a Domicilio Gratuito  
Cr. 21 No. 15 - 24

Tel. 420 12 06 - Cel. 315 6931028

Servicios Médicos Olimpus (ps) Ltda. Cr. 5 26B-15 ..... 434 7957

**LABORATORIOS DE DIAGNOSTICO**  
Ver: Laboratorios Clínicos

**LABORATORIOS DE SUELOS**

Ingeestudios Av. del Río 5-62 ..... 423 4643

**INGEESTUDIOS**  
INGENIERIA - LABORATORIO DE SUELOS Y PAVIMENTOS  
ESTUDIOS GEOTECNICOS  
ENSAYOS A MATERIALES  
CONTROL DE CALIDAD  
CALCULOS ESTRUCTURALES  
DISEÑOS - INTERVENTORIAS  
CONSULTORIAS

Av. del Río No. 5 - 62  
Tel. 423 46 43 - Fax 4319260  
Cels. 300 8143622 - 316 4527338  
E-mail: ingeestudios@hotmail.com

**LABORATORIOS DENTALES**

AB LABORATORIOS ODONTOLÓGICOS MARDENT

TELEFAX ..... Cr. 20 601 L31 431 3508  
B3 431 3182

directoriotelefonico.com.co

**LABORATORIOS DIESEL**

**Laboratorio DIESEL SANTANDER**  
Reparación y Calibración de Bombas de Inyección e Inyectores / Repuestos Diesel  
Troncal del Caribe Km. 2 Vía Gaira  
Telefax 430 81 73 - Cel. 314 551 7470 - 300 388 1890

**LABORATORIOS FARMACEUTICOS**  
Ver: Productos Químicos

**LABORATORIOS FOTOGRAFICOS**  
Ver: Fotografías

**LABORATORIOS MEDICOS**  
Ver: Laboratorios Clínicos

**LABORATORIOS VETERINARIOS**  
Ver: Diagnósticos Veterinarios  
Veterinarios

**LABORATORISTAS**  
Ver: Laboratorios Clínicos

**LADRILLOS**

Verificados & Gres e Hips Ltda. Cr. 22 13A-79 ..... 421 5815

**LAMINAS ACRILICAS**  
Ver: Acrílicos

**LAMINAS DE ACERO**

**SURTI FERRETERIAS S.A.**  
C/ 11 No. 11 A-50 Esq. PUNTO DE VENTA  
P.V.C. COLOMBIA - CEMENTO - MANIQUERIAS  
ZINC - CARBONO - PINTURAS - ALAMBRES  
Cableados Cableados ANILLOS - TUBERIAS - PLUMAS - VALLAS  
Cableados Cableados LAMINAS - CABLES - SOLDADURA - PINTURAS  
Tel. 430 48 78 - 421 67 13 - 421 68 47 - 421 68 83 - 421 67 28

Depósito Remotal Cr. 10 C19 Esq. .... 421 6031

**LAMINAS DE ALUMINIO**  
Ver: Ferreterías

**LAMINAS DECORATIVAS**  
Ver: Decoraciones

**LAMINAS PLASTICAS**  
Ver: Plásticos

**LANAS**

**LABORES Y CONFECCIONES**

Av. Ferroc 1500 421 7787 Oneyman Refrigeración Cr. 12 42-42 ..... 423 3524

**LANCHAS**

**Plast Fibra Ltda.**  
Fabricación de Embarcaciones para Pesca Artesanal - Carga y Turismo - Revestimientos

TODO EN FIBRA DE VIDRIO Y POLIURETANO

Venta de Materiales

Via a Gaira No. 46A - 30  
Frente a Zona Franca Industrial  
**430 17 86 - 430 53 16**  
Cel. 315 7253349  
E-mail: plastfibra1786@yahoo.com

**LANZADERAS**  
Ver: Máquinas de Coser

**LAPICEROS**

**LAPICEROS**  
MARCADOS TAMPOGRAFICAMENTE

Cr. 15 No. 5-77 - Tel. 420 0864 - Cel. 315 638028 - 316 2721541  
publicavichu@hotmail.com - Santa Marta

**LAPIDAS**

Artes Santa Marta Fundición y Relieves  
Av. del Ferroc 5-48 421 6943

**LATONERIAS**  
Ver: Plomerías

**LAVADO DE ALBERCAS**

Centro de Servicio Técnico Especializado Maratías ..... 432 4750

**LAVADO DE MUEBLES**

**Labores & Decoraciones**  
Servicio de Lavado

Av. Santa Rita Cr. 22 No. 17A-49 Tel. 420 87 13 Santa Marta

**LAVA SERVICIOS** Marz N Cr. 21 Nuevo Milenio 434 6543

**LAVADO DE TANQUES**

Centro de Servicio Técnico Especializado Maratías ..... 432 4750  
Tecnogas J.S. Ltda. Cr. 23 3-39 Rodadero Sur ..... 422 8032

**LAVADORAS**

**LAVANDERIAS DE ALFOMBRAS, CORTINAS, TAPETES Y TAPICERIA AUTOMOTRIZ**

**Labores & Decoraciones**  
Servicio de Lavado

Av. Santa Rita Cr. 22 No. 17A-49 Tel. 420 87 13 Santa Marta  
(CONTINUA EN LA PAGINA SIGUIENTE)

**DePrimeraclase**  
Guía de Viajes y Turismo

Una novedosa alternativa para atraer más clientes.

Pauta en De Primera Clase  
Línea Nacional:  
01 8000 930202

**LAS AMARILLAS DE DANARANJO**  
PARA VENDER MÁS

**Las Amarillas de Danaranjo**  
PARA VENDER MÁS

Encuentra aquí la información turística de la zona.

**DePrimeraclase**  
Guía de Viajes y Turismo

create index **paginas\_idx**  
on **paginas\_amarillas** (  
**tipo\_negocio**,  
**nombre\_negocio**);

select \*  
from **paginas\_amarillas**  
where **tipo\_negocio** =  
'Laboratorio clinicos';

select \*  
from **paginas\_amarillas**  
where **tipo\_negocio** =  
'Laboratorios clinicos'  
and **nombre\_negocio** =  
'Laboratorio clinico Martinez';

select \*  
from **paginas\_amarillas**  
where **nombre\_negocio** =  
'Laboratorio clinico Martinez';

\* El ultimo query NO USA EL INDICE

# MYSQL

**TIPOS DE MOTORES**  
**TIPOS DE ÍNDICES**

# TIPOS DE ÍNDICES MYSQL

Index Name	Columns		Index Type	
PRIMARY	id	...	PRIMARY	▼
personas1_idx	dni	...	UNIQUE	▼
personas2_idx	telefono	...	KEY	▼
personas3_idx	hoja_vida	...	FULLTEXT	▼

- **unique index:** can be no duplication of values
- **primary key:** a unique index in which no value can be NULL
- **simple, regular, or normal index**
- **fulltext index:** Sometimes you want to find the blob of text that contains a certain word or group of words or maybe you want to find a certain substring within the larger block of text.



# MOTORES DE MYSQL (1/4)

Table Name  Engine 

[default]

[default]

ARCHIVE

BLACKHOLE

CSV

InnoDB

MEMORY

MyISAM

## Archive

El mecanismo Archive está concebido para almacenar gran volumen de datos. Soporta INSERT y SELECT, pero no DELETE, REPLACE o UPDATE. Soporta el operador ORDER BY, atributos tipo BLOB y casi todos los demás, excepto los espaciales. Implementa bloqueo a nivel de fila (tupla).

## Blackhole

El mecanismo de almacenamiento Blackhole actúa como un agujero negro que acepta todo pero no lo almacena en ningún sitio. Todo intento de recuperación de información resultará infructuoso.

Al crear una tabla con el mecanismo Blackhole el servidor sí crea la definición del formato en el directorio de datos -un fichero con extensión.frm- pero no aparece ningún fichero más. Los datos de la tabla los envía todos a /dev/null. Soporta la definición de todo tipo de índices.

# MOTORES DE MYSQL (2/4)

Table Name  Engine 

[default]

[default]

ARCHIVE

BLACKHOLE

CSV

InnoDB

MEMORY

MyISAM

## CSV

Este mecanismo de almacenamiento almacena los datos en ficheros de texto separados por comas. Está disponible en todas las versiones del servidor MySQL.

Cuando se crea una tabla con el mecanismo CSV el servidor crea un fichero con el formato de la tabla en el directorio de datos con extensión frm, y crea un fichero para contener los datos con extensión CSV en el que se guardan los datos de cada tupla en una línea ASCII y separados por comas.

## Memory

Como indica la palabra, el mecanismo de almacenamiento Memory archiva las tablas en memoria volátil. Usa índices hash por defecto, lo que las hace muy rápidas de recorrer y muy útiles para crear tablas temporales. Sin embargo, cuando el servidor se apaga estas se borran. La definición de las tablas se guarda en ficheros con extensión.frm, de modo que al rearrancar el servidor las tablas estarán definidas pero vacías de datos.

# MOTORES DE MYSQL (3/4)

Table Name  Engine 

[default]

[default]

ARCHIVE

BLACKHOLE

CSV

InnoDB

MEMORY

MyISAM

## InnoDB

InnoDB es un mecanismo de almacenamiento transaccional -con características ACID- para MySQL. Realiza bloqueo a nivel de filas y lecturas no bloqueantes MVCC tipo Oracle, lo que aumentan la concurrencia y las prestaciones.

InnoDB pertenece a Oracle Corporation tras la adquisición de Innobase Oy en Octubre 2005. Tiene licenciamiento dual: se distribuye bajo licencia GPL, pero también se licencia a empresas que la quieran incluir en su software.

Sigue el modelo ACID con COMMIT, ROLLBACK y recuperación de caídas. Bloquea a nivel de fila (tupla), y proporciona concurrencia multiusuario para lectura.

Organiza los datos en disco para optimizar el uso de claves primarias.

Mantiene la integridad de datos mediante restricciones FOREIGN KEY, que se aplican en INSERT, UPDATE y DELETE.

# MOTORES DE MYSQL (4/4)

Table Name prueba

Engine

[default]

[default]

ARCHIVE

BLACKHOLE

CSV

InnoDB

MEMORY

MyISAM

## MyISAM (Indexed Sequential Access Method (ISAM))

Implementación de MySQL del original ISAM que efectúa el almacenamiento de datos de manera sencilla y robusta, y proporciona una alta velocidad de acceso y un tamaño pequeño de ficheros. El mecanismo MyISAM almacena cada tabla en tres ficheros:

.frm con el formato de la tabla

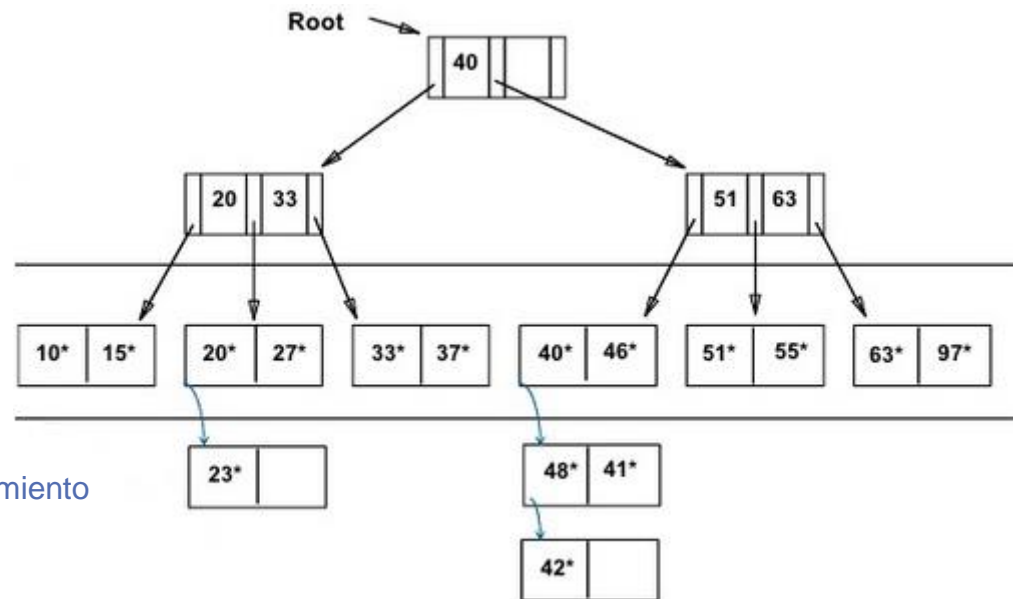
.MYD con los datos

.MYI con los índices

Paginas  
De Índices

Paginas  
De Hojas  
Primarias

Paginas  
De Desbordamiento



# AL FINAL TODO ESTA EN ARCHIVOS

Nombre	Fecha de modifica...	Tipo	Tamaño
abonos_cxc.frm	2020/04/16 6:42 a. ...	Archivo FRM	9 KB
abonos_cxc.ibd	2020/04/16 6:56 a. ...	Archivo IBD	112 KB
compras.frm	2020/04/16 6:42 a. ...	Archivo FRM	9 KB
compras.ibd	2020/04/16 6:42 a. ...	Archivo IBD	112 KB
db.opt	2020/04/16 6:42 a. ...	Archivo OPT	1 KB
detalle_abonos_cxc.frm	2020/04/16 6:42 a. ...	Archivo FRM	9 KB
detalle_abonos_cxc.ibd	2020/04/16 6:56 a. ...	Archivo IBD	128 KB
detallecompras.frm	2020/04/16 6:42 a. ...	Archivo FRM	9 KB
detallecompras.ibd	2020/04/16 6:42 a. ...	Archivo IBD	128 KB
detalleventa.frm	2020/04/16 6:42 a. ...	Archivo FRM	9 KB
detalleventa.ibd	2020/04/16 6:49 a. ...	Archivo IBD	128 KB
FTS_0000000000000127_0000000000000226_INDEX_1.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_0000000000000226_INDEX_2.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_0000000000000226_INDEX_3.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_0000000000000226_INDEX_4.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_0000000000000226_INDEX_5.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_0000000000000226_INDEX_6.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_BEING_DELETED.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_BEING_DELETED_CACHE.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_CONFIG.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_DELETED.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
FTS_0000000000000127_DELETED_CACHE.ibd	2020/04/27 2:21 p....	Archivo IBD	96 KB
personas.frm	2020/04/27 2:21 p....	Archivo FRM	9 KB
personas.ibd	2020/04/27 2:21 p....	Archivo IBD	144 KB
productos.frm	2020/04/16 6:42 a. ...	Archivo FRM	9 KB
productos.ibd	2020/04/16 6:42 a. ...	Archivo IBD	96 KB
ventas.frm	2020/04/16 6:42 a. ...	Archivo FRM	9 KB
ventas.ibd	2020/04/16 6:49 a. ...	Archivo IBD	112 KB

**Yes fragmented!**

# ESTADÍSTICAS (1/2)

Las estadísticas son uno de los elementos más importantes que ayudan al desempeño de los servidores y en general suelen pasar desapercibidos a las personas que se inician en Bases de Datos. Pero, ¿Qué son puntualmente estas estadísticas?. Son objetos que mantienen información sobre el contenido de las columnas de las tablas y cuya responsabilidad consiste en llevar control sobre: la selectividad, la cardinalidad y la densidad de los datos.

# ESTADÍSTICAS (2/2)

•**Cardinalidad:** Es una medida que describe la cantidad de valores únicos que tiene una tabla en una columna. Esta definición no debe confundirse con la definición de cardinalidad en modelado de datos, que se refiere al tipo de relación que tienen dos entidades/tablas y que puede ser: uno-a-uno, uno-a-muchos y muchos-a-muchos. Si queremos entonces calcular la cardinalidad de una columna en SQL básicamente tenemos que hacer:

```
SELECT COUNT(DISTINCT (MiColumna)) FROM MiTabla;
```

•**Densidad:** Es una medida que se utiliza para medir la frecuencia con que ocurren valores duplicados en una columna y es un valor “opuesto” a la cardinalidad. Opuesto en el sentido que dada una cantidad de filas en una tabla una mayor cardinalidad necesariamente implica una menor densidad. Si queremos calcular la densidad en SQL entonces usaríamos:

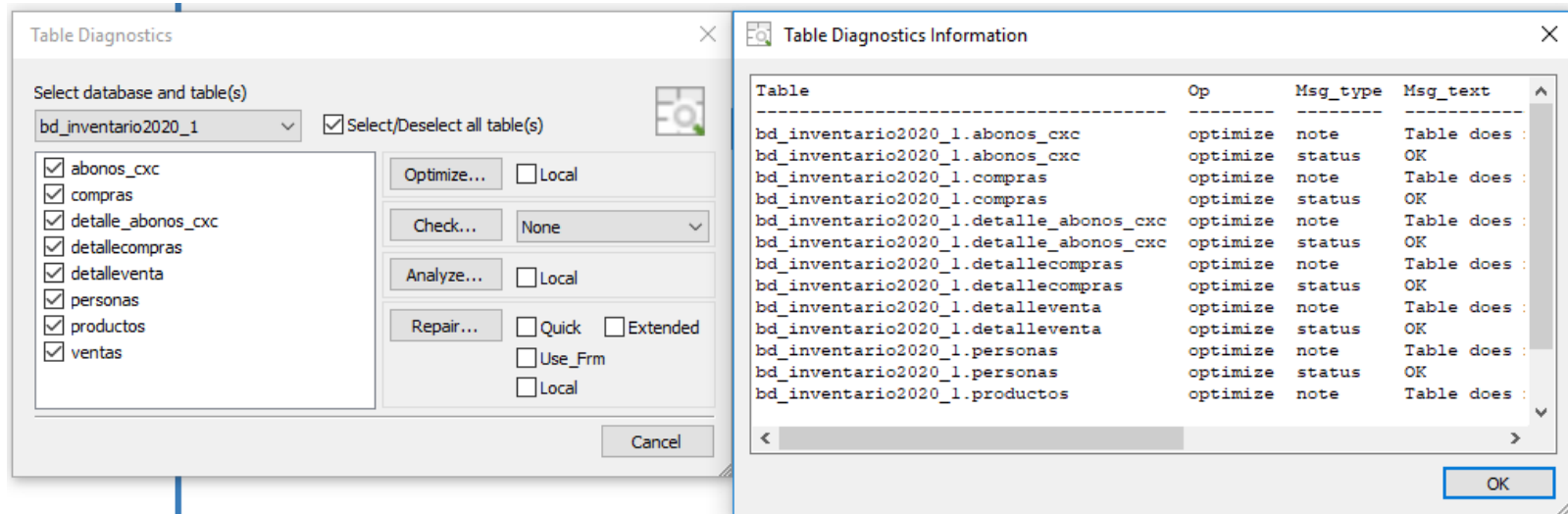
```
SELECT COUNT(*)/COUNT(DISTINCT (MiColumna)) FROM MiTabla;
```

La interpretación de la densidad es el promedio de filas duplicadas que para cada valor tiene una columna en una tabla.

•**Selectividad:** Es el número de filas que satisface un predicado. Para entender esta definición deber entonces definir predicado, como una expresión que se evalúa como verdadera o falsa. En SQL calcularíamos la selectividad como:

```
SELECT COUNT(*) FROM MiTabla WHERE MiColumna=Valor; -- Predicado
```

# CORRER RUTINAS DE OPTIMIZACIÓN Y DESFRAGMENTACIÓN



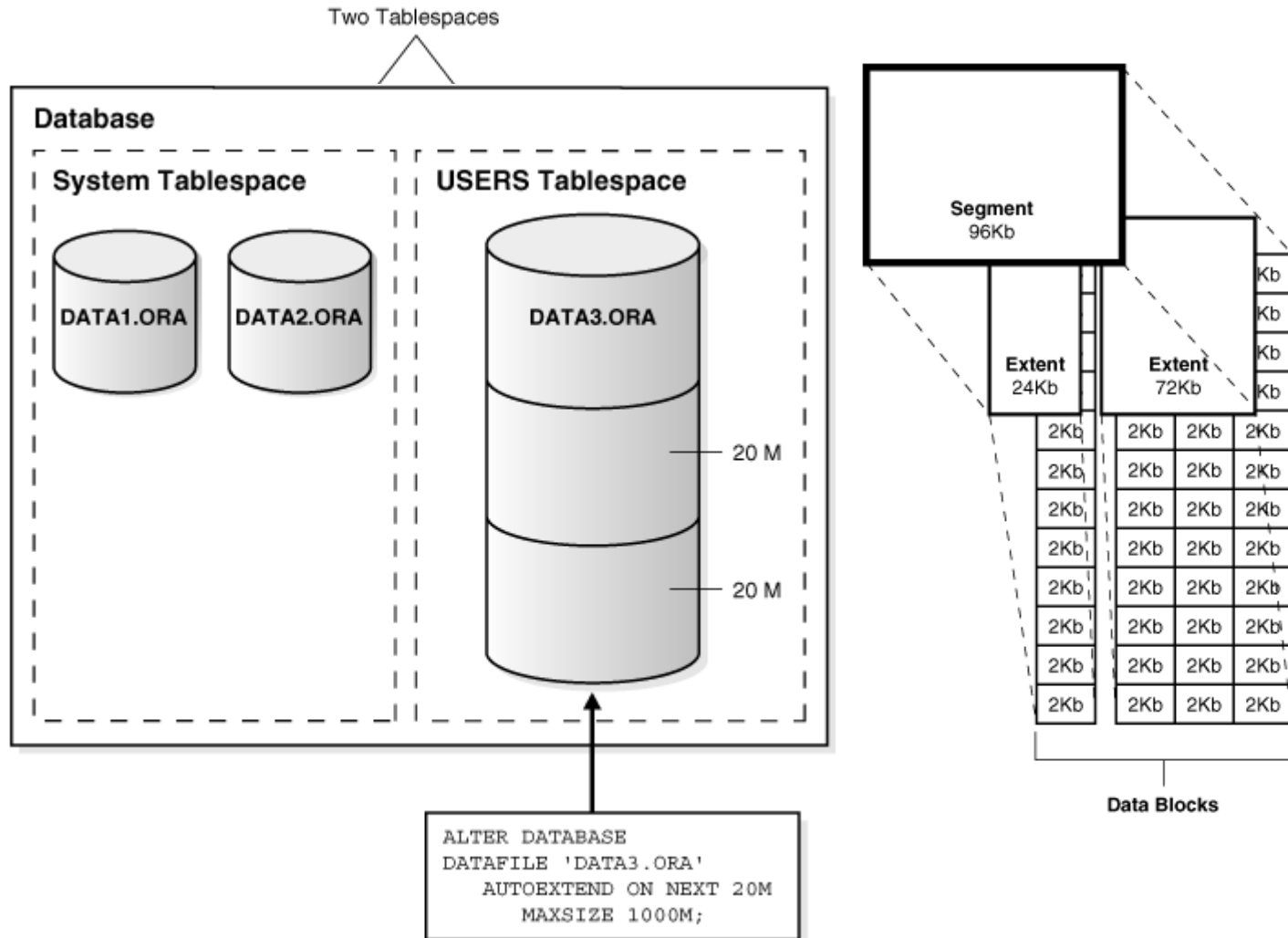
OPTIMIZE [NO\_WRITE\_TO\_BINLOG | LOCAL]  
TABLE tbl\_name [, tbl\_name] ...



# ORACLE

**DIMENSIONAMIENTO DE TABLAS**  
**TIPOS DE ÍNDICES**

# ALMACENAMIENTO EN ORACLE



# EXTENSIÓN

**Una extensión es una unidad lógica de almacenamiento que está formada por un número determinado de bloques de datos contiguos.**

**Al crear o, mejor dicho, asignar una nueva extensión al segmento, se está reservando espacio en el disco para almacenar los nuevos datos de dicho segmento. Por lo tanto, al crear la nueva extensión está totalmente vacía y todo su espacio está disponible para almacenar los datos del segmento y, además, en el disco debe haber espacio libre para que Oracle reserve todo el tamaño que necesita la extensión, y lo formatea de forma especial para poder utilizarlo. A partir de ese momento, en esa extensión solamente se podrán almacenar datos del segmento al que pertenece.**

**Cuando se llenan todos los bloques de datos de una extensión, el segmento solicita una nueva extensión al sistema para poder seguir almacenando información.**

# STORAGE CLAUSE (1/2)

Configuración del almacenamiento de tablas (CREATE TABLE), índices (CREATE INDEX), etc... en oracle.

## STORAGE opciones

**INITIAL** int K | M

**NEXT** int K | M

**PCTINCREASE** int

**MINEXTENTS** int

**MAXEXTENTS** int

**initial:** Especifica el tamaño (en bytes) de la primera extensión.

**next:** Especifica el tamaño (en bytes) de la segunda extensión.

**pctincrease:** Especifica el % de incremento en el tamaño de las siguientes extensiones.

El tamaño de una nueva extensión es el tamaño de la anterior multiplicado por **pctincrease**.

**minextents:** Especifica el número inicial de extensiones cuando se crea el objeto.

**maxextents:** Especifica el número máximo de extensiones que el objeto puede tener.

# STORAGE CLAUSE (2/2)

```
create table empleado (nombre varchar2(50))  
storage (initial 256K next 512K pctincrease 50 minextents 3 maxextents 10);  
.
```

Results

Explain

Describe

Saved SQL

History

Table created.

0.02 seconds

Select extent\_id, bytes, blocks from USER\_EXTENTS  
where segment\_name = 'EMPLEADO' order by extent\_id;

0	286720	35
1	532480	65
2	819200	100

# DIMENSIONAMIENTO PESIMISTA

Index Name	Column	Position	Length	Order
PK_E1_EMP	EMPNO	1	22	ASC

	Column Name	Type	Length
1	EMPNO	NUMBER	22
2	ENAME	VARCHAR2	10
3	JOB	VARCHAR2	9
4	MGR	NUMBER	22
5	HIREDATE	DATE	7
6	SAL	NUMBER	22
7	COMM	NUMBER	22
8	DEPTNO	NUMBER	22

Tamaño por registro:

$22 + 10 + 9 + 22 + 7 + 22 + 22 + 22 =$

**136**

Preguntas:

- Cuanto **años** se quiere guardar la información?
- Promedio de registros en un **años**?

Entonces si por ejemplo:

- 5 años
- 5000 registros por año

$$5 * 5000 * 136 = 3320K$$

# DIMENSIONAMIENTO ANÁLISIS

E1\_EMP

Table	Data	Indexes	Model	Constraints	Grants	Statistics
Analyze						
Rows	Blocks	Average Row Length	Sample Size	Last Analyzed		
14	2	41	14	7 seconds ago		

Mode: ☐ Compute Statistics  
☒ **Estimate Statistics**  
☐ Validate Structure

Type: ☒ **Percent** (?)  
☐ Row

Amount:  (?)

Preguntas:

- Cuanto **años** se quiere guardar la información?
- Promedio de registros en un **años**?

Entonces si por ejemplo:

- 5 años
- 5000 registros por año

$$5 * 5000 * 41 = 1001K$$

# INDICES (1/2)

## B-Tree Index

Este es el tipo de índices normal de Oracle, estos índices son los que más se utilizan, están organizados en una estructura de árbol B y por lo general se utilizan para las llaves primarias de manera que se pueda hacer una búsqueda por llave más rápido.

## Bitmap Index

Los índices de bitmap son muy recomendables en columnas en las cuales los valores se repiten y representan una división en categorías, por ejemplo columnas como género, estado civil, etc. También son muy recomendables cuando no cambian mucho, aunque sean muy variantes.

## Cluster Index

Este índice consiste en unas tablas que comparten los mismos bloques de datos, utilizan un valor llamado clúster key value que es el valor de las columnas involucradas para una fila en específico.

El clúster key sirve para guardar juntas las filas que tengan el mismo valor y solo guarda una vez cada valor distinto.



# INDICES (2/2)

## Hash Cluster Index

Este índice es muy parecido al anterior, pero se aplica una función de hash al cluster key para poder encontrar las demás filas con el mismo cluster key.

## Partitioned tables

Cuando las tablas crecen mucho, acceder a ellas puede significar un bajo rendimiento, por eso surge el concepto de tablas particionadas que es dividir las tablas en partes independientes, se le puede sacar backup independientemente, el rendimiento de las consultas puede aumentar significativamente, y se pueden hacer transacciones paralelas en particiones diferentes.

Es común que las diferentes particiones, se coloquen en un tablespace diferente para optimizar aun más el rendimiento.

## Function Based Index

Este tipo de índice es muy sencillo pero muy útil y puede llegar a mejorar los tiempos de respuesta increíblemente. Este índice no es más que un índice B-Tree pero sobre una función, por ejemplo para las consultas dadas para este proyecto se utiliza mucho la función upper entonces cree un índice B-Tree sobre la función upper del campo nombre, entonces en índice están los datos ya con la función upper aplicada.

# EJEMPLO ÍNDICE ORACLE

## Create Index

Schema: **CO\_A751\_PLSQL\_S29** ⓘ

Table **E1\_EMP** ⓘ

Type of Index ☒ **Normal** ⓘ  
☐ Text

- Los índices normales son para indexar números y la fecha VARCHAR
- Los índices de texto son para indexar columnas CLOB (Objetos binarios largos)

\* Index Name  ⓘ

☐ Preserve Case

Uniqueness  ⓘ

\* Index Column 1  ⓘ

# HERRAMIENTAS

# EXPLAIN PLAN

```
1  EXPLAIN
2  SELECT id, nombre, f_calcula_deuda_cxc( id )
3  FROM personas
4  ORDER BY nombre ASC ;
```

1 Result 2 Profiler 3 Messages 4 Table Data 5 I

(Read Only)

	id	select_type	table	partitions	type
	1	SIMPLE	personas	(NULL)	OK ALL

```
1  EXPLAIN
2  SELECT id, nombre, f_calcula_deuda_cxc( id )
3  FROM personas
4  WHERE id = 1036
5  ORDER BY nombre ASC ;
6
```

1 Result 2 Profiler 3 Messages 4 Table Data 5 I

(Read Only)

	id	select_type	table	partitions	type
	1	SIMPLE	personas	(NULL)	OK const

SQL Commands Object Browser

← → ↻ iacademy3.oracle.com/ords3/f?p=4500:1003:2798485718678:....

ORACLE® Application Express Application Builder

SQL Commands

Rows 10 ? Clear Command

```
select *
from e1_emp
where empno = 7369;
```

Results Explain Describe Saved SQL History

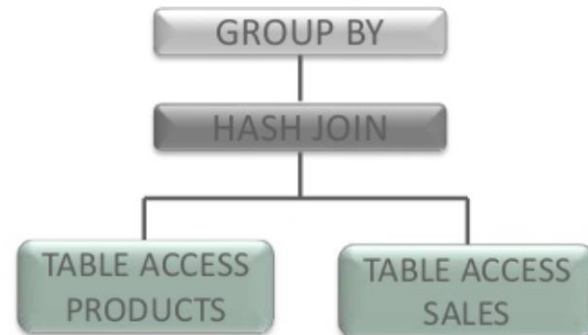
# EXPLAIN PLAN

```
SELECT prod_category, avg(amount_sold)
  FROM sales s, products p
 WHERE p.prod_id = s.prod_id
GROUP BY prod_category;
```

Tabular representation of plan

Id	Operation	Name
0	SELECT STATEMENT	
1	HASH GROUP BY	
2	HASH JOIN	
3	TABLE ACCESS FULL	PRODUCTS
4	PARTITION RANGE ALL	
5	TABLE ACCESS FULL	SALES

Tree-shaped representation of plan



# NOSQL - MONGO

Países.ciudadanos

DOCUMENTS 2 TOTAL SIZE 310B AVG. SIZE 155B INDEXES 1 TOT 3

Documents

Aggregations

Explain Plan

Indexes

**FILTER** {"pasaportes.idpais":1}

► OPTIONS

**EXPLAIN**

VIEW DETAILS AS VISUAL TREE RAW JSON

## Query Performance Summary

Documents Returned: 1

Index Keys Examined: 0

Documents Examined: 2

Actual Query Execution Time (ms): 0

Sorted in Memory: no

⚠ No index available for this query.