

# 10

## Fragmentación

**Bases de datos II**  
**Universidad del Cauca**  
**Ing. Wilson Ortega**

# Bases de datos distribuidas

- Una BDD (Base de Datos Distribuida) es un conjunto de Bases de Datos relacionadas lógicamente, pero que se encuentran físicamente localizadas en varios “sitios” de la red.
- Un SGBDD permite el manejo de la BDD y hace esta distribución transparente a todos los usuarios.

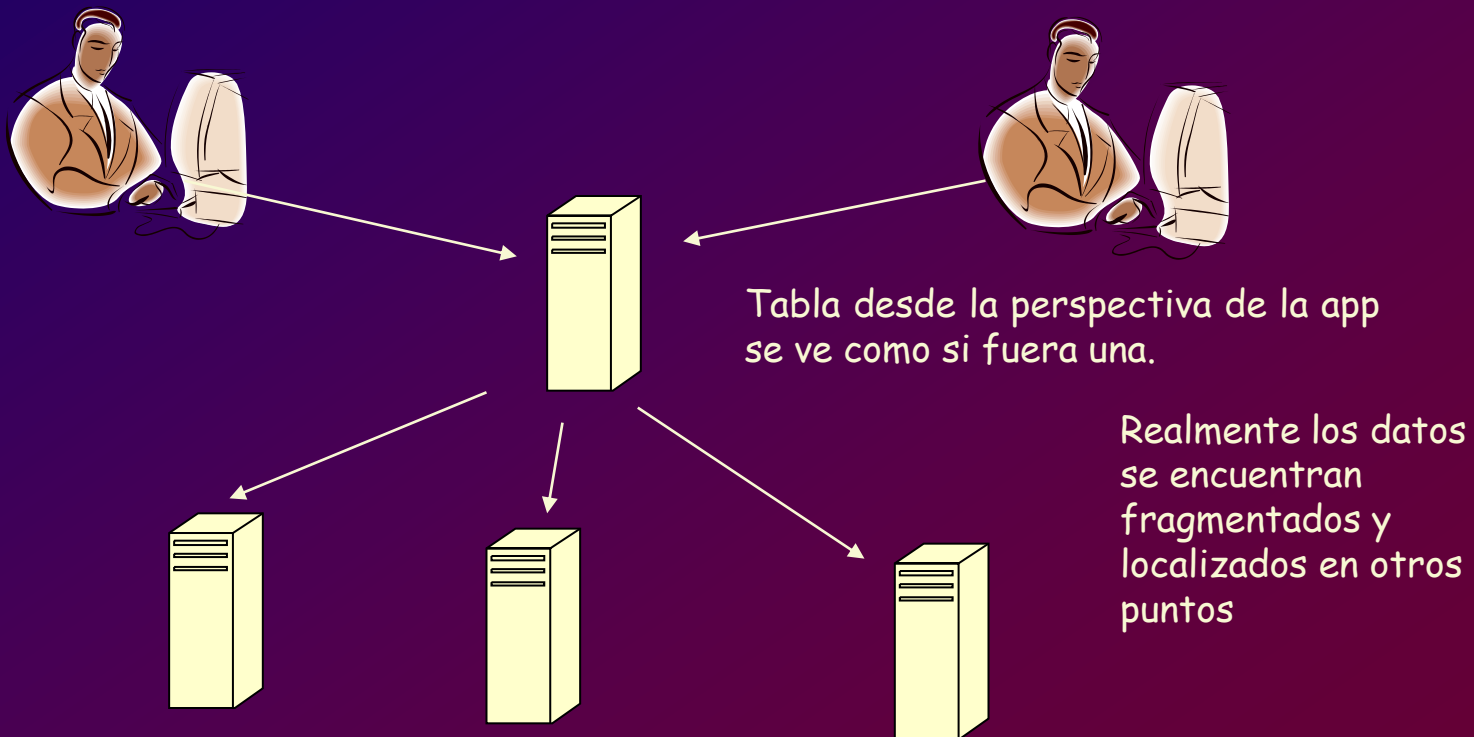
# Bases de datos distribuidas

- Cada punto de la red es una base de datos.
- Cada computador que maneja una BD de una BDD se le denomina NODO.
- BD Local: BD a la cual el usuario está directamente conectado.
- BD Remota: cualquier BD adicional.
- Arquitectura Cliente-Servidor.

# Fragmentación

- Cada tabla se puede separar en varios fragmentos. Cada fragmento se almacena en un solo nodo
- Es una técnica de optimización para el mejoramiento de los tiempos de respuesta de las consultas, donde las tablas son muy grandes.
- Son transparentes a las apps. Desde la perspectiva del usuario, una tabla fragmentada es igual que una no fragmentada.
- Con las mismas sentencias DML podemos acceder a las tablas seleccionadas.

# Fragmentación



Las tablas se descomponen en tablas mas pequeñas

# Tipos de Fragmentación

- Fragmentación Horizontal: subconjuntos de filas.
- Fragmentación Vertical: subconjuntos de columnas.
- Fragmentación Mixta: aplica las fragmentaciones anteriores a la vez.

# Tipos de Fragmentación

TABLA ESTUDIANTE

CODEST	NOMBRE	EDAD	CARRERA	PROMEDIO	BECA
10	Gustavo Correa	21	Ing. Sistemas	4.6	Si
20	Laura Muñoz	23	Ing. Electrónica	3.3	No
30	Oscar Rendón	19	Ing. Sistemas	3.5	No
40	Rubén Ordoñez	22	Ing. Electrónica	4.6	Si

Fragmentación Horizontal

CODEST	NOMBRE	EDAD	CARRERA	PROMEDIO	BECA
10	Gustavo Correa	21	Ing. Sistemas	4.6	Si
30	Oscar Rendón	19	Ing. Sistemas	3.5	No

CODEST	NOMBRE	EDAD	CARRERA	PROMEDIO	BECA
20	Laura Muñoz	23	Ing. Electrónica	3.3	No
40	Rubén Ordoñez	22	Ing. Electrónica	4.6	Si



# Tipos de Fragmentación

TABLA ESTUDIANTE

CODEST	NOMBRE	EDAD	CARRERA	PROMEDIO	BECA
10	Gustavo Correa	21	Ing. Sistemas	4.6	Si
20	Laura Muñoz	23	Ing. Electrónica	3.3	No
30	Oscar Rendón	19	Ing. Sistemas	3.5	No
40	Rubén Ordoñez	22	Ing. Electrónica	4.6	Si

Fragmentación Vertical

CODEST	NOMBRE	EDAD	CARRERA	PROMEDIO
10	Gustavo Correa	21	Ing. Sistemas	4.6
20	Laura Muñoz	23	Ing. Electrónica	3.3
30	Oscar Rendón	19	Ing. Sistemas	3.5
40	Rubén Ordoñez	22	Ing. Electrónica	4.6

CODEST	NOMBRE	EDAD	CARRERA	BECA
10	Gustavo Correa	21	Ing. Sistemas	Si
20	Laura Muñoz	23	Ing. Electrónica	No
30	Oscar Rendón	19	Ing. Sistemas	No
40	Rubén Ordoñez	22	Ing. Electrónica	Si



# Tipos de Fragmentación

TABLA ESTUDIANTE

CODEST	NOMBRE	EDAD	CARRERA	PROMEDIO	BECA
10	Gustavo Correa	21	Ing. Sistemas	4.6	Si
20	Laura Muñoz	23	Ing. Electrónica	3.3	No
30	Oscar Rendón	19	Ing. Sistemas	3.5	No
40	Rubén Ordoñez	22	Ing. Electrónica	4.6	Si

Fragmentación Mixta

CODEST	NOMBRE	EDAD	CARRERA	PROMEDIO
10	Gustavo Correa	21	Ing. Sistemas	4.6
30	Oscar Rendón	19	Ing. Sistemas	3.5

CODEST	NOMBRE	EDAD	CARRERA	BECA
10	Gustavo Correa	21	Ing. Sistemas	Si
30	Oscar Rendón	19	Ing. Sistemas	No

# Estructura física y lógica de la BD Oracle

- Estructura lógica:
  - Indica la composición y distribución teórica de la base de datos.
  - Sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando.
  - Se divide en unidades de almacenamiento lógicas llamadas Tablespaces.

# Estructura física y lógica de la BD Oracle

- Estructura lógica:
  - Cada Base de Datos estará formada por uno o más tablespaces. Cada tablespace se corresponde con uno o más archivos de datos
  - El Tablespace SYSTEM se crea automáticamente al hacer la instalación de Oracle
  - Al crear una Base de Datos, este Tablespace contiene el diccionario de datos.

# Estructura física y lógica de la BD Oracle

- Estructura física:
  - Es la estructura de los datos tan cual se almacenan en las unidades de disco. La correspondencia entre la estructura lógica y la física se almacena en la base de datos es decir en los metadatos.
  - Una B.D. tiene uno o más ficheros de datos. Estos ficheros son de tamaño fijo y se establecen en el momento en que se crea la base de datos o en el momento en el que se crean tablespaces.
  - Los datos del fichero de datos son leídos cuando se necesitan y situados en una caché de memoria compartida para que el próximo acceso a los mismos sea más rápido.

# Creación de Tablespaces

- Una base de datos está formada por una o varias unidades lógicas llamadas tablespaces
- Cada tablespaces está formado por uno o varios ficheros físicos que son los datafiles.
- Un datafile solamente puede pertenecer a un tablespace.

# Creación de Tablespaces

- Cuando se crea una base de datos, hay que crear al menos un tablespace (SYSTEM).
- Es recomendable crear otro tablespace distinto al SYSTEM de modo que todos los nuevos usuarios, tablas e índices se almacenarán en un tablespace diferente, esto evita que se bloquee toda la base de datos si ocurre algo en el tablespace SYSTEM.



# Creación de Tablespaces

- Debemos iniciar la sesión en la base de datos con un usuario con permisos de administración, el usuario SYSTEM  
Ej.

```
CREATE TABLESPACE TESTUDIANTES DATAFILE 'C:\ORACLE\ESTUDIANTES\TESTUDIANTES.DBF'  
SIZE 300M;
```

- Crear un usuario que trabaja sobre ese tablespace:

```
CREATE USER user_estudiantes IDENTIFIED BY oracle DEFAULT TABLESPACE  
TESTUDIANTES;
```

Si no se especifica un tablespace, la Base de Datos le asignará el tablespace USERS, que es el tablespace que se utiliza por defecto para los nuevos usuarios



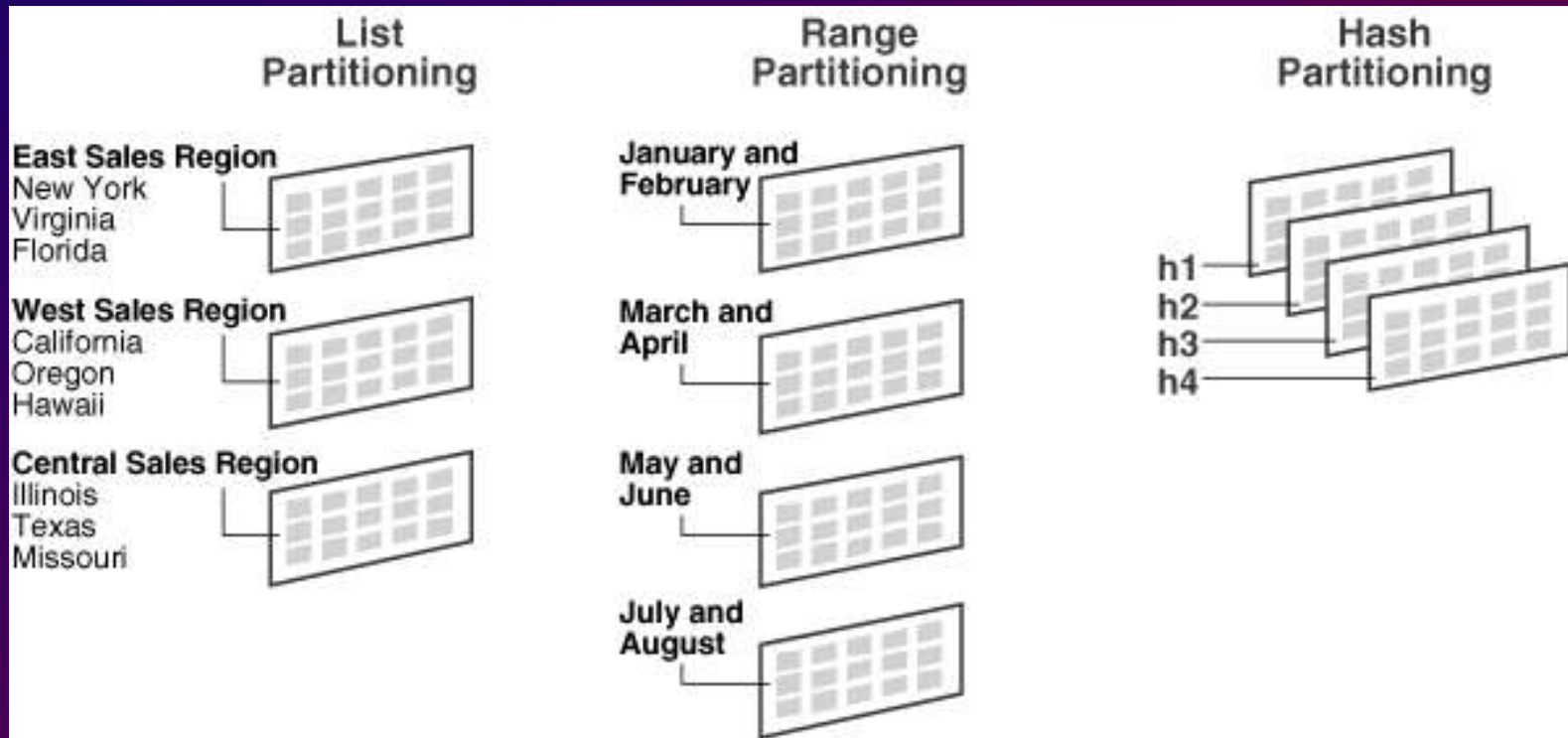
# Estrategias de fragmentación en Oracle

- Oracle 8.0: Particionamiento por Rango.
- Oracle 8i: Particionamiento por Rango, Hash y Composite.
- Oracle 9iR2/10g: Particionamiento por Rango, Hash, Composite y el tipo List
- Oracle 11g: Columnas virtuales para particionar (que no existen físicamente en la tabla), particionado de Sistema (donde podemos gestionar directamente en que partición de la tabla se insertan los registros) y el particionado por Intervalos.

# Cuándo particionar?

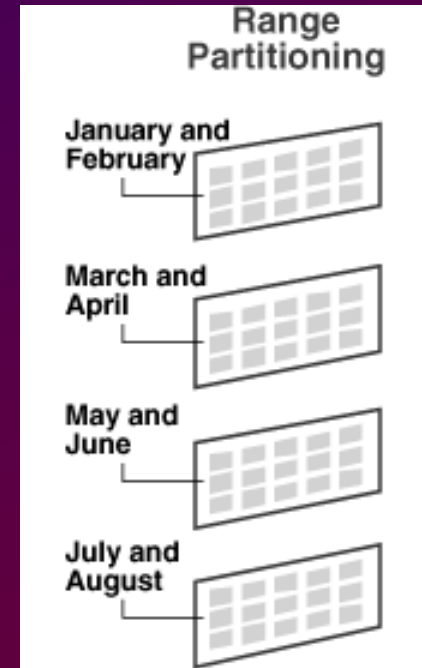
- Tablas con tamaño superior a 2 GB deberían ser consideradas candidatas.
- Tablas que contienen datos históricos, en las cuales los nuevos datos son añadidos en una nueva partición. Un ejemplo típico en una tabla histórica donde solo se actualizan datos del mes actual y los demás meses son solamente de lectura.
- Cuando el contenido de una tabla necesita ser distribuido a través de diferentes tipos de dispositivos de almacenamiento

# Estrategias básicas



# Particionado por rango

- El valor de la columna “partition key” determina la partición a la cual pertenecerá el registro
- Este tipo de particionamiento esta mejor situado cuando se tiene datos que tienen rango lógicos y que pueden ser distribuidos por este. Ej. Mes del Año o un valor numérico.



# Particionado por rango

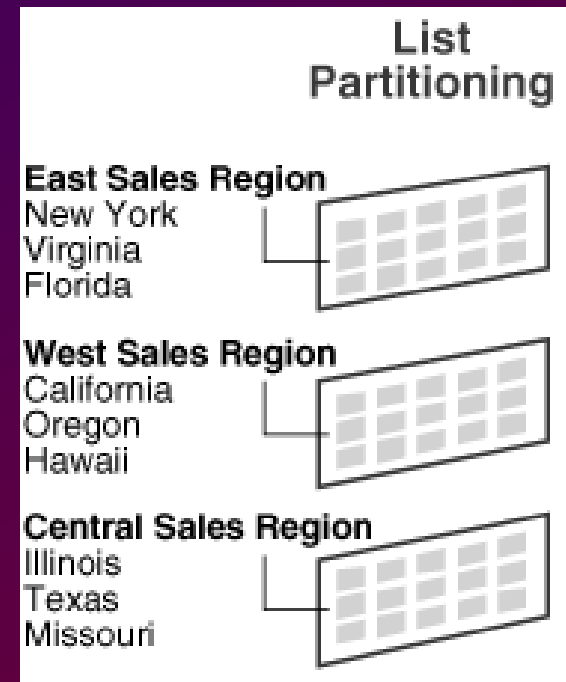
Ej.

```
CREATE TABLE ventas
( prod_id      NUMBER(6)
, cli_id       NUMBER
, fecha_id     DATE
, valor        NUMBER(10,2)
)
PARTITION BY RANGE (fecha_id)
( PARTITION ventas_s1_2014 VALUES LESS THAN (TO_DATE('01-04-2014','dd-mm-yyyy'))
TABLESPACE tss1
, PARTITION ventas_s2_2014 VALUES LESS THAN (TO_DATE('01-06-2014','dd-mm-yyyy'))
TABLESPACE tss2
, PARTITION ventas_s3_2014 VALUES LESS THAN (TO_DATE('01-09-2014','dd-mm-yyyy'))
TABLESPACE tss3
, PARTITION ventas_s4_2014 VALUES LESS THAN (TO_DATE('01-01-2015','dd-mm-yyyy'))
TABLESPACE tss4
, PARTITION ventas_otras VALUES LESS THAN MAXVALUE
);
```

Es opcional asignar un tablespace a cada partición.

# Particionado por listas

- Permite determinar el particionado según una lista de valores definidos sobre el valor de una columna específica.
- Los valores literales deben ser únicos en la lista, permitiendo el uso del valor NULL
- El valor DEFAULT sirve para definir la partición donde irán registros que no cumplen ninguna condición de las diferentes particiones.





# Particionado por listas

Ej.

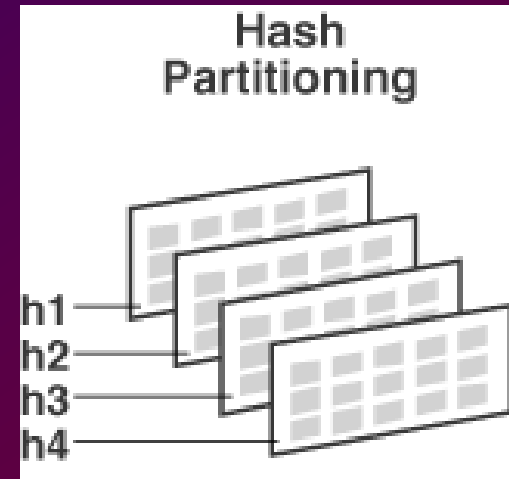
```
CREATE TABLE ventas
( prod_id      NUMBER(6)
, cli_id       NUMBER
, fecha_id     DATE
, valor        NUMBER(10,2)
, pais_nombre  VARCHAR2(20)
)
PARTITION BY LIST(pais_nombre)
(
PARTITION ventas_suramerica VALUES('Colombia', 'Ecuador'),
PARTITION ventas_norteamerica VALUES ('Estados Unidos', 'Mexico'),
PARTITION ventas_otras VALUES(DEFAULT)
);
```

Es posible asignar un tablespace a cada partición.



# Particionado por Hash

- Los registros de la tabla tienen su localización física determinada aplicando un valor hash a la columna del partition key.
- La función hash devuelve un valor automático que determina a qué partición irá el registro.
- Es una forma automática de balancear el particionado.



# Particionado por Hash

Ej1.

```
CREATE TABLE departamento (deptno NUMBER, deptnombre VARCHAR(32))  
    PARTITION BY HASH(deptno) PARTITIONS 16;
```

Ej2.

```
CREATE TABLE departamento (deptno NUMBER, deptnombre VARCHAR(32))  
    PARTITION BY HASH(deptno)  
    (PARTITION p1 TABLESPACE ts1, PARTITION p2 TABLESPACE ts2,  
     PARTITION p3 TABLESPACE ts1, PARTITION p4 TABLESPACE ts3);
```

Es opcional asignar un tablespace a cada partición.

# Otras estrategias de particionado

- <http://www.dataprix.com/blogs/respinosamilla/particionado-tablas-oracle>
- [http://docs.oracle.com/cd/B28359\\_01/server.111/b32024/partition.htm](http://docs.oracle.com/cd/B28359_01/server.111/b32024/partition.htm)

# Ejercicio

PROFESOR		
<u>Cedula</u>	NUMBER	<pk>
Nombre	VARCHAR2(100)	
Salario	NUMBER(10,2)	
Facultad_Id	NUMBER	<fk>

Crear la tabla Profesor usando:

- Particionado por rango
- Particionado por listas
- Particionado por Hash