



# INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

6to Semestre



## Ingeniería en Sistemas Computacionales

### ADMINISTRACIÓN DE BASES DE DATOS

Actividad: Cuestionario y Mapa conceptual  
“Configuración y administración del espacio en disco”

Docente: M.T.I., I.S.C. Salvador Acevedo Sandoval

Alumna: Pritschella Berenice Flores Estrada

Correo Electrónico: [prits99@hotmail.com](mailto:prits99@hotmail.com)

No. Control: S17070169

Jerez De García Salinas, Zac.

20/03/ 2020

Investigar los puntos que se indican a continuación y posteriormente crear un mapa conceptual en formato electrónico.

## **MySQL**

### **1. Asignación de espacio en disco para base de datos.**

Se hace a través de estructuras lógicas de almacenamiento que permiten que el espacio en disco se asigne de forma dinámica cuando la base de datos crece de tamaño.

### **2. Asignación de espacio en disco para tablas.**

Formado por uno o más datafiles, cada datafile sólo puede pertenecer a un determinado tablespace. Es decir, consiste en páginas de base de datos con un tamaño por defecto de 16KB.

### **3. Asignación de espacio en disco para usarlos.**

De manera similar a la de las tablas, es decir, formado por uno o más datafiles, cada datafile sólo puede pertenecer a un determinado tablespace.

### **4. Particionamiento de tablas.**

#### **a. ¿Qué es y para qué se utiliza?**

Es una división de una base de datos lógica o sus elementos constituyentes en partes independientes.

Se utiliza por razones de mantenimiento, rendimiento o manejo. También se utiliza para separar una tabla grande en tablas individuales más pequeñas.

#### **b. Tipos.**

\*Manual: El particionamiento lo podríamos realizar nosotros en nuestra lógica de procesos de carga ETL.

\*Automático: La gestión la realiza de forma automática el motor de base de datos tanto a la hora de insertar registros como a la hora de leerlos.

\*Range: La asignación de los registros de la tabla a las diferentes particiones se realiza según un rango de valores definido sobre una determinada columna de la tabla o expresión.

\*List: La asignación de los registros de la tabla a las diferentes particiones se realiza según una lista de valores definida sobre una determinada columna de la tabla o expresión.

\*Hash: Está pensando para repartir de forma equitativa los registros de la tabla entre las diferentes particiones.

**Key:** Similar al HASH, pero la función para el particionado la proporciona MySQL automáticamente.

**c. Limitaciones/restricciones**

- \*Construcciones prohibidas.
- \*Operadores aritméticos y lógicos.
- \*Número máximo de particiones.
- \*Claves foráneas no admitidas.

**d. Instrucciones de ejemplo para cada tipo.**

-Range:

```
CREATE TABLE `salaries_partitions_salary` (
    `emp_no` int NOT NULL,
    `salary` int NOT NULL,
    `from_date` date NOT NULL,
    `to_date` date NOT NULL
)
PARTITION BY RANGE(salary)
(
    PARTITION part1 VALUES LESS THAN(30000),
    PARTITION part2 VALUES LESS THAN(60000),
    PARTITION part3 VALUES LESS THAN(90000),
    PARTITION part4 VALUES LESS THAN(120000),
    PARTITION part5 VALUES LESS THAN(150000),
    PARTITION part_default VALUES LESS THAN MAXVALUE
);
```

-List:

```
CREATE TABLE `category_partitions_category_id` (
    `category_id` tinyint unsigned NOT NULL AUTO_INCREMENT,
    `name` varchar(25) NOT NULL,
    `last_update` timestamp NOT NULL DEFAULT
```

```
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
PRIMARY KEY (`category_id`))  
PARTITION BY LIST(category_id)(  
PARTITION accion VALUES IN(1, 2, 3, 4),  
PARTITION terror VALUES IN(5, 6, 7, 8),  
PARTITION ciencia_ficcion VALUES IN(9, 10, 11, 12),  
PARTITION romance VALUES IN(13, 14, 15, 16)  
);
```

-Hash:

```
CREATE TABLE `employees_partition_emp_no` (  
`emp_no` int NOT NULL,  
`birth_date` date NOT NULL,  
`first_name` varchar(14) NOT NULL,  
`last_name` varchar(16) NOT NULL,  
`gender` enum('M','F') NOT NULL,  
`hire_date` date NOT NULL,  
PRIMARY KEY (`emp_no`))  
PARTITION BY HASH(emp_no) PARTITIONS 5;
```

-Key:

```
CREATE TABLE `departments_partition_by_dept_no` (  
`dept_no` char(4) NOT NULL,  
`dept_name` varchar(40) NOT NULL,  
PRIMARY KEY (`dept_no`))  
PARTITION BY KEY() PARTITIONS 3;
```

## *PostgreSQL*

### **1. Asignación de espacio en disco para base de datos.**

Se realiza dinámicamente.

### **2. Asignación de espacio en disco para tablas.**

Puede configurarse manualmente o de forma dinámica sin exceder el límite.

### **3. Asignación de espacio en disco para usarlos.**

Similar a las tablas; puede configurarse manualmente o de forma dinámica sin exceder el límite.

### **4. Particionamiento de tablas.**

#### **a. ¿Qué es y para qué se utiliza?**

Se denomina particionado mediante herencia de tablas. Cada partición puede ser creada como una tabla hija de una única tabla padre.

Se utiliza para reducir la cantidad de datos a recorrer en cada consulta SQL y aumentar el rendimiento.

#### **b. Tipos.**

\*Por rangos: La tabla es particionada mediante rangos definidos en base a la columna de llave primaria o cualquier columna que no se solape entre los rangos de valores asignados a diferentes tablas hijas.

\*Por lista: La tabla es particionada listando los valores de cada una de las llaves en cada partición.

#### **c. Limitaciones/restricciones**

El tamaño de una tabla puede ser de hasta 32 TB, sin embargo, sin el mantenimiento adecuado y sin una estrategia, este tipo de tablas serán casi inutilizables.

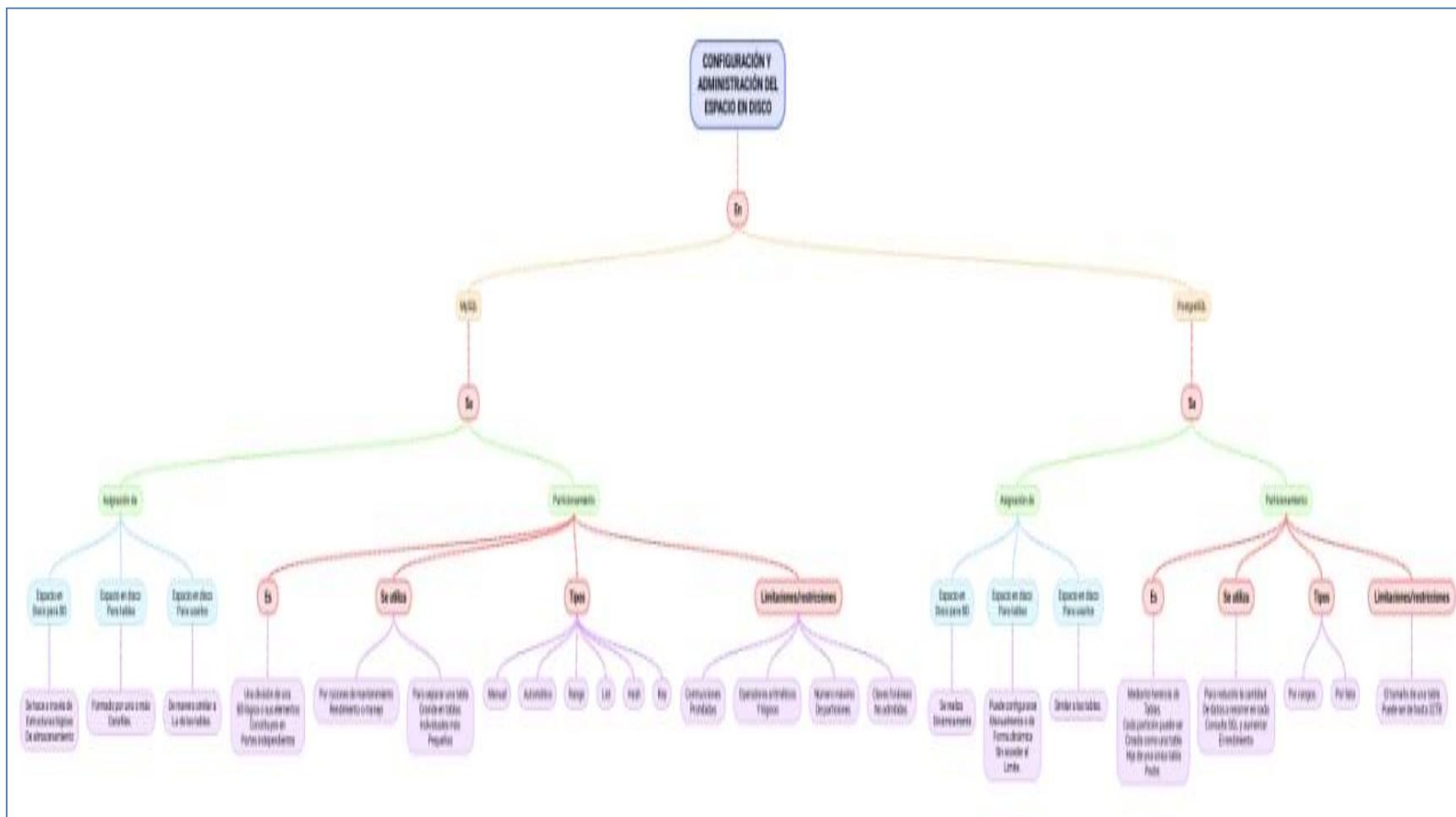
#### **d. Instrucciones de ejemplo para cada tipo.**

\*Tablas hijas

```
test=# CREATE TABLE prueba_1 (CHECK (col >0 AND col <1000001))
INHERITS (prueba);
```

\*Reglas adicionales

```
test=# CREATE RULE prueba_3_rule AS ON INSERT TO prueba WHERE
(col >2000000 AND col <3000001) DO INSTEAD INSERT INTO prueba_3
VALUES (NEW.*);
```



## BIBLIOGRAFÍA

- Desconocido. (s.f.). *Unidad III. Configuración y administración del espacio en disco.* Recuperado el 18 de Marzo de 2020, de Unidad III. Configuración y administración del espacio en disco: <http://itpn.mx/recursosisc/6semestre/administraciondebasesdedatos/Unidad%20III.pdf>
- PostgreSQL. (27 de Noviembre de 2009). *PostgreSQL. Particionamiento.* Recuperado el 20 de Marzo de 2020, de PostgreSQL. Particionamiento: <https://beastieux.com/2009/11/27/postgresql-particionamiento-de-tablas/>