



Ingeniería en Sistemas
Computacionales



INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ
JEREZ DE GARCÍA SALINAS A 20 DE MARZO DEL 2020

NOMBRE:
GUADALUPE VÁZQUEZ DE LA TORRE

NUMERO DE CONTROL:
S17070158

CORREOS:
guvadlt@gmail.com

CARRERA:
INGENIERÍA EN SISTEMAS COMPUTACIONALES

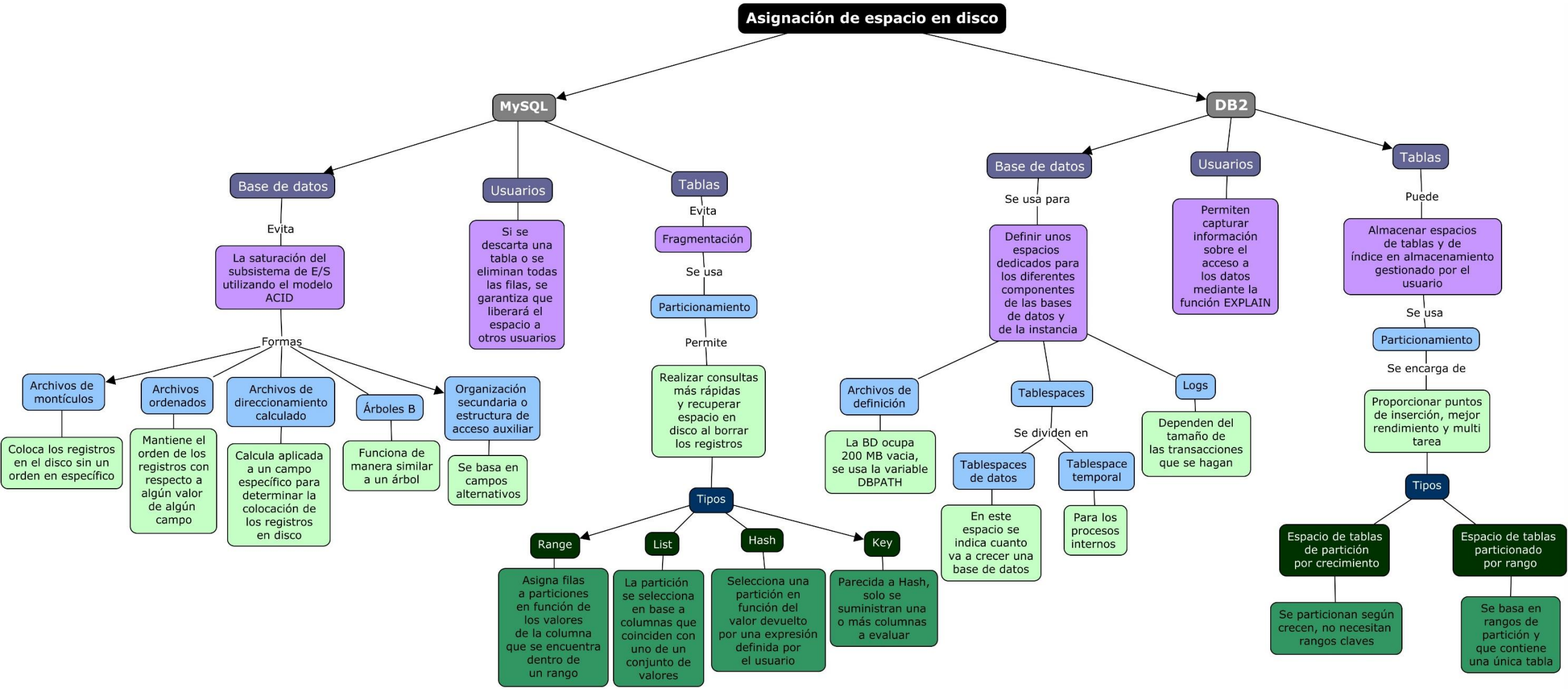
NOMBRE DE LA MATERIA:
ADMINISTRACIÓN DE BASES DE DATOS

SEXTO SEMESTRE

TEMA 3 - CONFIGURACIÓN Y ADMINISTRACIÓN DEL ESPACIO EN
DISCO

“ACTIVIDAD 2 - CUESTIONARIO Y MAPA CONCEPTUAL”

DOCENTE:
SALVADOR ACEVEDO SANDOVAL



MYSQL

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

Como DBA, debe administrar las E / S de disco para evitar que el subsistema de E / S se sature y administrar el espacio en disco para evitar llenar los dispositivos de almacenamiento. El modelo de diseño ACID requiere una cierta cantidad de E / S que puede parecer redundante, pero ayuda a garantizar la fiabilidad de los datos. Dentro de estas restricciones, InnoDB intenta optimizar el trabajo de la base de datos y la organización de los archivos de disco para minimizar la cantidad de E / S de disco.

Los archivos de datos que se definen en la configuración usando el `innodb_data_file_path`, los archivos son concatenados para formar el espacio de tabla del sistema.

Las bases de datos se almacenan en ficheros o archivos. Existen diferentes formas de organizaciones primarias de archivos que determinan la forma en que los registros de un archivo se colocan físicamente en el disco y, por lo tanto, cómo se accede a éstos.

Las distintas formas de organizaciones primarias de archivos son:

- Archivos de Montículos (o no Ordenados): esta técnica coloca los registros en el disco sin un orden específico, añadiendo nuevos registros al final del archivo.
- Archivos Ordenados (o Secuenciales): mantiene el orden de los registros con respecto a algún valor de algún campo (clave de ordenación).
- Archivos de Direccionamiento Calculado: utilizan una función de direccionamiento calculado aplicada a un campo específico para determinar la colocación de los registros en disco.
- Árboles B: se vale de la estructura de árbol para las colocaciones de registros.
- Organización Secundaria o Estructura de Acceso Auxiliar: Estas permiten que los accesos a los registros de un archivo basado en campos alternativos, sean más eficientes que los que han sido utilizados para la organización primaria de archivos.

2. Asignación de espacio en disco para tablas.

Para evitar los problemas que vienen con el almacenamiento de todas las tablas e índices dentro del espacio de tablas del sistema, puede habilitar la `innodb_file_per_table` opción de configuración (la predeterminada), que almacena cada tabla recién creada en un archivo de espacio de tabla separado (con extensión. `ibd`). Para las tablas almacenadas de esta manera, hay menos fragmentación dentro del archivo de disco, y cuando la tabla se trunca, el espacio se devuelve al sistema operativo en lugar de que InnoDB aún lo reserve dentro del espacio de tablas del sistema.

3. Asignación de espacio en disco para usuarios.

Cuando elimina datos de una tabla, InnoDB contrae los índices de árbol B correspondientes. El hecho de que el espacio liberado esté disponible para otros usuarios depende de si el patrón de eliminaciones libera páginas individuales o extensiones al espacio de tabla. Si se descarta una tabla o se eliminan todas las filas, se garantiza que liberará el espacio a otros usuarios, pero recuerde que las filas eliminadas se eliminan físicamente solo por la operación, que ocurre automáticamente un tiempo después de que ya no son necesarias para las reversiones de transacciones o lecturas consistentes.

4. Particionamiento de tablas

4a. ¿Qué es y para qué se utiliza?

Permite rotar la información de las tablas en diferentes particiones, consiguiendo así realizar consultas más rápidas y recuperar espacio en disco al borrar los registros.

4b. Tipos

- Particionamiento RANGE. Este tipo de particionamiento asigna filas a particiones en función de los valores de la columna que se encuentran dentro de un rango determinado.
- Particionamiento LIST. Similar a la partición RANGE, excepto que la partición se selecciona en base a columnas que coinciden con uno de un conjunto de valores discretos.
- Particionamiento HASH. Con este tipo de particionamiento, se selecciona una partición en función del valor devuelto por una expresión definida por el usuario que opera en valores de columna en filas para insertar en la tabla. La función puede consistir en cualquier expresión válida en MySQL que produzca un valor entero no negativo.
- Particionamiento KEY. Este tipo de particionamiento es similar al particionamiento por HASH, excepto que solo se suministran una o más columnas a evaluar, y el servidor MySQL proporciona su propia función de hashing. Estas columnas pueden contener valores distintos a los enteros, ya que la función de hash proporcionada por MySQL garantiza un resultado entero independientemente del tipo de datos de la columna.

4a. Limitaciones/restricciones

- Procedimientos almacenados, funciones almacenadas, UDF o complementos. Variables declaradas o variables de usuario.
- Operadores aritméticos y lógicos. El uso de los operadores aritméticos +, - y * está permitido en las expresiones de partición. Sin embargo, el resultado debe ser un valor entero o NULL (excepto en el caso de la partición [LINEAL] KEY).
- Los resultados de muchas funciones y operadores de MySQL pueden cambiar de acuerdo con el modo SQL del servidor y puede conducir a

cambios importantes en el comportamiento de dichas tablas, y podría conducir fácilmente a la corrupción o pérdida de datos.

- Índices, partition pruning. Al igual que con las tablas no particionadas, el uso adecuado de los índices puede acelerar significativamente las consultas en las tablas particionadas. Además, el diseño de tablas particionadas y consultas en estas tablas para aprovechar partition pruning puede mejorar drásticamente el rendimiento.
- Rendimiento con LOAD DATA. En MySQL 8.0, LOAD DATA utiliza el almacenamiento en búfer para mejorar el rendimiento. Hay que tener en cuenta que el búfer usa 130 KB de memoria por partición para lograr esto.
- Número máximo de particiones. El número máximo posible de particiones para una tabla dada que no usa el motor de almacenamiento NDB es 8192. Este número incluye subparticiones.

4b. Instrucciones de ejemplo para cada tipo

```
CREATE TABLE `salaries_partition_by_from_date` (  
  `emp_no` int(11) NOT NULL,  
  `salary` int(11) NOT NULL,  
  `from_date` date NOT NULL,  
  `to_date` date NOT NULL)  
PARTITION BY RANGE (YEAR(from_date))(  
  PARTITION part1 VALUES LESS THAN(1990),  
  PARTITION part2 VALUES LESS THAN(1995),  
  PARTITION part3 VALUES LESS THAN(2000),  
  PARTITION part6 VALUES LESS THAN maxvalue  
);  
  
CREATE TABLE `category_partition_category_id` (  
  `category_id` tinyint(3) 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)  
);
```

```
CREATE TABLE `employees_partitions_emp_no` (
  `emp_no` int(11) 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;

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

DB2

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

Cuando se monta una base de datos Db2, es importante definir unos espacios dedicados para los diferentes componentes de las bases de datos y de la instancia

Bases de datos: Una base de datos contiene 3 componentes cuando se habla de espacio usado: Logs, Tablespaces (Datos), archivos de definición.

- Archivos de definición: La base de datos recién creada ocupa 200 MB, donde no hay nada de datos, y solo existen datos en el tablespace del catálogo. Esto se define al crear la base de datos en la variable DBPATH
- Tablespaces: Los tablespaces se pueden dividir entre temporales y de datos.
- Tablespace de datos: Dependiendo del tamaño que vaya a tener la base de datos y la proyección de crecimiento, en este espacio se indica cuanto va a crecer.
- Tablespace temporal: para los procesos internos de Db2 como los reorgs o joins se usa un system temporary tablespace.
- Logs: Los logs dependen del tamaño de las transacciones que se hagan.

2. Asignación de espacio en disco para tablas.

Puede almacenar espacios de tablas y espacios de índice en almacenamiento gestionado por el usuario, almacenamiento gestionado por SMS o en grupos de almacenamiento gestionados por DB2. (Un grupo de almacenamiento es un conjunto de volúmenes de disco.)

3. Asignación de espacio en disco para usuarios.

Las tablas de usuario proporcionadas permiten capturar información sobre el acceso a los datos mediante la función EXPLAIN, con el objetivo de supervisar mediante perfiles de supervisión el rendimiento del sistema y las consultas, y de limitar, mediante el recurso de límite de recursos, el uso de los recursos que hacen algunos servidores de middleware o aplicaciones en concreto. Las instancias de estas tablas permiten también que las herramientas de optimización capturen, analicen y almacenen la información sobre el rendimiento de consultas.

4. Particionamiento de tablas

4a. ¿Qué es y para qué se utiliza?

- Proporciona múltiples puntos de inserción para una misma tabla.
- Potencialmente mejora el rendimiento haciendo posible que las operaciones de lectura/escritura sean distribuidas sobre múltiples dispositivos de datos.
- Mejora el rendimiento cuando el servidor está configurado para realizar procesamiento paralelo de consultas, ya que puede haber un worker processes por partición en una búsqueda basada en particiones.
- Ejecuta múltiples tareas bcp sobre una tabla particionada, permitiendo controlar a qué particiones se hace la copia de un bcp.

4b. Tipos

Espacio de tablas de partición por crecimiento: las tablas segmentadas se particionan según crecen, sin necesidad de rangos de claves, como resultado, las tablas segmentadas se benefician de límites aumentados de espacio de tablas y del paralelismo de programas de utilidad.

Espacio de tablas particionado por rango: es un tipo de espacio de tablas universal que se basa en rangos de partición y que contiene una única tabla

4a. Limitaciones/restricciones

- Cada partición de base de datos especificada por número debe estar definida en el archivo db2nodes.cfg (SQLSTATE 42729).
- Cada núm-partición-bd listado en la cláusula ON DBPARTITIONNUMS puede aparecer solo una vez (SQLSTATE 42728).
- Un número-partición-base-de-datos válido es un número comprendido entre 0 y 999, ambos inclusive (SQLSTATE 42729).
- La sentencia CREATE DATABASE PARTITION GROUP puede resultar anómala si una petición para añadir un servidor de particiones de base de datos está pendiente o en curso o si se añade en línea un servidor de particiones de base de datos nuevo a la instancia y no todas las aplicaciones saben de la existencia del servidor de particiones de base de datos nuevo.

4b. Instrucciones de ejemplo para cada tipo

```
-CREATE DATABASE PARTITION GROUP--nombre-grupo-particiones-bd-->

.-ON ALL DBPARTITIONNUMS-----
+-----
|                                     .,-----
|                                     V
'-ON--+-DBPARTITIONNUMS--(----partición-bd-número1--+-+-----+
      '-DBPARTITIONNUM--'                                '-TO--partición-bd-número2--'
```

- nombre-grupo-particiones-bd: Indica el nombre del grupo de particiones de base de datos.
- ON ALL DBPARTITIONNUMS: Especifica que el grupo de particiones de base de datos está definido en todas las particiones de base de datos definidas para la base de datos cuando se crea el grupo de particiones de base de datos.
- ON DBPARTITIONNUMS: Especifica las particiones de base de datos que se encuentran en el grupo de particiones de base de datos.
- partición-bd-núm-1: Especifique un número de partición de base de datos.
- TO partición-bd-número2: Especifique un rango de números de partición de base de datos.

Fuentes:

- Unidad 3: Configuración y Administración del Espacio en Disco (s.f), recuperado de: <https://sites.google.com/site/itjabd23/home/asignatura/plan-de-estudios/unidad-3-configuracion-y-administracion-del-espacio-en-disco>
- MySQL (s.f), 15.11.2 File Space Management, recuperado de: <https://dev.mysql.com/doc/refman/8.0/en/innodb-file-space.html>
- MySQL.com. (s.f). 23.1 Overview of Partitioning in MySQL., de MySQL.com Sitio web: <https://dev.mysql.com/doc/refman/8.0/en/partitioning-overview.html>
- MySQL.com. (s.f). 23.6 Restrictions and Limitations on Partitioning, de MySQL.com Sitio web: <https://dev.mysql.com/doc/refman/8.0/en/partitioning-limitations.html>
- MySQL.com. (s.f). 23.2 Partitioning Types, de MySQL.com Sitio web: <https://dev.mysql.com/doc/refman/8.0/en/partitioning-types.html>
- Santana Macías Jesús Daniel. (s.f). Administración de Bases de Datos, de BlogSpot.com Sitio web: <http://dan1456bd.blogspot.com/p/particiones-en-mysql-y-oracle.html>

- Space disks for DB2 (s.f), recuperado de:
<https://angocadb2.blogspot.com/2017/12/space-disks-for-db2.html>
- IBM (s.f), Asignación de espacios de tablas a almacenamiento físico, recuperado de:
https://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/intro/src/tpc/db2z_tablespacephysicalstorage.html
- IBM (s.f), Tablas de usuario proporcionadas por DB2, recuperado de:
https://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/usrtab/src/tpc/db2z_suptab.html
- Db Support (s.f), Base de Datos SQL Server, Oracle, Sybase, MySQL, sitio web:
<https://www.dbasupport.com.mx/index.php/bases-de-datos/sybase/sybase-admin/126-particionamiento-de-tablas>
- IBM, (s.f), Espacios de tablas universales, Sitio web:
https://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/wnw/src/tpc/db2z_09_universaltablespacewnw.html
- IBM (s.f), Sentencia CREATE DATABASE PARTITION GROUP, Sitio web:
https://www.ibm.com/support/knowledgecenter/es/SSEPGG_11.1.0/com.ibm.db2.luw.sql.ref.doc/doc/r0000921.html