INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

18 de marzo del 2020

Jerez, Zac

Ingeniería en sistemas computacionales

Semestre: 6

Alumna: Leticia carrera venegas

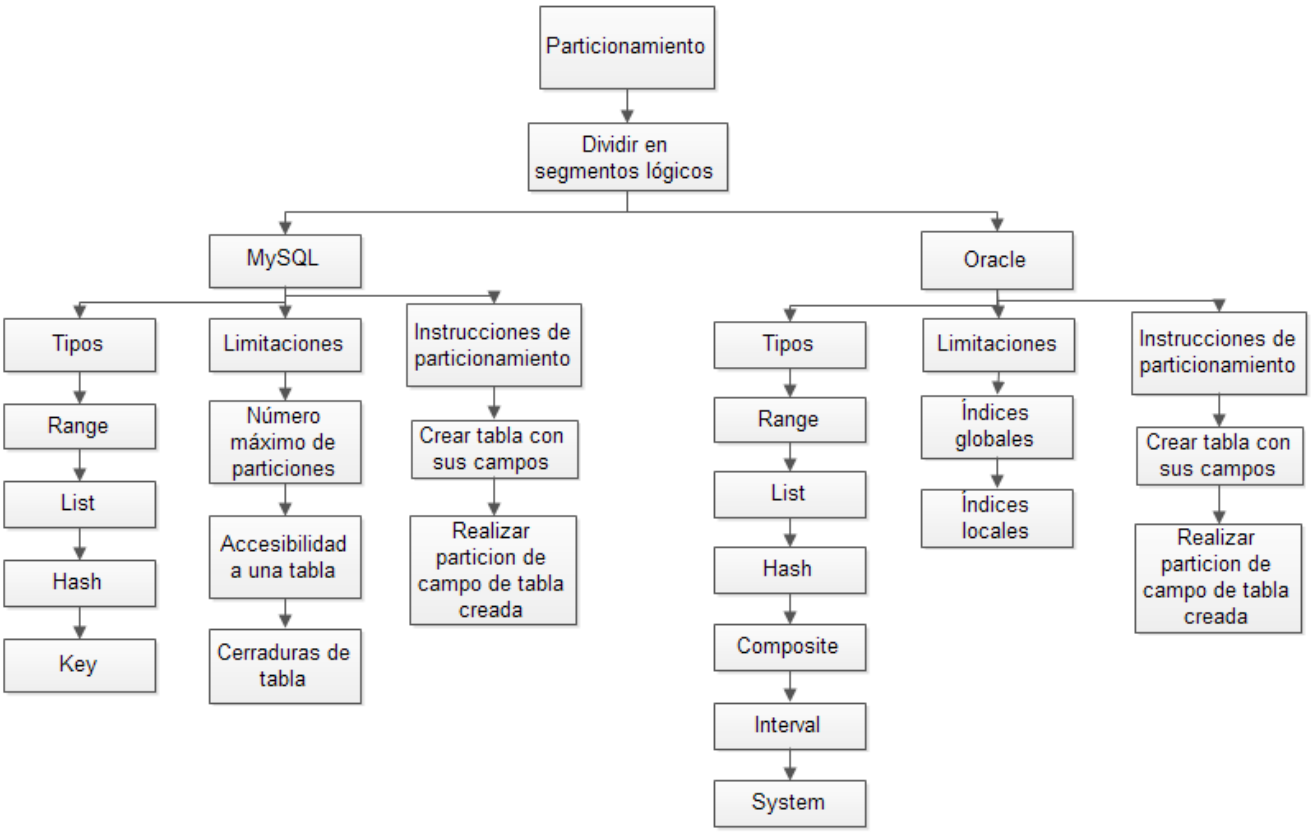
Correo: Letycv25@gmail.com

Num control:S17070155

Actividad: Mapa conceptual

Administración de bases de datos

Docente: MTI Salvador Acevedo Sandoval



PARTICIONAMIENTO

1.- ¿Qué es?¿Para qué sirve?

El particionamiento de tabla en las bases de datos es una técnica para dividir en segmentos lógicos las tablas de datos para que al momento de hacer búsquedas estas se puedan realizar sobre grupos más pequeños de datos.

2.- Tipos

Particionamiento RANGE. Este tipo de particionamiento asigna filas a particiones en función de los valores de columna que se encuentran dentro de un intervalo determinado.

Particionamiento LIST. la partición está seleccionada en función de 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 funciona en valores de columna en filas que se insertarán en la tabla. La función puede constar de cualquier expresión válida en MySQL que produzca un valor entero no negativo.

Particionamiento KEY. se proporcionan una o más columnas que se van a evaluar y el servidor MySQL proporciona su propia función de hash. Estas columnas pueden contener valores distintos de los enteros, ya que la función hash proporcionada por MySQL garantiza un resultado entero independientemente del tipo de datos de columna.

3.- Limitaciones

Número máximo de particiones. El número máximo posible de particiones para una tabla determinada que no utiliza el motor de almacenamiento NDB es 8192. Este número incluye subparticiones.

Accesibilidad a una tabla. un cambio en el SQL del servidor puede hacer que las tablas particionadas no se puedan usar.

Cerraduras de tabla. Por lo general, el proceso que ejecuta una operación de particionamiento en una tabla toma un bloqueo de escritura en la tabla. Las lecturas de estas tablas no se ven afectadas.

4.- Instrucciones para el particionamiento

```
CREATE TABLE ts (
        id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(30)
)
PARTITION BY KEY()
PARTITIONS 4;
```

Range

```
CREATE TABLE members (
    firstname VARCHAR(25) NOT NULL,
    lastname VARCHAR(25) NOT NULL,
    username VARCHAR(16) NOT NULL,
    email VARCHAR(35),
    joined DATE NOT NULL
)
PARTITION BY RANGE( YEAR(joined) ) (
    PARTITION p0 VALUES LESS THAN (1960),
    PARTITION p1 VALUES LESS THAN (1970),
    PARTITION p2 VALUES LESS THAN (1980),
    PARTITION p3 VALUES LESS THAN (1990),
    PARTITION p4 VALUES LESS THAN MAXVALUE
);
```

Oracle

5.- ¿Qué es?¿Para qué sirve?

El particionamiento de tabla en las bases de datos es una técnica para dividir en segmentos lógicos las tablas de datos para que al momento de hacer búsquedas estas se puedan realizar sobre grupos más pequeños de datos.

6.- Tipos

Particionado Range: la clave de particionado viene determinada por un rango de valores, que determina la partición donde se almacenará un valor.
Particionado Hash: la clave de particionado es una función hash, aplicada sobre una columna, que tiene como objetivo realizar una distribución equitativa de los registros sobre las diferentes particiones. Es útil para particionar tablas donde no hay unos criterios de particionado claros, pero en la que se quiere mejor el rendimiento.
Particionado List: la clave de particionado es una lista de valores, que determina cada una de las particiones.
Particionado Composite: los particionados anteriores eran del tipo simples, pues utilizamos un único método de particionado sobre una o más columnas. Oracle nos permite utilizar métodos de particionado compuestos, utilizando un primer particionado de un tipo determinado, y luego para cada partición, realizar un segundo nivel de particionado utilizando otro método. Las combinaciones son las siguientes (se han ido ampliando conforme han ido avanzando las versiones):

range-hash, range-list, range-range, list-range, list-list, list-hash y hash-hash (introducido en la versión 11g).
Particionado Interval: tipo de particionado introducido igualmente en la versión 11g. En lugar de indicar los rangos de valores que van a determinar cómo se realiza el particionado, el sistema automáticamente creará las particiones cuando se inserte un nuevo registro en la b.d.
Particionado System: se define la tabla particionada indicando las particiones deseadas, pero no se indica una clave de particionamiento. En este tipo de particionado, se delega la gestión del particionado a las aplicaciones que utilicen la base de datos.

7.- Limitaciones

Índices globales.

Índices locales.

8.- instrucciones para el particionamiento

Hash

CREATE TABLE dept (deptno NUMBER, deptname VARCHAR(32))

PARTITION BY HASH(deptno) PARTITIONS 16;

List

```
CREATE TABLE sales_list (salesman_id NUMBER(5), salesman_name VARCHAR2(30),
sales_state VARCHAR2(20),
sales_amount NUMBER(10),
sales_date DATE)
PARTITION BY LIST(sales_state)
(
PARTITION sales_west VALUES('California', 'Hawaii'),
PARTITION sales_east VALUES ('New York', 'Virginia', 'Florida'),
PARTITION sales_central VALUES('Texas', 'Illinois')
PARTITION sales_other VALUES(DEFAULT)
);
```

Composite

```
CREATE TABLE TAB2 (ord_id     NUMBER(10),
   ord_day    NUMBER(2),
   ord_month  NUMBER(2),
   ord_year   NUMBER(4)
   )
PARTITION BY RANGE(ord_year)
SUBPARTITION BY HASH(ord_id)
SUBPARTITIONS 8
 ( PARTITION q1 VALUES LESS THAN(2001)
   ( SUBPARTITION q1_h1 TABLESPACE TBS1,
     SUBPARTITION q1_h2 TABLESPACE TBS2,
     SUBPARTITION q1_h3 TABLESPACE TBS3,
     SUBPARTITION q1_h4 TABLESPACE TBS4
   ),
   PARTITION q2 VALUES LESS THAN(2002)
   ( SUBPARTITION q2_h5 TABLESPACE TBS5,
     SUBPARTITION q2_h6 TABLESPACE TBS6,
     SUBPARTITION q2_h7 TABLESPACE TBS7,
     SUBPARTITION q2_h8 TABLESPACE TBS8
   ),
   PARTITION q3 VALUES LESS THAN(2003)
   ( SUBPARTITION q3_h1 TABLESPACE TBS1,
     SUBPARTITION q3_h2 TABLESPACE TBS2,
     SUBPARTITION q3_h3 TABLESPACE TBS3,
     SUBPARTITION q3_h4 TABLESPACE TBS4
   ),
   PARTITION q4 VALUES LESS THAN(2004)
   ( SUBPARTITION q4_h5 TABLESPACE TBS5,
     SUBPARTITION q4_h6 TABLESPACE TBS6,
     SUBPARTITION q4_h7 TABLESPACE TBS7,
     SUBPARTITION q4_h8 TABLESPACE TBS8
   )
 )
```

Interval

```
CREATE TABLE T_11G(C1 NUMBER(38,0),
C2 VARCHAR2(10),
C3 DATE)
PARTITION BY RANGE (C3) INTERVAL (NUMTOYMINTERVAL(1,'MONTH'))
(PARTITION P0902 VALUES LESS THAN (TO_DATE('2009-03-01 00:00:00','YYYY-MM-DD HH24:MI:SS')));
```

System

```
create table t (c1 int,
                c2 varchar2(10),
                c3 date)
     partition by system
    (partition p1,
     partition p2,
     partition p3);
```

Referencias

Particionado de tablas en Oracle. (17 de marzo de 2020). Obtenido de https://www.dataprix.com/es/blog-it/respinosamilla/particionado-tablas-oracle

Server, M. (16 de marzo de 2020). *Chapter 23 Partitioning*. Obtenido de https://dev.mysql.com/doc/refman/8.0/en/partitioning.html