



TECNOLÓGICO
NACIONAL DE MÉXICO



Tecnológico Nacional de México

Instituto Tecnológico de Morelia

“José María Morelos y Pavón”

Especialidad: Ingeniería en Sistemas Computacionales

Semestre: Agosto - Diciembre 2019

Materia: Administración de Bases de Datos

Proyecto Final

Jaime Isai Velázquez Aguilar

Carlos Jahir Castro Cázares

Diego Zamora Delgado

Jesús Iván Lemus Cervantes

Brandon Alexis Rodríguez Molina

Profesor: Roque Trujillo Ramos

Fecha: 28/11/2019

Calificación: _____

Definición del SMBD

Como podemos ver existen multitud de base de datos que podemos utilizar.

MySQL es un sistema de base de datos relacional muy popular, y de los más utilizados para los sistemas de gestión de contenidos.

Al ser una base de datos que se utiliza en multitud de aplicaciones web existen multitud de tutoriales, foros, en la red en los que podemos encontrar la información que necesitamos.

La condición de open source de MySQL, hace que su utilización sea gratuita e incluso se pueda configurarse con total libertad.

Es un servidor multi-usuarios muy rápido y robusto de ejecución de instrucciones en paralelo, es decir, que múltiples usuarios distribuidos a lo largo de una red local o Internet podrán ejecutar distintas tareas sobre las bases de datos localizadas en un mismo servidor.

Este tipo de bases de datos puede ejecutar desde acciones tan básicas, como insertar y borrar registros, actualizar información ó hacer consultas simples, hasta realizar tareas tan complejas como nuestra la aplicación lo requiera.

Los Requerimientos básicos para MySQL son:

- 512 Mb de memoria Ram
- 1024 Mb maquina virtual
- 1 GB de espacio de disco duro
- Sistema operativo:Windows,Linux y Unix
- Arquitectura del sistema 32/64 bit
- Protocolo de red TCP/IP

El servidor de base de datos MySQL proporciona lo último en escalabilidad, luciendo la capacidad de manejar aplicaciones profundamente arraigadas con una huella de tan sólo 1 MB a la ejecución de los almacenes de datos masivos que llevan a cabo terabytes de información.

Permite a los DBA configurar el servidor de base de datos MySQL de una manera mas sencilla, con el resultado final de un rendimiento increíble.

MySQL tiene una variedad de opciones de alta disponibilidad de las configuraciones de replicación maestro / esclavo a alta velocidad.

Apoyo transaccional robusto: MySQL ofrece uno de los más poderosos motores de bases de datos transaccionales en el mercado (InnoDB).

Estas características aseguran el concepto "ACID", (aislamiento, durabilidad, atomicidad, consistencia) de transacciones, ilimitada bloqueo de filas, la capacidad de transacción distribuida, y el soporte de transacciones múltiples, donde los lectores no bloquean a los escritores y viceversa. La integridad de datos completa está asegurada a través de la integridad referencial obligadas por el servidor, los niveles de aislamiento de transacciones especializadas, y la detección de estancamiento instantánea.

También proporciona SSH y soporte SSL para asegurar conexiones seguras y protegidas.

Se pueden crear usuarios con ciertos privilegios para que los usuarios sólo vean los datos que deberían, y potentes funciones de cifrado y descifrado de datos para asegurar que los datos sensibles se protejan de accesos no autorizados.

Por último, las utilidades de copia de seguridad y recuperación proporcionadas a través de MySQL y proveedores de software de terceros permiten la copia de seguridad lógica y física completa, así como la recuperación completa y el punto en el tiempo.

Definición del espacio de la base de datos y table space

Para la asignación del espacio que será usado por nuestra base de datos, usaremos tablespace que serán asignados a cada una de las tablas de la base de datos, ya que nuestro sistema gestor de base de datos, no nos permite la asignación global de un solo tablespace para toda la base de datos si no que al momento de la creación de la base de datos se crea un directorio donde se aguardan cada una de las tablas de la base de datos, así que se crearan tablespace para cada una de las tablas de la base de datos para así llevar un control del espacio de la base de datos médico.

Los tamaños de los tablespace por cada tabla serán los siguientes:

Tabla	Tamaño TableSpace
PACIENTE	350 KB
COLONIA	50 KB
POBLACION	12 KB
MUNICIPIO	12 KB
ESTADO	2 KB
TIPO_PACIENTE	100 B
TELEFONO	64 KB
OCUPACION	56 KB
FACTURA	20 KB
CITA	550 KB
CITA-ESTUDIO	150 KB
ESTUDIO	150 KB
CITA_MEDICAMENTO_NUEVO	100 KB
CITA_MEDICAMENTO_ANTERIOR	100 KB

MEDICAMENTO	54 KB
--------------------	--------------

Estos son los valores iniciales de los tablespace que se usaran por cada una de las tablas, aunque parece que el espacio reservado para los tablespace es muy grande, se indicaron así ya que este espacio es suficiente como para tener 1000 registros por cada tabla más sus índices y otros elementos, así que esto ayudara a que no se tenga que alterar los tablespace en mucho tiempo, aun así si el espacio es superado se configurara que los tablespace se expandan en una cantidad de 5KB cada vez que se necesite mas espacio hasta un máximo de 1GB.

El tipo de almacenamiento secundario que se utilizará será en un disco duro (HDD), que contendrá la base de datos y la memoria primario será administrada por el sistema gestor de base de datos igualmente que la forma de organización de archivos.

Tablespace por usuario

En nuestra base de datos se usarán dos usuarios que serán las entidades que interactuaran con la base de datos, el usuario del médico que será quien tendrá control completo sobre la base de datos y recepcionista quien ingresara algunos registros sobre algunas tablas, para que estos usuarios trabajen sobre un espacio de memoria asignado se usaran 2 tablespace para cada usuario.

Usuario	Tamaño tablespace
Medico	600 KB
Recepcionista	600 KB

Se usará así ya que tanto el medico como la recepcionista ingresan datos a la base de datos continuamente sobre todo en las tablas de paciente y cita. Con un espacio de estos tamaños es suficiente para que se trabaje continuamente durante un gran periodo de tiempo sin hacer modificaciones en los tablespace igualmente si estos sobrepasan la capacidad asignada, crecerán automáticamente en 10KB, hasta un máximo de 1GB.

Usuarios y roles

Para nuestro proyecto (bd_medico) definiremos dos tipos de usuarios que serán médico y recepcionista.

```
-- crear usuarios
create user 'medico'@'localhost'
identified by '12345';

create user 'recepcionista'@'localhost'
identified by '12345';
```

Cada usuario desempeñará un rol distinto dentro de la base de datos, mientras que el médico tiene todos los privilegios de la base de datos, el recepcionista no tiene por qué tener acceso a todo, como al historial de la cirugía o a la historia clínica de un paciente, ya que es algo que no le corresponde. A continuación se muestra imágenes con los privilegios de cada usuario:

```
grant all privileges on bd_medico.*
to 'medico'@'localhost';
```

*El médico con todos los privilegios en la bd_medico para todas las tablas

```
grant delete, insert, update  
on bd_medico.cita  
to 'repcionista'@'localhost';
```

*permisos de recepcionista en la bd_medico en la tabla cita.

El usuario recepcionista podrá visualizar todas las citas, además de poder cambiar de fecha una, así como también cancelar o eliminar alguna o todas.

```
grant INSERT, UPDATE, Delete  
on bd_medico.factura  
to 'repcionista'@'%';
```

*permisos de recepcionista en la bd_medico en la tabla factura.

El usuario recepcionista podrá agregar facturas, además de poder cambiar de fecha una, así como también cancelar o eliminar alguna o todas.

```
grant delete, insert, update, select  
on bd_medico.cita_estudio  
to 'repcionista'@'localhost';
```

*permisos de recepcionista en la bd_medico en la tabla cita_estudio.

El usuario recepcionista podrá agregar estudios por cita, además de poder cambiar de fecha una, así como también cancelar o eliminar alguna o todas.


```
grant insert, select
on bd_medico.cita_medicamento_anterior
to 'repcionista'@'localhost';
```

*permisos de recepcionista en la bd_medico en la tabla cita_medicamento_anterior.

El usuario recepcionista podrá agregar medicamentos a una cita, así como también visualizarlos.

```
grant insert, select
on bd_medico.cita_medicamento_nuevo
to 'repcionista'@'localhost';
```

*permisos de recepcionista en la bd_medico en la tabla cita_medicamento_anterior.

El usuario recepcionista podrá agregar medicamentos a una cita, así como también visualizarlas.

Configuración del SGBD para soportar la carga de trabajo

En la base de datos “Medico” es necesario tener una configuración para tener un buen funcionamiento de esta, así que en este apartado explicaremos cómo optimizaremos un servidor MySQL para soportar nuestra carga de trabajo.

¿Para qué queremos optimizar un servidor MySQL? Pues para que pueda servir las peticiones más rápido usando menos recursos de CPU, RAM e I/O de disco, o simplemente para que aproveche mejor los recursos del sistema para garantizar una mejor estabilidad y una mayor velocidad de respuesta al acceder a datos almacenados en las bases de datos.

Las configuraciones del servidor MySQL se realizarán desde un único archivo (my.cnf)

A continuación, vamos a especificar algunos de los parámetros generales que influyen en el rendimiento y en la estabilidad de MySQL y que se deben parametrizar en el my.cnf. Lo dividiremos en tres categorías Memoria, Procesos y conexiones

Memoria

- **query_cache_type:** Sirve para activar o desactivar cache, pondremos 1 para activar el cache de consultas.
- **query_cache_size:** Este parámetro especifica el tamaño del cache de consultas, asignaremos 64 MB de RAM por cada 1 GB de memoria física usable.
- **sort_buffer_size:** Con este parámetro configuramos el tamaño del cache de búsquedas de MySQL, configuraremos 1 MB por cada 1 GB de memoria RAM física disponible por ser lo recomendable.
- **read_buffer_size:** Con este parámetro configuramos el tamaño del cache de lecturas de MySQL, configuraremos 1 MB por cada 1 GB de memoria RAM física disponible por ser lo recomendable.

Procesos

- **thread_cache_size:** Es el número máximo de hilos de ejecución que se pueden cachear y rehusar, configuraremos entre 32 y 64 para un uso normal.
- **thread_concurrency:** Especifica el número máximo de hijos de ejecución o procesos abiertos de MySQL, configuraremos 2 por cada núcleo de CPU disponible.
- **table_cache:** Especifica el máximo de tablas abiertas entre todos los threads o hilos de ejecución de MySQL, daremos un valor de 64 por ser lo recomendable.

Conexiones

- **max_connections:** Especifica el número máximo de conexiones totales que puede aceptar el servidor MySQL al mismo tiempo. Asignaremos 2, debido a que en la base de datos “medico” solo existen dos usuarios que continuamente estarán conectado.
- **wait_timeout:** Es el tiempo de espera que tarda MySQL en cerrar una conexión. Asignaremos 2 segundos.