

**Nombre:** Eduardo Neri Martínez

**Matricula:** 87339

**Grupo:** K045

**Materia**: K045 (21) BASES DE DATOS I

**Docente:** Mtro. Emmanuel Hernández Ortiz

Actividad de aprendizaje 1. Bases de datos

Ecatepec de Morelos, Estado de México a 12 de Enero del 2019

**Objetivo**

Identificar y describir las distintas bases de datos.

**Instrucciones**

Realiza una investigación sobre las siguientes bases de datos:

1. Mysql.

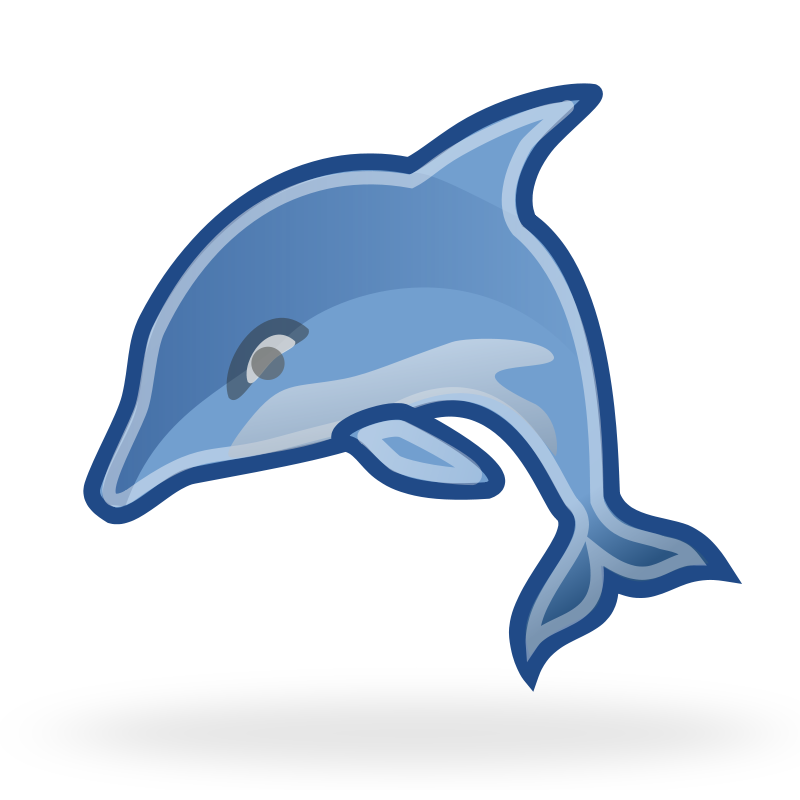
2. PostgreSQL

3. Oracle.

Tomando como base la información recabada, realiza un reporte, en el cual desarrolles los aspectos que se describen en la rúbrica de la actividad.

**MySql**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia pública general/Licencia comercial por Oracle Corporation, considerada como una de las bases datos de código abierto más popular del mundo, junto con Oracle y Microsoft SQL Server, sobre todo para entornos web.



**Componentes y características MySql**

La base de datos MySQL contiene los siguientes componentes:

**mysqld**: es el motor de MySQL. Es el programa servidor el cual proporciona las bases de datos que se encuentran en memoria. En los sistemas operativos Windows el servidor MySQL son los programas **mysqld-nt** o **mysql-max-nt**. Este último provee soporte para tablas BDB.

**MySQLCC**: Ésta interfaz gráfica permite controlar todos los aspectos del servidor MySQL. Permite crear bases de datos, tablas, usuarios, cambiar permisos, acceder y cambiar la información almacenada, etc., de una forma muy intuitiva y muy fácil

**mysql**: Ésta interfaz de texto permite también controlar todos los aspectos del servidor pero la interfáz es solamente texto.

**mysqlimport** permite importar datos a través de archivos de texto. Provee una interfaz de texto para los comandos LOAD, DATA, INFILE.

**mysqldump** permite hacer copias o respaldos de la información almacenada para restaurarlos en el mismo servidor o para exportarlo a otros servidores.

**mysqladmin** permite administrar el servidor con una interfáz gráfica y de una forma muy sencilla

**mysqlcheck** permite revisar la salud de la base de datos. Permite también reparar dichas bases si fuera necesario.

**myisamchk** revisa las tablas y realiza operaciones de reparación.

**myisampack** crea versiones comprimidas de solo-lectura de tablas MyISAM

**Privilegios**

En MySQL existen cinco niveles distintos de privilegios:

**Globales**: se aplican al conjunto de todas las bases de datos en un servidor. Es el nivel más alto de privilegio, en el sentido de que su ámbito es el más general.

**De base de datos**: se refieren a bases de datos individuales, y por extensión, a todos los objetos que contiene cada base de datos.

**De tabla**: se aplican a tablas individuales, y por lo tanto, a todas las columnas de esas tabla.

**De columna**: se aplican a una columna en una tabla concreta.

De rutina: se aplican a los procedimientos almacenados. Aún no hemos visto nada sobre este tema, pero en MySQL se pueden almacenar procedimietos consistentes en varias consultas SQL.

**Usuarios**

Aunque en la versión 5.0.2 de MySQL existe una sentencia para crear usuarios, CREATE USER, en versiones anteriores se usa exclusivamente la sentencia GRANT para crearlos.

En general es preferible usar GRANT, ya que si se crea un usuario mediante CREATE USER, posteriormente hay que usar una sentencia GRANT para concederle privilegios.

**Sistemas operativos**

El servidor MySQL y los clientes soportan los principales sistemas operativos existentes:

* Windows (Windows 95, 98, Me, NT, 2000 y XP). *MySQL no soporta MS-DOS o Windows 3.1*
* La familia UNIX y derivados:
* Los sistemas BSD (ejemplo: FreeBSD, OpenBSD, NetBSD, etc.),
* El sistema operativo MacOS X
* System V
* Solaris
* HP-UX
* Linux (como Fedora, RedHat, SuSE, Debian, Mandrake, Gentoo, Ubuntu, etc.)
* Novell Netware 6.5 y superior.

**Clientes**

Existen diferentes aplicaciones clientes para comunicarse con el servidor. Entre ellas:

* La Librería en C libmysqlclient: Esta librería es la más usada para crear aplicaciones en el lenguaje C que se comuniquen con el servidor MySQL. De esta librería se han derivado varias aplicaciones para conectar a PHP, Pascal, Pyton y otros con la base de datos.
* El conector ODBC: Este conector provee compatibilidad entre las aplicaciones que usan el lenguaje ODBC para que puedan comunicarse con la base de datos MySQL.
* El conector J: Este conector provee compatibilidad entre las aplicaciones Java que usan el lenguaje JDBC para que puedan comunicarse con la base de datos MySQL.

**Configuraciones**

Aumentar el tamaño de cache de consultas.

Cuando se habilita el cache de consultas en memoria y se dispone de suficiente de ésta, el desempeño del servidor se incrementa. El valor predeterminado del tamaño cache de consultas —query\_cache\_size— es 0, es decir está desactivado. Los valores permitidos son enteros múltiplos de 1024 (bytes). Para establecer un tamaño de cache de consultas de 32 MiB, el valor correspondiente para query\_cache\_size sería 33882112 bytes.

1. Ingresar a MySQL™ como root:

mysql -u root -p

1. Verificar el valor de la variable query\_cache\_size ejecutando lo siguiente:

SHOW VARIABLES LIKE '%query\_cache%';

La salida será similar a la siguiente

+------------------------------+---------+

| Variable\_name | Value |

+------------------------------+---------+

| have\_query\_cache | YES |

| query\_cache\_limit | 1048576 |

| query\_cache\_min\_res\_unit | 4096 |

| query\_cache\_size | 0 |

| query\_cache\_type | ON |

| query\_cache\_wlock\_invalidate | OFF |

+------------------------------+---------+

6 rows in set (0.00 sec)/pre>

1. Ejecutar para cambiar el valor de query\_cache\_size a 32 MiB:

SET GLOBAL query\_cache\_size = 33882112;

1. Verificar el cambio:

SHOW VARIABLES LIKE '%query\_cache%';

La salida será similar a la siguiente:

+------------------------------+----------+

| Variable\_name | Value |

+------------------------------+----------+

| have\_query\_cache | YES |

| query\_cache\_limit | 1048576 |

| query\_cache\_min\_res\_unit | 4096 |

| query\_cache\_size | 33882112 |

| query\_cache\_type | ON |

| query\_cache\_wlock\_invalidate | OFF |

+------------------------------+----------+

6 rows in set (0.00 sec)

1. Salir de MySQL.

exit;

1. El cambio prevalecerá hasta que sea reiniciado MySQL. Editar el archivo /etc/my.cnf para que el cambio sea permanente:

vi /etc/my.cnf

1. Añadir la siguiente opción en la sección [mysqld]:

query\_cache\_size = 32M

1. Reiniciar el servicio para aplicar los cambios:

service mysqld restart

1. Ingresar a MySQL™ para verificar el estado del cache de consultas:

mysql -u root -p

1. verificar el estado del cache de consultas:

SHOW STATUS LIKE '%Qcache%';

Soporte para UTF-8.

Algunas aplicaciones como vTigerCRM requieren sé que configuré UTF-8 como codificación predeterminada. Edite el archivo /etc/my.cnf:

vi /etc/my.cnf

1. Añadir las siguientes líneas resaltadas:

[mysql]

default-character-set=utf8

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

user=mysql

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

skip-name-resolve

query\_cache\_size = 32M

collation\_server=utf8\_unicode\_ci

character-set-server=utf8

init\_connect='SET collation\_connection = utf8\_general\_ci'

init\_connect='SET NAMES utf8'

[mysqld\_safe]

log-error=/var/log/mariadb/mariadb.log

pid-file=/var/run/mariadb/mariadb.pid

#

# include all files from the config directory

#

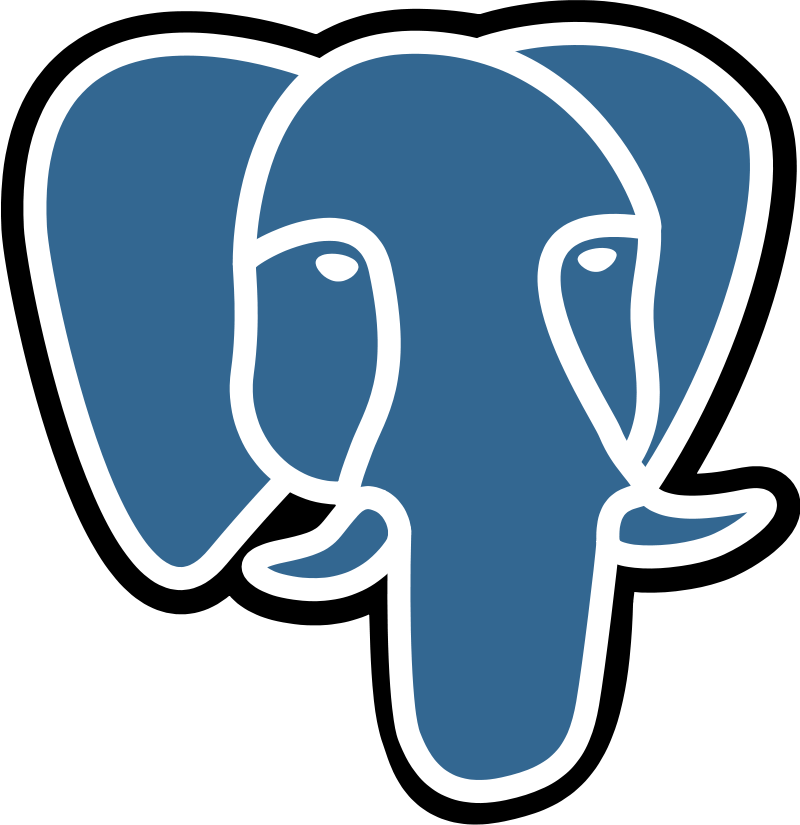
!includedir /etc/my.cnf.d

**Posgresql**

PostgreSQL es un sistema de gestión de bases de datos relacionales orientados a objetos y de código abierto, publicado bajo la licencia PostgreSQL.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL no tiene un gestor de defectos, haciendo muy difícil conocer el estado de sus defectos.



**Componentes y características Posgresql**

PostgreSQL se utiliza en diversos ámbitos, aquí te dejamos algunos de los ejemplos donde se hace uso de este gestor de bases de datos:

* Almacenamiento de datos (DWH).
* En servicios como Amazon Web Services Redshift.
* Para procesado de datos, almacenado tanto en la propia instancia como en otros servicios que puedan conectarse.
* En sistemas de información geográfica, como el servicio de mapas web o también en servicios móviles OpenStreetMap.
* En bases de datos para servicios web.
* En CMS como Drupal o WordPress.
* En la conocida base de datos de cine IMDb.

**pg\_hba.conf**: se utiliza para definir los diferentes tipos de accesos que un usuario tiene en el cluster.

**pg\_ident.conf**: se utiliza para definir la información necesaria en el caso que utilicemos un acceso del tipo ident en pg\_hba.conf .

**postgresql.conf**: para cambiar todos los parámetros de configuración que afectan al funcionamiento y al comportamiento de PostgreSQL en nuestra computadora.

**Pg\_hba.Conf**

Se utiliza para definir como, donde y desde que sitio un usuario puede utilizar nuestro cluster PostgreSQL. Todas las lineas que empiezen con el caracter # se interpretan como comentarios. El resto debe de tener el siguiente formato:

[Tipo de conexion][database][usuario][IP][Netmask][Tipo de autentificacion][opciones]

Dependiendo del tipo de conexion y del tipo de autentificacion, [IP],[Netmask] y [opciones] pueden ser opcionales. Vamos a explicar un poco como definir las reglas de acceso. El tipo de conexion puede tener los siguientes valores, local, host, hostssl y hostnossl. El tipo de metodo puede tener los siguientes valores, trust, reject, md5, crypt, password, krb5, ident, pam o ldap.

**Postgresql.Conf**

Los cambios que realicemos en este fichero afectaran a todas las bases de datos que tengamos definidas en nuestro cluster PostgreSQL. La mayoria de los cambios se pueden poner en produccion con un simple 'reload' (/usr/local/bin/pg\_ctl -D /var/pgsql/data reload), otros cambios necesitan que arranquemos de nuevo nuestro cluster (/usr/local/bin/pg\_ctl -D /var/pgsql/data restart).

**Usuarios**

Existe un comando en la carpeta de binarios de postgres que se llama createuser. Con este comando es posible crear un nuevo ROLE con permisos de acceso a la base de datos.

En PostgreSQL no existen los usuarios y los grupos como tal, sino que tenemos roles. El comando createuser es un wrapper que crea un ROLE con permisos de conexión.

Para poder crear un usuario (ROLE) es necesario tener permisos de super usuario o privilegios de CREATEROLE.

Debemos tener especial cuidado. Los usuarios tienen permisos de conexión, es decir, pueden conectar a nuestro cluster. Esto no quiere decir que tengan permisos de acceso a nuestra base de datos, para ello tenemos otros comandos.

La sintaxis para ejecutar createuser es la siguiente:

createuser [connection-option…] [option…] [username]

Por ello, en las opciones de conexión se debe utilizar las habituales para conectarnos y para ayudarnos a ello podemos hacer uso de las variables de entorno como PGHOST, PGPORT y PGUSER.

Por ejemplo, si queremos crear un usuario llamado todopostgre podemos utilizar la siguiente sintaxis:

$ createuser --interactive todopostgre

1

$ CREATEUSER --INTERACTIVE TODOPOSTGRE

Con este comando se crea un usuario llamado todopostgre.

**Bibliografía**

JulioProfe. (2017). Algebra Lineal. 2018, de Algebra Sitio web: [**https://julioprofe.net/categoria/algebra-lineal/**](https://julioprofe.net/categoria/algebra-lineal/)

[**https://www.edx.org/es/course/algebra-lineal-mexicox-acf-0903-1x**](https://www.edx.org/es/course/algebra-lineal-mexicox-acf-0903-1x)

[**https://www.youtube.com/watch?v=En6Xz6gyZT4**](https://www.youtube.com/watch?v=En6Xz6gyZT4)

[**https://www.youtube.com/watch?v=6u2Ds\_BdMKI**](https://www.youtube.com/watch?v=6u2Ds_BdMKI)