

Práctica 1: Instalación de SQL-Server

Fundamentos de Bases de Datos

López Soto Ramses Antonio

26 de agosto de 2019

Objetivo

Instalar el sistema manejador de datos (**SMBD**) *SQL-Server* en Linux.

Introducción

Las *bases de datos* se definen como un conjunto de datos que están relacionados y que nacen de la necesidad de almacenar información. Ésto se refiere a que las bases de datos logran preservar la información a lo largo del tiempo y el deterioro y que su manejo se lleva a cabo a través de un SMBD.

Los *sistemas manejadores de bases de datos (SMBD)* son un tipo de software que se encarga de manejar de forma sencilla y ordenada un conjunto de datos que se convierten en información importante para una empresa u organización.

Los propósitos principales de un SMBD son:

- Acceso rápido a datos.
- Manejar los datos de acuerdo a lo requerido por el cliente.
- Registrar el uso de la base de datos.
- La interacción con el manejador de datos.

Para todos los SMBD, el fin es el mismo y existen los casos que los hacen diferentes entre sí; cada uno tiene ciertas ventajas sobre los demás. Actualmente, en el mercado existen distintos SMBD que se clasifican de acuerdo a la licencia que éstos poseen, ya sea de código abierto o de manejo mediante web; tales como: *MySQL, SQLite, PostgreSQL, Microsoft SQL Server, Oracle*.

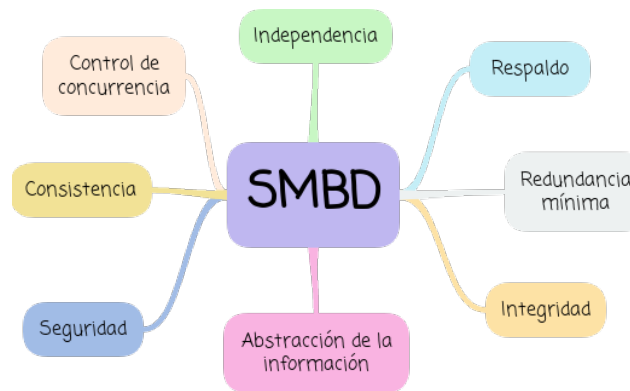


Figure 1: Características de SGBD

El SGBD que nos interesa es Microsoft SQL Server, pues posee características tales como:

- El soporte de transacciones.
- Estabilidad y seguridad.
- Soporte para procedimientos almacenados.
- Potente entorno gráfico que permite el uso de comandos DDL y DML.

y además se diferencia del resto de SGBD.

SGBD	Microsoft SQL-Server	MySQL	PostgreSQL	ORACLE
Descripción	SGBD basado en el modelo relacional	SGBD relacional multihilo y multiusuario	SGBD dirigido a una comunidad de desarrolladores	SGBD objeto-relacional
Plataformas	Microsoft Windows	Microsoft Windows, Linux Unix	Microsoft Windows, Mac OS, Linux Unix.	Microsoft Windows, Linux Unix
Lenguajes de programación	T-SQL	C, C++, Pascal, PHP, etc.	PL/ PostgreSQL, C, C++, Java, PL/Java, Web	PL/SQL, PHP, Java, .NET, XML
Herramientas	Management Studio	MySQL Workbench	Power Designer	Oracle designer
Almacenamiento	2 GB	200 MB	1.5 GB	11GB
Transacciones	Sí	Sí	Sí	Sí
Licencia	Libre	Libre	Libre	Libre
Ventajas	Agilidad de análisis y administración de datos	La más usada.	Orientada a objetos	Considerado uno de los más completos SGBD

Figure 2: Tabla de diferencias

Los SGBD pueden hacer uso de lenguajes de programación, en nuestro caso, se utiliza algo llamado *T-SQL*. El Transact-SQL se trata de un lenguaje muy potente que permite efectuar casi cualquier cosa que sea relacionado a la base de datos. Va más allá del lenguaje SQL pues

posee las características necesarias para que sea considerado un lenguaje de programación, las cuales permiten definir la lógica para mantener la información.

El Transact-SQL permite realizar acciones como:

- La definición de bloques de instrucciones SQL (unidades de ejecución).
- Ejecuciones condicionales.
- Ejecuciones iterativas.
- Manipular tupla a tupla el resultado de una consulta.

pero no permite crear interfaces de usuario ni aplicaciones ejecutables.

2014	2016	2017
<ul style="list-style-type: none"> • Procesamiento de Transacciones En Línea (OLTP). • Almacén de columnas en memoria. • extensión del grupo de búferes a SSD. • Siempre puede estar encendido. • Cifrado de datos transparente. • Compatibilidad con el cifrado de copias de seguridad. • Separación de tareas. • Copias de seguridad en Azure. • Imágenes virtuales en Azure y servicios de integración administrados como servidor. 	<ul style="list-style-type: none"> • Análisis operativo en tiempo real. • Grupos de disponibilidad básicos. • Cifrado en reposo y en movimiento. • Enmascaramiento dinámico de datos. • base de datos elástica. • Tablas temporales • Compatibilidad con <u>Json</u> y <u>PolyBase</u> para consultas <u>T-SQL</u>. 	<ul style="list-style-type: none"> • Procesamiento de consultas adaptable. • Funcionamiento en contenedores <u>docker</u> y compatibilidad con datos de gráficos.

Figure 3: Diferencias entre versiones

Así como existen las bases de datos SQL, también existen las *bases de datos NoSQL*. Las bases de datos no relacionales son un enfoque hacia la gestión de datos y el diseño de base de datos que es útil para grandes conjuntos de datos distribuidos. En pocas palabras, almacenan grandes volúmenes de datos (imágenes, videos, etc).

<u>NoSQL</u>	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Versatilidad • Crecimiento horizontal • Disponibilidad de recursos • Optimización 	<ul style="list-style-type: none"> • Atomicidad • Documentación del software • Estándares del lenguajes • Herramientas <u>GUI</u>

Figure 4: Bases de datos no relacionales

SQL-Server en Linux

Microsoft SQL-Server funciona bien en Microsoft Windows, pero ¿cómo podemos utilizarlo en Linux? Para eso existe un contenedor conocido como *Docker*.

Docker es uno de los proyectos más conocidos y utilizados en temas de virtualización. No se trata de un sistema operativo como tal, sino que es una plataforma de código abierto que hace uso de las funciones de aislamiento de recursos del kernel de Linux para poder dar lugar a contenedores independientes. También cuenta con una serie de repositorios, similares a los de Linux, donde los usuarios publican sus propios contenedores de manera que los usuarios que los necesiten los puedan bajar rápidamente desde allí. Generalmente se usa para la simplificación de configuraciones y la gestión de proyectos.

Desarrollo

La instalación de SQL-Server en Linux es un proceso largo pero fácil de seguir. Sólo basta con seguir los siguientes pasos al pie de la letra para que no se ocasione ningún inconveniente durante el proceso.

Pasos a seguir:

- 1.- Tenemos que actualizar nuestro repositorio local con el siguiente comando:

```
ramses@ramses:~$ sudo apt update
```

- 2.- Al terminarse de ejecutar el comando anterior instalaremos algunos paquetes como prerequisite:

```
ramses@ramses:~$ sudo apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```

3.- Agregamos el repositorio al sistema:

```
ramses@ramses:~$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
OK
```

4.- Agregamos al repositorio de Docker las fuentes APT:

```
ramses@ramses:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

5.- Ahora volvemos a actualizar el repositorio local:

```
ramses@ramses:~$ sudo apt update
```

6.- Verificar si se obtiene la siguiente salida al ejecutar el siguiente comando:

```
ramses@ramses:~$ apt-cache policy docker-ce
docker-ce:
  Instalados: 5:19.03.1~3-0~debian-stretch
  Candidato: 5:19.03.1~3-0~debian-stretch
  Tabla de versión:
```

7.- Instalar Docker:

```
ramses@ramses:~$ sudo apt install docker-ce
```

8.- Verificar si Docker se está ejecutando:

```
ramses@ramses:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-08-19 14:30:58 CDT; 1min 29s ago
     Docs: https://docs.docker.com
    Main PID: 5412 (dockerd)
    CGroup: /system.slice/docker.service
            └─5412 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

y Docker ya estará funcionando.

9.- Hacemos pull de la imagen SQL Server 2017 de Microsoft Container Registry:

```
ramses@ramses:~$ sudo docker pull mcr.microsoft.com/mssql/server:2017-latest
```

10.- Para ejecutar la imagen, utilizar el siguiente comando que se colocará por partes:

```
ramses@ramses:~$ sudo docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=<YourStrong@Passw0rd>" \
> -p 1433:1433 --name sql1 \
> -d mcr.microsoft.com/mssql/server:2017-latest
fe23209f7018c9ee21ca7e42e42962f440ada3851721132fcdad45fd9690f650
ramses@ramses:~$
```

11.- Para poder visualizar los contenedores, ejecutar el siguiente comando:

```
ramses@ramses:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
fe23209f7018   mcr.microsoft.com/mssql/server:2017-latest  "/opt/mssql/bin/sqls..." 30 seconds ago Up 28 seconds  0.0.0.0:1433->1433/t
cp            sql1
ramses@ramses:~$
```

12.- Ahora vamos a cambiar la contraseña. Colocamos la contraseña que elegimos en el paso 10 y escribimos otra:

```
ramses@ramses:~$ sudo docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd \
> -S localhost -U SA -P "<YourStrong@Passw0rd>" \
> -Q 'ALTER LOGIN SA WITH PASSWORD= "!"'

```

13.- Utilizamos el siguiente comando para interactuar en el contenedor colocando la contraseña elegida en el paso 12:

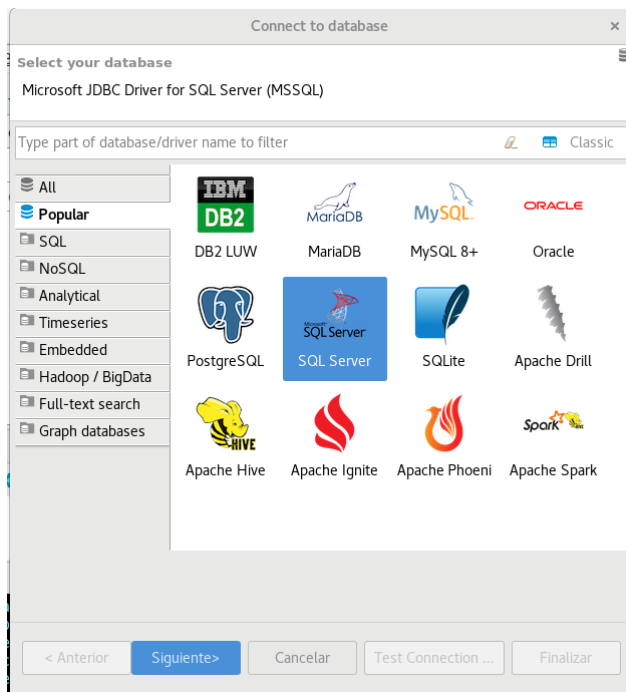
```
ramses@ramses:~$ sudo docker exec -it sql1 "bash"
root@fe23209f7018:/# /opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P "!"
1> CREATE DATABASE TestDB
2>
```

y así podemos usar SQL Server a través de Docker.

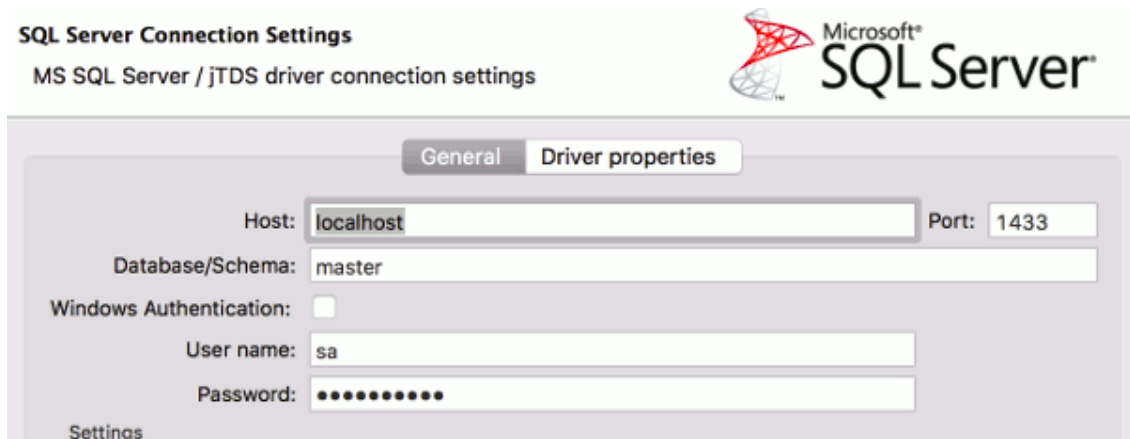
14.- Para poder utilizar a base de datos, además desde la terminal, instalaremos *DBeaver*. Descargamos el paquete .deb de <https://dbeaver.io/download/> y en la carpeta donde se descargó, ejecutar el siguiente comando:

```
ramses@ramses:~/Descargas$ sudo dpkg -i dbeaver-ce_6.1.5_amd64.deb
```

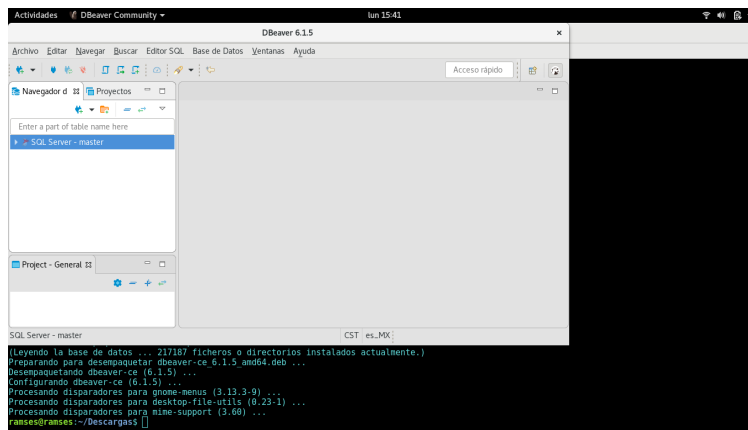
15.- Al finalizar la instalación de *DBeaver*, abrir el programa y elegir SQL Server:



16.- Después de seleccionar SQL Server, se solicitará el usuario (siempre es SA) y la contraseña. Al ingresar los datos, dar click en finalizar (finish):



17.- Y para finalizar la base de datos en DBeaver se verá así:



Si se siguen los pasos al pie de la letra, ya se podrá trabajar con SQL-Server 2017.

Conclusión

Como se pudo ver, la instalación de SQL-Server es un proceso largo, un poco más si se desea trabajar con Linux, pues es necesario el uso de un contenedor a diferencia de Microsoft Windows. Pero, se verá que es una herramienta realmente útil para trabajar con el manejo de las bases de datos.

Referencias

- <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-debian-9>
- <https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-2017#pivots=cs1-bash>
- <https://es.scribd.com/doc/137485506/Cuadro-Comparativo-Sistemas-Gestores-De-Bases-De-Datos>
- https://www.ecured.cu/Transact_SQL
- <https://medium.com/@marlonmanzo/sql-vs-nosql-ventajas-y-desventajas-849ccc9db3d4>
- <https://openwebinars.net/blog/5-usos-practicos-de-Docker-en-el-dia-a-dia/>
- <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>
- <https://concepto.de/base-de-datos/>