comment-installer-microsoft-sql-server-sur-debian-12

MSSQL Server or Microsoft SQL Server is an RDBMS (Relational Database Management System) developed by Microsoft. As a database server, it is used to store data for your applications. The SQL Server is ideal for multiple purposes of applications, you can use MS SQL Server as the database for your desktop applications, use it for your web applications, or any small server applications.

In this guide, we'll show you how to install Microsoft SQL Server 2022 Edition on Debian 12 server. You will also learn how to install MS SQL Tools for interacting with MS SQL Server. In addition to that, you will learn some basic queries of MS SQL Server.

Prerequisites

To complete this guide, ensure you have the following:

- A Debian 12 server.
- A non-root user with sudo privileges.

Adding Microsoft SQL Repository

In this guide, we'll be using the name **MS SQL Server** as a reference to the **Microsoft SQL Server**.

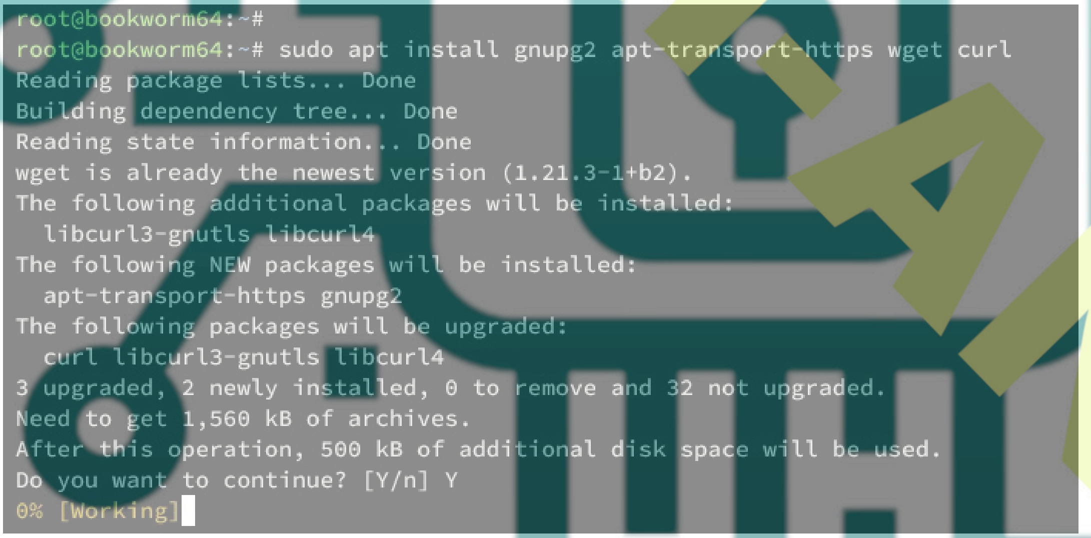
Before installing MS SQL Server, you must add the MS SQL Server repository to your Debian machine. And in this example, you will add the MS SQL Server 2022 repository to your Debian 12 server.

First, update your Debian repository and upgrade all packages using the command.

```
sudo apt update && sudo apt upgrade
```

Now install package dependencies using the following command. Type y to confirm the installation.

```
sudo apt install gnupg2 apt-transport-https wget curl
```



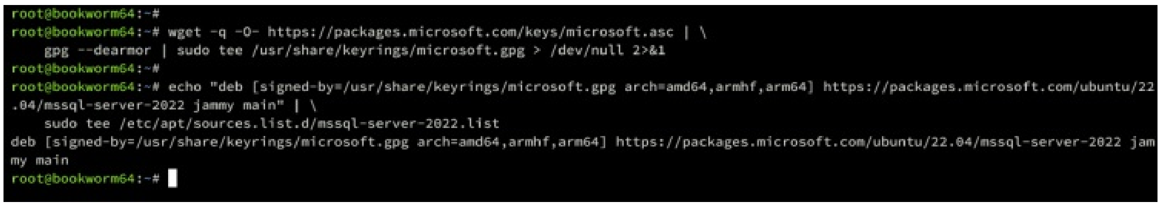
```
root@bookworm64:~#
root@bookworm64:~# sudo apt install gnupg2 apt-transport-https wget curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.3-1+b2).
The following additional packages will be installed:
  libcurl3-gnutls libcurl4
The following NEW packages will be installed:
  apt-transport-https gnupg2
The following packages will be upgraded:
  curl libcurl3-gnutls libcurl4
3 upgraded, 2 newly installed, 0 to remove and 32 not upgraded.
Need to get 1,560 kB of archives.
After this operation, 500 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
0% [Working]
```

After package dependencies are installed, add the GPG key for the MS SQL Server repository by running the command below.

```
wget -q -O- https://packages.microsoft.com/keys/microsoft.asc | \
gpg --dearmor | sudo tee /usr/share/keyrings/microsoft.gpg > /dev/null 2>&1
```

Then, add the MS SQL Server repository with the command below. In this guide, you will install the MS SQL Server 2022.

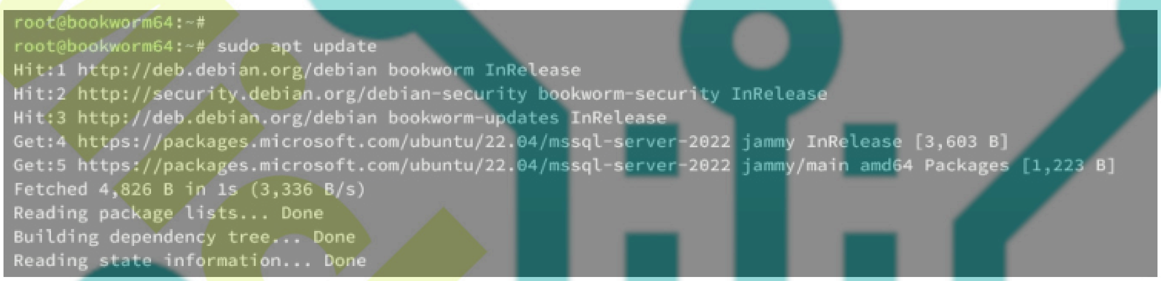
```
echo "deb [signed-by=/usr/share/keyrings/microsoft.gpg arch=amd64,armhf,arm64] https://packages.microsoft.com/ubuntu/22.04/mssql-server-2022 jammy main" | \
sudo tee /etc/apt/sources.list.d/mssql-server-2022.list
```



```
root@bookworm64:~#
root@bookworm64:~# wget -q -O- https://packages.microsoft.com/keys/microsoft.asc | \
    gpg --dearmor | sudo tee /usr/share/keyrings/microsoft.gpg > /dev/null 2>&1
root@bookworm64:~#
root@bookworm64:~# echo "deb [signed-by=/usr/share/keyrings/microsoft.gpg arch=amd64,armhf,arm64] https://packages.microsoft.com/ubuntu/22
.04/mssql-server-2022 jammy main" | \
    sudo tee /etc/apt/sources.list.d/mssql-server-2022.list
deb [signed-by=/usr/share/keyrings/microsoft.gpg arch=amd64,armhf,arm64] https://packages.microsoft.com/ubuntu/22.04/mssql-server-2022 jam
my main
root@bookworm64:~#
```

After adding the GPG key and repository of MS SQL Server, use the command below to refresh your Debian repository and retrieve package information for MS SQL Server.

```
sudo apt update
```



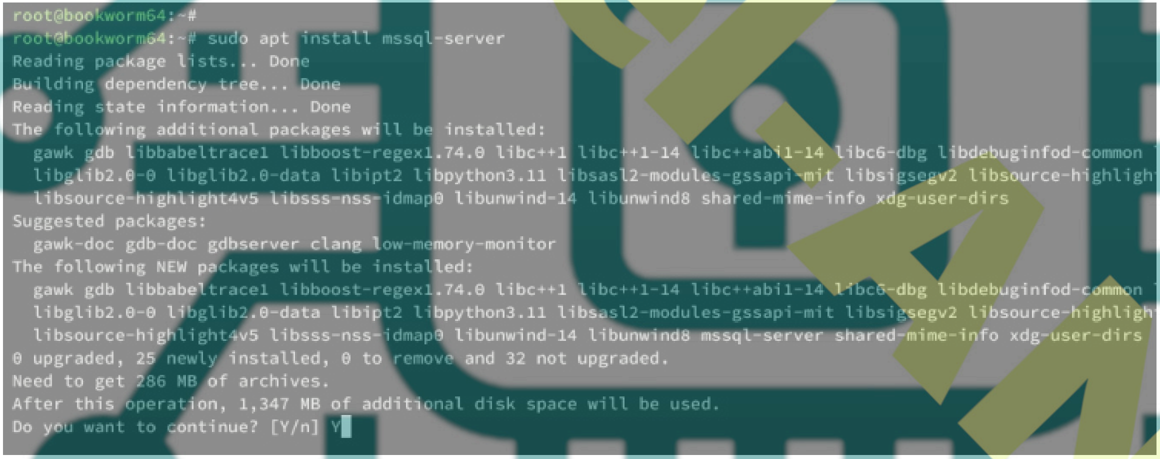
```
root@bookworm64:~#
root@bookworm64:~# sudo apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Get:4 https://packages.microsoft.com/ubuntu/22.04/mssql-server-2022 jammy InRelease [3,603 B]
Get:5 https://packages.microsoft.com/ubuntu/22.04/mssql-server-2022 jammy/main amd64 Packages [1,223 B]
Fetched 4,826 B in 1s (3,336 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Installing Microsoft SQL Server

Now that you've added the MS SQL Server repository to your Debian system, let's install it via the APT package manager. In this case, you will be installing MS SQL Server 2022 to your Debian server.

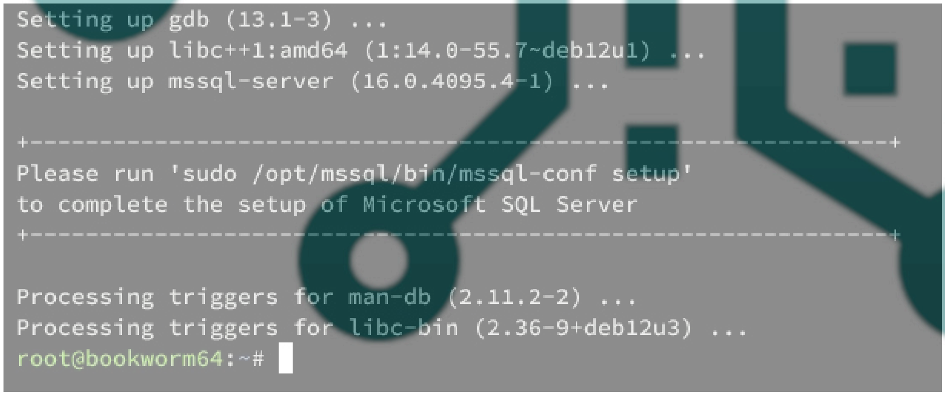
Install the MS SQL Server 2022 using the apt command below. When asked for confirmation, input y and press ENTER to confirm.

```
sudo apt install mssql-server
```



```
root@bookworm64:~#
root@bookworm64:~# sudo apt install mssql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  gawk gdb libbabeltrace1 libboost-regex1.74.0 libc++1 libc++1-14 libc++abi1-14 libc6-dbg libdebuginfod-common
  libglib2.0-0 libglib2.0-data libipt2 libpython3.11 libsasl2-modules-gssapi-mit libsigsegv2 libsource-highligh
  libsource-highlight4v5 libsss-nss-idmap0 libunwind-14 libunwind8 shared-mime-info xdg-user-dirs
Suggested packages:
  gawk-doc gdb-doc gdbserver clang low-memory-monitor
The following NEW packages will be installed:
  gawk gdb libbabeltrace1 libboost-regex1.74.0 libc++1 libc++1-14 libc++abi1-14 libc6-dbg libdebuginfod-common
  libglib2.0-0 libglib2.0-data libipt2 libpython3.11 libsasl2-modules-gssapi-mit libsigsegv2 libsource-highligh
  libsource-highlight4v5 libsss-nss-idmap0 libunwind-14 libunwind8 mssql-server shared-mime-info xdg-user-dirs
0 upgraded, 25 newly installed, 0 to remove and 32 not upgraded.
Need to get 286 MB of archives.
After this operation, 1,347 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

At the end of this process, you will see the instructions to complete the MS SQL Server installation.



```
Setting up gdb (13.1-3) ...
Setting up libc++1:amd64 (1:14.0-55.7~deb12u1) ...
Setting up mssql-server (16.0.4095.4-1) ...

+--------------------------------------------------------------+
Please run 'sudo /opt/mssql/bin/mssql-conf setup'
to complete the setup of Microsoft SQL Server
+--------------------------------------------------------------+

Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+deb12u3) ...
root@bookworm64:~#
```

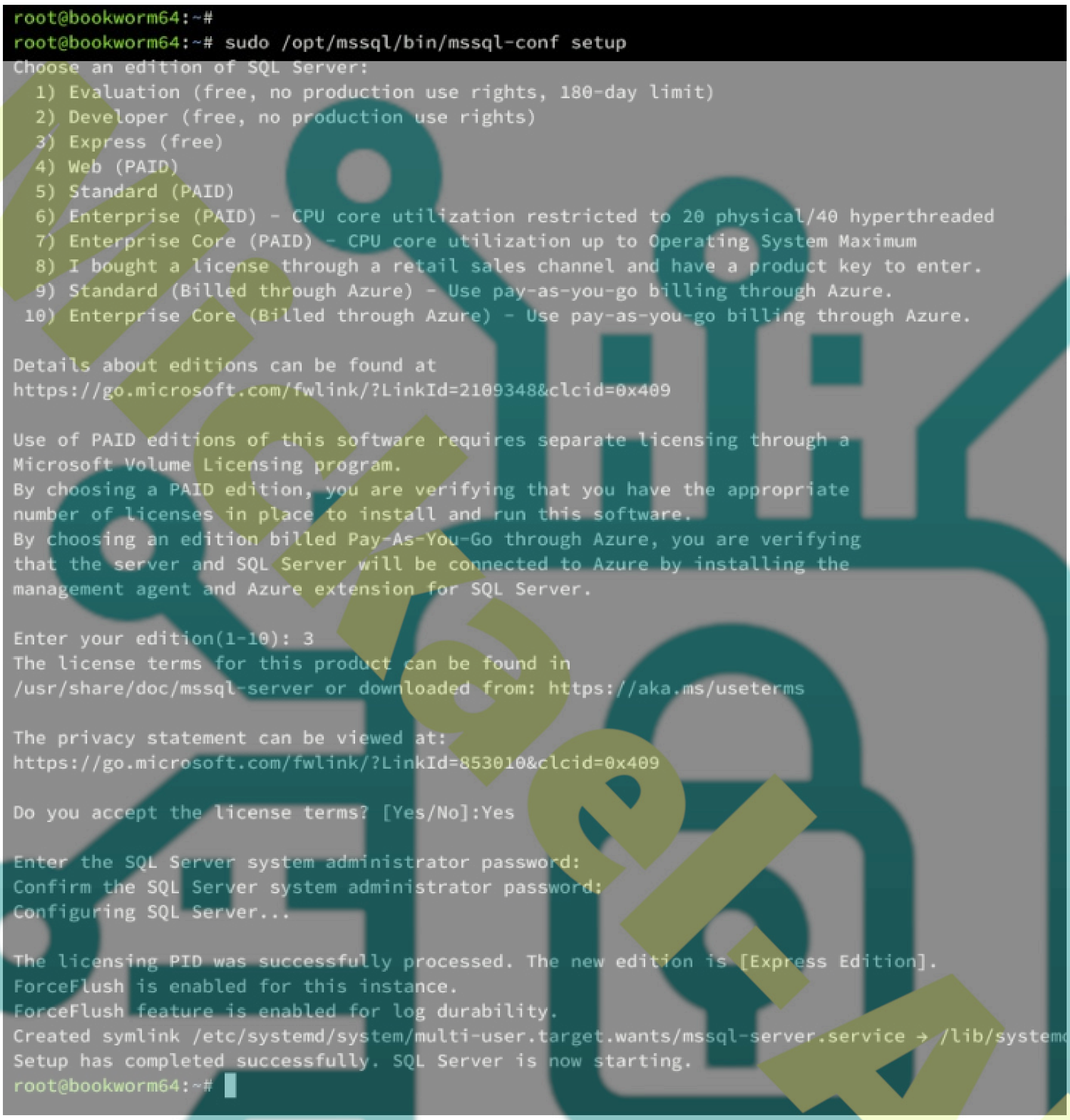
Run the command below to complete the MS SQL Server installation.

```
sudo /opt/mssql/bin/mssql-conf setup
```

Now you will be asked the following:

- Select the MS SQL Server edition? Type 3 to select the SQL Server Express.
- When asked for license terms? Type Yes to confirm.
- Now input the administrator password for your MS SQL Server and repeat.

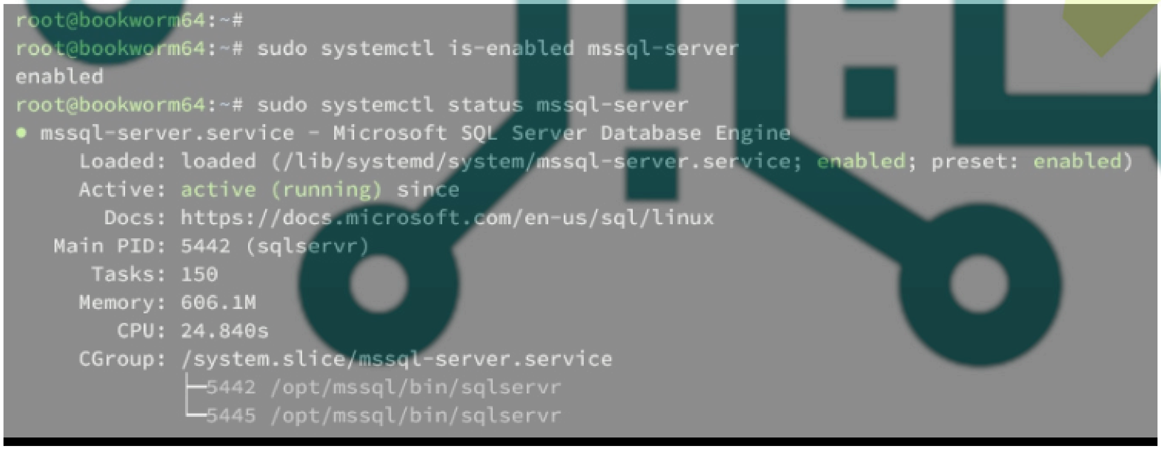
With this, the configuration of MS SQL Server should be completed. It should be running on your Debian server.



```
root@bookworm64:~#
root@bookworm64:~# sudo /opt/mssql/bin/mssql-conf setup
Choose an edition of SQL Server:
  1) Evaluation (free, no production use rights, 180-day limit)
  2) Developer (free, no production use rights)
  3) Express (free)
  4) Web (PAID)
  5) Standard (PAID)
  6) Enterprise (PAID) - CPU core utilization restricted to 20 physical/40 hyperthreaded
  7) Enterprise Core (PAID) - CPU core utilization up to Operating System Maximum
  8) I bought a license through a retail sales channel and have a product key to enter.
  9) Standard (Billed through Azure) - Use pay-as-you-go billing through Azure.
 10) Enterprise Core (Billed through Azure) - Use pay-as-you-go billing through Azure.

Details about editions can be found at
https://go.microsoft.com/fwlink/?LinkId=2109348&clcid=0x409

Use of PAID editions of this software requires separate licensing through a
Microsoft Volume Licensing program.
By choosing a PAID edition, you are verifying that you have the appropriate
number of licenses in place to install and run this software.
By choosing an edition billed Pay-As-You-Go through Azure, you are verifying
that the server and SQL Server will be connected to Azure by installing the
management agent and Azure extension for SQL Server.

Enter your edition(1-10): 3
The license terms for this product can be found in
/usr/share/doc/mssql-server or downloaded from: https://aka.ms/useterms

The privacy statement can be viewed at:
https://go.microsoft.com/fwlink/?LinkId=853010&clcid=0x409

Do you accept the license terms? [Yes/No]:Yes

Enter the SQL Server system administrator password:
Confirm the SQL Server system administrator password:
Configuring SQL Server...

The licensing PID was successfully processed. The new edition is [Express Edition].
ForceFlush is enabled for this instance.
ForceFlush feature is enabled for log durability.
Created symlink /etc/systemd/system/multi-user.target.wants/mssql-server.service → /lib/system
Setup has completed successfully. SQL Server is now starting.
root@bookworm64:~#
```

Verify the MS SQL Server service status using the command below. Ensure the MS SQL Server is running and enabled on your system.

```
sudo systemctl is-enabled mssql-server
sudo systemctl status mssql-server
```



```
root@bookworm64:~#
root@bookworm64:~# sudo systemctl is-enabled mssql-server
enabled
root@bookworm64:~# sudo systemctl status mssql-server
● mssql-server.service - Microsoft SQL Server Database Engine
     Loaded: loaded (/lib/systemd/system/mssql-server.service; enabled; preset: enabled)
     Active: active (running) since
       Docs: https://docs.microsoft.com/en-us/sql/linux
   Main PID: 5442 (sqlservr)
      Tasks: 150
     Memory: 606.1M
        CPU: 24.840s
     CGroup: /system.slice/mssql-server.service
             ├─5442 /opt/mssql/bin/sqlservr
             └─5445 /opt/mssql/bin/sqlservr
```

Setting Up UFW

After the MS SQL Server is running, the next step is to set up UFW and open port 1433/tcp for the MS SQL Server. Be sure to connect

Install UFW (Uncomplicated Firewall) to your Debian machine using the command below. Type y to confirm when asked.

```
sudo apt install ufw
```

Once UFW is installed, run the ufw command below to add the **OpenSSH** profile and port **1433/tcp**. And be sure to change the network subnet with your local network IP addresses.

The OpenSSH profile will allow traffic to SSH port **22**, while port **1433/tcp** is used for MS SQL Server traffic.

```
sudo ufw allow OpenSSH
sudo ufw allow 1433/tcp

or

sudo ufw allow from 192.168.1.0/24 to any port 1433
```

Now start and enable UFW using the command below. When asked, type y to confirm and start UFW.

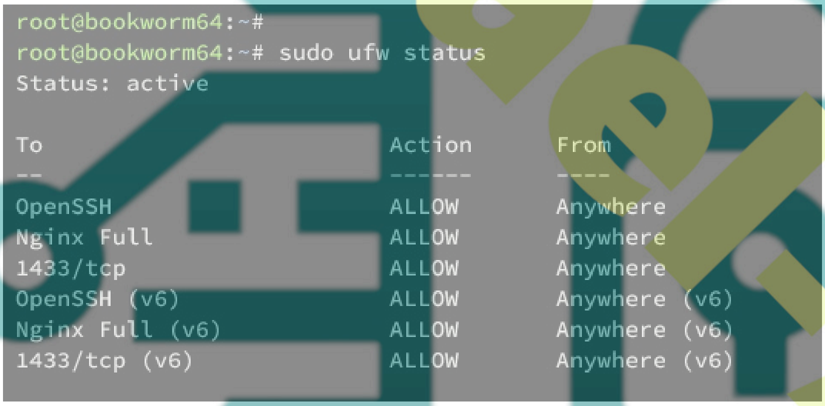
```
sudo ufw enable
```

If UFW is running, you will see the output '**Firewall is active and enabled on system startup**'.

You can now verify the list of open ports and enabled profiles on UFW using the command below.

```
sudo ufw status
```

Ensure the OpenSSH profile and port 1433/tcp are added to UFW.



```
root@bookworm64:~#
root@bookworm64:~# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Nginx Full                 ALLOW       Anywhere
1433/tcp                   ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Nginx Full (v6)            ALLOW       Anywhere (v6)
1433/tcp (v6)              ALLOW       Anywhere (v6)
```

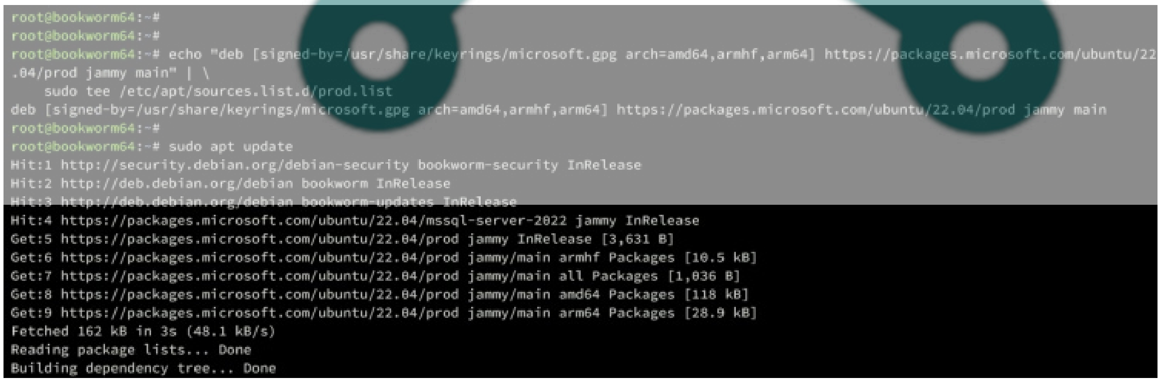
Installing Microsoft SQL Tools for Client

In this section, you will learn how to install MS SQL Tools on your client machine. This covers how to install MS SQL Tools on Linux Debian-based distributions and RedHat-based distributions.

Add the MS SQL Tools repository and refresh your package index in your Debian/Ubuntu client using the command below.

For Debian/Ubuntu client machine:

```
echo "deb [signed-by=/usr/share/keyrings/microsoft.gpg arch=amd64,armhf,arm64] https://packages.microsoft.com/ubuntu/22.04/prod jammy main" | \
sudo tee /etc/apt/sources.list.d/prod.list

sudo apt update
```



```
root@bookworm64:~#
root@bookworm64:~#
root@bookworm64:~# echo "deb [signed-by=/usr/share/keyrings/microsoft.gpg arch=amd64,armhf,arm64] https://packages.microsoft.com/ubuntu/22.04/prod jammy main" | \
    sudo tee /etc/apt/sources.list.d/prod.list
deb [signed-by=/usr/share/keyrings/microsoft.gpg arch=amd64,armhf,arm64] https://packages.microsoft.com/ubuntu/22.04/prod jammy main
root@bookworm64:~#
root@bookworm64:~# sudo apt update
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 https://packages.microsoft.com/ubuntu/22.04/mssql-server-2022 jammy InRelease
Get:5 https://packages.microsoft.com/ubuntu/22.04/prod jammy InRelease [3,631 B]
Get:6 https://packages.microsoft.com/ubuntu/22.04/prod jammy/main armhf Packages [10.5 kB]
Get:7 https://packages.microsoft.com/ubuntu/22.04/prod jammy/main all Packages [1,036 B]
Get:8 https://packages.microsoft.com/ubuntu/22.04/prod jammy/main amd64 Packages [118 kB]
Get:9 https://packages.microsoft.com/ubuntu/22.04/prod jammy/main arm64 Packages [28.9 kB]
Fetched 162 kB in 3s (48.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
```

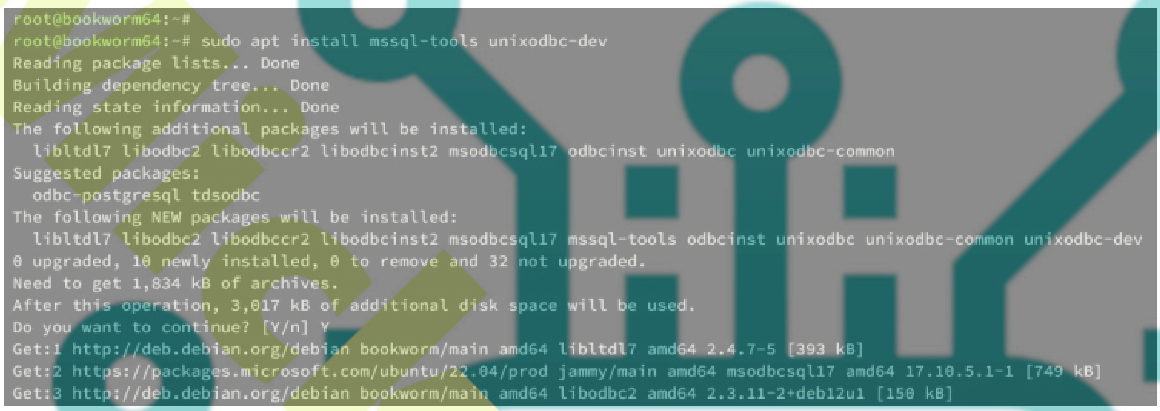
For clients' machine RedHat-based distributions:

```
sudo curl -o /etc/yum.repos.d/mssql-prod.repo https://packages.microsoft.com/config/rhel/9.0/prod.repo
sudo dnf repolist
```

Once the repository repository is added, run the command below to install MS SQL Tools on your client machine.

MS SQL Tools for Debian/Ubuntu distributions:

```
sudo apt install mssql-tools unixodbc-dev
```

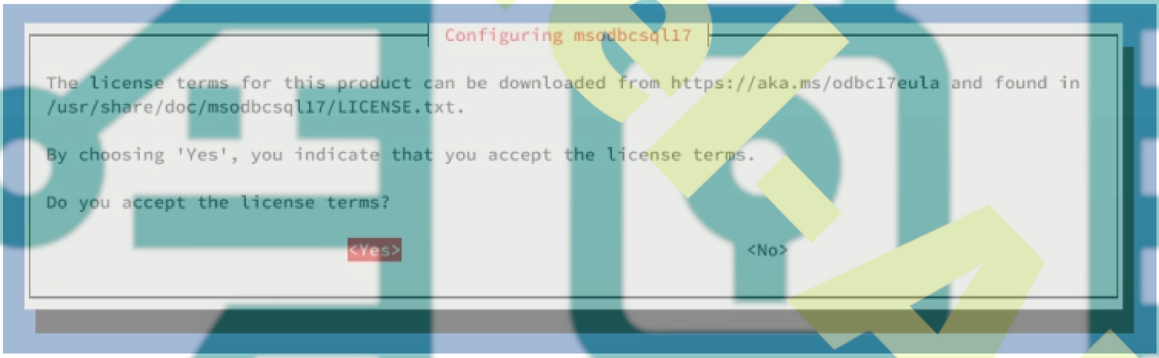


```
root@bookworm64:~#
root@bookworm64:~# sudo apt install mssql-tools unixodbc-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libltdl7 libodbc2 libodbccr2 libodbcinst2 msodbcsql17 odbcinst unixodbc unixodbc-common
Suggested packages:
  odbc-postgresql tdsodbc
The following NEW packages will be installed:
  libltdl7 libodbc2 libodbccr2 libodbcinst2 msodbcsql17 mssql-tools odbcinst unixodbc unixodbc-common unixodbc-dev
0 upgraded, 10 newly installed, 0 to remove and 32 not upgraded.
Need to get 1,834 kB of archives.
After this operation, 3,017 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian bookworm/main amd64 libltdl7 amd64 2.4.7-5 [393 kB]
Get:2 https://packages.microsoft.com/ubuntu/22.04/prod jammy/main amd64 msodbcsql17 amd64 17.10.5.1-1 [749 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 libodbc2 amd64 2.3.11-2+deb12u1 [150 kB]
```

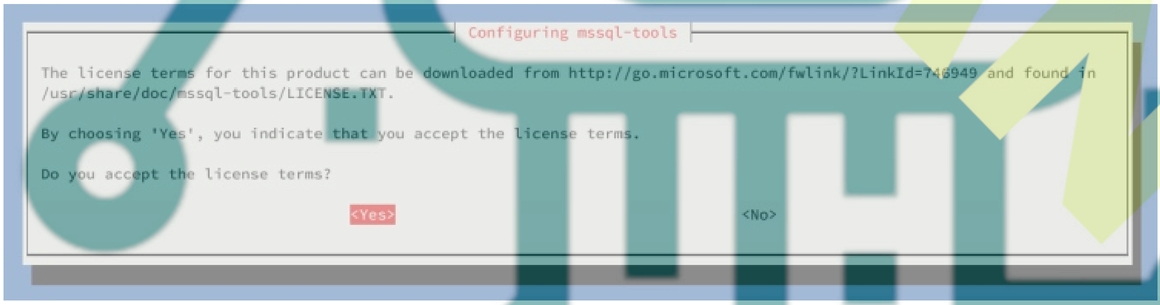
MS SQL Tools for RHEL-based distributions:

```
sudo dnf install mssql-tools unixODBC-devel
```

Input YES when asked for license terms of MS SQL Tools packages.



```
Configuring msodbcsql17

The license terms for this product can be downloaded from https://aka.ms/odbc17eula and found in
/usr/share/doc/msodbcsql17/LICENSE.txt.

By choosing 'Yes', you indicate that you accept the license terms.

Do you accept the license terms?

<Yes>                                              <No>
```



```
Configuring mssql-tools

The license terms for this product can be downloaded from http://go.microsoft.com/fwlink/?LinkId=746949 and found in
/usr/share/doc/mssql-tools/LICENSE.TXT.

By choosing 'Yes', you indicate that you accept the license terms.

Do you accept the license terms?

<Yes>                                              <No>
```

After the installation is complete, the MS SQL Tools should be installed in the */opt/mssql-tools/bin* directory. You can check the */opt/mssql-tools/bin* directory and you will see two binary files *sqlcmd* as the MS SQL client and *bcp* for importing data to the SQL Server.

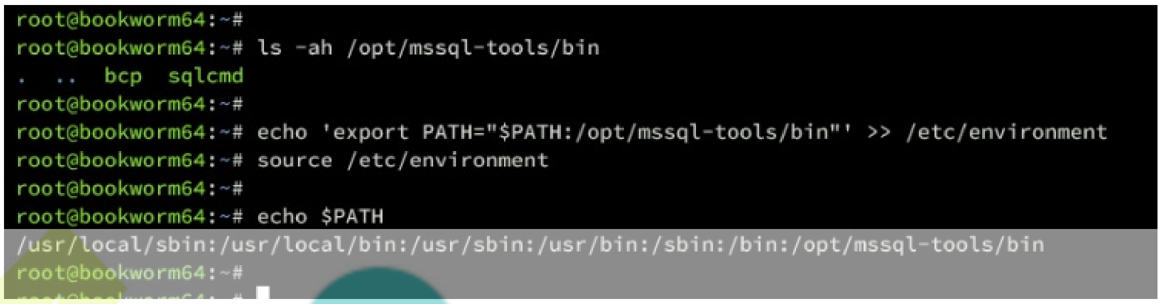
```
ls -ah /opt/mssql-tools/bin
```

Next, add the */opt/mssql-tools/bin* directory to the system **PATH** using the command below.

```
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> /etc/environment
```

Now load the /etc/environment file and verify PATH using the command below. Ensure the */opt/mssql-tools/bin* directory is added to your system **PATH**.

```
source /etc/environment
echo $PATH
```



```
root@bookworm64:~#
root@bookworm64:~# ls -ah /opt/mssql-tools/bin
.  ..  bcp  sqlcmd
root@bookworm64:~#
root@bookworm64:~# echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> /etc/environment
root@bookworm64:~# source /etc/environment
root@bookworm64:~#
root@bookworm64:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/mssql-tools/bin
root@bookworm64:~#
```

Lastly, run the following command to locate both binary files sqlcmd and bcp.

```
which sqlcmd
which bcp
```

You can now execute both '*sqlcmd*' and '*bcp*' commands like this:

```
sqlcmd -?
bcp -?
```

Connecting to MS SQL Server via sqlcmd

Now that you have installed the MS SQL Tools on your client machine, the next step is to connect to your MS SQL Server.

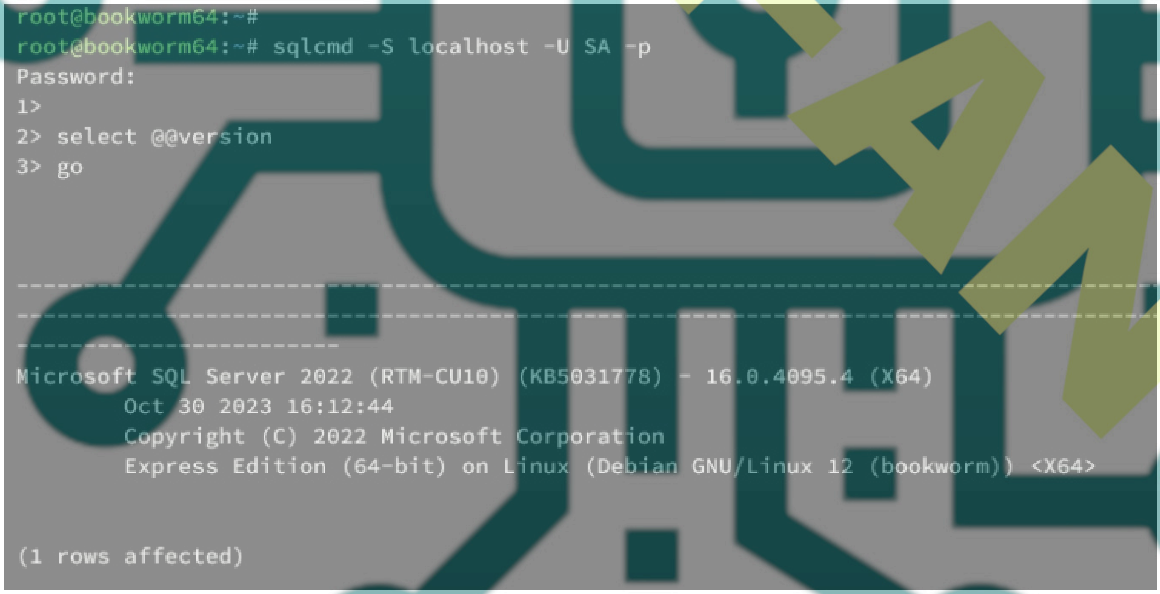
To connect to your MS SQL Server, run the sqlcmd command below. Input your administrator password when asked.

```
sqlcmd -S localhost -U SA -p
```

Once connected, run the following query to verify your MS SQL Server version.

```
select @@version
go
```

You will see that you've installed MS SQL Server 2022.

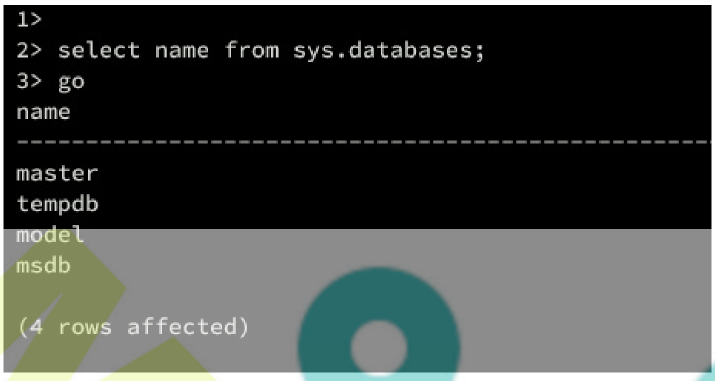


```
root@bookworm64:~#
root@bookworm64:~# sqlcmd -S localhost -U SA -p
Password:
1>
2> select @@version
3> go

----------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
------------------------
Microsoft SQL Server 2022 (RTM-CU10) (KB5031778) - 16.0.4095.4 (X64)
        Oct 30 2023 16:12:44
        Copyright (C) 2022 Microsoft Corporation
        Express Edition (64-bit) on Linux (Debian GNU/Linux 12 (bookworm)) <X64>


(1 rows affected)
```

Now verify the list of databases on your MS SQL Server using the query below.

```
select name from sys.databases;
go
```

You will see default databases master, tempdb, model, and msdb on the MS SQL Server.



```
1>
2> select name from sys.databases;
3> go
name
------------------------------------------------
master
tempdb
model
msdb

(4 rows affected)
```

Creating Database and User in MS SQL Server

Once you've connected to the MS SQL Server, now you can create a new user and database on the MS SQL Server.

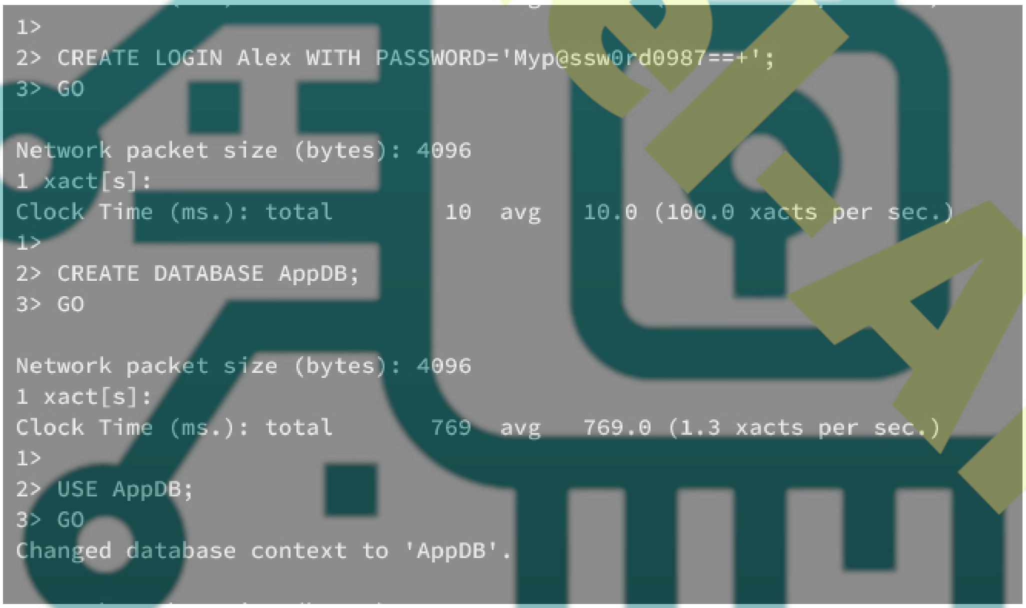
TO create a new login and user on MS SQL Server, execute the following queries. In this example, you will create a new login **Alex** with the password '**Myp@ssw0rd0987==+**'.

```
CREATE LOGIN Alex WITH PASSWORD='Myp@ssw0rd0987==+';
GO
```

After that create a new database '**AppDB**' and switch to it using the following queries.

```
CREATE DATABASE AppDB;
GO
```

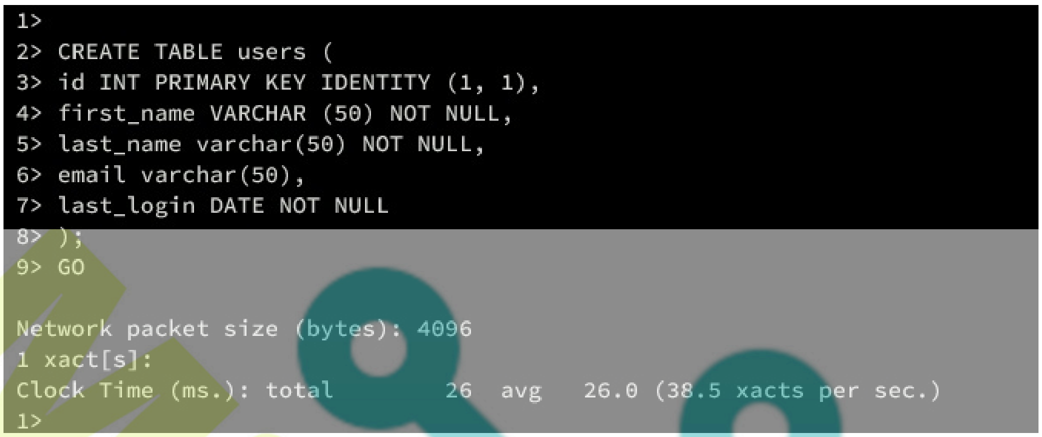
```
USE AppDB;
GO
```



```
1>
2> CREATE LOGIN Alex WITH PASSWORD='Myp@ssw0rd0987==+';
3> GO

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total         10  avg   10.0 (100.0 xacts per sec.)
1>
2> CREATE DATABASE AppDB;
3> GO

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total        769  avg   769.0 (1.3 xacts per sec.)
1>
2> USE AppDB;
3> GO
Changed database context to 'AppDB'.
```

Now run the following queries to create new table **users**.

```
CREATE TABLE users (
id INT PRIMARY KEY IDENTITY (1, 1),
first_name VARCHAR (50) NOT NULL,
last_name varchar(50) NOT NULL,
email varchar(50),
last_login DATE NOT NULL
);
GO
```



```
1>
2> CREATE TABLE users (
3> id INT PRIMARY KEY IDENTITY (1, 1),
4> first_name VARCHAR (50) NOT NULL,
5> last_name varchar(50) NOT NULL,
6> email varchar(50),
7> last_login DATE NOT NULL
8> );
9> GO

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total        26  avg   26.0 (38.5 xacts per sec.)
1>
```

Then create a new user '**Alex**' for login '**Alex**' using the query below.

```
CREATE USER Alex FOR LOGIN Alex;
GO
```

Once the user is created, run the query below to grant privileges of table **users** to user **Alex**.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON users TO Alex;
GO
```

Type quit to exit.

Insert and Retrieve Data in Microsoft SQL Server

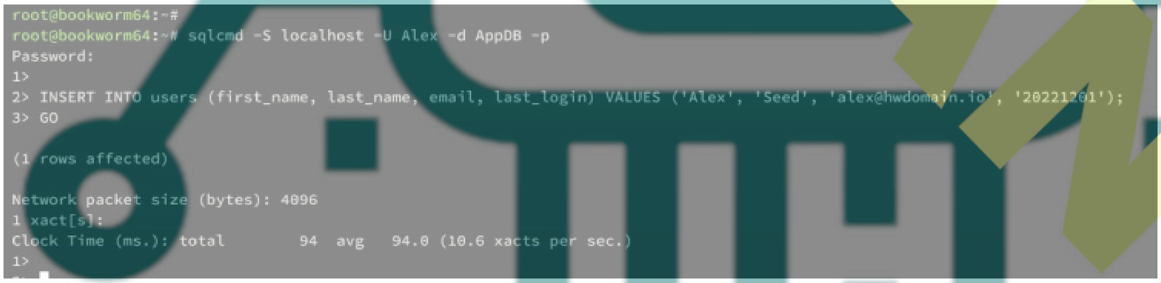
With the new database and user created, the next step is to verify your database by connecting to the MS SQL Server using your user, specifically to your database.

Now run the '*sqlcmd*' command below to connect to your database using your login user. Input your password when prompted.

```
sqlcmd -S localhost -U Alex -d AppDB -p
```

Once logged in, run the following query to insert new data to the table 'users'.

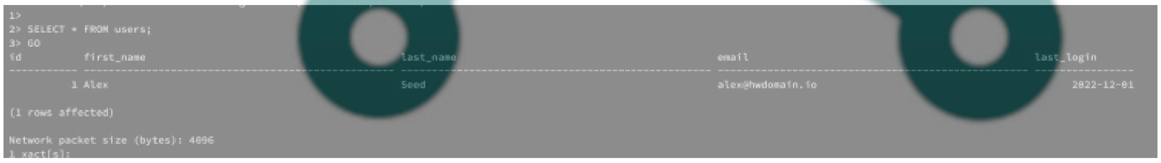
```
INSERT INTO users (first_name, last_name, email, last_login) VALUES ('Alex', 'Seed', 'alex@hwdomain.io', '20221201');
GO
```



```
root@bookworm64:~#
root@bookworm64:~# sqlcmd -S localhost -U Alex -d AppDB -p
Password:
1>
2> INSERT INTO users (first_name, last_name, email, last_login) VALUES ('Alex', 'Seed', 'alex@hwdomain.io', '20221201');
3> GO

(1 rows affected)

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total        94  avg   94.0 (10.6 xacts per sec.)
1>
```

Now you can run the SELECT query below to retrieve your data in MS SQL Server.

```
SELECT * FROM users;
GO
```



```
1>
2> SELECT * FROM users;
3> GO
id          first_name                                         last_name                                          email                                              last_login
----------- -------------------------------------------------- -------------------------------------------------- -------------------------------------------------- ----------
          1 Alex                                               Seed                                               alex@hwdomain.io                                   2022-12-01

(1 rows affected)

Network packet size (bytes): 4096
1 xact[s]:
```

You can now type quit to exit.

Conclusion

Congratulations! You have now completed the installation of Microsoft SQL Server 2022 on the Debian 12 server. You've also learned how to install MS SQL Tools on Debian-based and RedHat-based distributions and learned how to connect to MS SQL Server via the 'sqlcmd' command.

Furthermore, you have also learned the basic usage of MS SQL queries for creating databases and users, creating tables, and inserting and retrieving data in the MS SQL Server.