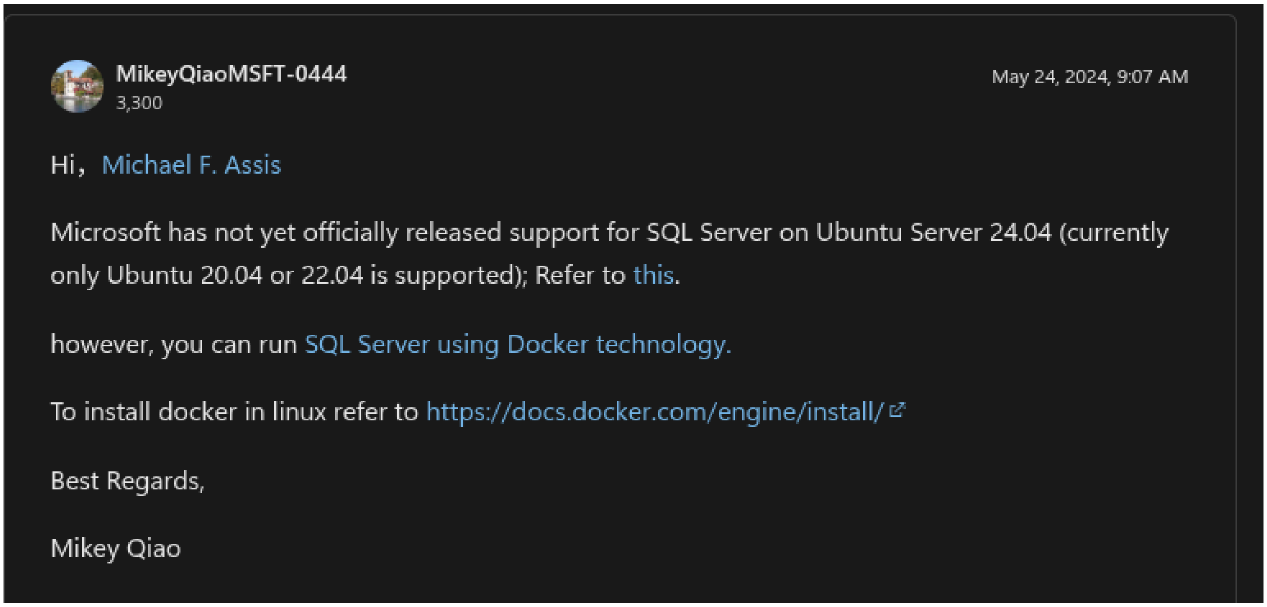
MIGRATING MS SQL DATABASE ACROSS EC2 INSTANCE

Some considerations before installing MS SQL Server in EC2 instance:

Ubuntu Version	SQL Server Support	Common Issue	Solution
20.04, 22.04	Supported	Missing libssl1.1	Install libssl1.1 (64-bit)
24.04+	Not Supported	Service fails to start	Downgrade to 22.04 or use Docker



MikeyQiaoMSFT-0444
3,300

May 24, 2024, 9:07 AM

Hi，Michael F. Assis

Microsoft has not yet officially released support for SQL Server on Ubuntu Server 24.04 (currently only Ubuntu 20.04 or 22.04 is supported); Refer to this.

however, you can run SQL Server using Docker technology.

To install docker in linux refer to https://docs.docker.com/engine/install/

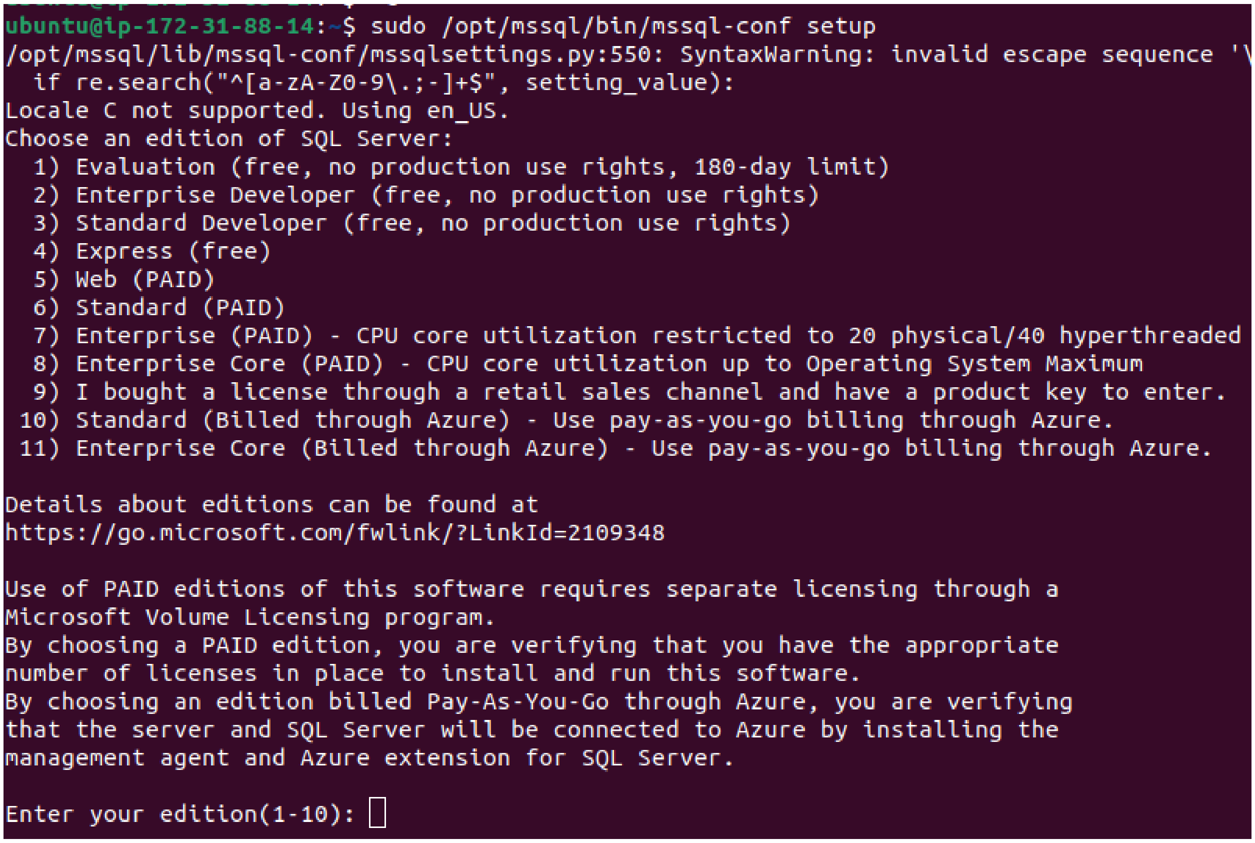
Best Regards,

Mikey Qiao

Install MS SQL in the *newer* Ubuntu Instance:

1. **FIrst install the MS SQL package from the official documentation(follow doc):** https://learn.microsoft.com/en-us/sql/linux/quickstart-install-connect-ubuntu?view=sql-server-ver17&tabs=ubuntu2004#install-sql-server

2. **Important: Install the SQL Command Line Tool:** https://learn.microsoft.com/en-us/sql/linux/sql-server-linux-setup-tools?view=sql-server-ver17&tabs=ubuntu-install



```
ubuntu@ip-172-31-88-14:~$ sudo /opt/mssql/bin/mssql-conf setup
/opt/mssql/lib/mssql-conf/mssqlsettings.py:550: SyntaxWarning: invalid escape sequence '\
  if re.search("^[a-zA-Z0-9\.;-]+$", setting_value):
Locale C not supported. Using en_US.
Choose an edition of SQL Server:
  1) Evaluation (free, no production use rights, 180-day limit)
  2) Enterprise Developer (free, no production use rights)
  3) Standard Developer (free, no production use rights)
  4) Express (free)
  5) Web (PAID)
  6) Standard (PAID)
  7) Enterprise (PAID) - CPU core utilization restricted to 20 physical/40 hyperthreaded
  8) Enterprise Core (PAID) - CPU core utilization up to Operating System Maximum
  9) I bought a license through a retail sales channel and have a product key to enter.
 10) Standard (Billed through Azure) - Use pay-as-you-go billing through Azure.
 11) 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

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): 
```

Choose 3 for MS SQL Standard Developer Edition

Important: The System Admin(SA) password should comply with the MS terms, so to avoid problems, password should be simple and at least 8 letter long
I.e Genese@1234

We have the following options for migrating the database:

Method	Preserves Logins/Jobs	Downtime	Complexity	Best For
Backup/Restore	Yes	Medium	Low	Most scenarios
BACPAC	No	Medium	Low	Schema + data only
Azure Data Studio (Cross - platform)	Yes	Depends on the backup configuration	Low (GUI)	Depends upon the configuration
Detach/Attach	Yes	High	Medium	Large DB, same OS/FS
EBS Snapshot	Yes	Low	Medium	Fast, same region
AWS DMS	No (by default)	Low	High	Ongoing sync, low downtime
Application Migration (MGN)	Yes	Low	High	Whole server migration

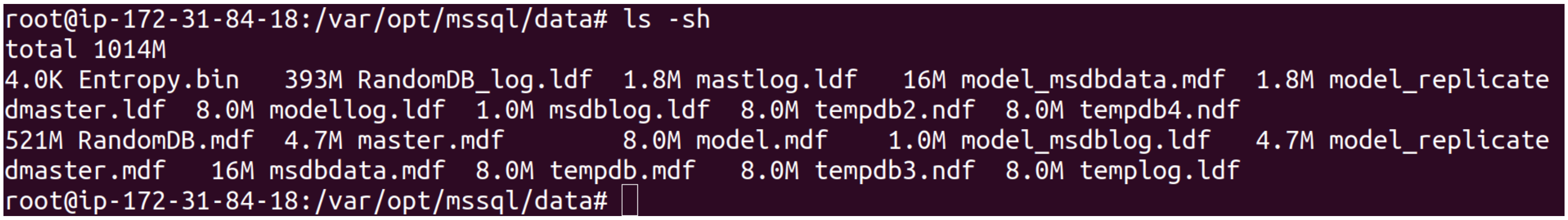
Considering feasibility of operations:

We are currently implementing a full restore option for ms sql database that is **MS SQL native** and not cloud native. We can use the **Azure Data Studio** too, but have to migrate the .bak file using the terminal, so it's the same thing as a manual SCP through the linux cmd.

Method 1: Backup and Restore (Recommended Method):

Note: This technique does not migrate transactional logs.

Slight overview of the **File Structure:**



```
root@ip-172-31-84-18:/var/opt/mssql/data# ls -sh
total 1014M
4.0K Entropy.bin   393M RandomDB_log.ldf  1.8M mastlog.ldf   16M model_msdbdata.mdf  1.8M model_replicate
dmaster.ldf  8.0M modellog.ldf  1.0M msdblog.ldf  8.0M tempdb2.ndf  8.0M tempdb4.ndf
521M RandomDB.mdf  4.7M master.mdf        8.0M model.mdf    1.0M model_msdblog.ldf   4.7M model_replicate
dmaster.mdf   16M msdbdata.mdf  8.0M tempdb.mdf   8.0M tempdb3.ndf  8.0M templog.ldf
root@ip-172-31-84-18:/var/opt/mssql/data# 
```

MS SQL Database is stored in the system with following file types:

- .**LDF**: Log File – stores transaction logs for recovery.
- **.MDF**: Primary Data File – contains system tables and startup data.
- **.NDF:** Secondary Data File – optional, used to store additional user data.

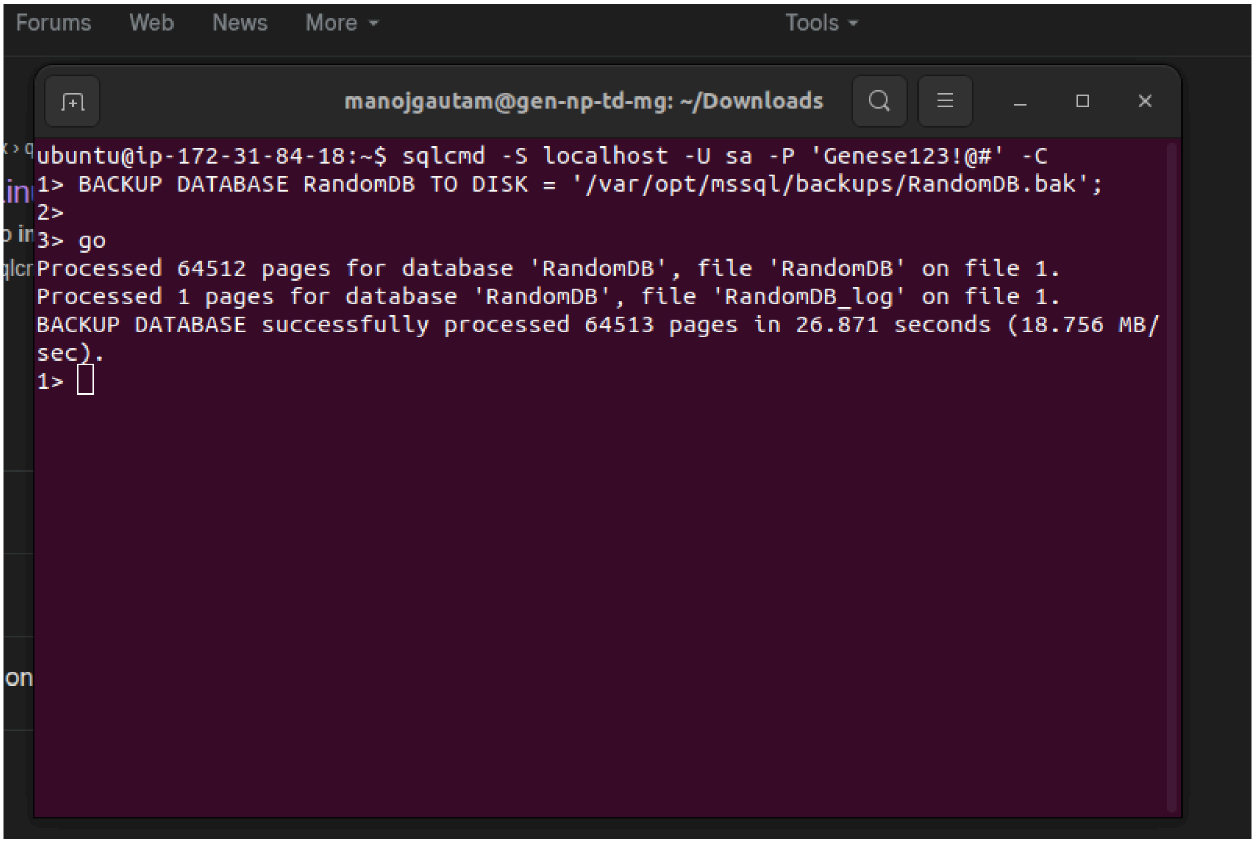
Backup options for the **Backup and Restore**:

	Method	**Description**
1.	Full Backup	A complete copy of the entire database (data + part of the log).
2.	Differential Backup	Captures only change since the last full backup.
3.	Transactional Log Backup	Backs up the transaction log, allowing point-in-time recovery.
4.	Copy-Only Backup	A special full or log backup that doesn't affect the backup chain.

Procedure 1: Full Backup

Backup the database in the desired location by running the command:

```
BACKUP DATABASE RandomDB TO DISK = '/var/opt/mssql/backups/RandomDB.bak';
```



```
ubuntu@ip-172-31-84-18:~$ sqlcmd -S localhost -U sa -P 'Genese123!@#' -C
1> BACKUP DATABASE RandomDB TO DISK = '/var/opt/mssql/backups/RandomDB.bak';
2>
3> go
Processed 64512 pages for database 'RandomDB', file 'RandomDB' on file 1.
Processed 1 pages for database 'RandomDB', file 'RandomDB_log' on file 1.
BACKUP DATABASE successfully processed 64513 pages in 26.871 seconds (18.756 MB/
sec).
1>
```

Send the backup file from current instance to *newer instance* using command:

```
rsync -avz -e "ssh -i /home/ubuntu/testkeymanoj.pem"
/var/opt/mssql/backups/RandomDB.bak ubuntu@34.227.148.41:~
```

SSH to *newer instance* and Move the .bak folder to the appropriate location:

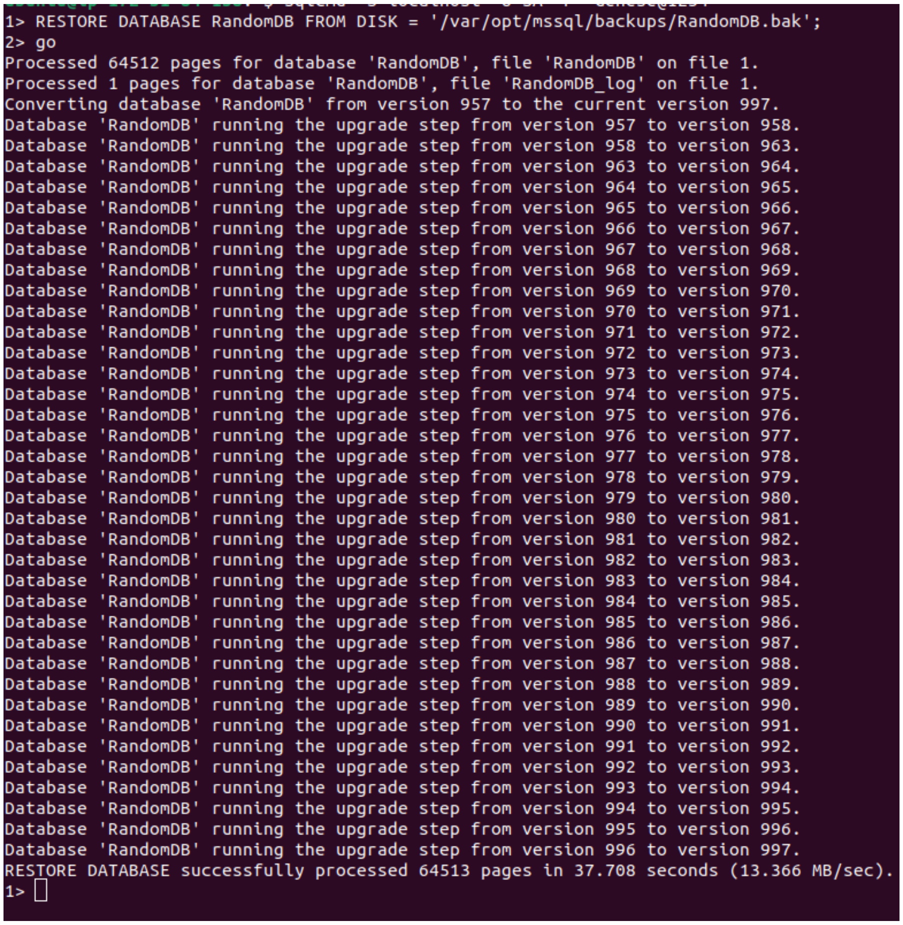
sudo mv /home/ubuntu/RandomDB.bak /var/opt/mssql/backups/

But first, *ensure ownership* of the file and the directory for the current user by:

```
sudo chmod 644 /var/opt/mssql/backups/RandomDB.bak
sudo chown mssql /var/opt/mssql/backups
sudo chmod 755 /var/opt/mssql/backups
```

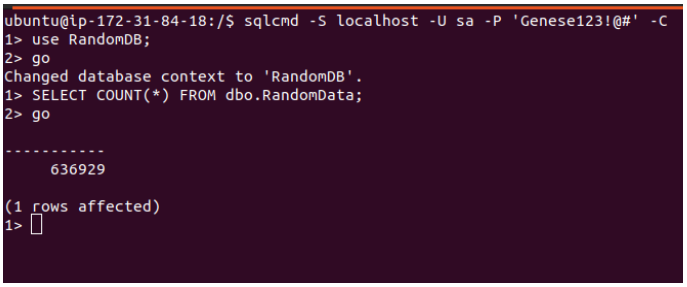
Now Run the restore command(on the *newer instance*):

```
RESTORE DATABASE RandomDB FROM DISK = '/var/opt/mssql/backups/RandomDB.bak';
GO
```



```
1> RESTORE DATABASE RandomDB FROM DISK = '/var/opt/mssql/backups/RandomDB.bak';
2> go
Processed 64512 pages for database 'RandomDB', file 'RandomDB' on file 1.
Processed 1 pages for database 'RandomDB', file 'RandomDB_log' on file 1.
Converting database 'RandomDB' from version 957 to the current version 997.
Database 'RandomDB' running the upgrade step from version 957 to version 958.
Database 'RandomDB' running the upgrade step from version 958 to version 963.
Database 'RandomDB' running the upgrade step from version 963 to version 964.
Database 'RandomDB' running the upgrade step from version 964 to version 965.
Database 'RandomDB' running the upgrade step from version 965 to version 966.
Database 'RandomDB' running the upgrade step from version 966 to version 967.
Database 'RandomDB' running the upgrade step from version 967 to version 968.
Database 'RandomDB' running the upgrade step from version 968 to version 969.
Database 'RandomDB' running the upgrade step from version 969 to version 970.
Database 'RandomDB' running the upgrade step from version 970 to version 971.
Database 'RandomDB' running the upgrade step from version 971 to version 972.
Database 'RandomDB' running the upgrade step from version 972 to version 973.
Database 'RandomDB' running the upgrade step from version 973 to version 974.
Database 'RandomDB' running the upgrade step from version 974 to version 975.
Database 'RandomDB' running the upgrade step from version 975 to version 976.
Database 'RandomDB' running the upgrade step from version 976 to version 977.
Database 'RandomDB' running the upgrade step from version 977 to version 978.
Database 'RandomDB' running the upgrade step from version 978 to version 979.
Database 'RandomDB' running the upgrade step from version 979 to version 980.
Database 'RandomDB' running the upgrade step from version 980 to version 981.
Database 'RandomDB' running the upgrade step from version 981 to version 982.
Database 'RandomDB' running the upgrade step from version 982 to version 983.
Database 'RandomDB' running the upgrade step from version 983 to version 984.
Database 'RandomDB' running the upgrade step from version 984 to version 985.
Database 'RandomDB' running the upgrade step from version 985 to version 986.
Database 'RandomDB' running the upgrade step from version 986 to version 987.
Database 'RandomDB' running the upgrade step from version 987 to version 988.
Database 'RandomDB' running the upgrade step from version 988 to version 989.
Database 'RandomDB' running the upgrade step from version 989 to version 990.
Database 'RandomDB' running the upgrade step from version 990 to version 991.
Database 'RandomDB' running the upgrade step from version 991 to version 992.
Database 'RandomDB' running the upgrade step from version 992 to version 993.
Database 'RandomDB' running the upgrade step from version 993 to version 994.
Database 'RandomDB' running the upgrade step from version 994 to version 995.
Database 'RandomDB' running the upgrade step from version 995 to version 996.
Database 'RandomDB' running the upgrade step from version 996 to version 997.
RESTORE DATABASE successfully processed 64513 pages in 37.708 seconds (13.366 MB/sec).
1>
```

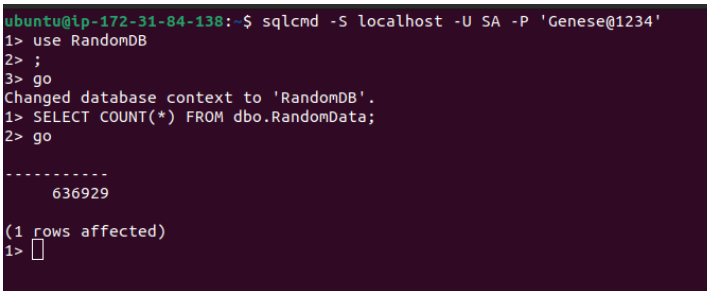
Check the integrity of the database:



```
ubuntu@ip-172-31-84-18:/$ sqlcmd -S localhost -U sa -P 'Genese123!@#' -C
1> use RandomDB;
2> go
Changed database context to 'RandomDB'.
1> SELECT COUNT(*) FROM dbo.RandomData;
2> go

-----------
     636929

(1 rows affected)
1> 
```



```
ubuntu@ip-172-31-84-138:~$ sqlcmd -S localhost -U SA -P 'Genese@1234'
1> use RandomDB
2> ;
3> go
Changed database context to 'RandomDB'.
1> SELECT COUNT(*) FROM dbo.RandomData;
2> go

-----------
     636929

(1 rows affected)
1> 
```

Some Important Database terminologies:

1. **back up (verb):**
   To create a copy of database or log data.
2. **backup (noun):**
   A saved copy of data used for recovery or relocation.
3. **backup device:**
   Disk, tape, or Azure Blob where backups are stored and restored from.
4. **backup media:**
   Tape or disk files holding one or more backups.
5. **data backup:**
   Backup of all or part of a database or filegroups.
6. **database backup:**
   A full or differential copy of a database.
7. **differential backup:**
   Stores changes since the last full backup.
8. **full backup:**
   Backs up all data and enough log to allow recovery.
9. **log backup:**
   Captures unbacked transaction log records.
10. **recover:**
    Bring a database to a consistent state.
11. **recovery:**
    Phase that makes the database transaction-consistent.
12. **recovery model:**
    Database setting that defines log handling and backup needs.
13. **Restore:** Process of applying backups to bring a database to a usable state