

Made by

Nevzat Tokmak
Nicholas Johansson
Rune Andreasen
 Serverteknologi Database

Relation Database

A relational database is a set of formally described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables. The standard user and application programming interface (API) of a relational database is the Structured Query Language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports.

Reference: https://en.wikipedia.org/wiki/Microsoft_SQL_Server

MS SQL Architecture

External Protocols Shared memory - Named pipes - TCP/IP - Virtual Interface Adapter (VIA)

Database Engine Database Tables/Indexes - Type System Transactions - Events/Exception Triggers - TSQL - Stored Procedure - SQLCLR

Storage Engine Transaction Services & File Manager (Utilities) – Buffer Manager & Lock Manager (Access Methods)

MS SQL Architecture (cont)

Query Parser - Optimizer - SQL

Processor Manager - Database Manager - Query Executer

SQL OS API Lock Manager – Synchronization Services – Thread Scheduler – Buffer Pool – Memory Manger – I/O Manager

Reference: <http://udayarumilli.com/sql-server-architecture-qa-3/>

Database Design and Resource use

Database planning Planning the database is very important in terms of creating a resourceful and working database. Even though it seems obvious – It can be one of the most demanding tasks, when creating a relational database.

Documentation The importance of documentation of a database, will make it easier for maintaining and altering changes further ahead in the database.

Database Design and Resource use (cont)

Naming conventions Even if you consider naming your tables, columns etc. to be easy – It is very important to use clear and understandable name giving for your database – So it can be understood by everyone who need to work on the database.

Normalize It very important to know the rules of normalization in terms of relational database structure. If the principles of normalization are done right. Your database will be efficient and logical to everyone. It is very important that your normalization is done right up to the Third Normalization.

Data types Giving your design the right data types is crucial for a working database. Therefore is very important to give your columns in your database the most appropriate data type for each and give you better understanding of the data.



By **Nicholade**
cheatography.com/nicholade/

Published 12th February, 2021.
 Last updated 22nd June, 2018.
 Page 1 of 3.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Database Design and Resource use (cont)

Primary keys It is very important in terms of relational database, that your primary and natural key is set correctly. Be sure that you define a primary key and natural key for your tables – It can be very important when you start to break up your tables when you start to normalize.

Data Integrity Data integrity is very important when you work with RDBMS and it is one of the core functions to uphold the data integrity in your database.

Indexing A proper indexing is a key to a successful database. It can be very complexed to index your database and deserves quite focus when creating your database.

Testing Testing your relational database is very important and can be crucial to avoid system failures. Quality assurance is there for key in database creating.

Reference: <https://www.quora.com/What-are-good-resources-to-learn-Database-design-best-practices>

Install SQL Server

Hardware Minimum Requirements

SQL Server requires a minimum of 6 GB of available hard-disk space.

SQL Server requires Super-VGA (800x600) or higher resolution monitor.

Memory: Express Edition 512 MB, All other Editions 1 GB

Processor Speed: x64 Processor: 1.4 GHz

Processor Type: x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support

Install SQL Server

```
Run setup.exe from your media ->
Installation -> Choose "New SQL
Server stand-alone installation
or add features to an existing
installation" -> Next -> Check I
agree to the terms -> Next
```

Advance to Install Setup Files and check for updates, if not move on and check your installation is correct. Click Next.

```
On Installation type choose new
or add features to existing and
click Next -> Advance -> Name
your Instance ID -> Create
Service Account -> Advance to
Install page -> Reboot when
Required to -> Your installation
is now complete
```

<https://docs.microsoft.com/en-us/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server?view=sql-server-2017>

Configure Windows Firewall

Open TCP port 1433

```
Run netsh.exe with admin permissions
firewall set portopening
protocol = TCP port = 1433 name
= SQLPort mode = ENABLE scope =
SUBNET profile = CURRENT
```

Tabeller

MASTER The master database records all the system-level information for a SQL Server system.

MODEL The model database is used as the template for all databases created on an instance of SQL Server.

MSDB The msdb database is used by SQL Server Agent for scheduling alerts and jobs and by other features such as SQL Server Management Studio, Service Broker and Database Mail

TEMPDB The tempdb system database is a global resource that is available to all users connected to the instance of SQL Server or connected to SQL Database. Tempdb is used to hold: *Temporary user objects that are explicitly created. Internal objects that are created by the database engine. tempdb is re-created every time SQL Server is started*



By **Nicholade**
cheatography.com/nicholade/

Published 12th February, 2021.

Last updated 22nd June, 2018.

Page 2 of 3.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Add users (RDMS)

Powershell

```
$newUser = New-RDMUser -Name "-
LoginName" -CreateSQLServerLogin
-IntegratedSecurity
```

GUI

SSMS > "SQL-Server" > Right click on Security > New - Login

Creating an user in Query

```
CREATE LOGIN "USERNAME"
WITH PASSWORD = 'Pa$$w0rd'
```

The Password need to follow the Windows policy requirement

```
GO
CREATE USER "USERNAME" FOR LOGIN
"USERNAME"
GO
```

User Permissions (RDMS)

Powershell

```
Set-RDMUserProperty -User
$newUser -Property "Add" -Value
$TRUE
Set-RDMUserProperty -User
$newUser -Property "Edit" -Value
$TRUE
Set-RDMUserProperty -User
$newUser -Property "Delete" -
Value $TRUE
```

GUI

SSMS > "SQL-Server" > Databases > "Database" > Security > Users > Right Click on an user > Permissions > Owned Schemas

Alter User with Query

```
EXEC sp_addrolemember ' "SCHEMA-
S" ', ' "USERNAME" '
```

Backup of Database

Create a Full Backup

Go to your Database in Object Explorer
Right click NORTHWIND -> Tasks -> Back UP -> OK

It creates our backup and sets it to default location

Backup of Database (cont)

Create a Full Backup to non-default location

Go to your Database in Object Explorer
Right click NORTHWIND -> Tasks -> Back UP

From here go to Destination -> General -> Select Disk from Drop-down menu

Click Remove until all existing backup files have been removed.

Click Add and enter your desired Path.
It creates our backup and sets it to your changed path

Recovery of Database

Recovery of Full Database Backup

Right click NORTHWIND -> Restore Database... -> From Device -> Add -> Browse your file -> OK -> Select point to restore

Optimize Database Operations

In SSMS open SQL Activity Monitor by pressing

CTRL + ALT + A

View Log Files gathered from SQL

```
%ProgramFiles%\Microsoft SQL Server\13-
0\Setup Bootstrap\LOG\
```

Import & Export Database MS Tool

Import and Export Wizard

Run the SQL server 2016 Import and Export Data Wizard

Click Next -> Pick Excel as Data source -> Choose Destination SQL server and modify the settings -> Choose Copy Data -> Next -> Pick your Table to copy -> OK -> Check Run Immediately -> Finish

Setup Replication

1. Configure Distributor and publisher

SSMS > "SQL-Server" > Right click Replication > Configure Distribution > Act as own Distributor > When asked for publisher add second server

2. Creating Snapshot Publication

SSMS > "Publisher Server" > Replication > Right click Local Publications > New publications > Use following server > Add > Select Distributor > Select Database > Snapshot Publication > Choose Tables > Create Snapshot Immediately

3. Create Subscription

SSMS > "Publisher server" > Replication > Local Publications > Right click the new replication > New Subscription > Add Subscription > Run all Agents at Distribution > Add SQL Server Subscription > Choose a Path > Add New Database > Agent Schedule > Run Continuously > Initialize > Immediately

Snapshot Agent

The Snapshot agent stores an executable file which stores schema and data of published tables and db objects. It's updated when synchronization jobs are recorded.

Distribution Agent

Works with Snapshot to distribute to subscribers. Moves transactions held at the distribution db out to the subscribers.

Merge Agent

It applies the initial snapshot to the Subscriber and moves incremental data changes that occur.

Queue Reader Agent

It runs at the Distributor and moves changes made at the Subscriber back to the Publisher.

<https://www.codeproject.com/Articles/71-5550/SQL-Server-Replication-Step-by-Step>

