

## Databashantering YH00541, Vecka 47

### Introduktion, Praktiska frågor

Information på <https://studentportal.nackademin.se/course/view.php?id=4158> uppdateras efter hand.

#### Obs!

På grund av vädret måndag 21 november har vi flyttat måndagens lektioner till tisdag, varpå tisdagens lektioner blir på onsdag. Tisdag 22 november, onsdag 23 november och torsdag 24 november blir det undervisning på distans.

| Vecka | Dag                                 | Område                                    |
|-------|-------------------------------------|---|
| 46    | Mån 14 nov                          | Introduktion, Använda SQL                 |
| 46    | Tis 15 nov                          | Fortsättning SQL                          |
| 46    | Tors 17 nov                         | Modellera data, Skapa databaser           |
| 47    | <del>Mån 21 nov</del><br>Tis 22 nov | Installera SQL Server (eller motsvarande) |
| 47    | <del>Tis 22 nov</del><br>Ons 23 nov | Hantera säkerhet, Optimera databaser      |
| 47    | Tors 24 nov                         | Olika typer av databaser                  |
| 48    | Mån 28 nov                          | Koppla applikation till databas           |
| 48    | Tis 29 nov                          | Lagra data, Olika driftmiljöer            |
| 48    | Fre 2 dec                           | Inlämningsuppgift, Summering              |

Kl 09.00 – 16.00, med undantag för torsdag 24 nov endast kl 09.00 – 13.00.

Lunch 11.30 – 12.30 (istället för 12.00 – 13.00).

### ***Kort repetition av förra veckan***

Tabeller, kolumner, datatyper. Default-värden. Kontroll av giltiga värden med “check”, “not null”, “unique”.

Primära och främmande nycklar. Kopplingar mellan två eller flera tabeller.

Sql-frågor för att definiera schema, manipulera data, göra urval. Transaktioner.

Urvalsvillkor, sortera och gruppera data. Inbyggda funktioner för att bearbeta siffror, text mm.

Modeller med ER-diagram. Översätta logisk modell till fysisk modell.

Normalisering av databas.

En liten återblick på kunskaper och kompetenser enligt kursplan. Vilka delar har vi klarat av?

## ~~Måndag 21 nov~~ **Tisdag 22 nov – Installera SQL Server**

Mer exakt Microsoft SQL Server 2019 (15.0).

Transact-SQL (T-SQL), Microsofts “dialekt” av SQL.

<https://learn.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15>

Installation av SQL Server 2019, Developer Edition.

- Windows: <https://learn.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server?view=sql-server-ver15>
- Linux: <https://learn.microsoft.com/en-us/sql/linux/sql-server-linux-overview?view=sql-server-ver15>
- MacOS mha Docker: <https://learn.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-ver15>  
Vilket förutsätter att Docker for Mac är installerat, se <https://docs.docker.com/desktop/install/mac-install/>  
Användbara tips på bland annat <https://www.twilio.com/blog/using-sql-server-on-macos-with-docker>

Klientprogram:

Förslagsvis DBeaver eller Azure Data Studio.

<https://dbeaver.io/download/>

<https://learn.microsoft.com/en-us/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver15>

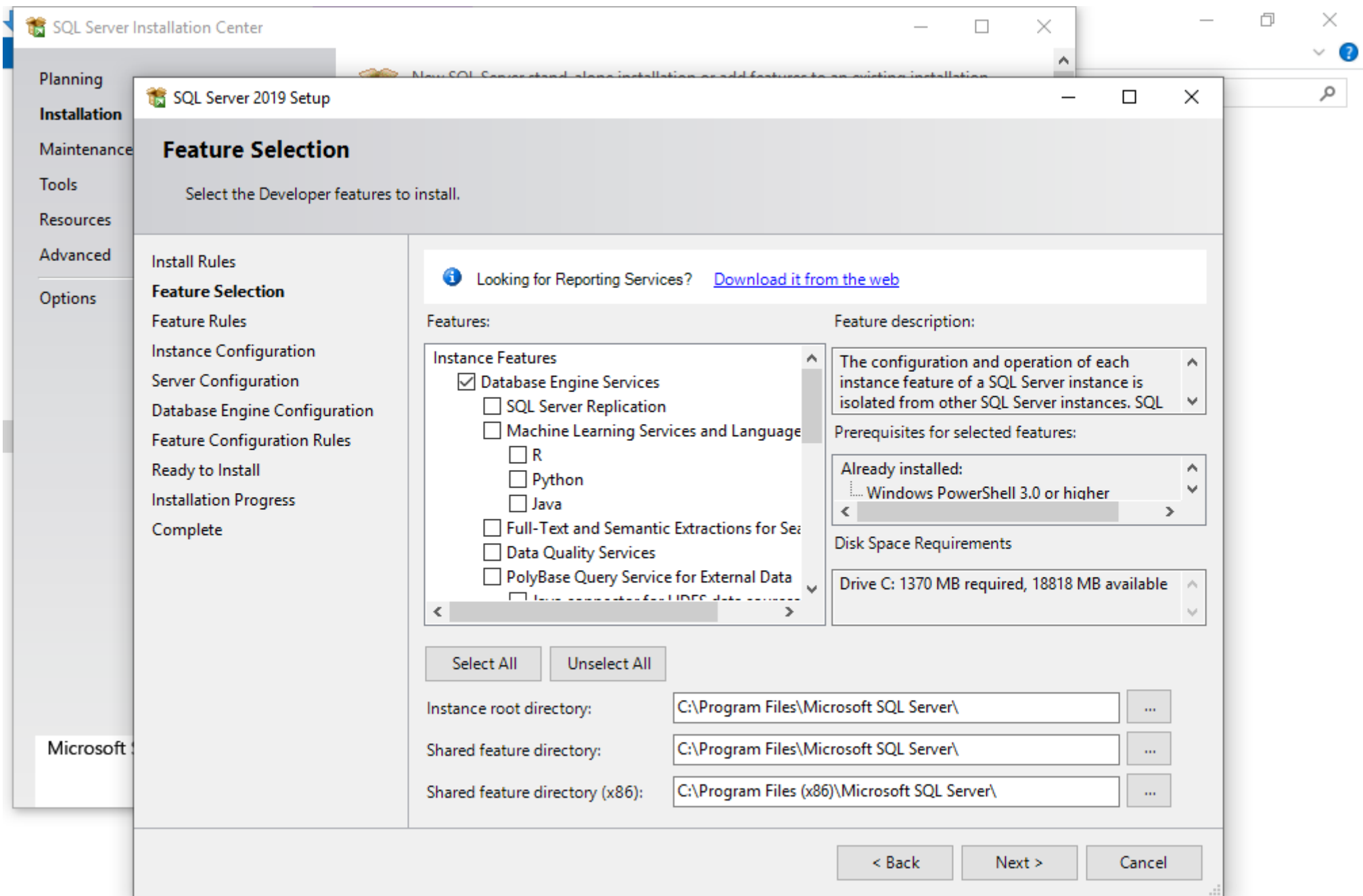
## ***Installation i Windows***

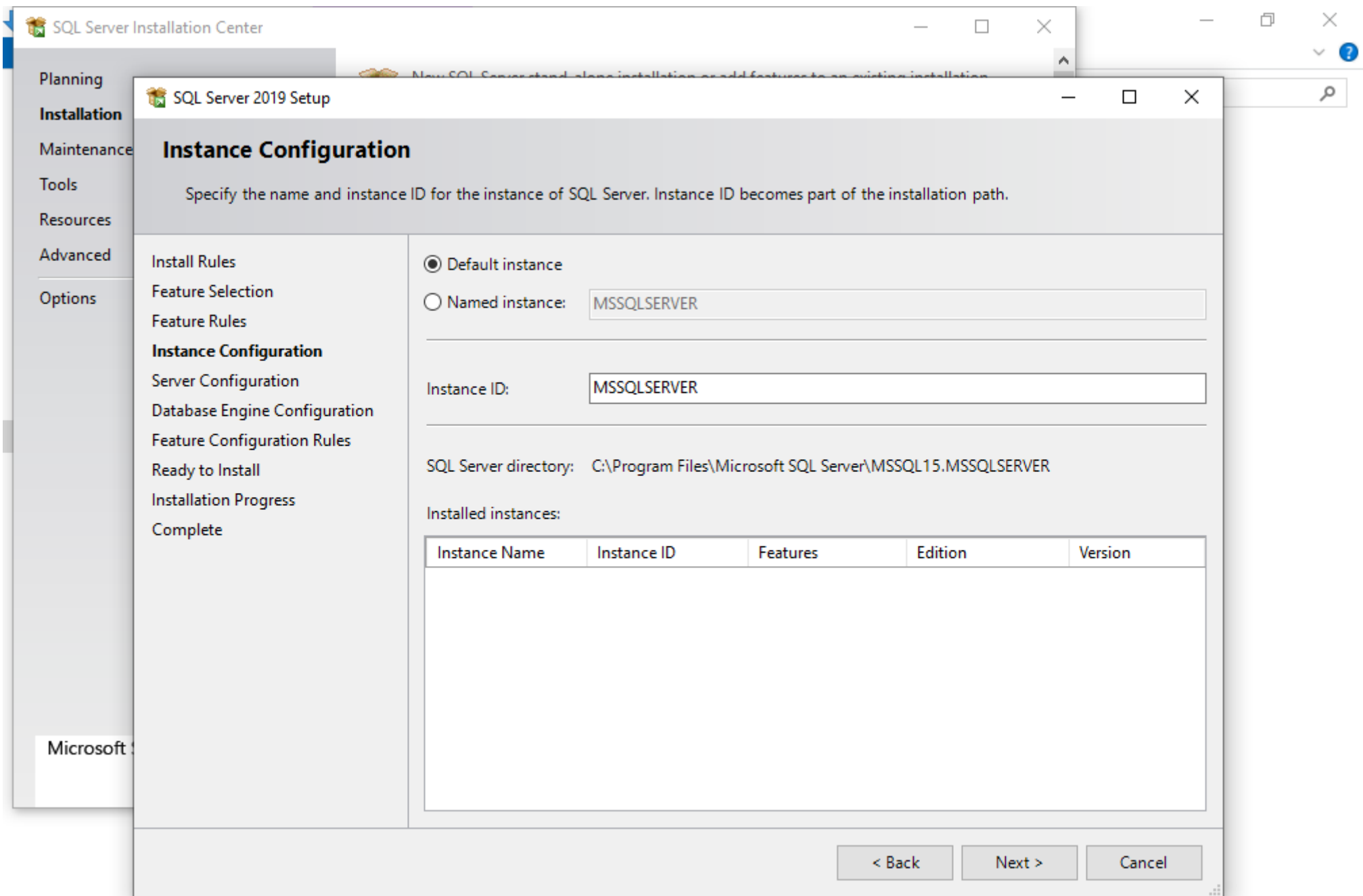
Listan nedan är på intet sätt fullständig, utan visar endast på de viktigaste inställningarna.

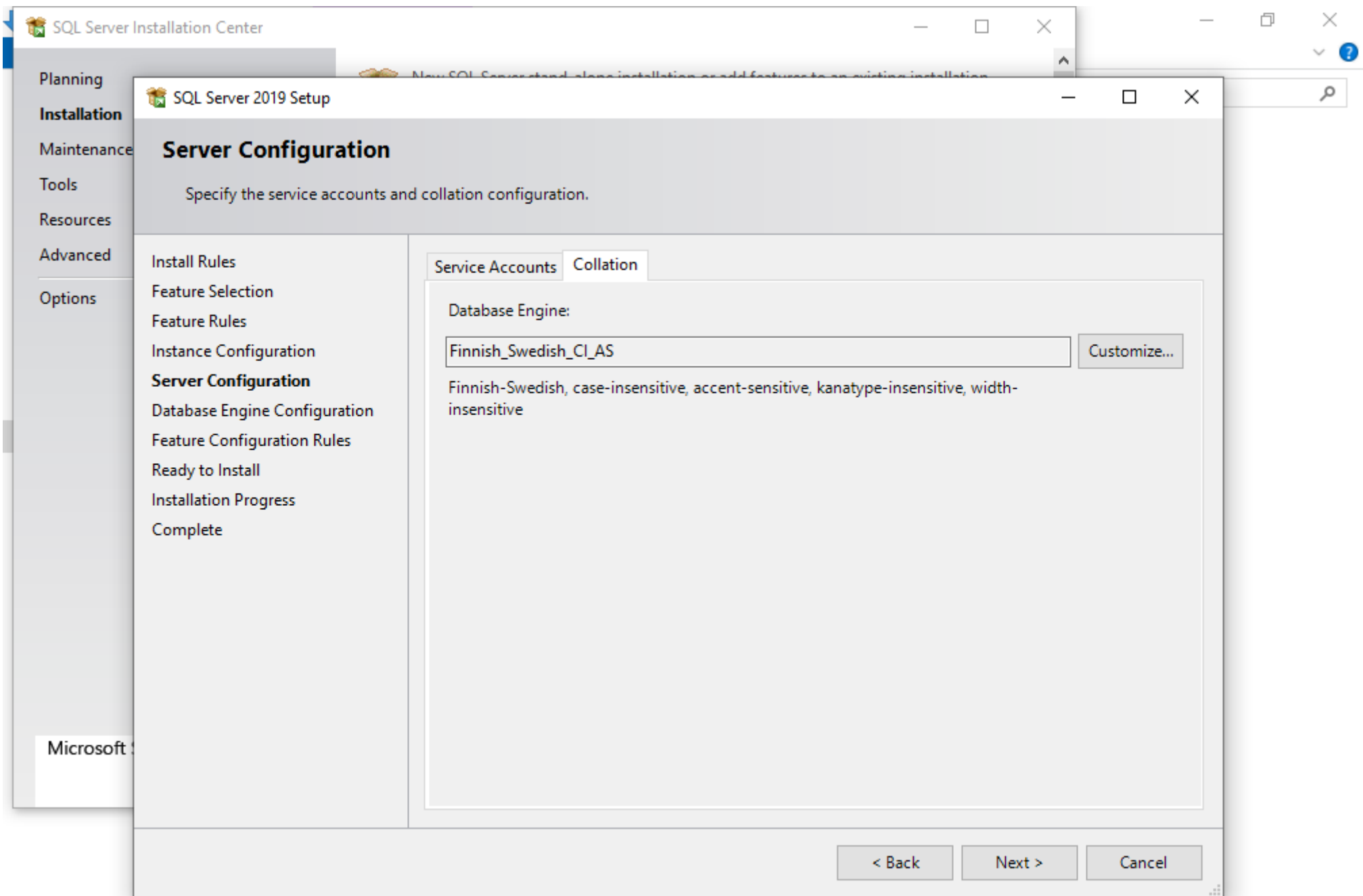
1. Välj “Custom installation type”.
2. Ange var installationsmedia finns.
3. Under “Installation”, välj “New SQL Server stand-alone installation”.
4. Under “Product Key”, välj “Developer” och acceptera villkor.
5. Vänta ut att “Install Rules” söker efter eventuella problem.
6. Under “Feature Selection”, kryssa för ...
  1. Database Engine Services
  2. Client Tools Connectivity
  3. Integration Services
  4. Client Tools Backward Compatibility
7. Under “Instance Configuration”, låt ID=MSSQLSERVER stå kvar.
8. Under “Server Configuration”, ange Collation=Finnish\_Swedish\_CI\_AS (case insensitive, accent sensitive).
9. Under “Database Engine Configuration” ...
  1. Välj Authentication=Mixed mode.
  2. Ange lösenord för kontot “sa”.
  3. Lägg till Current User som SQL Server admin.
10. Ready to install!

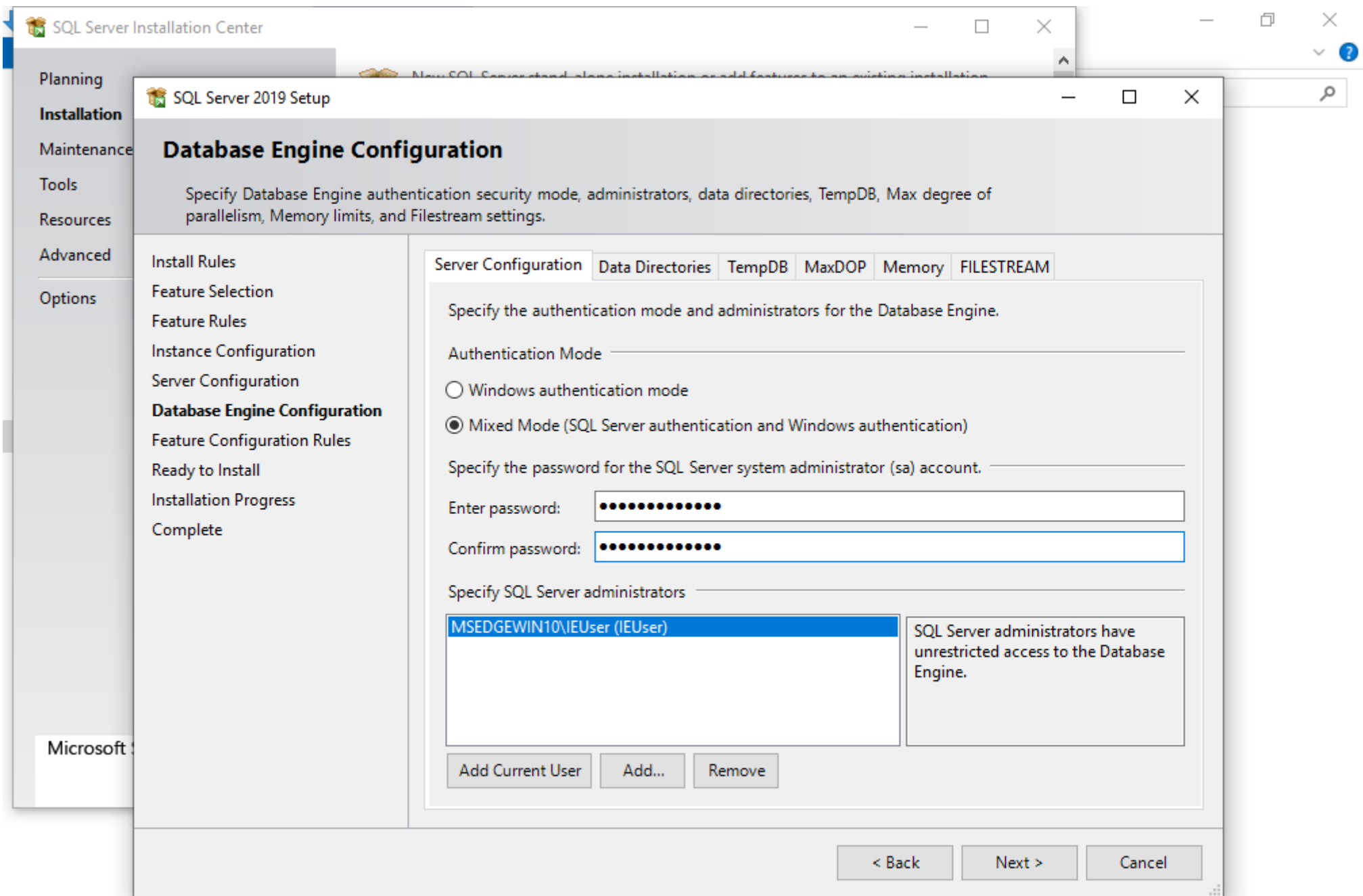
Testa att ansluta till databasen.

Vid problem, kolla att TCP/IP=Enabled under SQL Server Network Configuration.

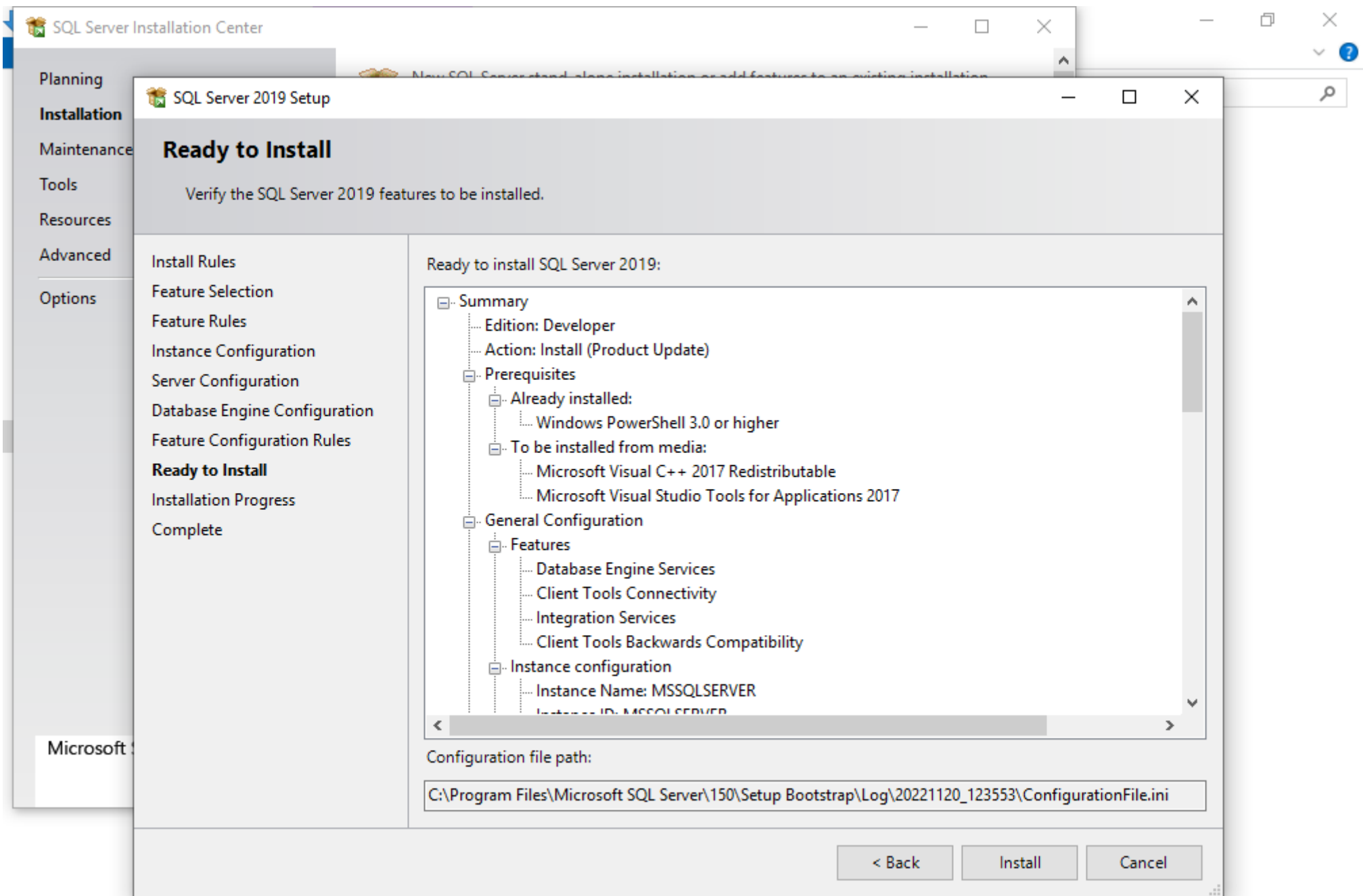












Computer Management

FileActionViewHelp

Computer Management (Local)

System Tools

Task Scheduler

Event Viewer

Shared Folders

Local Users and Groups

Performance

Device Manager

Storage

Disk Management

Services and Applications

Services

WMI Control

SQL Server Configuration Manager

SQL Server Services

SQL Server Network Configuration (32bit)

SQL Native Client 11.0 Configuration (32bit)

SQL Server Network Configuration

Protocols for MSSQLSERVER

SQL Native Client 11.0 Configuration

| Protocol Name | Status   |
|---------------|----------|
| Shared Memory | Enabled  |
| Named Pipes   | Disabled |
| TCP/IP        | Enabled  |

Actions

Protocols for MSSQLSERVER

More Actions

TCP/IP

More Actions

## ***Installation i Linux (Ubuntu 20.04)***

Kopierat från <https://learn.microsoft.com/en-us/sql/linux/quickstart-install-connect-ubuntu?view=sql-server-ver15>

1. Import the public repository GPG keys:  
`wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -`
2. Register the SQL Server Ubuntu repository:  
`sudo add-apt-repository "$(wget -qO- https://packages.microsoft.com/config/ubuntu/20.04/mssql-server-2019.list)"`
3. Run the following commands to install SQL Server:  
`sudo apt-get update`  
`sudo apt-get install -y mssql-server`
4. After the package installation finishes, run `mssql-conf setup` and follow the prompts to set the SA password and choose your edition.  
`sudo /opt/mssql/bin/mssql-conf setup`
5. Once the configuration is done, verify that the service is running:  
`systemctl status mssql-server --no-pager`
6. If you plan to connect remotely, you might also need to open the SQL Server TCP port (default 1433) on your firewall.

Se även ytterligare steg på [learn.microsoft.com](https://learn.microsoft.com/en-us/sql/linux/quickstart-install-connect-ubuntu?view=sql-server-ver15) för att installera `mssql-tools`. (Inte nödvändigt om du använder andra verktyg för att ansluta.)

För att avinstallera:

```
sudo apt-get remove mssql-server
```

För att även radera alla databasfiler:

```
sudo rm -rf /var/opt/mssql
```

## ***Användare och behörigheter***

Att använda “sa” för alla databaskopplingar är mindre bra.

Bättre skapa användare med minsta nödvändiga behörighet för olika ändamål.

Gruppera användare genom olika roller. (Mer om detta senare.)

## ***Hur är själva databasen lagrad?***

Datafiler med namn .mdf

- Uppdelade i logiska “pages”.

Loggfiler med namn .ldf

- Ändringar i schema.
- Varje insert, update, delete.
- Info om transaktioner.

Säkerhetskopior med namn .bak

## Övningar

Skissa på en lösning för att hantera Studenter och Kurser, med bland annat följande förutsättningar.

- En student har som minimum förnamn, efternamn och födelsedatum.
- En kurs har som minimum kursnamn, kurskod, antal poäng och en lärare, samt hör till ett visst övergripande ämnesområde.
- En kurs kan också ha start- och slutdatum.
- Samma student kan vara registrerad på flera olika kurser, men bara en gång per kurs.
- Man får inte radera en kurs som har studenter, eller en student som är registrerad på en kurs.
- Efter avslutad kurs får en student ett betyg (IG, G eller VG).

Rita först ER-diagram.

Översätt därefter till fysisk datamodell, och formulera lämpliga SQL-kommandon (för MS SQL Server 2019).

Befolka databasen med information om studenter, kurser mm.

Formulera SQL-frågor för till exempel ...

- Urval som visar vilka kurser de olika studenterna är registrerade på.
- Urval som visar hur många studenter som är registrerade på respektive kurs.
- Urval som visar betygsfördelningen på avslutade kurser.

Extrauppgifter:

Testa med olika användare med olika nivåer av behörighet, till exempel ...

- En användare med rätt att ändra schema för databasen.
- En användare med rätt att lägga till, ändra och radera data.
- En användare med (endast) rätt att göra urval.

Fundera över vilka olika index som skulle kunna vara användbara för databasen i exemplet ovan.

## **~~Tisdag 22 november~~ Onsdag 23 november – Hantera säkerhet, Optimera databaser**

Även denna dag undervisning på distans.

Före lunch mestadels teoretisk genomgång. Efter lunch mestadels egna övningar.

Vid kl 14.30 samling för summering av dagen och möjlighet att ställa frågor.

## ***Mer om användare, behörigheter, mm***

Lite repetition från gårdagens installation, “Windows authentication mode” vs “Mixed mode”.

Begreppen “login” vs “user”.

På servernivå “logins” och “server roles”.

På databasnivå “users” och “database roles”.

Vem är jag? Testa med “select current\_user”. Login “sa” är alltså mappat mot user “dbo”.

Skapa ett nytt login: create login my\_login with password='Testing!';

Kolla vilka login som finns: select \* from sys.sql\_logins;

Skapa en ny användare: create user my\_user for login my\_login;

Kolla vilka användare som finns: select \* from sys.database\_principals;

Men kan min användare göra något i databasen? Nedanstående bör resultera i “select permission was denied”.

execute as user='my\_user'; select \* from Students; revert;

Först måste vi dela ut några rättigheter: grant select on database::Test to my\_user;

Förkortat skrivsätt: grant control ... ger rätt att göra insert+update+delete.

Vad har jag för rättigheter? Testa med “select \* from fn\_my\_permissions('Students', 'OBJECT');”.

En snyggare lösning hade naturligtvis varit att skapa en roll, ge rollen rättigheter, och lägga till en eller flera användare som medlemmar av rollen.

Utforska gärna på egen hand. Sök efter “create role” och “alter role” + “add member”.

## ***Skapa index, begränsningar (constraints), mm***

Introduktion till index.

Kommer någon ihåg bibliotekens gamla kortregister med alfabetisk ordning och systematisk ordning?

Index går att skapa för enstaka kolumner eller för kombinationer av två eller flera kolumner.

Poängen är att förenkla och snabba upp sökning i tabeller.

Nyttan av index är i högsta grad beroende av vad för slags frågor vi ställer till databasen.

Ska index vara i stigande eller fallande ordning?

Ett smart drag är att indexera kolumner som används vid join-operationer.

Primära nycklar blir alltid också index, men inte samma sak för främmande nycklar.

Om “select” går fortare kommer dock “insert”, “update” och “delete” att gå långsammare.

När vi ändrar i en tabell måste ju även index uppdateras.

Via till exempel “Estimated plan” i Azure Data Studio går att skapa sig bilder av hur olika SQL-frågor utförs.

Det går även att använda sig av SQL-kommandot “set showplan\_xml on”.

En specialvariant av index är dem vi skapar med “unique”.

Andra slags begränsningar, eller “constraints”, vi har stött på tidigare:

- Default-värden (“default”).
- Giltiga värden (“check”).
- Obligatoriska fält (“not null”).



## ***Använda transaktioner***

Nyttan av att kunna använda transaktioner, sammanfattad med ett enkelt exempel:

```
begin transaction;  
update Students set name='Nisse' where name='Johan';  
select * from Students;  
rollback transaction;  
select * from Students;
```

Hade vi velat spara ändringarna skulle vi istället för “rollback” ha gjort “commit”.

“Allt eller inget.”

I princip ett måste att ha transaktionshantering när flera olika användare nyttjar samma databas.

Inne i en transaktion ser jag min uppdaterade eller ändrade “bubbla”. Övriga användare ser ännu inte ändringarna.

Men vad händer om två transaktioner låser varandra?

Lite överkurs: Med “save transaction” går också att sätta en “savepoint” i en transaktion.

## Övningar

Om man inte är klar med gårdagens övningar går att spara dem till senare.

Testa att skapa egna login och users till din databas, och dela ut olika slags rättigheter till dem.

Pröva att skapa roller för att gruppera databas användare, och ge rollerna rättigheter, istället för att ge enskilda användare rättigheter.

Utforska olika sätt att granska “query execution plans”, och testa vilka slags index som verkar förbättra (eller försämma?) prestandan.

Ta reda på hur man kan använda funktioner i en check-constraint (till exempel för att jämföra med innevarande års årtal).

Bekanta dig med transaktionshantering.

Använd två olika klientprogram och undersök hur poster blir låsta för en användare om annan användare är inne i en transaktion.

## **Presentation av inlämningsuppgift**

Inlämning senast fredag 2 december kl 16. I praktiken måndag 5 december kl 09.

Inlämning i pdf-format, ett samlat dokument, inga separata bilagor. Antalet sidor bör inte överstiga 20.

*”Please, do not consider the page limit as a target!  
It is in your interest to keep your text as concise as possible,  
since experts rarely view unnecessarily long proposals in a positive light.”*

Tänk på att hålla nere filstorleken. Om du infogar bilder, se till att komprimera dem först. Undvik tunga skärmdumpar.

Uppgiften ska redovisas individuellt. Under arbetets gång får du dock gärna ta hjälp av andra för att diskutera designval, problem mm.

Uppgiften och förutsättningarna är dessa:

Skapa en databas åt en hyresvärd i en stad som vill hålla reda på lägenheter uthyrda till personer.

En lägenhet har ett antal rum och en area i kvadratmeter.

En lägenhet är belägen på ett våningsplan och har ett ordningsnummer inom detta våningsplan (plan + nummer är unikt för varje adress).

En lägenhet har en gatuadress. På samma adress kan det finnas en eller flera lägenheter.

En gatuadress hör till ett hus. Ett hus kan ha en eller flera adresser (olika portar).

Ett hus ligger i ett kvarter, och har ett ordningsnummer inom det kvarter där det ligger. Ett kvarter kan ha ett eller flera hus.

Ett kvarter har en fastighetsbeteckning sammansatt av kommunnamn, traktnamn, blocknummer och enhetsnummer (tex “Stockholm Norrmalm 5:3”).

En person har personnummer, för- och efternamn, telefonnummer och epostadress. Telefonnummer och epostadress behöver inte vara unika.

En person kan endast hyra en lägenhet i taget. En lägenhet kan vara uthyrd till flera personer som delar på avtalet.

Ett hyresavtal gäller från ett startdatum och har en årshyra i kronor.

Om hyran ändras skapas ett nytt avtal med nytt startdatum, och det gamla avtalet förses med ett slutdatum som är dagen före det nya avtalets startdatum.

Se även uppgifterna på nästa sida för ledtrådar till vad som ska vara möjligt att få fram ur databasen.

Kom ihåg: Databasen kommer omöjligen kunna motsvara verkligheten fullständigt, men du ska åstadkomma bästa möjliga likhet med verkliga förhållanden.

Fortsättning på nästa sida ...

## ***Inlämningsuppgift, fortsättning***

Uppgiften ska lösas genom att ...

1. Rita ER-diagram. Identifiera entiteter och relationer, samt kardinaliteter. Använd helst “crow’s foot notation”.  
Beskriv kortfattat hur du har tolkat kraven, och motivera de viktigaste designvalen du har gjort.  
Redovisa med bild av diagram och korta förklarande texter.
2. Skapa tabeller, med ER-diagrammet som underlag. Koppla relaterade tabeller med primära och främmande nycklar.  
Lägg till de regler för giltiga värden, unika index mm du anser är de viktigaste, och motivera kortfattat varför.  
Redovisa med SQL-kommandon för att skapa tabeller mm, i textform, med korta förklarande texter.
3. Lägg till exempeldata, antingen genom att infoga en post i taget eller genom att importera från textfiler,  
med som minimum två kvarter med vardera minst två hus, med i sin tur minst två lägenheter per hus, alla uthyrda till olika personer.  
Redovisa med SQL-kommandon för att lägga till data.
4. Göra följande urval från databasen.
  1. Lista aktiva hyresgäster i bokstavsordning, med uppgifter om adresser och övriga egenskaper för de lägenheter de hyr.
  2. Lista kvartersnamn i bokstavsordning, med uppgifter per kvarter om antal hus, antal lägenheter, och totala antalet kvadratmeter.
  3. Lista uppgifter om antal kvarter, antal hus, antal lägenheter, och totala antalet kvadratmeter, grupperat per traktnamn (del av fastighetsbeteckning).
  4. Lista vad hyrorna kommer att bli nästa kalendermånad för alla icke avslutade hyresavtal.Redovisa med SQL-kommandon för urval, med vid behov korta förklarande texter.

För betyget G ska uppgifterna 1-3 och 4.1-4.2 vara utförda på ett i stora drag godtagbart sätt.

För betyget VG ska uppgifterna 1-2 dessutom vara utförda med välgrundade resonemang kring designen, samt även uppgifterna 4.3-4.4 vara utförda.