Windows Server og MS SQL

# SQL case 1

## Installation af Windows Server

For at installere Windows Server skal man lave en virtuel pc på Hyper-V hvorefter man linker Iso filen til den virtuelle så den loader windows når man starter den virtuelle, derefter følger du vejledningerne som Windows giver og til sidst er du ved login skærmen. Vi skal nu til at installere de services vi skal bruge for at have en Sql server. Hvis man klikker på add features knappen kommer der et vindue op hvor man kan vælge nogen services og pakker man kan installere. I det her tilfælde vil vi gerne installere Net Framework 3.5 R1 da det er en ting man skal bruge for at køre MS SQL, når installationen er færdig er det en god ide at genstarte serveren.

## Installation af MS SQL

Nu skal vi installere SQL, så vi starter med at mounte iso filen som vi har på computeren så vi kan se den i den virtuelle. Når det er gjort burde den gerne komme op som en cd, vi skal køre den setup som er på disken. Når vinduet kommer frem skal vi klikke på installation og derefter på den knap hvor der står install SQL. når den når til det step hvor den spørg dig om de ting der skal installeres vælger du bare dem alle, når installationen er færdig burde din SQL server virke.

## Oprettelse af DB og Tables

Nu skal vi til at oprette Databaser og tables i SQL serveren, det gør vi ved at åbne et program som hedder SQL Management. Når programmet starter op spørg den dig om du vil logge ind på en server, du skal vælge den SQL server du lige har installeret.

når du er logget på burde serveren ligge ovre til venstre. højreklik på den og klik på new query. Vi skal nu lave noget kode som får SQL serveren til at oprette en database og tables.

Koden som skal indtastes er følgende:

CREATE DATABASE Tec;

USE Tec;

CREATE TABLE PostNrBy (

PostNr int PRIMARY KEY,

ByNavn varchar(250)

);

CREATE TABLE Klasse (

KlasseID int PRIMARY KEY IDENTITY (1,1) I,

KlasseNavn varchar(250)

);

CREATE TABLE Elev (

ElevID int PRIMARY KEY IDENTITY (1,1),

Fornavn varchar(250) ,

Efternavn varchar(250),

Adresse varchar(250),

PostNr int FOREIGN KEY REFERENCES PostNrBy(PostNr)

);

CREATE TABLE Laerer (

LaererID int PRIMARY KEY REFERENCES Klasse(KlasseID),

Fornavn varchar(250),

Efternavn varchar(250),

Adresse varchar(250),

PostNr FOREIGN KEY REFERENCES PostNrBy(PostNr)

);

## Diagram af Tables

# 

# SQL Case 2 Task 1

## Opsætning

Før vi kan tilgå vores databaser skal vi først slå firewallen fra så vores forbindelser ikke bliver stoppet, derefter skal vi installere en til MS SQL Server. Når den er installeret skal vi oprette et table på begge servere, den ene skal hedde Personer og den skal indeholde ID, Fornavn, Efternavn, Adresse og Fødselsdag og den anden skal hedde Cprnr og den skal indeholde ID og CPR.

## C# program som kan kommunikere med de to servere

C# programmet som vi har brugt ligger inde på google drev på det her link: goo.gl/whbjB1

## Diagram af Tables

# 

## 

# SQL Case 2 Task 2

## Løsning

Løsningen til den her task er at man tager alle tables og får dem skrevet i csv så de kan uploades til google cloud, for at have backup på MS SQL serveren ville man skulle lave et program som kan tage tabellerne fra google cloud og lave dem om til xml fil og så indtsætte dem i MS SQL serveren.

# SQL Case 2 Task 3

## Kode til Powershell

### Kode til Variabel

$var = "Hej med dig"

$var

### Kode til Array

$MitArray = 10,"hej","med","dig"

$MitArray

### Kode til While

$x = 1

$MitArray = 10,"hej","med","dig"

[System.Collections.ArrayList]$MitArray1 = $MitArray

$MitArray1.GetType()

while ($x -le 10)

{

$MitArray1.Add($x)

$x = 1 + $x

}

$MitArray1

### Kode til Netværksshare

New-Item "C:\Shared" -type directory

New-SmbShare -Name "Shared" -Path "C:\Shared" -FullAccess "Everyone"

# SQL Case 2 Task 4

## Kode til powershell

### Kode til forbindelse til MS SQL

$SQLServer = "Server navn"

$SQLDBName = "Databasenavn"

$SQLPassword = "Kode"

$SQLUsername = "Brugernavn"

$SqlQuery = "select \* from test"

$SqlConnection = New-Object System.Data.SqlClient.SqlConnection

$SqlConnection.ConnectionString = "Server = $SQLServer;Database = $SQLDBName;User ID = $SQLUsername;Password = $SQLPassword; Integrated Security = True"

$sqlCmd = New-Object System.Data.SqlClient.SqlCommand

$sqlCmd.CommandText = $SqlQuery

$sqlCmd.Connection = $SqlConnection

$SqlAdapter = New-Object System.Data.SqlClient.SqlDataAdapter

$SqlAdapter.SelectCommand = $sqlCmd

$DataSet = New-Object System.Data.DataSet

$SqlAdapter.Fill($DataSet)

$SqlConnection.Close()

$DataSet.Tables[0]

# 

# 

# SQL Case 2 Task 5

## Kode til oprettelse af Xml

### Kode til RAW

SELECT \* FROM Test as t FOR XML RAW;

### Kode til AUTO

SELECT \* FROM Test as t FOR XML AUTO;

### Kode til PATH

SELECT \* FROM Test as t FOR XML PATH (‘Person’), root (‘Test’);

## Forklaring

Du kan bruge xml til at rykke data fra den ene server til den anden da xml koden er skrevet på en måde så serverne kan se hvordan det er struktureret.

# SQL Case 2 Task 6

## C# koden

Link til kode: <https://goo.gl/tscNda>

# SQL Case 2 Task 7

## Codeplex

Codeplex er et sted på nettet hvor man kan have sin kode og dele den så andre folk kan hjælpe en og bruge koden

## Forskellen mellem Codeplex og Github

Der er ikke så stor forskel på de to steder da de virker på samme måde og har samme formål, og det er også grunden til at codeplex lukker ned her til efteråret.

## Hvad kan man bruge Github til?

Man kan bruge github til at samarbejde om projekter selvom man ikke arbejder samme sted og så kan man bruge det til at få hjælp hvis der er noget man ikke kan finde ud af eller hvis der er en anden som har lavet en bedre kode kan man bruge den kode da det hele er open source

## Hvad kan man bruge Codeplex til?

Det samme som github men med lidt færre features da siden jo er ved at lukke.