

Datatyper och variabler

Microsoft SQL Server

Datatyper

- Olika typer av data lagras på olika sätt i en databas.
- När man skapar en tabell behöver man därför ange en datatyp för varje kolumn så databashanteraren vet hur informationen ska lagras.
- Olika databaser (mjukvaror) har olika datatyper.
- De flesta databaser har ett ganska stort antal olika datatyper att välja mellan, men oftast är det ett relativt fåtal som är vanligt förekommande.
- Vi ska nu kolla på några vanliga datatyper i MS SQL Server.

Vanliga datatyper

Datatyp	Lagrar	Exempel-värde
bit	Booleskt värde	1
int	Heltal	5392
float	Flyttal	3.141592
datetime2	Datum och tid	2019-09-24 18:50:00.000
nvarchar	Text	Hello world!

Identity

"Identity" är en egenskap man kan välja att sätta på en kolumn av datatyp 'int' (eller andra heltals-typer). Databashanteraren kommer då själv generera värden för denna kolumn enligt en given sekvens (vanligtvis 1, 2, 3, 4, 5 ... o.s.v)

När man sedan sätter in nya rader i tabellen hoppar man helt enkelt över att ange något värde för "Identity"-kolumnen, som automatiskt får nästa värde i sekvensen.

Null

- NULL är ett speciellt värde som alla datatyper kan anta, och som används för att representera "värde saknas".
- Ett fält som är NULL saknar alltså värde, och är **inte** samma sak som ett fält som är 0 eller en tom textsträng "".
- När man skapar en tabell kan man ange för varje kolumn om den ska godta NULL-värden eller inte.
- Man kan använda IS NULL i vilkor för att kolla om ett värde saknas.
- Exempel: `Select * from users where username is null`

Konvertera mellan datatyper

Ibland kan man behöva konvertera en datatyp till en annan. Säg att du vill läsa ut en datetime och t.ex lagra i ett nvarchar-fält (kanske för att du klippt in datumet i en längre text) då kan du använda "convert".

Exempel: `convert(nvarchar, @myDatetime, 121);`

121 är en formateringskod som anger hur DateTime-värdet ska representeras som en sträng.

Deklarera variabler

- För att använda variabler i SQL behöver man först deklarerat dem.
- Syntax: `DECLARE @variabelnamn (as) datatyp`
- Exempel: `Declare @username as nvarchar(max);`

Sätta värden på variabler

Man kan sätta värden på variabler när man deklarerar dem:

```
Declare @username as nvarchar(max) = 'Admin';
```

... eller så kan man sätta/ändra det senare:

```
SET @username = 'Fredrik';
```

Man kan även sätta ett värde från en tabell till en variabel:

```
SET @username = (Select top 1 username from users);
```


Sätta variabler med select

I stället för SET kan man använda SELECT för att sätta variabler:

```
select top 1 @name = user, @pass = password from users;
```

```
select @allUsers += user + ';' from users;
```

Referera till värden i variabler

Man kan använda "print" för att skriva ett värde till "meddelanden":

```
print @username;
```

Man kan även använda variabler i queries:

```
select * from users where username = @username;
```

Globala variabler

Globala variabler är variabler som SQL server sätter automatiskt men som vi kan läsa och använda oss av. Några exempel:

Variabel	Returnerar
@@ROWCOUNT	Antal rader som påverkats av senaste kommando.
@@SPID	Server process ID på
@@ERROR	Status på senast exekverade kommande. (Succeeded eller Failed)
@@IDENTITY	Senaste värde som satts på en identity-kolumn via insert eller select into.

Konfigurationsvariabler

Det finns även variabler för att styra hur SQL server beter sig. Exempel:

Syntax	Användning
SET NOCOUNT { ON OFF }	När NOCOUNT = ON så visas inte antal rader i meddelanden.
SET STATISTICS TIME { ON OFF }	Anger om statistik över tidsanvändning ska visas för uttryck.
SET SHOWPLAN_XML { ON OFF }	Får servern att inte exekvera queries utan istället visa en analys av hur.

Temporära tabeller

- Temporära tabeller kan användas för att spara resultatet från en query som man behöver (temporärt) komma åt flera gånger.
- Temporära tabeller namnges med # före namnet (#exempel) och går endast att komma åt från samma session där den skapades.
- ... om det inte är en global temporär tabell, då har den ## före namnet, och går att komma åt mellan sessioner.
- SQL server droppar automatisk temporära tabeller när sessionen som skapade dem stängs.
- Man kan dock även droppa dem manuellt om man så önskar.

Tabell-variabel

Man kan även lagra resultatet från en query i en variabel. Man deklarerar då variabeln som datatyp table:

```
DECLARE @people as TABLE  
(  
    name nvarchar(max),  
    age int  
);
```

... och kan sedan använda den som en vanlig tabell