NET2 - 3

SQL 1 - SELECT

Förra gången

- Introduktion till kursen NET2
- Grunderna om databaser
- Installation
  - Microsoft SQL Server
  - Microsoft SQL Management studio

Idag

- Utvecklingsmiljön
- SQL 1
  - SELECT

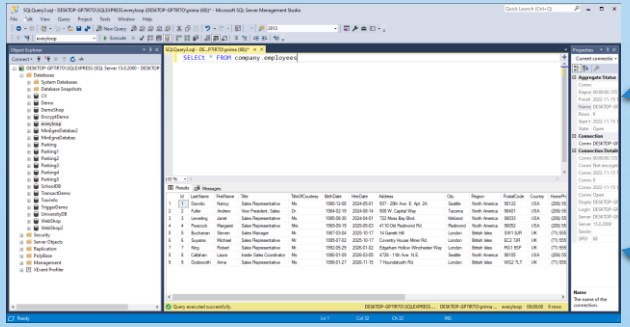
Utvecklingsmiljön

Utvecklingsmiljön

Vår dator

SQL Server

SQL Server Management Studio (SSMS)



SQL Server

SQL Server

Utvecklingsmiljön

- SQL Server
  - Saknar GUI
  - Är en process som körs hela tiden på datorn
  - Kan bara ta emot och svara på frågor/queries
  - Allt som sparas på SQL-servern stannar där I evighet (tills man tar bort det aktivt)

- SQL Server Management Studio (SSMS)
  - Erbjuder ett GUI
  - Är ett program som kommunicerar med olika SQL-servrar (En av dem har vi installerat på vår dator)
  - Skickar frågor och visar vilket svar vi får från SQL-servern
  - De frågor vi skriver och skickar kan vi spara, om vi vill

Introduktion till SQL

Structured Query Language

Relationsdatabaser

- En databas är en samling information som är organiserad för att man enkelt ska kunna söka och ändra enskilda delar av informationen.
- En databashanterare (DBMS) är den mjukvara som används för att söka och manipulera data i en specifik typ av databas.
- Det finns olika sorters databaser. Vi kommer fokusera på den kategori av databaser som kallas relationsdatabaser, som är väldigt vanliga.
- Relationsdatabaser lagrar data i tabellform, och ofta är det relationerna mellan olika tabeller som är det mest intressanta. Därav namnet relationsdatabas.

Structured Query Language (SQL)

- SQL är ett standardiserat programspråk för att hämta och modifiera data i relationsdatabaser (RDBMS).
- Språket utvecklades först av IBM under 70-talet.
- SQL uttalas vanligen bokstav för bokstav, alltså S-Q-L, men ibland hör man det även uttalas som engelskans ”sequel”.
- Olika databashanterare (t.ex Oracle, Postgres och MySQL) använder olika dialekter av SQL. Variationerna är dock oftast relativt små.
- I denna och kommande lektioner använder vi T-SQL som är den variant som Microsoft använder i sin SQL-server.

Queries (frågor)

- För att hämta ut eller ändra information i en relationsdatabas skickar man så kallade queries (frågor) till databashanteraren (servern).
- En query kan vara allt från väldigt enkel (”ge mig all data i tabell xyz”) till väldigt komplex (t.ex. korsreferera data från flera tabeller med en mängd villkor för exakt vilka data som man vill få ut).
- I denna lektion tittar vi på några av de vanligaste uttryck man använder i SQL för att plocka ut data från en tabell.

Syntax

- SQL är INTE case-sensitive, men det är bra att man anger alla SQL-kommandon med VERSALER, så de syns tydligare.
- Tabellnamn kan anges utan hakparenteser, men innehåller de mellanslag eller svenska tecken behöver de omslutas av halparenteser

- **SELECT * FROM users** och **SELECT * FROM [users]** fungerar lika bra.
- SQL behöver inte avslutas med ;, men vid mer komplexa uttryck är det lämpligt att göra det.

- Förslag på grundsyntax:
  - **SELECT * FROM users;**

CRUD

- Create
  - SQL: INSERT
- Read:
  - SQL: SELECT
- **Update:**
  - **SQL: UPDATE**
- Delete
  - SQL: Delete

SELECT

- När vi vill hämta och visa data från en tabell använder vi ”SELECT”:
  Syntax: SELECT (lista med) kolumnnamn FROM tabellnamn

```
SELECT id, username, password FROM users;
```

Id	Username	Password
1	Admin	Abc123
2	Claes	Secret

- För att ta ut alla kolumner i en tabell kan man skriva:

```
SELECT * FROM tabellnamn;
```

SELECT

- När vi vill hämta ut data från en databastabell så använder vi SQL-kommandot SELECT
- Syntax:

```
SELECT * FROM tabellnamn
```

- * = alla kolumner
- Exempel:
  - `SELECT * FROM users;`

Id	Username	Password
1	Admin	Abc123
2	Claes	Secret

SELECT

- Det går också bra att bestämma vilka kolumner som ska visas.

- Exempel:
  - `SELECT id, username FROM users;`

Id	Username
1	Admin
2	Claes

TOP X *

- Man vill i princip aldrig be om all data i en tabell (SELECT *) då t.ex. tabellen ”users” kan innehålla tusentals användare.
- Ett sätt att begränsa antal rader är genom ”TOP x” som begränsar resultatet till x rader.

```
SELECT TOP 10 * FROM USERS;
```

WHERE

- Ett annat sätt att begränsa är att bara be om rader som matchar ett givet vilkor. Detta gör man med ”WHERE”.

```
SELECT * FROM Users WHERE Username = 'Micke';
```

Id	Username	Password
2	Micke	Secret

Logiska operationer

```
SELECT * FROM countries
WHERE country = 'Sweden’ AND population > 10000;

Select * from countries
where country = 'Sweden' or country = 'Norway';


Select * from countries where not country = 'Sweden';
```

```
Select * from countries where country <> 'Sweden';
```

IN

- I exemplet med ”or” ovan så tog vi ut alla rader där country är 'Sweden' eller 'Norway'. Ett smidigare sätt är att använda ”in”.

```
Select * from countries where country in
('Sweden', 'Norway', 'Denmark');
```

- Man kan även använda ”not in” för att få alla städer från tabellen som inte ligger i de länder man anger.

```
Select * from countries where country not in ('Sweden', 'Norway',
'Denmark');
```

BETWEEN

- Man kan även använda **between** för att ange ett spann av värden:

```
Select * from countries where population between 500000
and 1000000;
```

```
Select * from airports
where IATA between 'AAF' and 'AAJ';
```

LIKE

- ”Like” används när man vill matcha textfält mot ett specifikt mönster.
- T.ex. alla textfält som börjar på 'B'.
- Uttrycken i tabellen kan kombineras till ett mönster som beskriver data.

```
SELECT * From countries where country like 'b%’;
/* Alla länder som börjar på B */

SELECT * From airports where IATA like '[acf]_[q-z]’;
/* Alla IATA som börjar på a, c eller f och slutar på q-z

SELECT * From countries where country like '%land%’;
/* Alla länder som innehåller ordet “land”
```

Uttryck	Beskrivning
%	Vilken sträng av tecken som helst.
_	Ett tecken, vilket som helst.
[ ]	Ett tecken i ett set [abcd] eller spann [a-d].

ORDER BY

- Man kan välja att sortera resultatet på en eller flera kolumner.
- Om man t.ex. vill sortera de rader man tar ut från tabellen ”people” på efternamn, och i andra hand på förnamn (i de fall flera personer har samma efternamn) kan man skriva:

```
select * from users order by firstname;

select * from users order by firstname DESC;

select * from users order by lastname, firstname;
-- Sorterar först på lastname, och sedan på
firstname
```

- Efter varje kolumn vi sorterar på så kan vi lägga till ”asc” (ascending) eller ”desc” (descending) för att ange om det ska vara stigande eller fallande ordning. Anger man inget så används ”asc” som default.

DISTINCT

- Vill man bara ha ut unika värden så kan man använda ”distinct”.

```
Select distinct Lastname from users
order by Lastname;
```

ALIASES

- Aliases används för att i sin query referera till t.ex kolumner eller tabeller med ett annat namn än de har i databasen.
- Detta kan vara användbart t.ex för att slippa skriva så mycket, eller för att man vill att resultatet ska visas med ett annat namn.

```
select id, username as 'Användarnamn', password as 'Nyckel'
from users;
```

id	Användarnamn	Nyckel
1	Admin	Abc123
2	Fredrik	Secret

CASE WHEN

- Med konstruktionen case-when kan man välja att visa olika värden i resultatet beroende på vilkor man angett (som utvärderas per rad).

```
Select City,
     case
          when population < 1500 then 'Village'
          when population < 50000 then 'Town'
          else 'City'
     end
as 'Classification'
From UScities;
```

Sätta ihop strängar

- Ibland vill man lägga till information I tabellposterna
- Då kan man använda ett +.

```
SELECT TOP 1000 [Name] + ' med koden ' + [Code] as 'Färger'
      ,[Red]
  FROM [Colors]
```

```
Formatera siffror, och de blir strängar:
SELECT TOP (1000) [Name] + ' med koden ' + [Code] as 'Färger'
      , FORMAT(Red, '000')
  FROM [Colors]
```

Matematiska uttryck

- Det går att utföra beräkningar med hjälp av SQL, på de värden som är lagrade I databasen.
- ```
  SELECT TOP (1000) [TrackId]
  ```
- ```
        ,[Name]
  ```
- ```
        ,([Milliseconds] / 1000) as Seconds
  ```
- ```
        ,([Bytes] *  0.000009536) as Mb
  ```
- ```
        ,[UnitPrice] * 12 as SEK
  ```
- ```
    FROM [everyloop].[music].[tracks]
  ```

FORMAT

- Tal kan vara bra att formattera på olika sätt.
- Exempel
  - `SELECT Productname, FORMAT(UnitPrice, '00.00') FROM company.products;`
  - `/* Ger t ex 18.00 om Unitprice är 18. */`

  - `SELECT Country, Population, FORMAT(Population, '### ### ###') FROM Countries`
  - `/* Ger 2 976 372  om Population är 2976372 */`

SQL är mer än bara SELECT-satser

- Man kan göra mycket mer med SQL än att bara plocka ut data från tabeller som vi gjort i exemplen ovan. Några viktiga användningsområden som vi kommer att se på framöver är:
  - Lägga till, uppdatera och ta bort rader i tabeller.
  - Skapa och ta bort tabeller och så kallade vyer.
  - Plocka ut data på aggregerad nivå.
  - Korsreferera data från flera tabeller.
  - Skriva funktioner och procedurer med flödeslogik som mer liknar ”vanlig” programmering så som vi är vana vid från t.ex c#.

Demo

- Importera data I database
  - Dokument: everyloop.sql
- Göra querys
  - Dokument: demosql.txt

Småövningar

- Databasen Everyloop
  - Sortera company.employees efter personernas ålder, äldst först.
  - Visa tabellen Moonmissions, bara kolumnerna Spacecraft och Outcome ska visas, sorterat på Launchdate.
  - Visa bara information om de här länderna, i tabellen Countries: 'Sweden', 'Thailand', 'Germany', 'Brazil'
  - Lista enbart färgnamnet på tabellen Colors där färgvärdet för Blue är högre än 128.
  - Lista bara Svenska flygplatser, från tabellen Airports

  - Alla övningar är enskilda.

Övningar - 1

- Använd databasen Everyloop
  - Skapa SQL-queries så de stämmer med screenshot
    - Lista på unik Region
  - Databastabell: company.customers

	Våra områden
1	British Isles
2	Central America
3	Eastern Europe
4	North America
5	Northern Europe
6	Scandinavia
7	South America
8	Southern Europe
9	Western Europe

Övningar - 2

- Använd databasen Everyloop
  - Ta ut data (select) från tabellen GameOfThrones på sådant sätt att ni får ut en kolumn 'Title' med titeln samt en kolumn 'Episode' som visar episoder och säsonger i formatet "S01E01", "S01E02", osv. Tips: FORMAT()

	Title	Episode
1	Winter Is Coming	S01E01
2	The Kingsroad	S01E02
3	Lord Snow	S01E03
4	Cripples, Bastards, and Broken Things	S01E04
5	The Wolf and the Lion	S01E05
6	A Golden Crown	S01E06
7	You Win or You Die	S01E07
8	The Pointy End	S01E08
9	Baelor	S01E09
10	Fire and Blood	S01E10
11	The North Remembers	S02E01
12	The Night Lands	S02E02
13	What Is Dead May Never Die	S02E03
14	Garden of Bones	S02E04
15	The Ghost of Harrenhal	S02E05

Övning - 3

- Använd Table: Countries
  - Lista de 5 mest folkrika länderna I världen

	Country	Region	Population
1	China	ASIA (EX. NEAR EAST)	1313973713
2	India	ASIA (EX. NEAR EAST)	1095351995
3	United States	NORTHERN AMERICA	298444215
4	Indonesia	ASIA (EX. NEAR EAST)	245452739
5	Brazil	LATIN AMER. & CARIB	188078227

  - Lista de 15 länder som är trängst att bo I
    - Ledtråd: population / [Area (sq# mi#)]

	Country	Region	PeoplePerSqMi
1	Monaco	WESTERN EUROPE	16271
2	Macau	ASIA (EX. NEAR EAST)	16183
3	Singapore	ASIA (EX. NEAR EAST)	6482
4	Hong Kong	ASIA (EX. NEAR EAST)	6355
5	Gibraltar	WESTERN EUROPE	3989
6	Gaza Strip	NEAR EAST	3968
7	Malta	WESTERN EUROPE	1266
8	Bermuda	NORTHERN AMERICA	1241
9	Maldives	ASIA (EX. NEAR EAST)	1196
10	Bahrain	NEAR EAST	1050
11	Bangladesh	ASIA (EX. NEAR EAST)	1023
12	Guernsey	WESTERN EUROPE	838
13	Jersey	WESTERN EUROPE	785
14	Barbados	LATIN AMER. & CARIB	649
15	Taiwan	ASIA (EX. NEAR EAST)	640

	Symbol	Name	Period	Form
1	Cl	Chlorine	3	Gas
2	Ar	Argon	3	Gas
3	Kr	Krypton	4	Gas
4	Xe	Xenon	5	Gas
5	N	Nitrogen	2	Gas
6	O	Oxygen	2	Gas
7	F	Fluorine	2	Gas
8	Ne	Neon	2	Gas
9	H	Hydrogen	1	Gas
10	He	Helium	1	Gas
11	Rn	Radon	6	Gas
12	Br	Bromine	4	Flytande
13	Hg	Mercury	6	Flytande
14	Tl	Thallium	6	Fast
15	Pb	Lead	6	Fast
16	Bi	Bismuth	6	Fast
17	Po	Polonium	6	Fast
18	At	Astatine	6	Fast
19	Li	Lithium	2	Fast
20	Be	Beryllium	2	Fast
21	B	Boron	2	Fast

Övning - 4

- Table: Elements
- Lista alla grundämnen sorterat på om de är fast/flytande/gas I rumstemperatur (20 grader) genom att titta på melting och boilingpoint (som dock är angina I Kelvin-grader)

Övning - 5

	Id	ProductName	QuantityPerUnit	UnitsInStock	ReorderLevel	MissingUnits
1	31	Gorgonzola Telino	12 - 100 g pkgs	0	20	-20
2	32	Mascarpone Fabioli	24 - 200 g pkgs.	9	25	-16
3	66	Louisiana Hot Spiced Okra	24 - 8 oz jars	4	20	-16
4	70	Outback Lager	24 - 355 ml bottles	15	30	-15
5	37	Gravad lax	12 - 500 g pkgs.	11	25	-14
6	3	Aniseed Syrup	12 - 550 ml bottles	13	25	-12
7	48	Chocolade	10 pkgs.	15	25	-10
8	45	Rogede sild	1k pkg.	5	15	-10
9	68	Scottish Longbreads	10 boxes x 8 pieces	6	15	-9
10	56	Gnocchi di nonna Alice	24 - 250 g pkgs.	21	30	-9
11	43	Ipoh Coffee	16 - 500 g tins	17	25	-8
12	11	Queso Cabrales	1 kg pkg.	22	30	-8
13	2	Chang	24 - 12 oz bottles	17	25	-8
14	64	Wimmers gute Semmelknödel	20 bags x 4 pieces	22	30	-8
15	49	Maxilaku	24 - 50 g pkgs.	10	15	-5
16	30	Nord-Ost Matjeshering	10 - 200 g glasses	10	15	-5
17	21	Sir Rodney's Scones	24 pkgs. x 4 pieces	3	5	-2

- Lista tabellen company.products där produkterna I lager (UnitsInStock)är lägre än reorderLevel, sorterat på hur manga Units som saknas

Övning 6 – Uppgiftstafett SQL

- Gruppuppgift:
  - Botanisera I Everyloop-databasen och försök själv komma på en uppgift som du sedan skickar till nästa grupp. (G1 till G2…G6 till G1 samt G7 byter med G8)
  - Lös först uppgiften själv, innan du skickar den.
  - Efter ett tag hörs ni, för "redovisning".

Länkar

- SQL Tutorial (w3schools.com)
  - SQL Syntax (w3schools.com)
  - SQL SELECT Statement (w3schools.com)
  - SQL SELECT DISTINCT Statement (w3schools.com)
  - SQL WHERE Clause (w3schools.com)
  - SQL AND, OR, NOT Operators (w3schools.com)
  - SQL ORDER BY Keyword (w3schools.com)
  - SQL SELECT TOP, LIMIT, FETCH FIRST ROWS ONLY, ROWNUM (w3schools.com) (Visar skillnader mellan olika SQL-dialekter)
  - SQL LIKE Operator (w3schools.com)
  - SQL IN Operator (w3schools.com)
  - SQL BETWEEN Operator (w3schools.com)
  - SQL Aliases (w3schools.com)
  - SQL CASE Expression (w3schools.com)
  - SQL Operators (w3schools.com)
  - SQL Wildcard Characters (w3schools.com)

On-Fr

- Jobba med SQL-övningarna
- Gå mer nogrannt igenom de utvalda länkarna
- Utmana varandra på SQL-uppgifter
- Fråga om allt som känns oklart, även från förra kursen