

Fredag 9 Mars 2018



Databasteknik

20 YHP

# Vad är en databas?



- Databas i regel synonymt med ett *databassystem*:  
**Database management system** och **data som hör ihop**
- Vanligaste formen är en "relationsdatabas"

# Komponenter i ett DBMS:



- **Procedurer**

- Definierar användningen av databasen

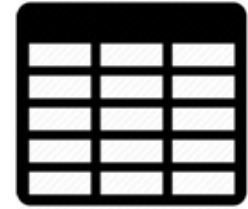
- **Vyer**

- Resultat av procedurer
- oftast en delmängd av datan i databasen

- **Schema**

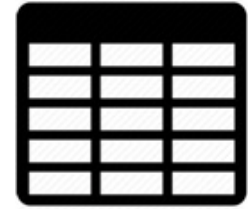
- En beskrivning av databasens struktur
- Man pratar ofta om hela databasens schema, men beroende på databasen kan det även vara relevant att prata om flera **scheman** i databasen.

# Vad är en relationsdatabas?



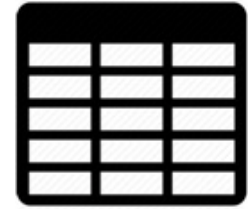
- Databas som fokuserar på relationen mellan sparad data.  
kända exempel: MSSQL och MySQL.
- Består av...
  - **Tabeller**
  - **Attribut (kolumner) med varsin domän**
  - **Rader (instanser)**
  - **Nycklar**

# Relationsdatabaser & DBMS



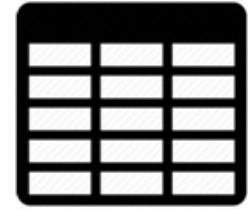
- Varje tabells attribut, domäner och nycklar definerar **tabellens schema**
- Alla tabellers scheman definerar **databasens schema**
- **Processer** och **vyer** varierar mellan relationsdatabaser, men består nästan alltid av SQL

# Nycklar i Relationsdatabaser



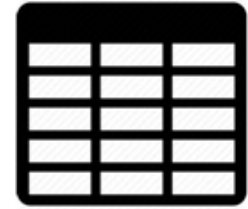
- Nycklar används för att hitta en viss rad i en tabell.
- En nyckel består av en eller flera kolumner
- En supernyckel är en nyckel som unikt identifierar endast EN rad.
- Varje tabell måste ha minst en supernyckel som kallas för "primärnyckeln"

# Nycklar i Relationsdatabaser



- Relationer defineras med hjälp av primärnycklen och främmande nycklar
- En främmande nyckel är en eller flera kolumner som tillsammans innehåller värden som är supernyckel i en annan tabell

# Regler i Relationsdatabaser



- Varje tabell har ett unikt namn
- Varje kolumn har ett unikt namn i den tabellen
- Varje cell innehåller exakt **1** värde
- Alla värden i en kolumn måste följa den kolumnens domän
- Varje rad är unik, **det finns inga dubletter**
- Ordningen på kolumner spelar ingen roll
- Ordningen av rader spelar ingen roll



# Vad är SQL?



- Structured Query Language
- Baserat på *relationsalgebra*
- Ett måttligt standardiserat språk 🙄 för att kommunicera med relationsdatabaser (DBMS) som har stöd för det.

# Vad är SQL?



- Är inte ett programmeringsspråk, utan ett språk som enbart kan ställa frågor till databasen (QUERIES)
- Det enda standardiserade databasspråket som blivit allmänt accepterat.
- Är en ISO standard, men dom flesta databaser använder sin egna dialekt

# Relationsdatabaser och SQL



- Kom ihåg att ett databassystem består av *scheman*, *vyer* och *procedurer*
- Vi har hittills gått igenom **schemat** för relationsdatabaser.
- Även enkla *vyer*, då alla tabeller är varsin **vy**.

# Relationsdatabaser och SQL



- SQL skrivs som enkla kommandon som körs efter varandra i den ordning dom skrivs.
- Input är i princip alltid en **vy** och en **operation** (procedur) och resultatet är alltid en ny **vy**
- På så sätt definerar vi med SQL hela databassystemet, scheman, vyer och procedurer.

# Relationsdatabaser och SQL



- Vad är en vy?
  - Oftast är en vy helt enkelt samma sak som en tabell
  - Men då en **tabell** alltid innehåller all data som är faktiskt sparad i databassystemet, så är vy ett bredare begrep...

# Relationsdatabaser och SQL



- Vad är en vy?
  - ...resultatet av en sökning
  - ...resultatet av ett filter
  - ...flera tabeller ihopslagna till en "virtuell tabell"

# Vad är SQL?



- **DDL** data definition language, är operationer som definierar hur datan ska se ut. Inom ramen för databasens schema (vi går igenom DDL senare)
- **DML** data manipulation language, är operationer för att hämta, lagra och uppdatera data.

# Vad är SQL?



- **DML operationer:**

- **SELECT** Används för att söka / välj ut / hämta data från en tabell
- **INSERT** Används för att lägga till data i en tabell
- **UPDATE** Används för att ändra på data i en tabell
- **DELETE** Används för att ta bort data från en tabell



# SQL Syntax



- **VERSALER** anger reserverade keywords i SQL
- **gemener** används för användardefinierade ord
- **|** används för att visa ett val mellan olika alternativ (eller)
- **{ }** indikerar en obligatorisk del
- **[ ]** indikerar valfri del
- **(...)** indikerar repetition av en del, noll till många gånger

# SQL Syntax



- ' ' strängar måste omges med enkla 'fnuttar'
- " " tabeller och attribut som innehåller mellanslag måste omges av "dubbelfnuttar"
- [ ] i MSSQL används klammrar på samma sätt som "citationstecken"
- ; avslutar ett SQL kommando.

# SQL SELECT



```
SELECT * FROM mytable;
```

# SQL SELECT



- Den första operationen vi lär oss
- Formell definition:

```
SELECT [DISTINCT | ALL] {*} | [kolumnUttryck [AS namn]]  
[, ...]}
```

```
FROM tabellNamn [AS alias | alias] [, ...]
```

```
[WHERE where-villkor]
```

```
[GROUP BY kolumnLista [HAVING having-villkor]] [ORDER  
BY kolumnLista [ASC | DESC]]
```

# SQL SELECT



- Den första operationen vi lär oss
- Förklaringar:
  - **SELECT** - Anger vilka kolumner som skall visas
  - **FROM** - Anger vilka tabeller som skall användas
  - **WHERE** – Filtrerar rader beroende på where-villkoret
  - **GROUP BY** – Grupperar rader med samma kolumnvärde
  - **HAVING** - Filtrerar grupper beroende på having-villkoret ! **ORDER BY**
    - Anger sorteringsordningen på raderna

# SQL SELECT



- SELECT körs i denna ordning:
  - **FROM** först väljs vilken tabell datan ska hämtas ifrån
  - **WHERE** sen filtreras raderna
  - **GROUP BY** raderna delas in i grupper
  - **HAVING** grupperna filtreras
  - **SELECT** kolumnerna/raderna fylls i med värden
  - **ORDERBY** raderna sorteras i rätt ordning

# SQL SELECT WHERE



- Om man inte anger en where så får man alla rader i tabellen man valt i FROM
- WHERE följs av en jämförelse
- Jämförelsen kan "vändas" om man använder WHERE NOT
- Flera jämförelser kan skrivas med AND och OR

# SQL SELECT WHERE



- AND och OR och NOT kan kombineras:
  - WHERE NOT ... OR ... AND ....
  - WHERE ... OR NOT ....



# SQL SELECT WHERE



- Ordningen bestäms med paranteser:
  - WHERE ( NOT ... OR ... ) AND ....
  - WHERE NOT ... OR ( ... AND .... )

# SQL SELECT WHERE



- Jämförelser skrivs ung. som ifsatser. i SQL är ett enkel likamed ( = ) alltid en check och inte an tilldeldning.

# SQL SELECT WHERE



- Möjliga jämförelser:
  - = Samma
  - <> Inte samma
  - > Större än
  - < Mindre än
  - >= Större än eller samma
  - <= Mindre än eller samma
- Det finns även mer avancerade jämförelser vi går igenom sen 😊

# Träna på SQL



**[sqlbolt.com](https://sqlbolt.com)**

Tack  
för  
idag.