**MÅNDAG: 23/10**

DDL  
DML

Båda är sequal, men ddl skapar struktur metodiskt och dml är det vi hantear med SELECT, INSERT..

**Nestlad fråga**: subfråga i frågan.

Om man kanske vill hämta från equals från en annan tabell = nestlad fråga. Sällan man jobbar men en tabell

**IN and NOT IN** – opposit

Where Namn IN (’Bowling’, ’Konstsim’) (alla som har bowling och konstsim)(kollar två värden i samma kolumn=

Where Sektionskod NOT IN (selectsektion from deltar)

**EXISST and NOT exsis**

Man kan vilja kolla om ngt finns. Deltar den här medlemmen i något – returnerar annars FALSe. Vill hämta en medlem men kolla om den verkligen deltar i ngt (exist – returnerar TRUE)

WHREE Deltar.Medlem = Medlem.Medlemnr) – Kollar bara Finns den här

Select \* FROM Medlem //hämtar själva datat

WHERE EXIST (SELECT \* FROM Deltar

WHERE Deltar.Medlem = Medlem.Medlemsnr) – LYSSNA //om man har satt index blir det snabbare

**SOME and ANY** är egenlitgen samma sak. Används för att jämför om ngt fall finns i subfrågan

SELECT \* FROM Medlem  
WHERE Medlemsnr > NY  
SELECT Medlemsnr FROM Medlem);

Ser inte some så ofta – använd bara any

hämtar alla som är större än medlemsnr om vi har 1, 2, 3, (1 och två har bpda ngt som är större). Returnerar bara det största numret. ???

**ALL anväds för att jämföra alla fall finns i sub-fråggan**

* Måste matcha allt
* Om man hämtar från en annan tabell så hade det fungerat
* Hårt – ALLA måste mappa det här

**JOINS – Finns två typer**

* **Inner join**
* **Outer join**

Outer:

* Left join
* Right join
* Full join

Inner join: Returnerar alla record som har en match i båda. Om total match så returnerar den från de båda vi joinat.

**Left join**: Returnerar allt i den tabellen plus matcharna, men då kan man få null värden. Kanske inte fanns ngt att referear till i den andra och då blir det null

**Right** **join:** är åt andra hållet:

**Full join:** får allt i båda tabellerna, men kan då få många null värden

**Viktit med tydliga quesries**

**SORTERING: ORDER BY desc**

Man kan specieficera att det ska specificera på namn. ORDER BY Namn DECS (kan tala om vilken väg)

Sorterar inte om ngt i tabellen, bara sökresultatet

**TOP:**

**Select TOP(10) –** Hur många jag vill ha 10. Hämtar första 10 resultaten. Ex om man vill kolla hur datan ser ut, eller vad man har för kolumner. De tio första resultatetn utifrån index. Om sorterar lista utifrån summa.

(CLUSTED INDEX: sortera om allt i själva tabellern, de andra index gör m

**Select TOP(10)Percent – hämtar top 10 procent av quesry. Select alla rader men returnera bara första tio procenten. Om hundra rader skulle dessa två ex returnerar samma. Men om 150 skulle procent returnera 15 o den övre TOP(10) tio stycken.**

**AGGREGATFUNKTIONER:**

Härledda attribut i ER-modellen. När detvar sträckade attribut som vi inte ville spara i databasen, men ville visa i modellen att detvar viktigt(men ändras hela tiden)

Behöver inte göra sina egna kalkylationer utan det finns redan

COUNT //räknar alla. Får tillbaka hur många rader det fanns

AVG medelvärde

MIN/MAX //selct MAX – högsta lönen – minst lön

SUM total summan av alla löner. Hur mkt betalr företag ut i lön per månan

**GROUP BY**

**SELECT Placering, AVG(Lön) FROM Arbetare  
GROUP BY Placering**

Gruperar man resultaten. Ex på hur många har den här lönen. Grupperas på hur många som har 15000.

Eller om det är GROUP BY Placering: de på Ica har den avergae, de på coop har..

Kan bryta ner till Ica Hökarhängen, Ica gamla stan.

**HAVING**

**SELECT Placering, AVG(Lön) FROM Arbetare  
GROUP BY Placering  
HAVING AVG(Lön) > 1000;**

Har ett condition på group by – en extension på group by

**VYER**

Om vet att ngt används hela tiden, alltid vill se och det alltid är samma fråga o samma resultat kan man skapa en vy – Virituell tabell

**Varför Vy?** För enkelheten. Ngt som görs ofta. Om gör uppdaterng till tabell, kanske man vill ha en vy så man inte tar sönder ngt. Så de tio applikationer man inte ändrats inte förstörs

Ger hög säkerthet: Du får bara se de här kolumnerna. Kan bestämma att just den personen bara får se viss infortmation

Resultatet som man får tillgång till

CREATE VIEEW AnställdaPerAvdelning  
AS

SELECT Efternamn, fornamn, namn  
 FROM Anställd – KOLLA EXEMPEL PW  
  
**KOLLA VAD SOM FINNS!!!**

Kan finnas kolumner i två tabeller som heter samma, o då vet den inte vilken den ska ta när man joinar

Select COUNT (UnitPrce) ger 77 st

Max (UnitPrice) 26300

Select COUNT(Unitprice), MAX(UP), AVG(UP), MIN(UP), SUM(UP)  
from Products

30 oktober

**Entity framwework**

* Kommer jobba orientobjekterat, men sen händer ”magi”, EF, hanterar det och översätter??
* Många olika sätt att jobba med det

**major major:**

* Modeling:
* Object layer

**Designval:**

* Hur stor
* Deployment cycle: Hur ska den se ut?
* Deployment?

Code first/Model first/Database first

**FYRA MAIN** grejer man kan göra – Empty Entity

-Högerklicka, får lista, add new entity

- scalar proppirty: Persnummer, Name

DbContext: Alltid lika oavsett (Code/Model/Database first)