

SQL-ESIMERKKILAUSEITA

yksinkertaisesta monimutkaiseen, esim. itseopiskelua varten

ASTERI LASKUTUS

Kaikkien laskuotsikoiden kaikki kentät:

```
SELECT * FROM Laskuotsikot
```

Vain valitut kentät:

```
SELECT Päivämäärä, Laskutusnimi, [ale mk], Loppusumma FROM Laskuotsikot
```

Annetaan kentille mukavammat nimet:

```
SELECT Päivämäärä, Laskutusnimi AS Asiakas, [ale mk] AS [Alennus euroa],  
Loppusumma FROM Laskuotsikot
```

Asiakaskohtaiset ostojen yhteissummat. Jos sama asiakas ollut eri nimillä, niin kukin nimi erikseen:

```
SELECT Asiakasnumero, Laskutusnimi AS Asiakas, Sum([Loppusumma]) AS [Ostanut euroa] FROM Laskuotsikot GROUP BY Asiakasnumero, Laskutusnimi
```

Asiakaskohtaiset ostojen yhteissummat. Jos sama asiakas ollut eri nimillä, niin kaikki nimet samaan summaan ja nimeksi aakkosissa viimeisin nimi:

```
SELECT Asiakasnumero, Max(Laskutusnimi) AS Asiakas, Sum([Loppusumma]) AS [Ostanut euroa] FROM Laskuotsikot GROUP BY Asiakasnumero
```

Yli 1000 eurolla ostaneet:

```
SELECT Asiakasnumero, Max(Laskutusnimi) AS Asiakas, Sum([Loppusumma]) AS [Ostanut euroa] FROM Laskuotsikot GROUP BY Asiakasnumero HAVING Sum([Loppusumma]) > 1000
```

Asiakaskortistosta kaikki asiakkaat:

```
SELECT * FROM Asiakkaat IN '%91'
```

..järjestettynä postinumeron mukaan:

```
SELECT * FROM Asiakkaat IN '%91' ORDER BY Paikka
```

..ne jotka asuvat Uudellamaalla

```
SELECT * FROM Asiakkaat IN '%91' WHERE Paikka < '2'
```

..ne jotka ovat kujalla

```
SELECT * FROM Asiakkaat IN '%91' WHERE Osoite LIKE '*kuja*'
```

..ne joiden nimi alkaa aakkosissa ennen K-kirjainta ja jotka sijaitsevat Uudellamaalla tai Turun seudulla

```
SELECT * FROM Asiakkaat IN '%91' WHERE Nimi < 'K' AND Paikka < '3'
```

..edellisten lisäksi Karhulassa sijaitsevat asiakkaat

HUOM 1: sulkujen käyttö!

HUOM 2: AND tarkoittaa, että vain ne otetaan, jotka täyttävät molemmat ehdot. OR tarkoittaa sekä ne, jotka täyttävät ensimmäisen ehdon että ne, jotka täyttävät toisen ehdon. Eli vähän eri asia kuin suomenkielen "ja" ja "tai"

```
SELECT * FROM Asiakkaat IN '%91' WHERE (Nimi < 'K' AND Paikka < '3') OR Paikka LIKE '*KARHULA*'
```

..ne joiden asiakasnumero on tietyllä välillä

```
SELECT * FROM Asiakkaat IN '%91' WHERE Numero BETWEEN '1003' AND '1006'
```

Laskut, joiden loppusumma on suurempi kuin 1111,11. Huom! desimaalierottimena piste!

```
SELECT Päivämäärä, Laskutusnimi, [ale mk], Loppusumma FROM Laskuotsikot WHERE Loppusumma > 1111.11
```

Asiakkaat järjestettynä Selite3-kentän mukaan, jos halutaan "aakkosjärjestys"

```
SELECT Nimi, Selite3 FROM Asiakkaat IN '%91' ORDER BY Selite3
```

Asiakkaat järjestettynä Selite3-kentän mukaan, jos halutaan "numerojärjestys"

```
SELECT Nimi, Selite3 FROM Asiakkaat IN '%91' ORDER BY Val(Selite3)
```

Laskut aikarajauksella

```
SELECT Päivämäärä, Laskutusnimi, Loppusumma FROM Laskuotsikot WHERE CDATE(Päivämäärä) BETWEEN DateValue('%1') AND DateValue('%2')
```

..ja rahamäärän muotoilulla kaksidesimaaliseksi

```
SELECT Päivämäärä, Laskutusnimi, Format(Loppusumma,'0.00') AS Summa FROM Laskuotsikot WHERE CDATE(Päivämäärä) BETWEEN DateValue('%1') AND DateValue('%2')
```

Rivit kahdesta eri talukosta UNION-kyselyllä

```
SELECT Nimi, Osoite, Paikka FROM Asiakkaat IN 'c:\winls\sqlloy\asiakas.wak'  
UNION  
SELECT Nimi, Osoite, Paikka FROM Asiakkaat IN 'c:\winti\malli\malli.wti'
```

Ristiintaulukointi

```
TRANSFORM Sum([netto mk]) SELECT Tuotenumero, Nimike  
FROM Tuoterivit  
WHERE CDate(Päivämäärä) BETWEEN DateValue('%1') AND DateValue('%2')  
GROUP BY Tuotenumero, Nimike  
ORDER BY Tuotenumero PIVOT DateValue(Päivämäärä)
```

NULL-pulmien esto IIF-ehdolauseella

```
SELECT * FROM Reskontra WHERE  
IIF(ISNULL([Viimeisin suorituspäivä]), FALSE,  
CDATE([Viimeisin suorituspäivä]) BETWEEN DateValue('%1') AND DateValue('%2'))
```

NULL-pulmien esto muuttamalla NULL-arvot tyhjiksi merkkijonoiksi käyttämällä " &

```
SELECT Nimi, Val(" & Lisätieto1) AS Lukuarvo FROM Asiakkaat IN '%91'
```

Liitos (JOIN)

```
SELECT Max(Laskuotsikot.Päivämäärä) AS Pvm, Max(Laskutusnimi) AS Asiakas,
Sum([brutto mk]) AS TuoterivitYhteensä FROM Laskuotsikot INNER JOIN Tuoterivit ON
Laskuotsikot.Laskuntunniste = Tuoterivit.Laskuntunniste GROUP BY
Laskuotsikot.Laskuntunniste
```

ASTERI ISÄNNÖINTI

Huoneistot:

```
SELECT * FROM Huoneistot
```

Asukkaat:

```
SELECT * FROM Asukkaat
```

Luettelo huoneistoista ja niiden asukkaista:

```
SELECT [Porras ja huoneisto], Nimi FROM Huoneistot INNER JOIN Asukkaat ON
Huoneistot.ID = Asukkaat.Huoneisto ORDER BY Huoneistot.ID
```

..mukana myös tyhjät huoneistot:

```
SELECT [Porras ja huoneisto], Nimi FROM Huoneistot LEFT JOIN Asukkaat ON
Huoneistot.ID = Asukkaat.Huoneisto ORDER BY Huoneistot.ID
```

Liitos kolmesta taulukosta ("nested join")

```
SELECT Max(Huoneistot.[Porras ja huoneisto]) AS Huoneisto, Max(Maksajat.Nimi) AS
Henkilö, Sum(Laskurivit.Summa) AS Laskutettu FROM Maksajat INNER JOIN (Laskurivit
INNER JOIN Huoneistot ON Laskurivit.Huoneisto = Huoneistot.ID) ON Maksajat.ID =
Laskurivit.Maksaja GROUP BY Laskurivit.Maksaja
```

Vuokraaikojen laskutetut ja suoritettut

Sisemmässä kyselyssä mm. UNION

Huomaa myös ulompi kysely

```
SELECT Vastike,
FORMAT(SUM(Tavoitteet),'0.00') AS Laskutettu,
FORMAT(SUM(Suoritukset),'0.00') AS Maksettu
FROM
(SELECT Laskurivit.Selite AS Vastike, SUM(Laskurivit.Summa) AS Tavoitteet,
0 AS Suoritukset
FROM Laskurivit
WHERE CDATE(Laskurivit.Eräpäivä) BETWEEN DateValue('%1') AND DateValue('%2')
GROUP BY Laskurivit.Selite
UNION
SELECT Suoritusrivit.Selite AS Vastike, 0 AS Tavoitteet,
SUM(Suoritusrivit.Suoritus) AS Suoritukset
FROM Suoritusrivit
WHERE CDATE(Suoritusrivit.Suorituspäivä)
BETWEEN DateValue('%1') AND DateValue('%2') GROUP BY Suoritusrivit.Selite)
GROUP BY Vastike
```