

# Räätälöidyt raportit

## Sisällysluettelo

- [Räätälöidyt raportit](#)
- [SQL-lauseiden ketjuttaminen](#)

## Räätälöidyt raportit

SKJ Pääohjelma Raportit Räätälöidyt raportit: voidaan tehdä asiakaskohtaisia raportteja.

Voit tuoda valmiita raportteja sisään ohjelmaan SKJ Pääohjelma Tiedosto Tuo raporttipohja -toiminnolla.

Raportteja voidaan ajaa myös yöajassa ajastetuilla raporteilla ([Raporttikon](#)). Raporttikonossa voi käyttää raportin parametreja tyyliä:

```
parametri:tyyppi=arvo
```

Näytön alalaidassa on kolme SQL-välilehteä, joista jokainen vastaa yhtä tietolähdettä (datapipeline) raportilla. Lisäksi on olemassa yksi tieto -niminen tietolähde, jota voidaan käyttää apuna, kun halutaan tulostaa vain yksi sivu/asia. Esimerkiksi Crosstab datapipeline1:n tiedoista.

SQL on normaalia Pervasiven SQL:a. Raportin rajausarvot voidaan saada mukaan parametreilla esimerkiksi:

```
select sum(summa) from tuotemyy where pvm between :spvmalku and :spvmloppu
```

(tuotemyy:n pvm on yyyyymmdd muotoinen merkkijono) tai

```
select sum(summa) from mt where tilauspaiva between :pvmalku and :pvmloppu
```

(mt:n tilauspaiva on pervasiven date kenttä yyyy-mm-dd muotoinen).

Jos raportille haluaa tulostaa nuo päivät, pitää raportille luoda Label ja Calc-välilehdelle laittaa ko. Labelin OnGetText:iin seuraava koodi:

```
procedure Label12OnGetText(var Text: String);  
begin  
Text := Report.Parameters['paivaalku']  
  
end;
```

Päivä loppu on vastaavasti

```
Report.Parameters['paivaloppu']
```

Lisäksi voidaan laittaa muita parametrejä esim. asiakasryhmä.

```
select nro, nimi, sum(summa) from mt,asiakas  
where asiakas.asiakas=mt.toimitusasiakas and asiakas.ryhma =:asryhma group by nro,nimi
```

Painamalla päivityä parametrit :asryhma ilmestyy ylös ja siihen voidaan ilmoittaa ryhmän numero.

Huom. parametrissa ei saa olla skandeja. :asryhmä ei ole siis kelvollinen.

## SQL-lauseiden ketjuttaminen

Usein on tarpeen tehdä laskentaa ennen raportin tulostusta, esimerkiksi kerätä asiakkaiden myynti aputauluun useasta kannasta.

Tätä varten on lisätty --EIVIRHE kommentti, jolla voidaan seuraavasta SQL-lauseesta tuleva virhe ohittaa.

Esimerkkinä alla olevat sql-lauseet keräävät saldosumma-tauluun neljästä kannasta tiettyjen asiakasryhmien tilisaldotapahtumat ja raportoi ne yhteen. Tässä taulun tyhjentäminen ja luomisen virheet on ohitettu, koska taulun uudelleenluominen tuottaisi virheen.

Lauseiden erottimena kannattaa käyttää puolipistettä. Ja huom. puolipistettä EI SAA laittaa heti edellisen sanan perään, ohjelma ei tunnista sitä.

```
--EIVIRHE
DELETE FROM SALDOSUMMA ;
--EIVIRHE
CREATE TABLE SALDOSUMMA (kanta varchar(20),asiakas integer ,summa float,primary key(kanta,asiakas)) ;

insert into SALDOSUMMA (select 'KUOPIO',a.asiakas,round(sum(if(tyyppi=0,-suoritus,suoritus)),2) from
KUOPIO.salsuor s, lielhti.asiakas a where s.asiakas = a.asiakas and
tyyppi in (0,1) and a.ryhma in (1,2,3,4,8,34) and pvm between :pvmalku and :pvmloppu
group by a.asiakas ) ;

insert into SALDOSUMMA (select 'TAMPERE',a.asiakas,round(sum(if(tyyppi=0,-suoritus,suoritus)),2) from
TAMPERE.salsuor s, elovainio.asiakas a where s.asiakas = a.asiakas and
tyyppi in (0,1) and a.ryhma in (1,2,3,4,8,34) and pvm between :pvmalku and :pvmloppu
group by a.asiakas ) ;

insert into saldosumma (select 'HELSINKI',a.asiakas,round(sum(if(tyyppi=0,-suoritus,suoritus)),2) from
HELSINKI.salsuor s, kulju.asiakas a where s.asiakas = a.asiakas and
tyyppi in (0,1) and a.ryhma in (1,2,3,4,8,34) and pvm between :pvmalku and :pvmloppu
group by a.asiakas ) ;

insert into saldosumma (select 'JKYLA',a.asiakas,round(sum(if(tyyppi=0,-suoritus,suoritus)),2) from
JKYLA.salsuor s, NOKIA.asiakas a where s.asiakas = a.asiakas and
tyyppi in (0,1) and a.ryhma in (1,2,3,4,8,34) and pvm between :pvmalku and :pvmloppu
group by a.asiakas ) ;

Select asiakas.asiakas asnro, nimi, sum(summa) as saldo
from asiakas left outer join saldosumma on asiakas.asiakas= saldosumma.asiakas where
ryhma in (1,2,3,4,8,34)
group by asnro,nimi
```