

# Summary Files Using SQL - A Technical Tip

submitted by [Steve Kontos](#).

There is a nice way to create summary files, on-the-fly, with SQL.

Recently, a customer asked to dynamically see their inventory position by company, division, and item class. The customer had dozens of divisions and classes, and 40,000 or so item balance (inventory) records.

Here's how I did it. First, I created a physical file, IOINSMWF, using DDS:

```
0.....1.....2.....3.....4.....5.....6.....7.....8
1234567890123456789012345678901234567890123456789012345678901234567890
A          R IOINSMWFR
A          WFCMP      R          REFFLD(IFCOMP ITEMBAL)
A          WFDIV      R          REFFLD(IFDIV  ITEMBAL)
A          WFCLS      R          REFFLD(IFCLS  ITEMBAL)
A          ONHAND          15  2
A          ONORDER         15  2
A          COMMITTED       15  2
A          AVAILABLE       15  2

A          K WFCMP
A          K WFDIV
A          K WFCLS
```

Then, I created an RPGLE program to run the following SQL statement:

```
0.....1.....2.....3.....4.....5.....6.....7.....8
1234567890123456789012345678901234567890123456789012345678901234567890

c/exec sql
c+      Insert Into QTEMP/IOINSMWF
c+      select ifcomp, ifdiv, ifcls,
c+          sum(ifqoh), sum(ifqpo), sum(ifqcm),
c+          ((sum(ifqoh) + sum(ifqpo)) - sum(ifqcm))
c+      from ITEMBAL
c+      group by ifcomp, ifdiv, ifcls
c/end-exec
```

On an iSeries 820, this summary file is built quickly (about two to three seconds) and is displayed in a subfile, as the customer requested. Whatever you think about two to three second response-time, it is certainly much faster than an RPGLE program could have read and summarized the entire file.

Even faster, (if you are using V5R3) is creating a table, via a SELECT statement, as shown below:

```
0.....1.....2.....3.....4.....5.....6.....7.....8
1234567890123456789012345678901234567890123456789012345678901234567890

* Delete the work file
c/EXEC SQL
c+      Drop Table QTEMP/IOINSMWF
c/END-EXEC

* Create work file
c          If          SQLCODE = *Zeros or SQLCODE = -204

c/EXEC SQL
c+      Create Table QTEMP/IOINSMWF
c+      as (select ifcomp, ifdiv, ifcls, sum(ifqoh) as onhand,
```

```
c+          sum(ifqpo) as onorder, sum(ifqcm) as committed
c+          from ITEMBAL
c+          group by ifcomp, ifdiv, ifcls
c+          order by ifcomp, ifdiv, ifcls)
c+ with data
c/END-EXEC
```

```
c          endif
```

Creating a table via an SQL SELECT statement is a pretty powerful thing.

The reference for this technical tip is the IBM Redbook, "Modernizing IBM eServer iSeries Application Data Access - A Roadmap Cornerstone." There is a nice chapter (Chapter 7) on embedded SQL.

The link to this Redbook is: <http://www.redbooks.ibm.com/abstracts/sg246393.html>

Enjoy!

Steve Kontos