



Tips / Tricks: Programming - Full Table Scans

Have you ever added a simple select statement to a report script but then noticed that the report takes much longer to run than it did previously? You may have run into the SQL trap of causing a Full Table Scan.

Full Table Scans (FTS) will occur when the SQL select does not use an index. For example, suppose you have a report that prints the project manager's employee number (pman.f), but you need the report to print the project manager's name. Knowing that the employee name is stored in the Employee Master (tccom001), you try to add tccom001.nama as an input field to the report. You then add the field to the report layout, but it displays blanks, causing you to resort to adding a select statement to the before.layout section of the report script.

```
.....
before.layout:
  select      tccom001.nama
  from        tccom001
  where       tccom001.emno = :pman.f
  selectdo
  endselect
```

The project manager's name now prints on the report, but the select statement is potentially causing a Full Table Scan. Because the select statement is trying to do a field compare, the database may check every record in the employee table (tccom001) to find the one that match the project manager field (pman.f). You may know that there is only one record, but based on the select statement, the database doesn't know that. If the select statement had been written as follows:

```
.....
before.layout:
  select      tccom001.nama
  from        tccom001
  where       tccom001._index1 = {:pman.f}
  selectdo
  endselect
```

the database would quickly read one record. Using indexes in the where clause will allow the SQL to take advantage of database index scanning. Don't be intimidated by the index, they are a table's best friend. In this example, the time for the first select compared to the second, may not be all that significant, but what if you were trying to find a particular set of records in some other large table in Baan (for example, sales order line history or integration transactions)?

You can also include a "hint" in the SQL statement to instruct the database driver to return the exact number of records you need. Syntax for hint is "as set with n rows", where 'n' stands for the number of records you need. The above example can be rewritten as follows to return only 1 row:

```
before.layout:
  select tccom001.nama
  from tccom001
  where tccom001._index1 = {pman.f}
  as set with 1 rows
  selectdo
  endselect
```

Indexes can have multiple parts. Let's make the example a bit more complicated. You want to select and sum the amount field on the Finalized Transactions table (tfgld106) for records that have a particular ledger account, transaction type, and fiscal year. You could write the following select statement:

```
select      sum(tfgld106.amnt):trx.amnt
from        tfgld106
where       tfgld106.leac = :hold.leac
and         tfgld106.otyp = :hold.otyp
and         tfgld106.oyer = 2010
selectdo
endselect
```

SQL will need to check every record in the table, summing only those records where the invoice number and item number equal the values you're interested in. You could save a significant amount of time by using the following statement.

```
select      sum(tfgld106.amnt):trx.amnt
from        tfgld106
where       tfgld106._index3 = {:hold.leac,
                                :hold.otyp}
and         tfgld106.oyer = 2010
selectdo
endselect
```

Now SQL will retrieve a small subset of records where the index equals the ledger account and transaction type you're interested in; then check only that small subset for records where the fiscal year is equal to what you're looking for. The results are the same, as far as the value of `trx.amnt`, but the second select will be much faster; how much will depend on the number of records in `tfgld106`.

Every table created in Baan must have at least one index, and the first index defined for the table must be unique. The indexes defined for a table can be found in Tools, under Application Customization, or Application Development, under the Domains and Tables submenu. Go to the session for Table Definitions. Under the specific pull-down menu there will be an option for indices (in Baan IV there is a button).