

SQL Tips

(Something for Everyone!)

Ted Holt

Senior Software Developer
Profound Logic Software
Starkville, Mississippi



Senior Technical Editor
Four Hundred Guru
www.itjungle.com



1: Sort by Column Numbers

You can use column numbers to indicate sort columns in the ORDER BY clause. This is especially helpful when sorting on calculated columns.

```
select account,  
       ((curyear - prevyear) * 0.5)  
from saleshist order by 2, 1
```

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/db2/rbafzmstintsel.htm>

2: Non-alphabetic Sorting Technique 1

Suppose you want data about Tennessee to sort ahead of all other states. Use a CASE construct in the ORDER BY clause.

```
select * from sales
order by
    case when state = 'TN'
        then 0 else 1 end,
    state
```

<http://www.itjungle.com/fhg/fhg100604-story02.html>

3: Non-alphabetic Sorting Technique 2

The following ORDER BY clause causes data for the Shipping department to sort ahead of data for the Receiving department.

```
select * from trans
order by locate(dept,
'ACCOUNTING SHIPPING RECEIVING')
```

<http://www.itjungle.com/fhg/fhg091306-story02.html>

4: Ignore Case Technique 1

Use national language support to ignore case in sorts and comparison.

```
exec sql set option srtseq=*langidshr
```

<http://www.itjungle.com/mgo/mgo111403-story01.html>

5: Ignore Case Technique 2

Upper and lower functions are easier to use than NLS features, but may not produce desired results.

```
SELECT * FROM SESSION.xyz  
ORDER BY UPPER(1stnam)
```

6: Access in Arrival Sequence

Use the RRN function to access a file by relative record number.

```
SELECT * FROM sometable AS s  
WHERE rrn(s) BETWEEN 3 AND 5
```

<http://www.itjungle.com/fhg/fhg072705-story02.html>

<http://www.itjungle.com/fhg/fhg022206-story02.html>

7: Access Multi-member Files

SQL does not support multi-member database files. To access a member other than the first, create an alias.

```
create alias mylib/pay2014  
for mylib/payhist(year2014)
```

```
select * from mylib/pay2014
```

<http://www.itjungle.com/guruo/mgo021302-story02.html>

8: Create or Replace

Use CREATE OR REPLACE to avoid having to drop an object before recreating it.

```
create or replace view Names as  
select Name from people
```

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/sqlp/rbafycrtrepl.htm

9: Create or Replace Table

If your release of SQL does not support CREATE OR REPLACE TABLE, you can fake it.

```
begin
    declare continue handler
                for sqlexception
                begin end;
    drop table customers;
end;
create table customers . . .
```

<http://www.itjungle.com/fhg/fhg012015-story01.html>

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_71/sqlp/rbafyreplacetable.htm

10: Clear a Table

Finally! SQL can clear a physical file.

```
truncate table daily
```

is equivalent to

```
CLRPFM DAILY
```

<http://www.itjungle.com/fhg/fhg062514-story01.html>

11: Create Temporary Tables

Use the DECLARE GLOBAL TEMPORARY TABLE command to create scratch tables in the QTEMP library.

```
DECLARE GLOBAL TEMPORARY TABLE TEMP1 AS
    (SELECT * FROM B
     WHERE SUBSTR (TRANSID,1,1) = 'T')
WITH DATA
```

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/d/b2/rbafzmstdggt.htm>

12: Keep Temporary Data

The system clears temporary tables on commit unless you tell it not to.

```
exec sql
  declare global temporary table Summary
    (state char(2), SumAmount dec(9,2))
  on commit preserve rows
```

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/db2/rbafzdgtt.htm

13: Avoid Temporary Tables

If possible, use a view or common table expression instead of a global temporary table.

```
declare global temporary table STotal
as (select state, sum(baldue) as Due
    from qiws/qcustcdt
    group by state)
with data with replace
```

<http://www.itjungle.com/fhg/fhg060215-story02.html>

The same query with a common table expression.

```
with Stotal as
(select state, sum(baldue) as Due
 from qiws/qcustcdt
 group by state)
```

```
select a.*, b.*
from qiws/qcustcdt as a
join Stotal as b
 on a.state = b.state
```

14: Insert Multiple Rows

Insert more than one row at a time using the VALUES clause of an INSERT statement. Separate rows with commas.

```
INSERT INTO PLANTS VALUES  
(92, 'Lost Angeles'), (200, 'New Yo1k')
```

<http://www.itjungle.com/fhg/fhg010406-story02.html>

15: Update one Table from Another

Use the MERGE statement to update one table with data in another table.

```
merge into customer as c
  using (select oldregion, newregion
        from fixes) as f
  on c.region = f.oldregion
 when matched then
  update set c.region = f.newregion
```

<http://www.itjungle.com/fhg/fhg040313-story02.html>

16: Use MERGE to “Upsert”

Update data where it exists, add it where it doesn't.

```
merge into people as tgt
using (select name, street
      from updates) as src
  on src.name = tgt.name
when matched then
update set tgt.street = src.street
when not matched then
insert values (src.name, src.street1
```

17:Fill with One Character

It is possible to fill a value with a non-blank character or sequence of characters.

```
insert into somefile (somefield)
values(repeat('*',128))
```

<http://www.itjungle.com/fhg/fhg020205-story01.html>

Here's the "old" way.

```
insert into somefile (somefield)
  values (replace(space(128), ' ', '*'))
```

18: Row Value Expressions

Simplify row selection by comparing rows of values, rather than single values, to one another.

```
select * From Project Where  
(PROJNO, ACTNO) = ('AD3100', '10')
```

<http://www.itjungle.com/fhg/fhg041206-story01.html>

Row value expressions are also great for joins.

```
SELECT *
  FROM cacmst a
 left outer join cinvp b
    on (a.cmlocn,a.cmcont) = (b.silocn,b.sicont)
 left outer join trudtp c
    on (b.silocn,b.sicont,b.siitem,b.siseqn,b.silino) =
       (c.tulocn,c.tucont,c.tuitem,c.tuseqn,c.tulino)
ORDER BY a.cmlocn, a.cmcont
```

<http://www.itjungle.com/fhg/fhg100913-story02.html>

19: Compare Alpha and Numeric Data

As of V5R3, data types do not have to match.

```
select * from SomeFile  
  where CharField > NumField
```

<http://www.itjungle.com/fhg/fhg051706-story02.html>

20: Null Does Not Equal Null

(and null does not not equal null)

If you want to compare null to null, use IS [NOT] DISTINCT FROM.

```
SELECT * FROM SomeTable
WHERE SomeField IS NOT DISTINCT FROM
AnotherField
```

<http://www.itjungle.com/fhg/fhg091405-story01.html>

21: Wildcard Search

If you need to search for one of the pre-defined wild card characters, use ESCAPE.

```
SELECT * FROM SomeTable  
WHERE DataField LIKE '%10\%%' ESCAPE '\'
```

<http://www.itjungle.com/fhg/fhg052307-story02.html>

22: Wildcards in RPG

The %SCAN BIF does not support wild cards, but SQL does.

```
dcl-s   Matched      char(1) ;
```

```
exec sql
```

```
    set :Matched = case
```

```
        when :inSource like :inPattern
```

```
            then '1' else '0' end;
```

<http://www.itjungle.com/fhg/fhg120314-story01.html>

<http://www.itjungle.com/fhg/fhg011205-story02.html>

If RPG doesn't have a BIF you need, use an SQL function.

```
exec sql
```

```
values hex(:Class) into :Token
```

<http://www.itjungle.com/mgo/mgo080803-story02.html>

<http://www.itjungle.com/fhg/fhg012804-story04.html>

23: Wildcards on Steroids

Regular expressions allow more specific searches than wildcards do.

```
select * from people as p
where regexp_like(p.street,
                  '^1\d{2} .*Main')
```

<http://http://www.itjungle.com/fhg/fhg051915-story01.html>

24: Remove Extra Spaces in a String

Replace two or more blanks with only one.

(E.g., "Joe Smith" becomes "Joe Smith".)

```
update mydata set name =  
  replace(  
    replace(  
      replace(name,  
        ' ', '<>'), '><', ''), '<>', ' ')
```

<http://mullinsconsulting.com/bp11.htm>

<http://www.itjungle.com/fhg/fhg101106-story02.html>

25: Easily Process Non-date Dates

Use the FMTCDATE function to convert numeric or alpha fields to manageable formats.

```
select fmtdate (DueDate, 'cymd', 'iso-'),  
        . . .  
from MfgOrders
```

<http://www.itjungle.com/fhg/fhg050515-story02.html>

<http://www.itjungle.com/fhg/fhg120209-story02.html>

<http://www.itjungle.com/fhg/fhg081909-story02.html>

Instead of this:

```
substr( digits(dec(19000000+crdate,8,0)), 5, 2 ) || '/' ||  
substr( digits(dec(19000000+crdate,8,0)), 7, 2 ) || '/' ||  
substr( digits(dec(19000000+crdate,8,0)), 1, 4 )
```

Do this:

```
fmtdate(CRDATE, 'CYMD', 'MDYY/')
```

26: Unusual Rounding

You don't have to round to the nearest power of ten. You can round to other whole numbers, or even to fractions.

```
select dec(  
    round(price * 4, 0) / 4, 11, 2)  
as Nearest_Qtr      from PriceBook
```

<http://www.itjungle.com/fhg/fhg092105-story02.html>

Multiply by the reciprocal of the number to which you are rounding.
Divide by the reciprocal.

In this example, multiply by four to round to the nearest one-fourth. Then divide by four.

Example:

Price	Rounded
=====	=====
1.00	1.00
1.12	1.00
1.13	1.25
1.25	1.25
1.49	1.50
1.51	1.50

27: Beware of Integer Division

If both operands of an arithmetic operation are integers, the system yields an integer result. Use the `DOUBLE` function to convert at least one of the operands to floating point.

```
select double(Shipments) /  
       double(Orders) as Ratio  
from Summary
```

28: SELECT * Is OK

Contrary to popular opinion, it's OK to use SELECT * in a program that contains embedded SQL.

```
D InputData   e ds  extname ('QCUSTCDT')
```

```
exec sql declare c1 cursor for  
        select * from qcustcdt
```

http://www.code400.com/forum/showthread.php/13061-SQLRPGLE-using-SELECT-*

29: Avoid Null Indicators

Using COALESCE with fields from secondary files in outer joins avoids the nuisance of null indicators.

```
select c.cusnum, c.lstnam, c.city,  
       c.state, coalesce(s.name, ' ')  
from qcustcdt as c  
left join states as s  
      on c.state = s.abbr
```

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_71/db2/rbafzscacoales.htm

30: How Many Rows Did I Fetch?

Use the SQLER3 or SQLERR(3) field to see how many rows were affected by the previous I/O operation (i.e., fetch, insert, update, delete).

```
exec sql delete from PriceBook
      where PrcFlag = 'X';
if SQLER3 > *zero;
```

<http://www.itjungle.com/fhg/fhg020404-story02.html>

31: Optimization Tip 1

If you only want one row, or a few rows, tell the computer.

```
select * from qcustcdt  
order by lstnam,init  
fetch first 5 rows only
```

<http://www.itjungle.com/fhg/fhg022305-story02.html>

32: Optimization Tip 2

When fetching rows in groups, use OPTIMIZE FOR n ROWS.

```
exec sql
  declare c1 cursor for
    select * from qcustcdt
    optimize for 3 rows;
```

<http://www.mcpressonline.com/database/db2/get-your-system-humming-7-great-tips-for-tuning-db2-for-i.html>

33: Update a Fetched Row

To update a row that has been fetched through a cursor, do two things:

- **Add the FOR UPDATE OF clause to the cursor declaration.**
- **Use the phrase WHERE CURRENT OF x, where x is the cursor name, in the UPDATE command.**

<http://www.itjungle.com/fhg/fhg062304-story03.html>

```
exec sql  declare EmpCursor cursor for
          select * from Employee
          for update of Salary;

exec sql  open EmpCusor;

dow whatever;

    exec sql  fetch EmpCursor into :EmpRec;

    exec sql  update Employee set Salary = :Salary
              where current of EmpCursor;

enddo;

exec sql  close EmpCursor;
```

34: Update a Row

If you're updating all the columns in row, you don't have to list the fields. Use the special word row.

```
update inventory  
    set row = ('B', 7, 8, 9)  
where key = 12345
```

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/index.jsp?topic=/sql/rbafyupdatediftbl.htm>

<http://www.itjungle.com/fhg/fhg031208-story02.html>

35: SELECT * Is OK

Contrary to popular opinion, it's OK to use SELECT * in a program that contains embedded SQL.

```
dcl-ds InputData extname ('QCUSTCDT')  
        qualified;  
end-ds;
```

```
exec sql declare c1 cursor for  
        select * from qcustcdt
```

```
ctl-opt  actgrp(*new);

dcl-ds InputData  extname('QCUSTCDT') qualified;
end-ds;

*inlr = *on;

exec sql
  declare c1 cursor for
    select * from qcustcdt;

exec sql open c1;

dow '1';
  exec sql fetch c1 into :InputData;
  if sqlstate = '02000';
    leave;
  endif;
enddo;

exec sql close c1;

return;
```

36: Details and Summaries in One Query

Use OLAP functions to include both details and summary amounts in one query.

```
select d.ItemNumber,  
       sum(d.Quantity) as Qty,  
       sum(d.Quantity * d.Price) as Extended  
from InvoiceLines as d  
group by d.ItemNumber  
with rollup  
order by 1
```

37: Conditions in Query/400

Use an SQL view for conditional Query/400 processing.

```
CREATE VIEW myschema.myview AS
(SELECT c.*,
      CASE WHEN c.baldue >= 500
           THEN '*' ELSE ' '
      END AS flag
FROM qiws.qcustcdt AS c)
```

<http://www-03.ibm.com/systems/i/software/db2/qry4case.html>

38: Load Data into IFS Files

Use Qshell's db2 utility to place data into an IFS file.

```
db2 "SELECT char(CUSNUM) || ',' || LSTNAM ||  
' ,' || char(baldue) from qiws.qcustcdt"  
| sed -n '/,/p' >> custdata.csv
```

<http://www.itjungle.com/fhg/fhg020205-story02.html>

39: Verify Once

SQL-defined files generally perform better than DDS-defined files. Here's one reason why.

- The system validates DDS-defined data when it reads.
- The system validates SQL-defined data when it writes.

40: Generate DDL

Don't convert physical files to DDL by hand!

- System i Navigator
- QSQGNDDL API
- QSYS2.GENERATE_SQL stored procedure
- Third-party utilities

<http://www.itjungle.com/mgo/mgo060502-story01.html>

<http://www.itjungle.com/mgo/mgo061202-story02.html>

<http://www.itjungle.com/fhg/fhg111214-printer01.html>

http://www.epi-software.com/convert_dds_to_sql.html

41: Avoid Goofy Field Names

DB2/400 creates ugly 10-character names for columns with long names.

```
create table SomeTable
  (ID integer, Name varchar(20), PreferredName
    for column
      prefname varchar(20))
```

<http://www.itjungle.com/fhg/fhg020608-story02.html>

42: Specify Record Format Name

RPG programs don't compile if file and record name are the same. (Definition in F specs only.)

```
create table mytable  
  (name char (20),  
   number dec (5))  
rcdfmt myrec
```

<http://www.itjungle.com/mgo/mgo062802-story02.html>

43: WHERE vs. HAVING

WHERE and HAVING select data. What's the difference?

- Use WHERE to select individual rows.
- Use HAVING to select summaries or representatives of rows.

<http://www.itjungle.com/fhg/fhg050907-story01.html>

44:Use an SQL Formatter

An SQL formatter can make your code easier to read.

- Set case of keywords, column names, functions, variables etc.
- Align similar parts of a statement.

<http://www.dpriver.com/pp/sqlformat.htm>

(Google “sql formatter” for more.)

45: SQLFiddle.com

SQL Fiddle allows you to test queries against different DBMS engines, and to share your questions with others.

<http://sqlfiddle.com/about.html>

An Inspirational Thought

from Ashley Phix



28.35 grams of
prevention is
worth 0.4536
kilograms of
cure.

46: Rule of Thumb 1

If you use GROUP BY, you probably want ORDER BY, too.

```
SELECT state, COUNT(*), SUM(baldue)
FROM qiws.qcustcdt
GROUP by state
ORDER BY state
```

<http://www.itjungle.com/guruo/guruo112101.html>

47: Rule of Thumb 2

Use a left outer join unless you have a reason to use some other type.

```
SELECT a.cusnum, a.lstnam, a.init,  
       a.city, b.statename  
FROM qcustcdt AS a  
LEFT JOIN states AS b  
ON a.state = b.abbr
```

48: Rule of Thumb 3

Do **not** put parentheses around the LIKE expression in CREATE TABLE unless you have a reason to do so. Including parentheses makes the system drop certain attributes.

```
create table someschema/v2  
→ (like someschema/vendors) ←
```

<http://www.itjungle.com/fhg/fhg101310-story02.html>

49: Rule of Thumb 4

Use UNION ALL, not UNION, by default.

```
SELECT Customer, Item, ShipQty
FROM CurrYear
UNION ALL
SELECT Customer, Item, ShipQty
FROM PrevYear
```

<http://www.itjungle.com/fhg/fhg092607-story01.html>

50: Rule of Thumb 5

Use correlation names, even when they're not required. Precede them with the optional word AS.

```
select c.cusnum, c.lstnam, c.city,  
       c.state, coalesce(s.name, ' ')  
from qcustcdt as c  
left join states as s  
on c.state = s.abbr;
```

- Improves readability.
- Helps avoid common errors.

<http://www.itjungle.com/fhg/fhg102407-story01.html>

<http://www.itjungle.com/fhg/fhg031208-story02.html>

<http://www.itjungle.com/fhg/fhg091515-story02.html>

```
select c.cusnum, c.lstnam, c.city,  
       c.state, coalesce(s.name, ' ')  
from qcustcdt as c  
left join states as s  
on c.state = s.abbr;
```

c = correlation name of qcustcdt

s = correlation name of state

