Stringbearbeitung in SAS-SQL

20 May, 2008 - 19:48 — marcosch

Hallo,

ich habe hier ein DB2-SQL welches mit vielen CASE, LEFT, RIGHT und SUBSTR eine Ziffernfolge aus einem Feld extrahiert. Leider funktioniert das nur schlecht und sehr lückenhaft. Das müsste doch mit einem SAS-SQL bestimmt einfacher gehen, aber ich finde keinen rechten Ansatz.

Kurzes Ist/Soll-Beispiel:

```
Ist                       -> Soll
HV 1234567-7              -> 001234567
HV-HA 123456.7-tecis 792  -> 000123456
HV SV1234567.8 - 672      -> 001234567
1234567                   -> 001234567
HV-HA 12345678.8-54409    -> 012345678
SV 12345/0083-703-6643    -> 000012345
1234567890123450          -> 123456789
```

Vereinfacht gesagt will ich die erste vorkommende Zahl in einem variablen String ermitteln und am liebsten ggf. gleich auf 9 Stellen vornullen. Und das ganze im SAS-SQL

Ich bekomme keinen brauchbaren Ansatz hin, außer wieder Stelle für Stelle abzufragen, ab schon eine Zahl vorliegt, oder ein Punkt oder beides. Reguläre Ausdrücke könnten vielleicht gehen, hab ich aber keine Erfahrung mit :-(
Kann sich jemand erbarmen und mir einen Fingerzeig geben? :-) Danke im voraus!

Gruß aus Hamburg
Marco Schmidt

Foren:
ETL & Base SAS

Log in or register to post comments

Stringbearbeitung mit SAS_Base

21 May, 2008 - 09:21 — HansKneilmann

Hallo,
warum mit SAS-SQL, es gibt doch auch wunderschöne Möglichkeiten mit SAS-Base:

```
data aaa;
ist="HV 1234567-7"; output;
ist="HV-HA 123456.7-tecis 792"; output;
ist="HV SV1234567.8 - 672"; output;
ist="1234567"; output;
ist="HV-HA 12345678.8-54409"; output;
ist="SV 12345/0083-703-6643"; output;
ist="1234567890123450"; output;
run;

data bbb;
  set aaa;
  length xx $20;
  put _N_=;
  put ist=;
  AnzVar=length(ist)-length(compress(ist,' .<(+&!$*);^-/,%|'))+1;
  put AnzVar=;
  do ii = 1 to AnzVar;
    NumTreffer='N';          /* Init vom Ja/Nein-Flag */
    xx=scan(ist,ii);
    put ii= xx=;
    if compress(translate(xx,' ','0123456789')) eq ''
    then do;            /* xx ist REIN und vollständig NUM */
      NumTreffer='J';
    end;
    else if length(xx) ne length(compress(translate(xx,' ','0123456789')
    then do;        /* Feld mit NUM-Anteil */
      /* CHAR-Zeichen löschen */
      xx=compress(translate(xx,' ',upcase('abcdefghijklmnopqrstuvwxyz'))
      NumTreffer='J';
    end;

    if NumTreffer eq 'J' then do;
      yy=put(input(xx, 20.), 20.);
      if length(compress(yy)) le 9
      then do;
        yy=put(input(xx, 20.), z9.);
        put ii= yy= " NUM-Treffer, kurz";
      end;
      else do;
        yy=substr(xx, 1, 9);
        put ii= yy= " NUM-Treffer, lang";
      end;
      ii=AnzVar;     /* Abbruch der Schleife */
    end;
  end;
run;
```

Ist vielleicht ein bischen umständlich, aber es klappt ...

Gruß

Hans Kneilmann, Schäfer Shop GmbH (SSI)

Log in or register to post comments

Klappt, aber...

21 May, 2008 - 16:09 — marcosch

Hallo Herr Kneilmann,

die "alte" DB2-SQL-Lösung war 142 Zeilen lang. Dagegen ist ihr Konstrukt deutlich kürzer und übersichtlicher :-)

Allerdings muss ich das bereits in dem SQL lösen, weil ich mit dem Ergebnis dann noch auf andere Tabellen joinen muss. Die extrem kurze Lösung von Herrn Daubmann bietet sich da besser an :-)

Gruß aus Hamburg
Marco Schmidt

Log in or register to post comments

Stringbearbeitung in SAS-SQL

21 May, 2008 - 09:32 — Ulrich Daubmann

Hallo,

so geht's auch, egal ob in SQL oder Data-Step:

```
neuerstring = put(input(subpad(scan(translate(alterstring,' ',collate(0,
```

Viel Erfolg
Ulrich Daubmann

Log in or register to post comments

knackig

21 May, 2008 - 16:12 — marcosch

Hallo Herr Daubmann,

knackig kurz und genial. Danke ! Allerdings musste ich das SUBPAD (erst ab SAS9) in SUBSTR ändern, weil ich hierfür SAS8.2 nutzen muss.

Nachmal Danke! Das spart mir viel Zeit (und ich kann wieder etwas protzen, was SAS alles kann) :-)

Gruß aus Hamburg
Marco Schmidt

Log in or register to post comments

nochmal gefreut

8 March, 2011 - 16:54 — marcosch

ich habe mich grad noch einmal über diese Lösung gefreut... fast drei Jahre später :-)

Notiz für mich auf dem Host mit EBCDIC funktioniert es mit
COLLATE(64,239) || COLLATE(250,255)

Log in or register to post comments