

# **DSNREXX**

Guide DB2 – 8 février 2007

# Connect DB2

```
ADDRESS TSO "SUBCOM DSNREXX"  
          /*HOST CMD ENV AVAILABLE?*/  
package= 'DSNREX' || ISO  
  
IF RC  
THEN S_RC =  
      RXSUBCOM( 'ADD' , 'DSNREXX' , 'DSNREXX' )  
  
ADDRESS DSNREXX "CONNECT" SUBSYS  
  
ADDRESS DSNREXX      "EXECSQL SET CURRENT"  
                    "PACKAGESET= ' "package" ' "
```

# Execute immediate

- Pour les ordres
  - DML
    - INSERT
    - UPDATE
    - DELETE
  - DDL
  - DCL

```
ADDRESS DSNREXX "EXECSQL EXECUTE IMMEDIATE" ,  
              " :SQLSTMT"
```

## Cursor « simple » pas de host variable

```
STM1="SELECT COL1 , COL2 FROM TEST.TABLE "
```

```
ADDRESS DSNREXX "EXECSQL DECLARE C1 CURSOR FOR S1"
```

```
ADDRESS DSNREXX "EXECSQL PREPARE S1 FROM :SQLSTMT1"
```

```
ADDRESS DSNREXX "EXECSQL OPEN C1"
```

```
WHILE SQLCODE = 0
```

```
    ADDRESS DSNREXX "EXECSQL FETCH C1 INTO :V1, :V2",
```

```
END
```

# Utilisation des Host Variables

- Valable pour tous ordres DML:

```
SMT2 = "SELECT COL1 , COL2 FROM TABLEA WHERE COLA : ?"
ADDRESS DSNREXX "EXECSQL DECLARE C2 CURSOR FOR S2"
ADDRESS DSNREXX "EXECSQL PREPARE S2 FROM :SQLSTMT2"
ADDRESS DSNREXX "EXECSQL DESCRIBE S2 INTO :OUTSQLD2"

/*Addressage de la Host Variable */
OUTSQLD2.1.SQLDATA = varx
ADDRESS DSNREXX "EXECSQL OPEN C2 " ,
                "USING :OUTSQLD2.1.SQLDATA"
ADDRESS DSNREXX "EXECSQL FETCH C2" ,
                "USING DESCRIPTOR :OUTSQLD2"

/* recuperation des colonnes */
Col1 = OUTSQLD2.1.SQLDATA
Col2 = OUTSQLD2.2.SQLDATA
```

# SQLCODE

- Ne pas oublier de tester le SQLCODE !!!

```
IF SQLCODE ^= 0 & SQLCODE ^= 100 then
  DO
    SAY "SQLCODE " SQLCODE "SQLSTATE" SQLSTATE
    SAY "SQLERRMC" SQLERRMC "SQLERRP " SQLERRP
    SAY "SQLERRD.1" SQLERRD.1 "SQLERRD.2" SQLERRD.2
    SAY "SQLERRD.3" SQLERRD.3 "SQLERRD.4" SQLERRD.4
    SAY "SQLERRD.5" SQLERRD.5 "SQLERRD.6" SQLERRD.6
    SAY "SQLWARN.1" SQLWARN.1 "SQLWARN.2" SQLWARN.2
    SAY "SQLWARN.3" SQLWARN.3 "SQLWARN.4" SQLWARN.4
    SAY "SQLWARN.5" SQLWARN.5 "SQLWARN.6" SQLWARN.6
    SAY "SQLWARN.7" SQLWARN.7 "SQLWARN.8" SQLWARN.8
    SAY "SQLWARN.9" SQLWARN.9 "SQLWARN.A" SQLWARN.10
  END
```

# SQLBATCH by LK

- Super SPUFI Batch ( version ISPF à developper )
- Principal Fonction
  - Bypass de SQLCODE négatifs à la demande
  - Compression d'ordre pour tenir dans 32K
  - Formate les sorties de select pour être réemployé en sysin par exemple pour un bind
  - et autres fonctionnalités
- License GPL

# SQLBATCH : JCL

```
//STEP1 EXEC PGM=IKJEFT1B
//STEPLIB DD DISP=SHR,DSN=DSN710.SDSNEXIT
//          DD DISP=SHR,DSN=DSN710.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
  EXEC 'MY.CLIST(SQLBATCH'
//SYSIN DD *
  SSID(dsn)
//SQLIN DD *
  SELECT * FROM SYSIBM.SYSDATABASE
//SQLOUT DD SYSOUT=*
```

# SQLBATCH : Option

- SYSIN non positionnelle.
- A part le SSID, pas de paramètre obligatoire.

SSID(ssid)

[HELP(NO/HELP)]

[COMMIT('\_'/YES/SYNC/ROLLB/nnn)]

[GENSYNC(N/F/V,Freq/Col)]

[SHRINKSQL(NO/YES)]

[RESTART(nnn/literal)]

[BYPASS(sqlcode1,[sqlcode2,[sqlcode3]])]

[ISO(UR/CS/RR/RS)]

[PAD(YES/NO)] [CPAD(\_/x)] [NPAD(\_/0)]

[LRECIN(nnn)] [LRECOUT(nnn)]

[DELCHAR(x/NODEL)] [ContChar(x)]