データベース統合支援システムAdbis(2) : DMPのユーティリティプログラムと機能追加

二村, 祥一
九州大学大型計算機センター研究開発部

松尾, 文碩
九州大学大型計算機センター研究開発部

高木, 利久
九州大学大型計算機センター研究開発部

https://doi.org/10.15017/1474961

出版情報：九州大学大型計算機センター広報. 14 (3), pp.388-398, 1981-09-10. 九州大学大型計算機センター
バージョン：
権利関係：

データベース統合支援システムAdbis(2)

—— DMPのユーティリティプログラムと機能追加 ——

二村　祥一,*　松尾　文碩,*　高木　利久*

1. 概　要

本稿では，Adbis/DMP[1]に追加になった下記のユーティリティプログラムと機能について仕様を述べる．

・ユーティリティプログラム

プログラム名	機　　能	記　事
ADUDDSYN	Adbis/DMP-DDLによるデータベース定義の構文チェック．	2．1
ADUINTEG	データベース創成用生データの保全性チェック．	2．2
ADUREVTR	データベースから創成用形式データへの逆変換．	2．3
ADUDISP	データベース用データセットの確保量および使用量の表示．	2．4

・追加機能

サブルーチン名	内　　容	記　事
FIND¥D	一部機能追加．	3．1
REVT¥D	新規追加．ユーティリティADUREVTRに対応する機能を持つ．	3．2
DISP¥D	新規追加．ユーティリティADUDISPに対応する機能を持つ．	3．3

ユーティリティプログラムは，いずれもバッチ処理のもとで動作する．このためのカタログドプロシジャADBISUTYを用意しているので，EXEC文は下記のような形式になる．

$$//\quad \text{EXEC}\quad \text{ADBISUTY},\text{PROG}=\text{プログラム名},\ \text{SYSOUT}=\left\{\begin{matrix}\underline{\text{A}}\\ \text{O}\\ \text{S}\\ \text{K}\end{matrix}\right\}$$

形式は，すべてのユーティリティプログラムに共通しているため，本文のEXEC文の説明では，SYSOUTパラメータは省略している．

* 九州大学大型計算機センター研究開発部

2. Adbis／DMPユーティリティプログラム

2.1 ADUDDSYN(ADU DDL−syntax check)

(1) 機能

Adbis／DMP-DDL で書かれたデータベース定義の構文をチェックする．データベース定義は，データベース定義ファイル上に作られているものとする．

(2) ジョブ制御文

a．EXEC文

```
// EXEC ADBISUTY,PROG=ADUDDSYN
```

b．SYSUT1 DD文

対象とするデータベース定義ファイルを指定する．

(3) ユーティリティ制御文

な し

(4) 用途

DEFD¥Dによるマスタファイルの作成のまえに，データベース定義の構文をチェックするのに用いる．

(5) 使用例

データベース定義ファイルF0001.DDL.DATA のデータベース定義の構文をチェックする．

```
// EXEC ADBISUTY,PROG=ADUDDSYN
//SYSUT1 DD DSN=F0001.DDL.DATA,DISP=SHR
//
```

出力例

```
***** ADBIS/DMP UTILITY ADUDDSYN  07/14/81 13:15:00 *****

** DATA DEFINITION LIST          ：データベース定義ファイルのデータセット名とその内容が表示される．
   DSN= F0001.DDL.DATA

CNO  COMMAND    OPERAND
-------------------------------------------------------------
     * THIS IS AN EXAMPLE OF DATABASE DEFINITION.
001  DATABASE   NAME(DBTEST)
                ADMINISTRATOR(F0014)
                EXPLANATION('MANUFACTURERS-CARS DATABASE')
                MASTER(MSEATR)

002  RELATION   NAME(SANUCARS)
                ALIAS(MC)
                VARIABLE(MNO,CNO)
                COMPLETENESS(10)
                EXPLANATION('JANUARY,1978--->DECEMBER,1980')
                SPACE(3000)

003  FUNCTION   NAME(COLOR)
                VARIABLE(CNO)
                TYPE(C6)
                RANGE('RED','BLUE','WHITE','BLACK')
                INVERSE

004  VECTFUNC   NAME(MANUFACT)
                ALIAS(M)
                VARIABLE(MNO)
                VECTOR(MNAME,PRODUCT,CITY)
                INVERSE(CITY)

005  VARIABLE   NAME(MNO)      TYPE(I4)
006  VARIABLE   NAME(CNO)      TYPE(X4)   DOMAIN(50,/100,200/)
007  VARIABLE   NAME(MNAME)    TYPE(C4)
008  VARIABLE   NAME(PRODUCT)  TYPE(R8)
009  VARIABLE   NAME(CITY)     TYPE(C10) DOMAIN('DETROIT','PARIS','TOYOTA')
010  END


** DIAGNOSTIC MESSAGES          ：ADUDDSYNによるデータベース定義の診断メッセージが表示される．

ADU003: NO WRITEPASSWORD IN 001
ADU006: NAME OPERAND ERROR IN 002
ADU012: TYPE OPERAND ERROR IN 006
```

2.2 ADUINTEG（ADU integrity check）

(1) 機能

創成用形式のデータの保全性のチェックを行う．データベース定義ファイルのデータベース定義に従ってデータが取りうる値の範囲内にあるかどうかをチェックする．同時にプルーフリストをとることもできる．

(2) ジョブ制御文

a．EXEC文

```
// EXEC ADBISUTY,PROG=ADUINTEG
```

b．SYSUT1 DD文

データベース定義ファイルを指定する．

c．SYSUT2 DD文

創成用形式データのデータセットを指定する．

d．SYSIN DD文

ユーティリティ制御文を含むデータセットを指定する．

(3) ユーティリティ制御文

形式：

```
ELEMENT＝名前 ［LIST=(［n1：］n2)］ ［ERRREC=(［n1：］n2)］
```

説明：

・ELEMENT＝名前：関係，関数あるいはベクトル関数の名前を指定する．創成用形式データはELEMENTで指示されたデータベース要素の定義情報に従ってチェックされる．

・LIST=(［n_1：］n_2)：創成用形式データの第n_1番目から第n_2番目までの作表を行う．n_1を省略した場合，1が指定されたものとみなす．

・ERRREC=(［n_1：］n_2)：創成用形式データの第n_1番目から第n_2までのエラーデータの作表を行う．n_1を省略した場合，1が指定されたものとみなす．ERRREC＝0と指定した場合，エラーメッセージのみが出力される．

(4) 用途

CREA¥Dによるデータファイル創成の前に，創成用形式データの保全性のチェックを行うのに使用する．

(5) 使用例

創成用形式データF0001.INPUT.DATAをデータベース定義ファイルF0001.DDL.DATAの関係RELA01に従ってチェックする．エラーデータを最初から第2000番目のデータについて出力する．

```
// EXEC ADBISUTY,PROG=ADUINTEG
//SYSUT1 DD DSN=F0001.DDL.DATA,DISP=SHR
//SYSUT2 DD DSN=F0001.INPUT.DATA,DISP=SHR
//SYSIN  DD *
   ELEMENT=RELA01  ERRREC=2000
//
```

出力例

```
***** ADBIS/DMP UTILITY ADUINTEG  07/14/81 16:15:00     *****

** RELATED FILE NAMES        ：データベース定義ファイルと入力データのデータセット名が表示される．
   DATA DEFINITION FILE= F0001.DDL.DATA
   INPUT FILE          = F0001.INPUT.DATA

** CONTROL STATEMENTS        ：ADUINTEGに対するユーティリティ制御文が表示される．
   ELEMENT= RELA01
   ERRREC = (1:2000)

** ERROR RECORD LIST         ：エラーレコードが表示される．エラーデータには，▼*▼が付加される．

RECORDNO      VALUE
-------------------------------------------------
#0000210        210          -30*       'RED    '
#0000404        404      3909100*       'BLUE   '
#0000512        512         1400        'XYZ    '*
#0000630        630           -1*       'WHITE  '
#0001300       1300         4500        'XXXXXX'*
```

2.3 ADUREVTR（ADU reverse transformation）

(1) 機能

データファイルから創成用形式のデータへの逆変換を行う．

(2) ジョブ制御文

a．EXEC文

```
// EXEC ADBISUTY,PROG=ADUREVTR
```

b．SYSUT1 DD文

創成用形式データ格納用のデータセットを指定する．

c．SYSIN DD文

ユーティリティ制御文を含むデータセットを指定する．

(3) ユーティリティ制御文

形式：

DATABASE=データベース名 ADMINISTRATOR=データベース管理者課題名 ELEMENT=名前

説明：

・DATABASE＝データベース名：データファイルを含むデータベースの名前を指定する．

・ADMINISTRATOR=データベース管理者課題名：データベース管理者の課題名を指定する．

・ELEMENT=名前：逆変換の対象とする関係，関数，ベクトル関数の名前を指定する．

(4) 使用例

データベース管理者F0001のデータベースDB01に創成ずみの関係RELA01から，創成用形式データをデータセットF0001.RE01.DATAに逆変換する．

```
// EXEC ADBISUTY,PROG=ADUREVTR
```

```
//SYSUT1   DD DSN=F0001.RE01.DATA,DISP=(NEW,CATLG),
//         UNIT=PUB,SPACE=(TRK,(50,50),RLSE)
//SYSIN DD *
     DATABASE=DB01 ADMINISTRATOR=F0001 ELEMENT=RELA01
//
```

2.4 ADUDISP (ADU display of datafile status)

(1) 機能

データファイルの確保量，使用量をデータベース全体あるいはデータファイル単位に表示する．

(2) ジョブ制御文

a．EXEC文

```
// EXEC ADBISUTY,PROG=ADUDISP
```

b．SYSIN DD文

ユーティリティ制御文を含むデータセットの名前を指定する．

(3) ユーティリティ制御文

形式：

DATABASE=データベース名 ADMINISTRATOR=データベース管理者課題名［ELEMENT=名前］

説明：

- DATABASE=データベース名：データベースの名前を指定する．
- ADMINISTRATOR=データベース管理者課題名：データベース管理者の課題名を指定する．
- ELEMENT=名前：対象とする関係，関数，ベクトル関数の名前を指定する．ELEMENTオペランド省略時はデータベースに登録されているすべての関係，関数，ベクトル関数の情報を出力する．

(4) 用途

データファイルの確保量，使用量を知るのに用いる．

(5) 使用例

データベース管理者F0001のデータベースDB01のデータファイルの使用状況を調べる．

```
// EXEC ADBISUTY,PROG=ADUDISP
//SYSIN DD *
   DATABASE=DB01 ADMINISTRATOR=F0001
//
```

出力例

```
***** ADBIS/DMP UTILITY ADUDISP   07/14/81 14:15:35    *****

** DATAFILE SPACES          ：データベース名，データベース管理者名およびデータファイルの確保量，使用量
   DATABASE NAME= DB01        が表示される．
   ADNINISTRATOR= F0001

NAME     ALLOC_SPACE(KB)      USED_SPACE(KB)
-----------------------------------------
RELA01              -
RELA02           1000          700(70.0%)
RELA03           1000          456(45.6%)
FUNC01              -
FUNC02              -
VECT01              -
VECT02            500          200(40.0%)
VECT03              -
```

3. Adbis/DMPの機能追加

3.1 FIND¥D(find)

(1) 機能

関係，関数，あるいはベクトル関数から，引数リストで指定された条件を満たすレコードのキー集合を求める．この機能は，例えば，5項関係(A,B,C,D,E)において，引数リストで(a,b,*,d,*)(a,b,dはそれぞれA,B,Dの値；*は不定を表す)と指定すると，A＝a，B＝b，D＝dであるすべての組に対するキー集合を求める．関数，ベクトル関数については，引数，関数値の組からなる関係と考える．FIND¥Dで求めたものが現キー集合であり，これをSAVE¥Dで保存することにより，集合演算の対象となる．キー集合から値集合を求めるにはEVAL¥Dを使用する．

(2) 呼び出し形式と引数

```
CALL FIND¥D(P1, P2, P3, P4)
```

P1：完了コード(Iv*4)

0＝正常終了．

2＝第3引数で指定された名前が，データベースに定義されていない．

3＝データファイルが創成されていない．

4＝引数リストの指定に誤りがある．

5＝検索項目に対して転置形が作成されていない．

8＝第4引数の検索条件が35，36，37，38の場合で，検索値の指定が正しくない．

10＝データファイルのリードエラーである．

50＝データベースがオープンされていない．

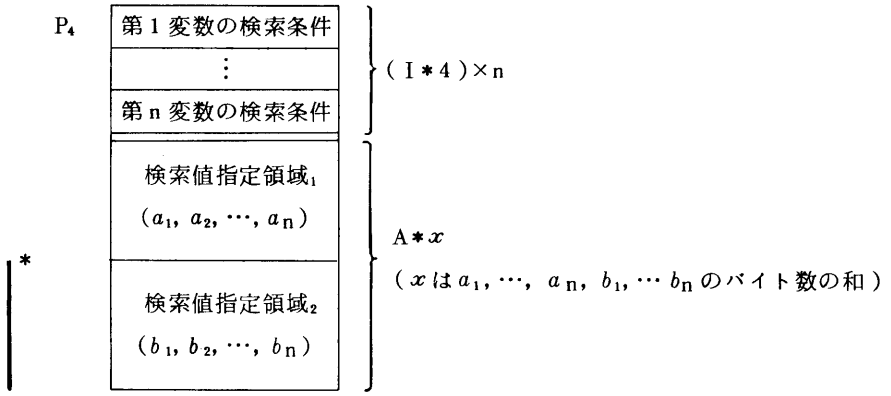
P_2: キー集合の要素数（Iv＊4）

求まったキー集合の要素数が設定される．

P_3: 名前（A＊8，Av＊8，Aarray＊8）

検索対象とする関係，関数，ベクトル関数の名前，あるいは別名を指定する．

P_4: 引数リスト（I＊4 array）



検索条件＝
- 0 … その変数の値が任意でよいことを示す．
- 1 … ＝，第i変数の値が指定値a_iと等しいレコードを探す．
- 2 … ≠，第i変数の値が指定値a_iと等しくないレコードを探す．
- 3 … ＞，第i変数の値が指定値a_iより大きいレコードを探す．
- 4 … ≧，第i変数の値が指定値a_iより大きいか等しいレコードを探す．
- 5 … ＜，第i変数の値が指定値a_iより小さいレコードを探す．
- 6 … ≦，第i変数の値が指定値a_iより小さいか等しいレコードを探す．
- 35 … 第i変数の値x_iが$a_i < x_i < b_i$を満足するレコードを探す．
- 36 … 第i変数の値x_iが$a_i < x_i \leqq b_i$を満足するレコードを探す．
- 45 … 第i変数の値x_iが$a_i \leqq x_i < b_i$を満足するレコードを探す．
- 46 … 第i変数の値x_iが$a_i \leqq x_i \leqq b_i$を満足するレコードを探す．

(3) 使用上の注意

・関係を構成する変数の属性などは，SHOW¥Dで調べる．

・演算結果の要素数が0でも，それは現キー集合となる．

(4) 適用

特定の条件を満たすレコードを求めるのに用いる．

(5) 使用例

変数A，B，Cからなる関係RELA01で，条件100≦A＜200，B＝任意，C＝▼ABCDEF▼を満足するキー集合を求める．ここで，A，B，Cの型はそれぞれI4，I2，C6とする．

＊｜は機能追加された部分を示す．

```
INTEGER ALIST(9)
      ⋮
ALIST(1)= 45
ALIST(2)=  0
ALIST(3)=  1
ALIST(4)=100
ALIST(5)=▼△△AB▼
ALIST(6)=▼CDEF▼
ALIST(7)=200
ALIST(8)=  0
ALIST(9)=  0
CALL FIND¥D(IRC,KNO,▼RELA01△△▼,ALIST)
```

ALIST

45
0
1
100
△*△*A B
C D E F
200
0*
0*

*任意の値でよい

3.2 REVT¥D(reverse transformation)

(1) 機能

データファイルから創成用形式のデータに逆変換する．Adbis／DMPユーティリティの ADURE VTRと同等の機能を持つ．

(2) 呼び出し形式と引数

```
CALL REVT¥D(P1, P2)
```

[P1]：完了コード(Iv*4)

0＝正常終了．

2＝第2引数で指定された名前がデータベースに定義されていない．

3＝データファイルが創成されていない．

4＝創成用形成データを格納するデータセットが割り当てられていない．

10＝データファイルのリードエラーである．

50＝データベースがオープンされていない．

P2：名前(A*8, Av*8, Aarray*8)

対象とする関係，関数，ベクトル関数の名前，あるいは別名を指定する．

(3) 使用上の注意

創成用形式のデータを格納する出力データセットは，あらかじめ領域を確保し，DD名=SYSUT1を割り当てておく必要がある．データセットの形式はVBとなる．

(4) 使用例

関係RELA01のデータファイルから創成用形式のデータをデータセットF0001.RE01.DATAに逆変換する．

i) 出力データセットの割り当て．

・JCL(バッチ処理)

```
//SYSUT1  DD DSN=F0001.RE01.DATA,DISP=(NEW,CATLG),
//          UNIT=PUB,SPACE=(TRK,(50,50),RLSE)
```

・TSSコマンド

```
ALLOC F(SYSUT1) DA(RE01.DATA) NE CA TRA SP(50 50)
```

ii) アプリケーションプログラムの実行

```
CALL REVT¥D(IRC,▼RELA01△△▼)
```

3.3 DISP¥D(display datafile status)

(1) 機能

データファイルの確保量，使用量を，データベース全体あるいはデータファイル単位に調べる．

DISP¥Dは，Adbis／DMPユーティリティのADUDISPと同等の機能を持つ．

(2) 呼び出し形式と引数

```
CALL DISP¥D(P1, P2, P3, P4)
```

[P1]: 完了コード(Iv*4)

0＝正常終了．

1＝第2，第3引数で指定した値設定領域が不足している．

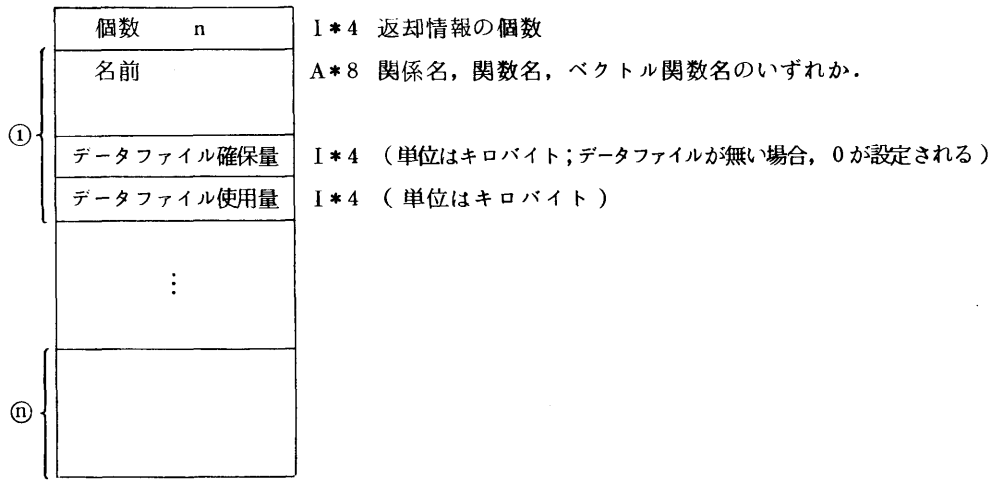
2＝第4引数の名前の指定がおかしい．

3＝データファイルのリードエラーである．

50＝データベースがオープンされていない．

[P2]: 値設定領域(I*4 array)

値設定領域には次の形式で情報が設定される．



P3: 値設定領域のバイト長(I*4，Iv*4)

P4: 名前あるいは▼#ALLNAME▼(A*8，Av*8，Aarray*8)

名前として，関係名，関数名，ベクトル関数名が指定できる．データベース全体について調べる場合▼#ALLNAME▼を指定する．

(3) 適用

データファイルの確保量，使用量を知るのに用いる．

(4) 使用例

関係RELA01のデータファイルの確保量，使用量を調べる．

```
INTEGER BUF(5)
   ⋮
CALL DISP¥D(IRC,BUF,20,▼RELA01△△▼)
```

参考文献

1. 松尾，二村，高木 データベース統合支援システムAdbis(1) — Adbisの概要とDMPの仕様 — ，九大大型計算機センター広報，**14**，2，1981，172-217.

参考文献1の正誤表

頁	行	正	誤
172	↓ 13	DMP(Data Manipulation	DMP(Data Base Manipulation
172	↑ 1	アプリケーションプログラム	アプリケーション
174	↓ 16	非職業的	非識業的
179	↓ 5	FACTORY	FACTORT
179	↑ 9	関数値	関係値
179	↑ 5	FATHER_OF	PATHER_OF
180	↓ 18	London	Londou
180	↑ 3	データベース	データーベース
181	↓ 1	マスタファイル	マスターファイル
182	↑ 14	データベース管理者が	データベース管理が
188	↓ 9	d.EXPLANATION	d.EXPLANAION
188	↓ 11	5.3.6	5.2.6
188	↓ 20	関係MANUCARS	関係MANCARS
193	↑ 9	F0001.DDL.DATA	F0001,DDL.DATA
194	↑ 11	(P_1 = 3で通知)	(P1 = 3で通知)
195	↑ 6	変数名またはそれらの別名が	変数名が
197	↓ 18	第1項目の長さ	第1項目の長
197	↓ 23	n_1×(A*8)	n_1×(I*4)
198	↓ 13 198 ↑ 7 198 ↑ 5	} range	} domain
202	↑ 6	定されていない)．	定されていない．
205	↓ 6	(xはa_1 … a_nのバイト数の和)	(長さは第3引数で指定された関係などの属性で決まる)
208	↓ 15	△△▼を指定する．	△△を指定する．
216	↑ 1	ユーティリティプログラム	ユーティティプログラム