OBTAIN

OBTAIN *operand1* **...**

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Einschränkung
- Syntax-Beschreibung
- Beispiele

Funktion

Das Statement `OBTAIN` dient dazu, ein oder mehrere Datenbankfelder zu lesen. Das `OBTAIN`-Statement erzeugt keinen ausführbaren Code im Natural-Objektprogramm. Es wird in erster Linie benutzt, um einen Wertebereich eines multiplen Feldes oder einen Bereich von Ausprägungen einer Periodengruppe zu lesen, so dass Teile dieser Bereiche nacheinander im Programm referenziert werden können.

Ein `OBTAIN`-Statement ist für jedes im Programm zu referenzierende Datenbankfeld nicht erforderlich, da Natural ja automatisch jedes in einem nachfolgenden Statement referenzierte Datenbankfeld liest (zum Beispiel, ein `DISPLAY`- oder `COMPUTE`-Statement).

Wenn multiple Felder oder Periodengruppen in der Form eines Arrays referenziert werden, muss das Array mit einem `OBTAIN`-Statement definiert werden, um sicher zu stellen, dass es für alle Ausprägungen des Feldes erstellt ist. Wenn einzelne multiple Felder oder Periodengruppen referenziert werden, bevor das Array definiert wird, werden die Felder nicht innerhalb des Arrays positioniert und existieren unabhängig vom Array. Die Felder enthalten denselben Wert wie die entsprechende Ausprägung innerhalb des Arrays.

Einzelne Ausprägungen oder multiple Felder oder Periodengruppen oder Subarrays können innerhalb eines vorher definierten Arrays gehalten werden, wenn die Array-Dimensionen der zweiten individuellen Ausprägung oder das Array innerhalb des ersten Arrays enthalten sind.

Referenzen auf multiple Felder oder Periodengruppen mit eindeutigem variablen Index können nicht in einem Array von Werten enthalten sein. Wenn einzelne Ausprägungen eines Arrays mit einem variablen Index verarbeitet werden sollen, muss dem Index-Ausdruck der eindeutige variable Index vorausgehen, um das individuelle Array zu kennzeichnen.

Einschränkung

Das `OBTAIN`-Statement gilt nur für den Reporting Mode.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur					Mögliche Formate												Referenzierung erlaubt	Dynam. Definition
operand1		S	A	G		A	U	N	P	I	F	B	D	T	L				ja	ja

Syntax-Element-Beschreibung:

operand1	Als *operand1* geben Sie die Felder an, die mit dem `OBTAIN`-Statement gelesen werden sollen.

Beispiele:

```
READ FINANCE OBTAIN CREDIT-CARD (1-10)
DISPLAY CREDIT-CARD (3-5) CREDIT-CARD (6-8)
SKIP 1 END
```

Das obige Beispiel führt dazu, dass die ersten 10 Ausprägungen des Feldes `CREDIT–CARD` (das in einer Periodengruppe enthalten ist) gelesen werden, und die Ausprägungen `(3–5)` und `(6–8)` angezeigt werden, wo die nachfolgenden Subarrays im ersten Array `(1–10)` residieren.

```
READ FINANCE
MOVE 'ONE' TO CREDIT-CARD (1)
DISPLAY CREDIT-CARD (1) CREDIT-CARD (1-5)
```

Ausgabe:

```
    CREDIT-CARD      CREDIT-CARD
------------------ ----------------


ONE                DINERS CLUB
                   AMERICAN EXPRESS



ONE                AVIS
                   AMERICAN EXPRESS



ONE                HERTZ
                   AMERICAN EXPRESS



ONE                UNITED AIR TRAVEL
```

Die erste Referenz auf `CREDIT–CARD (1)` ist nicht innerhalb des Arrays enthalten. Das Array, das nach der Referenz auf die eindeutige Ausprägung `(1)` definiert ist, kann rückwirkend keine eindeutige Ausprägung oder ein Array enthalten, das kürzer als das definierte Array ist.

```
READ FINANCE
OBTAIN CREDIT-CARD (1-5)
MOVE 'ONE' TO CREDIT-CARD (1)
DISPLAY CREDIT-CARD (1) CREDIT-CARD (1-5)
```

Ausgabe:

```
    CREDIT-CARD      CREDIT-CARD
------------------ ---------------


ONE                ONE
                   AMERICAN EXPRESS



ONE                ONE
                   AMERICAN EXPRESS



ONE                ONE
                   AMERICAN EXPRESS



ONE                ONE
```

Die individuelle Referenz auf `CREDIT–CARD (1)` ist innerhalb des im `OBTAIN`-Statement definierten Arrays enthalten.

```
MOVE (1) TO INDEX
READ FINANCE
DISPLAY CREDIT-CARD (1-5) CREDIT-CARD (INDEX)
```

Ausgabe:

```
    CREDIT-CARD      CREDIT-CARD
------------------- ---------------


DINERS CLUB         DINERS CLUB
AMERICAN EXPRESS



AVIS                AVIS
AMERICAN EXPRESS



HERTZ               HERTZ
AMERICAN EXPRESS



UNITED AIR TRAVEL  UNITED AIR TRAVEL
```

Die Referenz auf CREDIT–CARD mittels der variablen Index-Notation ist nicht innerhalb des Arrays enthalten.

```
RESET A(A20) B(A20) C(A20)
MOVE 2 TO I (N3)
MOVE 3 TO J (N3)
READ FINANCE
 OBTAIN CREDIT-CARD (1:3) CREDIT-CARD (I:I+2) CREDIT-CARD (J:J+2)
 FOR K (N3) = 1 TO 3
  MOVE CREDIT-CARD (1.K) TO A
  MOVE CREDIT-CARD (I.K) TO B
  MOVE CREDIT-CARD (J.K) TO C
  DISPLAY A B C
 LOOP /* FOR
LOOP / * READ
END
```

Ausgabe:

```
         A                    B                    C
------------------- ------------------- -------------------
CARD 01             CARD 02             CARD 03
CARD 02             CARD 03             CARD 04
CARD 03             CARD 04             CARD 05
```

Die drei Arrays können einzeln aufgerufen werden, indem Sie den eindeutigen Basis-Index als Kennzeichner für den Index-Ausdruck benutzen.

Ungültiges Beispiel 1

```
READ FINANCE
OBTAIN CREDIT-CARD (1-10)
FOR I 1 10
MOVE CREDIT-CARD (I) TO A(A20)
WRITE A
END
```

Das obige Beispiel erzeugt die Fehlermeldung NAT1006 (Wert für variablen Index = 0), weil als der Datensatz mit READ gelesen wurde, der Index I noch den Wert 0 enthielt.

Auf jeden Fall würden in dem obigen Beispiel nicht die ersten 10 Ausprägungen von CREDIT–CARD ausgegeben werden, weil die einzelne Ausprägung mit dem variablen Index nicht im Array enthalten sein kann und der variable Index (I) nur ausgewertet wird, wenn der nächste Datensatz gelesen wird.

Im Folgenden finden Sie die korrekte Methode, um das oben Beschriebene auszuführen:

```
READ FINANCE
OBTAIN CREDIT-CARD (1-10)
FOR I 1 10
MOVE CREDIT-CARD (1.I) TO A (A20)
WRITE A
END
```

Ungültiges Beispiel 2

```
READ FINANCE
FOR I 1 10
WRITE CREDIT-CARD (I)
END
```

Das obige Beispiel erzeugt die Fehlermeldung NAT1006, weil der Index I Null ist, wenn der Datensatz im READ-Statement gelesen wird.

Im Folgenden wird die korrekte Methode verwendet, um das oben Beschriebene auszuführen:

```
READ FINANCE
FOR I 1 10
GET SAME
WRITE CREDIT-CARD (0030/I)
END
```

Das GET SAME-Statement ist erforderlich, um den Datensatz erneut zu lesen, nachdem der variable Index in der FOR-Schleife aktualisiert worden ist.

Beispiele

- Beispiel 1 — OBTAIN-Statement
- Beispiel 2 — OBTAIN-Statement mit mehreren Bereichen

Beispiel 1 — OBTAIN-Statement

```
** Example 'OBTEX1': OBTAIN
************************************************************************
RESET #INDEX (I1)
*
LIMIT 5
READ EMPLOYEES BY CITY
  OBTAIN SALARY (1:4)
  /*
  IF SALARY (4) GT 0 DO
    WRITE '=' NAME / 'SALARIES (1:4):' SALARY (1:4)
    FOR #INDEX 1 TO 4
      WRITE 'SALARY' #INDEX  SALARY (1.#INDEX)
    LOOP
    SKIP 1
  DOEND
LOOP
*
END
```

Ausgabe des Programms OBTEX1:

```
Page      1                                                  05-02-08  13:37:48

NAME: SENKO
SALARIES (1:4):      31500      29900      28100      26600
SALARY    1      31500
SALARY    2      29900
SALARY    3      28100
SALARY    4      26600
```

```
NAME: HAMMOND
SALARIES (1:4):      22000      20200      18700      17500
SALARY    1      22000
SALARY    2      20200
SALARY    3      18700
SALARY    4      17500
```

Beispiel 2 — OBTAIN-Statement mit mehreren Bereichen

```
** Example 'OBTEX2': OBTAIN (with multiple ranges)
************************************************************************
RESET #INDEX (I1) #K (I1)
*
#INDEX  := 2
#K      := 3
*
LIMIT 2
*
READ EMPLOYEES BY CITY
  OBTAIN SALARY (1:5)
         SALARY (#INDEX:#INDEX+3)
  /*
  IF SALARY (5) GT 0 DO
    WRITE '=' NAME
    WRITE 'SALARIES (1-5):' SALARY (1:5) /
    WRITE 'SALARIES (2-5):' SALARY (#INDEX:#INDEX+3)
    WRITE 'SALARIES (2-5):' SALARY (#INDEX.1:4) /
    WRITE 'SALARY 3:' SALARY (3)
    WRITE 'SALARY 3:' SALARY (#K)
    WRITE 'SALARY 4:' SALARY (#INDEX.#K)
  DOEND
LOOP
```

Ausgabe des Programms OBTEX2:

```
Page      1                                                 05-02-08  13:38:31

NAME: SENKO
SALARIES (1-5):      31500      29900      28100      26600      25200

SALARIES (2-5):      29900      28100      26600      25200
SALARIES (2-5):      29900      28100      26600      25200

SALARY 3:      28100
SALARY 3:      28100
SALARY 4:      26600
```

Weitere Beispiele zur Benutzung des OBTAIN-Statements, siehe *Datenbank-Array referenzieren* im *Leitfaden zur Programmierung*.