ACCEPT/REJECT

```
{ ACCEPT } [IF] logical-condition
{ REJECT }
```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Verarbeitung mehrerer ACCEPT/REJECT-Statements
- Limit-Notation
- Hold-Status
- Beispiele

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Verwandte Statements: AT BREAK | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | HISTOGRAM | GET | GET SAME | GET TRANSACTION DATA | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Gehört zur Funktionsgruppe: *Datenbankzugriffe und Datenbankänderungen*

Funktion

Mit den Statements `ACCEPT` und `REJECT` können Sie eine logische Bedingung (*logical-condition*) angeben, aufgrund welcher ein gelesener Datensatz akzeptiert (`ACCEPT`) oder zurückgewiesen (`REJECT`) werden soll.

Beide Statements können in Verbindung mit Statements eingesetzt werden, die Datensätze in einer Verarbeitungsschleife lesen (`FIND`, `READ`, `HISTOGRAM`, `CALL FILE`, `SORT` oder `READ WORK FILE`). Die logische Bedingung wird erst ausgewertet, nachdem ein Datensatz ausgewählt/gelesen worden ist.

Wenn ein `ACCEPT`- bzw. `REJECT`-Statement ausgeführt wird, bezieht es sich auf die innerste gerade aktive Verarbeitungsschleife, die mit einem der oben genannten Statements initiiert wurde.

Befindet sich ein `ACCEPT`- bzw. `REJECT`-Statement in einer Subroutine und wird aufgrund der logischen Bedingung ein Datensatz zurückgewiesen, so wird die Subroutine automatisch beendet und die Verarbeitung mit dem nächsten Datensatz der innersten gerade aktiven Verarbeitungsschleife fortgesetzt.

Syntax-Beschreibung

IF	Eine IF-Klausel kann mit einem ACCEPT- oder REJECT-Statement verwendet werden, um logische Bedingungen über die Bedingungen hinaus anzugeben, die spezifiziert wurden, als der Satz mit einem FIND-, READ- oder HISTOGRAM-Statement ausgewählt/gelesen wurde. Die logischen Bedingungen werden ausgewertet, nachdem der Satz gelesen und seine Verarbeitung gestartet wurde.
logical-condition	Das Basis-Kriterium ist ein relationaler Ausdruck. Mehrere relationale Ausdrücke können mit logischen Operatoren zu komplexen Kriterien kombiniert werden (AND, OR). Arithmetische Ausdrücke können auch zur Bildung eines relationalen Ausdrucks benutzt werden. Felder können Datenbankfelder oder Benutzervariablen sein. Weitere Informationen zu logischen Bedingungen, siehe *Logische Bedingungen* im *Leitfaden zur Programmierung*. Wenn ACCEPT/REJECT mit einem HISTOGRAM-Statement benutzt wird, kann nur das im HISTOGRAM-Statement angegebene Datenbankfeld als logische Bedingung verwendet werden.

Verarbeitung mehrerer ACCEPT/REJECT-Statements

Pro Verarbeitungsschleife genügt in der Regel ein ACCEPT- bzw. REJECT-Statement. Wollen Sie in einer Verarbeitungsschleife mehrere ACCEPT/REJECT-Statements unmittelbar hintereinander verwenden, so beachten Sie bitte folgende Regeln:

- Befinden sich innerhalb einer Verarbeitungsschleife mehrere ACCEPT/REJECT-Statements direkt hintereinander, so werden sie in der angegebenen Reihenfolge verarbeitet.
- Wird aufgrund einer erfüllten ACCEPT-Bedingung ein Datensatz akzeptiert, so werden die unmittelbar nachfolgenden ACCEPT/REJECT-Statements ignoriert.
- Wird aufgrund einer erfüllten REJECT-Bedingung ein Datensatz zurückgewiesen, so werden die unmittelbar nachfolgenden ACCEPT/REJECT-Statements ignoriert.
- Geht die Verarbeitung bis zum letzten ACCEPT/REJECT-Statement, so entscheidet dieses letzte Statement, ob der betreffende Datensatz akzeptiert wird oder nicht.

Befindet sich zwischen zwei ACCEPT/REJECT-Statements ein anderes Statement, so werden beide ACCEPT/REJECT-Statements unabhängig voneinander verarbeitet.

Limit-Notation

Ist die Anzahl der Durchläufe einer Verarbeitungsschleife durch ein LIMIT-Statement oder eine andere Einschränkung begrenzt, so gilt diese für die Anzahl der gelesenen Datensätze, und zwar unabhängig davon, wieviele der gelesenen Datensätze aufgrund eines ACCEPT- oder REJECT-Statements akzeptiert

oder zurückgewiesen werden.

Hold-Status

Eine ACCEPT/REJECT-Verarbeitung hat keinen Einfluss darauf, ob ein im "Hold" gehaltener Datensatz freigegeben wird — es sei denn, der Profilparameter RI ist auf ON gesetzt (dieser Parameter steht nur auf Großrechnern zur Verfügung).

Beispiele

- Beispiel 1 — ACCEPT
- Beispiel 2 — ACCEPT / REJECT

Beispiel 1 — ACCEPT

```
** Example 'ACREX1': ACCEPT
************************************************************************
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 SEX
  2 MAR-STAT
END-DEFINE
*
LIMIT 50
READ EMPLOY-VIEW
  ACCEPT IF SEX='M' AND MAR-STAT = 'S'
  WRITE  NOTITLE '=' NAME '=' SEX 5X '=' MAR-STAT
END-READ
END
```

Ausgabe des Programms ACREX1:

```
NAME: MORENO               S E X: M     MARITAL STATUS: S
NAME: VAUZELLE             S E X: M     MARITAL STATUS: S
NAME: BAILLET              S E X: M     MARITAL STATUS: S
NAME: HEURTEBISE           S E X: M     MARITAL STATUS: S
NAME: LION                 S E X: M     MARITAL STATUS: S
NAME: DEZELUS              S E X: M     MARITAL STATUS: S
NAME: BOYER                S E X: M     MARITAL STATUS: S
NAME: BROUSSE              S E X: M     MARITAL STATUS: S
NAME: DROMARD              S E X: M     MARITAL STATUS: S
NAME: DUC                  S E X: M     MARITAL STATUS: S
NAME: BEGUERIE             S E X: M     MARITAL STATUS: S
NAME: FOREST               S E X: M     MARITAL STATUS: S
NAME: GEORGES              S E X: M     MARITAL STATUS: S
```

Beispiel 2 — ACCEPT / REJECT

```
** Example 'ACREX2': ACCEPT/REJECT
************************************************************************
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 SALARY    (1)
```

```
*
1 #PROC-COUNT (N8) INIT <0>
END-DEFINE
*
EMP. FIND EMPLOY-VIEW WITH NAME = 'JACKSON'
  WRITE NOTITLE *COUNTER NAME FIRST-NAME 'SALARY:' SALARY(1)
  /*
  ACCEPT IF SALARY (1) LT 50000
  WRITE *COUNTER 'ACCEPTED FOR FURTHER PROCESSING'
  /*
  REJECT IF SALARY (1) GT 30000
  WRITE *COUNTER 'NOT REJECTED'
  /*
  ADD 1 TO #PROC-COUNT
END-FIND
*
SKIP 2
WRITE NOTITLE 'TOTAL PERSONS FOUND ' *NUMBER (EMP.) /
              'TOTAL PERSONS SELECTED' #PROC-COUNT
END
```

Ausgabe des Programms ACREX2:

```
          1 JACKSON              CLAUDE              SALARY:      33000
          1 ACCEPTED FOR FURTHER PROCESSING
          2 JACKSON              FORTUNA             SALARY:      36000
          2 ACCEPTED FOR FURTHER PROCESSING
          3 JACKSON              CHARLIE             SALARY:      23000
          3 ACCEPTED FOR FURTHER PROCESSING
          3 NOT REJECTED


TOTAL PERSONS FOUND           3
TOTAL PERSONS SELECTED        1
```