

Statements nach Funktionen

Dieses Kapitel liefert eine Übersicht über die nach Funktionen eingeteilten Statements:

- Datenbankzugriffe und Datenbankänderungen
- Arithmetische Funktionen und Datenzuweisungen
- Schleifenverarbeitung
- Erstellen von Ausgabe-Reports
- Bildschirmgenerierung für interaktive Verarbeitung
- Verarbeitung logischer Bedingungen
- Aufrufen von Programmen und Unterprogrammen
- Beenden von Programmen und Sessions
- Verarbeitung von Arbeitsdateien
- Komponentenbasierte Programmierung
- Ereignisgesteuerte Programmierung
- Speicherverwaltung für dynamische Variablen/X-Arrays
- Natural Remote Procedure Call
- Internet und XML
- Sonstige Statements
- Reporting Mode-Statements

Anmerkungen:

1. Manche Statements können sowohl im Structured Mode als auch im Reporting Mode verwendet werden, während andere nur im Reporting Mode verwendet werden können. Siehe auch *Natural-Programmiermodi* im *Leitfaden zur Programmierung*
2. Die Statements DLOGOFF, DLOGON, SHOW, IMPORT und EXPORT sind nur verfügbar, wenn Entire DB installiert ist. Eine Beschreibung dieser Statements finden Sie in der *Entire DB*-Dokumentation.

Datenbankzugriffe und Datenbankänderungen

Natural-DML-Statements

Die folgenden Natural Data Manipulation Language-Statements (DML) dienen zum Zugriff auf und zum Ändern von in einer Datenbank gespeicherten Daten:

READ	Lesen einer Datei in physischer oder logischer Reihenfolge der Datensätze.
FIND	Auswählen von Datensätzen aufgrund bestimmter Kriterien.
HISTOGRAM	Lesen von Werten eines Datenbankfeldes.
GET	Lesen eines Datensatzes mit einer bestimmten ISN (Internal Sequence Number) bzw. SNR (Record Number).
GET SAME	Erneutes Lesen des gerade verarbeiteten Datensatzes.
ACCEPT/REJECT	Annehmen/Ablehnen von Datensätzen aufgrund bestimmter Kriterien.
PASSW	Angabe eines Passworts zur Zugriffsberechtigung auf eine passwortgeschützte Datei.
LIMIT	Begrenzen der Anzahl der Ausführungen einer READ-, FIND- oder HISTOGRAM-Schleife.
STORE	Anlegen eines neuen Datensatzes in der Datenbank.
UPDATE	Ändern eines Datensatzes in der Datenbank.
DELETE	Löschen eines Datensatzes von der Datenbank.
END TRANSACTION	Festlegen des Endes einer logischen Transaktion.
BACKOUT TRANSACTION	Abbrechen einer nicht vollständig abgeschlossenen logischen Transaktion.
GET TRANSACTION DATA	Lesen von Transaktionsdaten, die mit einem vorhergegangenen END TRANSACTION-Statement gespeichert wurden.
RETRY	Erneuter Versuch, einen Datensatz zu lesen, der vorher von einem anderen Benutzer benutzt wurde.
AT START OF DATA	Ausführen von Statements, wenn in einer Schleife der erste Datensatz verarbeitet wird.
AT END OF DATA	Ausführen von Statements, nachdem in einer Schleife der letzte Datensatz verarbeitet wurde.
AT BREAK	Ausführen von Statements bei einem Wertwechsel in einem bestimmten Feld (Gruppenwechsel).
BEFORE BREAK PROCESSING	Ausführen von Statements vor einer Gruppenwechsel-Verarbeitung.
PERFORM BREAK PROCESSING	Sofortiges Ausführen einer Gruppenwechsel-Verarbeitung.

Natural-SQL-Statements

Zusätzlich zu den Natural-DML-Statements bietet Natural auch SQL-Statements zur Benutzung in Natural-Programmen, so dass SQL unmittelbar verwendet werden kann.

Folgende SQL-Statements sind verfügbar:

CALLDBPROC	Dient dazu, eine "Stored Procedure" des SQL-Datenbanksystems, mit dem Natural verbunden ist, aufzurufen.
COMMIT	Entspricht dem END TRANSACTION-Statement. Es markiert das Ende einer logischen Transaktion und bewirkt, dass alle während der Transaktion gesperrten Daten freigegeben werden. Alle Datenänderungen werden gespeichert und sind damit definitiv.
DELETE	Löscht Reihen aus einer Tabelle, ohne einen Cursor zu verwenden ("searched" DELETE), oder löscht Reihen aus einer Tabelle, auf die der Cursor zeigt ("positioned" DELETE).
INSERT	Fügt einer Tabelle eine oder mehrere neue Reihen hinzu.
PROCESS SQL	Dient dazu, mit SQL-Statements auf eine Datenbank zuzugreifen.
READ RESULT SET	Liest einen Result Set, der von einer mit einem vorhergehenden CALLDBPROC-Statement aufgerufenen "Stored Procedure" erzeugt wurde.
ROLLBACK	Entspricht dem Statement BACKOUT TRANSACTION. Es macht alle seit dem Beginn der letzten "Recovery Unit" ausgeführten Datenbankänderungen rückgängig.
SELECT	Gemäß der Standard-SQL-Funktionalität unterstützt Natural sowohl das cursor-orientierte SELECT, mit dem eine beliebige Anzahl von Reihen gelesen werden kann, als auch das nicht cursor-orientierte "singleton" SELECT, das maximal eine Reihe liest.
UPDATE	Führt UPDATE-Operationen auf Reihen in einer Tabelle aus, ohne einen Cursor zu verwenden ("searched" UPDATE), oder auf Spalten in einer Reihe, in der ein Cursor positioniert ist ("positioned" UPDATE).

Arithmetische Funktionen und Datenzuweisungen

Die folgenden Statements werden verwendet, um arithmetische Operationen sowie Datenzuweisungen durchzuführen:

COMPUTE	Rechenoperationen ausführen oder Feldern Werte zuweisen.
ADD	Addieren von Operanden.
SUBTRACT	Subtrahieren von Operanden.
MULTIPLY	Multiplizieren von Operanden.
DIVIDE	Dividieren eines Operanden durch einen anderen.
EXAMINE TRANSLATE	Konvertiert die in einem Feld enthaltenen Zeichen in Groß- oder Kleinbuchstaben oder in andere Zeichen.
MOVE	Übertragen eines Operandenwertes in ein Feld oder mehrere Felder.
MOVE ALL	Übertragen sämtlicher Werte einer bestimmten Größe in ein anderes Feld.
COMPRESS	Aneinanderreihen mehrerer Feldwerte in einem Feld.
SEPARATE	Aufteilen eines Feldwertes in zwei oder mehr Felder.
EXAMINE	Absuchen eines Feldes nach einem bestimmten Wert und anschließend Ersetzen des Wertes und/oder Zählen, wie oft der Wert vorkommt.
RESET	Zurücksetzen eines Feldwertes auf Null (numerisches Feld) bzw. Leerwert (alphanumerisches Feld) oder auf einen Ausgangswert.

Schleifenverarbeitung

Die folgenden Statements werden in Verbindung mit der Ausführung von Verarbeitungsschleifen verwendet:

ESCAPE	Ausführung einer Verarbeitungsschleife abbrechen.
FOR	Initiieren einer Verarbeitungsschleife und Steuerung der Anzahl der Schleifendurchläufe.
REPEAT	Initiieren einer Verarbeitungsschleife (und Beenden in Abhängigkeit von einer bestimmten Bedingung).
SORT	Sortieren von Datensätzen.

Erstellen von Ausgabe-Reports

Die folgenden Statements werden bei der Erzeugung von Ausgabe-Reports verwendet:

FORMAT	Spezifizieren von Ausgabe-Parametern.
DISPLAY	Ausgabe von Feldwerten in Spalten untereinander.
WRITE / PRINT	Ausgabe von Feldwerten ohne Spalteneinteilung.
WRITE TITLE	Überschreiben einer Standard-Seitenüberschrift mit einer eigenen Seitenüberschrift.
WRITE TRAILER	Ausgabe eines Fußzeilentextes, der auf jeder Ausgabeseite erscheinen soll.
AT TOP OF PAGE	Spezifizieren der Verarbeitung, die beim Beginn einer neuen Ausgabeseite ausgeführt werden soll.
AT END OF PAGE	Spezifizieren der Verarbeitung, die beim Erreichen des Endes einer Ausgabeseite ausgeführt werden soll.
SKIP	Generieren von Leerzeilen in der Ausgabe.
EJECT	Seitenvorschub ohne Titel und Überschriften.
NEWPAGE	Seitenvorschub mit Titel und Überschriften.
SUSPEND IDENTICAL SUPPRESS	Aussetzen der "Identical Suppress"-Bedingung für einen einzelnen Datensatz.
DEFINE PRINTER	Bestimmen des Druckers oder logischen Zielorts, an dem ein Report ausgegeben werden soll.
CLOSE PRINTER	Schließen eines Druckers.

Bildschirmgenerierung für interaktive Verarbeitung

Die folgenden Statements werden in Verbindung mit der Verwendung von Bildschirmmasken (Maps) bei interaktiver Datenverarbeitung benutzt:

INPUT	Erstellen einer formatierten Map zur Datenausgabe/-eingabe.
REINPUT	Erneutes Ausführen eines INPUT-Statements (falls die auf das INPUT-Statement erfolgte Dateneingabe fehlerhaft war).
DEFINE WINDOW	Größe, Position und Attribute eines Windows festlegen.
SET WINDOW	Aktivieren und Deaktivieren eines Windows.
PROCESS PAGE	Erstellen eines Daten-Mappings auf einen Web Rich GUI-Schirm.
PROCESS PAGE USING	Ausführen einer GUI I/O-Verarbeitung unter Verwendung eines aus einem Seiten-Layout erzeugten Objekts vom Typ Adapter.
PROCESS PAGE UPDATE	Erneutes Ausführen eines PROCESS PAGE-Statements.
PROCESS PAGE MODAL	Initiieren eines Verarbeitungsblocks und Kontrolle des Lebenszyklus eines Rich GUI Window.

Verarbeitung logischer Bedingungen

Mit den folgenden Statements wird die Ausführung von Statements in Abhängigkeit von Bedingungen gesteuert, die während der Ausführung eines Natural-Programms auftreten:

IF	Ausführen von Statements aufgrund einer logischen Bedingung.
IF SELECTION	Prüfen, ob in einer Reihe von alphanumerischen Feldern genau eins einen Wert enthält.
DECIDE FOR	Ausführen von Statements aufgrund von logischen Bedingungen.
DECIDE ON	Ausführen von Statements aufgrund des Inhaltes einer Variablen.

Aufrufen von Programmen und Unterprogrammen

Die folgenden Statements werden zum Aufrufen von Programmen und Unterprogrammen verwendet:

CALL	Aufrufen eines Nicht-Natural-Programms von einem Natural-Programm aus.
CALLNAT	Aufrufen eines Natural-Subprogramms.
CALL FILE	Aufrufen eines Nicht-Natural-Programms, um einen Datensatz von einer Nicht-Adabas-Datei zu lesen.
CALL LOOP	Generieren einer Verarbeitungsschleife, die den Aufruf eines Nicht-Natural-Programms beinhaltet.
DEFINE SUBROUTINE	Definieren einer Natural-Subroutine.
ESCAPE	Abbrechen eines Unterprogramms.
FETCH	Aufrufen eines Natural-Programms.
PERFORM	Aufrufen einer Natural-Subroutine.
PROCESS COMMAND	Aufrufen eines Kommando-Prozessors.
RUN	Kompilieren und Ausführen eines Source-Programms.
Function Call	Aufrufen eines Natural-Objekts vom Typ Function.

Beenden von Programmen und Sessions

Die folgenden Natural-Statements dienen zum Beenden der Ausführung einer Anwendung oder der Natural-Session.

STOP	Beendet die Ausführung einer Anwendung.
TERMINATE	Beendet die Natural-Session.

Verarbeitung von Arbeitsdateien

Die folgenden Natural-Statements werden verwendet, um Daten auf eine physisch-sequentielle (nicht-Adabas) Arbeitsdatei zu schreiben bzw. von dieser zu lesen:

WRITE WORK FILE	Schreibt Daten auf eine Arbeitsdatei.
READ WORK FILE	Liest Daten von einer Arbeitsdatei.
CLOSE WORK FILE	Schließt eine Arbeitsdatei.
DEFINE WORK FILE	Weist einer Arbeitsdatei einen Dateinamen zu.

Komponentenbasierte Programmierung

Folgende Natural-Statements werden zur komponentenbasierten Programmierung verwendet:

DEFINE CLASS	Gibt innerhalb eines Natural-Klassenmoduls eine Klasse an.
CREATE OBJECT	Erstellt ein Objekt (auch bekannt als "Instanz") einer gegebenen Klasse.
SEND METHOD	Ruft eine Methode eines Objekts auf.
INTERFACE	Definiert eine Schnittstelle (eine Sammlung von Methoden und Eigenschaften) für eine bestimmte Funktion einer Klasse.
METHOD	Weist außerhalb einer Schnittstellendefinition ein Unterprogramm als Implementierung einer Methode zu.
PROPERTY	Weist außerhalb einer Schnittstellendefinition eine Objektdaten-Variable als Implementierung einer Eigenschaft zu.

Ereignisgesteuerte Programmierung

Die Natural-Statements werden zur ereignisgesteuerten Programmierung verwendet:

OPEN DIALOG	Öffnen eines Dialogs.
CLOSE DIALOG	Schließen eines Dialogs.
SEND EVENT	Auslösen eines benutzerdefinierten Ereignisses.
PROCESS GUI	Ausführen einer Standardprozedur in einer ereignisgesteuerten Anwendung.

Speicherverwaltung für dynamische Variablen/X-Arrays

EXPAND	Erweitert den zugewiesenen Speicher für dynamische Variablen auf eine gegebene Größe bzw. erweitert die Anzahl der Ausprägungen eines X-arrays.
REDUCE	Reduziert die Größe einer dynamischen Variablen bzw. die Anzahl der Ausprägungen eines X-arrays.
RESIZE	Passt die Größe einer dynamischen Variablen bzw. die Anzahl der Ausprägungen eines X-arrays an.

Natural Remote Procedure Call

OPEN CONVERSATION	Ermöglicht dem Client, eine Konversation zu öffnen und die Remote-Subprogramme anzugeben, die an der Konversation beteiligt sein sollen.
CLOSE CONVERSATION	Ermöglicht dem Client, eine Konversation zu schließen. Sie können die aktuelle Konversation, eine andere offene Konversation oder alle offenen Konversationen schließen.
DEFINE DATA CONTEXT	Dient zur Definition von als Kontext-Variablen bekannte Variablen, die für mehrere Subprogramme auf einem externen (Remote-)Rechner innerhalb einer Konversation zur Verfügung stehen sollen, ohne dass Sie die Variablen explizit als Parameter mit den entsprechenden CALLNAT-Statements übergeben müssen.

Internet und XML

PARSE	Gestattet es, XML-Dokumente von einem Natural-Programm aus zu analysieren und die darin enthaltenen Informationen der weiteren Verarbeitung zur Verfügung zu stellen.
REQUEST DOCUMENT	Ermöglicht den Zugang zu einem externen System.

Sonstige Statements

DEFINE DATA	Definiert die Datenelemente, die in einem Natural-Programm oder -Unterprogramm verwendet werden sollen.
END	Zeigt das Ende des Sourcecodes eines Natural-Programms bzw. -Unterprogramms an.
INCLUDE	Einfügen von Natural-Copycode während der Kompilierung.
ON ERROR	Abfangen von Laufzeitfehlern, die normalerweise eine Fehlermeldung und den Abbruch des ausgeführten Programms bewirken würden.
PROCESS REPORTER	Kommunikation mit dem Natural Reporter aus einem Programm heraus, um diesen anzuweisen, eine bestimmte Aktion auszuführen.
RELEASE	Löschen aller im Natural-Stack gehaltenen Daten; Freigabe aller Daten, die über eine RETAIN-Klausel in einem FIND-Statement gehalten wurden; Zurücksetzen von globalen Variablen auf die ursprünglichen Werte.
SET CONTROL	Ausführen eines Natural-Terminalkommandos aus einem Natural-Programm heraus.
SET KEY	Zuweisen von Funktionen zu Funktionstasten.
SET TIME	Setzen eines zeitlichen Bezugspunkts für eine *TIMD-Systemvariable.
STACK	Zwischenlagern von Daten bzw. Kommandos im Natural-Stack.

Reporting Mode-Statements

Die folgenden Statements gelten nur für den Reporting Mode:

CLOSE LOOP	Schließt eine Verarbeitungsschleife.
DO/DOEND	Spezifikation einer Gruppe von Statements, die auf der Grundlage einer logischen Bedingung ausgeführt werden sollen.
OBTAIN	Bewirkt, dass ein Feld oder mehrere Felder von einer Datei gelesen werden.
REDEFINE	Redefiniert ein Feld.

Die folgenden Statements können sowohl im Structured Mode als auch im Reporting Mode benutzt werden, allerdings ist die Statement-Struktur und bei einigen Statements auch die Funktionalität anders:

AT START OF DATA	Gibt Statements an, die ausgeführt werden sollen, wenn der erste in einer Reihe von Datensätzen in einer Verarbeitungsschleife verarbeitet wird.
AT END OF DATA	Gibt Statements an, die ausgeführt werden sollen, nachdem der letzte einer Reihe von Datensätzen in einer Verarbeitungsschleife verarbeitet wurde.
AT BREAK	Gibt Statements an, die ausgeführt werden sollen, wenn sich der Wert eines Kontrollfeldes ändert (Gruppenwechselverarbeitung).
AT TOP OF PAGE	Gibt eine Verarbeitung an, die ausgeführt werden soll, wenn eine neue Ausgabeseite gestartet wird.
AT END OF PAGE	Gibt eine Verarbeitung an, die ausgeführt werden soll, wenn das Ende einer Ausgabeseite erreicht ist.
BEFORE BREAK PROCESSING	Gibt Statements an, die vor Durchführung des Gruppenwechsels ausgeführt werden sollen.
CALL LOOP	Erzeugt eine Verarbeitungsschleife, die einen Aufruf an ein Nicht-Natural-Programm enthält.
CALL FILE	Ruft ein Nicht-Natural-Programm auf, um einen Datensatz aus einer Nicht-Adabas-Datei zu lesen.
COMPUTE	Führt arithmetische Operationen durch oder weist Feldern Werte zu.
DEFINE SUBROUTINE	Definiert eine Natural-Subroutine.
ESCAPE	Hält die Ausführung einer Verarbeitungsschleife an.
FIND	Wählt nach Benutzerkriterien Datensätze aus einer Datenbank- Datei aus.
GET SAME	Liest den gerade verarbeiteten Datensatz wieder ein.
HISTOGRAM	Liest die Werte eines Datenbankfeldes.
IF	Führt Statements in Abhängigkeit von einer logischen Bedingung aus.
IF SELECTION	Prüft, ob in einer Reihe von alphanumerischen Feldern wirklich nur eines einen Wert enthält.
ON ERROR	Fängt Laufzeitfehler ab, die ansonsten zu einer Natural-Fehlermeldung führen würden und beendet dann das Natural-Programm.
READ	Liest eine Datenbank-Datei in physischer oder logischer Reihenfolge der Datensätze.
READ WORK FILE	Liest Daten aus einer Arbeitsdatei.
REPEAT	Initiiert eine Verarbeitungsschleife (und beendet sie abhängig von einer angegebenen Bedingung).
SORT	Sortiert Datensätze.
STORE	Fügt der Datenbank einen neuen Datensatz hinzu.
UPDATE	Aktualisiert einen Datensatz in der Datenbank.