

# DELETE

`DELETE [RECORD] [IN] [STATEMENT] [( r)]`

This chapter covers the following topics:

- Function
- Restriction
- Syntax Description
- Database-Specific Considerations
- Examples

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

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

Belongs to Function Group: *Database Access and Update*

---

## Function

The DELETE statement is used to delete a record from a database.

## Hold Status

The use of the DELETE statement causes each record selected in the corresponding FIND or READ statement to be placed in hold status.

Record hold logic is explained in the section *Database Update - Transaction Processing* (in the *Programming Guide*).

## Restriction

A DELETE statement cannot be specified in the same statement line as a FIND, READ, or GET statement.

## Syntax Description

Syntax Element	Description
( <i>r</i> )	<p><b>Statement Reference:</b></p> <p>The notation (<i>r</i>) is used to reference the statement which was used to select/read the record to be deleted.</p> <p>If no statement reference is specified, the DELETE statement will reference the innermost active processing loop in which a database record was selected/read.</p>

## Database-Specific Considerations

<b>SQL Databases</b>	<p>The DELETE statement is used to delete a row from the database table. It corresponds with the SQL statement DELETE WHERE CURRENT OF CURSOR-NAME, that is, only the row which was read last can be deleted.</p> <p>With most SQL databases, a row that was read with a FIND SORTED BY or READ LOGICAL statement cannot be deleted.</p>
<b>XML Databases</b>	<p>The DELETE statement is used to delete an XML object from a database. For XML databases, this implies that only the record which was read last can be deleted.</p>

## Examples

- Example 1
- Example 2

### Example 1

In this example, all records with the name ALDEN are deleted.

```

** Example 'DELEX1': DELETE
**
**
CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
*
FIND EMPLOY-VIEW WITH NAME = 'ALDEN'
  /*
  DELETE
  END TRANSACTION
  /*
  AT END OF DATA
    WRITE NOTITLE *NUMBER 'RECORDS DELETED'
  END-ENDDATA
END-FIND
END

```

## Example 2

If no records are found in the VEHICLES file for the person named ALDEN, the EMPLOYEE record for ALDEN is deleted.

```
** Example 'DELEX2': DELETE
**
**
CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
1 VEHIC-VIEW VIEW OF VEHICLES
  2 PERSONNEL-ID
END-DEFINE
*
EMPL. FIND EMPLOY-VIEW WITH NAME = 'ALDEN'
/*
  VEHIC. FIND VEHIC-VIEW WITH PERSONNEL-ID = PERSONNEL-ID (EMPL.)
  IF NO RECORDS FOUND
    /*
      DELETE (EMPL.)
    /*
  END TRANSACTION
END-NOREC
END-FIND
/*
END-FIND
END
```