read ::= READ FIXED filename-expression [ BATCH SIZE integer-expression ]  
[ HEADER header-trailer-spec-1 [ HEADER header-trailer-spec-2 … ] ]  
{ DATA IGNORE | DATA data-spec-1 [ DATA data-spec-2 … ] }  
[ TRAILER header-trailer-spec-1 [ TRAILER header-trailer-spec-2 … ] ]

header-trailer-spec ::= IGNORE | header-trailer-fields

header-trailer-fields ::= header-trailer-field-1 [ , header-trailer-field-2 … ]

header-trailer-field ::= columns { validation-spec | keep-spec }

columns ::= [ COLUMNS ] start-expression end-expression

validation-spec ::= CONTAIN validation-expression

keep-spec ::= KEEP keep-variable

data-spec ::= data-fields INTO data-target

data-fields ::= data-field-1 [ , data-field-2 … ]

data-field ::= columns [ validation-spec | KEEP [ JOIN ] ] | LINE NUMBER [ KEEP [ JOIN ] ]

**Notes:**

If any data-target starts with SQL, it must be terminated by END SQL unless the entire task is otherwise terminated immediately with END TASK. This can only happen if there are no TRAILER clauses.

READ FIXED reads a fixed-record-layout file. The file may optionally have one or more header records, followed by data records in groups of one or more, optionally followed by one or more trailer records.

Each HEADER clause results in a leading record of the file being read. If IGNORE is specified, the record contents are ignored. Otherwise the contents are validated and/or retained in variables according to header-trailer-fields as described below.

For each COLUMNS clause, the start-expression and end-expression are integer expressions. They are evaluated and the results specify a start column number and end column number, with the first column being number 1.

For each CONTAIN clause, the validation-expression is a string expression. It is evaluated and the actual contents of the field spanning the columns of the preceding COLUMNS clause are compared to the evaluated expression. If they do not match, the READ FIXED task fails, except for special handling of the first trailer record as described below.

For each header-trailer-field KEEP clause, the actual contents of the header or trailer field spanning the columns of the preceding COLUMNS clause are stored to the indicated variable.

After the header records have been read, remaining records are either data records or trailer records.

The first TRAILER clause must include at least one CONTAIN clause. When each non-header record is read, it is compared to the CONTAIN clauses of the first TRAILER clause. If the record matches all the CONTAIN clauses, it is deemed to be the first trailer record. If it does not match, it is a data record.

The remainder of the file is scanned and data records are processed according to the DATA clauses until the first trailer record is found.

If the first DATA clause specifies IGNORE, all data records are skipped until the first trailer record is found.

Otherwise, the first DATA clause results in a single data record being read. Each data-field may specify a column span followed by a CONTAIN clause, the bare keyword KEEP, or neither. If neither is specified, then KEEP is implied.

A data field may also specify the keyword phrase LINE NUMBER optionally followed by the keyword KEEP optionally followed by JOIN. KEEP is implied if omitted.

At least one KEEP must be followed by JOIN if and only if this DATA clause is followed by one or more other DATA clauses.

For each data record matching the first DATA clause, each field with a CONTAIN clause is validated the same as for header and trailer records. Each field with an explicit or implicit KEEP clause is written to the data-target. For a COLUMNS field, the contents of those columns are written. For a LINE NUMBER field, the line number of the data record is written.

If present, the second and subsequent DATA clauses must each include at least one CONTAIN clause. Each may result in zero or more records being read.

After a record is matched to the first DATA clause, the next record is tested against the CONTAIN clauses of the second DATA clause. If it matches, the JOIN fields of the first data record followed by the KEEP fields of the next data record are written to data-target-2.

In the second DATA clause, at least one KEEP may be followed by JOIN if and only if this DATA clause is followed by one or more other DATA clauses. In that case, after a record Is matched to the second DATA clause, the next record is tested against the CONTAIN clauses of the third DATA clause. If it matches, the JOIN fields of the first data record followed by the JOIN fields of the second data record, if any, followed by the KEEP fields of the next data record are written to data-target-3. This nesting continues for any further DATA clauses.

In general, after a data record is matched to a first, second, or deeper level DATA clause, another data record is read and compared against the CONTAIN clauses of the DATA clause at next deeper level, if any. If there is no deeper level or the record does not match the deeper level, it is tested against the CONTAIN clauses of the DATA clause at the same level. If it does not match, the record is tested against the CONTAIN clauses of the DATA clause at each next higher level in turn until a match is found. If no match is found, the data records are considered to be exhausted and the first trailer record is expected.

After the first trailer record is found, remaining records are additional trailer records. There must be a TRAILER clause for each trailer record. If there are fewer or more trailer records than TRAILER clauses, the READ FIXED task fails.