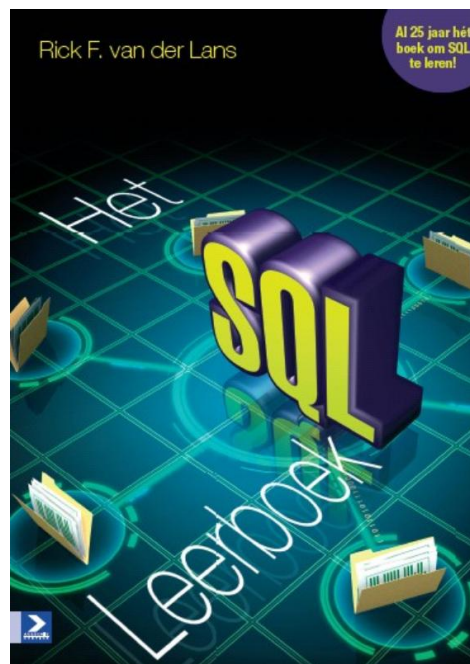


Het SQL Leerboek – zevende editie

Syntaxis van SQL



Auteur: Rick F. van der Lans

Versie: 1.0

Datum: Februari 2012

Alle rechten voorbehouden. Alle auteursrechten en databankrechten ten aanzien van deze uitgave worden uitdrukkelijk voorbehouden. Deze rechten berusten bij de auteur.

Behoudens de in of krachtens de Auteurswet 1912 gestelde uitzonderingen, mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voorzover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16 h Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (postbus 3060, 2130 KB Hoofddorp, www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich te wenden tot de Stichting PRO (Stichting Publicatie- en Reproductierechten Organisatie, Postbus 3060, 2130 KB Hoofddorp, www.cedar.nl/pro). Voor het overnemen van een gedeelte van deze uitgave ten behoeve van commerciële doeleinden dient men zich te wenden tot de uitgever.

Hoewel aan de totstandkoming van deze uitgave de uiterste zorg is besteed, kan voor de afwezigheid van eventuele (druk)fouten en onvolledigheden niet worden ingestaan en aanvaarden de auteur(s), redacteur(en) en uitgever deswege geen aansprakelijkheid voor de gevolgen van eventueel voorkomende fouten en onvolledigheden.

Syntaxis van SQL

1.1 Inleiding

Dit document bevat een toelichting op de notatiewijze die we in dit hoofdstuk gebruiken voor het definiëren van de instructies, een lijst met gereserveerde woorden en de definities van alle SQL-instructies die we in dit boek behandeld hebben.

De definities in dit document kunnen afwijken van die in de voorgaande hoofdstukken. De hoofdreden hiervoor is dat in de hoofdstukken sommige instructies en begrippen stap voor stap worden uitgelegd. Om niet te veel details tegelijk te tonen, zijn daarom soms versimpelde versies van de definities gebruikt. Dit document bevat de volledige definities.

1.2 De BNF-notatie

In dit document en in de diverse hoofdstukken gebruiken we een formele notatiewijze voor de beschrijving van de syntaxis van alle SQL-instructies en de basisbegrippen. Deze notatiewijze is een variant op de zogenaamde *Backus Naur Form* (BNF) die genoemd is naar John Backus en Peter Naur. De betekenis van de metasymbolen die we gebruiken is gebaseerd op die van de metasymbolen in de SQL-standaard.

Bij BNF wordt een taal ontleed in zogenaamde *substitutieregels*, ofwel *productieregels*, bestaande uit reeksen symbolen. In elke productieregel wordt een *symbool* gedefinieerd. Een symbool kan bijvoorbeeld een SQL-instructie, een tabelnaam of een puntkomma zijn. Een speciaal soort symbool is het *terminale symbool*. Alle symbolen, behalve de terminale symbolen, zijn met een productieregel in termen van andere symbolen gedefinieerd. Terminale symbolen zijn bijvoorbeeld het woord CREATE en de puntkomma.

Een productieregel kan vergeleken worden met een definitie van een begrip, waarbij in de definitie begrippen worden gebruikt die op hun beurt ergens anders gedefinieerd zijn. Een begrip komt in dit geval overeen met een symbool.

De volgende *metasymbolen* maken geen deel uit van de taal SQL, maar behoren bij de notatietechniek.

- < >
- ::=
- |
- []
- ...
- { }
- ;
- "

Elk van deze symbolen zullen we eerst toelichten.

De symbolen < en >

Niet-terminale symbolen worden met de punthaken < > aangegeven. Voor elk niet-terminaal symbool bestaat een productieregel. We zullen de namen van de niet-terminale symbolen in kleine letters weergeven. De <select-instructie> en de <tabelreferentie> zijn twee voorbeelden van niet-terminale symbolen.

Het symbool ::=

Met het ::= symbool wordt in een productieregel het niet-terminale symbool (links) dat gedefinieerd wordt, gescheiden van de definitie (rechts). Het symbool ::= moet gelezen worden als 'is gedefinieerd als.' Hieronder volgt als voorbeeld de productieregel voor de DROP INDEX-instructie:

```
<drop-index-instructie> ::= DROP INDEX <indexnaam>
```

Toelichting: De DROP INDEX-instructie bestaat dus uit de terminale symbolen DROP en INDEX gevolgd door het niet-terminale symbool indexnaam. Voor <indexnaam> behoort dan ook een productieregel te bestaan.

Het symbool |

Met het symbool | worden alternatieven aangegeven. Hieronder volgt als voorbeeld de productieregel voor het begrip <teken>:

```
<teken> ::= <cijfer> | <letter> | <speciaal-symbool> | ''
```

Toelichting: We moeten dit lezen als: een teken is een cijfer, een letter, een speciaal symbool, of twee aanhangstekens; een van de vier.

De symbolen [en]

Datgene wat tussen de vierkante haken [en] staat *mag* gebruikt worden. Hieronder volgt als voorbeeld de productieregel voor de ROLLBACK-instructie:

```
<rollback-instructie> ::= ROLLBACK [ WORK ]
```

Toelichting: Een ROLLBACK-instructie bestaat altijd uit het woord ROLLBACK en wordt eventueel gevolgd door het woord WORK.

Het symbool . . .

De drie punten geven aan wat een of meer malen herhaald mag worden. Hieronder volgt als voorbeeld de productieregel voor een geheel-getal:

```
<geheel-getal> ::= <cijfer>...
```

Toelichting: Een geheel-getal bestaat uit een reeks cijfers (minimaal één).

De drie punten gecombineerd met de vierkante haken geven ons de mogelijkheid om aan te geven dat een element nul, één of meer malen mag voorkomen:

```
<from-component> ::=
  FROM <tabelreferentie> [ , <tabelreferentie> ]...
```

Toelichting: Een FROM-component begint met het terminale symbool FROM en wordt gevolgd door minstens één tabelreferentie. Achter deze tabelreferentie kan eventueel een lijst van elementen volgen, waarbij elk element uit een komma bestaat gevolgd door een tabelreferentie. Vergeet niet dat de komma deel uitmaakt van SQL en niet van de notatietechniek.

De symbolen { en }

Alle symbolen tussen accolades vormen samen een groep. Accolades worden bijvoorbeeld samen met het symbool | gebruikt om aan te geven wat precies de alternatieven zijn. Hieronder volgt als voorbeeld een deel van de productieregel voor de float-constante:

```
<float-constante> ::=
  <mantisse> { E | e } <exponent>
```

Toelichting: Een float-constante begint met een mantisse en eindigt met een exponent. Daartussenin kan de hoofdletter E of een kleine letter e gebruikt worden. Maar een van de twee moet gebruikt worden.

Als we de accolades combineren met drie punten, kunnen we aangeven dat een element een of meer malen gebruikt moet worden. Dus in de productieregel A { B }... moet eerst het element A gebruikt worden, gevolgd door één of meer elementen B.

Het symbool ;

Sommige symbolen hebben dezelfde definitie. In plaats van deze te herhalen, kan de puntkomma gebruikt worden om de definities te verkorten. De volgende definitie

```
<character-constante> ;
<varchar-constante> ;
<long-varchar-constante> ::= <tekenreeks>
```

is gelijkwaardig aan de onderstaande drie definities

```
<character-constante> ::= <tekenreeks>
<varchar-constante> ::= <tekenreeks>
<long-varchar-constante> ::= <tekenreeks>
```

Het symbool "

Een klein aantal metasymbolen is onderdeel van bepaalde SQL-instructies zelf, zoals het symbool ". Om misverstanden te voorkomen, zijn deze symbolen omsloten met dubbele aanhalingstekens. Dit betekent onder andere dat het symbool " dat binnen SQL gebruikt wordt, in de productieregels wordt weergegeven als ""

Additionele opmerkingen:

- Wat in hoofdletters staat alsmede alle symbolen die geen deel uitmaken van de notatietechniek, moeten precies zo overgenomen worden.
- De volgorde van de symbolen in het rechterdeel van de productieregel staat vast.
- Spaties in productieregels hebben geen betekenis. We hebben ze meestal toegevoegd om de leesbaarheid van de productieregels te verbeteren. De volgende twee productieregels zijn gelijk:

```
<alfanumerieke-constante> ::= ' [ <teken>... ] '
```

en

```
<alfanumerieke-constante> ::= '[<teken>...]'
```

1.3 Gereserveerde woorden in SQL en MySQL

Elke programmeertaal en/of databasetaal (dus ook SQL) kent zogenaamde *gereserveerde woorden* ofwel *keywords*. SQL kent bijvoorbeeld SELECT en CREATE. Deze gereserveerde woorden mogen bij de meeste SQL-producten niet gebruikt worden als namen voor database-objecten zoals tabellen, kolommen, views en gebruikers. Elk product heeft een eigen verzameling gereserveerde woorden (uiteraard hebben twee SQL-producten veel overeenkomstige gereserveerde woorden). Voor welke dit zijn verwijzen we naar de respectievelijke documentatie. Hieronder staat de lijst met gereserveerde woorden zoals ze in de SQL3-standaard zijn gedefinieerd.

- ABSOLUTE, ACTION, ADD, ALL, ALLOCATE, ALTER, AND, ANY, ARE, AS, ASC, ASSERTION, AT, AUTHORIZATION, AVG
- BEGIN, BETWEEN, BIT, BIT_LENGTH, BOTH, BY
- CASCADE, CASCADED, CASE, CAST, CATALOG, CHAR, CHARACTER, CHAR_LENGTH, CHARACTER_LENGTH, CHECK, CLOSE, COALESCE, COLLATE, COLLATION, COLUMN, COMMIT, CONNECT, CONNECTION, CONSTRAINT, CONSTRAINTS, CONTINUE, CONVERT, CORRESPONDING, COUNT, CREATE, CROSS, CURRENT, CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP, CURRENT_USER, CURSOR
- DATE, DAY, DEALLOCATE, DEC, DECIMAL, DECLARE, DEFAULT, DEFERRABLE, DEFERRED, DELETE, DESC, DESCRIBE, DESCRIPTOR, DIAGNOSTICS, DISCONNECT, DISTINCT, DOMAIN, DOUBLE, DROP
- ELSE, END, END-EXEC, ESCAPE, EXCEPT, EXCEPTION, EXEC, EXECUTE, EXISTS, EXTERNAL, EXTRACT
- FALSE, FETCH, FIRST, FLOAT, FOR, FOREIGN, FOUND, FROM, FULL
- GET, GLOBAL, GO, GOTO, GRANT, GROUP
- HAVING, HOUR
- IDENTITY, IMMEDIATE, IN, INDICATOR, INITIALLY, INNER, INPUT, INSENSITIVE, INSERT, INT, INTEGER, INTERSECT, INTERVAL, INTO, IS, ISOLATION
- JOIN
- KEY
- LANGUAGE, LAST, LEADING, LEFT, LEVEL, LIKE, LOCAL, LOWER
- MATCH, MAX, MIN, MINUTE, MODULE, MONTH

- NAMES, NATIONAL, NATURAL, NCHAR, NEXT, NO, NOT, NULL, NULLIF, NUMERIC
- OCTET_LENGTH OF, ON, ONLY, OPEN, OPTION, OR, ORDER, OUTER, OUTPUT, OVERLAPS
- PARTIAL, POSITION, PRECISION, PREPARE, PRESERVE, PRIMARY, PRIOR, PRIVILEGES, PROCEDURE, PUBLIC
- READ, REAL, REFERENCES, RELATIVE, RESTRICT, REVOKE, RIGHT, ROLLBACK, ROWS
- SCHEMA, SCROLL, SECOND, SECTION, SELECT, SESSION, SESSION_USER, SET, SIZE, SMALLINT, SOME, SQL, SQLCODE, SQLERROR, SQLSTATE, SUBSTRING, SUM, SYSTEM_USER
- TABLE, TEMPORARY, THEN, TIME, TIMESTAMP, TIMEZONE_HOUR, TIMEZONE_MINUTE, TO, TRAILING, TRANSACTION, TRANSLATE, TRANSLATION, TRIM, TRUE
- UNION, UNIQUE, UNKNOWN, UPDATE, UPPER, USAGE, USER, USING
- VALUE, VALUES, VARCHAR, VARYING, VIEW
- WHEN, WHENEVER, WHERE, WITH, WORK, WRITE
- YEAR
- ZONE

Hierna volgt de lijst van gereserveerde woorden voor MySQL. De woorden die reeds in de bovenstaande lijst voorkomen, zijn weggelaten:

- ANALYZE, ASENSITIVE
- BEFORE, BIGINT, BINARY, BLOB
- CALL, CHANGE, CONDITION
- DATABASE, DATABASES, DAY_HOUR, DAY_MICROSECOND, DAY_MINUTE, DAY_SECOND, DELAYED, DETERMINISTIC, DISTINCTROW, DIV, DUAL
- EACH, ELSEIF, ENCLOSED, ESCAPED, EXIT, EXPLAIN
- FLOAT4, FLOAT8, FORCE, FULLTEXT
- HIGH_PRIORITY, HOUR_MICROSECOND, HOUR_MINUTE, HOUR_SECOND
- IF, IGNORE, INDEX, INFILE, INOUT, INT1, INT2, INT3, INT4, INT8, ITERATE
- KEYS, KILL
- LABEL, LEAVE, LIMIT, LINES, LOAD, LOCALTIME, LOCALTIMESTAMP, LOCK, LONG, LONGBLOB, LONGTEXT, LOOP, LOW_PRIORITY
- MEDIUMBLOB, MEDIUMINT, MEDIUMTEXT, MIDDLEINT, MINUTE_MICROSECOND, MINUTE_SECOND, MOD, MODIFIES
- NO_WRITE_TO_BINLOG
- OPTIMIZE, OPTIONALLY, OUT, OUTFILE
- PURGE
- RAID0, READS, REGEXP, RELEASE, RENAME, REPEAT, REPLACE, REQUIRE, RETURN, RLIKE
- SCHEMAS, SECOND_MICROSECOND, SENSITIVE, SEPARATOR, SHOW, SONAME, SPATIAL, SPECIFIC, SQLEXCEPTION, SQLWARNING, SQL_BIG_RESULT, SQL_CALC_FOUND_ROWS, SQL_SMALL_RESULT, SSL, STARTING, STRAIGHT_JOIN
- TERMINATED, TINYBLOB, TINYINT, TINYTEXT, TRIGGER
- UNDO, UNLOCK, UNSIGNED, USE, UTC_DATE, UTC_TIME, UTC_TIMESTAMP
- VARBINARY, VARCHARACTER
- WHILE
- X509, XOR
- YEAR_MONTH
- ZEROFILL

We raden u tevens aan de volgende adviezen te volgen wat betreft de naamgeving van database-objecten:

- Vermijd het gebruik van woorden die uit één letter bestaan, ook al komen ze niet in de lijst voor.
- Vermijd het gebruik van woorden die als afkortingen gezien zouden kunnen worden van woorden uit de lijst. Gebruik dus niet DATA, want het woord DATABASE komt in de lijst voor.
- Vermijd het gebruik van verbuigingen van woorden uit de lijst zoals meervoudsvormen en werkwoordsvormen. Gebruik dus niet TABLES (meervoudsvorm van TABLE) en niet ORDERING (werkwoordsvorm van ORDER).

1.4 Syntaxis-definities van SQL-instructies

Deze paragraaf bevat de definities van alle SQL-instructies zoals ze in dit boek behandeld zijn. In enkele instructies worden bepaalde basisbegrippen gebruikt, zoals conditie en kolommenlijst. Als een begrip tot slechts één instructie behoort, is het in paragraaf 1.4.2 bij de instructie zelf opgenomen. Anders wordt het in paragraaf 1.4.3 behandeld. We beginnen met het opsommen van alle verschillende groepen SQL-instructies.

1.4.1 Groepen SQL-instructies

In paragraaf **Fout! Verwijzingsbron niet gevonden.** hebben we aangegeven dat de verzameling SQL-instructies in groepen te verdelen is, zoals DDL-, DML- en DCL-instructies. En in hoofdstuk **Fout! Verwijzingsbron niet gevonden.** hebben we nog eens een onderscheid gemaakt tussen executeerbare en niet-executeerbare SQL-instructies. In deze paragraaf geven we precies aan welke instructie tot welke groep behoort.

SQL-instructie

```
<sql-instructie> ::=
  <executeerbare-instructie> |
  <niet-executeerbare-instructie>
```

Executeerbare instructie

```
<executeerbare-instructie> ::=
  <declaratieve-instructie> |
  <procedurele-instructie>
```

Declaratieve instructie

```
<declaratieve-instructie> ::=
  <ddl-instructie> |
  <dml-instructie> |
  <dcl-instructie>
```


DDL-instructie

```

<ddl-instructie> ::=
  <alter-database-instructie>
  <create-sequence-instructie>
  <alter-sequence-instructie>
  <create-database-instructie>
  <create-function-instructie>
  <create-index-instructie>
  <create-procedure-instructie>
  <create-sequence-instructie>
  <create-table-instructie>
  <create-trigger-instructie>
  <create-view-instructie>
  <drop-database-instructie>
  <drop-function-instructie>
  <drop-index-instructie>
  <drop-procedure-instructie>
  <drop-sequence-instructie>
  <drop-table-instructie>
  <drop-trigger-instructie>
  <drop-view-instructie>
  <rename-table-instructie>

```

DML-instructie

```

<dml-instructie> ::=
  <call-instructie>
  <close-instructie>
  <commit-instructie>
  <delete-instructie>
  <execute-immediate-instructie>
  <fetch-instructie>
  <insert-instructie>
  <lock-table-instructie>
  <open-instructie>
  <rollback-instructie>
  <savepoint-instructie>
  <select-instructie>
  <select-into-instructie>
  <set-instructie>
  <set-transaction-instructie>
  <start-transaction-instructie>
  <update-instructie>

```

DCL-instructie

```

<dcl-instructie> ::=
  <alter-user-instructie>
  <create-role-instructie>
  <create-user-instructie>
  <drop-role-instructie>
  <drop-user-instructie>
  <grant-instructie>
  <revoke-instructie>

```

Niet-executeerbare instructie

```

<niet-executeerbare-instructie> ::=
  <begin-declare-instructie> |
  <declare-cursor-instructie> |
  <end-declare-instructie> |
  <include-instructie> |
  <whenever-instructie>

```

Procedurele instructie

```

<procedurele-instructie> ::=
  <begin-end-blok> |
  <call-instructie> |
  <close-instructie> |
  <declare-condition-instructie> |
  <declare-cursor-instructie> |
  <declare-handler-instructie> |
  <declare-variable-instructie> |
  <fetch-cursor-instructie> |
  <flow-control-instructie> |
  <open-cursor-instructie> |
  <set-instructie> |
  <return-instructie>

```

Flow-control-instructie

```

<flow-control-instructie> ::=
  <if-instructie> |
  <case-instructie> |
  <while-instructie> |
  <repeat-instructie> |
  <loop-instructie> |
  <leave-instructie> |
  <iterate-instructie>

```

1.4.2 Definities van SQL-instructies**Alter database-instructie**

```

<alter-database-instructie> ::=
  ALTER DATABASE [ <databasenaam> ]
  [ <database-optie>... ]

```

Alter sequence-instructie

```

<alter-sequence-instructie> ::=
  ALTER SEQUENCE [ <gebruikersnaam> . ] <sequence-naam>
    [ <sequence-optie>... ]

<sequence-optie> ::=
  RESTART [ WITH <integer-constante> ] |
  INCREMENT BY <integer-constante> |
  { MAXVALUE <integer-constante> | NOMAXVALUE } |
  { MINVALUE <integer-constante> | NOMINVALUE } |
  { CYCLE | NOCYCLE } |
  { ORDER | NOORDER } |
  { CACHE <integer-constante> | NOCACHE }

```

Alter table-instructie

```

<alter-table-instructie> ::=
  ALTER TABLE <tabelspecificatie> <tabelstructuurwijziging>

<tabelstructuurwijziging> ::=
  <tabelwijziging> |
  <kolomwijziging> |
  <integriteitsregelwijziging> |
  <indexwijziging>

<tabelwijziging> ::=
  RENAME [ TO | AS ] <tabelnaam> |
  CONVERT TO CHARACTER SET { <karaktersetnaam> | DEFAULT } |
  [ COLLATE <collation-naam> ]

<kolomwijziging> ::=
  ADD [ COLUMN ] <kolomdefinitie>
    [ FIRST | AFTER <kolomnaam> ] |
  ADD [ COLUMN ] <tabelschema> |
  DROP [ COLUMN ] <kolomnaam> [ RESTRICT | CASCADE ] |
  CHANGE [ COLUMN ] <kolomnaam> <kolomdefinitie>
    [ FIRST | AFTER <kolomnaam> ] |
  MODIFY [ COLUMN ] <kolomdefinitie>
    [ FIRST | AFTER <kolomnaam> ] |
  ALTER [ COLUMN ] { SET DEFAULT <expressie> | DROP DEFAULT }

<integriteitsregelwijziging> ::=
  ADD <tabel-integriteitsregel> |
  DROP PRIMARY KEY |
  DROP CONSTRAINT <constraintnaam>

<indexwijziging> ::=
  ADD [ <indextype> ] INDEX <indexnaam>
    ( <kolom-in-index> [ , <kolom-in-index> ]... )

```

Alter user-instructie

```

<alter-user-instructie> ::=
  ALTER USER <gebruikersnaam> IDENTIFIED BY <wachtwoord>

```

Begin declare-instructie

```
<begin-declare-instructie> ::=
  BEGIN DECLARE SECTION
```

Call-instructie

```
<call-instructie> ::=
  CALL [ <databasenaam> . ] <procedurenaam>
  ( [ <scalaire-expressie> [ , <scalaire-expressie> ]... ] )
```

Case-instructie

```
<case-instructie> ::=
  { CASE <scalaire-expressie>
    WHEN <scalaire-expressie> THEN <instructielijst>
    [ WHEN <scalaire-expressie> THEN <instructielijst> ]...
    [ ELSE <instructielijst> ]
  END CASE } |
  { CASE
    WHEN <conditie> THEN <instructielijst>
    [ WHEN <conditie> THEN <instructielijst> ]...
    [ ELSE <instructielijst>
  END CASE }
```

Close-instructie

```
<close-instructie> ::=
  CLOSE <cursornaam>
```

Commit-instructie

```
<commit-instructie> ::=
  COMMIT [ WORK ]
```

Create database-instructie

```
<create-database-instructie> ::=
  CREATE DATABASE <databasenaam> [ <database-optie>... ]
```

Create function-instructie

```

<create-function-instructie> ::=
  CREATE FUNCTION <stored-functionnaam>
    ( [ <parameterlijst-voor-functie> ] )
    RETURNS <datatype>
    <functiebody>

<parameterlijst-voor-functie> ::=
  <parameterspecificatie-voor-functie>
    [ , <parameterspecificatie-voor-functie> ]...

<parameterspecificatie-voor-functie> ::=
  <parameter> <datatype>

<functiebody> ::= <begin-end-blok>

```

Create index-instructie

```

<create-index-instructie> ::=
  CREATE [ <indextype> ] INDEX <indexnaam>
    ON <tabelspecificatie>
    ( <kolom-in-index> [ , <kolom-in-index> ]... )

```

Create procedure-instructie

```

<create-procedure-instructie> ::=
  CREATE PROCEDURE <procedurenaam>
    ( [ <parameterlijst-voor-procedure> ] )
    <procedurebody>

<parameterlijst-voor-procedure> ::=
  <parameterspecificatie-voor-procedure>
    [ , <parameterspecificatie-voor-procedure> ]...

<parameterspecificatie-voor-procedure> ::=
  [ IN | OUT | INOUT ] <parameter> <datatype>

<procedurebody> ::= <begin-end-blok>

```

Create role-instructie

```

<create-role-instructie> ::=
  CREATE ROLE <rolnaam>

```

Create sequence-instructie

```

<create-sequence-instructie> ::=
  CREATE SEQUENCE [ <gebruikersnaam> . ] <sequence-naam>
    [ <sequence-optie>... ]

<sequence-optie> ::=
  START WITH <integer-constante>
  INCREMENT BY <integer-constante>
  { MAXVALUE <integer-constante> | NOMAXVALUE }
  { MINVALUE <integer-constante> | NOMINVALUE }
  { CYCLE | NOCYCLE }
  { ORDER | NOORDER }
  { CACHE <integer-constante> | NOCACHE }

```

Create table-instructie

```

<create-table-instructie> ::=
  CREATE [ TEMPORARY ] TABLE
    <tabelspecificatie> <tabelstructuur>

<tabelstructuur> ::=
  LIKE <tabelspecificatie>
  ( LIKE <tabelspecificatie> )
  <tabelinhoud>
  <tabelschema> [ <tabelinhoud> ]

```

Create trigger-instructie

```

<create-trigger-instructie> ::=
  CREATE TRIGGER <triggernaam>
  <trigger-moment>
  <trigger-event>
  [ <trigger-conditie> ]
  <trigger-actie>

<trigger-moment> ::=
  BEFORE | AFTER | INSTEAD OF

<trigger-event> ::=
  { INSERT | DELETE | UPDATE [ OF <kolommenlijst> ] }
  { ON | OF | FROM | INTO } <tabelspecificatie>
  [ REFERENCING { OLD | NEW | OLD_TABLE | NEW_TABLE }
    AS <tabelnaam> ]
  FOR EACH { ROW | STATEMENT }

<trigger-conditie> ::=
  WHEN ( <conditie> )

<trigger-actie> ::= <begin-end-blok>

```

Create user-instructie

```
<create-user-instructie> ::=
  CREATE USER <gebruikersnaam> IDENTIFIED BY <wachtwoord>
```

Create view-instructie

```
<create-view-instructie> ::=
  CREATE [ OR REPLACE ] VIEW <viewnaam> [ <kolommenlijst> ] AS
  <tabelexpressie>
  [ WITH [ CASCADED | LOCAL ] CHECK OPTION ]
```

Declare condition-instructie

```
<declare-condition-instructie> ::=
  DECLARE <conditiennaam> CONDITION FOR
  { SQLSTATE [ VALUE ] <sqlstate-waarde> } | <mysql-error-code>
```

Declare cursor-instructie

```
<declare-cursor-instructie> ::=
  DECLARE [ INSENSITIVE ] [ SCROLL ] <cursornaam> CURSOR FOR
  <tabelexpressie>
  [ <for-component> ]
```

Declare handler-instructie

```
<declare-handler-instructie> ::=
  DECLARE <handler-type> HANDLER FOR <conditiewaardelijst>
  <procedurele-instructie>

<handler-type> ::=
  CONTINUE |
  EXIT |
  UNDO

<conditiewaardelijst> ::=
  <conditiewaarde> [ , <conditiewaarde> ]...

<conditiewaarde> ::=
  SQLSTATE [ VALUE ] <sqlstate-waarde> |
  <mysql-error-code> |
  SQLWARNING |
  NOT FOUND |
  SQLEXCEPTION |
  <conditiennaam>
```

Declare variable-instructie

```
<declare-variable-instructie> ::=
  DECLARE <lokale-variabelelijst> <datatype>
  [ DEFAULT <scalaire-expressie> ]
```

Delete-instructie

```
<delete-instructie> ::=
DELETE
FROM <tabelreferentie>
[ WHERE { <conditie> | CURRENT OF <cursornaam> } ]
```

Drop database-instructie

```
<drop-database-instructie> ::=
DROP DATABASE <databasenaam>
```

Drop function-instructie

```
<drop-function-instructie> ::=
DROP FUNCTION [ <databasenaam> . ] <stored-functionnaam>
```

Drop index-instructie

```
<drop-index-instructie> ::=
DROP INDEX <indexnaam>
```

Drop procedure-instructie

```
<drop-procedure-instructie> ::=
DROP PROCEDURE [ <databasenaam> . ] <procedurenaam>
```

Drop role-instructie

```
<drop-role-instructie> ::=
DROP ROLE <rolnaam>
```

Drop sequence-instructie

```
<drop-sequence-instructie> ::=
DROP SEQUENCE [ <gebruikersnaam> . ] <sequence-naam>
```

Drop table-instructie

```
<drop-table-instructie> ::=
DROP TABLE <tabelspecificatie>
```

Drop trigger-instructie

```
<drop-trigger-instructie> ::=
DROP TRIGGER [ <tabelnaam> . ] <triggernaam>
```


Drop user-instructie

```
<drop-user-instructie> ::=  
  DROP USER <gebruikersnaam>
```

Drop view-instructie

```
<drop-view-instructie> ::=  
  DROP VIEW <tabelspecificatie>
```

End declare-instructie

```
<end-declare-instructie> ::=  
  END DECLARE SECTION
```

Execute immediate-instructie

```
<execute-immediate-instructie> ::=  
  EXECUTE IMMEDIATE <host-variabele>
```

Fetch-instructie

```
<fetch-instructie> ::=  
  FETCH [ <richting> ] <cursornaam>  
  INTO <host-variabelelijst>
```

<richting> ::=
 NEXT | PRIOR | FIRST | LAST |
 ABSOLUTE <geheel-getal> | RELATIVE <geheel-getal>

Grant-instructie

```

<grant-instructie> ::=
  <grant-tabelbevoegdheid-instructie> |
  <grant-databasebevoegdheid-instructie> |
  <grant-gebruikersbevoegdheid-instructie> |
  <grant-rol-instructie> |
  <grant-executebevoegdheid-instructie> |
  <grant-sequence-bevoegdheid-instructie>

<grant-tabelbevoegdheid-instructie> ::=
  GRANT <tabelbevoegdheden>
  ON <tabelspecificatie>
  TO <ontvangers>
  [ WITH GRANT OPTION ]

<grant-databasebevoegdheid-instructie> ::=
  GRANT <databasebevoegdheden>
  ON [ <databasenaam> . ] *
  TO <ontvangers>
  [ WITH GRANT OPTION ]

<grant-gebruikersbevoegdheid-instructie> ::=
  GRANT <gebruikersbevoegdheden>
  ON *.*
  TO <ontvangers>
  [ WITH GRANT OPTION ]

<grant-executebevoegdheid-instructie> ::=
  GRANT EXECUTE
  ON { <procedurenaam> |
      FUNCTION <stored-functionnaam> }
  TO <ontvangers>
  [ WITH GRANT OPTION ]

<grant-rol-instructie> ::=
  GRANT <rolnaam> [ , <rolnaam> ]...
  TO <ontvangers>

<grant-sequence-bevoegdheid-instructie> ::=
  GRANT <sequence-bevoegdheden>
  ON SEQUENCE <sequence-naam>
  TO <ontvangers>
  [ WITH GRANT OPTION ]

```

If-instructie

```

<if-instructie> ::=
  IF <conditie> THEN <instructielijst>
  [ ELSEIF <conditie> THEN <instructielijst> ]...
  [ ELSE <instructielijst> ]
  END IF

```

Include-instructie

```

<include-instructie> ::=
  INCLUDE <bestand>

```

Insert-instructie

```

<insert-instructie> ::=
    INSERT INTO <tabelspecificatie> <insert-specificatie>

<insert-specificatie> ::=
    [ <kolommenlijst> ] <values-component> |
    [ <kolommenlijst> ] <tabelexpressie>

```

Iterate-instructie

```

<iterate-instructie> ::=
    ITERATE <label>

```

Leave-instructie

```

<leave-instructie> ::=
    LEAVE <label>

```

Lock table-instructie

```

<lock-table-instructie> ::=
    LOCK TABLE <tabelspecificatie> IN <lock-type> MODE

<lock-type> ::= SHARE | EXCLUSIVE

```

Loop-instructie

```

<loop-instructie> ::=
    [ <label> : ] LOOP <instructielijst> END LOOP [ <label> ]

```

Open-instructie

```

<open-instructie> ::=
    OPEN <cursornaam>
    [ USING <host-variabele> [ , <host-variabele> ]... ]

```

Rename table-instructie

```

<rename-table-instructie> ::=
    RENAME TABLE <tabelnaamverandering>

<tabelnaamverandering> ::= <tabelnaam> TO <tabelnaam>

```

Repeat-instructie

```

<repeat-instructie> ::=
    [ <label> : ] REPEAT <instructielijst>
    UNTIL <conditie>
    END REPEAT <label>

```

Return-instructie

```
<return-instructie> ::=
    RETURN <scalaire-expressie>
```

Revoke-instructie

```
<revoke-instructie> ::=
    <revoke-tabelbevoegdheid-instructie> |
    <revoke-databasebevoegdheid-instructie> |
    <revoke-gebruikersbevoegdheid-instructie> |
    <revoke-rol-instructie> |
    <revoke-executebevoegdheid-instructie> |
    <revoke-sequence-bevoegdheid-instructie>

<revoke-tabelbevoegdheid-instructie> ::=
    REVOKE <tabelbevoegdheden>
    ON    <tabelspecificatie>
    FROM  <ontvangers>

<revoke-databasebevoegdheid-instructie> ::=
    REVOKE <databasebevoegdheden>
    ON    [ <databasenaam> . ] *
    FROM  <ontvangers>

<revoke-gebruikersbevoegdheid-instructie> ::=
    REVOKE <gebruikersbevoegdheden>
    ON    *.*
    FROM  <ontvangers>

<revoke-rol-instructie> ::=
    REVOKE <rolnaam>
    FROM  <ontvangers>

<revoke-executebevoegdheid-instructie> ::=
    REVOKE EXECUTE
    ON    { <procedurenaam> |
           FUNCTION <stored-functionnaam> }
    FROM  <ontvangers>

<revoke-sequence-bevoegdheid-instructie> ::=
    REVOKE <sequence-bevoegdheden>
    ON    SEQUENCE <sequence-naam>
    FROM  <ontvangers>
```

Rollback-instructie

```
<rollback-instructie> ::=
    ROLLBACK [ WORK ] [ TO [ SAVEPOINT ] <savepointnaam> ]
```

Savepoint-instructie

```
<savepoint-instructie> ::=
    SAVEPOINT <savepointnaam>
```

Select-instructie

```
<select-instructie> ::=
  <tabelexpressie>
  [ <for-component> ]
```

Select into-instructie

```
<select-into-instructie> ::=
  <select-component>
  <into-component>
  [ <from-component>
  [ <where-component> ]
  [ <group-by-component>
  [ <having-component> ] ] ]
```

Set-instructie

```
<set-instructie> ::=
  SET <lokale-variabele-definitie>
  [ , <lokale-variabele-definitie> ]...

<lokale-variabele-definitie> ::=
  <lokale-variabele> { = | := } <scalaire-expressie>
```

Set transaction-instructie

```
<set-transaction-instructie> ::=
  SET TRANSACTION ISOLATION LEVEL <isolation-level>

<isolation-level> ::=
  READ UNCOMMITTED |
  READ COMMITTED |
  REPEATABLE READ |
  SERIALIZABLE
```

Start transaction-instructie

```
<start-transaction-instructie> ::=
  START TRANSACTION
```

Update-instructie

```
<update-instructie> ::=
  UPDATE <tabelreferentie>
  SET <kolomtoekenning> [ , <kolomtoekenning> ]...
  [ WHERE { <conditie> | CURRENT OF <cursornaam> } ]

<kolomtoekenning> ::=
  <kolomnaam> = <scalaire-expressie>
```

Whenever-instructie

```

<whenever-instructie> ::=
    WHENEVER <whenever-conditie> <whenever-actie>

<whenever-conditie> ::= SQLWARNING | SQLERROR | NOT FOUND

<whenever-actie> ::= CONTINUE | GOTO <label>

```

While-instructie

```

<while-instructie> ::=
    [ <label> : WHILE <conditie> DO <instructielijst>
    END WHILE [ <label> ]

```

1.4.3 Basisbegrippen

Deze paragraaf bevat de algemene basisbegrippen die in diverse SQL-instructies gebruikt worden. De begrippen die een naam vormen staan alle aan het einde van deze paragraaf gegroepeerd.

```

<aggregatiefunctie> ::=
    COUNT ( [ DISTINCT | ALL ] { * | <scalaire-expressie> } ) |
    MIN ( [ DISTINCT | ALL ] <scalaire-expressie> ) |
    MAX ( [ DISTINCT | ALL ] <scalaire-expressie> ) |
    SUM ( [ DISTINCT | ALL ] <scalaire-expressie> ) |
    AVG ( [ DISTINCT | ALL ] <scalaire-expressie> ) |
    STDDEV ( [ DISTINCT | ALL ] <scalaire-expressie> ) |
    VARIANCE ( [ DISTINCT | ALL ] <scalaire-expressie> )

```

```

<alfanumeriek-datatype> ::=
    CHAR [ ( <lengte> ) ] |
    CHARACTER [ ( <lengte> ) ] |
    VARCHAR ( <lengte> ) |
    CHAR VARYING ( <lengte> ) |
    CHARACTER VARYING ( <lengte> ) |
    LONG VARCHAR

```

```

<alfanumerieke-constante> ::= <tekenreeks>

```

```

<alfanumerieke-expressie> ::=
    <alfanumerieke-scalaire-expressie> |
    <alfanumerieke-rijexpressie> |
    <alfanumerieke-tabelexpressie>

```

```

<alfanumerieke-scalaire-expressie> ::=
    <enkelvoudige-scalaire-expressie>
    COLLATE <collation-naam> |
    <samengestelde-scalaire-expressie>

```

```

<alternatieve-sleutel> ::= UNIQUE <kolommenlijst>

```

```
<any-all-operator> ::=
  <vergelijkingsoperator> { ALL | ANY | SOME }
```

```
<begin-end-blok> ::=
  [ <label> : ] BEGIN <instructielijst> END [ <label> ]
```

```
<blob-datatype> ::= BLOB
```

```
<boolean-constante> ::= TRUE | FALSE
```

```
<boolean-datatype> ::= BOOLEAN
```

```
<case-expressie> ::=
  CASE <when-definitie> [ ELSE <scalaire-expressie> ] END
```

```
<cast-expressie> ::=
  CAST ( <scalaire-expressie> AS <datatype> )
```

```
<check-integriteitsregel> ::= CHECK ( <conditie> )
```

```
<cijfer> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

```
<conditie> ::=
  <predikaat> |
  <predikaat> OR <predikaat> |
  <predikaat> AND <predikaat> |
  ( <conditie> ) |
  NOT <conditie>
```

```
<constante> ::=
  <numerieke-constante> |
  <alfanumerieke-constante> |
  <temporal-constante> |
  <boolean-constante> |
  <hexadecimal-constante>
```

```
<dagen> ::= <cijfer> [ <cijfer> ]
```

```
<databasebevoegdheid> ::=
```

```
SELECT
INSERT
DELETE
UPDATE
REFERENCES
CREATE
ALTER
DROP
INDEX
CREATE TEMPORARY TABLES
CREATE VIEW
CREATE ROUTINE
ALTER ROUTINE
EXECUTE ROUTINE
LOCK TABLES
```

```
<database-optie> ::=
```

```
[ DEFAULT ] CHARACTER SET <karaktersetnaam> |
[ DEFAULT ] COLLATE <collation-naam>
```

```
<datatype> ::=
```

```
<numeriek-datatype> |
<alfanumeriek-datatype> |
<temporal-datatype> |
<boolean-datatype> |
<blob-datatype>
```

```
<datumconstante> ::= ' <jaren> - <maanden> - <dagen> '
```

```
<datuminterval> ::=
```

```
INTERVAL <intervallengte> <datumintervaleenheid>
```

```
<datumintervaleenheid> ::=
```

```
DAY | WEEK | MONTH | QUARTER | YEAR
```

```
<decimal-constante> ::=
```

```
[ + | - ] <geheel-getal> [ . <geheel-getal> ] |
[ + | - ] <geheel-getal> . |
[ + | - ] . <geheel-getal>
```

```
<decimal-datatype> ::=
```

```
DECIMAL [ ( <precisie> [ , <schaal> ] ) ] |
DEC [ ( <precisie> [ , <schaal> ] ) ] |
NUMERIC [ ( <precisie> [ , <schaal> ] ) ] |
NUM [ ( <precisie> [ , <schaal> ] ) ]
```

```
<enkelvoudige-rijexpressie> ::=
```

```
( <scalaire-expressie> [ , <scalaire-expressie> ]... ) |
<rij-subquery>
```



```

<enkelvoudige-scalaire-expressie> ::=
  <enkelvoudige-numerieke-expressie> |
  <enkelvoudige-alfanumerieke-expressie> |
  <enkelvoudige-datumexpressie> |
  <enkelvoudige-tijdexpressie> |
  <enkelvoudige-timestamp-expressie> |
  <enkelvoudige-hexadecimal-expressie>

```

Elk van de bovenstaande enkelvoudige scalaire expressies kent de volgende verschillende vormen:

```

<enkelvoudige-scalaire-expressie> ::=
  <constante> |
  <kolomspecificatie> |
  <gebruikersvariabele> |
  <systemvariabele> |
  <cast-expressie> |
  <case-expressie> |
  NULL |
  ( <scalaire-expressie> ) |
  <scalaire-functie> |
  <aggregatiefunctie> |
  <scalaire-subquery> |
  <lokale-variabele> |
  <host-variabele> |
  <systemparameter> |
  <sequence-referentie>

```

```

<enkelvoudige-tabelexpressie> ::= <select-blok-kop>

```

```

<exponent> ::= <integer-constante>

```

```

<expressie> ::=
  <scalaire-expressie> |
  <rijexpressie> |
  <tabelexpressie>

```

```

<expressielijst> ::= <expressie> [ , <expressie> ]...

```

```

<float-constante> ::= <mantisse> { E | e } <exponent>

```

```

<float-datatype> ::=
  FLOAT [ ( <lengte> ) ] |
  REAL |
  DOUBLE [ PRECISION ]

```

```

<for-component> ::=
  FOR UPDATE [ OF <kolomnaam> [ , <kolomnaam> ]... ] |
  FOR READ ONLY

```

```

<from-component> ::=

```

```
FROM <tabelreferentie> [ , <tabelreferentie> ]...
```

```
<gebruikersbevoegdheid> ::=
  SELECT
  INSERT
  DELETE
  UPDATE
  REFERENCES
  CREATE
  ALTER
  DROP
  INDEX
  CREATE TEMPORARY TABLES
  CREATE VIEW
  CREATE ROUTINE
  ALTER ROUTINE
  EXECUTE ROUTINE
  LOCK TABLES
  CREATE USER
```

```
<gebruikersvariabele> ::= @ <variabelenaam>
```

```
<geheel-getal> ::= <cijfer>...
```

```
<group-by-component> ::=
  GROUP BY <group-by-specificatielijst>
  [ WITH { ROLLUP | CUBE } ]
```

```
<group-by-expressie> ::= <scalaire-expressie>
```

```
<group-by-specificatie> ::=
  <group-by-expressie> |
  <grouping-sets-specificatie> |
  <rollup-specificatie>
```

```
<group-by-specificatielijst> ::=
  <group-by-specificatie> [ , <group-by-specificatie> ]...
```

```
<grouping-sets-specificatie> ::=
  GROUPING SETS ( <grouping-sets-specificatielijst> )
```

```
<grouping-sets-specificatie> ::=
  <group-by-expressie> |
  <rollup-specificatie> |
  ( <grouping-sets-specificatielijst> )
```

```
<grouping-sets-specificatielijst> ::=
  <grouping-sets-specificatie>
  [ , <grouping-sets-specificatie> ]...
```

```
<having-component> ::= HAVING <conditie>
```

```
<hexadecimal-constante> ::= X ' <hexadecimalteken>... '
```

```
<hexadecimalteken> ::=
  <cijfer> | A | B | C | D | E | F | a | b | c | d | e | f
```

```
<host-variabele> ::= ":" <host-variabelenaam>
```

```
<host-variabele-element> ::=
  <host-variabele> [ <null-indicator> ]
```

```
<host-variabelelijst> ::=
  <host-variabele-element> [ , <host-variabele-element> ]...
```

```
<indextype> ::= UNIQUE | CLUSTERED
```

```
<instructie-in-body> ::=
  <declaratieve-instructie> |
  <procedurele-instructie>
```

```
<instructielijst> ::= { <instructie-in-body> ; }...
```

```
<integer-constante> ::= [ + | - ] <geheel-getal>
```

```
<integer-datatype> ::=
  SMALLINT |
  INTEGER |
  INT |
  BIGINT
```

```
<intervallengte> ::= <scalaire-expressie>
```

```
<into-component> ::=
  INTO <host-variabele> [ , <host-variabele> ]...
```

```
<jaren> ::= <geheel-getal>
```

```
<join-conditie> ::=
    ON <conditie> | USING <kolommenlijst>
```

```
<join-specificatie> ::=
    <tabelreferentie> <join-type> <tabelreferentie>
    [ <join-conditie> ]
```

```
<join-type> ::=
    [ INNER ] JOIN |
    LEFT [ OUTER ] JOIN |
    RIGHT [ OUTER ] JOIN |
    FULL [ OUTER ] JOIN |
    UNION JOIN |
    CROSS JOIN
```

```
<kolomdefinitie> ::=
    <kolomnaam> <datatype> [ <null-specificatie> ]
    [ <kolom-integriteitsregel> ] [ <kolom-optie>... ]
```

```
<kolom-in-index> ::= <kolomnaam> [ ASC | DESC ]
```

```
<kolom-integriteitsregel> ::=
    PRIMARY KEY |
    UNIQUE |
    <check-integriteitsregel>
```

```
<kolommenlijst> ::=
    ( <kolomnaam> [ , <kolomnaam> ]... )
```

```
<kolomnaam> ::= <naam>
```

```
<kolom-optie> ::=
    DEFAULT <constante> |
    COMMENT <alfanumerieke-constante>
```

```
<kolomspecificatie> ::=
    [ <tabelspecificatie> . ] <kolomnaam>
```

```
<kolom-subquery> ::= ( <tabelexpressie> )
```

```
<lengte> ::= <geheel-getal>
```

```
<letter> ::=
 a | b | c | d | e | f | g | h | i | j | k | l | m |
 n | o | p | q | r | s | t | u | v | w | x | y | z |
 A | B | C | D | E | F | G | H | I | J | K | L | M |
 M | O | P | Q | R | S | T | U | V | W | X | Y | Z
```

```
<like-patroon> ::= <scalaire-alfanumerieke-expressie>
```

```
<lokale-variabele> ::= <variabelenaam>
```

```
<lokale-variabelelijst> ::=
 <lokale-variabele> [ , <lokale-variabele> ]...
```

```
<maanden> ::= <cijfer> [ <cijfer> ]
```

```
<mantisse> ::= <decimal-constante>
```

```
<microseconden> ::= <geheel-getal>
```

```
<minuten> ::= <cijfer> [ <cijfer> ]
```

```
<mysql-error-code> ::= <geheel-getal>
```

```
<null-indicator> ::= <host-variabele>
```

```
<null-specificatie> ::= NOT NULL
```

```
<numeriek-datatype> ::=
 <integer-datatype> |
 <decimal-datatype> |
 <float-datatype>
```

```
<numerieke-constante> ::=
 <integer-constante> |
 <decimal-constante> |
 <float-constante>
```

```
<ontvangers> ::=
 <gebruikersnaam> [ , <gebruikersnaam> ]... |
 <rolnaam> [ , <rolnaam> ]... |
 PUBLIC
```

```
<order-by-component> ::=
  ORDER BY <sortering> [ , <sortering> ]...
```

```
<precisie> ::= <geheel-getal>
```

```
<predikaat> ::=
  <predikaat-met-vergelijking> |
  <predikaat-met-in> |
  <predikaat-met-between> |
  <predikaat-met-like> |
  <predikaat-met-null> |
  <predikaat-met-exists> |
  <predikaat-met-any-all>
```

```
<predikaat-met-any-all> ::=
  <scalaire-expressie> <any-all-operator> <kolom-subquery>
```

```
<predikaat-met-between> ::=
  <scalaire-expressie> [ NOT ] BETWEEN <scalaire-expressie>
  AND <scalaire-expressie>
```

```
<predikaat-met-exists> ::= EXISTS <tabel-subquery>
```

```
<predikaat-met-in> ::=
  <scalaire-expressie> [ NOT ] IN <scalaire-expressielijst> |
  <scalaire-expressie> [ NOT ] IN <kolom-subquery> |
  <rijexpressie> [ NOT ] IN <rijexpressielijst> |
  <rijexpressie> [ NOT ] IN <tabel-subquery>
```

```
<predikaat-met-like> ::=
  <scalaire-expressie> [ NOT ] LIKE <like-patroon>
  [ ESCAPE <speciaal-symbool> ]
```

```
<predikaat-met-null> ::=
  <scalaire-expressie> IS [ NOT ] NULL
```

```
<predikaat-met-vergelijking> ::=
  <scalaire-expressie> <vergelijkingsoperator>
  <scalaire-expressie> |
  <rijexpressie> <vergelijkingsoperator> <rijexpressie>
```

```
<primaire-sleutel> ::= PRIMARY KEY <kolommenlijst>
```

```
<pseudoniem> ::= <naam>
```

```
<refererende-actie> ::=
    ON UPDATE { CASCADE | RESTRICT | SET NULL } |
    ON DELETE { CASCADE | RESTRICT | SET NULL }
```

```
<refererende-sleutel> ::=
    FOREIGN KEY <kolommenlijst> <refererende-specificatie>
```

```
<refererende-specificatie> ::=
    REFERENCES <tabelspecificatie> [ <kolommenlijst> ]
    [ <refererende-actie>... ]
```

```
<rijexpressie> ::= <enkelvoudige-rijexpressie>
```

```
<rijexpressielijst> ::=
    ( <scalaire-expressielijst>
      [ , <scalaire-expressielijst> ]... )
```

```
<rij-subquery> ::= ( <tabelexpressie> )
```

```
<rollup-specificatie> ::=
    ROLLUP ( <group-by-expressielijst> ) |
    CUBE ( <group-by-expressielijst> ) |
    ( )
```

```
<samengestelde-alfanumerieke-expressie> ::=
    <scalaire-alfanumerieke-expressie> "||"
    <scalaire-alfanumerieke-expressie>
```

```
<samengestelde-datumexpressie> ::=
    <scalaire-datumexpressie> { + | - } <datuminterval>
```

```
<samengestelde-numerieke-expressie> ::=
    [ + | - ] <scalaire-numerieke-expressie> |
    ( <scalaire-numerieke-expressie> ) |
    <scalaire-numerieke-expressie>
    <wiskundige-operator> <scalaire-numerieke-expressie>
```

```
<samengestelde-scalaire-expressie> ::=
    <samengestelde-numerieke-expressie> |
    <samengestelde-alfanumerieke-expressie> |
    <samengestelde-datumexpressie> |
    <samengestelde-tijdexpressie> |
    <samengestelde-timestamp-expressie>
```

```
<samengestelde-tabelexpressie> ::=
    <tabelexpressie> <set-operator> <tabelexpressie>
```

```
<samengestelde-tijdexpressie> ::=
    ADDTIME( <scalaire-tijdexpressie> , <tijdinterval> )
```

```
<samengestelde-timestamp-expressie> ::=
    <scalaire-timestamp-expressie> { + | - } <timestamp-interval>
```

```
<scalaire-expressie> ::=
    <enkelvoudige-scalaire-expressie> |
    <samengestelde-scalaire-expressie>
```

```
<scalaire-alfanumerieke-expressie> ::=
    <enkelvoudige-alfanumerieke-expressie> |
    <samengestelde-alfanumerieke-expressie>
```

```
<scalaire-datumexpressie> ::=
    <enkelvoudige-datumexpressie> |
    <samengestelde-datumexpressie>
```

```
<scalaire-expressielijst> ::=
    ( <scalaire-expressie> [ , <scalaire-expressie> ]... )
```

```
<scalaire-hexadecimal-expressie> ::=
    <enkelvoudige-hexadecimal-expressie>
```

```
<scalaire-numerieke-expressie> ::=
    <enkelvoudige-numerieke-expressie> |
    <samengestelde-numerieke-expressie>
```

```
<scalaire-subquery> ::= ( <tabelexpressie> )
```

```
<scalaire-tijdexpressie> ::=
    <enkelvoudige-tijdexpressie> |
    <samengestelde-tijdexpressie>
```

```
<scalaire-timestamp-expressie> ::=
    <enkelvoudige-timestamp-expressie> |
    <samengestelde-timestamp-expressie>
```

```
<schaal> ::= <geheel-getal>
```

```
<seconden> ::= <cijfer> [ <cijfer> ]
```



```
<select-blok-kop> ::=
  <select-component>
  [ <from-component>
  [ <where-component> ]
  [ <group-by-component>
  [ <having-component> ] ] ]
```

```
<select-blok-staart> ::=
  [ <order-by-component> ]
```

```
<select-component> ::=
  SELECT [ DISTINCT | ALL ] <select-elementenlijst>
```

```
<select-element> ::=
  <scalaire-expressie> [ [ AS ] <kolomnaam> ] |
  <tabelspecificatie>.* |
  <pseudoniem>.*
```

```
<select-elementenlijst> ::=
  <select-element> [ , <select-element> ]... |
  *
```

```
<sequence-bevoegdheden> ::=
  <sequence-bevoegdheid> [ , <sequence-bevoegdheid> ]...
```

```
<sequence-bevoegdheid> ::= ALTER | USAGE
```

```
<sequence-referentie> ::=
  { NEXT | PREVIOUS ] VALUE FOR
  [ <gebruikersnaam> . ] <sequence-naam>
```

```
<set-operator> ::=
  UNION | INTERSECT | EXCEPT |
  UNION ALL | INTERSECT ALL | EXCEPT ALL
```

```
<sorteerrichting> ::= ASC | DESC
```

```
<sortering> ::=
  <kolomnaam> [ <sorteerrichting> ] |
  <scalaire-expressie> [ <sorteerrichting> ] |
  <volgnummer> [ <sorteerrichting> ]
```

```
<speciaal-symbool> ::= alle vreemde tekens, zoals !, # en *
```

```
<sqlstate-waarde> ::= <alfanumerieke-constante>
```

```
<subquery> ::= ( <tabelexpressie> )
```

```
<systeemparemeter> ::= @@ <variabelenaam>
```

```
<tabelbevoegdheden> ::=
  ALL [ PRIVILEGES ] |
  <tabelbevoegdheid> [ , <tabelbevoegdheid> ]...
```

```
<tabelbevoegdheid> ::=
  SELECT           |
  INSERT           |
  DELETE           |
  UPDATE [ <kolommenlijst> ] |
  REFERENCES [ <kolommenlijst> ] |
  ALTER           |
  INDEX           |
```

```
<tabel-element> ::=
  <kolomdefinitie>      |
  <tabel-integriteitsregel>
```

```
<tabelexpressie> ::=
  { <enkelvoudige-tabelexpressie> |
    ( <tabelexpressie> )           |
    <samengestelde-tabelexpressie> }
  [ <select-blok-staart> ]
```

```
<tabelinhoud> ::= [ AS ] <tabelexpressie>
```

```
<tabel-integriteitsregel> ::=
  [ CONSTRAINT <constraintnaam> ]
  { <primaire-sleutel>           |
    <alternatieve-sleutel>      |
    <refererende-sleutel>      |
    <check-integriteitsregel> }
```

```
<tabelreferentie> ::=
  <tabelspecificatie> [ [ AS ] <pseudoniem> ] |
  <join-specificatie>                       |
  ( <join-specificatie> )                   |
  <tabel-subquery> [ [ AS ] <pseudoniem> ]
```

```
<tabelschema> ::= ( <tabel-element> [ , <tabel-element> ]... )
```

```
<tabelspecificatie> ::=
    [ <databasenaam> . | <gebruikersnaam> . ] <tabelnaam>
```

```
<tabel-subquery> ::= ( <tabelexpressie> )
```

```
<teken> ::= <cijfer> | <letter> | <speciaal-symbool> | ''
```

```
<tekenreeks> ::= ' [ <teken>... ] '
```

```
<temporal-constante> ::=
    <datumconstante>      |
    <tijdconstante>       |
    <timestamp-constante>
```

```
<temporal-datatype> ::=
    DATE      |
    TIME      |
    TIMESTAMP
```

```
<tijdconstante> ::= ' <uren> : <minuten> [ : <seconden> ] '
```

```
<tijdinterval> ::= <scalaire-tijdexpressie>
```

```
<timestamp-constante> ::=
    ' <jaren> - <maanden> - <dagen> <spatie>
      <uren> : <minuten> [ : <seconden> [ . <microseconden> ] ] '
```

```
<timestamp-interval> ::=
    INTERVAL <intervallengte> <timestamp-intervaleenheid>
```

```
<timestamp-intervaleenheid> ::=
    MICROSECOND | SECOND | MINUTE | HOUR |
    DAY | WEEK | MONTH | QUARTER | YEAR
```

```
<uren> ::= <cijfer> [ <cijfer> ]
```

```
<values-component> ::=
    VALUES <rijexpressie> [ , <rijexpressie> ]...
```

```
<vergelijkingsoperator> ::=
    = | < | > | <= | >= | <>
```

```
<wachtwoord> ::= '<naam>'
```

```
<when-definitie> ::= <when-definitie-1> | <when-definitie-2>
```

```
<when-definitie-1> ::=
  <scalaire-expressie>
  WHEN <scalaire-expressie> THEN <scalaire-expressie>
  [ { WHEN <scalaire-expressie> THEN <scalaire-expressie> } ]...
```

```
<when-definitie-2> ::=
  WHEN <conditie> THEN <scalaire-expressie>
  [ { WHEN <conditie> THEN <scalaire-expressie> } ]...
```

```
<where-component> ::= WHERE <conditie>
```

```
<wiskundige-operator> ::= * | / | + | -
```

```
<collation-naam>      ;
<conditiennaam>      ;
<constraintnaam>     ;
<cursornaam>         ;
<databasenaam>       ;
<stored-functionnaam> ;
<gebruikersnaam>    ;
<hostnaam>           ;
<host-variabelenaam> ;
<indexnaam>          ;
<karaktersetnaam>    ;
<kolomnaam>          ;
<label>              ;
<procedurenaam>     ;
<rolnaam>            ;
<savepointnaam>     ;
<tabelnaam>          ;
<triggernaam>        ;
<variabelenaam>     ;
<viewnaam>           ::=
  <letter> { <letter> | <cijfer> | _ }...
```

De Auteur

Rick F. van der Lans is auteur van vele boeken over SQL. Naast dit SQL Leerboek dat in diverse talen vertaald is, waaronder Engels, Duits, Chinees en Italiaans, heeft hij SQL boeken geschreven voor producten als MySQL, Oracle, SQLite, Ingres en Pervasive PSQL.



Hij is onafhankelijk adviesur, auteur en docent gespecialiseerd in databasetechnologie, datawarehousing en applicatie-integratie. Hij is oprichter en directeur van R20/Consultancy. Door de jaren heen heeft hij veel organisaties geadviseerd op het gebied van IT-architecturen.

Als spreker op conferenties en seminars wordt hij internationaal gerespecteerd. Al meer dan vijftig jaar geeft hij over de gehele wereld lezingen, inclusief in de meeste Europese landen, Noord- en Zuid-Amerika en Australië. Hij is voorzitter van het jaarlijkse European Data Warehouse and Business Intelligence Conference. Hij schrijft een column voor Database Magazine en voor het internationale Beye-Network.com. Zeven jaar lang was hij lid van de Nederlandse ISO commissie verantwoordelijk voor ISO SQL Standaard.

Rick kan via de volgende kanalen bereikt worden:

Email: rick@r20.nl
Twitter: http://twitter.com/Rick_vanderlans
LinkedIn: <http://www.linkedin.com/pub/rick-van-der-lans/9/207/223>

Cursussen over de volgende onderwerpen kunnen door Rick F. van der Lans verzorgd worden

- Database-ontwerp en informatiemodellering
- De basis van SQL
- Het ontwikkelen van geavanceerde SQL queries
- Datawarehousing en business intelligence
- Data virtualisatie

Andere boeken geschreven door Rick F. van der Lans

