insightsoftware

Simba Java SQL Engine

Reference Guide

Version 10.3

August 2024

Copyright

This document was released in August 2024.

Contact Us

insightsoftware

www.insightsoftware.com

About This Guide

Purpose

This guide explains how to use the Simba Java SQL Engine to document what SQL grammar the JSQL engine supports.

Audience

The guide is intended for developers who have created a connector with the Simba Java SQL Engine. This guide is also intended for end users of the Simba Java SQL Engine.

Knowledge Prerequisites

To use the Simba Java SQL Engine, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Simba Java SQL Engine.
- Ability to use the data store to which the Simba Java SQL Engine is connecting.
- An understanding of the role of ODBC or JDBC technologies and driver managers in connecting to a data store.
- Experience creating and configuring ODBC or JDBC connections.

Variables Used in this Document

The following variables are used in this document:

Variable	Description
[DRIVER_NAME]	The name of your connector, as used in Windows registry keys and names of configuration files.
[INSTALL_DIR]	Installation directory for the Simba Java SQL Engine.

Contents

Introducing the Simba Java SQL Engine

An ODBC SQL Parsing and Execution Engine for Database Access, Simba Java SQL Engine is a component of SimbaEngine. It provides SQL processing in data drivers and data providers for non-SQL data stores built with SimbaEngine. Simba Java SQL Engine can also be used in custom solutions where SQL parsing or execution is required.

About

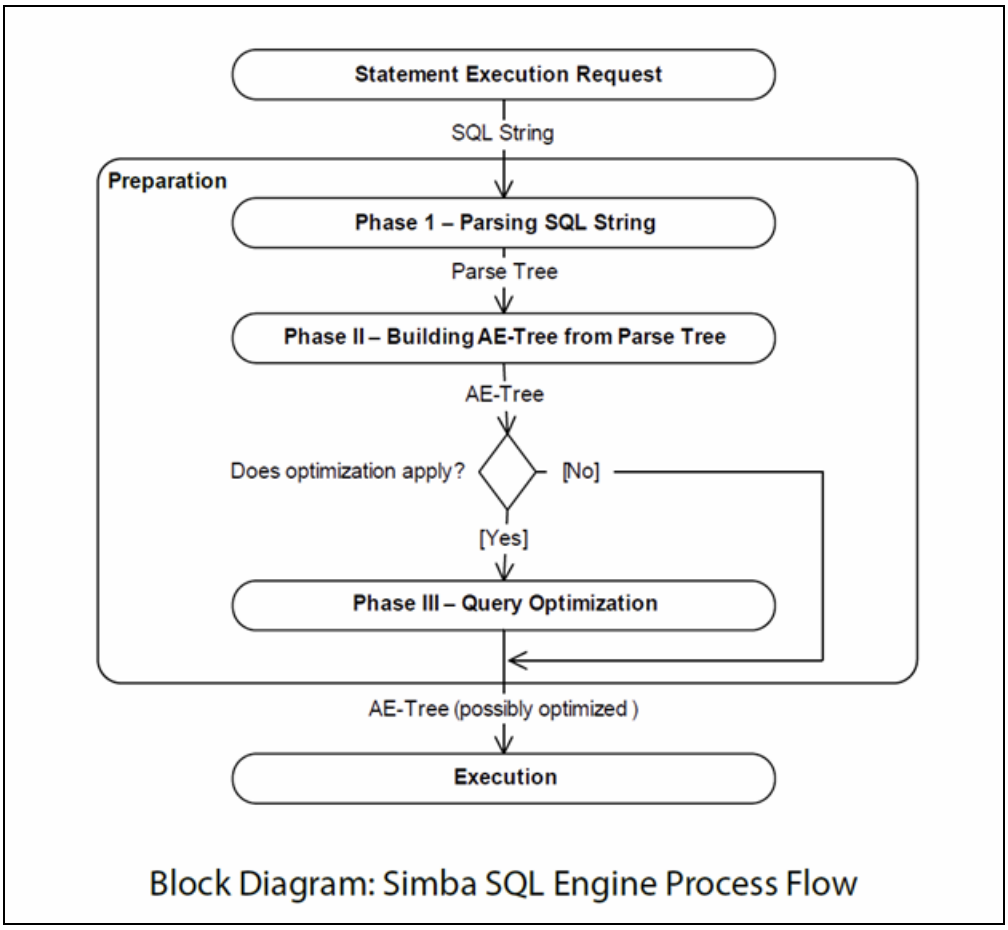
Simba Java SQL Engine is an easy-to-use, complete, high-performance SQL parser and execution engine for building ODBC, JDBC, ADO.NET or OLE DB data access solutions for non-SQL data stores. You can use it with other SimbaEngine components to create either local or remote data driver access to non-SQL data sources – such as web service, object-oriented, network, real time (e.g. SCADA) and ISAM databases – or you can use it in a custom solution and call it from your own code. Simba SQL Engine's clean and logical interfaces make it easy to work with, and it can either parse, or parse and execute a SQL statement and retrieve the result set. Included with the SimbaEngine is the Java Quickstart ODBC driver that is a working example of a JDBC driver built using SQL Engine that you can modify to demonstrate reading your own data in just a few days.

The Simba Java SQL Engine parser conforms to the SQL-92 specification, but it can be extended to implement any SQL function or feature. When Simba Java SQL Engine prepares a SQL statement, it validates the tables and columns referenced in the statement against the metadata from the underlying data store. Then it creates an intermediate version of the query called the Algebraic Expression Tree (AE-Tree).

The AE-Tree represents the metadata for the query, and is eventually transformed into an Execution Tree (E-Tree), which can be executed. By carefully moving or modifying nodes, the AE-Tree can be updated to change the way the query is executed without changing the result. The underlying data store can modify the AE-Tree, so that when the query is executed, Simba Java SQL Engine delegates some or all of the processing to the underlying data store. We call this Collaborative Query Execution. The advantage is that any high-performance processing available in the data store is available for use with Simba Java SQL Engine. If the entire query can be delegated, Simba Java SQL Engine will only retrieve the result set. This is particularly important for analytical databases with built-in, high-performance aggregation processing that end users expect to see even through JDBC.

Simba Java SQL Engine is a complete, compatible, high-performance and easy to use solution for SQL processing on non-SQL data stores. Everything you need to prepare and execute SQL queries is included with a clean and logical interface. With compliance to the SQL-92 standard, it is carefully designed to support common reporting and BI tools, so end users can get on with their work. Collaborative Query Execution allows you to use the high-performance processing built into your own data store to boost the performance of Simba Java SQL Engine. You can use Simba Java SQL Engine with the other components of SimbaEngine to quickly deploy local or remote data access, or you can use it alone, called from your own code. However you choose to deploy it, Simba Java SQL Engine is the easiest and most flexible SQL preparation and execution engine available today.

This flow of control is illustrated below:



Block Diagram: Simba SQL Engine Process Flow

Specifications

The following list details the specifications for the Simba Java SQL Engine:

- SQL-92-compliant parser with comprehensive support for ODBC 3.80-compliant data types, as well as scalar and aggregate functions, except interval types
- Maximum number of simultaneous connections limited only by memory.
- Maximum result set length limited only by memory and swap table space.
- Maximum 65,356 columns supported in a select list.
- Maximum 65,356 columns supported in a table.
- Maximum number of characters in a SQL statement is configurable and is only limited by memory.
- Maximum number of joined tables limited only by memory and swap table space.

SQL Engine Parser

The following is a list of commands that the SQL Engine Parser supports that the Execution component does not support. This means that a parse tree can be generated, however executing using the Execution component will fail.

- ALL PRIVILEGES
- ALTER TABLE
- CASCADE
- CREATE VIEW
- DROP VIEW
- EXCEPT
- EXCEPT ALL
- GRANT
- GRANT OPTION FOR
- PUBLIC
- REFERENCES
- RESTRICT
- REVOKE
- SET CATALOG
- SET SCHEMA
- USAGE
- WITH GRANT OPTION

SQL Engine SQL Commands

This section discusses commands that are supported by both the Parser and Execution components of the SQL Engine:

- CREATE TABLE
- CREATE VIEW
- DELETE
- DROP TABLE
- DROP VIEW
- INSERT
- SELECT
- UPDATE
- UPSERT

Functions

Simba Engine SQL has many built-in functions for performing calculations on data. There are several basic types and categories of functions. The following sections list some SQL Functions supported by Simba Engine SQL.

Aggregate Functions

An aggregate function takes a set of values (like a column of data) and returns a single value result from the set of values. As of SimbaEngine 9.3, custom aggregates are supported (excluding the ETree), which means that any aggregate function can be supported.

Aggregate Functions	SimbaEngine SQL Supported
AVG	Yes
COUNT	Yes
MAX	Yes
MIN	Yes
SUM	Yes

Scalar Function

A scalar function takes input argument(s) and returns a single value result. Following is a list of scalar function which supported by SimbaEngine SQL. As of SimbaEngine 9.3, custom scalars are supported, which means that any scalar function can be supported.

Scalar Function	SimbaEngine SQL Supported
String Functions	
ASCII	Yes
CHAR	Yes

Scalar Function	SimbaEngine SQL Supported
CONCAT	Yes
LCASE	Yes
LEFT	Yes
LENGTH	Yes
LOCATE	Yes
LTRIM	Yes
INSERT	Yes
REPEAT	Yes
REPLACE	Yes
RIGHT	Yes
RTRIM	Yes
SOUNDEX	Yes
SPACE	Yes
SUBSTRING	Yes
UCASE	Yes
BIT_LENGTH	No
CHAR_LENGTH	No
CHARACTER_LENGTH	No

Scalar Function	SimbaEngine SQL Supported
DIFFERENCE	No
OCTET_LENGTH	No
POSITION(_IN_)	No
Numeric Functions	
ABS	Yes
ACOS	Yes
ASIN	Yes
ATAN	Yes
ATAN2	Yes
CEILING	Yes
COS	Yes
COT	Yes
DEGREES	Yes
EXP	Yes
FLOOR	Yes
LOG	Yes
LOG10	Yes
MOD	Yes

Scalar Function	SimbaEngine SQL Supported
PI	Yes
POWER	Yes
RADIANS	Yes
RAND	Yes
ROUND	Yes
SIGN	Yes
SIN	Yes
SQRT	Yes
TAN	Yes
TRUNCATE	Yes
System Functions	
CAST	Yes
CONVERT	Yes
DATABASE	Yes
IFNULL	Yes
USER	Yes
SimbaEngine SQL	
COALESCE	Yes

Scalar Function	SimbaEngine SQL Supported
NULL	Yes
NULLIF	Yes

Date and Time Functions

The following is a table of all important Date and Time related functions available through Simba Engine SQL.

Date and Time and Interval Functions	SimbaEngine SQL Supported
CURRENT_DATE	Yes
CURRENT_TIME	Yes
CURRENT_TIME(time-precision)	Yes
CURRENT_TIMESTAMP	Yes
CURRENT_TIMESTAMP(time-precision)	Yes
CURDATE	Yes
CURTIME	Yes
DAYNAME	Yes
DAYOFMONTH	Yes
DAYOFWEEK	Yes
DAYOFYEAR	Yes
HOUR	Yes
MINUTE	Yes

Date and Time and Interval Functions	SimbaEngine SQL Supported
MONTH	Yes
MONTHNAME	Yes
NOW	Yes
DAYNAME	Yes
DAYOFMONTH	Yes
DAYOFWEEK	Yes
DAYOFYEAR	Yes
HOUR	Yes
MINUTE	Yes
MONTH	Yes
MONTHNAME	Yes
NOW	Yes
QUARTER	Yes
SECOND	Yes
TIMESTAMPDIFF	Yes
TIMESTAMPADD	Yes
WEEK	Yes
WEEK_ISO	Yes

Date and Time and Interval Functions	SimbaEngine SQL Supported
YEAR	Yes
EXTRACT	Yes

Joins

A SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them. SimbaEngine SQL supports all the join types which show in following table:

Joins	SimbaEngine SQL Supported
CROSS JOIN	Yes
JOIN	Yes
INNER	Yes
LEFT OUTER	Yes
LEFT	Yes
RIGHT OUTER	Yes
RIGHT	Yes
FULL	Yes
FULL OUTER	Yes

SELECT or Subquery Key Words

A subquery is a query within a query. A subquery is usually added in the WHERE Clause of the sql statement, and is usually used when you know how to search for a value using a SELECT statement but do not know the exact value in the database. Subqueries are an alternate way of returning data from multiple tables and can be used with the comparision operators, for example “=”, “<>”, “<”, “>”, “<=” and “>=”. They map contain any of the keywords used in the SELECT statement.

The following are other keywords that can be used within a SELECT statement, in any of the associated clauses:

Subquery Key Words	SimbaEngine SQL Supported
ALL	Yes
ANY	Yes
DISTINCT	Yes
GROUP BY	Yes
HAVING	Yes
LIMIT	Yes
TOP	Yes
UNION	Yes
UNION ALL	Yes
VALUES	Yes
AS	Yes
ASC	Yes
DATATYPE (one of the SQL_* types)	Yes
TSIDATATYPE (one of the SQL_TSI_* types)	Yes
DESC	Yes
DEFAULT	Yes
WHERE	Yes

Subquery Key Words	SimbaEngine SQL Supported
OR	Yes
AND	Yes
NOT	Yes
EQ (=)	Yes
NE (<>)	Yes
LT (<)	Yes
GT (>)	Yes
LE (<=)	Yes
GE (>=)	Yes
NOT BETWEEN	Yes
BETWEEN	Yes
NOT IN	Yes
IN	Yes
NOT LIKE	Yes
LIKE	Yes
ESCAPE	Yes
IS NOT NULL	Yes
IS NULL	Yes

Subquery Key Words	SimbaEngine SQL Supported
EXISTS	Yes

Simba Engine SQL Keywords

Keywords cannot be used as identifiers (names of database objects such as columns and tables) without quoting. Simba Engine SQL reserved keywords for defining, manipulating, and accessing databases.

Key Words	SimbaEngine SQL Supported
ADD *	Yes
ALL	Yes
ALTER *	Yes
AND	Yes
ANY	Yes
AS	Yes
ASC	Yes
AVG	Yes
BETWEEN	Yes
BY	Yes
CALL	Yes
CASCADE *	Yes
CASE	Yes

Key Words	SimbaEngine SQL Supported
CAST	Yes
CATALOG *	Yes
CHECK	Yes
COALESCE	Yes
COLUMN	Yes
CONVERT	Yes
COUNT	Yes
CREATE	Yes
CROSS	Yes
DATE	Yes
DAY	Yes
DEFAULT	Yes
DELETE	Yes
DESC	Yes
DISTINCT	Yes
DROP	Yes
ELSE	Yes
END	Yes

Key Words	SimbaEngine SQL Supported
ESCAPE	Yes
EXCEPT *	Yes
EXISTS	Yes
FN	Yes
FOR	Yes
FOREIGN *	Yes
FROM	Yes
FULL	Yes
GRANT *	Yes
GROUP	Yes
HAVING	Yes
HOUR	Yes
IF	Yes
IN	Yes
INDEX	Yes
INNER	Yes
INSERT	Yes
INTERVAL	Yes

Key Words	SimbaEngine SQL Supported
INTO	Yes
IS	Yes
JOIN	Yes
KEY	Yes
LEFT	Yes
LIKE	Yes
LIMIT	Yes
MAX	Yes
MIN	Yes
MINUTE	Yes
MONTH	Yes
NOT	Yes
NULL	Yes
NULLIF	Yes
ON	Yes
OPTION *	Yes
OR	Yes
ORDER	Yes

Key Words	SimbaEngine SQL Supported
OUTER	Yes
PERCENT	Yes
PRIMARY	Yes
PRIVILEGE *	Yes
PROCEDURE	Yes
PUBLIC *	Yes
REFERENCES *	Yes
RESTRICT *	Yes
REVOKE *	Yes
RIGHT	Yes
SCHEMA *	Yes
SECOND	Yes
SELECT	Yes
SET	Yes
SOME *	Yes
SUM	Yes
SVBEGIN	Yes
SVEND	Yes

Key Words	SimbaEngine SQL Supported
TABLE *	Yes
THEN	Yes
TIME	Yes
TIMESTAMP	Yes
TIMESTAMPADD	Yes
TIMESTAMPDIFF	Yes
TO	Yes
TOP	Yes
UNION	Yes
UNIQUE	Yes
UPDATE	Yes
USAGE *	Yes
USER	Yes
VALUES	Yes
VIEW *	Yes
WHEN	Yes
WHERE	Yes
WITH	Yes

Key Words	SimbaEngine SQL Supported
YEAR	Yes

* Parser Support only. This is not supported by the AETree or ETree components.

Appendix A: BNF Grammar for Java SQLEngine with Railroad Diagram

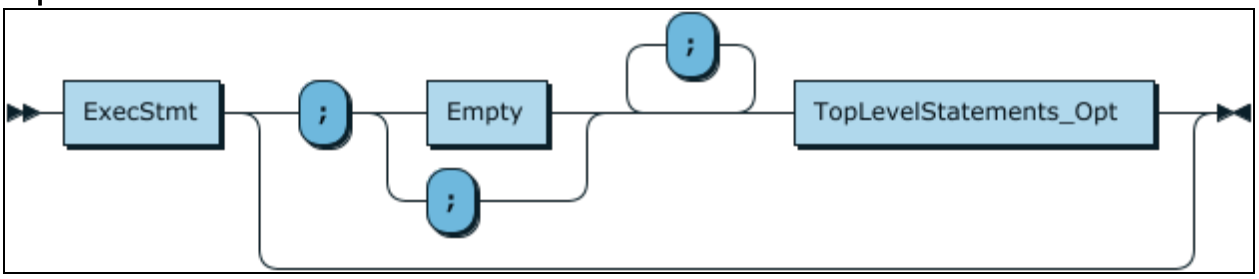
The following is the Rail Diagrams for the SQL BNF Grammar with their relations.

TopLevel:

TopLevel ::= TopLevelStatements

no references

TopLevelStatements:

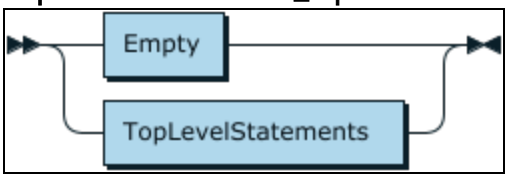



```
TopLevelStatements
    ::= ExecStmt ( ';' ( Empty | ';' ) ';'* TopLevelStatements_Opt )?
```

referenced by:

- TopLevel
- TopLevelStatements_Opt

TopLevelStatements_Opt:

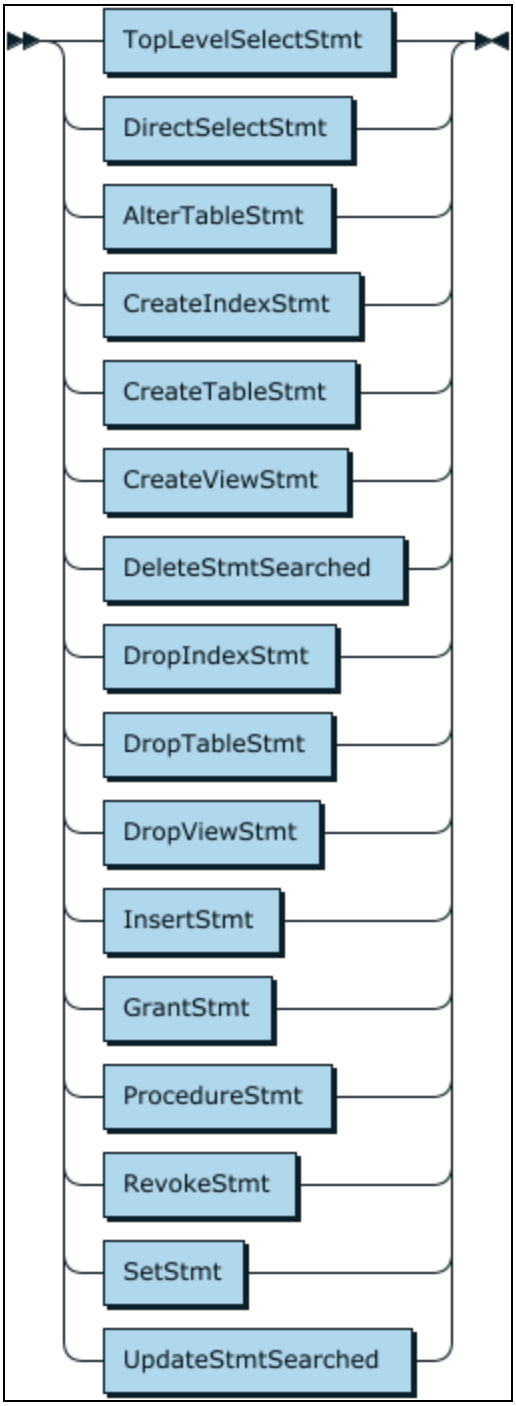



```
TopLevelStatements_Opt
    ::= Empty
     | TopLevelStatements
```

referenced by:

- TopLevelStatements

ExecStmt:



ExecStmt ::= TopLevelSelectStmt

```
    | DirectSelectStmt
    | AlterTableStmt
    | CreateIndexStmt
    | CreateTableStmt
    | CreateViewStmt
    | DeleteStmtSearched
    | DropIndexStmt
    | DropTableStmt
    | DropViewStmt
    | InsertStmt
    | GrantStmt
    | ProcedureStmt
    | RevokeStmt
    | SetStmt
    | UpdateStmtSearched
```

referenced by:

- TopLevelStatements

Empty:

```
Empty ::=
```

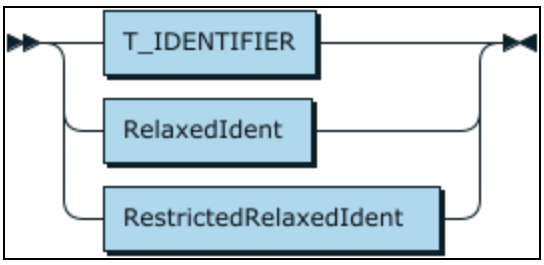
referenced by:

- AS_Clause_Opt
- AS_Opt
- CastFormat_Opt
- ColumnConstraintDefinition_Opt
- CreateTableStmt
- CreateViewStmt
- DataType

- DataTypeAttributeList_Opt
- DerivedColumnList_Opt
- ElseClause_Opt
- EscapeCharacter_Opt
- GrantOptionFor_Opt
- GroupBy_Opt
- Having_Opt
- Index_Type_Opt
- IntervalFractionalSecondsPrecision_Opt
- IntervalLeadingPrecision_Opt
- IntervalSign_Opt
- Limit_Opt
- OrderingSpecification_Opt
- ParamList_Opt
- Params_Opt
- PrivilegeColumnList_Opt
- ProcedureParams_Opt
- SelectLimit_Opt
- SetQuantifier_Opt
- ToDay_Opt
- ToHour_Opt
- ToMinute_Opt
- ToMonth_Opt
- ToSecond_Opt
- ToYear_Opt
- TopLevelStatements

- TopLevelStatements_Opt
- Unique_Opt
- WhereSearchCond_Opt
- WithGrantOption_Opt

Identifier:



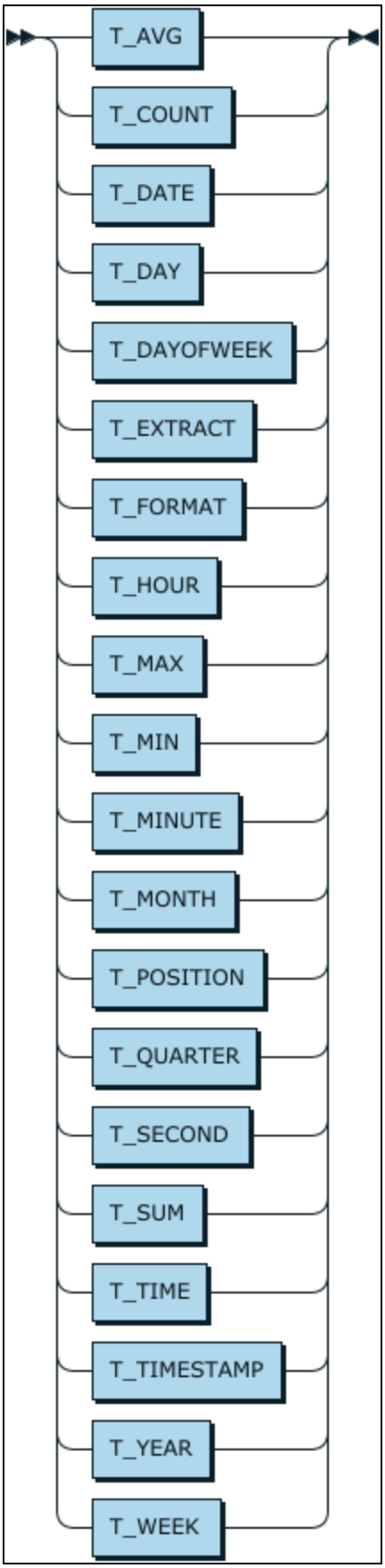

```
Identifier
    ::= T_IDENTIFIER
     | RelaxedIdent
     | RestrictedRelaxedIdent
```

referenced by:

- AuthorizationIdentifier
- CastParams
- ColumnName
- ColumnReference
- CorrelationName
- CreateIndexStmt
- DataType
- DataTypeAttributeValue
- DateTimeTSValueEscape
- DropIndexStmt
- Index_Type_Opt
- OrderColumn

- OuterJoinEscape
- ParameterName
- QualifiedIdentifier
- QualifiedName
- QualifiedProcedureName
- SQLDataType
- ScalarFn
- SelectSubListItem

RelaxedIdent:


T_AVG
T_COUNT
T_DATE
T_DAY
T_DAYOFWEEK
T_EXTRACT
T_FORMAT
T_HOUR
T_MAX
T_MIN
T_MINUTE
T_MONTH
T_POSITION
T_QUARTER
T_SECOND
T_SUM
T_TIME
T_TIMESTAMP
T_YEAR
T_WEEK

```
RelaxedIdent
    ::= T_AVG
     | T_COUNT
     | T_DATE
     | T_DAY
     | T_DAYOFWEEK
     | T_EXTRACT
     | T_FORMAT
     | T_HOUR
     | T_MAX
     | T_MIN
     | T_MINUTE
     | T_MONTH
     | T_POSITION
     | T_QUARTER
     | T_SECOND
     | T_SUM
     | T_TIME
     | T_TIMESTAMP
     | T_YEAR
     | T_WEEK
```

referenced by:

- Identifier
- RestrictedColumnName

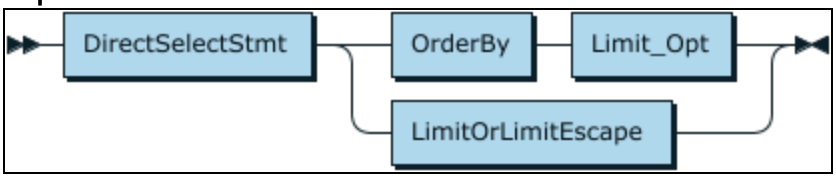
RestrictedRelaxedIdent:


```
RestrictedRelaxedIdent
    ::= T_USER
```

referenced by:

- Identifier

TopLevelSelectStmt:



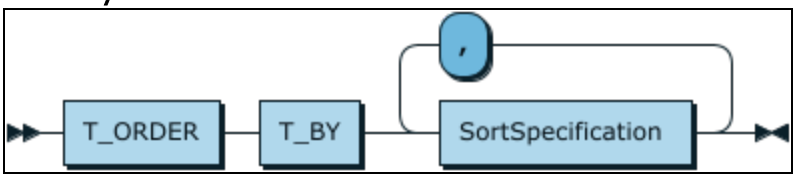
TopLevelSelectStmt

::= DirectSelectStmt (OrderByLimit_Opt | LimitOrLimitEscape)

referenced by:

- ExecStmt

OrderBy:



OrderBy ::= T_ORDER T_BY SortSpecification (',' SortSpecification)*

referenced by:

- SubQuery
- TopLevelSelectStmt

SortSpecification:

SortSpecification

::= SortKey OrderingSpecification_Opt

referenced by:

- OrderBy

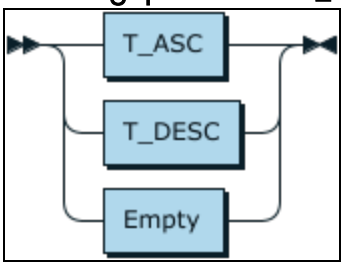
SortKey:


```
SortKey ::= Expression
```

referenced by:

- SortSpecification

OrderingSpecification_Opt:

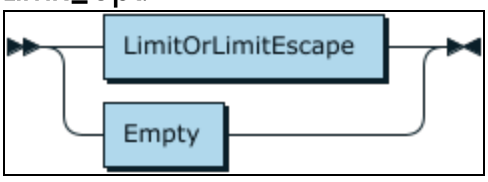



```
OrderingSpecification_Opt
    ::= T_ASC
     | T_DESC
     | Empty
```

referenced by:

- OrderColumn
- SortSpecification

Limit_Opt:



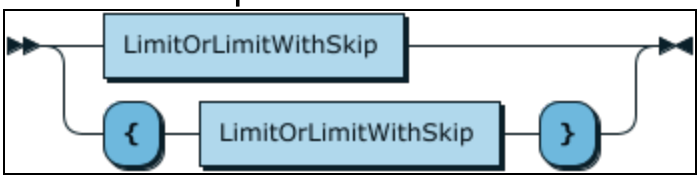

```
Limit_Opt
    ::= LimitOrLimitEscape
```

```
    | Empty
```

referenced by:

- NonJoinQueryExpression
- SubQuery
- TopLevelSelectStmt

LimitOrLimitEscape:

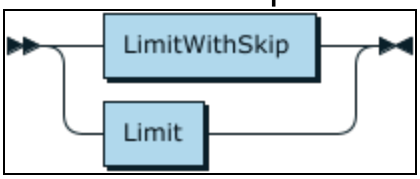



```
LimitOrLimitEscape
    ::= LimitOrLimitWithSkip
     | '{' LimitOrLimitWithSkip '}'
```

referenced by:

- Limit_Opt
- SubQuery
- TopLevelSelectStmt

LimitOrLimitWithSkip:

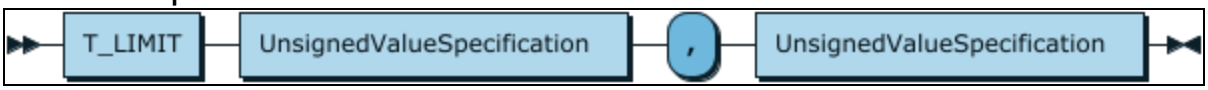



```
LimitOrLimitWithSkip
    ::= LimitWithSkip
     | Limit
```

referenced by:

- LimitOrLimitEscape

LimitWithSkip:




```
LimitWithSkip
    ::= T_LIMIT UnsignedValueSpecification ',' UnsignedValueSpecification
```

referenced by:

- LimitOrLimitWithSkip

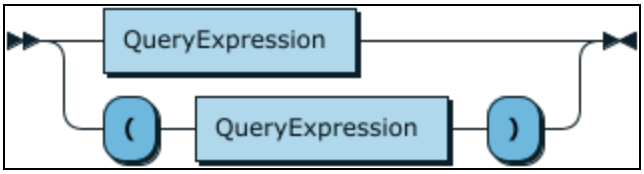
Limit:


```
Limit   ::= T_LIMIT UnsignedValueSpecification
```

referenced by:

- LimitOrLimitWithSkip

DirectSelectStmt:




```
DirectSelectStmt
    ::= QueryExpression
     | '(' QueryExpression ')'
```

referenced by:

- ExecStmt
- NonJoinQueryExpression
- TopLevelSelectStmt

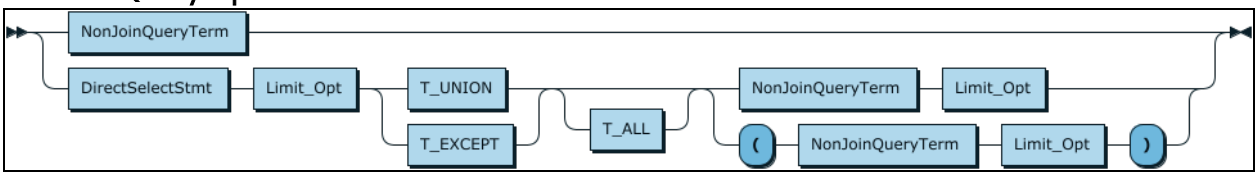
QueryExpression:


```
QueryExpression
    ::= NonJoinQueryExpression
```

referenced by:

- DirectSelectStmt
- InsertList
- SubQuery

NonJoinQueryExpression:




```
NonJoinQueryExpression
    ::= NonJoinQueryTerm
     | DirectSelectStmtLimit_Opt ( T_UNION | T_EXCEPT ) T_ALL? ( NonJoinQueryTermLimit_Opt | '('
                                 NonJoinQueryTermLimit_Opt ')' )
```

referenced by:

- QueryExpression

NonJoinQueryTerm:


```
NonJoinQueryTerm
    ::= NonJoinQueryPrimary
```

referenced by:

- NonJoinQueryExpression

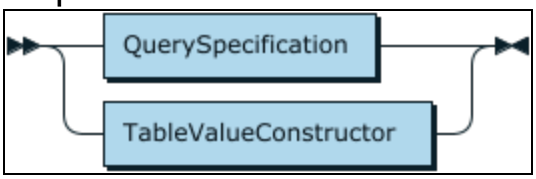
NonJoinQueryPrimary:


```
NonJoinQueryPrimary
    ::= SimpleTable
```

referenced by:

- NonJoinQueryTerm

SimpleTable:

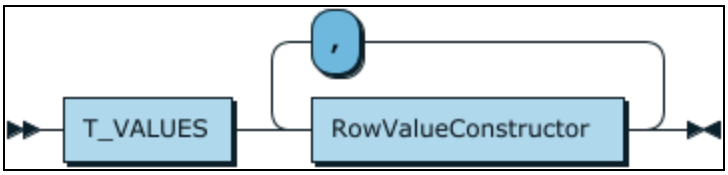



```
SimpleTable
    ::= QuerySpecification
     | TableValueConstructor
```

referenced by:

- NonJoinQueryPrimary

TableValueConstructor:

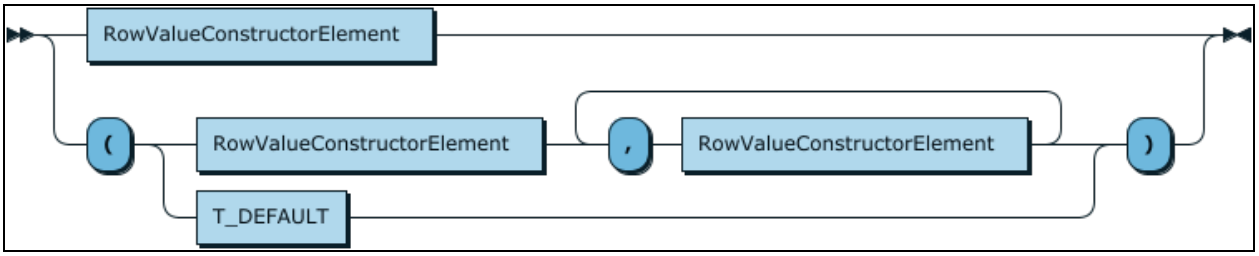



```
TableValueConstructor
    ::= T_VALUESRowValueConstructor ( ',' RowValueConstructor )*
```

referenced by:

- SimpleTable

RowValueConstructor:

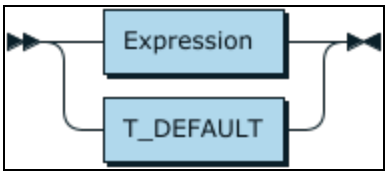



```
RowValueConstructor
    ::= RowValueConstructorElement
     | '(' ( RowValueConstructorElement ( ',' RowValueConstructorElement )+ | T_DEFAULT ) ')'
```

referenced by:

- BetweenPredicate
- ComparisonPredicate
- InPredicate
- NullPredicate
- QuantifiedComparisonPredicate
- TableValueConstructor

RowValueConstructorElement:

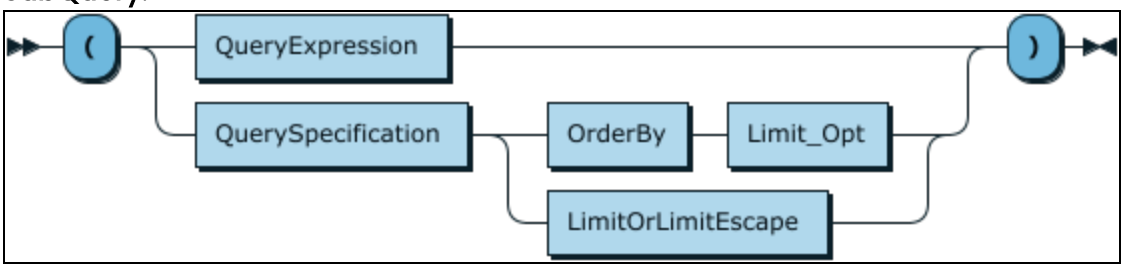



```
RowValueConstructorElement
      ::= Expression
       | T_DEFAULT
```

referenced by:

- RowValueConstructor

SubQuery:

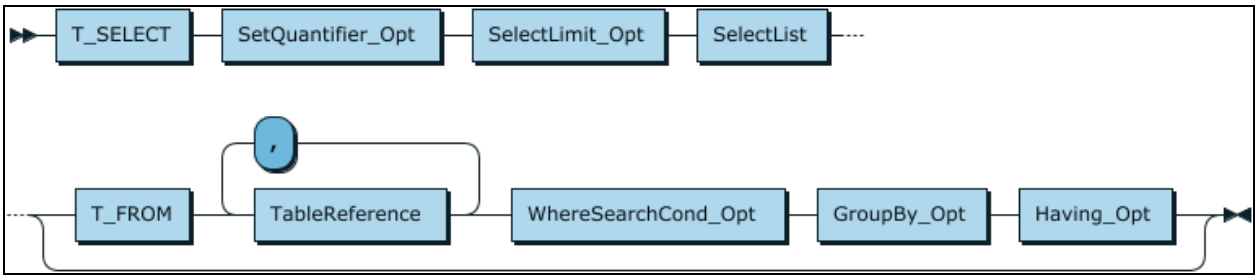



```
SubQuery ::= '(' ( QueryExpression | QuerySpecification ( OrderByLimit_Opt | LimitOrLimitEscape ) ) ')'
```

referenced by:

- ExistsPredicate
- InPredicateValue
- QuantifiedComparisonPredicate
- TableReference
- ValueExpressionPrimary

QuerySpecification:

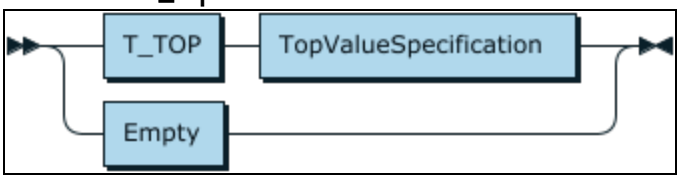



```
QuerySpecification
    ::= T_SELECT SetQuantifier_Opt SelectLimit_Opt SelectList ( T_FROMTableReference ( ','
                                TableReference )* WhereSearchCond_Opt GroupBy_Opt Having_Opt )?
```

referenced by:

- CreateViewStmt
- SimpleTable
- SubQuery

SelectLimit_Opt:

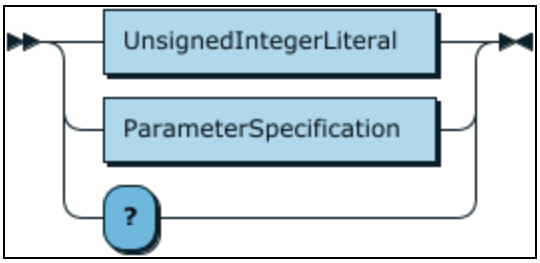



```
SelectLimit_Opt
    ::= T_TOP TopValueSpecification
     | Empty
```

referenced by:

- QuerySpecification

TopValueSpecification:

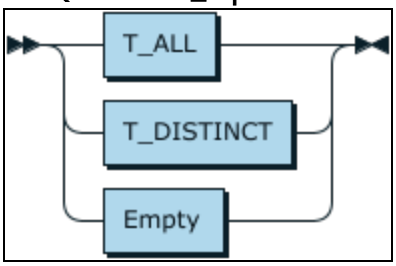



```
TopValueSpecification
    ::= UnsignedIntegerLiteral
     | ParameterSpecification
     | '?'
```

referenced by:

- SelectLimit_Opt

SetQuantifier_Opt:

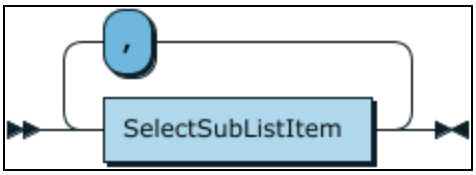



```
SetQuantifier_Opt
    ::= T_ALL
     | T_DISTINCT
     | Empty
```

referenced by:

- QuerySpecification
- SetFunction

SelectList:

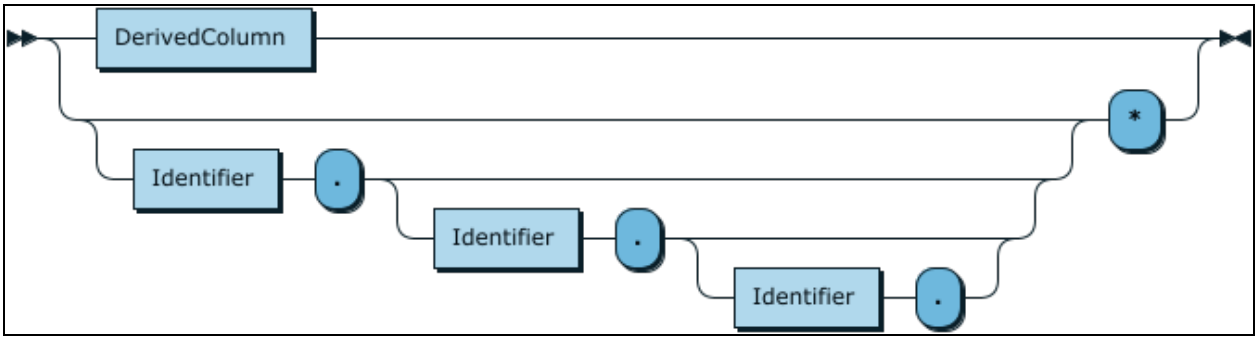



```
SelectList
    ::= SelectSubListItem ( ',' SelectSubListItem )*
```

referenced by:

- QuerySpecification

SelectSubListItem:

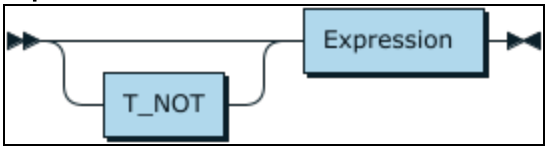



```
SelectSubListItem
    ::= DerivedColumn
     | ( Identifier '.' ( Identifier '.' ( Identifier '.' )? )? )? '*'
```

referenced by:

- SelectList

ExpressionGeneral:



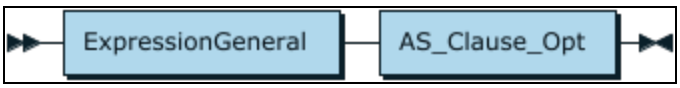

```
ExpressionGeneral
```

```
    ::= T_NOT? Expression
```

referenced by:

- DerivedColumn

DerivedColumn:

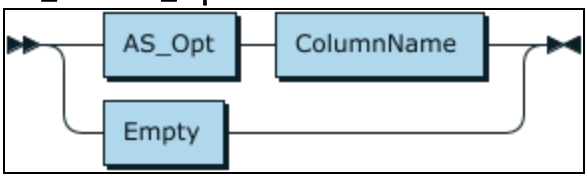



```
DerivedColumn
    ::= ExpressionGeneral AS_Clause_Opt
```

referenced by:

- SelectSubListItem

AS_Clause_Opt:

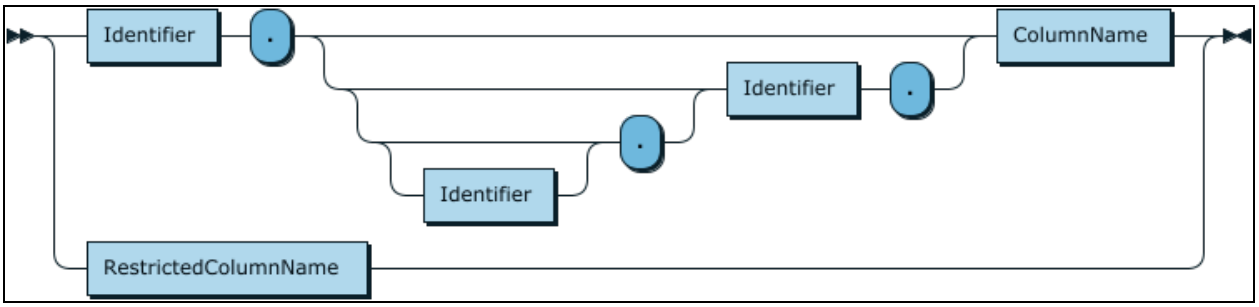



```
AS_Clause_Opt
    ::= AS_Opt ColumnName
     | Empty
```

referenced by:

- DerivedColumn

ColumnReference:

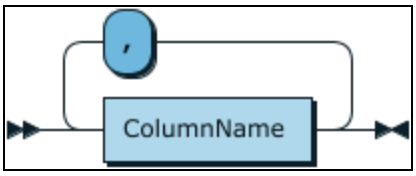



```
ColumnReference
    ::= Identifier '.' ( ( Identifier? '.' )? Identifier '.' )? ColumnName
     | RestrictedColumnName
```

referenced by:

- ValueExpressionPrimary

ColumnNameList:




```
ColumnNameList
    ::= ColumnName ( ',' ColumnName )*
```

referenced by:

- CreateViewStmt
- DerivedColumnList_Opt
- InsertList
- PrivilegeColumnList_Opt
- TableConstraintDefinition

ColumnName:


```
ColumnName
    ::= Identifier
```

referenced by:

- AS_Clause_Opt
- ColumnDefinition
- ColumnNameList
- ColumnReference
- SetClause

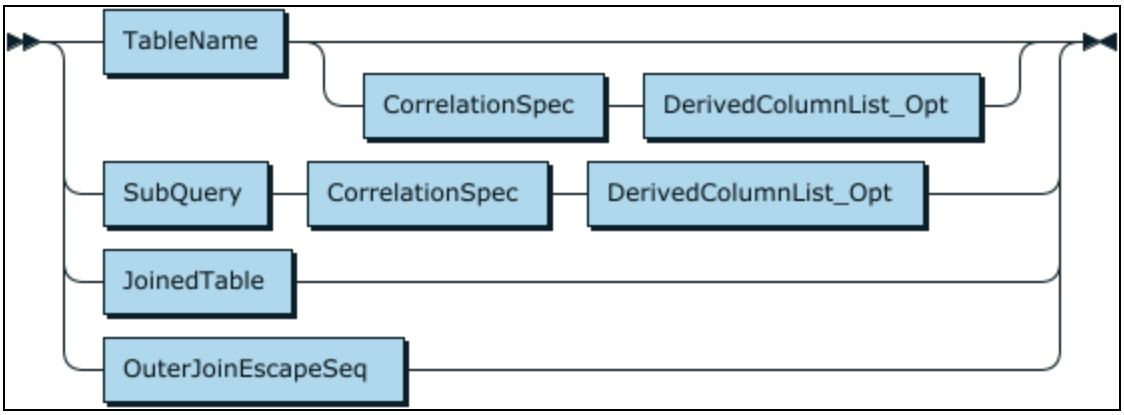
RestrictedColumnName:


```
RestrictedColumnName
    ::= T_IDENTIFIER
     | RelaxedIdent
```

referenced by:

- ColumnReference

TableReference:

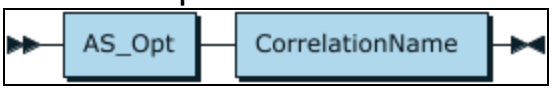



```
TableReference
    ::= TableName ( CorrelationSpecDerivedColumnList_Opt )?
     | SubQuery CorrelationSpec DerivedColumnList_Opt
     | JoinedTable
     | OuterJoinEscapeSeq
```

referenced by:

- JoinedTable
- QualifiedJoin
- QuerySpecification

CorrelationSpec:




```
CorrelationSpec
    ::= AS_Opt CorrelationName
```

referenced by:

- TableReference

CorrelationName:


```
CorrelationName
        ::= Identifier
```

referenced by:

- CorrelationSpec

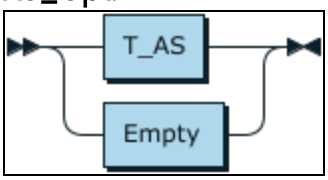
DerivedColumnList_Opt:


```
DerivedColumnList_Opt
        ::= '(' ColumnNameList ')'
          | Empty
```

referenced by:

- TableReference

AS_Opt:

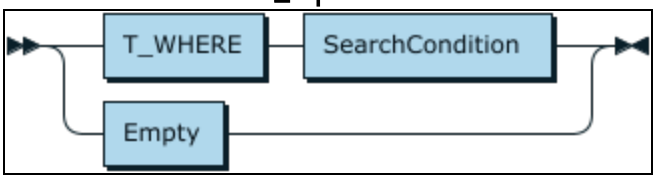



```
AS_Opt  ::= T_AS
          | Empty
```

referenced by:

- AS_Clause_Opt
- CorrelationSpec

WhereSearchCond_Opt:

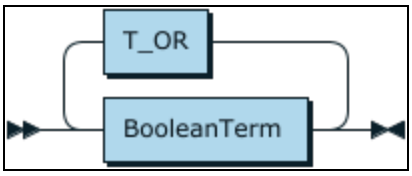



```
WhereSearchCond_Opt
    ::= T_WHERE SearchCondition
     | Empty
```

referenced by:

- DeleteStmtSearched
- QuerySpecification
- UpdateStmtSearched

SearchCondition:



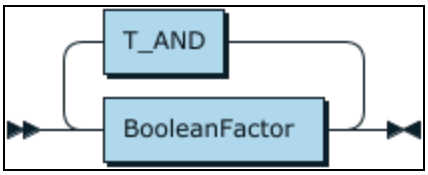

```
SearchCondition
    ::= BooleanTerm ( T_ORBooleanTerm )*
```

referenced by:

- BooleanPrimary
- Having_Opt
- QualifiedJoin

- SearchedWhenClause
- WhereSearchCond_Opt

BooleanTerm:

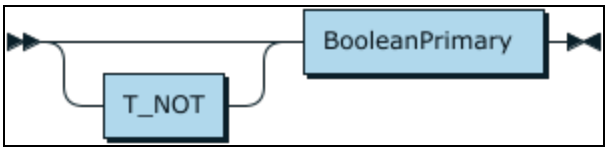



```
BooleanTerm
    ::= BooleanFactor ( T_AND BooleanFactor )*
```

referenced by:

- CastExpression
- SearchCondition

BooleanFactor:

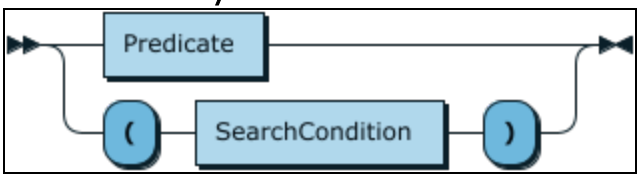



```
BooleanFactor
    ::= T_NOT? BooleanPrimary
```

referenced by:

- BooleanTerm

BooleanPrimary:

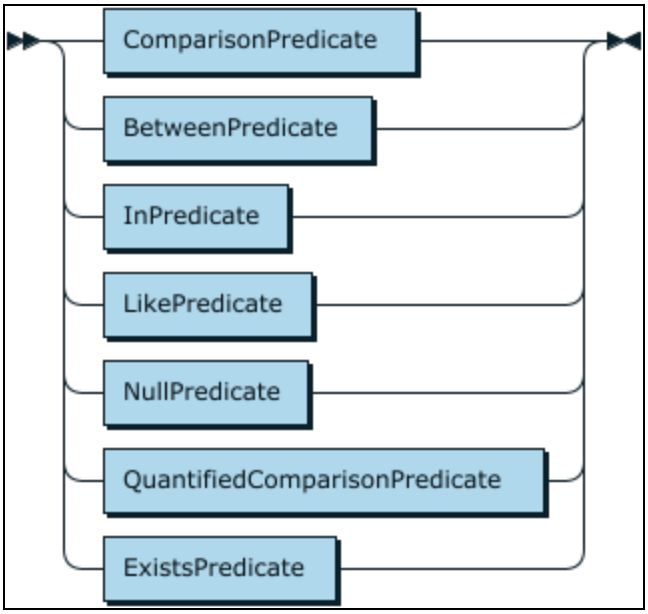


```
BooleanPrimary
    ::= Predicate
     | '(' SearchCondition ')'
```

referenced by:

- BooleanFactor

Predicate:




```
Predicate
    ::= ComparisonPredicate
     | BetweenPredicate
     | InPredicate
     | LikePredicate
     | NullPredicate
     | QuantifiedComparisonPredicate
     | ExistsPredicate
```

referenced by:

- BooleanPrimary

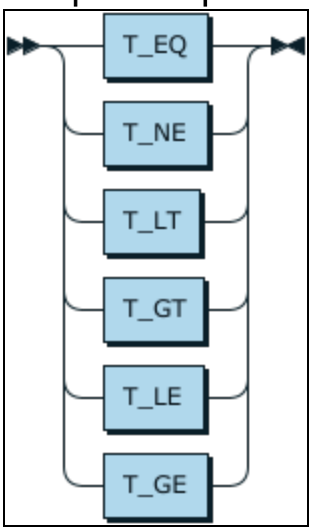
ComparisonPredicate:


```
ComparisonPredicate
    ::= RowValueConstructor ComparisonOp RowValueConstructor
```

referenced by:

- Predicate

ComparisonOp:

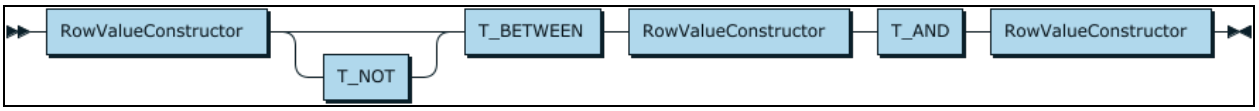



```
ComparisonOp
    ::= T_EQ
     | T_NE
     | T_LT
     | T_GT
     | T_LE
     | T_GE
```

referenced by:

- ComparisonPredicate
- QuantifiedComparisonPredicate

BetweenPredicate:



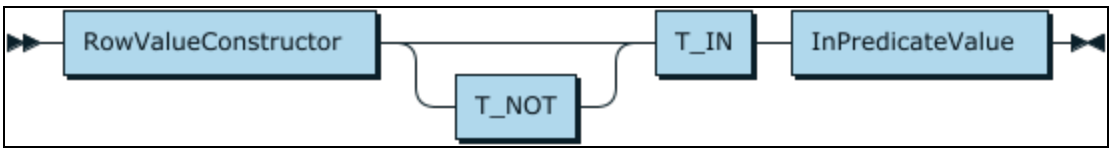
BetweenPredicate

::= RowValueConstructor T_NOT? T_BETWEEN RowValueConstructor T_AND RowValueConstructor

referenced by:

- Predicate

InPredicate:



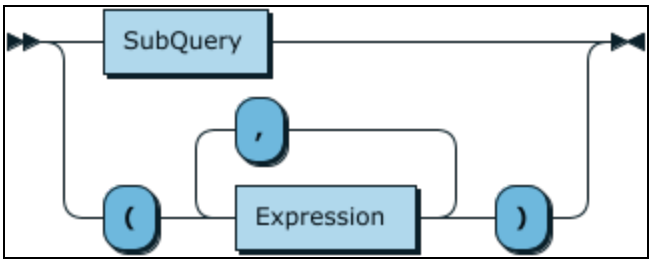
InPredicate

::= RowValueConstructor T_NOT? T_ININ PredicateValue

referenced by:

- Predicate

InPredicateValue:

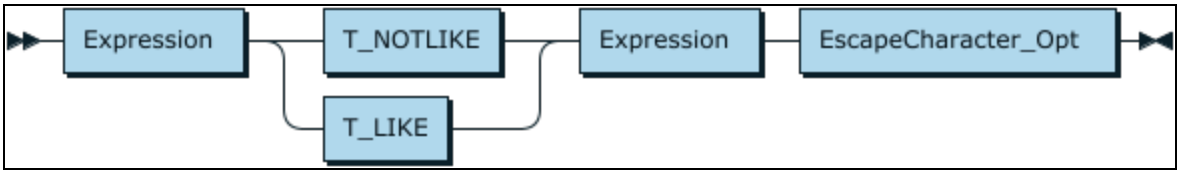


```
InPredicateValue
    ::= SubQuery
     | '(' Expression ( ',' Expression )* ')'
```

referenced by:

- InPredicate

LikePredicate:

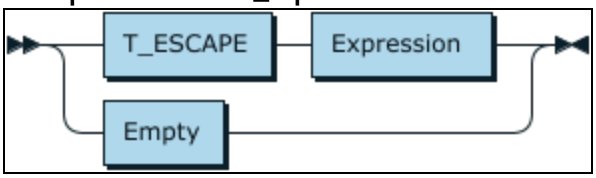



```
LikePredicate
    ::= Expression ( T_NOTLIKE | T_LIKE ) Expression EscapeCharacter_Opt
```

referenced by:

- Predicate

EscapeCharacter_Opt:

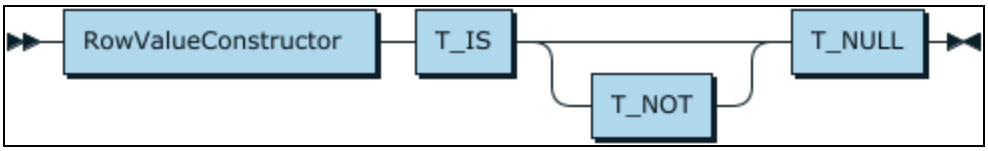



```
EscapeCharacter_Opt
    ::= T_ESCAPE Expression
     | Empty
```

referenced by:

- LikePredicate

NullPredicate:



NullPredicate

::= RowValueConstructor T_IS T_NOT? T_NULL

referenced by:

- Predicate

QuantifiedComparisonPredicate:

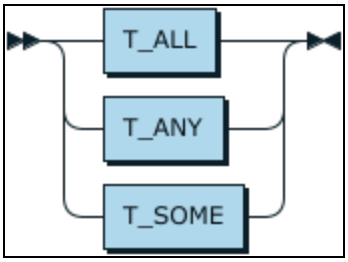
QuantifiedComparisonPredicate

::= RowValueConstructor ComparisonOp Quantifier SubQuery

referenced by:

- Predicate

Quantifier:



Quantifier

::= T_ALL

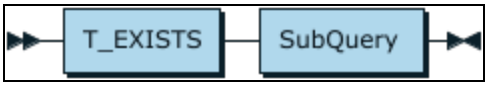
| T_ANY

| T_SOME

referenced by:

- QuantifiedComparisonPredicate

ExistsPredicate:




```
ExistsPredicate
    ::= T_EXISTS SubQuery
```

referenced by:

- Predicate

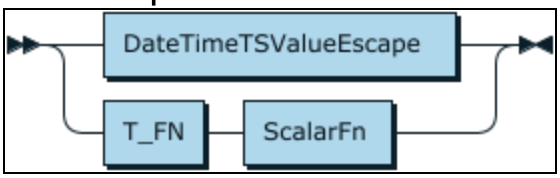
ValueEscapeSeq:


```
ValueEscapeSeq
    ::= '{' ValueEscape '}'
```

referenced by:

- ValueExpressionPrimary

ValueEscape:

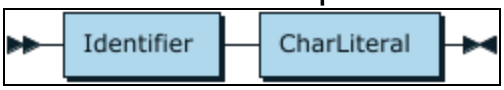



```
ValueEscape
    ::= DateTimeTSValueEscape
     | T_FN ScalarFn
```

referenced by:

- ValueEscapeSeq

DateTimeTSValueEscape:

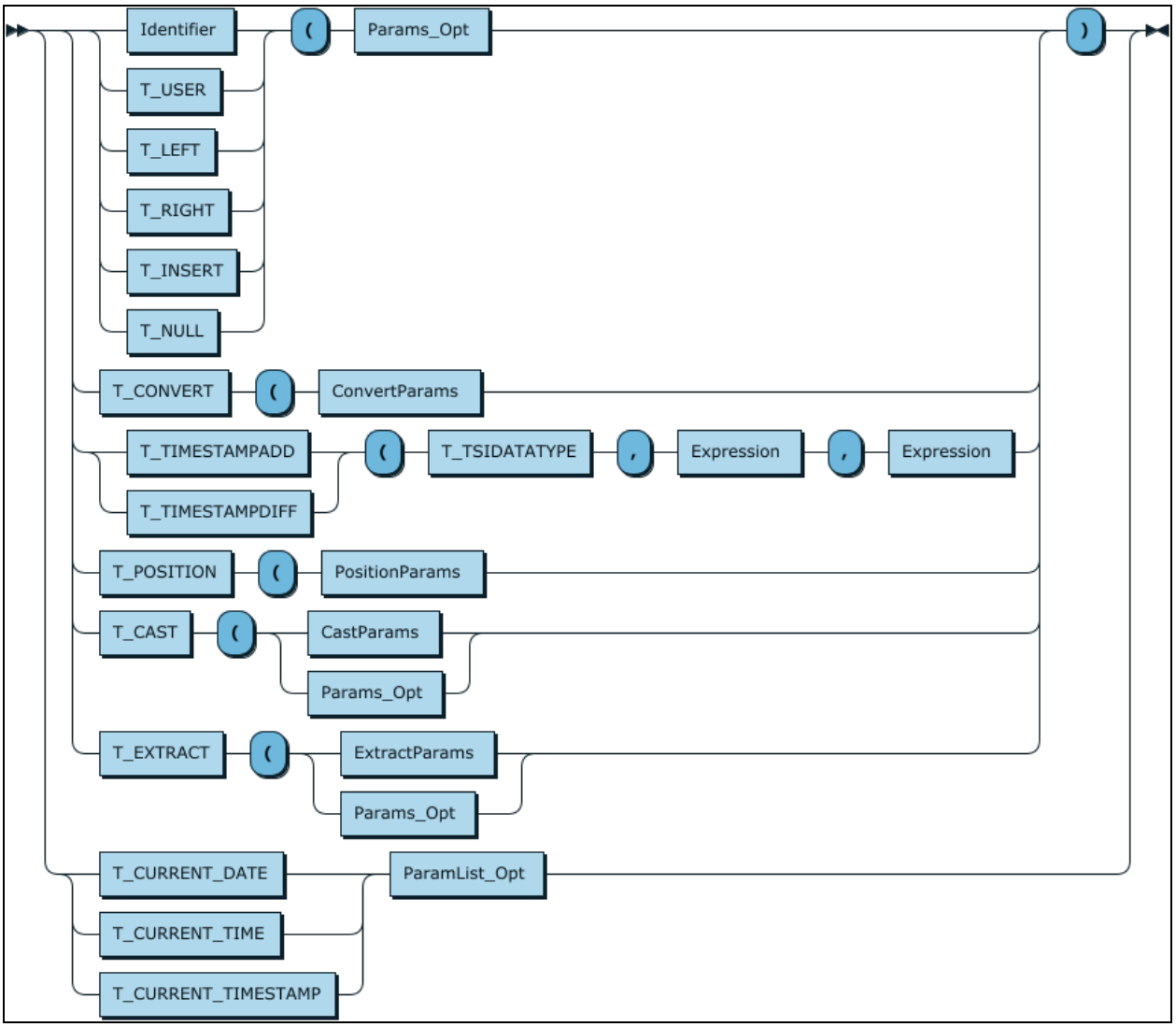



```
DateTimeTSValueEscape
    ::= Identifier CharLiteral
```

referenced by:

- ValueEscape

ScalarFn:

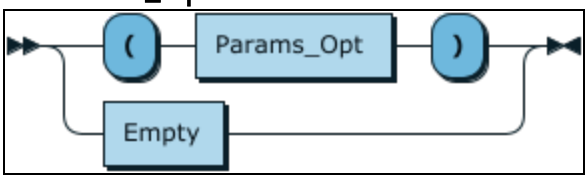



```
ScalarFn ::= ( ( Identifier | T_USER | T_LEFT | T_RIGHT | T_INSERT | T_NULL ) '(' Params_Opt | T_CONVERT
                                '(' ConvertParams | ( T_TIMESTAMPADD | T_TIMESTAMPDIFF ) '(' T_
                                TSIDATATYPE ',' Expression ',' Expression | T_POSITION '(' PositionParams | T_
                                CAST '(' ( CastParams | Params_Opt ) | T_EXTRACT '(' ( ExtractParams | Params_
                                Opt ) ) ')'
        | ( T_CURRENT_DATE | T_CURRENT_TIME | T_CURRENT_TIMESTAMP ) ParamList_Opt
```

referenced by:

- ValueEscape
- ValueExpressionPrimary

ParamList_Opt:

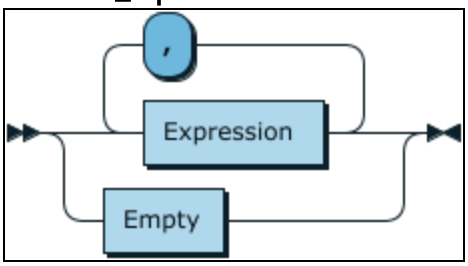



```
ParamList_Opt
    ::= '(' Params_Opt ')'
     | Empty
```

referenced by:

- ScalarFn

Params_Opt:

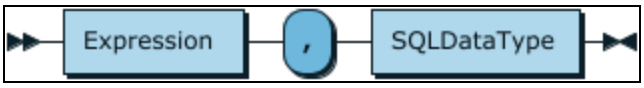



```
Params_Opt
    ::= Expression ( ',' Expression )*
     | Empty
```

referenced by:

- ParamList_Opt
- ScalarFn

ConvertParams:



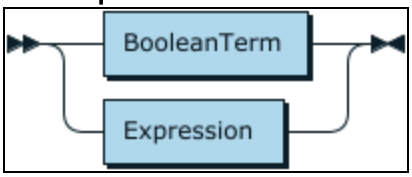

```
ConvertParams
```

::= Expression ',' SQLDataType

referenced by:

- ScalarFn

CastExpression:



CastExpression

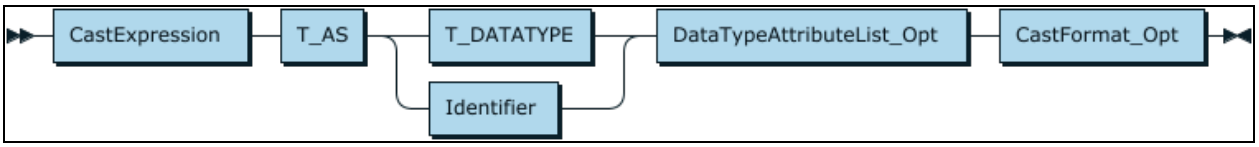
::= BooleanTerm

| Expression

referenced by:

- CastParams

CastParams:



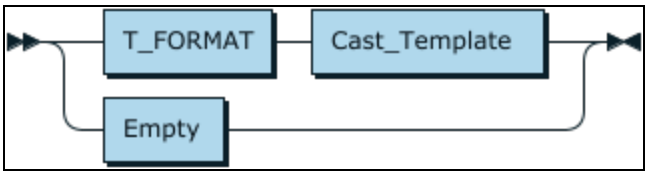
CastParams

::= CastExpressionT_AS (T_DATATYPE | Identifier) DataTypeAttributeList_OptCastFormat_Opt

referenced by:

- ScalarFn

CastFormat_Opt:




```
CastFormat_Opt
    ::= T_FORMAT Cast_Template
     | Empty
```

referenced by:

- CastParams

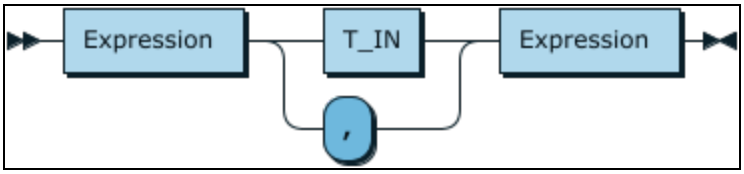
Cast_Template:


```
Cast_Template
    ::= T_CHAR_STR_LIT
```

referenced by:

- CastFormat_Opt

PositionParams:

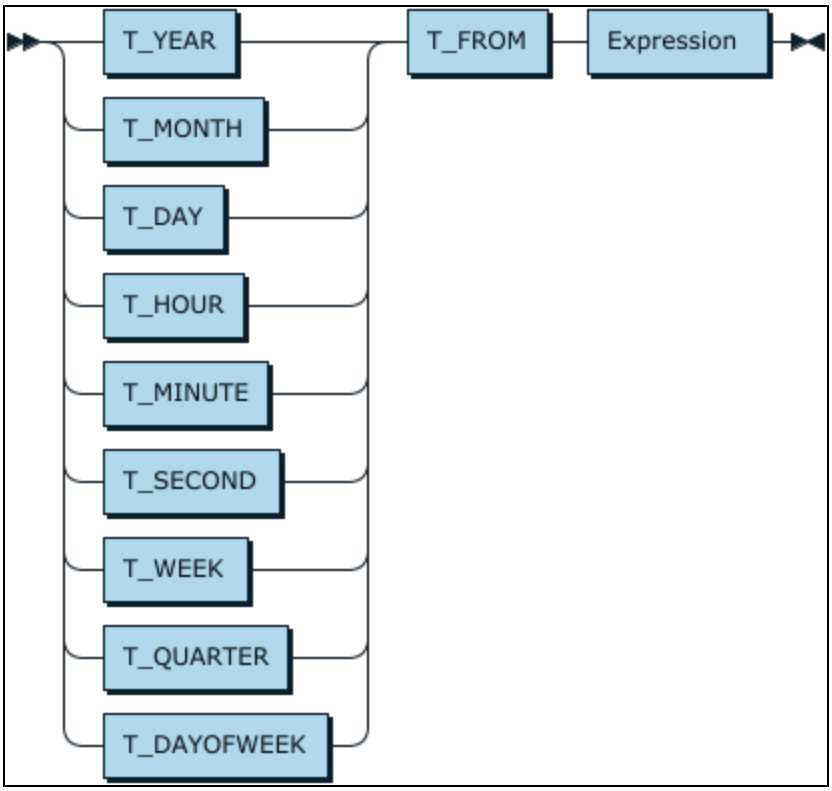



```
PositionParams
    ::= Expression ( T_IN | ',' ) Expression
```

referenced by:

- ScalarFn

ExtractParams:



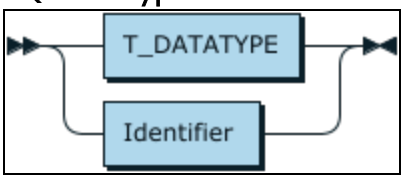
ExtractParams

::= (T_YEAR | T_MONTH | T_DAY | T_HOUR | T_MINUTE | T_SECOND | T_WEEK | T_QUARTER | T_DAYOFWEEK) T_FROMExpression

referenced by:

- ScalarFn

SQLDataType:



SQLDataType

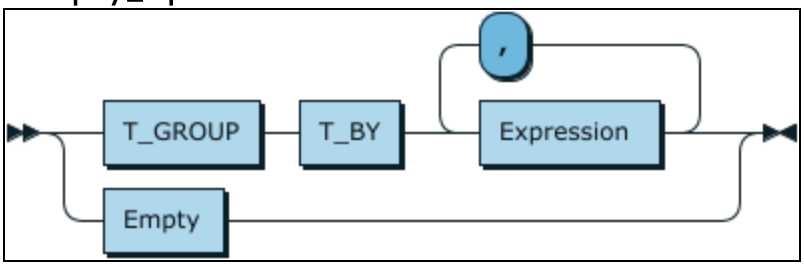
::= T_DATATYPE

| Identifier

referenced by:

- ConvertParams

GroupBy_Opt:

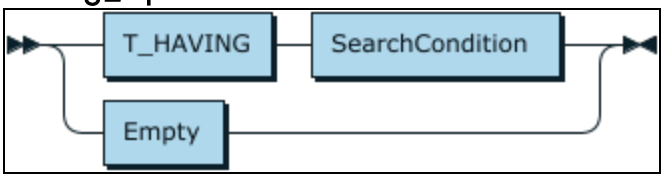



```
GroupBy_Opt
    ::= T_GROUP T_BY Expression ( ',' Expression )*
     | Empty
```

referenced by:

- QuerySpecification

Having_Opt:

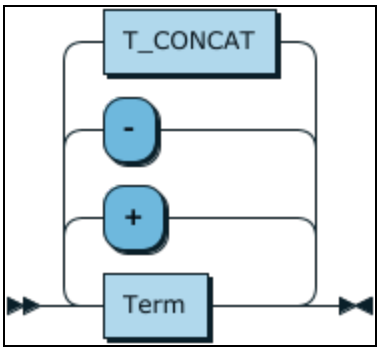



```
Having_Opt
    ::= T_HAVING SearchCondition
     | Empty
```

referenced by:

- QuerySpecification

Expression:



Expression

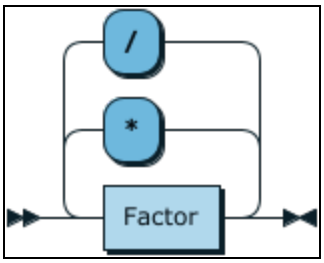
::= Term (('+' | '-' | T_CONCAT) Term)*

referenced by:

- CaseAbbreviation
- CastExpression
- CoalesceListExpression
- ConvertParams
- EscapeCharacter_Opt
- ExpressionGeneral
- ExtractParams
- GroupBy_Opt
- InPredicateValue
- LikePredicate
- Params_Opt
- PositionParams
- ProcedureParams_Opt
- Result
- RowValueConstructorElement
- ScalarFn

- SetFunction
- SimpleCase
- SimpleWhenClause
- SortKey
- UpdateSource
- ValueExpressionPrimary

Term:

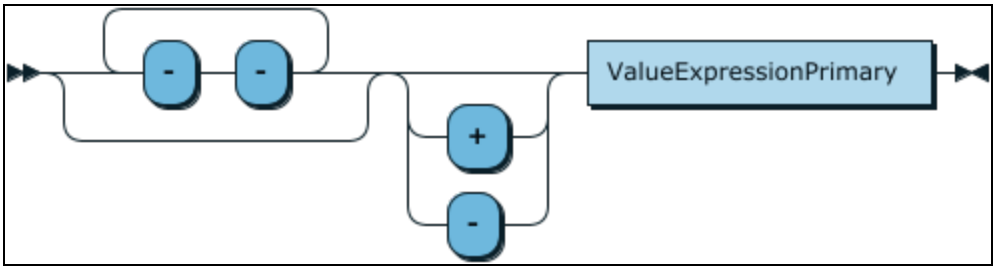



```
Term    ::= Factor ( ( '*' | '/' ) Factor )*
```

referenced by:

- Expression

Factor:

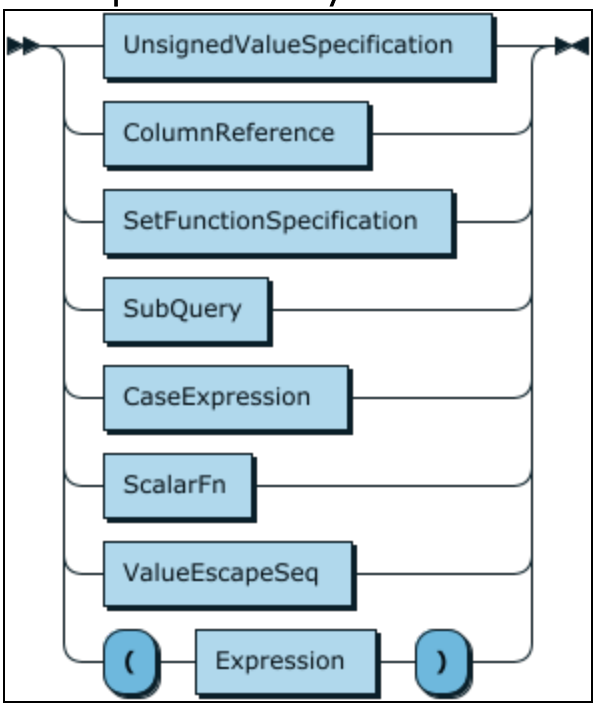



```
Factor  ::= ( '-' '-' )* ( '+' | '-' )? ValueExpressionPrimary
```

referenced by:

- Term

ValueExpressionPrimary:

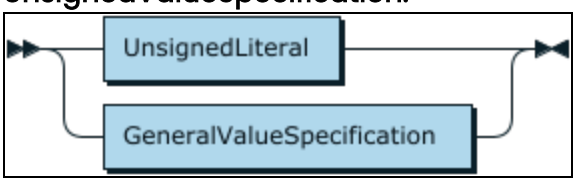



```
ValueExpressionPrimary
    ::= UnsignedValueSpecification
     | ColumnReference
     | SetFunctionSpecification
     | SubQuery
     | CaseExpression
     | ScalarFn
     | ValueEscapeSeq
     | '(' Expression ')'
```

referenced by:

- Factor

UnsignedValueSpecification:



UnsignedValueSpecification

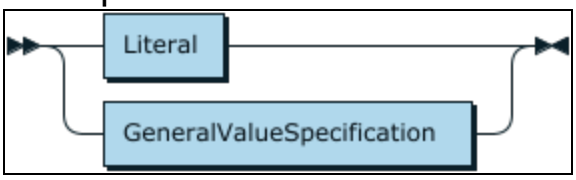
::= UnsignedLiteral

| GeneralValueSpecification

referenced by:

- Limit
- LimitWithSkip
- ValueExpressionPrimary

ValueSpecification:



ValueSpecification

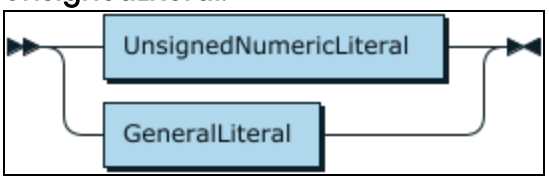
::= Literal

| GeneralValueSpecification

referenced by:

- SetStmt

UnsignedLiteral:



UnsignedLiteral

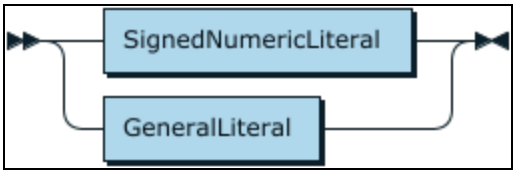
::= UnsignedNumericLiteral

| GeneralLiteral

referenced by:

- UnsignedValueSpecification

Literal:

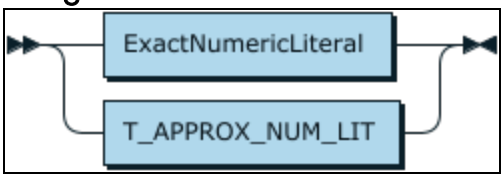



```
Literal  ::= SignedNumericLiteral
         | GeneralLiteral
```

referenced by:

- DataTypeAttributeValue
- ValueSpecification

UnsignedNumericLiteral:

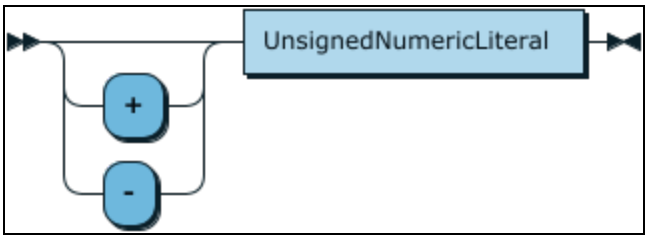



```
UnsignedNumericLiteral
       ::= ExactNumericLiteral
        | T_APPROX_NUM_LIT
```

referenced by:

- SignedNumericLiteral
- UnsignedLiteral

SignedNumericLiteral:

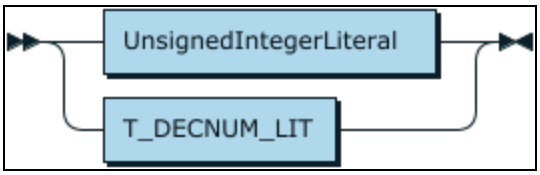



```
SignedNumericLiteral
    ::= ( '+' | '-' )? UnsignedNumericLiteral
```

referenced by:

- Literal

ExactNumericLiteral:

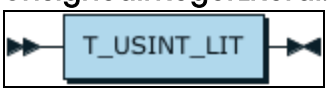



```
ExactNumericLiteral
    ::= UnsignedIntegerLiteral
     | T_DECNUM_LIT
```

referenced by:

- UnsignedNumericLiteral

UnsignedIntegerLiteral:

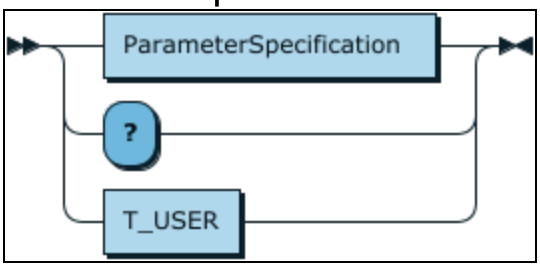



```
UnsignedIntegerLiteral
    ::= T_USINT_LIT
```

referenced by:

- ExactNumericLiteral
- TopValueSpecification

GeneralValueSpecification:

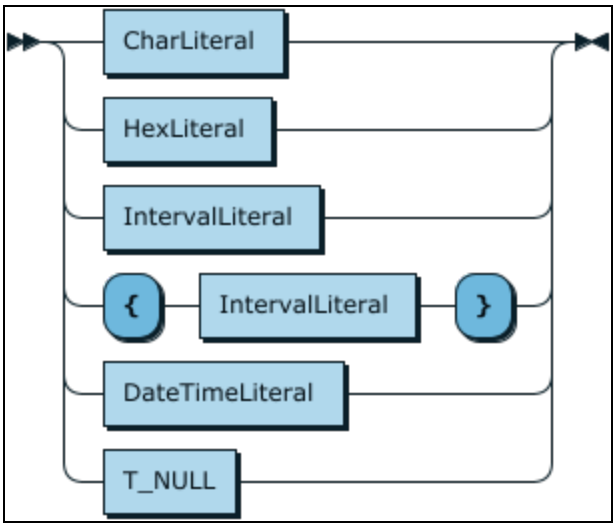



```
GeneralValueSpecification
    ::= ParameterSpecification
     | '?'
     | T_USER
```

referenced by:

- UnsignedValueSpecification
- ValueSpecification

GeneralLiteral:

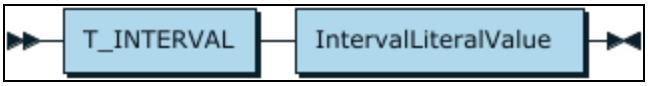


```
GeneralLiteral
    ::= CharLiteral
     | HexLiteral
     | IntervalLiteral
     | '{' IntervalLiteral '}'
     | DateTimeLiteral
     | T_NULL
```

referenced by:

- Literal
- UnsignedLiteral

IntervalLiteral:

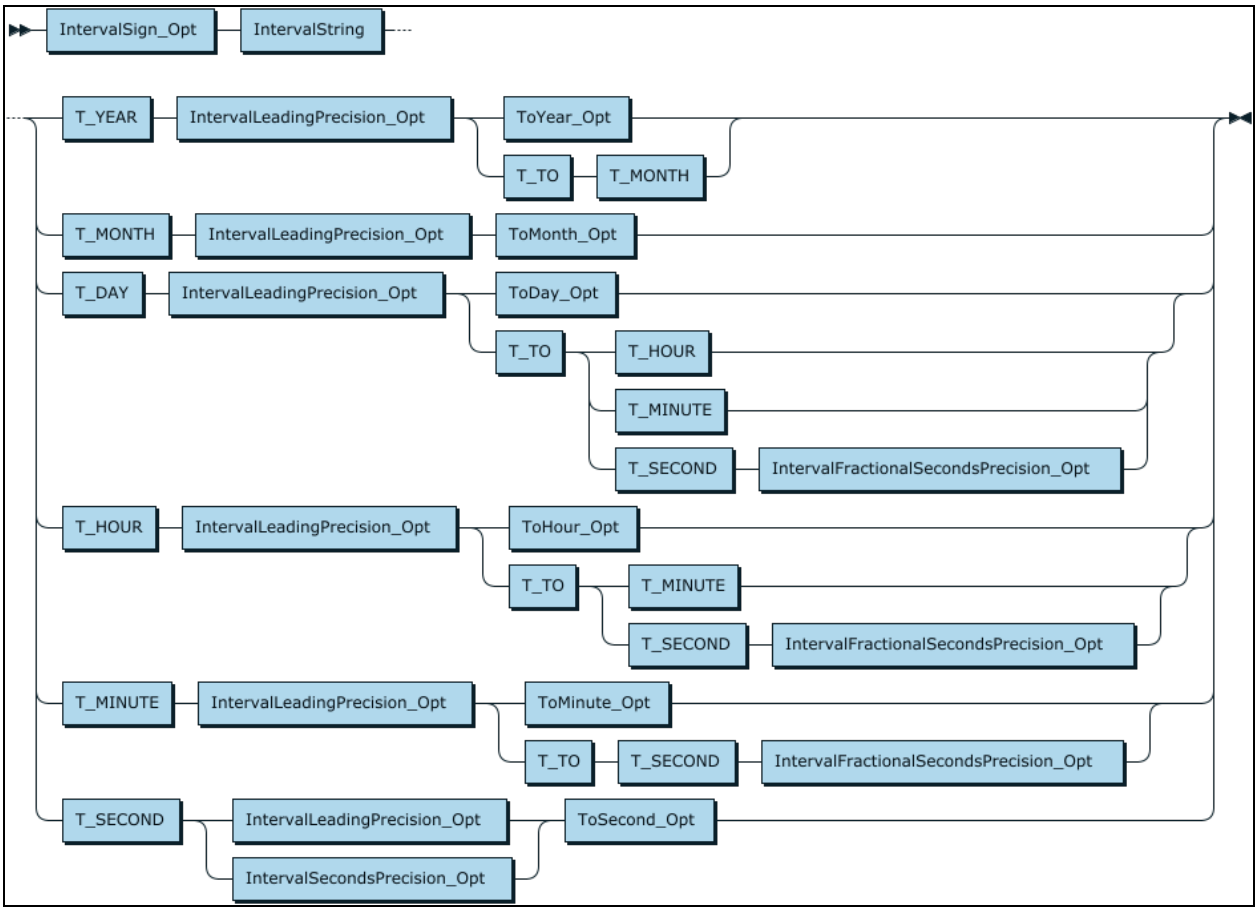



```
IntervalLiteral
    ::= T_INTERVAL IntervalLiteralValue
```

referenced by:

- GeneralLiteral

IntervalLiteralValue:




```
IntervalLiteralValue
      ::= IntervalSign_OptIntervalString
( T_YEAR IntervalLeadingPrecision_Opt
( ToYear_Opt | T_TO T_MONTH ) | T_MONTH IntervalLeadingPrecision_Opt ToMonth_Opt | T_DAY
                              IntervalLeadingPrecision_Opt
(ToDay_Opt | T_TO
(T_HOUR | T_MINUTE | T_SECOND IntervalFractionalSecondsPrecision_Opt ) ) | T_HOUR
                              IntervalLeadingPrecision_Opt
(ToHour_Opt | T_TO
(T_MINUTE | T_SECOND IntervalFractionalSecondsPrecision_Opt ) ) | T_MINUTE IntervalLeadingPrecision_
                              Opt
(ToMinute_Opt | T_TO T_SECOND IntervalFractionalSecondsPrecision_Opt ) | T_SECOND
(IntervalLeadingPrecision_Opt | IntervalSecondsPrecision_Opt ) ToSecond_Opt )
```

referenced by:

- IntervalLiteral

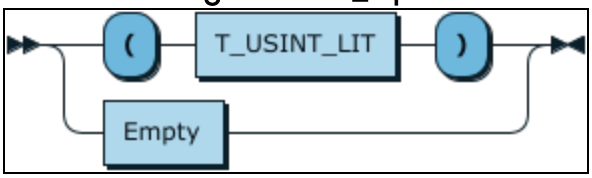
IntervalString:


```
IntervalString
    ::= CharLiteral
```

referenced by:

- IntervalLiteralValue

IntervalLeadingPrecision_Opt:

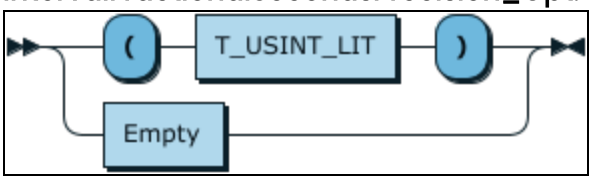



```
IntervalLeadingPrecision_Opt
    ::= '(' T_USINT_LIT ')'
     | Empty
```

referenced by:

- IntervalLiteralValue
- IntervalType

IntervalFractionalSecondsPrecision_Opt:



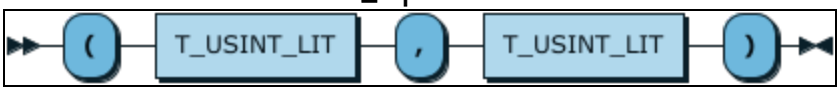

```
IntervalFractionalSecondsPrecision_Opt
```

```
::= '(' T_USINT_LIT ')'
 | Empty
```

referenced by:

- IntervalLiteralValue
- IntervalType

IntervalSecondsPrecision_Opt:

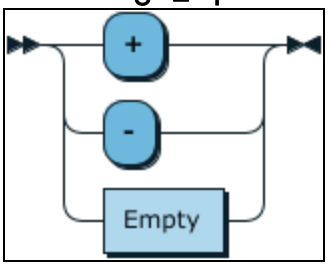



```
IntervalSecondsPrecision_Opt
    ::= '(' T_USINT_LIT ',' T_USINT_LIT ')'
```

referenced by:

- IntervalLiteralValue
- IntervalType

IntervalSign_Opt:

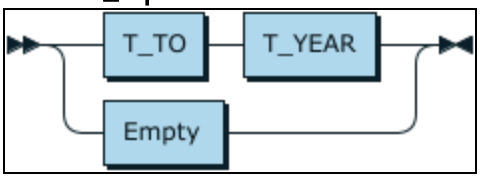



```
IntervalSign_Opt
    ::= '+'
     | '-'
     | Empty
```

referenced by:

- IntervalLiteralValue

ToYear_Opt:

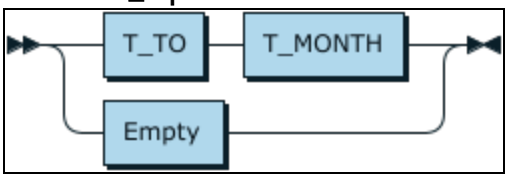



```
ToYear_Opt
    ::= T_TOT_YEAR
     | Empty
```

referenced by:

- IntervalLiteralValue
- IntervalType

ToMonth_Opt:

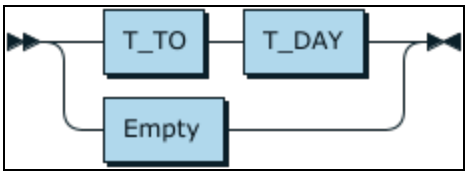



```
ToMonth_Opt
    ::= T_TO T_MONTH
     | Empty
```

referenced by:

- IntervalLiteralValue
- IntervalType

ToDay_Opt:

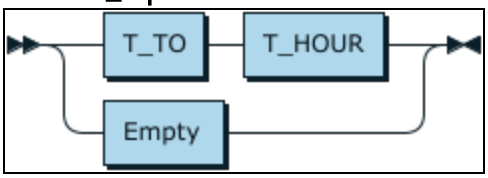



```
ToDay_Opt
      ::= T_TO T_DAY
       | Empty
```

referenced by:

- IntervalLiteralValue
- IntervalType

ToHour_Opt:

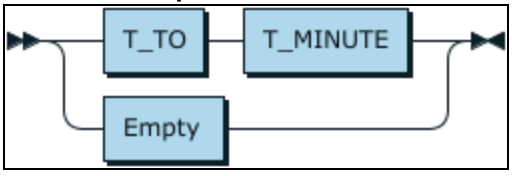



```
ToHour_Opt
      ::= T_TO T_HOUR
       | Empty
```

referenced by:

- IntervalLiteralValue
- IntervalType

ToMinute_Opt:



ToMinute_Opt

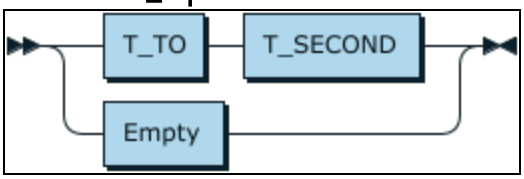
::= T_TO T_MINUTE

| Empty

referenced by:

- IntervalLiteralValue
- IntervalType

ToSecond_Opt:



ToSecond_Opt

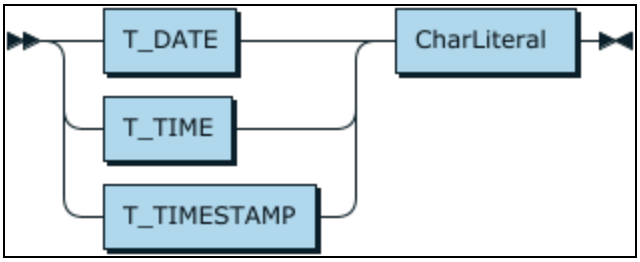
::= T_TO T_SECOND

| Empty

referenced by:

- IntervalLiteralValue
- IntervalType

DateTimeLiteral:



DateTimeLiteral

::= (T_DATE | T_TIME | T_TIMESTAMP) CharLiteral

referenced by:

- GeneralLiteral

CharLiteral:

CharLiteral

::= T_CHAR_STR_LIT

referenced by:

- DateTimeLiteral
- DateTimeTSValueEscape
- GeneralLiteral
- IntervalString

HexLiteral:

HexLiteral

::= T_HEX_LIT

referenced by:

- GeneralLiteral

ParameterSpecification:

ParameterSpecification

::= ParameterName

referenced by:

- GeneralValueSpecification
- TopValueSpecification

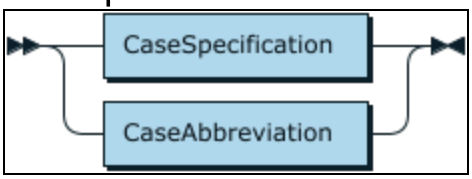
ParameterName:


```
ParameterName
    ::= ':' Identifier
```

referenced by:

- ParameterSpecification

CaseExpression:

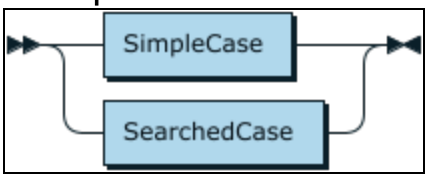



```
CaseExpression
    ::= CaseSpecification
     | CaseAbbreviation
```

referenced by:

- ValueExpressionPrimary

CaseSpecification:

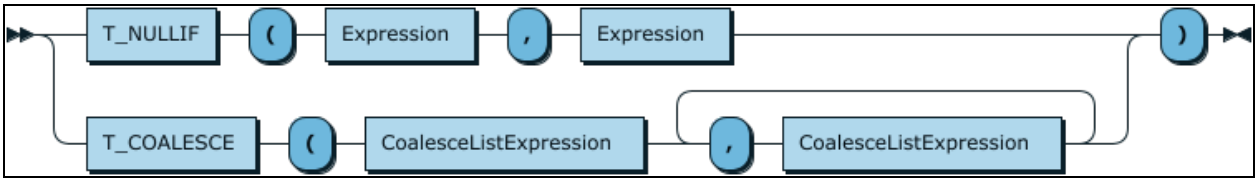


```
CaseSpecification
    ::= SimpleCase
     | SearchedCase
```

referenced by:

- CaseExpression

CaseAbbreviation:




```
CaseAbbreviation
    ::= ( T_NULLIF '(' Expression ',' Expression | T_COALESCE '(' CoalesceListExpression ( ','
                                CoalesceListExpression )+ ) ')'
```

referenced by:

- CaseExpression

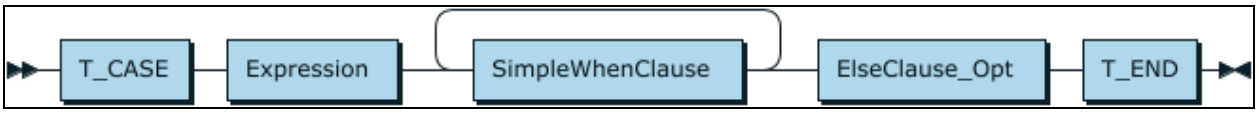
CoalesceListExpression:


```
CoalesceListExpression
    ::= Expression
```

referenced by:

- CaseAbbreviation

SimpleCase:



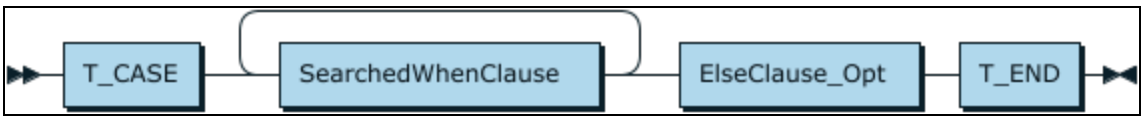
SimpleCase

::= T_CASE Expression SimpleWhenClause+ ElseClause_Opt T_END

referenced by:

- CaseSpecification

SearchedCase:



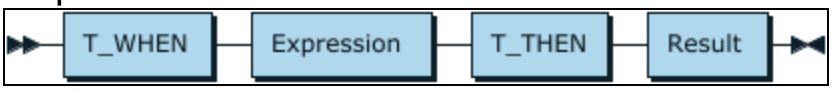
SearchedCase

::= T_CASE SearchedWhenClause+ ElseClause_Opt T_END

referenced by:

- CaseSpecification

SimpleWhenClause:



SimpleWhenClause

::= T_WHEN Expression T_THEN Result

referenced by:

- SimpleCase

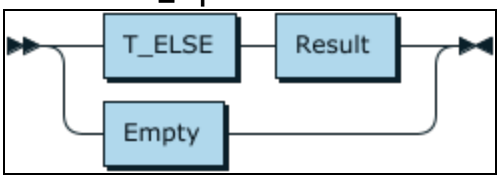
SearchedWhenClause:

```
SearchedWhenClause
    ::= T_WHEN SearchCondition T_THEN Result
```

referenced by:

- SearchedCase

ElseClause_Opt:




```
ElseClause_Opt
    ::= T_ELSE Result
     | Empty
```

referenced by:

- SearchedCase
- SimpleCase

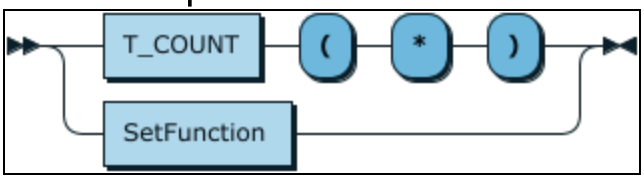
Result:


```
Result  ::= Expression
```

referenced by:

- ElseClause_Opt
- SearchedWhenClause
- SimpleWhenClause

SetFunctionSpecification:

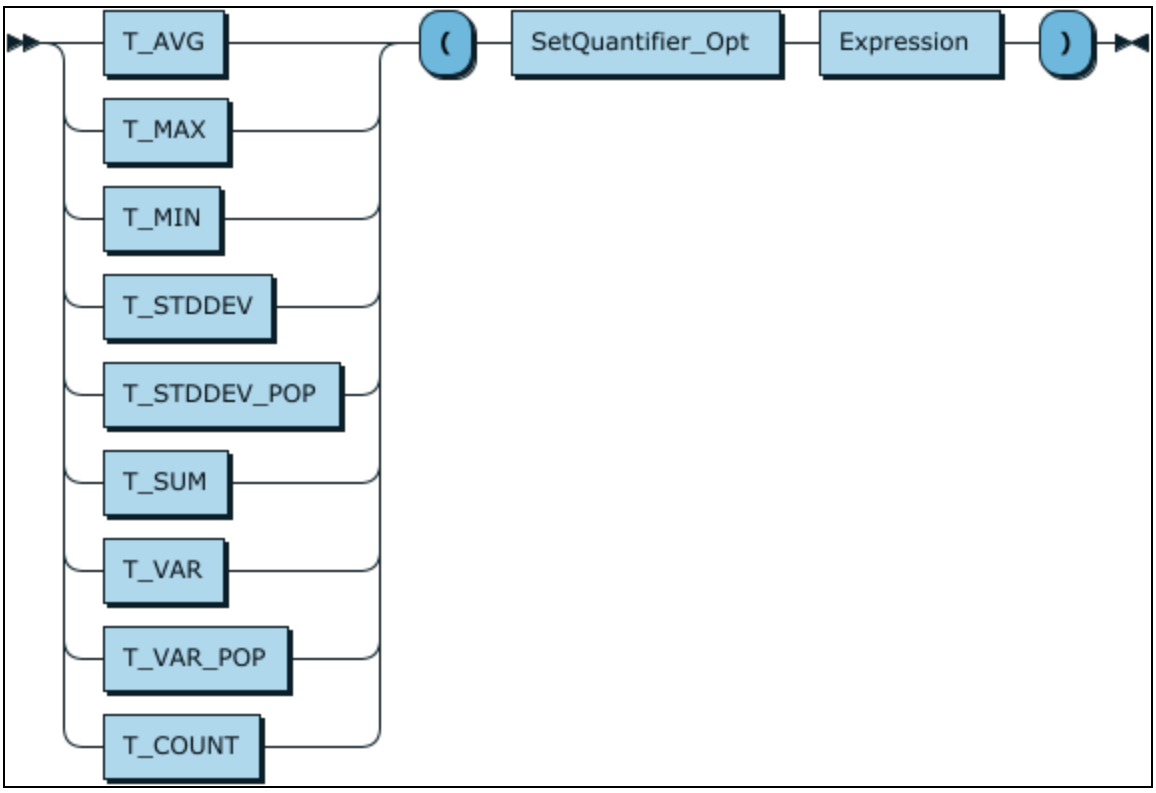



```
SetFunctionSpecification
    ::= T_COUNT '(' '*' ')'
     | SetFunction
```

referenced by:

- ValueExpressionPrimary

SetFunction:

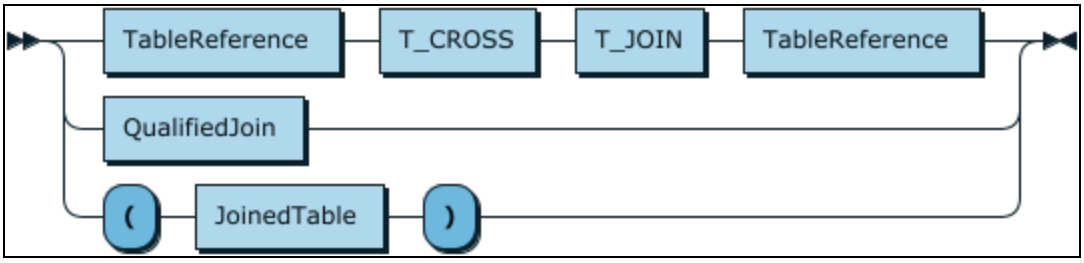



```
SetFunction
    ::= ( T_AVG | T_MAX | T_MIN | T_STDDEV | T_STDDEV_POP | T_SUM | T_VAR | T_VAR_POP | T_
                        COUNT ) '(' SetQuantifier_OptExpression ')'
```

referenced by:

- SetFunctionSpecification

JoinedTable:

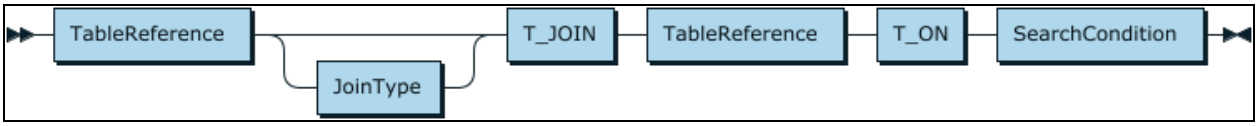



```
JoinedTable
    ::= TableReference T_CROSS T_JOIN TableReference
     | QualifiedJoin
     | '(' JoinedTable ')'
```

referenced by:

- JoinedTable
- OuterJoinEscape
- TableReference

QualifiedJoin:

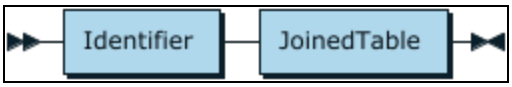



```
QualifiedJoin
    ::= TableReference JoinType? T_JOIN TableReference T_ON SearchCondition
```

referenced by:

- JoinedTable

OuterJoinEscape:




```
OuterJoinEscape
    ::= Identifier JoinedTable
```

referenced by:

- OuterJoinEscapeSeq

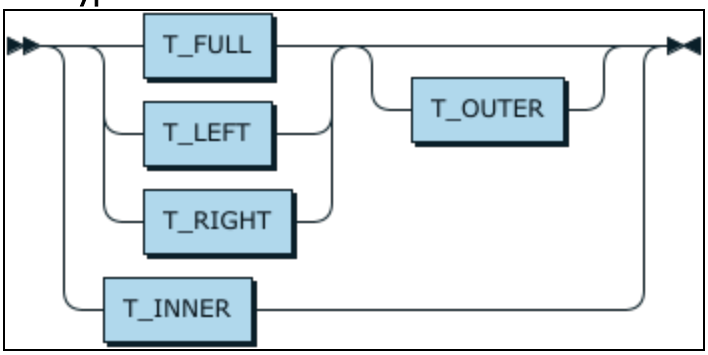
OuterJoinEscapeSeq:


```
OuterJoinEscapeSeq
    ::= '{' OuterJoinEscape '}'
```

referenced by:

- TableReference

JoinType:

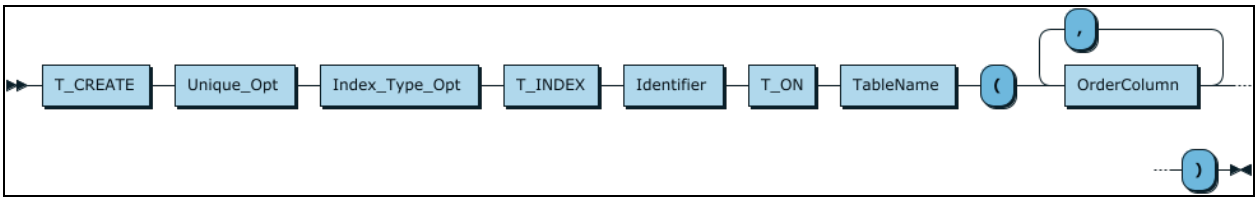



```
JoinType ::= ( T_FULL | T_LEFT | T_RIGHT ) T_OUTER?
    | T_INNER
```

referenced by:

- QualifiedJoin

CreateIndexStmt:

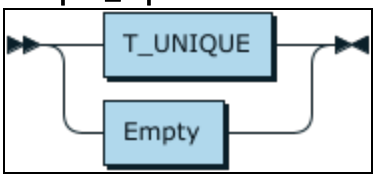



```
CreateIndexStmt
    ::= T_CREATE Unique_Opt Index_Type_Opt T_INDEX Identifier T_ON TableName '(' OrderColumn ( ','
                                OrderColumn )* ')'
```

referenced by:

- ExecStmt

Unique_Opt:

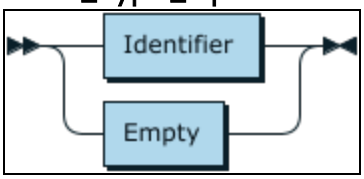



```
Unique_Opt
    ::= T_UNIQUE
     | Empty
```

referenced by:

- CreateIndexStmt

Index_Type_Opt:




```
Index_Type_Opt
    ::= Identifier
```

| Empty

referenced by:

- CreateIndexStmt

OrderColumn:

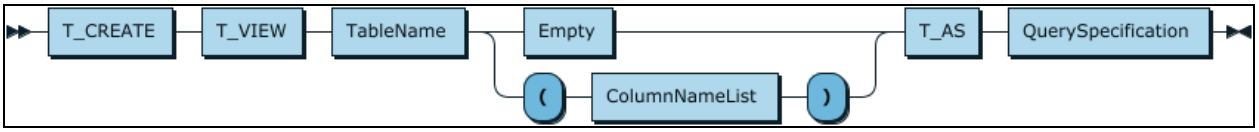
OrderColumn

::= Identifier OrderingSpecification_Opt

referenced by:

- CreateIndexStmt

CreateViewStmt:



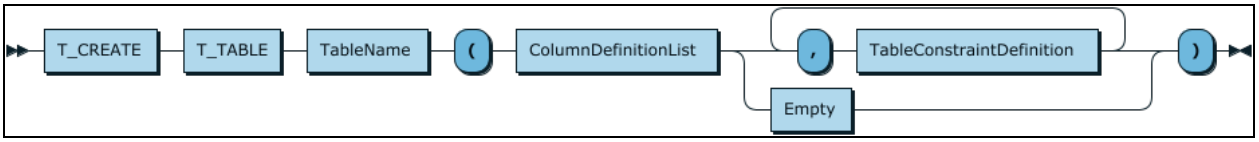
CreateViewStmt

::= T_CREATE T_VIEW TableName (Empty | '(' ColumnNameList ')') T_AS QuerySpecification

referenced by:

- ExecStmt

CreateTableStmt:



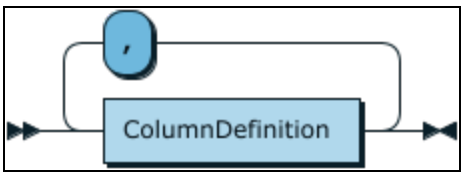
CreateTableStmt

::= T_CREATE T_TABLE TableName '(' ColumnDefinitionList ((',' TableConstraintDefinition)+ | Empty) ')'

referenced by:

- ExecStmt

ColumnDefinitionList:



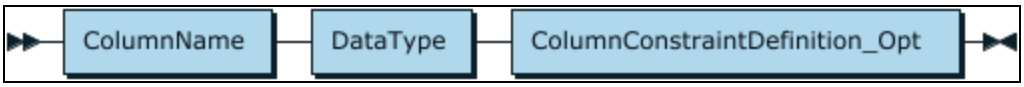
ColumnDefinitionList

::= ColumnDefinition (',' ColumnDefinition)*

referenced by:

- CreateTableStmt

ColumnDefinition:



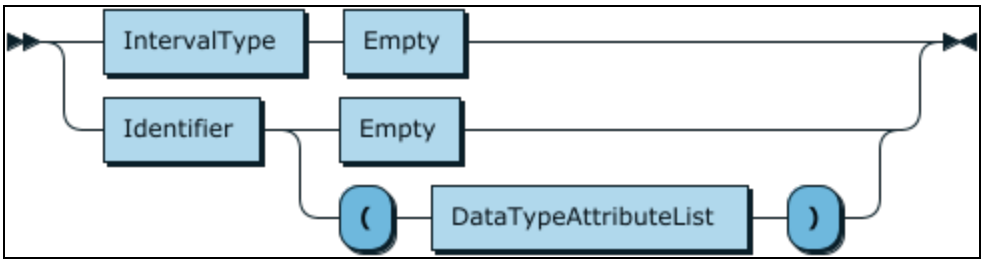
ColumnDefinition

::= ColumnName DataType ColumnConstraintDefinition_Opt

referenced by:

- AddColumnDefinition
- ColumnDefinitionList

DataType:

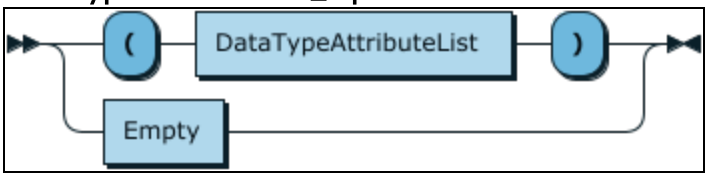



```
DataType ::= IntervalType Empty
        | Identifier ( Empty | '(' DataTypeAttributeList ')' )
```

referenced by:

- ColumnDefinition

DataTypeAttributeList_Opt:

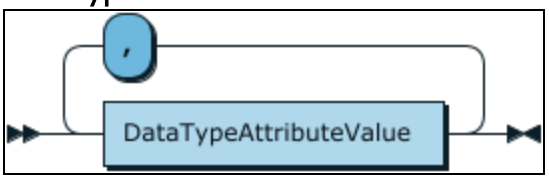



```
DataTypeAttributeList_Opt
        ::= '(' DataTypeAttributeList ')'
         | Empty
```

referenced by:

- CastParams

DataTypeAttributeList:

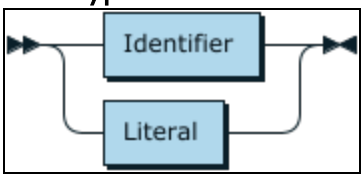



```
DataTypeAttributeList
        ::= DataTypeAttributeValue ( ',' DataTypeAttributeValue )*
```

referenced by:

- DataType
- DataTypeAttributeList_Opt

DataTypeAttributeValue:

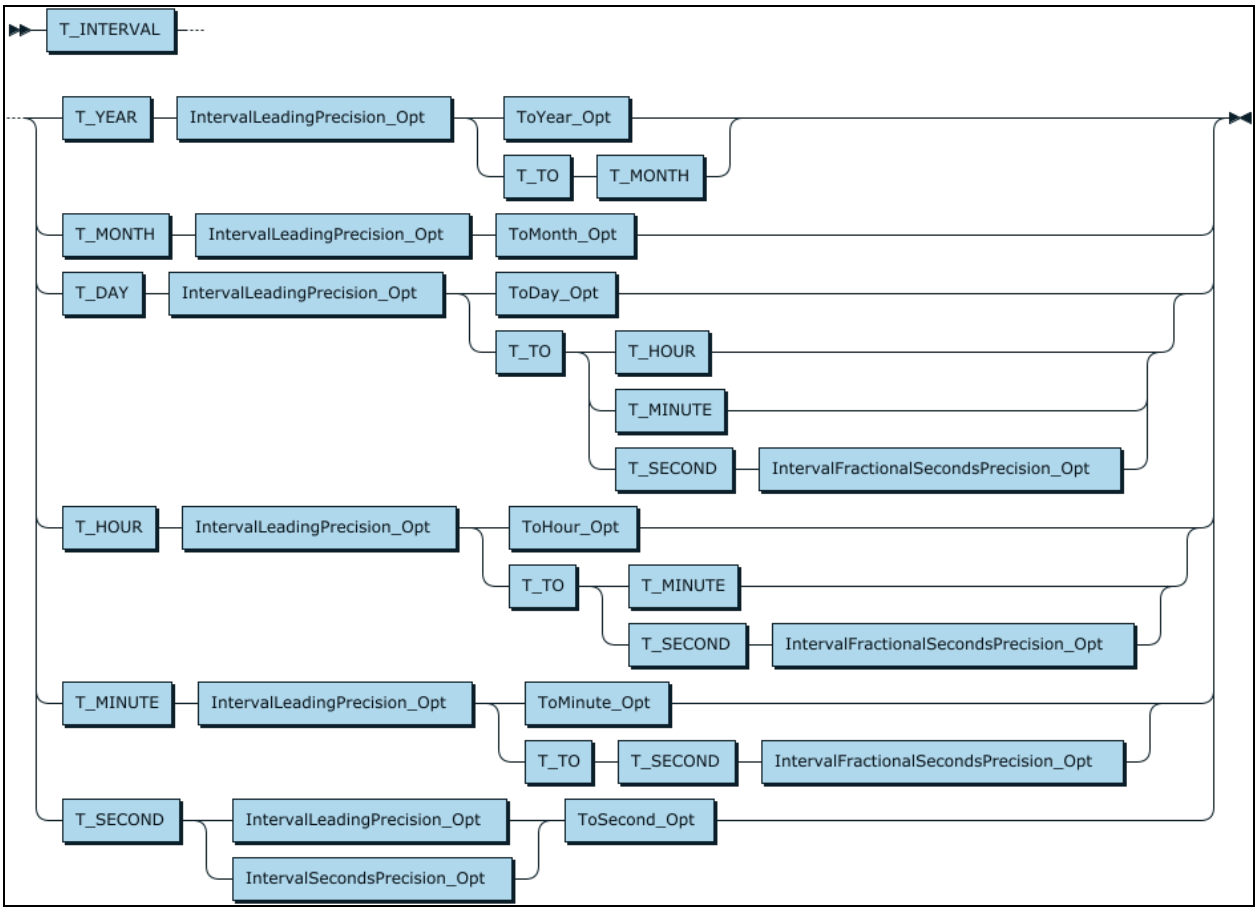



```
DataTypeAttributeValue
    ::= Identifier
     | Literal
```

referenced by:

- DataTypeAttributeList

IntervalType:



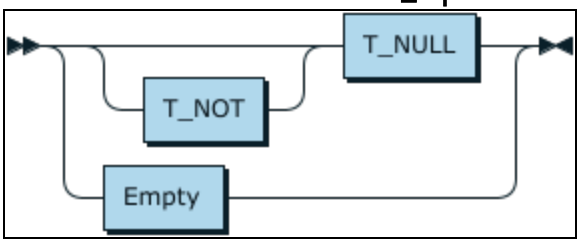
IntervalType

::= T_INTERVAL (T_YEAR IntervalLeadingPrecision_Opt (ToYear_Opt | T_TO T_MONTH) | T_MONTH IntervalLeadingPrecision_Opt ToMonth_Opt | T_DAY IntervalLeadingPrecision_Opt (ToDay_Opt | T_TO (T_HOUR | T_MINUTE | T_SECOND IntervalFractionalSecondsPrecision_Opt)) | T_HOUR IntervalLeadingPrecision_Opt (ToHour_Opt | T_TO (T_MINUTE | T_SECOND IntervalFractionalSecondsPrecision_Opt)) | T_MINUTE IntervalLeadingPrecision_Opt (ToMinute_Opt | T_TO T_SECOND IntervalFractionalSecondsPrecision_Opt) | T_SECOND (IntervalLeadingPrecision_Opt | IntervalSecondsPrecision_Opt) ToSecond_Opt)

referenced by:

- DataType

ColumnConstraintDefinition_Opt:

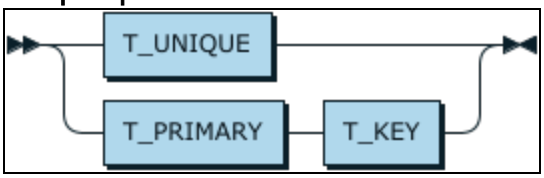



```
ColumnConstraintDefinition_Opt
    ::= T_NOT? T_NULL
     | Empty
```

referenced by:

- ColumnDefinition

UniqueSpecification:




```
UniqueSpecification
    ::= T_UNIQUE
     | T_PRIMARY T_KEY
```

referenced by:

- TableConstraintDefinition

TableConstraintDefinition:


```
TableConstraintDefinition
    ::= UniqueSpecification '(' ColumnNameList ')'
```

referenced by:

- CreateTableStmt

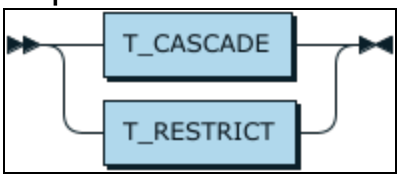
DropTableStmt:


```
DropTableStmt
    ::= T_DROP T_TABLE TableName
```

referenced by:

- ExecStmt

DropBehavior:




```
DropBehavior
    ::= T_CASCADE
     | T_RESTRICT
```

referenced by:

- DropViewStmt
- RevokeStmt

DropViewStmt:


```
DropViewStmt
    ::= T_DROP T_VIEW TableName DropBehavior
```

referenced by:

- ExecStmt

DropIndexStmt:

DropIndexStmt

::= T_DROP T_INDEX Identifier T_ON TableName

referenced by:

- ExecStmt

AlterTableStmt:

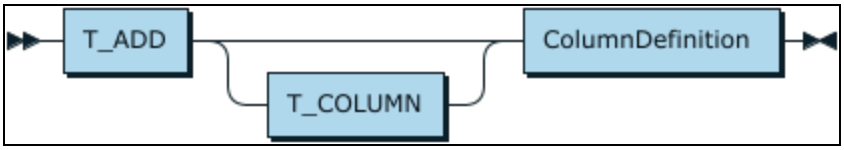
AlterTableStmt

::= T_ALTER T_TABLE TableName AddColumnDefinition

referenced by:

- ExecStmt

AddColumnDefinition:



AddColumnDefinition

::= T_ADD T_COLUMN? ColumnDefinition

referenced by:

- AlterTableStmt

DeleteStmtSearched:

DeleteStmtSearched

::= T_DELETE T_FROM TableName WhereSearchCond_Opt

referenced by:

- ExecStmt

TableName:

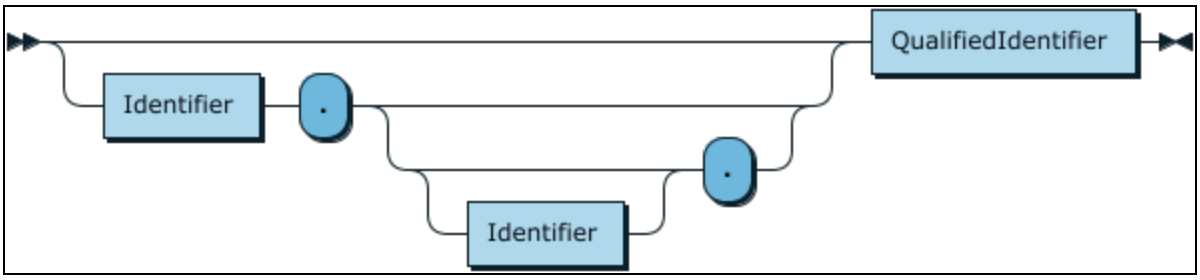
TableName

::= QualifiedName

referenced by:

- AlterTableStmt
- CreateIndexStmt
- CreateTableStmt
- CreateViewStmt
- DeleteStmtSearched
- DropIndexStmt
- DropTableStmt
- DropViewStmt
- InsertStmt
- ObjectName
- TableReference
- UpdateStmtSearched

QualifiedName:



QualifiedName

::= (Identifier '.' (Identifier? '.')?)? QualifiedIdentifier

referenced by:

- TableName

QualifiedIdentifier:

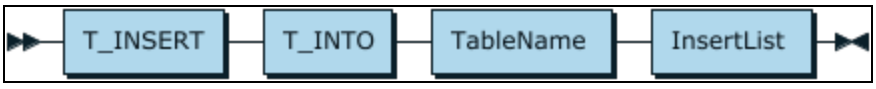
QualifiedIdentifier

::= Identifier

referenced by:

- QualifiedName

InsertStmt:



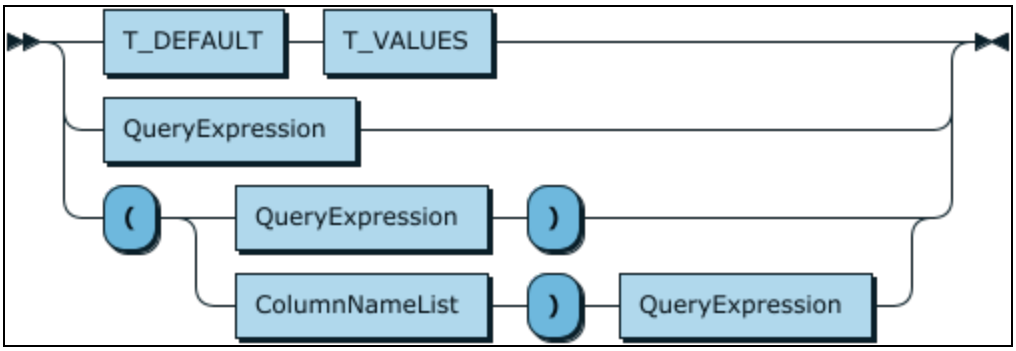
InsertStmt

::= T_INSERT T_INTO TableName InsertList

referenced by:

- ExecStmt

InsertList:

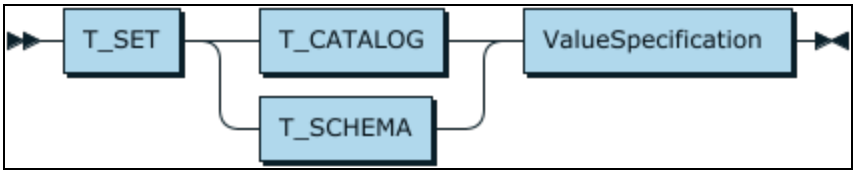



```
InsertList
    ::= T_DEFAULT T_VALUES
     | QueryExpression
     | '(' ( QueryExpression ')' | ColumnNameList ')' QueryExpression )
```

referenced by:

- InsertStmt

SetStmt:




```
SetStmt ::= T_SET ( T_CATALOG | T_SCHEMA ) ValueSpecification
```

referenced by:

- ExecStmt

ProcedureStmt:

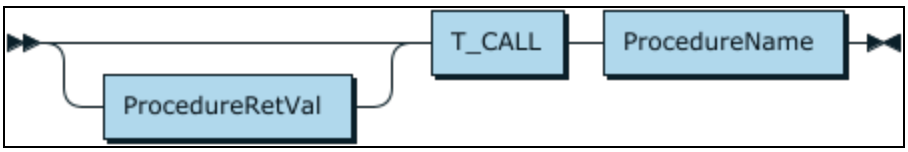

```
ProcedureStmt
```

::= '{' ProcedureCall '}'

referenced by:

- ExecStmt

ProcedureCall:



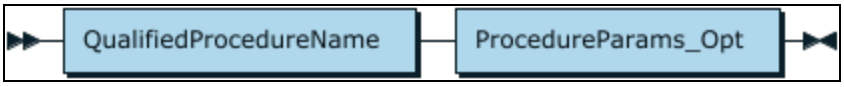
ProcedureCall

::= ProcedureRetVal? T_CALL ProcedureName

referenced by:

- ProcedureStmt

ProcedureName:



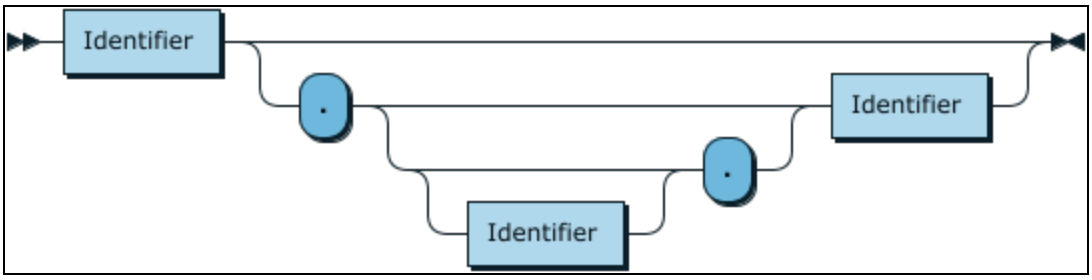
ProcedureName

::= QualifiedProcedureName ProcedureParams_Opt

referenced by:

- ProcedureCall

QualifiedProcedureName:

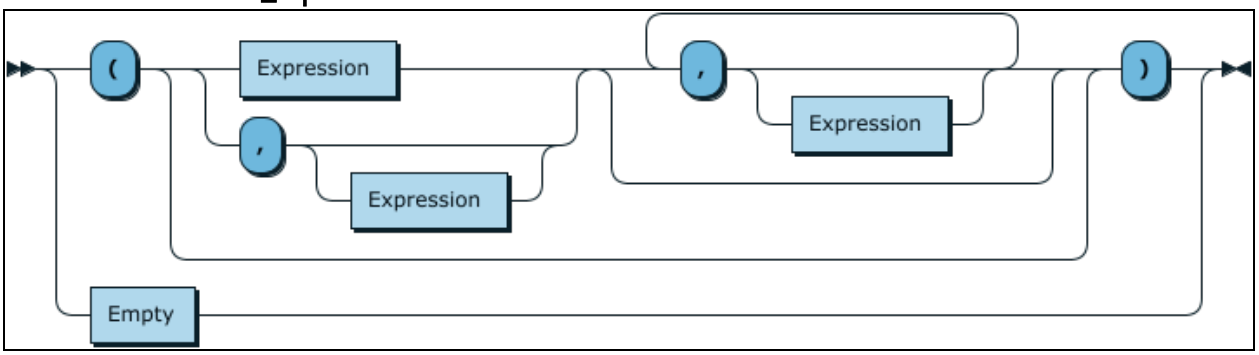



```
QualifiedProcedureName
    ::= Identifier ( '.' ( Identifier? '.' )? Identifier )?
```

referenced by:

- ProcedureName

ProcedureParams_Opt:




```
ProcedureParams_Opt
    ::= '(' ( ( Expression | ',' Expression? ) ( ',' Expression? )* )? ')'
     | Empty
```

referenced by:

- ProcedureName

ProcedureRetVal:

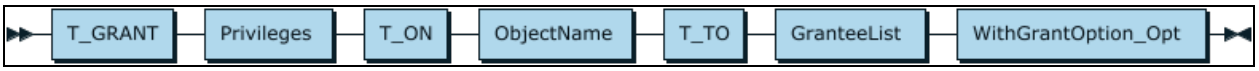
ProcedureRetVal

::= '?' T_EQ

referenced by:

- ProcedureCall

GrantStmt:



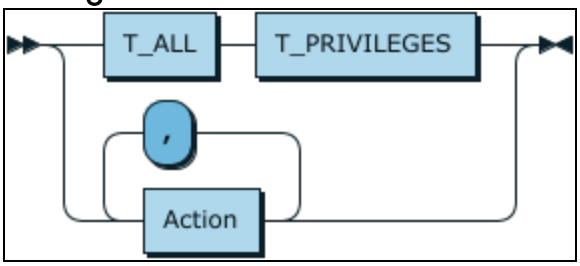
GrantStmt

::= T_GRANT Privileges T_ON ObjectName T_TO GranteeList WithGrantOption_Opt

referenced by:

- ExecStmt

Privileges:



Privileges

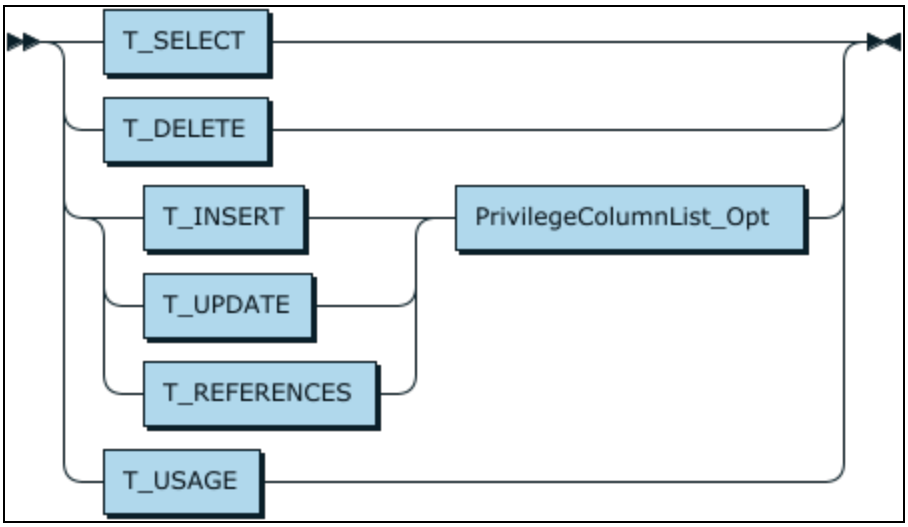
::= T_ALL T_PRIVILEGES

| Action (',' Action)*

referenced by:

- GrantStmt
- RevokeStmt

Action:

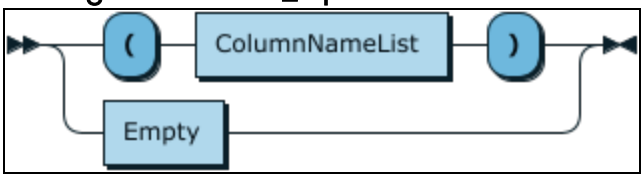



```
Action  ::= T_SELECT
        | T_DELETE
        | ( T_INSERT | T_UPDATE | T_REFERENCES ) PrivilegeColumnList_Opt
        | T_USAGE
```

referenced by:

- Privileges

PrivilegeColumnList_Opt:

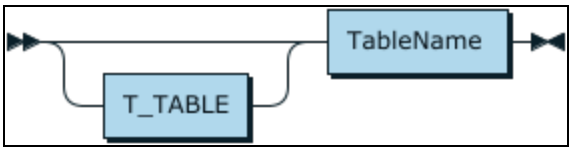



```
PrivilegeColumnList_Opt
      ::= '(' ColumnNameList ')'
       | Empty
```

referenced by:

- Action

ObjectName:

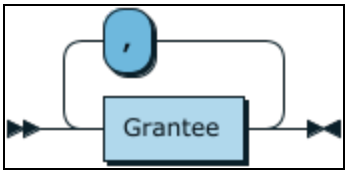



```
ObjectName
    ::= T_TABLE? TableName
```

referenced by:

- GrantStmt
- RevokeStmt

GranteeList:

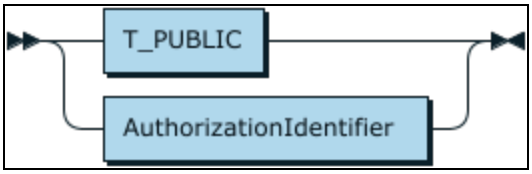



```
GranteeList
    ::= Grantee ( ',' Grantee )*
```

referenced by:

- GrantStmt
- RevokeStmt

Grantee:




```
Grantee  ::= T_PUBLIC
    | AuthorizationIdentifier
```

referenced by:

- GranteeList

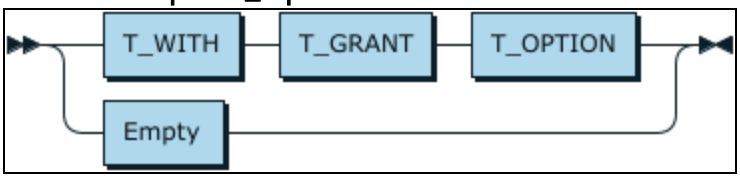
AuthorizationIdentifier:


```
AuthorizationIdentifier
    ::= Identifier
```

referenced by:

- Grantee

WithGrantOption_Opt:

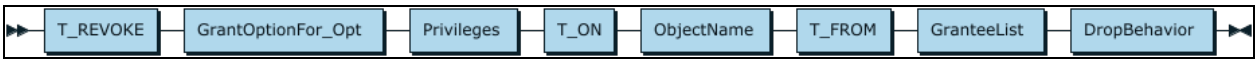



```
WithGrantOption_Opt
    ::= T_WITH T_GRANT T_OPTION
     | Empty
```

referenced by:

- GrantStmt

RevokeStmt:

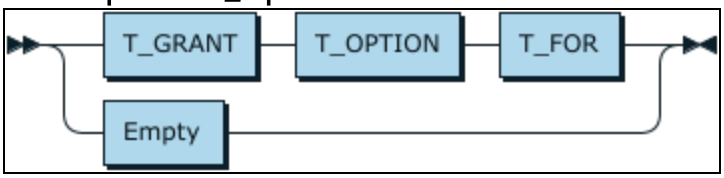



```
RevokeStmt
    ::= T_REVOKE GrantOptionFor_Opt Privileges T_ON ObjectName T_FROM GranteeList DropBehavior
```

referenced by:

- ExecStmt

GrantOptionFor_Opt:

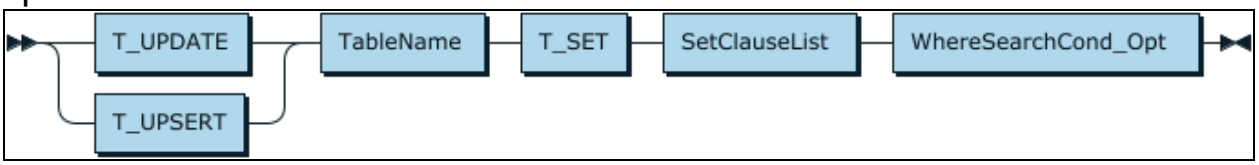



```
GrantOptionFor_Opt
    ::= T_GRANT T_OPTION T_FOR
     | Empty
```

referenced by:

- RevokeStmt

UpdateStmtSearched:

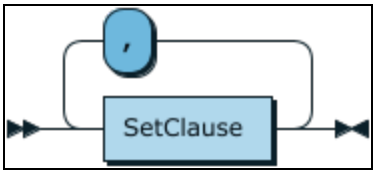



```
UpdateStmtSearched
    ::= ( T_UPDATE | T_UPSERT ) TableName T_SET SetClauseList WhereSearchCond_Opt
```

referenced by:

- ExecStmt

SetClauseList:




```
SetClauseList
```

```
    ::= SetClause ( ',' SetClause )*
```

referenced by:

- UpdateStmtSearched

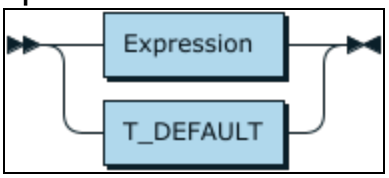
SetClause:


```
SetClause
    ::= ColumnName T_EQ UpdateSource
```

referenced by:

- SetClauseList

UpdateSource:




```
UpdateSource
    ::= Expression
     | T_DEFAULT
```

referenced by:

- SetClause

Appendix B: Token Definitions

This section defines the tokens for the Java SQL BNF Grammar described in Appendix A.

The following token matches the pattern:

T_APPROX_NUM_LIT

```
({DecNumLit}|{USInt})("E"|"e")("+"|"-")?{USInt}
```

T_BIB

```
{sqOpen}({didletter}|{dq})+{sqClose}
```

T_CALL

```
(c|C)(a|A)(l|L)(l|L)
```

T_CHAR_STR_LIT

```
({Chars}|"''"|"\"")*
```

T_DATATYPE

```
"SQL"("_"{letter}+)+
```

T_DECNUM_LIT

```
(({USInt}"."{USInt})|({USInt}".")|("."{USInt}))
```

T_FN

```
(f|F)(n|N)
```

T_HEX_LIT

```
0(x|X)[a-fA-F0-9]*
```

T_IDENTIFIER

```
({letter}|"_")({letter}|{digit}|"_")*
```

If an identifier matches one of the following strings (case-insensitive), it's mapped to the associated token in the following table:

Identifier	Token
"ADD"	T_ADD

Identifier	Token
"AND"	T_AND
"ANY"	T_ANY
"ALL"	T_ALL
"ALTER"	T_ALTER
"AS"	T_AS
"ASC"	T_ASC
"AVG"	T_AVG
"BETWEEN"	T_BETWEEN
"BY"	T_BY
"CASCADE"	T_CASCADE
"CASE"	T_CASE
"CAST"	T_CAST
"CATALOG"	T_CATALOG
"COALESCE"	T_COALESCE
"COLUMN"	T_COLUMN
"CONVERT"	T_CONVERT
"COUNT"	T_COUNT
"CREATE"	T_CREATE

Identifier	Token
"CROSS"	T_CROSS
"CURRENT_DATE"	T_CURRENT_DATE
"CURRENT_TIME"	T_CURRENT_TIME
"CURRENT_TIMESTAMP"	T_CURRENT_TIMESTAMP
"DATE"	T_DATE
"DAY"	T_DAY
"DAYOFWEEK"	T_DAYOFWEEK
"DEFAULT"	T_DEFAULT
"DELETE"	T_DELETE
"DESC"	T_DESC
"DISTINCT"	T_DISTINCT
"DROP"	T_DROP
"ELSE"	T_ELSE
"END"	T_END
"ESCAPE"	T_ESCAPE
"EXCEPT"	T_EXCEPT
"EXISTS"	T_EXISTS
"EXTRACT"	T_EXTRACT

Identifier	Token
"FORMAT"	T_FORMAT
"FROM"	T_FROM
"FULL"	T_FULL
"GRANT"	T_GRANT
"GROUP"	T_GROUP
"HAVING"	T_HAVING
"HOUR"	T_HOUR
"IN"	T_IN
"INDEX"	T_INDEX
"INNER"	T_INNER
"INSERT"	T_INSERT
"INTERVAL"	T_INTERVAL
"INTO"	T_INTO
"IS"	T_IS
"JOIN"	T_JOIN
"KEY"	T_KEY
"LEFT"	T_LEFT
"LIMIT"	T_LIMIT

Identifier	Token
"MAX"	T_MAX
"MIN"	T_MIN
"MINUTE"	T_MINUTE
"MONTH"	T_MONTH
"NULL"	T_NULL
"NULLIF"	T_NULLIF
"ON"	T_ON
"OPTION"	T_OPTION
"OR"	T_OR
"ORDER"	T_ORDER
"OUTER"	T_OUTER
"POSITION"	T_POSITION
"PRIMARY"	T_PRIMARY
"PRIVILEGES"	T_PRIVILEGES
"PUBLIC"	T_PUBLIC
"QUARTER"	T_QUARTER
"REFERENCES"	T_REFERENCES
"RESTRICT"	T_RESTRICT

Identifier	Token
"REVOKE"	T_REVOKE
"RIGHT"	T_RIGHT
"SCHEMA"	T_SCHEMA
"SECOND"	T_SECOND
"SELECT"	T_SELECT
"SET"	T_SET
"SOME"	T_SOME
"STDDEV"	T_STDDEV
"STDDEV_POP"	T_STDDEV_POP
"SUM"	T_SUM
"TABLE"	T_TABLE
"THEN"	T_THEN
"TIME"	T_TIME
"TIMESTAMP"	T_TIMESTAMP
"TIMESTAMPADD"	T_TIMESTAMPADD
"TIMESTAMPDIFF"	T_TIMESTAMPDIFF
"TO"	T_TO
"TOP"	T_TOP

Identifier	Token
"UNION"	T_UNION
"UNIQUE"	T_UNIQUE
"UPSERT"	T_UPSERT
"USAGE"	T_USAGE
"USER"	T_USER
"VALUES"	T_VALUES
"VAR"	T_VAR
"VAR_POP"	T_VAR_POP
"VIEW"	T_VIEW
"WEEK"	T_WEEK
"WHEN"	T_WHEN
"WHERE"	T_WHERE
"WITH"	T_WITH
"YEAR"	T_YEAR
{"||"}	T_CONCAT
"="	T_EQ
">="	T_GE
">"	T_GT

Identifier	Token
"<"	T_LT
"<="	T_LE
"<>" | "!="	T_NE

T_LIKE

```
(l|L)(i|I)(k|K)(e|E)
```

T_NOT

```
(n|N)(o|O)(t|T)
```

T_NOTLIKE

```
{Not}{white}+{Like}
```

T_TSIDATATYPE

```
(S|s)(Q|q)(L|l)"_"(T|t)(S|s)(I|i)"_"({letter}|"_")*
```

T_USINT_LIT

```
{digit}+
```

Reference

1. BNF for SQL-92: http://savage.net.au/SQL/sql-92.bnf.html#subquery
2. Credit to Syntax diagrams for BNF generate tool: http://bottlecaps.de/rr/ui

Contact Us

For more information or help using this product, please contact our Technical Support staff. We welcome your questions, comments, and feature requests.

Note:

To help us assist you, prior to contacting Technical Support please prepare a detailed summary of the Simba Java SQL Engine version and development platform that you are using.

You can contact Technical Support at www.insightsoftware.com.

Third-Party Trademarks

Simba, the Simba logo, SimbaEngine, Simba SDK, and Simba Technologies are registered trademarks of Simba Technologies Inc. in Canada, United States and/or other countries. All other trademarks and/or servicemarks are the property of their respective owners.

All other trademarks are trademarks of their respective owners.