

Appendix 4.A: SQL:2016 Syntax Summary

This appendix summarizes SQL:2016 syntax for the SELECT, INSERT, UPDATE, and DELETE statements presented in this chapter. The syntax is limited to the simplified statement structure presented in this chapter. More complex syntax is introduced in Part 5 of this textbook. The conventions used in the syntax notation are identical to those used at the end of Chapter 3.

Simplified SELECT Syntax

```
<Select-Statement>: { <Simple-Select> | <Set-Select> }  
    [ ORDER BY <Sort-Specification>* ]
```

```
<Simple-Select>:  
    SELECT [ DISTINCT ] <Column-Specification>*  
    FROM <Table-Specification>*  
    [ WHERE <Row-Condition> ]  
    [ GROUP BY ColumnName* ]  
    [ HAVING <Group-Condition> ]
```

```
<Column-Specification>: { <Column-List> | <Column-Item> }
```

```
<Column-List>: { * | TableName.* }  
    -- * is a literal here not a syntax symbol
```

```
<Column-Item>: <Column-Expression> [ AS AliasName ]
```

```
<Column-Expression>:  
    { <Scalar-Expression> | <Aggregate-Expression> }
```

```
<Scalar-Expression>:  
    { <Scalar-Item> |  
      <Scalar-Item> <Arith-Operator> <Scalar-Item> }
```

```
<Scalar-Item>:  
    { [ TableName.]ColumnName |  
      Constant |  
      FunctionName [ (Argument*) ] |  
      <Scalar-Expression> |  
      ( <Scalar-Expression> ) }
```

```
<Arith-Operator>: { + | - | * | / }  
    -- * and + are literals here not syntax symbols
```

```

<Aggregate-Expression>:
  { SUM ( {<Scalar-Expression> | DISTINCT ColumnName } )
  |
  AVG ( {<Scalar-Expression> | DISTINCT ColumnName } )
  |
  MIN ( <Scalar-Expression> ) |
  MAX ( <Scalar-Expression> ) |
  COUNT ( [ DISTINCT ] ColumnName ) |
  COUNT ( * ) } -- * is a literal symbol here, not a special syntax symbol

```

```

<Table-Specification>: { <Simple-Table> |
                        <Join-Operation> }

```

```

<Simple-Table>: TableName [ [ AS ] AliasName ]

```

```

<Join-Operation>:
  { <Simple-Table> [INNER] JOIN <Simple-Table>
    ON <Join-Condition> |
    { <Simple-Table> | <Join-Operation> } [INNER] JOIN
    { <Simple-Table> | <Join-Operation> }
    ON <Join-Condition> |
    ( <Join-Operation> ) }

```

```

<Join-Condition>: { <Simple-Join-Condition> |
                  <Compound-Join-Condition> }

```

```

<Simple-Join-Condition>:
  <Scalar-Expression> <Comparison-Operator>
  <Scalar-Expression>

```

```

<Compound-Join-Condition>:
  { NOT <Join-Condition> |
    <Join-Condition> AND <Join-Condition> |
    <Join-Condition> OR <Join-Condition> |
    ( <Join-Condition> )

```

```

<Comparison-Operator>: { = | < | > | <= | >= | <> }

```

```

<Row-Condition>:
  { <Simple-Condition> | <Compound-Condition> }

```

```

<Simple-Condition>:
  { <Scalar-Expression> <Comparison-Operator>
    <Scalar-Expression> |
    <Scalar-Expression> [ NOT ] IN ( Constant* ) |
    <Scalar-Expression> BETWEEN <Scalar-Expression> AND

```

```

    <Scalar-Expression> |
    <Scalar-Expression> IS [NOT] NULL |
    ColumnName [ NOT ] LIKE StringPattern }

<Compound-Condition>:
    { NOT <Row-Condition> |
      <Row-Condition> AND <Row-Condition> |
      <Row-Condition> OR <Row-Condition> |
      ( <Row-Condition> ) }

<Group-Condition>:
    { <Simple-Group-Condition> | <Compound-Group-Condition>
    }

<Simple-Group-Condition>:-- permits both scalar and aggregate expressions
    { <Column-Expression> ComparisonOperator
      < Column-Experssion> |
      <Column-Expression> [ NOT ] IN ( Constant* ) |
      <Column-Expression> BETWEEN <Column-Expression> AND
      <Column-Expression> |
      <Column-Expression> IS [NOT] NULL |
      ColumnName [ NOT ] LIKE StringPattern }

<Compound-Group-Condition>:
    { NOT <Group-Condition> |
      <Group-Condition> AND <Group-Condition> |
      <Group-Condition> OR <Group-Condition> |
      ( <Group-Condition> ) }

<Sort-Specification>:
    { ColumnName | ColumnNumber } [ { ASC | DESC } ]

<Set-Select>:
    { <Simple-Select> | <Set-Select> } <Set-Operator>
    { <Simple-Select> | <Set-Select> }

<Set-Operator>: { UNION | INTERSECT | EXCEPT } [ ALL ]

```

INSERT Syntax

```

INSERT INTO TableName ( ColumnName* )
    VALUES ( Constant* )

INSERT INTO TableName [ ( ColumnName* ) ]
    <Simple-Select>

```

UPDATE Syntax

```
UPDATE TableName  
  SET <Column-Assignment>*  
  [ WHERE <Row-Condition> ]
```

<Column-Assignment>: ColumnName = <Scalar-Expression>

DELETE Syntax

```
DELETE FROM TableName  
  [ WHERE <Row-Condition> ]
```

Appendix 4.B: Syntax Differences among Major DBMS Products

Table 4B-1 summarizes syntax differences among Microsoft Access (1997 to 2016 versions), Oracle 9i to 12c, Microsoft SQL Server, and IBM's DB2. The differences involve the parts of the SELECT statement presented in the chapter.

Table 4B-1: SELECT Syntax Differences among Major DBMS Products

Element	Product			
	<i>Oracle 9i to 12c</i>	<i>Access 97 to 2016</i>	<i>MS SQL Server</i>	<i>DB2</i>
Pattern-matching characters	%, _	*, ? although the % and _ characters can be used in the 2003 and later versions by setting the query mode	%, _	%, _
Case sensitivity in string matching	Yes	No	Depends on COLLATE clause ¹	Yes
Date constants	Surround in single quotation marks	Surround in # symbols	Surround in single quotation marks	Surround in single quotation marks
Inequality symbol	<>	<>	<>	<>
Join operator style	No in 8i, Yes in 9i and later versions	Yes	Yes	Yes
Difference operations	MINUS keyword	Not supported	EXCEPT keyword	EXCEPT keyword

¹ Case insensitive by default but default can be changed for a database or column using the COLLATE clause.