

## DML Syntax Rules

(for T-DB)

2017.1.23

- This SQL translator can translate any SQL statement that meets the following DML syntax rules.
- The SQL statement should also pass the semantic and permission check in standard T-SQL.
- Using Transact-SQL syntax and arguments convention.  
(<https://msdn.microsoft.com/en-us/library/ms177563.aspx>)

### Syntax

```
<SELECT statement> ::=
{
    SELECT <select_list>
    FROM <table_source>
    [WHERE <search_condition>]
    [<GROUP BY>]
    [HAVING <having_condition>]
    [ORDER BY {[table_name.]column_name [ASC | DESC]} [,...n]]
;}
```

```
<select_list> ::=
    {[table_name.]* | [table_name.]column_name | <method_name> (<expression>)} [,...n]
```

```
<method_name> ::=
    {MIN | MAX | COUNT | SUM | AVG}
```

```
<expression> ::=
    {[table_name.]column_name | positive_integer}
```

```
<table_source> ::=
    {table_name | <joined_table>} [,...n]
```

```
<joined_table> ::=
    {table_name <join_type> table_name ON <search_condition>}
```

```
<join_type> ::=  
  {[INNER | LEFT | RIGHT | FULL] JOIN}
```

```
<search_condition> ::=  
{  
  {<predicate> | (<search_condition>)}  
  [{ AND | OR } {<predicate> | (<search_condition>)}] [,...n]  
}
```

```
<predicate> ::=  
{  
  <expression> <comparison_operator> <expression>  
  | <expression> [NOT] BETWEEN <expression> AND <expression>  
  | <expression> IS [NOT] NULL  
  | <expression> [NOT] IN (<expression> [,...n])  
  | table_name.column_name [NOT] LIKE pattern [ESCAPE escape_character]  
  | <expression> <comparison_operator> (restricted_subquery)  
  | (restricted_subquery) <comparison_operator> <expression>  
  | [NOT] EXISTS (<subquery>)  
}
```

```
<comparison_operator> ::=  
  {<> | != | !=> | <= | !< | >= | = | > | <}
```

```
<subquery> ::=  
{  
  SELECT <select_list>  
  FROM <table_source>  
  [WHERE <search_condition>]  
  [<GROUP BY>]  
  [HAVING <having_condition>]  
}
```

```
<GROUP BY> ::=  
  {GROUP BY [ALL] {table_name.column_name} [,...n]}
```

```
<having_condition> ::=
{
  {<predicate> | <having_predicate> | (<having_condition>)}
  [{ AND | OR } {<predicate> | <having_predicate> | (<having_condition>)}] [,...n]
}
```

```
<having_predicate> ::=
{
  <method_name> (<expression>) IS [NOT] NULL
  | <having_expression> <comparison_operator> <having_expression>
  | <having_expression> [NOT] BETWEEN <having_expression> AND <having_expression>
  | <having_expression> [NOT] IN (<having_expression> [,...n])
  | table_name.column_name [NOT] LIKE pattern [ESCAPE escape_character]
  | {SUM | AVG} (<expression>) <comparison_operator> positive_integer
  | {SUM | AVG} (<expression>) [NOT] BETWEEN positive_integer AND positive_integer
  | {SUM | AVG} (<expression>) [NOT] IN (positive_integer [,...n])
}
```

```
<having_expression> ::=
  {<expression> | {MIN | MAX | COUNT} (<expression>)}
```

```
<INSERT statement> ::=
{
  INSERT [INTO] {table_name} [{column_name [,...n]}]
  {VALUES ({positive_integer | DEFAULT | NULL} [,...n]) [,...n] | <SELECT statement>}
;}
;
```

```
<UPDATE statement> ::=
{
  UPDATE {table_name} SET {column_name = {<expression> | DEFAULT | NULL}} [,...n]
  [FROM <table_source>]
  [WHERE <search_condition>]
;}
;
```

```
<DELETE statement> ::=
{
  DELETE [FROM] {table_name} [WHERE <search_condition>]
;}
;
```

## Arguments

*table\_name, column\_name* or *pattern*

cannot include any SQL keywords or the string "subquery".

*pattern*

can include the following valid wildcard characters: %, \_, [] and [^].

*restricted\_subquery*

can be considered a restricted <subquery> statement, where only a single value is returned by the subquery. SUM and AVG are not allowed in <method\_name>.

## Remarks

Unexpected errors and query results will be returned when these syntax rules are broken.