```
//v.2.6

// Start Symbol
start
  : ( getCollection             // 1 - 5.1        // il numero indica l'ID
  dell'istruzione
    | setIntermediateAs         // 2 - 5.1.1
    | saveAs                    // 3 - 5.1.2
    | spatialJoin               // 4 - 6
    | joinOfCollections         // 5 - 7.1
    | filter                    // 6 - 7.1.1
    | group                     // 7 - 7.1.2
    | expand                    // 8 - 7.1.3
    | mergeCollections          // 9 - 8.1
    | intersectCollections      // 10 - 8.2
    | subtractCollections       // 11 - 8.3
    | useDb                     // 12 - 5.1 nel nuovo paper
    | trajectoryMatching        // 13 - nuova istruzione
    )* EOF
  ;


collectionReference
  :
    ID ( AT ID )? ( AS ID )?
  ;


fieldRef
  :
    ( FIELD_NAME )+
  ;


value
  : INT
  | FLOAT
  | APEX_VALUE
  | QUOTED_VALUE
  | BOOLEAN
  ;



outputFieldSpec
  :
    fieldRef  ( COLON ( value
                      | fieldRef
                      | objectStructure
                      )
              )
  ;


generateAction
  :
    GENERATE
      ( objectStructure ( geometricOption )?
      | geometricOption
      | COUNT LP fieldRef RP
      )
  ;


objectStructure
  :
    LBR
      outputFieldSpec  ( COMMA outputFieldSpec )*
```

```
    RBR
  ;


geometricOption ]
  :
    KEEPING GEOMETRY
  | DROPPING GEOMETRY
  | SETTING GEOMETRY
            ( POINT LP fieldRef COMMA fieldRef RP
            | AGGREGATE LP fieldRef RP
            | fieldRef
            | TO_POLYLINE LP fieldRef RP
            )
  ;


caseClause
  :
    CASE
        ( whereCase )+
        others
  ;


whereCase
  :
    WHERE orCondition
      ( generateAction )?
  ;

others
  :
    KEEP OTHERS
  | DROP OTHERS
  ;


orCondition
  :
    andCondition ( OR andCondition )*
  ;


andCondition
  :
    notCondition ( AND notCondition )*
  ;


notCondition
  :
    ( NOT )? predicate
  ;


predicate
  :
      expression    ( comparator expression  )?
    | withPredicate
    | withoutPredicate
  ;


withPredicate
  :
    WITH ( ID | ARRAY )?
      fieldRef ( COMMA fieldRef )*
```

```
  ;


withoutPredicate
  :
    WITHOUT
      fieldRef ( COMMA fieldRef )*
  ;



expression
  :
    (ADD|SUB)? term ((ADD|SUB) term )*
  ;


term
  :
    factor ( (MUL|DIV) factor )*
  ;


factor
  :
    fieldRef
  | LP orCondition RP
  | INT
  | FLOAT
  | APEX_VALUE
  | QUOTED_VALUE
  | ID (LP (functionParams)? RP)?
  ;


functionParams
  :
    expression ( COMMA expression )*
  ;



// --------------- Basic operators -----------------


getCollection
  :
    GET COLLECTION
      ID ( AT ID )?
    SC
  ;



setIntermediateAs
  :
    SET INTERMEDIATE AS
      ID
    SC
  ;


saveAs
  : SAVE AS
      ID AT ID
    SC
  ;
```

```
spatialJoin
  :
    SPATIAL JOIN OF COLLECTIONS
      collectionReference COMMA collectionReference
      ( ON spatialJoinCondition )?
      SET GEOMETRY ( INTERSECTION | RIGHT | LEFT | ALL )
      ( caseClause )?
    SC
  ;



spatialJoinCondition
  :
    DISTANCE LP ID RP comparator numeric
  | AREA LP ID RP comparator numeric
  | ORIENTATION LP (LEFT|RIGHT) COMMA ID COLON numeric RP
  | INCLUDED LP (LEFT|RIGHT) RP
  | MEET
  | INTERSECT
  ;


comparator
  :
    ( EQ | NEQ | LT | GT | LE | GE )
  ;


numeric
  :
    ( FLOAT | INT )
  ;


joinOfCollections
  :
    JOIN OF COLLECTIONS
      collectionReference COMMA collectionReference
      ( caseClause )?
    SC

  ;


filter
  :
    FILTER
        caseClause
    SC
  ;


group
  : GROUP
      ( groupPartition )+
      others
    SC
  ;


groupPartition
  :
    PARTITION orCondition
      BY fieldRef ( COMMA fieldRef )*
      INTO fieldRef
      ( DROP GROUPING FIELDS )?
```

```
      ( ORDER BY fieldRef ( VERSUS )?  ( COMMA fieldRef ( VERSUS )? )* )?
      ( generateAction )?
  ;


expand
  :
    EXPAND
      ( unpack )+
      others
    SC
  ;


unpack
  :
    UNPACK  orCondition
      ARRAY fieldRef
      TO ID
      ( generateAction )?
  ;


mergeCollections
  :
    ( ALL )?
     MERGE COLLECTIONS
      collectionReference ( COMMA collectionReference )+
    SC
  ;


intersectCollections
  :
    INTERSECT COLLECTIONS
      collectionReference COMMA collectionReference
    SC
  ;


subtractCollections
  :
    SUBTRACT COLLECTIONS
      collectionReference COMMA collectionReference
    SC
  ;


useDb
  :
    USE
      DB (ID | APEX_VALUE) ( AS (ID | APEX_VALUE) )?
        ( COMMA DB (ID | APEX_VALUE) ( AS (ID | APEX_VALUE) )? )*
      ON
        ( DEFAULT SERVER
        | SERVER (ID | APEX_VALUE) ( (ID | APEX_VALUE) )?
        )
    SC
  ;


trajectoryMatching
  :
    TRAJECTORY MATCHING
      collectionReference COMMA collectionReference
      ( trajectoryPartition )+
      others
    SC
```

```
  ;


trajectoryPartition
  :
    PARTITION orCondition
      ( partitionMatching )+
  ;

partitionMatching
  :
    MATCHING fieldRef
    WRT fieldRef
    THRESHOLD LP ID  RP numeric
    ( WHERE orCondition )?
    INTO fieldRef
    ( ADDING fieldRef TO INPUT )
    ( MIN SIMILARITY numeric )?
  ;




// ************************************
// ***
// ***            SCANNER
// ***
// ************************************


fragment LETTER : 'A'..'Z'|'a'..'z';
fragment DIGIT0 : '1'..'9';
fragment DIGIT  : '0'..'9';
fragment WS     :   ( ' ' | '\t' | '\r' | '\n' )* ;


// boolean Operator
AND : 'AND';
OR  : 'OR';
NOT : 'NOT';

// keywords
ADDING        : 'ADDING';
AGGREGATE     : 'AGGREGATE';
ALL           : 'ALL';
AREA          : 'AREA';
ARRAY         : 'ARRAY';
AS            : 'AS';
BOOLEAN       : 'TRUE' | 'FALSE';
BY            : 'BY';
CASE          : 'CASE';
COLLECTION    : 'COLLECTION';
COLLECTIONS   : 'COLLECTIONS';
COUNT         : 'COUNT';
DB            : 'DB';
DEFAULT       : 'DEFAULT';
DISTANCE      : 'DISTANCE';
DROP          : 'DROP';
DROPPING      : 'DROPPING';
EXPAND        : 'EXPAND';
FIELDS        : 'FIELDS';
FILTER        : 'FILTER';
GENERATE      :  'GENERATE';
GEOMETRY      : 'GEOMETRY';
GET           : 'GET';
GROUP         : 'GROUP';
GROUPING      : 'GROUPING';
INCLUDED      : 'INCLUDED';
```

```
INPUT         : 'INPUT';
INTERMEDIATE  : 'INTERMEDIATE';
INTERSECT     : 'INTERSECT';
INTERSECTION  : 'INTERSECTION';
INTO          : 'INTO';
JOIN          : 'JOIN';
KEEP          : 'KEEP';
KEEPING       : 'KEEPING';
LEFT          : 'LEFT';
MATCHING      : 'MATCHING';
MEET          : 'MEET';
MERGE         : 'MERGE';
MIN           : 'MIN';
OF            : 'OF';
ON            : 'ON';
ORIENTATION   : 'ORIENTATION';
OTHERS        : 'OTHERS';
ORDER         : 'ORDER' | 'SORTED';
PARTITION     : 'PARTITION';
POINT         : 'POINT';
RIGHT         : 'RIGHT';
SAVE          : 'SAVE';
SERVER        : 'SERVER';
SET           : 'SET';
SETTING       : 'SETTING';
SIMILARITY    : 'SIMILARITY';
SPATIAL       : 'SPATIAL';
SUBTRACT      : 'SUBTRACT';
TO            : 'TO';
TO_POLYLINE   : 'TO_POLYLINE';
TRAJECTORY    : 'TRAJECTORY';
THRESHOLD     : 'THRESHOLD';
UNPACK        : 'UNPACK';
VERSUS        : 'DESC' | 'ASC';
USE           : 'USE';
WHERE         : 'WHERE';
WITH          : 'WITH';
WITHOUT       : 'WITHOUT';
WRT           : 'WRT';


INT: '0' | DIGIT0 DIGIT* ;

FLOAT: DIGIT0 DIGIT* DOT DIGIT+ | '0' DOT DIGIT+;

ID: LETTER (LETTER | DIGIT | '_')*;

ID2: (LETTER | DIGIT | '_')+;

FIELD_NAME: ( DOT (LETTER | DIGIT | '_')+ )
            | DOT '"' (~('"') )* '"'
            | DOT '~geometry'
            | '~geometry';


// puntuaction
AT    : '@';
EQ    : '=';
NEQ   : '!=';
LE    : '<=';
GE    : '>=';
LT    : '<';
GT    : '>';
DOT   : '.';
ADD   : '+';
SUB   : '-';
MUL   : '*';
DIV   : '\\';
```

```
COMMA : ',';
COLON : ':';
SC    : ';';
LP    : '(';
RP    : ')';
LB    : '[';
RB    : ']';
LBR   : '{';
RBR   : '}';
APEX  : '\'';
QUOTE : '"';
SLASH : '/';
TILDE : '~';
XXX   : '###TEST***';


WHITE_SPACES  : WS ;
APEX_VALUE    :  '\'' (~('\'') )* '\''  ;
QUOTED_VALUE  :  '"' (~('"') )* '"'  ;
SCAN_ERROR    : .  ;
```