



SQL to SparkQL

Continue the Development of SQL to SPARQL translation on Jyphon

University of Texas at Austin
CS347 - Database Management
Dr. Philip Cannata (Oracle Engineer)
Nelma Perera Rodriguez, Semur Nabiev, Marek Bejda

Building Instructions

1. ant clean
2. move dist to the trash
3. ant
4. ant
5. sudo dist/bin/jython

```
>> conn = connectTo 'jdbc:oracle:thin:@129.152.144.84  
/PDB1.usuniversi01134.oraclecloud.internal'  
'cs370_ontologies' 'orcl' 'rdf_mode' 'emp';  
>> SQL on conn 'select * from emp'
```

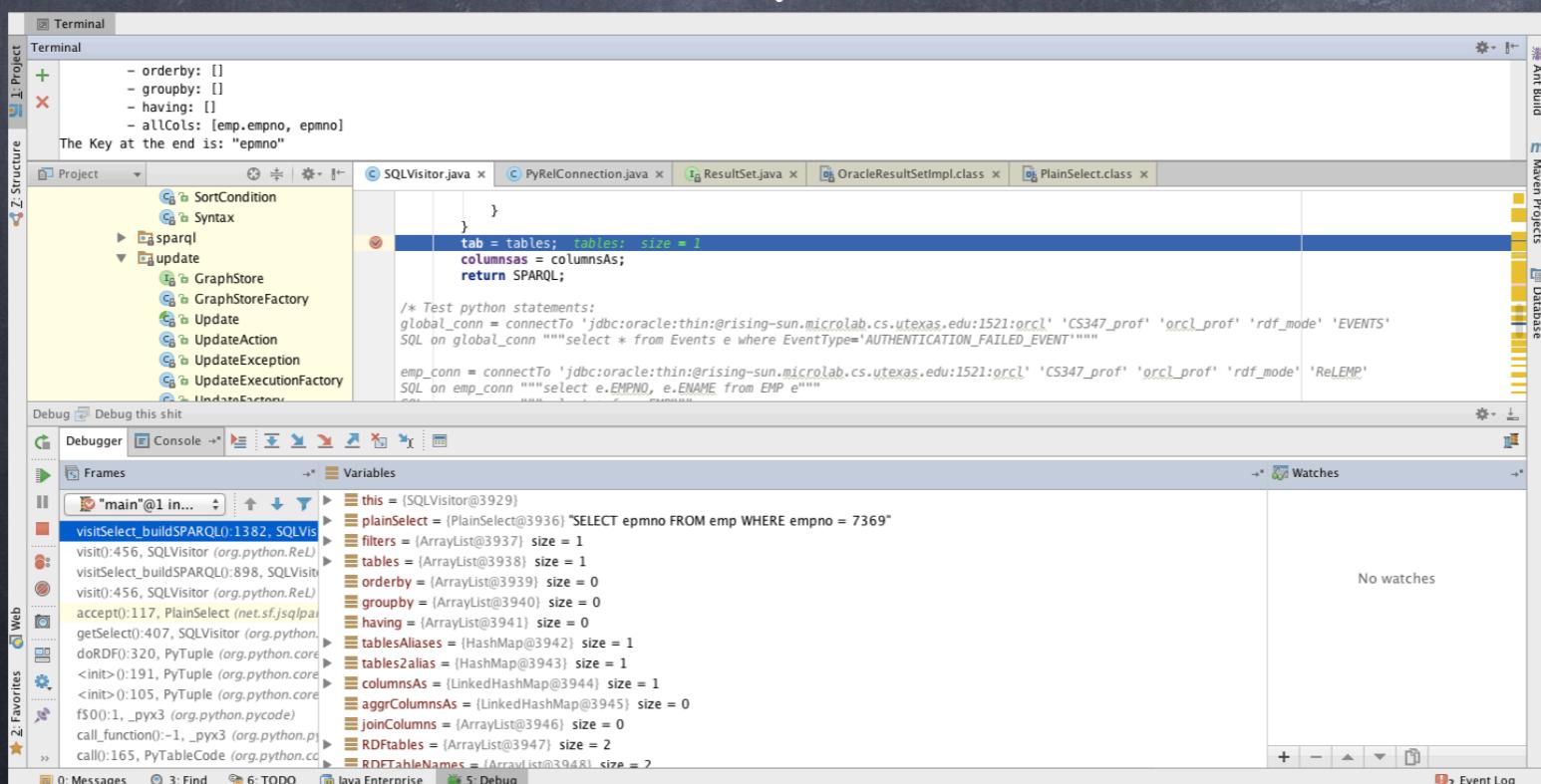
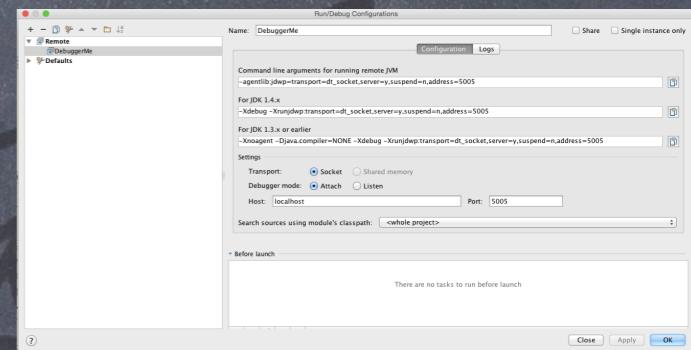


Debugging Instructions



Debugging Instructions

1. `export JAVA_OPTS="-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=5005"`
2. IntelliJ IDEA (IDE) -> Remote Configuration Profile
3. ant
4. dist/bin/jython
5. `>> conn = connectTo 'jdbc:oracle:thin:@129.152.144.84 /PDB1.usuniversi01134.oraclecloud.internal' 'cs370_ontologies' 'orcl' 'rdf_mode' 'emp';`
6. Set breakpoint in IDE
7. `>> SQL on conn 'select * from emp'`



Debugging Instructions

The screenshot shows an IDE interface with the following components:

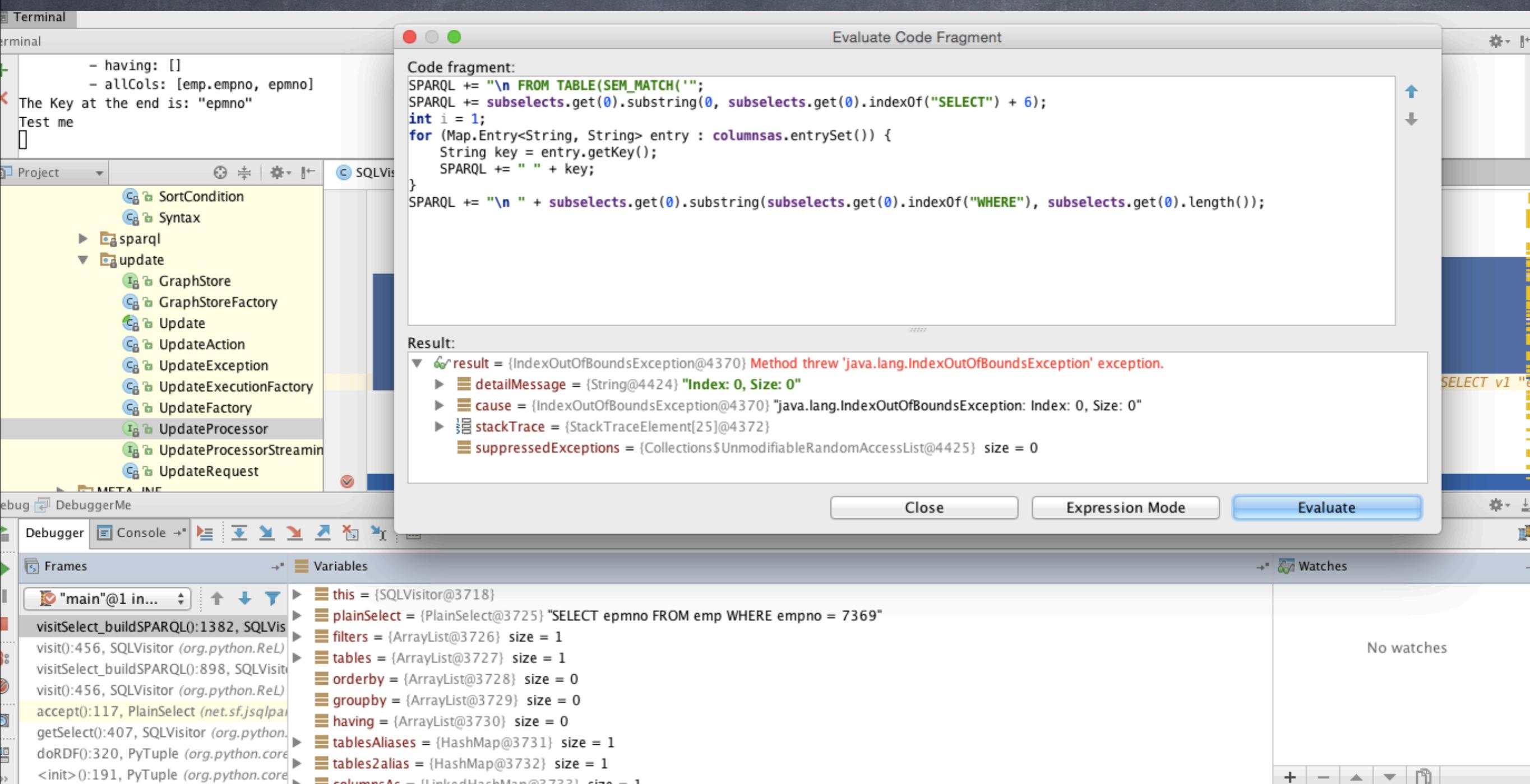
- Terminal:** Shows the output of a command-line session:

```
- orderby: []
- groupby: []
- having: []
- allCols: [emp.empno, empno]
The Key at the end is: "empno"
```
- Project:** Shows a tree view of the project structure under the `update` package:
 - SortCondition
 - Syntax
 - sparql
 - update
 - GraphStore
 - GraphStoreFactory
 - Update
 - UpdateAction
 - UpdateException
 - UpdateExecutionFactory
 - UpdateFactory
- Code Editor:** Displays the `SQLVisitor.java` file with the cursor on the line:

```
tab = tables; tables: size = 1
```
- Debugger:** Shows the debugger interface with the following details:
 - Frames:** Stack trace:

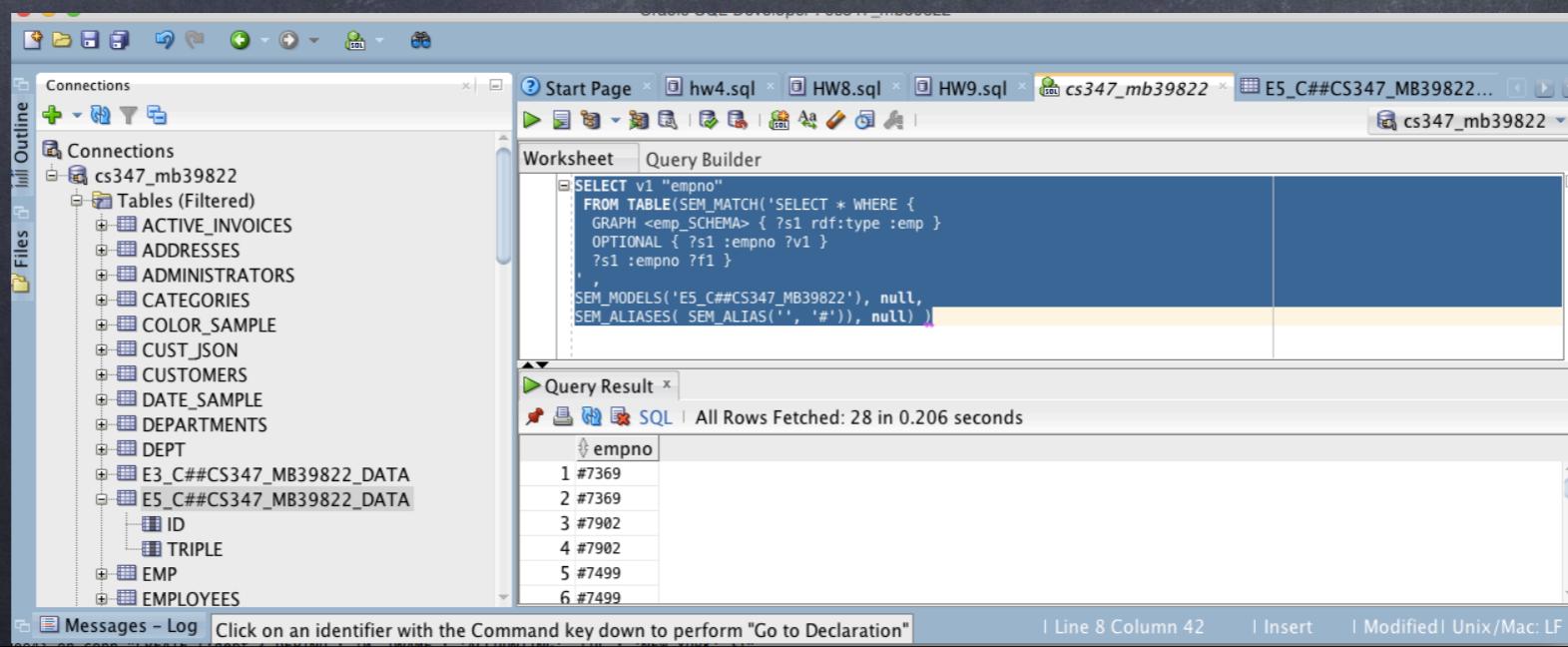
```
"main"@1 in ...
```
 - Variables:** A list of variables and their values:
 - this = {SQLVisitor@3929}
 - plainSelect = {PlainSelect@3936} "SELECT empno FROM emp WHERE empno = 7369"
 - filters = {ArrayList@3937} size = 1
 - tables = {ArrayList@3938} size = 1
 - orderby = {ArrayList@3939} size = 0
 - groupby = {ArrayList@3940} size = 0
 - having = {ArrayList@3941} size = 0
 - tablesAliases = {HashMap@3942} size = 1
 - tables2alias = {HashMap@3943} size = 1
 - columnsAs = {LinkedHashMap@3944} size = 1
 - aggrColumnsAs = {LinkedHashMap@3945} size = 0
 - joinColumns = {ArrayList@3946} size = 0
 - RDFtables = {ArrayList@3947} size = 2
 - RDFTableNames = {ArrayList@3948} size = 2
 - Watches:** No watches.
- Bottom Bar:** Includes tabs for Messages, Find, TODO, Java Enterprise, Debug, and Event Log.

Debugging Instructions



SparQL Executing Instructions

1. Login to a CS machine
2. wget https://www.dropbox.com/s/c0gshqi3s1fhx9n/neo4j_emp_db.txt?dl=1 -O emp_dept.neo4j.formatted.txt
3. Change the login credentials
conn = connectTo 'jdbc:oracle:thin:@sayonara.microlab.cs.utexas.edu:1521:orcl' 'C##cs347_UTEID' 'orcl_UTEID' 'rdf_mode' 'E5';
4. /u/cannata/ReL/dist/bin/jython emp_dept.neo4j.formatted.txt
5. Now open SQL Developer and you should see a table named "E5_C##cs347_UTEID"
6. Now you can run the SparQL queries.



How it works

SQL Query

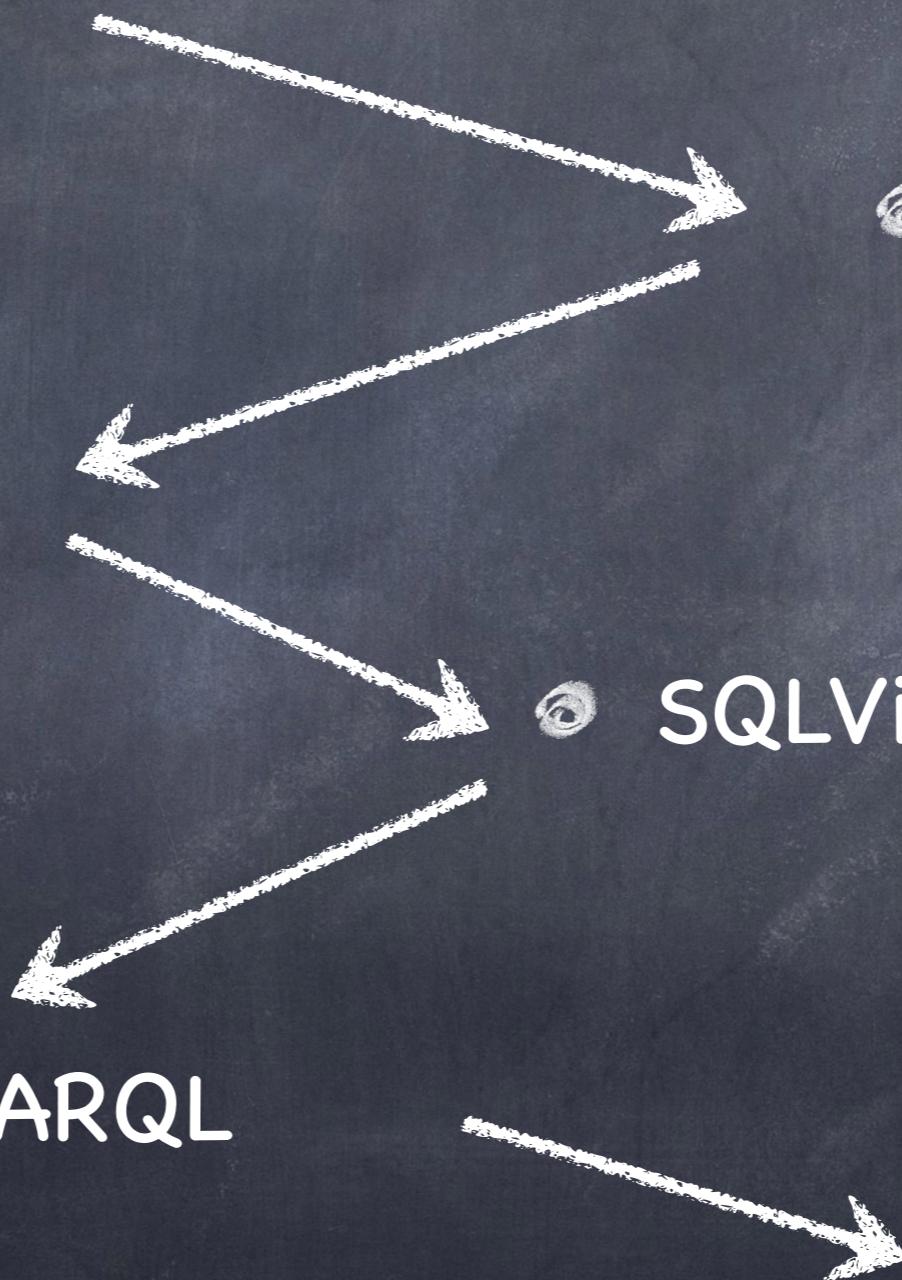
IF SELECT

visitSelect_buildSPARQL

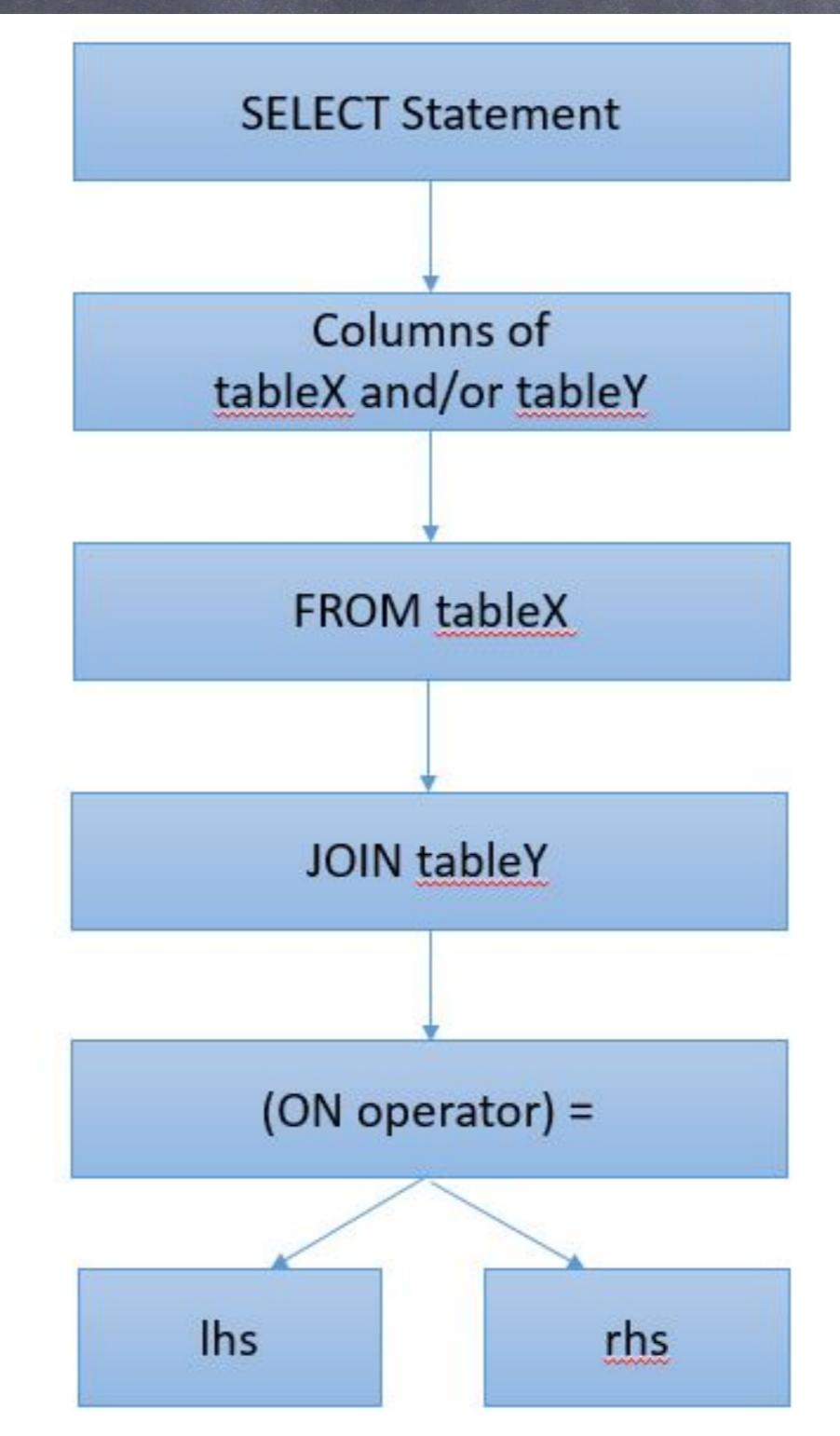
SQL Server

SQLVisitor.getSelect

SPARQL



old design



Example without subqueries

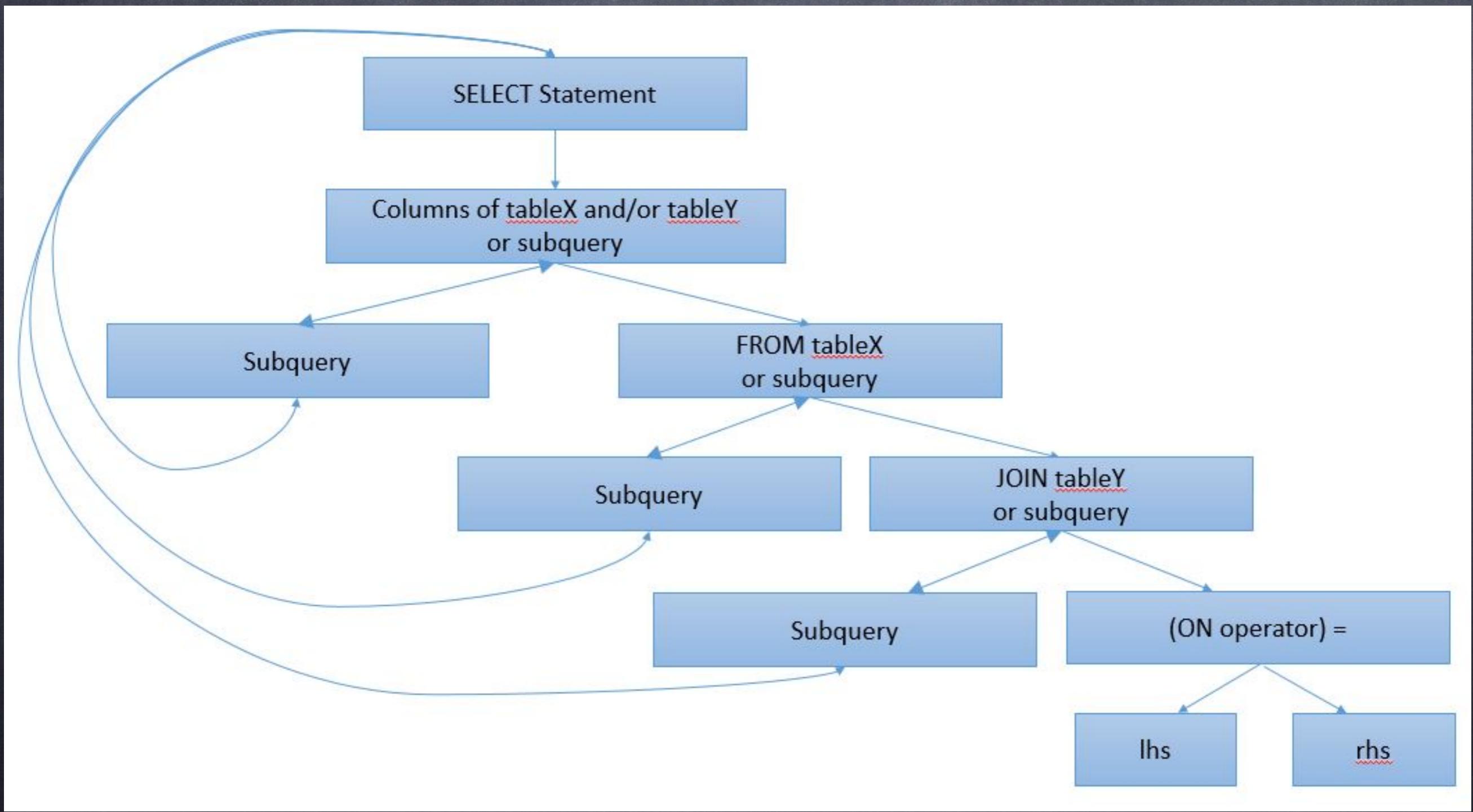
SQL:

```
SELECT * FROM emp;
```

SPARQL:

```
SELECT v1 "empno", v2 "deptno", v3 "comm", v4 "job", v5 "mgr", v6 "ename",
v7 "hiredate", v8 "sal", v9 "dept"
FROM TABLE(SEM_MATCH('SELECT * WHERE {
    GRAPH { ?s1 rdf:type :emp }
    OPTIONAL { ?s1 :empno ?v1 }
    OPTIONAL { ?s1 :deptno ?v2 }
    OPTIONAL { ?s1 :comm ?v3 }
    OPTIONAL { ?s1 :job ?v4 }
    OPTIONAL { ?s1 :mgr ?v5 }
    OPTIONAL { ?s1 :ename ?v6 }
    OPTIONAL { ?s1 :hiredate ?v7 }
    OPTIONAL { ?s1 :sal ?v8 }
    OPTIONAL { ?s1 :dept ?v9 } })
```

New design



Code Changes

- Create the List data structure subselects to store a list af all subqueries statements in the query, with size = to subDepth, and it will be filled depending on how many trues are in sq stack
- boolean subSelect will tell us if we processing a query of subquery inside visit(PlainSelect plainSelect) and visitSelect_buildSPARQL
- subDepth will tell us how many subqueries there are in the query
- Pass a list as an arg to visitSelect_buildSPARQL method
- Then in visitSelect_buildSPARQL we get the subquery from the SQL FROM OR WHERE statement then we modified the SPARQL FILTER statement to form the translations.

Subquery in WHERE

SQL:

```
SELECT empno FROM emp  
WHERE empno=(SELECT empno FROM emp WHERE empno=7369);
```

SPARQL:

```
SELECT v1 "empno"  
FROM TABLE(SEM_MATCH('SELECT * WHERE {  
    GRAPH <emp_SCHEMA> { ?s1 rdf:type :emp }  
    OPTIONAL { ?s1 :empno ?v1 }  
    {  
        SELECT * WHERE {  
            GRAPH <emp_SCHEMA> { ?s1 rdf:type :emp }  
            OPTIONAL { ?s1 :empno ?v1 }  
            ?s1 :empno ?f1 .  
            FILTER(?f1 = :7369)}  
    }'))
```

RESULTS: (('empno',), (7369,), (7369,))

Subquery with *

SQL:

```
SELECT * FROM emp  
WHERE empno=(SELECT epmno FROM emp WHERE empno=7369);
```

SPARQL:

```
SELECT v1 "empno" v2 "" v3...  
FROM TABLE(SEM_MATCH('SELECT v1 WHERE {  
    GRAPH { ?s1 rdf:type :emp }  
    OPTIONAL { ?s1 :empno ?v1 }  
    FILTER( ?v1 = :(SELECT empno FROM emp WHERE epmno=7369) )  
}'))
```

RESULTS:

Subquery in FROM

SQL:

```
SELECT empno, sal FROM (SELECT empno, dept, sal FROM emp);
```

SPARQL:

```
SELECT v1 "empno", v3 "sal"
FROM TABLE(SEM_MATCH('SELECT ?v1 ?v2 ?v3
WHERE {
    GRAPH <emp_SCHEMA> { ?s1 rdf:type :emp }
    OPTIONAL { ?s1 :empno ?v1 }
    OPTIONAL { ?s1 :dept ?v2 }
    OPTIONAL { ?s1 :sal ?v3 }
}'))
```

RESULTS: (('empno', 'sal'), (7369, 800), (7369, 800), (7369, 800),

Additional Resources

SQL Nested Queries in SPARQL <http://ceur-ws.org/Vol-619/paper5.pdf>

SubQuery Standard <http://www.w3.org/TR/sparql11-query/#subqueries>

SparQL By Example <http://www.cambridgesemantics.com/semantic-university/sparql-by-example>

New

