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

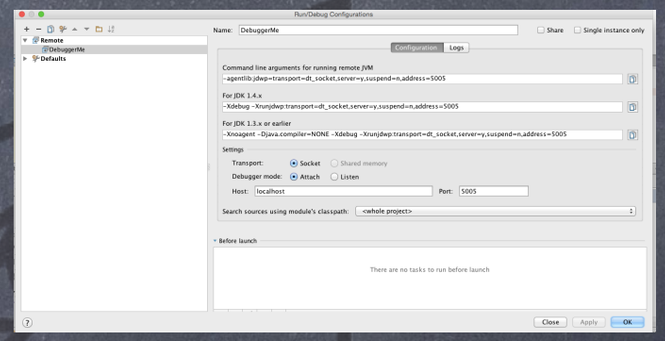
1. ant clean
2. move dist to the trash
3. ant
4. ant
5. sudo dist/bin/jython

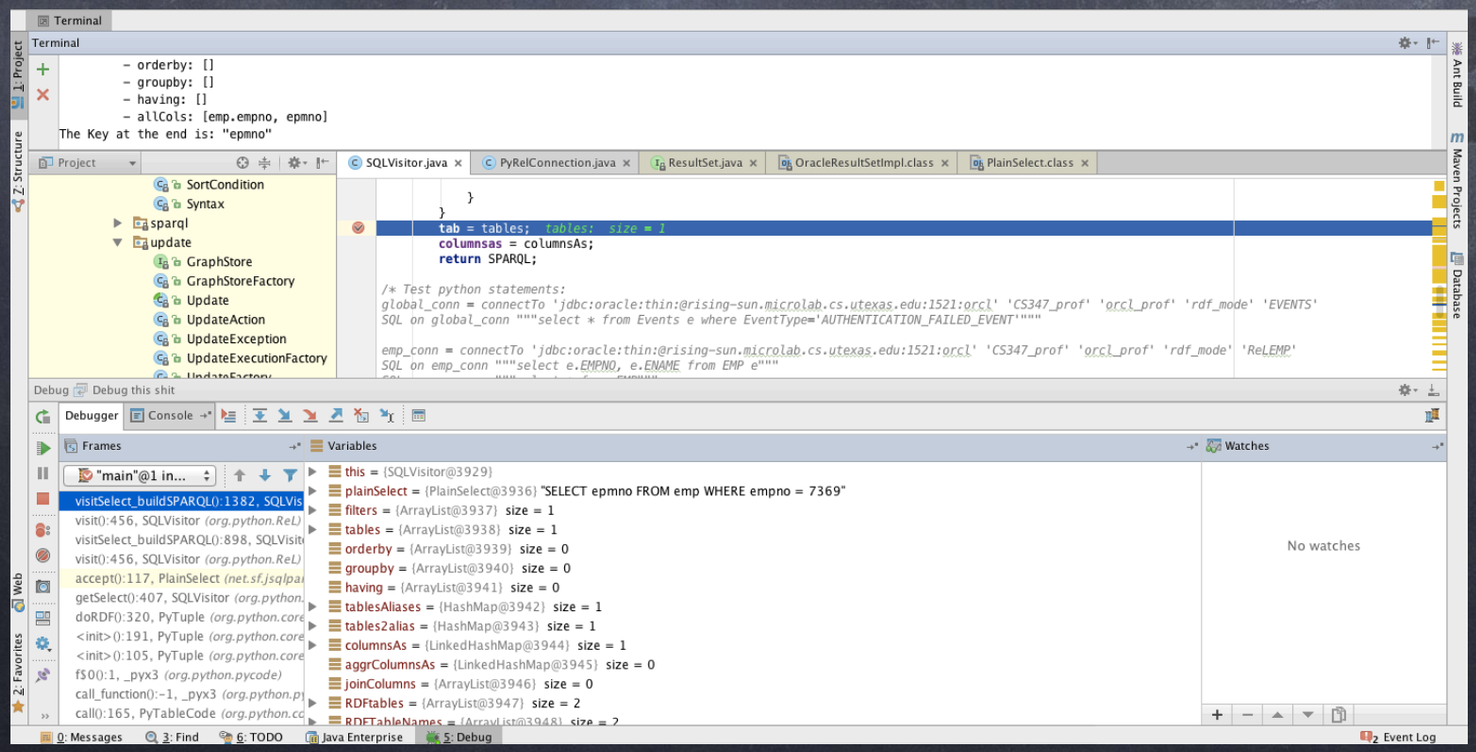
```
>> conn = connectTo 'jdbc:oracle:thin:@129.152.144.84
          /PDB1.usuniversi01134.oraclecloud.internal'
          'cs370_ontologies' 'orcl' 'rdf_mode' 'emp';
>> SQL on conn 'select * from emp'
```

Debugging Instructions

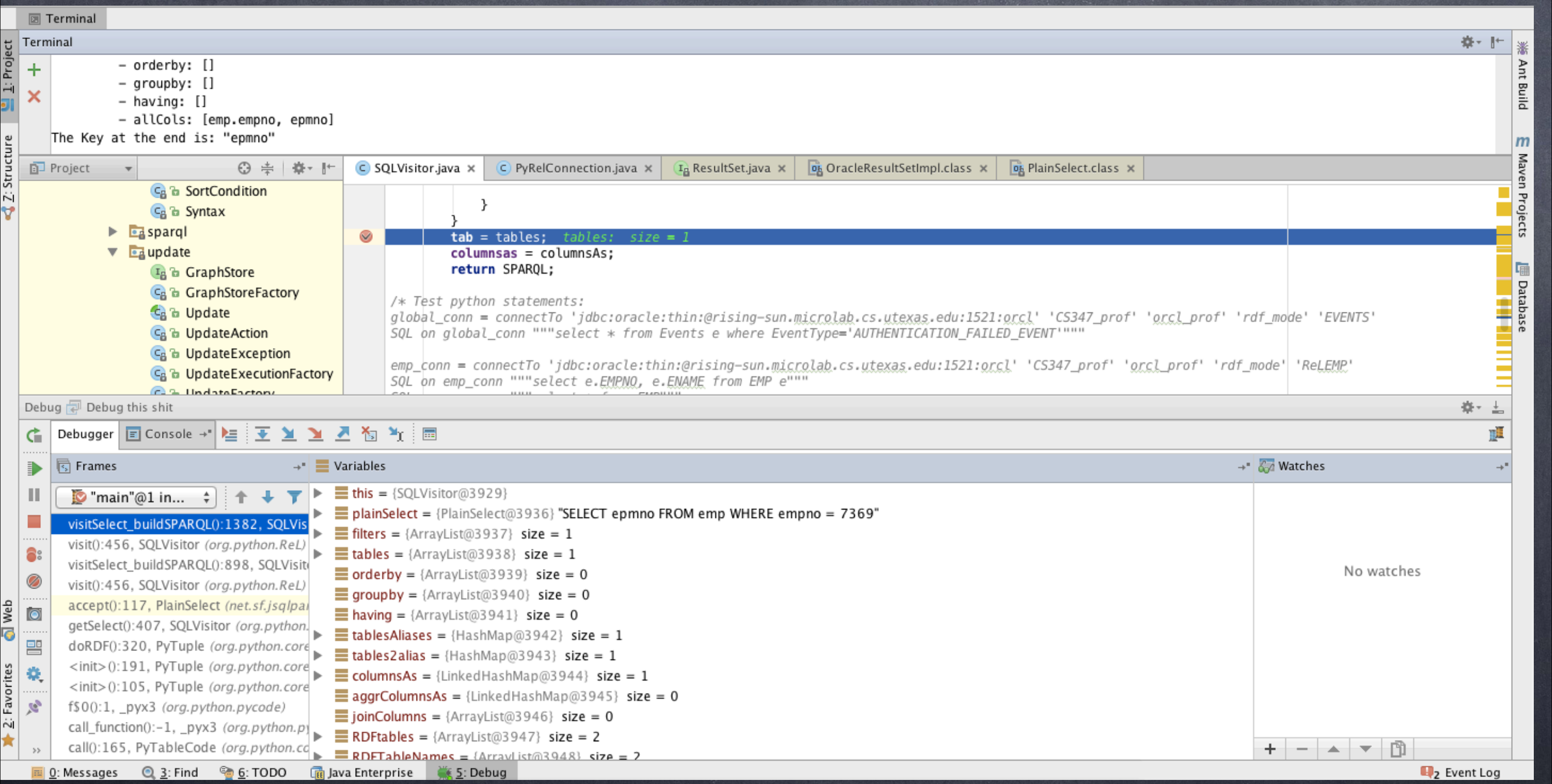
Debugging Instructions

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:@129.152.144.84:1521/PDB1.usuniversi01134.oraclecloud.internal' 'cs370_ontologies' 'orcl' 'rdf_mode' 'emp';
6. Set breakpoint in IDE
7. >> SQL on conn 'select * from emp'

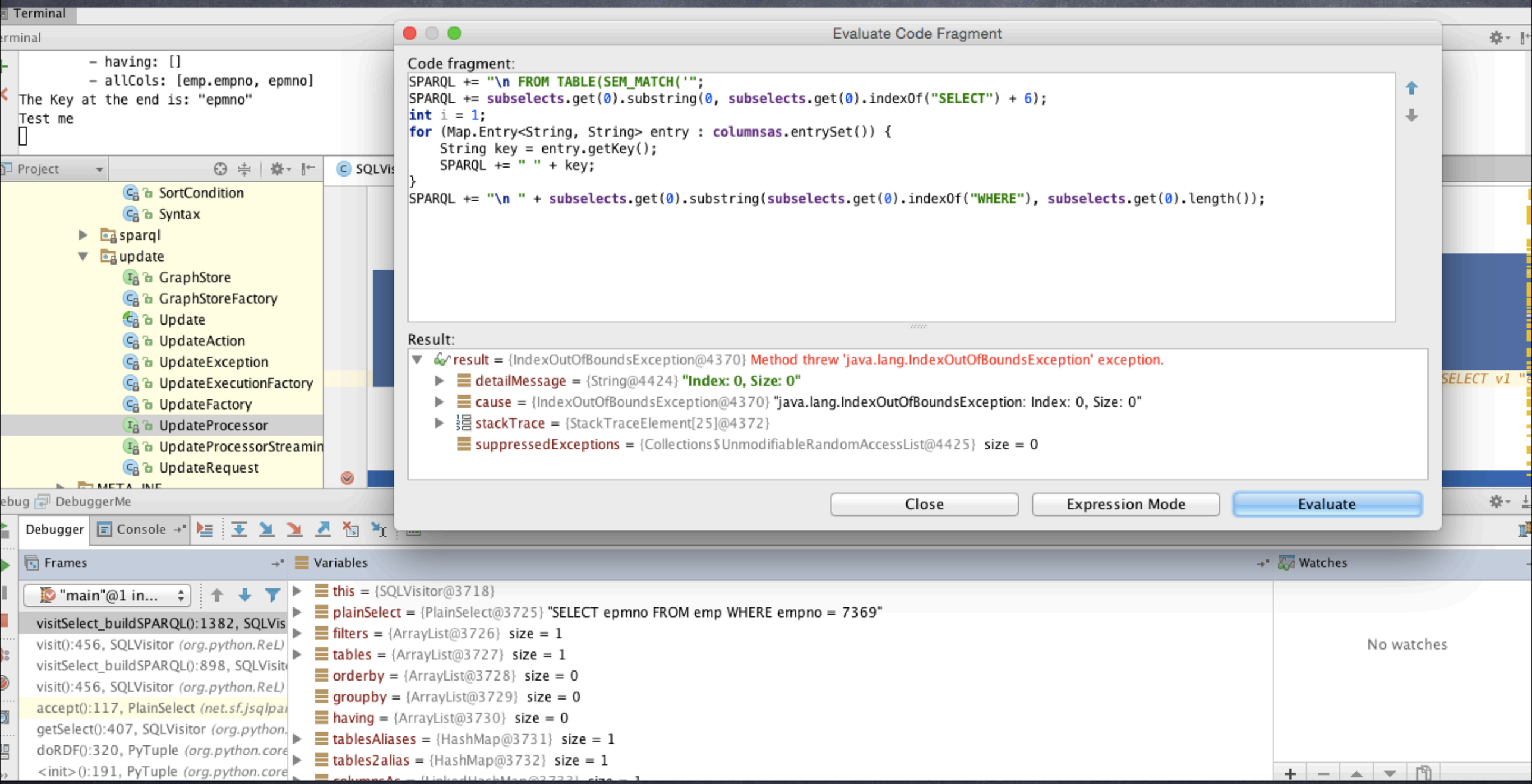




Debugging Instructions



Debugging Instructions



SparQL Executing Instructions

Oracle SQL Developer : cs347_mb39822

Connections

- Connections
  - cs347_mb39822
    - Tables (Filtered)
      - ACTIVE_INVOICES
      - ADDRESSES
      - ADMINISTRATORS
      - CATEGORIES
      - COLOR_SAMPLE
      - CUST_JSON
      - CUSTOMERS
      - DATE_SAMPLE
      - DEPARTMENTS
      - DEPT
      - E3_C##CS347_MB39822_DATA
      - E5_C##CS347_MB39822_DATA
      - EMP
      - EMPLOYEES
      - FLOAT_SAMPLE
      - GENERAL_LEDGER_ACCOUNTS
      - INVOICE_ARCHIVE
      - INVOICE_LINE_ITEMS
      - INVOICES
      - ITEMS
      - NULL_SAMPLE
      - ORDER_DETAILS
      - ORDER_ITEMS
      - ORDERS
      - PAID_INVOICES
      - PRODUCTS

Start Page | hw4.sql | HW8.sql | cs347_mb39822

cs347_mb39822

Worksheet | Query Builder

```
SELECT v1 "ename", v2 "dname"
 FROM TABLE(SEM_MATCH('
SELECT * WHERE {
    GRAPH <emp_SCHEMA> { ?s1 rdf:type :emp }
    GRAPH <dept_SCHEMA> { ?s2 rdf:type :dept }
    OPTIONAL { ?s1 :ename ?v1 }
    OPTIONAL { ?s2 :dname ?v2 }
    ?s1 :deptno ?j1 .
    ?s2 :deptno ?j1 .
}
' ,
SEM_MODELS('EMP_CS370_ONTOLOGIES'), null,
SEM_ALIASES( SEM_ALIAS('', '#')), null) );
```

Query Result

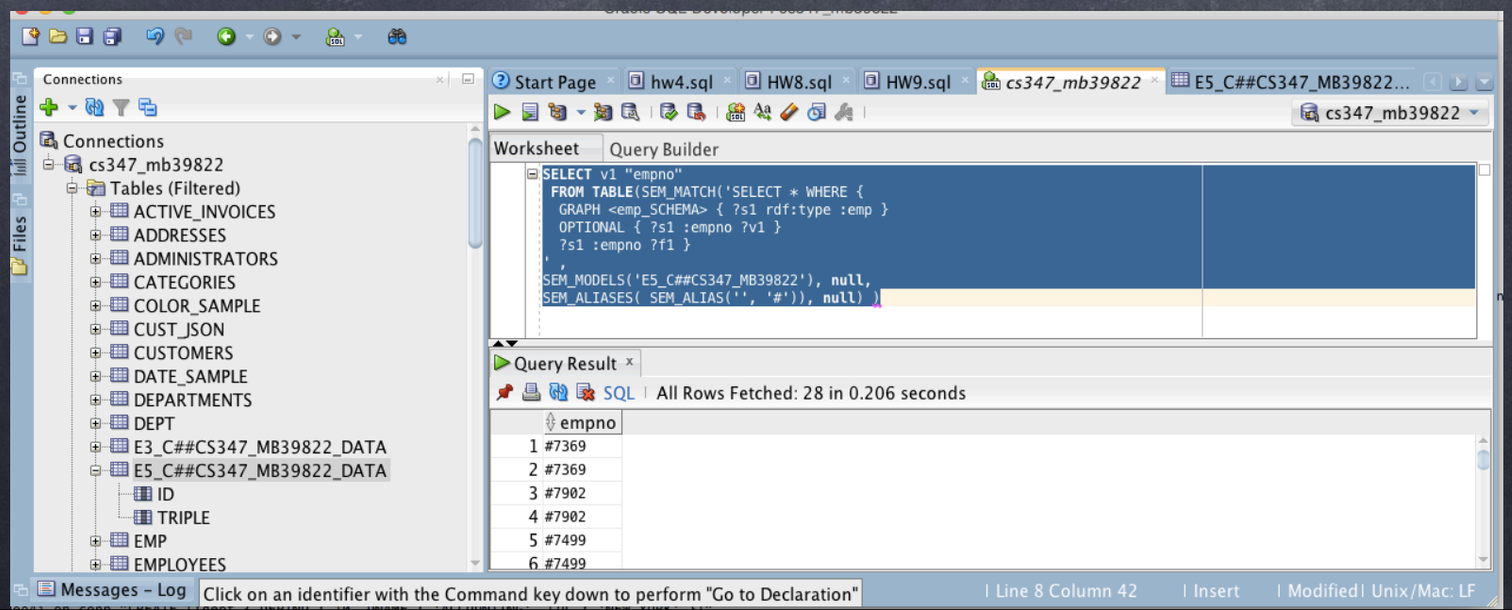
SQL | Executing:SELECT v1 "ename", v2 "dname" FROM TABLE(SEM_MATCH(' SELECT * WHERE { GRAPH <emp_S...

```
ORA-55300: model "EMP_CS370_ONTOLOGIES" does not exist
ORA-06512: at "MDSYS.MD", line 1723
ORA-06512: at "MDSYS.MDERR", line 17
ORA-06512: at "MDSYS.RDF_APIS_INTERNAL", line 1832
ORA-06512: at "MDSYS.RDF_MATCH_IMPL_T", line 381
ORA-06512: at line 1
55300. 0000 -  "model %s does not exist"
*Cause:    The specified model could not be found.
*Action:   Make sure that the model has been created.
```

Messages - Log

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.



Instructions

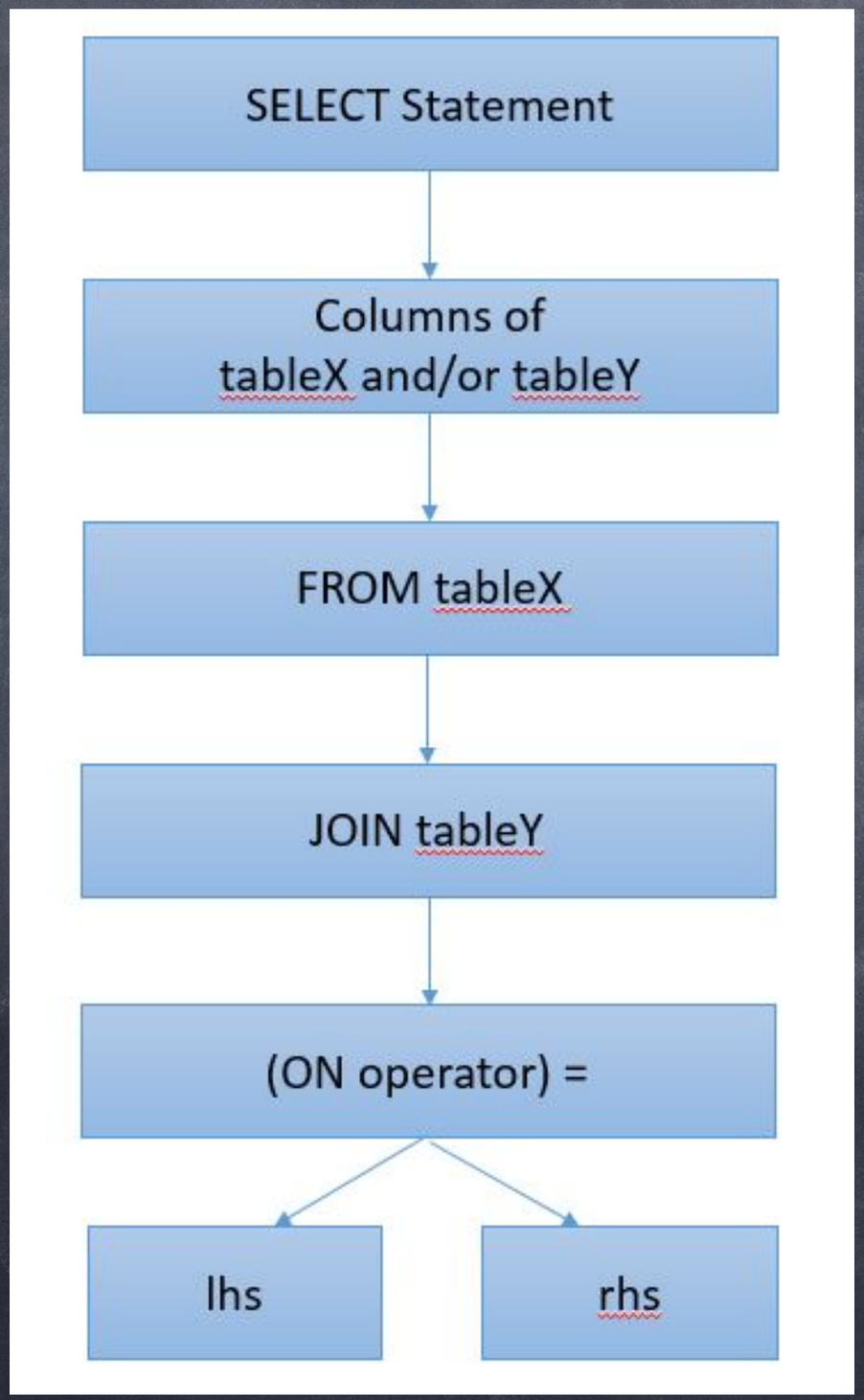
G. Implement the conversion of SQL subqueries to SPARQL subqueries in ReL (ReL Source Code).

- From ✓
- Where ✓
- Select
- JOIN

How it works

- SQL Query
- SQL Server
- IF SELECT
- SQLVisitor.getSelect
- visitSelect_buildSPARQL
- SPARQL

Old design


SELECT Statement
Columns of tableX and/or tableY
FROM tableX
JOIN tableY
(ON operator) =
lhs
rhs

Example without subqueries

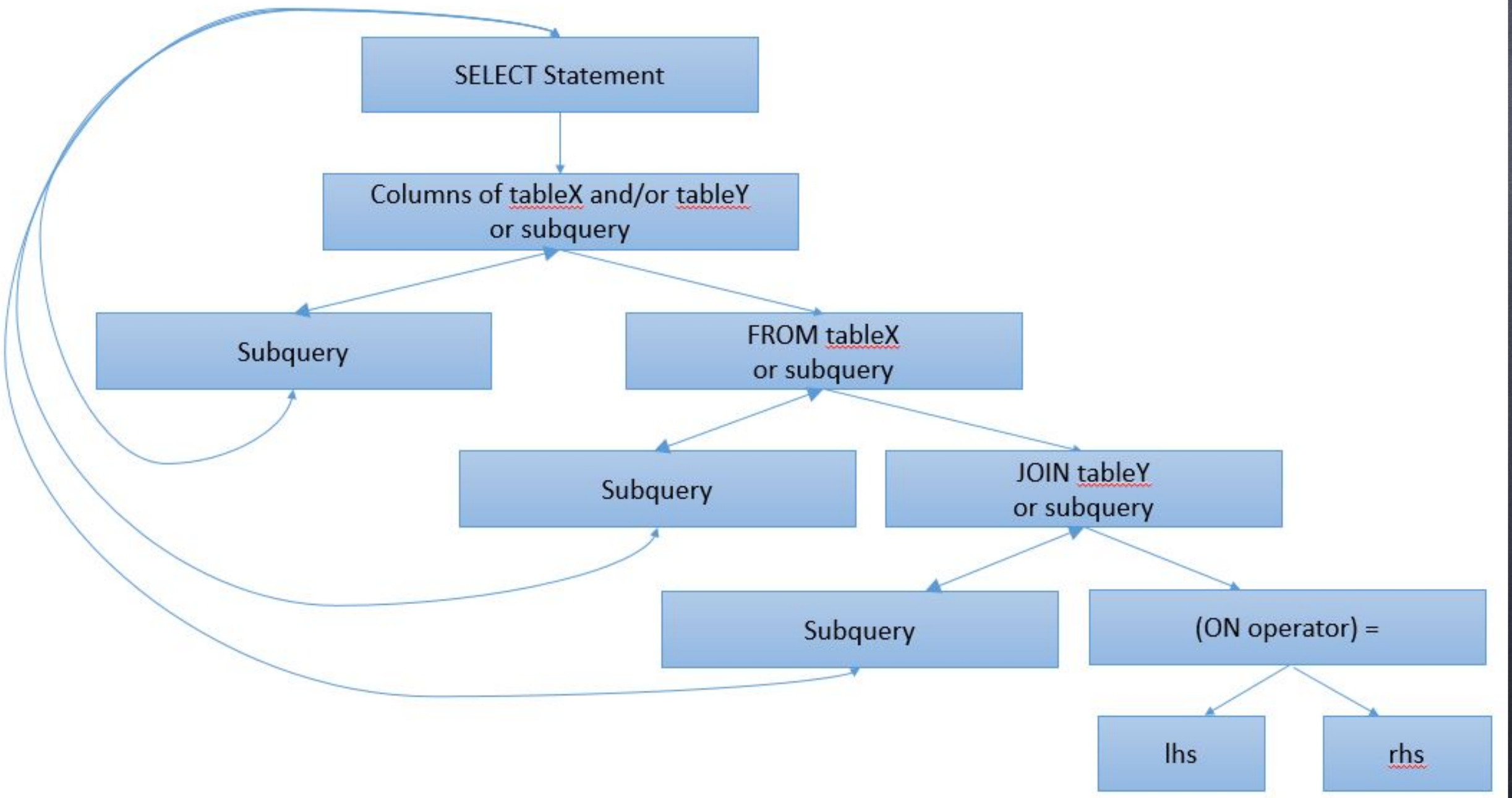
SQL:

```
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


SELECT Statement
Columns of tableX and/or tableY
or subquery
Subquery
FROM tableX
or subquery
Subquery
JOIN tableY
or subquery
Subquery
(ON operator) =
lhs
rhs

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(PlaintSelect plaintSelect) 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 epmno 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

Wiki SubSelect http://www.w3.org/2009/sparql/wiki/Design:SubSelect

https://apex.oracle.com/pls/apex/f?p=91561:3:13860963962818:::::