Test Report on gStore v0.3.0

Li Zeng[1]

October 5, 2016

[1] EECS of Peking University, zengli-bookug@pku.edu.cn

Contents

1 Preface

gStore[1] is a graph-based database management system, which keeps the structure of original RDF[2] data.

The data model is directed graph with labels, and each vertex corresponds to a subject or object.

Given a SPARQL[3] query(only select...where clause is well supported now), gStore will transfer it to a directed graph with labels first.

Then the query problem will be equivalent to a subgraph matching problem. An index called VSTree is used in gStore to speed up the matching process. For each variable in the SPARQL query, gStore acquires its candidates through VSTree, and finally a join process is performed to get the final result.

We compare the performance of gStore with apache-jena[4], openrdf-sesame[5] and virtuoso-openlinksw[6] on several RDF datasets. The items needing to be considered include the time to build database, the size of database and the time to answer each SPARQL query. In addition, we will give a special explanation if the query results of each database do not match. (we will not consider the memory and disk cost except for special cases)

[1] https://github.com/Caesar11/gStore

[2] http://www.w3school.com.cn/rdf/

[3] https://www.w3.org/TR/sparql11-query/

[4] http://jena.apache.org/

[5] http://www.rdf4j.org/

[6] http://virtuoso.openlinksw.com/

2 Environment Setup

We need to do some preparations before the experiment to ensure all datasets and corresponding queries can be run correctly by all database management systems. (**Notice that we only compare gStore and Jena in this report**) The limitations are listed below:

1. Jena does not support datasets with prefix declarations
2. Sesame does not support entities without appropiate prefixes
3. Virtuoso will remove the '<' and '>' for entities, '"' for literals

We should not include the time to load database indexes(called offline time) when comparing the time to answer SPARQL queries. And we need to empty the buffer and cache of operation system when the experiment for each database management system is over.

The datasets used include WatDiv[7], LUBM[8] and DBpedia[9]. DBpedia are the background data of wikipedia, while the others are generated by programs. SPARQL queries are generated by programs or written manually.

When testing DBpedia, some queries contain "^^" and gStore will ouput nothing in these cases. In addition, if the output results contain "^^", gStore and Virtuoso will ignore the properties linked by "^^". What is more, Sesame does not support LUBM due to invalid IRI, and it is unable to deal with too large datasets like dbpedia2014, watdiv_300M. Virtuoso can not deal with watdiv_300M, lubm_5000 and dbpedia2014, neither.

[7] http://dsg.uwaterloo.ca/watdiv/

[8] http://swat.cse.lehigh.edu/projects/lubm/

[9] http://wiki.dbpedia.org/

The experiment is finished on a Linux server with 128G memory and 4T disk. CentOS3.10.0 is installed and we require that the version of glibc should be at least 2.14. (78 server, whose ip is 172.31.222.78)

The versions of all database management systems used here are all open source. Latest versions are choosed, i.e. 3.0.1 for apache-jena, 4.1.1 for openrdf-sesame and 7.2 for virtuoso-openlinksw.

3 Experiment Result

All results are saved in load.log/, result.log/ and time.log/, and the format is TSV. Table 1 shows the index size and loading time of the datasets for different systems.

	Index Size(KB)		Loading Time(ms)	
Datasets	gStore	Jena	gStore	Jena
dbpedia170M	25,549,812	23,151,404	4,516,359	28,567,000
lubm6M	1,931,828	1,022,508	235,703	66,000
lubm66M	20,169,944	10,262,520	2,729,173	771,000
watdiv109M	14,146,572	12,731,136	3,253,110	1,884,000
watdiv219M	28,899,484	25,441,820	10,737,353	4,058,000
watdiv329M	43,276,644	37,950,456	11,549,843	6,409,000

Table 1: Offline Performance

The performance of different database management systems is shown in Figures 1, 2 and 3.

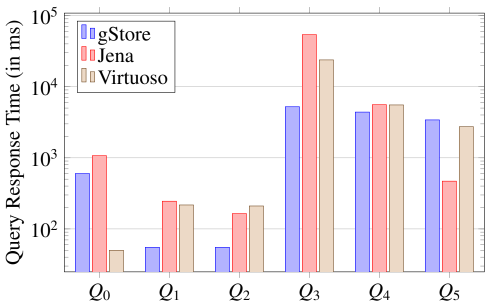
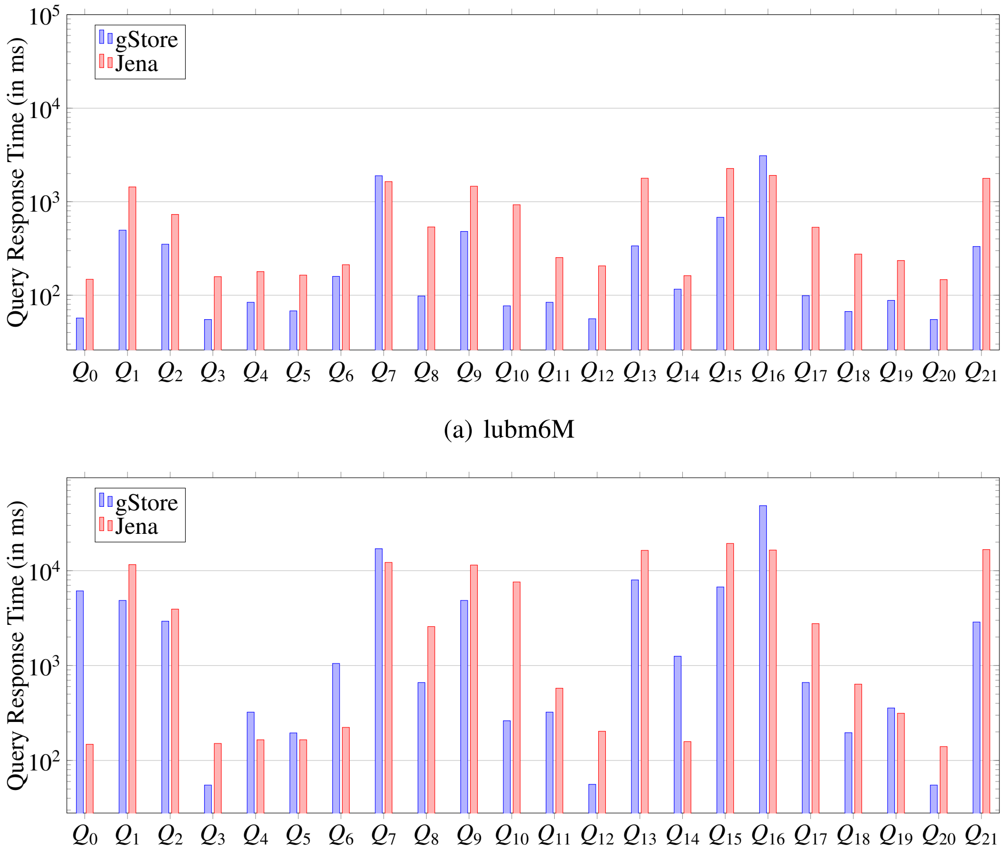


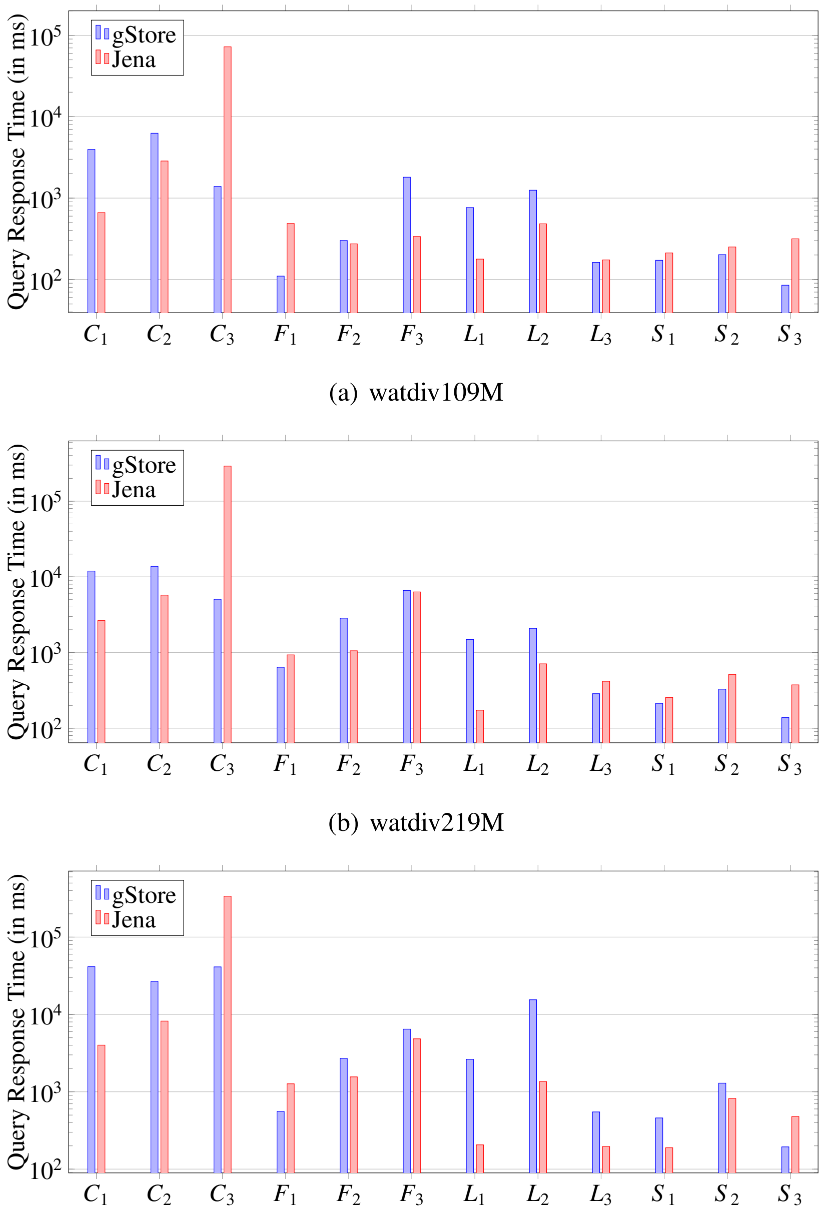
Figure 1: Query Performance over dbpedia170M



(a) lubm6M

(b) lubm66M

Figure 2: Query Performance over LUBM



(a) watdiv109M

(b) watdiv219M

(c) watdiv329M

Figure 3: Query Performance over WatDiv

4 Comparative Analysis

For most cases, query results of each database management system are matched, which means that the correctness of gStore is guaranteed. However, there are a few cases we need to pick out:

1. The answer of q0.sql in LUBM series are all empty if using Jena(should not be empty)
2. The answer of q4.sql in DBpedia2014 are not matched(containing unquoted numbers and ^^)
3. gStore produces more results for q6.sql of DBpedia2014 than Jena and consumes much more time

The first two questions are not the fault of gStore, but the third one does. However, we do not need to deal with it urgently because q6.sql is a rare case which can hardly be seen in real life.

Both time cost and disk cost of gStore are generally higher than the others when analysing the load.log/, about one order of magnitude. However, the size of the database generated by gStore can be reduced because some files are useless(for example, six_tuples and signaturebinary).

By analysing the time.log/, we can discover that gStore performs not much worser(even better) than Jena, including very complicated SPARQL queries and star-shape queries. These database management systems all performs well on small datasets. (query time usually less than 1s)

All datasets and queries we used are listed in this document, to provide a more thorough understanding of the experiment results.

Below is for the WatDiv datasets, and the corresponding queries are placed in WatDiv Queries.

Dataset	Size	Triple	Predicate	Entity	Literal
watdiv109M	15,743,004,966	109,795,918	86	5,212,745	5,077,247
watdiv219M	31,712,545,025	219,714,495	86	10,424,745	9,976,964
watdiv329M	47,676,280,476	329,584,783	86	15,636,745	14,748,846

Table 2: WatDiv series

For WatDiv datasets, gStore performs worser in C1.sql, C2.sql and L2.sql. The reason is that all vertices are filtered by VSTree, but the candidate num is still very large. There are many edges in L2.sql, so the cost of join process can be very high(the result set is small).

Below is for the LUBM datasets, and the corresponding queries are placed in LUBM Queries.

Dataset	Size	Triple	Predicate	Entity	Literal
lubm6M	8,134,671,485	66,718,642	17	16,437,950	0
lubm66M	801,112,089	6,652,613	17	1,648,692	0

Table 3: LUBM series

For LUBM datasets, gStore performs worser in q0.sql and q16.sql. However, Jena produce wrong result(empty) for q0.sql, so we do not consider the gap in q0.sql now. As for q16.sql, the result size is 65650, and this query is a circle consist of 3 variables.

Below is for the DBpedia datasets, and the corresponding queries are placed in DBpedia Queries.

For DBpedia datasets, gStore performs worser in q5.sql and q6.sql. The reason may be that the result set of this two are too large. Another feature of this two

Dataset	Size	Triple	Predicate	Entity	Literal
dbpedia170M	23,844,158,944	170,784,508	57,354	7,123,915	14,971,449

Table 4: DBpedia series

queries is that they are both star-like graphs.

5 Modification

We provide insertion and deletion in gStore v0.3.0. You can either insert/delete from a given RDF file, or just run sparql queries to insert/delete something. If you want to modify something, you need to delete it and reinsert. The cost of insertion and deletion are recorded in table 5, where the time unit is ms. The first two delete/insert 10000 triples, while the third one delete/insert 100000 triples.

Dataset	build	insert	delete
dbpedia170M	4,536,355	10,401,333	13,861,835
lubm66M	2,723,493	2,956,092	18,623,167
watdiv109M	3,246,542	3,964,283	2,156,458

Table 5: Insertion and Deletion

However, bugs do exist in insertion/deletion. When testing on lubm66M, the answer of q1.sql and q2.sql are not all right if the operation order is: build, delete, insert, query. More precisely, a few results are lost, though the proportion is really small. We do not care about the efficiency of insert/delete, but the correctness is a must, which means we will try to fix this bug as quickly as possible.

6 Conclusion

gStore can go well with RDF datasets which are in N-Triples format and TTL format, while the other database management systems may come across some questions. In addition, gStore outperforms other systems on many SPARQL queries. What is more, gStore is highly extensively because it uses graph model instead of relational model.

However, there are also some shortcomings for gStore:

1. RDF datasets in XML format are not supported
2. the disk cost is high
3. gStore is sometimes slower

Besides, gStore v0.3.0 does not generate solutions for satellites which are not selected. This will speed up the query answering process, while not keeping so many duplicates in the result set. For example, in below query, let's assume that ?s has only one unique answer, but ?o1 and ?o2 both have 10,000 answers. In previous versions of gStore, there are 100,000,000 records in the result set because we have to find the answer of ?s and generate the solutions for ?o1 and ?o2, even if only ?s is selected in the sparql query. However, in the v0.3.0, we find the answer of ?s and return directly. In this case, there won't be so many duplicates in the result set as before, but this is ok.

```
select  ?s  where
{
  ?s  <linkTo> ?o1  .
  ?s  <produce> ?o2  .
}
```

7 Prospective

Out of question, the performance of gStore can be improved a lot later. The future work is listed below:

1. fix the problem in insertion/deletion
2. support datasets of 1 billion triples in a single machine(only 500 million now)
3. add unit testing for the whole system(only black-box testing now)
4. do code level optimization(for example, large loop in Join module)
5. speed up the table join process using pipeline

8 Appendix

8.1 WatDiv queries

8.1.1 C1.sql

```
SELECT ?v0 ?v4 ?v6 ?v7 WHERE
{
  ?v0 <http :// schema.org/ caption> ?v1 .
  ?v0 <http :// schema.org/ text> ?v2 .
  ?v0 <http :// schema.org/ contentRating> ?v3 .
  ?v0 <http :// purl . org/ stuff / rev#hasReview> ?v4 .
  ?v4 <http :// purl . org/ stuff / rev# title > ?v5 .
  ?v4 <http :// purl . org/ stuff / rev#reviewer> ?v6 .
  ?v7 <http :// schema.org/ actor> ?v6 .
  ?v7 <http :// schema.org/language> ?v8 .
}
```

8.1.2 C2.sql

```
SELECT ?v0 ?v3 ?v4 ?v8 WHERE
{
  ?v0 <http :// schema.org/legalName> ?v1 .
  ?v0 <http :// purl . org/ goodrelations / offers > ?v2 .
  ?v2 <http :// schema.org/ eligibleRegion > <http :// db.uwaterloo . ca/~
        galuc / wsdbm/Country5> .
  ?v2 <http :// purl . org/ goodrelations / includes > ?v3 .
  ?v4 <http :// schema.org/ jobTitle > ?v5 .
  ?v4 <http :// xmlns.com/foaf/homepage> ?v6 .
  ?v4 <http :// db.uwaterloo . ca/~galuc / wsdbm/makesPurchase> ?v7 .
  ?v7 <http :// db.uwaterloo . ca/~galuc / wsdbm/purchaseFor> ?v3 .
  ?v3 <http :// purl . org/ stuff / rev#hasReview> ?v8 .
  ?v8 <http :// purl . org/ stuff / rev# totalVotes > ?v9 .
}
```

8.1.3 C3.sql

```
SELECT ?v0 WHERE
{
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/likes> ?v1 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/friendOf> ?v2 .
  ?v0 <http :// purl.org/dc/terms/Location> ?v3 .
  ?v0 <http :// xmlns.com/foaf/age> ?v4 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/gender> ?v5 .
  ?v0 <http :// xmlns.com/foaf/givenName> ?v6 .
}
```

8.1.4 F1.sql

```
SELECT ?v0 ?v2 ?v3 ?v4 ?v5 WHERE
{
  ?v0 <http :// ogp.me/ns#tag> <http :// db.uwaterloo.ca/~galuc/wsdbm/
        Topic103> .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v2 .
  ?v3 <http :// schema.org/ trailer > ?v4 .
  ?v3 <http :// schema.org/keywords> ?v5 .
  ?v3 <http :// db.uwaterloo.ca/~galuc/wsdbm/hasGenre> ?v0 .
  ?v3 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http://db.
        uwaterloo.ca/~galuc/wsdbm/ProductCategory2> .
}
```

8.1.5 F2.sql

```
SELECT ?v0 ?v1 ?v2 ?v4 ?v5 ?v6 ?v7 WHERE
{
  ?v0 <http :// xmlns.com/foaf/homepage> ?v1 .
  ?v0 <http :// ogp.me/ns# title > ?v2 .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v3 .
  ?v0 <http :// schema.org/caption> ?v4 .
  ?v0 <http :// schema.org/ description > ?v5 .
  ?v1 <http :// schema.org/ url> ?v6 .
  ?v1 <http :// db.uwaterloo.ca/~galuc/wsdbm/hits> ?v7 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/hasGenre> <http://db.
        uwaterloo.ca/~galuc/wsdbm/SubGenre35> .
}
```

8.1.6 F3.sql

```
SELECT ?v0 ?v1 ?v2 ?v4 ?v5 ?v6 WHERE
{
  ?v0 <http :// schema.org/ contentRating> ?v1 .
  ?v0 <http :// schema.org/ contentSize> ?v2 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/hasGenre> <http://db.
        uwaterloo.ca/~galuc/wsdbm/SubGenre59> .
  ?v4 <http :// db.uwaterloo.ca/~galuc/wsdbm/makesPurchase> ?v5 .
  ?v5 <http :// db.uwaterloo.ca/~galuc/wsdbm/purchaseDate> ?v6 .
  ?v5 <http :// db.uwaterloo.ca/~galuc/wsdbm/purchaseFor> ?v0 .
}
```

8.1.7 L1.sql

```
SELECT ?v0 ?v2 ?v3 WHERE
{
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/subscribes> <http :// db.
        uwaterloo.ca/~galuc/wsdbm/Website38303> .
  ?v2 <http :// schema.org/ caption> ?v3 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/likes> ?v2 .
}
```

8.1.8 L2.sql

```
SELECT ?v1 ?v2 WHERE
```

```
{
  <http://db.uwaterloo.ca/~galuc/wsdbm/City131> <http://www.geonames.org/ontology#parentCountry> ?v1 .
  ?v2 <http://db.uwaterloo.ca/~galuc/wsdbm/likes> <http://db.uwaterloo.ca/~galuc/wsdbm/Product0> .
  ?v2 <http://schema.org/nationality> ?v1 .
}
```

8.1.9 L3.sql

```
SELECT ?v0 ?v1 WHERE
{
  ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/likes> ?v1 .
  ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/subscribes> <http://db.uwaterloo.ca/~galuc/wsdbm/Website20769> .
}
```

8.1.10 S1.sql

```
SELECT ?v0 ?v1 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8 ?v9 WHERE
{
  ?v0 <http://purl.org/goodrelations/includes> ?v1 .
  <http://db.uwaterloo.ca/~galuc/wsdbm/Retailer391> <http://purl.org/goodrelations/offers> ?v0 .
  ?v0 <http://purl.org/goodrelations/price> ?v3 .
  ?v0 <http://purl.org/goodrelations/serialNumber> ?v4 .
  ?v0 <http://purl.org/goodrelations/validFrom> ?v5 .
  ?v0 <http://purl.org/goodrelations/validThrough> ?v6 .
  ?v0 <http://schema.org/eligibleQuantity> ?v7 .
  ?v0 <http://schema.org/eligibleRegion> ?v8 .
  ?v0 <http://schema.org/priceValidUntil> ?v9 .
}
```

8.1.11 S2.sql

```
SELECT ?v0 ?v1 ?v3 WHERE
{
  ?v0 <http :// purl .org/dc/terms/Location> ?v1  .
  ?v0 <http :// schema.org/ nationality > <http :// db.uwaterloo .ca/~galuc
          /wsdbm/Country23> .
  ?v0 <http :// db.uwaterloo .ca/~galuc/wsdbm/gender> ?v3 .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http://db.
          uwaterloo .ca/~galuc/wsdbm/Role2> .
}
```

8.1.12 S3.sql

```
SELECT ?v0 ?v2 ?v3 ?v4 WHERE
{
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http://db.
          uwaterloo .ca/~galuc/wsdbm/ProductCategory12> .
  ?v0 <http :// schema.org/ caption > ?v2  .
  ?v0 <http :// db.uwaterloo .ca/~galuc/wsdbm/hasGenre> ?v3 .
  ?v0 <http :// schema.org/ publisher > ?v4  .
}
```

8.2 LUBM queries

8.2.1 q0.sql

```
 select  ?x  where
{
  ?x        <ub:name>        <FullProfessor0 >.
}
```

8.2.2 q1.sql

```
select ?x where
{
  ?x      <rdf : type>      <ub:GraduateStudent>.
  ?y      <rdf : type>      <ub: University >.
  ?z      <rdf : type>      <ub:Department>.
  ?x      <ub:memberOf>   ?z.
  ?z      <ub:subOrganizationOf>   ?y .
  ?x      <ub:undergraduateDegreeFrom>    ?y .
}
```

8.2.3 q2.sql

```
select ?x where
{
  ?x      <rdf : type>      <ub:Course>.
  ?x      <ub:name>         ?y .
}
```

8.2.4 q3.sql

```
select ?x where
{
  ?x      <rdf : type>      <ub:UndergraduateStudent>.
  ?y      <rdf : type>      <ub: University >.
  ?z      <rdf : type>      <ub:Department>.
  ?x      <ub:memberOf>   ?z.
  ?z      <ub:subOrganizationOf>   ?y .
  ?x      <b:undergraduateDegreeFrom>     ?y .
}
```

8.2.5 q4.sql

```
elect ?x ?y1 ?y2 ?y3 where
{
  ?x      <ub:worksFor>   <http :// www.Department0.University0.edu>.
  ?x      <rdf:type>      <ub: FullProfessor >.
  ?x      <ub:name>       ?y1.
  ?x      <ub:emailAddress>        ?y2.
  ?x      <ub:telephone>  ?y3.
}
```

8.2.6 q5.sql

```
select ?x where
{
  ?x      <ub:subOrganizationOf>  <http :// www.Department0.
         University0.edu>.
  ?x      <rdf:type>      <ub:ResearchGroup>.
}
```

8.2.7 q6.sql

```
select ?x ?y where
{
  ?y      <ub:subOrganizationOf>  <http :// www.University0.edu>.
  ?y      <rdf:type>      <ub:Department>.
  ?x      <ub:worksFor>   ?y.
  ?x      <rdf:type>      <ub: FullProfessor >.
}
```

8.2.8 q7.sql

```
select  ?x  ?y  ?z  where
{
  ?x        <rdf:type>        <ub:UndergraduateStudent>.
  ?y        <rdf:type>        <ub: FullProfessor >.
  ?z        <rdf:type>        <ub:Course>.
  ?x        <ub:advisor>      ?y.
  ?x        <ub:takesCourse>            ?z.
  ?y        <ub:teacherOf>   ?z.
}
```

8.2.9 q8.sql

```
select  ?X where
{
  ?X        <rdf:type>        <ub:GraduateStudent>.
  ?X        <ub:takesCourse>            <http :// www.Department0.
          University0.edu/GraduateCourse0>.
}
```

8.2.10 q9.sql

```
select  ?X ?Y ?Z where
{
  ?X        <rdf:type>        <ub:GraduateStudent>.
  ?Y        <rdf:type>        <ub: University >.
  ?Z        <rdf:type>        <ub:Department>.
  ?X        <ub:memberOf>   ?Z.
  ?Z        <ub:subOrganizationOf>   ?Y.
  ?X        <ub:undergraduateDegreeFrom>     ?Y.
}
```

8.2.11 q10.sql

```
select  ?X where
{
  ?X      <rdf:type>       <ub:Publication>.
  ?X      <ub:publicationAuthor>   <http://www.Department0.
          University0.edu/AssistantProfessor0>.
}
```

8.2.12 q11.sql

```
select  ?Y1 ?Y2 ?Y3 where
{
  ?X      <rdf:type>       <ub:FullProfessor>.
  ?X      <ub:worksFor>    <http://www.Department0.University0.edu>.
  ?X      <ub:name>        ?Y1.
  ?X      <ub:emailAddress>        ?Y2.
  ?X      <ub:telephone>   ?Y3.
}
```

8.2.13 q12.sql

```
select  ?X where
{
  ?X      <ub:memberOf>   <http://www.Department0.University0.edu>.
}
```

8.2.14 q13.sql

```
select  ?X where
{
  ?X      <rdf:type>       <ub:UndergraduateStudent>.
}
```

8.2.15 q14.sql

```
elect  ?X ?Y where
{
  ?X      <rdf:type>      <ub:Student>.
  ?Y      <rdf:type>      <ub:Course>.
  ?X      <ub:takesCourse>        ?Y.
  <http :// www.Department0.University0.edu/ AssociateProfessor0>     <
          ub:teacherOf>   ?Y.
}
```

8.2.16 q15.sql

```
select  ?X where
{
  ?X      <rdf:type>      <ub:UndergraduateStudent>.
  ?Y      <rdf:type>      <ub:Department>.
  ?X      <ub:memberOf>   ?Y.
  ?Y      <ub:subOrganizationOf>  <http :// www.University0.edu>.
  ?X      <ub:emailAddress>       ?Z.
}
```

8.2.17 q16.sql

```
select  ?X ?Y ?Z where
{
  ?X      <rdf:type>      <ub:UndergraduateStudent>.
  ?Z      <rdf:type>      <ub:Course>.
  ?X      <ub:advisor>    ?Y.
  ?Y      <ub:teacherOf>  ?Z.
  ?X      <ub:takesCourse>        ?Z.
}
```

8.2.18 q17.sql

```
select ?X where
{
  ?X      <rdf:type>       <ub:GraduateStudent>.
  ?X      <ub:takesCourse>          <http :// www.Department0.University0.edu/GraduateCourse0>.
}
```

8.2.19 q18.sql

```
select ?X where
{
  ?X      <rdf:type>       <ub:ResearchGroup>.
  ?X      <ub:subOrganizationOf>  <http :// www.University0.edu>.
}
```

8.2.20 q19.sql

```
select ?X ?Y where
{
  ?Y      <rdf:type>       <ub:Department>.
  ?X      <ub:worksFor>   ?Y.
  ?Y      <ub:subOrganizationOf>  <http :// www.University0.edu>.
}
```

8.2.21 q20.sql

```
select ?X where
{
  <http :// www.University0.edu>    <ub:undergraduateDegreeFrom>    ?X
           .
}
```

8.2.22 q21.sql

```
select ?X where
{
  ?X       <rdf:type>       <ub:UndergraduateStudent>.
}
```

8.3 DBpedia queries

8.3.1 q0.sql

```
select ?v0 where
{
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http://
        dbpedia.org/ class /yago/LanguagesOfBotswana> .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type>   <http ://
        dbpedia.org/ class /yago/LanguagesOfNamibia> .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http://
        dbpedia.org/ontology/Language> .
}
```

8.3.2 q1.sql

```
select ?v0 where
{
  ?v0 <http :// dbpedia.org/ontology/associatedBand> <http :// dbpedia.
        org/ resource /LCD_Soundsystem> .
}
```

8.3.3 q2.sql

```
select  ?v2 where
{
  <http :// dbpedia.org/ resource /!! Destroy−Oh−Boy!!> <http://dbpedia.
          org/ property / title > ?v2 .
}
```

8.3.4 q3.sql

```
select  ?v0 ?v2 where
{
  ?v0 <http :// dbpedia.org/ontology/ activeYearsStartYear > ?v2 .
}
```

8.3.5 q4.sql

```
select  ?v0 ?v1 ?v2 where
{
  ?v0 <http :// dbpedia.org/ property / dateOfBirth> ?v2 .
  ?v1 <http :// dbpedia.org/ property /genre> ?v2 .
}
```

8.3.6 q5.sql

```
select  ?v0 ?v1 ?v2 ?v3 where
{
  ?v0 <http :// dbpedia.org/ property / familycolor> ?v1 .
  ?v0 <http :// dbpedia.org/ property / glotto> ?v2 .
  ?v0 <http :// dbpedia.org/ property / lc> ?v3 .
}
```

8.3.7 q6.sql

```
select  ?v0 ?v1 ?v2 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8 ?v9 where
{
  ?v0 <http://dbpedia.org/property/dateOfBirth> ?v1 .
  ?v0 <http://dbpedia.org/property/genre> ?v2 .
  ?v0 <http://dbpedia.org/property/instrument> ?v3 .
  ?v0 <http://dbpedia.org/property/label> ?v4 .
  ?v0 <http://dbpedia.org/property/placeOfBirth> ?v5 .
  ?v6 <http://dbpedia.org/property/name> ?v7 .
  ?v6 <http://dbpedia.org/property/occupation> ?v8 .
  ?v6 <http://dbpedia.org/property/placeOfBirth> ?v5 .
  ?v6 <http://dbpedia.org/property/instrument> ?v3 .
  ?v6 <http://dbpedia.org/property/notableInstruments> ?v9 .
}
```