

به نام خدا



دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر

پروژه  
وب معنایی

دکتر ابوالحسنی

۸۶۲۰۸۴۵۸

مسعود مشرف جوادی

۸۵۲۰۴۲۵۱

علیرضا صدیقیان

۸۶۲۰۸۷۰۳

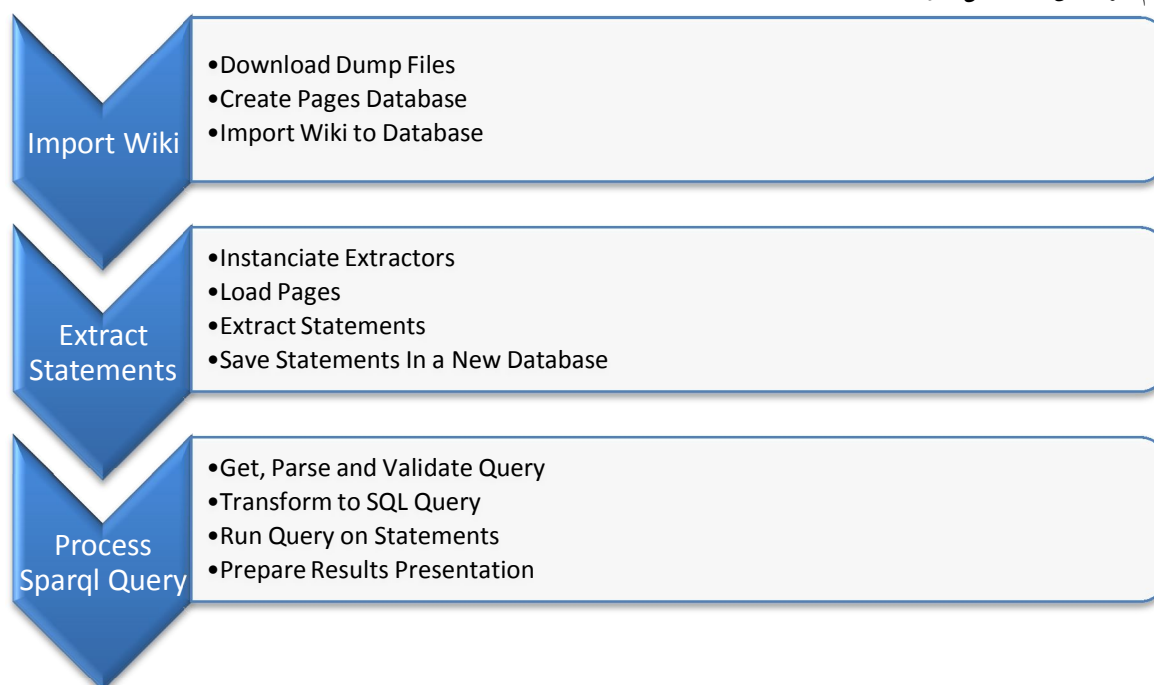
محمد قریب

بهار ۱۳۸۷

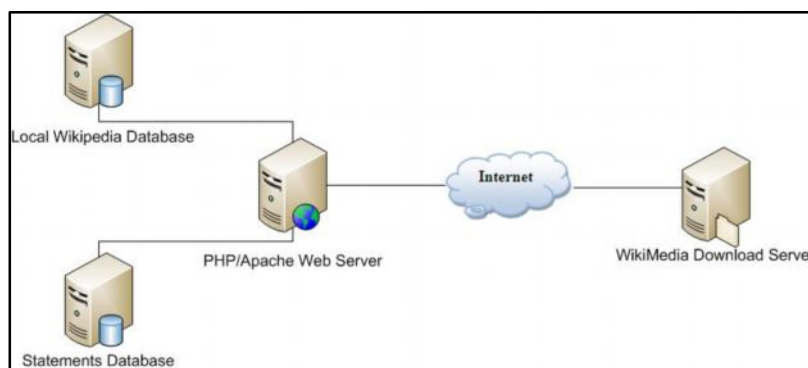
## فهرست

- ۱- فاز ورود داده‌ها ..... ۲
- ۲- استخراج عبارات سه‌تایی ..... ۴
- ۳- دریافت، پردازش پرسش و نمایش نتیجه ..... ۷

شکل ۱ سه فاز انجام پروژه را به صورت خلاصه نشان می‌دهد. همچنین سرورها و نحوه ارتباط آنها در شکل ۲ آمده است. در ادامه مراحل انجام هر بخش به تفصیل خواهد آمد.



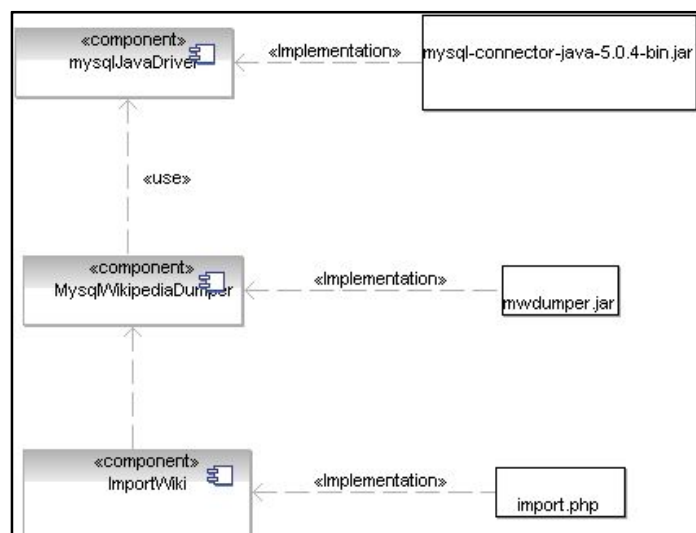
شکل ۱ خلاصه روند انجام پروژه



شکل ۲ ساختار و نحوه ارتباط سرورها

## فاز ورود داده‌ها

این فاز وظیفه دریافت فایل‌های dump پروژه ویکی‌پدیا فارسی و ورود آنها به یک پایگاه‌داده را به عهده دارد. در پایگاه‌داده ساخته شده دسته‌ها، مشخصات هر صفحه، متن صفحات، رابطه دسته‌ها با هم، دسته‌ها و صفحات، و صفحات با هم ذخیره می‌شوند. شکل ۳ ساختار مؤلفه‌ها و فایل‌های این بسته را نشان می‌دهد.



شکل ۳ نمودار مؤلفه اجزای فاز ورود داده‌ها

شکل ۴ ساختار جداولی از پایگاه داده نرم افزار ویکی پدیا که داده‌هایی در آن ذخیره شده‌اند را نشان می‌دهد.

|                 |               |          |
|-----------------|---------------|----------|
| <b>image</b>    |               |          |
| Column          |               |          |
| img_name        | VARCHAR(255)  | NOT NULL |
| img_size        | INT(10)       | NOT NULL |
| img_width       | INT           | NOT NULL |
| img_height      | INT           | NOT NULL |
| img_metadata    | MEDIUMBLOB    | NOT NULL |
| img_bits        | INT           | NOT NULL |
| img_media_type  | ENUM          | NULL     |
| img_major_mime  | ENUM          | NOT NULL |
| img_minor_mime  | VARBINARY(32) | NOT NULL |
| img_description | TINYBLOB      | NOT NULL |
| img_user        | INT(10)       | NOT NULL |
| img_user_text   | VARCHAR(255)  | NOT NULL |
| img_timestamp   | VARBINARY(14) | NOT NULL |
| img_sha1        | VARBINARY(32) | NOT NULL |

|                   |              |          |
|-------------------|--------------|----------|
| <b>imagelinks</b> |              |          |
| Column            |              |          |
| il_from           | INT(10)      | NOT NULL |
| il_to             | VARCHAR(255) | NOT NULL |

|                      |              |          |
|----------------------|--------------|----------|
| <b>categorylinks</b> |              |          |
| Column               |              |          |
| cl_from              | INT(10)      | NOT NULL |
| cl_to                | VARCHAR(255) | NOT NULL |
| cl_sortkey           | VARCHAR(70)  | NOT NULL |
| cl_timestamp         | TIMESTAMP    | NOT NULL |

|                   |              |          |
|-------------------|--------------|----------|
| <b>page</b>       |              |          |
| Column            |              |          |
| page_id           | INT(10)      | NOT NULL |
| page_namespace    | INT          | NOT NULL |
| page_title        | VARCHAR(255) | NOT NULL |
| page_restrictions | TINYBLOB     | NOT NULL |
| page_counter      | BIGINT       | NOT NULL |
| page_is_redirect  | TINYINT(3)   | NOT NULL |
| page_is_new       | TINYINT(3)   | NOT NULL |
| page_random       | VARCHAR(20)  | NOT NULL |
| page_touched      | BINARY(14)   | NOT NULL |
| page_latest       | INT(10)      | NOT NULL |
| page_len          | INT(10)      | NOT NULL |

|             |            |          |
|-------------|------------|----------|
| <b>text</b> |            |          |
| Column      |            |          |
| old_id      | INT(10)    | NOT NULL |
| old_text    | MEDIUMBLOB | NOT NULL |
| old_flags   | TINYBLOB   | NOT NULL |

|                      |              |          |
|----------------------|--------------|----------|
| <b>templatelinks</b> |              |          |
| Column               |              |          |
| tl_from              | INT(10)      | NOT NULL |
| tl_namespace         | INT          | NOT NULL |
| tl_title             | VARCHAR(255) | NOT NULL |

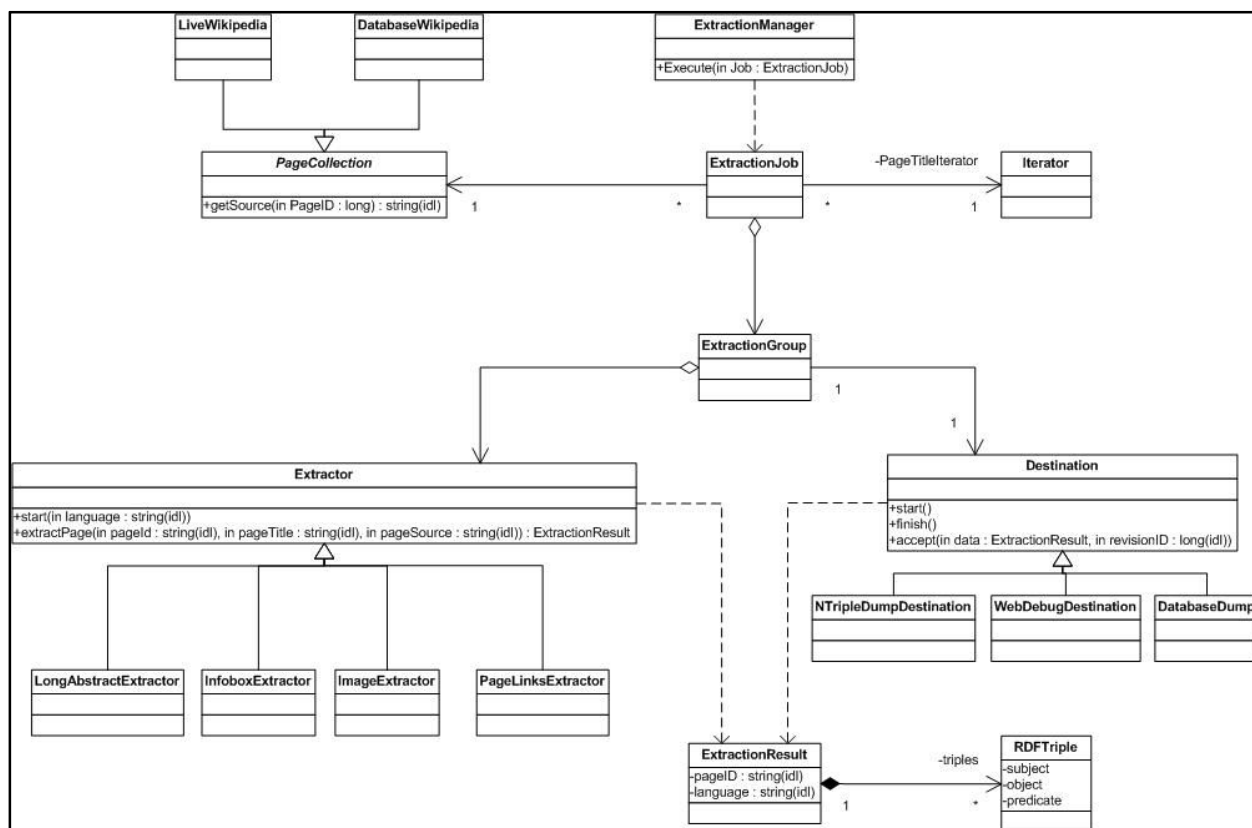
  

|                 |              |          |
|-----------------|--------------|----------|
| <b>category</b> |              |          |
| Column          |              |          |
| cat_id          | INT(10)      | NOT NULL |
| cat_title       | VARCHAR(255) | NOT NULL |
| cat_pages       | INT          | NOT NULL |
| cat_subcats     | INT          | NOT NULL |
| cat_files       | INT          | NOT NULL |
| cat_hidden      | TINYINT(3)   | NOT NULL |

شکل ۴ بخشی از ساختار پایگاه داده نرم افزار ویکی پدیا

## استخراج عبارات سه تایی

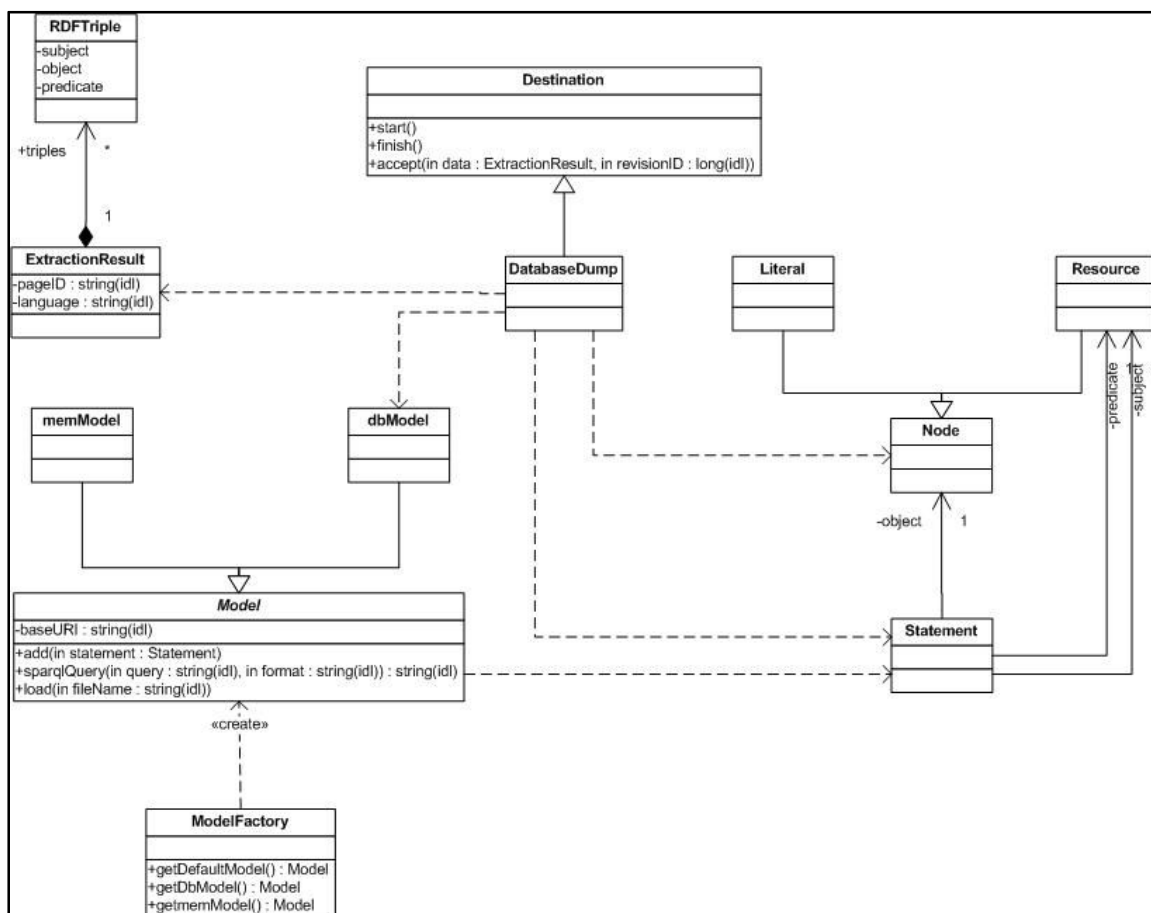
هدف این فاز استخراج عبارات سه تایی RDF از مستندات فارسی ویکی پدیا که اکنون در پایگاه داده ذخیره شده است می باشد. برای توضیح نحوه عملکرد این فاز لازم است ساختار کلاس های مربوط به استخراج در dbpedia را توضیح دهیم:



شکل ۵ ساختار کلاسی مؤلفه استخراج

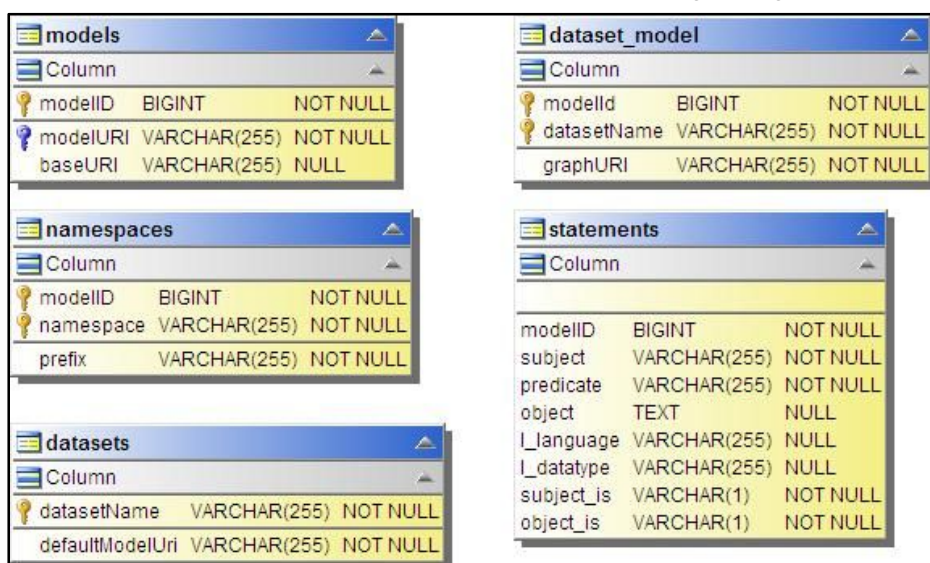
همانطور که در شکل ۵ نشان داده شده است. کلاس ExtractionManager مسئول مدیریت یک جلسه استخراج است. این مدیر یک شی از نوع ExtractionJob را اجرا می‌کند. هر ExtractionJob یک منبع صفحه از نوع PageCollection و یک Iterator برای حرکت روی این Collection یا انتخاب صفحات محدودی از این مجموعه دارد. همچنین هر ExtractionJob چندین ExtractionGroup دارد. هدف ساختن کلاس ExtractionGroup داشتن خروجی‌های مختلف به ازای یک بار عمل استخراج است. یعنی هر ExtractionGroup یک خروجی از نوع Destination و چندین Extractor دارد. Extractor ها در واقع وظیفه استخراج اطلاعات لازم از محتوا یا اطلاعات صفحه را برعهده دارند. و نتیجه را در قالب یک شی ExtractionResult که حاوی چندین RDFTriple است بازمی‌گردانند.

کلاس DatabaseDump که وظیفه ذخیره‌سازی statement های بدست آمده در پایگاه داده جدیدی را دارد به صورت شکل ۶ با کلاس‌های دیگر در ارتباط است:



شکل ۶ ساختار کلاسی مؤلفه ذخیره‌سازی سه‌تایی‌های RDF در پایگاه‌داده

ساختار پایگاه‌داده سه‌تایی‌های حاصل در شکل ۷ نشان داده شده است



شکل ۷ ساختار پایگاه‌داده سه‌تایی‌های RDF تولید شده

## دریافت، پردازش پرسش و نمایش نتیجه

این فاز شامل دریافت پرسش از فرم ورودی، پردازش و اعتبارسنجی آن، تبدیل به پرسش SQL روی پایگاه داده عبارات سه تایی و نمایش نتایج است.

پس از دریافت پرسش مشتری، یک شیء از نوع dbModel از پایگاه داده سه تایی ها می سازیم و برای اجرای پرسش Sparql از متد sparqlQuery کلاس Model استفاده می کنیم. همچنین برای نمایش نتیجه کافیت نوع نمایش لازم را به این متد در قالب String پاس دهیم. شکل ۸ کد مربوط به این قسمت در فایل sparqlStart.php را نشان می دهد.

```
$database = ModelFactory::getDbStore(
    $GLOBALS['dbConf']['type'],
    $GLOBALS['dbConf']['host'],
    $GLOBALS['dbConf']['database'],
    $GLOBALS['dbConf']['user'],
    $GLOBALS['dbConf']['password']
);

$strModel = "http://xmlns.com/foaf/0.1/";
$dbModel = $database->getModel($strModel);

if ($dbModel === false) {
    die('Database does not have a model ' . $strModel . "\n");
}

$querystring = $_GET["query"];
echo $dbModel->sparqlQuery($querystring, $_GET["format"]);
```

شکل ۸ کد مربوط به اجرای پرسش Sparql

نمونه فرم و نتیجه نمایش داده شده به صورت جدول HTML در شکل ۹ نشان داده شده است. همچنین در ابتدای نتایج جهت فهم بهتر پرسش آن را به صورت SQL اجرا شده روی پایگاه داده نیز نشان داده ایم.

**FADBPeida Sparql Interface for Semantic Web course of Sharif University of Technology**

This interface is designed to test new or improved extractors and sparql queries.

Query

Default Graph URI  
http://dbpedia.org

Query text  
select ?x ?y where {?x ?y ?z  
filter regex(?z, <اسفهان>)}

Display Results As: HTML Run Query Reset

1-select ?x ?y where {?x ?y ?z filter regex(?z, <اسفهان>)} Array ( [0] => Array ( [select] => SELECT 10.subject as "10.value\_v\_x", 10.subject\_is as "10.is\_v\_x", 10.predicate as "10.value\_v\_y" [from] => FROM statements as 10 [where] => WHERE 10.modelID = 1 AND ((CASE WHEN 10.1\_datatype = "http://www.w3.org/2001/XMLSchema#boolean" THEN IF(LOWER(10.object) = "true", TRUE, FALSE) ELSE 10.object END) REGEXP "اسفهان" AND 10.object\_is = T) )

SPARQL result with 1573 rows

| ?x                                                                | ?y                                           |
|-------------------------------------------------------------------|----------------------------------------------|
| http://www4.wiwiwss.fu-berlin.de/wikicompany/resource/اسفهان_مرکز | http://www.w3.org/2000/01/rdf-schema#comment |
| http://www4.wiwiwss.fu-berlin.de/wikicompany/resource/اسفهان_مرکز | http://dbpedia.org/property/abstract         |
| http://www4.wiwiwss.fu-berlin.de/wikicompany/resource/اسفهان      | http://dbpedia.org/property/نام_مجموعه       |
| http://www4.wiwiwss.fu-berlin.de/wikicompany/resource/اسفهان      | http://dbpedia.org/property/اسفهان           |
| http://www4.wiwiwss.fu-berlin.de/wikicompany/resource/اسفهان      | http://dbpedia.org/property/شهرستان          |

شکل ۹ نمونه نتیجه یک پرسش Sparql

همچنین می توان نتایج را به صورت فایل XML داشت. شکل ۱۰ آن را نشان می دهد.

Query text

```
select ?x ?y where {?x ?y ?z
filter regex(?z,<?مطه‌بان>)}
```

Display Results As: HTML

Run Query

Reset

```
1:select ?x ?y where {?x ?y ?z
filter regex(?z,<2{مطه‌بان}:Array
{
  [0] => Array
  {
    [select] => SELECT t0.subject as "t0.value_v_x", t0.subject_is as "t0.is_v_x", t0.predicate as "t0.value_v_y"
    [from] => FROM statements as t0
    [where] => WHERE t0.modelID = 1 AND ((CASE WHEN t0.l_datatype = "http://www.w3.org/2001/XMLSchema#boolean" TH
```

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="x"/>
    <variable name="y"/>
  </head>
  <results ordered="false" distinct="false">
    <result>
      <binding name="x">uri<http://www4.wiwiiss.fu-berlin.de/wikicompany/resource/مطه‌بان</uri></binding>
```

شکل ۱۰ نمونه پرسشی یک پرسش Sparql در قالب XML