DataImportHandlerDeltaQueryViaFullImport

Using query attribute for both full and delta import

The standard approach in Solr is to define one query for the initial import and a second query to fetch the IDs of documents that have changed and a third query to fetch the data that changed. If you expect a large number of changes this isn't very efficient. Furthermore if both in the initial import and the delta case you have the same SELECT list, its tedious to maintain 3 queries where two are almost identical and one still very similar. The fundamental idea is to only define one query for both the full and delta import, using SQL syntax to make the queries different based on the `clean` parameter.

Example

So take the example from the docs:

```
<entity name="item" pk="ID"
  query="SELECT * FROM item"
  deltaImportQuery="SELECT * FROM item
    WHERE id = '${dataimporter.delta.id}'"
  deltaQuery="SELECT id FROM item
    WHERE last_modified > '${dataimporter.last_index_time}'">
```

This can be rewritten as follows:

```
<entity name="item" pk="ID"
  query="SELECT * FROM item
    WHERE '${dataimporter.request.clean}' != 'false'
      OR last_modified > '${dataimporter.last_index_time}'">
```

When doing a normal full import solr defaults the `clean` to `true`. You can make this default explicit in the solrconfig.xml handler definition, in the "defaults" section. As a result the first part of the WHERE condition will be `'true' != 'false'`, which is always true. Most database servers will use that to evaluate the entire WHERE condition to true, ignoring the second clause after the "OR", so all rows will match. `http://localhost:8983/solr/core0/dataimport?command=full-import&clean=true`

Now when doing a delta import you do not use the delta-import command, but instead you do a normal full-import but with the `clean` url parameter set to `false`: `http://localhost:8983/solr/core0/dataimport?command=full-import&clean=false` In this case the first part of the WHERE will be `'false' != 'false'` which is always false. The "OR" clause is then evaluated to decide whether the row matches.

When want to use `deletedPkQuery`, running the `delta-import` command is still necessary, which means that you will still need `deltaQuery` and `deltaImportQuery`.

Efficiency Aspect

There might be situations where separate queries might be more efficient. Consider an example where the query that fetches the document data is a very complex join and the RDBMS doesn't do a great job with coming up with a good query plan and you can determine a relatively small number of ID's that change every night with a SELECT without a JOIN, then it might be more efficient to do one query to fetch the ID's and then one query per fetched ID that does the complex join.