

Assignment 12: XQuery

- This assignment will be published on January 20th, 2017.
- You may work on the assignments in groups of up to three people (if possible, keep the same groups as in the previous assignments).
- The groups will present the solution of this assignment on February 7th, 2017 at 11:30. We will meet in seminar room 0.124.
- All members of a group have to show up together for the grading of the assignment. For the grading, each group will need to have the query expressions (XQuery) and their results ready, e.g. using OPDS. Moreover, each group member might be asked questions about the solution.
- If you have questions, send an email to holger.schwarz@ipvs.uni-stuttgart.de.

Task 1 – XQuery

The goal of this exercise is to use FLWOR expressions of XQuery for XML processing.

- To test your XPath and XQuery expressions you should use the ODPS system:
<https://odps.informatik.uni-stuttgart.de>
You need to register in this system. We will provide the password for course NBA2016 with assignment 11.
- ODPS provides access to data in `xmshop.xml` which is structured as described in `xmlshop.xsd`.

Task 1.1 – XQuery

Provide XQuery expressions for the following queries. Test your queries in ODPS.

Query A

- Provide the list of all persons in `xmlshop.xml` and for each person the number of products (books, CDs, DVDs) he or she is involved in.

Query B

- Calculate the number of products a person is involved in (on average).

Query C

- Find all persons who are actor as well as directors.

Query D

- Find all companies selling CDs only.

Query E

- Find all persons who published a CD with a German company.

Query F

- Calculate the average price for each music genre in the shop.

Query G

- Provide a list that covers all persons and the value of all products a person was involved in. Sort this list on the value of products (descending) and the last name of the person (ascending).

Query H

- Find the cheapest product.

Query I

- Find the cheapest book, CD or DVD. The type of product should be provided as variable \$type in this query.

Query J

- Find all books of persons who wrote more than one book.

Query K

- Provide the result of Query J as follows:

```
<QueryResult>
<Position>
<Title>Title 1</Title>
<Price>Price 1</Price>
</Position>
...
</QueryResult>
```

Query L

- Provide the results of Query K ordered by title (descending).

Query M

- Extend Query L in such a way that each Position element gets an ID attribute with a value which corresponds to its position in the result.

Query N

- In addition to Query M calculate the sum of product prices and add it as last child node to the QueryResult.

Query O

- Add a price reduction of 10% to all products in query N that cost more than 30 Euro.