az XQuery egy magasszintu˝ lek´erdez˝o nyelv. A XQuery mellett felhozhat´o ´erbek k¨ozu¨l ez egyik legfontosabb az

egyszeru˝ formalizmus mellett az a t´eny, hogy a W3C terveiben az XQuery

m´ar nemcsak lek´erdez˝o nyelv, hanem adatkezel˝o nyelv is (XQuery Update Facility), mely funcki´o teljes eg´esz´eben hi´anyzik az XSLT nyelvbb˝ol.

Az XQuery legfontosabb,

a´tfog´o tulajdons´agai a k¨ovetkez˝okben foglal hat´ok o¨ssze:

halmazorient´alt: az XQuery kifejez´esekben lehet˝os´eg van arra, hogy egy szimb´olummal egy csom´oponthalmazt reprezent´aljunk, s a mu˝veletet (mint p´eld´aul egy szelekci´ot) ezen halmazra ´ertelmezzu¨nk;

•

gazdag kifejez˝oer˝o: az XQuery oper´atorai, mint egy algebra oper´atorai, tetsz˝oleges egym´asba a´gyaz´ast tesznek lehet˝ov´e, itt is ´el az egym´asba a´gyazott lek´erdez´esek lehet˝os´ege;

•

XML bemeneti adat ´es XML kimeneti adat: Az XQuery, az XSLT- hez hasonl´oan XML ´allom´anyokon ´ertelmezett ´es az eredm´eny is XML form´atumu´. Egy l´enyeges elt´er´es az XSLT-hez k´epest, hogy az XQuery parancs nem a forr´as XML dokumentumba ´ep´ıtett, mint az XSLT, ahol a forr´as XML dokumentumba van megadva az XSLT transzform´aci´o megh´ıv´asa. Az Xquery eset´eben egy ku¨l¨on ´allom´anyban van az XQuery parancs, amelyben hivatkozunk kell a forr´as ´allom´anyokra;

•

a parancsok nem XML form´atumban adottak: Egy l´enyeges elt´er´es a kor´abban megismert XML kezel˝o felu¨letekkel szemben, hogy itt nem t¨orekedtek az XML form´atum k¨ovetkezetes keresztu¨lvitel´ere. Az XQuery nyelv elemei alapvet˝oen nem XML szabv´anyu´ak, hab´ar l´etezik a sz-

•

abv´anynak egy XML alapu´ v´altozata is, az XQueryX v´altozat. Az

XML form´atumt´ol val´o elt´er´es oka, hogy ez´altal l´enyegret¨or˝obb, az em- ber sz´amm´ara a´ttekinthet˝obb lett a parancsnyelv;

XPath alapu´: A XQuery nyelv is a XML dokuemntumok fa reprezent´aci´oj´at haszn´alja. Az egyes csom´oponthalmazok kiv´alaszt´asa az XPath sz- abv´anyon alapszik, amely halmazorient´alt megk¨ozel´ıt´est biztos´ıt.A XQuery az XPath funkci´ot tov´abbi mu˝veletekkel eg´esz´ıti ki.

•

Az XQuery form´atum´anak szeml´eltet´es´ere egy kis mint´at veszu¨nk. Legyen adott az al´abbi forr´as XML dokumentum, melynek azonos´ıt´o neve xx9.xml.

m

<adatbazis>

<autok>

<auto rsz="r1">

<tipus> Opel</tipus>

<ar>214 </ar>

</auto>

<auto rsz="r2">

<tipus> Fiat</tipus>

<ar>165 </ar>

</auto>

<auto rsz="r4">

<tipus> Skoda</tipus>

<ar>365 </ar>

</auto>

</autok>

<emberek>

</emberek>

</adatbazis>

A lek´erdez´esi mintafeladat a 250-n´el olcs´obb aut´ok rendsz´am´anak ´es t´ıpusa´nak ki´ırat´asa. Az XQuery rendszerben a lek´erdez´est egy ku¨l¨on dokumentumban

kell megadni. Az lek´erdez´es alakja:

for $x in doc(’xx9.xml’)/adatbazis/autok/auto where $x/ar < 250

order by $x/tipus descending return

<car> {$x/@rsz} {$x/tipus/text()}</car>

Els˝o pillanatra feltu˝nik, hogy itt nem XML form´atum szerepel. A lek´erdez´es els˝o sor´aban megadjuk, higy az xx9.xml dokumentumb´ol emelju¨k ki a megadott el´er´esi u´tvonallal (/adatbazis/autok/auto) rendelkez˝o csom´opontokat. A kiemelt csom´opontok halmaz´aban sorra veszi a feldolgoz´o azon csom´opontokat, ahol

az ´ar mez˝o kisebb, mint 250. A kapott csom´opontokat t´ıpus szerint sorba rendezi, majd minden csom´opontra el˝o´all´ıt egy eredm´enydokumentumbeli csom´opontot. A l´etrej¨ott csom´opont tagneve ’car’ ´es az aut´o rendsz´am elem- tulajdons´ag´at ´es t´ıpus elem´ert´ek´et tartalmazza. A l´etrej¨ott XML dokumen- tum tartalma:

<car rsz="r2"> Fiat

</car>

<car rsz="r1"> Opel

</car>

Mint a p´elda is mutatja, az XSLT-hez hasonl´oan itt sem garant´alt, hogy az eredm´eny dokumentum helyesen form´alt XML dokumentum legyen. A lek´erdez´est kell form´alni, hogy a k´ıv´ant szerkezetu˝ v´alasz XML dokuemntu- mot szolg´altassa.

# Az XQuery lek´erdez´esi parancsa

Az XQuery lek´erd´es formalizmusa az al´abbi logik´ara ´epu¨l. Els˝ok´ent ki- jel¨olju¨k, hogy mely csom´oponthalmazt vagy csom´opontot k´ıv´anjuk feldol- gozni,hogy abb´ol inform´aci´ot mer´ıtsu¨nk. A kijel¨ol´es sor´an egy-egy v´altoz´ot rendelhetu¨nk a halmazhoz vagy csom´oponthoz. Halmaz eset´en a megadott v´altoz´o v´egigfut a halamz ´ert´ekein ´es minden ´ert´ekn´el elv´egzi a magban megadott mu˝veleteket. A kijel¨ol´es sor´an megadhat´o egy szelekci´os kifejez´es, amely leszu˝k´ıti a feldolgozott csomopontok k¨or´et. A halmazhoz m´eg egy fel-

dolgoz´asi sorrendis´eget kijel¨olo˝ mu˝velet is tartozhat. A magban a megadjuk,

hogy milyen kimeneti atomokat ´all´ıtson el˝o az ´eppen feldogoz´as alatt ´a´al´o csom´oponthoz. A feldogloz´as magj´aban egy u´jabb lek´erdez˝o kifejez´est lehet be´agyazni.

A lek´erdez´es ´altal´anos form´atuma:

FOR $v IN kif1 LET $w := kif2 WHERE kif3 ORDER BY kif4 RETURN kif5

A FOR kulcssz´o m¨og¨otti r´eszben lehet megadni a csom´oponthalmazt. A $v szimb´olum egy v´altoz´ot jel¨ol, a v´altoz´ok neve el˝ott mind´ıg szerepel a doll´arjel. A kif1 kifejez´es egy csom´oponthalmazt takar, melyben XQuery

´es XPath elemekre ´ep´ıtve adjuk meg a halmaz defini´al´as´at. A LET kulc- ssz´o m¨og¨ott olyan v´altoz´o a´ll, mely egyetlen ´ert´eket vesz fel, em¨og¨ott nincs

´ert´ekiter´aci´o. A WHERE tag a szelekci´ot, a csom´opont szu˝r´est jel¨oli ki. Csak azon csom´opontok keru¨lnek ´at az eredm´enybe, melyekre teljesu¨l a megadott kif3 logikai kifejez´es. A RETURN r´eszben kell megadni az eredm´eny doku- mentum fel´ep´ıt´es´et. Az egyes parancstagok kezd˝obetu˝ib˝ol fel´ep´ıtetten a fenti lek´erdez´es alakot FLOWER-kifejez´esnek is nevezik.

A forr´as csom´opontok kijel¨ol´es´e t¨obbf´elek´eppen t¨ort´enhet. Lehet

Konstanssal,liter´allal. Ekkor a kifejez´esben felsoroljuk a lista elemeit. P´eldak´ent egy h´aromelemu˝ list´at adunk meg:

•

for $x in (’a’,’b’,’s’)

XML forr´as dokumentum haszn´alunk. Ehhez az a´llom´any el´er´esi u´tvonal´at kell megadni, az el´er´esi u´tvonalat egy fn:doc() fu¨ggv´eny argumentum´aban kell szerepeltetni. A dokumentum kijel¨ol´es ut´an szerepelhet egy XPath kifejez´es is, mely a dokumentumon belu¨l szu˝kiti le az ´erintett csom´opontok k¨or´et.

•

for $x in fn:doc("xx9.xml")/adatbazis/autok/auto

A statikus adatok mellett a dinamikus elemekteszik rugalmass´a a rend- szer mu˝k¨od´es´et. A dinamikus elemekn´el egy kifejez´es ki´ert´ekel´esi ´ert´eke fog a´tkeru¨lni az eredm´eny dokumentumba. Az XQuery szabv´any megengedi, hogy mind a csom´opont s´ema jellemz˝ok, mind a sz¨oveges tartalom dinamiku- san gener´al´odjon. A dinamikus kifejez´es ki´ert´ekel´est oper´atora a kapcsos z´ar´ojelp´ar:

{ kifejez´es }

A fenti mu˝velet hat´as´ara a kifejez´es ´ert´eke keru¨l az eredm´enybe, mint azt az al´abbi p´elda is mutatja. Az

RETURN

<ar> {234 + 123} </ar>

utas´ıt´as hat´as´ara az eredm´eny csom´opont alakja:

<ar> 357 </ar>

Az XQuery nyelv egyes utas´ıt´asainak r´eszletes bemutat´as´as´ahoz egy minta XML ´allom´anyt veszu¨nk (minta.xml), melynek tartalma a k¨ovetkez˝o:

<adatbazis>

<autok>

<auto rsz="r1" tulaj="1">

<tipus> Opel</tipus>

<ar>214 </ar>

<szin>kek</szin>

</auto>

<auto rsz="r2" tulaj="2">

<tipus> Fiat</tipus>

<ar>165 </ar>

<szin>piros</szin>

</auto>

<auto rsz="r4" tulaj="1">

<tipus> Skoda</tipus>

<ar>365 </ar>

<szin>feher</szin>

</auto>

</autok>

<emberek>

<ember kod ="1">

<nev>Peter</nev>

<varos>Miskolc</varos>

</ember>

<ember kod="3">

<nev>Anna</nev>

<varos>Gyongyos</varos>

</ember>

<ember kod ="2">

<nev>Zoli</nev>

<varos>Miskolc</varos>

</ember>

</emberek>

</adatbazis>

## Egyetlen csomo´pont feldolgoz´asa.

Ha forr´as dokumentumb´ol, k¨oztes csom´oponthalmazb´ol csak egyetlen csom´opontra vagy ´ert´ekre van szu¨ks´egu¨nk, akkor haszn´alhatjuk a

let $valtozo := elem\_kijel¨ol´es

utas´ıt´ast, s az eztk¨ovet˝o utas´ıt´asr´eszekben a $v´altozo szimb´olum m¨og¨ott a kijel¨olt elem fog szerepelni. P´eld´aul, az

let $x := (’a’,’b’,’s’) return <a> {$x} </a>

hat´as´ara egyetlen ¡a¿ elem keru¨l ki´ır´asra, melynek tartalma a megadott el- emharmas:

<a> a

b s

</a>

Az aut´ok adatainak ki´ırat´as´ara szolg´al´o utas´ıt´as alakja:

let $a := fn:doc("minta.xml")/adatbazis/autok return

<lista> {$a} </lista>

## T¨obb csomo´pont feldolgoz´asa.

Ha egy csom´oponthalmaz, elemhalmaz elemeivel kell dolgozni, akkor egy iter´aci´os ciklust kell kijel¨olni a

for $v´altozo in elem\_halmaz

utas´ıt´assal. Ekkor a $v´altoz´o sorba felveszi a halmazbeli ´ert´ekeket. P´eld´aul a

for $v in (’a’,’b’,’c’) return

<elem> {$v} </elem>

utas´ıt´as hat´as´ara h´arom ’elem’ csom´opont fog l´etrej¨onni:

<elem> a </elem>

<elem> b </elem>

<elem> c </elem>

A k¨ovetkez˝o p´eld´aban az aut´ok t´ıpusat fogjuk ki´ıratni:

for $a in fn:doc("minta.xml")\\auto return

<typ> {$a/text()} </typ>

Vegyu¨k ´eszre, hogy az eredm´eny megad´as´an´al nem a

return

<typ> {$a} </typ>

utas´ıt´ast adtuk ki, mert a $a szimb´olum a teljes csom´opontot jel¨oli, s ebben az esetben a typ csom´opont alatt egy teljes auto csom´opont megjelenne. A

return

<typ> {$a/text()} </typ>

esetben viszont a text() XPath fu¨ggv´ennyel a csom´opont sz¨ovegtartalm´at emelju¨k ki ´es tesszu¨k ´at az eredm´enybe, teh´at nem fog megjelenni ku¨l¨on gyerekelem-csom´opont.

## Iter´aci´ok, ´ert´ekad´asok kapcsol´asa.

Egy XQuery utas´ıt´asban t¨obb let ´es for utas´ıt´asr´esz is szerepelhet. Ekkor a bels˝o iter´aci´o a ku¨ls¨o iter´aci´o minden egyes ´ert´ek´ere lefut, teh´at vez´erl´esi sz- erekezet megk¨ozel´ıt´esb˝ol ekkor egy be´agyazott ciklus szerkezetet kapunk. Ez a mechanizmus alkalmas p´eld´aul a Descartes-szorzat megval´os´ıt´as´ara, ame- lyet a k¨ovetkez˝o p´elda is szeml´eltet:

for $x in doc(’minta.xml’)/adatbazis/autok/auto for $y in doc(’minta.xml’)/adatbazis/emberek/ember return

<a> {$x} {$y} </a>

A p´eld´aban az o¨sszes aut´o-ember csom´opontp´aros meg fog jelenni az

eredm´enyben. rendelkezik.

O¨ sszesen 9 ’a’ elem j¨on l´etre, ´es mindegyik k´et gyerekelemmel

A bels˝o ciklusban felhaszn´alhat´ok a ku¨ls˝o ciklusban defini´alt v´altoz´ok, azaz a bels˝o r´esz mind´ıg a ku¨kls˝o r´esz tov´abbi szu˝k´ıt´es´enek tekinthet˝o. Az el˝oz˝o p´eld´at a´talak´ıtva u´gy, hogy a ku¨ls˝o v´altoz´ot a bels˝o ´ert´ekad´asn´al fel- haszn´aljuk, az al´abbi parancsot kapjuk:

let $b := doc(’minta.xml’)/adatbazis for $x in $b/autok/auto

for $y in $b/emberek/ember return

<a> {$x} {$y} </a>

Az eredm´eny struktu´ra el˝oall´ıt´asakor figyelni kell arra, hogy az ´atvitelkor az elemnek a forr´asban bet¨olt¨ott szerepe is a´ta´adsra keru¨l, azaz nemcsak az

´ert´ek megy a´t. ´Igy p´eld´aul a

let $b := doc(’minta.xml’)/adatbazis for $x in $b/autok/auto

for $y in $b/emberek/ember return

<a>

<car>{$x/@rsz}{$x/tipus/text()}</car>

<owner>{$y/nev/text()}</owner>

</a>

parancsban az ’rsz’ csom´opont elemjellemz˝ok´ent fog bekeru¨lni a ’car’ csom´opont al´a, mivel a forr´asban is elemjellemz˝o szerepet t¨olt¨ott be.

## Szelekcio´ elv´egz´ese.

A feldolgozott elemekb˝ol valo´ v´alogat´ast, szu˝r´est k´etf´ele m´odon is megold-

hatjuk. Egyr´eszt a kijel¨ol˝o XPath kifejez´esbe ´ep´ıtu¨nk be szelekci´os r´eszt, vagy az XQuery FLOWER parancsban szerepeltetu¨nk egy

WHERE felt´etel

r´eszt. Ekkor csak a felt´etelnek elehet t´ev˝o elemek keru¨lnek feldogoz´asra a parancs belsej´eben. A k¨ovetkez˝o p´eld´aban a 222-n´el dr´ag´abb aut´ok t´ıpus´at

´es rendsz´am´at ´ıratjuk ki. A rendsz´am tov´abbra is elemjellemz˝o marad.

for $x in doc(’minta.xml’)/adatbazis/autok/auto where $x/ar > 222

return

<car>{$x/@rsz}{$x/tipus/text()}</car>

A fenti mechanizmusra ´ep´ıtve m´ar szelekci´os join mu˝veletet is v´egre tudunk hajtani. Feladatk´ent az o¨sszetartoz´o aut´o-ember p´arosokat k´erdezzu¨k le:

for $a in fn:doc(’minta.xml’)//auto for $e in fn:doc(’minta.xml’)//ember where $a/@tulaj eq $e/@kod

return

<a>

<car>{$x/@rsz}{$x/tipus/text()}</car>

<owner>{$y/nev/text()}</owner>

</a>

A szelekci´os felt´etelben skal´ar´ert´ek-alapu´ o¨sszehasonl´ıt´ast v´egeztu¨nk.

## Elemek rendez´ese.

Iter´aci´o eset´en l´enyeges lehet a halmazelemek megjelen´esi sorrendje is. A rendez´es

order by kifejez´es m´od

utas´ıt´as´aval megadhatjuk, hogy a halmaz elemeit mely kifejez´es szerint rendezze, s a m´oddal rendez´es n¨ovekv˝o (ascending) csak cs¨okken˝o (descend- ing) jelleg´et lehet be´all´ıtani. A p´eld´aban a 222-n´el olcs´obb aut´ok adatait

´ıratjukki t´ıpusn´ev szerint cs¨okken˝o sorrendben:

for $x in doc(’minta.xml’)/adatbazis/autok/auto where $x/ar < 222

order by $x/tipus descending return

<car> {$x/@rsz} {$x/tipus/text()} </car>

Mivel az XQuery k¨ornyezetben a sorrendis´eg alapfogalom, az order by parancsot a bels˝o, be´agyazott parancsokban is lehet haszn´alni.

## Csom´opontok l´etrehoz´asa.

A dinamikus lehet˝os´egek k¨ozott szerepel azeredm´eny szerkezet dinamikus kialak´ıt´asa is. Lehet˝os´eg van arra, hogy az eredm´eny dokumentum csom´opontjait l´eterhoz´o parancsokat haszn´aljunk, melyben a csom´opont param´eteri, mint p´eld´aul az azonos´ıt´o neve, kifejez´esk´ent adhat´o meg. Az XQuuery rendszer legfontosabb csom´opont l´etrehoz´o utas´ıt´asai:

return

{element {nev} {ertek}}

{attribute {nev}{ertek}}

{text {ertek}}

{processing-instruction {nev} {ertek}}

{comment {sz¨oveg}}

Az element utas´ıt´as elemcsom´opontot gener´al, az attribute elemjellemz˝ot, m´eg a text egy sz¨ovegcsom´opontot hoz l´etre. A feldolgoz´o utas´ıt´acc csom´opontj´at a processing-instruction parancs, m´ıg a megjegyz´es csom´opont´et a comment parancs v´altja ki. A parancsokban mind az azonos´ıt´o n´ev, mind a tartalom dinamikusan adhat´o meg. Az el˝oz˝o aut´o kilist´az´o parancsot most dinamikus elemkijel¨ol´essel oldjuk meg:

for $x in doc(’minta.xml’)/adatbazis/autok/auto where $x/ar > 222

return

{element car {text{$x/@rsz}}}

A k¨ovetkez˝o p´eld´aban a kapcsol´od´o aut´ok ´es meberek adatait is dinamikus szerkezetfekl´ep´ıt´essel adjuk meg:

for $a in fn:doc(’minta.xml’)//auto for $e in fn:doc(’minta.xml’)//ember where $a/@tulaj eq $e/@kod

return

element eredmeny { element auto {$a}, element tulaj {$e}

}

Ezt a dinamikus csom´opont fel´ep´ıt´est haszn´alhtjuk ki arra, hogy a forr´as elemjellemz˝oj´eb˝ol az eredm´eny elem´ebe vigyu¨nk ´at adatot. Az al´abbi p´elda a rendsz´amot viszi ´at elemjellemz˝ob˝ol sz¨ovegcsom´opontba.

for $a in fn:doc(’minta.xml’)//auto return

element auto {text{$a/@rsz}}

Egy m´eg nagyobb l´ept´ek˝o ´atrendez´est ad meg a k¨ovetkez˝o feladat: Irassuk ki az aut´okat u´gy, hogy az elem neve a rendsz´am ´ert´ek legyen, sz¨ovegtartalma a t´ıpus´ert´ek, ´es az a´r ´ert´ek elemjellemz˝obe megy a´t. Az a´talak´ıt´as parancsa:

for $a in fn:doc(’minta.xml’)//auto return

element {$a/@rsz} {attribute price {$a/ar} , text{$a/tipus}}

Mind m´ar azt a bevezet˝oben eml´ıtettu¨k, az XQuery rendszer nem fogja automatikusan gener´alni a helyesen form´alts´agot az eredm´eny dokumen- tumokn´al. A kor´abbi p´eld´aink egyik jelent˝os hib´aja, hogy nincs gy¨okere a dokumentumnak. Ha egy olyan eredm´enyt szeretn´enk el˝o´all´ıtani, melyben l´etezik ilyen gy¨ok´erelem, akkor azt a lek´erdez´esbe is be kell ´ep´ıteni. Mivel a gy¨ok´erelem meg´aba foglalja a t¨obbi elemet, a lek´erdez´est le´ır´o parancsot ebbe a gy¨ok´erelembe foglaljuk be az al´abbi m´odon:

<gy¨ok´er>

FLOWER lek´erdez´es

</gy¨ok´er>

P´eldak´ent ´ırassuk ki az aut´ok t´ıpusait u´gy, hogy a kapott dokumentum tartalmazzon gy¨ok´er elemet:

<adatok>

{

for $a in fn:doc(’minta.xml’)//auto return

element car {$a/tipus}

}

</adatok>

A p´eld´aban a gy¨ok´erelem elnevez´ese ’adatok’.

## Felt´eteles parancsv´egrehajt´as.

Ha a kimenti ´allom´any egyes r´eszei felt´etelt˝ol fu¨gg˝oen v´altoznak, akkor a felt´eteles ki´ert´ekel´est megval´os´ıt´o paranccsal ´ep´ıtju¨k fel a kimeneti dokumen- tumot. A parancs form´atuma:

return

{

if (kifejezes1) then kifejezes2 else kifejezes3

}

A kifejez´es1 igaz ´ert´eke eset´en az eredm´eny a kiefejez´es2 helyettes´ıt´esi

´ert´eke lesz, m´ıg hamis ´ert´ek eset´en a kifejez´es3 ´ert´ek´et kapjuk vissza. A k¨ovetkez˝o p´eld´aban a sz´am´ert´ekkel megadott ´ar ´ert´eket egy sz¨oveges le´ır´assal eg´esz´ıtju¨k ki, megadva, hogy az ´arat kev´esnek vagy soknak tal´aljuk-e:

for $x in doc(’minta.xml’)/adatbazis/autok/auto return

<car>

{$x/@rsz} {$x/tipus/text()} {$x/ar}

{if ($x/ar>151) then ’sok’ else ’keves’}

</car>

M´asodik kapcsol´od´o p´eld´ankban az ha az doll´arban lenne megadva:

a´r ´ert´ek´et kell forintra konvert´alni,

<adatok>

{

for $a in fn:doc(’minta.xml’)//auto return

element car { text {$a/tipus},

element ar {if ($a/ar/@valuta eq ’USD’) then text {250\*$a/ar} else text {$a/ar}}

}

</adatok>

A t¨obbsz¨or¨os el´agaz´ast csak az if szerkezetek egym´asba a´gyaz´as´aval lehet megoldani:

if (felt´etel1a) then felt´etel2a

else if (felt´etel1b) then fet´etel2b else ...

## Aggreg´aci´os fu¨ggv´enyek.

A lek´erdez´esek egyik tipikus k¨ore, amikor ¨osszes´ıt˝o ´ert´ekeket kell el˝o´all´ıtani a r´eszletez˝o adatokb´ol. A XQuery egyik hi´anyoss´aga, hogy nincs ilyen k¨ozvetlen csoportk´epz˝o oper´ator, mint a GROUP BY parancs az SQL nyelvben. Itt csak keru¨l˝ou´ton lehet megoldani a csoportos´ıt´as probl´em´aj´at. A csopor- tos´ıt´ashoz egy

distinct-values()

fu¨gg´eny haszn´alhat´o, amely az argumentum´aban megadott ´ert´ekhalmazb´ol egy olyan ´ert´ekhalmazt ´all´ıt el˝o, melyben nem fordul el˝o elemism´etl˝od´es.

P´eldak´ent ´ırassuk ki az emberek v´arosait, u´gy hogy ne forduljon el˝o benne ism´etl˝od´es:

<varosok>

{

for $v in fn:distinct-values( fn:doc(’minta.xml’)//ember/varos ) return

<city>{$v}</city>

}

</varosok>

A m´asik fontos mu˝velet az aggreg´alt ´ert´ekek lek´erdez´ese. Ehhez egy sor aggreg´aci´os fu¨ggv´eny ´all rendelkez´esre. A XQuery aggreg´aci´os fu¨ggv´enykezel´es´enek saj´atoss´aga, hogy itt a fu¨ggv´eny argumentum´aban explicite ki kell jel¨olni a feldolgozand´o halmazt. A defini´alt fontosabb aggreg´aci´os fu¨ggv´enyek:

* max(): maximum ´ert´ek
* min(): minimum ´ert´ek
* sum():o¨sszeg
* count(): darabsz´am
* avg():´atlag
* A fu¨ggv´enyek haszn´alat´ara vegyu¨k az al´abbi p´eld´at: aut´ot´ıpusonk´ent iras- suk ki a hoz´a tartoz´o aut´okat:
* <adatok>

{

for $t in fn:distinct-values( fn:doc(’minta.xml’)//auto/tipus ) return

element tipus {attribute tip {$t}, element autok {

for $a in fn:doc(’minta.xml’)//auto where $a/tipus eq $t

return

element auto {$a/@rsz}

}

}

}

</adatok>

A p´eld´aban a be´agyazott lek´edez´es is megfigyelhet˝o, hiszen a ku¨ls˝o lek´erdez´es RETURN r´esz´eben egy m´asik FLOWER lek´erdez´es foglal helyet. A bels˝o lek´erdez´esben felhaszn´aljuk a ku¨ls˝o lek´erdez´esben l´etrehozott $t v´altoz´ot, il- letve egy saj´at fu¨ggetlen v´altoz´ot is, az $a-t is defini´aljuk.

Ha a lek´erdez´esben az aut´ok t´eteles felsorol´asa helyett csak a darabsz´amukat ig´enyelju¨k, akkor a lek´erdez´es alakja a k¨ovetkez˝ore v´altozik:

<adatok>

{

for $t in fn:distinct-values( fn:doc(’minta.xml’)//auto/tipus ) return

element tipus {attribute tip {$t}, attribute db {count(

for $a in fn:doc(’minta.xml’)//auto where $a/tipus eq $t

return

element auto {$a/@rsz}

)

}

}

}

</adatok>

Ebben az esetben a darabsz´amot a count() fu¨ggv´ennyel k´erdezzu¨k le, melynek argumentum´aba megadjuk a megsz´aml´aland´o csom´oponthalmazt. A csom´oponthalmazt most egy u´jabb FLOWER lek´erdez´essel ´all´ıtjuk el˝o.

## Saj´at fu¨ggv´enyek defin´al´asa.

Az XSLT nyelvhez hasonl´oan az XQuery nyelv is t´amogatja az egyedi fu¨ggv´enyek l´etrehoz´as´at. Azonban itt is telejsu¨l az a keretfelt´etel, hogy a k´odot a funkcion´alis programoz´as elvei ment´en kell k´esz´ıteni, teh´at is sem a hagyom´anyos pro-

gramoz´asi nyelvekben megismert vez´erl´esi szerkezetek ´es v´altoz´okezel´es a´ll

rendelkez´esre. A v´altoz´ok itt is csak egyszer vehetnek fel ´ert´eket ´es a ciklu- sok helyett rekurz´ıv fu¨ggv´enyhiv´asokkal lehet megoldani az iter´aci´okat.

A fu¨ggv´enyek l´etrehoz´as´ahoz itt is szu¨ks´eg van saj´at n´evt´er kijel¨ol´esre. A n´evt´er ´es fu¨ggv´eny l´etrehoz´as keretutas´ıt´asai:

declare namespace prefix=kifejezes;

declare function prefix:fnev ($p1 as t1,..) as rtip

{

utasitasok

return kifejezes

};

A return utas´ıt´as szolg´al a fu¨ggv´eny visszat´er´esi ´ert´eknek megad´as´ara. A k¨ovetkez˝o p´eld´aban egy olyan fu¨ggv´enyt k´esz´ıtu¨nk, mely egy decim´alis, tizessz´amrendszerbeli sz´amot vesz a´t az argumentum´aban ´es annak a fel´et adja vissza.

declare namespace ll="http:me.kl";

declare function ll:felez($x as xs:decimal) as xs:decimal

{

let $c:=2

return $x div $c

};

for $x in doc(’minta.xml’)/adatbazis/autok/auto where $x/ar < 222

order by $x/tipus descending return

<car> {$x/@rsz} {$x/tipus/text()} {$x/ar} {ll:felez($x/ar)}</car>

A mint´aban a fu¨ggv´eny megh´ıv´asa´t is l´athatjuk. A kiejez´es ki´ert´ekel´es oper´ator´an belu¨l haszn´alhatjuk, ahol megadjuk a h´ıv´as aktu´alis param´etereit

is. P´eld´ankban az aktu´alis aut´o csom´opont a´r gyerekelem´enek az ´ert´eke

keru¨l a´t a felez fu¨ggv´enyhez. A param´eter´atad´as sor´an egy automatikus

t´ıpuskonverzi´ot is v´egzett a rendszer.

Egy iter´aci´ot is tartalmaz´o mintak´ent vegyu¨k azt a feladatot, amikor az

els˝o n sz´am ¨osszeg´et kell o¨sszeadni. Ugyan tudjuk, hogy ez z´art alakban

is megadhat´o, de most a gyalogm´oszerrel elv´egezzu¨k az elemek egyenk´enti o¨sszead´as´at:

declare namespace ll="http:me.kl";

declare function ll:ossz($r as xs:integer) as xs:integer

{

let $y := 1 return (

if ($r > 0) then

$r + ll:ossz($r -1) else

0

)

};

let $x := 5 return

<ered> {ll:ossz($x)}</ered>

Az XQuery Update nyelv feldolgoz´asi modellj´et illet? legfontosabb v´altoz´as, hogy a v´egrehajt´o motorban a kifejez´esek eredm´enye nemcsak XDM (vagy In- foset) kimeneti atom lehet, hanem egy m´odos´ıt´ast le´ır´o inform´aci´olista lista is (Update List). A m´odos´ıt´asi lista c´elja megadni, hogy milyen m´odos´ıt´asokat kell majd elv´egezni az egyes csom´opontokn´al. A m´odos´ıt´asi lista elemei az u´gynevezett m´odos´ıt´asi atomok (Update Primitives), melyek mindegyike egy o¨n´alloan elv´egezhet? m´odos´ıt´ast ´ır le. A m´odos´ıt´asi atom k´et komponense:

* C´elcsom´opont, amin el kell v´egezni a m´odos´ıt´ast
* M?velet, az elv´egzend? tev´ekenys´egatom

A tev´ekenys´egatom egy bels?leg ´ertelmezett a´talak´ıt´ast ad meg, a fel-

haszn´al´o adatm´odos´ıt´ast el?´ır´o m?veletei ezen tev´ekenys´egatomokra ´epu¨lnek. A XQuery modell az al´abbi tev´ekenys´egatomokat defini´alja:

* insertBefore: csom´opont beszu´r´asa egy m´asik csom´opont el´e
* insertAfter: csom´opont beszu´r´asa egy m´asik csom´opont m¨og´e
* insertInto: csom´opont beszu´r´asa egy m´asik gyerekei k¨oz´e
* insertIntoAsFirst: csom´opont beszu´r´asa egy m´asik els? gyerekek´ent
* insertIntoAsLast: csom´opont beszu´r´asa egy m´asik ut´ols´o gyerekek´ent
* insertAttributes: elemjellemz? beszu´r´asa
* delete: csom´opont t¨orl´ese
* replaceNode: csom´opont helyettes´ıt´ese egy m´asik csom´oponttal
* replaceValue: csom´opont sz¨oveg´ert´ek´enek helyettes´ıt´ese
* replaceElementContent: csom´opont tartalm´anak helyettes´ıt´ese
* rename: csom´opont nev´enek megv´altoztat´asa

A fenti atomokb´ol o¨ssztett felhaszn´al´o parancsokat k´epeztek, melyek m´ar kiadhat´ok az XQuery k¨ornyezetben. A m´odos´ıt´ashoz kapcsol´od´o m?veleteknek o¨t t´ıpusa van:

* insert (beszu´r´as, b?v´ıt´es)
* delete (t¨orl´es)
* replace (helyettes´ıt´es)
* rename (´atnevez´es)
* transform (´atalak´ıt´as)
* A csom´opontfelvitelnek h´arom fontosabb szintakszis valtozata van. A gyerekcsom´opont felvitel´enek alakja:

insert nodes forras\_csomopont as (first|last) into cel\_csomopont

P´eldak´ent vigyu¨nk fel egy u´j aut´o le´ır´o elemet:

insert nodes <auto rsz="r5"> <tipus> Opel</tipus> <ar>365 </ar> </auto> as last into fn:doc("minta.xml")/adatbazis/autok

Az elem el´e t¨ort´en? beszu´r´as:

insert node forras\_csomopont before cel\_csomopont

N´ezzu¨k meg, hogyan lehet minden aut´o le´ır´o csom´opontot kieg´esz´ıteni egy evjarat elemmel:

for $v in fn:doc("minta.xml")//ar return

insert nodes <evjarat> 1999</evjarat> after $v

Az elem m¨og´e t¨ort´en? beszu´r´as:

insert node forras\_csomopont after cel\_csomopont

Elem t¨orl´ese:

delete node cel\_csomopont

P´eldak´ent t¨or¨olju¨k ki az Opel aut´ok arat le´ır´o elem´et:

delete node fn:doc("minta.xml")//auto[tipus = "Opel"]/ar

Elem cser´eje:

replace node cel\_csomopont with forras\_csomopont

Elem sz¨oveges tartalm´anak cser´eje:

replace value of node cel\_csomopont with forras\_csomopont

A p´eld´aban megn¨ovelju¨k a Skoda aut´ok ´art´a 10

replace value of node fn:doc("minta.xml")//auto[tipus="Skoda"]/ar with fn:doc("minta.xml")//auto[tipus="Skoda"]/ar\*1.1

Elem nev´enek m´odos´ıt´asa:

rename node cel\_csomopont as nev

Elemek ´atm´asol´asa m´odos´ıt´assal:

copy $v := cel\_csomopont modify modosito\_kifejezes return eredmeny

Az al´abbi p´eld´aban ki´ıratjuk az aut´ok adatait az ar elem n´elku¨l:

for $a in fn:doc("minta.xml")//auto return

copy $v := $a

modify delete node $v/ar return $v

P´eldak´ent m´eg n´eh´any tov´abbi feladatot vessu¨nk sorra.

1. feladat. A Peter nev? emberhez egy u´j Skoda aut´ot felvitele:

for $a in fn:doc("minta.xml")//ember[nev="Peter"]/@kod return

insert node

<auto rsz+"r23">

{attribute tulaj {$a}}

<tipus>Skoda</tipus>

</auto>

into fn:doc("minta.doc")/adatbazis/autok

1. feladat. Peter aut´oinak ´ar´at az eddigi maximum ´ar + 1 ´ert´ekre m´odos´ıtjuk

let $m := max(fn:doc("minta.xml")//auto/ar) return

replace value of node fn:doc("minta.xml")//ember[nev="Peter"]/ar with {$m+1}

1. feladat. Az a´tlag´arnal olcso´bb aut´ok a´r´anak megn¨ovel´ese 10

let $autok := max(fn:doc("minta.xml")/adatbazis/autok let $r := avg($autok/auto/ar)

for $a in $autok/auto/ar where $a le $r

return

replace value of node $a with {data($a)+1}

1. feladat. Azon emberek kit¨orl´ese, akiknek nincs aut´ojuk:

let $aa := fn:doc("minta.xml")/adatbazis for $e in $aa//ember

let $c := count($aa//auto[@tulaj = $e/kod]) where $c eq 0

return

delete node $e