

DIPLOMAMUNKA

Szűcs Béla

**Debrecen
2009**

Debreceni Egyetem

Informatika Kar

XML ALAPÚ SZOLGÁLTATÁSOK

Témavezető:

Dr. Adamkó Attila
egyetemi adjunktus

Készítette:

Szűcs Béla
programtervező-informatikus

Debrecen
2009

Tartalomjegyzék

Tartalomjegyzék	2
1. Bevezetés az XML-be	3
1.1 Az XML szintaktikai szabályai.....	4
2. EBEAD	6
2.1 Fizetési-mérleg és adatszolgáltatások elektronikus rendszere	6
2.1 Séma leíró módszerek	9
3. A DTD (Document Type Definition)	10
3.1 Elemtípusok deklarációja.....	10
3.2 Attribútumok	13
4. XML séma	14
4.1 Egyszerű típus	16
4.2 Összetett típus	19
4.3 XML és az Excel kapcsolata	21
5. XPath.....	28
5.1 Elérési-út (location path)	28
5.2 Lépések (location steps)	31
5.3 Irányok	31
5.4 Típusok.....	32
5.5 Feltételes kifejezés.....	33
5.6 Függvények	34
6. XQuery	38
6.1 XQuery alapok	38
6.2 Kifejezések	40
7. Összefoglalás	43
8. Irodalomjegyzék	45
9. Függelék.....	47

1. Bevezetés az XML-be

Az XML rövidítés az Extensible Markup Language kifejezésből származik, ami magyarra Bővíthető Jelölő Nyelvként fordítható. Egyik érdekessége, hogy a W3C (Word Wide Web Consortium) fejlesztette ki, így egyetlen nagy szoftverfejlesztő cégnek sem tartozik a fennhatósága alá, ami megakadályozza, hogy egyplatformossá váljon. Az XML-ben írt fájlok szerkezete ugyanúgy szöveges alapú, mint a HTML fájloké, maga a nyelvezet is hasonlít hozzá. Ez már csak azért sem véletlen, mert HTML és SGML nyelv jelentette az alapját. Annak ellenére, hogy a 2000 - 2001 környékén kezdett rohamosan terjedni nem új dologról van szó, mert a nyelv alapjai a 60-as években jöttek létre, de csak az SGML használatával kezdett ismertté válni. Hivatalosan XML néven 1998-ban jelent meg az 1.0-ás verzió. Az SGML nyelv részhalmazának tekinthető, kihasználva annak számos előnyös tulajdonságát, de nélkülözve a bonyolultságát.

Strukturált adatformátumú szövegek leírására alkalmas nyelvezet. Szöveges volta miatt platform független, mint a HTML - Windows, Linux, OS/2 és egyéb operációs rendszerek alatt egyaránt használható. Strukturált felépítése és az operációs rendszerektől való függetlensége lehetőséget biztosít a különböző rendszerek közötti adatátvitel megvalósítására. A tisztán szöveges kódnak köszönhetően méretük általában kisebb, mint a bináris fájloké és egy szövegszerkesztővel - akár a "Jegyzetömbbel" ("Notepad") is módosíthatók.

Napjainkban a szöveg- és adattárolás megannyi formátuma áll rendelkezésre, kezdve a TXT-től, az RTF-en át a DOC vagy XLS fájlokig. A rengeteg szabvány nagyon megnehezíti a keresőalkalmazások dolgát. Az XML-ben való keresés egyrészt azért könnyű, mert kódolatlan szöveges állományról van szó, másrészt meg van jelölve, hogy melyik része szöveg, adat, kép, stb., (ezért jelölőnyelv) így elegendő csak a szövegrészben keresni. Tehát különválasztja az információt és az információ ábrázolását. A különböző megjelenítési formátumokra vonatkozóan nincsenek megkötések, a nyelv önleíró. Mindig a célnak legmegfelelőbb formátum nyerhető ki belőle, ami magyarázatot jelent a széleskörű elterjedésre (használható web böngészőkben, mobiltelefonokban, adattárolásra, de nyomtatható információ is kinyerhető). Hordozhatóságának köszönhetően az XML vált az egyik legnépszerűbb adatközvetítő technológiává az adatbázisok és a felhasználók asztali gépe között.

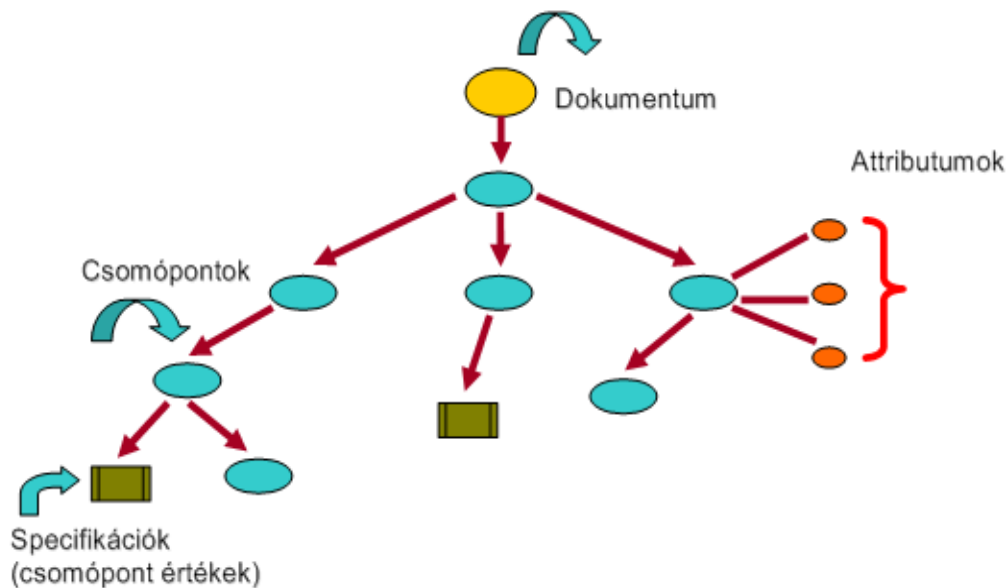
XML felhasználási területeire:

- adatok tárolása (adatbázis funkció) - mivel egyszerű szöveges állomány világos szerkezettel, ezért könnyedén feldolgozható, vagy olvasható.
- az XML dokumentumok faszerkezetű struktúrája bármely (logikusan) csoportosított adat tárolását lehetővé teszi, ami stíluslap használatával könnyen megjeleníthetővé válik, ezért alkalmas bármilyen adat tárolására.
- egyedi dokumentum-típus definíció létrehozásával, saját jelölőnyelv alakítható ki:
 - WML - wap oldalakat leíró jelölőnyelv
 - VML - vektorgrafika leírására létrehozott jelölőnyelv
 - OFX - pénzügyi információk leírására létrehozott jelölőnyelv
 - NML - internetes hírcsere

1.1 Az XML szintaktikai szabályai

Az XML-ben az adatokat elemek (TAG-ek) jelölik. Az elemeket az XML (A HTML-hez hasonlóan) kisebb-nagyobb jelekkel ("**<**", "**>**"), mint határoló karakterekkel jelöli meg, és a határoló elemek közötti szó - címke -írja le magát az elemet:

```
<name> ... </name>
```



1. ábra: XML adattípus modell

Az XML dokumentumban található adatok egy fa struktúrát alkotnak. Minden dokumentum egy gyökér elemből indul. Az XML dokumentumban a gyökér elem a legmagasabb szintű elem, az összes többi elemet a gyökér elembe kell beágyazni. A gyökérelem belsejében az egyes XML ágak (node) már tetszőlegesen ismétlődhetnek! Minden elemnek kell, hogy tartalmazzon nyitó- és záró jelölőelemet, tehát minden megnyitott elemet be kell zárni!

Egy XML dokumentum tetszőleges elemeket és tulajdonságokat tartalmazhat, valamint tetszőleges felépítéssel rendelkezhet. Mindezek a paraméterek meghatározzák a dokumentum típusát, amelyet leírva a dokumentum típus definícióját, vagy sémáját kapjuk. Egy dokumentum típus definíciója számos módon felhasználható. Felhasználható több, különböző forrásból származó XML dokumentum „kompatibilitásának” a vizsgálatára (hogy vajon a dokumentumok azonos típusúak, így ugyanabban az alkalmazásban egyaránt felhasználhatóak-e). Felhasználható annak meghatározására, hogy egy készítő dokumentumnak milyen típusal kell rendelkeznie, milyen konvenciókat kell betartania, hogy egy adott alkalmazásban felhasználható legyen, valamint a típus definíció alapján vizsgálható egy dokumentum érvényessége.

2. EBEAD

2.1 Fizetési-mérleg és adatszolgáltatások elektronikus rendszere

Az előző fejezetben leírtak ismeretében, úgy gondolom az XML lesz a legalkalmasabb a fájl típusok közül a lent ismertetett feladat megoldására.

Azoknak a gazdasági szervezeteknek kell jelenteniük a Magyar Nemzeti Banknak fizetési mérleg összeállításához, akikre a 3/2007. (II.21.) MNB rendeletben meghatározott feltételek vonatkoznak.

A fizetésimérleg-statisztikák összeállításához szükséges információk:

Azonosító kód	Megnevezés	Az adatszolgáltatók köre	Az adatszolgáltatás		
			gyakorisága	teljesítésének módja	határideje
R15	Egyéb befektetések negyedéves adatszolgáltatása - nem pénzügyi vállalatok, biztosítók és nyugdíjpénztárak, valamint háztartásokat segítő nonprofit intézmények	I. A. pontja szerinti A) Nem pénzügyi vállalatok, F) Biztosítók és nyugdíjpénztárak, valamint K) Háztartásokat segítő nonprofit intézmények szektorába sorolt – gazdasági szervezetek, amelyek esetében a tárgynegyedév első vagy utolsó napján az adatszolgáltatásban szereplő összes követelés vagy összes tartozás állománya eléri a 250 millió Ft-ot	negyedéves	elektronikus úton	tárgyidőszakot követő hónap 10. munkanap
R19	Nem pénzügyi vállalatok tájékoztató mérlegadatai	a jegybanki információs rendszerhez szolgáltatandó információk köréről, a szolgáltatás módjáról és határidejéről szóló MNB rendeletben elrendelt F97 adatszolgáltatásra ki nem jelölt, a jelen melléklet szerinti valamely más adatszolgáltatás teljesítésére kötelezett – az e rendelet 2. mellékletének I. A.	negyedéves	elektronikus úton	tárgyidőszakot követő hónap utolsó munkanap

		pontja szerinti A) Nem pénzügyi vállalatok sorolt szervezetek	szektorába –gazdasági		
--	--	---	-----------------------	--	--

Tekintettel arra, hogy az EBEAD minden jegybanknak adatot szolgáltató gazdálkodó szervezet számára kötelezővé válik 2008-tól a Magyar Nemzeti Bank elkészítette a fizetési-mérleg és adatszolgáltatások elektronikus benyújtását támogató számítástechnikai rendszerét.

MAGYAR NEMZETI BANK

EBEAD jelentésnapló

Szűrési feltételek

Feltöltés ideje	Adatszolgáltató	AGY. kód	Gyakoriság	Vonatkozási idő vége	Státusz
-tól			- Mind -	-tól	- Mind -
-ig				-ig	

Frissítés

Feltöltés ideje	Fájlnév	Adatszolgáltató	AGY. kód	Gy.	Von. idő vége	Státusz	Részletek	Kiv.
2009.11.03 13:31:34	R159N310668586	10668586	R15	NÉ	2009.09.30	Elküldve	Részletek	<input type="checkbox"/>
2009.11.03 10:22:19	R199N310668586	10668586	R19	NÉ	2009.09.30	Elküldve	Részletek	<input type="checkbox"/>
2009.07.30 09:59:02	R199N210668586	10668586	R19	NÉ	2009.06.30	Elküldve	Részletek	<input type="checkbox"/>
2009.07.29 15:45:16	R159N210668586	10668586	R15	NÉ	2009.06.30	Elküldve	Részletek	<input type="checkbox"/>
2009.05.20 14:24:13	R159N110668586	10668586	R15	NÉ	2009.03.31	Elküldve	Részletek	<input type="checkbox"/>
2009.05.20 14:19:24	R158N410668586	10668586	R15	NÉ	2008.12.31	Elküldve	Részletek	<input type="checkbox"/>
2009.05.20 14:18:50	R158N310668586	10668586	R15	NÉ	2008.09.30	Elküldve	Részletek	<input type="checkbox"/>
2009.05.20 13:44:26	R158N210668586	10668586	R15	NÉ	2008.06.30	Elküldve	Részletek	<input type="checkbox"/>
2009.05.20 13:36:08	R158N110668586	10668586	R15	NÉ	2008.03.31	Elküldve	Részletek	<input type="checkbox"/>
2009.05.06 10:08:55	R199N110668586	10668586	R19	NÉ	2009.03.31	Elküldve	Részletek	<input type="checkbox"/>

Kiválasztott jelentések beküld... Kiválasztott jelentések törlése E-mailban értesítés

2. ábra: EBEAD az MNB adatszolgáltatási elektronikus rendszere

Az EBEAD-dal az MNB-nek az a célja, hogy hatékony támogatást nyújtson adatszolgáltatóknak a jelentések elkészítéséhez, beküldéséhez, ellenőrzéséhez, továbbá hogy segítse a jegybank és az adatszolgáltatók közötti kommunikációt. Ennek érdekében kifejlesztett egy olyan alkalmazást, amely minden adatszolgáltató számára alacsony költségen

elérhető, megfelelő adatbiztonságot nyújt, platform független és jól automatizált. Az MNB szándéka az is, hogy az adatszolgáltatók széles köre számára biztosított legyen az elektronikus adatszolgáltatás, hiszen mindkét fél számára ez jelent optimális megoldást.

Az EBEAD többféle szerkezetű jelentésfájl fogadására képes. A beérkezett jelentésfájlról az EBEAD érkeztető modulja értelmezi, hogy a jelentés mely szerkezetben érkezett, és annak megfelelően dolgozza fel. Ezzel lehetőség van a jelenleg előírt szerkezetű jelentésfájlok beküldésére és így az új adatszolgáltatási előírásokra történő fokozatos átállásra, de az új adatküldési lehetőségek akár azonnali kihasználására is.

Az EBEAD által kezelt jelentésfájl szerkezetek:

- a) XML formátumú jelentésfájl, MNB által előírt fájlszerkezet
- b) CSV formátumú, PSZÁF által előírt fájlszerkezet

A EBEAD rendszer által fogadott XML jelentésfájlok kétfázisú ellenőrzésen mennek át. Az első fázis elsődlegesen az XML fájl formai ellenőrzését W3C XML-XSD séma szabvány, <http://www.w3.org/XML/Schema#dev> alapján. A második fázisban az XSD fájl által ki nem fejezett további formai és tartalmi szabályok kerülnek ellenőrzésre. Csak azon jelentések kerülnek elfogadásra, melyek mindkét ellenőrzésnek megfelelnek.

Diplomamunkám célja ezen jelentési kötelezettséghez elkészítéséhez megfelelő környezet kialakításának bemutatása, szükséges XML fájlok elkészítése, amelyen keresztül betekintés nyújtok az XML alapvető szolgáltatásaiba, alkalmazásukba.

2.1 Séma leíró módszerek

Az EBAD-hoz történő Jelentések elkészítéséhez, majd a feltöltéshez mindenképpen szükséges az alábbi séma leíró módszerek:

- 1) A jelenleg egyik legismertebb a **DTD** (Document Type Definition). Az XML fájlban szereplő adatok típusát és viselkedését határozza meg. Szabályok összessége, melyek az XML dokumentum teljes szerkezetét leírják. Legfontosabb előnye, hogy szabványos, valamint több XML dokumentumhoz is felhasználható, ami nem elhanyagolható, figyelembe véve a sokszor időigényes elkészítését. Hátránya, hogy nem XML alapú, nehéz megtanulni, és bonyolult, ráadásul mindezek miatt a jövőben valószínűleg el fog tűnni.
- 2) Az első elkészült XML alapú séma leírási módszer az **XDR**. (XML alapú annyit tesz - maga a leírás is egy XML dokumentum, a nyelvi elemek XML elemek is egyben.). A Microsoft fejlesztette ki, mindközül a legkönnyebben megtanulható, végleges, a Microsoft eszközei széles körűen támogatják, ugyanakkor nem szabványos, más vállalatok termékei csak elvétve ismerik.
- 3) Az **XSD** (XML Schema Definition) vagy XML séma szabványos, széles körűen támogatott, általános séma leíró nyelv. Vélhetően hamarosan ez lesz az XML-hez hivatalosan ajánlott dokumentum típus leíró nyelv.

3. A DTD (Document Type Definition)

Érvényes XML dokumentumnak nevezzük azon jól formázott XML dokumentumokat, melyek logikai felépítése és tartalma teljes mértékben megegyezik az XML dokumentumban meghatározott (vagy külső fájlban meghatározott és az XML dokumentumhoz csatolt) szabályoknak. Ezen szabályok megfogalmazhatóak (megírhatóak) **Dokumentum Típus Definíció** segítségével. A DTD segítségével biztosítva van, hogy a megírt vagy feldolgozni kívánt dokumentum megfelel az elvártaknak. Ráadásul az egyes DTD-k ismerete alapján bárki készíthet érvényes dokumentumot vagy akár feldolgozó alkalmazást, vagyis a DTD nem más mint egy tervrajz az XML dokumentumokhoz. A DTD-nek tartalmazni kell az összes elemet és jellemzőt, amelyet a dokumentum tartalmazhat.

3.1 Elemtípusok deklarációja

Az elemtípusok deklarálásának általános alakja:

```
<!ELEMENT elemneve tartalomleírás>
```

ahol “elemneve” az éppen deklarálni kívánt elem neve, tartalomleírás pedig meghatározza, hogy az elem milyen tartalommal rendelkezhet. Amennyiben az elem gyermekelemeket tartalmaz (elemtartalommal rendelkezik), az összes gyermekelem nevét fel kell tüntetni a szülőelem deklarációjában, majd az egyes gyermekelemeket egyesével deklarálni kell.

Az elem-név, annak az elemnek a neve, amelyhez a tulajdonság tartozik. A tulajdonság-név és az alapértelmezett érték egyértelmű, a tulajdonság típus pedig a következők valamelyike lehet:

Típus	Magyarázat
PCDATA	Ellenőrzött karakter adat
CDATA	Nem ellenőrzött karakter adat

(en1len2l..)	Az érték egy számozott lista valamely eleme lehet
ID	Az érték egy egyedi azonosító
IDREF	Az érték egy másik elem egyedi azonosítója
IDREFS	Más elemek egyedi azonosítóinak listája
NMTOKEN	Az érték egy érvényes XML név
NMTOKENS	Az érték érvényes XML nevek listája
ENTITY	Az érték egy egyed
ENTITIES	Az érték egyedek listája
NOTATION	Az érték egy megjegyzés neve

A fentiek alkalmazásával a mérleg saját tőke részét tartalmazó R199N310668586.xml nevű fájlhoz a DTD a következőképpen néz ki:

```
<?xml encoding="UTF-8"?>
<!ELEMENT Jelentes (AdatgyujtesKod,AdatszolgaltatoAzonosito,Gyakorisag,
                    VonatkozasiIdoTol,VonatkozasiIdoIg,
                    AdatkuldoAzonosito,Tabla+)>
<!ELEMENT AdatgyujtesKod (#PCDATA)>
<!ELEMENT AdatszolgaltatoAzonosito (#PCDATA)>
<!ELEMENT Gyakorisag (#PCDATA)>
<!ELEMENT VonatkozasiIdoTol (#PCDATA)>
<!ELEMENT VonatkozasiIdoIg (#PCDATA)>
<!ELEMENT AdatkuldoAzonosito (#PCDATA)>
<!ELEMENT Tabla (Sor)+>
<!ATTLIST Tabla
    Tablakod NMTOKEN #REQUIRED>
<!ELEMENT Sor (Sorkod,Cellaadat)>
<!ELEMENT Sorkod (#PCDATA)>
<!ELEMENT Cellaadat (#PCDATA)>
```

A lenti forráskód(dtd), a R159N310668586 fájlnevű - mely a külfölddel kapcsolatos követelések és kötelezettségek adatait tartalmazó - xml fájlhoz készült:

```
<!ELEMENT Jelentes (AdatgyujtesKod,AdatszolgaltatoAzonosito,Gyakorisag,
VonatkozasiIdoTol,VonatkozasiIdoIg, AdatkuldoAzonosito,Tabla+)>
<!ELEMENT AdatgyujtesKod (#PCDATA)>
<!ELEMENT AdatszolgaltatoAzonosito (#PCDATA)>
<!ELEMENT Gyakorisag (#PCDATA)>
<!ELEMENT VonatkozasiIdoTol (#PCDATA)>
<!ELEMENT VonatkozasiIdoIg (#PCDATA)>
<!ELEMENT AdatkuldoAzonosito (#PCDATA)>
<!ELEMENT Tabla (Nemleges|Sor+)>
<!ATTLIST Tabla
    Tablakod NMTOKEN #REQUIRED>
<!ELEMENT Nemleges (#PCDATA)>
<!ELEMENT Sor (Sorkod,Cellaadat+)>
<!ELEMENT Sorkod (#PCDATA)>
<!ELEMENT Cellaadat (#PCDATA)>
```

Példában létrehoztunk kettő DTD-t az MNB Jelentést tartalmazó XML fájlunkhoz. Elsőként magát a dokumentumelemet kellett deklarálnunk, amely elemtartalmú és az AdatgyujtesKod, AdatszolgaltatoAzonosito, Gyakorisag, VonatkozasiIdoTol, VonatkozasiIdoIg, AdatkuldoAzonosito, Tabla nevű gyermekelemeket fogja tartalmazni a felsorolás sorrendjében. Az egyes gyermekelemek felsorolásánál, a vessző (,) megadásával pontosan egyszer fognak szerepelni. Amennyiben a Tábla gyerekelem után a pluszt (+) elhagynánk, a dokumentumelem csak egy Tabla típusú gyermekelemet tartalmazhatna, de így akár többet is.

A használható előfordulást jelző karakterek a következők:

- ? - nulla vagy egy az előtte álló elemből
- + - egy vagy több az előtte álló elemből
- * - nulla vagy több az előtte álló elemből

3.2 Attribútumok

A deklarációban az attribútum-típust követően egy kulcsszó is szerepelhet, mely az attribútum alapértelmezett értékének jelölésére szolgál. Ezek a kulcsszavak az alábbiak lehetnek:

- **#REQUIRED** (kötelező): Mindenképpen meg kell adnunk valamely értéket;
- **#IMPLIED** (hallgatólagos, bennfoglalt): Nem muszáj megadnunk értéket – ilyen például az előzőekben az ID esete;
- **#FIXED** (rögzített): jelzi, hogy az attribútum neve után megadott érték rögzített, tehát nem változik, állandó;
- **#CURRENT** (aktuális, legutóbbi): Ha az adott elemnél nem adunk meg attribútumot, akkor az attribútum az ilyen típusú elemnél legutóbb megadott attribútum értéket veszi fel.

A leírtak alapján fenti kódot kiegészíthetjük az alábbiakkal:

```
<!ELEMENT Tabla (Sor)+>
<!ATTLIST Tabla
  Tablakod NMTOKEN #REQUIRED>
```

A következő lépés az EBEAD ellenőrzési folyamat alapján a megfelelő XML séma előállítása. A következő fejezetben ezt fogom reprezentálni.

4. XML séma

Az XML séma egy újabb keletű XML séma nyelv, amit a W3C a DTD utódaként definiáltak. Gyakran XSD (XML Schema Definition) néven is szokták emlegetni. Az XSD lényegesen többre képes a DTD-nél az XML nyelvek leírása terén. Sokoldalú adattípus rendszert használ, ami részletesebb megkötéseket tesz lehetővé az XML dokumentum logikai szintjén, de ezért sokkal robusztusabb érvényesítő keretrendszert követel meg. Ráadásul az XSD XML-alapú formátumon alapul, minek következtében szokványos XML eszközöket lehet használni a létrehozásához és feldolgozásához, bár az implementációk sokkal többet kívánnak, mint az egyszerű XML olvasási képesség. A DTD talán legnagyobb hátránya a külön szintaktika és némely helyzetekben a dokumentum tartalma feletti gyér ellenőrzés. Ezek után fordíthatunk egy kicsit a megközelítésen és ugyanezen két dolgot sorolhatjuk fel az XML-séma előnyeként. Következő fontos eltérés a deklaráció elhelyezésénél lép fel, mivel a DTD-ket deklarálhattuk az XML dokumentumon belül, vagy akár kívül egy külső fájlban, amelyet egy hivatkozás segítségével linkeltünk be. Az XML-sémáknál ez nem így van. Az XML séma-definíciót egy külön fájlban kell létrehozni, amelyre ugyanazon jól formázottsági szabályok vonatkoznak, mint egy XML dokumentumra, melyeket pár kiegészítés követ:

- egyetlen legfelső szintű elemet tartalmazhat a séma
- az összes sémában használt elemnek és jellemzőnek az "http://www.w3.org/2001/XMLSchema" névtérhez kell tartozni, ezzel is jelezve, hogy XML-sémáról van szó

Mivel az XML-séma elég sok lehetőséget biztosít a dokumentum tartalmának leírására, ezért célszerű forráskódot elemezve végigmenni a XML-séma definíción. A Jelentés XML séma fájl a következőképpen néz ki:

```

<?xml version="1.0" ?>
<xs:schema id="StartAdatGyujtes" targetNamespace="http://www.mnb.hu/StartAdatGyujtes"
xmlns="http://www.mnb.hu/StartAdatGyujtes"
xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="qualified"
elementFormDefault="qualified">
  <xs:element name="Jelentes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="AdatgyujtesKod" minOccurs="1" maxOccurs="1">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:length value="3" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="AdatszolgaltatoAzonosito" minOccurs="1" maxOccurs="1"
type="AdatszolgaltatoAzonosito"></xs:element>
        <xs:element name="Gyakorisag">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Napi"/>
              <xs:enumeration value="Heti"/>
              <xs:enumeration value="Havi"/>
              <xs:enumeration value="Negyedéves"/>
              <xs:enumeration value="Féléves"/>
              <xs:enumeration value="Éves"/>
              <xs:enumeration value="Eseti"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="VonatkozasiIdoTol" type="xs:date" minOccurs="1"
maxOccurs="1" />
        <xs:element name="VonatkozasiIdoIg" type="xs:date" minOccurs="1"
maxOccurs="1" />
        <xs:element name="AdatkuldoAzonosito" minOccurs="0" maxOccurs="1"
type="AdatszolgaltatoAzonosito" />
        <xs:choice maxOccurs="1" minOccurs="1">
          <xs:element name="Nemleges" type="xs:string" fixed="igen"
maxOccurs="1" minOccurs="1" />
          <xs:element name="Tabla" minOccurs="1" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:choice maxOccurs="1" minOccurs="1">
                  <xs:element name="Nemleges" type="xs:string"
fixed="igen" maxOccurs="1" minOccurs="1" />
                  <xs:element name="Sor" minOccurs="0"
maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:choice>
                          <xs:element name="Sorkod"
maxOccurs="1" minOccurs="1">
                            <xs:simpleType>
                              <xs:restriction
base="xs:string">
                                <xs:maxLength value="40" />
                                <xs:minLength value="1" />
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:choice>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                </xs:choice>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

type="xs:positiveInteger" maxOccurs="1" minOccurs="1" />
                                <xs:element name="Sorszam"
                                </xs:choice>
                                <xs:element name="Cellaadat" minOccurs="0"
maxOccurs="unbounded">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="1024" />
                                    </xs:restriction>
                                </xs:simpleType>
                                </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                </xs:choice>
                                </xs:sequence>
                                <xs:attribute name="Tablakod" form="unqualified"
use="required">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="10" />
                                    </xs:restriction>
                                </xs:simpleType>
                                </xs:attribute>
                                </xs:complexType>
                                </xs:element>
                                </xs:choice>
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                <xs:simpleType name="AdatszolgáltatoAzonosito">
                                    <xs:restriction base="xs:string">
                                        <xs:length value="8" />
                                    </xs:restriction>
                                </xs:simpleType>
                                </xs:schema>

```

A „Jelentés” alatti gyerekelemeket „AdatgyujtesKod”, „AdatszolgáltatóAzonosító”, „Gyakoriság”, „VonatozasiIdoTol”, „VonatozasiIdoIg” elemeket kötelező megadni, ebben a sorrendben. Az „AdatkuldoAzonosító” elemet csak akkor kell megadni, ha az adatküldő nem egyezik meg az adatszolgáltatóval. A felsorolt gyerekelemek mind egyszerű típusúak.

4.1 Egyszerű típus

Amennyiben **egyszerű típusú** elemet (nem tartalmaz gyermekelemeket) deklarálók nem származtatott tartalom-típussal, úgy magát az elem (tartalom) típust is az “element” elemen belül kell megadni:

```

<xs:element name="VonatokozasiIdoTol" type="xs:date" minOccurs="1"
maxOccurs="1" />
    <xs:element name="VonatokozasiIdoIg" type="xs:date" minOccurs="1"
maxOccurs="1" />

```

Az egyes elemek előfordulása korlátozható a **minimális** (minOccurs) illetve **maximális** (maxOccurs) előfordulás megadásával. Amennyiben nem adunk meg minimális és maximális előfordulást, úgy az elemnek pontosan egyszer kell előfordulnia.

Az XML-séma egyszerű típusai közül néhány:

- **string** - bármilyen érvényes karakter
- **integer** - <-126789, 126789> intervallumból bármilyen egész szám, vagy nulla. Pl. (-100, 0, 15)
- **positiveInteger** - <1, 126789> intervallumból bármilyen egész szám. Pl. (2, 50, 999)
- **negativeInteger** - <-126789, -1> intervallumból bármilyen egész szám. Pl. (-25, -559, -78902)
- **nonNegativeInteger** - <0, 126789> intervallumból bármilyen egész szám. Pl. (0, 50, 98745)
- **nonPositiveInteger** - <-126789, 0> intervallumból bármilyen egész szám. Pl. (0, -50, -98745)
- **decimal** - pozitív illetve negatív tizedes szám
- **time** - idő a következő formátumban: óó:pp:mm[.mm+-UTC], Pl. (14:30:00, 16:30:00.0, 16:30:00-01:00, 16:30:00.00-01:00)
- **dateTime** - dátum és idő a következő formátumban: éééé-hh-nnTóó:pp:mm.mm+-UTC. Pl. (2004-03-15T15:30:00.0-01:00)
- **gYear** - év a következő formátumban: éééé. Pl. (2004, 1983)
- **gMonth** - hónap a következő formátumban: --hh--. Pl. (--03--, --12--) -> (március, december)
- **gDay** - nap a következő formátumban: ---nn. Pl. (---01, ---23)
- **gYearMonth** - év és hónap a következő formátumban: éééé-hh. Pl. (1983-02)

Ezen típusok úgynevezett alaptípusok, amelyekből lehetőség van saját típus létrehozására úgy, hogy az egyes alaptípusokból egy saját típust származtatunk.

Ahogy az a definíciórészletből is kiderül a típuszármaztatás a **restriction** elem segítségével történik amely a simpleType gyermekeleme, egyszerű típusról lévén szó. A **restriction** elem **base** jellemzőjének adjuk meg azt a típust, amelyből származtatni szeretnénk a saját típusunkat. A **restriction** elem gyermekelemeiként megadhatjuk az új típusra vonatkozó megszorításokat:

- Jelen esetben ezek a megszorítások a minimális (**minLength**) és maximális (**maxLength**) karakterhossz meghatározása.

```
<xs:element name="Sorkod" maxOccurs="1" minOccurs="1">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="40" />
      <xs:minLength value="1" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **Reguláris kifejezést (pattern)** - segítségével egy olyan sablont hozhatunk létre, amellyel pontosan leírhatjuk egy karaktersorozatban használható karakterek listáját illetve sorrendjét.
- **Felsorolást (enumeration)** - melynél egy előre definiált lista elemei közül választhatunk egyet:

```
<xs:element name="Gyakorisag">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Napi"/>
      <xs:enumeration value="Heti"/>
      <xs:enumeration value="Havi"/>
      <xs:enumeration value="Negyedéves"/>
      <xs:enumeration value="Féléves"/>
      <xs:enumeration value="Éves"/>
      <xs:enumeration value="Eseti"/>
    </xs:restriction>
  </xs:simpleType>
```

4.2 Összetett típus

A gyökérelem a „Jelentes”, ami egy összetett elem. **Összetett típusúnak** nevezzük azon elemeket, melyek gyermekelemeket, jellemzőket, vagy karakteres adatokat és gyermekelemeket is tartalmaznak egyidejűleg. Amennyiben összetett típusú elemeket szeretnénk használni, a következő lehetőségek közül választhatunk:

- **általános összetett típus** (anyType) használata esetén az elem tartalmazhat bármilyen jellemzőt, elemet és karakteres adatot
- **saját összetett típus** létrehozása akár névtelenül, akár nevesített módon

Összetett típusokat hasonló módon hozhatunk létre, mint az egyszerű típusokat. Az egyes gyermekelemek a complexType elemen belül deklarált sequence, choice vagy all gyermekeiként adjuk meg, attól függően, hogy:

- **felsorolást** szeretnénk **kötött elemsorrenddel** (sequence)
- **választólistát** szeretnénk(choice)

```
<xs:element name="Sor" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element name="Sorkod" maxOccurs="1" minOccurs="1">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="40" />
              <xs:minLength value="1" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:choice>
      <xs:element name="Cellaadat" minOccurs="0" maxOccurs="unbounded">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="1024" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

- **felsorolást szeretnék kötetlen elemsorrenddel (all)**

Az XML egyik sajátossága az attribútumok, ahhoz, hogy egy elem tartalmazzon jellemzőket összetett típusúnak kell lennie. Az egyes jellemzőket a complexType gyermekelemeiként deklaráljuk. Az egyes jellemzők csakis kizárólag egyszerű típusúak lehetnek, tehát vagy az alaptípusok egyikét használjuk, vagy származtatunk egy új típust. A származtatás és létrehozás nagyban hasonlít az egyszerű típusú elemek létrehozásához. Az egyes jellemző deklarációk a következő jellemzőket támogatják:

- **default** - alapérték, amelyet felvesz a jellemző akkor, ha nem adunk meg neki értéket
- **fixed** - értékéül megadhatjuk azt az értéket, amelyet mindenképpen fel kell venni a jellemzőnek, tehát ha nem adunk meg értéket, akkor a fixedben megadott értéket veszi fel a jellemző, viszont, ha megadunk értéket akkor csakis kizárólag a fixedben megadott értéket adhatjuk.
- **use** - értékéül megadhatjuk
 - **optional** - a jellemző elhagyható, ami egyben az alapérték is
 - **prohibited** - a jellemző le van tiltva, olyan mintha nem is deklaráltuk volna
 - **required** - a jellemző használata kötelező, pl: kódrészletben:

```
<xs:attribute name="Tablakod" form="unqualified" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="10" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

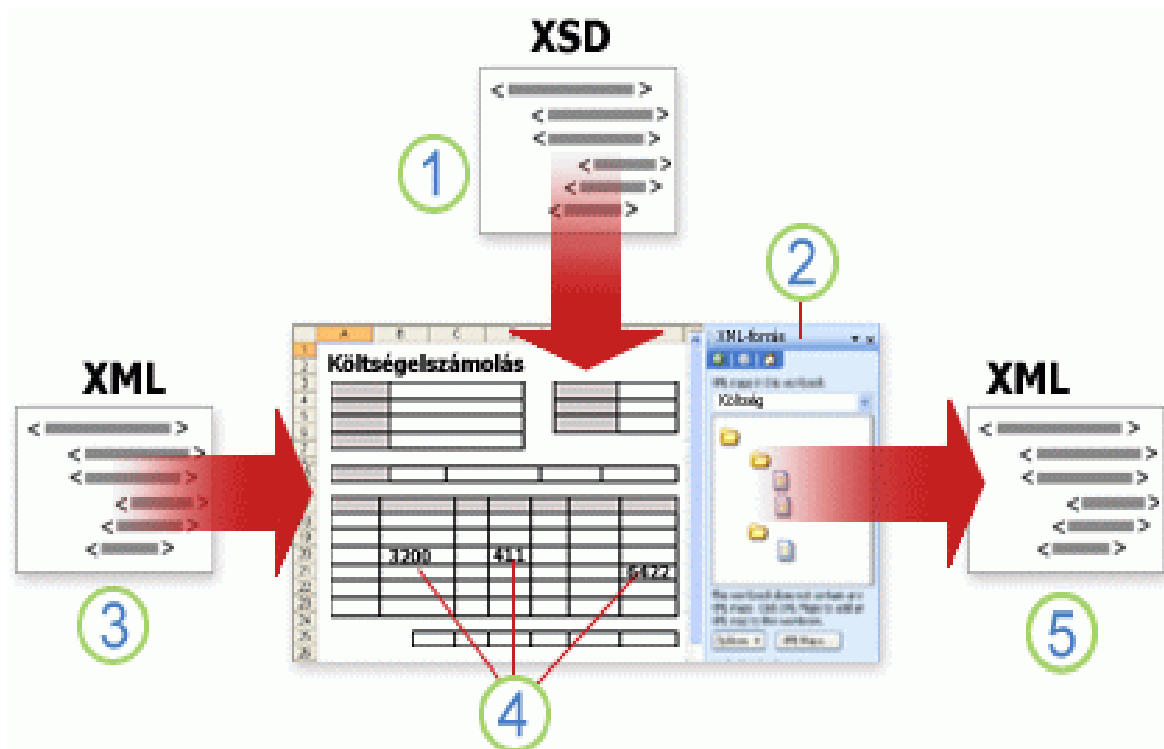
4.3 XML és az Excel kapcsolata

Az így elkészített XML séma alapján hozzákezdhetünk az XML előállításához. Az XML és az Excel együttes használatával olyan módon tudjuk kezelni a munkafüzeteket és az adatokat, amely korábban lehetetlen vagy nagyon nehézkes volt. XML-megfeleltetések használatával egyszerűen lehet üzleti adatokat felvenni, azonosítani és kinyerni az Excel-dokumentumokból. Ezeket az adatokat minden nehézség nélkül importálhatjuk különféle adatbázisokból és alkalmazásokból, áttekinthetjük és javíthatjuk az adatokat, majd exportálhatjuk azokat ugyanazon vagy más adatbázisokba és alkalmazásokba.

Az Excel és a XML-szolgáltatásokkal együtt megoldható eseteket:

- XML-elemek már létező celláknak történő megfeleltetésével bővíthető a már meglévő Excel-sablonok szolgáltatási köre, ezáltal egyszerűbbé válik az XML-adatok sablonokból történő kinyerése és azokban történő elhelyezése, és a sablonok újratevezésére sincs szükség.
- XML-adatok használhatók a meglévő munkalapok bemeneteként az XML-elemek és a meglévő számítási modellek megfeleltetésével.
- XML-adatfájlok importálhatók új munkafüzetekbe.
- XML-tartalom importálható web szolgáltatásokból az Excel-munkafüzetbe.
- A megfeleltetett cellákban lévő adatok XML-adatfájlokba exportálhatók a munkafüzet egyéb adataitól függetlenül. *(Mi ezt fogjuk kihasználni!)*

A következő diagram azt szemlélteti, hogy miként működnek együtt a különböző fájlok és műveletek az XML Excel alkalmazásbeli használatakor. Lényegében öt feldolgozandó fázis különböztethető meg:



3. ábra: XML-XSD felhasználási lehetőségek (forrás: <http://office.microsoft.com>)

- 1) .xsd kiterjesztésű XML-sémafájl hozzáadása munkafüzethez
- 2) XML-sémaelemek megfeleltetése egyes celláknak vagy XML-táblázatoknak
- 3) .xml kiterjesztésű XML-adatfájl importálása és az XML-elemek megfeleltetett cellákhoz kötése, el is hagyható, helyette mi készíthetjük, vihetjük be az adatokat a megfeleltetés alapján a cellákba.
- 4) Adatok bevitele, megfeleltetett cellák áthelyezése és az Excel funkcióinak alkalmazása az XML-struktúra és a definíciók megőrzése mellett
- 5) Ellenőrzött adatok exportálása megfeleltetett cellákból XML-adatfájlba

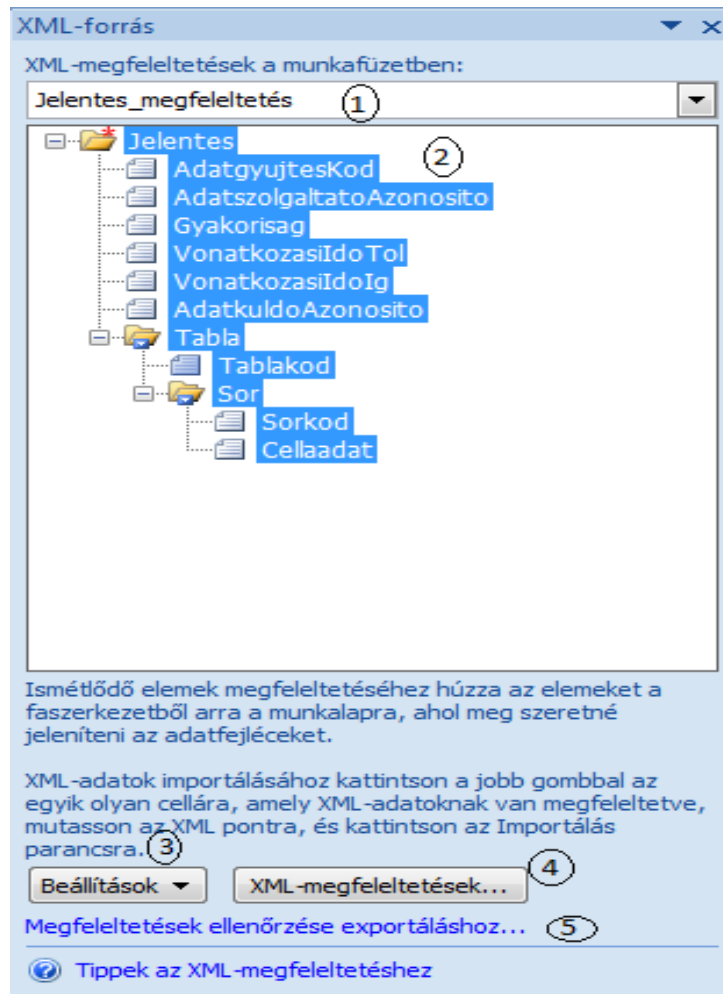
Az Excel programban létrehozott vagy megnyitott munkafüzethez .xsd kiterjesztésű XML-sémafájl csatolható, majd az XML-forrás munkaablakkal megfeleltethetők a séma XML-elemei a különböző celláknak vagy táblázatoknak. Ezt követően XML-tartalom importálható és exportálható a megfeleltetett cellákba, illetve a cellákból. Az .xsd kiterjesztésű XML-sémafájl munkafüzetbe adásakor XML-megfeleltetés jön létre, amely általában a megfeleltetett cellák létrehozására, valamint e cellák és az XML-sémában található elemek

kapcsolatának kezelésére használatos. Ezen kívül .xml kiterjesztésű XML-adatfájlok importálásakor és exportálásakor e megfeleltetésekkel köthetők össze a sémában található elemek és a megfeleltetett cellák tartalma.

Kétféle megfeleltetett cella hozható létre: *egyetlen hozzárendeléssel* rendelkező cellák és (XML-táblázatokként megjelenő) ismétlődő cellák. A munkalap hatékonyabb kialakítása érdekében a megfeleltetett cellák bárhová húzhatók a munkalapon, és sorrendjük sem kötött – a sorrend még az XML-sémáétól is különbözhet. Megadható az is, hogy a program mely elemeket képezze le.

Az XML-megfeleltetések használatakor a következő szabályokat kell szem előtt tartani:

- A munkafüzetben több XML-megfeleltetés is lehet.
- Egyszerre csak egy elem képezhető le egy adott helyre a munkafüzetben.
- Minden egyes XML-megfeleltetés önálló egység, még akkor is, ha a munkafüzet több XML-megfeleltetése utal ugyanarra a sémára.
- Az XML-megfeleltetések csak egy gyökérelemet tartalmazhatnak. Ha olyan sémát adunk hozzá, amely több gyökérelemet határoz meg, akkor a program kéri, hogy válassza ki az új XML-megfeleltetéshez használandó gyökérelemet.
- Az XML-forrás munkaablakkal az XML-megfeleltetéseket lehet kezelni. A következő ábra e munkaablak fő funkcióit szemlélteti:



4. ábra: Excel XML forrás ablak (Saját forrás)

- 1) A munkafüzethez adott XML-megfeleltetések listája
- 2) Az aktuális XML-megfeleltetés XML-elemeinek hierarchikus listája
- 3) Az XML-forrás munkaablak és az XML-tartalom beállításainak megadása (például az adatok megtekintésének és a fejlécek ellenőrzésének módja)
- 4) Az XML-megfeleltetések hozzáadásához, törléséhez és átnevezéséhez használható XML-megfeleltetések párbeszédpanel megnyitása
- 5) Annak ellenőrzése, hogy az XML-tartalom exportálható-e az aktuális XML-megfeleltetésen keresztül

Ezek után elkezdhetjük a cellák beillesztését a munkalapra, de figyelniük kell az adatok típusára. Az egyetlen hozzárendeléssel rendelkező cella olyan cella, amely egy nem ismétlődő

XML-elemhez van rendelve. Ilyen cella létrehozásához nem ismétlődő XML-elemeket kell az XML-forrás munkablakból a munkalap egy adott cellájára húzni.

Nem ismétlődő XML-elem munkalapra húzásakor az elem nevét intelligens címkével címsorként az egy hozzárendeléssel rendelkező cella fölé vagy attól balra írhatjuk. Címsorként egy létező cellaértéket is használhatjuk. Egyetlen hozzárendeléssel rendelkező cellákban képlet is használható, feltéve, hogy a cella olyan XML-sémadefinícióval kapcsolódik egy XML-elemhez, amelyet az Excel számként, dátumként vagy időpontként értelmez.

Lehetőség van *ismétlődő cellák* használatára az XML-táblázatokban, ilyenkor, az XML-táblázatok megjelenésüket és ábrázolásukat tekintve hasonlóak az Excel-táblázatokhoz: ez utóbbiak legalább egy XML-elemnek vannak megfeleltetve. Az XML-táblázat valamennyi oszlopa egy-egy XML-elemet jelöl:

	A	B	C	D	E	F	G	H	I
1									
2									
3	AdatgyujtesKod	AdatszolgáltatoAzonosito	Gyakorisag	VonatkozasidoToL	VonatkozasidoToI	AdatkuldoAzonosito	Tablakod	Sorkod	Cellaadat
4	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	ELOLAP	ELOLAP03	szucsbel@kite.hu
5	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	ELOLAP	ELOLAP02	54/480-401/353
6	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	ELOLAP	ELOLAP05	54/480-401/784
7	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	ELOLAP	ELOLAP04	Nyitrai Ibolya
8	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	ELOLAP	ELOLAP07	2009-07-30
9	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	ELOLAP	ELOLAP06	nyitraibolya@kite.hu
10	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	ELOLAP	ELOLAP01	Szűcs Béla
11	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ11	0
12	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ12	913
13	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ04	1776
14	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ07	1489
15	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ13	17368
16	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ01	31813
17	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ03	41124
18	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ02	486
19	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ05	5
20	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ09	0
21	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ08	0
22	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ06	0
23	R19	10668586	Negyedeves	2009.04.01	2009.06.30	10668586	TAJ	TAJ10	14966
24									
25									
26									

5. ábra: XSD megfeleltetések (Saját forrás)

Az XML-elemeknek megfeleltetett oszlopokban képletek is használhatók olyan XML-sémadefiníció (XSD) szerinti adattípussal, amelyet az Excel számként, dátumként vagy időpontként értelmez, jelen esetben ezt nem használtuk, mivel konkrét adatokat vittünk be.

Adatok érvényesítése a séma alapján történik. Exportáláshoz megadhatjuk, hogy az adatok exportálásakor az Excel érvényesítse az adatokat az XML-megfeleltetés alapján. Ezután az XML-adatok exportálása a munkalapon lévő megfeleltetett cellák tartalmának exportálásával történik. Adatok XML-be való exportálása után nincs más hátra, mint az elkészült XML fájl feltöltése az MNB EBEAD rendszerébe:

6. ábra: Elkészített XML visszaigazolás (Forrás: EBEAD)

Ha megnézzük az MNB rendszerében az adatokat, akkor láthatjuk, hogy az állomány formátuma XML és alatta a Jelentés státusza „Elküldve”, ami azt jelenti, hogy az adatokat a korábban leírt módszer alapján ellenőrizte (mind a jelentés formátuma, mind számvitelileg megvizsgálta mérlegsorokat, de erre nem tértünk ki számviteli volta miatt) és rendben találta, így elküldte feldolgozásra statisztikai jelentéshez, amit negyedévente az MNB publikál.

Összefoglalva elmondható, hogy sikerült megvalósítanunk az eredeti elképzelésünket, továbbá hogy az XML nagyban megkönnyíti az adatok alkalmazások, adatbázisok és szervezetek közötti definiálását, átvitelét, érvényesítését és értelmezését! Továbbiakban két olyan XML szolgáltatást mutatunk be, amely lehetőséget teremt az XML elemek közvetlen elérésére (XPath), illetve hibakeresés vagy elemzés elkészítése során előfordulhat, hogy csak bizonyos feltételek alapján kiválasztott, rendezett elemekre van szükségünk, melyet az SQL-hez hasonló, de XML-hez írt lekérdező nyelv segítségével valósíthatunk meg (XQuery).

5. XPath

Az XSLT és az XPointer szabványokhoz szükség volt egy módszerre, amelynek a segítségével leírható egy XML dokumentum adott része, ezért jött létre az XPath nyelv. Szintaxisa nem XML alapú, az XML dokumentumban való navigáción kívül lehetőséget ad logikai, számtani és string műveletek elvégzésére is. Egy XML dokumentum adott részének elérésén túl az XPath nyelv alkalmas annak leírására is, hogy egy csomópont megfelel-e egy adott mintának.

Az XPath értelmezésében egy XML dokumentum 7 féle részből állhat, ezek:

1. Gyökér csomópontok (root nodes),
2. csomópontok (element nodes),
3. szöveges csomópontok (text nodes),
4. tulajdonságok (attribute nodes), névterek (namespace nodes),
5. vezérlési utasítások (processing instruction nodes) és
6. megjegyzések (comment nodes).

Egy XPath kifejezésnek az eredménye lehet csomópontok egy csoportja, logikai érték (igen vagy nem), szám vagy szöveg. Egy kifejezés kiértékelésekor rendszerint figyelembe vevődik az aktuális helyzet, amit az XSLT vagy XPointer értelmező határoz meg. Az aktuális helyzetet több információ együttesen írja le, például az aktuális csomópont, annak pozíciója, mérete, stb.

5.1 Elérési-út (*location path*)

Minden elérési-út lépésekből áll (location steps), mely lépések egymástól egy „/” jellel vannak elválasztva, ezen lépések összessége határozza meg az XML dokumentum adott pontjaihoz vezető utat. Ez nagyon hasonlít a fájl rendszerben megszokott elérési-utakhoz, ahol az egyes könyvtárak jelentik a lépéseket, itt azonban az egyes lépések jóval komplexebbek is lehetnek az XML dokumentum elemeinél, hiszen akár több elemet is jelölhetnek. Akárcsak a

fájl rendszerek esetén, itt is kétféle útvonal leírás létezik. Az egyik a relatív, ahol az elérési út kiinduló pontja az aktuális pozíció, a másik az abszolút leírás, ahol a leírás egy „/” jellel kezdődik és a kiindulási pont az XML dokumentum gyökere (root). Az abszolút leírást lehet úgy tekinteni, mint egy olyan relatív leírást, ahol az aktuális pozíció ideiglenesen – erre az egy számításra – a gyökérre állítódik át.

Egy elérési-út által elérhető csomópontok körének meghatározásakor az elérési út értelmezése mindig balról jobbra történik. Az első lépés során a lépés kiinduló pontja az aktuális – vagy abszolút leírás esetén a gyökér –elem, és a lépés során kiszámítódik mindazon csomópontok köre, amelyet az adott lépés leír.

A `child::Sor` kifejezés tehát az összes olyan elemet jelenti, amelynek a neve „Sor” és amelynek a szülője az aktuális elem.

Ha a teljes elérési-út több lépésből áll, akkor a következő lépés kiinduló pontja - a lépés esetén értelmezett aktuális elem – az összes az előző lépésben elért elemek halmaza lesz, és az ő általa elért elemek köre ehhez képest relatív.

A „`child::Sor/child::Sorkod`” elérési út tehát az összes olyan „Sorkod” elemet jelenti, amelynek szülő eleme „Sor”, és amelynek nagyszülője az aktuális elem (vagyis az aktuális elem összes „Sor” gyermekének az összes „Sorkod” gyermeke). Ha csak egy ilyen van, akkor ez az elérési út egy konkrét elemet jelöl, ha több is van, akkor több elemet is leír.

A korábbi fejezetben elkészített Jelentés XML dokumentum részlet:

```
<?xml version="1.0" encoding="utf-8"?>
<Jelentes>
  <AdatgyujtesKod>R19</AdatgyujtesKod>
  <AdatszolgaltatoAzonosito>10668586</AdatszolgaltatoAzonosito>
  <Gyakorisag>Negyedeves</Gyakorisag>
  <VonatkozasiIdoTol>2009-04-01</VonatkozasiIdoTol>
  <VonatkozasiIdoIg>2009-06-30</VonatkozasiIdoIg>
  <AdatkuldoAzonosito>10668586</AdatkuldoAzonosito>
  <Tabla Tablakod="ELOLAP">
    <Sor>
      <Sorkod>ELOLAP03</Sorkod>
      <Cellaadat>szucsbelakite.hu</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP02</Sorkod>
      <Cellaadat>54/480-401/353</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP07</Sorkod>
      <Cellaadat>2009-07-30</Cellaadat>
    </Sor>
  </Tabla>
</Jelentes>
```

```

    </Sor>
  </Tabla>
<Tabla Tablakod="TAJ">
  <Sor>
    <Sorkod>TAJ11</Sorkod>
    <Cellaadat>0</Cellaadat>
  </Sor>
  <Sor>
    <Sorkod>TAJ12</Sorkod>
    <Cellaadat>913</Cellaadat>
  </Sor>
  <Sor>
    <Sorkod>TAJ04</Sorkod>
    <Cellaadat>1776</Cellaadat>
  </Sor>
  <Sor>
    <Sorkod>TAJ07</Sorkod>
    <Cellaadat>1489</Cellaadat>
  </Sor>
  <Sor>
    <Sorkod>TAJ13</Sorkod>
    <Cellaadat>17368</Cellaadat>
  </Sor>
</Tabla>
</Jelentes>

```

Ha az aktuális elem a Jelentes, amely egyben a gyökér elem is akkor:

Kifejezés	Eredmény
Jelentes	kiválasztja a Jelentes összes gyermek elemét
/Jelentes	kiválasztja a Jelentes gyökérelmet
Jelentes/Tabla	kiválasztja a Jelentes összes Tabla nevű gyerekét
//Celladat	kiválasztja az összes Celladat elemet, akár hol is van a dokumentumban
//@Tablakod	kiválasztja az összes Tablakod attribútumot
//Tabla[@Tablakod='TAJ']	kiválasztja az összes Tabla elemet, amelynek van „TAJ” attribútuma
/Jelentes/Tabla[1]	kiválasztja az első Tabla elemet, amely gyereke a Jelentes-nek
/Jelentes/Tabla[Last()]	kiválasztja a legutolsó gyerek elemét a Jelentes-nek
//Sor/Sorkod //Sor/Cellaadat	kiválasztja az összes Sorkod és Cellaadat elemét az összes Sor elemek
/Jelentes/Tabla/Sorkod //Gyakorisag	kiválasztja az összes Sorkod elemet amely gyereke Tabla-nak, amely gyereke a Jelentes-nek és az összes Gyakorisag elemet a dokumentumban

1. táblázat: Kifejezések kiértékelése

5.2 Lépések (location steps)

Egy XPath lépés három részből áll, felépítése a következő:

irány :: csomópont típus [feltételes kifejezés]

5.3 Irányok

Az irány, a lépéssel megjelölni kívánt csomópontok és az aktuális elem viszonyát írja le.

Lehetséges irányok:

child:: Gyermek. Az aktuális elem gyermek eleme. Amennyiben nincs irány megadva, akkor ez az alapértelmezés szerinti irány.

descendant:: Leszármazott. Az aktuális elem összes leszármazottja (a gyermekei és azoknak is a gyermekei rekurzívan).

parent:: Szülő. Az aktuális elem szülő elemét jelöli ki.

ancestor:: Felmenő. Az aktuális elem összes felmenőit jelöli ki. (A saját szülőjét és annak a szülőjét is rekurzívan, így természetesen szükségszerűen mindig a gyökér elemet is.)

following-sibling:: Következő testvér. Az aktuális elem következő testvérét jelöli ki. Ha az aktuális csomópont tulajdonság vagy névtér, a következő elem halmaz üres.

preceding-sibling:: Előző testvér. Az aktuális elem előző testvérét jelöli ki. Ha az aktuális csomópont tulajdonság vagy névtér, az előző elem halmaz üres.

following:: Következők. Az aktuális elemet követő minden elem a dokumentum sorrendjében.

preceding:: Megelőzők. Az aktuális elemet megelőző minden elem a dokumentum sorrendjében.

attribute:: Tulajdonság. Az aktuális elem tulajdonságait jelöli ki. Üres, ha az aktuális elem nem egy csomópont.

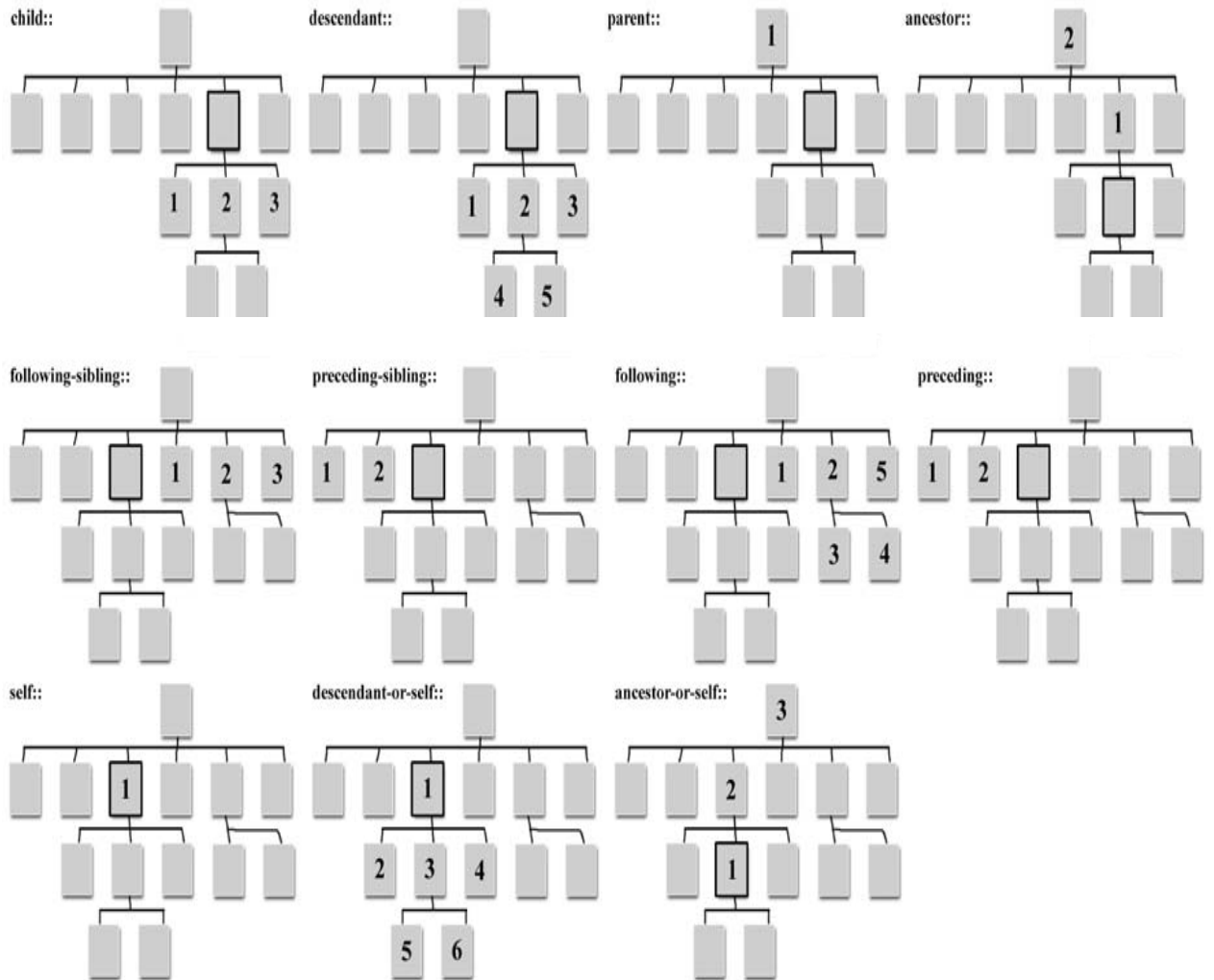
namespace:: Névtér. Az aktuális elem névtereit jelöli ki. Üres halmaz, ha az aktuális elem nem egy csomópont.

self:: Önmaga. Ez az aktuális elemet jelöli ki.

descendant-or-self:: Leszármazottak vagy önmaga. Az aktuális elemet és a leszármazottait jelöli.

ancestor-or-self:: Felmenő vagy önmaga. Az aktuális elemet és a felmenőit jelöli.

Az XPath lépések irányai (az aktuális elem van bekeretezve):



7. ábra: XML elemtípusok viszonyai (Forrás: <http://prog.hu>)

5.4 Típusok

XPath 7 féle csomópont típust különböztet meg, ebből három alap típus – névtér, tulajdonság, csomópont –, és négy a csomópont alaptípus speciális esete – szöveges csomópont, megjegyzés, gyökér, vezérlési utasítás. Az XPath-ban minden iránynak van egy elsődleges típusa, ez tulajdonság, ha az aktuális irány tulajdonság, névtér, ha az aktuális irány névtér, és

minden más esetben csomópont. Egy típust két módon lehet megadni, vagy név megadásával, vagy csomópont típus kiválasztó utasítások segítségével.

Lehetséges csomópont típusok:

„név” A „név” által megadott nevű tulajdonságot (attribute:: irány esetén), vagy elemeket választja ki. „*” Az adott iránynak megfelelő bármilyen nevű, tehát az adott iránynak megfelelő összes elemet kiválasztja.

- comment() Az adott iránynak megfelelő minden megjegyzés elemet kiválasztja.
- text() Az adott iránynak megfelelő minden szöveges elemet kiválasztja.
- processing-instruction() Az adott iránynak megfelelő minden vezérlési utasítást kiválasztja.
- node() Az adott iránynak megfelelő minden elemet a típusától függetlenül, kiválasztja.

Kifejezés	Eredmény
child::Gyakoriság	A Jelentes elem gyermekei közül a „Gyakorisag” nevűt választja ki.
child::*	A Jelentes elem összes gyermeke Adatkod, AdatszolgaltatoAzonosito, Gyakorisag, Tabla, stb.
child::Bekuldo	A Jelentes elem gyermekei közül a Bekuldo nevűt, de miután a Jelentes elemnek nincs Bekuldo nevű gyermeke, ezért ez egy üres halmaz.
descendant::Tabla	A Tabla elem leszármazottai közül a Sor, Sorkod, Cellaadat nevűek, tehát végigmegyünk a fán lefelé.
child::node()	A Jelentes elem összes gyermeke.
ancestor::Sor	A sor elem össze felmenőit jelöli, tehát Tabla, Jelentes elemeket

2. táblázat: Kifejezés kiértékelés

5.5 Feltételes kifejezés

A feltételes kifejezés egy tetszőleges összetett kifejezés, amely a csomópontok körének meghatározására egyéb XPath kifejezéseket, függvényeket, és a csomópontok tartalmát is felhasználhatja. A kifejezés kiértékelése előtt létrejön egy az irány és egy a típus által meghatározott csomópont halmaz. Ezután az értelmező végig megy a halmaz elemein, és minden egyes elemre kiértékeli a feltételes kifejezést. Ha a kifejezés az adott elemre igaz értékű, akkor a csomópont bekerül az XPath lépés eredmény halmazába.

Egy feltételes kifejezés egy vagy több kifejezésrészből épül fel, amelyek között logikai műveletek teremtenek kapcsolatot. A kifejezéshez a következő műveleti jelek használhatók fel:

- **or** (logikai vagy)
- **and** (logikai és)
- = (egyenlőség vizsgálat), != (nem egyenlőség vizsgálat)
- <= (kisebb vagy egyenlő), < (kisebb), >= (nagyobb vagy egyenlő), > (nagyobb)

Itt a sorrend a művelet rendjét is jelzi (tehát az „or” művelet a legmagasabb rendű). A kifejezés pontos megadásához használható zárójelezés. A feltételes kifejezés nem kötelező, elhagyása esetén a szögletes zárójelek sem szükségesek. Egy kifejezésrész lehet valamilyen szöveg, szám, függvényhívás vagy XPath kifejezés.

Kifejezés	Eredmény
//Tabla[@Tablakod='TAJ']	kiválasztja az összes Tabla elemet, amelynek van „TAJ” attribútuma
//Sor[Sorkod='TAJ12']	kiválasztja azon Sor elemeket, amelynek van „TAJ12” tartalmú gyermek elemük
//Sor/Sorkod //Sor/Cellaadat	kiválasztja az összes Sorkod és Cellaadat elemét az összes Sor elemek
/Jelentes/Tabla/Sorkod //Gyakorisag	kiválasztja az összes Sorkod elemet amely gyereke Tabla-nak, amely gyereke a Jelentes-nek és az összes Gyakorisag elemet a dokumentumban
//Sor[Cellaadat>100]/Cellaadat	kiválasztja azon Celladat elemeket, melynek nagyobb az értéke, mint 100

3. táblázat: Kifejezés kiértékelés

5.6 Függvények

Csomópont halmaz függvények

- last() Megadja az aktuális halmaz utolsó elemének indexét. (vagyis a halmaz méretét)

- `position()` Megadja az aktuális elem indexét. `count(halmaz)` Megadja a paraméterként megadott halmaz csomópontjainak számát.
- `id(objektum)` Az adott azonosítóval rendelkező elemet adja vissza. Csak séma definícióval rendelkező dokumentum esetében működik, hiszen csak ebben az esetben állapítható meg, hogy mi az azonosító.
- `local-name(halmaz)` Ha az aktuális elem kiterjesztett névvel rendelkezik, akkor a függvény megadja ennek a kiterjesztett névnek a helyi (local) részét.
- `namespace-uri(halmaz)` Ha az aktuális elem kiterjesztett névvel rendelkezik, akkor a függvény megadja ennek a kiterjesztett névnek a névtér URI részét.
- `name(halmaz)` Vissza adja az aktuális elem nevét.

Kifejezés	Eredmény
<code>/Jelentes/Tabla[1]</code>	kiválasztja az első Tábla elemet, amely gyereke a Jelentesnek
<code>/Jelentes/Tabla[Last()]</code>	kiválasztja a legutolsó elemét gyermek elemét a Jelentesnek
<code>//Table/Sor[position()<3]</code>	kiválasztja az első 2 Sor elemet, amely gyereke a Tabla elemnek

4. táblázat: Kifejezés kiértékelés

String függvények

- `string(objektum)` A paraméterként megadott objektumot szöveggé alakítja át.
- `concat(String, String*)` A paraméterként megadott - korlátlan számú - karakterláncot összefűzi.
- `starts-with(String, String)` Igaz, ha a paraméterként megadott első sztring a második sztringgel kezdődik.

- `contains(String, String)` Igaz, ha a paraméterként megadott első sztring tartalmazza a második sztringet.
- `substring-before(String, String)` Vissza adja az első string mindazon részét, ami a második string első stringben való első előfordulása előtt van. Ha az első string nem tartalmazza a második stringet, az eredmény üres string.
- `substring-after(String, String)` Vissza adja az első string mindazon részét, ami a második string első stringben való első előfordulása után van. Ha az első string nem tartalmazza a második stringet, az eredmény üres string.
- `substring(string, number, number?)` Az első stringnek az első szám által meghatározott pozíciótól kezdődő a második szám által meghatározott számú karakterét adja vissza. Ha a második szám hiányzik, akkor a visszaadott string az eredeti string végéig tart.
- `string-length(string?)` A paraméterként megadott, vagy annak hiányában az aktuális elem értékét számmá alakítja.
- `normalize-space(string?)` A paraméterként megadott stringből, vagy annak hiányában az aktuális eleméből kiveszi a fölösleges kihagyás jellegű karaktereket.
- `translate()` Kis-nagy betű konverzióra szolgáló függvény.

Logikai függvények

- `boolean(objektum)` A paraméterként megadott objektumot logikai értéké alakítja át.
- `not(boolean)` A paraméterként megadott logikai értéknek az ellentettjét adja vissza.
- `true()` Mindig igaz értéket ad vissza. Nincs paramétere.
- `false()` Mindig hamis értéket ad vissza. Nincs paramétere.
- `lang()` Megadja, hogy az adott csomópont azonos nyelvű-e a paraméterként megadott nyelvvel. Egy csomópont (ez a továbbiakban minden leszármazottra öröklődik) nyelvét az `xml:lang` tulajdonság segítségével lehet megadni. Ritkán használják.

Számtani függvények

- `number(objektum)` A paraméterként megadott objektumot számmá alakítja át.
- `sum(halmaz)` Megadja a paraméterként megadott halmaz számértékeinek összegét.

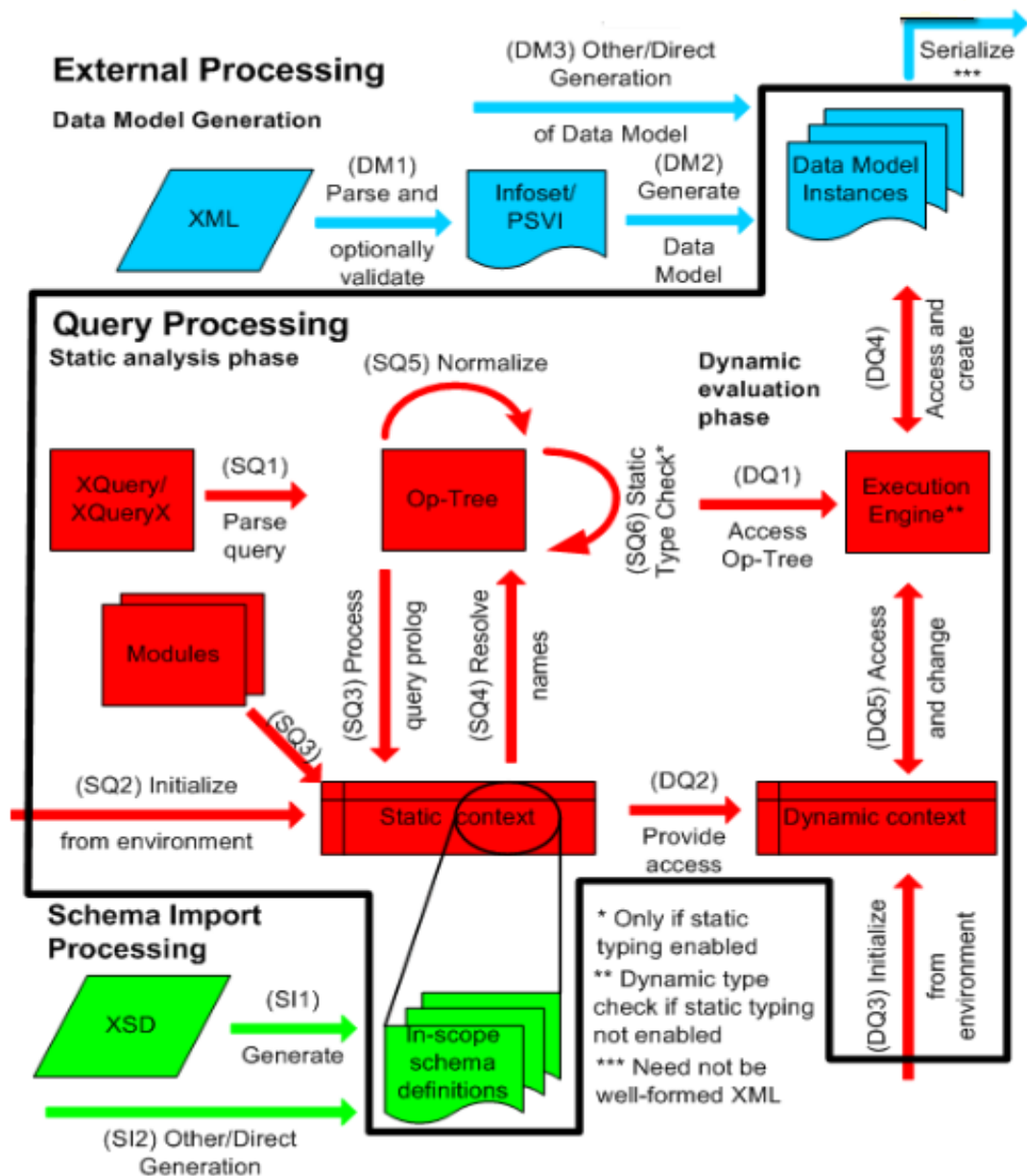
- `floor(number)` A paraméterként megadott számot lefelé kerekíti.
- `ceiling(number)` A paraméterként megadott számot felfelé kerekíti.
- `round(number)` A paraméterként megadott számot kerekíti.

6. XQuery

Mivel az XPath nyelv nem elegendő az XML dokumentumok lekérdezésére, csak egy címzési lehetőség az XQuery az XPath elérési-út kifejezéseit használja az XML dokumentum részeinek a visszatérítésére. XQuery olyan funkcionális nyelv, amelyik XML alapú adatforrásokat tud lekérdezni. A lekérdező eszköz remény szerint az XML-adatelérés szabványa lesz majd. A lekérdezett állománynak nem kell feltétlenül XML-formátumú adatokat tartalmaznia, de támogatnia kell az XML-t, és az eredményt mindenképpen XML-formátumban kapják a felhasználók.

6.1 XQuery alapok

- Az XQuery 1.0 2007. január 23-án vált hivatalos W3C szabvánnyá. Ennek köszönhetően az összes (legelterjedtebb) böngésző és adatbáziskezelő-rendszer érti és végre is tudja hajtani az XQuery utasításokat.
- W3C szabványa mely lapján a lekérdezési formula szerkezete:
 - alaphalmaz kiválasztása:* több hierarchia szelet is kiválasztásra kerülhet, a változók köthetők az egyes szeletekhez.
 - eredmény generálása*
- Az XQuery szerepe az XML adatbázisok világában olyan, mint az SQL szerepe a relációs adatbázisok esetében, de az SQL relációs, míg XML hierarchikus. Segítségével lekérdezhajjuk, módosíthatjuk adatunkat és új elemeket adhatunk hozzá.
- A gráfban való navigáláshoz XPath kifejezéseket használunk.
- Az XQuery ugyanazt az adatmodellt használja, mint a korábban megismert XPath, vagyis tételek (items) szekvenciáit kezeli és a függvényeik is közösek.



8. ábra: XQuery folyamatára (forrás: <http://prog.hu>)

- Funkcionális nyelv, vagyis tetszőleges XQuery kifejezést szerepelhet bármilyen olyan helyen, ahol kifejezés állhat, amelyek aztán kiértékelődnek.

6.2 Kifejezések

- Elsődleges kifejezések: literál, változó vagy egy függvény alkalmazása, a változó \$-jellel kezdődik.
- A sztring literálok aposztrófok vagy idézőjelek közt szerepelnek: 'alma', "alma".
- XPath kifejezések: XQuery ezen kifejezéseket használja, hogy navigáljon az XML dokumentumban szereplő elemek között.
- FLWOR kifejezés
- Szekvencia: A , (vessző) konstruktor sorozatokat generál. A sorozatok nem ágyazhatóak egymásba, azaz az (1, (2, 3), 4) sorozat egyenértékű az (1, 2, 3, 4) sorozattal.
- Direkt (literális) konstrukciók: az XML szintaktikával megadott elemek XML elemekként is értelmeződnek: <a>5.
- Dinamikus konstrukciók, amelyek futásidőben értékelődnek ki.
- Feltételes kifejezések

Doc függvény

A doc függvénnyel tetszőleges dokumentumot "olvashatunk be".

Alakja: doc(dokumentum_név.xml).

```
doc("R159N210668586.xml")  
doc("R159N210668586.xml")/Jelentes/Tabla/Sor[Sorkod='BEFK3AFK0010']
```

Flwor kifejezés

Jelentése egy mozaikszó, amely összefoglaló neve a For, Let, Where, Order by, Return záradékoknak.

```
for $x in doc("R199N210668586.xml")/Jelentes/Tabla[@Tablakod="TAJ"]/Sor
where $x/Cellaadat>3000
order by $x/Sorkod
return $x/Cellaadat | $x/Sorkod
```

A **for** záradék : Az XML dokumentumból kiválasztja az XPath kifejezésnek megfelelő Sor elemeket, mely szülőjének (TABLA) a Tablakod attribútuma „TAJ” és belepakolja az x változóba. Tablakod=„TAJ” szűkítésre azért volt szükség, mert így kiestek azon Cellaadat elemek melynek típus String, különben hibával térne vissza.

A **where** záradék: az SQL WHERE záradékához hasonlóan "szűrő" feltételeket tartalmazhat. Kiválasztja azon Cellaadat elemeket melynek értéke 3000 nagyobb.

Az **order by** záradék (opcionális) használatával az eredmény a megadott elem szerint rendezetten jelenik meg.

A **return** segítségével adhatjuk meg a lekérdezés végeredményét, mely elem vagy elemekkel térjen vissza a lekérdezésünk. Ez az egyetlen kötelező záradék

Egy Flwor kifejezés eredménye – hasonlóan bármely XQuery kifejezéshez – adatok szekvenciája. Az adatok szekvenciáját a return - záradékhoz csatolt kifejezés állítja elő az addig előállított adatok szekvenciájából.

Let záradék

Alakja: let *változó* := *kifejezés*

A let záradék (opcionális) segítségével szintén szekvenciákat rendelhetünk a változókhoz, ám nem történik semmiféle iteráció. Abban különbözik a **for**-tól, hogy a változó nem egyenként veszi fel a hozzárendelt kifejezés összes értékét, hanem a kifejezés összes értékét egyszerre rendeli a változóhoz. Használható a **for** kulcsszóval együtt vagy nélküle is. Ha nincs **for** a

kifejezésben, akkor a változó csak egy értéket vesz fel, mely valójában a hozzárendelt kifejezés összes értékének a sorozata.

```
let $x := doc("R199N210668586.xml")//Cellaadat
return $x
```

Egy lekérdezés elején tetszőleges számú for és let záradék szerepelhet tetszőleges sorrendben, de egy let vagy for záradéknak szerepelnie kell. A lekérdezést pontosan egy return záradéknak kell zárnia!

Elágazás

Alakja: *if (kifejezés1) then kifejezés2 else kifejezés3*

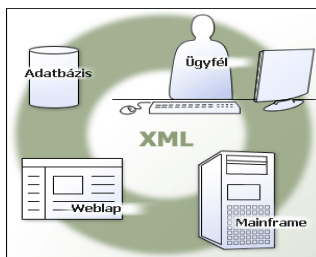
Az else ág megadása kötelező. Viszont, ha utána az üres szekvencia: () szerepel, akkor olyan, mintha nem lenne else ág.

```
for $keszito in doc("R199N210668586.xml")//Sor
  return
    if (contains($keszito/Cellaadat/text(),"Szűcs") or
contains($keszito/Cellaadat/text(),"Béla"))
      then $keszito
      else $keszito/text()
```

Az XQuery erősebben támogatott a nagy szoftvergyártók részéről is, ami elősegítheti elterjedését, így az XQuery egy erős, jól használható, a többi szabványhoz jól illeszkedő nyelv lesz.

7. Összefoglalás

Szakdolgozatomban betekintést nyújtottam az XML felépítésébe, a séma készítés szabályaiba, valamint a ráépülő szolgáltatások (XPath, XQuery) alkalmazásába. Munkahelyi feladatom - melyet a diplomamunkám témájává is választottam - hogy a 2008-ba életbelépő törvény értelmében egy bizonyos árbevétel meghaladó kereskedelmi cégek kötelesek negyedévenként a mérleg szerinti saját tőke, illetve a devizakülfölddel szemben követelések és kötelezettségek változásairól jelentést készíteni a Magyar Nemzeti Bank felé. Az MNB által kívánt jelentés elkészítéséhez ideális választás az XML nyelv.



Az XML eredetileg webes adatcserére lett tervezve. Jelenleg nyilvános és széles körben elfogadott szabványa a legkülönbözőbb rendszerek közötti adatcserének. Platform független ez azt jelenti, hogy az XML használatát támogató bármely program képes olvasni és feldolgozni az XML-adatokat, függetlenül a számítógépek operációs rendszerétől és hardvereitől. Ez a képesség lehetővé teszi, hogy az asztali alkalmazásokkal, amit legtöbb felhasználó használ munkában vagy otthon, például a Microsoft Word vagy a Microsoft Excel alkalmazással közvetlenül adatbázisból lekérdezett adatokon dolgozzon.

Ezen lehetőséget használtam fel a dolgozatomban a jelentés elkészítése során. Az adatbázis rendszerben tárolva vannak a szükséges pénzügyi adatok, azonban nem XML formátumban, így ki kell választani, mire volt szükségem, és azt egy lekérdezés során munkafüzetbe kellett importálni. Ezután elkészítenem az XML leíráshoz a DTD-t, illetve a XML sémafájlt a megfeleltetésekhez, majd a bevitt adatok alapján elkészülhetett a feltölteni kívánt két XML fájl. XML-megfeleltetések használatával egyszerűen sikerült üzleti adatokat felvenni, azonosítani és kinyerni az Excel-dokumentumokból. Az .xml kiterjesztésű XML-adatfájlok importálásakor és exportálásakor e megfeleltetésekkel köthetők össze a sémában található elemek és a megfeleltetett cellák tartalma. Az XML-adatok exportálása a munkalapon lévő megfeleltetett cellák tartalmának exportálásával történt.

Az XML előnyei közé tartozik továbbá, hogy lehetővé teszi a benne szereplő adatelemek feldolgozását, lekérdezését és megjelenítését az XPath, valamint az XQuery alkalmazásával. Keresés helyett lekérdezéssel juthatunk információhoz, így könnyen alkalmazható, ha nagy terjedelmű – mint jelen esetben - XML fájlokból valamilyen adatra vagy jelentés részre lenne szükségünk elemzés készítéséhez. Véleményem szerint sikerült megvalósítani a kitűzött cél és egy könnyen kezelhető, újra felhasználható rendszert sikerült összeállítanom a jelentések elkészítéséhez valamint adatok eléréséhez, lekérdezéséhez.

Meg kell azonban említeni az XML-nek egy másik, de igen fontos alkalmazási lehetőségét a szemantikus webet, amely szoros kapcsolatban van az XML-el, illetve az XPath, XQuery szolgáltatásokkal. A Szemantikus Web egy olyan fejlesztés, amely a weben található dokumentumokat a számítógép számára értelmezhető jelentéssel, szemantikával látja el. Mivel a HTML-nyelvnek eredeti funkciója szerint nem feladata, hogy tartalmilag feltárja egy weboldal szövegét, hanem elsődlegesen a dokumentum képi megjelenítéséért felelős. A szemantikus web ezt a hiányt hivatott kiküszöbölni a testre szabható XML nyelv segítségével. Véleményem szerint a jövő mindenképpen ez, különösen, ha figyelembe vesszük, hogy a Microsoft és egyre több másik gyártó is támogatja az XML-t, így elterjedése és használata még egyszerűbb lehet. Ez azt jelenti, hogy az idők során az XML és hozzá kapcsolódó szolgáltatások használata egyre költségkímélőbbé válik.

8. Irodalomjegyzék

Felhasznált irodalom:

- Michael Morrison: Tanuljuk meg az XML használatát 24 óra alatt (Kiskapu könyvkiadó Kft., Budapest 2006)
- Eliotte Rusty Harold: Hatékony XML (Kiskapu Könyvkiadó Kft., Budapest 2006)
- Neil Bradley: Az XML kézikönyv (Kiskapu könyvkiadó Kft., Budapest 2005)
- G. Ken Holman: Definitive XSLT and XPath (PrenticeHall, 2005)
- Michael J. Young: XML Lépséről Lépésre (Kiskapu Könyvkiadó Kft., Budapest 2002)
- Priscilla Walmsley: XQuery (Priscilla Walmsley: XQuery (O'Reilly Media Inc, 2007)

Internet címek:

Technológia megismerésében segítségemre volt a W3schools megfelelő oldalai:

- <http://www.w3schools.com/xml/default.asp>
- <http://www.w3schools.com/dtd/default.asp>
- <http://www.w3schools.com/xpath/default.asp>
- <http://www.w3schools.com/xquery/default.asp>

Microsoft Office Help

- <http://office.microsoft.com/hu-hu/help/HA100340221038.aspx>

Prog.hu

- <http://prog.hu/cikkek/sorozatok/Bevezet%E9s+az+XML-be.html>

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani konzulensemnek, Dr. Adamkó Attila egyetemi adjunktusnak, aki segítette dolgozatom elkészítését.

9. Függelék

Az EBEAD-ba feltöltött R199N210668586.xml nevű fájl

```
<?xml version="1.0" encoding="utf-8"?>
<Jelentes>
  <AdatgyujtesKod>R19</AdatgyujtesKod>
  <AdatszolgaltatoAzonosito>10668586</AdatszolgaltatoAzonosito>
  <Gyakorisag>Negyedebes</Gyakorisag>
  <VonatkozasiIdoTol>2009-04-01</VonatkozasiIdoTol>
  <VonatkozasiIdoIg>2009-06-30</VonatkozasiIdoIg>
  <AdatkuldoAzonosito>10668586</AdatkuldoAzonosito>
  <Tabla Tablakod="ELOLAP">
    <Sor>
      <Sorkod>ELOLAP03</Sorkod>
      <Cellaadat>szucsbelakite.hu</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP02</Sorkod>
      <Cellaadat>54/480-401/353</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP05</Sorkod>
      <Cellaadat>54/480-401/784</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP04</Sorkod>
      <Cellaadat>Nyitrai Ibolya</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP07</Sorkod>
      <Cellaadat>2009-07-30</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP06</Sorkod>
      <Cellaadat>nyitraiibolyakite.hu</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>ELOLAP01</Sorkod>
      <Cellaadat>Szúcs Béla</Cellaadat>
    </Sor>
  </Tabla>
  <Tabla Tablakod="TAJ">
    <Sor>
      <Sorkod>TAJ11</Sorkod>
      <Cellaadat>0</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>TAJ12</Sorkod>
      <Cellaadat>913</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>TAJ04</Sorkod>
      <Cellaadat>1776</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>TAJ07</Sorkod>
      <Cellaadat>1489</Cellaadat>
    </Sor>
    <Sor>
      <Sorkod>TAJ13</Sorkod>
      <Cellaadat>17368</Cellaadat>
    </Sor>
  </Tabla>
</Jelentes>
```

```

</Sor>
<Sor>
  <Sorkod>TAJ01</Sorkod>
  <Cellaadat>31813</Cellaadat>
</Sor>
<Sor>
  <Sorkod>TAJ03</Sorkod>
  <Cellaadat>41124</Cellaadat>
</Sor>
<Sor>
  <Sorkod>TAJ02</Sorkod>
  <Cellaadat>486</Cellaadat>
</Sor>
<Sor>
  <Sorkod>TAJ05</Sorkod>
  <Cellaadat>5</Cellaadat>
</Sor>
<Sor>
  <Sorkod>TAJ09</Sorkod>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>TAJ08</Sorkod>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>TAJ06</Sorkod>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>TAJ10</Sorkod>
  <Cellaadat>14966</Cellaadat>
</Sor>
</Tabla>
</Jelentes>

```

Az EBEAD-ba feltöltött R159N210668586.xml nevű fájl

```

<?xml version="1.0" encoding="utf-8"?>
<Jelentes>
  <AdatgyujtesKod>R15</AdatgyujtesKod>
  <AdatszolgaltatoAzonosito>10668586</AdatszolgaltatoAzonosito>
  <Gyakorisag>Negyedebes</Gyakorisag>
  <VonatkozasiIdoTol>2009-04-01</VonatkozasiIdoTol>
  <VonatkozasiIdoIg>2009-06-30</VonatkozasiIdoIg>
  <AdatkuldoAzonosito>10668586</AdatkuldoAzonosito>
  <Tabla Tablakod="BEFK1AFK">
    <Nemleges>igen</Nemleges>
  </Tabla>
  <Tabla Tablakod="BEFK2AFK">
    <Nemleges>igen</Nemleges>
  </Tabla>
  <Tabla Tablakod="BEFK3AFK">
    <Sor>
      <Sorkod>BEFK3AFK0014</Sorkod>
      <Cellaadat>KERHITK</Cellaadat>
      <Cellaadat>R</Cellaadat>
      <Cellaadat>SK</Cellaadat>
      <Cellaadat>EUR</Cellaadat>
      <Cellaadat>0</Cellaadat>
      <Cellaadat>0</Cellaadat>
      <Cellaadat>0</Cellaadat>
      <Cellaadat>0</Cellaadat>
    </Sor>
  </Tabla>

```

```

    <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFK3AFK0013</Sorkod>
  <Cellaadat>KERHITK</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>SI</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFK3AFK0009</Sorkod>
  <Cellaadat>KERHITK</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>IT</Cellaadat>
  <Cellaadat>USD</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFK3AFK0010</Sorkod>
  <Cellaadat>KERHITK</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>NL</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>327499</Cellaadat>
  <Cellaadat>249723</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>577222</Cellaadat>
  <Cellaadat>577222</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFK3AFK0012</Sorkod>
  <Cellaadat>KERHITK</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>RO</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>19335</Cellaadat>
  <Cellaadat>-17197</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>2138</Cellaadat>
  <Cellaadat>2138</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFK3AFK0006</Sorkod>
  <Cellaadat>KERHITK</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>FR</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFK3AFK0005</Sorkod>
  <Cellaadat>KERHITK</Cellaadat>

```

```

<Cellaadat>R</Cellaadat>
<Cellaadat>DE</Cellaadat>
<Cellaadat>EUR</Cellaadat>
<Cellaadat>7657312</Cellaadat>
<Cellaadat>-7556399</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>100913</Cellaadat>
<Cellaadat>100913</Cellaadat>
</Sor>
<Sor>
<Sorkod>BEFK3AFK0011</Sorkod>
<Cellaadat>KERHITK</Cellaadat>
<Cellaadat>R</Cellaadat>
<Cellaadat>NL</Cellaadat>
<Cellaadat>USD</Cellaadat>
<Cellaadat>862500</Cellaadat>
<Cellaadat>-546250</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>316250</Cellaadat>
<Cellaadat>316250</Cellaadat>
</Sor>
<Sor>
<Sorkod>BEFK3AFK0007</Sorkod>
<Cellaadat>KERHITK</Cellaadat>
<Cellaadat>R</Cellaadat>
<Cellaadat>GR</Cellaadat>
<Cellaadat>EUR</Cellaadat>
<Cellaadat>130212</Cellaadat>
<Cellaadat>-130212</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
</Sor>
<Sor>
<Sorkod>BEFK3AFK0002</Sorkod>
<Cellaadat>KERHITK</Cellaadat>
<Cellaadat>R</Cellaadat>
<Cellaadat>BA</Cellaadat>
<Cellaadat>EUR</Cellaadat>
<Cellaadat>27939</Cellaadat>
<Cellaadat>14279</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>42218</Cellaadat>
<Cellaadat>42218</Cellaadat>
</Sor>
<Sor>
<Sorkod>BEFK3AFK0001</Sorkod>
<Cellaadat>KERHITK</Cellaadat>
<Cellaadat>R</Cellaadat>
<Cellaadat>AT</Cellaadat>
<Cellaadat>EUR</Cellaadat>
<Cellaadat>850516</Cellaadat>
<Cellaadat>-648250</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>202266</Cellaadat>
<Cellaadat>202266</Cellaadat>
</Sor>
<Sor>
<Sorkod>BEFK3AFK0008</Sorkod>
<Cellaadat>KERHITK</Cellaadat>
<Cellaadat>R</Cellaadat>
<Cellaadat>IT</Cellaadat>
<Cellaadat>EUR</Cellaadat>
<Cellaadat>1017242</Cellaadat>
<Cellaadat>-1015610</Cellaadat>

```

```

    <Cellaadat>0</Cellaadat>
    <Cellaadat>1632</Cellaadat>
    <Cellaadat>1632</Cellaadat>
</Sor>
<Sor>
    <Sorkod>BEFK3AFK0003</Sorkod>
    <Cellaadat>KERHITK</Cellaadat>
    <Cellaadat>R</Cellaadat>
    <Cellaadat>BE</Cellaadat>
    <Cellaadat>EUR</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
    <Sorkod>BEFK3AFK0004</Sorkod>
    <Cellaadat>KERHITK</Cellaadat>
    <Cellaadat>R</Cellaadat>
    <Cellaadat>BG</Cellaadat>
    <Cellaadat>EUR</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
</Sor>
</Tabla>
<Tabla Tablakod="BEFK4AFK">
    <Nemleges>igen</Nemleges>
</Tabla>
<Tabla Tablakod="BEFK5AFK">
    <Nemleges>igen</Nemleges>
</Tabla>
<Tabla Tablakod="BEFT1AFK">
    <Nemleges>igen</Nemleges>
</Tabla>
<Tabla Tablakod="BEFT3AFK">
    <Sor>
        <Sorkod>BEFT3AFK0013</Sorkod>
        <Cellaadat>KERHITT</Cellaadat>
        <Cellaadat>R</Cellaadat>
        <Cellaadat>IE</Cellaadat>
        <Cellaadat>EUR</Cellaadat>
        <Cellaadat>0</Cellaadat>
        <Cellaadat>0</Cellaadat>
        <Cellaadat>0</Cellaadat>
        <Cellaadat>0</Cellaadat>
    </Sor>
    <Sor>
        <Sorkod>BEFT3AFK0014</Sorkod>
        <Cellaadat>KERHITT</Cellaadat>
        <Cellaadat>R</Cellaadat>
        <Cellaadat>IL</Cellaadat>
        <Cellaadat>EUR</Cellaadat>
        <Cellaadat>0</Cellaadat>
        <Cellaadat>22828</Cellaadat>
        <Cellaadat>0</Cellaadat>
        <Cellaadat>22828</Cellaadat>
    </Sor>
    <Sor>
        <Sorkod>BEFT3AFK0023</Sorkod>
        <Cellaadat>KERHITT</Cellaadat>
        <Cellaadat>R</Cellaadat>

```

```
<Cellaadat>TR</Cellaadat>
<Cellaadat>USD</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0020</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>SE</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0024</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>US</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>12377</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>12377</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0001</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>AT</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>7719</Cellaadat>
  <Cellaadat>-2228</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>5491</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0002</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>BE</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>30307</Cellaadat>
  <Cellaadat>3022</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>33329</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0003</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>CZ</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>82397</Cellaadat>
  <Cellaadat>510591</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>592988</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0004</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
```

```
<Cellaadat>R</Cellaadat>
<Cellaadat>DE</Cellaadat>
<Cellaadat>EUR</Cellaadat>
<Cellaadat>5260563</Cellaadat>
<Cellaadat>3028670</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>8289233</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0005</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>DE</Cellaadat>
  <Cellaadat>USD</Cellaadat>
  <Cellaadat>127078</Cellaadat>
  <Cellaadat>131824</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>258902</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0006</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>DK</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0007</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>EE</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>7525</Cellaadat>
  <Cellaadat>-3256</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>4269</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0008</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>ES</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>38157</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>38157</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0009</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>FR</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>409748</Cellaadat>
  <Cellaadat>-131061</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>278687</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0011</Sorkod>
```

```
<Cellaadat>KERHITT</Cellaadat>
<Cellaadat>R</Cellaadat>
<Cellaadat>GB</Cellaadat>
<Cellaadat>GBP</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
<Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0010</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>GB</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>15035</Cellaadat>
  <Cellaadat>252074</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>267109</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0022</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>TR</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0012</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>GR</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>0</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0015</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>IT</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>499187</Cellaadat>
  <Cellaadat>-386643</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>112544</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0021</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>SK</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>18719</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>18719</Cellaadat>
</Sor>
<Sor>
```

```

    <Sorkod>BEFT3AFK0017</Sorkod>
    <Cellaadat>KERHITT</Cellaadat>
    <Cellaadat>R</Cellaadat>
    <Cellaadat>NL</Cellaadat>
    <Cellaadat>USD</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
    <Cellaadat>0</Cellaadat>
  </Sor>
<Sor>
  <Sorkod>BEFT3AFK0016</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>NL</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>105319</Cellaadat>
  <Cellaadat>-80502</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>24817</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0019</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>RO</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>153005</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>153005</Cellaadat>
</Sor>
<Sor>
  <Sorkod>BEFT3AFK0018</Sorkod>
  <Cellaadat>KERHITT</Cellaadat>
  <Cellaadat>R</Cellaadat>
  <Cellaadat>PL</Cellaadat>
  <Cellaadat>EUR</Cellaadat>
  <Cellaadat>94064</Cellaadat>
  <Cellaadat>-60024</Cellaadat>
  <Cellaadat>0</Cellaadat>
  <Cellaadat>34040</Cellaadat>
</Sor>
</Tabla>
<Tabla Tablakod="BEFT4AFK">
  <Nemleges>igen</Nemleges>
</Tabla>
<Tabla Tablakod="BEFT5AFK">
  <Nemleges>igen</Nemleges>
</Tabla>
<Tabla Tablakod="ELOLAP">
  <Sor>
    <Sorkod>ELOLAP03</Sorkod>
    <Cellaadat>szucsbel@kite.hu</Cellaadat>
  </Sor>
  <Sor>
    <Sorkod>ELOLAP02</Sorkod>
    <Cellaadat>54/480-401/353</Cellaadat>
  </Sor>
  <Sor>
    <Sorkod>ELOLAP05</Sorkod>
    <Cellaadat>54/480-401/784</Cellaadat>
  </Sor>
  <Sor>
    <Sorkod>ELOLAP04</Sorkod>

```

```
<Cellaadat>Nyitrai Ibolya</Cellaadat>
</Sor>
<Sor>
  <Sorkod>ELOLAP07</Sorkod>
  <Cellaadat>2009-07-29</Cellaadat>
</Sor>
<Sor>
  <Sorkod>ELOLAP06</Sorkod>
  <Cellaadat>nyitraiibolya@kite.hu</Cellaadat>
</Sor>
<Sor>
  <Sorkod>ELOLAP01</Sorkod>
  <Cellaadat>Szűcs Béla</Cellaadat>
</Sor>
</Tabla>
</Jelentes>
```