Mérési jegyzőkönyv – Adatbázisok Laboratórium

5. mérés: XML

|  |  |
| --- | --- |
| Név: | **Szabó Bence Farkas** |
| Neptun kód: | **RF57V5** |
| Feladat kódja: | **26 - SZALL** |
| Mérésvezető neve: | **Böjti Bence** |
| Mérés időpontja: | **2017-04-20 16:15** |
| Mérés helyszíne: | **R4K** |
| *A működő alkalmazás elérhetősége:* | *rapid.eik.bme.hu/~rf57v5/xml/* |
| Megoldott feladatok: | **1,2,3,4** |
| Elérhető pontszám (plusz pontok nélkül): | **28** |

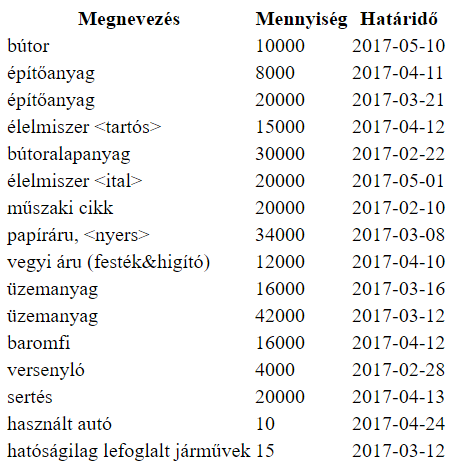
# Mérési feladatok megoldása

## feladat

Megrendelések kiíratása

### Magyarázat

Az első feladatban a megrendelések adatait kellett kiíratni XSL transzformációval XML formátumba. Ezt a list.xsl állományban valósítottam meg. A feladat megoldásához jó alapul szolgált a kiadott példaprogram, melyen nem is kellett sokat módosítani. Eleinte problémák voltak a dátum megjelenítésével, de erre a segédletben találtam megoldást. A dátum két attribútumot tartalmaz, ezek a date és time. Mivel nekünk csak a date-re van szükségünk, a sablonban azt kell kiválsztani. A dátum formátumát nem kellett módosítani, mivel az már helyesen volt tárolva a bemeneti xml-ben. A megoldáshoz a mintaprogramhoz hasonlóan több XSL sablont használtam, melyek rekurzívan hajtják végre a transzformációt. A vezérlő script a kimenetet a list.html állományba teszi. Ez az alábbi kimenetet adja:



## feladat

Adatok szűrése

### Magyarázat

Ebben a feladatban a 19000-nél nagyobb mennyiségű megrendelések adatait kellett kiírni XML file-ba. Ehhez a példaprogramban található két sablon egészítettem ki úgy, hogy az első sablon kiválasztja a bemeneti XML állományból az orders csomópontokat, majd a második lemásolja azokat és a gyerekeiket. Itt a segédletben is megemlített **xsl:copy-of** elemet felhasználva, annak select attribútumában megadtam a feltételt. Ennek használatát az interneten talált anyagokból fejtettem meg[[1]](#footnote-1). A megoldás a filter.xsl állományban található, a kimenet a fenti link /filter.xml kiegészítésén található, melyből lentebb látható egy részlet.

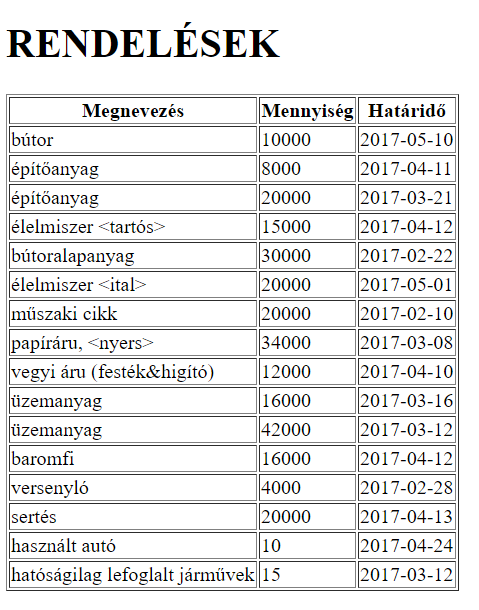
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <szall element-type="dataset">  <orders element-type="recordset">  <record element-type="record" order\_id="2017/000003">  <description is-null="False">építőanyag</description>  <vehicle\_type is-null="False">D</vehicle\_type>  <quantity is-null="False">20000</quantity>  <origin is-null="False">Milánó (IT)</origin>  <destination is-null="False">Szeged (HU)</destination>  <order\_date is-null="False" date="2017-01-14" time="00:00:00"/>  <deadline\_date is-null="False" date="2017-03-21" time="12:00:00"/>  <comment\_text is-null="True"/>  </record>  <record element-type="record" order\_id="2017/000005">  <description is-null="False">bútoralapanyag</description>  <vehicle\_type is-null="False">D</vehicle\_type>  <quantity is-null="False">30000</quantity>  <origin is-null="False">Varsó (PL)</origin> |

stb…

## feladat

1. Az index.html lap módosítása

### Magyarázat

A feladat során az index.html oldal tartalmát módosítjuk a megrendelések adataival a list.xsl állományt felhasználva. Ebben a feladatban is a példaprogramot írtam át illetve egészítettem ki. Ahhoz, hogy a megfelelő adatokat kinyerhessük, a sablon gyökérelemének az „//orders” Xpath kifejezést használtam, mely minden orders csomópontra illeszkedik, így minden megrendelés adatait kinyerhetjük. Ezek után már csak egy táblázatot és annak fejlécét kellett kialakítani, illetve meghívni a list.xsl sablonjait, melyek feltöltik a táblázat sorait. A táblázatnak a szebb és áttekinthetőbb megjelenés érdekében egy szegélyt is adtam. A stíluslap a feldolgozott adatokat html formátumban írja ki a kimenetre, melyet a vezérlés az index.html állományba tesz. Ez a fenti link megnyitásával tekinthető meg (illetve a lenti ábrán is látható a kimenet).

## feladat

## Táblák összekapcsolása.

### Magyarázat

A feladatban a megrendelések listázásában hozzá kellett csatolni a megrendeléshez tartozó fuvarokat. A megoldáshoz szinté jó alapul szolgált a kiadott példaprogram, melyet átalakítottam a feladatnak megfelelően. Először az első sablonban változattam meg a sablon paramétereit, hogy az orders tábla adatait dolgozza fel. Ezek után a megrendelés adatainak kiíratása következett. Itt a legnagyobb nehézséget a shipments tábla adatainak elérése jelentette. Ezt a laborvezetőtől kapott útmutatás segítségével tudtam megoldani, miszerint egy változóban tároljuk az aktuális megrendelés azonosítóját, majd ez alapján alkalmazzuk a következő sablont.

**<xsl:variable** name="my\_id" select="@order\_id" />

**<xsl:apply-templates** select="//shipments/record[order\_id=$my\_id]"/>

Itt a shipments record elemeit akkor dolgozzuk fel, ha a shipment-ben tárolt order\_id azonosító megegyezik az változónkban tárolt értékkel.

Ezek után a szállítmány adatainak listázása nem volt nehéz, itt egy harmadik sablont kellett megírni, hasonlóan az előzőhöz.

A vezérlő skript a kimenetet a join.xml állományba teszi, mely a fenti link /join.xml kiegészítésén található, illetve lentebb látható belőle egy részlet.

<orders xmlns:functx="http://www.functx.com" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<order id="2017/000001">

<description>bútor</description>

<quantity>10000</quantity>

<deadline\_date>2017-05-10</deadline\_date>

<shipments>

<shipment vehicle\_id="0" arrival\_date="2017-01-06"/>

<shipment vehicle\_id="0" arrival\_date="2017-01-07"/>

</shipments>

</order>

<order id="2017/000002">

<description>építőanyag</description>

<quantity>8000</quantity>

<deadline\_date>2017-04-11</deadline\_date>

<shipments>

<shipment vehicle\_id="0" arrival\_date="2017-01-09"/>

</shipments>

</order>

<order id="2017/000003">

<description>építőanyag</description>

<quantity>20000</quantity>

<deadline\_date>2017-03-21</deadline\_date>

<shipments>

<shipment vehicle\_id="2" arrival\_date="2017-01-15"/>

<shipment vehicle\_id="2" arrival\_date="2017-01-16"/>

</shipments>

1. https://www.w3schools.com/xml/ref\_xsl\_el\_copy-of.asp [↑](#footnote-ref-1)