# *Adattárház és jelentéskészítés OLAP-pal a Pentaho Mondrian és JPivot nyílt forráskódú eszközök használatával*

## Bevezetés

Jelen cikk betekintést ad az adattárházak és a többdimenziós adatmodell alapfogalmaiba, valamint konkrét példát is mutat az ingyenes, nyílt forráskódú Pentaho Mondrian Java nyelven implementált OLAP (online analitical processing) szerver, valamint a JPivot JSP custom tag library felhasználásával, mely OLAP táblát jelenít meg interaktív formában, mellyel a legtöbb OLAP művelet elvégezhető. A cikknek nem célja a fogalmak és technológiák alapos, precíz ismertetése, inkább egy bevezetőt nyújt ezen eszközök használatába, egy gyakorlati példán keresztül.

## Előélet

Képzeljük el, hogy olyan feladatot kapunk, ahol a megrendelő (akit egy üzleti döntéshozó személyesít meg) egy bonyolult komplex lekérdezéseket összeállítani képes, ennek futtatását és megjelenítését végző rendszert rendel meg. Hamar megállapítjuk, hogy nem elegendő előre gyártott riportok megfogalmazása, mert szükség van arra, hogy interaktív legyen, ad-hoc módon lehessen benne lekérdezéseket megfogalmazni, csoportműveleteket, aggregációkat végezni. A problémát meg lehet közelíteni felhasználói felület oldalról, ahol hamar beleütközhetünk abba, hogy egy interaktív nem csak informatikusok számára használható felület megtervezése nem egyszerű feladat. Ha a megvalósítás oldaláról közelítjük meg, hamar rájöhetünk, hogy az interfészen végzett műveletek alapján nekünk kell dinamikusan összeállítani (pl. relációs adatbázis esetén az SQL), manipulálni lekérdezéseket. Ha esetleg mindkettő sikerülne, akkor is nehézkes lehet a programunk továbbfejlesztése, valamint nagy adathalmaz esetén a sebessége.

Szerencsére nem kell kétségbe esnünk, vannak már ingyenes, nyílt forráskódú eszközök is ilyen feladatok megoldására, egy ilyent fog bemutatni ez a cikk is.

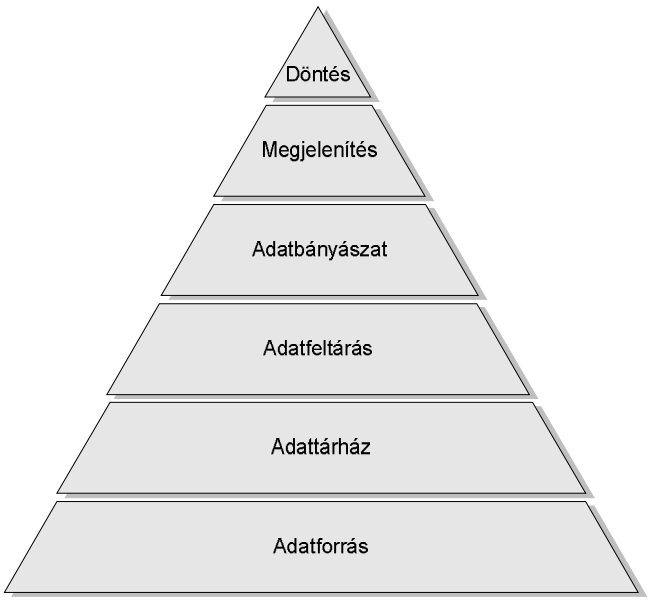
## Alapfogalmak

Napjainkban hatalmas adat- és információrobbanás tanúi lehetünk, hiszen új adatok, adattípusok, eszközök, törvények (pl. archiválásra vonatkozó) jelennek meg, és az adathordozók ára is rohamosan csökken. Szinte minden tevékenységünkről rögzül valamilyen adat egy számítógépes rendszerben, legyen az egy üzletben történő vásárlás, Internetes böngészésünk, banki tranzakcióink. Ezen adatok ezen rendszerekben nyers adatként jelennek meg, melyek viszonylag kevés információt hordoznak önmagukban. Gondoljunk például arra, hogy egy nagy hipermarket napi forgalmának eltárolása mekkora adatmennyiség. Ezen feldolgozatlan adatok azonban még nem alkalmasak arra, hogy az adott üzletlánc vezetői következtetéseket vonjanak le. John Naisbitt szerint „Megfulladunk az adatoktól, miközben tudásra éhezünk.” Az adat életútja két irányban folytatódhat. Eltűnhet valamilyen adattemetőben, és soha többé nem lesz visszakeresve, vagy valamilyen feldolgozás során a többi adatokkal együtt, azokkal aggregálva, összegezve tudássá válik. Tudássá, melyet csak így lehet megszerezni, tudássá, mely alapján döntéseket lehet hozni.

Az üzleti intelligencia (business intelligence - BI) egyre nagyobb szerepet kap, hiszen ez azon szakértelmek, technikák, alkalmazások, bevált gyakorlatok, és az ezek alapján előállt úgy információk összessége, melyek segítenek az üzleti környezet megértésében. Ezeknek a rendszereknek a felhasználói nem informatikusok, programozók, hanem vállalatvezetők, döntéshozók, akik az adatokból tudást akarnak megszerezni, hogy ezek segítsék az üzleti

döntéseiket. Ezen rendszerek a ’80-as évek vezetői információs és döntéstámogató rendszereikből nőtték ki magukat, és manapság egyre többet hangoztatott fogalom az üzleti teljesítménynövelés (business performance management - BPM) is, mely már indikátorok segítségével (key process indicator - KPI) méri is az üzleti hatékonyságot. Ezek alapján, ezek segítségével üzleti célok fogalmazhatóak meg (pl. 5%-kal növeljük az eladást), és ezek elérésére üzleti folyamatokat kell kidolgozni.

A döntés támogatására különböző szintű eszközök állnak rendelkezésre, melyek egymásra épülnek.




**1. ábra Döntéstámogatás eszközei**

A legalsó szinten az adatforrások helyezkednek el, melyek a nyers, feldolgozatlan adatokat tartalmazzák heterogén tárolási módon és formában. Lehetnek papír alapú és elektronikus iratok, adatbázisok, melyek a cégen belül, de kívül is elhelyezkedhetnek.

Az adatok alapvetően a következő forrásokból származhatnak:

- Cégen kívüli adatforrások
- Adatpiacok (lsd. később)
- OLTP (online transaction processing) rendszerek: klasszikus, adatbázison alapuló, napi működést biztosító rendszerek

Az adatok innét az adattárházba kerülnek, mely egy elemzési és lekérdezési céllal létrehozott speciális adatbázis, mely egységesített, megtisztított és értékes adatokat tartalmaz, hisztorikusan. Az adatok az adattárházba általában ETL (extract – transform - load) eszköz felhasználásával kerülnek be, mely az adatforrásból kinyeri az adatokat, transzformálja, majd betölti őket az adattárházba. Újabban megjelentek az ELT eszközök is (nagy adatbázisgyártóknál), ahol a transzformáció is az adatbázisban történik, szabályozott módon. Az adatpiac egy kisebb adattárház, és általában csak egy-egy témával kapcsolatos adatokat tartalmaz, pl. egy cég egyik osztályának működése során előállt adatokat.

A következő tevékenység az adatfeltárás, mely általában (de nem kötelezően) az adattárházra épül, és célja a meglévő adatokon lekérdezések, jelentések és statisztikai elemzések futtatása.

Az adatbányászat már olyan technológiák, eszközök gyűjteménye, melyek segítségével új információk állíthatóak elő, sőt, előrejelzések is készíthetők.

Az előállt adatokat a döntéshozók felé egyszerű, könnyen emészthető formában kell átadni. Itt mivel nagymennyiségű, időben változó adatokról van szó, megnőtt a szerepe a különböző vizualizációs technikáknak, valamint a megfelelő szintű interakció biztosításának. Az így kinyert és előállt adatokat lehetőleg egy egységes felületen kell megmutatni, ami azonnali rálátást biztosít a főbb információkra (un. dashboard - irányítópult), amiket aztán alá lehet bontani, un. „lefúrást” lehet elvégezni. Hasznos, ha ezen felületek szervesen beépülnek a vállalati portál felületébe.

# Adattárházak, OLAP

Ezen cikk a felsorolt eszközök közül főleg az adattárházakra, az ezekkel kapcsolatos fogalmakra és az OLAP-ra koncentrál. Ahogy említettem, az adattárház elemzési és lekérdezési céllal létrehozott speciális adatbázis. Az OLAP egy olyan megközelítés, mely használatával gyorsan lehet elemzési kérdésekre válaszolni.

A cikk során a könnyebb megértés érdekében egy egyszerű példát fogok bemutatni, méghozzá egy ingatlanközvetítő ügyviteli rendszerből kinyert adatok alapján felépített OLAP rendszert. Adottak az ingatlanközvetítő irodák, a hozzájuk tartozó ügynökökkel, valamint az eladó ingatlanok.

Az ingatlanközvetítő cég vezetőinek bizonyos üzleti döntések meghozatalához szükségük van olyan adatokra, melyekből kiderül, hogy milyen ingatlanok kerültek eladásra a közreműködésükkel, melyek a legeredményesebb irodák és ügynökök, valamint az ingatlanokat milyen áron sikerült értékesíteni.

Ezen igényeknek megfelelően ezen kimutatások adatköre sokkal szűkebb, mint a teljes rendszeré, tehát az OLTP rendszerből csak az adatok bizonyos részhalmazát kell átemelni az OLAP rendszerbe.

Az irodáról és ügynökről elegendő a nevet megjeleníteni, ingatlanról típusát (pl. tégla lakás, panel lakás, ház), új-e, vagy használt, valamint az ingatlan elhelyezkedését (egyszerűség kedvéért a megyét és a várost). Az ajánlatról az ajánlat idejét és az irányárat tartalmazza, az eladásról az eladás idejét és a konkrét árat.

Az adatokat egy Microsoft Excel táblázatban a következőképpen lehet elképzelni.

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Típus | Új | Ajánlat ideje | Irányár | Eladás ideje | Ár | Iroda | Név | Ország | Megye | Város |
| 2 | 2 | 1 | 2004.06.13 | 90710018 | 2004.07.26 | 89563347 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 3 | 2 | 1 | 2006.03.30 | 21371731 | 2006.09.20 | 19536199 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 4 | 2 | 1 | 2002.03.06 | 20819905 | 2003.02.24 | 16140661 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 5 | 2 | 1 | 2006.02.17 | 74772034 | 2006.08.22 | 74319364 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 6 | 3 | 0 | 2003.03.31 | 57265281 | 2003.10.10 | 53530018 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 7 | 2 | 1 | 2000.04.21 | 52097292 | 2000.06.22 | 47836931 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 8 | 3 | 0 | 2007.09.23 | 31624344 | 2008.08.15 | 30775563 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 9 | 3 | 0 | 2002.02.27 | 14280084 | 2002.12.19 | 13119913 | Budai iroda | Kovács Károly | Magyarország | Budapest | Budapest I. ker. |
| 10 | 2 | 1 | 2006.12.09 | 35319794 | 2007.04.18 | 32906758 | Budai iroda | Szabó Sándor | Magyarország | Budapest | Budapest I. ker. |
| 11 | 1 | 1 | 2004.05.28 | 61673704 | 2005.03.01 | 59936101 | Budai iroda | Szabó Sándor | Magyarország | Budapest | Budapest I. ker. |
| 12 | 3 | 1 | 2000.02.25 | 83271133 | 2000.04.04 | 81813545 | Budai iroda | Szabó Sándor | Magyarország | Budapest | Budapest I. ker. |
| 13 | 1 | 0 | 2001.03.25 | 91244112 | 2001.06.30 | 87550932 | Budai iroda | Szabó Sándor | Magyarország | Budapest | Budapest I. ker. |

**2. ábra Ingatlanközvetítő cég nyers adatai Microsoft Excel táblában**

Az adattárházat összehasonlítva az OLTP rendszerekkel lényeges különbségeket lehet felfedezni. Ezen különbségeket a következő táblázat mutatja.

| **OLTP rendszer adatbázisa** | **Adattárház** |
|---|---|
| Fő célja a napi adatrögzítés, lekérdezés | Döntéshozatal támogatása |
| Aktuális, naprakész adatokat tartalmaz | Történeti, összesített, integrált adatokat tartalmaz |
| Ismétlődő, rövid tranzakciók | Ad-hoc, komplex lekérdezések |
| Írás és olvasás egyaránt | Kizárólag olvasási műveletek, írás csak betöltéskor |
| Egy tranzakcióban átlagosan 10 rekord van érintve | Egy tranzakcióban akár több millió rekord is érintett |
| Mérete több mega vagy esetleg gigabájtos | Mérete akár a terrabájtos nagyságrendet is elérheti |

**1. Táblázat OLTP és adattárház rendszerek összehasonlítása**

A ’90-es évek elején Codd használta az OLAP (online analitical processing) kifejezést olyan módszertanra és követelményrendszerre, mellyel elemző és analitikai rendszerek alakíthatóak ki. Ezen rendszereknek fő feladata megosztott többdimenziós adatok gyors analízise. Alapja a többdimenziós adatszerkezet, melyen különböző lekérdezéseket, interaktív, iteratív felületeken különböző műveleteket lehet elvégezni.

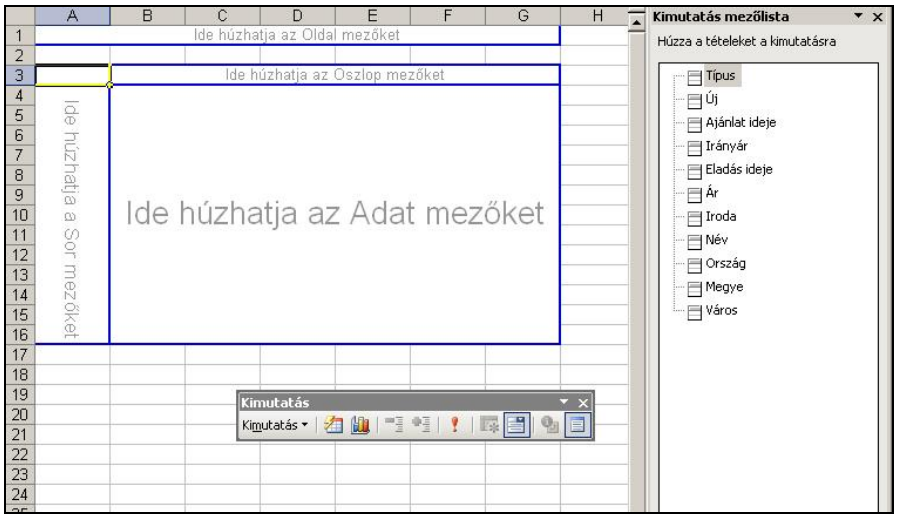
Az adatok egy un. többdimenziós térben jelennek meg. Ezek adatok egyrészt a konkrét értékek, valamint a különböző hierarchiák, melyeken az értékek értelmezve vannak. A hierarchia szintekből áll, és mindegyik szintnek vannak tagjaik.

A példa rendszeren hierarchia pl. az ingatlan elhelyezkedése, ahol a szintek az ország, megye és város. A város szint tagjai Budapest, Vác, Gödöllő, stb. Speciális hierarchia az ajánlat és az eladás ideje, ahol a szintek az év és hó, és a hó szint tagjai a január, február, stb. Konkrét érték az irányár és az eladási ár.

A hierarchia elemeiből és ezek metszéspontjain (cellákban) elhelyezkedő értékekből egy hatalmas, többdimenziós kocka áll elő. Ezen kockán különböző műveleteket lehet végezni, mint a „roll up - felgöngyölítés”, mellyel távolabbról nézzük a kockát, és bizonyos értékeket aggregálunk (ez az aggregálás lehet összeadás, de bármilyen más csoportos művelet is, pl. átlag, stb.). Ezt lehet dimenziók eltüntetésével, vagy dimenziószint tagjainak összevonásával, magasabb szintre lépéssel megvalósítani. A „drill down - lefúrás” ennek az ellentéte, mikor a mélyebben lévő adatokra vagyunk kíváncsiak, megnézzük, hogy milyen adatokból áll össze az aggregáció. Az „rotate – elforgatás” csak egy megjelenítésbeli művelet, dimenziók megcserélésével valósítható meg. A „slice - szeletelés” művelettel egy feltételt adhatunk meg, mely gyakorlatilag egy szűrés, így a kocka egy darabját levágtuk. A „dice – kockázás” művelettel különböző szeletek metszésével egy részkockát jelölhetünk ki.

A Microsoft Excel is korlátozott lehetőséget biztosít ilyen kockáknak a kezelésére, méghozzá az „Adatok/Kimutatás vagy kimutatásdiagram” (PivotTable wizard) menüpont segítségével.

A menüpont használatakor először ki kell választani az adatokat tartalmazó táblázatot, majd egy üres felület jelenik meg.




**3. ábra Kimutatás Microsoft Excel-ben**

Ezen a felületen interaktív módon állíthatjuk össze a lekérdezésünket úgy, hogy a mezőlistán szereplő mezőket a megfelelő Oldal mezőként, Sor mezőként vagy Adat mezőként felvesszük. Állítsunk össze pl. egy olyan lekérdezést, mely azt adja vissza, hogy melyik irodában melyik ügynök mennyi ingatlan adásvételében vett részt, megyékre, azon belül pedig városokra lebontva. Ekkor a „Megye”, majd a „Város” mezőt húzzuk a sorok közé, az „Iroda” és „Név” mezőt az oszlopok közé, és az árat az adathoz. Az Oldal mezőnél különböző szűrési feltételeket tudnánk megadni a mezőkre megadva.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 3 | Összeg / Ár | | Iroda | Név | | | |
| 4 | | | Budai iroda | | Pesti iroda | | |
| 5 | Megye | Város | Kovács Károly | Szabó Sándor | Horváth Anna | Kiss Tibor | Varga Katalin |
| 6 | Budapest | Budapest I. ker. | 344821996 | 546388899 | 104824730 | 459890723 | 485286165 |
| 7 | | Budapest II. ker. | 218512904 | 330644182 | 227500150 | 282646452 | 395886024 |
| 8 | | Budapest III. ker. | 237003542 | 503830834 | 193012394 | 452980549 | 128326933 |
| 9 | | Budapest IV. ker. | 128131123 | 307142669 | 211914985 | 266828531 | 70646860 |
| 10 | | Budapest IX. ker. | 166108372 | 189087451 | 364351139 | 270840664 | 357056805 |
| 11 | | Budapest V. ker. | 473495598 | 375499064 | 210216262 | 245230889 | 244949933 |
| 12 | | Budapest VI. ker. | 274752676 | 273176830 | 342842996 | 256672220 | 176734976 |
| 13 | | Budapest VII. ker. | 291861700 | | 211561977 | 431844701 | 368811947 |
| 14 | | Budapest VIII. ker. | 274918243 | 160340213 | 176090217 | 293856797 | 475218851 |
| 15 | | Budapest X. ker. | 149886163 | 442067735 | 301060057 | 168453829 | 388835140 |
| 16 | Pest | Budaörs | 315729850 | 155312618 | 303015339 | 275844915 | 118901434 |
| 17 | | Bunakeszi | 292940312 | 137265743 | 65073076 | 26622757 | 273719805 |
| 18 | | Cegléd | 73072910 | 128700499 | 241908817 | 138958206 | 209252792 |
| 19 | | Fót | 178663949 | 387169003 | 386302967 | 32890075 | 79349342 |
| 20 | | Göd | 38577277 | 159896082 | 214377943 | 302733570 | 335485224 |
| 21 | | Gödöllő | 570331749 | 235939756 | 389556305 | 239946125 | 37452634 |
| 22 | | Ráckeve | 260334338 | 250395428 | 274584077 | 270099842 | 280163130 |
| 23 | | Szentendre | 287023355 | 391291123 | 182712784 | 179794544 | 237400760 |
| 24 | | Szob | 416836477 | 342097497 | 475071458 | 446221491 | 313263449 |
| 25 | | Vác | 246079796 | 312322228 | 192235502 | 280773411 | 327203245 |

**4. ábra Elkészített kimutatás Microsoft Excel-ben**

A Microsoft Excel kitűnően elboldogul pár ezer sorral is, azonban ha milliós nagyságrendben gondolkodunk, komolyabb rendszerre lesz szükségünk. Olyan rendszerre, mely hatalmas adatmennyiséget képes kezelni, elosztottan használható, felhasználóbarát felülettel rendelkezik, és képes bizonyos adatokat jogosultság függvényében elrejteni.

Az adattárházakban az adatok tárolását többféleképpen meg lehet oldani, de minden esetben fontos az aggregáció, azaz a műveletek előre kiszámolása, hogy a lekérdezéseket gyorsabban lehessen végrehajtani. A tárolás történhet relációs adatbázisban (ROLAP), ilyenkor a klasszikus táblákban történik a reprezentáció. Történhet csillag sémában, mikor egy ténytábla van középen, mely tartalmazza a tény adatokat, értékeket, valamint a külső kulcsokat a dimenziókra, és a körülötte helyezkednek el a dimenzió táblák. Az előbbi 3. normál formában, míg az utóbbiak 2. normál formában vannak, azaz egy dimenzió egy táblára van „kiterítve". A hópehely séma esetén már egy dimenziót több tábla is leírhat. A galaxis séma esetén több tény tábla van, amelyek a közös dimenzió táblákkal vannak összekötve. A ROLAP esetén az aggregációt aggregációs táblák biztosítják.

Létezik többdimenziós tárolási módszer is (MOLAP), valamint a kettőnek a keveréke (HOLAP). A relációs adatbázisokhoz hasonlóan a többdimenziós adatmodellre is alkottak egy lekérdező nyelvet, ez a MultiDimensional eXpressions nyelv (MDX). Ez kezdetben a Microsoft által kiadott OLE DB for OLAP API (szabványos API OLAP rendszerek megszólítására) része volt, és később vált de facto szabvánnyá.

# Eszközök

Speciális felhasználási körük miatt ezen eszközök először csak kereskedelmi forgalomban megjelenő eszközök voltak. A terület fontosságát és a területen lezajló gyors változásokat a nagyszámú felvásárlások is jelzik. A piacon főleg a nagy gyártók termékei vannak jelen, melyek különböző felvásárlásokkal igyekeztek a termékpalettájukat a BI területen teljessé tenni, ilyen az Oracle, SAP, IBM, Microsoft, stb., A független „kisebb" cégek ezen felvásárlások áldozataivá váltak, sort az Oracle indította a Hyperion felvásárlásával, majd következett az SAP-Business Objects, majd az IBM-Cognos fúzió. Ezen kívül maradt még pár specialista cég is, mely egy adott témára fókuszálva próbál meg fennmaradni, ilyen pl. a Microstrategy, SPSS, Informatica és Teradata.

Ezen kívül elkezdtek megjelenni ingyenes, nyílt forráskódú termékek is. Az ezeket gyártó cégek kevert üzleti modellt alkalmaznak, a termékeik nyílt forráskódúvá tétele mellett fizetős, integrált eszközöket is gyártanak, valamint termékeikhez támogatást árulnak. Ezen gyártók is felismerték, hogy üzletileg kifizetődőbb, hogyha nem csak egy terméket fejlesztenek, hanem szélesítik a portfóliót, és komplex megoldásokat nyújtanak. Ilyen a BIRT jeletés készítő rendszer mögött álló Actuate, a JasperReports-al elhíresült JasperSoft, valamint a cikkem témáját nyújtó Pentaho is.

A Pentaho szintén komplett BI csomagot kínál, melynek tagjai a külön is letölthető, nyílt forráskódú Kettle adatintegrációs, ETL eszköz; a Pentaho Reporting jelentés készítő eszköz, a Weka adatbányász szoftver és a Mondrian Java alapú OLAP motor.

A Mondrian alapvetően egy kliens-szerver architektúrájú ROLAP eszköz, mely az adatokat relációs adatbázisban tarja, és a sebességet előre kiszámolt aggregált táblákkal biztosítja (csak bizonyos rekordszám felett kell használni). A Mondrian-hoz tartozik a Schema Workbench is, mellyel a relációs séma és a többdimenziós séma megfeleltetéséért felelős XML állományt tudjunk grafikusan szerkeszteni, valamint az Aggregation Designer az aggregációs táblák szerkesztéséhez.

A Mondrian nem tartalmaz saját grafikus felületet, helyette a JPivot nevű ingyenes JSP tag library-t használja.

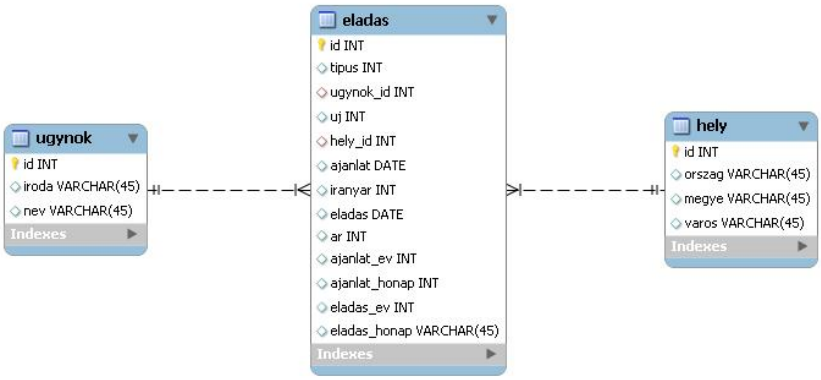
A Mondrian képes MDX utasítások elemzésére, feldolgozására, SQL utasításokká való átfordítására, futtatására.

A Mondrian a megfelelő sebesség eléréséhez az aggregált táblákon kívül az adatokat a memóriában is cache-eli, és a cache-elést programozó is befolyásolhatja (pl. adatbetöltés után ürítheti a cache-t, vagy a cache egy részét). A jogosultságkezelés különböző szerepkörök megadásával oldható meg, és a szerepkörökhöz dimenziószintek rendelhetőek. A Mondrian könnyen kiterjeszthető plug-in-ek írásával. Megadhatóak pl. felhasználó által definiált függvények, különböző formázást végző komponensek. Támogatja a többnyelvűsítést. Megvalósítja az XML/A (XML for Analysis) szabványt, mely egy gyártó független API OLAP eszközök elérésére. A HTTP, SOAP, és XML webes szabványokra épít, és az MDX lekérdezőnyelvet használja.

Implementálja a kidolgozás alatt álló OLAP4J szabványt is, melynek egyik írója az a Julian Hyde, aki a Mondrian megalkotója is. Sok szabványhoz hasonlóan ez is egy megfelelően dokumentált interfész gyűjtemény, JUnit teszt esetekből álló Test Compatibility Kit. A referencia implementáció a Mondrian OLAP4J driver-e, mely a definiált interfészeket megvalósítja. A szabvány definiálja a JDBC-hez hasonlóan a kapcsolatok, statement-ek kezelését, de itt a lekérdezőnyelv az SQL helyett az MDX, és a visszatérési érték (result set) többdimenziós.

A JPivot JSP tag library, mely a többdimenziós eredmény (főleg 2 dimenziós) megjelenítésére képes, valamint lehetőséget biztosít a kockában történő navigációra. Külön projekt, bár a letölthető Mondrian csomagban is megtalálható. Elvileg megvalósítás független, ugyanis az MDX előállításra, manipulációra saját osztályokat használ, így nem kötődik a Mondrian-hoz, elméletileg bármilyen XML/A interfészhez rendelkező OLAP motorhoz képes csatlakozni.

Az ingatlanközvetítő cég OLAP rendszerének relációs adatbázis alapú reprezentációja csillag séma, melyet a következő ábra mutat. Az adatbázist MySQL adatbázis-kezelőben hoztam létre, és az Excel-ből CSV-ként kiexpotált adatokat töltöttem be a „LOAD DATA” utasítás segítségével.




**5. ábra Csillagséma**

A többdimenziós séma létrehozásához a többdimenziós sémát és a relációs sémát összerendelő XML állományt kell létrehozni. Ehhez használhatjuk a külön letöltendő Schema Workbench-et, de akár megírhatjuk kézzel is.

A Workbench használata azért javasolt, mert grafikus felületet biztosít a szerkesztéshez, képes az adatbázishoz csatlakozni, és a relációs adatbázis séma alapján automatikusan felajánlja a tábla és mezőneveket, valamint azonnal MDX utasításokat is tudunk benne futtatni. Ehhez a letöltött állományt ki kell tömöríteni, majd az alkalmazást indító workbench.bat állományba állítsuk be, hogy az adatbázis driver szerepeljen a classpath-ban. Ezzel már el is indítható az alkalmazás.

Első lépésként tervezzük meg a sémát (File/New/Schema), majd mentsük el az XML állományt. Második lépésként próbaképp futtassuk le azt az MDX-et (File/New/MDX Query), melynek eredménye körülbelül megegyezik a Microsoft Excel-ben megjelenített kimutatással. Ezen MDX a következő:

```
SELECT
    {Crossjoin({[Measures].[Ár]}, [Ügynök].Members)} ON COLUMNS,
    {[Cím].Members} ON ROWS
FROM [Eladás]
```

A jelen cikk kereteit meghaladja az MDX nyelvnek a bemutatása, de egy több, mint 70 cikkből álló, nagyon jó tutorial található a http://www.databasejournal.com/features/article.php/3593466/MS-SQL-Series.htm címen

MDX Essentials Series névvel.

Az XML állományban definiálni kell a sémát, abban a kockát, és hogy melyik ténytáblára épül. Ezen belül kell definiálni a dimenziókat és a mértékeket. Az XML eleje tehát:

```
<Schema name="RealEstate">
    <Cube name="Eladás" cache="true" enabled="true">
        <Table name="eladas" />
...
```

Amivel definiáltunk egy „RealEstate” nevű sémát, egy „Eladás” kockát és megadtuk, hogy az „eladas” táblából vegye az adatokat.

A következő XML részlet a „Cím” hierarchiát mutatja, aminek felső szintje az ország, középső szintje a megye, alsó szintje a város.

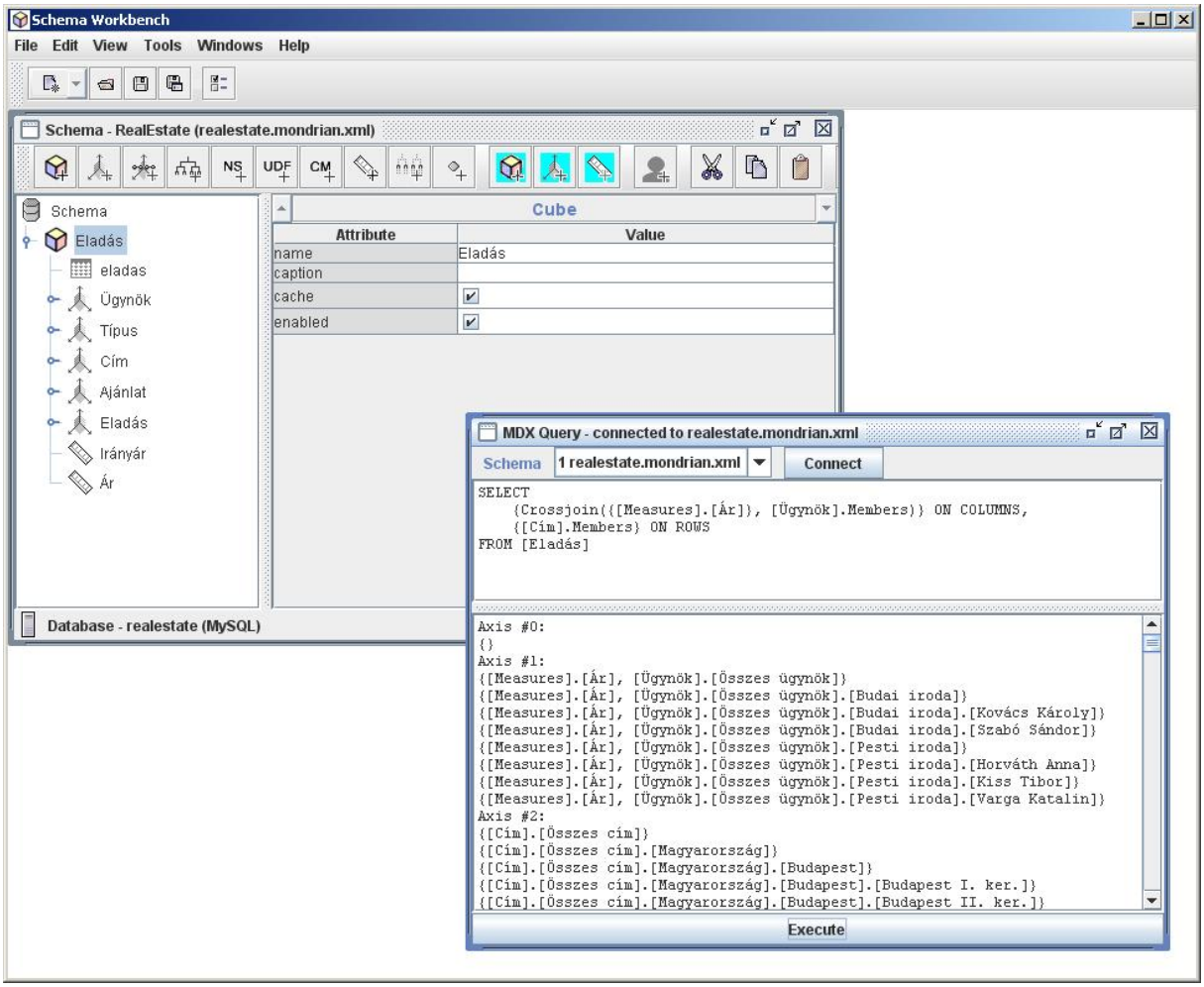
```
...
        <Dimension       type="StandardDimension"       foreignKey="hely_id"
name="Cím">
            <Hierarchy     hasAll="true"     allMemberName="Összes     cím"
primaryKey="id">
                <Table name="hely" />
                <Level    name="Ország"    table="hely"    column="orszag"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
                <Level     name="Megye"     table="hely"     column="megye"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
                <Level     name="Város"     table="hely"     column="varos"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
            </Hierarchy>
        </Dimension>
...
```

Az „eladas” tény táblában a „hely_id” külső kulcs mutat a „hely” táblában szereplő „id” nevű elsődleges kulcsra. A szinteket a „hely” tábla „orszag”, „megye” és „varos” mezői tartalmazzák.

Az értékeket a következő XML részlet mutatja:

```
...
<Measure name="Irányár" column="iranyar" aggregator="sum" visible="true" />
<Measure name="Ár" column="ar" aggregator="sum" visible="true" />
...
```

A teljes XML a cikk végén kerül közlésre. Mentsük el az állományt realestate.mondrian.xml néven.




**6. ábra Schema Workbench**

Amennyiben ezzel megvagyunk, nincs más dolgunk, csak az alkalmazást kell megírnunk. Ehhez töltsük le a Mondrian csomagot, és a lib könyvtárból tömörítsük ki a mondrian.war állományt, és készítsünk belőle egy projekt-et a kedvenc IDE-nkben.

Ahhoz, hogy az adatbázishoz kapcsolódjunk, egy DataSource-t kell felvennünk. Én Tomcat web konténert használtam, így ezt a server.xml-ben kellett megadni a GlobalNamingResources tag alatt a következőképp:

```
. . .
```

```
        <Resource            name="RealEstateDS"            auth="Container"
type="javax.sql.DataSource"
                  maxActive="100"        maxIdle="30"        maxWait="10000"
username="realestate" password="realestate12"
                  driverClassName="com.mysql.jdbc.Driver"

url="jdbc:mysql://localhost/realestate?autoReconnect=true"/>
...
```

Ezután a context.xml-t kellett kiegészítenem, hogy a globális JNDI nevet lokálisra mappelje:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/szat">
      <ResourceLink      global="RealEstateDS"      name="jdbc/RealEstateDS"
type="javax.sql.DataSource"/>
</Context>
```

Eztán a realestate.mondrian.xml állományt kell a WEB-INF/classes könyvtárba másolni, majd a WEB-INF/queries/ könyvtárba hozzunk létre egy realestate.jsp állományt, melynek tartalma a következő:

```
<%@ page session="true" contentType="text/html; charset=ISO-8859-1" %>
<%@ taglib uri="http://www.tonbeller.com/jpivot" prefix="jp" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>

<jp:mondrianQuery        id="query01"        dataSource="jdbc/RealEstateDS"
catalogUri="/WEB-INF/classes/realestate.mondrian.xml">
SELECT {Crossjoin({[Measures].[Ár]}, [Ügynök].Members)} ON COLUMNS,
  {[Cím].Members} ON ROWS
FROM [Eladás] </jp:mondrianQuery>
<c:set var="title01" scope="session">Real estate</c:set>
```

A JSP oldalt a testpage.jsp include-olja, a megfelelő paraméterrel. A fenti JSP-ben megadtuk a DataSource nevét, a realestate.mondrian.xml séma állományt, valamint az MDX lekérdezést. A web konténert elindítva a http://localhost:8084/szat/testpage.jsp?query=realestate címet beírva tekinthető meg a következő oldal:

**Real estate**

| | Measures | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Ár | | | | | | | |
| | Ügynök | | | | | | | |
| Cím | -Összes ügynök | -Budai iroda | Kovács Károly | Szabó Sándor | -Pesti iroda | Horváth Anna | Kiss Tibor | Varga Katalin |
| -Összes cím | 26 562 939 099 | 10 867 650 184 | 5 239 082 330 | 5 628 567 854 | 15 695 288 915 | 5 068 213 175 | 5 323 130 291 | 5 303 945 449 |
| -Magyarország | 26 562 939 099 | 10 867 650 184 | 5 239 082 330 | 5 628 567 854 | 15 695 288 915 | 5 068 213 175 | 5 323 130 291 | 5 303 945 449 |
| -Budapest | 14 252 044 090 | 5 687 670 194 | 2 559 492 317 | 3 128 177 877 | 8 564 373 896 | 2 343 374 907 | 3 129 245 355 | 3 091 753 634 |
| Budapest I. ker. | 1 941 212 513 | 891 210 895 | 344 821 996 | 546 388 899 | 1 050 001 618 | 104 824 730 | 459 890 723 | 485 286 165 |
| Budapest II. ker. | 1 455 189 712 | 549 157 086 | 218 512 904 | 330 644 182 | 906 032 626 | 227 500 150 | 282 646 452 | 395 886 024 |
| Budapest III. ker. | 1 515 154 252 | 740 834 376 | 237 003 542 | 503 830 834 | 774 319 876 | 193 012 394 | 452 980 549 | 128 326 933 |
| Budapest IV. ker. | 984 664 168 | 435 273 792 | 128 131 123 | 307 142 669 | 549 390 376 | 211 914 985 | 266 828 531 | 70 646 860 |
| Budapest IX. ker. | 1 347 444 431 | 355 195 823 | 166 108 372 | 189 087 451 | 992 248 608 | 364 351 139 | 270 840 664 | 357 056 805 |
| Budapest V. ker. | 1 549 391 746 | 848 994 662 | 473 495 598 | 375 499 064 | 700 397 084 | 210 216 262 | 245 230 889 | 244 949 933 |
| Budapest VI. ker. | 1 324 179 698 | 547 929 506 | 274 752 676 | 273 176 830 | 776 250 192 | 342 842 996 | 256 672 220 | 176 734 976 |
| Budapest VII. ker. | 1 304 080 325 | 291 861 700 | 291 861 700 | | 1 012 218 625 | 211 561 977 | 431 844 701 | 368 811 947 |
| Budapest VIII. ker. | 1 380 424 321 | 435 258 456 | 274 918 243 | 160 340 213 | 945 165 865 | 176 090 217 | 293 856 797 | 475 218 851 |
| Budapest X. ker. | 1 450 302 924 | 591 953 898 | 149 886 163 | 442 067 735 | 858 349 026 | 301 060 057 | 168 453 829 | 388 835 140 |
| -Pest | 12 310 895 009 | 5 179 979 990 | 2 679 590 013 | 2 500 389 977 | 7 130 915 019 | 2 724 838 268 | 2 193 884 936 | 2 212 191 815 |
| Budaörs | 1 168 804 156 | 471 042 468 | 315 729 850 | 155 312 618 | 697 761 688 | 303 015 339 | 275 844 915 | 118 901 434 |
| Bunakeszi | 795 621 693 | 430 206 055 | 292 940 312 | 137 265 743 | 365 415 638 | 65 073 076 | 26 622 757 | 273 719 805 |
| Cegléd | 791 893 224 | 201 773 409 | 73 072 910 | 128 700 499 | 590 119 815 | 241 908 817 | 138 958 206 | 209 252 792 |
| Fót | 1 064 375 336 | 565 832 952 | 178 663 949 | 387 169 003 | 498 542 384 | 386 302 967 | 32 890 075 | 79 349 342 |
| Göd | 1 051 070 096 | 198 473 359 | 38 577 277 | 159 896 082 | 852 596 737 | 214 377 943 | 302 733 570 | 335 485 224 |
| Gödöllő | 1 473 226 569 | 806 271 505 | 570 331 749 | 235 939 756 | 666 955 064 | 389 556 305 | 239 946 125 | 37 452 634 |
| Ráckeve | 1 335 576 815 | 510 729 766 | 260 334 338 | 250 395 428 | 824 847 049 | 274 584 077 | 270 099 842 | 280 163 130 |
| Szentendre | 1 278 222 566 | 678 314 478 | 287 023 355 | 391 291 123 | 599 908 088 | 182 712 784 | 179 794 544 | 237 400 760 |
| Szob | 1 993 490 372 | 758 933 974 | 416 836 477 | 342 097 497 | 1 234 556 398 | 475 071 458 | 446 221 491 | 313 263 449 |
| Vác | 1 358 614 182 | 558 402 024 | 246 079 796 | 312 322 228 | 800 212 158 | 192 235 502 | 280 773 411 | 327 203 245 |

**7. ábra JPivot táblázat**

A felső ikonsor biztosítja az interaktív navigációt, az OLAP kocka összerakását, dimenziók megadását oszlopnak vagy sornak, szűrő feltételek megadását, forgatást. Az MDX ikonra kattintva megjelenik a legenerált MDX utasítás.

A táblázatban lévő plusz jellel lehet lefúrni, és a mínusz jellel felgöngyölíteni.

A táblázat grafikonként kirajzolható, kinyomtatható, vagy Excel fájlként lementhető.

# Tapasztalatok

A Mondrian dokumentációja terjedelmes, folyamatosan fejlődik, de azért néha szükség lehet a forráskód vizsgálatára. Hibáktól ugyan nem mentes, de a listája aktív, több kérdésére is 1-2 napon belül kaptam választ.

Sajnos a JPivot-tal már nem voltam teljes mértékben megelégedve. Egy saját webes keretrendszert használ, WCF néven. Ez egy nagyon ritkán használt keretrendszer (Interneten nem is lehet találni más felhasználását). Egy külön controller servlet tartozik hozzá, a komponenseket XML-ben (DOM-fa) tárolja, ráadásul a session-ben, majd XSLT transzformációval jeleníti meg őket a JSP oldal futtatásakor. Saját expression language-el rendelkezik, melyet # karakterrel kell használni. Új alkalmazás készítésekor a jelenlegi példákból érdemes kiindulni, nincs olyan dokumentáció, hogy hogyan lehet elölről kezdve felépíteni egy alkalmazást. A JPivot és a WCF dokumentációja is szegényes és elavult. Sajnos több hiba is van benne, melybe én is belefutottam, és azóta sem javították ki. Több magyarországi projektről is tudok, amelyben használták, és mindegyikben erősen bele kellett nyúlni a forráskódjába.

# Konklúzió

Amennyiben olyan rendszer készítése a feladatunk, mely feladata a döntéshozók kiszolgálása egy olyan interaktív felületen, melyen ad-hoc módon tudnak informatikai tudás nélkül lekérdezéseket megfogalmazni, szükség esetén lefúrni, vagy felgöngyölíteni, nem szükséges magunk megvalósítanunk egy ilyen rendszert, és nem kötelező a drága kereskedelmi termékek közül választatnunk, hiszen már ingyenes megoldások is léteznek erre a feladatra. A Mondrian egy nagyon stabil, szabványokon alapuló, nyílt forráskódú, Java nyelven implementált OLAP motor. Sajnos a megjelenítésre ez már nem mondható el. Amennyiben egy professzionális, személyre szabható Pivot táblát szeretnénk megjeleníteni, vagy ne válasszuk a JPivot eszközt, vagy készüljünk fel arra, hogy nagyon bele kell nyúlni a forráskódjába.

# Köszönetnyilvánítás

A cikk elkészítése kapcsán köszönetet kell mondanom Arató Bencének, aki meghívott az I. Nyílt forráskódú BI konferenciára, ezzel figyelmemet jobban a téma felé fordította. Kovács Zoltánnak, akinek a TechNet Magazinban megjelent nagyszerű OLAP-alapok cikkéből sok mindent értettem meg. Valamint a JUM (Java User Meetings - http://jum.hu) közösségének, hogy végighallgatták az előadásomat.

Budapest, 2009. március 23.

# Források

Dr. Abonyi János: Adatbányászat a hatékonyság eszköze
Kovács Zoltán: OLAP-alapok

Arató Bence: Az üzleti intelligencia évkönyve
I. Nyílt forráskódú BI konferencia
http://www.bi.hu
http://www.wikipedia.org
http://mondrian.pentaho.org
http://jpivot.sourceforge.net
http://www.xmlforanalysis.com/
http://www.olap4j.org/

# A szerzőről

Viczián István a Debreceni Egyetem programtervező szakán végzett 2001-ben. Azóta szoftverfejlesztéssel, Java technológiákkal foglalkozik. Szabadidejében írja a Java technológiákkal foglalkozó blogját a http://jtechlog.blogspot.com címen.

E-mail cím: viczian.istvan a gmail-en

Honlap: http://delfin.unideb.hu/~vicziani/

# Melléklet – séma XML

Ez a melléklet tartalmazza a relációs és többdimenziós séma összerendelését végző XML állományt.

```
<Schema name="RealEstate">
    <Cube name="Eladás" cache="true" enabled="true">
        <Table name="eladas" />
        <!-- Ugynok -->
        <Dimension      type="StandardDimension"      foreignKey="ugynok_id"
name="Ügynök">
            <Hierarchy    hasAll="true"    allMemberName="Összes    ügynök"
primaryKey="id">
                <Table name="ugynok" />
                <Level    name="Iroda"    table="ugynok"    column="iroda"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
                <Level     name="Ügynök"     table="ugynok"     column="nev"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
            </Hierarchy>
        </Dimension>
        <!-- Tipus -->
        <Dimension foreignKey="tipus" name="Típus">
            <Hierarchy    hasAll="true"    allMemberName="Összes    típus"
primaryKey="id">
                <InlineTable alias="tipus">
                    <ColumnDefs>
                        <ColumnDef name="id" type="Integer">
                        </ColumnDef>
                        <ColumnDef name="leiras" type="String">
                        </ColumnDef>
                    </ColumnDefs>
                    <Rows>
                        <Row>
                            <Value column="id">
                                        1
                            </Value>
                            <Value column="leiras">
                                        Tégla lakás
                            </Value>
                        </Row>
                        <Row>
                            <Value column="id">
                                        2
                            </Value>
                            <Value column="leiras">
                                        Panel lakás
                            </Value>
                        </Row>
                        <Row>
                            <Value column="id">
                                        3
                            </Value>
                            <Value column="leiras">
                                        Ház
                            </Value>
                        </Row>
                    </Rows>
                </InlineTable>
```

```
                <Level    name="Típus"    column="id"    nameColumn="leiras"
type="String" uniqueMembers="true" levelType="Regular" hideMemberIf="Never"
/>
            </Hierarchy>
        </Dimension>
        <!-- Hely -->
        <Dimension       type="StandardDimension"       foreignKey="hely_id"
name="Cím">
            <Hierarchy     hasAll="true"     allMemberName="Összes     cím"
primaryKey="id">
                <Table name="hely" />
                <Level    name="Ország"    table="hely"    column="orszag"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
                <Level     name="Megye"     table="hely"     column="megye"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
                <Level     name="Város"     table="hely"     column="varos"
type="String"           uniqueMembers="false"           levelType="Regular"
hideMemberIf="Never" />
            </Hierarchy>
        </Dimension>
        <!-- Ajanlat -->
        <Dimension name="Ajánlat" type="TimeDimension">
            <Hierarchy hasAll="true" allMemberName="Összes ajánlat">
                <Level  name="Év"  column="ajanlat_ev"  uniqueMembers="true"
levelType="TimeYears" type="Numeric"/>
                <Level          name="Hónap"          column="ajanlat_honap"
uniqueMembers="false"     ordinalColumn="ajanlat"     levelType="TimeMonths"
type="Numeric"/>
            </Hierarchy>
        </Dimension>
        <!-- Eladas -->
        <Dimension name="Eladás" type="TimeDimension">
            <Hierarchy hasAll="true" allMemberName="Összes eladás">
                <Level  name="Év"  column="eladas_ev"  uniqueMembers="true"
levelType="TimeYears" type="Numeric"/>
                <Level          name="Hónap"          column="eladas_honap"
uniqueMembers="false"     ordinalColumn="eladas"     levelType="TimeMonths"
type="Numeric"/>
            </Hierarchy>
        </Dimension>

        <Measure    name="Irányár"    column="iranyar"    aggregator="sum"
visible="true" />
        <Measure name="Ár" column="ar" aggregator="sum" visible="true" />
    </Cube>
</Schema>
```