

# TechNet

2008. SZEPTEMBER – OKTÓBER

MAGAZIN *MÉLYVÍZ, CSAK ÚSZÓKNAK!*

## Üzleti intelligencia

**MICROSOFT OFFICE  
PERFORMANCE  
POINT SERVER  
2007**

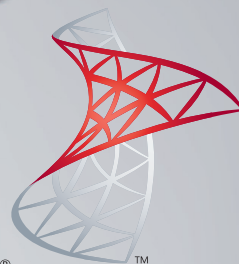
**Egy termékben  
a teljesítménymenedzsment**

**VIRTUÁLIS  
MINDENFELE**

**Megérkezett a Microsoft  
Hyper-V Server 2008**

**GROUP POLICY  
PREFERENCES**

**Csoportházirend,  
de nem kötelező érvényű  
a felhasználóra?**



Microsoft®  
**SQL Server® 2008**

**BŐSÉGTÁL:**

**SOK-SOK KALAPÁCS**

**ADATTÁRHÁZ-ÉPÍTÉS  
LÉPÉSRŐL LÉPÉSRE**

**ANALYSIS SERVICES**

**CSÁKÁNYT A KÉZBE!**

**KI A LEGÉNY A GÁTON?**

Ára: 999 Ft



9 771586 518005 08003

**Microsoft TechNet**

# Gondoljon a jövőre!

Vegyen részt **BIZTOSAN** induló őszi/téli képzéseinken, és a befizetett tanfolyami díj **50%**-át felhasználhatja 2009-ben!

**Több mint 40 féle tanfolyam közül válogathat!**

A képzések listája a résztvevőknek megfelelően folyamatosan bővíthet, kérjük, figyelje weboldalunkat!

Néhány tanfolyami téma ízelítőül:

- Windows Server 2003, 2008 rendszergazdai tanfolyamok
- SQL Server 2005 üzemeltetés, implementáció
- .NET, Visual Studio 2005 fejlesztés (ASP.NET, ADO.NET, Windows alkalmazások, osztott alkalmazások)
- Cisco tanfolyamok CCNA, CCNP és CCIP témában
- ITILv3 Foundation

a mi tudásunk  
az **Ön** sikere

Telefon: 203-0304/4122 m.

[www.szamalk.hu/tisza](http://www.szamalk.hu/tisza)



Az akciós feltételekkel és jelentkezéssel, valamint az induló képzésekkel kapcsolatos további információért, kérjük, látogassa meg weboldalunkat (Akciók) vagy keresse munkatársainkat! A képzésekre szakképzési hozzájárulás illetve SA oktatási utalványok is felhasználhatók!

# VENI, VIDI, BI

Háromévnyi száguldás, három tételben.



**Havas Levente**  
vezető tanácsadó

**H**a három éve megkérdezik tőlem – független szakértőtől –, hogy Microsoft-eszközökkel milyen BI(Business Intelligence) rendszert tudnék javasolni, kivitelezni, akkor leginkább a relációs adatbázis, az OLAP-motor és az Excel-tábla alapú megoldások jutottak volna eszembe, hála Kószó Károly SQL 2005 Launch- és TechNet-előadásainak, amelyeket mind végigültem. Magyarán: az összegyűjtött tényadatok vastagklienses elemzését. Ez indulásnak nem is rossz, főleg ha figyelembe vesszük azt a hazai Excel-dzsungelt, amit a felhasználók az üzletük nyomására kínjukban barkácsoltak össze. Pedig már akkor is volt SQL Server 2005, csak ezek a képességei nem voltak idehaza kidomborítva (*Veni*).

Amikor megkérdeztek két éve – mint a Microsoft hazai szakértőjét –, hogy mit tudnék javasolni, akkor már az előzőeken felül volt vékonykliens-felület a SharePoint jóvoltából. Abba már az Office Business Scorecard Manager 2005-öt tudtuk integrálni, és az éppen felvásárolt ProClarityvel az összetettebb elemzések, a ProClarity Analytics Serverrel azok intelligens tárolása, csoportmunka alá vonása is megoldható lett. Azaz a tényszámok elemzése már összetettebbek lehettek az elemzőknek, a vezetők pedig szofisztikált webes, jogosultságokkal kezelt felület révén ismerhették fel az üzleti kihívásokat. Mindez egy év leforgása alatt! És akkor már tudtuk, hogy érkezik a „mindent vivő” Office PerformancePoint Server és az SQL Server 2008. A belső webcastokon láttuk, hogy napról napra miként nyeri el végső formáját, funkcionalitását a PerformancePoint, és miképpen tölti be a korábbi üröket az SQL Server: látványos, speciális, magával ragadó diagramok (Dundas Charts, Gauge), nagyméretű riportok kezelése, IIS nélküli report server, egyszerűbb felületű Report Manager stb. (*Vidi*).

Amikor ebben az évben Steve Ballmer megtisztelte az országot, és eljött bejelenteni az SQL Server 2008-at – és ahol magam is kaptam egy kis szerepet –, elgondolkodtam. Elgondolkodtam azon, hogy három éve mit mondtam volna, és most miről fogok beszélni. Pestiesen szólva: „nem semmi”. Ez a mondás egyben a felpörgött BI-piacnak, illetve a Microsoftnak szólt (*BI*). Mára van minden a redmondi szoftvergyár készletében, ami kellhet egy összetettebb projekthez, csak győzzük igényekkel, szakértő kivitelezőkkel, tanácsadókkal, fejlesztőkkel, üzemeltetőkkel, tréningekkel, hogy az Excel-dzsungelek megpihenhessenek végre.

A TechNet Magazin mostani száma ezt a száguldást állítja meg egy pillanatra; hagyja beszállni az érdeklődőket, hogy majd az ismeretek, ötletek birtokában továbbshunjanak a megoldások dimenzióiba.

## Business Intelligence Solutions & Services

A BISS Kft. a Microsoft Magyarország 2008-as év díjazott BI partnere

### Előnyei:

- ! Jelentős időt spórolhat
- ! Intranetbe, BI portálba integrálható tartalom
- ! A manuális dokumentációnak vége
- ! Standardizálhatja beszállító partnereinek dokumentációit

### Próbálta SQL adatbázisának dokumentációját frissen tartani?

#### Gombnyomásra

- SQL 2000 és 2005 adatbázis dokumentáció
- Integration Services (ETL) dokumentáció
- Analysis Services (OLAP) dokumentáció
- Reporting Services dokumentáció
- Egész szavas keresés a tartalomban
- Meglévő Office dokumentációk csatolása
- HTML-es publikációs felület
- Töltési folyamatok diagramos megjelenítése



<http://www.bidocumenter.com/Sample/Index.htm>



A BISS Kft. fő profilja Üzleti Intelligencia és Folyamatmenedzsment megoldások szállítása, illetve az azzal kapcsolatos szakértői támogatás.

A tevékenység kiterjed a megoldásokkal kapcsolatos bevezetési, tanácsadási, audit, licenz szállítási, support és oktatási tevékenységekre.

A Microsoft Magyarország kiemelt BI szakértő partnere a BISS Kft.

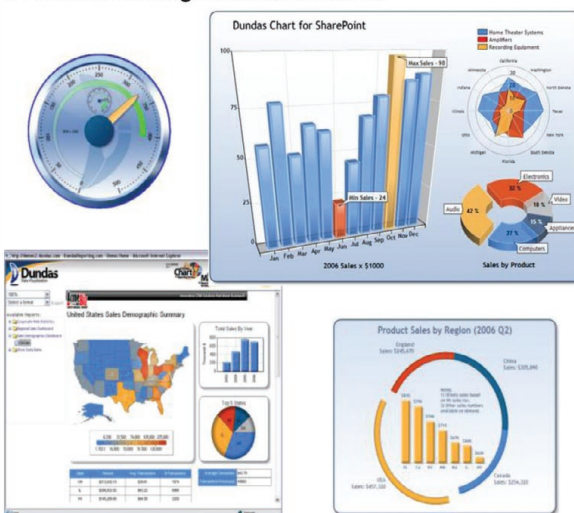
A BISS Kft. a BI Documenter illetve a Dundas termékek hazai képviselője.

### Előnyei:

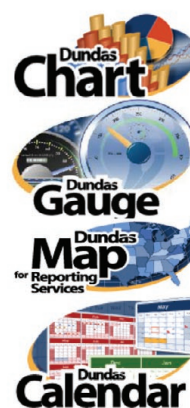
- ! Professzionális információ megjelenítés
- ! Dashboard készítés egyszerűen
- ! Sharepoint-ban, SQL Reporting Services-ben alkalmazható
- ! A Microsoft által javasolt, alkalmazott (SQL Server 2008)

### Fókuszáltak, lényegretörők, látványosak vállalata információi ?

#### A felületek magukért beszélnek



<http://www.dundas.hu>, [www.dundas.com](http://www.dundas.com)



Kapcsolat: Szommer Ákos – [szommer.akos@biss.hu](mailto:szommer.akos@biss.hu); Gosztonyi Szilvia – [gosztonyi.szilvia@biss.hu](mailto:gosztonyi.szilvia@biss.hu)

**Tapasztalat, Megoldások, Értékek**

# Címlapon

## Üzleti intelligencia

### Sok-sok kalapács

(Nagy Levente)

Akinek kalapácsa van, az mindent szögnek néz – avagy mire használható az üzleti intelligencia

6

### Adattárház-építés lépésről lépésre

(Kövári Attila)

Az adattárházak feltöltésének titkai – mit tud az Integration Services 2008?

10

### Analysis Services

(Kövári Attila)

Az üzletiintelligencia-rendszerek alapja

16

### Csákányt a kézbe!

(Balássy György)

Adatbányászat SQL Server 2008 Analysis Services segítségével

23

### Ki a legény a gáton?

(Hangyál Zoltán)

Az első találkozás a Reporting Services-zel

30

### Microsoft Office PerformancePoint Server 2007

(Kovács Zoltán)

Egy termékben a teljesítménymenedzsment

35

## Infrastruktúra

### Virtuális mindenféle

(Somogyi Csaba)

Megérkezett a Microsoft Hyper-V Server 2008

43

### Group Policy Preferences

(Kapás Tibor)

Egy beállításgyűjtemény, ami olyan, mint a csoportházirend, de nem kötelező érvényű a felhasználóra?

48

# TechNet

MAGAZIN

#### SZERKESZTŐSÉG

##### Főszerkesztő

Sziebig Andrea – [asziebig@itbusiness.hu](mailto:asziebig@itbusiness.hu)

##### Szakmai lektor

Budai Péter – [pbudai@microsoft.com](mailto:pbudai@microsoft.com)

##### Főszerkesztő-helyettes

Varga János – [jvarga@itbusiness.hu](mailto:jvarga@itbusiness.hu)

##### Nyomdai előkészítés

Graffaello Kft.

Korrektor

Matula Zsolt

Lapterv és címlap

Emotion Bt.

#### Szerkesztőség és kiadó címe

IT-Business Publishing Kft.

1072 Budapest, Rákóczi út 28.

Tel.: 577-7970, fax: 577-7995

#### KIADÓ

Kiadja a Microsoft Magyarország megbízásából az IT-Business Publishing Kft.

#### A kiadásért felel

Sziebig Andrea ügyvezető

[asziebig@itbusiness.hu](mailto:asziebig@itbusiness.hu)

Tel.: 577-7999, fax: 577-7995

A TechNetben közölt cikkek fordítása, utánnomása, sokszorosítása és adattrendszerekben való tárolása kizárólag a kiadó engedélyével történhet. A megjelent cikkeket szabadalmi vagy más védettségre való tekintet nélkül használjuk fel.

#### Médiareferensek

Csabai Enikő – [ecsabai@itbusiness.hu](mailto:ecsabai@itbusiness.hu)

Tel.: 577-7971

Simorjay Ágnes – [asimorjay@itbusiness.hu](mailto:asimorjay@itbusiness.hu)

Tel.: 577-7973

Szabó K. Lajos – [lszabo@itbusiness.hu](mailto:lszabo@itbusiness.hu)

Tel.: 577-7972

Fax: 577-7995

#### Terjesztés

Terjesztett példányszám: 4700

Előfizethető a kiadó ügyfélszolgálatán:

[terjesztés@itbusiness.hu](mailto:terjesztés@itbusiness.hu)

Az éves előfizetés díja 4990 forint.

MCP-k számára ingyenes!

#### Nyomda:

Pauker Nyomdaipari Kft.

1047 Budapest, Baross utca 11-15.

Felelős vezető: Vértés Gábor ügyvezető igazgató

ISSN 1586-5185

# SOK-SOK KALAPÁCS

Akinek kalapácsa van, az mindent szögnek néz – avagy mire használható az üzleti intelligencia.

**H**a egy mondatban kellene összefoglalni, akkor az üzleti intelligencia az az informatikai eszköztárszer és alkalmazáshalmaz, amely segít egy szervezetnek a rendelkezésre álló, folyamatosan termelődő adatokból döntéstámogató információkat kinyerni. Mivel nagyon szerteágazó a terület, az üzleti intelligencia sokféle jelentéssel bír, és figyelni kell arra, hogy ki mit ért rajta. Sokat segíthet az, ha végignézzük, hogy milyen szituációkban, milyen eszközöket lehet használni és mire. Ebben a cikkben a leggyakoribb üzletiintelligencia-eszközök alkalmazási lehetőségeit tekintjük át.

## Adatok összegzése több szempont szerint (Pivot-tábla)

Az informatika gyökerei az adattároláshoz és adatfeldolgozáshoz nyúlnak vissza. Ennek megfelelően még a kezdő felhasználók is hamar kerülnek szembe kisebb-nagyobb adathalmazokkal, amelyek leggyakrabban Excel-táblázatokban jelennek meg. Az egyszerű összegeket és függvényeket mindenki képes használni, de kevesen ismerik az Excel egyik leghasznosabb funkcióját, a Pivot-táblát (a pivot angolul forgáspontot, forgatást jelent). Ez egy dinamikus táblázat (egy másik belső vagy külső táblázat adatain alapul), amellyel a különböző szempontok szerint összegezhethetjük, átlagolhatjuk, sorba rendezhetjük adatainkat. Szükség van hozzá egy adatokat tartalmazó táblára, amelynek a fejlécében található a mezőnevek. Ezután a beszúrás-szalagon (Office 2007) található Pivot-tábla gomb segítségével létrehozhatjuk a táblánkat, majd a jobb oldalon megjelenő dobozok segítségével megtölthetjük adatokkal. Az egyes mezőket ide-oda húzogatva láthatjuk, hogy gyorsan és kényelmesen elemezhetjük az adatokat különböző szempontok szerint.

A Pivot-tábla látványos testvére a Pivot Chart, ami ugyanezt a rugalmasságot adja grafikon formájában. Legegyszerűbben úgy tudjuk kipróbálni, ha a Pivot-táblánk közepére állva meg-

nyomjuk az F11 gombot, ami automatikusan elkészíti nekünk a grafikonot.

Viszont ennyi elég is a Pivot-tábláról, hiszen az nem az üzleti intelligencia csúcsa, hanem a kezdete, bár sokak szerint minden üzleti intelligencia ügyis Excelben végződik. Mindenesetre a Pivot-tábla megismerése jó kezdés ahhoz, hogy a további eszközök előnyeit lássuk, ezért ha még nem találkozott vele, akkor érdemes félretenni a magazint, és egy kicsit játszani vele, mielőtt továbbolvas.

## Adattárolás

Nagyon leegyszerűsítve a legtöbb üzletiintelligencia-funkció visszavezethető arra, hogy valamilyen módon kinötte az Excelt. Az egyik ilyen az, ha adattárolás szempontjából értük el a határokat. Lehet, hogy már nem férnek el az adatok (bár a 2007-es Excelben már 1048576 sor tárolható), de akár az is, hogy többen egyszerre szeretnék elérni az adatokat, más és más jogosultságok szerint, és szeretnénk, ha egy központi helyen,

Név	Város	SQL Server verzió
Nagy Levente	Budapest	2008
Budai Péter	Budapest	2008
Gipsz Jakab	New York	2005
Adatbázis Anna	Sidney	2005
Select Sándor	Peking	2000

Egy egyszerű lista

Row Labels	Count of Név
2000	1
2005	2
2008	2
<b>Grand Total</b>	<b>5</b>

Az elkészült Pivot-tábla

biztonságban, esetleg titkosítva, rendszeres mentéssel stb. lennének tárolva az adatok. Ekkor érdemes az SQL Serverben gondolkodni, hiszen az Excel képes közvetlenül az SQL Server-táblákhoz csatlakozni és onnan adatokat feldolgozni, így az összes fenti előny mellett élvezhetjük továbbra is az Excel felületét. Kezdsnek akár elegendő lehet egy SQL Server Express változat is, amelyet ingyenesen letölthetünk a Microsoft honlapjáról.

## Jelentéskészítés és megjelenítés

Sokszor van szükség arra, hogy bizonyos kimutatásokat folyamatosan figyeljünk, így a megfelelő döntéseket mindig időben meghozhatjuk. Ilyen például az, ha egy vállalat értékesítési vezetője mindennap áttekinti az üzlet pillanatnyi állását, hogy nyomon követhesse a teljesítményt. Egy ilyen kimutató tartalmaz bizonyos összegzéseket (például mennyi volt összesen az értékesítés ebben a hónapban), részletes adatokat (például az előző hét legnagyobb értékű tranzakciói) és akár még grafikai elemeket is, mint egy tortadiagram arról, hogy az egyes termékcsoportok között hogyan oszlott meg az értékesítés. Ilyen kimutatásokat – jelentéseket (report) – az SQL Server Reporting Services segítségével központilag definiálhatunk, amelyeket aztán a felhasználók egy webes felületen (vagy akár egy Office Sharepoint Portálba ágyazva) érhetnek el.

A jelentéskészítésnek két formája létezik az SQL Server világában: az egyik a standard jelentéskészítés – ekkor a fejlesztők vagy az üzleti tanácsadók előre elkészítenek jelentéseket, és ezeket használhatják a felhasználók. Emellett az SQL Server Reporting Services lehetővé teszi azt is, hogy a szerverten publikált adatmodell alapján maguk a felhasználók készítsenek jelentéseket (ad hoc jelentéskészítés) a saját igényeik szerint, és azokat akár vissza is publikálhatják a kiszolgálóra, hogy mások is elérjék és használják őket.

## OLAP-elemzés

Az adatbázisok nagyon jók az adatok tárolásában, visszakeresésében, összegzésében és még számos más dologban. Viszont amikor üzleti adatelemzésre használjuk őket, előfordulhat, hogy sokan, sokszor egymáshoz nagyon hasonló kérdéseket tesznek fel nekik. Egy vállalat értékesítéseit millió és egy szempont szerint lehet elemezni: termékcsoport

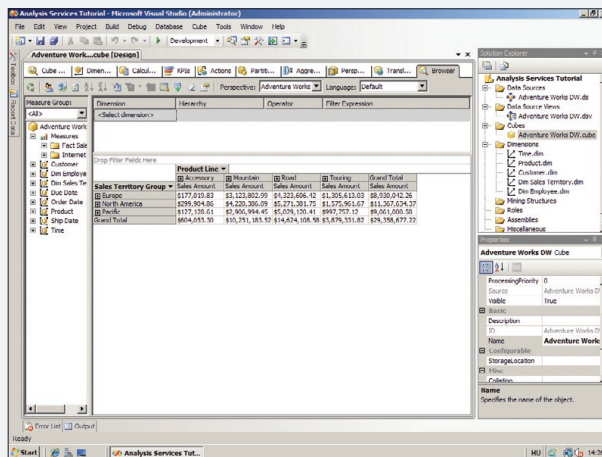
és termék szerint, a vásárlás helye szerint, az értékesítő személye, a divízió, a kereskedelmi csatorna, az ár és még nagyon sok paraméter szerint. Ezeket a szempontokat nevezzük az OLAP-terminológiában dimenzióknak. Az egyes dimenziók lehetnek laposak (csak csoportosító szempontok vannak bennük), vagy tartalmazhatnak hierarchiákat is, amelyek mentén összegezhethetjük vagy részletezhetjük az adatokat. Ilyen hierarchia lehet például az értékesítés helye: város, megye, régió, ország, földrész. Amikor valaki lát egy kimutatást, akkor – amint megértette – elkezd azon gondolkodni, hogy vajon milyen más összefüggések vannak még az adatok között.

Ahogy láttuk, a Pivot-tábla nagyon jó lehetőséget nyújt arra, hogy az adatokat pillanatok alatt megforgassuk és egy más nézőpontból más összefüggéseket fedezzünk fel közöttük. Az OLAP (OnLine Analytical Processing) feladata, hogy az ilyen elemzést elősegítse. Ezt úgy teszi, hogy az adatokat úgynevezett kockákba szervezi, amelyek sokdimenziós adattáblák, és az egyes dimenziók szerint előre összegzik is az adatokat, hogy gyorsabb válaszokat tudjanak adni. Az így előkészített adatokat aztán elérhetjük Excelből, webes felületen vagy valamilyen kliensalkalmazás segítségével. A Microsoft SQL Server OLAP-funkcióit a beépített SQL Server Analysis Services (SSAS) nyújtja. Persze felmerülhet a kérdés, hogy ha ügyis Excelből érjük el az OLAP-kockát, akkor miért nem használunk inkább egy Pivot-táblát, de ennek több magyarázata is van.

- Az OLAP-kiszolgálóval olyan mennyiségű adatot is tudunk kezelni, ami messze túlmutat az Excel képességein, és mindezt gyorsan, hiszen az OLAP előre összegzi, átlagolja nekünk az adatokat.
- A kiszolgáló lehetővé teszi, hogy egyszerre sokan használjuk ugyanazt az adatforrást.
- Az OLAP-kockákban történő adattárolás lehetővé teszi, hogy különleges jogosultságokat adjunk a felhasználóknak. Amíg adattáblákban tároljuk az adatokat, addig egy felhasználónak vagy van hozzájuk joga-

sultása, vagy nincs (egy kicsit leegyszerűsítve persze). Az OLAP-ban lehetőségünk van arra, hogy bizonyos felhasználóknak csak aggregált szintű adatokhoz – vagy épphogy csak a részletekhez – legyen jogosultságuk.

- Az OLAP-kiszolgálón egy sor olyan dolgot is definiálhatunk, amelyeket Pivot-táblában nem vagy csak nagyon áttételesen lehetne. Például:
  - Definiálhatunk halmazokat, amelyek dinamikusan változnak, például az „előző hónap”.
  - Készíthetünk úgynevezett kulcs-teljesítménymutatókat (key performance indicator), amelyek egy adott mutató értékét céltértekekkel hasonlítják össze, és így felhasználhatók scorecardok alapjául.



Egy OLAP-kocka belülről

– Összetett kalkulációkat definiálhatunk, amelyek akár programkódot is tartalmaznak (például dinamikusan végezhetünk devizaárfolyam-kalkulációkat).

Az Excelen kívül az OLAP-kockákat elérhetjük webes felületen is az Office PerformancePoint Serverben található ProClarity Analytics Server, illetve a szintén a Performance Point Serverben található ProClarity-kliens segítségével is.

## Adatbányászat

Van füllentés, van hazugság, és van a statisztika – tartja a mondás. Bár mindannyian ismerjük a statisztika ilyen felhasználását, de azért azt is tudjuk, hogy sokszor nagyon hasznos az adathalmazokban matematikai eszközökkel kutatni az összefüggéseket. Ami a legtöbb esetben visszatartja ettől a földi halandót,

az az, hogy jelentős matematikai háttér nélkül igen nehéz belebonyolódni a statisztikai elemzésekbe. Az adatbányászat nem tesz mást, mint statisztikai módszerekkel segít összefüggéseket keresni adathalmazokban és az ilyen összefüggések alapján jóslatokat adni. Az SQL Server Analysis Services másik funkciója az OLAP-elemzések mellett éppen ez. Nyolc statisztikai algoritmust (ügynevezett adatbányászati modellt) tartalmaz, amelyek a kiszolgálón lévő adatok alapján képesek elemezni és jóslni. Az adatbányászati komponensek elérhetők programozási felületen keresztül is, hogy saját alkalmazásokba is beépíthessük és a Microsoft külön T-SQL kiegészítést is készített (DMX – Data Mining eXtensions), hogy még könnyebben tudjuk megfogalmazni a kérdéseinket. Van viszont egy még ennél is kényelmesebb elérési módja ennek a funkciónak: a már jól ismert Excelből. Létezik ugyanis egy Excel add-in – a Data Mining Additions for Office Excel –, amelyet ha letöltünk a Microsoft honlapjáról és telepítünk, akkor kapunk egy új szalagot az Excelben, amelyről az SQL Server Analysis Services majdnem minden adatbányászati funkciója elérhető. Így könnyen kísérletezhetünk vele, illetve a kollégák számára is nagyon egyszerűen elérhetővé tudjuk tenni.

## Integráció (ETL) és adattárházak

Ha már belejöttünk az adatok megjelenítésébe és elemzésébe, akkor elgondolkodunk azon, hogy milyen jó lenne ezeket az adatokat más forrásból származó adatokkal is összevetni. Mondjuk, két alkalmazás (egy vállalatirányítási rendszer, illetve egy ügyfélkapcsolati rendszer) adatait együtt is érdemes volna elemezni. Ekkor merül fel az adatintegráció lehetősége. Ahhoz, hogy ezeket az adatokat tényleg össze tudjuk egymással vetni, először össze kell gyűjteni őket egy központi helyre. Ezt nevezzük adattárháznak. Az adattárházra jellemző, hogy egységes formában és egy egységesen jó minőségben tartalmazza az adatokat.

Az adattárházak feltöltése a forrásrendszerből az integráció. Közkeletű rövidítéssel ETL-nek nevezik, ami a három fontos lépésre utal:

**Extract – kinyerés.** Az adatokat ki kell olvasni a különböző adatforrásokból, amelyek lehetnek más adatbázisok, szövegfájlok, webszolgáltatások és gyakorlatilag bármilyen más adatforrás.

**Transform – átalakítás.** Általában az ada-

## Mi közülük az adattárházaknak egy csillagos téli éjszakához?

Az adattárházakban az adatokat nem eredeti formájukban tároljuk, hanem az ügynevezett csillagséma szerint. Ez azért hatékony, mert az egyes dimenziók elemeit külön táblák tartalmazzák, amelyeket csak az idegen kulcsok kapcsolnak az elemzendő adatokat tartalmazó tény táblához. Így a tényadatok dimenzióinformációi nincsenek redundánsan tárolva. Ez különösen fontos, hiszen egy adattárházban sok millió vagy akár annál is több tényadatot tárolunk. Ha egy ilyen tény táblát ábrázolunk, köré felrajzoljuk a hozzá kapcsolódó dimenzió táblákat, és összekötjük a kulcsok mentén a kapcsolódó mezőket, akkor egy szép csillag formáldók ki előttünk, ezért a csillagséma elnevezés. A hóhély séma nagyon hasonló, mindössze annyi a különbség, hogy bizonyos esetekben az egyes dimenzió táblák másokon keresztül kapcsolódnak a tény táblához, és az ilyen többszörös kapcsolatok ábrázolva egy hópihére kezdenek hasonlítani. És ahogy nincs két egyforma hópihé, úgy nincs két egyforma adattárház sem.

tokra nem olyan formában van szükségünk, mint amilyenben a forrásrendszerekben vannak tárolva. Szükség lehet egyszerűbb átalakításokra, mint például adattípus- vagy karakterkódolás-konverzió, de akár összetettebb műveleteket is végezhetünk, mint

Mindezen átalakítások mellett szükség van arra is, hogy az adatokat az adattárház sémájának megfelelőre konvertáljuk (lásd a kereset írást a csillag- és hópihé sémáról).

**Load – betöltés.** Az előkészített adatokat ezután be kell tölteni az előmelegített adattárházba.

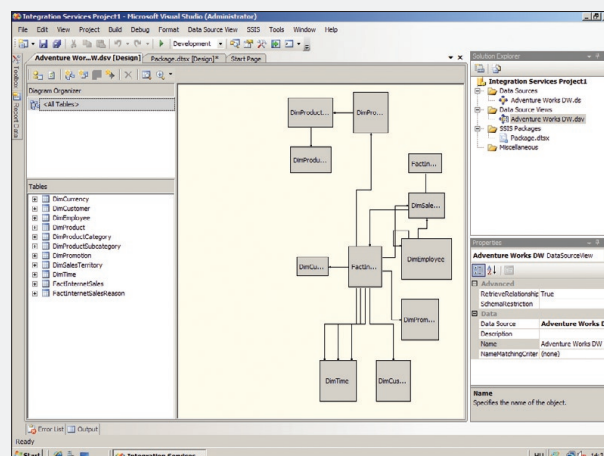
Az SQL Server Integration Servicest persze nemcsak adattárházak feltöltésére lehet használni, hanem minden olyan helyzetben, ahol szükség van nagy mennyiségű adat bevitelére, átalakítására<sup>1</sup>.

## Adattisztítás és Master Data Management

Az adattisztítás egy külön válfaja az üzleti intelligenciának. Része lehet egy integrációs folyamatnak is, de önállóan is megállja a helyét. Lényege, hogy amikor több adatforrásból szeretnénk hasonló adatokat integrálni, akkor előfordulhat, hogy az egyes forrásokban ugyanazok az adatok kicsit eltérően szerepelnek, de szeretnénk, ha a végeredményben ezek egymásra találnának. Képzelnék el például két vállalatot, amelyek egyesülésük után szeretnének egy közös ügyféladatbázist használni. Léteznek olyan ügyfelek, akik mindkettővel kapcsolatban álltak, és olyanok is, akik csak az egyikkel. Esetleg más-más időpontban, ezért más adatok szerepelnek az egyik és a másik rendszerben. Ehhez hasonló szituációt akár egy vállalat két különböző rendszere között is el tudunk képzelni.

Az adattisztítás során (ami akár a napi integrációs folyamat része is lehet) korábban már említett fuzzy lookup és adatbányászat, illetve más eszközök segítségével ezeket a különbségeket lehet kiküszöbölni.

Az adattisztítással szoros összefüggés a Master Data Management, amely azt célozza meg, hogy ha már fáradtságos munkával sikerült tiszta adatokat létrehozni, akkor azokat használjuk is, tehát úgy építsük fel a vállalati rendszereket, hogy a központi, tiszta adatokat vegyék alapul, és ezek alapján dolgozzanak. A Master



### A kockán belüli adatrelációk

az árfolyam-kalkulációk, adategyeztetés stb. Az SQL Server Integration Services például tartalmaz egy fuzzy-lookup nevű funkciót, amellyel a nem pontosan illeszkedő listákat is össze tudjuk rendelni ebben a fázisban.

hát úgy építsük fel a vállalati rendszereket, hogy a központi, tiszta adatokat vegyék alapul, és ezek alapján dolgozzanak. A Master

<sup>1</sup> Hangyál Zoltán cikke a Magazin 30. oldalán



Data Management egy, az adattárházakhoz hasonló komplex, folyamatos üzemű, dinamikus rendszer erre.

## Megjelenítés és csoportmunka

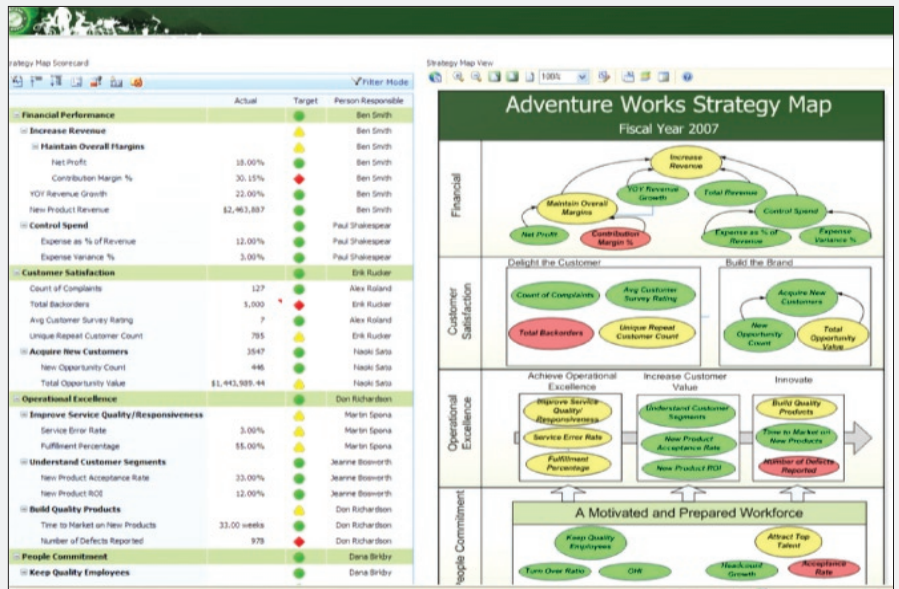
Számos üzletiintelligencia-rendszer „rövidlátó”: az elemzések előállítását után már nem foglalkozik azok sorsával, és a felhasználók hagyományos eszközeire hagyatkozik a döntések meghozatala során. Pedig tudjuk azt, hogy a szervezetek hatékonyságát csökkentő állásidők nem az informatikai rendszerekben keletkeznek. A vállalatok hatékony működését az szolgálja igazán, ha az informatikai rendszerekből érkező adatok feldolgozását is segítjük a megfelelő eszközökkel. Ezért van az, hogy a Microsoft minden üzleti intelligencia lehetőségét integrálja az Office Rendszerrel, hiszen így a felhasználók a számukra ismert és általuk szeretett Office-környezetben tudnak foglalkozni a döntések előkészítésével. Ezt terjeszti ki például az Office Sharepoint Server, amely lehetővé teszi, hogy a döntéstámogató információkat intra- és internetoldalakon tegyük közzé, az Excel Services segítségével a böngészőben is megjelenítsük és formális munkafolyamatokat, jóváhagyást, véleményezést és döntési folyamatokat szervezhessünk körükük. Az Office Sharepoint Server számos módon integrálódik a Microsoft üzletiintelligencia-platformjának többi elemével:

- A Sharepoint-oldalakon megjeleníthetünk SQL Server Reporting Services-jelentéseket, akár paraméterezhetjük is azokat a webpartokon keresztül.
- Az Excel Services segítségével megjeleníthetünk táblákat és grafikonokat a mögöttes adatkapcsolatokra alapozva.
- Definiálhatunk kulcs-teljesítménymutató listákat, amelyek a portálba ágyazva segítik a gyors áttekintést.
- Munkafolyamatokat definiálhatunk a megismert információk feldolgozására.

Az SQL Server üzletiintelligencia-lehetőségei az Office Rendszer eszközeivel kiegészülve képeznek teljes megoldást, és így együtt tudnak egy vállalat hasznára lenni.

## Vállalati teljesítménymenedzsment

Az üzleti intelligencia csúcsmegoldása az, amikor az összes eddig ismertetett eszköz egy egységes alkalmazásban foglaljuk, és ezt állítjuk a vállalat sikerének szolgálatába. Ennek



## A PerformancePoint Server lehetőségei

hangzatos neve a Corporate Performance Management. A Microsoft eszközkészletében ezt a szerepet tölti be az Office PerformancePoint Server (OPPS), amely előre integráltan, könnyen elérhetővé teszi ezeket a funkciókat a vállalatok számára.

## Üzleti modellezés

Lényege, hogy elkészül a vállalat működésének logikai modellje: milyen mutatók jellemzik a működést, ezek milyen paraméterektől függenek, és milyenek szerint lehet elemezni őket. Aki figyelmesen olvasott eddig, az sejtetheti, hogy egy OLAP-kocka modelljét építik meg a szakértők ebben a fázisban. A PerformancePoint Server Business Modeler segít abban, hogy ezt hatékonyan, csoportmunkában, verziókezelten, jogosultságkezelten és sok felhasználó bevonásával lehessen megtervezni. Egy vállalatot akár több modell is leírhat – az egyes részegységeknek, régióknak saját modelljeik lehetnek, amelyek a megfelelő szabályok szerint konszolidálódnak egy központi modellbe.

## Üzleti tervezés

Ha már van egy modellünk, azt meg is kell tölteni adatokkal. Az elsők a tervszámok, amelyek megadják, hogy az egyes mutatók mentén milyen eredményeket szeretne elérni a vállalat. Egy adott szervezeti méret fölött ez már nem egy egyszerű megállapítás, hanem a felülről-lefelé és az alulról-felfelé folytatott egyeztetések rendszerezett hulláma. A PerformancePoint Server egy Excel add-in segít

széggel lehetővé teszi, hogy a szervezet egyes szereplői kommunikálják a központi rendszer felé a saját tervjavasolataikat.

Az üzleti tervek teljesülését ellenőrizni is kell, erre szolgál az OPPS monitoring-funkcionalitása. Ennek segítségével kulcs-teljesítménymutatókat definiálhatunk (lásd az OLAP-os résznél), amelyeket aztán scorecardokba és vezérlőpultokba (dashboard) szervezhetünk. Ezek segítségével a szervezet különböző szereplői a nekik megfelelő összesítő nézetekben láthatják a pillanatnyi teljesítményt és a trendeket.

A teljesítmény állapota nem mindig mutatja meg egyértelműen az okokat és az összefüggéseket. Itt jön képbe az elemzés lehetősége, amit a mögöttes OLAP-modell és infrastruktúra lehetővé tesz. Az OPPS tartalmaz egy webes alapú ProClarity Analytics Servert és a kliensoldalon a ProClarity elemző klienst is.

## Összefoglalás

Ahogy láthattuk, a Microsoft üzletiintelligencia-platformján sok kalapácsunk van. Ezek ismerete segíthet minket abban, hogy megértsük mit is lehet megvalósítani ezen a téren, és tudjuk, hogy hova nyúljunk, amikor legközelebb szembejön velünk egy szög. A magazin többi cikkében épp az első ütések mutatjuk be ezekkel az eszközökkel. Mindenkinek jó kalapálást kívánok!

Nagy Levente  
(Nagy.Levente@microsoft.com)  
Microsoft Magyarország

# ADATTÁRHÁZ-ÉPÍTÉS LÉPÉSRŐL LÉPÉSRE

Ismerje meg az adattárházak feltöltésének titkait, és derüljön fény arra, hogy mit tud az Integration Services 2008!

**M**ég tisztán emlékszem az első BI-projektemre, amely 11 hónapig állt, mert a forrásrendszerek adatminősége olyan silány volt, hogy nem tudtuk betölteni őket az OLAP-kockákba. Hónapokig csak vártunk, vártunk, míg végre olyan szintre kerültek, hogy el tudtuk kezdeni dolgozni.

Persze, ha előtte megvizsgáltuk volna az adatok minőségét, más lett volna a helyzet. Ehhez azonban két dologra lett volna szükség.

1. Zöldfülűként nem kellett volna elhinnem, hogy „Nálunk az adatok jók”.

2. Kellott volna egy olyan kis eszköz, ami a 2008-as Integration Services-ben már van, és amivel gépi úton még betöltés előtt tudjuk ellenőrizni az adatok minőségét.

De ne szaladjunk ennyire előre. Az Integration Services 2008 valóban alkalmas a forrásrendszerek adatminőségének felmérésére, de nem ez az elsődleges feladata. Elsősorban nem erre használjuk.

## Mire való az Integration Services?

Ha megkérdezzük valakit, hogy szerinte mire való az Integration Services, akkor rögtön rá fogja vágni: arra, hogy adatot töltsünk be vele egyik adatbázisból egy másikba.

A legfontosabb feladata tényleg ez. Lehetőleg minél többfajta adatforrásból tudjon adatokat kiolvasni, és azt minél gyorsabban bele tudja pumpálni az SQL Serverbe. És ebben nagyon jó. Nem tudom, hogy tudja-e a kedves olvasó, hogy a jelenlegi betöltési sebességi világrekordot épp ez az Integration Services 2008 tartja, amelyről a cikk szól. Nem kevesebb, mint 1 terabájtnyi text-fájl betöltéséhez csak 25 perc 20 másodpercre van szüksége, ami jelenleg még elég a világszínvonalhoz.

Sokan azt vallják, hogy az SSIS egy általános célra használható adatbetöltő (ETL-) eszköz, csak éppen nem lehet az SQL Server programcsomagtól külön megvásárolni. Ez igaz is meg nem is. (Mint Mátyás király meséjében...)

Igaz, mert tényleg szinte tetszőleges adatforrásból szinte tetszőleges adatszerkezetbe tudjuk segítségével mozgatni az adatokat (hasonlóan egy általános célra kifejlesztett ETL-szoftverhez). De nem igaz, mert az Integration Services csak arra van kihegyezve, hogy az SQL Server adatbázisába villámgyorsan be lehessen tölteni vele az adatokat. Arra már nincs, hogy más adatbázisokba is villámgyorsan át tudja tölteni azokat. Ez nem azt jelenti, hogy nem tudunk idegen adatbázisokba (például: Oracle vagy IBM) adatokat tölteni, hanem azt, hogy ezt csak relatíve lassan tudjuk megtenni.

Ez nem véletlenül van így. A Microsoft elsődleges célja, hogy az adatok bejutását (integrálását) az SQL Serverbe minél könnyebbé és gyorsabbá tegye (innen az Integration Services név). Az, hogy SQL Server-adatokat minél gyorsabban tudjunk áttenni másik adatbázisgyártók ter-

mékeibe – érthető módon – nem kapott akkora fókuszot a fejlesztések során. (Hasonlóan egyébként más adatbázisgyártók ETL-eszközeihez.)

Úgy fest azonban, hogy az Integration Services 2008-ban e téren is változás fog beállni. Ha igazak a hírek, kb. 2-3 hónap múlva az Enterprise verziót használók ingyenesen le fognak tudni tölteni „konnektorokat” Oracle-höz, SAP-hez és a Teradatahoz. Ezek segítségével már az adatok exportálását is gyorsan meg fogjuk tudni oldani, és egyre közelebb kerülünk ahhoz, hogy az Integration Services tényleg egy SQL Serverbe csomagolt általános célú ETL-eszköz legyen.

## Alapozás

Nemsokára belescapunk a lecsóba, és elkezdünk szakszavakkal dobálózni, úgyhogy előtte ejtsünk még néhány szót az SSIS és az adattárház-építés alapjairól.

Nemrég hallottam valakit, aki úgy mutatta be az Integration Services-t, hogy „az Integration Services az, amivé a DTS szeretett volna válni”. Ebben a mondatban minden benne van, hiszen a DTS – Data Transformation Services – az SQL 7.0-ban bemutatkozó, majd az SQL 2000 után kihalt adatbetöltő eszköz volt, az Integration Services pedig egy vadiúj eszköz, zéróról fejlesztve. Szerencsére. Hiszen akik a DTS előnyeiről beszélnek, rendszerint azt mondják róla, hogy az egy „egyszerű eszköz volt egyszerű feladatokra”. Mint amikor egy nőre azt mondják, hogy „aranyos”.

Nézzük a szakszavakat.

Az Integration Services (SSIS) package vagy magyarul csomag az, amit futtatunk. Ez az, ami a betöltési munkát végzi. Hívhatnánk betöltőprogramnak is, de maradjunk a micro-softos terminológiánál, és hívjuk csomag-  
nak. Egy SSIS-csomag valójában egy dtsx kiterjesztésű fájl, amit akár parancssorból is meghívhatunk:

```
dtxexec /f „c:\SSIS-Csomagom.dtsx”
```

Az Integration Services-csomagokat a BI Development Studióban fejlesztjük. Ez az az eszköz – egyébként egy Visual Studio –, amely a vizuális interfészt biztosítja a fejlesztéshez és a futtatáshoz.

Egy Integration Services-csomag két fő összetevőből épül fel. Van az úgynevezett Control flow része, ami a vezérlést végzi és egy data flow része, ami mozgatja az adatokat.

A Control flow valósítja meg a betöltési logikát. Ha például szeretnék egy olyan betöltőcsomagot írni, ami lefuttat egy tárolt eljárást, és ha az sikeresen lefutott, akkor küld egy e-mailt, akkor ezt a BI Development Studióban így kell megírni:

Az 1. ábrán látható dobozokat nevezik taszkoknak. A taszk az SSIS-csomag legkisebb önállóan futtatható egysége, és a taszkok között függőségeket állíthatunk fel. (Ezt reprezentálják a különböző színű nyilak a taszkok között).

Az ábra szerinti csomagban például a levélküldési taszk csak akkor indul el, ha az adatbetöltés sikeresen lefutott.

Az 1. ábra SSIS-csomagja tehát adatokat nem mozgat, csak meghív egy tárolt eljárást, és ha az sikeresen lefutott, akkor küld egy levelet. Ha szeretnénk egy olyan betöltőcsomagot írni, ami lefuttat egy tárolt eljárást, majd meghív egy adatbetöltő folyamatot (data flow), és ha az adatbetöltés nem sikerül, akkor küld egy e-mailt, akkor a 2. ábrán látható módon kell átalakítanunk a csomagot.

Az adatbetöltő taszk komponensekből épül fel, és egy adatforrás-komponense, egy adattranszformációs komponense és egy célkomponense.

A forráskomponens felolvassa az adatokat a forrásrendszerből, a transzformáció-komponens valamilyen módosítást hajt rajta vég-

re, a célkomponens pedig beszúrja az adatokat a céladatbázisba.

Most, hogy már minden szakszó ismert:

## Csapjunk a lecsóba!

Egy ETL-szoftvertől, mint amilyen az Integration Services is, joggal várja el az ember, hogy maximálisan támogassa az adattárházak feltöltése során használt módszereket. Ilyen például az adattárházba érkező rekordok auditálása, verziózása, a mesterséges kulcsok kiosztása vagy az adatok tisztítása. Ezek a fogalmak most még talán kínainak tűnnek, de ha végigolvassa valaki a cikket, akkor tisztában lesz az adattárház-betöltések folyamatával.

Mint a bevezetőben említettem, zöldfülűként elhittem, hogy „az adatok nálunk jók”, és ennek katasztrófális következményei lettek. Ma már máshogy csinálom: visszakerdezek, mint a Windows, hogy „biztos? ☺”, aztán az első lépések egyikeként megprofilozom a forrásrendszereket. Megpróbálom felderíteni azokat az anomáliákat, adatminőségi problémákat, amelyek eddig rejtve maradtak a forrásrendszerekben, hogy ezek ne akkor kerüljenek napvilágra, amikor már minden kész, csak fel kéne tölteni az adattárházat, vagy a BI-rendszert.

Ugorjunk neki, és nézzük meg, mi az (1. ábra).

## Adatprofilozás

Az adatprofilozás az a folyamat, amelynek során megvizsgáljuk a forrásrendszerek adatait, azokról statisztikákat készítünk (például hány NULL értéket tartalmaznak az oszlopok), és információkat gyűjtünk az adatok minőségéről (mennyire tiszták az adatok).

Az adatprofilozás során találkozunk először az adattárház-tervező az éles adatokkal. Ekkor kezd el kialakulni benne egy kép az adatminőségről, és a profilozás befejezésekor

már elegendő információja lesz ahhoz, hogy meg tudja tervezni a fizikai adatmodellt (milyen adattípusokat kell majd használni az adattárházban).

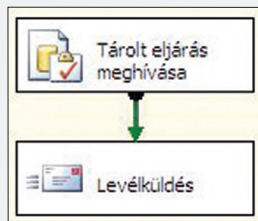
Az adatok profilozását gépi úton végzzük. Eddig vagy saját magunk írtunk olyan szkripteket, amelyek elvégzik az adatok profilozását, és a végén kiköpnék egy riportot a profilozás eredményéről, vagy egy célszoftverre bíztuk mindezt. Eddig, mert mostanra a Microsoft kifejlesztett egy adatprofilozó alkalmazást, amit jó szokása szerint becsomagolt az Integration Services 2008-ba. (Tette mindezt úgy, hogy közben változatlanul hagyta az SQL Server árát. A 2008-as SQL Server pont annyiba kerül, mint 2005-ös elődje.)

Az Integration Services 2008 adatprofilozó taszkja a következő profilozó eljárásokat támogatja:

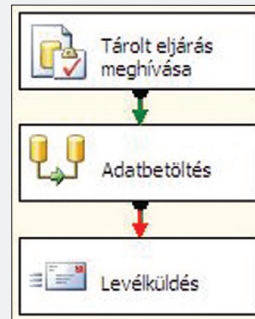
- a kitöltöttséganalízis segítségével képet kaphatunk arról, hogy egy oszlop hány százaléka tartalmaz null értéket;
- az adathosszeloszlás-elemzés megmutatja nekünk, hogy

hány darab 1 hosszú, 2 hosszú, 3 stb. hosszú szöveget tartalmaz az oszlopunk (lásd a 4. ábrát);

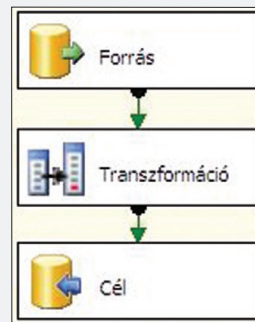
- kulcsképesség-elemzés, amellyel meggyőződhetünk arról, hogy egy kulcsnak gondolt mező tényleg az-e (különösen szövegfájlokban keresztül érkező adatoknál lehet erre szükségünk);
- minták keresése, amellyel telefonszámok, rendszámok, irányítószámok vagy egyéb kötött struktúrájú, ugyanakkor szabadszöveges mezőként tárolt adatainkat analizálhatjuk;
- oszlopstatisztikák, amelyek visszaadják nekünk az oszlopok statisztikai jellemzőit (mint például az oszlop minimuma, maximuma, átlaga vagy szórása);
- értékeloszlás-analízis, amely kimutatja például, hogy hány Béla van a keresztnév oszlopban;
- összefüggés-vizsgálat, amellyel hierarchiákat kereshetünk táblákon belül;



1. ábra. Control flow az SSIS-csomagokban



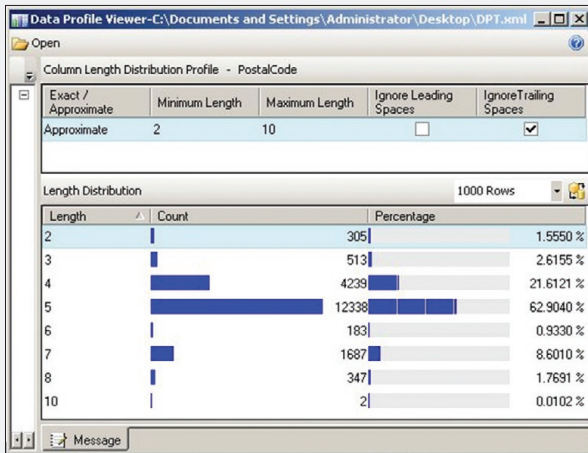
2. ábra. Átalakítva



3. ábra. Az adatbetöltő taszk

- olyan részhalmazok keresése, amelyekkel adatkapcsolatokat deríthetünk fel két tábla között.

Az Integration Services adatprofilozó taszkját azonban nemcsak a forrásrendszerek



4. ábra. Adathosszozslás-elemzés az Integration Services adatprofilozó taszkjával

felmérésére használhatjuk, hanem az adattárházba érkező adatok ellenőrzésére is. A napi betöltések során még a betöltések előtt megvizsgálhatjuk például azt, hogy a frissen érkező adatok szórása milyen képet mutat a már az adattárházban lévő adatok szórásához képest, és ha jelentős eltérést tapasztalunk, akkor dönthetünk a betöltés elhalasztásáról.

Ha viszont minden rendben a betöltésre váró adatokkal, akkor kezdjük el feltölteni az adattárházunkat. Első lépésként válogassuk le a lehetőleg csak az utolsó leválogatás óta keletkezett adatokat a forrásrendszerekből, és töltsük be őket egy ideiglenes, úgynevezett staging adatbázisba. Ezt a folyamatot nevezik extract fázisnak, és ez a teljes ETL- (adattárház-betöltési) folyamat (Extract, Transform, Load) első fázisa.

## Az E betű az ETL szóból

Az adattárházak feltöltése során egy forrásrendszeri adat sok lépcsőn megy keresztül, amíg eljut a felhasználók számára is látható végleges adatbázisba. Először bemásoljuk egy átmeneti (Stage) adatbázisba, majd elvégzünk rajta egy-két átalakítást, és végül begyömöszöljük az adattárházba. Ezt a folyamatot mutatja az 5. ábra.

Az adattárház-feltöltések első (Extract) fázisában gyakorlatilag a forrásrendszerek szerkezetével megegyező módon átmásoljuk azok

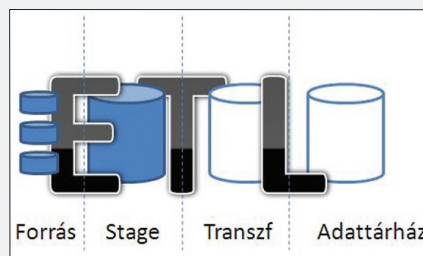
adatait. Semmi átalakítás, semmi módosítás. Az adatokat úgy, ahogy vannak, átemeljük a saját szerverünkre. Csak arra kell törekednünk, hogy a forrásrendszereket minél rövidebb ideig és a lehető legkevésbé terheljük, és lehetőleg csak azokat az adatokat hozzuk át, amelyek még nem szerepelnek az adattárházban. Aztán ha már nálunk vannak az adatok, és leszakadtunk a forrásrendszerekről, akkor kezdődhetnek az erőforrás-igényes átalakítások.

## Az új adatok leválogatása

Egy adattárházat nem lehet mindig nulláról feltölteni, mert olyan adatmennyiségekkel kell dolgoznunk, amelyek teljes betöltése több hetet vehet igénybe.

Így az egyik napi betöltés még be sem fejeződik, és máris belefutnánk a következő betöltésbe. (Úgy járnánk, mint szegény Lewis Fry Richardson meteorológus, aki a 20-as években 1 hónap alatt számolta ki, hogy mennyivel fog változni a légköri nyomás 6 óra múlva.) De az adattárház újratöltésének van másik hátulütője is: elvesztenénk azokat az információkat, amelyek historikusan csak az adattárházban vannak meg, a forrásrendszerekben már nem.

Azért, hogy ne kelljen átcibálni mindig



5. ábra. Az E

reggel az összes forrásadatot, ki kell találnunk valamilyen adatleválogatási módszert, amelynek segítségével meg tudjuk állapítani, hogy melyek azok a rekordok, amelyek újak, vagy megváltoztak az utolsó betöltés óta.

Ha a forrásrendszerünk egy SQL 2008-as adatbázis, és az üzemeltetők bekapcsolják nekünk az úgynevezett Change Data Capture

szolgáltatást, akkor nagy szerencsénk van. Ez a Change Data Capture ugyanis elkapja a változásokat a forrásrendszerben, és kitekíti őket egy külön táblába, így nem kell azzal bajlódnunk, hogy kitaláljuk, mi változott meg az utolsó leválogatás óta. Ha nincs ilyen szerencsénk, akkor nekünk kell kitalálnunk valamilyen módszert az új vagy megváltozott adatok leválogatására (például időbélyeg vagy rekordazonosító alapján történő szűrés). S ha megvan a módszer, akkor indulhat a leválogatás és az új adatok betöltése az úgynevezett Staging adatbázisba.

## Audit

Adattárházba rekordot úgy nem töltünk be, hogy ki ne egészítenénk a származására vonatkozó információkkal. Legalább annyit kell tudnunk egy adattárházban csücsülő rekordról, hogy:

- melyik forrásrendszerből (esetleg táblából) jött az adott rekord;
- melyik (mikori) betöltéssel került be;
- és melyik SSIS-csomag töltötte be.

E három információ már elegendő ahhoz, hogy mindent tudjunk a betöltött rekord eredetéről, amit hibakeresésnél vagy egy esetleges hibás betöltés visszavonásánál használhatunk.

Ezeket, a származásra vonatkozó információkat nevezük auditinformációknak. Az auditinformációkat a bejövő rekordokhoz az Integration Services Audit névre hallgató taszkjával tudjuk hozzáadni. Ez a taszk nemcsak egy általunk meghatározott szöveget vagy kifejezést képes a befelé áramló rekordokhoz fűzni, hanem olyan belső változók tartalmát is, mint a gép vagy az SSIS-csomag nevét vagy az adott futás egyedi azonosítóját (amit remekül használhatunk kulcsként bonyolultabb auditrendszerek kialakításához).

Nos. Minden szükséges adat az auditinformációkkal együtt ott csücsül a saját szerverünk staging adatbázisában. A forrásrendszerekről leszakadhatunk, kezdődhet az adatok átalakítása, transzformálása.

## A nagy T betű az ETL szóból

Az adatok transzformálása során két fő feladatot hajtunk végre:

- megtisztítjuk
  - és előfeldolgozzuk őket,
- hogy a következő (Load) fázis betöltési munkáit – amelynek során végleges helyükre ke-

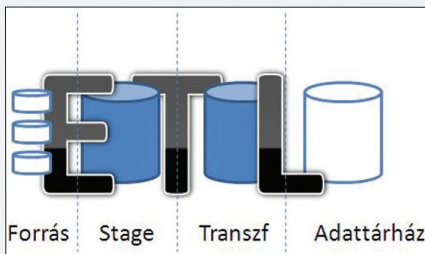
rülnek majd az adatok az adattárházban – a lehető legegyszerűbbé tegyük.

## Adattisztítás

Az Integration Services két olyan taszkot is tartalmaz, amelyek segíthetnek nekünk a pontatlan vagy hiányos adatok megtisztításában, a duplikált adatok összefésülésében. Ez a két taszk a fuzzy grouping és a fuzzy lookup taszkok. Míg az előbbi a duplikált adatok összehozására (például két azonos, de különböző forrásrendszerekben is szereplő vevőből egyet csinálni), addig az utóbbi a hiányos, elgépelt adatok kitisztítására szolgál.

Mindkét taszk hasonlósági alapon tisztítja az adatokat. Ha két adat elér egy általunk meghatározott hasonlósági indexet, akkor a taszkok ennek megfelelően javítják a hibás adatokat. Ha a hasonlóság kisebb, mint az általunk meghatározott küszöbszám, akkor marad a kézi tisztítás.

Fontos tudni, hogy e két taszk nyelvfüggetlen, azaz nem veszi figyelembe a magyar



7. ábra. AT

és egyéb nyelvi sajátosságokat. Neki a sírály és a siráj szó között két karakter eltérés lesz, és nem fogja észrevenni, hogy a két szó ugyanazt jelenti. (Ahhoz tudnia kéne, hogy magyarul kétfajta, jé hangot jelölő betűt is használunk.)

A fuzzy grouping és lookup taszkok már a 2005-ös Integration Services-ben is léteztek, azok nem a 2008-as Integration Services újdonságai. Ami újdonság e téren, az a webszervizek hívhatósága a data flow taszkon belülről. Látszólag persze a webszerviz hívhatóságának semmi köze az adattisztításhoz. De ez csak a látszat.

Webszervizt eddig is tudtunk hívni SSIS-ből, ez nem újdonság. A nagy újdonság az, hogy ezt immáron nemcsak a Control Flow-n

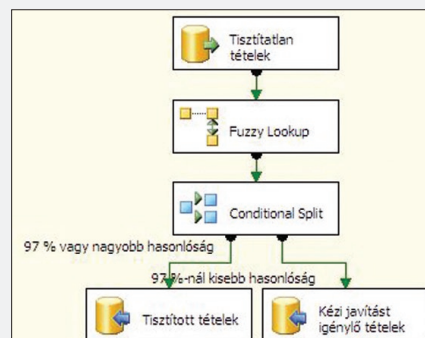
belülről, hanem a Data Flow-n belülről is megtehetjük, azaz a betöltés közben minden egyes sort elküldhetünk egy webszerviz felé. Tegyük fel, hogy önnek címadatokot kell tisztítania. Milyen lehetősége volt eddig? Fogta, letöltötte a Posta honlapjáról az irányítószámok nevű xls-t, abból épített egy referencia-várostörzset, és a fuzzy lookup taszkkal kikereszte az adatokat a várostörzsből.

Ma már azonban a lehetősége megvan rá, hogy meghívjon egy olyan webszervizt, amely megtisztítva visszaküldi önnek a helyes címeket. Nem önnél van a referencia-adatbázis, nem ön tartja azt karban, ha-

lalom valaki más. Valaki más, aki tudja, hogy a József A. utca az a József Attila utca, és hogy a Bp. az a Budapest. Ma persze még nem tudok ilyen magyar nyelvet támogató webszervizről, de sokat gondolkodtam rajta, hogy kéne csinálni egyet. Ezt ugyanis nemcsak az adattárházak, hanem a web- és az alkalmazásfejlesztők is használhatnák, ami már egy kicsit nagyobb piac.

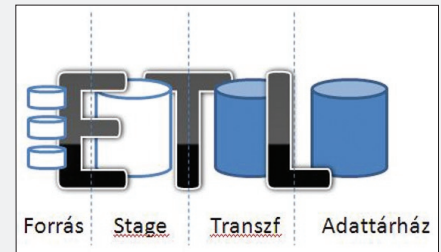
De a suszter maradjon a kaptafájánál, úgy-hogy térjünk vissza az adattárházakhoz, hiszen adataink már tiszták, és alig várják, hogy egyre beljebb töltsük őket az adattárházba, egyre közelebb kerüljenek a felhasználókhoz.

Az adattárház-feltöltés következő lépése az adatok előfeldolgozása. Az előfeldolgozás során a friss adatokat átranzformáljuk az adattárház formátumának megfelelő alakra. Néha letároljuk őket egy ideiglenes adatbázisban (nevezzük ezt „Transzf” adatbázisnak), néha röptében töltsük őket tovább az adattárházba. Most a könnyebb magyarázhatóság kedvéért tároljuk le őket ebben a köztes transzf-adatbázisban.



8. ábra. Adattisztítás a fuzzy lookup taszkkal

Ebben az esetben a transzf-adatbázis szerkezete – néhány oszlop kivételével, amiről később lesz szó – tökéletesen egyezik az adattárház szerkezetével. Mindkét adatbázisban megtalálhatóak ugyanazok a dimenzió- és tény táblák, ugyanazok az oszlopok, csak míg az adattárházban több évre visszamenőleg tartalmaznak adatokat, addig a temporális transzf-adatbázis csak az utolsó betöltés óta keletkezett friss adatokat tartalmazza.



9. ábra. Az L

Elérkeztünk oda, hogy az adatok megtisztítva, az adattárházba töltéshez előkészítve várják, hogy betöltsük őket a végleges helyükre: az adattárházba.

## A Load fázis

Az eddig bemutatott átalakítások mind-mind csak előfeldolgozások voltak. Csak azt a célt szolgálták, hogy az adatokat könnyen be tudjuk tölteni az adattárházba. (Abba az adatbázisba, amelyet a felhasználók használni fognak.)

Mielőtt rátérnénk erre az úgynevezett „load” fázisra, essen néhány szó az adattárház adatszerkezetéről, hogy tudjuk, mégis milyen szerkezetbe kell betölteni az adatokat.

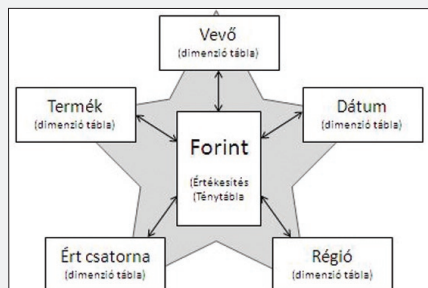
Az adattárházak adatszerkezete lehet normalizált, vagy lehet csillagsémás. Mindkét adatszerkezetnek van előnye, és van hátránya is a másikkal szemben, de mi most csak a csillagsémás adattárházak építésére koncentrálnak.

A csillagséma központi eleme a tény tábla, amely tartalmazza a mutatószámokat, és ekörül helyezkednek el csillag alakban a dimenzió táblák, amelyek leírják a tény táblában szereplő mutatószámokat.

A klasszikus példában a tény táblában tároljuk, hogy mennyit értékesítettünk, a dimenzió táblákkal pedig leírjuk, hogy az adott értékesítés milyen termékből, melyik vevőnek és mikor történt.

A tény tábla és a dimenzió tábla között a kapcsolatot egy általunk generált, jelentés

nélküli, úgynevezett mesterséges kulcs teremti meg. Bár használhatnánk a dimenzióelemek forrásrendszeri kulcsát, mint például a vevőkódot vagy a cikk-kódot, de nem ezt tesszük. A miértéről hamarosan, most nézzük először a folyamatot:



10. ábra. Csillagséma

Tegyük fel, hogy bejön egy 311001-es vevőkód a forrásrendszerből. Megnézzük, hogy ez a vevő létezik-e már az adattárházban, és ha nem, akkor beszurjuk a dimenzióáblába. Kap egy új azonosítót, és a ténytáblához ezzel az azonosítóval fogjuk majd kötni. Ezt az azonosítót nevezzük mesterséges kulcsnak, helyettesítő kulcsnak vagy surrogate key-nek.

## Mesterséges kulcs generálása

A mesterséges kulcs lesz a dimenzióáblában szereplő sorok egyedi azonosítója. Mint az előbbiekben említettem, az adattárházakban ezt az úgynevezett mesterséges kulcsot használjuk a dimenzióelemek egyedi azonosítójaként, és ezen a kulcson keresztül fognak kapcsolódni a ténytáblákhoz.

Mesterséges kulcs gyanánt jelentés nélküli egész számokat használunk. Lehet ez egy, az adatbázis által karbantartott automatikusan növekvő egész szám (Identity), vagy generálhatjuk mi is az Integration Services segítségével.

Miért nem használjuk a forrásrendszerekből már mindenki által jól ismert vevőkódot? Miért kell helyettük egy jelentés nélküli mesterséges kulcsot használnunk?

A mesterséges kulcs elsődleges feladata, hogy segítségével meg tudjuk oldani az adattárházba érkező rekordok verziójását. Ha például a vevőnek megváltozik a telephelye – és ez az információ fontos számunkra –, akkor a cím felülírása helyett eltároljuk a vevőnek mind a két állapotát: az 1-es mesterséges vevőkóddal tároljuk a vevőt a régi címével, a 2-es mesterséges vevőkóddal pedig beszurjuk az új címével.

A mesterséges kulcsok elsődleges szerepe tehát az, hogy segítségével megoldhassuk a dimenzióelemek változásainak nyomon követését. Mindemellett a mesterséges kulcs használatával:

- elszakadhatunk a forrásrendszerek kódolásától, így azok esetleges változását (például egy forrásrendszercserét) viszonylag fájdalommentesen átvészelhetünk;
  - egyszerre több forrásrendszerből jövő „vevőkódot” is fel tudunk dolgozni;
  - felvehetünk a dimenzióba olyan dimenzióelemeket, amelyek nem léteznek a forrásrendszerekben;
  - az egész számként tárolt mesterséges kulcs hatékonyabb, mint a szöveges természetes kulcs: kevesebb helyet foglal, könnyebben megbirkózik vele a relációs adatbázis-kezelő és az Analysis Services is, így hatékonyabb lesz a lekérdezés és a feldolgozás is.
- Most, amikor tudjuk, hogy a forrásrendszerek természetes kulcsát ki kell cserélnünk egy általunk generált mesterséges kulcsra, már csak egy kérdés maradt: Hogyan?

## Slowly Changing Dimensions (SCD)

A Slowly Changing Dimensions – vagy más néven SCD – igazából egy technika, egy olyan technika, amelynek segítségével nyomon követhetjük dimenzióelemeink változását. Az SCD technikának két tiszta formája létezik:

- az SCD type-1 módszer lényege, hogy nem követi a dimenzióelemek változását, nem őrzi meg például a vevők korábbi jellemzőit (mint például a telephely), hanem azokat helyben felülírja;
- az SCD type-2 módszer lényege, hogy a dimenzióelem megváltozása esetén létrehozza annak egy újabb verzióját, nem írja felül a vevő korábbi telephelyét, hanem létrehoz egy új vevőt az új telephellyel, úgy, hogy közben megmarad a régi is.

Megpróbálom elmagyarázni egy másik példán keresztül is. Tegyük fel, hogy vevőinkről – akik most legyenek személyek – összesen két információt tárolunk az adattárházban: 1. házasek, 2. mi az e-mail-címük. És most kezdünk el gondolkodni a felhasználók fejével! Fontos nekünk, hogy tudjuk, mi volt a vevők e-mail-címe, mielőtt megváltozott volna? Valószínűleg nem. És azt fontos tudnunk, hogy mikor változott meg a vevők családi állapota? Bizony fontos, hiszen ha sze-

retnek olyan lekérdezést készíteni, hogy hány forintot költöttek a házások és a nőtlenek, akkor tudnom kell, hogy mennyit vásárolt Gipsz Jakab, amíg nőtlen volt és mennyit vásárolt miután bekötötték a fejét.

Modellezzük le mindezt. Gipsz Jakab mint nőtlen vásárló bekerül az adattárházvevő dimenzióáblájába, és megkapja az 50-es mesterséges kulcsot:

ID	Ügyf	Cs. Áll.	E-mail	Érv. -tól	Érv. -ig
50	Gipsz J.	Nőtlen	GJ@MOL.hu	1900.01.01	9999.12.31

11. ábra. Gipsz Jakab, a vásárló

Nem sokkal később Gipsz Jakab megházasodik. Mivel tudni szeretnénk, hogy mennyit vásárolt Gipsz Jakab, amíg nőtlen volt, és mennyit vásárolt majd, mint nős ember, ezért felvesszünk egy másik Gipsz Jakabot a vevő-dimenzióáblába. A régi Gipsz Jakab rekordját „lejárattjuk”, azaz beírjuk, hogy Gipsz Jakab a mai napig bezárólag nőtlen volt, és az új Gipsz Jakabot pedig felvesszük az 51-es mesterséges kulccsal:

ID	Ügyf	Cs. Áll.	E-mail	Érv. -tól	Érv. -ig
50	Gipsz J.	Nőtlen	GJ@MOL.hu	1900.01.01	2007.05.15
51	Gipsz J.	Nős	GJ@MOL.hu	2007.05.16	9999.12.31

12. ábra. Gipsz Jakab 2.0

Telnek-múlnak a mézes hetek, és Gipsz Jakab e-mail-címe megváltozik. Érdekel minket, hogy mi volt Gipsz Jakab régi e-mail-címe? Nem. Minket csak az érdekel, hogy mi Gipsz Jakab mostani e-mail-címe. Ezért mindkét Gipsz Jakab (a nős és a nőtlen) e-mail-címét is megváltoztatjuk az újra. Íme:

ID	Ügyf	Cs. Áll.	E-mail	Érv. -tól	Érv. -ig
50	Gipsz J.	Nőtlen	GJ@OTP.hu	1900.01.01	2007.05.15
51	Gipsz J.	Nős	GJ@OTP.hu	2007.05.16	9999.12.31

13. ábra. Gipsz Jakabok összesítő

Ezek voltak azok a módszerek, amelyekkel nyomon tudjuk követni a dimenzióelemeink változásait. Nézzük meg, hogyan ültethető át mindez a gyakorlatba.

Az Integration Services tartalmaz egy Slowly Changing Dimension nevű taszkot, amelynek a feladata pontosan a fenti technika megvalósítása. Ez remek. A való élet azonban a fenti eseteknél sokkal cifrábbakat is produkál. Képzeld el egy olyan dimenzióelemet, amelynek változásait csak egy

bizonyos idő után akarjuk nyomon követni. Miután megtörtént rá például az első értékesítés. Szerencsére a Slowly Changing Dimension taszk erre is fel van készítve. Erre mutat példát az alábbi, valós életből vett SSIS-csomag Data Flow taszkja.

Mint a 14. ábrából talán látszik, a Slowly Changing Dimension taszk elsődleges feladata, a bejövő rekordok összehasonlítása az adattárházban csúcslévelel, és az, hogy eldöntse, melyek az új rekordok, melyek a megváltozottak, és ha megváltoztak, akkor szétválassza őket: melyeket kell egy új mesterséges kulccsal beszúrni a dimenziótáblába, és melyeket kell csak egyszerűen felülírni.

Vessen még egy pillantást a 14. ábrára. Mivel ez egy data flow taszk belseje, ezért itt a nyílak az adatfolyam irányát mutatják. A Select doboz felszedi a lemezlől az adatokat, majd továbbküldi az SCD taszknak. Az SCD taszk elküldi balra az új rekordokat, jobbra pedig azokat, amelyek nem változtak az utolsó betöltés óta. Lefelé mennek azok, amelyeken valamilyen változás történt. Nekünk már csak le kell kezelni a változásokat: beszúrni az újakat vagy a type-2 szerint változókat, frissíteni a type-1 szerint historizált attribútumokat és azokat, amelyek változását addig nem akarjuk követni, amíg meg nem történt az első értékesítés (Inferred member ág).

Ezzel feltöltöttük a dimenziótáblánkat, már csak a tény táblák betöltését kell megoldanunk.

## Lookup

Amikor egy rekordot betöltünk az adattárház tény táblájába, akkor az abban szereplő természetes kulcsokat ki kell cserélnünk a dimenziótáblákban található mesterséges kulcsokra. Tegyük fel, hogy Gipsz Jakab vásárolt valamit, és a tranzakció összege megjelenik a tény táblába betöltendő rekordok között. Ebben az esetben meg kell néznünk, hogy a vásárlás időpillanatában Gipsz Jakab mely rekordja volt érvényes a dimenziótáblában (ezt megmondják nekünk az érvényesség kezdete és vége oszlopok), és Gipsz Jakab természetes kulcsát ki kell cserélni a dimenziótáblában található mesterséges kulcsra. (Ha Gipsz Jakab házas volt a vásárlás pillanatában, akkor a tény tábla sor megkapja az 51-es mesterséges kulcsot, ha nem, akkor megkapja az 50-est.) Így az adott vásárlás Gipsz Jakab vásárláskori családi állapotával lesz összekap-

csolva, lehetőséget teremtve így a nőtlenek és a házások forgalmának pontos kimutatására.

Ezt a folyamatot, amikor a tény táblák betöltése során az azokban szereplő természetes kulcsokat kicseréljük azok megfelelő mesterséges kulcspárjaikra, lookup-nak nevezzük. Ezt a lookup-ot megvalósíthatjuk adatbázisoldalon és az SSIS Lookup taszkjának segítségével is. Melyiket használjuk?

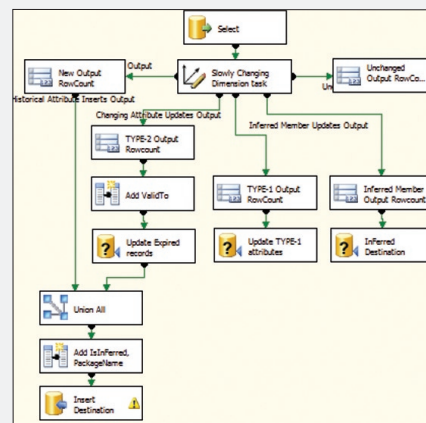
Sokszor az adatbázis-kezelő gyorsabban oldja meg ezt a problémát, mint az Integration Services, de sokszor nem. A 2005-ös Integration Services használatakor voltak ökölszabályok, hogy mikor nem érdemes SSIS-t használni (például ha túl sok olyan elemet tartalmaznak a bejövő tény táblasorok, amelyek nem szerepelnek a dimenziótáblában), de ezeket a szabályokat a 2008 SSIS fel fogja rúgni. Még nincs éles tapasztalatom a 2008-as lookup taszkkal, de dokumentációkból kiderül, hogy egyiken jelentősen megnövelték a teljesítményét. Az egyik ilyen teljesítménynövelő fejlesztés egyébként pont az említett sok ismeretlen elemeket tartalmazó tény táblák feldolgozásának hatékonyságán javít. A nagygyű azonban kétség kívül a lookup taszk gyorsítótárának továbbfejlesztése, amelynek eredményeképpen jelentősen felgyorsulnak majd a betöltéseink.

Az adattárház elkészült. A csillagsémáink korrektül fel vannak töltve adatokkal. Már csak fel kell összegeznünk az OLAP processz taszkkal vagy egy általunk írt szkripttel az Analysis Services adatkockáit, időzítenünk kell a betöltéseket, hogy azok mindennap lefuthassanak, és a felhasználók máris elkezdhetik lekérdezni adattárházunk mind relációs, mind többdimenziós oldalát.

## Hátország

Minden reggel, a betöltések lefutása után az adattárház-üzemeltetőnek rá kell néznie a betöltés eredményére, és ha az valamilyen okból nem sikerült (például megtelt a vinyó ☺), akkor tájékoztatni kell a felhasználókat, hogy ma csak a tegnapi adatok érhetőek el, a tegnapiak még nem. Ehhez pedig az üzemeltetőnek szükségük van egy olyan monitoring rendszerre, ahol az összes adattárházban zajlott folyamatot nyomon tudják követni. Mely csomagok futottak le, melyek nem, melyik jelzett hibát, melyik nem indult egyáltalán, melyik hány rekordot töltött be, melyik mennyi ideig futott, és ez mennyivel több, mint a megszokott, és még sorolhatnám.

Egy ETL-eszköztől, mint például az SSIS azt is el kell várunk, hogy biztosítsa a háttér szöveget az üzemeltetőknek. Miközben ezerral darálnak a betöltések, arról is gondoskodni kell, hogy az adattárház eseményeit folyamatosan nyomon tudjuk követni. Bár az SSIS-nek van beépített naplózási funkciója, ez annyira részletes, hogy ahhoz csak akkor kell



14. ábra. Késleltetett változáskövetés

nyúlnunk, ha tényleg baj van. Ezért célszerű ezt a naplót kiegészíteni saját magunk által írt naplózással is, amihez az SSIS minden segítségét megad.

Minden SSIS-csomagnak, taszknak van OnError, OnFinish és még sorolhatnám eseménye. Ezeket kiegészítve például az audit, a row count taszkokkal és a beépített rendszerváltozókkal olyan betöltési naplót tudunk készíteni, ami mind a fejlesztők, mind az üzemeltetők igényeit maximálisan ki fogják elégíteni.

Összefoglalva: egy forrásadatnak sok-sok lépcsőn, úgynevezett ETL-alrendszeren kell keresztülmennie ahhoz, hogy eljusson végleges helyére, az adattárházba. Először át kell esnie a kötelező szűrővizsgálatokon (az adatprofilozáson), ahol analizáljuk minőségét és szerkezetét. Ha nincs óriási probléma, akkor megtervezük a leválogatás módját, majd betöltjük őket az úgynevezett staging adatbázisba. Itt bevárjuk a még más forrásrendszerből érkező adatokat, tisztítunk rajtuk egy kicsit, előkészítjük őket az adattárházba való betöltésre. Végül új mesterséges kulcsokat adva betöljük őket végleges helyükre, az adattárházba.

Kővári Attila  
(www.biprojekt.hu)

BI-bevezetési tanácsadó, SQL Server MVP

# ANALYSIS SERVICES

Az üzletiintelligencia-rendszerek alapja.

Ha szeretné, hogy önt se zaklassák többé a felhasználók riportigényeikkel, akkor olvassa el a cikket.

**N**emrég egy előadáson jártam, ahol egy nyugdíjas mérnök-közgazdász beszélt az OLAP-ról és az üzleti intelligenciáról. Az első diáján ez állt: Mi az OLAP? Hmmm. Gondoltam, most jön a jól ismert, E. F. Codd által még 1985-ben alkotott OLAP-definíció vagy Nigel Pendse FASMI (Fast Analysis of Shared Multidimensional Information) meghatározása. Ehelyett az öregúr kiállt, és mindenféle kódosítás nélkül, nemes egyszerűséggel azt mondta, hogy az „OLAP-ot szoftveroldalról érdemes meghatározni, mert kézzelfoghatóan szoftver formájában jelenik meg”, és elmondta szépen érthetően, hogy mi az OLAP.

## Mi az OLAP?

Mi is ezt az utat fogjuk végigjárni, és szoftveroldalról fogjuk megközelíteni az OLAP-ot, azaz esetünkben az Analysis Services 2008-at. Meg fogjuk vizsgálni, hogy mire való, mint adatbázis-kezelőnek milyen jellemzői vannak, és milyen hasznot realizálnak azok a vállalatok, amelyek a bevezetése mellett döntenek.

Előljáróban azonban annyit el kell árulnom, hogy az OLAP egy speciális adatbázis-kezelő, amely szabványos adatszerkezete miatt rém egyszerűen lekérdezhető az informatikában járatan üzleti szakemberek számára is. Ennek persze ára van, amiről lesz még szó bőven, de előtte érdemes megismerni a motivációt, ami másokat arra sarkallt, hogy belefogjanak egy OLAP-rendszer bevezetésébe.

## Miért vezetnek be a vállalatok OLAP-rendszereket?

Az idevonatkozó „szakirodalom” szerint azért, hogy

- csökkenjenek a papírköltségek;
- csökkenjenek a bérköltségek;
- javuljon a termelékenység;
- növekedjen a bevétel.

Vajon tényleg ezért akarnak a cégek OLAP-rendszereket bevezetni? Ki kell ábrándítanom mindenkit, és azt kell, hogy mondjam: nem. Ezek mind hangzatos marketingszövegek, de még nem hallottam Magyarországon olyan céget, amelyik a fenti okokat választotta volna e kérdés kapcsán. Ehelyett ilyeneket említettek:

- „Beléptünk egy új piaci szegmensre, amelyet sok kis vevő jellemez, és ezek forgalmát már képtelenek voltunk a meglévő eszközökkel monitorozni...”
- „Az xxx riport elkészítéséhez kevés lett az Excel 65 ezer sora” – Jó, ez nem egy mai példa, de remekül szemlélteti azt, hogy a felhasználók hogyan fejezik ki magukat, amikor azt akarják mondani, hogy kinőttek már az Excel korlátait.
- Volt ilyen is: „Az anyavállalat adattárházában egy új egyéni hierarchia létrehozása több hónapot vesz igénybe, és nekünk ezt nincs időnk kivárni”.
- „Jogosultsági problémák miatt egész egyszerűen nem férhet hozzá

egy személy minden vagy a számára fontos adatforráshoz. Nem kaphat XY jogosultságot a Z forrásrendszerhez csak azért, mert egy szám kell neki belőle. Kell nekünk egy rendszer, ahol saját magunk tudjuk akár cellaszinten szabályozni a jogosultságokat.”

- Egy controller problémája: „Az adatszolgáltatók vagy az adatgazdák nem feltétlenül a kérdésemre adják meg a választ, ami miatt megnőnek az amúgy is hosszú válaszidők”.
- Vagy: „Egyre több ad hoc lekérdezésünk van, amelyet az adatszolgáltatók már nem tudnak kielégíteni”.
- És végül: „Szeretnénk a zárást követően napokon belül riportálni a menedzsmentnek. Most hetekig tart a jelentések elkészítése”.

Ezek közül egy sem célozta meg a költségcsökkentést. Mindenki saját folyamatait akarja hatékonyabbá tenni. Céljuk, hogy a rendszergazdák, az elemzők is azt csinálják, amihez a legjobban értenek. Ezáltal, persze, végső soron csökkennek majd a költségek, és nőnek a bevételek, de az a hatékonyság növekedésének lesz az eredménye, nem közvetlenül az OLAP-é.

Most, hogy már tudjuk, mit várnak a cé-

Ad hoc analysis	65,60%
General Data Warehouse reporting	48,50%
Sales and Marketing Analysis	42,30%
Budgeting an Planning	41,40%
Dashboards and EIS	37,20%
...	...
ClickStream Analysis	2,10%

Forrás: Nigel Pendse, BI Survey 7

1. ábra. Mire használják az OLAP-ot?



gek bevezetés előtt, nézzük meg, hogy mire használják az OLAP-rendszereket immár bevezetés után.

## Mire használják az OLAP-rendszereket?

Nigel Pendse, aki rendszeresen kutatja az OLAP-rendszerek piacát, a 2008-as BI Survey 7-ben azt publikálta, hogy a megkérdezett felhasználók az OLAP-ot elsősorban ad hoc elemzésre (66 százalék) és általános adattárházi jelentéskészítésre (48 százalék) használják.

## BI, DSS, MIS, VIR, EIS, CPM...

Ha egy másik megközelítésből vizsgáljuk meg, hogy mire használják az OLAP-rendszereket, akkor azt találjuk, hogy OLAP-szívek doboznak az

- üzletiintelligencia-rendszerekben (BI – Business Intelligence);
- döntéstámogató rendszerekben (DSS – Decision Support System);
- vezetői információs rendszerekben (VIR vagy angolul MIS – Management Information System).

De előszeretettel használják az OLAP-technológiát

- analitikus CRM- (Customer Relationship Management) rendszerek alapjaként;
- kiegyensúlyozott mutatószám-rendszerek kialakításához (BSC – Balanced Scorecard);
- CPM – Corporate Performance Management rendszerekhez;
- irányítópultok (Dashboardok) kialakításához;
- és nem utolsósorban tervezéshez és előrejelzéshez (Budgeting and Forecasting) vagy konszolidációhoz.

Mindezekből talán látszik, hogy sokkal több rendszer használja az OLAP-technológiát, vagy – hogy még egy szinonimát említsek – a többdimenziós adatbázis-kezelőket, mint az elsőre látszik. 10-20 éve még az OLAP-tól voltak hangosak a marketinganyagok, ma már újabb, divatosabb szavakat használnak a rendszerek leírására, de ettől függetlenül, mindegyik rendszer alatt továbbra is egy többdimenziós adatbázis-kezelő látja el a kulcsfeladatokat.

## Többdimenziós adatbázis-kezelő

Az OLAP – ahogy az Analysis Services is – egy többdimenziós adatbázis-kezelő, és mint ilyen, egy sor specialitással rendelkezik a relációs

adattár-kezelőkhöz képest. Az egyik legfontosabb specialitása, hogy nagyon gyors lekérdezési sebességeket produkál. A válaszüz – akár milyen bonyolult lekérdezést állítunk össze – másodpercekben mérhető. Ehhez azonban az kellett, hogy szakítsanak a relációs alapokkal, és kidolgozzanak egy teljesen új tárolási módot. Ezt a tárolási és adatkezelési módot látták el aztán az OLAP betűszóval.

Mint minden jóban, az OLAP-ban is van valami rossz. A gyors adatelérésért és a rugalmasságért cserébe nem tudjuk gyorsan beszűrni az OLAP-adattárba az adatokat.

Mint a későbbiekben majd látni fogjuk, az OLAP-adattárakban kockákban tároljuk az adatokat, és egy adatkockába adatokat beszűrni meglehetősen nehéz. Az MDX-nek – az Analysis Services és még egy néhány OLAP-adattár-kezelő lekérdezőnyelvének – például nincs INSERT utasítása. Csak SELECT és UPDATE. Épp ezért az OLAP-kockákat ritkán, naponta jellemzően egyszer töltjük, és ekkor vagy zéróról újraépítjük az egészet, vagy az új adatokat hozzáillesztjük a már meglévő kockához.

## Adatkockák és dimenziók

Az OLAP-adattárakban az adatokat úgynevezett adatkockákban és dimenziókban tároljuk. (Hasonlóan ahhoz, ahogy a relációs adattárban az adatokat táblákban tároljuk.) A kockák 3 dimenziós testek, egy Analysis Services-adatkockának azonban 128 dimenziója lehet. Igaz, a való életben jellemzően 5-10 dimenziós kockákat építünk, mert az üzleti területek adatkörei rendszerint ennyi dimenzióval leírhatóak.

Az adatkockákban témaorientáltan tároljuk a mutatószámokat, és jellemzően egy-egy üzleti vagy elemzési terület igényei köré szervezzük őket. Mondok egy példát. A pénzügyi területnek biztos szüksége lesz egy olyan adatkockára, amelyben a főkönyvi mozgásokat követheti nyomon. (Ebben az esetben az elemzési terület vagy témakör a főkönyv, az adatkocka a főkönyvi tranzakciókat fogja tartalmazni.)

A főkönyv nevű adatkockánk dimenzióvalva lesz a főkönyvi számlaszámokkal, idővel, költségköltséggel, költségközponttal és költségközponttal. (A dimenziók segítségével címkézzük fel az adatokat: mikori állapotot tükrözne, melyik költségköltséggel érintik stb.) Ezeket a dimenziókat az üzleti élet határozta meg,

hiszen amikor egy könyvelési tételt rögzítünk a vállalatirányítási rendszerben, akkor ezeket a „könyvelési dimenziókat” kell megadnunk. Nekünk csak az a feladatunk, hogy ezeket a létező dimenziókat feltérképezzük, és lemondjuk az OLAP-adattárban.

Ha az értékesítést szeretnénk elemezni, akkor létre kell hoznunk egy értékesítés kockát, és azt, hogy milyen dimenziói lesznek, szintén nekünk kell meghatároznunk. Pontosabban az üzlet már régen meghatározta, nekünk csak ki kell derítenünk, hogy melyek ezek. Az idő, a termék és a vevő biztos, a többi ki kell nyomozni.

Egy OLAP-kocka tervezésekor itt kell a leginkább ésszerű lenni. Ha jól határozzuk meg a kocka dimenzióit, akkor a felhasználóink szeretni fogják az OLAP-ot, hiszen az a valós üzleti problémát modellel, ráadásul mindezt az ő gondolkodásuknak megfelelően. Ha nem, akkor készítettünk szuper gyors, szuperjót rendszert a magunk dicsőségére, mert a felhasználók nem fognak benne otthonosan mozogni.

## Csillagsémás gyökerek

Az OLAP-kockák szerkezete nagyon hasonló képet mutat a relációs csillagsémás adatmodellekhez. A csillagsémában középen helyezkednek el a mérőszámok, körülöttük pedig csillag alakban a mérőszámokat leíró dimenziótáblák. (Bővebben lásd az Adattárház-építés lépésről lépésre című cikkben.)

Az OLAP-adattár dimenziói a csillagsémák dimenziótábláinak, az adatkocka pedig a ténytáblának feleltethető meg. Ez a megfeleltetés egy jól felépített csillagséma esetén 1:1-es kapcsolatot jelent, azaz pont ugyanazokat az adatokat találjuk meg a dimenziókban, mint a dimenziótáblákban, és ugyanazokat találjuk meg a kockában, mint a ténytáblában. Épp ezért nagyon sokszor adattárházak ülnek az OLAP (Analysis Services) adattárak alatt. (Lásd Nigel Pendse kutatását, amely szerint az OLAP technológia második leggyakoribb alkalmazási területe az általános adattárházi jelentéskészítés.)

Sokszor azonban az OLAP-adattárak alá is csillagsémás relációs adatmodellt tervezünk. Ennek legfőbb oka az, hogy sokkal könnyebb felépíteni relációs oldalon egy csillagsémát és arra ráültetni egy OLAP-kockát, mint a forrásrendszerekből közvetlenül tölteni azokat. Sokkal nagyobb irodalma van,

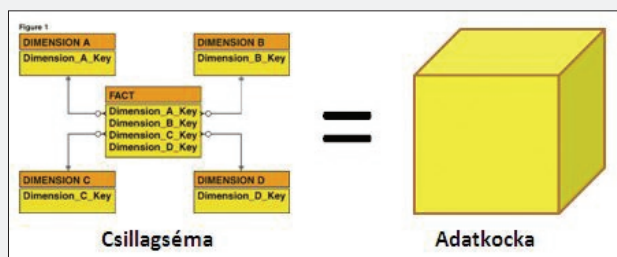
sokkal több eszköz létezik hozzá, és sokkal több szakértő ért a relációs adatbázisokhoz, mint az OLAP-hoz.

Az Analysis Servicest egyébként pont az ilyen csillagsémás adatmodellekre optimalizálták, így két legyet ütünk egy csapásra. Egyrészt a feladatok egy részét áttettük a relációs oldalra, ahol sokkal könnyebben oldhatunk meg bizonyos problémákat, másrészt az elkészített OLAP-rendszer sokkal hatékonyabban fog működni, mint ha csillagséma helyett egy normalizált modellből töltenénk.

Ja, és az Analysis Serviceshez ingyen kapjuk a relációs adatbázis-kezelőt és az adatbetöltést támogató Integration Servicest is. Ilyen feltételek mellett pedig tényleg vétek lenne paragon hagyni ezeket a szoftvereket, pláne úgy, hogy nagyon sok beépített szolgáltatást tartalmaznak a csillagsémás adatmodellek menedzseléséhez és feltöltéséhez.

## Miért nem elég az adattárház?

Említettük, hogy egy adattárházban felépített csillagséma egy az egyben megfeleltethető az OLAP-kockáknak. Akkor miért építünk ugyanarra az adatkörre egy relációs és egy többdimenziós adatbázist is?



2. ábra. Csillag és kocka

A kérdés teljesen jogos. Miért kell megduplázunk az adatokat és letárolni két fajta adatbázis-kezelőben is, amikor a relációs csillagsémát is simán le tudják kérdezni a felhasználók? Hiszen pont ezért építjük az adattárházakat.

Egy OLAP-adatbázis-kezelő azonban rendelkezik néhány olyan előnnyel, amellyel relációs társa nem.

- Lényegesen jobb a lekérdezési sebessége, mint relációs társáé. Nem csoda, hiszen erre optimalizálták (és nehéz úgy megfektetni egy lekérdezéssel, mint a relációt).
- Sokkal kifinomultabb és hatékonyabb jogosultságkezeléssel rendelkezik, mint a relációs adatbázisok. (Az OLAP-adatbázis-

ban akár cellaszinten kezelhetünk jogosultságokat.)

- Lényegesen fejlettebb elemzést támogató funkcióval rendelkezik, mint a relációs társa. Például az előző év azonos időszakának kimutatására az OLAP már több mint 10 éve tartalmaz beépített függvényeket, a relációs adatbázis-kezelők továbbfejlesztési terveiben pedig csak most kezd feltűnni.
- És ami a legfontosabb: klasszisokkal jobb riportkészítő és lekérdezőeszközök léteznek hozzá, mint a relációshoz, és ez az, amivel lehetőséget teremtünk a felhasználóknak, hogy saját maguk – az IT támogatása nélkül – legyenek képesek riportokat készíteni.

Összefoglalva elmondhatjuk, hogy OLAP-rendszereket azért vezetünk be, hogy segítsük az üzleti felhasználók munkáját, hogy ne az informatikát terheljük érthetetlen kérésekkel, hanem saját maguk abban a pillanatban elő tudják állítani a kívánt riportot, amikor eszükbe jut. Ezért duplikáljuk meg a vállalati adatokat, és tároljuk le őket még egyszer egy másik adatbázis-kezelőben is, és ezért fordítunk heteket, hónapokat arra, hogy kiépítsünk egy újabb rendszert, ami végül is alapadatszinten nem tartalmaz több adatot, mint a vállalati rendszerek. (Igaz, nagyon sok előre kiszámított adatot tárol, ami viszont már nem található meg a forrásrendszerekben.)

Most, hogy már kivesszük mi az az OLAP, illetve miért vezetnek be

a vállalatok OLAP alapú rendszereket, vizsgáljuk meg, hogy milyen speciális jellemzői vannak az Analysis Servicesnek, és milyen új-donságokat hozott a 2008-as verzió.

## A Microsoft OLAP-szervere, az Analysis Services

Az Analysis Services a Microsoft többdimenziós adatbázis-kezelője, az SQL Server programcsomag része. Az Analysis Services teljesen független az SQL Servertől, akár annak telepítése nélkül is használhatjuk, de külön nem tudjuk megvásárolni.

Bizonyára sokan tudják: az Analysis Services idén novemberben fogja ünnepelni 10. születésnapját. Ez alatt a 10 év alatt nagyon

sokat fejlődött. Az SQL 7.0-ban debütáló MS OLAP Server névre hallgató Analysis Services még meglehetősen szerény képességekkel volt felruházva. A 2000-es OLAP-szerver már egy robusztus OLAP-motornak számított, de a nagy változást a 2005-ös verzió jelentette.

Ekkor ugyanis jelentősen megváltoztatták az OLAP-szerver architektúráját, és a régi kódok 90 százalékát újraírták. Ekkor mutatkozott be az úgynevezett Unified Dimensional Model – vagy más néven UDM –, ami teljesen felforgatta az Analysis Services korábbi dimenzionális adatmodelljét.

## Unified Dimensional Model (UDM)

Beszéljünk egy kicsit arról, hogy mi is az az UDM, mert marketinganyagokban sokszor fog vele találkozni, és talán már meg is tanulta, hogy szuperflexibilis, meg nagyteljesítményű, de valószínűleg azt még nem tudja, mi áll a háttérben.

Ez nem az ön hibája, hiszen nem nagyon jelent meg irodalom erről a témáról. (Én a webkettővel voltam így: már a csapból is webkettő folyt, de nekem még mindig fogalmam sem volt róla, hogy mi is az.)

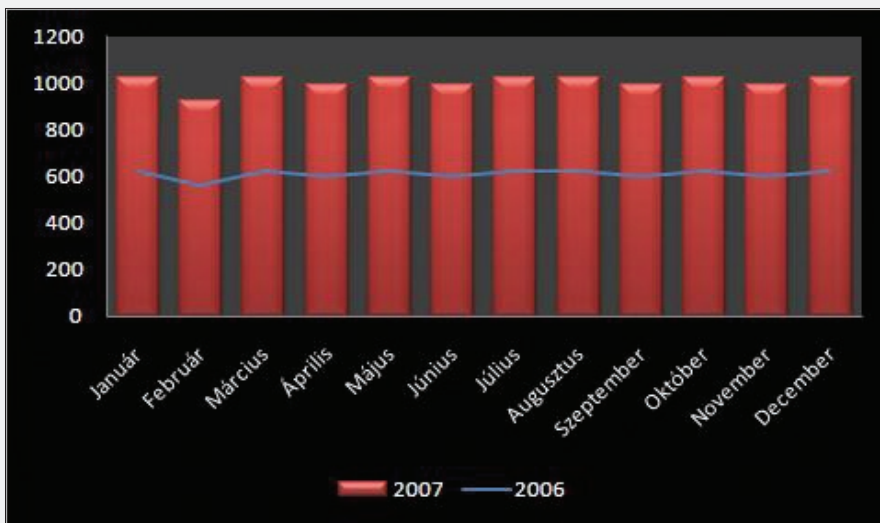
Az UDM az a modell vagy szerkezet, ahogy a 2005-ös Analysis Servicesben a kockák és a dimenziók felépülnek, ideértve, hogy egy adatkocka immáron több különböző forrásból táplálkozhat, a kulcs-teljesítménymutatókat (KPI), a többnyelvűség támogatását és még sok minden más új-donságot, amit a 2005-ös Analysis Services-kockák szerkezetének változása hozott. Az UDM tehát az a szerkezet, ahogy az Analysis Services 2005 a kockákat tárolja. Inkább marketingfogalom, semmint műszaki meghatározás. Az UDM-nek, vagyis a 2005-ös adatkockák szerkezeti változásának legfontosabbika, az attribútum alapú adatmodell megjelenése volt.

## Attribútum alapú dimenzionális modell

A 2005-ös Analysis Services szakított az addigi dimenzió és hierarchia alapú felépítéssel, és áttért az úgynevezett attribútum alapú megközelítésre. Az attribútum alapú dimenzionális modellben az attribútumoké a főszerep. Mindent, amit a korábbi Analysis Servicesekben a dimenziókkal tudtunk csinálni, most már attribútumokkal tudjuk megtenni.

Régebben dimenziókat tudtunk a táblázat soraira, illetve oszlopokra húzni, ma már mindez az attribútumokkal lehetséges. Olyat például nem tudtunk megtenni, hogy egy dimenzió (egy hierarchia) egyik attribútumát kitettük a táblázat sorára, a másikat pedig a táblázat oszlopára.

Joggal kérdezhetné, hogy minek a táblázat sorára kitenni például a vevő nevét, ami a vevődimenzió egy attribútuma, az oszlopára pedig a címét (ami egy másik attribútuma a vevődimenzióknak), hiszen kapnánk egy olyan táblázatot, aminek csak az átlójában vannak adatok. Az ég egy adta világon semmi. A táblázat oszlopára kitenni a hónapokat, soraira pedig az éveket, annak már van értelme. (Pedig mindkét attribútum az idődimenzió attribútuma.) Íme:



3. ábra. Szezonálisvizsgálat: soron a hónapok, oszlopon az évek

Néhány esetben tényleg zavaró lehet egy kicsit, hogy bármely attribútumot kirakhatjuk a táblázat sorára vagy oszlopára, de néha (például az előbb említett szezonálisvizsgálatoknál) nagyon jól jön.

Ahogy néha az is nagyon jól jön, hogy egy adatkockába betölthetjük az egész adattárházat, annak minden adatát: értékesítést, termelést, pénzügyet, mindent. Így például az idődimenzióra felfűzve a vállalat minden adatát, egy kockában láthatjuk a vállalat adott időszak alatt bekövetkezett összes eseményét. Az más kérdés, hogy így maga a kocka bazi nagy lesz, és nem mindenki számára lesz átlátható, de a szabadságnak ez az ára. A döntés pedig rajtunk áll: vagy sok kis kockát építünk, vagy egy nagyot, amit aztán az úgynevezett perspektívákkal (Perspective) szét-

darabolunk kis kockákra. (Hasonlóan ahhoz, ahogy egy táblára nézeteket definiálunk a relációs adatbázisokban.)

Míg az UDM, vagy az attribútum alapú dimenzionális modell a 2005-ös Analysis Services újdonságai voltak, nézzük meg, mi mindent tartogat számunkra a 2008-as verzió.

### Az Analysis Services 2008

A 2008-as Analysis Services messze nem hozott annyi újdonságot, mint 2005-ös elődje. A redmondiai rámentek a hatékonyság fejlesztésére, és ennek szentelték minden energiájukat. Összesen egy darab olyan újdonságot tartalmaz a szoftver, ami vadiújnak tekinthető: ez pedig a dinamikus nevesített halmaz. Az összes többi fejlesztés a teljesít-

2005 alapú rendszert, és arra jöttek rá, hogy a fejlesztők messze nem használják ki még az Analysis Services lehetőségeit. Hiába tombolnak a lőerők az Analysis Services alatt, ha annak fejlesztői behúzva felejtik a kézféket.

Az Analysis Serviceshez kevés a jó dokumentáció (mármint nem rendszer, hanem rendszerépítési), a felhasználói interfészek sem intuitívak, és nem figyelmeztetik a fejlesztőt, ha behúzva felejtik a féket. Mondok egy példát: a 2005-ös Analysis Servicesben minden gond nélkül létre tudtunk hozni redundáns attribútumrelációkat (ez rontja a teljesítményt). Az SP2 telepítése után már kaptunk egy halványsárga figyelmeztetést: ez így nem biztos, hogy jó. 2008-tól kezdve pedig átalakították az attribútumreláció-tervezőt, hogy ne nagyon lehessen redundáns relációt létrehozni, de ha mégis sikerül, akkor a BI Development Studio pirossal figyelmeztet minket, hogy ejnye-bejnye.

A 4. ábra az Analysis Services új attribútumreláció-tervezőjét mutatja. Érdeemes megnézni a 2005-ös verzió attribútumreláció-tervezőjét (lásd az 5. ábrát). Mindkét eszköz ugyanazt a feladatot látja el, ám míg a felsőből látszik, hogy a hónap attribútum a negyedév és a nap attribútummal áll kapcsolatban, addig a 2005-ös attribútumreláció-tervezőből ez már kevésbé állapítható meg. Pedig ugyanazt a dimenziót mutatja mindkét eszköz.

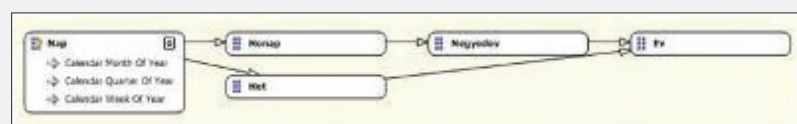
Ezek az apró javítások nagyban hozzájárulnak ahhoz, hogy a BI-fejlesztők könnyebben tudjanak hatékony BI-rendszereket készíteni. A fenti attribútumkapcsolatokból például egy szakavatott szem egyből észreveszi, hogy azok gyémánt alakot rajzolnak ki (ami szintén teljesítményromboló hatású), de az 5. ábrából ez sohasem fog kiderülni.

### Tervezési figyelmeztetések

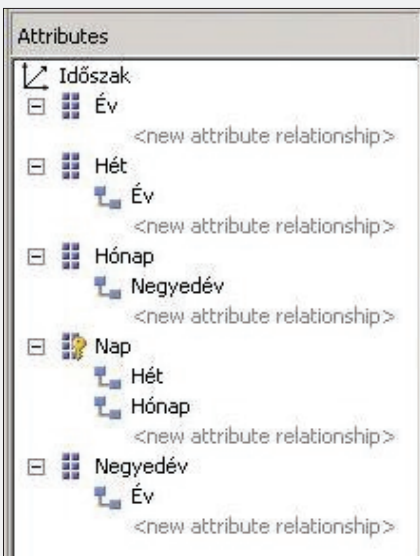
Az Analysis Services 2008, hogy hatékonyabban támogassa az OLAP-adatbázisok tervezését, több mint 40, az előzőhöz hasonló tervezési irányelv teljesülését ellenőrzi, és ha valamelyik sérül, akkor szolidan figyelmeztet. Mindezt nagyon kulturáltan teszi. Nem ugrálnak fel ablakok ötsoros hibaüzenettel és

mény növelését és a felhasználóbarátságot célozta meg.

Amikor kint jártam az Egyesült Államokban az MVP-meetingen, akkor ott valaki megkérdezte: miért csak ennyi újdonságot pakoltak az új Analysis Servicesbe. A fejlesztők erre egyértelműen elmondták, hogy „időt kell hagyni, amíg a BI-fejlesztők megtanulják használni az Analysis Servicest”. Ez így önmagában lehetne cinikus válasz is, de tovább folytatták, és elmagarázták, hogy megvizsgáltak x darab működő Analysis Services



4. ábra. Az Analysis Services új attribútumreláció-tervezője



5. ábra. A 2005-ös verzió attribútumreláció-tervezője

OK gombbal, csak diszkrétan jelzi, illetve egy listán gyűjti a javaslatait.

Mindezen figyelmeztetéseket a BI Development Studióba, az Analysis Services rendszer fejlesztőeszközeibe építették be, így már a fejlesztési fázisban tudomást szerezhethetünk az esetleges tervezési hibákról. (A Microsoftnak van már egy hasonló feladatokat ellátó Best Practice Analyzer nevű alkalmazása, de az teljesen külön futtatható a BI Development Studiótól.)

A BI Development Studio az adatbázisokat, adatkockákat, dimenziókat, adatforrásokat, aggregációkat és particiókat ellenőrzi, és megvizsgálja, hogy azok felépítése meg egyezik-e a legjobban bevált gyakorlattal. Ha például túl sok aggregációt tervezünk, akkor szól, hogy ettől nem lesznek gyorsabbak a lekérdezések, viszont a felösszegzési idők jelentősen meg fognak nőni.

### Erőforrás-monitorozás

Az Analysis Services fejlesztői egy egész csoport (több mint 50) úgynevezett adatmenedzsment-nézetet (Data Management View) tettek elérhetővé a 2008-as verzióban. Ezek lekérdezésével megtudhatunk mindent szerverünk aktuálisan futó folyamatairól. Olyan kérdésekre kaphatunk választ belőlük, mint például mennyi ideig futott az utolsó lekérdezés, ki futtatta azt, mennyi CPU-t használt hozzá, mi volt a lekérdezés szövege, milyen objektumokat érintett a lekérdezés, és még sorolhatnám.

Azért hívják nézetnek, mert hasonlóan viselkedik a relációs nézetekhez (view-khoz). Le

tudjuk kérdezni őket egy select utasítással, de a háttérben a lekérdezésünk átirányítódik más adatbázis-objektumokra. Azt, hogy mikor lettek utoljára felösszegezve az adatkockák, például a következő DMX-lekérdezéssel tudjuk visszakapni:

```
Select
    CUBE_NAME,
    LAST_DATA_UPDATE
from $system.mdschema_Cubes
where CUBE_SOURCE = 1
```

Az aktuális folyamatokon kívül egyes adatmenedzsment-nézetek információt adnak azoknak a tervezési kérdéseknek a megválaszolására is, hogy vajon elég aggregációt tervezünk-e egy adott kockára, vagy létezik-e olyan kocka, amelyre több aggregációt tervezve javítani tudnánk a lekérdezések sebességén?

És ez csak a jéghegy csúcsa. Az 50 adatmenedzsment-nézet ennél sokkal több információt tartalmaz. És hol voltak ezek az információk eddig? Egy részük, az adatbázis-leírásokkal kapcsolatosak eddig is elérhetőek voltak, viszont az Analysis Services eszközkészletén kívülről. Például VB-ben lehetett programot írni az OLAP adatbázissémájának felderítésére, ami nagy segítség volt az alkalmazás-fejlesztőknek, de egy mezei BI-szakértőnek mintha ott se lett volna. Sosem használta.

Az adatmenedzsment-nézetekkel megint kaptunk egy kis segítséget, amivel könnyebben tudjuk monitorozni OLAP-rendszerünk működését, és aminek eredményeképpen aztán hatékonyabb megoldásokat tudunk majd készíteni. Térjünk át a teljesítménynövelés másik forrására, az adatbázismotor továbbfejlesztésére.

### Chiptuning

Az új verzió nemcsak ráncfelvarráson ment keresztül, hanem komolyan hozzányúltak a 128 dimenziós erőforráshoz is: gyorsítottak a lekérdezéseken, lerövidítették a mentéshez szükséges időt, lehetővé tették az erőforrás monitorozását, és megváltoztatták a visszaírás architektúráját is. (Hogy csak a legfontosabbakat említsem.)

### A lekérdezések felgyorsítása (blokkonkénti kiértékelés)

A lekérdezések gyorsítására a legegyszerűbb optimalizációs technika most minden bizonynyal az Analysis Services 2005-ös verziójá-

nak frissítése 2008-ra. Ezzel nem azt akarom mondani, hogy el kell felejtetni a hagyományos lekérdezőoptimalizációt, vagy azt, hogy az Analysis Services 2008 telepítése után minden lekérdezésünk szupergyors lesz, hanem azt, hogy a meglévő lekérdezéseink jelentős része fel fog gyorsulni.

A lekérdezések felgyorsulását az eredményezi, hogy megváltoztatták az Analysis Services lekérdező optimalizálóját, amelynek eredményeképpen az optimalizáló a cellánkénti kiértékelés helyett már képes a NULL tartalmú cellák blokkonkénti kiértékelésére is. Ez az, aminek következtében az Analysis Services 2008-nak sokkal kevesebb műveletet kell végeznie, mint 2005-ös társának, és végül is ennek köszönhetően gyorsulnak fel a lekérdezések. Hogy mennyivel, az sok mindentől függ. Például attól, hogy mennyire lyukacsos a lekérdeztet adathalmaz, vagy attól, hogy olyan MDX-utasításokat használunk-e, amelyek optimalizálva lettek a blokkonkénti kiértékelésre. (Az MDX az Analysis Services lekérdezőnyelvére, hasonlóan ahhoz, ahogy az SQL nyelv az SQL Serveré.)

Én még CTP5-ös verzióon teszteltem a sebességnövekedést, és ugyanarra a lekérdezésre 6-szor gyorsabban kaptam választ az Analysis Services 2008-cal, mint a 2005-ös verzióval. (Mások hasonló tesztet futtatva 20-szoros sebességnövekedésről számoltak be.) Az igazsághoz azonban az is hozzátartozik, hogy ugyanez a sebességnövekedés kézi tuningolással is elérhető volt a 2005-ös verzióban. A 2008-as Analysis Services lekérdezőoptimalizáló azonban már saját maga felismeri az ilyen helyzeteket, és nem nekünk kell trükközniük, hogy észrevegye.

### Oldalra skálázás

Három olyan újdonságot is tartalmaz az Analysis Services 2008, ami hozzájárul ahhoz, hogy immáron oldalra is skálázhassuk BI-rendszerünket.

- Olvashatóvá (read-only) tehető az Analysis Services adatbázisa. Bármilyen furcsa, az Analysis Services adatbázisai eddig nem voltak csak olvashatóvá (read only) tehetőek. Épp ezért kellett leállítani a szervert fájlmásolásakor – lásd a következő mentési fejezetet –, hiszen módosulhattak a fájlok a másolás alatt, így verzióeltérések miatt könnyen előfordulhatott, hogy nem sikerült visszaállítani az adatbázist.

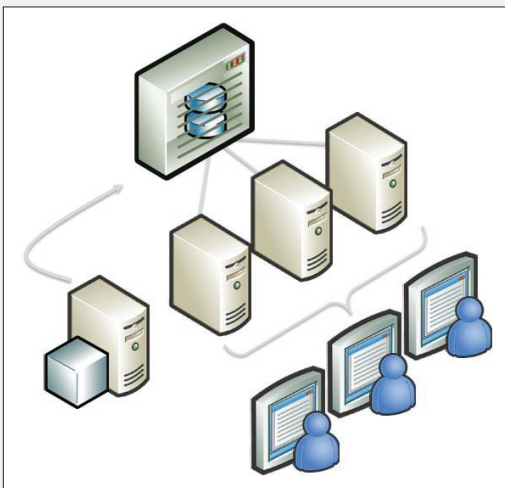
- Az Analysis Services adatfájlnak mindig azon a gépen kellett lennie, ahol a szerver futott. Eddig. A 2008-as verziótól kezdve átrakhatóak másik gépre.
- És le-, illetve visszakapcsolhatjuk az adatbázist (attach/detach).

E három újdonság már lehetővé teszi, hogy kialakítsunk egy olyan architektúrát, amelyben egyszerre több kisebb teljesítményű szerver válaszol a felhasználók lekérdezésére. (Egy nagy és drága gép helyett.)

## Új backup-fájlszerkezet

Az üzemeltetés hatékonyabbá tétele érdekében megváltoztatták az Analysis Service-adatbázisok mentéseként előálló backup-fájl szerkezetét. A régi fájl szerkezettel az volt a probléma, hogy elérve a 20 gigabájtos méretet, elkezdett exponenciálisan nőni a mentés elkészítésének időszükséglete.

Hogy ezt elkerüljék, az üzemeltetők mentés helyett inkább leállították a szervizt, és

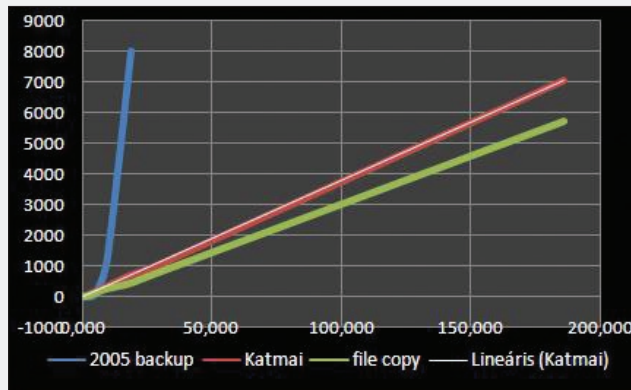


6. ábra. Oldalra skálázás az Analysis Services 2008-cal.

átmásolták a DATA könyvtár tartalmát egy backup területre. Aztán ha végeztek, visszakapcsolták a szervizt, és folytathatták tovább a munkát.

A 7. ábra az Analysis Services 2005, 2008 mentési sebességét hasonlítja össze a fájl másolás segítségével. (Az y tengelyen a másodpercek szerepelnek, az x tengelyen pedig megabájtok.)

Az ábrából leolvasható, hogy a 2008-as Analysis Services mentésének készítése nem sokkal lassabb, mint a fájl másolás sebessége, és hogy a mentés időszükséglete immáron lineárisan függ az adatbázis méretétől.



7. ábra. Mentési sebességek összehasonlítása

## Az üzleti tervezés támogatása

Mint a bevezetőben említettem, az OLAP-rendszereket ad hoc jelentéskészítés mellett előszeretettel használják tervezésre is (4. helyen volt a „mire használják az OLAP-rendszereket” listán), hiszen a többdimenziós felépítés, a hierarchiák menti felösszegzések (például: havi tervből éves terv előállítás) mind-mind szükségesek a tervezés támogatásához.

Az Analysis Services mint OLAP-eszköz természetesen támogatja az egyszerűbb tervezési folyamatokat. Tudja fentről lefelé szétosztani, vagy lentől felfelé konszolidálni a tervszámokat, de ezt már a 2000-es verzió óta tudja.

Ami új a 2008-as Analysis Services tervezéstámogatásában (visszaírás, vagy más néven write back), az az architektúra megváltozása. Az Analysis Services eddig a tervvariánsok közti különbséget vagy első verzió esetén magát a tervet a relációs adatbázis-kezelőben tárolta, a 2008-as verziótól azonban már átke-

rült a többdimenziós adatbázis-kezelőbe, jelentősen felgyorsítva ezzel a tervezett adatok lekérdezésének sebességét.

## Elemzés támogatása: nevesített halmazok

Most az Analysis Services 2008 egyetlen olyan újdonságát fogom bemutatni, ami nem valami meglévő modulon vagy a fejlesztők hatékonyságán hivatott javítani, hanem valami vadonatújat hozott létre. Ez pedig a dinamikus nevesített halmaz vagy más néven Dynamic Named Sets.

Nevesített halmazok (Named Sets) eddig

is léteztek az Analysis Servicesben, és segítségével dimenzióelemekből összeállíthatunk egy halmazt, ami aztán úgy viselkedett, mint egy dimenzió. Ki lehetett rakni őket táblázat soraira, illetve oszlopokra.

E halmazok tartalmát, azaz a halmazba beválogatott elemeket kézzel tudtuk módosítani,

de olyan halmazokat nem tudtunk létrehozni, amelyek elemei dinamikusan változnak az aktuális lekérdezéssel együtt. Ha például a TOP 5 vevőt tartalmazó nevesített halmazt hoztuk létre, akkor az mindig azt az 5 vevőt tartalmazta, amely a létrehozásának pillanatában benne volt az 5 legjobb között. Ha viszont módosítottuk a lekérdezést, mondjuk, megváltoztattuk az évet 2008-ról 2007-re, akkor a nevesített halmaz változatlanul a 2008-as 5 legjobb vevőt mutatta.

Ezzel szemben a dinamikus nevesített halmazok dimenzióelemei dinamikusan változnak a lekérdezéssel együtt, azaz nem létrehozásukkor töltődnek fel, hanem minden egyes olyan alkalommal, amikor megváltozik a rá hivatkozó lekérdezés. Így a 2008-as Analysis Services-zel már képesek leszünk olyan nevesített halmazokat létrehozni, amelyek mindig az aktuális lekérdezésnek megfelelő 5 legjobb vevőt fogják visszaadni:

```
CREATE DYNAMIC SET CURRENTCUBE.[5 legjobb vevő]
AS TopCount
(
    [Vevő].Members,
    3,
    [Measures].[Terv-tény eltérés %]
)
```

Ezzel kivégeztük az Analysis Services 2008 újdonságainak bemutatását, és már csak egy témánk van hátra.

## Az adatbázisban tárolt adatok kiaknázása

Kezdjük talán azzal, hogy az Analysis Servicesben épített BI-rendszerek fölé tetszőlegesen választhatunk front-endet, azaz olyan szoftvereket, amelyekkel az adatbázis lekérdezhető. Az Analysis Services lekérdezhe-

tó az Excelből, Reporting Servicesen és a PerformancePoint alkalmazásokon (ProClarity, Scorecard Manager, Planning) keresztül, de választhatunk harmadik szállító termékei közül is, és fejleszthetünk saját front-end alkalmazást is.

Ezek a front-endek más és más igényt elégitenek ki, és ennek megfelelően a valódi BI-rendszerek sem egyfajta front-endet használnak az OLAP-adatbázis lekérdezésére, hanem legalább egy webes lekérdezőből és egy munkaállomásokra telepített elemző eszközökből álló portfóliót használunk.

A Reporting Services alapvetően a webes lekérdezési igényeket elégíti ki, így érthetően nem tud annyit, mint egy munkaállomásra telepített Excel. A PerformancePoint három speciális célalkalmazást is tartalmaz, amelynek révén az Analysis Services-zel épített BI-rendszerek adatbázisát lekérdezhettük vagy módosíthatjuk. Ezek az üzleti tervezés támogatására kifejlesztett Planning, a vállalati célok megjelenítését és lebontását lehetővé tevő Scorecard Manager és az elemzési munkát professzionálisan támogató ProClarity. Mivel ezekről az eszközökről részletesen olvashat a TechNet Magazinban, mi maradunk a mindenki gépére telepített arany középutánál, az Excelnél.

## Az Excel 2007

„Előbb-utóbb minden adat az Excelben landol” – tartja a népi hiedelem, és mint minden hiedelemnek, ennek is van valóságtartalma. Nincs ugyanis még egy olyan riporting eszköz, amellyel annyira könnyen lehetne testre szabni a riportokat, mint az Excellel. Gyönyörűen nyomtatható, minden egyes cellája akár külön-külön is megformázható, és az adatok módosítására, statisztikai függvények egész hada áll rendelkezésre.

Ha már úgyis Excelben landolnak az adatok, akkor kézenfekvőnek tűnhet, hogy tegyük képessé az Excelt az OLAP-adatok lekérdezésére is. Ez eszükbe jutott más OLAP-adatbázisgyártóknak is, és sokan fejlesztettek front-end alkalmazásokat saját OLAP-adatbáziskezelőjük fölé. A Microsoft is készített néhány add-int még a 2003-as Excelhez, de alapvetően a Pivot-táblát javasolták az Analysis Services alapú adatbázisok lekérdezésére. Míg azonban az Excel 2003 még csak támogatta az OLAP-adatok megjelenítését a Pivot-táblában, addig a 2007-es verzióban helyet ka-

pó új Pivot-tábla tervezésekor már figyelembe vették, hogy legyen alkalmas hatékonyan együttműködni az Analysis Services-zel.

## Pivot-tábla

A Pivot-tábla egy Excelbe épülő alkalmazás, amely az Excel 5 óta része az Excel táblázatkezelőnek. Konceptiója azon alapult, hogy keltsene a felhasználók kezébe adni egy eszközt, amelynek segítségével adathalmazokat hozhatnak létre, majd ezeket az adathalmazokat tetszőlegesen mozgathatják táblázatok soraira, oszlopaira, összegezhetik, rendezhetik tartalmukat. Ez a koncepció nőtt azán ki magát addig, hogy az Excel 97-ben megjelent a varázsló, amellyel már sokkal egyszerűbben lehetett az adathalmazokat definiálni, majd a 2000-es verzióba beépült a kimutatásdiagram (Pivot chart), amely szinkronizálva volt a Pivot-tábla adataival. A fejlődés még ma is tart, és az Excel 2007 Pivot-táblája már sokkal felhasználó- és „elemzésbarátabb”, mint a korábbi verziók voltak.

Teljesen megváltoztatták a Pivot-tábla felépítését, egyszerűsödött a rendezés, bővült a szűrés támogató funkciók köre, új adatvizualizációs eszközök épültek az Excelbe, és a legfontosabb, hogy optimalizálták az OLAP-adatbázisok lekérdezésére.

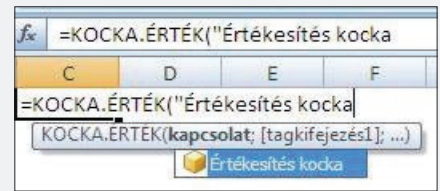
Van azonban a Pivot-táblának egy olyan kötöttsége, amely hosszú évek múltán sem fog megszűnni: segítségével képtelenek leszünk létrehozni úgynevezett szabad formátumú riportokat, amelyek sorain vagy oszlopain nem dimenziók szerepelnek, hanem különböző dimenziók elemeiből összeállított listák. (Egyes hatóságok előszeretettel kérnek ilyen kimutatásokat ☺.)

Képzelsen el egy olyan munkalapot, amely több egymástól teljesen független riportot tartalmaz, de minden egyes riport szinkronizálva van a dátum szűrőfeltétellel. Ha változtatunk a dátumon, akkor minden riport frissül a kiválasztott dátumnak megfelelően. Ezt a Pivot-táblával nem tudjuk elkészíteni, de az úgynevezett kockafüggvényekkel igen.

## Kockafüggvények

A kockafüggvények az Excel 2007-ben mutatkoztak be először, és segítségükkel le tudjuk kérdezni az OLAP-adatkockákat. Nem kell semmi trükkös történetre gondolni, a kockafüggvények ugyanolyan függvények, mint mondjuk a SZUM(), vagy az ÁTLAG() csak

nem összeadják vagy átlagolják a paraméterként megadott számokat, hanem lefuttatják a paraméterben megadott lekérdezést.



8. ábra. Kockafüggvények

A kockafüggvényeket felparaméterezhetjük saját magunk is (megadhatjuk, hogy melyik adatkocka melyik mutatószámát, milyen szűrőfeltételek mentén szeretnénk lekérdezni), de egy már meglévő Pivot-táblát is átalakíthatunk kockafüggvényekké.

Fontos, hogy ez a paraméter (például egy mutatószám neve) lehet egy fix szöveg, például „Forint”, de lehet egy cella értéke is – mondjuk A1 –, és ebben az esetben az A1-es cellába beírt szöveget küldi el az Excel az Analysis Services felé.

A kockafüggvények paramétereit előállíthatjuk más függvények segítségével is. Például az aktuális dátumot átadva a kockafüggvényeknek olyan riportot készíthetünk, amely mindig az aktuális napi adatokat fogja mutatni.

A kockafüggvények segítségével tehát ki-lephetünk a Pivot-tábla zárt 2 dimenziós világából, és képessé tehetjük az üzleti felhasználót akár teljesen amorf alakú riportok létrehozására is. Olyan újdonsága ez az Excel 2007-nek, amely minden bizonnyal az elemzők nagy kedvence lesz.

## Zárszó

Az Analysis Servicesre épített üzletiintelligencia-rendszerek egyik legnagyobb előnye, hogy megteremtik az önálló elemzés és riportkészítés lehetőséget az üzleti felhasználók számára. A bevezetés után jelentősen csökkenni fognak a beszámoló előállításához szükséges idők, és szépen lassan le fognak szokni a felhasználók az IT-tól való információkérésről is. A végén pedig mindenki azt fogja csinálni, amihez a legjobban ért: az informatikus az üzemeltetéshez, illetve a fejlesztéshez, a kontroller pedig az elemzéshez.

Kővári Attila  
(www.biprojekt.hu)

BI-bevezetési tanácsadó, SQL Server MVP

# CSÁKÁNYT A KÉZBE!

## Adatbányászat SQL Server 2008 Analysis Services segítségével

Az információs technológia robbanásával nemcsak az vált teljesen természetessé, hogy adatainkat számítógépen tároljuk, hanem mára már azon sem csodálkozunk senki, hogy az adataink szinte elárasztanak bennünket. Mindegy, hol tároljuk őket, fájlrendszerben, Excel-munkalapokon vagy adatbázistáblákban, egy kritikus méret felett áttekinthetetlené válnak. Persze az évek során számos technológia fejlődött ki az adatok karbantartására és keresésére, gondoljunk csak az operációs rendszerbe épített dokumentum- és e-mail-kereső szolgáltatásra vagy éppen a T-SQL nyelvű lekérdezésekre. Sajnos, ezeknek a módszereknek van egy közös gyenge pontjuk: csak akkor adnak eredményt, ha pontosan megadjuk, mit keresünk.

De mi van akkor, ha nem egy konkrét fájl vagy egy rekordot szeretnénk megtalálni, hanem összefüggésekre vagyunk kíváncsiak? A relációs adatbázis-kezelés eszközeivel hogyan kereshetünk kapcsolatot az egyes rekordok vagy egy tábla attribútumai között? Ebben az esetben jutnak szerephez az *adatbányászati technológiák*, amelyek lehetővé teszik, hogy az adathalmazokból olyan nem triviális, korábban nem ismert vagy potenciálisan hasznosnak vélt *információkat* bányásszunk ki automatizálható módszerekkel, amelyek alapján akár előrejelzéseket is készíthetünk. Ellentétben az OLAP-rendszerekkel, amelyek csak az információ megtalálásában segítenek, az adatbányászat célja *tudásfeltárás*.

Az adatbányászat mögött igen komoly matematikai háttér van, nem véletlen, hogy az adatbázis-technológiákkal kombinált összetett statisztikai és valószínűség-számítási módszerek sokak számára igen elrettentőnek tűnnek. Ma már azonban az adatbányászat nemcsak az akadémikusok játéka, hanem végfelhasználói közelségbe került: algoritmusok részletes ismerete nélkül, Excelben kattintgatva kaphatunk választ olyan kérdésekre, mint hogy mi befolyásolja a vevőink elégedettségét, milyen termékeket vásárolnak együtt, vagy éppen adhatunk-e hitelt egy adott ügyfélnek.

### Adatbányászat és üzleti intelligencia

Bár az adatbányászati technológiákat és módszereket számos területen alkalmazzák a spamszűréstől kezdve az orvostudományon, a játékiparon és a forgalomelemzésen keresztül egészen a terroristaelhárításig, leggyakrabban üzleti adatok elemzéséhez vetik be őket. Íme néhány tipikus prediktív elemzési üzleti probléma, amelyek megoldásához jól használható az adatbányászat:

**Bevásárlókosár-elemzés.** Mely termékeket veszik általában együtt a vevők, mit érdemes ajánlani nekik?

**Elégedetlenségi elemzés.** Mi kell ahhoz, hogy megtartsunk egy ügyfelet, vagy éppen mi készíti a vevőinket, hogy átpártoljanak a konkurenciához?

**Piacelemzés.** Tulajdonságaik, szokásaik és igényeik alapján milyen szegmensekbe sorolhatók az ügyfeleink?

**Előrejelzés.** Várhatóan mennyi fog fogyni egyes termékekből az elkövetkező időszakban, mennyit kell raktárra rendelnünk?

**Kampányelemzés.** Az egyes vásárlói szegmensek számára mik a fontos paraméterek, hogyan hirdessük számukra a termékeinket és szolgáltatásainkat?

**Az adatminőség javítása.** Ennek a mézőnek helyes az értéke, vagy vajon elégepelés történt, és ha ez utóbbi, mit akarhatott írni a felhasználó?

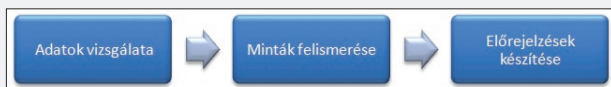
**Csalások felismerése.** Bár formailag helyes, elfogadható-e egy adott mező értéke, vagy gyanúsán kilóg a sorból?

**Szövegelemzés.** Mik a leggyakoribb beérkező kérdések vagy kifogások?

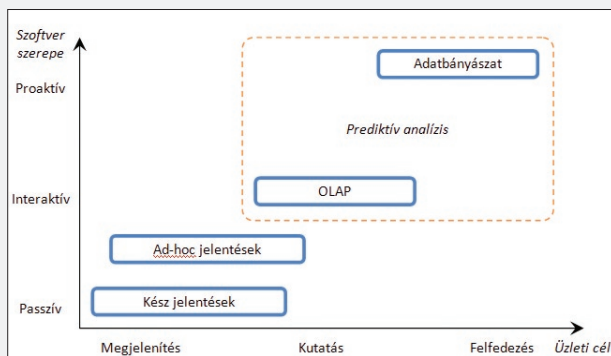
### Adatbányászati feladatok

A fenti felsorolás alapján talán már látható, hogy az adatbányászat számtalan üzleti probléma megoldásához használható. Ezek a problémák jellegük alapján tipikus csoportokba sorolhatók.

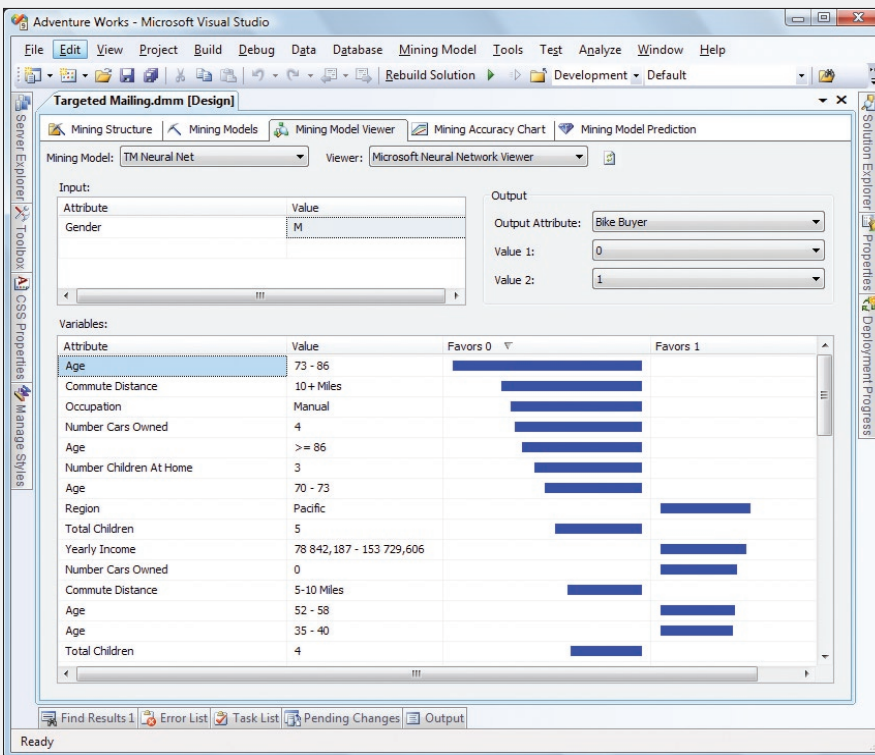
**Osztályozás.** Az osztályozás az új elemek automatikus besorolását jelenti az előzetesen meghatározott kategóriák valamelyikébe, a predikció célja tehát az osztály meghatározása. A megoldáshoz szükségünk van egy modellre, amely az elemek tulajdonságainak függvényében képes meghatározni az *osztály* attribútumot. A modellt ezek után betanítjuk olyan minták alapján, amelyek esetén már ismert az osztály, tehát a modell maga ismeri fel az egyes osztályok elemei közötti hasonlóságot. Tipikus osztályozási algoritmusok a döntési fák, neurális hálók és a Naive Bayes algoritmus, alkalmazási terü-



1. ábra. Az adatbányászat szerepe



2. ábra. Prediktív elemzések kontra hagyományos jelentéskészítés



3. ábra. Mitől függ, hogy a férfiak vesznek-e kerékpárt?

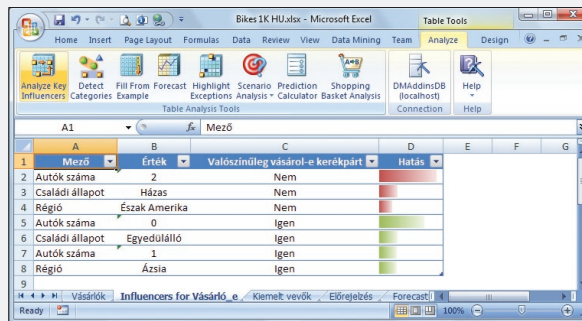
letek pedig a hitelbírálat, kockázatelemzés, kampányelemzés.

**Klaszterezés.** A klaszterezés az osztályozással rokon feladat; célja, hogy egy halmaz elemeit a megadott számú csoportba soroljuk. Itt tehát nincs előre megadott minta, az algoritmus feladata annak meghatározása, hogy mely elemek tartoznak össze – és ezzel az osztályok megtalálása. A piacelemzés például tipikus klaszterezési feladat.

**Asszociációs szabályok keresése.** Az asszociációs szabályok keresése tipikusan „ha X akkor Y is” típusú szabályok felderítése, amelynek leggyakoribb alkalmazási területe vásárlói kosarak elemzése. Ebben az esetben nemcsak a szabályok, hanem a gyakori elemhalmazok és a hozzájuk tartozó valószínűség (például sör és chips együtt 70 százalékos valószínűséggel) felderítése is cél.

**Szekvenciák keresése.** A szekvenciakeresés során nemcsak az egyes esetek tulajdonságai számítanak, hanem kiemelt jelentősége van azok sorrendjének, azaz az idődimen-

ziónak. Fontos tényező az időablak mérete, amellyel meghatározzuk, hogy milyen időintervallumon belül figyeljük az események egymásutánosságát. Tipikus alkalmazási terü-



4. ábra. Az elemzés eredménye: akinek van 2 autója, házas és amerikai, az szinte biztosan nem vesz kerékpárt

letei közé tartozik a riasztások kezelése és kereskedelmi területen a vásárlások sorrendjének elemzése, az utóbbi időben pedig a DNS-lánckok és a webkiszolgálónaplóinak elemzése (click-stream analízis) miatt került előtérbe.

Előrejelzés. az előre-

jelzési feladat egyfajta szekvenciakeresésnek fogható fel, ahol egy változó jövőbeli értékét kell meghatározni korábbi minták alapján. Gyakori alkalmazási terület az eladás- és raktárkészlet-kezelés, valamint a tőzsdei árfolyamok alakulásának előrejelzése.

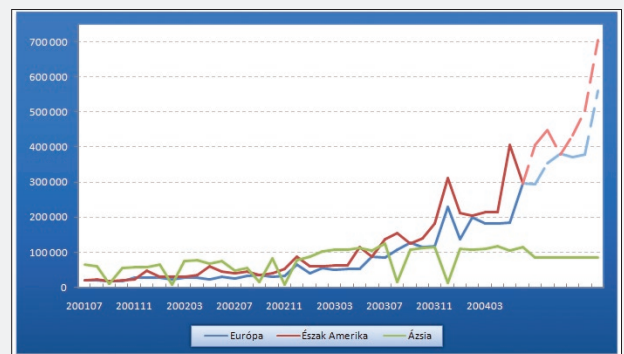
**Eltéréselemzés.** Az eltéréselemzés feladata azoknak a ritka elemeknek a megtalálása, amelyek jelentősen eltérnek a többitől, azaz a korábban megfigyelt mintától. Leggyakoribb alkalmazása a bankkártyacsalások felderítése, de használható hálózati forgalom elemzésére, behatolások és DOS-támadások jelzésére is.

Egy konkrét üzleti probléma megoldása esetén gyakran a fenti alapfeladatok kombinációját, azaz több algoritmust használunk. Például annak meghatározásához, hogy mitől lesz nagyobb hasznuk egy ügyfélből, első lépésben szegmentálhatjuk a partnereinket (klaszterezés), majd kapcsolatot kereshetünk a profit és az ügyfelek tulajdonságai között (osztályozás), felderíthetjük az igényeiket (asszociációs szabály keresése), és megfigyelhetjük a vásárlói szokásaikat (szekvenciák keresése), végül pedig az így kapott információk alapján határozhatjuk meg vállalati stratégiánkat.

## Hogyan lesz az adatból információ?

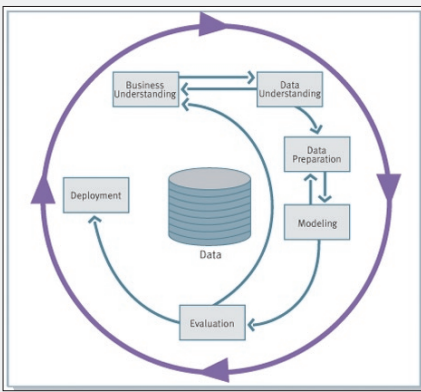
Az adatbányászat folyamatának de facto szabványa 1999-ben született meg CRoss Industry Standard Process for Data Mining, azaz CRISP-DM néven. A szabvány érdekessége, hogy nem elméleti szakemberek alkották meg zárt ajtók mögött, hanem az alkotók valós projektek tapasztalatait osztották meg a világgal, ezért nem csoda, hogy a Microsoft is csatlakozott a kezdeményezéshez, és az SQL Server 2008-ban is ennek a metodológiának az elemeivel találkozhatunk.

A CRISP-DM 1.0 referenciamodell hierar-



5. ábra. Hat hónapos előrejelzés Excelben





6. ábra. Fázisok a CRISP-DM 1.0 modellben

chikus, négy absztrakciós szinten definiálja az adatbányászat lépéseit. Magas szinten a folyamat hat fázisból áll.

Az első lépés a projekt céljainak felmérése üzleti szemmel, majd átültetésük az adatbányászat világába azzal, hogy mérnöki szemmel definiáljuk a problémát. Miután adott a feladat, fel kell derítenünk a rendelkezésünkre álló adatokat, hogy képünk legyen arról, mely adathalmazok segíthetnek a rejtett információk kinyerésében. A rendelkezésre álló adathalmazból általában nincs szükségünk mindenre, ezért következik egy adat-előkészítő lépés, amelynek során transzformációk sorozatával kiválasztjuk a szükséges attribútumokat és rekordokat, majd megtisztítjuk az adatokat, mielőtt felhasználjuk őket. A következő fázis egy adatbányászati algoritmus kiválasztása és a hozzá kapcsolódó modell megépítése, majd a modell számos paraméterének finomhangolása. Könnyen előfordulhat, hogy több modellel is kísérletezünk a feladat megoldása érdekében, és bizony megeshet, hogy egy másik modell más bemeneti adatokat követel meg, ezért vissza kell lépnünk, és újra faragnunk kell kicsit az adatokon. Ha már van egy modellünk, érdemes alaposan tesztelnünk, hogy lássuk, mennyire pontos, megbízható és hasznos az üzleti problémánk szemszögéből. Végül miután megbizonyosodtunk a modell életképességéről, hadrendbe állíthatjuk, azaz beépíthetjük saját alkalmazásainkba és vállalati folyamatainkba.

Ezzel persze még nem vagyunk készen, figyelembe kell vennünk ugyanis, hogy modellünk valószínűleg nem időtálló, ezért a változó üzleti igények követéséről nekünk kell gondoskodnunk. Ez a gyakorlatban lehet, hogy csak a modell paramétereinek pontosítását jelenti, de az is előfordulhat, hogy a megváltozott igényeket csak egy másik modell megalkotásával tudjuk kielégíteni – azaz idővel kezdődik előlről a folyamat.

A fenti hat fázison belül a referenciamodell meghatározza az egyes lépésekben elvégzendő feladatokat is. Mivel a meghatározások elég általánosak ahhoz, hogy minden adatbányászati projektben használhatóak legyenek, a CRISP-DM gyártó- és eszközfüggetlen módon válhatott iparági szabvánnyá.

## Irány a bányára!

A CRISP-DM által elméletben megfogalmazott feladatokat a gyakorlatban a *Business Intelligence Development Studio*, azaz a BIDS segítségével végezhetjük el. A BIDS a Visual Studio 2008 testre szabott változata, amelyet az SQL Server 2008-cal együtt telepíthetünk akár a kiszolgálóra, akár a munkállomásra, és amely fel van készítve Analysis, Integration és Reporting Services projektek fejlesztésére.

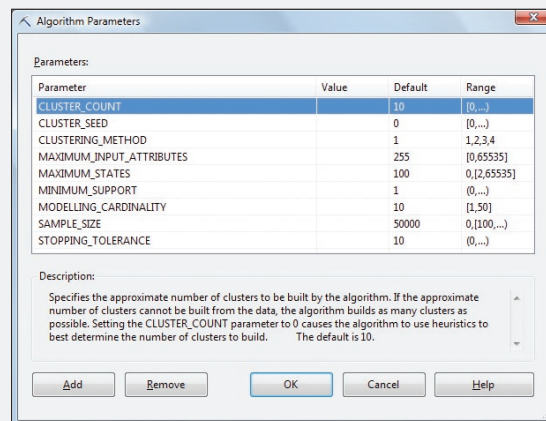
Egy *Analysis Services Projectben* az első lépés egy *Data Source* objektum létrehozása. A data source segítségével határozzuk meg azt az adatforrást, amellyel dolgozni fogunk, ehhez a varázslóban az adatok helyét és a kapcsolódáshoz használt felhasználói fiókot kell megadnunk. Az adatforrás bármilyen OLE DB-n vagy .NET-es adatszolgáltatón (*provid-*

amelyeknek általában csak egy részhalmazára van szükségünk adatbányászati feladatunk megoldásához. Azonosítanunk kell tehát az a táblát, amely az elemzendő eseteket tartalmazza (*case table*), azon belül azokat a sorokat és oszlopokat, amelyek valóban relevánsak, továbbá azokat a kapcsolódó táblákat, amelyek még szükségesek a probléma megoldásához. A teljes adatforrás egy részének ilyen formán történő kivágásához *Data Source View* (DSV) objektumot kell létrehozunk. A DSV lehetőséget ad számított oszlopok definiálására is, így akár bővíthetjük is az adatszerkezetünket.

A bemeneti adatok meghatározása után a következő lépés a két legfontosabb objektum, a *Mining Structure* és azon belül a *Mining Model* létrehozása. A *Mining Structure* objektumban írjuk le, hogy a bemeneti adatokat hogyan szeretnénk használni: itt adjuk meg például az egyes oszlopok adattípusát (szám, szöveg, dátum stb.), a tárolt adatok típusát (folytonos, diszkrét stb.), eloszlását (normál, logaritmus stb.) és célját (bemenet, jóslandó stb.). Mindezek a paraméterek jelentősen befolyásolják a választott algoritmus működését, amelyet a struktúrához rendelt modellben határozunk meg és paraméterezünk fel.

A Microsoft kilenc algoritmust szállít az SQL Server 2008-cal, amelyek közül a megfelelő kiválasztása és optimális paraméterezése tapasztalatot, gyakorlást vagy legalább szerencsét igényel, hiszen egy adott feladatra kevésbé alkalmas algoritmus jelentősen eltérő eredményt produkálhat. Szerencsére a *Books Online Algorithm Reference* fejezete sok segítséget ad az induláshoz, a profibbakk pedig akár saját algoritmus implementálásával is bővíthetik a rendszert.

Ha megvan a modell, be kell tanítanunk azt, amit a modell feldolgozásának (*Process*) is neveznek. Az SQL Server 2008 újdonsága, hogy nem kell külön



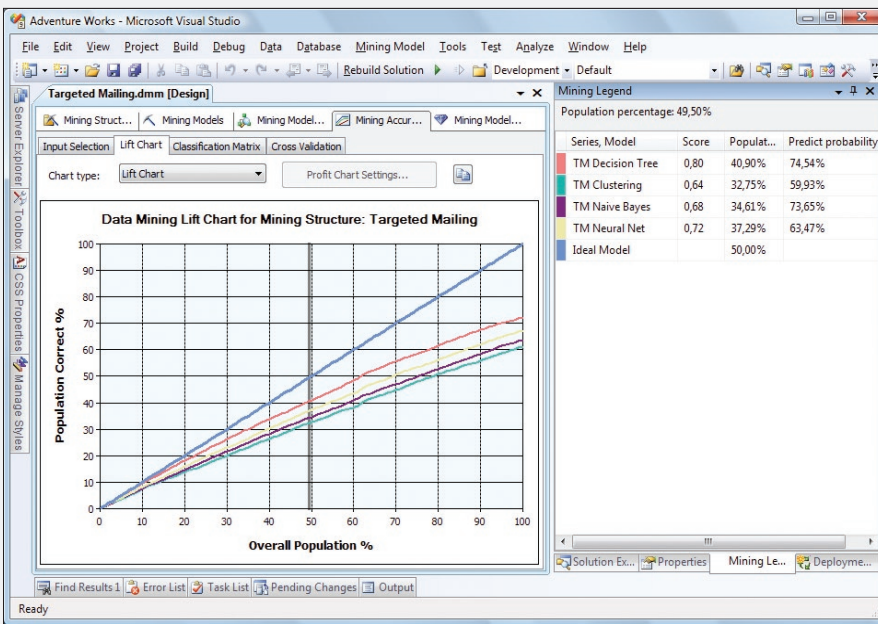
7. ábra. A modellek finomhangolása gyakorlott bányászoknak

tanító- és tesztelőadatokat biztosítanunk, elég megadnunk, hogy a bemeneti adatok hány százalékát használja a rendszer tesztelésre – ez az úgynevezett *holdout*, és tipikusan 30 százalék.

A betanított modell már használható, futtathatunk lekérdezéseket rajta. Előtte azonban célszerű megvizsgálni, hogy a modellünk

tanító- és tesztelőadatokat biztosítanunk, elég megadnunk, hogy a bemeneti adatok hány százalékát használja a rendszer tesztelésre – ez az úgynevezett *holdout*, és tipikusan 30 százalék.

A betanított modell már használható, futtathatunk lekérdezéseket rajta. Előtte azonban célszerű megvizsgálni, hogy a modellünk



8. ábra. Modellek pontosságának összehasonlítása Lift Chart segítségével BIDS-ben

mennyire sikerült jól. Itt három szempontot szokás figyelembe venni.

**Pontosság.** Valóban helyes értékeket jósol a modell?

**Megbízhatóság.** Jól működik-e a modell más bemeneti adatokra is?

**Hasznosság.** Feltárulnak-e újabb összefüggések a modell használatával, vagy csak nyilvánvaló kapcsolatokat ismert fel a rendszer?

A modell pontosságának meghatározását a BIDS vizuális eszközökkel támogatja: *Lift Chart*, *Profit Chart*, *Scatter Plots* típusú diagramok és *Classification Matrix* áll rendelkezésünkre.

További segítség az SQL Server 2008-ban bevezetett *Cross Validation* funkció, amely a tanító- és tesztelő-adathalmazok particionálásával és forgatásával segíti a modell megbízhatóságának meghatározását.

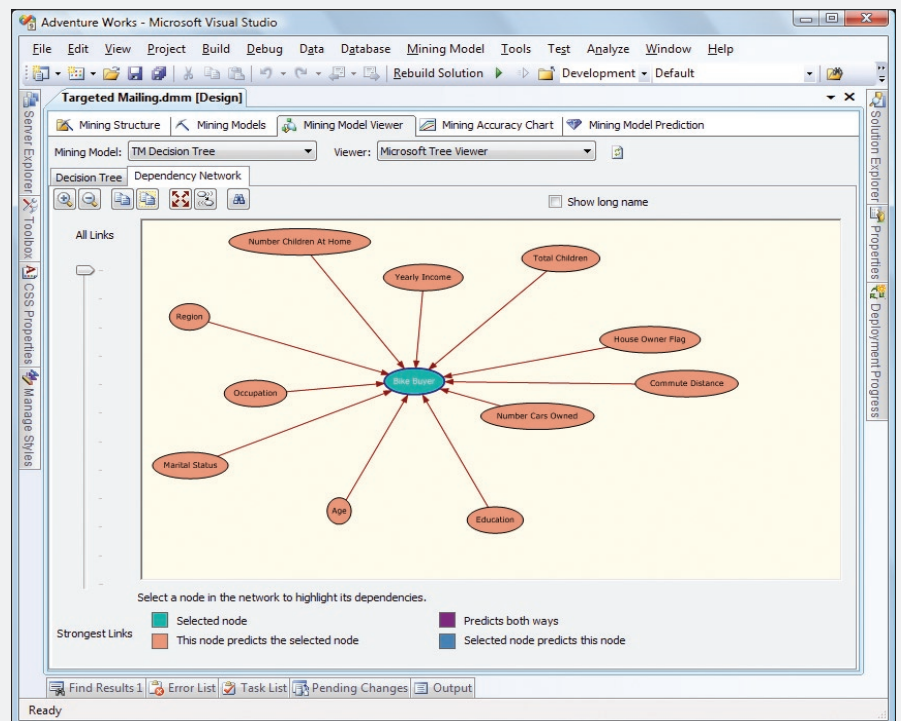
Miután megbizonyosodtunk modellünk helyességéről, bevethetjük éles használatra. Ennek legegyszerűbb változata, ha közvetlenül a BIDS eszközeivel végzünk lekérdezéseket a modellen. Hasonló grafikus eszközöket találunk a Management Studióban is, de akár Reporting Services-ből, Excelből és Visióból is kapcsolódhatunk a modellhez.

## Halandóknak

Az előző fejezetben bemutatott folyamat, az objektumok létrehozása és paraméterezése nemcsak szakértelmet kívánó, de időigényes feladat is. A Microsoft is belátta, hogy ha-

landó felhasználók számára az igazi adatkezelő és -elemző eszköz az Excel, ezért készített hozzá egy ingyenes *Data Mining Add-ins for Microsoft Office 2007* nevű kiegészítést, amely a modell összeállítását a lehető legjobban automatizálja, hogy a felhasználó a modell lekérdezésére, a valódi jóslásra és tudásfeltárára tudjon koncentrálni.

A bővítmény valójában két Excel- és egy



9. ábra. Attribútumok közötti függőségi viszonyok elemzése BIDS-ben

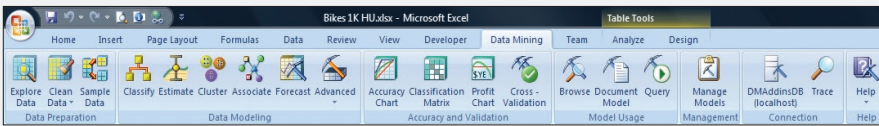
## Tipp

Az SQL Server 2008 adatbányászati funkcióihoz tartozó Excel-bővítményeket igazán érdemes kipróbálni, nagyon gyorsan érhetünk el velük látványos eredményeket, és tanulhatunk bele az adatbányászatba. Ha nincs kéznél telepített Analysis Services, akkor is kipróbálhatjuk a *Table Analysis Tools* funkciókat, ha ellátogatunk a <http://www.sqlserverdata-mining.com/cloud> címre. Csak egy böngészőre lesz szükségünk!

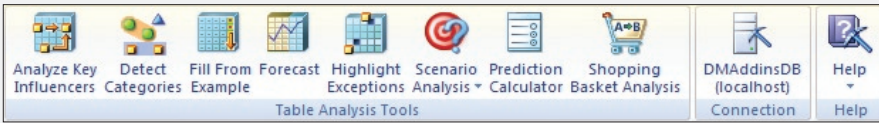
Visio-kiegészítést tartalmaz. A *Data Mining Client for Excel* a szalagon jeleníti meg a *Data Mining* fület, amelyen közvetlenül végezhetjük el a BIDS-ben már megismert műveletek többségét. Bár szinte minden gomb egy-egy varázslót indít el, ezek használatához célszerű ismerni az adatbányászat fogalmait.

A bővítmény másik komponense a jóval egyszerűbb *Table Analysis Tools*, amely az Excel-tábla objektumait okosítja fel. A telepítés után, ha bármelyik táblára kattintunk, megjelenik a *Table Tools* csoportban egy *Analyze* fül és a hozzá tartozó szalagon számos adatbányászati funkció.

Ezek a gombok is varázslókat indítanak el, azonban ezek a varázslók kevésbé technikai



10. ábra. Data Mining-szalag az Excel 2007-ben



11. ábra. Az Analyze-szalag az Excel 2007-ben

részleteket kérdeznek, átlagfelhasználók számára könnyebben emészthetőek.

Bármelyik szalag funkcióit használjuk, ne felejtsük el, hogy a háttérben szükség van az SQL Server Analysis Services-re és azon belül egy adatbázisra, amelyben a felhasználónak jogosultnak kell lennie objektumok létrehozására. A szükséges kiszolgálóparaméterek beállítását és munkaadatbázis létrehozását a bővítmény telepítő könyvtárában található *Microsoft.SqlServer.DataMining.Office.ServerConfiguration.exe* alkalmazás segítségével tudjuk könnyen elvégezni.

## A parancssor varázsa

Üzemeltetők és SQL-guruk fejében felmerülhet, hogy persze szép ez a sok varázsló a Business Intelligence Development Studióban, de hogy lehet ezt elvégezni parancssorból? Az adatbányászati feladatok megoldásához nem elég a T-SQL kifejezőereje, helyette a *Data Mining Extensions* (DMX) nyelvet használhatjuk. A két nyelv között óriási hasonlóság – a DMX a T-SQL kiterjesztéseként is felfogható –, mindkettőben találunk az adatok szerkezetére (*data definition statements*) és az adatok kezelésére (*data manipulation statements*) vonatkozó utasításokat.

Adatbázis-objektumainkat a megszokott CREATE, ALTER és DROP utasításokkal kezelhetjük, létezik például CREATE MINING STRUCTURE és ALTER MINING MODEL utasítás. Természetesen itt is meg kell adnunk a mezők nevét és típusát, továbbá az értékek típusát és a mező célját, valamint modell esetén a választott algoritmust is. Egy új modellt például így hozhatunk létre:

```
CREATE MINING MODEL [KerekparEladasok]
(
    [Azonosító] LONG KEY,
    [Nem] TEXT DISCRETE,
```

```
[Autók Száma] LONG DISCRETE,
[Vásárolt-e] TEXT DISCRETE PRE-
DICT
)
USING MICROSOFT_DECISION_TREES
```

Szintén a *data definition* csoportba tartoznak a mentéshez és a visszatöltéshez használható EXPORT és IMPORT utasítások.

Miután létrehoztuk a *mining structure* és *mining model* objektumainkat, a következő lépés a modell betanítása, azaz feltöltése adatokkal, melyhez az INSERT INTO utasítást használhatjuk. Végül a betanított modellt a SELECT utasítással kérdezhetjük le, amelynek számos formája közül prediktív elemzési feladatokhoz valószínűleg legtöbbször ezt fogjuk használni:

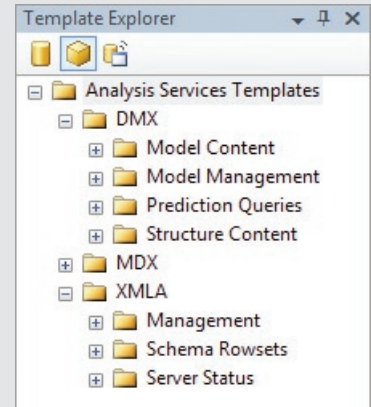
```
SELECT [TOP <row count>] <select expression list>
FROM <model>
[
    [NATURAL] PREDICTION JOIN
    <source data> AS <alias>
    [ ON <column mapping> ]
    [ WHERE <filter expression> ]
    [ ORDER BY <expression> ]
]
```

A DMX nyelv függvényeket is tartalmaz. A SELECT kulcsszó után megadott kifejezésben használhatjuk például a Predict függvényt egy oszlop jóslat értékének lekérdezésére, vagy a PredictProbability függvényt, ha azt akarjuk tudni, hogy egy oszlop milyen valószínűséggel vesz fel egy adott értéket. Szintén függvényeket használhatunk az esetek szűrésére a WHERE kulcsszó után: az IsTrainingCase és az IsTestCase függvények akkor térnek vissza true értékkel, ha az adott eset a modell betanítására vagy tesztelésére használt.

A DMX mellett érdemes megismerkednünk az Analysis Services szkriptelésére – tehát nem csak adatbányászati feladatokra – alkalmas *Analysis Services Scripting Language*

## Tipp

Szkriptek írásához nagy kezdőlapot kaphatunk, ha a Management Studióban megnyitjuk a *Template Explorer* ablakot, és átváltunk az *Analysis Services Templates* csoportra, itt ugyanis 30 DMX- és 18 XMLA-szkript-sablon segíti a leggyakoribb feladatok megoldását.



### Template Explorer az SQL Server Management Studióban

DMX-szkriptek esetén további segítség az IntelliSense-támogatás, illetve hogy a sablon megnyitása után a Query menüben a *Specify Values for Template Parameters* menüpontra kattintva gyorsan megadhatjuk a szükséges paramétereket. Ha bővíteni szeretnénk a sablonok listáját, egyszerűen mentjük el a szkriptjeinket a %ProgramFiles%\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\SqlWorkbench-ProjectItems\AnalysisServices mappába.

(ASSL) nyelvvel is. Az ASSL a szabványos XML for Analysis (XMLA) API-t használja a parancsok és paramétereik leírására, amelyeket SOAP protokollon keresztül, TCP vagy HTTP felett küldhetünk el a szervernek, ahogy azt a BIDS és a Management Studio is teszi. Mindegy, hogy OLE DB, ADO, ADOMD.NET vagy bármilyen más kliensből fordulunk a szerverhez, végső soron a *provider* XMLA-t állít elő, ez ugyanis az Analysis Services platform- és nyelvfüggetlen kommunikációs protokollja.

Az XMLA-szkriptek írását segíti, hogy amikor a Management Studióban Analysis Services objektumokon vagy a *Properties* ablakban kattintunk a *Script* gombra, szintén XMLA-szkriptek keletkeznek. Példaként íme egy modell törlését végző szkript:

```
<Delete xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Object>
    <DatabaseID>Bikes</DatabaseID>
    <MiningStructureID>Forgalom</MiningStructureID>
    <MiningModelID>KereparEladasok</MiningModelID>
  </Object>
</Delete>
```

Ezeket a szkripteket legegyszerűbben az SQL Server 2008 Integration Services vagy a példaprogramok között megtalálható *ascmd.exe* segítségével futtathatjuk, sőt akár saját alkalmazásainkba is beépíthetjük őket az *ADOMD.NET provider* segítségével.

A más platformokkal való együttműködés és kifogástalan kompatibilitás érdekében az SQL Server 2008 Analysis Services a DMX és az ASSL mellett támogatja a *Data Mining Group* által elfogadott, XML alapú *Predictive Model Markup Language* (PMML) nyelvet is.

## Telepítés

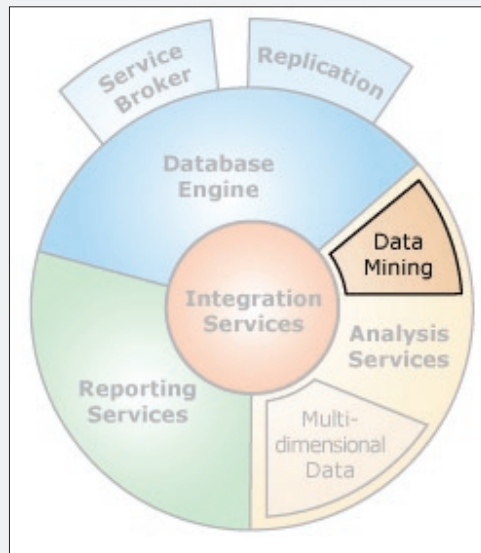
Amennyiben adatbányászati feladatokat kell megoldanunk, az SQL Server 2008 telepítésekor mindenképp válaszunk ki az *Analysis Services* (SSAS) komponenst, ez fogja tárolni az adatainkat és futtatni a modellünket. A telepítő még meg fogja kérdezni, hogy hol szeretnénk tárolni az adatbázisainkat (bátran tegyük át másik partícióra), és hogy mely felhasználókkal szeretnénk futtatni és üzemeltetni a szolgáltatást. Mivel a telepítőt teljesen újraírták, és részben ez veszi át a korábbi *Surface Area Configuration Tool* szerepét, már a telepítésnél gondoljuk végig, hogyan használjuk minimális jogosultságokkal a szolgáltatást. A *Database Engine*-hez hasonlóan, egy számítógépre több, nevesített SSAS-példány (*instance*) is telepíthető egymás mellé, sőt akár SQL Server 2000 és 2005 verziók mellé is telepíthető SQL Server 2008 Analysis Services.

Az adatbázisban található objektumok megtekintésére és szkriptek futtatására szokás szerint az *SQL Server Management Studiót* (SSMS) használhatjuk, csak bejelentkezésnél adjuk meg, hogy nem *Database Engine*, hanem *Analysis Services* típusú kiszolgálóhoz szeretnénk csatlakozni. Ne lepődjünk meg azon, hogy gyakorlatilag nem kapunk grafi-

kus felületet új objektumok létrehozására, ez a funkció ugyanis egy másik eszközben, a *Business Intelligence Development Studióban* kapott helyet.

Az SSMS és a BIDS megfelelő jogosultságok birtokában képes távolról csatlakozni a kiszolgálóhoz, így telepíthetjük őket bányásztársaink munkaállomásaira is. Érdeemes még külön letölteni az *SQL Server Feature Pack* részeként elérhető *Microsoft SQL Server 2008 Data Mining Add-ins for Microsoft Office 2007* kiegészítést, ezzel közvetlenül az Excelből varázsolhatunk modelleket, és futtathatunk lekérdezéseket.

Az egyes eszközökhöz található súgó szókás szerint az *SQL Server Books Online*-ban olvasható, amelyben a *Basic Data Mining Tutorial* címszó alatt hasznos bevezetőt ta-



12. ábra. Az adatbányászati funkciók helye az SQL Serverben

lálunk az eszköz használatába. Az SQL Server példaprojektjei és példaadatbázisai azonban nincsenek a telepítőcsomagban, hanem új helyen, a <http://codeplex.com/SqlServerSamples> címen érhetők el.

## Jogosultságszabályozás

Az adatbányászathoz felhasznált forrásadatok igen értékesek, és sokszor érzékenyek, ezért természetesen kiemelt gondot kell fordítanunk az adatok biztonságára. Az SQL Server Analysis Services hozzáférés-szabályozása aránylag könnyen átlátható: Windows-felhasználóinkat vagy inkább -csoportjainkat szerepkörökhöz rendelhetjük, és ezekre meghatározhatjuk, hogy az adatbázis mely

objektumát érhetik el. Fontos, hogy windowsos integrált hitelesítésről van szó, és hogy felhasználóknak közvetlenül nem adhatunk jogosultságot, csak szerepköröknek. Az SQL Server relációs motorjával ellentétben itt nem használhatunk tiltó (*DENY*) engedélyeket, így egy felhasználó eredő jogosultsága a szerepköreikhez rendelt (megengedő, azaz *ALLOW*) jogosultságok uniója lesz.

Az SSAS kétféle szerepkört ismer: kiszolgálószintű és adatbázis szintű szerepkört. Kiszolgálószinten csak egy *Server Administrators* szerepkör létezik, amely felhasználó ennek tagja, az tetszőleges objektumhoz hozzáférhet, és tetszőleges műveletet végezhet az adott SSAS kiszolgálópéldányban. Noha ez a felhasználói felületen nem látszik, alapértelmezés szerint az operációs rendszer helyi *Administrators* csoportjának felhasználói tagjai lesznek ennek a szerepkörnek<sup>1</sup>, de a telepítő is külön rákérdez, hogy milyen felhasználói fiókokkal szeretnénk üzemeltetni a kiszolgálót.

Az Analysis Services minden egyes adatbázisában definiálhatunk adatbázis szintű szerepköröket. Itt adhatunk *Full control* (*Administrator*), *Process database* vagy *Read definition* jogot az egész adatbázisra, de akár részletesen is megadhatjuk, hogy a szerepkör tagjai mely objektumokhoz férhetnek hozzá. Az 1. táblázat összefoglalja, milyen jogosultságok közül válogathatunk:<sup>2</sup>

Mint látható, van lehetőségünk finoman szabályozni a felhasználó jogosultságait, néhány problémába azonban könnyen belefutunk:

- Az esetek nagy részében az SSAS felhasználóinak nem kell elérniük azokat az adatforrásokat, amelyekre az adatbázis épül. Adatbányászati feladatok esetében azonban ennek épp az ellenkezője igaz, legalább olvasási jog szükséges. Ezenkívül minden egyes adatforrásnál megadhatjuk, hogy a rendszer egy adott felhasználói fiókkal vagy megszemélyesítéssel próbáljon csatlakozni az adatforráshoz.
- Ha a felhasználónak adunk *Drill Through* jogot, akkor lesz lehetősége lefűrni és meg-

<sup>1</sup> Ha ezt nem szeretnénk, a kiszolgáló beállításai között állítsuk a *Security \ Built-In AdminsAreServerAdmins* tulajdonság értékét *false*-ra. Az alapértelmezés *true*.

<sup>2</sup> Amennyiben OLAP-adatokkal dolgozunk, kockákra és dimenziókra is meghatározhatjuk a hozzáférési jogosultságokat.

## 1. táblázat. A jogosultságok választéka

Objektum	Jogosultságok
Database	Full control (Administrator), Process database, Read definition
Data source	None, Read, Read definition
Mining structure	None, Read, Drill Through, Read definition, Process
Mining model	None, Read, Read/Write, Drill Through, Read definition, Browse

nézni az egyes rekordokat, amelyekre a modellünk épül. Amennyiben azok az adatok érzékenyek, ne engedélyezzük ezt a jogot, vagy használjunk Data Source View-t a kényes sorok vagy oszlopok kiszűrésére.

- Ahhoz, hogy egy üzleti elemző vagy egy fejlesztő modelleket hozzon létre vagy teszteljen a szerveren, számára rendszergazdai jogot kell adnunk az adott adatbázisban. Ezzel természetesen arra is feljogosítjuk, hogy bármilyen műveletet végezzen az adatbázis bármely objektumán, akár törölje is azokat. Éles üzemben tehát el kell döntőnünk, hogy mi fontosabb: a fejlesztő szabadsága vagy a már elért eredmények (létrehozott objektumok) védelme. Amennyiben mindkettő, kénytelenek leszünk több adatbázissal dolgozni: az egyikben engedélyezzük új modellek létrehozását és tesztelését, a másikban pedig csak a lekérdezést – ez utóbbihoz ugyanis már nem szükségesek rendszergazdai jogosultságok.

- Amennyiben ki szeretnénk használni a BIDS offline képességeit, mindenképpen Server Administrators jogkörrel kell rendelkezniük. Ha ez számunkra fontos szolgáltatás, akkor célszerű üzembe állítani egy fejlesztői szervert, amelyen a modellek készítői teljes rendszergazdai jogosultságokkal bírnak és egy másik szervert, amelyen a végfelhasználók a lekérdezéseiket futtatják – természetesen minimális jogosultságokkal.

A hozzáférési engedélyek állítását elvégeztetjük grafikusán az SQL Server Management Studióból, a Business Intelligence Developer Studióból vagy akár XMLA-szkriptből is.

### Adatbázisok mentése és szinkronizálása

Ha úgy döntünk (például biztonsági okokból), hogy külön Analysis Services-példányt használunk fejlesztésre és éles üzembe, akkor

érdemes megismerkednünk az adatbázisok mozgatójának lehetőségeivel.

A legegyszerűbb lehetőség a már jól megszokott (ugye megszokott!?) mentés és visszaállítás. Erre találunk lehetőséget az SQL Server Management Studióban, de akár szkriptből is elvégezhetjük. Bár ez utóbbi esetben *Analysis Services Scripting Language* (ASSL) formátumú XML-t kell írunk, nem kell megijednünk a feladattól, messze nincs annyi opció, mint hagyományos SQL-adatbázisok mentése esetén. Íme a mentést végző szkript, amelynek kimenete egyetlen *.abf* (Analysis Services Backup File) fájl:

```
<Backup xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Object>
    <DatabaseID>Bikes</DatabaseID>
  </Object>
  <File>D:\Backup\Bikes.abf</File>
  <AllowOverwrite>false</AllowOverwrite>
  <ApplyCompression>true</ApplyCompression>
  <Password>T1tko5je152o!</Password>
  <Security>CopyAll</Security>
</Backup>
```

Bár az *ApplyCompression* és a *Password* elemek megadása opcionális, ha elhagyjuk, az adatbázisban lévő *Data Source* objektumoknál megadott *Connection String* kódolatlanul kerül a fájlba, azaz bárki elolvashatja az adatforrásokhoz történő kapcsolódáshoz használt jelszót.

A visszaállítás nagyon hasonló, csak a *Backup* helyett a *Restore* elemet kell használnunk:

```
<Restore xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <File>D:\Backup\Bikes.abf</File>
  <DatabaseName>Bikes</DatabaseName>
  <AllowOverwrite>true</AllowOverwrite>
  <Password>T1tko5je152o!</Password>
  <Security>CopyAll</Security>
  <DbStorageLocation xmlns="http://schemas.microsoft.com/analysis-
```

```
services/2008/engine/100/100">
  D:\Data
  </DbStorageLocation>
</Restore>
```

Ennél egyszerűbb megoldás, ha a *Synchronize* elem segítségével egy lépésben végezzük el mindezt, azonban ehhez mindenképp két Analysis Services-példányra lesz szükségünk, ez a módszer ugyanis egy példányon belül nem használható.

Amennyiben XML helyett inkább a T-SQL-re emlékeztető DMX nyelvet preferáljuk, használhatjuk az *EXPORT* és *IMPORT* utasításokat *Mining Structure* és *Mining Model* objektumok mentésére és visszatöltésére. Egy modell mentését az összes hivatkozott objektumával együtt így menthetjük el:

```
EXPORT MINING MODEL [BikesModel] TO 'D:\Backup\Bikes.abf' WITH DEPENDENCIES
```

A szkriptelést kevésbé kedvelők számára a legegyszerűbb megoldás a Start menüben található *Analysis Services Deployment Wizard* elindítása. Ez egy gyorsan végigkattintható varázsló, amely a BIDS projektünk kimeneként előállt *.asdatabase* kiterjesztésű fájl alapján ASSL szkriptet készít, és akár közvetlenül futtat is.

Akarmelyik megoldást választjuk, rendszergazdai jogosultságokra lesz szükségünk a szerverben vagy az adott adatbázisban. Figyeljünk oda a jogosultságok visszaállítására (lásd *Security* elem), mert előfordulhat, hogy ha a mentés idején még nem rendelkezünk rendszergazdai joggal az adott adatbázisban, most a visszaállítás után nem fogunk tudni hozzáférni az adatainkhoz.

## Összefoglalás

A prediktív analízis mind a mai napig komoly kutatási és tudományos területnek számít, de az eszközeink már kezdik megközelíteni azt a szintet, hogy a felhasználók a matematikai modellek részletes ismerete nélkül, az elemzés céljaira koncentrálva oldjanak meg adatbányászati feladatokat. Az SQL Server 2008-cal a Microsoft élen jár az ilyen eszközök fejlesztésében, a bányászmunka egyre kevésbé izzasztó. Csákányt a kézbe, a bányára vár!

Balassy György

Microsoft regionális igazgató

balassy@aut.bme.hu

http://balassy.spaces.live.com

# KI A LEGÉNY A GÁTON?

## Az első találkozás a Reporting Services-zel.

**M**indannyian tudjuk a választ a címben feltett kérdésre, hiszen emlékszünk a történetére, tanúi voltunk boldogságos pokoljárásának. Pelikán Józsefnek hívják a gát hőst. Azért volt hős, mert bukásai ellenére mindig volt ereje csillogó szemmel újrakezdeni. Vajon modern korunk hősei, a jelen Pelikán elvtársai hogyan boldogulnak?

Egyszer még kérni fogunk valamit, Pelikán elvtárs! Ugye, ismerős? Kitől nem kértek még semmit? A saját Virág elvtársam a saját Pelikán elvtársamtól utoljára szerencsére azt kérte, amiben a legjobb volt. Nem kellett az uszodát vagy az Angolparkot, sem a narancstermelést igazgatnia, egyszerűen csak figyelje a Dunát, ami egyre inkább áradt. Pelikánom tette is a dolgát sikeresen, küldözte a jelentéseket a gátról meg a vízállásról. A sikeren felbuzdulva jött az újabb kérés Virág elvtárstól. Kellene gyártani egy jelentést, de nem ám a hagyományos papiros irkafirkát, hanem olyan országosat, amit interneten keresztül az elvtársak is nézegethetnek. Pelikán érezte a veszét: Nem lehetne erre egy képzetesebb elvtársat megkérni? De már tudta a választ: A nemzetközi helyzet egyre fokozódik, nincs időnk egyéni sérelmeinket dédelgetni. Egyelőre csak annyit kérünk magától, hogy szedje össze a gondolatait. Így történt, hogy Pelikánom elkezdett ismerkedni a Reporting Services-zel.

### Kezdjünk hozzá!

A kezdet nem volt egyszerű, hiszen az élet nem habos torta. Pelikán elvtárs pont úgy járt az országos vízállásadatokkal, mint a tűzoltókkal, amikor dohányzás közben elaludt. Ezek sem érkeztek meg időben, ezért nekiállt, és generált egy tesztadatbázist néhány sornyi véletlen adattal. Az adatbázisgyártó kódban van egy példa az SQL 2008-ban újdonságként megjelenő row constructorra, amelyik feltölti a folyók táblát névvel, kóddal és kezdőszinttel. Egy egyszerű CTE-re, amelyik feltölti a vízállástáblát a kezdő és a vége dátum között a folyó vízállásának emelkedésével. Végül megtalálható benne egy trükkös update utasítás, amelyik a kezdeti szinthez, majd az előző naphoz képest minden napra beállítja a vízszint változása alapján a folyó aktuális vízszintjét.

```
create database VizallasDemo;
use VizallasDemo;
create table Folyok (
    folyoazon int,
    folyonev nvarchar(30),
    kezdoszint decimal(8,2));

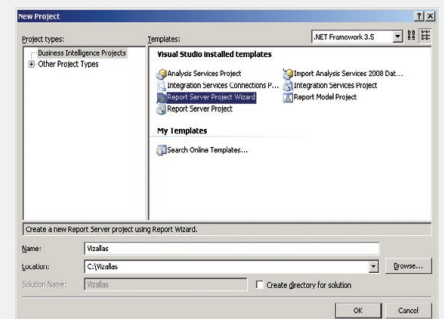
insert Folyok values
(1, 'Duna', convert(decimal(8,2), abs(checksum(NEWID()) % 4) + 1 + abs(CHECKSUM(NEWID()) % 100)/100.0)),
(2, 'Tisza', convert(decimal(8,2), abs(checksum(NEWID()) % 4) + 1 + abs(CHECKSUM(NEWID()) % 100)/100.0)),
(3, 'Dráva', convert(decimal(8,2), abs(checksum(NEWID()) % 4) + 1 + abs(CHECKSUM(NEWID()) % 100)/100.0)),
(4, 'Száva', convert(decimal(8,2), abs(checksum(NEWID()) % 4) + 1 + abs(CHECKSUM(NEWID()) % 100)/100.0));
```

```
create table Vizallas(
    folyoazon int NOT NULL,
    datum date NOT NULL,
    ertekek decimal(8, 2) NULL,
    emelkedes int NULL,
    constraint PK_vizallas primary key clustered
    ( folyoazon asc,
    datum asc));

declare @startdate date = '2006.1.1';
declare @enddate date = '2008.12.31';
;with napok as (
    select @startdate as datum
    union all
    select dateadd(d, 1, datum)
    from napok
    where datum < @enddate)
insert Vizallas
select folyoazon, datum, 0, case when datum = @
startdate then 0 else checksum(NEWID()) % 5 end
as emelkedes
from napok cross join Folyok
option (maxrecursion 0)

declare @szint decimal(8,2);
update Vizallas
set @szint = ertekek = case
    when datum = @startdate then kezdoszint
    else @szint + emelkedes /100.0
end
from Vizallas with(index(1)) inner join Folyok
on Vizallas.folyoazon = Folyok.folyoazon;
```

A tesztadatok előállítás után neki lehetett kezdeni az első riportnak. Az SQL Server 2008 programjai közül a Business Intelligence Development Studiót kell elindítanunk és létrehozni vele egy új Business Intelligence Projektet. A sablonok közül válasszuk a Report Server Project Wizardot. A



1. ábra. Report Server Project Wizard

varázsló végigvezet azon a néhány egyszerű lépésen, amellyel el tudjuk készíteni a vízállásjelentésünket.

Rögtön, nulladik lépésként el is indul a jelentéskészítő varázsló. Ennek a lapnak a megjelenítése opcionális, akár ki is tudjuk kapcsolni, ha teszünk egy pipát az ablak alján levő jelölőnégyzetbe.

Első lépésként meg kell adni, hogy honnan jönnek az adatok. Gyakorlatilag egy connection stringet kell megadnunk adatforrással



2. ábra. Üdvözlő a nagy varázsló

és adatbázissal, de ebben is segít a varázsló. Állítsuk be a helyi gép VizallasDemo adatbázisát. Az adatforrás lehet osztott is. Ebben az esetben elég egyszer megadni, és több jelentésben is fel tudjuk használni. Ha változik a szerver vagy az adatbázis neve (a tesztszerver tesztadatbázisából az éles szerver éles adatbázisára kell átállítani), akkor csak egy helyen, az osztott adatforrásban kell módosítani, és az összes jelentés, amelyik ezt használja, átáll az éles adatbázisra (3. ábra).

A következő lépésben meg kell adni azt a lekérdezést, amelyik az előbb meghatározott adatforrásban fut le, és visszaadja a jelentés sorait, oszlopait. Itt is van grafikus segítség, ha a Query Builderre kattintunk (4. ábra). A mi Pelikánunknak egyszerűbb kézzel beírnia a lekérdezést. Átállítja a nyelvet magyarra, így a DATENAME függvény a hónapok neveit magyarul adja vissza:

```
set language hungarian
```

```
select folyonev, YEAR(datum) as Év,
MONTH(datum) as honpszam, DATENAME(m,
datum) as Hónap, DAY(datum) as Nap, ertek
from Vizallas inner join Folyok
on Vizallas.folyoazon = Folyok.folyoazon
order by Év, Hónap, Nap
```

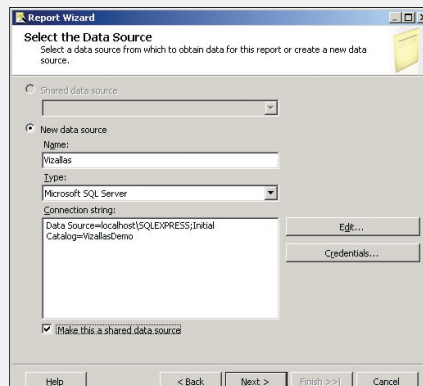
Ha a lekérdezés is megvan, akkor ki kell választani a jelentés megjelenítésének típusát. Itt két irányban haladhatunk tovább a táblázatos vagy a mátrixos megjelenítés felé. Táblázatot választva a varázsló az 5. ábrán látható ablakot jeleníti meg.

Táblázat esetében lapok (Page), csoportok (Group) és tételsorok (Details) közé vihetjük át a lekérdezésben definiált mezőket. Melyik mit jelent?

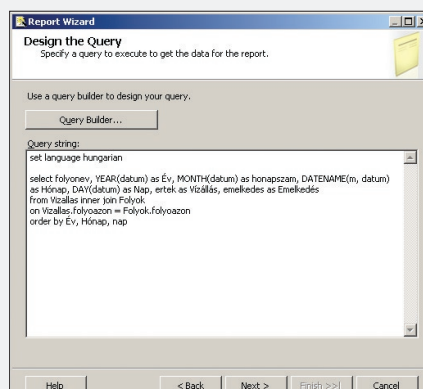
- Page: új értéknél lapot dob a jelentés.
- Group: csoportosítási lehetőség, aggregáló függvények segítségével összeget, átlagot, legkisebb és legnagyobb értéket lehet számolni.
- Details: a lekérdezésből érkező tételsorok. A táblázat elrendezése (6. ábra) lehet lépcsőzetes (Stepped) vagy tömbös (Block).

Az elrendezés kiválasztása után előre definiált stílusokból választhatunk egyet. Az itt kiválasztott stílusnak megfelelően színezi táblázatunkat a Reporting Services. Ezeket a színeket természetesen utólag változtathatunk. Utolsó lépésként egy összefoglalót jelenít meg a varázsló. Ha valami nem úgy van, ahogy szerettük volna, akkor visszaléphetünk és megváltoztathatjuk. Ha minden rendben, akkor már csak nevet kell adni a jelentésünknek és a Finish gombon kattintani (7. ábra).

Készen is vagyunk az első jelentéssel. Nézzük meg, hogy mit varázsoltunk! A táblázat fölötti Preview fülre kattintva a Reporting



3. ábra. Az adatforrás megadása

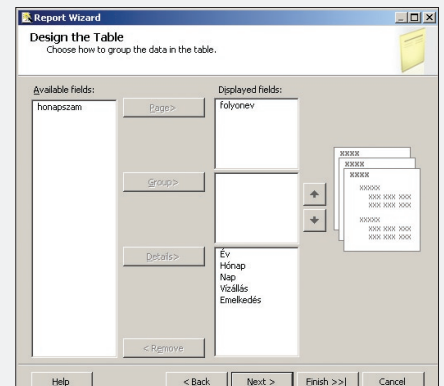


4. ábra. Lekérdezéstervező

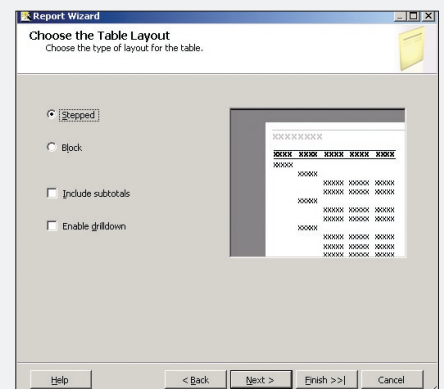
Services elkészíti a riportot. Az adatforrásunkban lefuttatja a lekérdezésünket, majd a színes-szagos táblázatunkban oldalakra törölve megjeleníti. Őszintén? Kicsit sárga, kicsit savanyú, de a mienk! Igazítsunk rajta egy keveset!

## Finomítsunk az eredményen!

Az egyik probléma, hogy a jelentés neve nem jelenik meg csak az első oldalon, kezdjük ezzel. Minden jelentésnek van egy fejléce és egy lábléce. Amit a fejléce vagy láblécebe teszünk, az minden lap tetején, illetve alján megjelenik. A Report menü Add Page Header menüpontján vagy a Report eszközsor Page Header gombján kattintva megjelenik a fejléc. A Vízállás szöveget tartalmazó szövegdobozt mozgassuk át az egérrel a fejlécbe. A fejléc magasságát állítsuk be a szövegdoboz magasságára. (A fejléc vízszintes szaggatott vonalát húzzuk feljebb a szövegdoboz aljáig.) Az átmozgatott szövegdoboz helyén keletkezett rést foltozzuk be, nehogy áttörjön rajta az árvíz, a List1 téglalapot mozgassuk feljebb, közvetlenül a fejléc alá. Ha most megnézzük a jelentést a Preview fülre kattintva, akkor



5. ábra. Táblázattervező



6. ábra. Táblázat elrendezése

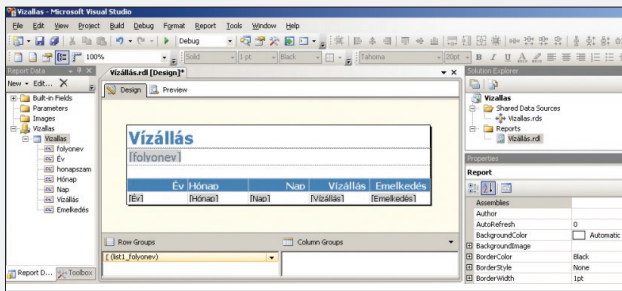
láthatjuk, hogy minden lap tetején ott a jelentés neve.

Egy másik apró probléma a táblázat formázása. A varázsló a szöveges adatokat tartalma-

amelyet február követ, pont, mint a Gergely-naptárban. Pelikán elvtárs adhat magának egy szóbeli vállon veregetést, de csak az egyik vállára. Az elkészült jelentést exportálhat-

juk a Preview fülön a kis kék floppyemlerezre kattintva. A 2008-as változat tartalmazza a Wordbe exportálás lehetőségét is (11. ábra).

A Wordbe exportált jelentést egy e-mailhez csatolva elküldhetjük Virág elvtársnak: nézze meg, erre gondolt-e? Sajnos, mostanában sok



7. ábra. Az első jelentés

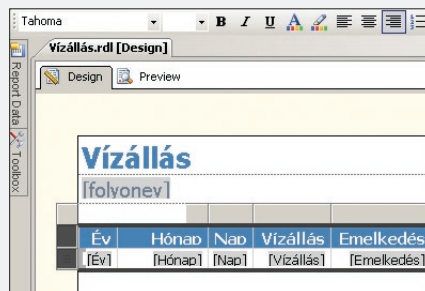
zó mezőket balra, a számokat tartalmazókat pedig jobbra igazítja. Igazítsuk jobbra az összes mezőt. A részletes adatokat tartalmazó táblázat sorain kell kattintani, majd a Report Formatting eszközsor jobbra igazítás (Align Right) gombján. Mindjárt jobban néz ki a jelentés, nincsenek összenöve az Év és a Hónap oszlopok. A táblázaton még lehet igazítani az oszlopok szélességén, például a Nap oszlop simán lehet keskenyebb is, mert sem a mező neve, sem a benne megjelenő értékek nem tartalmaznak sok betűt vagy számot (8. ábra).

A harmadik apróság már picit súlyosabb, és a figyelmetlenségéből adódik. A jelentésben az első hónap az április, majd augusztus következik, aztán december. Első pillantásra össze-vissza vannak a hónapok, pedig nem is. Csak a nevük szerint vannak növekvő sorrendbe rendezve. Ki rendezte őket így? A kezel beírós figyelmetlen Pelikán barátunk, aki villantani akart SQL-tudásával, de a hónap sorszáma helyett a neve alapján rendezett a lekérdezésben. Két módon javíthatjuk ki a hibát a lekérdezésben: kicseréljük a Hónap mezőt a honapszam mezőre, vagy a jelentésben belül kényszerítjük ki a sorrendet. A sorokat megjelenítő táblázatra kell kattintani, erre megjelenik a Sor csoportok (Row groups) között a table1\_Details\_Group. Ezen az egér jobb gombjával kell kattintani és kiválasztani a Csoport tulajdonságok (Group Properties) elemet (9. ábra).

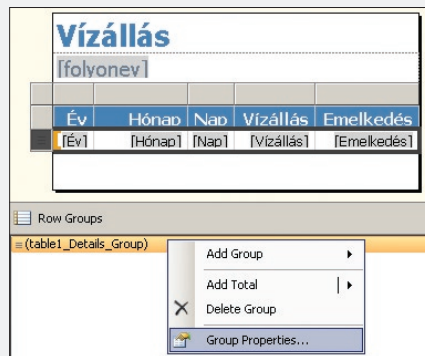
A megjelenő ablakban (10. ábra) a Rendezés (Sorting) fület választva adjuk hozzá az Év, a honapszam és a Nap mezőket A-Z rendezési sorrenddel.

Nézzük meg ismét a jelentést a Preview fülön. Most már januárral kezdődik az év,

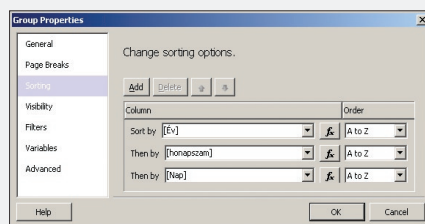
az álmatlan éjszakája, de tetszett neki, továbbította is a Minisztériumba, ahol újabb igényrel álltak elő. Szükség lenne egy csinos kis grafikonra, amelyik vonalakkal mutatja a vízszint ingadozását. Nekem kint kellene lennem



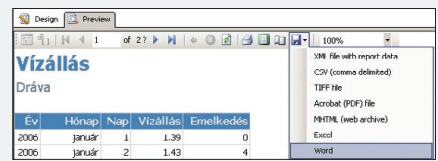
8. ábra. Táblázat formázása



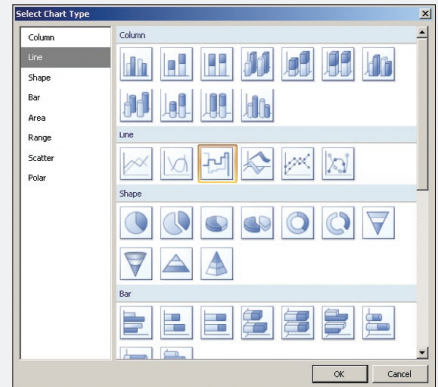
9. ábra. Tételsorok tulajdonságai



10. ábra. Rendezés a hónap száma alapján



11. ábra. Exportálás Wordbe



12. ábra. A Lépcsős vonal diagram

a gáton, tessék engem ebből kihagyni, Virág elvtárs! De megint nem sikerült megmunkálni, mert jött a válasz: menet közben kell az önbizalmat megszerezni, Pelikán elvtárs.

### Grafikon készítése

Gyártunk hát grafikon! Szerencsére nagy halom grafikonfajta közül válogathatunk a 2008-as verzióban. A View menü Toolbox menüpontjára kattintva megjelennek a jelentésre pakolható elemek. Húzzunk a List\_Contents téglalapba egy grafikon (Chart). Válasszuk ki a Vonall diagramok (Line) közül balról a harmadikat (12. ábra), a Lépcsős vonal diagramot (Stepped Line).

Állítsuk be a diagram tulajdonságait, a tengelyek nevét kell még megváltoztatni. A View menü Report Data menüpontot választva előjönnek a lekérdezésben definiált mezők. Ezeket kell áthúzni a megfelelő helyre (13. ábra):

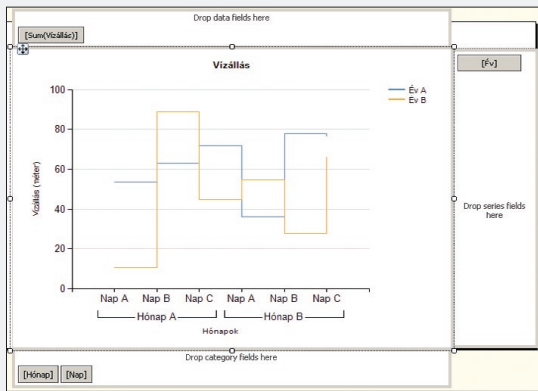
- a Vízállás oszlopot felülre az adatmezőkhöz (Drop data fields here);
- az Év oszlopot jobbra a sorozatokhoz (Drop series field here);
- a honapszam és Nap oszlopokat pedig alulra a kategóriákhoz (Drop category fields here).

Egyetlen dolog van már csak hátra, a hónapok sorrendbe rendezése, de szerencsére ez sem bonyolult. Az előző lépésben a kategóriákhoz bedobott Hónap mezőre kell jobb gombbal kattintani, és a felbukkanó menüben a Kategória-csoporttulajdonságok



(Category Group Properties) menüpontot választani. A megjelenő ablakban a Rendezés (Sorting) fület kiválasztva fel kell venni a listába a honapszám oszlopot az Add gombon kattintva. Ugyanezt a műveletsort kell megismételni a Nap kategóriánál, csak a Nap oszlopot kell hozzáadni a rendezési feltételekhez. A Preview fülön ismét exportálni lehet a jelentést, és elküldeni Virág elvtársnak (14. ábra).

Az eredmények láttán Virág elvtársnak



13. ábra. Diagram testre szabása

könnybe lábada a szeme az örömtől, és megállapítja a makacs tény. Kiadtuk a jelszót: legyen vízállásjelentés, és lett. Mi nem ígértünk a levegőbe. Básty elvtárs most elégedett. Újabb kérés érkezik. Szükség lenne egy másik jelentésre is. (Miről is volt szó eredetileg? Egyszer kérni fogunk valamit! De ez már a hányadik egyszer?) A meglévő jelentés túl hosszú, nagyon sok sora van. A sorokban csak az éveket és a hónapokat kellene megjeleníteni, a napokat pedig az oszlopokban, a táblázat celláiba pedig a napi vízállás adata kerülne. Ezzel jelentősen csökkenne a jelentés hossza. Szerencsére ez sem teljesíthetetlen kérés. Az előző jelentés varázslása közben a táblázatos módot választottuk, de volt még ott egy mátrixos lehetőség is. Nekünk meg pont a mátrixra van szükségünk. A táblázat és a mátrix abban hasonlít egymásra, hogy mindkettőnek vannak sorai és oszlopai, a különbség pedig az, hogy a táblázatnak fixek az oszlopai, a mátrixnak pedig dinamikusak.

Nézzük meg a gyakorlatban. Adjunk a projekthez még egy riportot. Jobb oldalt a Solution Explorerben a Reportsra kattintva válasszuk ki az Add New Report elemet. Elindul a már ismerős jelentéskészítő varázsló. Adatforrásnak válasszuk az előzőleg létrehozott osztott adatforrást. Lekérdezzünk

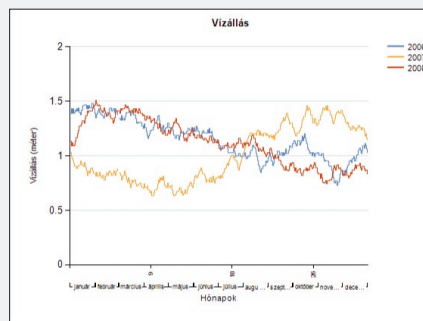
adjuk meg az előzőleg már jól bevált lekérdezést. A jelentés típusánál viszont evezünk bátran ismeretlen vizekre, és válasszuk ki a Mátrix típust (15. ábra).

Itt négy részre van osztva az ablak. Mit jelentenek az egyes részek?

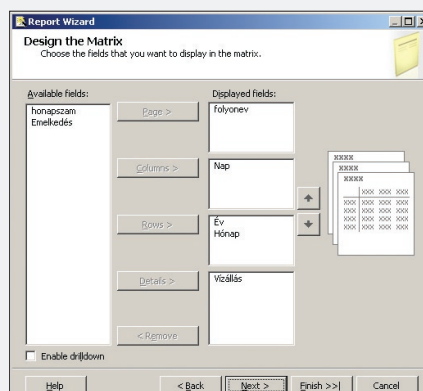
- Page: új értéknél lapot dob a jelentés (pont, mint a táblázatnál);
- Columns: mi kerüljön a mátrix egyes oszlopaiba;

- Rows: mi alapján csoportosítsuk a mátrix sorait;
- Details: a lekérdezésből érkező tételesorok (pont, mint a táblázatnál).

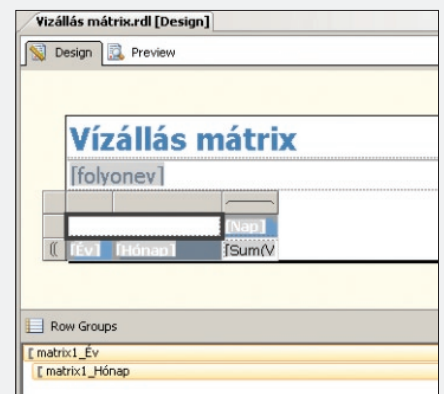
A különbség abból adódik, hogy a táblázatnál levő sor szerinti csoportosítás mellett itt az oszlopok szerint is lehet csoportosítani. Az ábrán látható módon válasszuk ki a mezőket a megfelelő régióba. A következő lépésben a táblázathoz hasonlóan megadhatjuk a mátrix stílusát. Utolsó lépésként adjunk nevet a jelentésnek, és már vége is a varázslásnak. Az Év és a Nap oszlopok most is szélesebbek, mint amire szükség van, nyugodtan vegyük őket



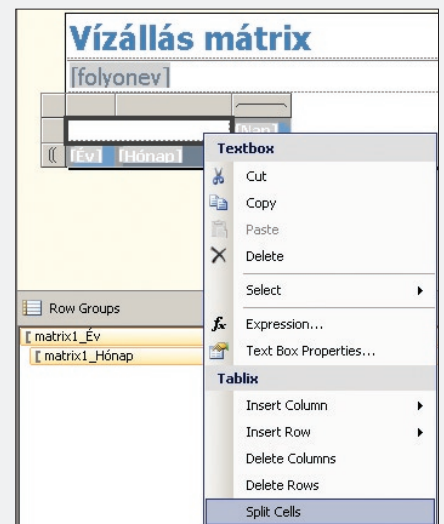
14. ábra. A kész vízállásdiagram



15. ábra. Mátrixtervező



16. ábra. Mátrixsor- és oszlopcsoportok

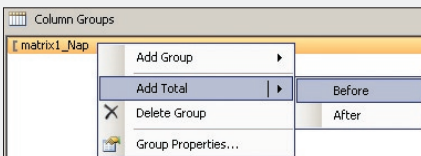


17. ábra. Fejléccella kettéhasítása

keskenyebbre. A jelentés neve most is csak az első oldalon látszik, ha szeretnénk, hogy minden lapon látszódjon, a táblázatnál megismert módon kell eljárunk. Szintén ismert probléma a hónapok sorrendje. Itt is április az első. A javítás szintén hasonlóan történik, mint a táblázatnál. Meg kell változtatni a sorrendet a megfelelő csoportosítási szinten. A táblázatra kattintva megjelennek a sor- és oszlopcsoportok. A matrix1\_Hónap sor csoporton jobb egérgombbal kattintva válasszuk a Csoporttulajdonságok (Group Properties) menüelemet. A felbukkanó ablak Rendezés (Sorting) fülén kattintva változtassuk meg a rendezési sorrendet a Hónap oszlopról a honapszám oszlopra (16. ábra).

További hiányosság, hogy az Év és Hónap oszlopoknak nincs feliratum. Ezt is könnyen orvosolhatjuk. Hasítsuk ketté a bal felső cellát, majd írjuk be a bal oldaliba az Év, a jobb oldaliba pedig a Hónap szavakat. Színezhetjük is a két új cellát. Az Év cella hát-

térszínét (BackColor) állítsuk be az alatta levő cella színével megegyező világoskékre (#6e9eca), a Hónap cella háttérszínét pedig az alatta levővel megegyező SlateGrayre. Mindkét cellánál állítsuk a megjelenés színét (Color) fehérre (White), a betűtípust (Font összetett tulajdonság FontWeight tulajdonsága) pedig Félkövrre (Bold). Így egységes megjelenést adtunk a mátrix fejlécének.



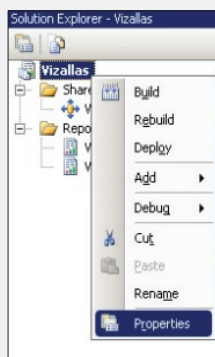
18. ábra. Összesítő oszlop hozzáadása

Vizállás mátrix												
Dráva												
Év	Hónap	Min	Max	1	2	3	4	5	6	7	8	9
2006	január	1.38	1.48	1.39	1.43	1.39	1.43	1.39	1.40	1.41	1.38	1.39
	február	1.35	1.44	1.35	1.38	1.39	1.41	1.43	1.40	1.41	1.38	1.39
	március	1.25	1.41	1.34	1.34	1.33	1.37	1.33	1.34	1.31	1.31	1.27
	április	1.16	1.37	1.29	1.26	1.22	1.19	1.16	1.18	1.25	1.26	1.30
	május	1.15	1.26	1.26	1.22	1.22	1.25	1.25	1.23	1.24	1.21	1.21
	június	1.12	1.27	1.26	1.22	1.23	1.27	1.25	1.23	1.13	1.12	1.13
	július	0.99	1.13	1.13	1.10	1.09	1.05	1.08	1.09	1.02	0.99	1.00
	augusztus	0.85	1.10	1.01	1.03	0.99	0.97	0.98	1.01	0.91	0.91	0.95
	szeptember	0.91	1.11	1.00	1.03	0.99	0.95	0.95	0.95	0.93	1.09	1.08
	október	0.99	1.20	1.11	1.14	1.13	1.11	1.09	1.1	0.91	1.07	1.11
	november	0.73	1.03	1.03	1.01	1.01	1.00	1.02	1.1	0.98	0.79	0.83
	december	0.84	1.11	0.86	0.86	0.84	0.87	0.85	0.8	0.91	1.07	1.11
2007	január	0.82	1.03	1.03	0.99	0.97	0.96	0.93	0.9	0.90	0.89	0.86
	február	0.79	0.88	0.81	0.85	0.81	0.81	0.82	0.7	0.82	0.86	0.88
	március	0.70	0.86	0.84	0.84	0.82	0.78	0.79	0.7	0.86	0.75	0.72
	április	0.63	0.81	0.74	0.71	0.71	0.72	0.69	0.6	0.68	0.72	0.71
	május	0.63	0.76	0.71	0.71	0.72	0.73	0.69	0.6	0.63	0.76	0.75
	június	0.75	0.89	0.78	0.81	0.82	0.80	0.80	0.7	0.84	0.79	0.79
	július	0.79	1.02	0.81	0.81	0.79	0.80	0.80	0.7	0.85	0.90	0.93
	augusztus	1.04	1.24	1.04	1.07	1.10	1.13	1.13	1.1	1.18	1.12	1.19
	szeptember	1.15	1.39	1.17	1.19	1.19	1.17	1.16	1.1	1.16	1.19	1.30
	október	1.18	1.46	1.26	1.24	1.21	1.18	1.16	1.1	1.21	1.34	1.34

19. ábra. A vizállásmátrix végeredménye

Virág elvtárs ismét boldog. Tiszta szerencse, hogy van két füle, különben körbeérne a mosoly a fején. Bátya elvtárs is örül, de fent a Minisztériumban azt is szeretnék látni, hogy havonta mennyi volt a legalacsonyabb és a legmagasabb vizállás (17. ábra).

Ezt is roppant könnyű megoldani. A mátrixunk matrix1\_Nap nevű oszlop csoportját az éget jobb gombjával kattintva az Add Total elemek közül a Before-t választva kapunk egy pluszoszlopot, amelynek a Total szövegét cseréljük le Minre, az alatta levő cellában a képletet (jobb gomb a cellán, majd Expression) is cseréljük le a =Sum(Fields!Vizállás.Value) értéket =Min(Fields!Vizállás.Value)-ra. Ezzel már megkaptuk a havi legkisebb vizállást (18. ábra).



20. ábra. Projekttulajdonságok

A legmagasabb vizállást ugyanígy adhatjuk a jelentéshez, csak a Total oszlopnevet kell Maxra cserélni, a képletben pedig a Sum függvényt Maxra. Íme a végeredmény (19. ábra).

## Közzététel

Örülnek az elvtársak, egyelőre nincs további kérésük. Csak Pelikán elvtárs mosolya nem teljesen őszinte. Ha megjönnek a valós adatok, akkor folyamatosan zaklatni fogják majd a Minisztériumból, hogy generálja le a jelentéseket a legfrissebb adatokkal? Hát azért ez már mindennek a teteje lenne! Szerencsére megoldható, az elkészült jelentések közzététele a minisztériumi dolgozók és az elvtársak számára. A projekt tulajdonságai között (jobb gomb a Solution Explorerben a projekt nevének, majd a Properties menüpont) (20. ábra).

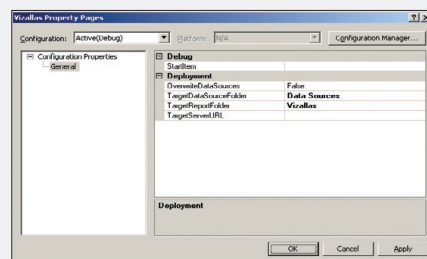
A megjelenő ablakban a következőket állíthatjuk be:

- TargetServerURL: ez a Report Server virtuális könyvtára az Internet Information Services-en (IIS);
- TargetDataSourceFolder: ebbe a könyvtárba kerülnek az adatforrások;
- TargetReportFolder: ebbe a könyvtárba kerülnek a projektben levő jelentések;
- OverwriteDataSources: ha már a megadott helyen van azonos nevű adatforrás, akkor felülírja-e? Alapértelmezetten hamisra van állítva az értéke, így tesztkörnyezetben nyugodtan továbbfejlesztjük a jelentéseket, és amikor publikáljuk őket,

akkor az éles szerverre mutató adatforrásokat nem írjuk felül a tesztkörnyezetben levő szerver- és adatbázisnévvel (21. ábra).

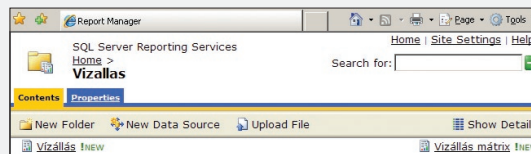
Honnan tudjuk kideríteni, hogy hol is van ez a virtuális könyvtár? Az SQL Server 2008 programok között a Configuration Toolsban található Reporting Services Configuration Manager segít nekünk ebben. Kapcsolódás után a bal oldali részen a Web Service URL-re kattint-

va a legelső adat, a Report Server Web Service URLs, tartalmazza a szükséges virtuális könyvtár címét. A másik fontos címhez



21. ábra. Jelentés közzététele

ismét a bal oldalon kell kattintani a Report Managerre. Itt kapjuk meg azt az URL-t, amelyet elküldhetünk az elvtársaknak, ezentúl ne minket zaklassanak, hanem látogassák meg a levélben elküldött oldalt, és onnan töltsék le magunknak a jelentést, olyan formátumban, amilyenben szeretnék. Miután megszereztük és beállítottuk a közzétételhez szükséges adatokat, cselekedjük meg, amit



22. ábra. Ezt látják az elvtársak a közzététel után a Vizállás mappában

megkíván a haza. Jobb gomb a projekt nevének, és válasszuk a Deploy menüpontot.

Mindenki nagyon elégedett a pilotprojekt eredményével, Pelikán is visszament a gátára, mert a vizállás egyre emelkedett, és kellett a legény. Azonban a nemzetközi helyzet fokozódik. Megjelent a dekadens nyugatról Gustaaf van Dijk, a holland gátör, és nem füves cigivel, hanem ingyenes vizállásjelentő megoldásával kezdte kábítani dicső vezetőinket. De az ifjú Pelikánunkat sem kell féltetni, már a kezdet kezdetén a torkánál ragadta meg a kérdést, az övé is pont ugyanannyiba kerül. A Microsoft Letöltőközpontból szerelte be a projekthez szükséges komponenseket. Letöltötte, majd feltelepítette az ingyenes .NET Framework 3.5 SP1-et, Windows Installer 4.5-öt, az ingyenes PowerShell 1.0-át, és az ingyenes Microsoft SQL Server 2008 Express with Advanced Services-t, majd erre a konfigurációra készítette el a riportokat. A pénz beszél, a kutya ugat, a tények pedig makacs dolgok, szegény Gustaaf barátunk pedig tekerhetett egyet hazáig.

Hangyál Zoltán

zoltan.hangyal@netacademia.net  
oktató, MCDBA, MCAD, MCT

# MICROSOFT OFFICE PERFORMANCEPOINT SERVER 2007

Egy termékben egyesíti azokat a funkciókat, amelyeket összefoglaló néven csak teljesítménymenedzsmentként szoktunk emlegetni.

**A** PerformancePoint Server (röviden PPS) a Microsoft üzletiintelligencia-termécsaládjának a zászlóshajója. A cikkben áttekintjük, hogy mire tudjuk használni a PerformancePoint Servert, és azt, hogy miből áll össze egy PerformancePoint Serverre épülő teljesítménymenedzsment-alkalmazás.

## Teljesítménymenedzsment

Ahhoz, hogy a PerformancePoint Server szerepét megértsük, először azt kell tisztáznunk, hogy mit is értünk teljesítménymenedzsmenten.

Ideális esetben egy cégnek vannak céljai, és a célok eléréséhez van kidolgozott stratégiája, a stratégiához megfelelő végrehajtási terv kapcsolódik, a folyamatokat, eredményeket pedig mérni és ellenőrizni lehet. Ezt egy bizonyos cégmért felett ma már nem lehet kézi irányítással és személyes ellenőrzéssel, ad hoc beavatkozással megvalósítani. Szükség van tehát arra, hogy a célok eléréséhez, a stratégia megvalósításához szükséges folyamatokat kialakítsuk, a cég egészének, egyes részeinek és dolgozóinak tevékenységét a stratégiához igazítsuk. Képesnek kell lennünk annak mérésére, hogy az egyes tevékenységek mennyire felelnek meg a céloknak, a stratégiának, és arra, hogy szükség esetén beavatkozzunk a folyamatokba. Fontos az is, hogy képesek legyünk a folyamatokon változtatni, javítani és a változások hatását nyomon követni. A teljesítménymenedzsment feladata az, hogy a cég céljai megvalósuljanak, a stratégia és a tevékenységek összhangban legyenek, folyamatosan nyomon tudjuk követni és előre tudjuk jelezni a cég teljesítményét.

A teljesítménymenedzsment megvalósításához rengeteg információra van szükség, és nem mindegy az sem, hogy ezek az információk milyen formában, mennyi idő alatt és mennyire megbízhatóan állnak rendelkezésre. Az értelmezhető, releváns információkon kívül nagyon fontos

az is, hogy azok el is jussanak a felhasználóig. Ugyancsak fontos, hogy a teljesítménymenedzsment mint folyamat megfelelően szabályozott és megbízható legyen, hiszen ezen keresztül mérjük magát céget, ez alapján avatkozunk be a cég működésébe, így az itt elkövetett hibák nagyon sokba kerülhetnek.

A teljesítménymenedzsmentet három fő területre oszthatjuk.

**Teljesítménymonitorozás.** A teljesítmény méréséhez kapcsolódó tevékenységeket soroljuk ide, mint például a mutatószámok, scorecardok, jelentések kialakítása, figyelemmel kísérése, azaz a cég és egyes részelei, a dolgozók teljesítményének számszerűsítése és megjelenítése. A teljesítménymonitorozás tehát azokra a kérdésekre ad választ, hogy mi történt – vagy aktuálisan mi történik – a vállalatnál.

**Elemzés.** Az előre meghatározott mutatószámokon, jelentéseken kívül gyakran van szükség arra, hogy bizonyos adatokat tovább elemezzünk, az adatok összefüggéseit vizsgáljuk, hogy jobban megérthessük a háttérben zajló folyamatokat. Az elemzés segítségével tudunk olyan kérdésekre válaszolni, miért így alakultak a mutatószámok, miért ilyenek az eredmények, és milyen eredmények várhatóak.

**Tervezés.** A tervezés a teljesítménymenedzsment motorja, terv nélkül nincs mit



1. ábra. Teljesítménymenedzsment

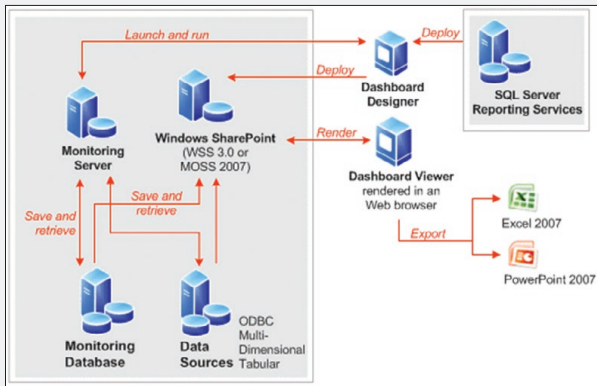
mérni, nincs mihez hasonlítani a teljesítményt, és nem tudjuk azt sem, hogyan fog alakulni az üzlet. Ahhoz, hogy a céljainkat el tudjuk érni, meg kell terveznünk azt, hogy hogyan akarjuk azokat elérni. Terveznünk

dásunkat, ráadásul a megoldás megépítésének jelentős része felhasználói feladattá egyszerűsíthető, az üzleti elemzők maguk állíthatják elő a webes megjelenítőfelületeket, mutatószámokat, riportokat. Az egész rendszer egyetlen felhasználói felületen kezelhető az adatforrások definiálásától kezdve a riportok integrálásán át az eredmények publikációjáig, megszabadítva minket számos munkaidőesztendő feladat végrehajtásától.

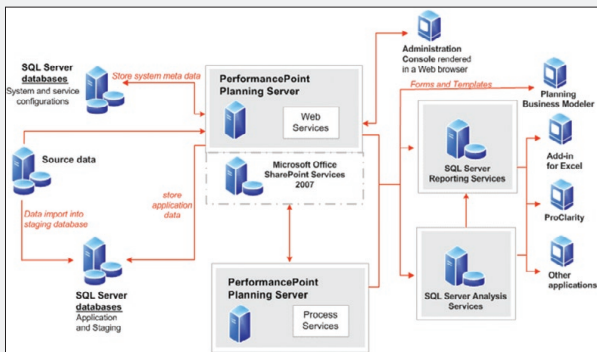
javosultsági rendszerrel, valamint adatintegrációs eszközökkel támogatja a tervezést. Felhasználói felülete az Excel, ebben történik az adatbeviteli formok kialakítása és a felhasználók az Excelen keresztül tölthetik be, kérdezhetik le a tervezési adatokat. Az Excelfelület mögött azonban egy relációs és egy OLAP-adatbázis található, így az Excelben adattárolás nem történik, az adatok mindig egy központi adatbázisban tárolódnak, amelyet a PerformancePoint Server komponensei tartanak karban, így biztosítva a megbízhatóságot és a konzisztenciát. A tervezési folyamatot tervezési ciklusok, feladatok kialakításával támogathatjuk, amelyek automatikusan értesítik a felhasználót a végrehajtandó tevékenységekről, és felügyelik a jóváhagyási folyamatot is. Mindehhez egy igen részletes menő jogosultsági rendszer tartozik, amellyel korlátozhatjuk a modellekhez, dimenziókhoz és az egyes elemi adatokhoz történő hozzáférést is. Mindezekkel az eszközökkel olyan megoldásokat építhetünk, amelyek teljesen le tudják fedni, és nagymértékben automatizálják egy-egy cég pénzügyi, kontrolling, értékesítési, humán erőforrás- stb. tervezését.

Elemzés. Az elemzőnézetek, az Excel és a ProClarity (az OLAP-kockák elemzőfelülete) segítségével a felhasználók részleteiben elemezhetik az adatokat. A PerformancePoint Server adatelemzési funkcionalitása erősen támaszkodik az SQL Server Analysis Server OLAP-kockáira, amelyeket egy monitorozási megoldáshoz külön eszközökkel kell összeállítanunk, a tervezési modul azonban eleve OLAP-kockákra épül, és ezek azonnal használható elemzési célra is.

A továbbiakban megnézzük, hogy a PerformancePoint Serverrel hogyan lehet mindenre kiterjedő teljesítménymenedzsment-megoldást megvalósítani.



2. ábra. A Monitoring Server architektúrája



3. ábra. A Planning Server architektúrája

kell, és előre kell jeleznünk a bevételeinket, a költségeinket, az erőforrásokat, az ügyféleggedettséget, és még folytathatnánk a sort. A tervezés tehát arra koncentrál, hogy mit szeretnénk elérni, azt hogyan akarjuk megvalósítani, és szerintünk mi fog történni.

### Mire használhatjuk a PerformancePoint Servert?

A továbbiakban részletesen megnézzük majd, hogy miből áll egy PerformancePoint Serverre épülő teljesítménymenedzsment-megoldás, de előbb érdemes néhány pontban áttekinteni, hogy mire is használhatjuk a PerformancePoint Servert.

**Teljesítménymonitorozás.** A PerformancePoint Serverben foghatjuk össze a teljes monitorozó eszköztárat a teljesítménymutatóktól kezdve a komplex elemzőnézetekig. Különböző adatforrásokból, számos riporttípusból, elemzőnézetből építhetjük fel a megol-

**Tervezés.** A PerformancePoint Server kész pénzügyi tervezési modellekkel, előre definiált dimenziókkal, számításokkal, az adatbázisok automatikus létrehozásával és karbantartásával, automatikus és szkriptelhető deploymenttel, fejlett

### Architektúra, szükséges infrastruktúra

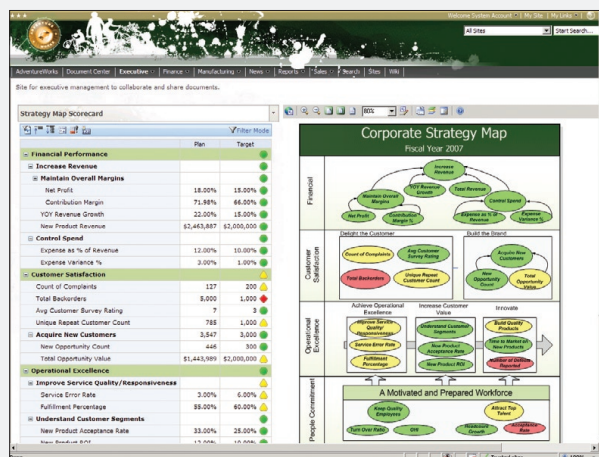
Megoldásunkhoz először is szükségünk van valamilyen infrastruktúrára. A PerformancePoint Server két fő komponensből áll:

- a Monitoring and Analysis: a teljesítménymonitorozást és az elemzést tartalmazó szolgáltatások;

- Planning: az üzleti tervezést támogató szolgáltatások.

A két komponenst egymástól függetlenül is használhatjuk, telepíteni is külön kell őket. Ha tehát a teljesítménymenedzsment-alkalmazásunk nem tartalmazza a tervezést, akkor elegendő csak a monitorozáshoz szükséges komponenseket telepítenünk.

A Monitoring Server architektúráját a 2. ábra szemlélteti.



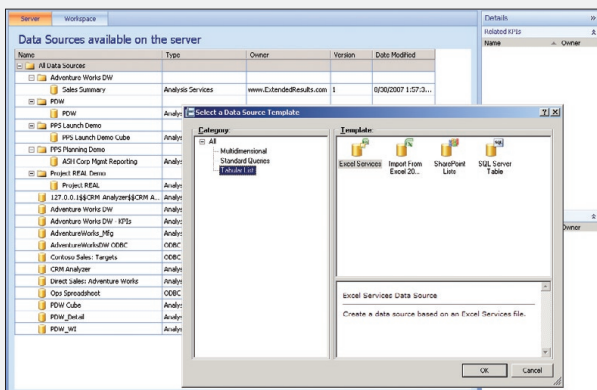
4. ábra. Scorecard és a kapcsolódó stratégiai térkép

A komponens központi eleme a Monitoring and Analysis Service, egy webszolgáltatás, amely egy SQL Server-adatbázisban tárolja a különböző elemek (KPI-k, scorecardok, irá-

tatás és a Planning webszolgáltatás. Planning szolgáltatáshoz az SQL Server Enterprise változatára van szükségünk, és természetesen itt is kell egy Windows 2003-as szerver, az IIS és az Active Directory.

A Planning egyes komponenseit – a PerformancePoint Business Modelert, amely a modellek kialakításának eszköze, valamint a tervezési űrlapok elkészítéséhez, kitöltéséhez szükséges Excel Add-int – a munkaállomásokra kell telepíteniük.

Egyszerű esetben minden komponens egy gépre feltelepíthető, de célszerű legalább az SQL Servert és a MOSS szervert külön gépre telepíteniük. Szükség esetén a Monitoring és Planning komponensek is külön gépen lehetnek.



5. ábra. Adatforrások definiálása a Dashboard Designerben

nyitópultok stb.) definícióit, és gondoskodik a KPI-k, scorecardok, táblázatok, grafikonok forrásadatainak eléréséről. Az irányítópultok megjelenítése a SharePointon történik, a SharePointra telepített PerformancePoint komponensek segítségével. A fentiekből már kiolvasható, hogy a Monitoring komponenshez a SharePointon (WSS vagy MOSS) kívül legalább egy Windows Server 2003-ra, IIS-re és SQL Serverre van szükségünk, és természetesen Active Directoryra a bejelentkezéshez és a jogosultsági rendszer támogatására. (Megjegyzés: ha a KPI-k, táblázatok, grafikonok adatforrásoként használt SQL Server nem ugyanazon a gépen van, mint a Monitoring szerver, és a felhasználó identitását tovább szeretnénk vinni az adatforrás felé a jogosultsági rendszer miatt, akkor Kerberos-otentikációt is kell implementálnunk.)

Ügyféloldalon a megjelenítő eszköz a böngésző, Internet Explorer 6 vagy későbbi változat. A fejlesztőeszköz pedig a Dashboard Designer, egy Office 2007 felhasználói interfésszel rendelkező click-once alkalmazás, amellyel grafikusan szerkeszthetjük a monitorozási funkciók felhasználói interfészeként szolgáló irányítópultokat és azok építőelemeit. A Dashboard Designert csak azoknak a felhasználóknak kell telepíteniük, akik az irányítópultokat publikálják, karbantartják.

Amennyiben az üzleti tervezést is szeretnénk támogatni, szükségünk van a Planning Server komponensre is.

A Planning Server két legfontosabb eleme a Planning Process Service Windows-szolgál-

### Teljesítménymonitorozás

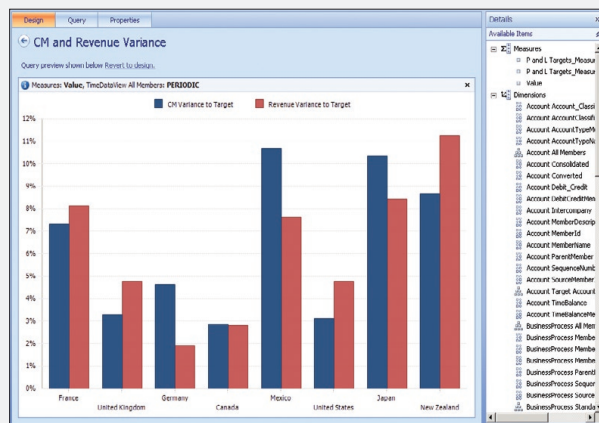
A PerformancePoint Server a teljesítménymonitorozást webes irányítópultokkal támogatja. Az irányítópultok olyan SharePointba ágyazható weblapok, amelyek speciális web-

kat a legfontosabb mérőszámokat, amelyek a teljesítmény méréséhez szükségesek. Ilyen lehet például a tervezett és aktuális bevétel viszonya, a nyitott lehetőségek ideális és aktuális száma, az eladott termékek mennyiségének, összetételének alakulása. Ezekből alakíthatjuk ki a KPI-eket és a scorecardokat. A scorecardokhoz kapcsolódóan érdemes elgondolkozni azon, hogy stratégiatérképen is szemléltessük az egyes mutatók összefüggéseit. Ezek után meg kell határozni, hogy milyen további riportokra, elemzési lehetőségekre van szükségünk, és hogyan, milyen formában fogjuk ezeket a felhasználói interfészen megjeleníteni. Ha ezzel készen vagyunk, azaz pontosan tudjuk, hogy mit akarunk megvalósítani, akkor meg kell vizsgálnunk, hogy a mutatószámok, jelentések előállításához milyen adatokra van szükségünk, ezek hol találhatóak, és milyen módon állnak rendelkezésre. Előfordulhat az is, hogy a megoldás részeként SQL- vagy OLAP-adatpiacokat kell építeniük, és ezekből kell előállítani a szükséges információkat. (Ha gyors megoldást akarunk, adatforrásként akár egy Excel-táblázatot is használhatunk, és ha van SharePoint-szerverünk, a jelentéseket is készíthetjük Excelben.)

A következő lépés, hogy a PerformancePoint Server Dashboard Designerrel implementáljuk és publikáljuk az irányítópultokat.

A PerformancePoint irányítópultok a következő elemeket tartalmazhatják:

**Teljesítménymutató (röviden: KPI).** A teljesítménymutatók konkrét, számszerűsített információt tartalmaznak a cég, egy speciális terület vagy egy-egy közreműködő teljesítményére vonatkozóan. Négy részből



6. ábra. Jelentés készítése a Dashboard Designerben

partokat tartalmaznak, egy-egy lapra csoportosítva az egy kérdéskörhöz tartozó mutatószámokat, jelentéseket és egyéb fontos információkat. Az irányítópultokon a felhasználók szűrőfeltételeket állíthatnak be, egyszerre szűrhetik a különböző elemeket, és interaktívan dolgozhatnak az egyes webpartokkal.

Képzelnék el, hogy feladatunk egy cég kereskedelmi teljesítményének a monitorozása. Ehhez először meg kell határozni azo-

állnak, egy célértékből, egy aktuális értékből és a jelenlegi státuszt, valamint a trendet jelképező egy-egy grafikus indikátorból, amelyek általában háromértékűek, egy-egy ikon jelöli a célérték és az aktuális érték, illetve trend alapján kiértékelte teljesítményt. Ilyen teljesítménymutató lehet például az árbevétel alakulása a kitűzött célhoz képest. A KPI-eket az irányítópultokra nem közvetlenül vezetjük ki, hanem scorecardokba foglaljuk.

**Scorecard.** A scorecard egy vagy több teljesítménymutatót tartalmaz. A teljesítménymutatók a scorecardon csoportosíthatók, összesíthetők, és dimenziók szerinti bontás-

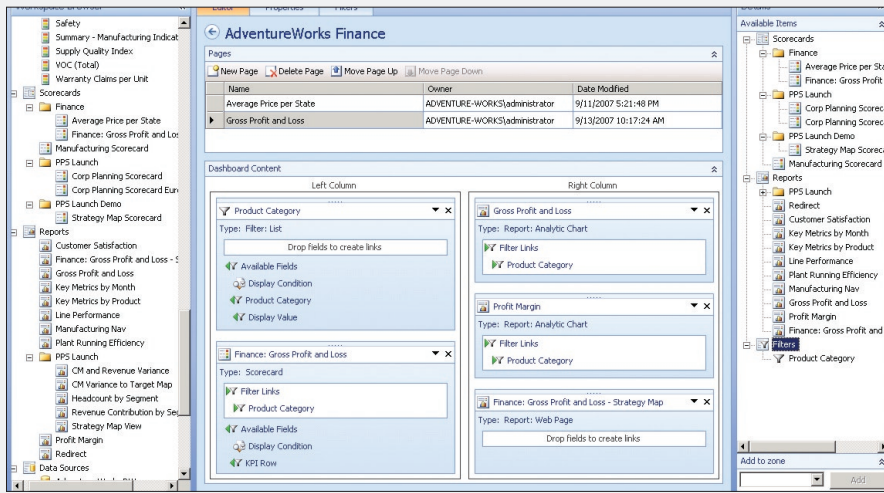
lyek lehetnek SQL Server-, Analysis Services- vagy ODBC-adatforrások, Excel-fájlok vagy Excel Servicesbe publikált munkafüzetek és SharePoint-listák.

peznek a Dashboard Designerben összeállítható táblázatok és grafikonok (Analytic Grid, Chart), amelyeket helyben készítünk el.

Végül elhelyezzük az egyes elemeket az irányítópultokon, definiáljuk a szűrőfeltételeket, és összekapcsoljuk azokat az egyes irányítópult-elemekkel. A szűrőfeltételek is táplálkozhatnak adatforrásokból, könnyen definiálhatunk például egy olyan szűrőt, amely a kereskedelmi területek listáját egy adatbázisból tölti fel. A szűrőfeltételek között kitüntetett szerepe van az időnek, lehetőségünk van például arra, hogy olyan feltételt állítsunk be, amely mindig az utolsó három hónapot vagy az aktuális évet tartalmazza.

Ha elkészült az irányítópult, a Dashboard Designer segítségével ki is próbálhatjuk azt egy erre a célra rendelkezésre álló tesztalkalmazással.

Miután minden teszteltünk, az irányítópultokat a Dashboard Designerből publikáljuk a megfelelő SharePoint-dokumentumtárakba.



7. ábra. Irányítópult tervezése a Dashboard Designerrel

ban jeleníthetők meg. A PerformancePoint scorecardjai szűrhetők, megjegyzéseket fűzhetünk hozzájuk, és összekapcsolhatjuk őket más irányítópult-elemekkel. Egy adott KPI-hez például hozzákapcsolhatunk egy jelentést, amely a KPI-re kattintva megjelenik az irányítópulton.

**Riport.** Több riporttípus közül választhatunk, attól függően, hogy milyen tartalmat szeretnénk megjeleníteni, milyen fokú interaktivásra van szükségünk, és milyen technológiával rendelkezünk. Megjeleníthetünk SQL Server Reporting Services-jelentéseket, Excel Servicesbe publikált Excel-dokumentumokat, ProClarityvel készített elemzéseket, stratégiai térképeket, interaktív PerformancePoint-táblázatokat és -grafikonokat, valamint tetszőleges weblapokat is, amelyek az url-en keresztül akár paraméterezhetők is.

**Stratégiai térkép.** A stratégiai térkép olyan ábra, amelyen vizuálisan szemléltetjük a cég vagy egy részleg teljesítményének különböző elemeit és azok összefüggéseit. A PerformancePoint stratégiai térkép nem más, mint egy Visio-ábra, amelyet egy scorecardhoz illeszthetünk, és a scorecardon szereplő KPI-k színét és adatait a Visio objektumaihoz kapcsolhatjuk.

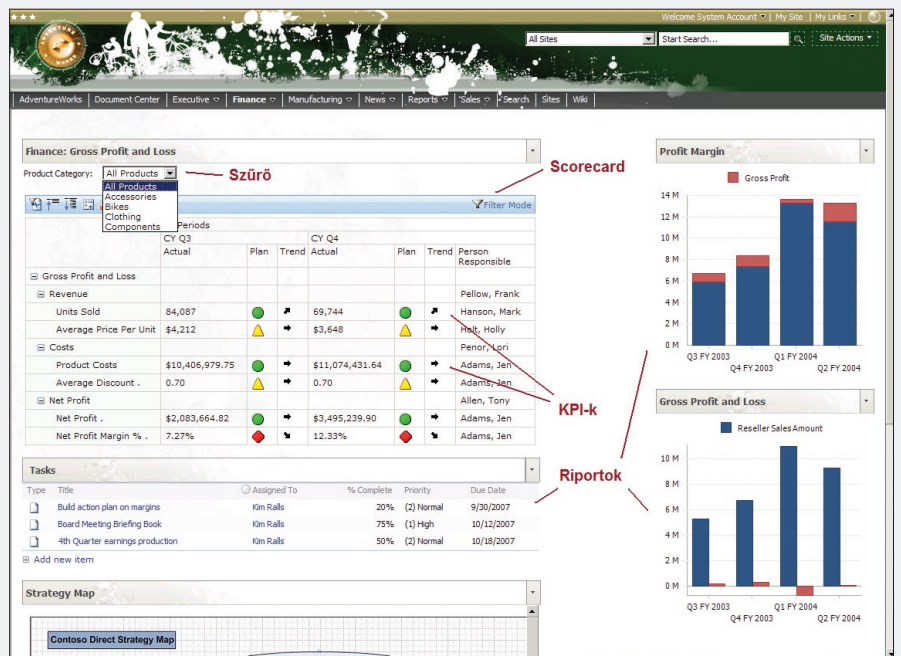
Tekintsük át most lépésről lépésre, hogyan is áll össze a teljesítménymonitorozási megoldás!

Először az adatforrásokat definiáljuk, ame-

Az adatforrások után a KPI-eket definiáljuk, majd a KPI-eket scorecardokhoz kapcsoljuk. Elkészítjük a stratégiai térképeket, és hozzákapcsoljuk azokat a megfelelő scorecardokhoz. Definiáljuk a jelentéseket is; ez abból áll, hogy az általunk választott platfor-

### Elemzés

A PerformancePoint Server számos lehetőséget kínál arra, hogy az adatainkat elemezzük. Az adatelemzéshez az SQL Server Analysis



8. ábra. SharePointra publikált PerformancePoint-irányítópult

mon – Excel Services, Proclarity, SQL Server Reporting Services – elkészítjük, telepítjük őket, és hivatkozunk rájuk. Ez alól kivételt ké-

Servicesben létrehozott OLAP-kockákat használhatjuk, amelyekhez az irányítópultokon publikált interaktív PerformancePoint-

tablázatokkal, grafikonokkal és ProClarity, valamint Excel Services-jelentéseken keresztül férhetünk hozzá. Az elemzések alapját tehát az OLAP kockákon definiált különböző típusú riportok szolgáltatják. Természetesen az elemzési funkcionalitás nagyban függ at-

Mik is ezek a problémák? Elsősorban az okozza a gondot, hogy a tervezéshez nagyon sok különböző információra van szükség, ezeket be kell szerezni, rendszerezni kell, számításokat kell végezni rajtuk, majd többször át kell dolgozni, végül pedig megfelelő formában közzé is kell tenni őket.

A mai gyakorlatban ez tipikusan Excelben történik, mivel az Excel nagyon sok segítséget ad a munkához. Az Excelre épülő megoldásoknak van viszont néhány komoly hátulütőjük: nem elég biztonságos, könnyen módosítható, a változások nehezen követhetők, egyszerre többen nem szerkeszthetik ugyanazt az állományt, és a munkafüzetek megbízható tárolásáról, verziókezeléséről, valamint a tervezési folyamatok ve-

zérítéséről is külön kell gondoskodnunk.

A PerformancePoint Server ezzel szemben az adatokat egy központi adatbázisban tárolja, miközben a felhasználók Excel-felületen dolgoznak, és az Excel teljes funkcionalitását kihasználhatják a tervezés során. A tervezés nem ad hoc módon, hanem szabályozottan zajlik, munkafolyamatokat alakíthatunk ki feladatok előírásával, jó-

vághagyással és részletesen szabályozott jogosultsági rendszerrel. A tervek, előrejelzések állapotát pedig a monitoring-komponens irányítópultjain követhetjük nyomon. Mindezek mellett, a tervezési modellek kialakítását és a külső adatok integrálását is nagymértékben egyszerűsíti, így a projektek átfutási ideje is jelentősen csökkenthető.

Ha egy mondatban kell összefoglalni a PerformancePoint szerver legfontosabb előnyeit a hasonló megoldásokkal szemben, akkor azt emelhetjük ki, hogy a PPS a teljes tervezési folyamatot átfogja, annak minden elemére jó megoldást ad, és megszűnteti azokat az anomáliákat, amelyek nehézkessé, lassúvá és megbízhatatlanná teszik a tervezést.

Ezek után tekintjük át, hogy miről áll egy PerformancePoint tervezési megoldás!

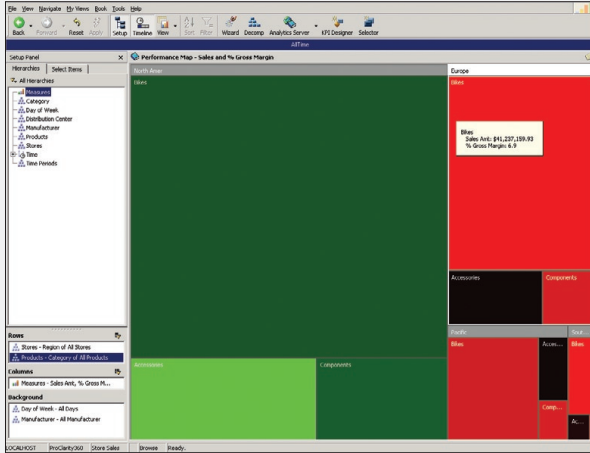
A PerformancePoint Server tervezési megoldását alkalmazások, modellek, a modellek közötti adatátadást meghatározó asszociációk, a tervezési folyamatot támogató tervezési ciklusok, a tervezés során kitöltendő űrlapok, a jogosultsági rendszer kialakítása és a tervezéshez szükséges adatok kezelése alkotja.

## Alkalmazások

A PerformancePoint Server alkalmazás nem más, mint az egymáshoz szorosan kapcsolódó dimenziók és modellek gyűjteménye. Az alkalmazáson belül több úgynevezett model site alakítható ki, amely a dimenziók és modellek további strukturálását teszi lehetővé. Fizikailag minden alkalmazáshoz tartozik egy relációs alkalmazás-adatbázis, egy, a számításokat és összesítéseket támogató OLAP-adatbázis, valamint egy betöltést segítő relációs átmeneti adatbázis. Az alkalmazásokhoz és site-okhoz külön jogosultsági rendszert építhetünk fel, és ezeken belül definiáljuk a tervezési ciklusokat is.

## Dimenziók és modellek

A tervezési modellek kialakítása előtt magát a tervezési folyamatot kell áttekintenünk. Egy kereskedelmi tervezés például indulhat azzal, hogy meghatározzuk az egyes termék-



9. ábra. Adatelemzés ProClarityvel

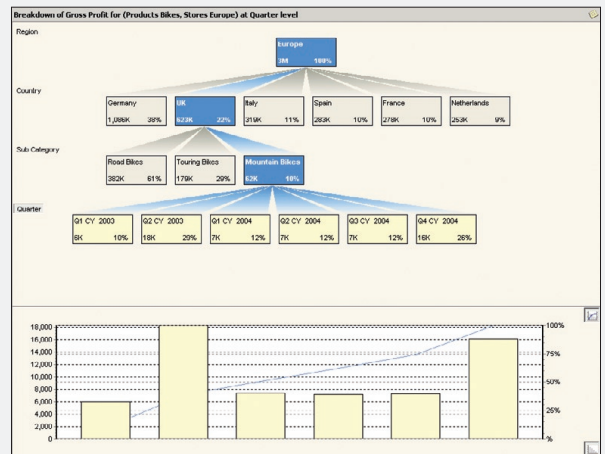
től, hogyan alakítjuk ki az elemzések alapját képező OLAP-kockákat, ezzel a magazin egy másik cikkében foglalkozunk.

Nézzünk egy szemléletes példát a lehetséges elemzésekre!

A 9. ábrán egy ProClarity-teljesítményterképet látunk, amelyen az egyes régiók és termékcsoportok értékesítési adatai láthatók, a területek nagysága az eladott áruk összértékét, a színek pedig a profitot reprezentálják, a nagyobb piros területek arra figyelmeztetnek, hogy az adott termékcsoportból nagyobb értékben, de alacsony profittal értékesítünk. A jelentést a ProClarity Web Professional alkalmazásban megnyitva egy dekompozíciós fát tudunk készíteni az adott termékcsoportból az értékesítés földrajzi megoszlása szerint. A 10. ábrán a profit alakulásának bontását látjuk egy dekompozíciós fával szemléltetve.

## Tervezés

Az üzleti tervezés a PerformancePoint Server legösszetettebb szcenáriója, mivel a tervezési modellek megalkotása, a szükséges számítások, előrejelzések definiálása, az űrlapok, folyamatok kialakítása igen munkáigényes. Mégis megéri azonban ezzel foglalkozni, mert a PerformancePoint Server sok olyan problémát megold, amely az üzleti tervezésben jelentkezik, és komoly kihívást jelent a területtel foglalkozó szakembereknek.

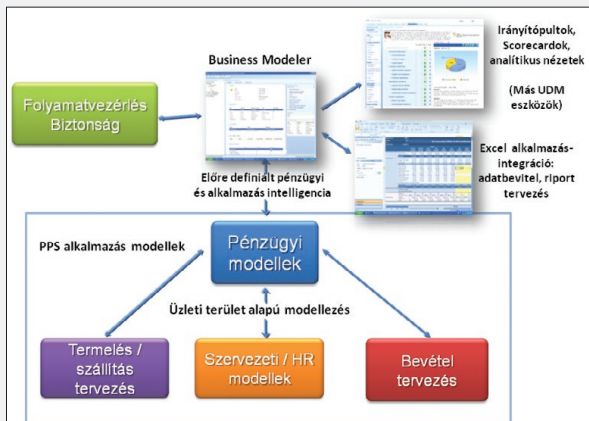


10. ábra. A profit elemzése dekompozíciós fá alkalmazásával

csoportok és kereskedelmi régiók elvart bevételét és profitját, majd ezeket a felső szintű terveket régióként és termékcsoportonként tovább finomítjuk, lebontva az irányszámokat konkrét termékekre, kereskedőkre, ügyfelekre. Ezután az így kialakított terveket viz-

zacsatolhatjuk, összehasonlíthatjuk a felső szintű tervekkel, majd véglegesítjük őket. A folyamat azonban itt nem áll meg, hiszen év közben szükségünk van arra, hogy a terveket

ügyfél (például szegmensekre bontva), az idő (év, negyedév, hónap). A PerformancePoint Server számos előre definiált dimenzióval rendelkezik, amelyek megkönnyítik a model-



11. ábra. A PerformancePoint tervezési megoldása

össze tudjuk hasonlítani az aktuális teljesítménnyel, előrejelzéseket tudunk készíteni, és bizonyos esetekben a tervek módosítására is sor kerülhet.

A tervezés alapja az, hogy megfelelő tervezési modellel, modellekkel rendelkezünk. A PerformancePoint Serverben a tervezési modellek kialakítása felhasználói felületen keresztül történik, míg az implementáció SQL Server relációs és OLAP-adatbázisokban valósul meg. Egy tervezési modell tulajdonképpen nem más, mint egy OLAP-kocka a hozzá tartozó relációs adattáblákkal. A tervezési modellek kialakítása során a modellek struktúráit az OLAP-kockákhoz hasonlóan dimenziókból és mértékekből állítjuk elő, definiáljuk a szükséges számításokat és a modellek közötti összefüggéseket. A felhasználói felületen összeállított modellek a PerformancePoint Server Planning adatbázisában tárolódnak. Az elkészült modelleket a felhasználói felületről telepíthetjük az SQL- és az OLAP-kiszolgálóra.

Egy kereskedelmi tervezés során tervezhetjük a bevételeinket, a költségeinket és profittunkat termékcsoportokra, termékekre, értékesítési csoportokra, értékesítőkre, régiókra, ügyfélszegmensekre, konkrét ügyfelekre nézve. Terveink lehetnek éves, negyedéves, havi bontásúak.

Az egyes tervezési szempontokból dimenziókat építünk, ilyen dimenzió lehet a termék (például termékcsoportokra bontva), a kereskedelmi régió, az értékesítési csoport, az

előre definiált dimenzióval rendelkezik, amelyek megkönnyítik a modellek kialakítását, mivel az előre definiált dimenzióhoz előre definiált üzleti logika is tartozik. Ha például egy pénzügyi tervezési modelt akarunk kialakítani, a beépített dimenziókból és az ezekhez kapcsolódó előre definiált számításokból a modellünk jelentés részét fel tudjuk építeni. A beépített funkcionalitásra jó példa az átváltási árfolyamok kezelése, a főkönyvi számlastruktúra támogatása, az időbeni összesítések,

halmozott értékek számítása vagy a cégek közötti számlamegzások konszolidációja.

A dimenziók létrehozása és adatokkal történő feltöltése a Business Modelerben történik. Egy-egy dimenzióhoz több különböző hierarchiát adhatunk meg, így az egyes modellekben másképp csoportosíthatjuk a dimenzióknak adatait. Tipikus példa erre a főkönyvi struktúra különböző jelentésekhez, kimutatásokhoz alkalmazott eltérő csoportosítása. A dimenziók a későbbiek során tetszőlegesen bővíthetők, és külső adatforrásból, csv-állományból vagy adatbázisból is feltölthetők.

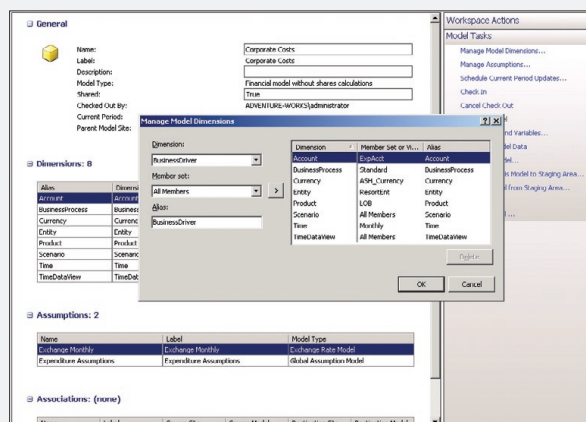
A dimenziókat felhasználva kialakítjuk az egyes modelleket. Egy tervezési modell dimenziókból, tényadatokból és a modellel definiált számításokból áll. A modell kialakítása úgy történik, hogy kiválasztjuk a modell-

hez szükséges dimenziókat, majd definiáljuk a számításokat. Számításokra azért van szükség, mert a tervezés során számos adat, mutatószám meglévő adatokból jön létre, és célszerű ezeket a modellünkbe befoglalni, ezáltal egyszerűsíthetjük és gyorsíthatjuk a riportolást és az elemzések készítését. Tipikus példa erre a különböző eltérések például terv-tény különbségének meghatározása vagy az átváltási árfolyamok alapján az értékek konverválása.

Ha a tervezési folyamatunk egyszerű, akkor akár egyetlen modell kialakítása is elegendő lehet, ha viszont egy fentebb leírt többszintű tervezési folyamatot kell implementálnunk, akkor több modellt érdemes építeni. Ilyenkor gondoskodnunk kell a modellek közötti adatátadásról is, amelyet a PerformancePoint Server deklaratív módon, adatelemek közötti kapcsolatok, asszociációk segítségével támogat, így nincs szükség programozásra vagy külső ETL-eszközök alkalmazására. A modelleket egymásba is ágyazhatjuk, így a bonyolultabb modellek kisebb, újrafelhasználható elemekből is felépíthetők. Erre jó példa az átváltási árfolyamok kezelésére szolgáló, előre definiált struktúrájú Exchange Rate Model, amelyet egyszerűen felhasználhatunk a pénzügyi modelljeinkben. A PerformancePoint Server több előre definiált modellt is rendelkezésünkre bocsát, ezek elsősorban a pénzügyi modellezést, elemzést, konszolidációt támogatják.

## Tervezési űrlapok

A tervezési űrlapokat Excelben készítjük el a PerformancePoint Excel Add-in segítségével. Az űrlapok kialakítása a megfelelő dimenziók és szűrőfeltételek kiválasztásával és beállításával történik. Az űrlapok létrehozása és használata közben élhetünk az Excel formázási lehetőségeivel, a számításokhoz pedig az Excel függvényeit is igénybe vehetjük. A kész űrlapokat a PerformancePoint Serverre publikáljuk, a felhasználók pedig a tervezési ciklusoknak megfelelően használhatják azokat. A tervezés során a kitöltött űrlapok adatait az Excel felölti az alkalmazás adatbá-



12. ábra. Tervezési modell kialakítása a Planning Business Modelerben



zásába, így azok a többi felhasználó számára is elérhetővé válnak. Lehetőség van az űrlapok offline kitöltésére is, ilyenkor az adatok a lokális gépen tárolódnak, majd a szervertől történő kapcsolódás után a felhasználó kezdeményezheti azok adatbázisba írását.

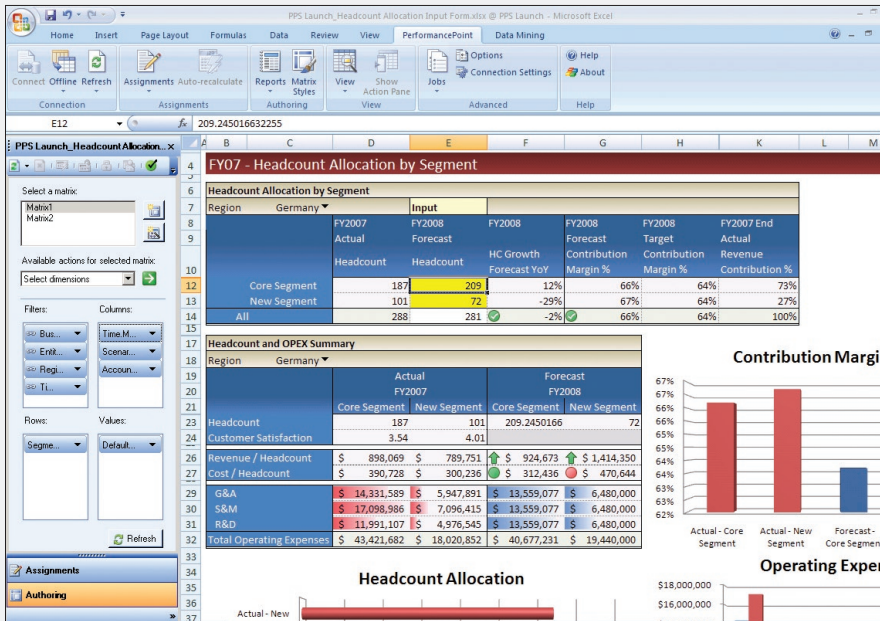
Az űrlapok kitöltésénél lehetőség van alakításra is, azaz egy-egy adat szétoztására

írhatjuk, hogy az adott ciklusban az adatokat melyik időperiódusra vonatkozóan módosíthatják a tervezők.

Például, ha a következő év terveit október 1. és november 30. között kell elkészíteniük bizonyos felhasználóknak, és ehhez ki kell tölteniük a szükséges űrlapokat, a terveket pedig egy másik felhasználó hagyja jóvá, ak-

tokra, mint például a főkönyvi számlastruktúra, az ügyféltörzs vagy az aktuális értékesítés, számlázás adataira.

A külső adatok integrációját egy átmeneti adatbázis és egy szinkronizációs megoldás támogatja, amely az átmeneti adatbázisba betöltött dimenzió- és tényadatokat szinkronizálja az alkalmazás-adatbázissal. Ez nagy könnyebbséget jelent az adatbetöltő interfészek kialakításánál, mivel a szinkronizáció során az adatok közötti összefüggések is ellenőrződnek, így minimalizálhatjuk a hibás adatbetöltésből eredő problémákat. A másik nagy előny, hogy az átmeneti adatbázist szükség esetén újra előállíthatjuk és szinkronizálhatjuk az alkalmazás-adatbázissal, így a betöltési folyamat bármikor újratekinthető. Ehhez egy parancssoros eszköz, a ppscmd.exe is rendelkezésre áll, így a folyamat szkriptelhető, automatizálható.



13. ábra. Tervezési űrlap az Excelben

egy meghatározott dimenzió tagjai mentén egyenletesen, vagy már meglévő értékek százalékos arányában. Igen kényelmes funkció az űrlapok újraszámításának ki-be kapcsolhatósága, ami különösen jól jön, ha bonyolult, sok számítást igénylő modellekkel dolgozunk.

### Tervezési ciklusok

A tervezési ciklusok annak a meghatározására szolgálnak, hogy kinek, mikor, milyen feladatai vannak a tervezés során. A tervezési ciklus tulajdonképpen egy olyan speciális munkafolyamat, amellyel előírjuk és biztosítjuk, hogy a felhasználók határidőre kitöltsék a megfelelő adatbeviteli űrlapokat, és megmondhatjuk azt is, hogy kinek kell azokat jóváhagynia, véglegesítenie.

A PerformancePoint Serverben tervezési űrlapot csak azok a felhasználók tölthetnek ki, akiket a tervezési ciklusban erre kijelöltünk, és kizárólag a tervezési ciklus futamideje alatt dolgozhatnak az űrlapokkal. A tervezési ciklus definiálása során azt is elő-

kor erre egy tervezési ciklust definiálunk. Bonyolultabb esetekben természetesen több tervezési ciklusunk lesz, amelyekhez több kitöltendő űrlap, határidő és felhasználó kapcsolódik.

### Adatintegráció

A tervezéshez gyakran van szükségünk más rendszerekben már rendelkezésre álló ada-

### Összefoglalás és további információk

Ha röviden össze akarjuk foglalni az eddigieket, azt mondhatjuk, hogy a PerformancePoint Serverrel az egyszerű scorecard-megoldásoktól kezdve a bonyolultabb többszintű pénzügyi tervezésig igen sokféle teljesítmény-menedzsment-alkalmazást ki lehet alakítani. Ha szeretnénk részletesebben megismerni, és lépésről lépésre végigkövetni egy monitorozási vagy egy tervezési megoldás kialakítását, akkor az alábbi hivatkozáson, nyilvánosan elérhető online tréninget javasoljuk:

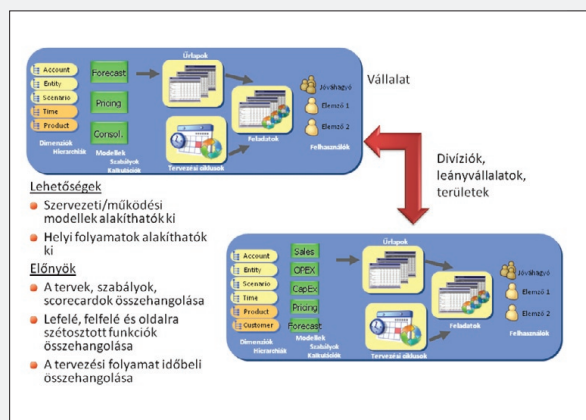
<http://www.microsoft.com/business/PerformancePoint/resources/training.aspx>

Több könyv is megjelent a témában, ezek közül a kezdőknek a Rational Press két kiadványát érdemes elolvasniuk, amelyek az alábbi címeken jelentek meg: Planning with Microsoft

Office PerformancePoint Server 2007 és Monitoring and Analyzing with Microsoft Office PerformancePoint Server 2007.

További hasznos információk a termékkel és a Microsoft üzletiintelligencia-megoldásaival kapcsolatban a <http://www.microsoft.com/hun/bi/> oldalon található.

Kovács Zoltán  
(zoltanko@microsoft.com)  
Microsoft Magyarország



14. ábra. A tervezési folyamat a PerformancePointban

# Üzleti Intelligencia Microsoft platformon az IQSYS-től

Az IQSYS és elődcegei a '90-es évek eleje óta foglalkoznak az ügyfelek jelentéskészítési feladatainak informatikai támogatásával. A megoldott problémák a banki kötelező jelentéskészítéstől (például PSZÁF, MNB) az IFRS, Basel II és egyéb fedezeti számításokon át a legkülönbözőbb tulajdonosi riportokig terjednek. Szöllőssy Gyula, az IQSYS Microsoft technológiákkal foglalkozó kompetencia központjának vezetője elmondta, hogy köszönhetően a Microsoft innovatív, új termékeinek (Excel 2007, SQL Server 2005, .NET 3.x, WPF) lehetővé vált, hogy a cégben felhalmozódott többéves jelentéskészítési tapasztalatot átültessék egy erre a platformra készült termékbe. Az MS Casir az adattárház építési, jelentéskészítési és elemzési problémák támogatására kínál egy kedvező, kis- és középvállalatok által is megfizethető licenz költségű megoldást.

Az MS Casir működésének lényege, hogy

összegyűjti az elemzésekhez, számításokhoz illetve a jelentésekhez szükséges adatokat a forrásrendszerekből és a különböző nyilvántartásokból egy adattárház építési elveken alapuló rendszerbe. Az adatok ellenőrzését, integrációját és a szükséges kalkulációkat a feldolgozási folyamatok keretében a rendszer elvégzi, továbbá a jelentések előállításához és az elemzések támogatásához OLAP kockákat készít. Az így előállt adatokon a jogosult felhasználók különféle elemzéseket végezhetnek, illetve a standard, kötött szerkezetű jelentések automatikusan állnak elő. Az MS Casir tulajdonképpen egy Microsoft platformra készült keretrendszer, amely támogatást, fogalmi és technikai keretet nyújt ezen adatok átlátható, auditálható végrehajtására, mondta Szöllőssy Gyula.

Az IQSYS a megoldást több ügyfelénél sikeresen bevezette, az egészen kis szer-



**Szöllőssy Gyula, Microsoft kompetencia központ vezető, IQSYS Zrt.**

vezetektől a nagyokig. Például az ElectroCoord Magyarország Kht-nél a tulajdonosi, a kontrolling és a VPOP felé kötelezően átadandó jelentések előállítására használják, az MNB-nél a Fizetési Mérleg Feldolgozó Rendszert, míg a Provident Pénzügyi Zrt-nél egy, az ügyfelek és tranzakcióik elemzésére használt rendszert valósítottak meg a segítségével. A megoldás mindhárom környezetben teljesítette az eredeti célkitűzéseket, illetve további ügyfeleknél jelenleg is folyamatban van a bevezetése. (X)

## Átfogó

# BI megoldások

- Adattárház és adatpiac építés
- Analitikus alkalmazások
- Pénzügyi alkalmazások
- Vállalati tervezés és teljesítmény monitorozás (CPB, CPM)
- Adatminőség menedzsment
- Törzsdát kezelés
- Adatbányászat

**IQSYS**  
A MAGYAR TELEKOM CSOPORT TAGJA

IQSYS Zrt. • H-1135 Budapest, Hun u. 2. • Telefon: (1)-236-6400  
Fax: (1)-236-6464 • info@iqsys.hu • www.iqsys.hu

PLATFORMFÜGGETLEN  
ÜZLETI  
INTELLIGENCIA



TELJES KÖRŰ BI SZOLGÁLTATÁSOK

# VIRTUÁLIS MINDENFÉLE

Szalad az idő – a virtualizációs megoldások területén kiváltképp.

Néhány hónappal ezelőtt, amikor virtualizációs megoldásaink palettáját mutattuk be, még sok helyen jövő időt kellett használnunk. Mára már alig maradt ilyen terület, sőt lassan látszanak a következő generációs megoldások körvonalai is.

## Helyreigazítás

Igazából nem is egy kiigazítandó hibáról van szó, csupán arról, hogy már a júniusi lapszámban megjelent cikk felett is eljárt az idő. A napokban megérkezett ugyanis a Microsoft Hyper-V Server 2008, ami egy kicsivel tovább árnyalja a szervervirtualizációs megoldások palettáját. Így az akkori cikkben ismertetett forgatókönyvet is.

Elsőként azt vegyük észre a néven, hogy ez nem Windows. Vagy mégis? Tulajdonképpen igen is meg nem is. Nevében semmiképpen sem az: tudatosan kihagyták belőle a Windows Serverre utaló megnevezést ezzel is hangsúlyozva, hogy itt egy olyan önálló célmegoldással van dolgunk, amelyik nem rendelkezik a széles funkcionalitással, mint a teljes értékű Windows Serverek. Másfelől igen, mind a telepítési folyamat, mind az alapvető beállítások módja tipikusan windowsos (leginkább persze a Server Core beállításaira hasonlít).

Hová tegyük tehát a Hyper-V Servert a Hyper-V szerepkört nyújtó szerverválatatok sorában? Tegyük mindjárt a legelejére, ott a helye. A Windows Server 2008 Standard változattól ugyanis mindössze két és fél dologban tér el.

Az első – és a legfontosabb –, hogy lényegesen szűkebb funkciókat nyújt a szerepkörököt illetően, mint a legkisebb Windows Server (amiben van Hyper-V). A szerepkörök és képességek rövidke listájáról még lesz szó a későbbiekben. A másik fontos különbség a licenclés módja: a Hyper-V Server ingyenes termék, szabadon hozzájuthat bárki az interneten keresztül. A rajta futó virtuális gépekhez viszont megfelelő egyedi licencekkel kell rendelkeznie a felhasználónak.

Csak egy zárójeles megjegyzés: igen, eleinte 30 dollár körüli árról volt szó, de azóta bebizonyosodott, Steve Ballmernek igaza volt, amikor azt mondta febr-

ruár végén, hogy a Hyper-V demokratizálni fogja a szervervirtualizációs piacot. Ma már van ingyenes ESX Server (VMware ESXi), és van ingyenes Hyper-V Server, lehet tehát válogatni a hypervisor alapú megoldások között, bár a választást némiképp egyszerűsíti, hogy használható és ingyenes felügyeleti konzol csak a Hyper-V-hez van, amivel akár egy Vista SP1 kliensről is irányíthatjuk valamennyi Hyper-V-t futtató kiszolgálónkat. (A részletes konfigurálási lépéseket megtaláljuk John Howard blogjában a 2008. áprilisi cikkek között: <http://blogs.technet.com/jhoward>.)

A két eltérés megvolt, mi a fél? A grafikus felület. A Windows Server esetében mindig van választási lehetőségünk, hogy teljes grafikus vagy Server Core telepítési opciót válasszunk. A Hyper-V Server esetén a választási lehetőség nincs meg: csak parancssori változatot telepíthetünk.

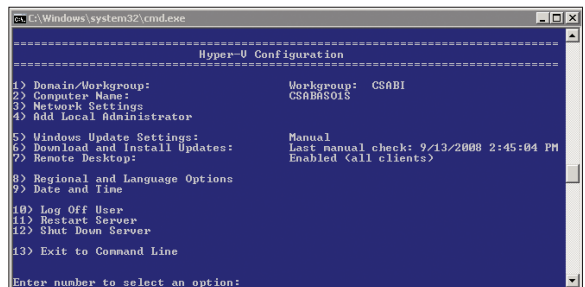
Egyéb képességekben a Hyper-V Server a Windows Server Standard változatára épül, így ugyanazok a hardvermegkötések vonatkoznak rá, ha a nagyobb verziókhoz hasonlítjuk: legfeljebb 4 fizikai processzorral, maximum 32 gigabájt memóriával dolgozik, nem támogatja a fürtözést, és így a magas rendelkezésre állású virtuális gépeket sem.

A telepítési folyamat ugyanaz, mint a Windows Serverek esetén – csak a verzió kiválasztása és a termékkulcs beírása hiányzik, mert ilyenek nincsenek. A rendszergazdai jelszó beállítása és az első bejelentkezés után egyszerű kis konzol fogad bennünket, amellyel a legfontosabb beállításokat véghezvesszük.

Meg kell mondjam, szívesen látnék egy hasonló beállítási menüt a Server Core-on is, sok időt lehetne megtakarítani vele. A Hyper-V szerepkör alából telepítve van már, nem kell tehát külön bekapcsolni, a hardveres előfeltételeknek viszont itt is meg kell felelnünk (Intel-VT- vagy AMD-V-támogatás a procesz-

Virtualization Needs	Microsoft Hyper-V Server 2008	Windows Server 2008 Standard	Windows Server 2008 Enterprise	Windows Server 2008 Datacenter
Server Consolidation	✓	✓	✓	✓
Test and Development	✓	✓	✓	✓
Mixed OS Virtualization (Linux and Windows)	✓	✓	✓	✓
Local Graphical User Interface		✓	✓	✓
High Availability—Clustering			✓	✓
Quick Migration			✓	✓
Large Memory Support (Host OS) > 32 GB RAM			✓	✓
Support for > 4 Processors (Host OS)			✓	✓
Ability to Add Additional Server Roles		✓	✓	✓
Virtualization Rights per Server License	Each VM Guest requires a server license	1 Physical + 1 VM	1 Physical + 4 VMs	1 Physical + Unlimited VMs

1. ábra. A Hyper-V Server és a Windows Serverek összehasonlítása



2. ábra. A Hyper-V Server főmenüje

szorban, hardveres Data Execution Prevention-támogatás a BIOS-ban). Mi a helyzet viszont a további szerepkörökkel? Nos, szerepkörből csak egy van, ez a Hyper-V, és a képességek listája is rövid, hiszen ez nem egy Windows Server. Ragadjuk ki csak a legfontosabbakat ezek közül.

Biztonsági szempontból fontos, hogy a Hyper-V Server merevlemezei is védhető BitLocker technológiával, ami létfontosságú lehet például telephelyi elhelyezés esetén. A kiszolgálónk lehet iSCSi-kliens, ehhez konfigurálhatunk többutas elérést a tárolórendszer felé (MultiPathIO), és beléphetünk egy terheléelosztási fűrtbe is (NLBS). Mentéshez pedig ugyanazokat a parancsokat használhatjuk, mint a Windows Serveren (ott is sokkal többre megyünk ezekkel, mint az egyszerűsített grafikus felülettel).

A termék frissítései természetesen a Windows Update-en keresztül érkeznek, választásunk szerint automatikusan vagy kézi frissítéssel. Kézi frissítés esetén még követhetjük is a folyamatot, ami kifejezetten hasznos, hogy lássuk: szükséges-e újraindítani a rendszert.

A Server Core-on használt parancsok döntő többsége megél ebben a környezetben is, a tűzfalat (igen, van és aktív!) például a netsh megfelelő használatával nyitogathatjuk ki, hogy a gépet grafikus eszközökkel felügyelhessük távolról.

Hogyan árnyalja az új változat az előző cikkben leírt forgatókönyveket, mik lehetnek a Hyper-V Server tipikus alkalmazási módjai?

Általánosságban jó választás lehet olyan esetekben, amikor a magas rendelkezésre állás nem követelmény, a virtuális gépekhez megfelelő licencekkel rendelkezünk, és nem is tervezünk további virtuális gépet létrehozni. Ha akár egyetlen további Windows Servert is szeretnénk futtatni virtuális gépként, amihez még nincsen licencünk, akkor már jobban megéri egy Windows Server Standard vagy magasabb változat (érdemes

visszatérni az első ábrán látható táblázat utolsó sorához, abból látszik, hogy miért).

Nézzünk néhány példát! Lehetnek például kiöregedett kiszolgálóink, amelyeken üzleti leg kritikus alkalmazások futnak, amelye-

meztakarítani, ott is vannak olyan szerepek, amelyek alacsony kihasználtsággal futnak önálló fizikai gépeken, és a Hyper-V Serveren keresztül ráadásul ezeket is egy egységes felügyeleti platform alá integrálhatjuk. De ez már átvezet bennünket a virtuális rendszergazdák világába...

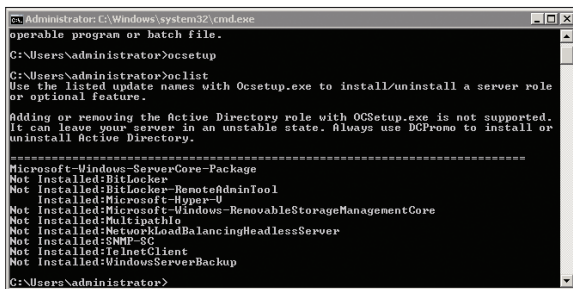
## Virtuális rendszergazda – fogalomzavar vagy jövőkép?

Az alcím talán egy kissé provokatív, de gondoljunk bele: a felügyeleti rendszereinkben alkalmazott valamennyi automatizmus éppen azokat az unalmas, ismétlődő feladatokat veszi le a vállunkról, amelyeket egyébként igyekszünk elkerülni és gyakornokok vagy kezdő kollégák nyakába varrni. Nincs ebben semmi szégyellnivaló, ilyen az emberi psziché: az új, ismeretlen feladat az vonzó a számunkra, érdeklí az agyunkat, és szívesen elvégezzük. Az unalmas, már ismert feladatokat pedig továbbadjuk. Csak gondoljunk arra, hogy a gyakornok is ember. Vagy mégsem? Talán éppen a virtualizáció lehet az a fontos eszköz, ami hosszabb távon kiszabadíthat bennünket a rutinfeladatok csapdájából. Az eszközök – úgy tűnik – lassan előállnak, most már a kreatív hozzáálláson és a lehetőségek legjobb kihasználásán múlik, hogy a robotos munkát valóban robotokra bizzuk.

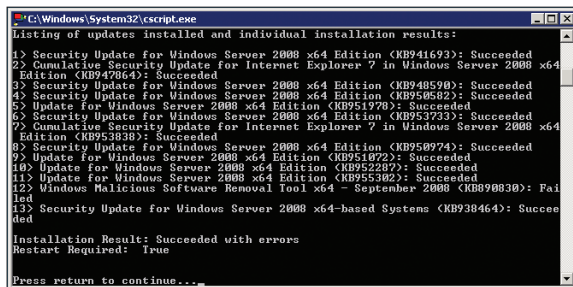
A cél elérésének két kulcsszereplője a System Center Operations Manager 2007 (SP1) és a Virtual Machine Manager 2008 lesz. A két alkalmazás finom összehangolása nagy lépést jelent előre a szabályokra épülő informatikai környezet irányába, ami egyúttal azt is jelenti, hogy a megfelelő szabályok létrehozásával csökkenthetjük az emberekre nehezedő rutinfeladatok nyomását.

Azt javaslom, induljunk egy képzeletbeli sétára, egy képzeletbeli számítóközpontban, és nézzük, hol találunk olyan feladatokat, amelyeket egy virtuális rendszergazdára bízhatunk. Az egyszerűség kedvéért foglalkozunk csak a kiszolgálókkal, és tekintsük adottaknak a hálózati szolgáltatásokat (bár ott is adódnak lehetőségek automatizációra), továbbá feltételezzük azt, hogy a szolgáltatások nagymértékben virtuális gépeken futnak.

Induljunk a kézzelfogható eszközök irányából, kezdjük a fizikai hardverrel! Ma már minden komolyabb kiszolgáló rendelkezik olyan felülettel, amelyen keresztül felügyelhető és irányítható a hardver, így akár távol-



3. ábra. A Hyper-V Server telepítése



### Kézi frissítés a Windows Update-ről

ket nem tudunk újratelepíteni vagy frissíteni (nincs forrás, nincs szakember, nem fut az új operációs rendszereken, vagy egyszerűen nincs keret egy újabb verzió vásárlására). Ebben az esetben járható út lehet ezeknek a gépeknek a virtualizálása. A Hyper-V Server lehet például az ilyen gépek futtatási környezete, míg a korszerű alkalmazásokat és operációs rendszereiket futtathatjuk Windows Servereken, ami egyrészt nagyobb rugalmasságot jelent a több funkció miatt, másrészt gazdaságosabb a licencelés módja miatt.

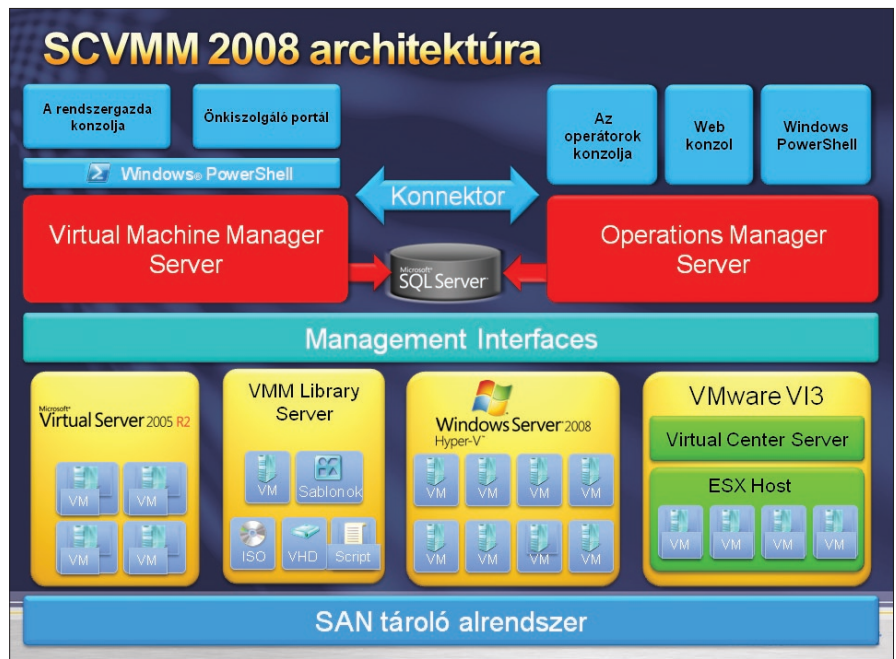
Építhetünk a Hyper-V Serverre tesztkörnyezetet, ahol akár az élesben használt virtuális gépeinket is megépíthetjük (nincsenek platformkorlátok, dolgozhatunk 64 bites virtuális gépekkel is), majd azokat átköltöztethetjük végleges helyükre, és élesíthetjük a gazdagép kulcsával.

Ha pedig tovább bővül a Linux alapú rendszerekhez tartozó integrációs komponensek köre (ahogy a tendencia már látszik a Xen, a SuSe és a Sun példáján), akkor akár ezeket a kiszolgálókat is virtualizálhatjuk Hyper-V Serverre. Miért? Mert energiát, felügyeleti ráfordítást azokon a gépeken is lehet

ról kapcsolhatjuk ki vagy be a kiszolgálóinkat, hogy csak a két leghétköznapibb műveletet említsük. Már a kiszolgálók telepítésekor is hasznát vehetjük ennek, ha például System Center Configuration Managerrel telepítjük a kiszolgálóinkat (is). Technikailag teljesen lehetséges az a forgatókönyv, amikor az SCOM észleli, hogy valamennyi virtuális gazdagépünk elérte azt a teljesítményhatárt, amikor már nem terhelhető tovább a felhasználók korlátozása nélkül. Kiad hát egy parancsot egy tartalék kiszolgálónak, a hardverfelügyeleti interfészen keresztül bekapcsolja, majd az SCCM-re bizza az operációs rendszer telepítését (amit az például a hálózati kártya egyedi címe alapján képes is megtenni). A telepített lemezkép már eleve tartalmazhatja a virtualizációs szerepkört és akár azt a kis scriptet is, ami automatikusan a megfelelő gazdacsoportba sorolja az új kiszolgálót. A SCOM – észlelve az új gazdagépet – most már akár automatikusan is képes lehet arra, hogy a leginkább erőforrásszűkében lévő virtuális gépeket átköltöztesse az új hardverre.

Futurisztikusan hangzik? Egyelőre igen, de a meglévő eszközökkel már megvalósítható. Ha nincs Configuration Managerünk, akkor használhatjuk a Windows Deployment Servicest, kiegészítve néhány konfigurációs XML állománnyal, ami a telepítést vezérli. Ebben az esetben igénybe kell vennünk valós gyakornokunkat, hogy a szükséges 5-6 gombnyomást tegye meg.

Valós gépeink hardvereinek és operációs rendszerének felügyeletét ismét csak az SCOM-ra bízhatjuk. A legtöbb hardvergyártó már biztosít olyan felügyeleti csomagokat, amelyekkel aprólékosan követhetjük a hardver „életét”, értesülhetünk a hibákról és a túlterhelésről. Ha pedig tudunk valamiről,



5. ábra. SAN tároló-alrendszer

akkor létrehozhatunk automatikus választokat. Így például egy processzor túlmelegedése esetén mondhatjuk az SCOM-nak, hogy ilyenkor néhány vagy minden virtuális gépet helyezzen át más kiszolgálóra. Ha kicsit merészebbek vagyunk, még azt is megvalósíthatjuk, hogy a szoftverfrissítések telepítése – illetve az amiatt szükséges újraindítás – váltson ki egy hasonló folyamatot, tehát minden virtuális gép költözzön el arra az időre, amíg a gazdagép újraindul. Hogy ez lehetetlen? Így a Live Migration első nyilvános bemutatása után azt mondanám: már nem sokáig.

Kicsit távolabbi tekintve: ha megoldódik a virtuális processzorok és a memória dinamikus kezelése, vagyis bármikor adhatunk és elvehetünk processzort és memóriát a virtuális gépeinkből, akkor az SCOM-ra esetleg

olyan döntést is bizhatunk, hogy éjszakára vonja össze a virtuális gépeket néhány kiszolgálóra, kapcsolja ki a feleslegessé vált fizikai gépeket, majd reggel térjen vissza a nagyobb teljesítményű állapothoz.

Gondolom, sokakban ott motoszkál már a kérdés, hogyan lehetséges ez? Hogyan tudunk ilyen együttműködést elérni a fizikai és virtuális környezet között? Először is: a System Center különböző felügyeleti megoldásai pontosan érzékelik, hogy mikor van dolguk fizikai és mikor virtuális rendszerrel, és ebben meglehetősen egyedülállóak a hasonló megoldások között. Az esetek jelentős részében a fizikai vagy virtuális környezetnek nincs jelentősége (például az egy-egy alkalmazásra vonatkozó eseménynapló-bejegyzések értelmezésénél), más esetekben viszont nagyon is: például, amikor egy virtuális gép azért teljesít gyengén, mert egy másik leköti az erőforrásokat. Ilyen esetekben jön a képbe az a speciális felügyeleti csomag, ami a Virtual Machine Managerrel érkezik és a PRO névre hallgat. A PRO itt csak vizuálisan utal a professionalizmusra, valójában betűszó (mi lenne más): Performance and Resource Optimization a feloldása.

Vessünk egy pillantást a Virtual Machine Manager architektúrájának ábrájára, és máris megértjük, hogyan épül fel a két alkalmazás kapcsolata!

Kicsit leegyszerűsítve: az SCVMM ismeri a

Management Packs (59)			
Look for:	Find Now	Clear	
Name	Version	▲	Sealed
Microsoft Virtualization Core Library	1.0.2627.0		Yes
Microsoft Virtual Server 2005 R2	1.0.2627.0		Yes
Microsoft System Center Virtual Machine Manager 2007	1.0.2627.0		Yes
Microsoft Virtualization Reports	1.0.2627.0		Yes
System Center Virtual Machine Manager PRO Library	2.0.3194.0		Yes
System Center Virtual Machine Manager PRO HyperV Host Performance	2.0.3194.0		Yes
System Center Virtual Machine Manager PRO VM Right Size	2.0.3194.0		Yes
Microsoft System Center Virtual Machine Manager 2008	2.0.3194.0		Yes
Microsoft Virtualization Reports 2008	2.0.3194.0		Yes
System Center Internal Library	6.0.6278.0		Yes

4. ábra. Virtuális gépek felügyeleti csomagjai az SCOM-ban

Name	Library Server	Type	Operating System	Owner	Status
120G DE VHD Template	CONTOSO-DC.contoso.local	Virtual Hard Disk	64-bit edition of Win...	CONTOSO\administr...	OK
Blank Disk - Large	CONTOSO-DC.contoso.local	Virtual Hard Disk	None	CSABASO15\Administr...	OK
Blank Disk - Small	CONTOSO-DC.contoso.local	Virtual Hard Disk	None	CSABASO15\Administr...	OK
EBS-RC1-MGMT	CONTOSO-DC.contoso.local	Virtual Machine	Unknown	Unknown	Stored
en_win_srv_2003_r2_standard_with_sp2_cd1...	CONTOSO-DC.contoso.local	ISO	Unknown	Unknown	OK
en_win_srv_2003_r2_standard_with_sp2_cd2...	CONTOSO-DC.contoso.local	ISO	Unknown	Unknown	OK
en_windows_server_2008_datacenter_enterpri...	CONTOSO-DC.contoso.local	ISO	Unknown	Unknown	OK
en_windows_server_2008_datacenter_enterpri...	CONTOSO-DC.contoso.local	ISO	Unknown	Unknown	OK
en_windows_vista_with_service_pack1_x86_d...	CONTOSO-DC.contoso.local	ISO	Unknown	Unknown	OK
en_windows_xp_professional_with_service_pac...	CONTOSO-DC.contoso.local	ISO	Unknown	Unknown	OK
SBS2008	VMMServer	Template	64-bit edition of Win...	CONTOSO\administr...	OK
Vista Base	CONTOSO-DC.contoso.local	Virtual Machine	Windows Vista	CONTOSO\administr...	Stored

6. ábra. Növekszik a könyvtár

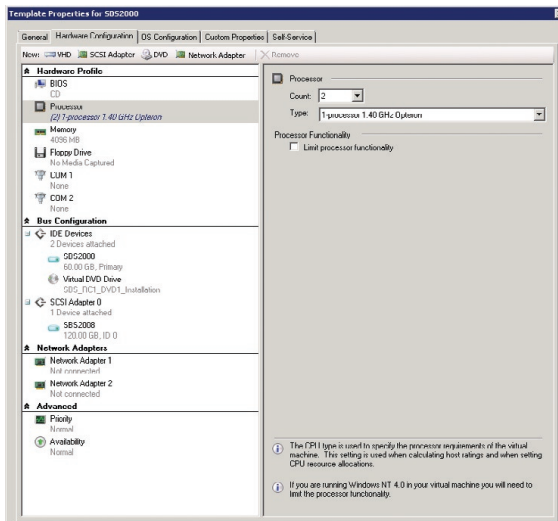
gazdagépeket, és tudja, melyiken milyen virtuális gépek futnak. Ezt a tudását megosztja az SCOM-mal, amelyik valamilyen eseményt értékelve figyelembe veszi ezeket az információkat. Például ha egy virtuális gépen futó alkalmazás elvárt válaszideje megnő, akkor képes megnézni a gazdagépet és az azon futó többi virtuális gépet, hogyők okozzák-e a lassulást. A két rendszer közös beszélt nyelve pedig a Windows Powershell, ami a grafikus konzolok alatt, azokon messze túlmutató funkcionalitást nyújt a hardver- és szoftverállapot lekérdezéséről, annak módosításáig.

Ha mindehhez hozzáteszük, hogy az SCVMM képes a VMware VI3 rendszerek irányítására is, az SCOM pedig hamarosan képes lesz nem Windows alapú rendszerek felügyeletére is, akkor hamarosan az automatizálási lehetőségek hatványozódásával számolhatunk.

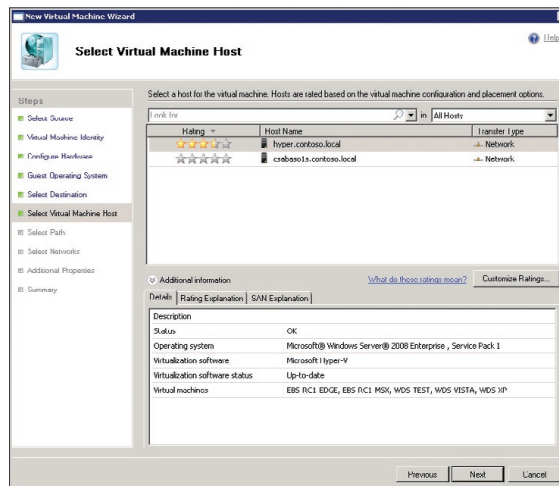
Ha még néhány gondolat erejéig visszatérünk a képzeletbeli számítóközpontunkhoz, akkor érdemes megnézni, milyen lehetőségeink vannak a virtuális gépek kezelésére. Ezen a területen a Virtual Machine Managerre támaszkodhatunk. Az alkalmazás egyik legfontosabb szolgáltatása a Library, vagyis a könyvtár. Ez igazából csak egy vagy több megosztás, ahol sokféle információt tárolhatunk: lehetnek itt ISO lemezek, sysprep-pel előkészített és üres virtuális diszkek, hardver- és szoftverkonfigurációs sablonok.

Ha úgy adódik, bármikor elővehetjük őket, és néhány kattintással virtuális gépet készíthetünk belőlük. A kombinációk száma jelentős: készíthetünk hardversablonok-

kat, amelyekkel különféle teljesítményű virtuális gépekhez juthatunk (variálhatjuk a processzorok számát, a memóriát, a hálózati



7. ábra. Virtuális gép sablonja



8. ábra. A virtuális gép kiválasztása

csatolókat és a diszkeket). A hardversablonokkal telepíthetünk virtuális gépet kézzel és automatikusan, ekkor egy XML-fájl kell

létrehozunk, ami a telepítést vezérli (ezért is szükséges a háttérben a Windows Automated Installation Kit). A sysprep-pel előkészített virtuális diszkekből is indulhatunk, akár kézi, akár automatizált telepítéssel. Az átmenetileg feleslegessé vált gépeinket pedig vissza is vonhatjuk a könyvtárba, amíg újra szükségünk nem lesz rá (esetleg egy másik gazdagépen).

Amikor egy új virtuális gépet elindítunk, az SCVMM javaslatot tesz arra, hogy melyik gazdagépre helyezük el azt. A számításhál figyelembe veszi az aktuális terheléelosztást és a gazdagépek kapacitását. Az eredményt egyszerű 5 csillagra épülő minősítésként kapjuk, néhány kiegészítő információ kíséretében (például hogy a gazdagép SAN-ra csatlakozik-e vagy sem).

A tárolórendszer valóban kulcsfontosságú a virtualizált környezetben. Azon túl, hogy magas rendelkezésre állású virtuális gépek létrehozásához nélkülözhetetlen, az SCVMM-nek is megvan az a képessége, hogy a SAN-on belül mozgasson virtuális diszket, ha a könyvtármegosztás és a virtuális diszkek szánt LUN ugyanazon a tárolórendszeren található (így tehermentesíthető a hálózat, nem is beszélve a sokkal nagyobb sebességről).

Ha visszatérünk a virtuális rendszergazda gondolatához, akkor azt látjuk, hogy az SCVMM minden művelete lefordítható Powershell-parancsokká, és ebből következően automatizálható. Rajtunk múlik, hogy meghatározzuk azokat az eseteket, amikor valamilyen automatikus választ várnunk a rendszereinktől, és hogy milyen döntési pontokban akarunk mi közbeavatkozni.

Talán rendhagyó ez a cikk abból a szempontból, hogy nem tételesen halad végig egy új termék funkcióin, de sokkal fontosabbnak éreztem, hogy felvillanjanak a hosszú távú lehetőségek, amelyek benne rejlének. Mert tudhat a technológia akármennyit, ténylegesen csak annyit ér, amennyit kihasználunk belőle...

Somogyi Csaba  
(Csaba.Somogyi@microsoft.com)  
Microsoft Magyarország

# it security today



## Az itbusiness napi informatikai biztonsági online tájékoztatója

Informatikai döntéshozóknak technológiai szakembereknek

Az elmúlt 24 óra legfontosabb hazai és külföldi informatikai biztonsági és információ biztonsági hírei

Ingyenes napi online hírlevél

**it security today**  
Az itbusiness napi informatikai biztonsági online tájékoztatója

2008. 01. 10. **KEDVES OLVASÓ RÉSZÉRE** Klikk a jobb oldali címsorban!

**TERMÉKÉRTÉKELÉSEK**  
A Security Nowveler által fejlesztett FastChannel és HyperScan olyan intelin alapú, gyorsan futó biztonsági skánerok, amelyek azonosítják a kártékony vírusokat, azonosítják a potenciális fenyegetéseket, és azonnal jeleznek a felhívásról. A skánerok a kártékony vírusokat azonosítják, és azonosítják a potenciális fenyegetéseket, és azonnal jeleznek a felhívásról.

**A NAGYVILÁGBÓL**  
A Spies VOP és egyéni...  
A Spies VOP és egyéni...  
A Spies VOP és egyéni...

**KONYVJÁRÓ**  
Harding: The Art of Exploitation, 2nd Edition...  
Harding: The Art of Exploitation, 2nd Edition...  
Harding: The Art of Exploitation, 2nd Edition...

**JÓVÓ HÉTEN REZSZEK**  
BCI Cél-Menedzsment Workshop...  
BCI Cél-Menedzsment Workshop...  
BCI Cél-Menedzsment Workshop...

**Regisztráljon!**  
[www.it-business.hu/hirlevel](http://www.it-business.hu/hirlevel)

**Regisztráljon!**  
[www.it-business.hu/hirlevel](http://www.it-business.hu/hirlevel)

2008. IT Business Publishing Rt. Főosztály: 1052 Budapest, Főv. Területi Hivatala, 1052 Budapest, Főv. Területi Hivatala. Képzés: A képzés célja az informatikai biztonsági tudás, készségek fejlesztése. A képzés során az résztvevők megismerik az informatikai biztonsági fogalmakat, és megismerik a legújabb technológiákat. A képzés során az résztvevők megismerik az informatikai biztonsági fogalmakat, és megismerik a legújabb technológiákat.

## Regisztráljon!

[www.itbusiness.hu/hirlevel](http://www.itbusiness.hu/hirlevel)

# GROUP POLICY PREFERENCES

Mi ez egyáltalán? Mikor először hallottam róla, valaki úgy jellemezte, hogy ez egy beállításgyűjtemény, ami olyan, mint a csoportházirend, de nem kötelező érvényű a felhasználóra. Rendben, de akkor mi értelme? Főleg, hogy a felhasználó megváltoztathatja ugyan a beállítást, de a policy újbóli alkalmazásával (például ki- és bejelentkezés) a beállítások újból érvényre jutnak. Nem láttam hasznát.

**H**osszabb ideig ennyiben is maradtunk a GP Preferences és én, míg nemrég egy ügyfél (az MVM-holding egyik tagvállalata) kívánságának kellett eleget tennem, nagyon rövid idő alatt. A kívánság pedig az volt, hogy több hálózati meghajtót kell felcsatolni a felhasználóknak automatikusan AD csoporttagság alapján.

Elsőre teljesen átlagos feladatnak tűnt, és mivel mindig is szerettem scriptet írni, most is rögtön a logon script mellett döntöttem. Rövid gondolkodás után azonban rájöttem, hogy ha rendesen meg szeretném írni a scriptet, akkor nem olyan egyszerű a feladat, mint az elsőre látszik. Mégpedig a fő gond a beágyazott csoportokkal lesz.

Ekkor jutott ismét eszembe a GP Preferences. Az MVMI Informatikánál nemrég tértünk át a Windows Server 2008-as operációs rendszerre a Microsoft Consulting Services-zel közösen végzett infrastruktúrafejlesztési projekt során. A 2008-as DC-ken elérhető újdonságok között ott a GP Preferences is, és emlékeztem, hogy abban van lehetőség a hálózati meghajtók csatlakoztatására is. Rövid nézelődés után (kb. 10 perc) össze is kattintgattam a megfelelő beállításokat, és mehetett „élesbe” a policy. Scriptelve biztosan nem végeztem volna ennyi idő alatt. Ráadásul egyik – scriptet írni nem szerető – kollégám is megörült, hogy így már könnyebben követheti ő is a logon-kor érvényre jutó beállításokat. Ezek után már mindenképpen mélyebben bele kellett nézmem a GP Preferences lehetőségeibe, hogy egy másik probléma megoldását is leegyszerűsítsem ilyen módon.

## Ismerős helyzet

Sokaknak ismerős lehet az a probléma, hogy a saplogon.ini konfigurációs fájl helye a Windows mappa. Ha több felhasználó különböző saplogon-beállításokkal dolgozik ugyanazon a gépen, akkor ezt a fájlt minden bejelentkezéskor felül kell írni vagy a bejegyzéseit módosítani. Erre a GP Preferencesben nagyon jó megoldás lehet az „ini files” opció, ahol megadhatjuk, hogy bizonyos feltételek esetén milyen bejegyzések kerüljenek az ini fájl egyes szekcióiba. Mindezt pedig úgy tudjuk beállítani, hogy nem kell a felhasználónak jogot adni a fájl módosítására, hiszen a

GP Preferences a rendszer nevében is módosíthatja a fájlt, nem csak a bejelentkezett felhasználó nevében.

Tovább bonyolítom az esetet egy nálunk előfordult problémával. Kihasználtuk ugyan is a Windows Server 2008 Terminal Server RemoteApp alkalmazás publikációs lehetőségeit, és az új 7.10 SAP GUI-t elérhetővé tettük néhány felhasználónak, hogy otthonról használhassák. Ezeknek a felhasználóknak az éles környezetben is és a terminálkörnyezetben is kellett dolgozniuk, más-más saplogon-beállításokkal. Erre adott kiváló megoldást GP Preferences „targeting” lehetősége, ahol a saplogon.ini bejegyzéseit be tudtam állítani arra az esetre is, ha az SAP GUI-t terminálkapcsolaton keresztül használják.

Nézzük akkor, milyen lehetőségeink vannak a GP preferences használatával.

Két csoportra oszlik a lehetőségek listája, az egyik a Windows beállításaié, ahol az általánosabb, szinte minden szervezetenél előforduló alapbeállítások kaptak helyet, a másik a Vezérlőpult beállításaié, ahol a vezérlőpulton helyet foglaló konfigurációs eszközök beállításait szabhatjuk testre.



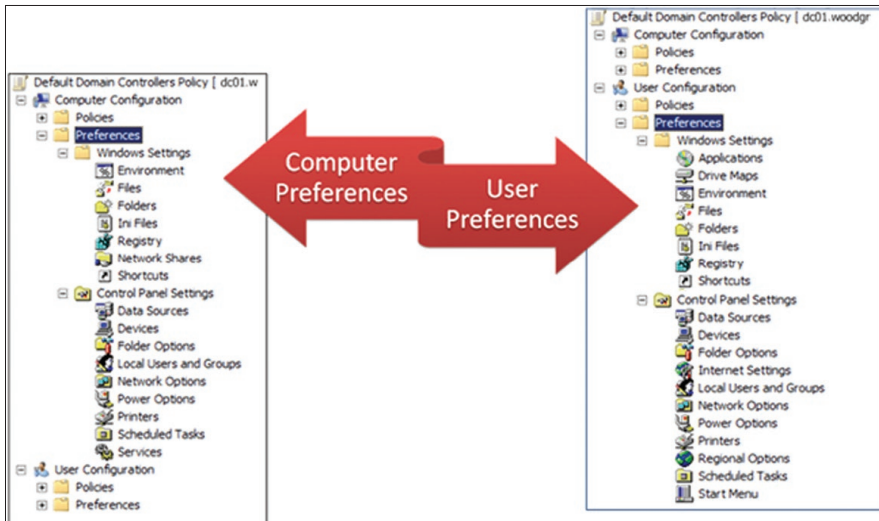
Az első csoportban található eszközökkel létrehozhatunk fájlokat, mappákat, környezeti változókat és registry-bejegyzéseket, módosíthatjuk vagy törölhetjük azokat az adott szervezeti egység vagy szoftver kiválmáinak megfelelően. Itt kapott helyet az „ini files” opció is, amellyel könnyen felhasználókat szabhatjuk például a saplogon.ini-t.

Érdekesebbek a Vezérlőpult beállításai,

„Internet settings” alatt beállíthatunk mindent, amit az Internet Explorer internetbeállítások dialógus ablakában megtehetünk.

Bármelyik lehetőségre kattintunk, megkapjuk a már bekonfigurált elemek listáját, ahol hozzáadhatunk, módosíthatunk vagy törölhetünk elemeket. A módosítások konfigurációs dialógusablakban történnek. Itt van egy speciálisan a kiválasztott elemre vonatko-

a házirend vagy a pozicionálás hatókörén. Az alapértelmezett esetben nem történik semmi, és az előzőleg beállítottak maradnak érvényben, de ha bepipáljuk, akkor – amint a kliensre már nem lesz érvényes a házirendünknek ez a része – visszaáll a beállítás előtti állapot. Ez az opció nem minden GP Preferences-lehetőség esetén érhető el (például az energiagazdálkodási lehetőségek esetén nem). A végén még megadhatjuk: akarjuk-e, hogy minden esetben kiértékelődjön a beállítás, vagy csak egyszer, és többet ne. Ez jól jöhet olyan esetben, ha egy alkalmazás első indításához szükséges beállításokat akarjuk elvégezni központilag, és a továbbiakban mindenki testre szabhatja azt magának. Az utolsó lehetőség a pozicionálás beállítása. Itt egy új dialógust kapunk, ahol ismét rengeteg új lehetőségünk van. Talán ez a legérdekesebb része a GP Preferences-nek.



**1. ábra.** Szinte minden elem elérhető felhasználóhoz kötötten és számítógéphez kötötten is, ami segíti a konfigurálás pontosabb pozicionálását, és nem kell belefutni abba a csapdába, amit GPO esetén csak a loopback processing bekapcsolásával lehet megoldani

ahol például az ODBC-kapcsolatokat kezelhetjük, testre szabhatjuk a mappák kinézetét és tulajdonságait mind Vista, mind XP operációs rendszer esetén, csatlakoztathatunk nyomtatókat, vagy ütemezett feladatokat állíthatunk be néhány kattintással.

Hasznos lehet a „devices” lehetőség, ennek segítségével letilthetünk olyan eszközöket, amelyeket nem akarunk, hogy a felhasználók elérhessenek.

Jó megoldás lehet ez például banki terminál vagy bérszámfejtő gépek esetén, ahol letilthetjük az USB- vagy floppyeszközöket, hogy ne lehessen bizalmas adatokat lemásolni hordozható médiára. Jó lehetőségnek tartom még a „Local Users and Groups” használatát olyan esetben, ha a számítógépen egyedileg bekonfigurált csoportokhoz hozzáadni szeretnénk felhasználót vagy csoportot, esetleg törölni akarunk a meglévő csoportok közül. GPO-val ezt nem lehet megoldani, hiszen a „Restricted Groups” opció mindegyik lecserele a felhasználókat és csoportokat a policyban megadott csoportokra.

zó beállításokat összegyűjtő lap, ahol összeállíthatjuk, hogy pontosan mi az, amit végre kell hajtani a kliensen, és egy „Common” lap, ami minden elem esetében ugyanaz.

### Kiértékelés, lefutás

Az Common lapon olyan beállítások végezhetők el, amelyek az elem kiértékelését vagy lefutását szabályozzák. Elsőként beállítható, hogy ha az elem kiértékelése hibába ütközik, folytatódjon-e a házirend kiértékelése a többi elemmel. Ez főleg akkor lehet hasznos, ha a beállítások valamilyen módon függenek egymástól (például fájlt akarunk másolni egy előzőleg létrehozott mappába stb.). A következő jelölővel kiválaszthatjuk, hogy a bejelentkezett felhasználó nevében vagy a rendszer nevében történjen-e a beállítás.

Nagyon hasznos (ezzel lehet elérni például, hogy ne kelljen a felhasználónak jogot adni a saplogon.ini módosítására), természetesen csak a felhasználókra vonatkozó beállításoknál érvényes. Ezután megadhatjuk, hogy mi történjen, ha a kliens már kívül esik

### Elemenként

A targeting vagy pozicionálás segítségével a házirenden belül minden egyes elemre meghatározhatjuk, mi legyen a hatóköre, azaz milyen feltételek fennállása esetén lehessen alkalmazni. Ezek a feltételek roppant széleskörűen definiálhatók. Nemcsak arra kell gondolni, amit a csoportházirend is tud, ha megfelelő WMI-lekérdezést írunk hozzá, hanem teljeskörű dialógustámogatást kapunk, ahol pár kattintással tudunk speciális feltételeket kialakítani és ezeket a feltételeket logikai csoportokba rendezni. Tudunk feltételeket szabni csoporttagság alapján vagy csak bizonyos felhasználókra. Alapozhatunk egy bizonyos szabad tárterületméret, minimális processzorsebesség vagy minimum-memóriaméret meglétére vagy éppen hiányára, egy bizonyos fájl vagy mappa létezésére, az operációs rendszer verziójára és így tovább.

A mellékelt képen az összes lehetőségtípus látszik. Érdekes lehet a „language” opció, amely segíthet a többnyelvű beállítások vagy megjelenítendő figyelmeztetések (például EULA) konfigurálásában, akár a felhasználó, akár a számítógép lokalizációs beállításait figyelembe véve.

Hasznos lehet még az „MSI Query”, amelyvel köthetjük a beállításaink lefutását például egy bizonyos telepített patch vagy update meglétéhez. Hasznos lehetőség még a „portable computer”, hogy más beállításokat küldjünk hordozható és asztali számítógépek

Group Policy Preferences	Group Policy	
Kötelező-e	<ul style="list-style-type: none"> <li>• A beállítások kötelező jellegűek</li> <li>• A felhasználói beállítás dialógusa nem tiltott</li> <li>• Lehet csak egyszer alkalmazni</li> </ul>	<ul style="list-style-type: none"> <li>• A beállítások kötelező jellegűek</li> <li>• A felhasználói beállítás dialógusa tiltott</li> <li>• A beállítások kötelezően frissílnék időről időre</li> </ul>
Rugalmasság	<ul style="list-style-type: none"> <li>• Egyszerűen kezelhető registry-beállítások, fájlkezelés stb.</li> <li>• Importálható egyedi registry-beállítások, vagy akár egész registryágak a helyi vagy egy távoli gépről</li> </ul>	<ul style="list-style-type: none"> <li>• Nem lehet registry-beállításokat létrehozni, módosítani, nincs lehetőség fájlkezelésre stb.</li> <li>• Új lehetőségek hozzáadásához admin template-et kell létrehozni, és az alkalmazásnak is támogatnia kell a csoportházirend lehetőségeit.</li> </ul>
Helyi házirend	<ul style="list-style-type: none"> <li>• A beállítások nem érhetők el a helyi házirendben</li> </ul>	<ul style="list-style-type: none"> <li>• A beállítások elérhetők a helyi házirendben is</li> </ul>
Használhatósági kör	<ul style="list-style-type: none"> <li>• Használható nem csoportházirendre felkészített alkalmazás esetén is</li> </ul>	<ul style="list-style-type: none"> <li>• Csak csoportházirendre felkészített alkalmazással használható</li> </ul>
Tárolás	<ul style="list-style-type: none"> <li>• Az eredeti beállítások felülíródnak</li> <li>• A beállítás eltávolításával nem minden esetben kerül vissza az eredeti beállítás</li> </ul>	<ul style="list-style-type: none"> <li>• Az eredeti beállítás nem változik</li> <li>• A registry külön erre fenntartott ágában tárolódik</li> <li>• A beállítás eltávolításával visszaáll az eredeti állapot</li> </ul>
Pozicionálás és szűrés	<ul style="list-style-type: none"> <li>• Felhasználói felületen beállítható sokféle szűrési és pozicionálási lehetőség</li> <li>• A pozicionálás a házirend-kiegészítés minden elemére külön beállítható</li> </ul>	<ul style="list-style-type: none"> <li>• A szűrés és pozicionálás csak WMI segítségével valósítható meg, és ehhez külön WMI-lekérdezéseket kell írni</li> <li>• A szűrés csak a házirend egészére állítható be</li> </ul>

re. Nagyon megörültem a „Terminal Session” opciónak, mert ezt fel tudtam használni a már korábban említett saplogon-konfigurációs problémánk megoldására, ugyanis ezzel az RDP-kapcsolatban indított alkalmazásokhoz köthetünk különböző konfigurációs beállításokat.

A „Collection” gomb segítségével a feltételeket logikai csoportokba rendezhetjük, és tetszőleges mélységig egymásba ágyazhatjuk, így igen összetett feltételeket is összeállíthatunk. Megadhatunk például egy olyan összetett feltételt, hogy egy registry kulcs/érték módosítás csak akkor történjen meg, ha a számítógép hordozható, és Vista SP1 van rajta, vagy ha a számítógép asztali XP SP3 operációs rendszerrel telepített, és még a titok.txt fájl is létezik a System mappában.

Látható, hogy a lehetőségeink széles körben variálhatók, szerintem mindenki megtalálja benne a lehetőséget, hogy egyszerűsítse a kliensbeállítás folyamatait.

### Kliensoldali kiegészítés

Van azonban még néhány dolog, amire oda kell figyelnünk, mielőtt úgy döntünk, hogy kihasználjuk a GP Preferences lehetőségeit. Az első az, hogy kell hozzá kliensoldali kiegészítés, a neve pedig GP client side extensions. Ez csak a Windows Server 2008-nak

része, sem a Vista SP1, sem az XP SP3 nem tartalmazza. Szabadon letölthető azonban (és frissítésként is érkezik) a Microsoft oldaláról. XP SP2 vagy Server 2003 SP1 előtti operációs rendszerekre azonban nem lehet telepíteni. A második, hogy konfigurálni csak Server 2008-as gépről a beépített GPMC segítségével, vagy Vista SP1-ről a telepített RSAT GPMC használatával lehetséges.

Végezetül még egy kis táblázat a csoportházirend és a csoportházirend kliensoldali kiegészítések összehasonlítására, összefoglalva mindazt amit az előzőekben leírtam (lásd a fenti táblázatot).

### Sokoldalú segédlet

A GP Preferencesre tehát nem úgy kell tekinteni, mint a hagyományos csoportházirendre. Ennek a kiegészítésgyűjteménynek nem az a lényege, hogy valamit ráerőltessünk a felhasználóra, és az sem lényeges, hogy a felhasználó meg tudja változtatni ezeket a beállításokat, hiszen az itt megfogalmazott és beállított tulajdonságok mind a felhasználók életét segítik. Ezért inkább úgy kell gondolni rá, mint egy sokoldalú segédletre, ami a startup és a logon script helyett vagy mellett használható, leegyszerűsítve így a scriptelni nem tudó vagy időhiánnyal küzdő rendszergazdák életét.



2. ábra. A targeting lehetőségei

Azért nem kell teljesen eldobni a jó öreg scripteket és a tudást, amivel ezek készültek, hiszen a GP Preferences sem mindenható, biztos állíthatom, hogy még jó sokáig lesz olyan feladat, aminek a megoldásához szükség lesz a régi jól bevált megoldásokra, de ezek után már több időnk lesz az ilyen feladatok megoldására – vagy ahogy GT szokta mondani; több időnk marad a nyaralásra –, és ez mindenképpen jó hír.

Kapás Tibor  
MCSE, MVM Informatika Zrt.

Rendeljen már most  
képzéseket 2009-re!

## Idén fizet, jövőre kap akció a NetAcademiánál



Akár már a 2008-as költségkeret  
terhére is rendelhet tanfolyamokat a  
jövő évre. A képzésre szánt összeget  
mi most megtoldjuk +20%-kal.

Minden év végén a vállalatok életében elérkezik a pillanat, amikor kétségbeesetten próbálnak megszabadulni a rajtuk maradt éves költségvetési keret maradványaitól. Ilyenkor születnek a legfurcsább beruházások: két tonna gemkapocs, 70 km UTP-kábel, 40 folyóméter Post-it.

Az **Idén fizet, jövőre kap akció\*** választ kínál a „Mire költjük el” dilemmára, hiszen képzésre költeni mindig jó befektetés. Az akció keretében minden idén megrendelt és kifizetett 2009-es tanfolyamdíjat a NetAcademia +20% értékű szabadon felhasználható tanfolyamkerettel emel meg. Ha például az Önök cége 1 millió Ft-ot fordít még idén a költségkeretéből 2009-es tanfolyamokra, akkor oktatóközpontunk azt 1,2 millió Ft-ra egészíti ki.

### Képzési javaslatunk:

- Microsoft üzemeltetői képzések,
- Microsoft fejlesztői képzések,
- SQL Server 2005 és Üzleti intelligencia (BI) képzések,
- SharePoint,
- Microsoft Office 2003/2007 képzések,
- ITIL és IT folyamatirányítás képzések,
- Cisco képzések.

\*Az akció a 2008. december 15-ig beérkezett megrendelésekre érvényes.

**Microsoft®**  
**GOLD CERTIFIED**

Partner

Tanfolyamainkról bővebb listát és  
információt honlapunkon talál:  
[www.netacademia.net/training.aspx](http://www.netacademia.net/training.aspx)

Az Idén fizet, jövőre kap akció  
részleteiről kérjük, érdeklődjön  
Szántó Zoltánál (tel.: 20/369-6947).

NetACADEMIA

A LEGJOBBAKAT TANÍJTJUK.

Microsoft



Nem eszik,  
nem alszik,  
dolgozik!



Windows Server 2008