

A hierarchikus adatbázis struktúra jellemzői

Az első adatbázis-kezelő rendszerek a hierarchikus modellen alapultak. Ennek az volt a magyarázata, hogy az élet sok területén első közelítésben elég jól lehet ábrázolni az adatokat egyszerű hierarchiával.

Olyan helyzetekben, amikor az alá- és fölérendeltségi viszonyok egyértelműek, a hierarchikus modell jól és gépileg hatékonyan megvalósítható módon írja le a valóságot.

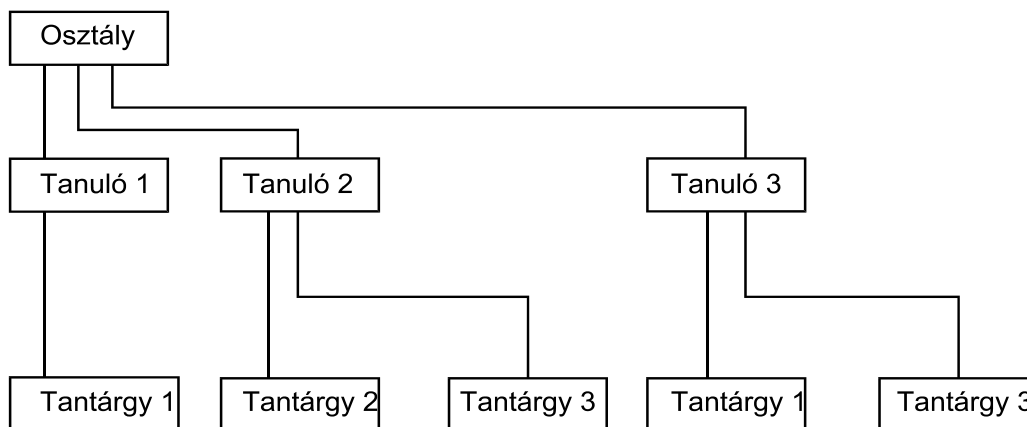
A gyakorlatban a legelterjedtebb hierarchikus rendszer az IBM által kifejlesztett Information Management System (IMS) 1968-ban jelent meg a piacon. Azóta bővítették, módosították és még ma is alkalmazzák.

A jellemzője, hogy a rekordok kisebb részekre, úgynevezett szegmensekre vannak felosztva, amelyek több szinten kapcsolódnak egymáshoz.

A felsőbb szinten lévő szegmenshez (szülő), alsóbb szinten elhelyezkedő szegmens (gyermek) kapcsolódik. A legfelső szintet gyökérnek (root) nevezzük. A szabály, hogy egy szülőhöz több gyermek is kapcsolódhat, ugyanakkor egy gyermeknek csak egy szülője lehet.

Sokan tartják ezt a megoldást korszerűtlennek, valójában lényegesen hatékonyabb gépi megvalósítást tesz lehetővé, mint a korszerű relációs adatbázisok, ha a feldolgozás sorrendje megegyezik a hierarchiával.

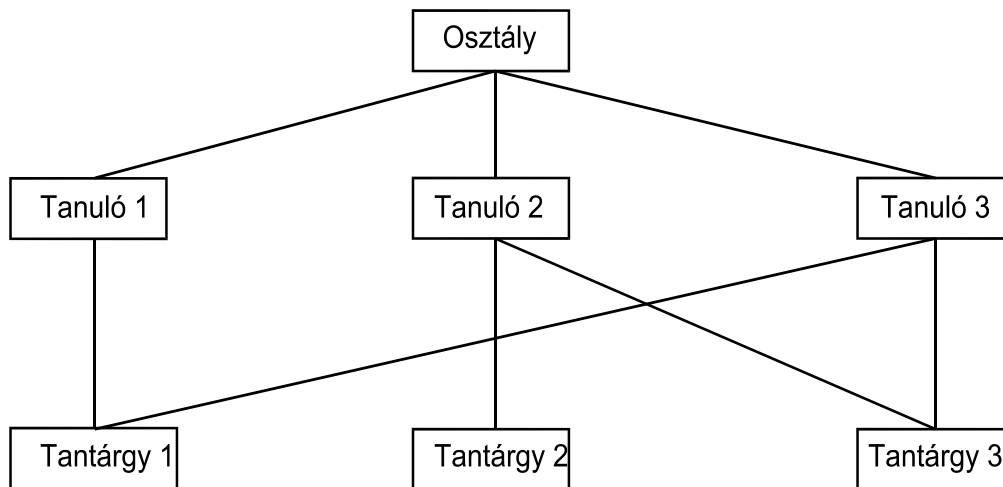
Az alábbiakban egy hierarchikus modell példáját láthatják:



Hálós adatbázis struktúra

A hálószerkezet az adatstruktúrában olyan általánosított hierarchikus kapcsolatot jelent, ahol az elemek között bármilyen kapcsolat fennállhat. A hálós hierarchiában lehetnek olyan szintek, ahol a kapcsolat nemcsak lefelé, hanem fölfelé is többértékű lehet, azaz a hierarchikus modell terminológiáját használva egy elemnek több közvetlen szülője is lehet.

A hálós adatbázis-struktúra az alábbi példán tekinthető át.



Valójában mind a hierarchikus, mind pedig a hálós adatbázis-struktúrának számos előnye van, melyek közül a legjelentősebb az, hogy a nagyméretű adatbázisokban gyors hozzáférést biztosít.

A legnagyobb hátrányuk viszont az, hogy rugalmatlanok. Bármilyen struktúraváltoztatás sok munkát és időt, ennek megfelelően tetemes költséget jelent. A másik jelentős hátrány, hogy nem támogatja az ad-hoc lekérdezéseket.

Relációs adatbázis struktúra

A relációs modellben az adatokat táblázatokban ábrázoljuk és a közöttük lévő különböző kapcsolatokat relációs műveletek segítségével hozzuk létre.

A táblázatok az adatok szerkezetének meghatározását, az adatok leírását teszik lehetővé a modell keretében, míg az adatkezelést a relációs műveletekkel valósítjuk meg.

Ezen a két alapvető feladaton kívül a relációs modell lehetőséget nyújt az adatok integritásának biztosítására, valamint a modell alkalmazásának hatékonyságát növelő szerkezetek kialakítására.

A relációs adatbázis struktúra a legelterjedtebb struktúra az adatbázis-kezelő rendszerek közül.

A relációs struktúra előnyei:

- nagy rugalmasság;
- az ad-hoc lekérdezések támogatása;
- az adatbázis tervezése, módosítása, karbantartása nagyon egyszerű;
- a táblázatos ábrázolásmód következtében az átlagos felhasználói szemlélethez ez áll a legközelebb;

- matematikai szempontból - elsősorban Codd ¹ alapvető munkái következtében - ezt a modellt dolgozták ki legrészletesebben;
- ebben a modellben valósítható meg legjobban, legegyszerűbben és legnagyobb mértékben a fizikai és logikai adatfüggetlenség.

A személyi számítógépekre adaptált adatbázis-kezelő rendszerek csaknem kivétel nélkül a relációs modellen alapulnak, és az utóbbi évtizedben a közép és nagygépekre elkészített rendszerek zöme is relációs.

Ahhoz, hogy ezek a táblázatok valóban relációs adatmodellt alkossanak, az egyes táblázatokban a következő feltételeknek kell teljesülnie:

- **Minden relációnak neve van**, mely egyértelműen azonosítja az adatmodellben.
- Adott reláció minden sorában azonos számú oszlop van. **Az oszlopok számát nevezzük a reláció fokának.**
- Minden **oszlopnak az adott reláción belül egyértelmű neve** van, amellyel azonosítható.
- Minden oszlop csak **meghatározott értékeket vehet fel** és azonos típusú elemi adatot tartalmazhat.
- **Bármely** sorban bármelyik **oszlop csak egyetlen értéket vehet fel** a megengedett értékek közül.
- Az oszlopok sorrendje nem befolyásolja az adatmodellt.
- Mivel a reláció, a táblázat egyed előfordulásai az egyes sorok, a táblázatnak **nem lehet két teljesen azonos sora**, azaz két olyan bejegyzése, melyben minden oszlopérték megegyezik.
- **A sorok számát nevezzük a reláció kiterjedésének.**

A relációs modell hátrányai:

- Az egyetlen modellezési alapelem nem teszi lehetővé az egyedhalmazok és a kapcsolatok egyértelmű szétválasztását. Mint láttuk, a táblázat vagy egy egyedhalmazt, vagy egy egyedhalmazt és kapcsolatait, vagy két egyedhalmaz kapcsolatát írja le.
- A relációs modellben az egyedek azonosítása és a kapcsolatleírás egyaránt értékalapú, ezért az integritási kényszerek állandó ellenőrzése, és a szükséges műveletek elvégzése plusz feladatot jelent az adatbázis-kezelők számára.
- A különböző táblázatokból az összetartozó adatok lekérdezése a táblázatok összekapcsolását igényli, ami tárolásigényes és lassú.
- A tulajdonságok elemi adattípusának megkövetelése miatt az összetett adatstruktúrákat elemekre kell bontani, és több önálló táblázatban kell tárolni. A táblázatok összekapcsolása a már említett hátránnyal jár.
- A modell az adatokon végzett műveletek leírását a koncepcionális modell szintjén egyáltalán nem teszi lehetővé. Az adatmodell külső szintjén az állandó jellegű lekérdezési műveletek definiálhatók, a felújítási jellegű műveletek azonban nem.
- Az adatbázist kezelő felhasználói programok készítése az SQL-en kívül egy másik programnyelvet is igényel. Mivel az SQL deklaratív nyelv, így nem

¹Codd matematikus, a relációs adatbázis-kezelés elméletének megteremtője

tartalmaz vezérlőszerkezeteket, azaz szekvenciát, elágazást, ciklust, valamint a perifériákat kezelő utasításokat stb.

A relációs modell vázolt hátrányait az objektumorientált adatmodell tudta legteljesebben kiküszöbölni.

Az objektumorientált koncepció

Az objektumorientált (OO) koncepció az objektumorientált programozásból (OOP) ered, melyet a hagyományos programozási módszerek alternatívájaként fejlesztettek ki. A hagyományos programozási nyelvekben az adatok és a rajtuk végrehajtott műveletek elkülönültek egymástól. Ezzel szemben az objektumorientált környezetben az alapvető strukturális elem, az objektum az adatok mellett az adatokon végrehajtott műveleteket is tartalmazza.

Az objektum a valós világ egy egyedét reprezentálja, attribútumokból és módszerekből áll, egyértelműen azonosítható, rendelkezik a bezárás, elrejtés, üzenetküldés, öröklés és a polimorfizmus tulajdonságokkal. Az azonos felépítésű objektumok egy osztályba tartoznak.

Az **objektum** tehát osztályának **egy példánya**, ezért szokásos **az objektumpéldány** elnevezés is. Az **osztály fogalom** alatt az **objektumok közös szerkezetének definícióját** értjük, ami megfelel az **egyedhalmaz** absztrakt fogalmának.

Az attribútumok az egyedeket jellemző tulajdonságok. Minden attribútumnak van egy egyedi neve és egy adattípusa. Az adattípusok a hagyományos elemi és a strukturált adattípusok egyaránt lehetnek.

Minden műveletet, amit egy objektumon végre kell hajtani, egy **metódus** képvisel. A metódusok megváltoztathatják, illetve lekérdezhetik az objektum attribútumainak értékét. A metódusok lényegében megegyeznek a hagyományos programozási nyelvek eljárásaival. Minden metódus egy azonosító névből és egy törzsből áll. A törzs valamilyen programnyelven megírt, a számítógépnek szóló utasítások sorozata.

Az objektumok szerkezetüket tekintve zártak, ez a belső szerkezet rejtett a külvilág előtt. Az attribútumok és a metódusok törzsét alkotó utasítások közvetlenül nem érhetők el egyetlen másik objektumból sem.

Egy objektum egy másik objektum metódusait **üzenetküldéssel** aktivizálhatja. Egy objektum abból a célból küldhet üzenetet egy másik objektumnak, hogy megváltoztassa vagy lekérdezze annak állapotát. Az üzenetküldés nem más, mint a fogadó objektum számára a metódus nevének és a szükséges paramétereknek az előírása.

Az **osztályok**, az egyed-kapcsolat modell egyedhalmaz hierarchiájához hasonló **osztályhierarchiába** rendezhetők. **Egy objektum több osztályból is származhat.** A hierarchia kialakításának folyamatát az objektumorientált koncepcióban általánosításnak nevezzük. Az osztályhierarchia felső szintjein lévő osztályok,

melyekből más osztályok származnak, a **főlérendelt osztályok**, az alsóbb szinteken lévő osztályok pedig az **alosztályok**.

Egy osztályhierarchián belül lehetséges az öröklés. Ez azt jelenti, hogy az alosztály öröklí a főlérendelt osztályoktól az attribútumokat és a metódusokat. Az adatok és metódusok öröklődése az osztály-hierarchiában fentről lefelé történik. Az öröklés lehet egyszeres vagy többszörös. Egyszeres öröklésről akkor beszélünk, ha egy osztálynak csak egy közvetlen főlérendelt, szülő osztálya van. A jelenlegi objektumorientált rendszerek legtöbbje az egyszeres öröklést támogatja. A többszörös öröklés esetén egy osztálynak több közvetlen főlérendelt osztálya is van. Néhány objektumorientált rendszer megengedi a felhasználó által definiált öröklési szabályok használatát is. Az alosztályok definiálásakor kell megadni a szülő osztályokat, és ekkor kell rendelkezni az öröklésről is.

Az objektumorientált terminológiában a **polimorfizmus** azt jelenti, hogy az osztályhierarchia különböző szintjeinek osztályaiban definiált metódusok esetén megengedett ugyanazon metódusnév használata, de a metódus törzsek különbözők lehetnek. A főlérendelt osztályok metódusait az alosztályok szintjén újradefiniálhatjuk, természetesen más-más alosztályban különböző módon.

Az osztályok és objektumpéldányok **grafikusan is ábrázolhatók**. A grafikus ábrázolás előnye, hogy eltekinthetünk a programnyelvek szintaktikai szabályainak ismeretétől, hiszen az osztályokat és az objektumpéldányokat egy-egy téglalap képviseli. Osztály esetén, a téglalapot három részre osztjuk, a felső részben az **osztály megnevezését**, a középsőben az **attribútumokat**, az alsóban pedig a **paraméterlistáikkal együtt a metódusokat** tüntetjük fel.

Az osztályoknál az öröklést egy nyíllal jelöljük, a nyíl az alosztályból indul, és a főlérendelt osztályra mutat. Az objektumpéldányoknál a téglalapot két részre bontjuk, a felső részben az objektumazonosító értéke, az alsó részben az attribútumok és aktuális értékeik jelennek meg, melyek együttesen képviselik az objektum állapotát.

Objektumorientált adatmodell

Az objektumorientált adatmodell még nem olyan kiforrott, mint az egyed-kapcsolat, illetve a relációs modell. 1991-ben alakult meg az Object Database Management Group (ODMG), melynek célja az objektumorientált adatbázis-kezelőkre vonatkozó szabvány kidolgozása. A munkába bekapcsolódtak a főbb OODBMS (Object Oriented Database Management System) gyártók is. Az együttes munka eredményeképpen 1995-ben megszületett az ODMG-93 szabvány. 1997-ben publikálták a továbbfejlesztett ODMG 2.0 szabványt.

A szabvány, ajánlást tartalmaz az objektummodellre, az objektum-definíciós nyelvre (Object Definition Language - ODL) és az objektum-lekérdező nyelvre (Object Query Language - OQL). Az objektumorientált adatbázis-kezelő rendszerek objektumokat kezelő nyelve (OML - Object Manipulation Language) a C++, a Smalltalk, a Java objektumorientált programozási nyelvek egyike. A szabványban alkalmazott modellezési alapelemeket a gyakorlatban ezen nyelvek típusrendszerének feleltetik meg.

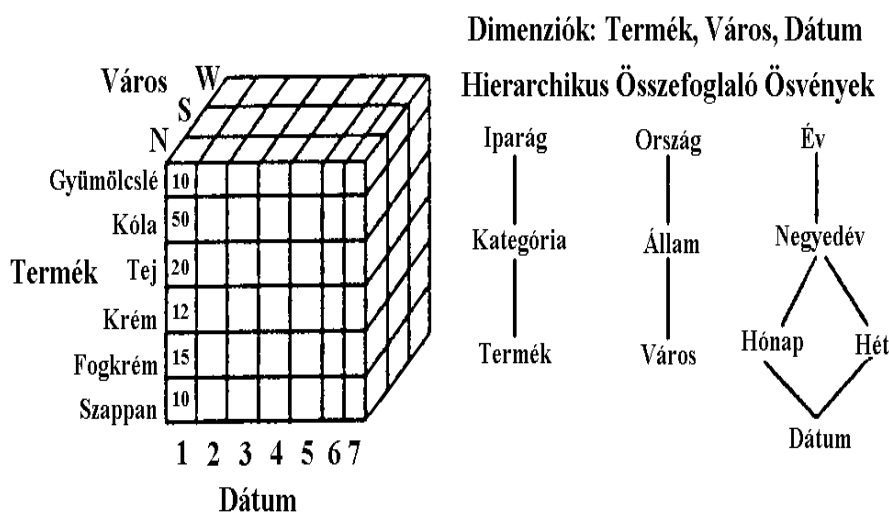
Az elmúlt években jó néhány OODBMS került kereskedelmi forgalomba világszerte és Magyarországon egyaránt. Jelentős az objektumorientált adatmodellen alapuló, a gyakorlatban megvalósított adatbázisok száma is. Magyarországon is találunk már néhányat.

A modell, a programozási nyelvekhez hasonlóan egy típusrendszerrel segíti az egyedekek és kapcsolataik leírását. A típusrendszer két alapvető típusal - két modellezési alapelemmel - rendelkezik, melyek az objektum típus és a literál típus. Az objektum típusú elemek mindig rendelkeznek egy egyértelmű azonosítóval, a literál típusúak viszont nem. A literál típusok felelnek meg a programozási nyelvek elemi és összetett adattípusainak, az objektum típusok pedig az objektum-orientáltság objektum fogalmának. Mint látni fogjuk, a literál típusokkal az attribútumokat, az objektum típusokkal pedig az osztályokat és kapcsolataikat definiáljuk.

Multidimenzionális adatszerkezet

A döntéstámogatás eléggé eltérő követelményeket támaszt az adatbázis technológiával szemben, összehasonlítva a **hagyományos online tranzakció feldolgozó** alkalmazásokkal.

A multidimenzionális adatszerkezet egy olyan kocka alakú, többdimenziós adathalmazként képzelhető el (lásd az alábbi ábrán), melyben a táblák tartalma és kapcsolata a gyors lekérdezés lehetőségét célzó.



1. ábra

Az **adattárház** és az **OLAP (on-line analytical processing)** a döntés támogatás alapvető elemei, amelyek mindinkább az adatbázis ipar fókuszába kerültek. Manapság több kereskedelmi termék és szolgáltatás is elérhető, és az összes vezető adatbázis management rendszer gyártónak van kínálata ezen a területen.

Adatbányászat

Az információrobbanás korát éljük. A legtöbb nemzetközi cég néhány nap alatt több információt állít elő, mint amennyit sok ember egész életében képes elolvasni. Az intraneteken kívül óriási mennyiségű információ van felhalmozva az Interneten is. Ezen információk közül sok **strukturált formában található (különböző adatbázisok)**, de a legnagyobb rész (becslések szerint több mint 80%) strukturálatlan adat, leginkább szöveges állomány.

Az információ fontos termelési tényezővé lépett elő. Az irányításban a jó döntésekhez minél átfogóbb, relevánsabb információk szükségesek.

Új technikák jöttek létre, amelyek segítségével fontos, új ismeretekhez juthatunk meglévő adatbázisok adataiból. Ilyen új technikák az adatbányászat (data mining), illetve adatbázisok tudásfeltárása (KDD: Knowledge Discovery in Databases). A két kifejezés különböző, de átfedéseket is tartalmazó technikákat jelöl. Sokszor egymás szinonimájaként használják, bár a KDD fogalma bővebb.

Az adatbányászat módszertanának lényeges összetevői évtizedek óta fejlesztés alatt állnak számos kapcsolódó terület kutatásainál, mint pl. statisztika, mesterséges intelligencia, szakértői rendszerek, az ún. „machine learning” technikák. Az adatbányászat elterjedésében a hatalmas teljesítményű számítógépek köznapivá válásán túl szerepet játszott az alapjait adó matematikai és logikai módszerek nagymértékű fejlődése, valamint azon szoftverek megjelenése, amely az adatbányászat folyamatát automatizálták, s ezáltal az széles körben elterjedté válhatott.

A KDD folyamata

- **Az adatok kiválogatása**

Ahhoz, hogy jó döntéseket lehessen hozni a vállalat számára, ismerni kell a múltat, s meg kell határozni a jelentős trendeket. Az ehhez szükséges információk különböző adatbázisokban lelhetők fel, amelyekben a napi tranzakciók eredménye van tárolva. Ezekből az úgynevezett operációs adatbázisokból építünk fel egy új, az előzőektől elkülönített adatbázist.

- **Az adatok tisztítása**

Az egyesített adatbázist át kell vizsgálni, hiszen számos hibás adatot tartalmazhat, amely a későbbi eredményt torzíthatja. Az adattisztítás folyamán sor kerül a duplikált tételek kizárására, az ellentmondásos adatok, a még nem befejezett tranzakciók által módosított ún. „piszkos adatok” kiszűrésére.

- **Az adatok bővítése**

A meglévő egyedtípusok tulajdonság adatait – amennyiben rendelkezésre állnak vagy beszerezhetők – bővíthetjük további tulajdonságokkal, amivel újabb következtetések tehetők, s a döntésekhez további információk nyerhetők.

- **Az adatok kódolása**

Az előző lépésben extra információkkal gazdagítottuk meglévő rekordjainkat. Előfordulhatnak azonban számos rekordnál hiányzó adatok. Ilyenkor dönteni kell: elegendő információt tartalmaz-e a rekord vagy törölni kell? Eddig a pontig az

előzőekben felsorolt lépések SQL utasításokkal végrehajthatók. Lehetséges azonban, hogy a rekordokban tárolt információk túl részletesek ahhoz, hogy mintafelismerési algoritmust használjunk. Ilyenkor a túlon túl részletes információt kódolni kell. Például a teljes születési dátum helyett elegendő egy vásárló korát 10 éves intervallumokban meghatározni, vagy lakcím helyett elegendő a régió kódját megadni. Az előző négy lépés eredményeként létrejött adatbázist *adattárháznak* nevezzük. Az adattárházak hatalmas méretűek és régebbi adatokat tartalmaznak

Adatbányászat Az adatbányászat nem annyira egy technikát jelöl, mint inkább egy eszmét, miszerint több információt tartalmaznak az adatok, mint ahogy az az első ránézésre látszik. Bármely technika, amely egy adatbázis adataiból további hasznos információt hoz elő, hasznos lehet.

Ezen technikák az alábbi területekről származnak:

- Adatbázisok lekérdező eszközei
- Statisztikai módszerek
- Adatok megjelenítése, vizualizáció
- Kimutatások, grafikonok készítése

A globalizáció, integráció korában számos helyi adatbázist építenek adattárházzá. Természetesen az **adattárház** felépítése, s benne az adatbányászat nem öncélú művelet: Az adatbányászat eredményét össze kell kapcsolni a különböző *döntéstámogató rendszerekkel* ahhoz, hogy a kapott információ valóban a jó, hatásos döntések meghozatalát segítse elő.

Adattárház (data warehouse)

Az adattárház egy „témaorientált, integrált, történeti, nem változó adatgyűjtemény, amelyet elsősorban a szervezési döntéshozásban használnak.”

Jellemzők:

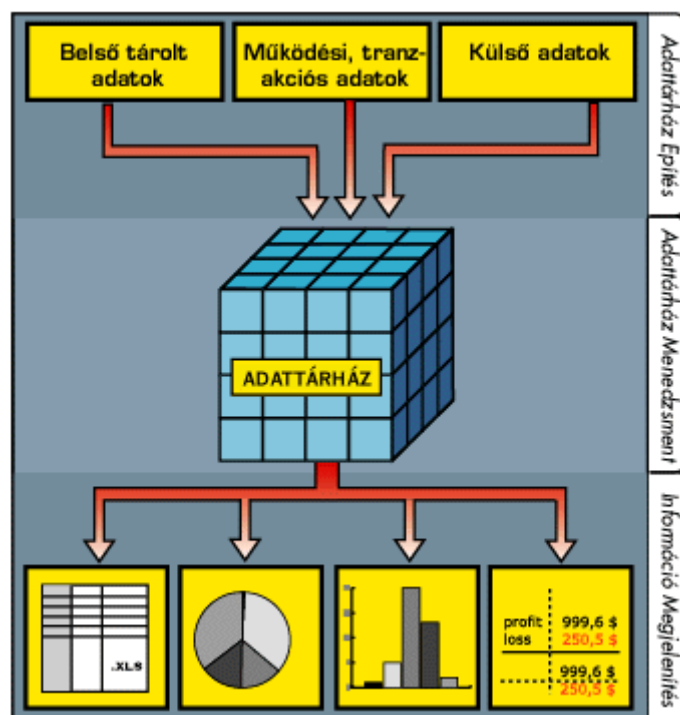
- Az adatok **időtől való függése**. Az információ és a létrehozás időpontja között kapcsolat van.
- **Állandóság**. Az adatokat az adattárházban soha nem írják felül, csak lekérdezik. Az adatok módosítása az operációs adatbázisokban történik.
- Az adattárházak **tárgyorientáltak**. Az operációs adatbázisokból csak a témának megfelelő adat kerül be az adattárházba.
- Az operációs adatbázisokban különböző neveken, különböző típusokkal használhatjuk ugyanazt az egyedet. Az adattárházakban a nevek, típusok **egységessé**nek

Napjainkra a nagyvállalatok hatalmas adatmennyiséget kezelnek és generálnak ügyfeleikről, a nekik nyújtott szolgáltatásról és az üzletmenetről általában. Az adat azonban még nem információ és a felhalmozott adatok mennyisége sokszor inkább akadálya, mint elősegítője a tisztánlátásnak. Világossá vált, hogy a folyamatosan duzzadó adathalmazok elemzése és a belőlük levonható következtetések stratégiai fegyverként szolgálhatnak a globalizálódó piacon egyre keményebb versenyben álló vállalatok számára. Éppen ezért ezek az elemző eszközök és technológiák a vállalati

struktúrában a stratégiai szempontokat megtestesítő menedzsment és azon belül is különösen a pénzügy, controlling, marketing és ügyfélszolgálat területei számára készülnek.

Az erős versenyhelyzettel, és gyors változásokkal jellemezhető üzleti környezetben az azonnali vállalatirányítási döntésekhez elengedhetetlen a szükséges tények, és megbízható adatok elérhetőségének biztosítása. A döntések megfelelő előkészítésében a legnagyobb kihívás legtöbbször az információk összegyűjtése a szervezet különböző informatikai rendszereiből, illetve az összegyűjtött adatok azonos formában történő prezentálása, adott esetben az egyes részadatok újrafeldolgozási lehetőségének biztosítása.

Az elemzésekhez persze adat kell, és az elemzői munka sajátos igényei és szempontjai, valamint a hardver- és szoftvertechnológia fejlődése nyújtotta új lehetőségek végre megteremtették a kifejezetten elemzési, lekérdezési céllal létrehozott, nagyméretű elkülönített adatbázist, a Data Warehouse-t. A „data warehouse” egy elkülönített környezetben, dedikált adatbázisra épül, mely adatbázis különböző forrásokból kaphat adatokat, hogy a széleskörű lekérdezéseket és a részletekbe menő elemzéseket egyaránt támogathassa.



2. ábra

Az adattárház technológia a döntéstámogató eszközök generációjának legújabb és leghatékonyabb eszköze. Világméretű elterjedése annak köszönhető, hogy segítségével megalapozott és gyors döntések hozhatók. A felhasználók elemezni tudják a különböző események hatását és a tevékenységükkel kapcsolatos trendeket. Ennek eredményeként a felhasználók idejüket hasznosan, az adatok megkeresése, összegyűjtése, rendezése helyett a fontos információk analizálásával, elemzésével és a szükséges intézkedések megtételével tölthetik.

Az adattárház alkalmazásának előnyei

<p>Üzleti döntések</p> <ul style="list-style-type: none"> • Fejlett döntési folyamat • Megalapozott stratégiai tervezés • A döntések minőségi javítása • A jó döntések száma növelhető • Fejlettebb kiértékelési lehetőségek • A trendek kimutathatósága • Fejlett költséganalízis • Fejlett készletkezelés • Az üzleti folyamatok átláthatók 	<p>Adatelérés</p> <ul style="list-style-type: none"> • Fejlett adatrendelkezésre állás • Jobb adatminőség • Fejlett adatintegritás • Jobb időbeni áttekintés • Könnyebb adatelérés • Adatbányászati lehetőség • Eddig el nem ért adatok láthatóvá tétele • A felhasználó részére több adat biztosítható
<p>Költségek</p> <ul style="list-style-type: none"> • Kevesebb munkaerő • A veszteségforrások láthatóvá tétele • Az erőforrások kihasználása javul • Alacsonyabb raktárkészlet • A készlet forgási sebessége nő 	<p>Termelékenység</p> <ul style="list-style-type: none"> • Programozás nélküli adatelérés • Egyszerűbb adatelemzés • A manuális adatkezelési munka csökkenése • A meglévő rendszer hiányosságainak kiküszöbölése

3. ábra

A legtöbb data warehouse megoldás hosszadalmas és aprólékos bevezetési folyamatot követel meg, és integrációja is komoly nehézségek elé állítja a szakembereket. Az egyik legnagyobb probléma a data warehouse hozzáillesztése a felhasználó szervezet üzleti folyamataihoz. Ezen kívül a legtöbb rendszer egyszerűen csak információt szolgáltat, és nem biztosítja azokat az eszközöket, melyek szükségesek a megszerzett ismereteken alapuló konkrét lépések megtételéhez.

Osztott adatbázisok

A **telekommunikáció** fejlődése lehetővé tette, hogy a létrehozott adatbázist több, fizikailag egymástól távol lévő helyen tároljuk és az adatkezelés, adatfeldolgozás is elosztott legyen.

Az osztott adatbázis-kezelő rendszer² képes arra, hogy a felhasználó egy logikailag egységes adatbázist használhasson, mely egyébként fizikailag különálló egységekből épül fel.

Az osztott adatbázisoknak különféle megvalósításai léteznek, ilyenek:

- a **központi adatbázis egy részét** távoli felhasználónál tároljuk (particionált adatbázis),
- a **teljes adatbázist** távoli felhasználónál tároljuk (duplikált adatbázis),
- **nem létezik központi adatbázis**, csak központi index, az összes rekord csak a távoli felhasználóknál van tárolva. A lekérdezés alkalmával a központi index adja meg a keresett rekord helyét. Ezeket a rekordokat a felhasználó az osztott adatbázis-kezelő rendszer segítségével kapja meg.

Az osztott adatbázis-kezelés az utóbbi években igen gyorsan terjed, s ez is egy lehetséges megoldás. Csupán rövid betekintést kívántunk adni erről a jövő szakembereinek.

²Distributed DataBase Management System (DDBMS)