



Adatbázisok

- Alapfogalmak
- Relációs adatmodell



Alapfogalmak



Adatbázis

➤ Számítógépes adatbázis

**A valós világ egy részének leírásához
használt adatok összefüggő, rendezett,
tárolt halmaza.**



Adatbázis-kezelő rendszer

- DataBase Management System, **DBMS**
 - Szoftverrendszer
 - Adattárolás speciális adatstruktúrában
 - Hozzáférés az adatbázis adataihoz
 - pl. Oracle DB, MS SQL, IBM DB2, MySQL, PostgreSQL, ElasticSearch, MongoDB



DBMS jelentősége

- **NAGY** mennyiségű adat kezelése
- Adataok **BIZTONSÁGOS** tárolása
- **HATÉKONY** adatelérés,
- felhasználók **IZOLÁCIÓ**ja



DBMS szolgáltatásai

- 1. Adatműveletek**
- 2. Adatintegritás**
- 3. Adatbiztonság**
- 4. Tranzakciókezelés**
- 5. Adatvédelem**
- 6. Több felhasználó, párhuzamos hozzáférés**
- 7. Hatékony keresési és módosítási lehetőségek**



Adatműveletek

- Create_(Létrehozás)
- Read (Kiolvasás)
- Uppdate (Módosítás)
- Delete (Törlés)



Adatintegritás

➤ Formai/tartományi

- Az adatok egy előre meghatározott tartományból kerülnek ki
 - pl testmagasság, cipőméret, életkor (típus + tartomány)

➤ Referenciális

- Egy mező értéke egy máshonnan kiolvasott mező értékével meg kell, hogy egyezzen (számla - megrendelő)

➤ Strukturális

- A tervezéskori feltételezések sérülnek-e
 - pl nevek egyedisége (*egyértelműség*), kényszerek (*constraint*) sérülése

➤ ➔ DBMS biztosítja (CRUD műveleteknél)



Adatbiztonság

- Megbízható tárolás
 - Ne sérülhessenek meg adatok,
 - Ne veszhessenek el adatok
- Megoldás
 - Naplózás
 - Redundancia, többszörös tárolás
 - Paritás bit külön HDD-n
 - Időszakos mentés, visszaállítás



Tranzakciókezelés

- **Atomi műveletsorozat:** oszthatatlan, tehát vagy végrehajtódik a lánc minden eleme, vagy egy sem.
- **Tranzakció:** atomi módon végrehajtott műveletek sorozata



Adatvédelem

➤ Illetéktelen felhasználók elleni védelem

■ Szoftveres védelem (jelszó)

- Nem invertálható tárolás (pl MD5)
- Brute force: pl 10 karakter => $(26+26+10)^{10} \sim 8 \cdot 10^{17}$ lehetőség
 - 1G/s => 26 év

■ V.ö. – fizikai védelem

- Fizikailag elzárva (külön gép, terem)
- Hardveres védelem (hardver kulcs – pl. mágneskártya)



Párhuzamos hozzáférés (Szinkronitás)

- DBMS - általában többfelhasználós rendszer
- Párhuzamos módosítás => hibalehetőség, lassúság
 - pl raktárkészlet olvasása és írása egyidőben
- Megoldás:
 - Konfliktuskezelés
 - Párhuzamosítás
 - Jogosultságkezelés



Többféle nézet

- Az egyes felhasználók ugyanazon adatok más részleteit láthatják
 - Amire szükségük van
 - Amire jogosultságuk van
- Virtuális adatok
 - Származtatott, nem tárolt adatok



Hatékonyság

- Hatékony tárolás
 - Akár lekérdezésre optimalizált
- Index struktúrák
 - ~ tartalomjegyzék
 - Hatékony keresés, illesztés
- Hatékony programozás
 - Saját lekérdező / módosító nyelv (pl SQL)



DBMS szolgáltatásai

- 1. Adatműveletek**
- 2. Adatintegritás**
- 3. Adatbiztonság**
- 4. Tranzakciókezelés**
- 5. Adatvédelem**
- 6. Több felhasználó, párhuzamos hozzáférés**
- 7. Hatékony keresési és módosítási lehetőségek**



Összefoglalás

- Miért olyan fontosak?
- Miért nem egyszerűen fájlokban tároljuk?
- Miért nem írunk sajátot?
- Mikor nem érdemes használni?



Személyek

- Képzetlen, vagy külső felhasználó
 - Nem kell ismerni az adatbázis szerkezetét
 - Az adatbázis által biztosított jogosultságokkal
 - *Speciális kliensen keresztül használja az adatbázist*
- Alkalmazás programozó
 - Ismerni kell az adatbázis szerkezetét
 - Programozói ismeretek szükségesek
 - *A speciális alkalmazást készítő*
- Adatbázis adminisztrátor
 - Teljes jogosultság



Személyek - Adminisztrátor

➤ Létrehozás

- Felépítés meghatározása, állományszerkezet kijelölése

➤ Szerkezetmódosítás

- meglévő adatok nem sérülhetnek

➤ Jogosultságok karbantartása

- A hozzáférések jogának naprakészen tartása, módosításai

➤ Mentés-visszaállítás

- Időszakos biztonsági mentések
- Teljes vagy inkrementális mentések



DBMS típusok adatmodell szerint

➤ Adatmodell

■ Az adatbázisban tárolt adatok leírásának módja

- Adatok leírására szolgáló jelölés
- Adatok struktúráját leíró szabályok
- Kényszerek, megszorítások

➤ DBMS-ek adatmodell szerint:

■ Relációs adatbázisok – RDBMS

■ Nemrelációs adatbázisok

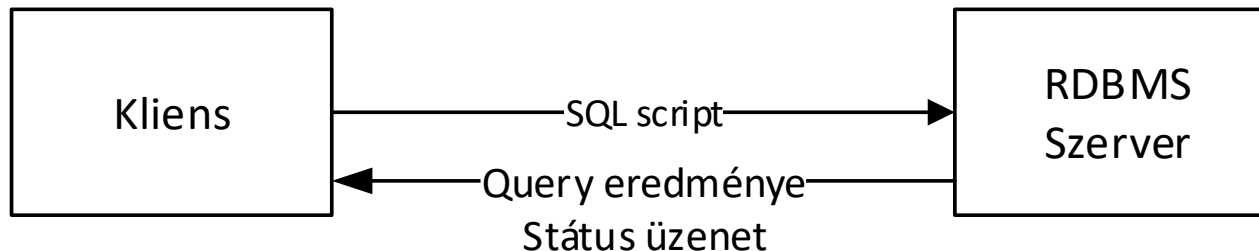
- Dokumentum alapú adatbázisok (pl. MongoDB)
- Kulcs-érték táruk
- Gráf adatbázis



RDBMS

➤ Programozási nyelv

- SQL – Structural Query Language (minden művelethez)
- Szabványos (ISO+ANSI)
- Nem általános célú
- Műveletek:
 - Séma szerkesztés
 - Adminisztráció (pl. jogosultságok)
 - CRUD
 - Query: lekérdezés
- Nagymértékű optimalizálás





Nem relációs adatbázisok

- Nem SQL alapú programozási struktúra
- Összefoglaló nevük: NoSQL adatbázisok



DBMS felhasználása

- DBMS szerver
 - API – programoói interfész
 - Programozási nyelv
- Általános grafikus kliens
 - API-n keresztül
- Feladat specifikus kliens



Adatbázis-kezelő rendszer

➤ DEMO



DBMS

- IBM DB2
- Oracle DBMS
- MS (Microsoft) SQL Server
- MySQL (MariaDB)



Relációs adatmodell



Relációs adatmodell

➤ Mi az az adatmodell?

■ Az adatbázisban tárolt adatok leírásának módja

- Struktúra (bonyult adattípusok, nem csak lista, tömb)
- Kényszerek (pl tartomány vagy sokkal bonyolultabb)
- Műveletek (korlátozott => magas szint, optimalizálható vs C)

➤ A legtöbb adatbáziskezelő alapja

➤ Halmazelméleti relációkon alapul

➤ Definíció: *Relációnak nevezzük halmazok Descartes-szorzatának részhalmazát.*



Relációs adatmodell - definíció

- Legyenek adottak a $T_1, T_2, \dots, T_n$ halmazok és képezzük ezen halmazok $T = T_1 \times T_2 \times \dots \times T_n$ **Descartes-szorzatát**.
- A T szorzat elemei olyan $(t_1, t_2, \dots, t_n)$ elem n -esek, amelyekre $t_1 \subseteq T_1, t_2 \subseteq T_2, \dots, t_n \subseteq T_n$
- A T szorzathalmaz egy R résztartományát **relációnak**,
- a $T_1, T_2, \dots, T_n$ halmazokat pedig a reláció **tartományainak, attribútumainak** nevezzük.



Relációs adatmodell

= TÁBLÁZAT



Relációs adatmodell – példa

- $T1 = \{1,2,3\}, \quad T2 = \{a,b,c\}$
- $T1 \times T2 = \{(1,a), (1,b), (1,c), (2,a), (2,b), (2,c), (3,a), (3,b), (3,c)\}$
- Relációk: tetszőleges részhalmaz, pl
 - $R1 = \{(1,a), (2,b), (3,a)\}$
 - $R2 = \{(1,b), (2,b), (3,a), (3,c)\} \dots$

■ $R1:$

T1	T2
1	a
2	b
3	a



Relációs adatmodell – példa / 2

➤ Egy „kicsit” hétköznapiabb példa

- $Name = \{a..z, A..Z, ' '\}^*$, $Age = N_{0..150}$, $Gender = \{f, m\}$
- EMPLOYEE reláció:

Name	Age	Gender
Dona Saylor	42	f
Jessie Dermody	21	m
Red John	36	m
Hillary Wesner	25	f

➤ Nevet rendelünk hozzá, felsoroljuk az **attribútumokat**

- Ezt hívjuk **SÉMÁNAK** = adat tárololási struktúrája
 - $PI\ EMPLOYEE (Name, Age, Gender)$
- Egy **ADATBÁZIS** összefüggő relációk együttese



A reláció tulajdonságai

➤ A reláció **foka** (arity, aritás)

- A relációban található oszlopok (attribútum, tulajdonság, tartomány) száma

- EMPLOYEE: 3

➤ A reláció **számossága**

- A relációban található sorok száma

- EMPLOYEE: 4

➤ A reláció **kulcsa** (**elsődleges kulcs**)

- Az a tulajdonság (halmaz), amely a reláció (táblázat) egy-egy sorát egyértelműen azonosítja

- EMPLOYEE: (Name, Age) ? => Egy vállalatban belül ez „valószínű” egyedi, egyébként célszerű egy egyedi azonosítót (sorszámot) is használni (~szigszám, taj szám, neptun kód...)

- Jelölés: aláhúzással: EMPLOYEE (Name, Age, Gender)



A reláció tulajdonságai – 2

- Nem tartalmaz két azonos sort
 - $\sim$ halmazok keresztszorzatának részhalmaza
 - konkrét implementációknál ez nem feltétlen lesz így
 - célszerű betartani
- Az oszlopok neve egyértelmű, sorrendjük nem számít
 - Csak ha sorszám alapján hivatkozunk az oszlopra
- A sorok sorrendje nem számít
- Tetszőleges számú sort tartalmazhat
- Egy sor és egy oszlop kereszteződésében egy érték szerepelhet



Adatbázis példa 1 – 1

➤ Ország nyilvántartás

■ ORSZÁG(Név, Terület, Lakosság, Pénznem)

<u>Név</u>	Terület	Lakosság	Pénznem
Magyarország	93 030	9 896 000	HUF
Olaszország	301 318	60 768 217	EUR
Dánia	43 094	5 655 750	DKK



Adatbázis példa 1 – 2

➤ Ország – város nyilvántartás

- ORSZÁG(Név, Terület, Lakosság, Pénznem)
- VÁROS(OrszágNév, Név, Terület, Lakosság)
 - Név nem egyedi (~ Budapest Georgia-ban)
 - VÁROS(OrszágNév, Név, Terület, Lakosság)



Adatbázis példa 1 – 3

➤ Ország

■ ORSZÁG(Név, Terület, Lakosság, Pénznem)

<u>Név</u>	Terület	Lakosság	Pénznem
Magyarország	93 030	9 896 000	HUF
Olaszország	301 318	60 768 217	EUR
Dánia	43 094	5 655 750	DKK

➤ Város

■ VÁROS(OrszágNév, Név, Terület, Lakosság)

<u>OrszágNév</u>	<u>Név</u>	Terület	Lakosság
Magyarország	Budapest	525	1 744 665
Magyarország	Pécs	163	146 581
Dánia	Koppenhága	77	1 246 611



Külső kulcs

➤ *Külső kulcs (idegen kulcs)*

- Egy másik (vagy akár ugyanazon) reláció elsődleges kulcsa, a külső kulcs értéke a másik reláció létező elsődleges kulcsai közül kerülhet ki (OrszágNév)
- Ezen keresztül hivatkozik az egyik rekord a másikra



Adatbázis példa 1 – 3

➤ Mesterséges kulcs

- ORSZÁG(Id, Név, Terület, Lakosság, Pénznem)
- VÁROS(Id, OrszágId, Név, Terület, Lakosság)

<u>Id</u>	Név	Terület	Lakosság	Pénznem
1	Magyarország	93 030	9 896 000	HUF
2	Olaszország	301 318	60 768 217	EUR
3	Dánia	43 094	5 655 750	DKK

<u>Id</u>	OrszágId	Név	Terület	Lakosság
1	1	Budapest	525	1 744 665
2	1	Pécs	163	146 581
3	3	Koppenhága	77	1 246 611



Adatbázis példa 2 – 1

- Raktár és rendelés nyilvántartás
 - TERMÉK (Név, Ár, Raktárkészlet)
 - MEGRENDELÉS (MegrendelőNév, Cím, Dátum)
 - MEGRENDELÉSTÉTEL (TermékNév, MegrendelőNév, MegrendelőCím, MegrendelésDátum, Db)



Adatbázis példa 2 – 2

➤ Termék:

■ TERMÉK (Név, Ár, Raktárkészlet)

<u>Név</u>	Ár	Raktárkészlet
Alma	150	50
Körte	280	48
Szilva	120	63

➤ Megrendelés

■ MEGRENDELÉS (Megrendelő, Cím, Dátum)

<u>MegrendelőNév</u>	<u>Cím</u>	<u>Dátum</u>
Kovács Béla	1111 Budapest...	2011.09.15
Kiss Pista	5000 Szolnok...	2012.04.21
Nagy Anna	9000 Győr ...	2012.10.04



Adatbázis példa 2 – 3

➤ MegrendelésTétel

- Mindkét reláció kulcsai szerepelnek
- MEGRENDELÉSTÉTEL (TermékNév, MegrendelőNév, MegrendelőCím, MegrendelésDátum, Db)

<u>TermékNév</u>	<u>MegrendelőNév</u>	<u>MegrendelőCím</u>	<u>MegrendelésDátum</u>	Db
Alma	Kovács Béla	1111 Budapest ...	2011.09.15	5
Körte	Kiss Pista	5000 Szolnok...	2012.04.21	3
Alma	Kiss Pista	5000 Szolnok...	2012.04.21	1
Szilva	Nagy Anna	9000 Győr ...	2012.10.04	2



Adatbázis példa 2 – 4

- TERMÉK (Id, Név, Ár, Raktárkészlet)
- MEGRENDELÉS (Id, Megrendelő, Cím, Dátum)
- MEGRENDELÉSTÉTEL (TermékId, MegrendelőId, Db)

➤ Termék:

	<u>Id</u>		Név	Ár	Raktárkészlet
	1		Alma	150	50
	2		Körte	280	48
	3		Szilva	120	63



Adatbázis példa 2 – 5

➤ Megrendelés

<u>Id</u>	Megrendelő	Cím	Dátum
1	Kovács Béla	1111 Budapest...	2011.09.15
2	Kiss Pista	5000 Szolnok...	2012.04.21
3	Nagy Anna	9000 Győr ...	2012.10.04

➤ MegrendelésTétel

<u>TermékId</u>	<u>MegrendelőId</u>	Db
1	1	5
2	1	3
1	2	1
3	3	2



Adatbázis példa 3 – 1

➤ Film adatbázis

- Studiók, filmek, színészek; ki – miben szerepel
- **STUDIÓ** (Név, Székhely)
 - => **STUDIÓ** (Id, Név, Székhely)

<u>Id</u>	Név	Székhely
1	Universal	Universal City
2	Columbia Pictures	Culver City
3	20th Century Fox	Los Angeles



Adatbázis példa 3 – 2

➤ Film adatbázis

■ Studiók, filmek, színészek; ki – miben szerepel

■ FILM (Cím, év, StudióNév, Hossz, Műfaj)

□ => FILM (Id, StudióId, Cím, Év, Hossz, Műfaj)

<u>Id</u>	<u>StudióId</u>	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action



Adatbázis példa 3 – 3

➤ Film adatbázis

- Studiók, filmek, színészek; ki – miben szerepel
- SZÍNÉSZ (Név, Nem, Születésnap)
 - => **SZÍNÉSZ (Id, Név, Nem, Születésnap)**

<u>Id</u>	Név	Nem	Születésnap
1	Sylvester Stallone	M	1946.07.06
2	Vera Miles	F	1929.08.23
3	Liam Neeson	M	1952.06.07
4	Anthony Perkins	M	1932.09.12
5	Finn Carter	F	1960.03.09



Adatbázis példa 3 – 4

➤ Film adatbázis

- Studiók, filmek, színészek; ki – miben szerepel
- SZEREPEL(Név, Születésnap, Cím, év)
 - => **SZEREPEL(FilmId, SzínészId)**

FilmId	SzínészId
4	1
1	2
5	3
1	4
2	5