

Adatbázis-kezelő rendszerek

I.

dr. Siki Zoltán

Adatbázis fogalma

- adatok valamely célszerűen rendezett, szisztéma szerinti tárolása

Az informatika elterjedése előtt is számos adatbázis létezett pl.

- Vállalati személyzeti nyilvántartás
- Könyvtári kartoték rendszerek

Manapság adatbázis-kezelő rendszerekről beszélünk, melyekbe beleértjük az adatok mellett az adatok kezelésére szolgáló szoftvert.



Történeti áttekintés

i.e. az írásbeliség kialakulásával jöttek létre az első „adatbázisok”

1846 lyukszalag

1884 lyukkártya

1910- kartotékrendszerek

1946 első elektronikus számítógép (ENIAC 80 byte memória)

1950 első mágnesszalag egység

1951 első kereskedelmi számítógép

1. generációs programnyelvek (gépi kódú programozás)

2. generációs programnyelvek (Assembler nyelvek, processzor függő)

1956 első HD (IBM RAMAC 4 MB)

3. generációs programnyelvek (FORTRAN 1954, Algol 1960, C 1970)

1969 Unix operációs rendszer

1976 első 5^{1/4}” floppy egység

4. generációs nyelvek (SQL 1970, PL/SQL 1991, MATLAB 1984)

1980 első CD lemez

5. generációs programnyelvek (Prolog, Lisp)

Objektum orientált nyelvek (SmallTalk 1972, C++ 1981, Java 1994)

Script nyelvek (Perl 1987, Python 1991, PHP 1995, JavaScript)

Miért kell adatbázisról tanulnia egy térinformatikusnak?

- Az adatok egyre szélesebb körét adatbázisokban találjuk meg (már a térinformatikai)
- A térinformatikusnak szót kell értenie az adatbázis tervező, programozó szakemberekkel, hogy a számukra szükséges adatbázis környezet álljon elő
- Tudják mit várhatnak el egy adatbázis-kezelő rendszertől

Adatbázis-kezelők szembeni követelmények

- Függetlenség az aktuális hardver konfigurációtól
- Függetlenség az adatelérés módjától
- Függetlenség az adatstruktúráktól

Fájl alapú adatkezelő megoldás

Háttértár

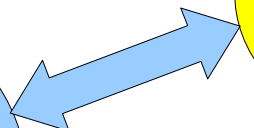


Felhasználói program

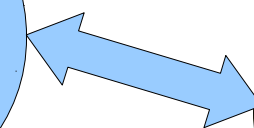
Adatok tárolási logikáját a program ismeri (fizikai és logikai), az adatszerkezet módosulása esetén a programot is módosítani kell. A konkurens adatelérést a operációs rendszerek nem támogatják. Jogosultságok kezelése?

Adatbázis alapú megoldás

DBMS



Felhasználói program



Felhasználói program

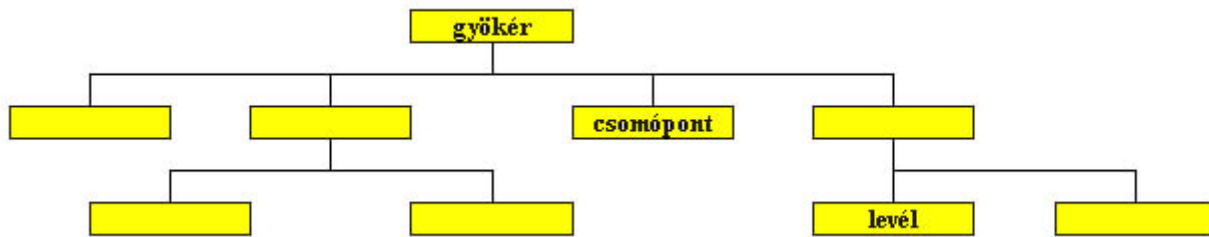
Adatok tárolási logikáját a DBMS ismeri (fizikai és logikai). A DBMS-ben egy központi helyen lehet megoldani a konkurens adat elérés és a jogosultságok kérdését.

Adatbázis-kezelők előnyei

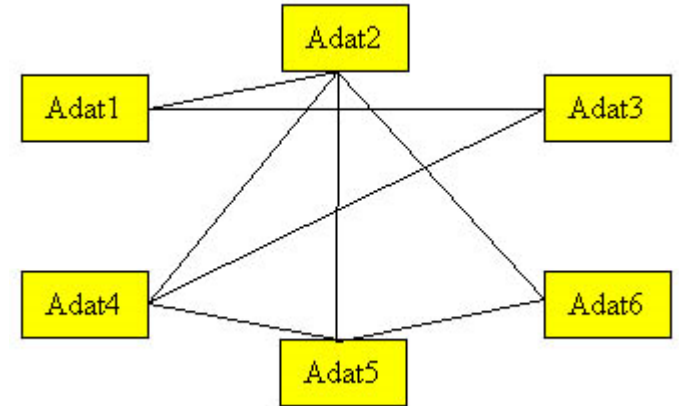
- Elfedi az adatok fizikai tárolási szerkezetét, a felhasználóknak, programoknak csak a logikai adatszerkezetet kell ismernie
- Hatékony adatelérést biztosítanak
- Adatintegritás ellenőrzése, jogosultságok kezelése
- Konkurens adatelérés és adatvesztés elleni védelem (hardver hiba esetén is)
- Lerövidíti a programfejlesztés idejét

Adatbázis modellek

Hierarchikus (1:n kapcsolatok)



Hálós adatmodell (n:m kapcsolatok)



Relációs (1:1 kapcsolatok)

| name | birth | address | ... |
|-------|-------|---------|-----|
| Kis | 1996 | | |
| Nagy | 1959 | | |
| Közép | 1988 | | |
| ... | | | |

NoSQL (strukturálatlan adatok)

Dokumentum tárolók
Gráf adatbázisok (hálós modell)
Kulcs – érték adatbázisok

Relációs adatbázis modell (Codd 1970)

1:1 típusú kapcsolatok tárolása

Komplex adatszerkezetek, és kapcsolatrendszer helyett a kapcsoló adatok többszörös tárolása

Az 1:1 kapcsolatban lévő adatok rekordokat alkotnak, az azonos típusú rekordokat táblázatokba (relációkba) rendezzük

| | | | | |
|-----|--------|--|------|--|
| | Oszlop | | | |
| Sor | | | | |
| | | | Mező | |

tábla = reláció
oszlop = attribútum

- Az adatbázis relációk halmaza
- A relációkhoz egyedi nevet rendelünk
- A reláció attribútumaihoz egyedi neveket rendelünk (a táblázaton belül)
- Az attribútumokhoz a tárolt adattípust megadjuk (szöveg, szám, dátum)
- Egy reláción belül nem lehet két egyforma rekord, minden rekordnak megkülönböztethetőnek kell lennie a többitől
- Egy reláció egy mezője nem tartalmazhat egy másik relációt
- Adatbázis séma – a relációk szerkezetének leírása

CODD 12 szabálya

1. Információs szabály: minden információt táblázatokban szereplő értékekkel fejezünk ki.
2. Garantált elérési szabály: minden adat logikailag elérhető a táblázatnév, az oszlopnév és az elsődleges kulcs értékének megadásával.
3. Hiányzó információ szabálya: a null értékek megjelenítése független az adat típusától.
4. Rendszerkatalógus szabálya: az adatbázis logikai leírása úgy áll rendelkezésre, hogy arra jogosult felhasználó ugyanolyan módon kérdezheti le, mint az adatbázisban tárolt adatokat

CODD 12 szabálya folyt.

5. Széleskörű nyelv szabálya: egy jól definiált szintaxissal bíró, karaktersorozatokkal leírható, magas szintű relációs nyelv áll rendelkezésre, amelynek segítségével leírhatók a következők: adat és nézet definíciója, integritási feltételek, interaktív és programozható adatkezelés, tranzakció-kezdés, -érvényesítés és -elutasítás.
6. Nézetmódosítás szabálya: a rendszer képes végrehajtani minden elméletileg lehetséges módosítást a nézeteken.

CODD 12 szabálya folyt.

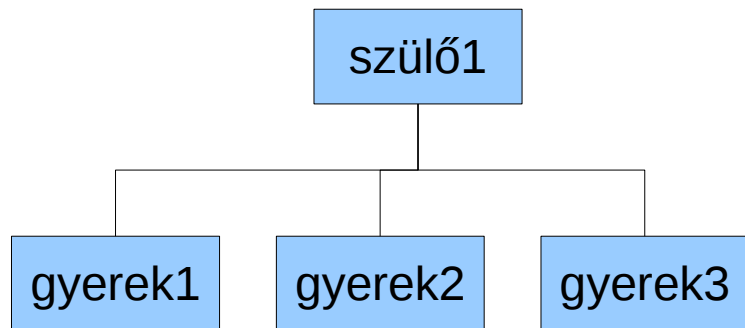
7. Halmazszintű módosítás szabálya: a rendszer képes egész táblázatok együttes kezelésére.
8. Fizikai adatfüggetlenség szabálya: a felhasználó és alkalmazói programok tevékenysége független a fizikai adattárolásban vagy adatelérési módszerekben bekövetkező változásoktól.
9. Logikai adatfüggetlenség szabálya: a felhasználó és alkalmazói programok tevékenysége független az adatbázis táblázatainak logikai struktúrájában bekövetkező változásoktól, feltéve, hogy azok nem járnak információvesztéssel.

CODD 12 szabálya folyt.

10. Integritásfüggetlenségi szabály: az egyed- és hivatkozásintegritási feltételek az 5. szabály által meghatározott relációs nyelv segítségével definiálhatók.
11. Elosztásfüggetlenségi szabály: a felhasználó és alkalmazói programok tevékenysége független az adatok elhelyezésétől.
12. Megkerülés-tiltási szabály: egy alacsonyabb szintű eljárásorientált programnyelv használatával nem lehet megkerülni az integritási vagy biztonsági feltételeket.

Relációs adatbázis modell folyt.

1:n típusú kapcsolatok leképezése a relációs modellben
(n db. 1:1 kapcsolat)

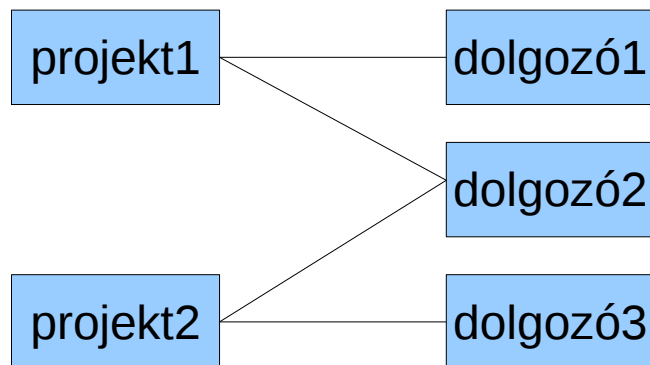


szülő-gyerek

| szülők | gyerekek |
|--------|----------|
| szülő1 | gyerek1 |
| szülő1 | gyerek2 |
| szülő1 | gyerek3 |

Relációk és attribútumok elnevezése legyen logikus

n:m típusú kapcsolatok leképezése a relációs modellben
(n x m db. 1:1 kapcsolat)



projekt-dolgozó

| projektek | dolgozók |
|-----------|----------|
| projekt1 | dolgozó1 |
| projekt1 | dolgozó2 |
| projekt2 | dolgozó2 |
| projekt2 | dolgozó3 |

Néhány elterjedtebb relációs relációs adatbázis-kezelő rendszer (RDBMS)

Kereskedelmi szoftverek:

Oracle, MS SQL Server, DB2

Korlátozott ingyenes változatok pl. MS SQL Server Express



Nyíltforrású szoftverek:

PostgreSQL, MySQL, MariaDB, SQLite

Fizetős változatok pl. MySQL Server Enterprise



Néhány elterjedtebb NoSQL adatbázis-kezelő rendszer

Apache Cassandra

MongoDB

Redis

