Adatbázisok 1.
Bevezetés

Adatbázisok használata

Az előadás diasoraihoz felhasználtam az ELTE IK-s munkatársaim által korábban kidolgozott munkákat, továbbá *Dr. Siki Zoltán* (BME), *Jeffrey D. Ullman* (Stanford University) és *Andy Pavlo* (Carnegie Mellon University) ötleteit, jegyzeteit, diasorait

Elérhetőségek, információk

- Előadó
  - Szalai-Gindl János Márk (ELTE IK, Információs Rendszerek Tanszék)
  - Email: szalaigindl@inf.elte.hu
  - Szoba: 2.507 (déli tömb)
  - Honlap: https://szalaigj.web.elte.hu

Elérhetőségek, információk

- Előadások időpontjai, helye:
  - 1. csoport: péntek 8:25-9:55, D.T. 0-821 (Bolyai János terem), teljesen jelenléti formában!
  - 2. csoport: kedd 12:25-13:55, É.T. -1.75 (Konferenciaterem), teljesen jelenléti formában!
- Az előadás weboldala:
  - https://canvas.elte.hu/courses/43350
- Irodalom: Jeffrey D. Ullman, Jennifer Widom: Adatbázisrendszerek - Alapvetés, 2. kiadás, Panem, Budapest, 2008.

Tantárgy tematikája

- Bevezetés és relációs adatmodell
- Relációs algebra lekérdezések optimalizációja
- SQL
- Relációs adatbázis tervezés
- Egyed kapcsolat modell
- Objektum-relációs ismeretek
- XML séma és lekérdezőnyelvek
- Indexek

Vizsga

- **Számonkérés az előadás teljesítéséhez: írásbeli vizsga, amely a vizsgaidőszakban lesz esedékes**
- Ennek előfeltétele: legalább elégséges (2) gyakorlatjegy
- A vizsga az egész féléves anyagra fog épülni, elméleti és gyakorlati feladatokból lesz összeállítva a kérdéssor
- Az előadásokon elhangzott és a diasorokon szereplő definíciók, összefüggések és a belőlük levonható következtetések, továbbá az ismertetett módszerek együttesen alkotják a tantárgy anyagát.
- Szaknyelvi követelmények: (a diasorokon is szereplő) angol nyelvű szakkifejezések számonkérése **szódolgozat** formájában
- A részleteket az utolsó előadáson fogjuk pontosítani

Plagiarizmus figyelmeztetés!!!

- A tárgyhoz kapcsolódó összes munkát (az ellenőrző kérdéseket, a zárthelyik, a házi feladatok és a vizsgadolgozat megírását) **ÖNÁLLÓAN** kell elvégezni!

Mi is az az adatbázis?

- Hétköznapi értelemben:
- Adatbázis (*database*): összefüggő adatok praktikusan rendezett gyűjteménye és ezeknek valamilyen rendszerben való tárolása, amely a való világot valamilyen szempontból modellezi
- Adathalmaz ≠ adatbázis
- Adatbázisok korábban nem feltétlen számítógépen voltak, pl. a könyvtári kartoték rendszer
- Az adatbázisok ma már az élet számos területén alapvető fontossággal bírnak (Google, Amazon, Flickr, Youtube stb.)

Adatbázis példa

- Tegyük fel, hogy egy olyan adatbázist szeretnénk, amely az egyetemi oktatásról készít modellt.
- A dolgok, amelyeket tárolnunk kell:
  - Infók az oktatókról
  - Tárgyakról
  - Termek használatáról

Adatbázis példa

- Oktatókhoz tartozó információk: *azonosítószám*, *név*, *beosztás*
- Tárgyakról: *név*, *oktató azonosítószám*, *bevezetés éve*
- Teremhasználatról: *épület*, *szobaszám*, *időköz*, *tárgynév*
- Egy tárgyat egy vagy több oktató oktathat, egy oktató egy vagy több tárgyat oktathat
- Egy időben egy teremben csak egy tárgyat oktathatnak

Ötlet: nyers fájlok

- Tároljuk az adatokat például CSV fájlokban
- Használjunk egy-egy fájlt az oktatókra, a tárgyakra és a teremhasználatra!
- Az alkalmazásunknak majd szintaktikai elemzést kell végezni minden egyes alkalommal a fájlokon, amikor olvasni/frissíteni fogja a bejegyzéseket

Oktatók(azon., név, beoszt.)

```
1,"Nagy Morgána","docens"
2,"Kovács István","adjunktus"
…
```

Tárgyak(név, azon., év)

```
"Adatbázisok", 1, 1998
"Hálózatok", 2, 2000
…
```

Ötlet: nyers[illegible]ok

- Tároljuk az adatokat [illegible]SV fájlokban
- Használjunk egy-egy fá[illegible]ókra, a tárgyakra és a teremhasználatra!
- Az alkalmazásunknak majd sz[illegible]lemzést kell végezni minden egyes alkalommal a fájlokon, am[illegible]i/frissíteni fogja a bejegyzéseket

Keressük ki azt az évet, amikor az Adatbázisok tárgyat bevezették!

Oktatók(azon., név, beoszt.)

```
1,"Nagy Morgána","docens"
2,"Kovács István","adjunktus"
…
```

Tárgyak[illegible]év)

```
"Adatbázis[illegible]
"Hálózatok", 2[illegible]
…
```

Ötlet: nyers fájlok

```
for line in file:
    record = parse(line)
    if "Adatbázisok" == record[0]:
        print int(record[2])
```

Nyers fájlok: adatintegritás

- Hogyan tudjuk azt biztosítani, hogy egy tárgy az összes teremhasználatnál ugyanabban formában jelenjen meg?
  - Pl. adatbázis vagy adatbázisok vagy AB stb.
- Mi történik, ha valaki véletlenül felülírja a tárgy bevezetésének évét egy érvénytelen sztringgel?
  - Pl. 1993 helyett ezerkilencszázkilencvenhárom
- Hogyan, milyen formában tároljuk, ha több oktató oktat egy tárgyat?

Nyers fájlok: implementáció

- Hogyan keresünk meg egy adott bejegyzést?
- Mi van, ha egy új alkalmazást akarunk készíteni, amelyik ugyanazt az adatbázist használja?
- Mi van, ha két szál egyidőben ugyanabba a fájlba próbál írni?

Nyers fájlok: tartósság

- Mi van, ha a gép elszáll, miközben éppen egy bejegyzést frissítünk?
- Mi van, ha a magas rendelkezésre állás érdekében többszörözni/másolni akarjuk ugyanazt az adatbázist több gépen?

Mi is az az adatbázis?

- Az adatbázisnak új meghatározást adunk:
- Adatbázis (*database*): olyan adatok együttese, amelyet egy adatbázis-kezelő rendszer (DBMS: *Database Management System*) kezel.
- Az általános célú DBMS megoldandó feladatai:
  - új adatbázisok létrehozása, ezek logikai szerkezetének, sémájának definiálása, adatdefiníciós nyelv (DDL, *Data Definition Language*),
  - adatok lekérdezése, módosítása, adatmanipulációs nyelv (DML, *Data Manipulation Language*),
  - nagyméretű adatok hosszú időn keresztül történő tárolása, adatok biztonsága meghibásodásokkal, illetéktelen hozzáférőkkel szemben, hatékony adatbázis-hozzáférés,
  - egyszerre több felhasználó egyidejű hozzáférésének biztosítása.
- Példa: banki rendszerek (még a hőskorszakból).

Történelem I.

- Ókori „adatbázisok”: kőtáblák, papirusz tekercsek
- Később kartoték rendszerek
- 60-as évektől az adattárolás a számítógépek mágneses tárjainak felhasználásával történik
  - Az egyik első DBMS: Integrated Data Store (IDS), 1964, GE 235 mainframe
- Eleinte adatbázis alkalmazások születtek egyedi feladatokra
- Később a fejlesztők elkezdtek törekedni arra, hogy minél általánosabb formában történjenek az adatokkal kapcsolatos műveltek → szabványosítás

Történelem II.

- Legelőször olyan helyzetekben alkalmaztak DBMS-t (adatbáziskezelő-rendszert), ahol sok kicsi adatelem szerepelt, sokan akartak hozzáférni az adatokhoz egyszerre és gyakoriak voltak a módosítások (a banki rendszerek mellett még pl. repülőgép-helyfoglalás).
- Az első modellek: hierarchikus és hálós adatmodell, az előbbi fa-, utóbbi gráfszerkezetben ábrázolta az adatokat.
- A modelleket végül szabványosították CODASYL jelentésben (Committee on Data Systems and Languages).
- Hátrányuk: nem támogattak magasabb szintű lekérdezőnyelvet (pl. add meg, hogy Sziszi számláin összesen mennyi pénz van), hanem csak a mutatók mentén pontról-pontra lehetett haladni a gráfban, minden lekérdezés külön programkódot igényelt.

Történelem III.

- Ted Codd 1970-ben publikált egy cikket, amelyben azt javasolta, hogy az adatokat táblázatokban, relációkban tárolják.
- Az ötlet, habár igen egyszerűnek tűnik, meglepően sikeresnek bizonyult, a 90-es évek elejére a relációs adatbázisok lettek a legelterjedtebb rendszerek.
- Egyik fő előnyük, hogy lehetővé teszik az adatok magasszintű programnyelvvel, strukturált lekérdezőnyelvvel: SQL (*Structured Query Language*) történő lekérdezését:
  - pl. Sziszi számláin összesen mennyi pénz van. Kell egy művelet, ahol a megfelelő sorokat választjuk ki: név = 'Sziszi', kell egy másik, ahol a kívánt oszlopot (összeg), végül az oszlopban lévő számokat összegezni kell. Az SQL-ben ez három utasítás, s az implementációval nem kell törődnünk, sem azzal, hogy az adatokat valójában miként tárolja a rendszer.

Egy példa...

név	számla_azon	összeg
Sziszi	SZ01	45000
Peti	SZ02	543000
Sziszi	SZ03	120000

Mostani irányvonalak

- A rendszerek már nem csupán egyszerű adatok tárolására, hatékony lekérdezésére stb. képesek, hanem igen összetett adatokat is hatékonyan kezelnek (pl. térinformatikai rendszerek).
- Egyre nagyobb mennyiségű adatot kell eltárolni. Ma már egyáltalán nem számít kirívó esetnek, ha egy vállalat terabájtnyi adatot (10^{12}) tárol, de vannak petabájtnyi (10^{15}) adattal dolgozó rendszerek.
- Több adatbázis fölé egy „összefogó” adatbázis felépítése (adattárház, middleware).