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

- Előadások időpontja, helye:
  - hétfő 8:30-10:00 (1. csop.) és kedd 8:30-10:00 (2. csop.), MS Teams felület
- 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
- Az előadás weboldala: https://canvas.elte.hu/courses/16792
- Irodalom: Jeffrey D. Ullman, Jennifer Widom: Adatbázisrendszerek - Alapvetés, 2. kiadás, Panem, Budapest, 2008.
- **Az előadásokról és a gyakorlatról maximum 3 hiányzás megengedett**

Tantárgy tematikája

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

Feladatterv áttekintése

Feladat	**A jegy összetétele**	**Várható ideje**
1. Zárthelyi	25% (gyakorlat)	4. gyakorlaton
2. Zárthelyi	25% (gyakorlat)	8. gyakorlaton
3. Zárthelyi	25% (gyakorlat)	12. gyakorlaton
Ellenőrző kvízkérdések	25% (gyakorlat)	várhatóan minden héten a szorgalmi időszakban a 2. gyakorlattól kezdve
Vizsgadolgozat	100% (előadás)	vizsgaidőszakban

Ponthatárok

- Mind a gyakorlat, mind az előadás esetében a táblázatban látható módon alakulnak a jegyek
- Gyakorlatjegy százalék =

(ZH1_szerzett_pontok / ZH1_maxpontszám) * 25
+
(ZH2_szerzett_pontok / ZH2_maxpontszám) * 25
+
(ZH3_szerzett_pontok / ZH3_maxpontszám) * 25
+
(kvízeken_szerzett_pontok / kvízeken_szerezhető_összpontszám) * 25

Százalék	Érdemjegy
0 – 40 %	Elégtelen (1)
40 – 55 %	Elégséges (2)
55 – 70 %	Közepes (3)
70 – 85 %	Jó (4)
85 – 100 %	Jeles (5)

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.
- A részleteket később fogjuk pontosítani a járványhelyzet miatt

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: 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 egy digitális zeneboltról készít modellt.
- A dolgok, amelyeket tárolnunk kell:
  - Infók a előadókról
  - Albumok, amelyeket kiadtak az előadók
  - Műsorszámok az albumokról

Adatbázis példa

- Előadókhoz tartozó infók: név, kezdési év, származási ország
- Albumokról: név, kiadás éve
- Műsorszámokról: név, sorszám
- Egy albumon egy vagy több előadó szerepelhet, és egy előadó egy vagy több albumon szerepelhet
- Egy albumon egy vagy több műsorszám is lehet
- De egy műsorszám csak egy albumon jelenik meg.

Ötlet: nyers fájlok

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

Előadók(név, év, ország)

```
"Wu Tang Clan",1992,"USA"
"Notorious BIG",1992,"USA"
"Ice Cube",1989,"USA"
…
```

Album(név, előadó, év)

```
"Enter the Wu Tang", "Wu Tang Clan", 1993
"St.Ides Mix Tape", "Wu Tang Clan", 1994
…
```

Ötlet: nyers fájlok

- Tároljuk az adatokat p l CSV fájlokban
- Használjunk egy-egy fájlt a előadókra, az albumokra és a műsorszámokra!
- Az alkalmazásunknak majd szint i elemzést kell végezni minden egyes alkalommal a fájlokon, amik asni/frissíteni fogja a bejegyzéseket

Keressük ki azt az évet, amikor Ice Cube debütált!

Előadók(név, év, ország)

"Wu Tang Clan",1992,"USA"
"Notorious BIG",1992,"USA"
"Ice Cube",1989,"USA"
…

Album(ne dó, év)

"Enter the W ". "Wu Tang Clan", 1993
"St.Ides Mix Tape u Tang Clan", 1994
…

Ötlet: nyers fájlok

```
for line in file:
    record = parse(line)
    if "Ice Cube" == record[0]:
        print int(record[1])
```

Nyers fájlok: adatintegritás

- Hogyan tudjuk azt biztosítani, hogy egy előadó az összes albumánál ugyanabban formában jelenjen meg?
  - Pl. O'Shea Jackson vagy Ice Cube, Wu Tang Clan vagy Wu-Tang Clan stb.
- Mi történik, ha valaki véletlenül felülírja az album évét egy érvénytelen sztringgel?
  - Pl. 1993 helyett ezerkilencszázkilencvenhárom
- Hogyan, milyen formában tároljuk, ha egy albumon több előadó is szerepel?

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: 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 (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 legelterjedtebbek rendszerek.
- Egyik fő előnyük, hogy lehetővé teszik az adatok magasszintű programnyelvvel, 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.
- Peer-to-peer fájlmegosztó rendszerek.
- Több adatbázis fölé egy „összefogó” adatbázis felépítése (adattárház, middleware).