

Adatbázisok I.

1

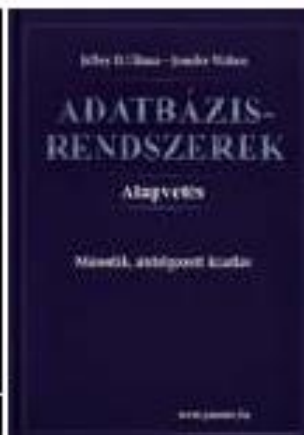
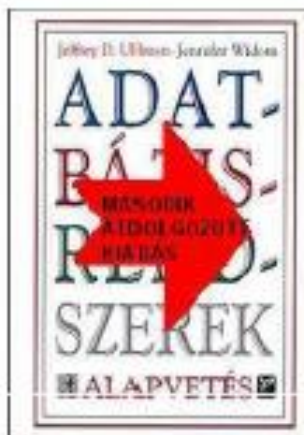
Jánosi-Rancz Katalin Tünde

tsuto@ms.sapientia.ro

327A

Ajánlott irodalom

BSc Adatbázisok-1

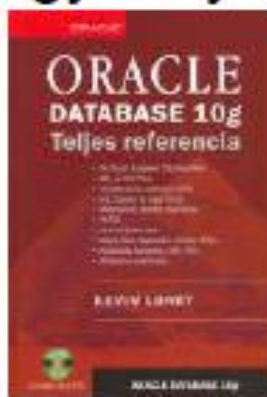
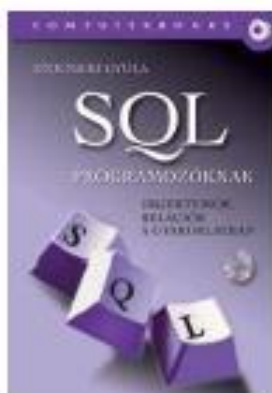


TANKÖNYV:
Ullman-Widom:
Adatbázisrendszerek
Alapvetés (Második
átdolgozott kiadás)
(a kék könyv, nem a fehér)

Erre épül BSc-n
Adatbázisok-2



További magyar nyelvű irodalom, SQL és PL/SQL-hez



Ajánlott magyar nyelvű irodalom

- Jeffrey D. Ulmann, Jenifer Widom: Adatbázisrendszerek Alapvetés, Második, átdolgozott kiadás, Panem kiadó, Budapest, 2009
- Varga Ibolya: Adatbázisrendszerek, Egyetemi Kiadó, Kolozsvár, 2005
- <http://people.inf.elte.hu/kiss/>
- <http://people.inf.elte.hu/sila/>
- <http://15415.courses.cs.cmu.edu/fall2015/slides>

Jegy megállapítás

- Laboron a jelenlet kötelező, előadáson IS.
- Labor tevékenység: 10 pont
- Laborvizsga csak évközben tehető le, pótszesszióban nem. 10 pont
- Projekt: 30 pont. Fontos: projektet pótszesszióban nem lehet leadni.
- Kizáró jellegű interjú kérdések!!!
- Hivatalból 10 pont
- Írásbeli: 40 pont
 - 2 részből áll:
 - tervezés (egyed-kapcsolat, illetve normalizálás) -20 pont (átmenőre legalább 10 pontot el kell érni)
 - SQL nyelv - 20 pont (átmenőre legalább 10 pontot el kell érni)
- Írásbeli vizsgára jöhet:
 - - aki legtöbb 3 laboron hiányzott,
 - - elért legalább 5 pontot a labortevekenységből,
 - - elért legalább 5 pontot a laborvizsgán
 - - 15 pontot a projektnél.
- Pótszesszióban csak az írásbelit lehet megismételni!

Legjobb Online tanulási lehetőségek

- **W3Schools – “SQL Tutorial”**
 - <http://www.w3schools.com/sql/default.asp>
- **Vertabelo Academy**
 - <https://academy.vertabelo.com/>
- **Codecademy – “Learn SQL”**
 - <https://www.codecademy.com/learn/learn-sql>
- **Khan Academy – “Intro to SQL”**
 - <https://www.khanacademy.org/computing/computer-programming/sql>
- **SQLZoo**
 - <http://sqlzoo.net/>
- **Tutorials Point – “Learn SQL”**
 - <http://www.tutorialspoint.com/sql/>
- **Udacity – “Intro to Relational Databases”**
 - <https://www.udacity.com/course/intro-to-relational-databases--ud197>
- **SQL Problems and Solutions**
 - <http://www.sql-tutorial.ru/>
- **Tuts+ – “SQL for Beginners”**
 - <https://code.tutsplus.com/articles/sql-for-beginners--net-8200>

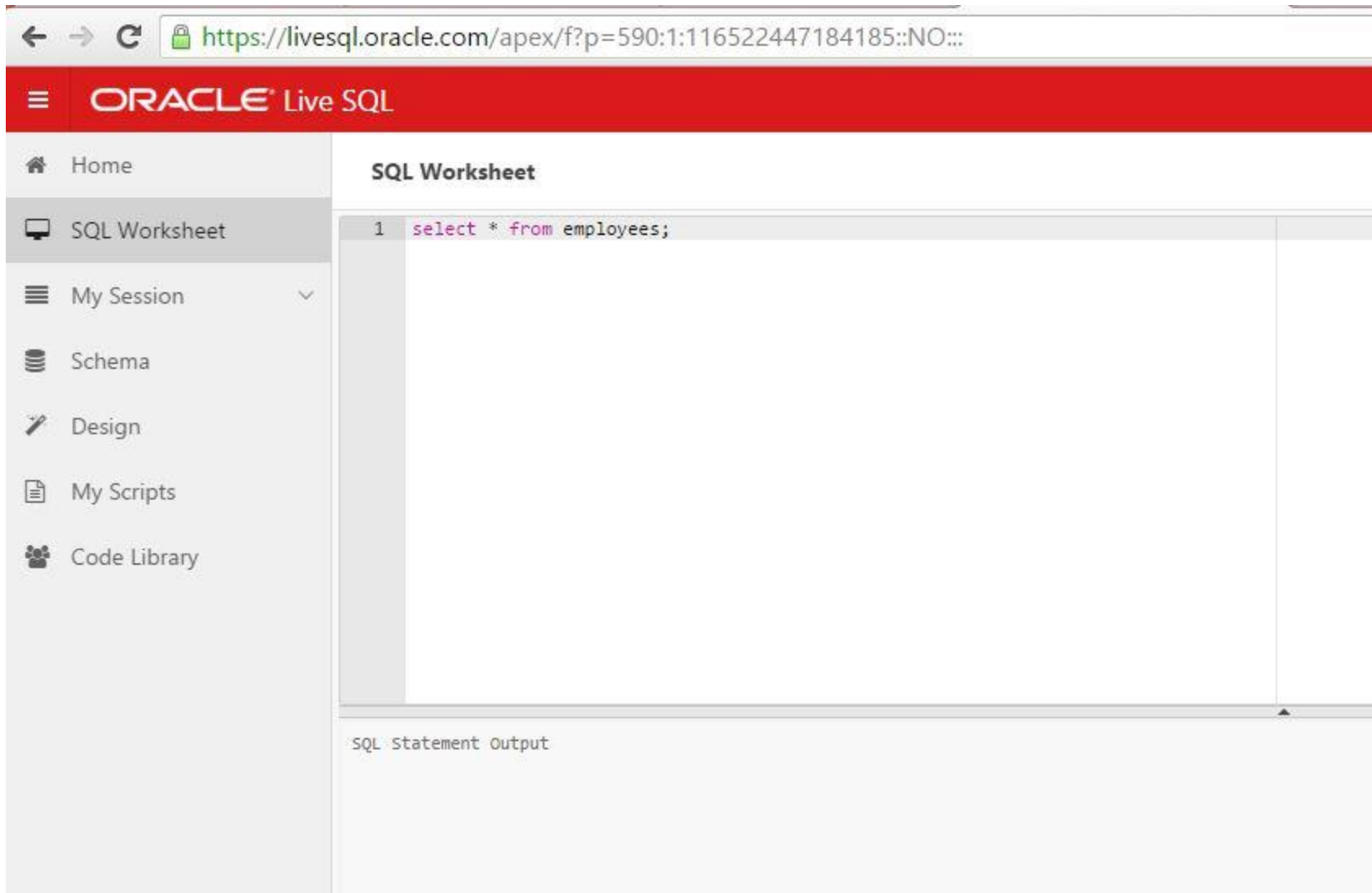
Legjobb Online tanulási lehetőségek -2

- **Essential SQL**
 - <http://www.essentialsql.com/>
- **Learn SQL The Hard Way**
 - <https://learncodethehardway.org/sql/>
- **Udemy – SQL Tutorials**
 - <https://www.udemy.com/courses/search/?q=sql&src=ukw&price=price-free&lang=en>
- **MySQL Tutorial**
 - <http://www.mysqltutorial.org/>
- **SQLCourse**
 - <http://www.sqlcourse.com/>
- **Database Journal**
 - <http://www.databasejournal.com/>
- **Stanford Database Course**
 - <https://lagunita.stanford.edu/courses/Home/Databases/Engineering/about>
- **Oracle Database SQL reference**
 - http://docs.oracle.com/cd/B19306_01/server.102/b14200/toc.htm

Online gyakorlási lehetőség

▶ Oracle Live SQL

- <https://livesql.oracle.com//>



The screenshot displays the Oracle Live SQL web interface. The browser's address bar shows the URL <https://livesql.oracle.com/apex/f?p=590:1:116522447184185::NO::>. The page features a red header with the Oracle logo and the text "ORACLE Live SQL". A left-hand navigation menu includes options for Home, SQL Worksheet (which is currently selected), My Session, Schema, Design, My Scripts, and Code Library. The main content area is titled "SQL Worksheet" and contains a single line of SQL code: `1 select * from employees;`. Below the code editor, there is a section labeled "SQL Statement Output" which is currently empty.

Online gyakorlási lehetőség

- **SQL Fiddle**
- <http://sqlfiddle.com/>

The screenshot displays two web-based tools used for database development. The top window is SQL Fiddle, showing SQL code for creating tables 'client' and 'product', and a query to select a record from the 'client' table. The bottom window is Vertabelo, showing a database model structure with tables like 'book', 'author', 'purchase', and 'customer', along with their relationships and properties.

SQL Fiddle Code:

```
1 -- tables
2 -- Table client
3 CREATE TABLE client (
4   id int NOT NULL,
5   full_name varchar(255) NOT NULL,
6   email varchar(255) NOT NULL,
7   CONSTRAINT client_pk PRIMARY KEY (id)
8 );
9
10 -- Table product
11 CREATE TABLE product (
12   id int NOT NULL,
13   product_category_id int NOT NULL,
14   sku char(10) NOT NULL,
15   name varchar(255) NOT NULL,
16   price decimal(12,2) NOT NULL,
17   description varchar(1000) NOT NULL,
18   image blob NOT NULL
19   CONSTRAINT
20 );
21
22 -- Table produ
23 CREATE TABLE pr
24   id int NOT
25   name varchar
```

SQL Fiddle Query:

```
1 SELECT * FROM client WHERE id=7;
```

Vertabelo Model Structure:

- Tables: archived_purchase, archived_purchase_i..., author, book, book_author, book_comment, customer, purchase, purchase_item, purchase_status
- References: Sequences, Text notes, Views, Subject areas
- Problems: 7 errors - final model (Edit mode)

Table Properties (Purchase):

Primary data	
Name:	purchase
Comment:	

Columns	
Name	Type
id	int
customer_id	int
purchase_sta	int
create_ts	timestamp
update_ts	timestamp
update_info	text

Table Problems:

- Errors (0)
- Warnings (0)
- Hints (0)

- **Vertabelo**
- <http://www.vertabelo.com/>

Konferenciák

- ▶ Adatbázisokkal foglalkozó legrangosabb évente megrendezett konferenciák:
- ▶ VLDB (Very Large Databases)
- ▶ PODS (Symposium on Principles of Database Systems)
- ▶ ICDT/EDBT (International Conference on Database Theory/International Conference on Extending Database Technology)
- ▶ ADBIS (East-European Conference on Advances in Databases and Information Systems)
- ▶ DEXA (Conference on Database and Expert Systems Applications)
- ▶ folyóiratok:
- ▶ ACM Transactions on Database Systems

Célkitűzések

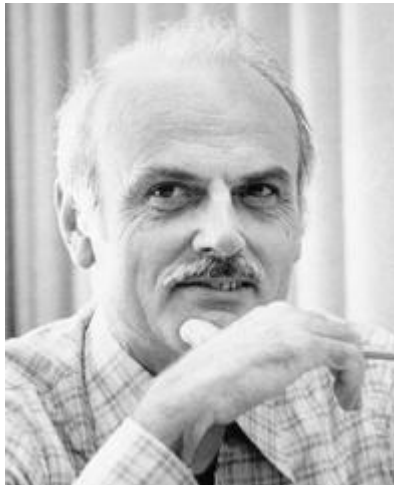
- Az adatbázisok tanulmányozása három nézőpontból:
 - adatbázis felhasználó
 - adatbázis tervező és manager
 - adatbázis-kezelő rendszer implementáló
- Megérteni az adatbázis-kezelő rendszerek (ABKR=DBMS) működési elvét és lekérdező nyelveit (SQL)

Tematika

1. **Adatbázis-kezelő rendszerek** általános jellemzői.
2. **A relációs adatmodell**, a relációs algebra műveletei, használata
3. **Az SQL nyelv részei** (ORACLE specifikusan):
 - DDL, DML QL, triggererek, jogosultságok, megszorítások, PL/SQL, függvények, procedúrák, cursorok használata, programozás,
4. **Adatmodellezés**, egyed-kapcsolat modell, az E/K diagram átalakítása relációs adatmodellé.
5. Tárolás, Lekérdezés végrehajtás, Konkurencia vezérlés, Helyreállítás

Történelmi háttér

- A régi adatbázisok kérdései relevánsak ma is!
- A mai adatbázisok legtöbb ötlete nem új!
- 1960-as évek:
 - Charles Bachmann kifejlesztette az első ABKR-t Honeywell-ben (IDS)
 - ❖ Háló modell, amelyben a kapcsolatokat gráfokkal reprezentálták
 - Az IBM kifejlesztette az első kereskedelmi célú ABKR-t (IMS)
 - ❖ Hierarchikus modell, amelyben a kapcsolatokat fa-struktúrával ábrázolták
- 1970 Ted **Codd** az IBM-nél definiálta a **relációs adatmodellt**
 - Az adatbázist egyszerű adatstruktúrában, táblákban tárolja, SQL –el kérdezi le



Codd

1970-es évek

- ▶ 1976: Peter Chen definiálta az Egyed-Kapcsolat (ER) modellt
- ▶ System R – IBM
- ▶ INGRES – Berkeley
- ▶ **ORACLE** – Larry Ellison

ORACLE®



Lary Ellison (7-ik leggazdagabb ember a világon!)

1980-as évek

- ▶ DB2 - IBM
- ▶ Az SQL standardizálása ISO szabványként, igazi fejlődési periódus
- ▶ **Oracle** a nyerő a piacon!!!
- ▶ Postgres
- ▶ Informix
- ▶ Sybase



1980-as évek -Objektum Orientált Adatbázisok

```
class Employee{  
    int id;  
    string name;  
    int salary;  
    string phone[];  
}
```

- ▶ kevés létezik már azokból amit a 80-as években fejlesztettek
- ▶ Hátrány: komplex lekérdezés, nincs standard API
- ▶ lett helyette JSON és XML

```
{  
    "id": 123,  
    "name": "Jim",  
    "salary": 56000,  
    "phone": {  
        "00-77-88-99-888",  
        "00-77-88-99-999"  
    }  
}
```

1990-es évek

- ▶ SQL Server - Microsoft
- ▶ MySql
- ▶ Postgres készít SQL szuportot



2000-es évek –Internet bumm

- ▶ az eddig bemutatott ABKR-ek drágák lettek
- ▶ sok cég saját ABKR-t készít

2000-es évek – Adattárházak

- ▶ Data Warehouse – OLAP DBMS
- ▶ adatelemzésre, Lynuxban
 - ▶ osztott
 - ▶ képes relációs/SQL
 - ▶ zárt forráskód



2000-es évek –NoSQL

- ▶ **NOT ONLY** SQL
- ▶ magas elérhetőségre és nagy skálázhatóságra fókuszál
- ▶ dokumentum, gráf, kulcs-érték
- ▶ nyílt forráskód
 - ▶ séma nélküli
 - ▶ nem biztosítja az ACID-ot
 - ▶ custom API-kat használt SQL helyett
 - ▶ most majdnem mindenik biztosít SQL szuportot (kivételet a MongoDB és a Redis)



2010-es évek –NewSQL

- ▶ biztosítja ugyanazt a teljesítményt mint a NoSQL, de nem rugja fel az ACID elveket
 - ▶ képes relációs/SQL
 - ▶ többnyire osztott
 - ▶ zárt forráskód



Napjainkban–Hibrid rendszerek

- ▶ Hybrid Transactional Analytical Processing
 - ▶ OLTP és OLAP lekérdezéseket hajt végre gyorsan
 - ▶ osztott
 - ▶ nyílt és zárt forráskód
 - ▶ képes relációs/SQL



Történelem következtetések

- ▶ Történelem ismétli önmagát
 - ▶ 70-80-as években IBM volt az élen, most a Google a trendsetter
 - ▶ Oracle kölcsönvesz ötletket másoktól, sokat tesztel mielőtt kiad vmit és ezért nagyon megbízható, nagyon széleskörben használt
- ▶ A relációs adatmodell nyer az operációs adatbázisok közül, amikor tranzakciókezelés szükséges. Amikor pl. gépi tanulás kell akkor más modelleket kell használni

Az adatbázis-kezelés alapjai

Adat és információ

Adatbázis: logikailag összefüggő adatok összessége

Az adatbázisok a legtöbb számítógépes alkalmazás alapvető elemei!

Példák:

- Napi szinten: Google, Yahoo, Amazon.com, Neptun
- egyetemi oktatás (kurzusok, szakok, termek, hallgatók, oktatók, órarend)
- vállalat (főosztályok, osztályok, dolgozók, fizetések)
- raktár (árúk, készlet, mozgások)
- könyvtár (olvasók, könyvek, kiadók, szerzők)
- lakás (helyiségek, bútorok, elektromos szerelvények)
- munkahely (helyiségek, számítógépek, telefonok, dolgozók)
- városi tömegközlekedés (buszjáratok, megállók, menetrend)
- pizzeria (pizzák, ügyfelek, rendelések)
- bank (folyószámlák, ügyfelek, átutalások)
- ingatlan (telkek, épületek, tulajdonosok)
- vasúti-, légi helyfoglalás (járatok, helyek, ügyfelek)

Minden adatbázisnak van egy belső struktúrája *sémája*. Ez tartalmazza az összes adatelem és a köztük lévő kapcsolatok definícióját, leírását. Ezek a **Metaadatok**.

Adatbázis-kezelő rendszernek (ABKR) DataBase Management System (DBMS)

ABKR: egy programrendszer (soft), melyet arra terveztek, hogy nagy mennyiségű adatot tudjon tárolni és használni.

Egy ABKR által használt technikák:

- programozási nyelvek (objektum-orientáltságot is)
- operációs rendszerek
- konkurens programozás
- adatstruktúrák
- algoritmusok
- párhuzamos és osztott rendszerek
- felhasználói felületek
- mesterséges intelligencia
- statisztika.

Az ABKR feladatai

- Támogatja új adatbázisok létrehozását, azok struktúrájának, tárolási módjának kialakítását.
- Megvalósítja az adatbázisban tárolt adatok kezelését, karbantartását.
- Lehetővé teszi a tárolt adatok feldolgozását, lekérdezését, módosítását. Új információkat képes előállítani, illetve a meglévő információknak a felhasználók által igényelt formában történő megjelenítését támogatja.
- Garantálja az adatok biztonságát, konzisztenciáját, a hozzáférések szabályozását, vagyis a felhasználói műveleteket csak az arra jogosult személyek végezhessek, és ezek a műveletek ne veszélyeztessék az adatok integritását.
- Lehetővé teszi az adatbázisok megosztását több felhasználó között.
- Rendszerhibák esetén képes egy helyes adatbázist visszaállítani.

Miért adatbázis technológia?

- Adatintegritás: az adatok korrektek, konzisztensek és aktuálisak
- Redundancia csökkentése (adat ismétlődés)
- Megoszthatóság
- Standardok
- Biztonság, adatvédelem (tranzakció vezérlés)

ABKR Típusok -1

▶ OLTP (On-Line Transaction Processing)

- ★ Gyors műveleteket végeznek kis mennyiségű adaton minden alkalommal csak olvasnak/frissítenek



ABKR Típusok -2

▶ OLAP (On-Line Analytical Processing)

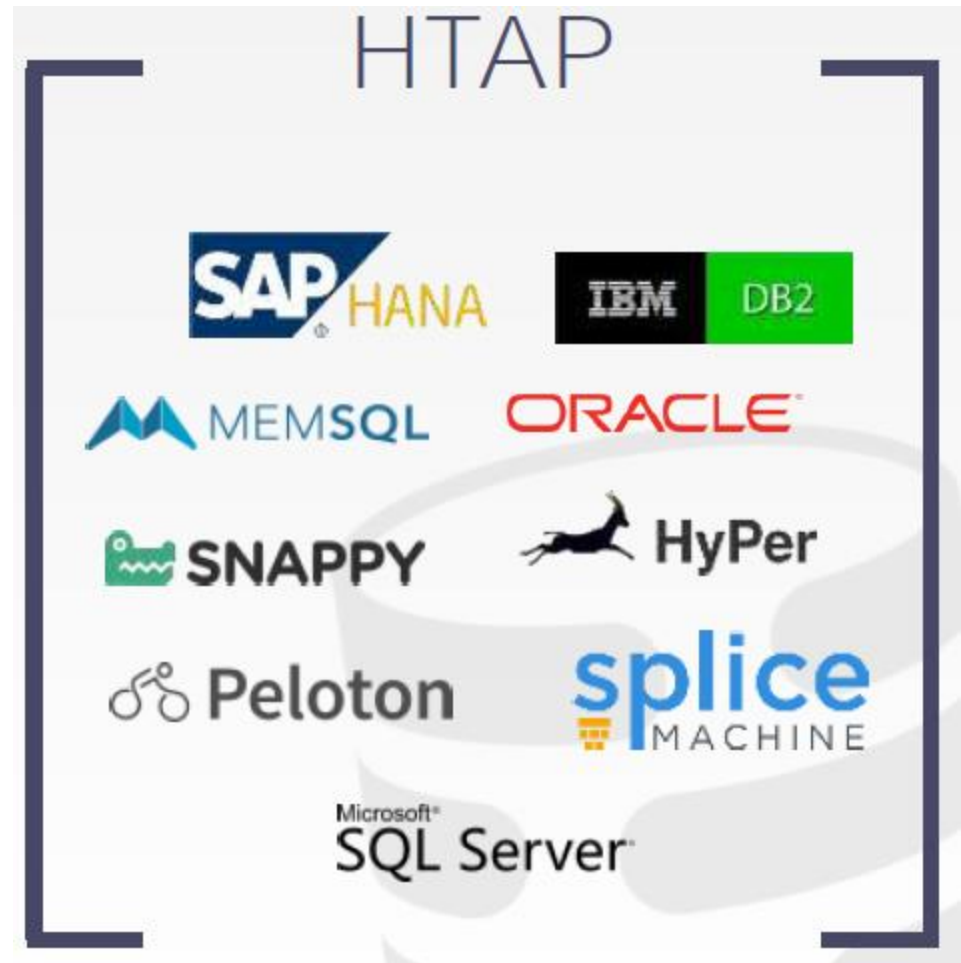
- ★ Komplex lekérdezésekhez, melyek nagy mennyiségű adatot használnak ahhoz, hogy aggregátumokat számítsanak ki



ABKR Típusok -3

▶ HTAP (Hybrid Transactional Analytical Processing)

- ★ OLAP és OLTP együtt ugyanazon az adatbázis példányon



What Database Does Facebook Use?

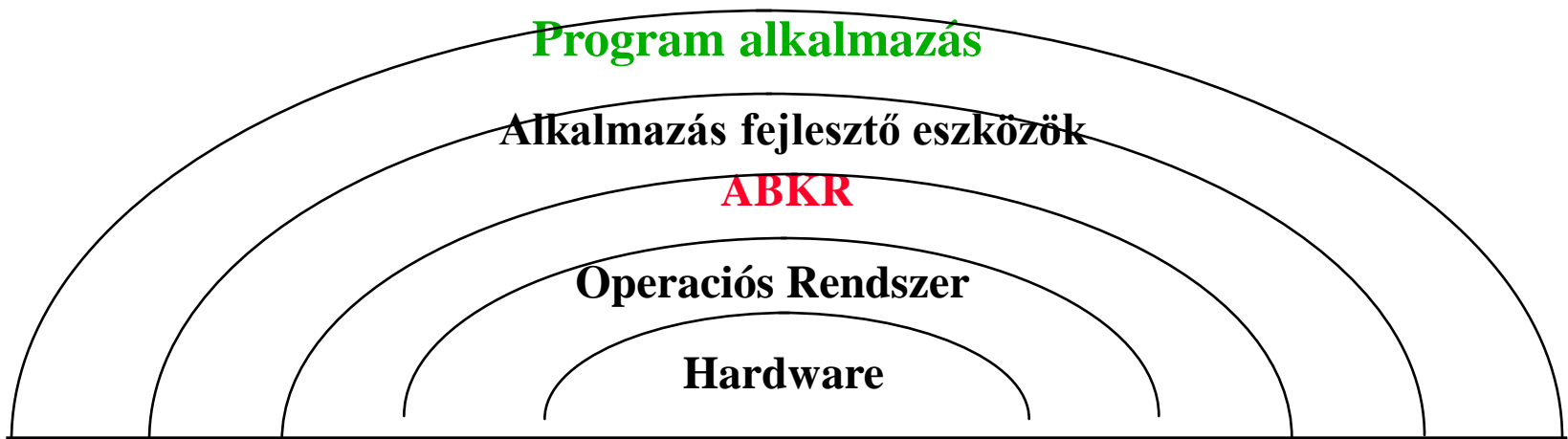
- ▶ **MySQL** is the core, main database used by Facebook for storing all the social data. It started with the **InnoDB MySQL database engine** & then wrote **MyRocksDB**, which was eventually used as the MySQL Database engine.
- ▶ **Memcache** sits in front of MySQL & is used for caching.
- ▶ To Handle BigData Facebook uses **Apache Hadoop, HBase, Hive, Apache Thrift, PrestoDB**. All these are used for [data ingestion](#), warehousing & running analytics.
- ▶ **Apache Cassandra** is used for the inbox search.
- ▶ **Beringei & Gorilla**, a high-performance time series storage engines for infrastructure monitoring.
- ▶ **LogDevice**, a distributed data store for storing logs.
- ▶ <https://www.8bitmen.com/what-database-does-facebook-use-a-1000-feet-deep-dive/>

Oracle- t fogunk tanulni!

- ▶ Oracle: nagy teljesítményű rendszer, nagy adatbázisok, sok felhasználó, különleges biztonsági követelmények esetén ajánlott.
- ▶ Certificate lehetőség!

ORACLE®

ABKR helye a számítógépes rendszerben



A világ legnagyobb adatbázisai - 2018

1. Klímaadatok Világközpontja - Németország
2. Az USA Nemzeti Energiakutató Tudományos Számítóközpontja
3. AT&T, Távközlési szolgáltató
4. Google
5. Sprint
6. ChoicePoint
7. Youtube
8. Amazon
9. CIA – az USA központi hírszerző ügynöksége
10. USA Kongresszusi Könyvtár

<http://www.edunow.ga/2018/01/top-10-largest-databases-in-world-2018.html>

Adatbázis-kezelők felépítése

▶ Lekérdezés-feldolgozó

- ★ Lekérdezés szintaktikai ellenőrzése
- ★ Adatbázis-objektumok létezésének, és a hozzáférési jogoknak az ellenőrzése (metaadatbázis, rendszertáblák)
- ★ Lekérdezés optimális átfogalmazása
- ★ Végrehajtási tervek készítése
- ★ Az adatstruktúrák, méretek statisztikái alapján várhatóan minimális költségű végrehajtási terv kiválasztása
- ★ Az optimális végrehajtási terv lefuttatása

▶ Tranzakció-kezelő:

- ★ Tranzakciók párhuzamos végrehajtásának biztosítása (atomosság, következetesség, elkülönítés, tartósság)

▶ Tárkezelő (állománykezelő):

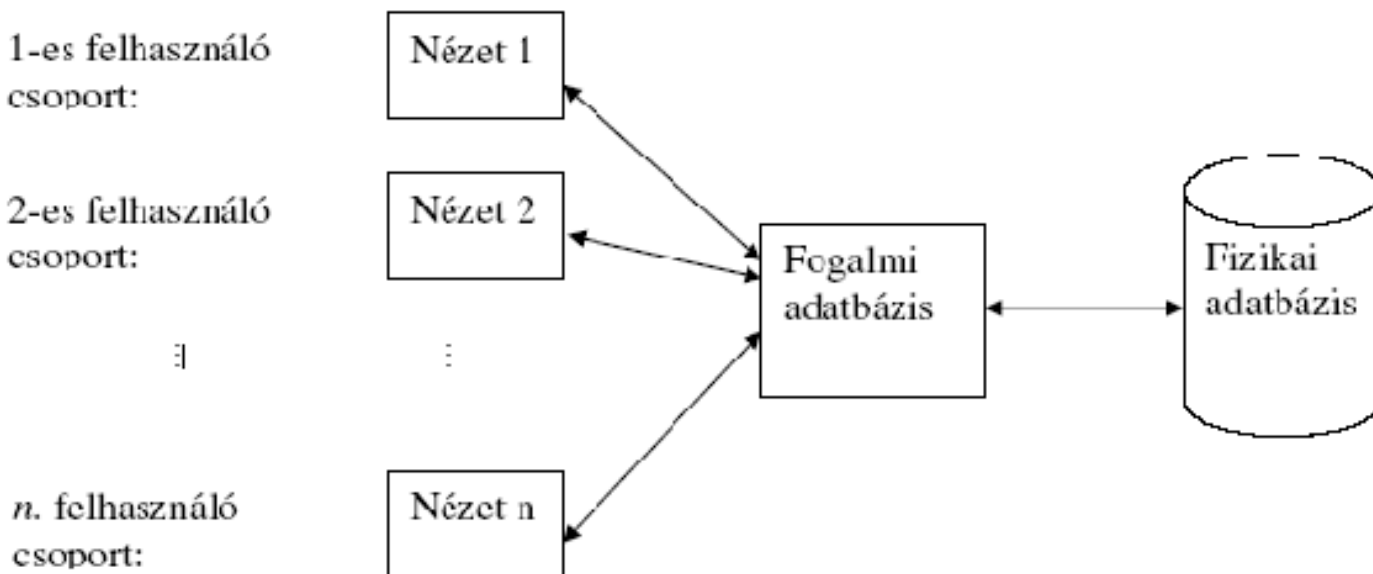
- ★ fizikai adatstruktúrák, táblák, indexek, pufferek kezelése

Az adatbázisok ANSI/SPARC architektúrája

1. **Fogalmi szint:** (conceptual level): egy alkalmazási környezet valamennyi ismeretét és azok valamennyi összefüggését egyetlen közös adatmodellben kell leírunk -> **egyed/kapcsolat adatmodell**, **STUDENT(sid, name)**,

2. **Belső szint:** Az adatoknak a számítógépes rendszerben való aktuális reprezentációját jelenti. **? byte**

3. **Külső szint:** a felhasználó szemszögéből vizsgálja az adatokat. **Select name from student**



Adatbázisok nyelvei

- Lekérdező nyelv
 - SQL: egy komplett programozási nyelv minden lehetséges adatbázis műveletre
 - Relációs algebra: egy alapnyelv adatlekérdezésre
 - Beágyazott SQL: egy nyelv amely összekombinálja az SQL-t és egy host nyelvet
 - 4GL program aplikációk SQL kiterjesztéssel
 - ODBC/JDBC/Dynamic SQL standard protokollok az adatbázisok elérésére egy távoli gépről

Adatbázisok nyelvei

- **Adatleíró nyelvek** (Data Definition Language - DDL)

- az egyedek és a köztük lévő kapcsolatok leírására

```
CREATE TABLE Csoportok (  
CsopKod VARCHAR(3) PRIMARY KEY,  
Evfolyam NUMBER(3),  
SzakKod CHAR(3));
```

- **Adatkezelő nyelvek** (Data Manipulation Language - DML)

- Lekérdezés, beszúrás, módosítás, törlés

```
SELECT Név  
FROM Diákok  
WHERE CsopKod = 531;
```

- **Adatvezérlő nyelvek** (Data Control Language – DCL)

- Védelem, tranzakciókezelés

```
GRANT SELECT ON diakok TO istvan;
```

Adatbázis felhasználók

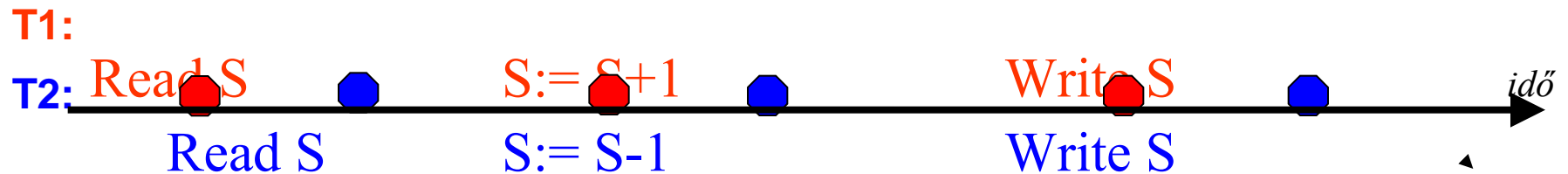
- Vég felhasználó
 - alkalmi felhasználó
 - hozzákapcsolódik az adatbázisokhoz formok vagy front-end applikációk használata során
 - DML-t használnak
- Programfejlesztő
 - megtervez és implementál olyan applikációkat, amelyek hozzákapcsolódnak az adatbázishoz
- Adatbázis adminisztrátor (DBA)
 - meghatározza és kezeli a fogalmi sémát
 - definiál applikáció és felhasználó nézeteket
 - monitorizálja és összhangban tartja az ABKR teljesítményét
 - futtatja és újratervezi az adatbázist
 - felelős a biztonságért

Több felhasználó támogatása

- ▶ **Felhasználói csoportok**
- ▶ **DBA** – adatbázis-adminisztrátor
- ▶ **Jogosultságok** (objektumok olvasása, írása, módosítása, készítése, törlése, jogok továbbadása, jogok visszavonása)
- ▶ Jogosultságok tárolása rendszertáblákban történik

Tranzakció-kezelés

- ▶ **Tranzakció**: adatkezelő műveletekből (adategység írása, olvasása) álló sorozat
- ▶ Cél: tranzakciók párhuzamos végrehajtása



- ▶ A tranzakció-kezelő biztosítja: (ACID)
 - ★ **Atomosság** (a tranzakció egységesen lefut vagy nem)
 - ★ **Következetesség** (a tranzakció futása után konzisztens legyen az adatbázis)
 - ★ **Elkülönítés** (párhuzamos végrehajtás eredménye egymás utáni végrehajtással egyezzen meg)
 - ★ **Tartósság** (a befejezett tranzakció eredménye rendszerhiba esetén sem veszhet el)

Tranzakció-kezelés

▶ Zárolások (Lock, Unlock)

T1: (Lock S, Read S, $S:=S+1$, Write S, Unlock S)

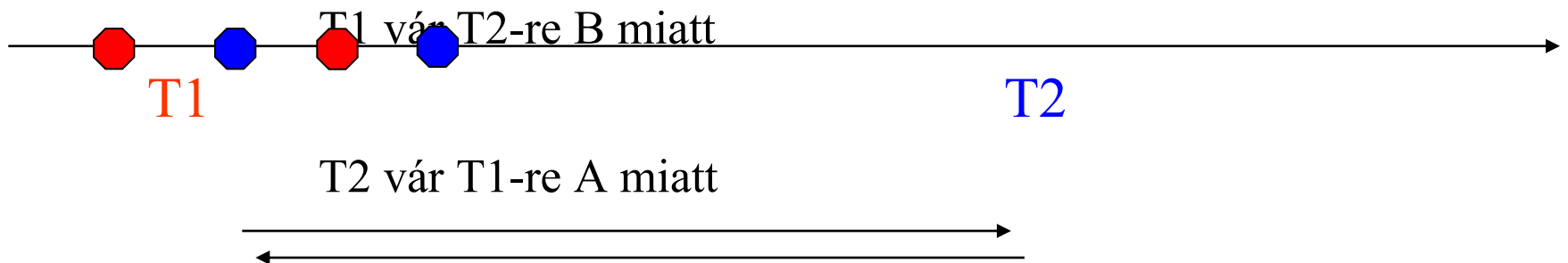
T2: (Lock S, Read S, $S:=S-1$, Write S, Unlock S)

- ▶ A zár kiadásához meg kell várni a zár feloldását.
- ▶ Csökken a párhuzamosíthatóság
- ▶ Záruk finomsága (zárolt adataegység nagysága, zárolás típusa) növeli a párhuzamosíthatóságot

▶ Holtpont probléma:

Lock A Read A Lock B Read B $C:=A+B$

Lock B Read B Lock A Read A $D:=A+B$



Tranzakció-kezelés

- ▶ **Kétfázisú protokoll** – a tranzakció elején zárolunk minden szükséges adatalelemet, a végén minden zárat feloldunk
- ▶ **Tranzakciók érvényesítése**, naplózás, **Commit, Rollback, Checkpoint**
- ▶ **Ütemező** (tranzakciók műveleteinek végrehajtási sorrendjét adja meg)
- ▶ **Szerializálhatóság** (az ütemezés ekvivalens a tranzakciók egymás utáni végrehajtásával)
- ▶ Tranzakciók állapotát, elvégzett műveleteket rendszertáblák tárolják

Helyreállíthatóság

- ▶ Szoftver- vagy hardverhiba esetén az **utolsó konzisztens állapot visszaállítása**
- ▶ Rendszeres **mentések**
 - ★ Statikus adatbázis (módosítás nem gyakori)
 - ★ Dinamikus adatbázis (módosítás gyakori)
- ▶ **Naplóállományok**
- ▶ Összefügg a tranzakciókezeléssel

Mi is az adatmodell?

Az adatmodell a valóság fogalmainak, kapcsolatainak, tevékenységeinek magasabb szintű ábrázolása

- ▶ azaz az adatok (mit nevezünk adatnak?) leírására szolgáló jelölés. Az adat és valóság viszonya.
- ▶ Ez a leírás általában az alábbi három részből áll:
 1. **Az adat struktúrája** (leírhatóvá tesszük a valóságra vonatkozó adatokat, adatok típusa, formálisan megadható)
 2. **Az adaton végezhető műveletek** (lekérdezések, módosítások, feldolgozások legyenek megfogalmazhatók)
 3. **Az adatokra tett megszorítások** (milyen adatokat engedélyezünk? később: megszorítások, triggerek)
- ▶ Implementáció kérdései: Nagy tömegben hatékony és sok felhasználó számára biztonságos legyen a megvalósítás.

Adatmodellek

- Relációs modell: az adatok kétdimenziós táblákban tárolódnak
- Félig-strukturált adatmodell: tartalmazza az XML-t és a hozzá tartozó szabványokat
- Hálós modell: a rekordok pointerekkel kapcsolódnak egymáshoz –elavult, ritka
- Hierarchikus: a rekordok fastruktúra-szerű hierarchiába rendezettek –elavult, ritka
- Objektum-orientált modell: objektumokkal definiálja a DB-t
- Objektum-relációs modell: a relációs modell bővítése objektum-orientált lehetőségekkel
- Osztott modell – több független ABKR fut a kommunikációs hálózat csomópontjain
- Kulcs/Érték, Gráf, Dokumentum, Column Family –**NoSQL**
- Tömb/Mátrix – gépi tanuláshoz

Relációs adatmodell

Reláció = Tábla

- ▶ Reláció = egy sorokból álló halmaz
- ▶ Előfordulás v. példány = a tábla sorai
- ▶ Séma = a relációk szerkezetének leírása

- ▶ Megszorítások
 - ★ Primary Key
 - ❖ Egyértelműen azonosít egy elemet
 - ❖ Automatikus generálás (Sequence, Auto_Increment)

 - ★ Foreign Key
 - ❖ Meghatározza, hogy egy attributum az egyik relációból a másik reláció melyik sorának feleltethő meg

- ▶ Pl: Employees (Employee_id, First_name, Last_name, Email, Phone_number, Hire_date, Job_id, Salary, Commission_pct, Manager_id, Department_id)

Példa relációkra

(BSc-n itt) az adatokat természetes, táblázatos formában kezeljük

Egy reláció sémája: Sör (név, gyártó).

Az adatbázis sémája: Sör (név, gyártó), Bár (név, város, tulaj),
Felszolgál (sör, bár, ár).



Egy relációs sorokból álló halmaz, nem pedig lista!!!

Relációs adatmodell – relációs séma

- ▶ A relációk táblákban jelennek meg. A tábláknak egyedi neve van. A relációk oszlopait az attribútumok címzik. A tábla sorait tetszőlegesen megcserélhetjük, sorok sorrendje lényegtelen
- ▶ **R - relációnev** (vagy táblanév)
- ▶ adattípusa: **sortípus** - A sortípus megadása:
- ▶ **<attributumnev1: ertektípus1, ... , attr.nevk: ert.tipusk>**
- ▶ röviden **<A₁, ... , A_k>**
- ▶ **Relációséma**: Relációnev(sortípus) vagyis
- ▶ **R(A₁: értéktípus₁, ... , A_k: értéktípus_k)**
- ▶ röviden **R(A₁, ... , A_k)**
- ▶ jelölés **U = {A₁, ... , A_k}**

Relációs adatmodell - előfordulás

- A táblázatos szemléltetésből áttérhetünk a sorok egy másik szemléltetésére:
- $t \in R$ sor felfogható függvényként is
- $t_i : U \rightarrow \text{értékek}$, ahol $U = \{A_1, \dots, A_k\}$
Ezzel a jelöléssel $t_i(A_j) = v_{ij}$
ekvivalens jelöléssel $t_i[A_j]$ vagy $t_i.A_j$ (obj.orient.jel.)
- Jelölések: Tekintsük U tetszőleges X részalmazát
Ha $X = \{A_{i_1}, \dots, A_{i_l}\}$ akkor $t[X]$ típusa:
 $\langle A_{i_1}: \text{értéktípus}_{i_1}, \dots, A_{i_l}: \text{értéktípus}_{i_l} \rangle$

Relációs adatbázis felépítése

- ▶ az adatbázis tulajdonképpen relációk halmaza
- ▶ a megfelelő relációsémák halmaza adja az adatbázissémát (jelölése vastag szárú \mathbf{R})

$$\mathbf{IR} = \{\mathbf{R}_1, \dots, \mathbf{R}_n\}$$

- ▶ a hozzá tartozó előfordulások az adatbázis-előfordulást

Példa megszorításokra - Kulcs

- ▶ **Elsődleges Kulcs - Primary Key**
- ▶ Az attribútumok egy halmaza egy kulcsot alkot egy relációra nézve, ha a reláció bármely előfordulásában nincs két olyan sor, amelyek a kulcs összes attribútumának értékein megegyeznének.
- ▶ $R(A_1, \dots, A_k)$ séma, $X = \{A_{i1}, \dots, A_{il}\}$ kulcs, ha nincs két olyan t_1, t_2 sor, amely különbözne, pedig az X -en megegyezik $t_1[X] = t_2[X]$
- ▶ **Megjegyzés:** egy kulcs nem feltétlenül egy attribútumból áll. Például a Filmek táblában a filmcím és év együtt alkotják a kulcsot, nem elég a filmcím, uis van például

(King Kong, 1933), (King Kong, 1976) és (King Kong, 2005).

- ▶ A kulcsot aláhúzás jelöli: Filmek (cím, év, hossz, ...)

Idegen kulcs és hivatkozási épség

- ▶ **Idegen kulcs, FOREIGN KEY**
- ▶ $R(A_1, \dots, A_k)$ séma, $X = \{A_{i1}, \dots, A_{il}\}$ kulcs,
- ▶ $S(B_1, \dots, B_n)$ séma, $Y = \{B_{j1}, \dots, B_{jl}\}$ idegen kulcs,
- ▶ ami az X -re hivatkozik a megadott attribútum sorrendben:
- ▶ B_{j1} az A_{i1} -re, ..., B_{jl} az A_{il} -re
- ▶ **Hivatkozási épség, REFERENCES**
- ▶ megszorítás a két tábla együttes előfordulására:
- ▶ Ha $s \in S$ sora, akkor létezik $t \in R$ sor,
- ▶ hogy $s[B_{j1}, \dots, B_{jl}] = t[A_{i1}, \dots, A_{il}]$
- ▶ Ekkor az S -en Y idegen kulcs, ami hivatkozik az R kulcsára