

# DATABASE



## Adatbázisok

# Kapcsolattartás és egyéb infók

- **Kommunikációs csatornák:** MS Teams (kód: **dixihs5**), Canvas, e-mail:

Dóka-Molnár Andrea-Éva

andrea.molnar@math.ubbcluj.ro

Gaskó Noémi

noemi.gasko@ubbcluj.ro

- **Online alkalmak színhelye:** MS Teams (kód: **dixihs5**)

# Követelmények

- 25% - labortevékenység
- 75% - vizsga
  - 55% - írásbeli vizsga
  - 20% - laborvizsga
- **Minimális követelmény:**
  - összesen minimum 50p
  - írásbelin minimum 27.5p, 12.5p labortevékenységből, 10p laborvizsgán

**További infók canvas-en (Syllabus)**

# Irodalomjegyzék

1. J.D. Ullman, J. Widom: *Adatbázisrendszerek. Alapvetés - Második átdolgozott kiadás*, Panem Kiadó, 2009.
2. Varga Ibolya: *Adatbázisrendszerek (A relációs adatbázisoktól az XML adatokig)*, Egyetemi Kiadó, Kolozsvár, 2005. Megtalálható canvas-en:  
[Files/kurzus/konyveszet/ABKurzusOsztoXML.pdf](#)
3. Gajdos Sándor: *Adatbázisok* (egyetemi jegyzet), 2016.  
(Online elérhető)

**További könyvészet is elérhető: Syllabus**

# Áttekintés/Tematika

## Bevezetés

- Bevezetés az adatbázis-kezelő rendszerek világába

## Adatmodellezés

- Egyed/kapcsolat adatmodell
- Relációs adatmodell
- E/K diagram átalakítása relációs adatmodellé

## Relációs algebra

- Relációs algebra műveletei, használata

## SQL nyelv (MS SQL specifikusan):

- DDL, DML, QL, triggerek, jogosultságok, függvények, procedúrák, kurzorok használata.

## Relációs adatbázisok tervezése

- anomáliák, funkcionális- és többértékű függőségek, függőségörzés ellenőrzése, normálformák

## NoSQL adatbázisok

- Jellemzők, relációs adatbázisokkal való összehasonlítás, NoSQL adatbázisok típusai, multimodell adatbázisok

# Bevezetés

- *Hol találkozunk adatbázisokkal? Miért van szükségünk adatbázisokra?*

# Bevezetés

- *Hol találkozunk adatbázisokkal? Miért van szükségünk adatbázisokra?*

Adatbázis alapú alkalmazások:

- Banki ügyintézés (folyószámlák, ügyfelek, átutalások)
- Vasút- és légitársaságok (járatok, menetrendek, foglalások, ügyfelek)
- Egyetemek (kurzusok, szakok, termek, hallgatók, oktatók, órarend)
- (Online) eladások (vásárlók, termékek, beszerzések)
- Humán erőforrások (alkalmazottak, fizetések, adómódosulások)
- Könyvtárak (olvasók, könyvek, kiadók, szerzők)

# Bevezetés

- *Hol találkozunk adatbázisokkal? Miért van szükségünk adatbázisokra?*

Adatbázis alapú alkalmazások:

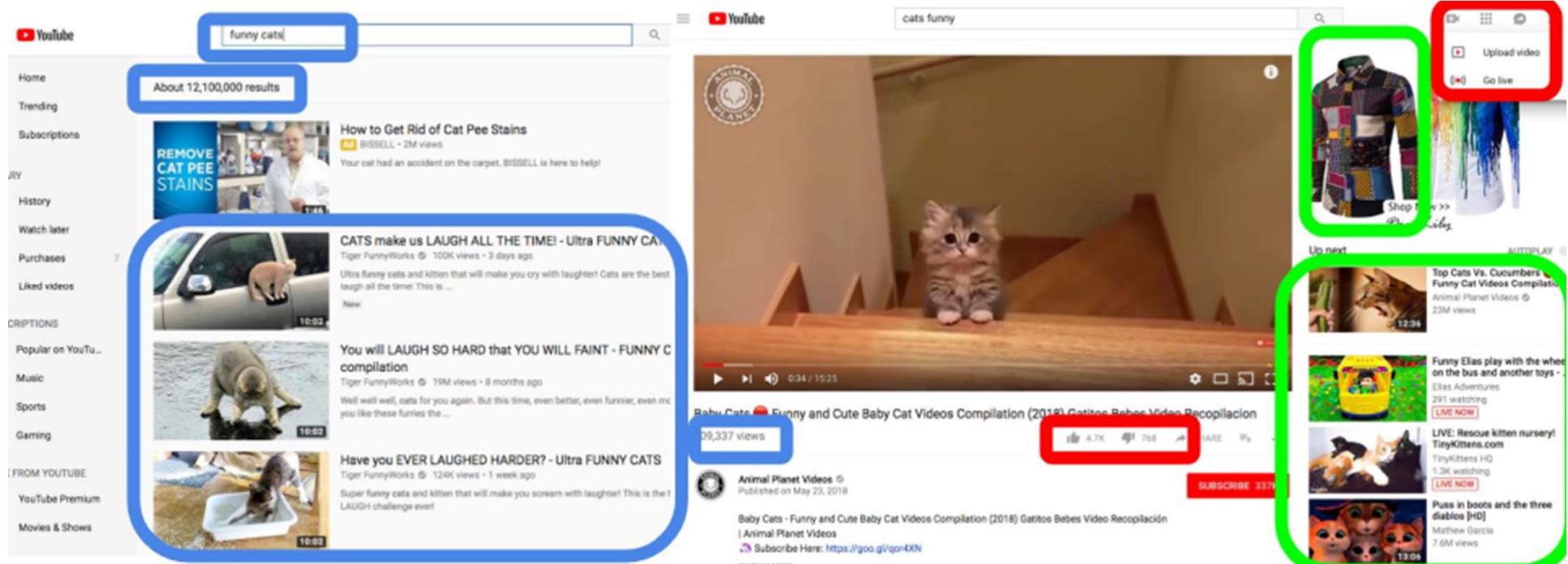
- Banki ügyintézés (folyószámlák, ügyfelek, átutalások)
- Vasút- és légitársaságok (járatok, menetrendek, foglalások, ügyfelek)
- Egyetemek (kurzusok, szakok, termek, hallgatók, oktatók, órarend)
- (Online) eladások (vásárlók, termékek, beszerzések)
- Humán erőforrások (alkalmazottak, fizetések, adómódosulások)
- Könyvtárak (olvasók, könyvek, kiadók, szerzők)

*Máspéldák?*



# Bevezető példák

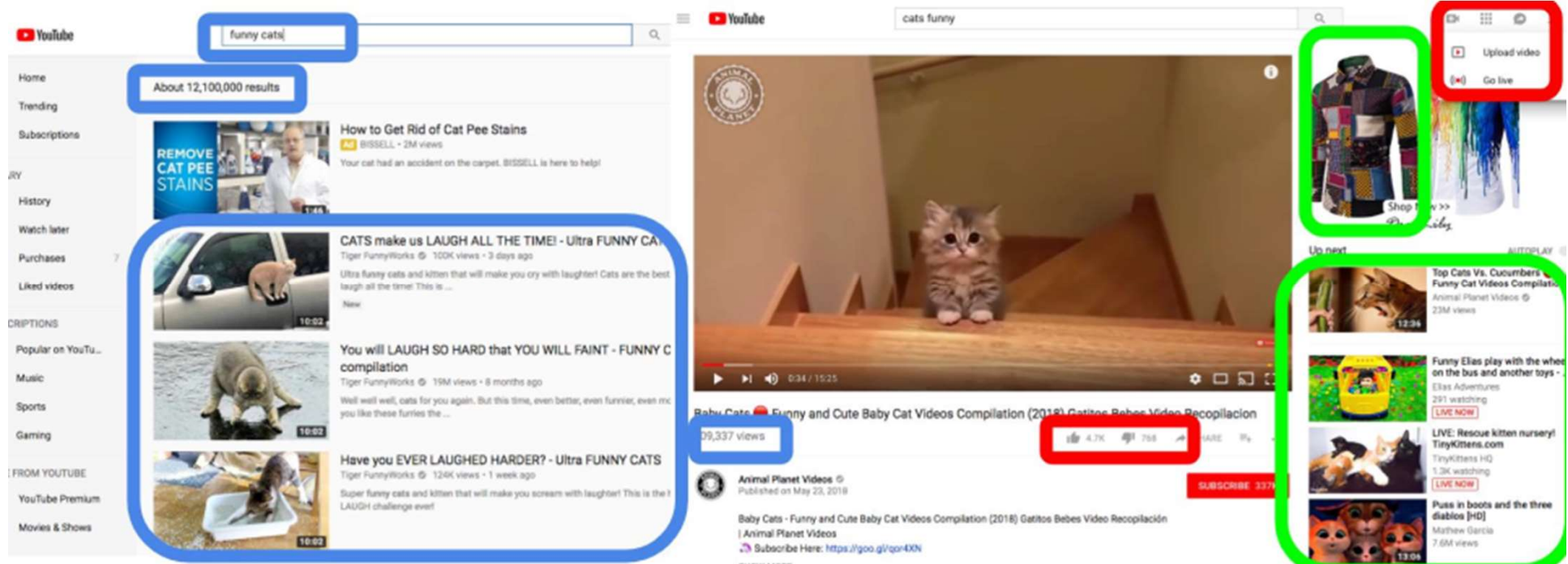
## YouTube adatbázisa



Forrás: <https://bit.ly/2HD4f40>

# Bevezető példák

## YouTube adatbázisa

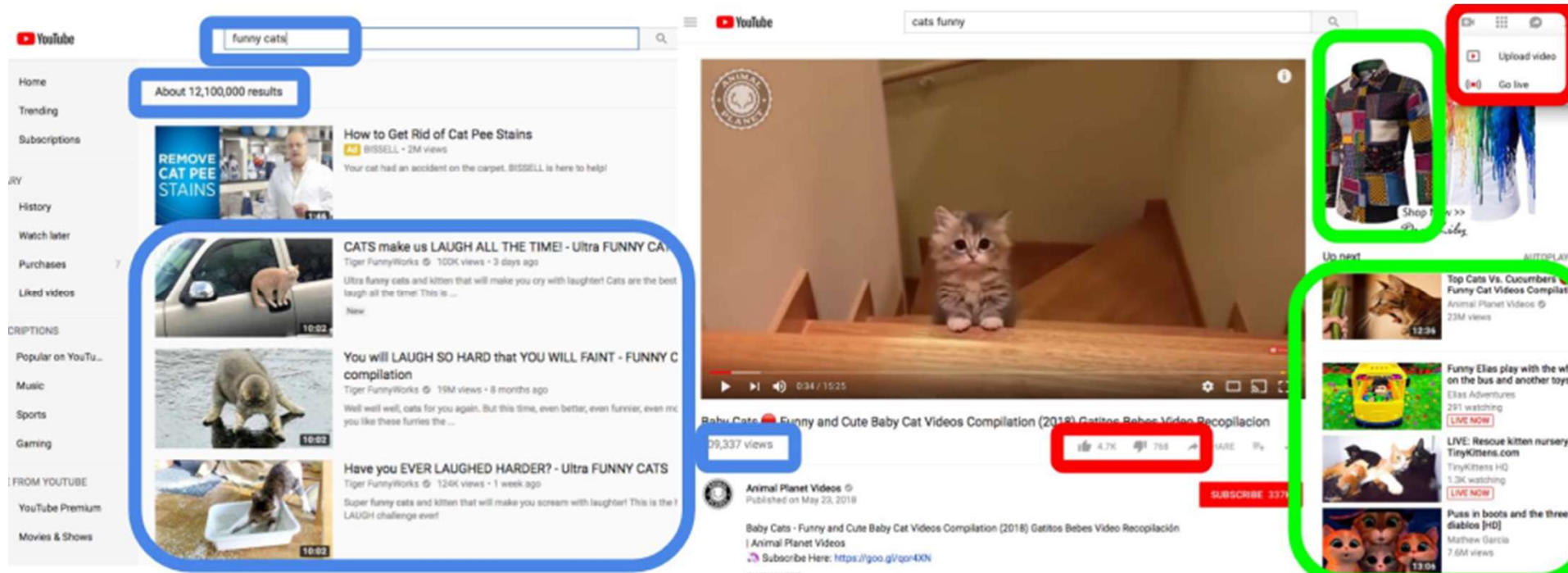


Forrás: <https://bit.ly/2HD4f40>

- 3 művelet:
  - **olvasás/adatlekérdezés:** találatok listája: megtekintések, lájkok, videók és leírások

# Bevezető példák

## YouTube adatbázisa

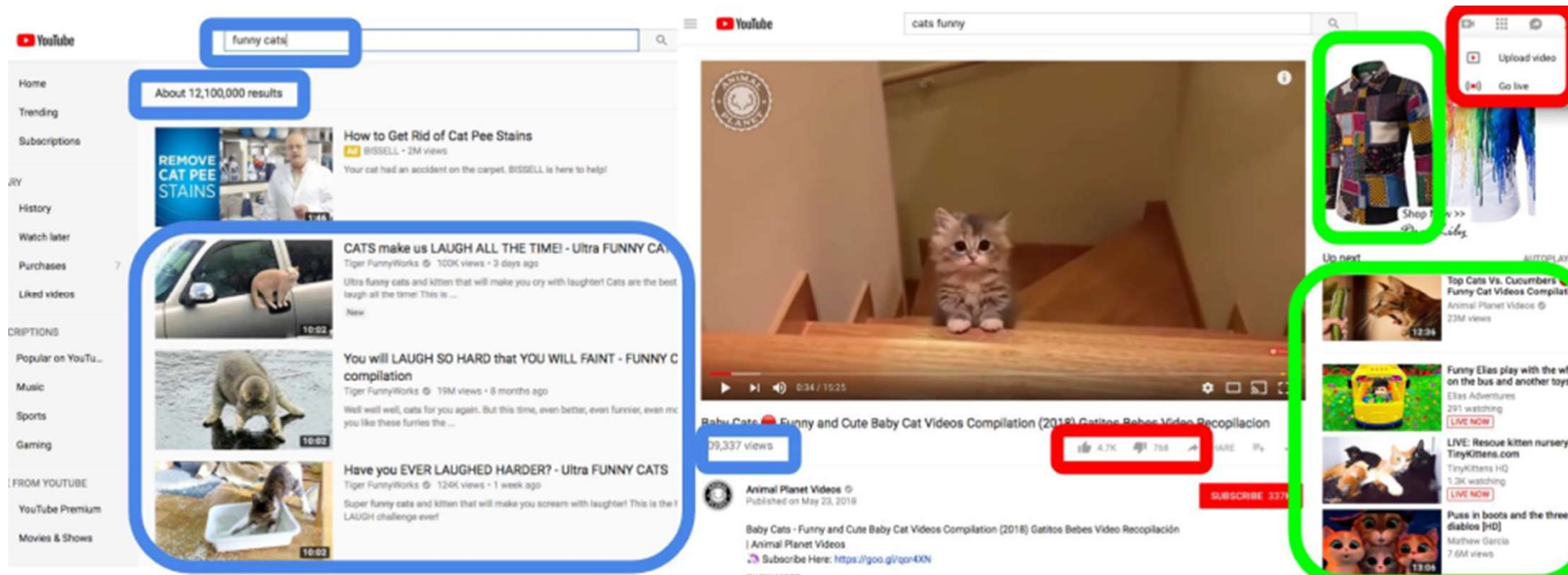


Forrás: <https://bit.ly/2HD4f40>

- 3 művelet:
  - **olvasás/adatlekérdezés**: találatok listája: megtekintések, lájkok, videók és leírások
  - **(adat)módosítás**: feltöltés, lájk, visszajelzés

# Bevezető példák

## YouTube adatbázisa



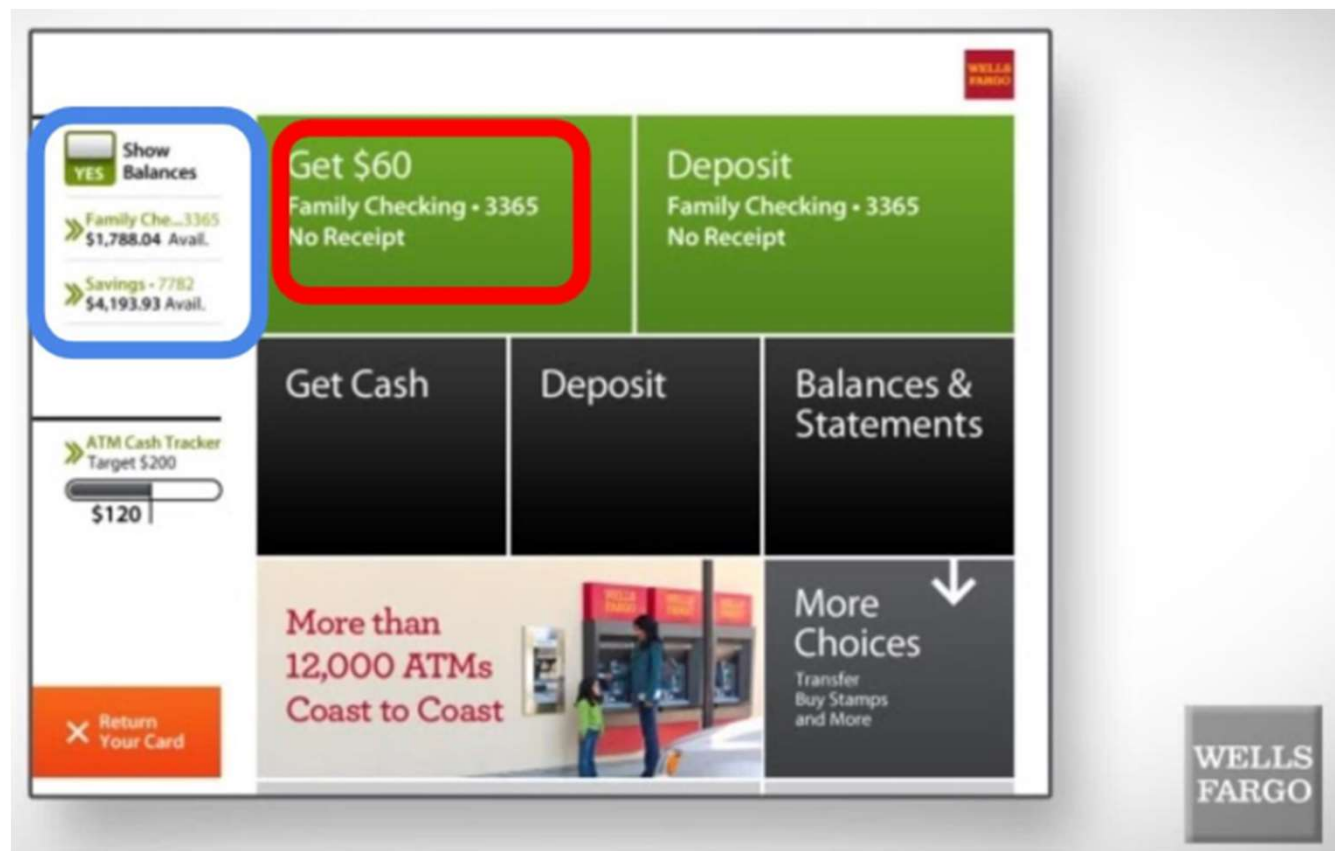
Forrás: <https://bit.ly/2HD4f40>

- 3 művelet:
  - **olvasás/adatlekérdezés**: találatok listája: megtekintések, lájkok, videók és leírások
  - **(adat)módosítás**: feltöltés, lájk, visszajelzés
  - **„tanulás”**: hasonló videók, releváns reklámok



# Bevezető példák

## ATM adatbázisa



- Műveletek:
  1. Egyenleg lekérése
  2. Pénz kiadása
  3. Egyenleg módosítása

Forrás:<https://bit.ly/2HD4f40>

# Adatbázis-kezelés alapjai

## ■ Adatbázis:

- Nagyméretű adatgyűjtemény (adatok összessége), mely hosszú időn keresztül (akár évekig) létezik és elérhető.
- **Más def.:** adatok gyűjteménye, amelyeket egy adatbázis-kezelő rendszer (ABKR) kezel.

# Adatbázis-kezelés alapjai

- **Adatbázis** = adatok gyűjteménye, amelyeket egy adatbázis-kezelő rendszer (ABKR) kezel.
- **Adatbázis-kezelő rendszer (ABKR; database management system (DBMS))**: olyan speciális szoftverrendszer, amellyel lehetséges az adatbázisokban tárolt adatok *definiálása; kezelése; karbantartása; felügyelete*.
  - *NEMfelhasználói program (főfeladata NEM felhasználói igények kielégítése)*

# ABKR-ek előnyei, feladatai

- Felhasználók számára új adatbázisokat létrehozásának lehetővé tétele + adatbázisok sémájának (logikai struktúrájának) egy adatdefiníciós nyelven való megadása (ld. később: **DDL**).
- Felhasználók számára annak lehetővé tétele, hogy az adatokat egy megfelelő nyelv segítségével lekérdezhessék, illetve módosíthassák (ld. később: **DML+(D)QL**).
- Nagyon nagy mennyiségű adat hosszú időn keresztüli tárolásának támogatása.
- Lehetőség az adatbázisok megosztására több felhasználó között.



# Az ABKR-ek előnyei, feladatai

- Garantálja az adatok biztonságát, konzisztenciáját, a hozzáférések szabályozását.  $\leftrightarrow$  A felhasználói műveleteket csak az arra jogosult személyek végezhetik, és ezek a műveletek nem veszélyeztethetik az adatok integritását.  
Pl. új alkalmazott fizetésének bevezetése:
  - jogosultságok ellenőrzése
  - megszorítások ellenőrzése (pl. befér-e a költségvetésbe?)
- Rendszerhibák esetén helyes adatbázis visszaállításának képessége.
- Egy adatmodellre épül (multimodell adatbázisoknál többre):
  - adatbázis megtervezése, adatok érthető formában való megtekintése



## Relációs ABKR-k

# NoSQL és NewSQL ABKR-k



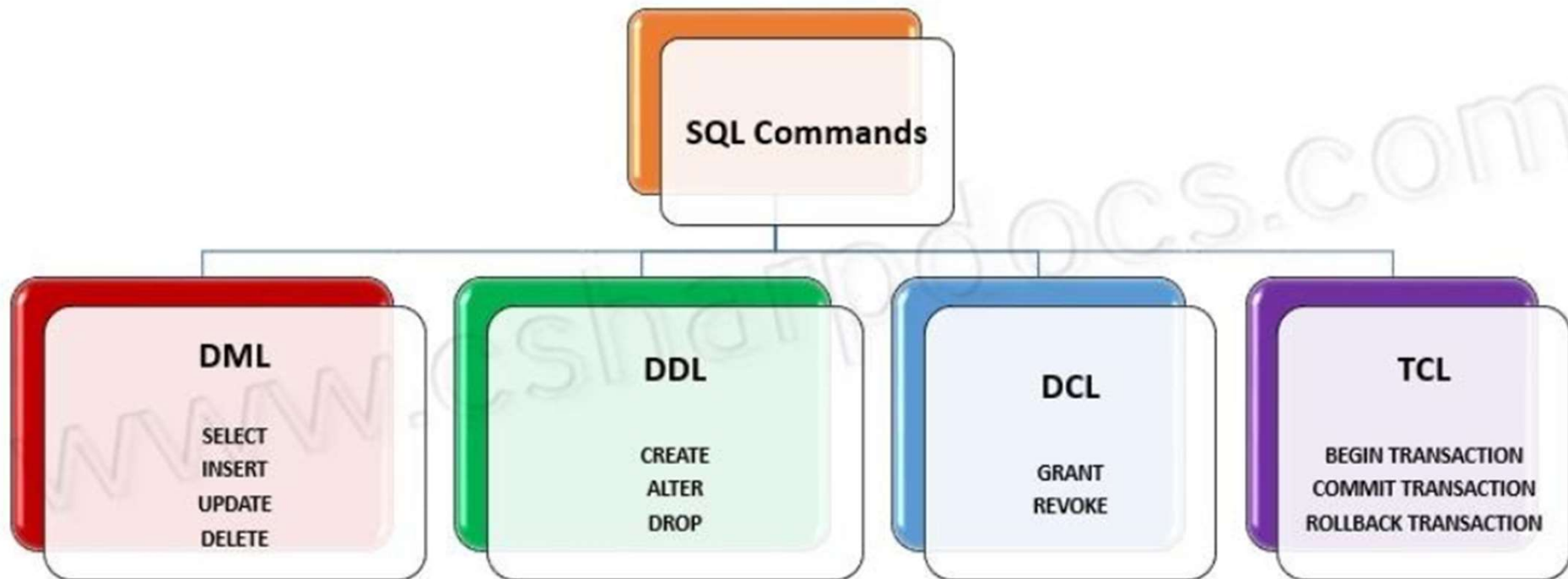
# ABKR-k népszerűsége 2021-ben

Rank			DBMS	Database Model	Score		
Jul 2021	Jun 2021	Jul 2020			Jul 2021	Jun 2021	Jul 2020
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1262.66	-8.28	-77.59
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1228.38	+0.52	-40.13
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	981.95	-9.12	-77.77
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	577.15	+8.64	+50.15
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	496.16	+7.95	+52.68
6.	↑ 7.	↑ 8.	Redis +	Key-value, Multi-model ⓘ	168.31	+3.06	+18.26
7.	↓ 6.	↓ 6.	IBM Db2	Relational, Multi-model ⓘ	165.15	-1.88	+1.99
8.	8.	↓ 7.	Elasticsearch +	Search engine, Multi-model ⓘ	155.76	+1.05	+4.17
9.	9.	9.	SQLite +	Relational	130.20	-0.33	+2.75
10.	↑ 11.	10.	Cassandra +	Wide column	114.00	-0.11	-7.08
11.	↓ 10.	11.	Microsoft Access	Relational	113.45	-1.49	-3.09
12.	12.	12.	MariaDB +	Relational, Multi-model ⓘ	97.98	+1.19	+6.86
13.	13.	13.	Splunk	Search engine	90.05	-0.22	+1.78
14.	14.	14.	Hive	Relational	82.68	+2.98	+6.25
15.	15.	↑ 18.	Microsoft Azure SQL Database	Relational, Multi-model ⓘ	75.22	+0.43	+22.59
16.	16.	16.	Amazon DynamoDB +	Multi-model ⓘ	75.20	+1.43	+10.62
17.	17.	↓ 15.	Teradata	Relational, Multi-model ⓘ	68.95	-0.39	-7.02
18.	18.	↑ 22.	Neo4j +	Graph	57.16	+1.41	+8.24
19.	19.	↑ 20.	SAP HANA +	Relational, Multi-model ⓘ	53.81	-0.29	+2.48
20.	20.	↓ 19.	Solr	Search engine, Multi-model ⓘ	51.79	-0.30	+0.15

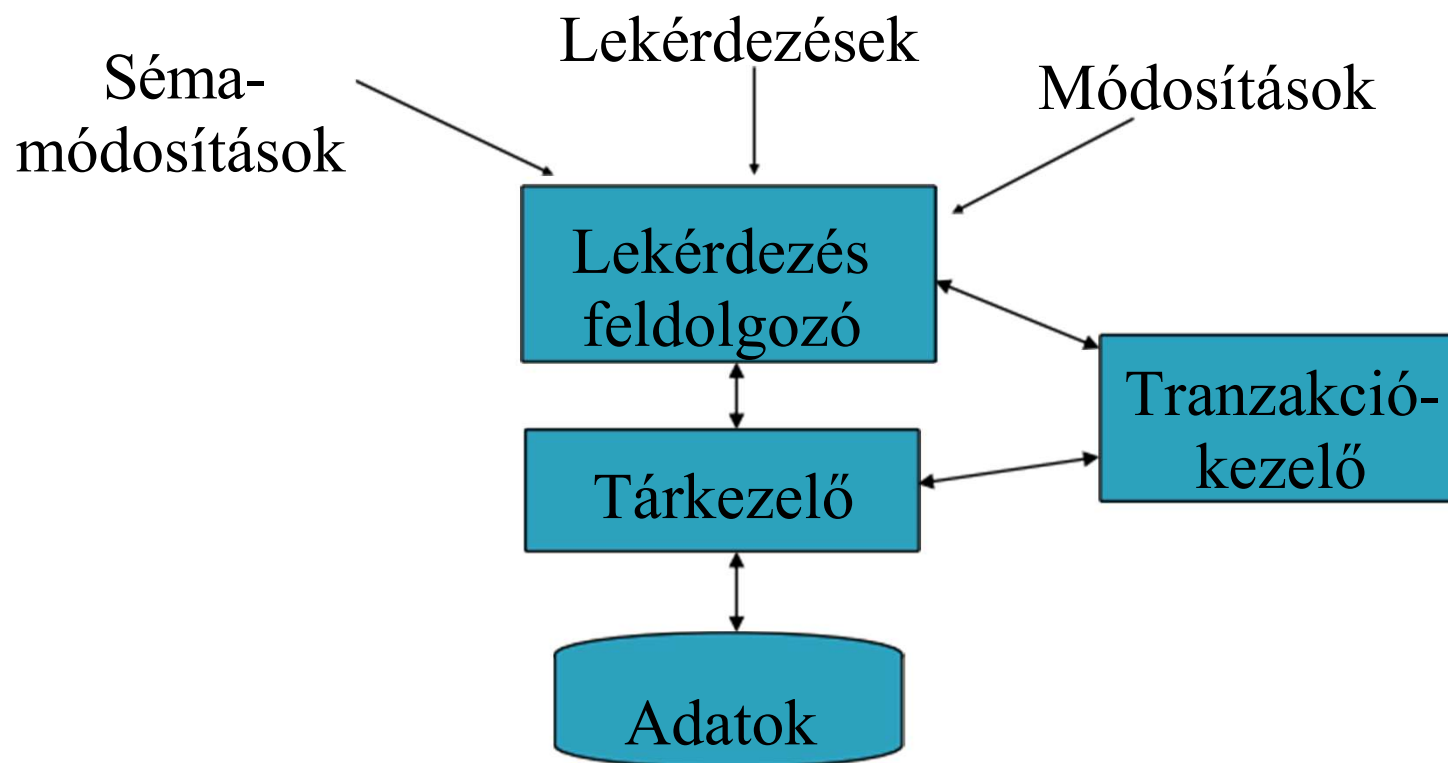


# Adatbázisok nyelvei

- relációs ABKR ↔ SQL (Structured Query Language)



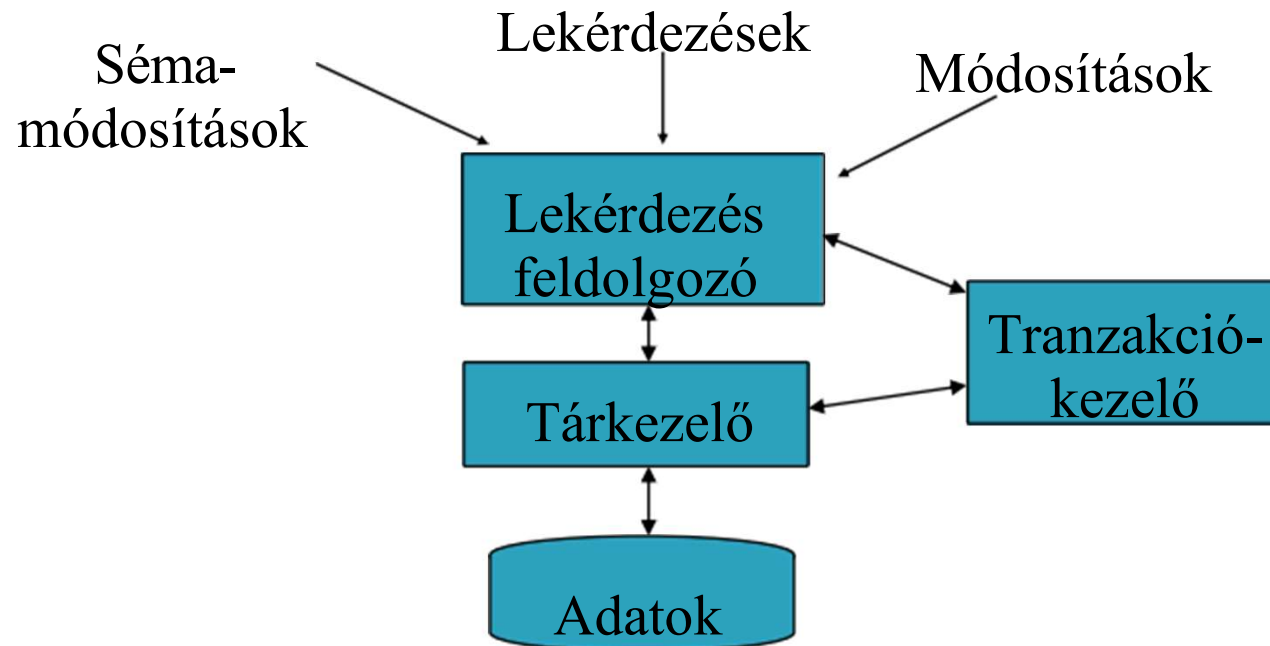
# Adatbázis-kezelők felépítése



# Adatbázis-kezelők felépítése

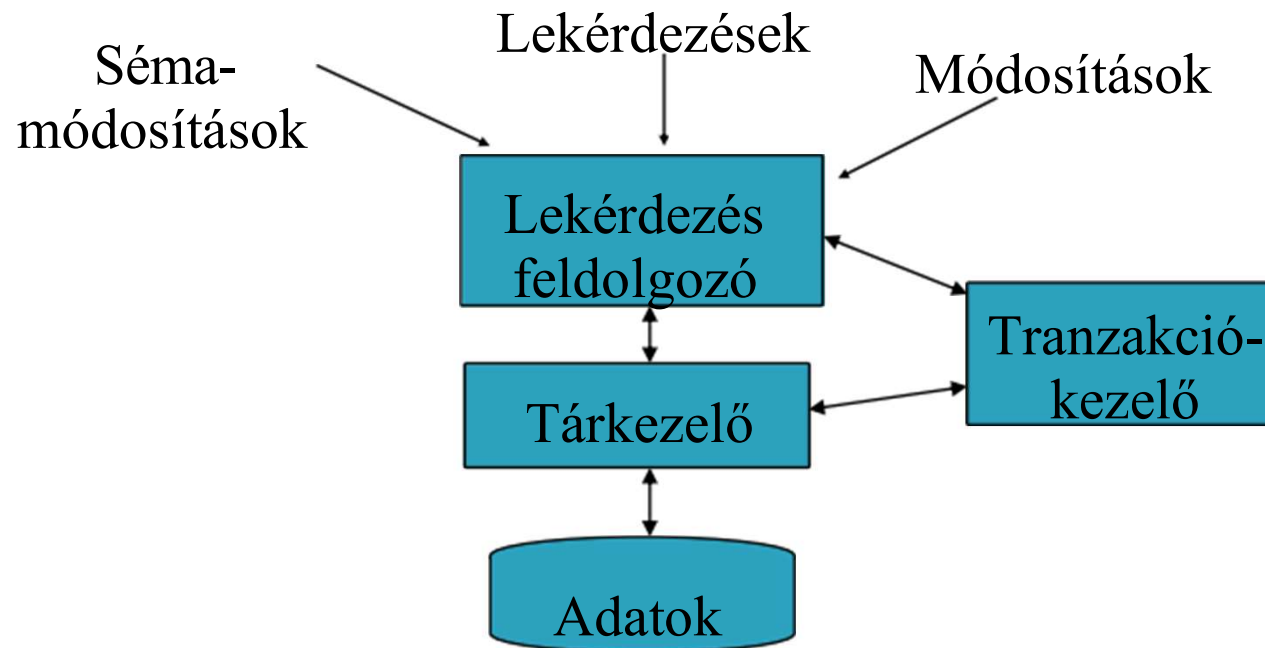
## ■ Adatok, metaadatok

Fizikailag valahol tárolódnak az adatok (pl. milyen nevű utas, melyik gépre foglalt helyet) és a metaadatok (mik a relációk nevei és attribútumai és ezek típusai, illetve pl. milyen indexek vannak létrehozva a kereséshez).



# Adatbázis-kezelők felépítése

- **Tárkezelő (állománykezelő)**
  - fizikai adatstruktúrák, táblák, indexek, pufferek kezelése.
  - Részei: fájlkezelő, pufferkezelő





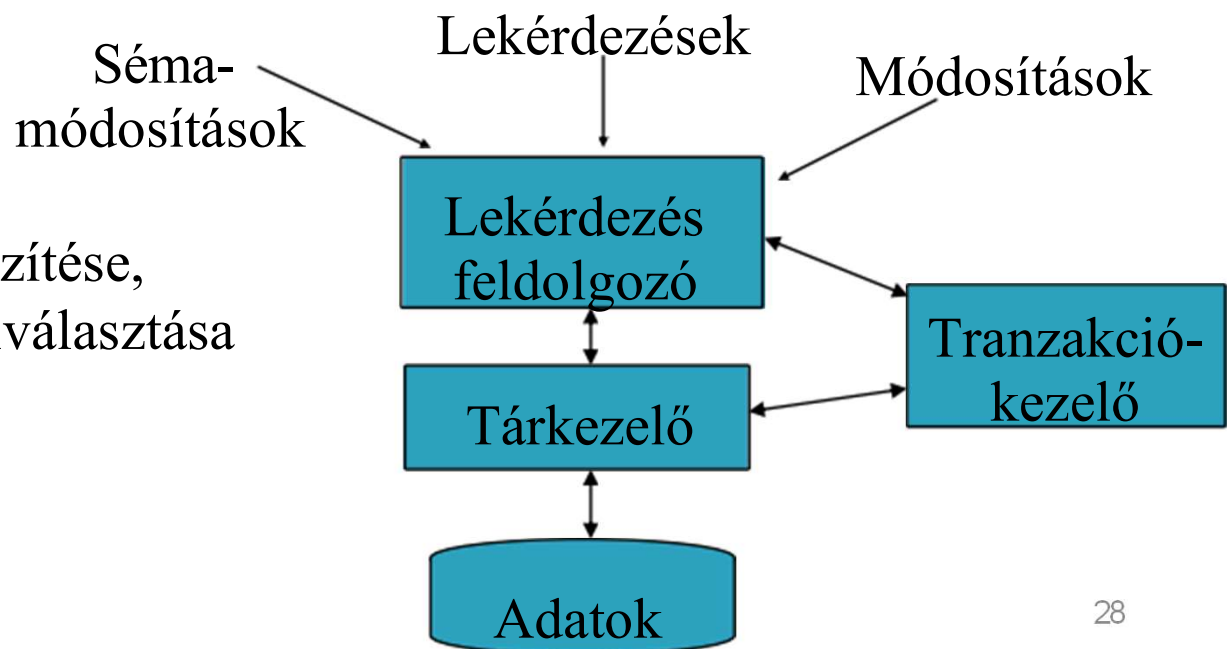
# Adatbázis-kezelők felépítése

## ▪ Lekérdezés-feldolgozó

- Lekérdezés szintaktikai ellenőrzése
- Sémaműveletek: az adatbázis logikai struktúrájának kialakítása, módosítása
- Adatmódosítás: az adatbázis tartalmának módosítása, beszúrás, törlés
- Adatbázis-objektumok létezésének, és a hozzáférési jogoknak az ellenőrzése (metaadatbázis, rendszertáblák)

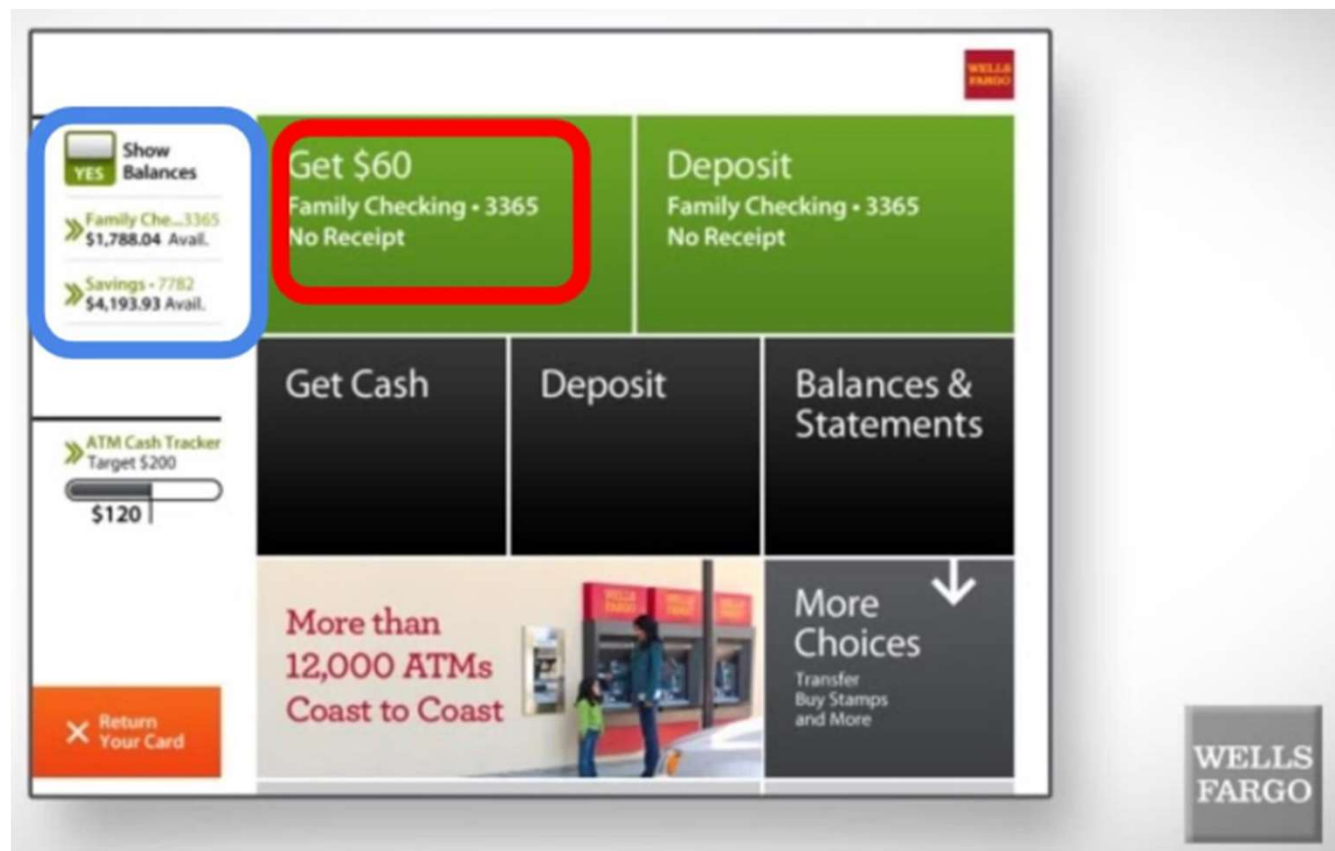
- Lekérdezés optimális átfogalmazása,

végrehajtási tervek készítése,  
ebből: min. költségű kiválasztása  
és végrehajtása



# Bevezető példák

## ATM adatbázisa



- Műveletek:

1. Egyenleg lekérése

2. Pénz kiadása

3. Egyenleg módosítása

Forrás: <https://bit.ly/2HD4f40>

*Mi történik, ha valamelyik lépésnél hiba lép fel?  
Definiálhatunk-e más sorrendet  
a műveletek esetén?*

# Tranzakciókezelő

- **Két nagyobb problémakör** megoldására jó:
  - Több felhasználó egyszerre használja az adatbázist, egyidejű hozzáférések kezelése.
  - Rendszerhibák, ABORT-ok hatásainak kivédése: ezek bekövetkeztekor sem veszhetnek el adatok, nem maradhat az adatbázis inkonzisztens állapotban.
- Ezek megoldására: alapfogalom a **tranzakció**: egy felhasználóhoz tartozó, összetartozó utasítások olyan sorozata, melyek vagy mind végrehajtódnak vagy semelyik sem (atomiság).
  - *Pl. banki átutalásnál nem lehet, hogy csak a pénz levonása történik meg az egyik számlán, de nem íródik jóvá a másikon.*

# Elvárások a tranzakciókezelésben

- **A (atomicity, atomiság):** egy tranzakció vagy teljesen végrehajtódik vagy semmi se hajtódik végre belőle
- **C (consistency) – konzisztencia:** a tranzakció a helyesség (konzisztencia) egysége, az adatbázist egy helyes állapotból egy másik helyes állapotba alakítja.
- **I (isolation) – izoláció:** különböző tranzakciók egymástól elszigetelten futnak, mintha egymás után hajtódnának végre; *valójában:* egyidejűleg versengenek az adatbázis elemekért.
- **D (durability) – tartósság:** ha a tranzakció elért a végpontjához (COMMIT), az általa végzett adatbázis-módosítások véglegesek, még ha közben esetleg hiba is lép fel.

→ tranzakciók **ACID** tulajdonságai

# Példa: Repülőgépes helyfoglalás

- **Adatelemek:** indulási-, érkezési időpont, honnan indul, hova érkezik, ár, repülőjegyek darabszáma, utas neve stb.
- **Lekérdezések:** van-e még hely, mennyi az ára, mikor indul a gép
- **Módosítások:** új utas bevitele, helyfoglalás
- **Párhuzamosság:** egyszerre több jegyeladás és lekérdezés is mehet
- **Védelem:** helyfoglalás nem veszhet el

