

# Adatbázisok elmélete 1. előadás

Katona Gyula Y.

Budapesti Műszaki és Gazdaságtudományi Egyetem

Számítástudományi Tsz.

I. B. 137/b

`kiskat@cs.bme.hu`

`http://www.cs.bme.hu/~kiskat`

2005

# Tudnivalók

Katona Gyula   [kiskat@cs.bme.hu](mailto:kiskat@cs.bme.hu)   <http://www.cs.bme.hu/~kiskat/adatbazis/>

## Tudnivalók

Katona Gyula [kiskat@cs.bme.hu](mailto:kiskat@cs.bme.hu) <http://www.cs.bme.hu/~kiskat/adatbazis/>

### Előadás:

- Kedd 12:15-14:00, I. B. 27.
- Csütörtök 8:15-10:00, I. B. 27.

## Tudnivalók

Katona Gyula [kiskat@cs.bme.hu](mailto:kiskat@cs.bme.hu) <http://www.cs.bme.hu/~kiskat/adatbazis/>

### Előadás:

- Kedd 12:15-14:00, I. B. 27.
- Csütörtök 8:15-10:00, I. B. 27.

### Tankönyv

- [Gajdos Sándor](#): Adatbázisok, Műegyetemi kiadó, 2004. (új kiadás)

## Tudnivalók

Katona Gyula [kiskat@cs.bme.hu](mailto:kiskat@cs.bme.hu) <http://www.cs.bme.hu/~kiskat/adatbazis/>

### Előadás:

- Kedd 12:15-14:00, I. B. 27.
- Csütörtök 8:15-10:00, I. B. 27.

### Tankönyv

- **Gajdos Sándor**: Adatbázisok, Műegyetemi kiadó, 2004. (új kiadás)
- **Ullman – Widom**: Adatbázisrendszerek, alapvetés, PANEM – Prentice-Hall, 1998
- **Garcia-Molina – Ullman – Widom**: Adatbázisrendszerek magvalósítása, PANEM – John Wiley & Sons, 2001

## Tudnivalók

Katona Gyula [kiskat@cs.bme.hu](mailto:kiskat@cs.bme.hu) <http://www.cs.bme.hu/~kiskat/adatbazis/>

### Előadás:

- Kedd 12:15-14:00, I. B. 27.
- Csütörtök 8:15-10:00, I. B. 27.

### Tankönyv

- **Gajdos Sándor:** Adatbázisok, Műegyetemi kiadó, 2004. (új kiadás)
- **Ullman – Widom:** Adatbázisrendszerek, alapvetés, PANEM – Prentice-Hall, 1998
- **Garcia-Molina – Ullman – Widom:** Adatbázisrendszerek magvalósítása, PANEM – John Wiley & Sons, 2001
- **Letölthető fóliák és egyéb anyagok a honlapon.**

## Aláírás és vizsga

Az **aláírásért** (kb. 9-10. hét) az évközi zárhelyit legalább elégségesre meg kell írni. Ez egy 60 pontos zárthelyi lesz, stílusában hasonló, mint a tavalyiak, nem feltétlenül 6 darab 10 pontos feladat lesz benne (természetesen az egyes feladatok pontszámai rajta lesznek a feladatsoron). A 60 pontból 20 pont kell a ketteshez, 30 pont a hármas szinthez, 40 pont a négyes szinthez és 50 pont az ötös szinthez. A zh pontszáma beleszámít a vizsgajegybe, amennyiben jobb, mint a vizsga pontszáma (erről lásd a vizsgánál írottakat.)

## Aláírás és vizsga

Az **aláírásért** (kb. 9-10. hét) az évközi zárhelyit legalább elégségesre meg kell írni. Ez egy 60 pontos zárthelyi lesz, stílusában hasonló, mint a tavalyiak, nem feltétlenül 6 darab 10 pontos feladat lesz benne (természetesen az egyes feladatok pontszámai rajta lesznek a feladatsoron). A 60 pontból 20 pont kell a ketteshez, 30 pont a hármas szinthez, 40 pont a négyes szinthez és 50 pont az ötös szinthez. A zh pontszáma beleszámít a vizsgajegybe, amennyiben jobb, mint a vizsga pontszáma (erről lásd a vizsgánál írottakat.)

Akinek van régebről aláírása, annak nem kötelező ZH-t írni, de írhat, ha akar. A régi aláírás egy esetleges rossz zhval sem vész el.



## Aláírás és vizsga

Az **aláírásért** (kb. 9-10. hét) az évközi zárhelyit legalább elégségesre meg kell írni. Ez egy 60 pontos zárthelyi lesz, stílusában hasonló, mint a tavalyiak, nem feltétlenül 6 darab 10 pontos feladat lesz benne (természetesen az egyes feladatok pontszámai rajta lesznek a feladatsoron). A 60 pontból 20 pont kell a ketteshez, 30 pont a hármas szinthez, 40 pont a négyes szinthez és 50 pont az ötös szinthez. A zh pontszáma beleszámít a vizsgajegybe, amennyiben jobb, mint a vizsga pontszáma (erről lásd a vizsgánál írottakat.)

Akinek van régebből aláírása, annak nem kötelező ZH-t írni, de írhat, ha akar. A régi aláírás egy esetleges rossz zhval sem vész el.

**Pótzárthelyi** kb. 2 héttel a ZH után lesz, ezen már csak az aláírást lehet megszerezni, az eredmény semmiképpen nem számít a vizsgajegybe.

## Aláírás és vizsga

Az **aláírásért** (kb. 9-10. hét) az évközi zárhelyit legalább elégségesre meg kell írni. Ez egy 60 pontos zárthelyi lesz, stílusában hasonló, mint a tavalyiak, nem feltétlenül 6 darab 10 pontos feladat lesz benne (természetesen az egyes feladatok pontszámai rajta lesznek a feladatsoron). A 60 pontból 20 pont kell a ketteshez, 30 pont a hármas szinthez, 40 pont a négyes szinthez és 50 pont az ötös szinthez. A zh pontszáma beleszámít a vizsgajegybe, amennyiben jobb, mint a vizsga pontszáma (erről lásd a vizsgánál írottakat.)

Akinek van régebből aláírása, annak nem kötelező ZH-t írni, de írhat, ha akar. A régi aláírás egy esetleges rossz zhval sem vész el.

**Pótzárthelyi** kb. 2 héttel a ZH után lesz, ezen már csak az aláírást lehet megszerezni, az eredmény semmiképpen nem számít a vizsgajegybe.

**Sem a ZH-n, sem a pótzh-n nem lehet használni semmilyen segédeszközt (a zh és a pótzh closed book).**

## Aláírás és vizsga

Akinek nem sikerül a pótzs sem, az első három vizsgahét valamelyik vizsgáján próbálkozhat, feltéve, hogy az egyetemi adminisztráció által előírt formai követelményeket (iv vagy különjárás díj, engedély, stb.) teljesíti.

Csak egyetlen alkalommal lehet megpróbálni ilyen módon az aláírás megszerzését, az aláíráshoz a vizsgán legalább kettest kell elérni. Az itt elért pontszám nem számít bele a későbbi vizsgajegybe.

## Aláírás és vizsga

Akinek nem sikerül a pótzh sem, az első három vizsgahét valamelyik vizsgáján próbálkozhat, feltéve, hogy az egyetemi adminisztráció által előírt formai követelményeket (iv vagy különjárás díj, engedély, stb.) teljesíti.

Csak egyetlen alkalommal lehet megpróbálni ilyen módon az aláírás megszerzését, az aláíráshoz a vizsgán legalább kettest kell elérni. Az itt elért pontszám nem számít bele a későbbi vizsgajegybe.

A vizsga: A vizsgaidőszakban három vagy négy vizsgaalkalom lesz. Mindegyik írásbeli, itt sem lehet használni semmilyen segédeszközt. A vizsgán 60 pontot lehet szerezni, ennek egy részét elméleti kérdésekkel, a maradékot feladatokkal. Stílusában hasonló lesz, mint a korábbi vizsgák, de itt is igaz, hogy nem 6 darab 10 pontos feladat lesz.

## Aláírás és vizsga

**A megajánlott vizsgajegy:** 20 ponttól kettes, 30-tól hármas, 40-től négyes és 50-től ötös, illetve ha a vizsgán szerzett pontszám legalább 20 (a vizsga legalább kettes) és a félévközi zh eredménye jobb, mint a vizsgáé, akkor a két pontszám átlagára kapott osztályzat. Ha a megajánlott jegy egyes, az mindenképp bekerül az indexbe. **Ha a megajánlott jegy legalább kettes, lehetőség van egy szóbeli kérdésre felelni a kiosztáskor (ezt mindenki maga dönti el, hogy kéri-e), és ezzel a megajánlott jegyen legfeljebb lehet egyet javítani vagy rontani.** Az eredményhirdetésre és a szóbeli feleletekre az írásbelik után néhány nappal, egy időben kerül sor.

# Adatbázis



telefonkönyv, könyvtár, notesz, internet

NEPTUN.WORLD - Számítástudományi és Információelméleti Tanszék

Leírás | Hallgatók | Oktatók | **Tárgyak** | Publikációk


Aktív tárgyak


Kód	Tárgy neve
BMEVIMA2025	A számítástudomány alapjai
BMEVIMA9000	A Szemantikus világháló és az ontológiakezelés alapjai
BMEVIMA9043	Adat- és jeltömörítés
BMEVIMA5309	Adatbányászat laboratórium
BMEVIMA9031	Adatbányászati algoritmusok
▶ BMEVIMA3232	Adatbázisok
BMEVIMAD084	Adattömörítés
BMEVIMA9039	Alakfelismerés
BMEVIMA5308	Alakfelismerés és adatbányászat
BMEVIMA2207	Algoritmusok elmélete
BMEVIMA9312	Algoritmusok és adatstruktúrák valószínűségi alapú elemzése


Keresés:

Szűrési feltétel

Szerkesztés Minden

 Bővebb

 Nyomtat

 Bezár

The screenshot shows a Mozilla browser window displaying the Matáv website. The browser's address bar shows the URL `http://tudakozo.matav.hu/`. The website header includes the Matáv logo and navigation links for 'tudakozók', 'telefonkönyvek', and 'előfizetők adatai'. Below the header is a banner for 'On-line Tudakozó és Telefonkönyv' with a navigation bar numbered 1 to 6.

The main content area is titled 'Tájékoztató' and contains introductory text. Below this, there are three search sections:

- On-line Tudakozó:** Includes input fields for 'Név:' and 'Település:', a 'keresés »' button, and a 'szűkítés:' section with fields for 'név:', 'hasonló nevűeket is:' (checkbox), 'keresés típus:' (dropdown menu), 'előfizető típus:' (dropdown menu), 'település:', 'irányítószám:', 'utca/tér:', 'házzám:', and 'mezők kiürítése' with a 'keresés »' button.
- On-line Telefonkönyv:** Includes input fields for 'Név:', 'Általános keresőszó:', 'szűkítés:', 'Megye:', 'Település:', 'Utca / Házzám:', 'Közet / Telefon', 'Irányítószám:', 'szórészlet:' (dropdown menu), 'és/Vagy kapcsolat:' (dropdown menu), 'Max. találat:' (dropdown menu), and 'mezők kiürítése' with a 'keresés »' button.
- Arany Oldalak:** Includes input fields for 'Szakmák:', 'Általános keresőszó:', 'keresés »', 'szűkítés:', 'Cégnév:', 'Megye:', 'Település:', 'Utca / Házzám:', 'Irányítószám:', 'Közet / Telefon:', 'Internet:', and 'E-mail:'.



Japanese <-> English Dictionary Server - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop http://rut.org/cgi-bin/j-e/sjis/S=48/fg=r/nocolor/dict Search Print

Home Bookmarks Keresők Újságok Radio Japán Jap<-> Eng Sport Adatbazis Matek Számítógép Currency Conv Big Broth

## 和英／英和辞典 - Japanese↔English Dictionary

Welcome to Jeffrey's Japanese↔English Dictionary Server! Also see the [searchable database of kanji information](#). If the Japanese doesn't come out, [check here](#). [また、日本語のページもあります](#). This page available [with colors](#), etc.

This server has a mirror in [Melbourne, Australia](#), [Vancouver, Canada](#), [Lund, Sweden](#), and another in [San Jose, USA](#). This server's web space kindly provided by [Robert Henney](#).

Instructions and such are at the bottom of the page.

Search in dictionaries of [  words ;  names ;  legal terms ;  life-science terms ;  
 Four-Character Idiomatic Compounds ;  aviation terms ;  computer terms ;  compound verbs terms ;  
 concrete terms ;  engineering and science terms ;  environment terms ;  finance terms ;  forestry terms ;  
 geological terms ;  Japanese place names ;  linguistics terms ;  marketing terms ;  pulp and paper terms ;  
 star and constellation names ;  Buddhist Terms ; ] with search key:  
word starting with pattern  given as the  Japanese  text

[ disable [fuzzy search](#) for Japanese lookups  ]

[Search only [commonly-used words](#) ] [Restrict to [word-class](#) (EDICT tags)  Do not restrict to a word class

You can customize various things for all interactions with my dictionary system ([see here](#)), or use the following for some one-time changes for the dictionary lookup only: [ [Anchors to Large Japanese](#)  ] [link [kanji information](#)  ]  
[link [external search engines](#)  ]


**News:** You can now [customize](#) various things in this server. See the [Change notes](#) for more info.

The Aria Database - Search the Database - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop  Search Print

Home Bookmarks Keresők Újságok Radio Japán Jap<-> Eng Sport Adatbázis Matek Számítógép Currency Conv Big Broth



## Search the Database

Complete Site Index :

Enter as many or as few terms in the search form as you wish and click on "Search the Database!" to bring back the entries matching your search terms. See [search instructions](#) below for help.

All Fields :

Aria :

Composer :

Opera :

Role :

Range : From   To

Range does not matter -  [Help on Range Notation](#)

Find only arias with : Sound File -  Translation -  Aria Text -

Choose Database : Arias-  Ensembles-  Operatic Roles-

[Difference between databases](#)


Number of Arias Initially Displayed : First 20 Only-  All-

RecipeSource: Your Source for Recipes on the Internet - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop  Search Print

Home Bookmarks Keresők Újságok Radio Japán Jap<-> Eng Sport Adatbazis Matek Számítógép Currency Conv Big Bro



**Welcome to RecipeSource!**  
*RecipeSource is the new home of  
 SOAR: The Searchable Online Archive of Recipes  
 and your source for recipes on the Internet.*

September 9, 2004 Home : Your Source for Recipes on the Internet

**Search for Recipes:**  
 **GO**  
[Advanced Search](#)

**Browse by region:**  
[Africa & Middle East](#)  
[Asia & Pacific Ocean](#)  
[Europe](#)  
[North & South America](#)  
[Non-regional](#)

**Browse by type:**  
[Main Dishes](#)  
[Soups & Stuff](#)  
[Baked Goods](#)  
[Fruits, Grains & Vegetables](#)  
[On the Side](#)  
[Sweets & Desserts](#)  
[Snacks & Appetizers](#)  
[Holiday Foods](#)  
[Restricted & Special Diets](#)  
[Miscellaneous & Other](#)

**Other info:**  
[About Us](#)  
[Contact Us](#)  
[Site Map](#)  
[Site News](#)  
[Submit a Recipe](#)

**Our Recipe Collection**

We've organized our recipes into two major groups - recipes primarily identified with an ethnic cuisine are broken down by region and ethnic group, while other recipes are categorized by the type of dish.

**Ethnic cuisines by region:**

*Africa & Middle East:* [Armenian](#) · [Egyptian](#) · [Ethiopian](#) · [Lebanese](#) · [Moroccan](#) · [Persian](#) · [Turkish](#) · [Other African](#) · [Other Middle Eastern](#)

*Asia & Pacific Ocean:* [Australian](#) · [Burmese](#) · [Chinese](#) · [Filipino](#) · [Hawaiian](#) · [Indian & Pakistani](#) · [Indonesian](#) · [Japanese](#) · [Korean](#) · [New Zealand](#) · [Singaporean](#) · [Tahitian](#) · [Thai](#) · [Tibetan](#) · [Vietnamese](#)

*Europe:* [Austrian](#) · [Basque](#) · [Belgian](#) · [British](#) · [Croatian](#) · [Czech](#) · [Danish](#) · [Dutch](#) · [Finnish](#) · [French](#) · [German](#) · [Greek](#) · [Hungarian](#) · [Icelandic](#) · [Irish](#) · [Italian](#) · [Norwegian](#) · [Polish](#) · [Portuguese](#) · [Rumanian](#) · [Russian](#) · [Scandinavian](#) · [Scottish](#) · [Serbian](#) · [Spanish](#) · [Swedish](#) · [Swiss](#) · [Ukrainian](#) · [Welsh](#)

*North & South America:* [Argentinian](#) · [Brazilian](#) · [Cajun](#) · [Canadian](#) · [Caribbean](#) · [Colombian](#) · [Eskimo](#) · [Mexican](#) · [Native American](#) · [Peruvian](#) · [Venezuelan](#)

*Non-regional:* [Jewish](#)

**Recipes by type of dish:**

*Main Dishes:* [Breakfast Dishes](#) · [Burgers](#) · [Casseroles](#) · [Crockpot Cooking](#) · [Dinner Pies](#) · [Fish & Seafood](#) · [Meat](#) · [Pasta](#) · [Pizza](#) · [Poultry](#) · [Sandwiches](#)

*Soups & Stuff:* [Chili](#) · [Soups](#) · [Stews](#) · [Stocks](#)

*Baked Goods:* [Bagels](#) · [Biscuits](#) · [Breads](#) · [Buns](#) · [Desserts & Sweets](#) · [Muffins](#) · [Pastries](#) · [Rolls](#) · [Scones](#)

*Fruits, Grains, & Vegetables:* [Bean Salads](#) · [Beans & Grains](#) · [Fruits](#) · [Fruit Salads](#) · [Pasta Salads](#) · [Pickles](#) · [Pilafs](#) · [Polenta](#) · [Potato Salads](#) · [Rice](#) · [Salads](#) · [Stuffed Vegetables](#) · [Stuffing](#) · [Vegetables](#)

# Adatbáziskezelő rendszerek (DBMS) jellemzői

- Komplex hardver-szoftver rendszer adatok kezelésére (tárolás, módosítás, lekérdezés)

## Adatbáziskezelő rendszerek (DBMS) jellemzői

- Komplex hardver-szoftver rendszer adatok kezelésére (tárolás, módosítás, lekérdezés)
- Nagy adatmennyiség

## Adatbáziskezelő rendszerek (DBMS) jellemzői

- Komplex hardver-szoftver rendszer adatok kezelésére (tárolás, módosítás, lekérdezés)
- Nagy adatmennyiség
- Gazdag logikai struktúra (adatmodell)  $\implies$  magas szintű lekérdezés, módosítás (például egy légitársaságos adatbázisnál kategóriák lehetnek: utas, járat, pilóta)

## Adatbáziskezelő rendszerek (DBMS) jellemzői

- Komplex hardver-szoftver rendszer adatok kezelésére (tárolás, módosítás, lekérdezés)
- Nagy adatmennyiség
- Gazdag logikai struktúra (adatmodell)  $\implies$  magas szintű lekérdezés, módosítás (például egy légitársaságos adatbázisnál kategóriák lehetnek: utas, járat, pilóta)
- hosszú életciklus (jól kell tőnnie a hardver módosításait, a fizikai tárolás változásait)

## Elvárások a DBMS-mel szemben

- Új séma létrehozása (fogalmak, metaadatok megadása) pl.: legyen egy *utas* kategória, ennek attribútumai legyenek: *név*, *lakcím*, *vegetáriánus-e*,...
- Ehhez adott egy rendszertől függő **DDL** (Data Definition Language)



## Elvárások a DBMS-mel szemben

- Új séma létrehozása (fogalmak, metaadatok megadása) pl.: legyen egy *utas* kategória, ennek attribútumai legyenek: *név*, *lakcím*, *vegetáriánus-e*, . . .  
Ehhez adott egy rendszertől függő **DDL** (Data Definition Language)
- Adatok beillesztése, módosítása (az adatbázis fogalmi keretének feltöltése) pl.: új *utas* felvétele, *elromlott gép* törlése . . .  
Ehhez adott egy rendszertől függő **DML** (Data Manipulation Language)

## Elvárások a DBMS-mel szemben

- Új séma létrehozása (fogalmak, metaadatok megadása) pl.: legyen egy *utas* kategória, ennek attribútumai legyenek: *név*, *lakcím*, *vegetáriánus-e*, ...  
Ehhez adott egy rendszertől függő **DDL** (Data Definition Language)
- Adatok beillesztése, módosítása (az adatbázis fogalmi keretének feltöltése)  
pl.: új *utas* felvétele, *elromlott gép* törlése ...  
Ehhez adott egy rendszertől függő **DML** (Data Manipulation Language)
- Lekérdezés (infót kiszedni az adatbázisból)  
Ehhez **Query Language** (pl.: **SQL**)  
**Cél:** gyakori kérdéseket könnyű legyen kérdezni (ehhez jól végiggondolt fogalmi keret kell, hatékony tárolás  $\implies$  fontos a jó tervezés)

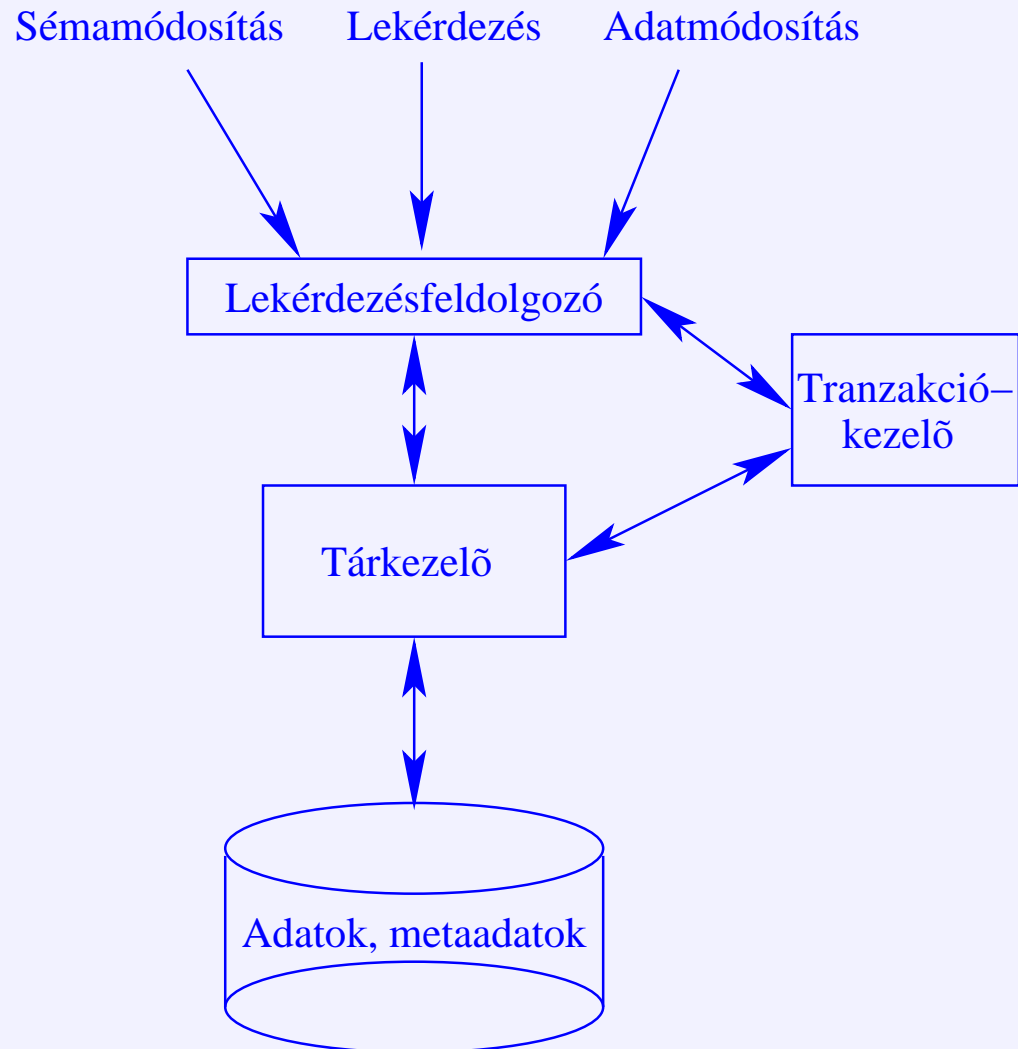
## Elvárások a DBMS-mel szemben

- Új séma létrehozása (fogalmak, metaadatok megadása) pl.: legyen egy *utas* kategória, ennek attribútumai legyenek: *név*, *lakcím*, *vegetáriánus-e*, ...  
Ehhez adott egy rendszertől függő **DDL** (Data Definition Language)
- Adatok beillesztése, módosítása (az adatbázis fogalmi keretének feltöltése) pl.: *új utas felvétele*, *elromlott gép törlése* ...  
Ehhez adott egy rendszertől függő **DML** (Data Manipulation Language)
- Lekérdezés (infót kizsedni az adatbázisból)  
Ehhez **Query Language** (pl.: **SQL**)  
**Cél:** gyakori kérdéseket könnyű legyen kérdezni (ehhez jól végiggondolt fogalmi keret kell, hatékony tárolás  $\implies$  fontos a jó tervezés)
- Nagy mennyiségű adat tárolása biztonságosan (jogosultságok, illetéktelen hozzáférés megakadályozása illetve rendszerhibák elleni védelem)

## Elvárások a DBMS-mel szemben

- Új séma létrehozása (fogalmak, metaadatok megadása) pl.: legyen egy *utas* kategória, ennek attribútumai legyenek: *név*, *lakcím*, *vegetáriánus-e*, ...  
Ehhez adott egy rendszertől függő **DDL** (Data Definition Language)
- Adatok beillesztése, módosítása (az adatbázis fogalmi keretének feltöltése) pl.: *új utas felvétele*, *elromlott gép törlése* ...  
Ehhez adott egy rendszertől függő **DML** (Data Manipulation Language)
- Lekérdezés (infót kiszedni az adatbázisból)  
Ehhez **Query Language** (pl.: **SQL**)  
**Cél:** gyakori kérdéseket könnyű legyen kérdezni (ehhez jól végiggondolt fogalmi keret kell, hatékony tárolás  $\implies$  fontos a jó tervezés)
- Nagy mennyiségű adat tárolása biztonságosan (jogosultságok, illetéktelen hozzáférés megakadályozása illetve rendszerhibák elleni védelem)
- Többfelhasználós működés támogatása (egyidejű hozzáférés, de ne legyen hibás, inkonzisztens állapot emiatt)

# A DBMS felépítése, részei



## Adatok, metaadatok

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

## Adatok, metaadatok

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

## Tárkezelő

A kért **információ beolvasása, módosítása**. Kb. mint az OR-ek file kezelője, de itt néha (**többfelhasználós működésnél pl.**) mi mondjuk meg, hova történjen az írás (**háttártárra, pufferbe**), nem kezelheti teljesen szabadon a puffert.

## Adatok, metaadatok

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

## Tárkezelő

A kért **információ beolvasása, módosítása**. Kb. mint az OR-ek file kezelője, de itt néha (**többfelhasználós működésnél pl.**) mi mondjuk meg, hova történjen az írás (**háttártárra, pufferbe**), nem kezelheti teljesen szabadon a puffert.

### Részei:

- **File kezelő**: nyilvántartja a DB állományát; fizikai I/O-t végez, ha a pufferkezelő kéri; indexstruktúrába rendezni az adatokat (pl. B-fa) (**I/O mindig lemezblokkonként, hatékonyság mértéke az ilyen műveletek száma**)



## Adatok, metaadatok

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

## Tárkezelő

A kért **információ beolvasása, módosítása**. Kb. mint az OR-ek file kezelője, de itt néha (**többfelhasználós működésnél pl.**) mi mondjuk meg, hova történjen az írás (**háttártárra, pufferbe**), nem kezelheti teljesen szabadon a puffert.

### Részei:

- **File kezelő:** nyilvántartja a DB állományát; fizikai I/O-t végez, ha a pufferkezelő kéri; indexstruktúrába rendezni az adatokat (pl. B-fa) (**I/O mindig lemezblokkonként, hatékonyság mértéke az ilyen műveletek száma**)
- **Pufferkezelő:** kezeli a memóriát, tárolja a filekezelő által beolvasott blokkokat

# Lekérdezésfeldolgozó

**Alapfeladata:** sémadefiníciós, adatmódosítás és lekérdezős kérések fogadása, kezelése

# Lekérdezésfeldolgozó

**Alapfeladata:** sémadefiníciós, adatmódosítás és lekérdezős kérések fogadása, kezelése

- **Sémaműveletek:** a DB logikai struktúrájának kialakítása, módosítása  
eredménye: maga az adatbázisséma, plusz kiegészítő metaadatok (pl. mit hogyan tároljunk, indexek)  
Nagyon fontos, meghatározza a további működést

# Lekérdezésfeldolgozó

**Alapfeladata:** sémadefiniálás, adatmódosítás és lekérdezős kérések fogadása, kezelése

- **Sémaműveletek:** a DB logikai struktúrájának kialakítása, módosítása  
eredménye: maga az adatbázisséma, plusz kiegészítő metaadatok (pl. mit hogyan tároljunk, indexek)  
Nagyon fontos, meghatározza a további működést
- **Lekérdezések:** keresőkérdések a DB-re vonatkozóan  
két lehetőség erre: vagy egy külön lekérdezőfelületen át vagy alkalmazói programból (a második esetben a host language-nek van utasításkészlete a DB-hez fordulásra)

# Lekérdezésfeldolgozó

**Alapfeladata:** sémadefiniálás, adatmódosítás és lekérdezős kérések fogadása, kezelése

- **Sémaműveletek:** a DB logikai struktúrájának kialakítása, módosítása  
eredménye: maga az adatbázisséma, plusz kiegészítő metaadatok (pl. mit hogyan tároljunk, indexek)  
Nagyon fontos, meghatározza a további működést
- **Lekérdezések:** keresőkérdések a DB-re vonatkozóan  
két lehetőség erre: vagy egy külön lekérdezőfelületen át vagy alkalmazói programból (a második esetben a host language-nek van utasításkészlete a DB-hez fordulásra)
- **Adatmódosítás:** a DB tartamának módosítása, beszúrás, törlés  
Itt is lehet külön felület vagy program

# Lekérdezésfeldolgozó tennivalói

Hogyan kezeli ezeket a kéréseket?

# Lekérdezésfeldolgozó tennivalói

Hogyan kezeli ezeket a kéréseket?

## 1. Végrehajtási terv készítése

A magas szintű kérdéseket átalakítjuk elemi utasítások sorozatává.

A (legtöbbször) deklaratív kérdésből (ahol nem mondjuk meg, hogy milyen úton akarom megkapni az eredményt, csak azt, hogy mit akarok, pl. SQL) procedurális kérdést csinálunk (ahol már egy konkrét végrehajtási terv látszik).

# Lekérdezésfeldolgozó tennivalói

Hogyan kezeli ezeket a kéréseket?

## 1. Végrehajtási terv készítése

A magas szintű kérdéseket átalakítjuk elemi utasítások sorozatává.  
A (legtöbbször) deklaratív kérdésből (ahol nem mondjuk meg, hogy milyen úton akarom megkapni az eredményt, csak azt, hogy mit akarok, pl. SQL) procedurális kérdést csinálunk (ahol már egy konkrét végrehajtási terv látszik).

Pl: Ha egy banki séma két relációja  
*ügyfél*(név, cím, számszám) és  
*számla*(számszám, számlaszám, egyenleg)  
és mi Bill Gates számlájának egyelegét szeretnénk megkapni, akkor erre egy SQL kérdés:



```
SELECT egyenleg FROM ügyfél, számla  
WHERE ügyfél.szemszám=számla.szemszám AND ügyfél.név="Bill Gates"
```

```
SELECT egyenleg FROM ügyfél, számla  
WHERE ügyfél.szemszám=számla.szemszám AND ügyfél.név="Bill Gates"
```

Ez csak azt mondja meg, hogy mely relációkból, mit akarok megkapni, de azt nem, hogy hogyan kell ezt megszerezni. Erre egy (elnagyolt) végrehajtási terv lehet pl. az, hogy ha van index a névre az *ügyfél*-ben, akkor az alapján keressük meg B.G. személyi számát, aztán ez alapján a másik relációban keressük meg a megfelelő sort és olvassuk ki az egyenleget.

```
SELECT egyenleg FROM ügyfél, számla  
WHERE ügyfél.szemszám=számla.szemszám AND ügyfél.név="Bill Gates"
```

Ez csak azt mondja meg, hogy mely relációkból, mit akarok megkapni, de azt nem, hogy hogyan kell ezt megszerezni. Erre egy (elnagyolt) végrehajtási terv lehet pl. az, hogy ha van index a névre az *ügyfél*-ben, akkor az alapján keressük meg B.G. személyi számát, aztán ez alapján a másik relációban keressük meg a megfelelő sort és olvassuk ki az egyenleget.

## 2. "Optimalizálás"

Több lehetséges végrehajtási terv közül kiválasztani egy "legjobbat". Nem valódi optimalizálás, nem a legjobbat keressük, csak egy elég jót.

```
SELECT egyenleg FROM ügyfél, számla  
WHERE ügyfél.szemszám=számla.szemszám AND ügyfél.név="Bill Gates"
```

Ez csak azt mondja meg, hogy mely relációkból, mit akarok megkapni, de azt nem, hogy hogyan kell ezt megszerezni. Erre egy (elnagyolt) végrehajtási terv lehet pl. az, hogy ha van index a névre az *ügyfél*-ben, akkor az alapján keressük meg B.G. személyi számát, aztán ez alapján a másik relációban keressük meg a megfelelő sort és olvassuk ki az egyenleget.

## 2. "Optimalizálás"

Több lehetséges végrehajtási terv közül kiválasztani egy "legjobbat". Nem valódi optimalizálás, nem a legjobbat keressük, csak egy elég jót.

### Tanulság:

1. A kapott kérdés nem ad támpontot, hogy hogyan kell megkapni az eredményt, de nem is kötelez semmire.

**SELECT** egyenleg **FROM** ügyfél, számla  
**WHERE** ügyfél.szemszám=számla.szemszám **AND** ügyfél.név="Bill Gates"

Ez csak azt mondja meg, hogy mely relációkból, mit akarok megkapni, de azt nem, hogy hogyan kell ezt megszerezni. Erre egy (elnagyolt) végrehajtási terv lehet pl. az, hogy ha van index a névre az *ügyfél*-ben, akkor az alapján keressük meg B.G. személyi számát, aztán ez alapján a másik relációban keressük meg a megfelelő sort és olvassuk ki az egyenleget.

## 2. "Optimalizálás"

Több lehetséges végrehajtási terv közül kiválasztani egy "legjobbat". Nem valódi optimalizálás, nem a legjobbat keressük, csak egy elég jót.

### Tanulság:

1. A kapott kérdés nem ad támpontot, hogy hogyan kell megkapni az eredményt, de nem is kötelez semmire.
2. Érdeemes szöszölni egy jobb végrehajtási terv keresésével, mert nagyok lehetnek a különbségek.

# Tranzakciókezelő

Két nagyobb problémakör megoldására jó:

- Több felhasználó egyszerre használja a DB-t, egyidejű hozzáférések kezelése

# Tranzakciókezelő

Két nagyobb problémakör megoldására jó:

- Több felhasználó egyszerre használja a DB-t, egyidejű hozzáférések kezelése
- Rendszerhibák, ABORT-ok hatásainak kivédése: ezek bekövetkeztekor sem veszhetnek el adatok, nem maradhat a DB inkonzisztens állapotban

# Tranzakciókezelő

Két nagyobb problémakör megoldására jó:

- Több felhasználó egyszerre használja a DB-t, egyidejű hozzáférések kezelése
- Rendszerhibák, ABORT-ok hatásainak kivédése: ezek bekövetkeztekor sem veszhetnek el adatok, nem maradhat a DB inkonzisztens állapotban

Ezek megoldására: alapfogalom a

**tranzakció**: egy felhasználóhoz tartozó, összetartozó utasítások olyan sorozata, melyek vagy mind végrehajtódnak vagy semelyik sem (ez a tulajdonság az atomiság)



# Tranzakciókezelő

Két nagyobb problémakör megoldására jó:

- Több felhasználó egyszerre használja a DB-t, egyidejű hozzáférések kezelése
- Rendszerhibák, ABORT-ok hatásainak kivédése: ezek bekövetkeztekor sem veszhetnek el adatok, nem maradhat a DB inkonzisztens állapotban

Ezek megoldására: alapfogalom a

**tranzakció**: egy felhasználóhoz tartozó, összetartozó utasítások olyan sorozata, melyek vagy mind végrehajtódnak vagy semelyik sem (**ez a tulajdonság az 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

## Elvárások a tranzakciókezelésben

- **A** (atomicity, atomiság): egy tranzakció vagy teljesen végrehajtódik vagy semmi se hajtódik végre belőle
- **C** (consistency, konzisztencia): vannak a rendszernek helyes, konzisztens állapotai, ezek között mozog. Inkonzisztens állapot csak ideiglenesen lehet.

## Elvárások a tranzakciókezelésben

- **A** (atomicity, atomiság): egy tranzakció vagy teljesen végrehajtódik vagy semmi se hajtódik végre belőle
- **C** (consistency, konzisztencia): vannak a rendszernek helyes, konzisztens állapotai, ezek között mozog. Inkonzisztens állapot csak ideiglenesen lehet.
- **I** (isolation, elkülönítés): több tranzakció egyidejű lefutása után úgy nézzen ki a rendszer, mintha a tranzakciók egymás után, elkülönülten futottak volna le.

## Elvárások a tranzakciókezelésben

- **A** (atomicity, atomiság): egy tranzakció vagy teljesen végrehajtódik vagy semmi se hajtódik végre belőle
- **C** (consistency, konzisztencia): vannak a rendszernek helyes, konzisztens állapotai, ezek között mozog. Inkonzisztens állapot csak ideiglenesen lehet.
- **I** (isolation, elkülönítés): több tranzakció egyidejű lefutása után úgy nézzen ki a rendszer, mintha a tranzakciók egymás után, elkülönülten futottak volna le. Tehát pl. az alábbi párhuzamos lefutás:



hatása legyen ugyanaz, mint vagy a tr1 tr2 sorrend, vagy a tr2 tr1 sorrend hatása.

## Elvárások a tranzakciókezelésben

- **A** (atomicity, atomiság): egy tranzakció vagy teljesen végrehajtódik vagy semmi se hajtódik végre belőle
- **C** (consistency, konzisztencia): vannak a rendszernek helyes, konzisztens állapotai, ezek között mozog. Inkonzisztens állapot csak ideiglenesen lehet.
- **I** (isolation, elkülönítés): több tranzakció egyidejű lefutása után úgy nézzen ki a rendszer, mintha a tranzakciók egymás után, elkülönülten futottak volna le. Tehát pl. az alábbi párhuzamos lefutás:



hatása legyen ugyanaz, mint vagy a tr1 tr2 sorrend, vagy a tr2 tr1 sorrend hatása.

- **D** (durability, tartósság): ha egy tranzakció sikeresen befejeződött, akkor annak eredménye nem veszhet el később.

## Eszközök a tranzakciókezelésben- ízelítő

- **Zárolás:** ha egy tranzakció csinálni akar valamit egy adattal, akkor zárat rak rá, ekkor más nem, vagy csak korlátozottan fér hozzá (lehet többféle zárat is használni). Zárolni lehet sort, blokkot, egész táblát is: minél nagyobbat zárolunk, annál könnyebb az adminisztráció, de annál többet kell a tranzakcióknak várniuk egymásra.

## Eszközök a tranzakciókezelésben- ízelítő

- **Zárolás:** ha egy tranzakció csinálni akar valamit egy adattal, akkor zárat rak rá, ekkor más nem, vagy csak korlátozottan fér hozzá (lehet többféle zárat is használni). Zárolni lehet sort, blokkot, egész táblát is: minél nagyobbat zárolunk, annál könnyebb az adminisztráció, de annál többet kell a tranzakcióknak várniuk egymásra.
- **Naplózás:** van egy biztos, hibáktól védett helyen tartott napló, ahol minden történést feljegyzünk. Hiba esetén ez megmarad, innen vissza lehet állítani egy helyes állapotot.



## Eszközök a tranzakciókezelésben- ízelítő

- **Zárolás:** ha egy tranzakció csinálni akar valamit egy adattal, akkor zárat rak rá, ekkor más nem, vagy csak korlátozottan fér hozzá (lehet többféle zárat is használni). Zárolni lehet sort, blokkot, egész táblát is: minél nagyobbat zárolunk, annál könnyebb az adminisztráció, de annál többet kell a tranzakcióknak várniuk egymásra.
- **Naplózás:** van egy biztos, hibáktól védett helyen tartott napló, ahol minden történést feljegyzünk. Hiba esetén ez megmarad, innen vissza lehet állítani egy helyes állapotot.

Ez az eddig felsorolt sok alkotórész eloszlik a kliens és a szerver között. A szerver tartja a kapcsolatot az fizikai adatbázissal, a kliens pedig a felhasználóval, a többi funkció eloszlása nagyon változhat rendszertől függően.

## Az adatbáziskezelő használati szintjei

- **Felhasználó:** egyszerűbb SQL kérdést fogalmaz meg, adatmódosítást csinál, alkalmazói programot futtat

## Az adatbáziskezelő használati szintjei

- **Felhasználó:** egyszerűbb SQL kérdést fogalmaz meg, adatmódosítást csinál, alkalmazói programot futtat
- **Adatbázis programozó:** összetett kérdések írása, alkalmazói program írása. Jól ismeri a DB felépítését, a DBMS-ben használt technikákat.

## Az adatbáziskezelő használati szintjei

- **Felhasználó:** egyszerűbb SQL kérdést fogalmaz meg, adatmódosítást csinál, alkalmazói programot futtat
- **Adatbázis programozó:** összetett kérdések írása, alkalmazói program írása. Jól ismeri a DB felépítését, a DBMS-ben használt technikákat.
- **Adatbázis tervező:** sémát hoz létre, fizikai szervezésbe beleszólhat.

## Az adatbáziskezelő használati szintjei

- **Felhasználó:** egyszerűbb SQL kérdést fogalmaz meg, adatmódosítást csinál, alkalmazói programot futtat
- **Adatbázis programozó:** összetett kérdések írása, alkalmazói program írása. Jól ismeri a DB felépítését, a DBMS-ben használt technikákat.
- **Adatbázis tervező:** sémát hoz létre, fizikai szervezésbe beleszólhat.
- **Adatbázisrendszer megvalósító:** az adatbáziskezelőt magát tervezi és írja.

## A félév anyaga

Főleg a tervezés és megvalósítás kérdései:

## A félév anyaga

Főleg a tervezés és megvalósítás kérdései:

- **Tervezés:**
  - ★ **Adatmodellezés:** adatbázis-tervező technikák (ODL, objektumos tervezés és E/K (egyed-kapcsolat) diagram, grafikus jelölésrendszer). Az adatbázis fogalmi keretének megadására jók, tervet lehet velük készíteni, amit aztán majd át kell alakítani az adatbáziskezelő által használt formális megadási módra.

## A félév anyaga

Főleg a tervezés és megvalósítás kérdései:

- **Tervezés:**
  - ★ **Adatmodellezés:** adatbázis-tervező technikák (ODL, objektumos tervezés és E/K (egyed-kapcsolat) diagram, grafikus jelölésrendszer). Az adatbázis fogalmi keretének megadására jók, tervet lehet velük készíteni, amit aztán majd át kell alakítani az adatbáziskezelő által használt formális megadási módra.
  - ★ **Relációs adatmodell:** nagyon fontos, tipikusan ilyenek a DBMS-ek mostanában.



## A félév anyaga

Főleg a tervezés és megvalósítás kérdései:

- **Tervezés:**
  - ★ **Adatmodellezés:** adatbázis-tervező technikák (ODL, objektumos tervezés és E/K (egyed-kapcsolat) diagram, grafikus jelölésrendszer). Az adatbázis fogalmi keretének megadására jók, tervet lehet velük készíteni, amit aztán majd át kell alakítani az adatbáziskezelő által használt formális megadási módra.
  - ★ **Relációs adatmodell:** nagyon fontos, tipikusan ilyenek a DBMS-ek mostanában.
- **Tervezés/megvalósítás:**
  - ★ fizikai szervezés

## A félév anyaga

Főleg a tervezés és megvalósítás kérdései:

- **Tervezés:**
  - ★ **Adatmodellezés:** adatbázis-tervező technikák (ODL, objektumos tervezés és E/K (egyed-kapcsolat) diagram, grafikus jelölésrendszer). Az adatbázis fogalmi keretének megadására jók, tervet lehet velük készíteni, amit aztán majd át kell alakítani az adatbáziskezelő által használt formális megadási módra.
  - ★ **Relációs adatmodell:** nagyon fontos, tipikusan ilyenek a DBMS-ek mostanában.
- **Tervezés/megvalósítás:**
  - ★ fizikai szervezés
  - ★ tranzakciókezelés

## A félév anyaga

Főleg a tervezés és megvalósítás kérdései:

- **Tervezés:**
  - ★ **Adatmodellezés:** adatbázis-tervező technikák (ODL, objektumos tervezés és E/K (egyed-kapcsolat) diagram, grafikus jelölésrendszer). Az adatbázis fogalmi keretének megadására jók, tervet lehet velük készíteni, amit aztán majd át kell alakítani az adatbáziskezelő által használt formális megadási módra.
  - ★ **Relációs adatmodell:** nagyon fontos, tipikusan ilyenek a DBMS-ek mostanában.
- **Tervezés/megvalósítás:**
  - ★ fizikai szervezés
  - ★ tranzakciókezelés
  - ★ lekérdezésfeldolgozás

## A félév anyaga

Főleg a tervezés és megvalósítás kérdései:

- **Tervezés:**
  - ★ **Adatmodellezés:** adatbázis-tervező technikák (ODL, objektumos tervezés és E/K (egyed-kapcsolat) diagram, grafikus jelölésrendszer). Az adatbázis fogalmi keretének megadására jók, tervet lehet velük készíteni, amit aztán majd át kell alakítani az adatbáziskezelő által használt formális megadási módra.
  - ★ **Relációs adatmodell:** nagyon fontos, tipikusan ilyenek a DBMS-ek mostanában.
- **Tervezés/megvalósítás:**
  - ★ fizikai szervezés
  - ★ tranzakciókezelés
  - ★ lekérdezésfeldolgozás
- **Programozás:** nem nagyon lesz, majd laboron a következő félévben, de azért: SQL, adatmódosítás, adatdefiniálás, triggerek, megszorítások kezeléséről lesz szó.