Halász Noel órai feladat

1. Adatbázis jelentése: Minimális definícióként azt mondhatjuk, hogy az adatbázis egymáshoz kapcsolódó információk bármilyen gyűjteménye. Ha egy bevásárlólistát írunk fel egy papírlapra, máris egy kis analóg adatbázist hozunk létre. De mit nevezünk adatbázisnak a számítástudományban? Itt az "adatbázist" úgy definiáljuk, mint információk olyan gyűjteményét, amelyet adatokként tárolunk egy számítógépes rendszeren. Ilyen lehet többek között a környékbeli szupermarket készletleltára is. Forrás: <https://azure.microsoft.com/hu-hu/resources/cloud-computing-dictionary/what-are-databases>
2. Adatbázis fajták: A jelenleg működő korszerű adatbázisok négy alapvető adatmodellt realizálnak:

* [- a hierarchikus adatmodellt,](http://www.agt.bme.hu/tutor_h/terinfor/t12.htm#fa)
* [- a hálós adatmodellt,](http://www.agt.bme.hu/tutor_h/terinfor/t12.htm#halo)
* [- a relációs adatmodellt](http://www.agt.bme.hu/tutor_h/terinfor/t12.htm#rel)
* [- objektum orientált adatmodelt.](http://www.agt.bme.hu/tutor_h/terinfor/t12.htm#obj)

Forrás: <http://www.agt.bme.hu/tutor_h/terinfor/t12.htm>

1. SQL jelentése: A fentebb említett adatbázisok kezelésére adatbázis-kezelőket használunk. Az SQL (ami az angol Structured Query Language szóra utal) egy relációsadatbázis-kezelő lekérdezési nyelv. Bár nem tekintjük programozási nyelvnek, de mégis hasonlóan kell elképzelni, itt is vannak utasítások, amelyek segítségével műveleteket végezhetünk. Létrehozhatunk adatbázisokat, adatokat adhatunk az adatbázishoz, törölhetünk belőle, módosíthatjuk az egyes adatokat, lekérdezhetjük azokat, vagy rendezhetjük.

Forrás: <https://webiskola.hu/sql-ismeretek/relacios-adatbazis-sql-fogalmak-peldak/>

1. Programozási nyelvek generációi:
   1. *Nulladik generáció:* Ez nem hivatalos elnevezés. Gyakorlatilag az egyedi, gépközeli nyelvekről van szó a legelső számítógépekben (pl.: ENIAC, EDVAC). A kód számok sorozata; általában egy szám egy utasítás. Relatíve sok utasítás. Nincs változónév, sem eljárásnév, sem ciklus. Minden egyes memóriacímre hivatkozás számkóddal történik.
   2. *Első generáció (1GL):* A programkódok megjelenése. Elemi típusok megszületése (egész, valós, logikai, karakter, tömb, rekord). Alapvető utasítások kialakulása (elágazás, ciklus, értékadás, eljárás). Hozzájuk tartozó elmélet: assemblerek, kódgenerálás, makroprocesszor, hatékonyság.
   3. *Második generáció (2GL):* Alapvető utasítások finomodása, adattípusok és kifejezések kialakulása (pl.: fa, teljes zárójelezettség, lengyel formula). Hozzájuk tartozó elmélet: szintaxis, kifejezés-kiértékelés. A nyelvek immár nem túlságosan gép-közeliek, de magukat a nyelveket többnyire egy-egy problémacsoport megoldására lehetett alkalmazni. Az egyes utasítások viszont processzor-családra és környezetre jellemzőek.  A kódokat közvetlenül a programozók írják és (főleg ők) olvassák. Hátrány: egyes CPU-utasításításokban kellett gondolkodni, nem magasabb szintű logikában; kezelni kellett a memóriát és más hardver-részeket
   4. *Harmadik generáció (3GL):* Magas szintű általános nyelvek megjelenése. Jellemzők: típusok további finomodása, strukturált programozás, nagyobb gép-függetlenség és programozó-barátibb megvalósítás. Az alapvető utasításokról itt már a számítógép (és a fordító) gondoskodik, tehát nem a programozónak kell minden részlettel törődnie. Jellemző a gép-függetlenség. Az első (főleg az Algol 60-ra épülő) speciális nyelvek megjelenése. Elméleti fogalmak: helyesség-bizonyítás, programszintézis, szemantika-definiálás, szimbolikus végrehajtás, strukturált programozás, objektum-orientáltság.
   5. *Negyedik generáció (4GL):* modularitás, specifikáció, absztrakt típus (értékhalmaz és műveletek). A legelső ilyen nyelvek az 1980-as évek közepén jelentek meg. Előnyük: hatékonyabb, egyszerűbb, biztonságosabb programkészítés – a nagyobb logikai egységek definiálása és mozgatása a konkrét bitekre és bájtokra koncentrálás helyett. A nyelvek logikája közelít az emberi gondolkodáshoz. Támogatott a grafikus felület (GUI), az adatbázis-kezelés, a jelentés-készítés, a matematikai optimalizálás, valamint a webfejlesztés. Sok tekintetben egybemosódik a 3GL és a 4GL.
   6. *Ötödik generáció (5GL):* párhuzamosság, hálózatok programozása. Felhasználása pl. központilag vezérelt számítógép-hálózatok, döntéstámogató rendszerek fejlesztése; valamint a mesterséges intelligenciát fejlesztők eszköze. Ezek a nyelvek általában már alkalmasak arra, hogy a számítógép a problémát a programozó nélkül oldja meg. Több nyelv is a LISP-re épül és átvett onnan elemeket. Sokáig úgy vélték, hogy az 5. generációs nyelvek elavulttá teszik, így felváltják a régebbi nyelveket, de kiderült, hogy egyes elemekhez feltétlenül szükséges van emberi programozóra, így továbbra is van létjogosultsága a régebbi nyelveknek.

*Forrás:* <https://tferi.hu/programozasi-nyelvek-generacioi>

1. E-K model és részei:

Az egyed valamilyen típusú absztrakt objektum. Hasonló egyedek összessége egyedhalmazt alkot. Az egyed fogalma bizonyos értelemben hasonlít az objektumorientált programozás objektumfogalmához; hasonló viszony mondható el egyegyedhalmaz és objektumok egy osztálya között. Ugyanakkor az E/K modell statikus fogalom, mely csupán az adatok szerkezetéről szól, a rajtuk végezhető műveletekről nem. Így tehát az osztályoktól eltérően, az egyedhalmazokhoz nem tartoznak metódusok. Az E/K egyes változataiban az attribútumok típusai lehetnek: 1. atomiak, mint ahogy az itt bemutatott változatban, 2. rekordok, adott számú atomi komponensből felépítve (a C nyelvű ”sruct”-hoz hasonlóan). 3. atomi vagy rekord típusú elemekből álló halmazok, adott elemtípussal. Például egy attribútum típusa lehetne párok halmaza, melyek első eleme egész szám, második eleme karaktersorozat.

Forrás: https://people.inf.elte.hu/zirtaai/adatb1/adat1\_kidolgozott\_tetelek\_EKmodellek.pdf