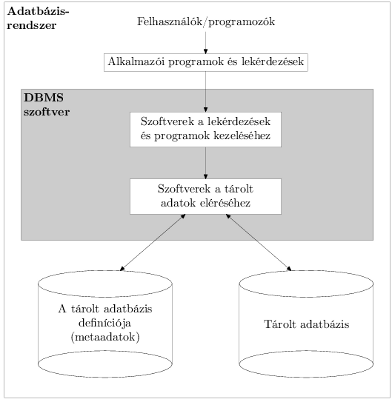
**Adatbázis, adatbázisrendszer, adatbázis-kezelő rendszer (DBMS) fogalma és jellemzői**

* Adatbázis: Egymással logikailag összefüggő, egymáshoz kapcsolódó, belsõ jelentéssel bíró adatok összessége (kollekciója, együttese).
  + Az adatbázis a következő implicit tulajdonságokkal rendelkezik:
  + Az adatbázis a valós világ valamely részét reprezentálja, amelyet néha minivilágnak vagy a modellezés tárgyának nevezünk. A minivilágban bekövetkező változások megjelennek az adatbázisban.
  + Az adatbázis adatok logikailag összetartozó gyűjteménye a benne rejlő jelentéssel együtt. Az adatok egy véletlenszerű összességét nem tekinthetjük adatbázisnak.
  + Az adatbázist egy konkrét céllal tervezzük és építjük, továbbá konkrét céllal tárolt adatokkal töltjük föl. Jól meghatározott felhasználói csoport használja jól meghatározott, számukra értelmes céllal.
  + Más szóval az adatbázisnak van valamilyen forrása, ahonnan az adatok származnak, valamilyen szinten kölcsönhatásban áll a valós világ eseményeivel, és van egy közönsége, amelyet aktívan érdekel a tartalma. Egy adatbázis tetszőleges méretű és összetettségű lehet.
  + A DBMS egy általános célú szoftverrendszer, amely lehetővé teszi különböző felhasználók és alkalmazások számára az adatbázisok definiálását, létrehozását, manipulálását és megosztását
* Adatbáziskezelő rendszer (DBMS): Olyan szoftvercsomag/rendszer, amely számítógépes adatbázisok létrehozását és karbantartását támogatja. Egy általános célú szoftverrendszer, amely lehetővé teszi különböző felhasználók és alkalmazások számára az adatbázisok definiálását, létrehozását, manipulálását és megosztását.
* Adatbázisrendszer: az adatbázist, a felhasználókat, a DBMS szoftvert, illetve a DBMS-t futtató és az adatbázist tároló hardvert együtt adatbázisrendszernek nevezzük.



**Egyed, tulajdonság és kapcsolat fogalma és tulajdonságai**

**Egyed**

A valós világnak az az eleme (tárgy, jelenség, elképzelés, személy, fogalom stb.), amely a modellezés tárgyát képezi. Egyednek tekinthetjük például: Kovács Péter, másodéves PTI-s hallgatót

**Tulajdonság**

Az egyednek a modellezés szempontjából lényeges jellemzője. Tulajdonságnak tekinthetjük például: a Kovács Péter nevet, a második évfolyamot, a PTI szakot

**Kapcsolat**

A két vagy több egyedtípus egyedei között fennálló viszony. Kapcsolatnak tekinthetjük például: Kovács Péter felvette az Adatbázisrendszerek tárgyat

Az **egyed** egy olyan valós világbeli objektumot vagy fogalmat reprezentál, amely szerepel az adatbázisban, mint például egy alkalmazott vagy egy projekt. A **tulajdonság** vagy **attribútum** az egyedet leíró, a modell szempontjából érdekes jellemző, mint például az alkalmazott neve vagy fizetése. Egy két vagy több egyed közötti **kapcsolat** az egyedek között fennálló valamilyen viszonyt jelöl. Ilyen például a „dolgozik rajta” kapcsolat egy alkalmazott és egy projekt között.

**Relációs, objektum-relációs és NoSQL adatbázisok jellemzése**

Relációs

Előnyei:

* egyszerűség
* matematikai megalapozottság

Alapja a matematikai reláció fogalma, ami hasonló egy értékekkel kitöltött táblázathoz (mátrixhoz).

Elméleti alapjai:

* halmazelmélet
* elsőrendű predikátumkalkulus (matematikai logika)

A relációs modellben az adatbázis relációknak egy halmaza lesz. Minden reláció durván egy flat fájlként fogható fel. A reláció értékek egy táblázata, amely sorok egy halmazából áll.

Minden egyes sor adat elemei a modellezett kisvilág egy egyed-előfordulásáról vagy egy kapcsolat-előfordulásáról tartalmaznak tényeket (információt).

A sorokat a formális modellben elem n-eseknek vagy rekordoknak fogjuk nevezni.

Minden egyes oszlop egy oszlop fejléccel (címmel) rendelkezik, amely az illető oszlopban lévő adatok jelentéséről ad információt. Az oszlop fejléceket a formális modellben attribútum neveknek (röviden attribútumoknak) fogjuk nevezni.

Egy D tartomány atomi értékek egy halmaza.

Atominak nevezünk egy olyan értéket, amely már nem bontható tovább a relációs modell szempontjából.

Relációséma alatt az R(A1;A2; : : : ;An) jelölést értjük, ahol R a relációséma neve, A1;A2; : : : ;An pedig attribútumok. A relációs adatmodellben egy r (R) reláció nem más, mint egy

dom(A1); dom(A2); : : : ; dom(An) tartományokon értelmezett n-ed fokú matematikai reláció, amely részhalmaza azon tartományok Descartes-szorzatának, amelyek R-et definiálják:

r (R) \_ dom(A1) \_ dom(A2) \_ : : : \_ dom(An).

Objektum-relációs

Az objektum-relációs modellek például az alap relációs modellt egészítik ki többek között a viselkedés leírására szolgáló eszközökkel.  
Fő jellemző és előny: a tervező specifikálni tudja mind a sokszorosan összetett objektumok szerkezetét, mind az ezeken az objektumokon alkalmazható műveleteket.  
Az objektum-relációs rendszerek (ORDBMS) a tradicionális RDBMS-ek OO szemlélettel való bővítései.  
Az objektumok az OO programozási nyelvekben tranziensek (átmenetiek), a futás befejeztével törlődnek Az objektumok az ODMS-ekben perzisztensek (állandóak), eltárolódnak, később kinyerhetőek és megoszthatóak más programokkal.

NoSQL

A NoSQL adatbázisok nem táblákban tárolják az adatokat, és általában nem használnak SQL nyelvet lekérdezésre.  
A legtöbb NoSQL adatbázis szerver erősen optimalizált írás és olvasás műveletekre, míg ezen túl nem sok műveletet támogatnak.

A jobb sebesség és a skálázhatóság érdekében olyan adatszerkezeteket használnak tárolásra, mint a kulcs-érték párok, gráfok, dokumentum-adatbázisok vagy oszlopcsaládok.

Leginkább a big data-ban és a real-time web alkalmazásokban használatosak.

**A funkcionális függés fogalma**

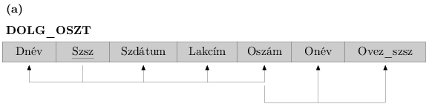
A funkcionális függés egy olyan megszorítás, amely az adatbázis két attribútumhalmaza között áll fenn. Tegyük fel, hogy a relációs adatbázissémánknak *n* attribútuma van: *A*1, *A*2, ..., *An*; és gondoljunk az egész adatbázisunkra úgy, hogy azt egyetlen, **univerzális** *R* = { *A*1, *A*2, …, *An* } relációsémával írjuk le. Ez nem jelenti azt, hogy az adatbázisunkat egyetlen univerzális táblaként fogjuk tárolni; ezt a fogalmat csak az adatfüggőség formális elméletének a kialakításához használjuk.

**Definíció.** Az *R* két attribútumhalmaza, *X* és *Y* között értelmezett, *X* → *Y* alakú **funkcionális függés** egy *megszorítást* ír elő az *R* relációséma bármely *r* relációjának lehetséges rekordjaira. A megszorítás az, hogy bármely két *r*-beli *t*1 és *t*2 rekord esetén, amelyekre *t*1[*X*] = *t*2[*X*] teljesül, teljesülnie kell *t*1[*Y*] = *t*2[*Y*]-nak is.

Ez azt jelenti, hogy *r*-beli rekord *Y* komponensének az értékei *függnek* az *X* komponens értékektől; más szavakkal egy rekord *X* komponensének az értékei egyértelműen (vagy **funkcionálisan**) *meghatározzák* az *Y* komponens értékeit. Azt is mondhatjuk, hogy *X* **funkcionálisan meghatározza** *Y*-t, vagy *Y* **funkcionálisan függ** *X*-től. Az *X* attribútumhalmazt a funkcionális függés **bal oldalának**, az *Y*-t pedig a **jobb oldalának** nevezzük.

Tekintsük a 2-es (b) ábrán szereplő DOLG\_PROJ relációsémát; az attribútumok jelentése alapján tudjuk, hogy a következő funkcionális függéseknek kell fennállniuk:

1. Szsz → Dnév

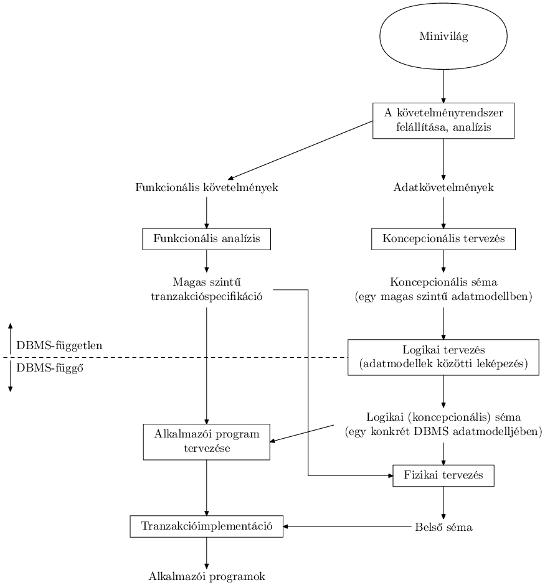


Ezek a funkcionális függések azt mutatják, hogy (a) egy dolgozó személyi száma (Szsz) egyértelműen meghatározza a dolgozó nevét (Dnév)

**Koncepcionális adatbázis-tervezés, az ER modell és leképezése relációs modellre**

A koncepcionális adatmodellek olyan fogalmakat használnak, mint például az egyed, a tulajdonság és a kapcsolat.  
A tervezés lépései:

1. **a követelményrendszer felállítása és analízise**
2. az adatbázis **koncepcionális sémájának** létrehozása: a koncepcionális séma a felhasználók adatkövetelményeinek egy tömör leírása
3. az adatbázis tényleges megvalósítása: ezt a lépést **logikai tervezésnek** vagy **az adatmodell leképezésének** nevezzük
4. **fizikai tervezés:** megadjuk az adatbázis állományainak belső tárolási szerkezetét, indexeit, elérési útjait és állományszervezési módjait



Az ER modell az adatokat mint *egyedeket*, *kapcsolatokat* és *attribútumokat* írja le. Az ER modell által kezelt alapvető objektum az **egyed**, amely a valós világnak egy olyan *darabja*, amely önálló léttel bír. Minden egyednek vannak **attribútumai** — az őt leíró tulajdonságok. Az egyedeket leíró attribútumértékek fogják alkotni az adatbázisban tárolt adatok nagy részét.

Az ER séma leképezése relációs sémára:

1. **Erős egyedtípusok leképezése**
2. **Gyenge egyedtípusok leképezése**
3. **Bináris 1:1 számosságú kapcsolattípusok leképezése**
4. **Bináris 1:N számosságú kapcsolattípusok leképezése**
5. **Bináris M:N számosságú kapcsolattípusok leképezése**
6. **Többértékű attribútumok leképezése**
7. **Az *n*-edfokú kapcsolattípusok leképezése**

**Az SQL elemei: DDL, DML, DCL,** **egyszerű lekérdezések és táblák összekapcsolása**

* **DDL- Data Definition Language**: A DBA és adatbázis tervezők használják azért, hogy az adatbázis koncepcionális sémáját meghatározzák. Az SQL DDL nyelve ad eszközöket a tartománymegszorítás, kulcsmegszorítás és egyéb (not null, unique, stb.) kényszerítésre.
* **DML - Data Manipulation Language**: Arra használjuk, hogy az adatbázisból való keresést illetve az adatbázis frissítését specifikáljuk.  
  A DML utasítások beágyazhatóak olyan általános célú programozási nyelvekbe mint a COBOL, C, C++, JAVA.

Fajtái:

* + magas szintű vagy nem-procedurális nyelvek. Ilyen pl. az SQL. „Halmaz” orientáltak, azt mondják meg, hogy mit keresünk és nem azt, hogy hogyan. Deklaratív nyelvnek is nevezik.
  + alacsony szintű vagy procedurális nyelvek. Az adatokat egy rekord egy időben elv alapján keresik. Ciklusok szükségesek több rekord kinyeréséhez mutatók pozicionálása útján.
* **DCL - Data Control Language:** A DCL nyelv egy, az adatbázisok hozzáférésének szabályozásához használatos nyelv. DCL parancsok például a GRANT és a REVOKE.

SELECT \* FROM surveys

WHERE year >= 2000;

SELECT genus, species

FROM species

WHERE taxa = 'Bird'

ORDER BY species\_id ASC;

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;