

Adatbázisrendszerek

2. előadás: Adatbázis-rendszer

Adatbázisrendszer, adatbázis-kezelő rendszer,
háromséma-architektúra

Ispány Márton

2024. március 1.



**DEBRECENI EGYETEM
INFORMATIKAI KAR**





2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- Egy konkrét adatbázis **definiálása** adattípusai, szerkezete és megszorításai révén.
- A kezdeti adatbázis tartalom **betöltése** a (másodlagos) tároló eszközön.
- Az adatbázis kezelése
 - **Kinyerés (keresés)**: lekérdezések, riportok előállítás.
 - **Módosítás**: beszúrás, törlés és a tartalom frissítése.
 - Az adatbázis elérése web alkalmazásokon keresztül.
- Feldolgozás és megosztás konkurens felhasználók és alkalmazói programok egy halmaza között úgy, hogy az **összes adat érvényes és konzisztens** maradjon.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

További feladatok:

- Védelmi és biztonsági szolgáltatások a jogosulatlan elérés megelőzésére.
- „Aktív” feldolgozás az adatokon való belső műveletek végrehajtására.
- Adatok megjelenítése és vizualizációja.
- Az adatbázis és a kapcsolódó programok karbantartása az adatbázis alkalmazás teljes életciklusa alatt. Ezeket egyenként adatbázis, szoftver és rendszer karbantartásnak nevezzük.



Az adatbázisrendszer önleíró természetete

- Egy DBMS **katalógus** egy önálló adatbázis leírását tárolja (adatszerkezetek, adattípusok, megszorítások).
- A leírás ún. **metaadatokból** (az adatra vonatkozó adat) áll.
- Ez teszi lehetővé, hogy a DBMS szoftver különböző adatbázis alkalmazásokkal működjön együtt.

A programok és az adatok elszigetelése

- Ezt **program-adat függetlenségnek** nevezzük.
- Lehetővé teszi az adatszerkezetek és a tárolás módjának megváltoztatását anélkül, hogy a DBMS-t elérő programot meg kellene változtatni.



Adat absztrakció

- Egy **adatmodell**et használunk arra, hogy a tárolási részleteket elrejtjük és csak az adatbázis koncepcionális képét jelenítsük meg a felhasználóknak.
- A programok az adatmodellre hivatkoznak az adattárolási részletekkel szemben.

Az adatok többféle nézetének támogatása

- Minden felhasználó különböző képet láthat az adatbázisról, ami csak a számára érdekes adatokat írja le.



Adatok megosztása és többfelhasználós tranzakció feldolgozás

- Megengedi **konkurens felhasználó**knak az adatkinyerést és frissítést ugyanazon adatbázisban.
- A konkurencia ellenőrzés a DBMS-en belül garantálja, hogy minden **tranzakció** helyesen hajtódik végre vagy szakad meg.
- A helyreállító alrendszerek biztosítják, hogy minden végrehajtott tranzakció állandó bejegyzésre kerül az adatbázisban.
- A közvetlen tranzakció feldolgozás (**OLTP** - Online Transaction Processing) az adatbázis alkalmazások fő része. Ez teszi lehetővé konkurens (egyidejű) tranzakciók százainak másodpercek alatti végrehajtását.



Adatmodell (naív definíció)

Fogalmak egy olyan összessége, amely leírja az adatbázis szerkezetét, azokat a műveleteket amelyekkel ez a szerkezet módosítható és bizonyos megszorításokat (kényszereket), melyeket az adatbázisnak ki kell elégítenie.

Az adatmodell szerkezete és megszorításai:

- Az adatbázis szerkezetét **konstruktorokkal** (építőelemek) definiáljuk.
- Jellemző konstruktorok az elemek (és az ő adattípusaik), elemek csoportjai (egyed, rekord, tábla) és ezen csoportok közötti kapcsolatok.
- A **megszorítások** korlátozásokat vezetnek be az érvényes adatokon. Ezeket a megszorításokat minden időben kikényszerítjük.



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- Ezeket a műveleteket arra használjuk, hogy specifikáljuk az adatbázisbeli (vissza)keresést és frissítést az adatmodell konstruktoraira való hivatkozással.
- Az adatmodell műveletei állhatnak alpműveletekből (generikus beszúrás, törlés, frissítés) és felhasználó által definiáltakból (számolj_kreditindex).

Adatmodellek fajtái

- **Koncepcionális** (magas szintű, szemantikus): olyan fogalmakkal dolgozik, amelyek közel vannak ahhoz, ahogy a legtöbb felhasználó gondolkodik az adatokról. (Nevezik egyed vagy objektum alapú adatmodellnek is.)
- **Fizikai** (alacsony szintű, belső): olyan fogalmakkal dolgozik, amelyek azt írják le ahogy az adatok eltárolódnak a számítógépben.
- **Implementációs** (reprezentációs): olyan fogalmakkal dolgozik, amelyek a fenti két típus között helyezkednek el. A legtöbb DBMS implementáció ezt használja (pl. a relációs modell).



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- **Adatbázis séma:** az adatbázis leírása. Az adatbázis szerkezetének, az adattípusoknak és a megszorításoknak a leírását tartalmazza.
- **Séma diagram:** az adatbázis séma (több szempontból való) szemléltető megjelenítése.
- **Séma konstruktor:** a séma vagy a sémán belül egy objektum egy **komponense**. Pl.: HALLGATÓ, KURZUS

Adatbázis állapota

- Egy **időpillanatban** az adatbázisban tárolt aktuális adatok összessége.
- Nevezik az adatbázis egy **előfordulásának** (példányának vagy pillanatképének). Az előfordulás kifejezést használjuk az egyedi adatbázis komponensekre is, pl. rekord-előfordulás, egyed-előfordulás.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- **Adatbázis állapot:** az adatbázis egy időpillanatbeli tartalmát jelenti.
- **Kezdeti adatbázis állapot:** az az adatbázis állapot, amelyet kezdetnek betöltünk az adatbázisba.
- **Érvényes állapot:** egy olyan állapot, amely eleget tesz az adatbázis szerkezetének és megszorításainak.
- **Különbség:** Az adatbázis séma nagyon ritkán változik, míg az adatbázis állapot minden alkalommal amikor frissítjük az adatbázist.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- számítógép
- adatok
 - fizikai adatbázis
 - adatszótár (metaadatbázis)
- szoftver
- felhasználók
 - eseti
 - naív vagy parametrikus
 - szakértő
 - adatbázis-adminisztrátor



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

A felhasználók két nagy csoportra oszthatóak:

- Azok, akik valójában használják és ellenőrzik az adatbázis tartalmát, és azok, akik tervezik, fejlesztik és karbantartják az adatbázis alkalmazásokat. (**Szereplők a színpadon**)
- Azok, akik DBMS szoftvereket és kapcsolódó eszközöket terveznek és fejlesztenek, valamint ilyen rendszereket üzemeltetnek. (**Dolgozók a színpad mögött**)



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

Adatbázis adminisztrátorok

- Felelősek az adatbázis ellenőrzött eléréséért, annak használatának koordinálásáért és monitorozásáért, szoftver és hardver eszközök telepítéséért és használatuk ellenőrzéséért és a műveletek hatékonyságának monitorozásáért.

Adatbázis tervezők

- Felelősek a tartalom, a szerkezet, a megszorítások definiálásáért, és az adatbázis felé intézett funkciókért és tranzakciókért. Együtt kell működniük a végfelhasználókkal és meg kell érteniük az igényeiket.



Végfelhasználók

- Az adatokat lekérdezésekre, riportok készítésére használják, illetve néhányan közülük frissítik az adatbázis tartalmát. A végfelhasználók osztályozhatók mint:
 - **eseti** - alkalmyszerűen érik el az adatbázist, amikor az szükséges
 - **naív** vagy parametrikus - a végfelhasználók nagy részét teszik ki

A naív vagy parametrikus felhasználók

- Korábban jól definiált függvényeket használnak „konzerv” tranzakciók formájában az adatbázis felé.
- Példák: bankpénztárosok, helyfoglalással foglalkozó ügynökök, akik ezt a tevékenységet műveletek során keresztül végzik.



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlesztése

Szofisztikált felhasználók

- Üzleti elemzők, tudósok, mérnökök és mások, akik ismerik a rendszer képességeit.
- Sokan olyan eszközt használnak szoftvercsomagok formájában, amelyek a tárolt adatbázishoz közel működnek.

Önálló felhasználók

- Főként személyes adatbázist tartanak fenn készen csomagolt alkalmazásokkal.
- Példa: egy adóbevallást végrehajtó program, amely létrehozza saját belső adatbázisát.
- Példa: egy felhasználó, aki címjegyzéket tart karban.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

Azért javasolták, hogy az alábbi DBMS tulajdonságokat támogassák:

- Program-adat függetlenség
- Az adatok többféle nézetének támogatása

Közvetlenül nem használják az üzleti DBMS termékekben, azonban hasznosnak bizonyult az adatbázisrendszerek szervezésének elmagyarázásában.



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

A DBMS sémákat három szinten definiálja:

- Belső séma belső szinten a szerkezet és az elérési utak (pl. indexek) fizikai tárolásának leírására. Jellemzően fizikai adatmodellt használ.
- Konceptcionális séma konceptcionális szinten a teljes adatbázis szerkezetének és megszorításainak leírására a felhasználók közössége számára. Jellemzően konceptcionális vagy implementációs adatmodellt használ.
- Külső sémák külső szinten a különböző felhasználói nézetek leírására. Rendszerint ugyanazt az adatmodellt használja mint a konceptcionális séma.



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

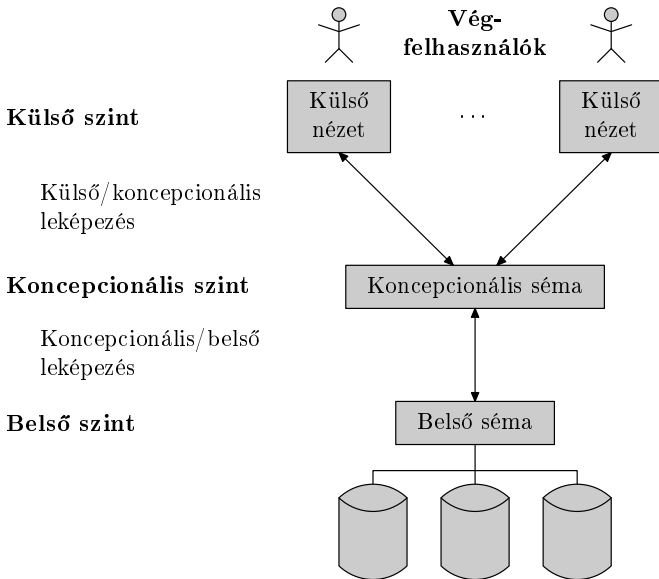
Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése





2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

Séma szintek közötti leképezések szükségesek a követelmények és az adatok közötti transzformációk leírására.

- A programok egy külső sémára hivatkoznak, és leképeződnek a DBMS által a belső sémára, hogy végrehajthatók legyenek.
- A belső DBMS szintből kinyert adatok újraformázzódnak, hogy illeszkedjenek a felhasználó külső nézetéhez. Pl. egy SQL lekérdezés eredménye formázásra kerül, hogy egy weblapon megjelenjen.



- **Logikai adatfüggetlenség:** Annak képessége, hogy a koncepcionális séma anélkül változzon meg, hogy a külső sémáknak és a hozzájuk rendelt alkalmazói programoknak meg kellene változni.
- **Fizikai adatfüggetlenség:** Annak képessége, hogy a belső séma anélkül változzon meg, hogy a koncepcionális sémának meg kellene változnia. Pl. a belső séma megváltozhat azért, hogy bizonyos fájl szerkezeteket átszervezünk vagy új indexeket hozunk létre az adatbázis hatékonyságának a javítása miatt.
- Amikor egy alacsonyabb szintű séma megváltozik, akkor csak ez és az eggyel magasabb szintű sémák közötti leképezésnek kell változnia.
- A magasabb szintű sémák változatlanok maradnak. Ezért az alkalmazói programoknak nem szükséges módosulniuk, mivel azok a külső sémákra hivatkoznak.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- Adatleíró nyelv (**DDL- Data Definition Language**)
- Adatmanipulációs nyelv (**DML - Data Manipulation Language**)
- Magas szintű vagy nem-procedurális nyelvek: ilyen pl. az SQL relációs nyelv. Használhatóak önállóan vagy egy programozási nyelvbe beágyazva.
- Alacsony szintű vagy procedurális nyelvek: ezek egy programozási nyelvbe kell, hogy beágyazva legyenek.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- A DBA (adatbázis adminisztrátor) és adatbázis tervezők használják azért, hogy az adatbázis koncepcionális sémáját meghatározzák.
- Sok DBMS-ben a DDL-t arra is használják, hogy a belső és a külső sémákat (nézeteket) definiálják.
- Egyes DBMS-ekben elkülönített tárolásleíró (SDL - storage definition language) és nézetleíró (VDL - view definition language) nyelveket használnak a belső és külső sémák külön-külön leírására. Az SDL jellemzően a DBA és az adatbázis tervezők által kiadott DBMS utasításokban valósul meg.



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- 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. A programozási nyelvek szintén nyújthatnak olyan függvény-könyvtárakat, melyekkel DBMS-t érhetünk el.
- Alternatív módon, önálló DBMS utasításokat alkalmazhatunk közvetlenül (lekérdező nyelvek nevezik).
- Magas szintű vagy **nem-procedurális** nyelvek. Ilyen pl. az SQL. „Halmaz” orientáltak, azt mondják meg, hogy mit keresünk és nem azt, hogy hogyan. **Deklaratív** nyelvek is nevezik.
- Alacsony szintű vagy **procedurális** nyelvek. Az adatokat egy rekord egy időben elv alapján keresik. Ciklusok szükségesek több rekord kinyeréséhez mutatók pozicionálása útján.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlesztése

- Önálló lekérdező nyelvi interfészek: SQL*Plus az Oracle-ben.
- Programozói interfészek a DML programozási nyelvekbe való beágyazására.
 - Beágyazott megközelítés: beágyazott SQL (C, C++), SQLJ (Java).
 - Eljárás hívás megközelítés: JDBC a Java, ODBC más programozási nyelvek számára.
 - Adatbázis programozási nyelv megközelítés: PL/SQL az Oracle-ben.



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlesztése

- Felhasználóbarát interfészek:
 - menü alapú (web-böngészésre),
 - form alapú (naív felhasználóknak),
 - grafikus (point and click, drag and drop),
 - természetes nyelvi,
 - továbbá ezek kombinációi.
- További interfészek:
 - beszéd alapú,
 - web alapú,
 - parametrikus (bankpénztáros csak bizonyos billentyű kombinációkat használ),
 - DBA interfészek (felhasználói accountok létrehozása, jogosultság-kezelés, a rendszer paramétereinek beállítása, a séma illetve az elérési utak megváltoztatása).



Az adatbázis-kezelő rendszer komponensei és környezete

26

2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

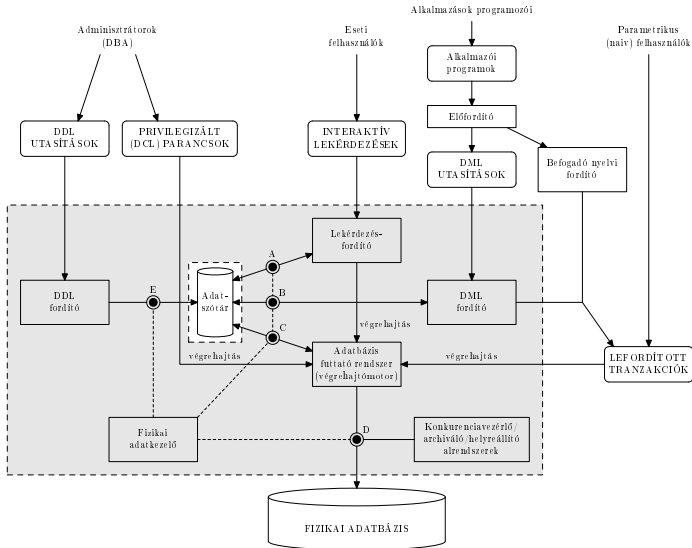
Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése





A következő funkciók végrehajtását támogatják:

- Fájlokban tárolt adatok betöltése az adatbázisba.
- Az adatbázis periodikus mentése szalagra vagy más háttértárra.
- A fájl-szerkezet újraszervezése.
- Riport generálás.
- Hatékonyság monitorozás.
- Rendezés, felhasználók monitorozása, adat-tömörítés stb.

Más eszközök: adatszótár/adatrepozitrium

- Arra használjuk, hogy a sémaleírást és más olyan információt, mint tervezői döntések, alkalmazói programok leírása, felhasználói információk, használati standardok, eltároljuk.
- Az aktív adatszótárat mind a DBMS szoftver, mind a felhasználók és a DBA elérheti, míg a passzív adatszótárat csak az utóbbiak.



- **Centralizált DBMS.** Mindent egy rendszerbe egyesít beleértve a DBMS szoftvert, hardvert, alkalmazói programokat és a felhasználói interfészeket feldolgozó szoftvereket. A felhasználók távoli terminálokon keresztül kapcsolódhatnak a rendszerhez, azonban az összes feldolgozás egy központosított helyen megy végbe.
- **Két rétegű kliens-szerver architektúra.** Több különböző célfeladatra dedikált szerverből (nyomtatás, fájl-kezelés, DBMS, web, email stb.) és kliensekből áll. A kliensek szükség szerint érhetik el a specializált szervereket.
- **Három rétegű kliens-szerver architektúra.** Általánosan elterjedt a webalkalmazások számára. A korábbi két réteg egy közbenső réteggel egészül ki, amelyet **alkalmazás-szervernek** vagy **web-szervernek** nevezünk.



A DBMS szerver:

- Adatbázis lekérdezési és tranzakciós szolgáltatásokat nyújt a klienseknek.
- A relációs DBMS szervert gyakran nevezik SQL szervernek, lekérdező szervernek vagy tranzakciós szervernek.
- A klienseken futó alkalmazások API-t használnak standard interfészekkel a szerver adatbázis elérésére, pl. ODBC, JDBC.
- Kliens és szerver oldalon egyaránt telepítve kell, hogy legyenek a megfelelő szoftver modulok az ODBC vagy JDBC számára.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

A kliensek:

- Megfelelő interfészeket nyújt kliens szoftver modulokon keresztül a különböző szerver szolgáltatások elérésére és használatára.
- Egyaránt lehetnek diszk-mentes vagy diszkkal ellátott PC-k illetve munkaállomások.
- Különböző típusú hálózatokon (LAN, wireless) keresztül kapcsolódnak a szerverekhez.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

- A webhez való kapcsolatot biztosító és az alkalmazás üzleti logikáját, mely az adatbázis-szerver megfelelő adataihoz való hozzáférést biztosítja, tartalmazza.
- Egyfajta vezetékként működve részlegesen feldolgozott adatokat küld az adatbázis-szerver és a kliensek között.
- **Biztonság.** Az adatbázis-szerver csak a közbenső rétegen keresztül érhető el. A kliensek közvetlenül nem érhetik el az adatbázis-szervert.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

Kezdeti adatbázis alkalmazások

- A hierarchikus és hálózati modelleket a 60's évek közepén vezették be és a 70's évekig domináltak.
- Számos olyan világszerte elterjedt adatbázis alkalmazás fut még, amely ilyen, pl. hierarchikus modellt használ.

Relációs modell alapú rendszerek

- A relációs modellt 1970-ben vezették be, erősen kutatták és kísérleteztek vele az IBM Researchnél és több egyetemen.
- Relációs DBMS termékek a 80's évek elején jelentek meg.



Objektum-orientált és haladó alkalmazások

- Az objektum-orientált adatbáziskezelő-rendszereket (OODBMS) a 80's évek végén és a 90's évek elején vezették be CAD és más alkalmazásokban előforduló összetett adatok feldolgozásának igényét kielégítendő.
- Használatuk nem nagyon terjedt el.
- Sok relációs DBMS kibővült objektum adatbázis fogalmakkal egy új kategóriát az ún. **objektum-relációs DBMS**t (ORDBMS) létrehozva.
- A kiterjesztett relációs rendszerek további képességekkel bővültek, pl. multimédia adatok, XML és más adattípusok.



2. előadás: Adatbázis- rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

Adatok a Web és E-kereskedelmi alkalmazásokban

- A Web HTML-ben tartalmazza az adatokat az oldalak közötti linkekkel együtt.
- Ez új alkalmazások sokaságát hívta életre és az e-kereskedelem olyan új szabványokat használ, mint az XML.
- Az olyan script programozási nyelvek mint a PHP és a Javascript dinamikus weboldalak generálását teszik lehetővé, amelyek részben egy adatbázisból generálódnak. Fordítva, szintén megengedik az adatbázis frissítését weblapokon keresztül.



2. előadás:
Adatbázis-
rendszer

Ispány
Márton

Adatbázis-
rendszer,
adatbázis,
adatmodell

Felhasználók

Háromséma-
architektúra

DBMS
nyelvek és
interfészek

Adatbázis-
kezelő
rendszer

Adatbázis-
technológiák
fejlődése

A DBMS-ek új funkcionalitásokkal bővültek az alábbi területeken:

- Tudományos alkalmazások
- XML
- Képek tárolása és kezelése
- Audió és videó adatok kezelése
- Adattárházak és adatbányászat
- Térbeli adatok kezelése (GIS)
- Idősorok és historikus adatok kezelése

A fentiek új kutatási és fejlesztési területeket hívtak életre beleértve új adattípusokat, összetett adatszerkezeteket, új műveleteket és új tárolási és indexelési sémákat az adatbázis-rendszerekben.



A DBMS használatának fő korlátai:

- Magas belépési költségek és további hardver igények.
- Szükségtelen túllépés az általánosság, a biztonság, a konkurencia kontrol, a visszaállítás és az integritási funkciókban.

Amikor a DBMS szükségtelen lehet:

- Ha az adatbázis és az alkalmazások egyszerűek, jól definiáltak és várhatóan nem változnak.
- Ha olyan szoros valós idejű követelmények vannak, melyek nem teszik lehetővé DBMS alkalmazását az (idő)túllépés miatt.
- Ha nem szükséges, hogy az adatokat több felhasználó is elérje.



Amikor a DBMS-mentesség megfelelő:

- Ha az adatbázisrendszer nem képes kezelni az adatok összetettségét a modellezési korlátok miatt.
- Ha az adatbázis felhasználóknak olyan speciális műveletekre van szükségük, amelyet a DBMS nem támogat.