

1.előadás/2.része: Adatbázisok-I.

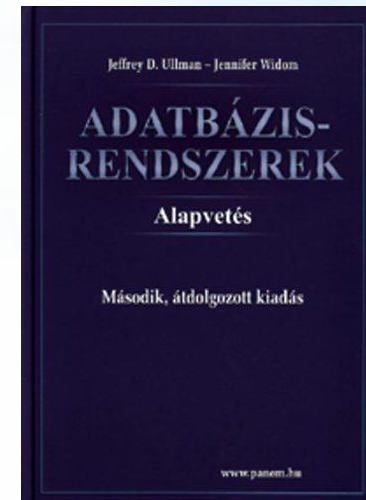
dr. Hajas Csilla (ELTE IK) (2020)

<http://sila.hajas.elte.hu/>

SQL gyakorlatban: SQL DDL és DML

2.3. Relációsémák definiálása SQL-ben
(Táblák létrehozása és előkészítése)

Kiegészítés: Leckék Oracle gyakorlatra
[./sql/bevezetes.pdf](#)



SQL története, szabványok

- Szabvány adatbázis-kezelő nyelv: SQL
- SQL (angol kiejtésben SEQUEL) uis az SQL előfutára IBM fejlesztette ki a 70-es években: SEQUEL → SQL más is volt pl. Ingres : QUEL (ez kalkulus alapú lekérdezés)
- Szabványok (ANSI, ISO)
SQL86, SQL89, SQL92 (SQL2), **SQL:1999** (SQL3),
SQL: 2003, SQL:2006, SQL:2008
- Nyelvjárások (**Oracle**, Sybase, DB2, Progress, MSSQL, mySQL, SQL Server, PostgreSQL, Access,...)
- Az SQL megvalósítások között vannak különbségek, gyakorlatokon az **Oracle SQL**-t nézzük meg részletesen.

SQL fő komponensei

- **Az SQL elsődlegesen lekérdező nyelv** (Query Language)
SELECT utasítás (az adatbázisból információhoz jussunk)
- **Adatkezelő nyelv, DML** (Data Manipulation Language)
INSERT, UPDATE, DELETE, SELECT
- **Sémaleíró nyelv, DDL** (Data Definition Language)
CREATE, ALTER, DROP
- **Adatvezérlő nyelv, DCL** (Data Control Language)
GRANT, REVOKE
- **Tranzakció-kezelés**
COMMIT, ROLLBACK, SAVEPOINT
- **Procedurális kiterjesztések**
SQL/PSM és a gyakorlatban Oracle PL/SQL

SQL DDL

Adatbázis relációsémák definiálása

- Az SQL tartalmaz **adateleíró részt (DDL)**, az adatbázis **objektumainak** a leírására és megváltoztatására. **Objektumok** leíró parancsa a **CREATE** utasítás.
- A relációt az SQL-ben táblának (TABLE) nevezik, az SQL alapvetően háromféle táblát kezel:
 - Alaptáblák (permanens) CREATE TABLE
 - Nézet táblák CREATE VIEW
 - Átmeneti munkatáblák (WITH utasítás)
- **Alaptáblák** megadása: **CREATE TABLE**

Tábla/reláció sémák SQL-ben

- A legegyszerűbb formája:

```
CREATE TABLE relációnév (  
    Attribútum deklarációk listája,  
    Kiegészítő lehetőségek  
);
```

- Az attribútum deklaráció legalapvetőbb elemei:

Attribútumnév típus [kiegészítő lehetőségek]

- itt: a **típus** olyan, amit az SQL konkrét megvalósítása támogat (gyakorlaton Oracle környezetben nézzük meg),
Típusok, pl: INTEGER, REAL, CHAR, VARCHAR, DATE
- **Kiegészítő lehetőségek** például [DEFAULT], [UNIQUE], [PRIMARY KEY], [FOREIGN KEY, REFERENCES], stb.

Egyszerű példák táblák létrehozására

```
CREATE TABLE Sörözők (  
    név CHAR(20) ,  
    város VARCHAR2(40) ,  
    tulaj CHAR(30) ,  
    engedély DATE DEFAULT SYSDATE  
);
```

```
CREATE TABLE Felszolgál (  
    söröző CHAR(20) ,  
    sör VARCHAR2(20) ,  
    ár NUMBER(10,2) DEFAULT 100  
);
```

Az SQL értékekről (bővebben gyakorlaton)

- INTEGER, REAL, stb, a szokásos értékek, számok.
- STRING szintén, de itt egyes-aposztróf közé kell tenni a 'szöveget' (vagyis nem „macskaköröm” közé).
Két egyes-aposztróf = egynek felel meg, például
' Joe' ' s Bar' megfelel a Joe's Bar szövegnek.
- DATE és TIME típusok is vannak az SQL-ben.
- A dátum formátumát meg kell adni DATE 'yyyy-mm-dd'
Például: DATE '2007-09-30' (2007. szept. 30)
- Az idő formátumát is meg kell adni TIME 'hh:mm:ss'
Például: TIME '15:30:02.5' (délután fél 4 múlt két és fél másodperccel)
- Bármely érték lehet **NULL** hiányzó érték:

Hiányzó értékek: NULL

- Az SQL lehetővé teszi a táblákban **a hiányzó értékeket**, vagyis a relációk soraiban az attribútum értéke ne legyen megadva, hanem egy speciális **NULL** nullérték legyen.
- **A nullérték értelmezésére** több lehetőségünk is van:
 - **Nem-ismert érték:** például tudom, „Joe’s Bár”-jának van valamilyen címe, de nem tudom, hogy mi az.
 - **Nem-definiált érték:** például a házastárs attribútumnak egyedülálló embereknél nincs olyan értéke, aminek itt értelme lenne, nincs házastársa, ezért nullérték.
 - stb (van olyan cikk, amely több százféle okot felsorol)

Táblák létrehozása után a táblák feltöltése adatokkal

Eddig láttuk, hogy a CREATE TABLE utasítással hogyan tudunk létrehozni táblákat és megadni a kulcsokat:

- **SQL DDL: sémaleíró nyelv** (Data Definition Language)
CREATE TABLE, ALTER TABLE, DROP TABLE

Most nézzük meg a táblák tartalmának módosítását, hogyan tudjuk INSERT utasítással a táblát feltölteni adatsorokkal:

- **SQL DML: adatkezelő nyelv** (Data Manipulation Language)
INSERT, UPDATE, DELETE, SELECT

SQL DML

Adatbázis tartalmának módosítása

- **A módosító DML utasítások** az adatbázisban tárolt táblák tartalmát változtatják meg.
- 3-féle módosító utasítás létezik:
 - INSERT** - sorok beszúrása
 - DELETE** – sorok törlése
 - UPDATE** – sorok komponensei értékeinek módosítása
- Most a tábla előkészítésénél csak az elsőt nézzük meg, hogy viszünk fel adatsorokat:

Beszúrás (insert into)

Két alakja van:

- 1.) ha egyetlen sort szúrunk be:

INSERT INTO <reláció>

VALUES (<konkrét értékek listája>);

- 2.) INSERT második alakját később tanuljuk hogyan tudunk több sort beolvasni a táblába, egy lekérdezés eredményét:

INSERT INTO <reláció>

(<alkérdés>);

Beszúrás, attribútumok megadása

- **Példa:** A Szeret táblába beírjuk, Zsu szereti a Bud sört.

```
INSERT INTO Szeret  
VALUES('Zsu', 'Bud');
```

- A reláció neve után megadhatjuk az attribútumait.
- Ennek alapvetően két oka lehet:
 1. elfelejtettük, hogy a reláció definíciójában, milyen sorrendben szerepeltek az attribútumok.
 2. Nincs minden attribútumnak értéke, és azt szeretnénk, ha a hiányzó értékeket NULL vagy default értékkel helyettesítenék.

Példa:

```
INSERT INTO Szeret(sör, név)  
VALUES('Bud', 'Zsu');
```

Default értékek megadása

- A CREATE TABLE utasításban az oszlopnevet **DEFAULT** kulcsszó követheti és egy érték.
- Ha egy beszúrt sorban hiányzik az adott attribútum értéke, akkor a default értéket kapja.

```
CREATE TABLE Sörivók (  
    név CHAR(30)  
    cím CHAR(50) DEFAULT 'Sesame St'  
    telefon CHAR(16) );  
INSERT INTO Sörivók(név)  
VALUES ('Zsu');
```

Az eredmény sor:

név	cím	telefon
Zsu	Sesame St	NULL

Tankönyv példa: Filmek séma

Filmek(

cím:string,
év:integer,
hossz:integer,
műfaj:string,
stúdióNév:string,
producerAzon:integer)

FilmSzínész(

név:string,
cím:string,
nem:char,
születésiDátum:date)

Stúdió(

név:string,
cím:string,
elnökAzon:integer)

Mit jelentenek az aláhúzások?

Tankönyv példája, hibás fordítás:

title=(film)cím és address=(lak)cím

Tervezéssel később foglalkozunk, ez a példa hibás, az elnevezések, de így jó lesz, hogy a lekérdezéseknél megnézzük hogyan kezeljük.

SzerepelBenne(

filmCím:string,
filmÉv:integer,
színészNév:string)

GyártásIrányító(

név:string,
cím:string,
azonosító:integer,
nettóBevétel:integer)

Példa megszorításokra: Kulcs

- Előző példában: attribútumok aláhúzása mit jelent?
- Kulcs = minimális szuperkulcs (azonosító attribútumok)
- Filmek: elvárjuk, hogy ne legyen a megengedett előfordulásokban két különböző sor, amelyek megegyeznek cím, év attribútumokon.
- Egyszerű kulcs egy attribútumból áll, de egy kulcs nem feltétlenül áll egy attribútumból, ez az összetett kulcs. Például a **Filmek** táblában a cím és év együtt alkotják a kulcsot, nem elég a cím, ugyanis van például (King Kong, 1933), (King Kong, 1976) és (King Kong, 2005).
- A kulcsot aláhúzás jelöli: **Filmek** (cím, év, hossz, ...)

Kulcs megadása

- **PRIMARY KEY** vagy **UNIQUE**
- Nincs a relációnak két olyan sora, amely a lista minden attribútumán megegyezne.
- Kulcs esetén nincs értelme a DEFAULT értéknek.
- Kulcsok megadásának két változata van:
 - Egyszerű kulcs (egy attribútum) vagy
 - Összetett kulcs (attribútumok listája)

Egyszerű kulcs megadása

- Ha a kulcs egyetlen attribútum, akkor ez az attribútum deklarációban megadható

<attribútumnév> <típus> **PRIMARY KEY**

vagy <attribútumnév> <típus> **UNIQUE**

- Példa:

```
CREATE TABLE Sörök (  
    név          CHAR(20) UNIQUE,  
    gyártó      CHAR(20)  
);
```

Összetett kulcs megadása

- Ha a kulcs több attribútumból áll, akkor a CREATE TABLE utasításban az attribútum deklaráció után a kiegészítő részben meg lehet adni további tábla elemeket: **PRIMARY KEY (attrnév₁, ... attrnév_k)**
- Példa:

```
CREATE TABLE Felszolgál (  
    söröző      CHAR(20) ,  
    sör         VARCHAR2(20) ,  
    ár          NUMBER(10,2) ,  
    PRIMARY KEY (söröző, sör)  
);
```

PRIMARY KEY vs. UNIQUE

- Csak egyetlen **PRIMARY KEY** lehet a relációban, viszont **UNIQUE** több is lehet.
- **PRIMARY KEY** egyik attribútuma sem lehet **NULL** egyik sorban sem. Viszont **UNIQUE**-nak deklarált attribútum lehet **NULL**, vagyis a táblának lehet olyan sora, ahol a **UNIQUE** attribútum értéke **NULL** vagyis **hiányzó érték**.
- az SQL lekérdezésnél adjuk meg hogyan kell ezzel a speciális értékkel gazdálkodni, hogyan lehet **NULL**-t kifejezésekben és hogyan lehet feltételekben használni
- Következő héten visszatérünk a megszorítások és a hivatkozási épség megadására.

Mit jelent az insert utasítások után a COMMIT vagy a ROLLBACK?

- **A COMMIT utasítás** a tranzakció sikeres befejeződését eredményezi. Egy sikeresen befejeződött tranzakció a kezdete óta végrehajtott utasításainak módosításait tartósan rögzíti az adatbázisban
 - vagyis a módosítások **véglegesítődnek**.
- **A ROLLBACK utasítás** megszakítja a tranzakció végrehajtását, és annak sikertelen befejeződését eredményezi. Az így befejezett tranzakció SQL utasításai által végrehajtott módosításokat a rendszer meg nem történtekké teszi
 - vagyis az összes utasítás **visszagörgetésre kerül**, a módosítások nem jelennek meg az adatbázisban.

Kérdés / Válasz

- Köszönöm a figyelmet! Kérdés/Válasz?
- Jön a köv.előadáson: SQL SELECT utasítás

Melléklet

- Oracle gyakorlatra: rövid összefoglaló
- **Oracle gyakorlaton:** Az attribútumok típusának megadásakor (például az Oracle implementációban) **milyen standard típusok** közül választhatunk, mi a különbség a CHAR és a VARCHAR2 között?
- Létrehozzuk a táblákat és feltöltjük adatsorokkal.

Kiegészítés: Leckék Oracle gyakorlatra

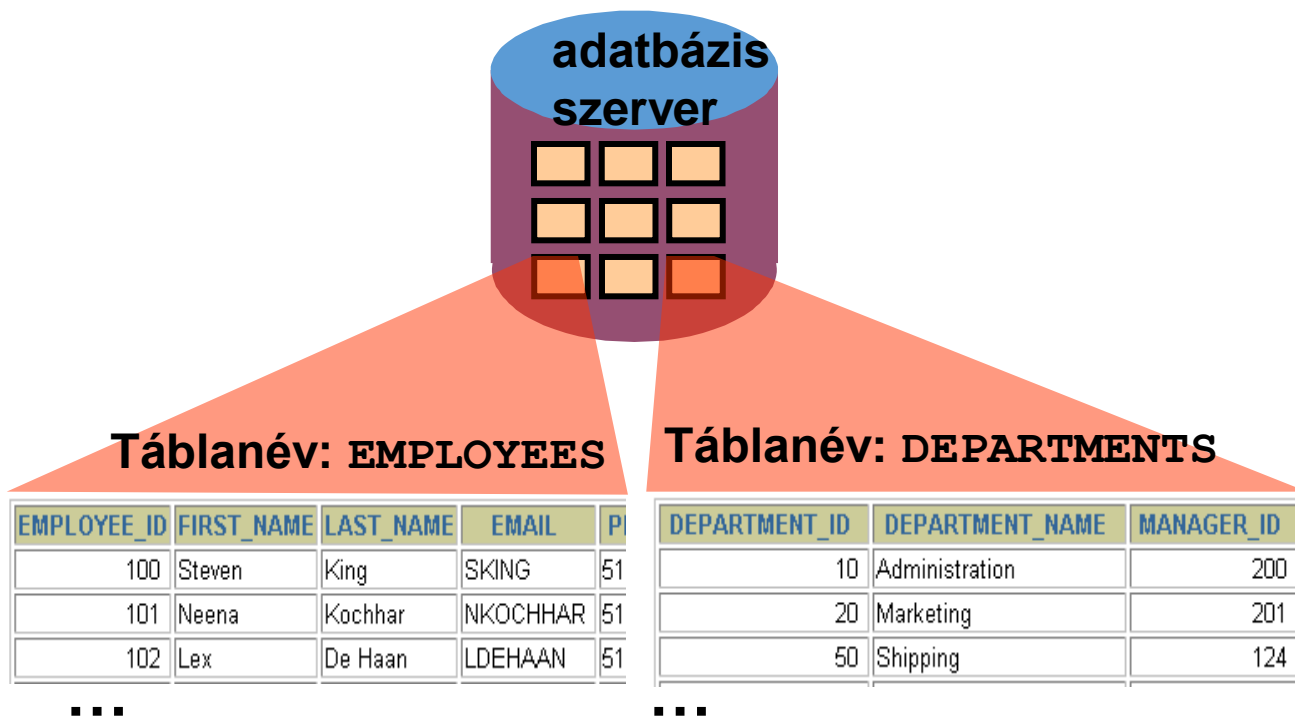
[./sql/bevezetes.pdf](#)

A relációs adatbázis fogalma

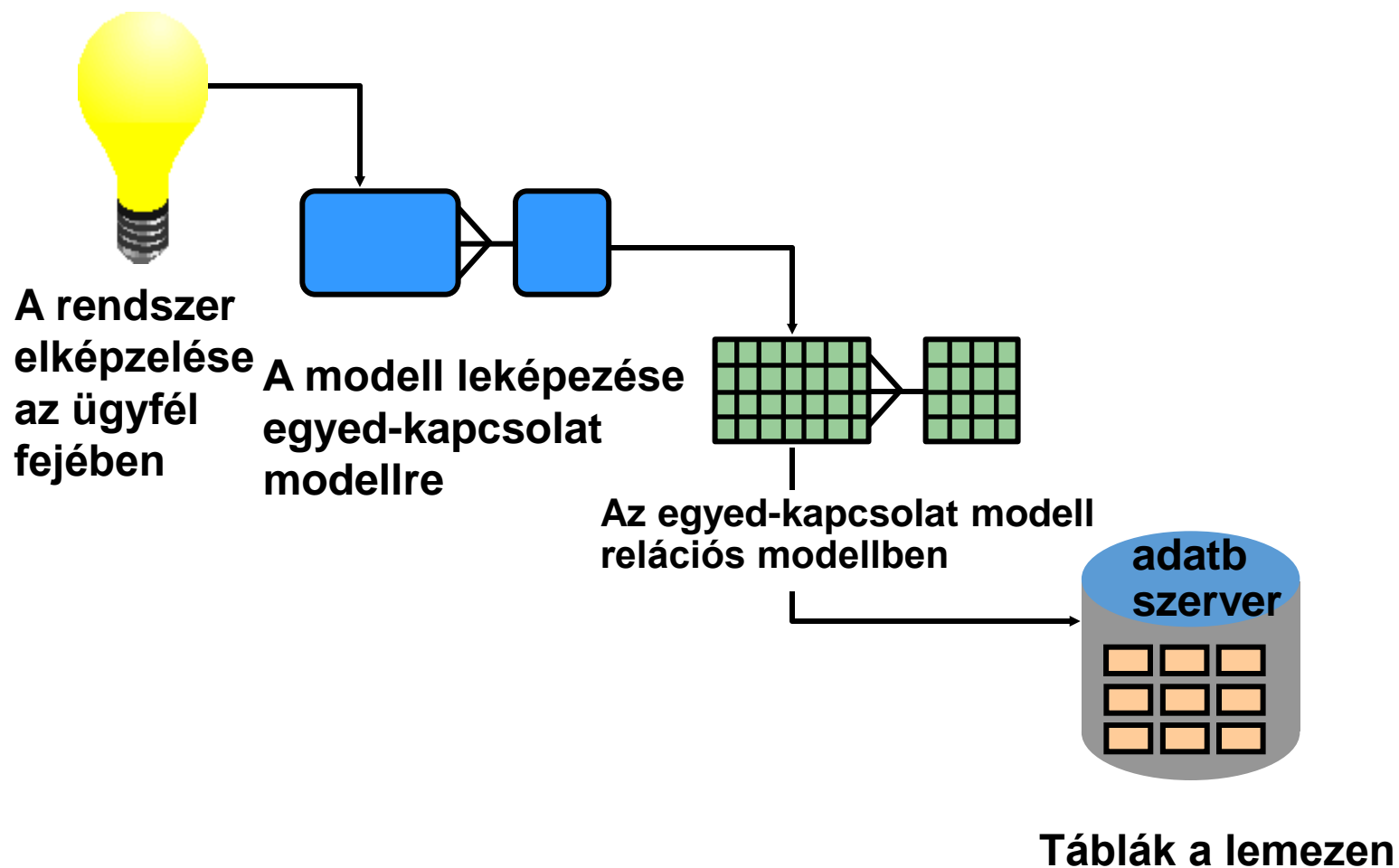
- Dr. E. F. Codd javasolta a relációs modellt 1970-ben.
- Ez az alapja a relációs adatbázis-kezelőknek (Relational database management system - RDBMS).
- **A relációs modell a következőkből áll:**
 - Objektumok vagy relációk (táblák) gyűjteményéből
 - Relációkból relációkat készítő műveletekből
 - A konzisztenciát és hibátlanságot megadó megszorításokból

A relációs adatbázis definíciója

- A relációs adatbázis kétdimenziós táblák, relációk gyűjteményét jelenti.



Adatmodellezés



Relációs adatbázisok terminológiája

2

3

4

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90
102	De Haan	Lex	17000		90
103	Hunold	Alexander	9000		60
104	Ernst	Bruce	6000		60
107	Lorentz	Diana	4200		60
124	Mourgos	Kevin	5800		50
141	Rajs	Trenna	3500		50
142	Davies	Curtis	3100		50
143	Matos	Randall	2600		50
144	Vargas	Peter	2500		50
149	Zlotkey	Eleni	10500	.2	80
174	Abel	Ellen	11000	.3	80
176	Taylor	Jonathon	8600	.2	80
178	Grant	Kimberely	7000	.15	
200	Whalen	Jennifer	4400		10
201	Hartstein	Michael	13000		20
202	Fay	Pat	6000		20
205	Higgins	Shelley	12000		110
206	Gietz	William	8300		110

1

6

5

- 1 – sor
- 2 – elsődleges kulcs
- 3 – nem kulcs oszlop
- 4 – idegen kulcs
- 5 – mező- vagy oszlopérték
- 6 - nullérték

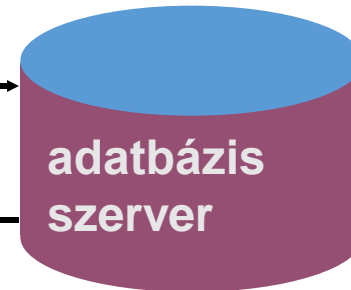
Kapcsolat az adatbáziskezelővel SQL-en keresztül

SQL utasítás kiadása.

```
SELECT department_name  
FROM departments;
```

DEPARTMENT_NAME
Administration
Marketing
Shipping
IT
Sales
Executive
Accounting
Contracting

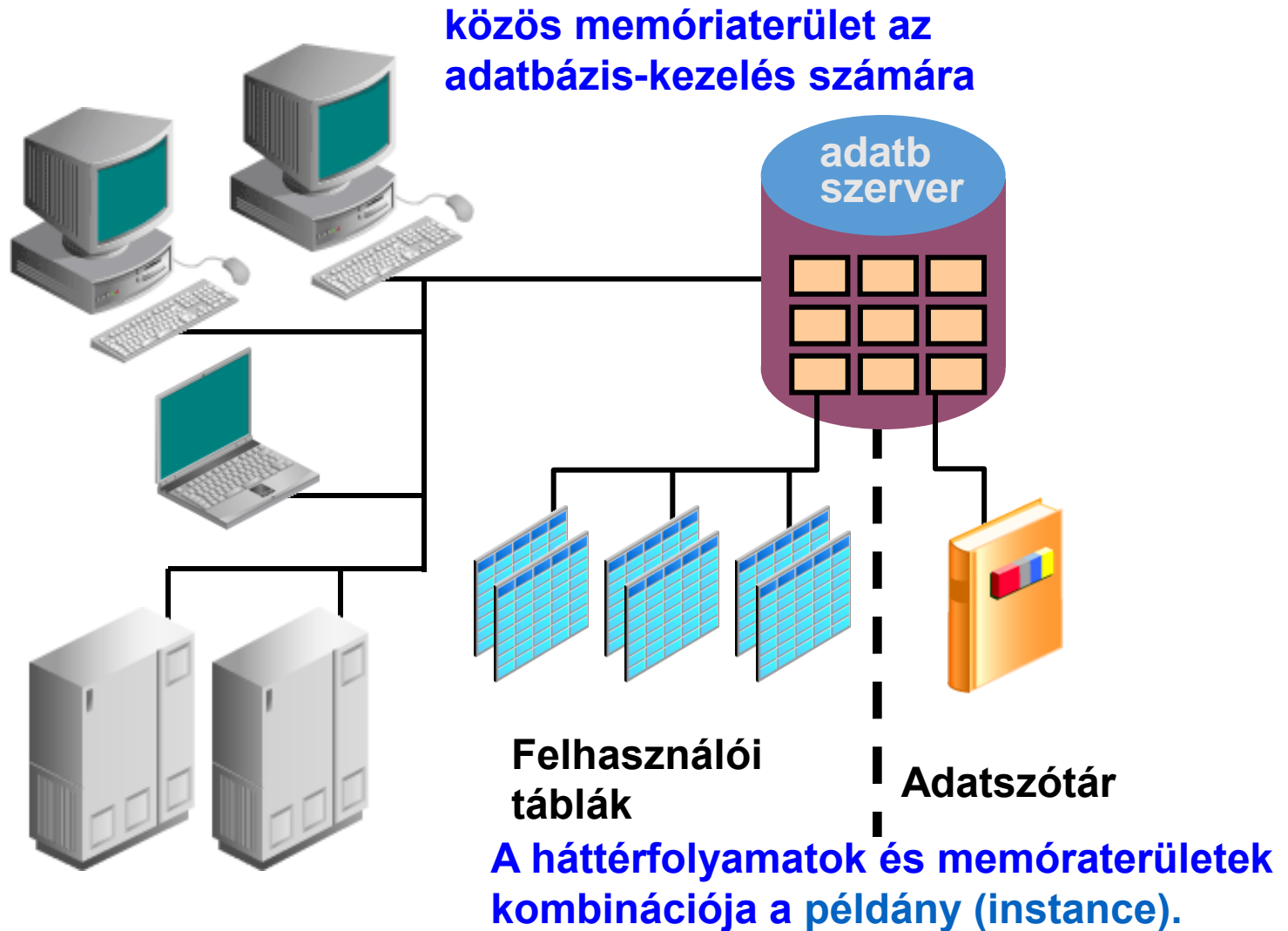
**Az utasítás átküldése az
adatbázis szerverhez.**



**Az
eredmény
előállítás,
megjeleníté
se, vagy
megfelelő
helyre
küldése**

**Az utasítás
hatékony
feldolgozása**

A relációs adatbázis-kezelő rendszer felépítése



SQL utasítások csoportosítása

- SELECT
- INSERT
- UPDATE
- DELETE
- MERGE

Adatkezelő nyelv:
Data manipulation language (DML)

- CREATE
- ALTER
- DROP
- RENAME
- TRUNCATE
- COMMENT

Adatdefiniáló nyelv:
Data definition language (DDL)

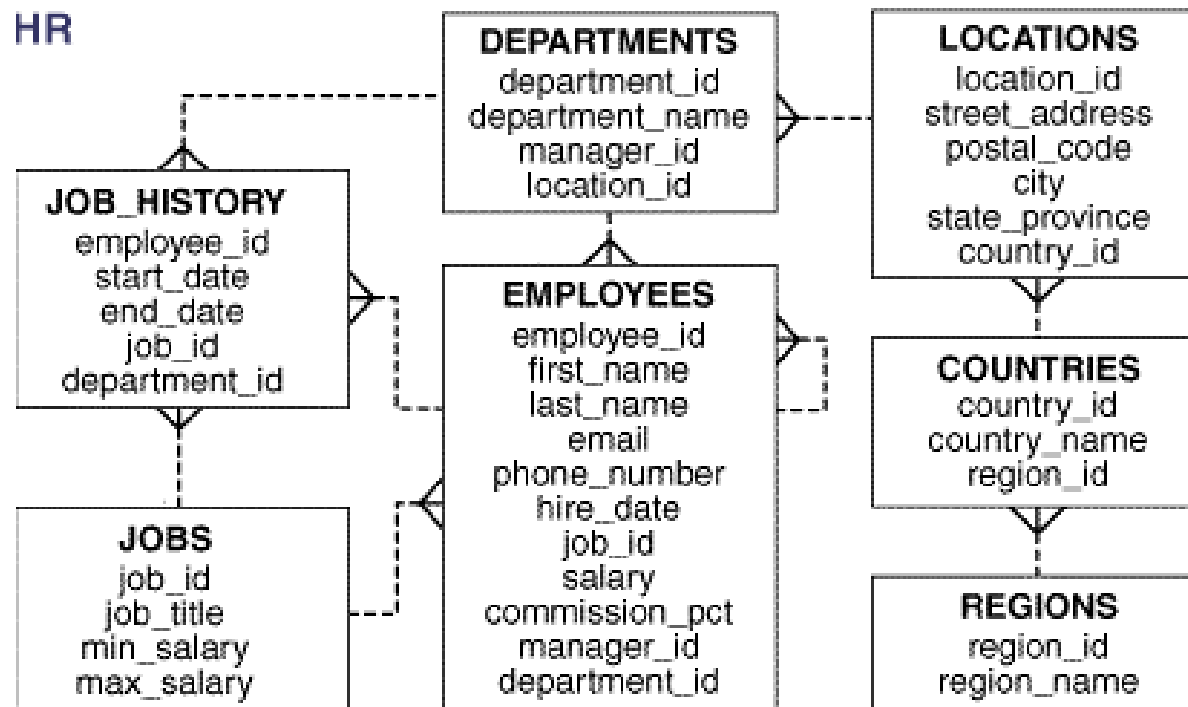
- GRANT
- REVOKE

Adatvezérlő nyelv:
Data control language (DCL)

- COMMIT
- ROLLBACK
- SAVEPOINT

Tranzakcióvezérlő nyelv:
Transaction control

A mintaadatbázis sémája



A mintaadatbázis táblái

EMPLOYEES (DOLGOZÓK)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALA
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	240
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	170
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	170
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	90
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	42
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	58
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	35
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	31

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

OS	650.121.2874	15-MAR-98	ST_CLERK	26
GAS	650.121.2004	09-JUL-98	ST_CLERK	25
KEY	650.121.4204	09-JAN-99	ST_MAN	45

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

DEPARTMENTS (OSZTÁLYOK)

JOB_GRADES (FIZETÉSI FOKOZATOK)