RELÁCIÓS ADATBÁZIS

* TÁBLÁK és azok kapcsolata
* TÁBLA: adatok halmaza
  + egyedi neve van
  + sor = rekord: minden sor különböző, számossága: változó, aktuális számát a reláció számossága
  + oszlop = attribútum/mező: a táblán belül egyedi neve van, minden mezőnek van típusa, a mező minden sorában szerepel egy elemi érték, legalább a NULL érték; oszlopok száma = reláció foka
* kulcs: feladata, hogy azonosítsa a reláció sorait; olyan attribútum(halmaz), ami erre alkalmas; sémaszintű definíció
  + elsődleges kulcs (PK): az a minimális kulcs, amit használatra választunk
  + minimális kulcs: olyan kulcs, amiből bármely attribútumot elhagyva a maradék már nem bír a kulcs tulajdonsággal, egy ***egyszerű*** kulcs mindig minimális; a ki nem választott kulcsjelölteket ***alternatív*** kulcsnak nevezzük (***összetett*** kulcs, ami több attribútumból áll)
  + idegenkulcs (FK): egy másik tábla elsődleges kulcsára hivatkozunk egy relációból

MŰVELETEK  
- DDL = DATA DEFINITION LANGUAGE:

ADATBÁZIS LÉTREHOZÁSA:

CREATE DATABASE adatbázisnév  
 DEFAULT CHARACTER SET utf8 -- karakterkódolás  
 COLLATE utf8\_hungarian\_ci; -- rendezés sorrend

TÁBLA LÉTREHOZÁSA:

CREATE TABLE táblanév(

mezőnév típus(szélesség) mezőszintű megszorítások,

….

mezőK típus(szélesség) [mezőszintű megszorítások],

[táblaszintű megszorítások]

);

MEZŐSZINTŰ MEGSZORÍTÁSOK:

* [~~NULL,~~] NOT NULL = kötelezően kitöltendő mező
* PRIMARY KEY = elsődleges kulcs, kötelezően kitöltendő
* ~~FOREIGN KEY REFERENCES táblanév (mezőnév)~~
* DEFAULT érték = ha a mező nem kap értéket, ezt állítja be
* AUTO\_INCREMENT = a rekordok automatikus sorszámozása a bevitel sorrendjében

TÁBLASZINTŰ MEGSZORÍTÁSOK:

* PRIMARY KEY (mezőnév, …, mezőnév)
* FOREIGN KEY (mezőnév) REFERENCES táblanév(mezőnév)
* UNIQUE (mezőnév, …, mezőnév) = alternatív kulcs

MEZŐTÍPUSOK:

* INT(hossz): egész
* DATE, DATETIME
* CHAR(hossz), VARCHAR(hossz), TEXT
* DOUBLE(hossz, pontosság)
* BOOLEAN

RELÁCIÓSÉMA MÓDOSÍTÁSA

új mező hozzáadása egy relációhoz

ALTER TABLE táblanév ADD mezőnév típusa [megszorítás];

mező módosítása: ha a reláció üres, esetleg szélesség bővítése, ha már van benne adat

ALTER TABLE táblanév MODIFY mezőnév típus(új\_szélesség) [esetleges megszorítás];

mező eltávolítása

ALTER TALBE táblanév DROP mezőnév;

reláció törlése

DROP TABLE táblanév;

adatbázis törlése

DROP DATABASE adatbázisnév;

DML = DATA MANIPULATION LANGUAGE

* adat bevitele valamelyik relációba:

INSERT INTO táblanév (mezők) VALUES (értékek);

INSERT INTO táblanév (mezők) VALUES (értékek), (új\_rekord), (új\_rekord2);

* adatok módosítása:

UPDATE táblanév SET mező=új érték WHERE feltétel

* adatok törlése:

DELETE FROM táblanév WHERE feltétel

DQL = DATA QUERY LANGUAGE

SELECT [DISTINCT] \*|táblanév.mezőnév, … táblanév.mezőnév [AS ’mezőnév\_megjelenik’], kifejezés, (alselect)

FROM táblanév [INNER JOIN tábla2 ON (kapcsolat)]

WHERE rekordokra vonatkozó feltételek megadása

GROUP BY táblanév.mezőnév [, táblanév.mezőnév…]

HAVING aggregátum függvényre vonatkozó feltétel

ORDER BY táblanév.mező ASC/DESC, táblanév.mező2 ASC/DESC

LIMIT n k;

Feltételek megfogalmazása

* összehasonlító operátorok:   
  kisebb (<), nagyobb vagy egyenlő (>=), egyenlő (=), nem egyenlő (<>), nagyobb (>), kisebb vagy egyenlő (<=)

szöveg: LIKE ’szöveg’; helyettesítő karakter (% = bármennyi darab, \_ = 1 db), NOT LIKE

* IN (elem1, elem2, elem3…) = a halmaz elemei közül bármelyikkel való egyezés megfelel
* BETWEEN a AND b = [a,b]
* feltételek összekapcsolása: AND, OR
* IS NULL, IS NOT NULL

Aggregátum függvények

* SUM(), AVG(), MIN(), MAX(),
* COUNT(táblanév.mezőnév) = akkor számlál, ha a mező értéke nem NULL, COUNT(\*) = sorok számát számlálja meg, COUNT(DISTINCT táblanév.mezőnév) = a különböző elemek számának meghatározása

LAZA ILLESZTÉS

táblák összekapcsolása

1. LEFT JOIN a kapcsolat bal oldalán álló reláció minden sora megjelenik az eredményben, a jobb oldali tábla esetén a mezők NULL értékkel kapcsolódnak a rekordhoz, ha nincs benne megfelelő sor
2. RIGHT JOIN szimmetrikus left join