

Érettségi követelmények

2.2.5. Adatbázis-kezelés

Készítette: Kuttor Patrik

Adatdefiníciós parancsok

Tudjon új adatbázist létrehozni a saját számítógépén futtatott MySQL adatbázis-kezelő alkalmazás segítségével.

Tudjon táblákat létrehozni (CREATE TABLE) és feltölteni (INSERT INTO) SQL parancsfájl (script) segítségével MySQL adatbázis-kezelőn.

Tudja a legfontosabb adattípusokat: szöveges, logikai, dátum, idő, numerikus (egészek és valós típusok).

CREATE TABLE : Tábla létrehozása. A tábla nevének megadásán túl fel kell sorolni a tábla mezőit (attribútumok) és típusát.

pl.: CREATE TABLE gyumolcsok(

id INT(11),

megnevezes VARCHAR(50),

leiras TEXT

);

INSERT INTO : Új rekord(ok) beszúrása az adattáblába.

pl.: INSERT INTO gyumolcsok(megnevezes, leiras) VALUES('alma', 'Rövid leírás');

Lehetőség van egyszerre több rekord beszúrására is

pl.: INSERT INTO gyumolcsok(megnevezes, leiras) VALUES
('alma', 'Rövid leírás'),
('barack', 'Rövid leírás'),
('dinnye', 'Rövid leírás');

A tábla neve után célszerű felsorolni a mezők neveit, amiket szeretnénk feltölteni adatokkal. Ez opcionális lehetőség, ha elhagyjuk, a mezők sorrendben kerülnek feltöltésre adatokkal.

Lekérdező parancs

Ismerje és tudja alkalmazni a SELECT parancsot. Lekérdezéseknél tudjon számított mezőket létrehozni, táblákat összekapcsolni szoros illesztéssel.

SELECT: A lekérdezéseket mindig a SELECT kulcsszóval kezdjük. A SELECT után kell megadni azokat a mezőket, amiket eredményként, visszakapunk.

Minden nev és leiras mező értékének lekérdezése a gyumolcsok táblából

```
SELECT nev, leiras FROM gyumolcsok
```

Minden mező lekérdezése a gyumolcsok táblából

```
SELECT * FROM gyumolcsok
```

Záradékok és módosítók

Ismerje és tudja alkalmazni egyszerű parancsok készítéséhez a legfontosabb záradékokat, módosítókat:

AS, DISTINCT,

FROM: A lekérdezések része a forrás adattábla (k) meghatározása, melyet használtunk az előző példákban is.

GROUP BY: A megadott feltétel (mező értéke) szerint azonos rekordok csoportosítása, melyet jellemzően aggregáló függvényekkel kiegészítve használunk.

Ebben a lekérdezésben minden rekord külön sorként kerül listázásra és mindenhol 1 lesz a második mező értéke.

```
SELECET gyumolcsnev, COUNT(*) FROM kiszallitasok;
```

Ebben a lekérdezésben az azonos nevű gyümölcsök csoportosítva kerülnek feldolgozásra és a COUNT függvény a csoportosított rekordok számát fogja visszaadni.

```
SELECT gyumolcsnev, COUNT(*) FROM kiszallitasok GROUP BY gyumolcsnev;
```

HAVING: A HAVING segítségével a már csoportosított (GROUP BY) eredményhalmazra tudunk extra feltételeket megadni.

```
SELECT gyumolcsnev, COUNT(*) FROM kiszallitasok GROUP BY gyumolcsnev  
HAVING COUNT(*) > 5;
```

LIMIT: A visszatérési rekordok számának limitálása.

Az első 5 sort adja vissza eredményként.

```
SELECT * FROM gyumolcsok LIMIT 5
```

ORDER BY: Rendezési feltétel meghatározása. Megadhatjuk melyik mezőre történjen a rendezés és milyen irányban.

Növekvő sorrendben történő rendezés

```
SELECT * FROM gyumolcsok ORDER BY nev ASC
```

Csökkenő sorrendben történő rendezés

```
SELECT * FROM gyumolcsok ORDER BY nev DESC
```

WHERE: A lekérdezés eredményének szűkítésére szolgál. A mezőnevekre megadott logikai műveletek segítségével szűkíti az eredmény halmazt.

Lekérdezzük, az összes gyümölcsöt ahol ki van töltve a leírás.

```
SELECT * FROM gyumolcsok WHERE leiras <> "";
```

Kifejezések, operátorok, függvények

Tudjon záradékokban és a számított mezőkben kifejezéseket összeállítani.

Ismerje és tudja alkalmazni a következő operátorokat és függvényeket: aritmetikai operátorok, relációs operátorok, AND, AVG(), BETWEEN, COUNT(), IN, IS NULL, IS NOT NULL, LIKE, MAX(), MIN(), NOT, OR, SUM()