**SQL**

SQL: Structured Query Language – Strukturált Lekérdező Nyelv

Fejlesztése: 1970-től Edgar Codd, IBM-nél (SEQUEL)

1986-tól megjelennek a szabványok.

Felépítése: angol mondatok, amit az értelmező optimalizál és feldolgoz.

Az SQL az adatbázis-kezelő rendszer integrált része, nem egy önálló szoftver. Számos programfejlesztő rendszerbe is beépítették. Az SQL egy hatékony eszköz, mely a felhasználókat, és a programokat összeköti a relációs adatbázisban tárolt adatokkal.

Szintaktika:

* Az SQL parancsokat ; (pontosvessző) zárja.
* A kis és nagybetűket nem különbözteti meg.
* A parancsok szóközökkel, tabulátorokkal tagolhatók és több sorba írhatók.
* Szöveg konstansok szimpla aposztrófok között vannak, pl. 'alma'
* Azonosítók:
* Maximum 128 karakter
* Tartalmazhatja az angol ábécé betűit, számjegyeket, \_, #, $ karaktereket.
* Betűvel kell, hogy kezdődjön.

SQL parancsok osztályozása:

**Adatdefiníciós parancsok**: az adatbázist, a táblákat, mezők létrehozása, tulajdonságok, kapcsolatot definiálása.

**Adatmanipulációs parancsok**: új rekordok létrehozása, törlése, mezők módosítása.

**Lekérdező parancsok**: választó lekérdezés létrehozása (SELECT parancs).

**Adatvezérlő parancsok**: file-ok megnyitását, lezárását, mutatók mozgatását, parancsok végrehajtásának jóváhagyását, stb. tudjuk vezérelni.

Megjegyzések az sql parancsállományokban:

* Egy soros megjegyzés: két – jel és egy szóköz után következik a megjegyzés szövege  
  -- megjegyzés
* Több soros megjegyzés: /\* és \*/ közötti szöveg, egy soron belül is alkalmazható.

**MySQL adattípusok**

**Szám adattípusok:**

tinyint 1 bájt 0-255 vagy -128-+128

smallint 2 bájt 0-65535 vagy -32768-+32768

int (integer) 4 bájt 0-~4 milliárd vagy ±2 milliárd között

**Lebegőpontos típusok:**

float, double, decimal

Dátum és idő típus:

date dátum ÉÉÉÉ-HH-NN formátumban

time idő ÓÓ:PP:MM formátumban

datetime dátumot és időt együtt tároló típus; tárolási formátum: ÉÉÉÉ-HH-NN ÓÓ:PP:MM

**Szöveges típus:**

char(hossz) fix hosszúságú szöveg. hossz: 1-től 255-ig

varchar(hossz) változó hosszúságú karakterlánc, hossz: 1-től 255-ig

blob vagy text max 65535 karakter hosszú szöveg

**Logikai adattípus:**

használható a bool típus és a true, false értékek, de a megvalósítása tinyint típussal történik, ahol true = 1, false = 0

**SQL utasítások**

Adatbázis lekérdezése:

SHOW DATABASES

Adatbázis létrehozása UTF8 kódolással, magyar ábécé szerinti rendezéssel:

CREATE DATABASE *adatbázisnév*

DEFAULT CHARACTER SET utf8

COLLATE utf8\_hungarian\_ci;

Adatbázis létrehozása:

CREATE DATABASE *adatbázisnév*;

Belépés egy adatbázisba:

USE *adatbázisnév*;

Adatbázis törlése:

DROP DATABASE *adatbázisnév*;

Tábla létrehozása:

CREATE TABLE táblanév (

*mezőnév\_1* adattípus\_1,

…

*mezőnév\_n* adattípus\_n );

A definícióban, a mezőnév után, egyéb tulajdonságok is megadhatók:

* mezőméret (varchar típusnál)
* megszorítások:
  + NULL 🡪 lehet üres a mező
  + NOT NULL 🡪 nem lehet üres a mező
  + PRIMARY KEY 🡪 elsődleges kulcs
  + AUTO\_INCREMENT 🡪 automatikus növelés, sorszámozás
* alapértelmezett érték megadása: DEFAULT ’érték’
* indexelés beállítása: INDEX *indexnév* (*mezőnév*)

Az adatbázis meglévő tábláinak lekérdezése:

SHOW TABLES;

Tábla törlése:

DROP TABLE *táblanév*;

Tábla átnevezése:

RENAME TABLE *régi\_név* TO *új\_név;*

Mezőnevek és mezőtulajdonságok lekérdetése:

SHOW COLUMNS FROM *táblanév*;

Helyette használható a DESCRIBE *táblanév*; utasítás is.

Tábla módosítása, mező hozzáadása:

ALTER TABLE *táblanév*

ADD (mezőnév adattípus [,mezőnév adattípus…]);

Tábla módosítása, mező törlése:

ALTER TABLE *táblanév*

DROP *mezőnév*;

Tábla módosítása, index hozzáadása:

ALTER TABLE *táblanév*

ADD INDEX *indexnév* (*mezőnév*);

Tábla módosítása, index törlése:

ALTER TABLE *táblanév*

DROP INDEX *indexnév*;

Tábla módosítása, elsődleges kulcs hozzáadása:

ALTER TABLE *táblanév*

ADD PRIMARY KEY (*mezőnév*);

Tábla módosítása, elsődleges kulcs törlése:

ALTER TABLE *táblanév*

DROP PRIMARY KEY (*mezőnév*);

Tábla módosítása, meglévő mező módosítása:

ALTER TABLE *táblanév*

MODIFY (mezőnév adattípus [,mezőnév adattípus…]);

Új adatsor (rekord) bevitele:

INSERT INTO *táblanév*

[(mezőnév-lista)]

VALUES (értéklista);

A mezőnevek és az értékek listájának elemeit vesszővel kell elválasztani. A mezőnevek listája csak akkor hagyható el, ha minden mező értéket kap.

Rekordok törlése:

DELETE FROM *táblanév*

[WHERE feltétel];

A tábla rekordjainak módosítása:

UPDATE *táblanév*

SET oszlopnév=kifejezés [,oszlopnév=kifejezés,…]

[WHERE logikai kifejezés];

Lekérdezés készítése:

**SELECT** mező1 AS név1, mező2 AS név2 …

**FROM** táblanév

**WHERE** feltételek

**GROUP BY** csoportképző mező(k)

**HAVING** feltétel az összesítő függvényekre és csoportképző mezőkre

**ORDER BY** sorrendképző mező ASC|DESC

**LIMIT** elemek száma;

A SELECT tartalmazhat mezőket, számításokat, összesítő függvényeket.

A mezőnév előtti **DISTINCT** kifejezés hatása, hogy az azonos értékek csak egyszer jelennek meg.

A LIMIT a megjelenő elemek számát adja meg vagy –tol –ig sorszámot, ekkor az első elem sorszáma a 0.

Két tábla kapcsolata SELECT utasításban:

**FROM tábla1 INNER JOIN tábla2 ON tábla1.mezőnév = tábla2.mezőnév**

Példák:

SELECT 5+3\*4

FROM iskola;

SELECT \*

FROM iskola;

SELECT nev, cim

FROM iskola

WHERE varos<>’Ajka’

ORDER BY nev;

SELECT SUM(ar)

FROM eladas

Operátorok

|  |  |  |  |
| --- | --- | --- | --- |
| = | egyenlő | BETWEEN x AND y | adott értékek közé esik |
| <> | nem egyenlő | IN (a, b, c, ...) | az értékek között található |
| > | nagyobb | LIKE minta | hasonlít a mintára |
| >= | nagyobb egyenlő | NOT | Logikai tagadás |
| < | kisebb | AND | Logikai és |
| <= | kisebb egyenlő | OR | Logikai vagy |

LIKE esetén MySQL-ben a % több karaktert helyettesít, a \_ pedig egyet

Precedencia:

1. =, <>, >, >=, <, <=

2. NOT

3. AND

4. OR

Üres mező feltétele: IS NULL, tagadása: IS NOT NULL

Összesítő függvények:

AVG(mezőnév) a mező értékeink átlaga

SUM(mezőnév) a mező értékeink összege

MAX(mezőnév) a mező legkisebb eleme

MIN(mezőnév) a mező legnagyobb eleme

COUNT(mezőnév) a mezőben található elemek száma

Matematikai függvények:

MOD(osztandó, osztó) maradékképzés

ABS(szám) abszolút érték

POW(alap, kitevő) hatványozás

SQRT(szám) gyökvonás

ROUND(szám) kerekítés egészre

ROUND(szám, számjegy) kerekítés helyiérték szerint (0 - egész, 1 – tized, 2 - század … -1 – tízes …)

TRUNCATE (szám) levágás egész helyiértéknél

Dátum és idő függvények:

NOW() Aktuális dátum és idő

CURDATE() Aktuális dátum

CURTIME() Aktuális idő

YEAR(dátum/idő) A dátum adatból visszaadja az évet

MONTH(dátum/idő) A dátum adatból visszaadja a hónapot

DAY(dátum/idő) A dátum adatból visszaadja a napot

HOUR(dátum/idő) Az idő adatból visszaadja az órát

MINUTE(dátum/idő) Az idő adatból visszaadja a percet

SECOND(dátum/idő) Az idő adatból visszaadja a másodpercet

DAYNAME(dátum/idő) A nap szöveges nevét adja meg