



Blahota István

SQLite alapok

oktatási segédanyaghoz tartozó

gyakorlatok

Társadalmi Megújulás Operatív Program
Határon átnyúló együttműködés a szakképzés és a felnőttképzés
területén c. pályázati felhívás keretében megvalósított

***Mobil alkalmazásfejlesztés az informatikai tudás innovatív
alkalmazásával*** című, **TÁMOP-2.2.4-11/1-2012-0055** kódszámú
projekt keretében valósult meg.

2013.





Tartalomjegyzék

1. Kidolgozott SQLite példák megoldásokkal.....	3
2. SQLite és PHP.....	17
3. SQLite és Android.....	23

1. Kidolgozott SQLite példák megoldásokkal

1. Az alábbiakban néhány tipikus SQLite műveletet fogunk bemutatni.

Ehhez először létrehozunk, majd feltöltünk néhány adattal egy mintaadatbázist. Ezek után módosításokat, lekérdezéseket hajtunk végre rajta.

Az adatbázis elkészítése, szerkezetének módosítása

1. Feladat

Hozzuk létre a „minta” nevű adatbázist.

Megoldás:

Parancssorból adjuk ki az

```
sqlite3 minta.sqlite
parancsot. (A kiterjesztés opcionális.)
```

2. Feladat

Hozzuk létre a „konyvek” nevű táblát. A mezőnevek és típusaik a következők legyenek:

„id”: egész, elsődleges kulcs, számláló típus,
 „szerzok”: szöveg típus,
 „cim”: szöveg típus, nem lehet üres string,
 „tipus”: egész típus,
 „ev”: (a kiadás éve) egész típus.

Megoldás:

```
CREATE TABLE konyvek (
id INTEGER PRIMARY KEY AUTOINCREMENT,
szerzok TEXT,
cim TEXT NOT NULL,
tipus INTEGER,
ev INTEGER);
```

3. Feladat

Hozzuk létre a „tipusok” nevű táblát. A mezőnevek és típusaik a következők legyenek:

„tipus”: egész típus, elsődleges kulcs,
 „tipusnev”: szöveg típus, nem lehet üres string, egyedi név.

Megoldás:

```
CREATE TABLE tipusok (  
tipus INTEGER PRIMARY KEY,  
tipusnev TEXT NOT NULL UNIQUE);
```

4. Feladat

Indexeljük a „konyvek” táblát a „cim” mező alapján, ABC szerinti növekvő sorrendben. Az index neve legyen „cimlista”.

Megoldás:

```
CREATE INDEX cimlista  
ON konyvek (  
cim ASC);
```

5. Feladat

Módosítsuk a „konyvek” táblát, szűrjük be az „alcim” nevű szöveg típusú mezőt.

Megoldás:

```
ALTER TABLE konyvek  
ADD COLUMN alcim TEXT;
```

Az adatbázis feltöltése

6. Feladat

Töltsük fel a „tipusok” táblát adatokkal. A típusnevek legyenek: szakkönyv, szépirodalom, ifjúsági, egyéb.

Megoldás:

```
INSERT INTO tipusok (tipus, tipusnev)  
VALUES (1, „szakkönyv”);  
INSERT INTO tipusok (tipus, tipusnev)  
VALUES (2, „szépirodalom”);  
INSERT INTO tipusok (tipus, tipusnev)  
VALUES (3, „ifjúsági”);
```

```
INSERT INTO tipusok (tipus, tipusnev)
VALUES (4, "egyéb");
```

7. Feladat

Töltsük fel a „konyvek” táblát adatokkal. (Elvileg mindegy, milyen adatokkal töltjük fel a táblát, viszont a továbbiakban tárgyalt műveletek eredményei a megoldásban látható adatokból származnak.)

Megoldás:

```
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Szép Jenő","Analízis",1,1972);
INSERT INTO konyvek (szerzok,cim,alcim,tipus,ev)
VALUES ("Dr. Csernyák László","Analízis","Matematika
üzemgazdászoknak",1,1988);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Walter Rudin","A matematikai analízis alapjai",1,1978);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Szidarovszky Ferenc","Bevezetés a numerikus
módszerekbe",1,1974);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Pólya György","A gondolkodás iskolája",1,2000);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Pólya György","A problémamegoldás iskolája",1,1985);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Slawomir Mrozek","Zuhanás közben",2,1983);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Ray Bradbury","Marsbéli krónikák",2,1982);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Gerald Durrel","A halak jelleme",3,1989);
INSERT INTO konyvek (szerzok,cim,alcim,tipus,ev)
VALUES ("Timkóné Sebestyén Izabella","Hegyedül a
kisegér..." ,"Mondókák, énekek, tréfás dalok óvodásoknak",3,2002);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("Csörsz István","Elhagyott a közérzetem",4,1986);
INSERT INTO konyvek (szerzok,cim,tipus,ev)
VALUES ("William J. Broad","Teller háborúja",4,1996);
```

Az adatbázis lekérdezései

8. Feladat

Írassuk ki a teljes „konyvek” táblázati tartalmat.

Megoldás:

```
SELECT * FROM konyvek;
```

9. Feladat

Írassuk ki a „konyvek” tábla „szerzok” és „cim” oszlopainak tartalmát.

Megoldás:

```
SELECT szerzok, cim FROM konyvek;
```

10. Feladat

Írassuk ki a „konyvek” tábla „cim” oszlopának tartalmát az elemek esetleges ismétlése nélkül.

Megoldás:

```
SELECT DISTINCT cim FROM konyvek;
```

11. Feladat

Írassuk ki a „konyvek” tábla „szerzők” és „cim” oszlopának tartalmát az elemek címük szerinti növekvő ABC sorrendjében.

Megoldás:

```
SELECT szerzok, cim FROM konyvek
ORDER BY cim ASC;
```

12. Feladat

Írassuk ki a „konyvek” tábla „szerzők” és „cim” oszlopának tartalmát típusuk azonosítójának csökkenő sorrendjében, azon belül pedig címük szerinti növekvő ABC sorrendben.

Megoldás:

```
SELECT szerzok, cim FROM konyvek
ORDER BY tipus DESC, cim ASC;
```

13. Feladat

Írassuk ki a „konyvek” tábla azon rekordjainak adatait melyek típusának azonosítója 1.

Megoldás:

```
SELECT * FROM konyvek
```

```
WHERE tipus=1;
```

14. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvcímeket, melyekhez alcím tartozik.

Megoldás:

```
SELECT cim FROM konyvek
WHERE alcim NOT NULL;
```

15. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvcímeket, melyekhez nem tartozik alcím.

Megoldás:

```
SELECT cim FROM konyvek
WHERE alcim IS NULL;
```

16. Feladat

Írassuk ki a „konyvek” táblában szereplő azon szerzőket, akiknek jelent meg 2000 előtt könyve (és az benne van az adatbázisunkban). Kerüljük el a duplikációkat!

Megoldás:

```
SELECT DISTINCT szerzok FROM konyvek
WHERE ev<2000;
```

17. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a hozzájuk tartozó szerzőket, amelyek 1980 és 1990 között jelentek meg.

Megoldás:

```
SELECT cim, szerzok FROM konyvek
WHERE ev BETWEEN 1980 AND 1990;
```

18. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét, melyek szerzője Pólya György.

Megoldás:

```
SELECT cim FROM konyvek
WHERE szerzok="Pólya György";
```

19. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét, melyek szerzőjének neve „S” betűvel kezdődik.

Megoldás:

```
SELECT cim, szerzok FROM konyvek
WHERE szerzok LIKE "S%";
```

20. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyveket és a hozzájuk tartozó szerzőket, amelyek 1972-ben, 1983-ban vagy 1986-ban jelentek meg.

Megoldás:

```
SELECT cim, szerzok FROM konyvek
WHERE ev IN (1980,1983,1986);
```

21. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyveket, a hozzájuk tartozó szerzőket, valamint megjelenésük évét, amelyek 1995-ben vagy korábban jelentek meg, és a szerzőjének névében benne van az „er” karaktersorozat.

Megoldás:

```
SELECT cim, szerzok, ev FROM konyvek
WHERE ev<=1995 AND szerzok LIKE "%er%";
```

22. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a megjelenésük évét, amelyek a múlt évezredben jelentek meg és vagy szerzőjüknek névében benne van a „György” karaktersorozat, vagy van alcímük.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE ev<=2000 AND
(szerzok LIKE "%György%" OR alcim NOT NULL);
```


23. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a megjelenésük évét, amelyek vagy a múlt évezredben jelentek meg és vagy szerzőjüknek névében benne van a „György” karaktersorozat , vagy van alcímük.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE (ev<=2000 AND szerzok LIKE "%György%") OR
alcim NOT NULL;
```

24. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a megjelenésük évét, amelyeknek van alcímük és vagy a múlt évezredben jelentek meg, vagy szerzőjüknek névében benne van a „György” karaktersorozat.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE alcim NOT NULL AND
(ev<=2000 OR szerzok LIKE "%György%");
```

25. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a megjelenésük évét, amelyeknek van alcímük és nem a múlt évezredben jelentek meg és szerzőjüknek névében nincs benne a „György” karaktersorozat.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE alcim NOT NULL AND
NOT (ev<=2000 OR szerzok LIKE "%György%");
```

vagy:

```
SELECT cim, ev FROM konyvek
WHERE alcim NOT NULL AND
NOT (ev<=2000) AND
NOT(szerzok LIKE "%György%");
```

26. Feladat

Írassuk ki a „konyvek” táblában szereplő azon nem szakkönyvek (típusuk száma nem 1) címét és a megjelenésük évét, amelyek vagy a múlt évezredben jelentek meg és vagy szerzőjüknek névében benne van a „György” karaktersorozat, vagy van alcímük.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE tipus IS NOT 1 AND
((ev<=2000 AND szerzok LIKE "%György%") OR alcim NOT NULL);
```

27. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a megjelenésük évét, amelyek szerzőjének névében benne van a „György” karaktersorozat és a múlt évezredben jelentek meg vagy van alcímük és nem szakkönyvek (típusuk száma nem 1).

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE (szerzok LIKE "%György%" AND ev<=2000) OR
(alcim NOT NULL AND tipus IS NOT NULL);
```

28. Feladat

Írassuk ki a „konyvek” táblában szereplő könyvek címét és a megjelenésük évét a megjelenés évének sorrendjében.

Megoldás:

```
SELECT cim, ev FROM konyvek
ORDER BY ev;
```

29. Feladat

Írassuk ki a „konyvek” táblában szereplő szakkönyvek (típusuk száma 1) címét és a megjelenésük évét a címek fordított ABC sorrendjében.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE tipus=1
ORDER BY cim DESC;
```

30. Feladat

Írassuk ki a „konyvek” táblában szereplő szakkönyvek (típusuk száma 1) címét és a megjelenésük évét, de minden évből csak egyet.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE tipus=1
GROUP BY ev;
```

31. Feladat

Írassuk ki a „konyvek” táblában szereplő szakkönyvek (típusuk „szakkönyv”) címét és a megjelenésük évét, de minden évből csak egyet.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE tipus IN (SELECT tipus FROM tipusok WHERE
tipusnev="szakkönyv")
GROUP BY ev;
```

32. Feladat

Írassuk ki a „konyvek” táblában szereplő könyvek címét és a megjelenésük évét a szerzők ABC sorrendjében, de minden évből csak egyet.

Megoldás:

```
SELECT cim, ev FROM konyvek
GROUP BY ev
ORDER BY szerzok ASC;
```

33. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a megjelenésük évét, amelyeknek van alcímük és a múlt évezredben jelentek meg, vagy szerzőjüknek névében benne van a „György” karaktersorozat, viszont minden szerzótől legfeljebb egyet. A kiíratás sorrendje a megjelenésük évének sorrendje legyen.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE (alcim NOT NULL AND ev<=2000) OR
szerzok LIKE "%György%"
```

```
GROUP BY szerzok
ORDER BY ev ASC;
```

34. Feladat

Írassuk ki a „konyvek” táblában szereplő azon könyvek címét és a megjelenésük évét, amelyeknek van alcímük és a múlt évezredben jelentek meg, vagy szerzőjüknek névében benne van a „György” karaktersorozat, viszont minden szerzótől legfeljebb egyet.

Megoldás:

```
SELECT cim, ev FROM konyvek
WHERE (alcim NOT NULL AND ev<=2000) OR
szerzok LIKE "%György%"
GROUP BY szerzok;
```

35. Feladat

Írassuk ki a „konyvek” tábla „cim” mezőjének elemeit, mellettük pedig azt, hogy azok hány karakterből állnak.

Megoldás:

```
SELECT cim, LENGTH(cim) FROM konyvek;
```

36. Feladat

Írassuk ki a „konyvek” tábla „cim” mezőjének elemeit nagybetűssé alakítva.

Megoldás:

```
SELECT UPPER(cim) FROM konyvek;
```

37. Feladat

Írassuk ki a „konyvek” tábla „cim” mezőjének elemeit kisbetűssé alakítva.

Megoldás:

```
SELECT LOWER(cim) FROM konyvek;
```

38. Feladat

Írassuk ki a „konyvek” tábla „cim” mezőjének elemeit, úgy, hogy az „e” betűket „qqq” jelsorozatra cseréltük.

Megoldás:

```
SELECT REPLACE(cim,"e","qqq") FROM konyvek;
```

39. Feladat

Írassuk ki a „konyvek” tábla „cim” mezőjének elemeit, mellettük pedig azt, hogy azok hány „e” betűt tartalmaznak.

Megoldás:

```
SELECT cim, LENGTH(cim)-LENGTH(REPLACE(cim,"e","")) FROM konyvek;
```

40. Feladat

Írassuk ki a „konyvek” tábla „cim” mezőjének elemeit, mellettük pedig azt, hogy az „e” betű hányadik karakterként fordul elő a címben.

Megoldás:

```
SELECT cim, INSTR(cim,"e") FROM konyvek;
```

41. Feladat

Írassuk ki a „konyvek” tábla „cim” mezője elemeinek számát.

Megoldás:

```
SELECT COUNT(cim) FROM konyvek;
```

42. Feladat

Írassuk ki a „konyvek” tábla „ev” mezője elemeinek minimumát, maximumát, átlagát és összegét.

Megoldás:

```
SELECT MIN(ev), MAX(ev), AVG(ev), SUM(ev) FROM konyvek;
```

43. Feladat

Írassuk ki a „konyvek” tábla „ev” mezője elemeinek átlaguktól való eltéréseinek átlagát.

Megoldás:

```
SELECT SUM(ABS(ev-(SELECT AVG(ev) FROM konyvek))) FROM konyvek;
```

44. Feladat

Írassuk ki az aktuális dátumot.

Megoldás:

```
SELECT DATE();
```

vagy:

```
SELECT DATE("now");
```

45. Feladat

Írassuk ki az aktuális időt.

Megoldás:

```
SELECT TIME();
```

vagy:

```
SELECT TIME("now");
```

46. Feladat

Írassuk ki az aktuális dátumot és időt.

Megoldás:

```
SELECT DATETIME();
```

vagy:

```
SELECT DATETIME("now");
```

47. Feladat

Írassuk ki a három hónap múlva aktuális dátumot.

Megoldás:

```
SELECT DATE("now", "3 months");
```

48. Feladat

Írassuk ki a tíz nappal ezelőtti aktuális dátumot.

Megoldás:

```
SELECT DATE("now", "-10 days");
```

49. Feladat

Írassuk ki a hányadika lesz az aktuális hét negyedik napján.

Megoldás:

```
SELECT DATE("now", "weekday 4");
```

50. Feladat

Írassuk ki a mai nap év az aktuális év kezdetének idejét.

Megoldás:

```
SELECT DATETIME("now", "start of day"), DATETIME("now", "start of year");
```

51. Feladat

Írassuk ki a mai nap dátumát formázva.

Egy lehetséges megoldás:

```
SELECT STRFTIME("%Y. év, %m. hónap, %d. nap van ma");
```

52. Feladat

Írassuk ki hány nap múlva lesz vége a 2013-as évnek.

Megoldás:

```
SELECT STRFTIME("%j", "2013-12-31")-STRFTIME("%j", "now");
```

53. Feladat

Írassuk ki hány változás történt az adatbázisban (a DELETE vagy az INSERT vagy az UPDATE hatására), mióta megnyitottuk.

Megoldás:

```
SELECT TOTAL_CHANGES();
```

54. Feladat

Írassuk ki az SQLite verziószámát.

Megoldás:

```
SELECT SQLITE_VERSION();
```

55. Feladat

Írassuk ki a „konyvek” tábla címeiből hármat véletlenszerűen (könyvajánló).

Megoldás:

```
SELECT cim FROM konyvek  
ORDER BY RANDOM() LIMIT 3;
```


2.SQLite és PHP

Ismerkedjünk meg az SQLite ezen klasszikus használatával. Tegyük fel, hogy működik gépünkön egy Apache webservert PHP-val, illetve telepítettük az SQLite-ot is. Az SQLite és PHP együttes működéséhez szükséges még az SQLite PHP modul telepítése is. Ez Ubuntu alatt a `php5-sqlite` csomag telepítését jelenti.

Mindez együtt egy lépésben is megtehető. Adjuk ki az alábbi parancsot egy terminálból:

```
sudo apt-get install apache2 libapache2-mod-php5
sqlite3 php5-sqlite
```

(Ubuntu 12.10 LTS alatt tesztelve.)

A teljesség igénye nélkül néhány tipikus alkalmazást fogunk bemutatni.

Az alábbi egyszerű PHP kóddal ellenőrizhetjük, hogy működik-e az Apache-PHP-SQLite hármas. Kíratjuk ugyanis a PHP és az SQLite verziószámát.

```
<?php
print (phpversion());
print ("<br/>");
print_r (SQLite3::version());
?>
```

Mentsük a programot `ver.php` néven és másoljuk (tipikus megoldásként) a `/var/www` könyvtárba. Ha a telepítés rendben megtörtént, van olvasási jogunk a fájlra, továbbá van belépési jogunk a könyvtárba, akkor futtatás után valami ilyesmit kapunk:



1. ábra: Az Apache-PHP-SQLite használatra kész

Megjegyezzük, hogy az interneten található felhasználói leírások igen gyakran elavultak, még sokszor akkor is, ha friss fórumbejegyzések mutatnak rájuk. Óvatosan bánjunk hát velük, ha egy kód semmiképpennem akar működni, nem biztos, hogy a „mi készülékünkben van a hiba”.

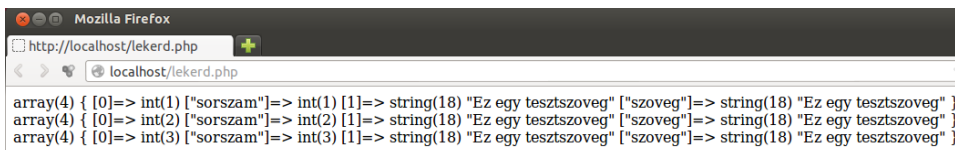
A következő példában megnyitjuk a `teszt.sqlite` adatbázist (ha még nem létezik, létrehozuk), azon belül beillesztünk egy rekordot a `tabla` nevű táblába, mindezt természetesen PHP-ből. Ha a tábla nem létezik, akkor létrejön. Egymás utáni végrehajtására tehát újra és újra ugyanazt a rekordot vesszük fel a táblába.

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('adatok/teszt.sqlite');
    }
}
$adatb = new MyDB();
$adatb->exec('CREATE TABLE tabla (sorszam INTEGER
PRIMARY KEY AUTOINCREMENT, szoveg STRING)');
$adatb->exec("INSERT INTO tabla (szoveg) VALUES
('Ez egy tesztszoveg')");
?>
```

Fontos, hogy ezúttal írási jogunk is legyen az adatok könyvtárba (ahol a `teszt.sqlite` fájl létrejön, illetve ahol szerkesztjük).

A következőekben az előzőleg feltöltött adatbázis tartalmát listázzuk ki a böngészőben. (Az egyes rekordok végén sort törtünk.)

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open("adatok/teszt.sqlite");
    }
}
$adatb = new MyDB();
$eredmeny = $adatb->query("SELECT * FROM tabla");
while ($sor = $eredmeny->fetchArray()) {
    var_dump($sor);
    print("<br/>");
}
?>
```



```
array(4) { [0]=> int(1) ["sorszam"]=> int(1) [1]=> string(18) "Ez egy tesztszoveg" ["szoveg"]=> string(18) "Ez egy tesztszoveg" }
array(4) { [0]=> int(2) ["sorszam"]=> int(2) [1]=> string(18) "Ez egy tesztszoveg" ["szoveg"]=> string(18) "Ez egy tesztszoveg" }
array(4) { [0]=> int(3) ["sorszam"]=> int(3) [1]=> string(18) "Ez egy tesztszoveg" ["szoveg"]=> string(18) "Ez egy tesztszoveg" }
```

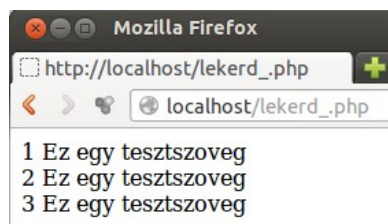
2. ábra: A `var_dump` kimenete

Szebb eredményt kapunk, ha a fenti kódban kicseréljük a `var_dump($sor);`

sort a

```
print($sor["sorszam"]." ". $sor["szoveg"]);
```

sorra.



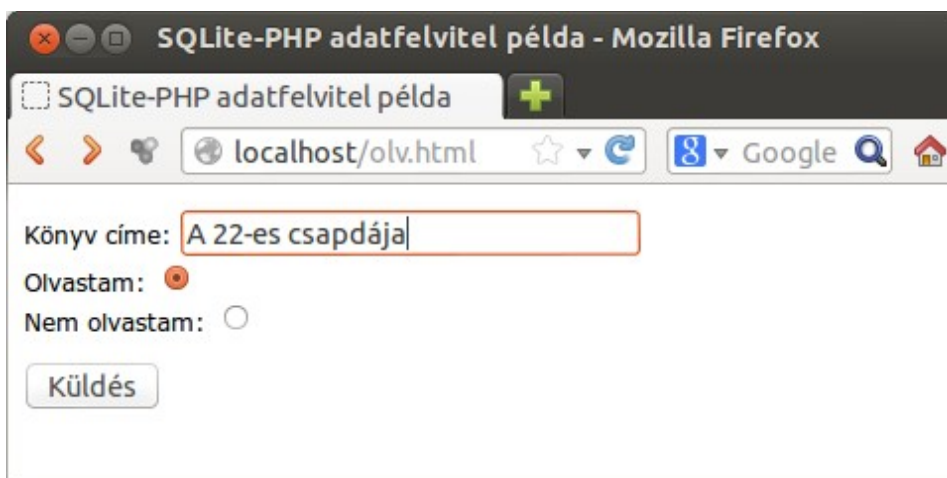
```
1 Ez egy tesztszoveg
2 Ez egy tesztszoveg
3 Ez egy tesztszoveg
```

3. ábra: Formázott kimenet

A következőekben a webes felületű adatbevitelre látunk egy egyszerű példát. Könyveink címét fogjuk elmenteni egy SQLite adatbázisba, valamint azt is, hogy olvastuk-e már. Lesz egy html fájlunk a felület kialakításához, és egy php fájl, ami az adatbázis és a felület közti kapcsolatot megteremti.

A valid html olv.html kódja:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<title>SQLite-PHP adatfelvitel példa</title>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
</head><body>
<form action="olv.php" method="post">
<p>
Könyv címe: <input type="text" name="cim"/><br/>
Olvastam: <input type="radio" value="I"
name="olv"/><br/>
Nem olvastam: <input type="radio" value="N"
name="olv"/>
</p>
<p>
<input type="submit"/>
</p>
</form>
</body>
</html>
```



4. ábra: Adatbevitel webes felületről

Az olv.php kódja:

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open("adatok/olv.sqlite");
    }
}
$adatb = new MyDB();
$olv = $_POST['olv'];
$cim = $_POST['cim'];
if (!empty($cim))
{
    $adatb->exec("CREATE TABLE tabla (cim STRING, olv
    STRING)");
    $adatb->exec("INSERT INTO tabla (cim,olv) VALUES
    ('$cim', '$olv')");
}
?>
```

(A szakirodalomban, fórumbejegyzésekben más paraméter átadási módokat is leírnak, azonban nem mindegyik működik a legújabb SQLite és PHP verziókkal.)

Végezetül fontosnak tartjuk felhívni az olvasó figyelmét arra az ismert tényre, hogy az adatbázis kezelők helytelen használata a webes sebezhetőségek egyik melegágya. Ez akár egy külön könyv is lehetne, ezért itt csupán álljon egy rövid leírás és néhány hivatkozás.

A weboldalak biztonságának tekintetében leggyakrabban a nem szűrt speciális karakterek és a helytelen típuskezelés okozza a legtöbb problémát. Ezeket használják ki az úgynevezett „SQL injection” támadás során. Az SQL injection lényege az, hogy a rosszindulatú felhasználók egy nem megfelelően ellenőrzött adatbeviteli mezőn keresztül hajtanak végre adatbázis lekérést, amely a probléma mélységétől függően akár bármilyen végrehajtható kódot is jelenthet.

A támadás alapvető ellenszere, hogy minden olyan mezőt, ami megbízhatatlan forrásból (például felhasználók számára küldött oldalakról) érkezett, alaposan meg kell vizsgálni és a problémás részeket hatástalanítani kell.

Mindezek mellett gondoskodni kell arról is, hogy illetéktelenek minél kevesebb információval rendelkezzenek magáról az adatbázisról, például ne tudhassák meg a táblák nevét beszédes hibaüzenetekből.

- http://en.wikipedia.org/wiki/SQL_injection
- http://pezia.hu/content/2009/03/08/weboldalak_biztons%C3%A1ga_1_sql_injection
- <http://duplaklikk.nemkf.com/node/782>

3.SQLite és Android

Androidon több különböző alternatívánk van az adattárolásra. Ezek közül nyilvánvalóan az aktuális igények alapján választunk. Egyik lehetőségünk az SQLite adatbázis használata, melyet az Android a legmagasabb szinten támogat.

Az Android tartalmaz egy C/C++ könyvtárat – természetesen az alap könyvtáron kívül, amely a JAVA funkcionalitás nagy részét megvalósítja –, amit az Android rendszer több összetevője is felhasznál. Ezek közül az egyik az SQLite adatbázis-motor. Az SQLite minden Android alkalmazás számára hozzáférhető.

Ha Androidon egy alkalmazás futása során jön létre egy SQLite adatbázis, akkor csak az használhatja, más alkalmazások számára elérhetetlen lesz. Ha létrejött az SQLite adatbázis, azt a `/data/data/<csomag_nev>/databases` könyvtárban találjuk.

Androidon SQLite adatbázist az `SQLiteOpenHelper` osztály segítségével hozhatunk létre. Ehhez létrehozunk egy saját alosztályt, melyben átírjuk az `onCreate()` metódust. A stringként megadott SQL utasítást az `execSQL` hajtja végre. Erre látunk példát az alábbiakban.

```
public class SOpenHelper extends SQLiteOpenHelper
{
    private static final int ABZ_VER=1;
    private static final String TABLA_NEV="konyvek";
    private static final String TABLA_LETREHOZASA=
    "CREATE TABLE " + TABLA_NEV + " (id INTEGER
    PRIMARY KEY AUTOINCREMENT, szerzok TEXT, cim TEXT
    NOT NULL, alcim TEXT);";
    SOpenHelper(Context context)
    {
        super(context, ABZ_NEV, null, ABZ_VER);
    }
    @Override
    public void onCreate(SQLiteDatabase abz)
    {
        abz.execSQL(TABLA_LETREHOZASA);
    }
}
```

```
}  
}
```