

SQLite

FTS eli full text search, tutustu lähemmin:

<http://www.xojo.com/blog/en/2014/03/full-text-search-with-sqlite.php>

SQL-komennot saavat olla lowercase tai uppercase.

Pääteikkunassa laita komentojen jälkeen ;

SELECT UNION SELECT
SELECT EXCEPT SELECT
SELECT nimi FROM (SELECT nimi, idhlo ...)
UNION-yhdistelmälle voi laittaa loppuun yhteisen ORDER BY

aiemmin olin käyttänyt aputauluja, nyt ei aina tarpeen:

```
SELECT idKas, Sunimi, Kaksois, LatSuku, Latlaji, HeiNimi, kpl,
KasHuom
FROM ((Kasvit LEFT JOIN Suvut ON iSuku = idSuku) LEFT JOIN Heimot ON
iHeimo = idHei)
LEFT JOIN (SELECT iKasvi, Count(*) AS kpl
FROM Havainnot INNER JOIN Kohteet ON idHava = iHava
GROUP BY iKasvi) ON idKas = iKasvi
WHERE kpl > 3;
```

```
sqlite> select idKas, Sunimi, kpl from Kasvit left join
...> (select iKasvi, count(*) as kpl from Havainnot inner join
...> Kohteet on idHava = iHava group by iKasvi)
...> on idKas = iKasvi where Sunimi like '%korpi%';
2007|Korpi-imarre|1
2008|Korpikaisla|5
2009|Korpipaatsama|3
sqlite> █
```

```
SELECT iKasvi, kpl, Kasvit.latlaji
FROM (SELECT iKasvi, count(*) AS kpl FROM Havainnot GROUP BY iKasvi
HAVING kpl > 1), Kasvit
WHERE Kasvit.idKas = iKasvi
```

Ohjelmakoodissa noukit joukon arvoja ja tekemättä tilapäistaulua haluat selvittää mikä niistä ei ole tietokannassa ennestään:

```
SELECT hlo FROM (SELECT 35 AS hlo UNION SELECT 36 UNION SELECT 44) AS
E WHERE hlo NOT IN (SELECT iHlo FROM HenkAlbt WHERE iAlbumi=5)
```

Kentän tietotyyppin muuttaminen (cast) hakulausekkeessa:

```
SELECT idPaikka, cast(Koord as text) AS koordt, ....
```

'SELECT COUNT DISTINCT' tehdään näin:
SELECT COUNT(*) AS kpl FROM (SELECT DISTINCT...)

alikyselyt

```
SELECT idKas, Sunimi FROM Kasvit WHERE idKas IN (SELECT DISTINCT iKas
FROM Havainnot)
```

```
-----  
SELECT idKas, Sunimi, LENGTH(Sunimi) FROM Kasvit  
WHERE LENGTH(Sunimi) > 20
```

(tai upper, length)

LENGTH(null) ei ole mitään, esim. seuraava antaa 0:

esim SELECT COUNT(*) FROM Paikat WHERE LENGTH(Kartta)<1

SELECT iKuva, LENGTH(Bitmap) FROM Bitmapit WHERE ...

Vaihda integer tekstiksi kyselyssä:

SELECT idPaikka, CAST(Koord AS Text) AS koortxt, ...

Funktiot UPPER(), LOWER() ja LIKE eivät selviä skandeista

```
-----  
SELECT COUNT(*) FROM Kasvit WHERE Kukkiaku = 5  
SELECT Painimi, COUNT(*) FROM Paikat GROUP BY Painimi  
...HAVING COUNT(*) > 20
```

AVG(), SUM(), MIN(), MAX()

antaa arvon 0 jos sarakkeita ei löydy:

SELECT MIN(idKoh) FROM Kohteet WHERE iHava = -8

```
-----  
... WHERE Arvo IS null  
... WHERE Arvo IS NOT null  
... WHERE Arvo = ''  
... WHERE NULLIF(Arvo, '') IS NULL (Arvo on joko NULL tai '')  
... WHERE Sunimi = 'Pujo' (case sensitive)  
... WHERE Painimi = 'Kenny's station'  
... WHERE Sunimi LIKE 'Metsä%' (not case sensitive)
```

Huom! LIKE ei selviä skandeista lower/upper

```
... WHERE Painimi NOT LIKE '%Kotipaha%'  
... WHERE Sunimi GLOB 'Metsä*' (case sensitive)
```

```
-----  
... ORDER BY Sunimi DESC  
... ORDER BY (SELECT COUNT(iPaik)....
```

operaattorit

```
Latsuku || ' ' || Latlaji  
'Kenny's station'
```

CASE expressions within select:

```
SELECT customerid, firstname, lastname,  
CASE country  
WHEN 'USA' THEN 'Dosmetic'  
ELSE 'Foreign'  
END CustomerGroup  
FROM customers ORDER BY LastName, FirstName
```

Tuloksena kentät customerid, firstname, lastname, CustomerGroup

```
-----  
CREATE TABLE testi (id integer primary key, sisus text, muu text)  
CREATE TABLE testi AS SELECT idi, nimi, maa FROM ...  
CREATE TEMP TABLE testi (...  
CREATE TEMP TABLE testi AS SELECT iKasvi, Laji FROM Kohteet
```

```
DROP TABLE testi
DROP TABLE IF EXISTS zapu
```

storage classes integer, real, text, blob, null

```
CREATE INDEX kas_sunimi ON Kasvit (Sunimi)
CREATE VIEW Kaslat AS SELECT ....
```

```
UPDATE Kasvit SET Sunimi = 'Pujo', Harvi = 1, Huom = null WHERE ...
```

UPDATE taulua käyttäen toisen taulun dataja:

taulussa tilapx on id ja polku, viedään taulun Kuvat kenttään Polku:

```
UPDATE Kuvat SET Polku =
(SELECT tilapx.polku FROM tilapx WHERE tilapx.id = Kuvat.idKuva)
WHERE EXISTS (SELECT * FROM tilapx WHERE tilapx.id = Kuvat.idKuva)
```

```
INSERT INTO testi (sisus, muu) VALUES ('Hesari', 'jotain')
INSERT INTO testi VALUES (null, 'Apu', 'jotain2')
INSERT INTO tilapx SELECT id, arvo FROM taulux
```

```
SELECT last_insert_rowid();    Puolipiste oltava mukana!
int lastid = [fmdb lastInsertRowId];
```

```
DELETE FROM Paikat WHERE id=413
```

```
SELECT date('now')      -> 2012-12-21
SELECT strftime('%m', '2012-09-21') -> '09'    '%d' -> '21'
SELECT strftime('%m%d', '2012-09-21') -> '0921'
SELECT strftime('%Y', '2012-12-21') -> 2012
SELECT date(Pvm) FROM Kuvat WHERE idKuva=789 -> 2011-06-16
SELECT time(Pvm) FROM Kuvat WHERE idKuva=789 -> 13:30:00
```

http://www.sqlite.org/lang_datefunc.html

Transactions

```
BEGIN
(commands)
COMMIT / ROLLBACK
(tai softan sisällä ohjelmallisesti)
```

UUSI KANTA

defaults, constraints, collations voidaan määritellä jos halutaan
Omat kantani aakkostaa skandit oikein ilman collation-määrittelyä

cache_size voidaan asettaa isommaksi kuin default jos paljon binääridataa

YLLÄPITO

pääteikkunassa aloita: `sqlite3 Herbaario.db`
pääteikkunasta pois: `.exit`
tee uusi tietokanta: `sqlite3 testi.db`

pääteikkunassa laita komentojen jälkeen ;

taulujen tutkimista:

```
PRAGMA table_info (Kasvit)
```

```
PRAGMA cache_size
```

```
PRAGMA temp_store
```

```
SELECT name FROM sqlite_master // taulujen nimet
```

```
SELECT type, name, tbl_name, sql FROM sqlite_master
```

```
... FROM sqlite_temp_master
```

cleaning the database: **VACUUM**

taulun nimen vaihtaminen:

```
ALTER TABLE nytnimi RENAME TO uusinimi
```

sarakkeen lisääminen tauluun:

```
ALTER TABLE Kuvat ADD COLUMN Maa TEXT
```
