Cruise sql cookbook

A collection of useful sql snipits to perfrom complex tasks on cruise database files

Table of Contents

[General Commands 2](#_Toc409538654)

[Foreign Key Check 2](#_Toc409538655)

[Finding out what tables reference each other 2](#_Toc409538656)

[Delete all field data 2](#_Toc409538657)

[Delete data from whole table columns 2](#_Toc409538658)

[Fixes for known issues 4](#_Toc409538659)

[Foreign key error from deleted stratum 4](#_Toc409538660)

[Cutting Unit Table 6](#_Toc409538661)

[Delete Cutting Unit Records 6](#_Toc409538662)

[Stratum Table 7](#_Toc409538663)

[Delete Stratum Records 7](#_Toc409538664)

[Sample Group Table 8](#_Toc409538665)

[Delete Sample Group Records 8](#_Toc409538666)

[Count Tree 9](#_Toc409538667)

[Create Count Records: Tally by Sample Group 9](#_Toc409538668)

[Create Count Records: Tally by Species 9](#_Toc409538669)

# General Commands

## Foreign Key Check

The foreign key check command is a useful command for checking the integrity of the data in the database. It is especially important when modifying data using a SQL editor. When ran it generates a list of each record in the database where a foreign key reference is broken.

pragma foreign\_key\_check;

## Finding out what tables reference each other

With a large number of tables in the database it can become confusing figuring out where all the references point. Here is a command that will find all the references pointing to the stratum table. Replace Stratum with any other table name, to use for other tables

SELECT \* FROM sqlite\_master WHERE type = 'table' and sql GLOB '\*REFERENCES Stratum[ ,]\*';

Finding out what tables a table references is easy. Use the following command; replace table-name with the name of the desired table.

PRAGMA foreign\_key\_list(‘table-name’);

## Delete all field data

PRAGMA foreign\_keys = OFF;--optional--

UPDATE CountTree Set TreeCount = 0, SumKPI = 0;

UPDATE SampleGroup Set SampleSelectorState = '';

DELETE FROM Log;

DELETE FROM Stem;

DELETE FROM LogStock;

DELETE FROM TreeCalculatedValues;

DELETE FROM Tree;

DELETE FROM Plot;

DELETE FROM ErrorLog WHERE TableName IN ('Tree', 'Plot', 'Log', 'Stem');

UPDATE CuttingUnit Set TallyHistory = NULL;

## Delete data from whole table columns

There are two methods that can be used when deleting data from the columns in a table. The simplest method, useful when only a few columns need to be cleared, is to use an update command. ( See SQLite ref: <http://sqlite.org/lang_update.html> )

UPDATE Tree Set DBH = 0;

The down side to this is that it doesn’t automatically account for the default value of the column (if one exists) and the command will need to be updated as more columns are added to the table.

The next method uses a combination of Insert/Replace and a Select command. This involves two steps. First we use the select command to read the data we want to keep from the table. If the table has a primary key(PK) then the select command should include the PK. The second step we use an INSERT OR REPLACE command to reinsert the data back into the table. The advantages of using this method is that allows you to erase column data by specifying what data you want to keep rather than what data you don’t want to keep, also it allows you to set the data you don’t want to keep back to its default value without requiring you to know what the default value should be.

INSERT OR REPLACE INTO Tree (Tree\_CN, CuttingUnit\_CN, Stratum\_CN, Plot\_CN, TreeNumber, CreatedBy, CreatedDate)

SELECT Tree\_CN, CuttingUnit\_CN, Stratum\_CN, Plot\_CN, TreeNumber, CreatedBy, CreatedDate FROM Tree;

# Fixes for known issues

## Foreign key error from deleted stratum

Prior to Cruise Manager version 8.05 there was a bug when deleting a stratum, would create foreign key errors.

The table where foreign key errors would be created were: SampleGroup, TreeFieldSetup, and LogFieldSetup. Note that there should not be any foreign key errors in the tree table because Cruise Manager doesn’t allow stratum to be deleted if it contains trees, but it is a possibility.

Before we start deleting references to the deleted stratum, check to see if there are and references to deleted strata in the tree, plot or CountTree tables to make sure we are not going to delete useful data.

SELECT \* FROM Tree LEFT JOIN Stratum USING (Stratum\_CN) WHERE Stratum.Stratum\_CN IS NULL;

SELECT \* FROM Plot LEFT JOIN Stratum USING (Stratum\_CN) WHERE Stratum.Stratum\_CN IS NULL;

SELECT \* FROM CountTree JOIN SampleGroup USING (SampleGroup\_CN) LEFT JOIN Stratum USING (Stratum\_CN) WHERE Stratum.Stratum\_CN IS NULL;

Here are the commands that will delete the records with invalid references to the deleted strata. They are in a specific order because some need to be executed before others, but we will turn on foreign key checking to insure that these commands are executed in the proper order.

PRAGMA foreign\_keys = ON;

DELETE FROM Treefieldsetup WHERE rowid IN (SELECT lft.rowid FROM TreeFieldSetup AS lft LEFT JOIN stratum USING (stratum\_cn) WHERE stratum.Stratum\_CN IS NULL);

DELETE FROM logfieldsetup WHERE rowid IN (SELECT lft.rowid FROM LogFieldSetup AS lft LEFT JOIN stratum USING (stratum\_cn) WHERE stratum.[Stratum\_CN] IS NULL);

--next two commands optional, if we need to delete hanging tree or plot records—

DELETE FROM Plot WHERE Plot\_cn in (SELECT Plot.Plot\_CN FROM Plot LEFT JOIN Stratum USING (Stratum\_CN) WHERE Stratum.Stratum\_cn IS NULL);

DELETE FROM Tree WHERE Tree\_cn in (SELECT Tree.Tree\_CN FROM Tree LEFT JOIN Stratum USING (Stratum\_CN) WHERE Stratum.Stratum\_cn IS NULL);

DELETE FROM SampleGroupTreeDefaultValue WHERE SampleGroup\_cn IN (SELECT samplegroup\_cn FROM sampleGroup LEFT JOIN stratum USING (stratum\_cn) WHERE stratum.stratum\_cn IS NULL);

DELETE FROM samplegroup WHERE samplegroup\_cn IN (SELECT samplegroup\_cn FROM sampleGroup LEFT JOIN stratum USING (stratum\_cn) WHERE stratum.stratum\_cn IS NULL);

PRAGMA foreign\_key\_check;--optional, will tell us if all foreign key errors were fixed--

# Cutting Unit Table

## Delete Cutting Unit Records

Delete From CuttingUnitStratum WHERE CuttingUnit\_CN = {0};

DELETE FROM Log WHERE Tree\_CN IN (SELECT Tree\_CN FROM Tree WHERE Tree.CuttingUnit\_CN = {0});

DELETE FROM LogStock WHERE EXISTS (SELECT 1 FROM Tree WHERE Tree.Tree\_CN = LogStock.Tree\_CN AND Tree.CuttingUnit\_CN = {0});

DELETE FROM TreeCalculatedValues WHERE EXISTS (SELECT 1 FROM Tree WHERE Tree.Tree\_CN = TreeCalculatedValues.Tree\_CN AND Tree.CuttingUnit\_CN = {0});

DELETE FROM Tree WHERE CuttingUnit\_CN = {0};

DELETE FROM Plot WHERE CuttingUnit\_CN = {0};

DELETE FROM CountTree WHERE CuttingUnit\_CN = {0};

DELETE FROM CuttingUnit WHERE CuttingUnit\_CN = {0};

# Stratum Table

## Delete Stratum Records

DELETE From CuttingUnitStratum WHERE Stratum\_CN = {0};

DELETE FROM Log WHERE Tree\_CN IN (SELECT Tree\_CN FROM Tree WHERE Tree.Stratum\_CN = {0});

DELETE FROM LogStock WHERE Tree\_CN IN (SELECT Tree\_CN FROM Tree WHERE Tree.Stratum\_CN = {0});

DELETE FROM TreeCalculatedValues WHERE Tree\_CN IN (SELECT 1 FROM Tree WHERE Tree.Stratum\_CN = {0});

DELETE FROM Tree WHERE Stratum\_CN = {0};

DELETE FROM Plot WHERE Stratum\_CN = {0};

DELETE FROM TreeEstimate WHERE CountTree\_CN IN (SELECT CountTree\_CN FROM CountTree JOIN SampleGroup USING (SampleGroup\_CN) WHERE Stratum\_CN = {0});

DELETE FROM CountTree WHERE SampleGroup\_CN IN (SELECT SampleGroup\_CN FROM SampleGroup WHERE SampleGroup.Stratum\_CN = {0});

DELETE FROM SampleGroupTreeDefaultValue WHERE SampleGroup\_CN IN (SELECT SampleGroup\_CN FROM SampleGroup WHERE SampleGroup.Stratum\_CN = {0});

DELETE FROM SampleGroup WHERE Stratum\_CN = {0};

DELETE FROM TreeFieldSetup WHERE Stratum\_CN = {0};

DELETE FROM LogFieldSetup WHERE Stratum\_CN = {0};

DELETE FROM Stratum WHERE Stratum\_CN = {0};

# Sample Group Table

## Delete Sample Group Records

DELETE FROM Log WHERE Tree\_CN IN (SELECT Tree\_CN FROM Tree WHERE Tree.SampleGroup\_CN = {0});

DELETE FROM LogStock WHERE Tree\_CN IN (SELECT Tree\_CN FROM Tree WHERE Tree.SampleGroup\_CN = {0});

DELETE FROM TreeCalculatedValues WHERE Tree\_CN IN (SELECT Tree\_CN FROM Tree WHERE Tree.SampleGroup\_CN = {0});

DELETE FROM Tree WHERE SampleGroup\_CN = {0};

DELETE FROM TreeEstimate WHERE CountTree\_CN IN (SELECT CountTree\_CN FROM CountTree WHERE SampleGroup\_CN = {0});

DELETE FROM CountTree WHERE SampleGroup\_CN = {0};

DELETE FROM SampleGroupTreeDefaultValue WHERE SampleGroup\_CN = {0};

DELETE FROM SampleGroup WHERE SampleGroup\_CN = {0};

# Count Tree

## Create Count Records For newly added Cutting Units

The fallowing command looks for cutting units that don’t have any records in the count tree table, then cross references that with the strata in those units, then finds preexisting count tree records for sample groups in those strata, then creates new count tree records using the preexisting ones as a guide.

To test the output of the command before running it, run the SELECT portion of the command first:

SELECT CuttingUnitStratum.CuttingUnit\_CN, CountTree.SampleGroup\_CN, CountTree.TreeDefaultValue\_CN, CountTree.Tally\_CN, 'script' as cb

FROM CountTree

JOIN SampleGroup USING (SampleGroup\_CN)

JOIN CuttingUnitStratum USING (Stratum\_CN)

WHERE CuttingUnitStratum.CuttingUnit\_CN IN

(SELECT CuttingUnit\_CN FROM CuttingUnit EXCEPT SELECT CuttingUnit\_CN FROM CountTree)

GROUP BY CountTree.SampleGroup\_CN, CountTree.TreeDefaultValue\_CN;

Then run the whole command if the SELECT command output looks correct:

INSERT OR IGNORE INTO CountTree (CuttingUnit\_CN, SampleGroup\_CN, TreeDefaultValue\_CN, Tally\_CN, CreatedBy) SELECT CuttingUnitStratum.CuttingUnit\_CN, CountTree.SampleGroup\_CN, CountTree.TreeDefaultValue\_CN, CountTree.Tally\_CN, 'script' as cb FROM CountTree JOIN SampleGroup USING (SampleGroup\_CN) JOIN CuttingUnitStratum USING (Stratum\_CN) WHERE CuttingUnitStratum.CuttingUnit\_CN IN (SELECT CuttingUnit\_CN FROM CuttingUnit EXCEPT SELECT CuttingUnit\_CN FROM CountTree) GROUP BY CountTree.SampleGroup\_CN, CountTree.TreeDefaultValue\_CN;

## Create Count Records: Tally by Sample Group

note: replace {1} with the SampleGroup\_CN value

INSERT OR Fail INTO CountTree (CuttingUnit\_CN, SampleGroup\_CN, CreatedBy)

Select CuttingUnitStratum.CuttingUnit\_CN, SampleGroup.SampleGroup\_CN, 'NA' AS CreatedBy

From SampleGroup

INNER JOIN CuttingUnitStratum

ON SampleGroup.Stratum\_CN = CuttingUnitStratum.Stratum\_CN

WHERE SampleGroup.SampleGroup\_CN = {1};

## Create Count Records: Tally by Species

note: replace {1} with the SampleGroup\_CN value

INSERT OR IGNORE INTO CountTree (CuttingUnit\_CN, SampleGroup\_CN, TreeDefaultValue\_CN, CreatedBy)

Select CuttingUnitStratum.CuttingUnit\_CN, SampleGroup.SampleGroup\_CN, SampleGroupTreeDefaultValue.TreeDefaultValue\_CN, 'NA' AS CreatedBy

From SampleGroup

INNER JOIN CuttingUnitStratum

ON SampleGroup.Stratum\_CN = CuttingUnitStratum.Stratum\_CN

INNER JOIN SampleGroupTreeDefaultValue

ON SampleGroupTreeDefaultValue.SampleGroup\_CN = SampleGroup.SampleGroup\_CN

WHERE SampleGroup.SampleGroup\_CN = {1};

# Appendix i: Recommended Software

## SQLite Shell/Command line interface

The SQLite shell is a command line application that provides very basic interaction with a SQLite database, but it offers some very useful functionality. Most notably it allows the ability to dump table data, can export data in various text formats, such as, CSV. More information on the command line interface can be found here: <http://sqlite.org/cli.html>

Download: <http://sqlite.org/download.html>

## SQLiteman

SQLiteman is an open source GUI application with simple interface, but offers some useful features out of the box. It offers the ability export data to CSV, XML, SQL, and Clipboard, as well it has the ability to generate random data. For non-programmers this is probably the best option.

Download: <http://sqliteman.yarpen.cz>

## SQLite Expert

SQLite Expert is another GUI application for editing databases. It offers a paid and free version with some features disabled in the latter. This application excels mainly in customizability. Its allows you to select SQLite library dll, and allows importing from any other ADO database.

Download: <http://www.sqliteexpert.com>