

# Introduction to Database Systems: CS312

## Updating Tables

Oliver Bonham-Carter

18 Feb 2019

# Alter a Table's Name

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples

Deleting  
Entries

DB Browser

- ALTER TABLE command modifies an existing table without performing a full dump and reload of the data.
- Rename an existing table

```
ALTER TABLE database_name.table_name  
    RENAME TO new_table_name;
```

# Alter a Table's Name

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples

Deleting  
Entries

DB Browser

- Create a new table using *Insert*.

```
INSERT INTO first_table_name  
[(column1, column2, ... columnN)]  
    SELECT column1, column2, ...columnN  
FROM second_table_name  
[WHERE condition];
```

# Change a Table's Name

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples

Deleting  
Entries

DB Browser

- Add a new table

```
CREATE TABLE newTable(  
    Id varchar,  
    name varchar,  
    totCred numeric  
);
```

.tables

- Change table's name

```
ALTER TABLE newTable RENAME TO oldTable;
```

.tables

# Create and Populate New Table Using AS Keyword

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples

Deleting  
Entries

DB Browser

- Used to create a table from an existing table by copying the existing table's columns with data structures

```
CREATE TABLE new_table AS SELECT expressions  
FROM existing_tables  
[WHERE conditions];
```

```
DROP TABLE sTable;  
CREATE TABLE sTable AS  
    SELECT id, name, totCred  
    FROM student  
    WHERE ID in ("S1","S2","S3","S4","S5");  
  
.tables  
.schema sTable
```

```
INSERT INTO sTable values("S100","Fletcher",4);
```

# Removing Tables or Table Data

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples

Deleting  
Entries

DB Browser

- Remove only table's data but leave the empty table in the DB

```
DELETE FROM tableName;
```

- Remove entire table *and* its data too

```
DROP TABLE tableName;
```

# Delete Tuples From a Table

First, let's add a row (i.e., a tuple) to delete ...

- Used to delete data in a table

```
DELETE FROM table_name  
WHERE [condition];
```

- Add a record to erase for this example

```
select * from instructor;
```

```
INSERT INTO instructor  
VALUES ( "E10101", "Potter",  
        "ES1", "English", 90000.00 );
```

```
/* Did the tuple arrive into the table?*/  
select * from instructor;
```

# Delete Tuples From a Table

Second, we delete the recently added row (i.e., a tuple)

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples

Deleting  
Entries

DB Browser

- Delete all tuples in the Instructor for a particular condition

```
delete from instructor where name == "Potter";
```

- Be more specific: add to the conditional part:

```
delete from instructor where name == "Potter"  
and deptName == "English";
```



# Update an Entry

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples

Deleting  
Entries

DB Browser

- Create a new table using *Update*.

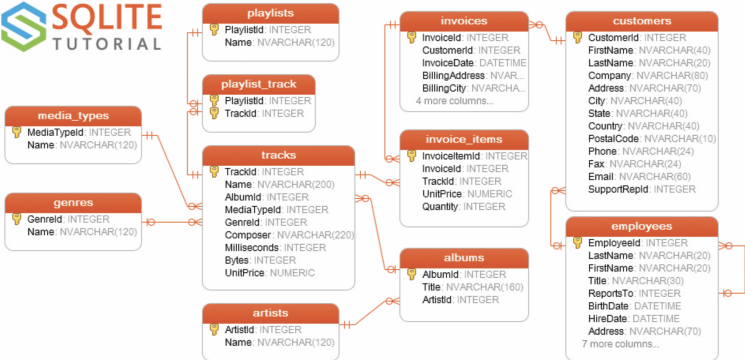
```
UPDATE table_name  
    SET column1 = value1,  
        column2 = value2....,  
        columnN = valueN  
WHERE [condition];
```

- Add an update to the Student table: Replace the ID and credits belonging to "Beuller"

```
UPDATE student SET ID = "XS5" where name == "Beuller";
```

```
UPDATE student SET ID = "S5", totCred = 55  
WHERE name == "Beuller";
```

# New database: Chinook's Schema



- Ref: <http://www.sqlitetutorial.net/sqlite-sample-database/>

Modifying  
Tables

Copying  
tables

Removing  
Data

Deleting  
Tuples


Deleting  
Entries

DB Browser

SQLite Database Browser - /Users/jc/tmp/example.db

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table:  

New Record Delete Record

|   | list          | month      | members |
|---|---------------|------------|---------|
|   | Filter        | Filter     | Filter  |
| 1 | gluster-board | 2013-09-05 | 99999   |
| 2 | gluster-users | 2013-09-05 | 99999   |

< 1 - 2 of 12 >

Go to:

SQL Log

Show SQL submitted by  Clear

```
PRAGMA foreign_keys = "1";
PRAGMA encoding
SELECT type, name, sql, tbl_name FROM sqlite_master;
SELECT COUNT(*) FROM (SELECT rowid,* FROM 'total_members' ORDER BY 'rowid' ASC);
SELECT rowid,* FROM 'total_members' ORDER BY 'rowid' ASC LIMIT 0, 50000;
```

UTF-8

## Let's Try It Out!

- Locate the sandbox database called `sandbox/chinook.sqlite3`
- Test-out the SQLite tool

**THINK**