Introduction to Database Systems: CS312
Updating Tables

Oliver Bonham-Carter

15 Feb 2019

Alter a Table's Name

- 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

- 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

- Add a new table

```
CREATE TABLE newTable(
  Id varchar(5),
  name varchar(20),
  totCred numeric(3,0)
  );
```

- Change a table's name

```
 ALTER TABLE newTable RENAME TO oldTable;

.tables
```

Create Table Using AS Keyword

- Used to create a table from an existing table by copying the existing table's columns.

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

Create a new table ("sTable")

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

Add data to test

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

Removing Tables or Table Data

- 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);

   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

- Create a new table using *Update*.

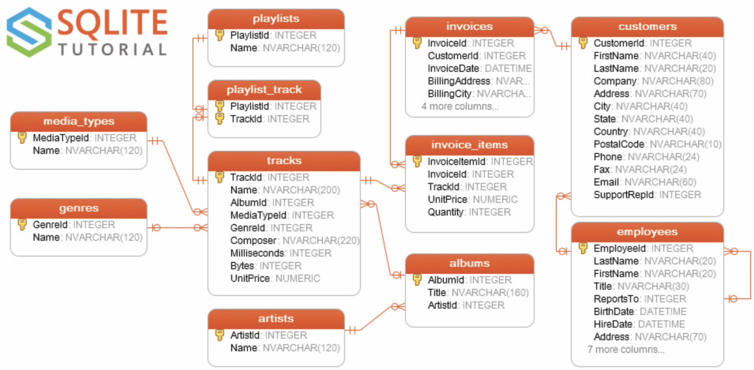
```
UPDATE table_name
   SET column1 = value1,
   column2 = value2....,
   columnN = valueN
   WHERE [condition];
```

- 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



playlists
PlaylistId: INTEGER
Name: NVARCHAR(120)

playlist_track
PlaylistId: INTEGER
TrackId: INTEGER

media_types
MediaTypeId: INTEGER
Name: NVARCHAR(120)

genres
GenreId: INTEGER
Name: NVARCHAR(120)

tracks
TrackId: INTEGER
Name: NVARCHAR(200)
AlbumId: INTEGER
MediaTypeId: INTEGER
GenreId: INTEGER
Composer: NVARCHAR(220)
Milliseconds: INTEGER
Bytes: INTEGER
UnitPrice: NUMERIC

artists
ArtistId: INTEGER
Name: NVARCHAR(120)

invoices
InvoiceId: INTEGER
CustomerId: INTEGER
InvoiceDate: DATETIME
BillingAddress: NVAR...
BillingCity: NVARCHA...
4 more columns...

invoice_items
InvoiceItemId: INTEGER
InvoiceId: INTEGER
TrackId: INTEGER
UnitPrice: NUMERIC
Quantity: INTEGER

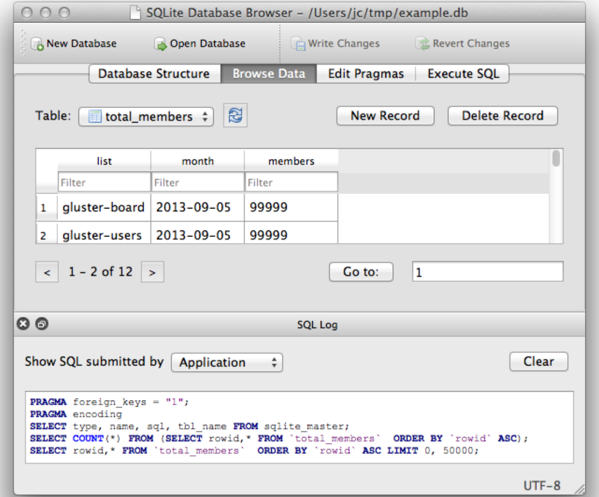
albums
AlbumId: INTEGER
Title: NVARCHAR(160)
ArtistId: INTEGER

customers
CustomerId: INTEGER
FirstName: NVARCHAR(40)
LastName: NVARCHAR(20)
Company: NVARCHAR(80)
Address: NVARCHAR(70)
City: NVARCHAR(40)
State: NVARCHAR(40)
Country: NVARCHAR(40)
PostalCode: NVARCHAR(10)
Phone: NVARCHAR(24)
Fax: NVARCHAR(24)
Email: NVARCHAR(60)
SupportRepId: INTEGER

employees
EmployeeId: INTEGER
LastName: NVARCHAR(20)
FirstName: NVARCHAR(20)
Title: NVARCHAR(30)
ReportsTo: INTEGER
BirthDate: DATETIME
HireDate: DATETIME
Address: NVARCHAR(70)
7 more columns...

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

DB Browser



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

```
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;
```

DB Browser

Let's Try It Out!

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