

Owen Dix

9/16/2015

Basic SQL Statement

5.

```
Command Prompt - mysql -u root -p
Database changed
mysql> show tables;
+-----+
| Tables_in_itunesdb |
+-----+
| album               |
| artist              |
+-----+
2 rows in set (0.02 sec)

mysql> describe album;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| albumID    | int(11)       | NO   |     | NULL    |       |
| artistID   | mediumint(9)  | NO   |     | NULL    |       |
| albumName  | varchar(60)   | NO   |     | NULL    |       |
| trackList  | varchar(200)  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql> describe artist;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| artistID   | mediumint(9)  | NO   |     | NULL    |       |
| artistName | varchar(60)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql>
```

6.

```
cmd Command Prompt - mysql -u root -p

mysql> describe album;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| albumID    | int(11)       | NO   |     | NULL    |       |
| artistID   | mediumint(9)  | NO   |     | NULL    |       |
| albumName  | varchar(60)   | NO   |     | NULL    |       |
| trackList  | varchar(200)  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql> describe artist;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| artistID   | mediumint(9)  | NO   |     | NULL    |       |
| artistName | varchar(60)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> alter table album MODIFY albumID INT UNSIGNED NOT NULL;
Query OK, 0 rows affected (0.57 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table artist MODIFY artistID MEDIUMINT UNSIGNED NOT NULL;
Query OK, 0 rows affected (0.23 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

7. I modified the Album table's albumID field to make it an unsigned int data type (not null). I modified the Artist table's artistID field to make it an unsigned mediumint (not null).

8. I already had a field for a primary key in both of my tables, in the first column position, but I modified them so they were auto-incrementing and made them primary keys.

9. The Artist table has two fields: artistID, and artistName. ArtistID I have set aside as the primary key. ArtistName would not make a good primary key because, like a person's name, an artist's name may not be unique.

The Album table has four fields: albumID, artistID, albumName, trackList. AlbumID is the primary key. artistID would not make a good primary key because an artist can have multiple albums, so it would not be unique. The albumName would not make a good primary key because different artists could have albums with the same name, so it is also not unique. The trackList might not be unique either, but if there is a set of tracks with the same name it would be unlikely. However, besides possibly not being unique, the trackList is also a very long entry. With up to 200 characters, trackList would not be efficient at accessing the data.

10.

```
Command Prompt - mysql -u root -p

D PRIMARY KEY (albumID);
Query OK, 0 rows affected (0.23 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table artist MODIFY artistID MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT, ADD PRIMARY KEY (artistID);
Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe album;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| albumID    | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
| artistID   | mediumint(9)        | NO   |     | NULL    |                |
| albumName  | varchar(60)         | NO   |     | NULL    |                |
| trackList  | varchar(200)        | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)

mysql> describe artist;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| artistID   | mediumint(8) unsigned | NO   | PRI | NULL    | auto_increment |
| artistName | varchar(60)         | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql>
```

11.

```
mysql> insert album set artistID=5, albumName='The Two Hump Kind', trackList='no water, we spit';
Query OK, 1 row affected (0.36 sec)

mysql> insert album set artistID=6, albumName='We Rock', trackList='bark, ruff, scratch, lick, eat, smile';
Query OK, 1 row affected (0.36 sec)

mysql> insert album set artistID=6, albumName='We Roll...in Stuff', trackList='happy to see you - single';
Query OK, 1 row affected (0.09 sec)

mysql> select * from artist;
+-----+
| artistID | artistName |
+-----+
| 1 | aarons, The |
| 2 | aardvarks, The |
| 3 | applesauce metalheads |
| 4 | badger barbers |
| 5 | camel captains |
| 6 | doggy destroyers |
+-----+
6 rows in set (0.00 sec)

mysql> select * from album;
+-----+-----+-----+-----+
| albumID | artistID | albumName | trackList |
+-----+-----+-----+-----+
| 1 | 1 | Hi Aaron | aaron1, aaron2, aaron3 |
| 2 | 2 | Rolling Strong | we roll, we rock, we eat stuff |
| 3 | 3 | The Spicy Kind | motts, umm, some other applesauce...with metal in it |
| 4 | 4 | Grrr | hairy mammal, i eat, i mate, i reproduce, im angry |
| 5 | 5 | The Two Hump Kind | no water, we spit |
| 6 | 6 | We Rock | bark, ruff, scratch, lick, eat, smile |
| 7 | 6 | We Roll...in Stuff | happy to see you - single |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

12.     a. What are the names of the Albums by artistID = 1?
- b. What are the tracks on the album named Grrr?
- c. What are the names of the Albums and Tracks by artistID = 6?
- d. How many artists are in the database?
- e. What is the artist name of artistID = 3?
- f. What are all artist names that begin with the letter “a”? (I know this requires more than this lesson but it just took a simple internet search, so I hope you’ll allow it.)
13.     a.

```
mysql> select albumName from album where artistID = 1;
+-----+
| albumName |
+-----+
| Hi Aaron |
+-----+
1 row in set (0.00 sec)
```

b.

```
mysql> select trackList from album where albumName = "Grrr";
+-----+
| trackList |
+-----+
| hairy mammal, i eat, i mate, i reproduce, im angry |
+-----+
1 row in set (0.04 sec)
```

c.

```
mysql> select albumName, trackList from album where artistID = 6;
+-----+-----+
| albumName | trackList |
+-----+-----+
| We Rock | bark, ruff, scratch, lick, eat, smile |
| We Roll...in Stuff | happy to see you - single |
+-----+-----+
2 rows in set (0.00 sec)
```

d.

```
mysql> select count(*) from artist;
+-----+
| count(*) |
+-----+
| 6 |
+-----+
1 row in set (0.08 sec)
```

e.

```
mysql> select artistName from artist where artistID = 3;
+-----+
| artistName |
+-----+
| applesauce metalheads |
+-----+
1 row in set (0.00 sec)
```

f.

```
mysql> select artistName from artist where artistName like "a%";
+-----+
| artistName |
+-----+
| aarons, The |
| aardvarks, The |
| applesauce metalheads |
+-----+
3 rows in set (0.05 sec)
```

14.

```
mysql> create table testArtist as select * from artist;
Query OK, 6 rows affected (0.17 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> create table testAlbum as select * from album;
Query OK, 7 rows affected (0.22 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> show tables;
+-----+
| Tables_in_itunesdb |
+-----+
| album               |
| artist              |
| testalbum           |
| testartist          |
+-----+
4 rows in set (0.02 sec)
```

15.

16 (before Update 1 to testAlbum):

```
mysql> select * from testalbum;
+-----+-----+-----+-----+
| albumID | artistID | albumName | trackList |
+-----+-----+-----+-----+
| 1 | 1 | Hi Aaron | aaron1, aaron2, aaron3 |
| 2 | 2 | Rolling Strong | we roll, we rock, we eat stuff |
| 3 | 3 | The Spicy Kind | motts, umm, some other applesauce...with metal in it |
| 4 | 4 | Grrr | hairy mammal, i eat, i mate, i reproduce, im angry |
| 5 | 5 | The Two Hump Kind | no water, we spit |
| 6 | 6 | We Rock | bark, ruff, scratch, lick, eat, smile |
| 7 | 6 | We Roll...in Stuff | happy to see you - single |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Update 1 to testAlbum:

```
mysql> update testalbum set trackList='aaron1, aaron2, aaron3, aaron4' where albumID=1;
Query OK, 1 row affected (0.10 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

16 (after Update 1 to testAlbum):

```
mysql> select * from testalbum;
+-----+-----+-----+-----+
| albumID | artistID | albumName | trackList |
+-----+-----+-----+-----+
| 1 | 1 | Hi Aaron | aaron1, aaron2, aaron3, aaron4 |
| 2 | 2 | Rolling Strong | we roll, we rock, we eat stuff |
| 3 | 3 | The Spicy Kind | motts, umm, some other applesauce...with metal in it |
| 4 | 4 | Grrr | hairy mammal, i eat, i mate, i reproduce, im angry |
| 5 | 5 | The Two Hump Kind | no water, we spit |
| 6 | 6 | We Rock | bark, ruff, scratch, lick, eat, smile |
| 7 | 6 | We Roll...in Stuff | happy to see you - single |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

16 (before Update 2 to testAlbum):

```
mysql> select * from testalbum;
+-----+-----+-----+-----+
| albumID | artistID | albumName          | trackList          |
+-----+-----+-----+-----+
| 1       | 1       | Hi Aaron           | aaron1, aaron2, aaron3, aaron4 |
| 2       | 2       | Rolling Strong     | we roll, we rock, we eat stuff |
| 3       | 3       | The Spicy Kind     | motts, umm, some other applesauce...with metal in it |
| 4       | 4       | Grrr               | hairy mammal, i eat, i mate, i reproduce, im angry |
| 5       | 5       | The Two Hump Kind  | no water, we spit |
| 6       | 6       | We Rock            | bark, ruff, scratch, lick, eat, smile |
| 7       | 6       | We Roll...in Stuff | happy to see you - single |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Update 2 to testAlbum:

```
mysql> update testalbum set albumName='We Roll...In Stuff' where albumID=7;
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

16 (after Update 2 to testAlbum):

```
mysql> select * from testAlbum;
+-----+-----+-----+-----+
| albumID | artistID | albumName          | trackList          |
+-----+-----+-----+-----+
| 1       | 1       | Hi Aaron           | aaron1, aaron2, aaron3, aaron4 |
| 2       | 2       | Rolling Strong     | we roll, we rock, we eat stuff |
| 3       | 3       | The Spicy Kind     | motts, umm, some other applesauce...with metal in it |
| 4       | 4       | Grrr               | hairy mammal, i eat, i mate, i reproduce, im angry |
| 5       | 5       | The Two Hump Kind  | no water, we spit |
| 6       | 6       | We Rock            | bark, ruff, scratch, lick, eat, smile |
| 7       | 6       | We Roll...In Stuff | happy to see you - single |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

16 (before Update 1 to testArtist):

```
mysql> select * from testartist;
+-----+-----+
| artistID | artistName |
+-----+-----+
| 1       | aarons, The |
| 2       | aardvarks, The |
| 3       | applesauce metalheads |
| 4       | badger barbers |
| 5       | camel captains |
| 6       | doggy destroyers |
+-----+-----+
6 rows in set (0.00 sec)
```

Update 1 to testArtist:

```
mysql> update testartist set artistName='Zod';
Query OK, 6 rows affected (0.08 sec)
Rows matched: 6  Changed: 6  Warnings: 0
```

16. (after Update 1 to testArtist):

```
mysql> select * from testartist;
+-----+-----+
| artistID | artistName |
+-----+-----+
| 1 | Zod |
| 2 | Zod |
| 3 | Zod |
| 4 | Zod |
| 5 | Zod |
| 6 | Zod |
+-----+-----+
6 rows in set (0.00 sec)
```

16 (before Update 2 to testArtist):

```
mysql> select * from testartist;
+-----+-----+
| artistID | artistName |
+-----+-----+
| 1 | Zod |
| 2 | Zod |
| 3 | Zod |
| 4 | Zod |
| 5 | Zod |
| 6 | Zod |
+-----+-----+
6 rows in set (0.00 sec)
```

Update 2 to testArtist:

```
mysql> update testartist set artistName='Joe Mama' where artistID = 3 or artistID = 4;
Query OK, 2 rows affected (0.16 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

16 (after Update 2 to testartist):

```
mysql> select * from testartist;
+-----+-----+
| artistID | artistName |
+-----+-----+
| 1 | Zod |
| 2 | Zod |
| 3 | Joe Mama |
| 4 | Joe Mama |
| 5 | Zod |
| 6 | Zod |
+-----+-----+
6 rows in set (0.00 sec)
```

17. (Created delTestAlbum)

18.

```
mysql> delete from deltestalbum where artistID=2;
Query OK, 1 row affected (0.09 sec)
```

19.

```
mysql> delete from deltestalbum;
Query OK, 6 rows affected (0.08 sec)
```



20.

```
mysql> select * from deltestalbum;  
Empty set (0.00 sec)
```

21.

```
mysql> insert into deltestalbum select * from album;  
Query OK, 7 rows affected (0.36 sec)  
Records: 7 Duplicates: 0 Warnings: 0
```

22.

```
mysql> select * from deltestalbum;  
+-----+-----+-----+-----+  
| albumID | artistID | albumName | trackList |  
+-----+-----+-----+-----+  
| 1 | 1 | Hi Aaron | aaron1, aaron2, aaron3 |  
| 2 | 2 | Rolling Strong | we roll, we rock, we eat stuff |  
| 3 | 3 | The Spicy Kind | motts, umm, some other applesauce...with metal in it |  
| 4 | 4 | Grrr | hairy mammal, i eat, i mate, i reproduce, im angry |  
| 5 | 5 | The Two Hump Kind | no water, we spit |  
| 6 | 6 | We Rock | bark, ruff, scratch, lick, eat, smile |  
| 7 | 6 | We Roll...in Stuff | happy to see you - single |  
+-----+-----+-----+-----+  
7 rows in set (0.00 sec)
```

23.

```
mysql> truncate table deltestalbum;  
Query OK, 0 rows affected (0.08 sec)
```

24.

```
mysql> drop table deltestalbum;  
Query OK, 0 rows affected (0.14 sec)
```

25.

```
mysql> show tables;  
+-----+  
| Tables_in_itunesdb |  
+-----+  
| album |  
| artist |  
| testalbum |  
| testartist |  
+-----+  
4 rows in set (0.00 sec)
```

27. With this exercise, I've learned how to alter the structure of a table. I've learned how to search for (select) specific data from tables based on simple criteria of my choosing, and even a form of aggregation of the data (count(\*)). I've learned how to insert, update, and delete data from a table, perhaps based on some criteria. I've also learned to truncate data from a table. I think this is very relevant to the database world and am looking forward to more complicated queries and establishing relationships between tables.