COP4710 – Theory and Structure of Databases
Summer 2014

Mike Black

1. Title: Music Curator

A curation system that a songwriter, or a music production company, can use to keep track of lyrics, music, files, songs, record releases and compositional credits.

2. Outline

The database has 7 tables

1. **Record:** Information about the record release, such as the album title, format, length, SKU, etc.
2. **Song:** Tracks about the music, including the actual audio files.
3. **Lyrics:** Stores the words for the song (if any).
4. **Composer:** Tracks writing credit. It separately tracks the music and lyrical credits.
5. **TrackListing:** An association table that links Record to Song. It allows you to maintain separate track-listings of songs for each record, including the track order.
6. **CreditMusic:** An intersection table that links Song to Composer.
7. **CreditLyric:** An intersection table that links Lyric to Composer.

Description

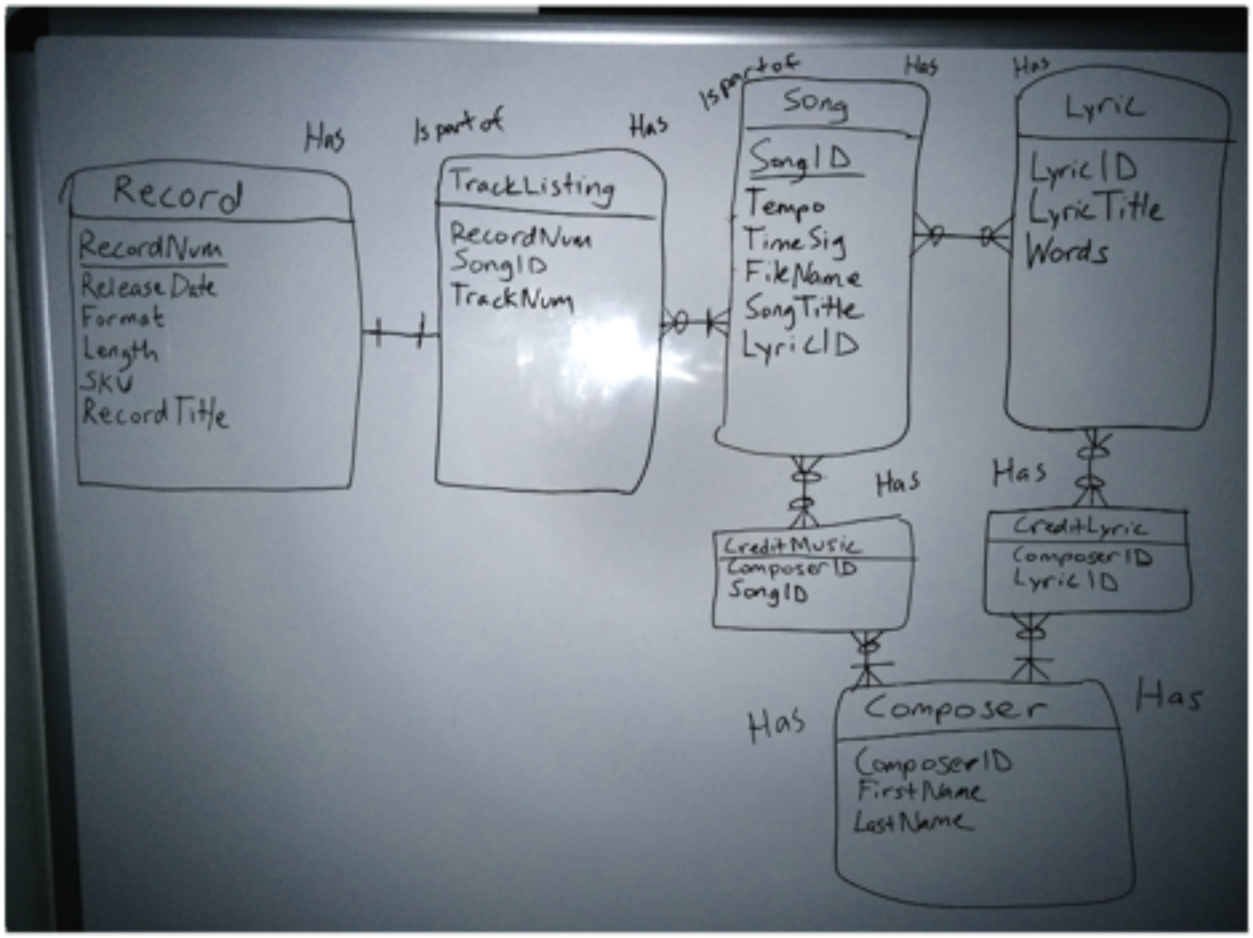
Keeping track of songs, music, lyrics and various revisions is complicated. I would like to create a system to organize this process.

I feel like the end user will need to have the ability to insert values directly into any field in the database. It will need to assume that the user is either a songwriter, or someone who is "music-industry-savvy" enough to understands the fields. In the real world, some of these IDs and revision numbers should be auto-generated to relieve the burden of having to "make up" random IDs and avoid duplication.

Note: I ended up having to scrap the original database and rebuilding it after running into a variety of challenges. Originally, there were four tables. I had to increase this in order to make it possible to store track listings with records, as to be able to maintain a separate list of composers for music and lyrics.

3. ER Diagram

Note: I am completely “re-doing” this because table has changed so much since the last update.



Note: I intentionally left most of the relationships with the Credits and Composer as optional. This is because I felt that it would be annoying if a “solo artist” was constantly forced to type his own name into the credits every time he, or she, added a song or a lyric.

4. Table Schemas

Record (RecordNum, ReleaseDate, Format, Length, SKU, RecordTitle)
Lyric (LyricID, LyricTitle, Words)
Song (SongID, Tempo, TimeSig, FileName, SongTitle, LyricID)
Composer (ComposerID, FirstName, LastName)
TrackListing (RecordNum, SongID, TrackNum)
CreditMusic (ComposerID, SongID)
CreditLyric (ComposerID, LyricID)

Note: It just occurred to me that I do not have primary keys in the association or intersection tables. As I am typing this, I am not sure if that is correct, but the database seems to work fine. In the interest of time, I will just leave it as it.

5. A Note About Normalization

I had a couple of goals for this database that increased the complexity:

- The ability to track lyrics separately from the music
- The ability to track composers

In order to accomplish this, I had to “decompose” my original tables, adding three more:
An association table called “TrackListing” to link Record and Song
Two intersection tables called “CreditMusic” and “CreditLyric” to link the Composer table with both the Song and Lyric tables.

I believe that I this database is normalized at least up to BCNF. There shouldn’t be a problem with atomicity, or redundancy. As far as partial, or transitive, non-key determinants, I am hoping that my aggressive decomposition of the tables as brought me to to 5NF, but I am not certain of this.

6. DDL

```
CREATE TABLE `Record` (
  `RecordNum` int(3) NOT NULL default '0',
  `ReleaseDate` date default NULL,
  `Format` enum('LP','EP','Single','Double LP','Video','Box Set') default NULL,
  `Length` int(11) default NULL,
  `SKU` varchar(20) NOT NULL,
  `RecordTitle` varchar(80) default NULL,
  PRIMARY KEY  (`RecordNum`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `Lyric` (
  `LyricID` int(11) NOT NULL default '0',
  `LyricTitle` varchar(80) default NULL,
  `Words` text,
  PRIMARY KEY  (`LyricID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `Song` (
  `SongID` int(11) NOT NULL default '0',
  `Tempo` float default NULL,
  `TimeSig` char(10) default NULL,
  `FileName` char(80) default NULL,
  `SongTitle` varchar(80) default NULL,
  `LyricID` int(11) default NULL,
  PRIMARY KEY  (`SongID`),
  KEY `LyricID` (`LyricID`),
  CONSTRAINT `Song_ibfk_1` FOREIGN KEY (`LyricID`) REFERENCES `Lyric` (`lyricID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `Composer` (
  `ComposerID` int(7) NOT NULL default '0',
  `FirstName` varchar(20) default NULL,
  `LastName` varchar(20) default NULL,
  PRIMARY KEY  (`ComposerID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `TrackListing` (
  `RecordNum` int(3) default NULL,
  `SongID` int(11) default NULL,
  `TrackNum` int(4) default NULL,
  KEY `RecordNum` (`RecordNum`),
  KEY `SongID` (`SongID`),
  CONSTRAINT `TrackListing_ibfk_2` FOREIGN KEY (`SongID`) REFERENCES `Song`
(`SongID`),
  CONSTRAINT `TrackListing_ibfk_1` FOREIGN KEY (`RecordNum`) REFERENCES `Record`
(`RecordNum`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `CreditMusic` (
  `ComposerID` int(7) default NULL,
  `SongID` int(11) default NULL,
  KEY `ComposerID` (`ComposerID`),
  KEY `SongID` (`SongID`),
  CONSTRAINT `CreditMusic_ibfk_1` FOREIGN KEY (`ComposerID`) REFERENCES
`Composer` (`ComposerID`),
  CONSTRAINT `CreditMusic_ibfk_2` FOREIGN KEY (`SongID`) REFERENCES `Song`
(`SongID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


CREATE TABLE `CreditLyric` (
  `ComposerID` int(7) default NULL,
  `LyricID` int(11) default NULL,
  KEY `ComposerID` (`ComposerID`),
  KEY `LyricID` (`LyricID`),
  CONSTRAINT `CreditLyric_ibfk_1` FOREIGN KEY (`ComposerID`) REFERENCES
`Composer` (`ComposerID`),
  CONSTRAINT `CreditLyric_ibfk_2` FOREIGN KEY (`LyricID`) REFERENCES `Lyric` (`LyricID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

7. Available Functions

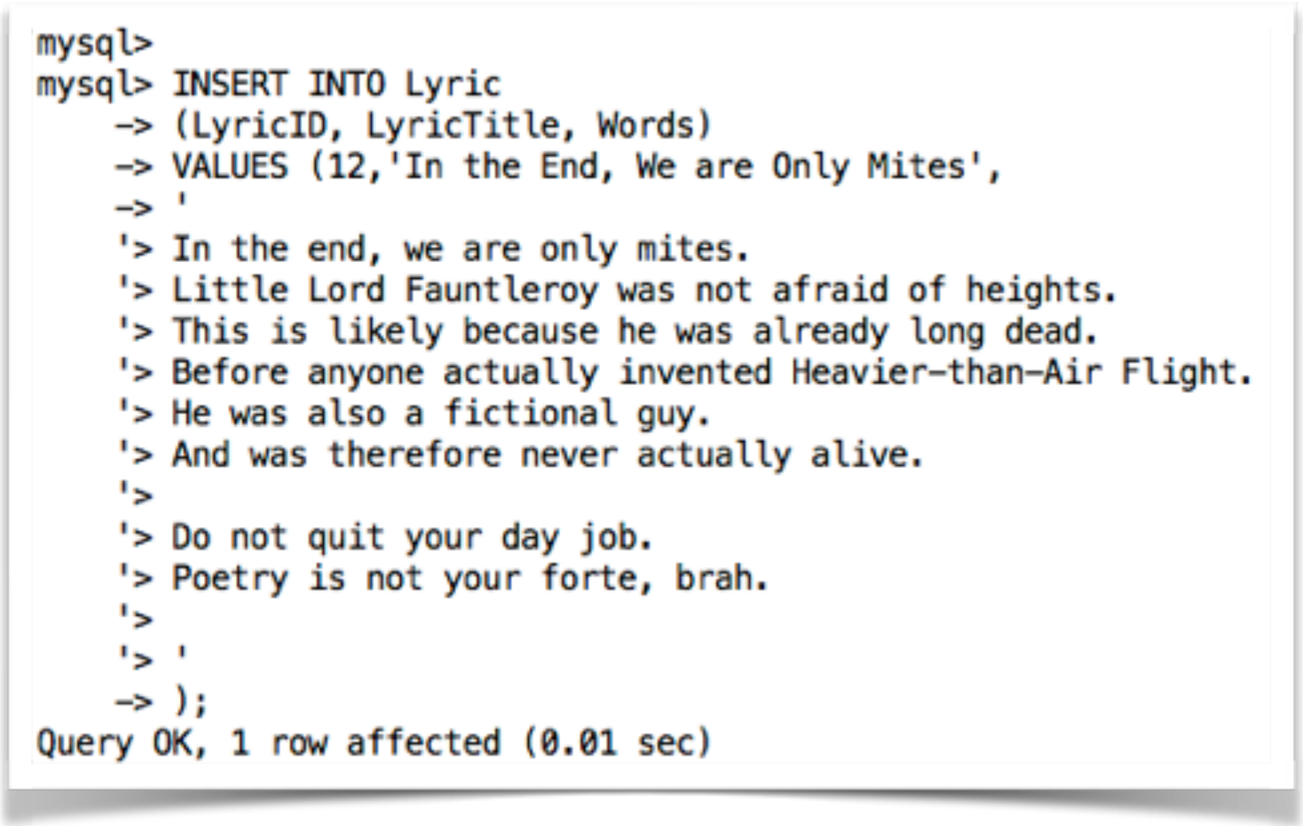
Add a new lyric

A lyric can be added independently of the music. This gives the composer the ability to store lyrics that may not yet have music.

Syntax

INSERT INTO Lyric
(LyricID, LyricTitle, Words)
VALUES (<New ID>,<Lyric Title>,
'
<Actual Lyric/Words>
'
);

Example



```
mysql>
mysql> INSERT INTO Lyric
    -> (LyricID, LyricTitle, Words)
    -> VALUES (12,'In the End, We are Only Mites',
    -> '
    '> In the end, we are only mites.
    '> Little Lord Fauntleroy was not afraid of heights.
    '> This is likely because he was already long dead.
    '> Before anyone actually invented Heavier-than-Air Flight.
    '> He was also a fictional guy.
    '> And was therefore never actually alive.
    '>
    '> Do not quit your day job.
    '> Poetry is not your forte, brah.
    '>
    '> '
    -> );
Query OK, 1 row affected (0.01 sec)
```

Add a new song

Adding a song essentially means entering the following:
- Name of a song
- SongID, and the
- Name of the file that contains the audio

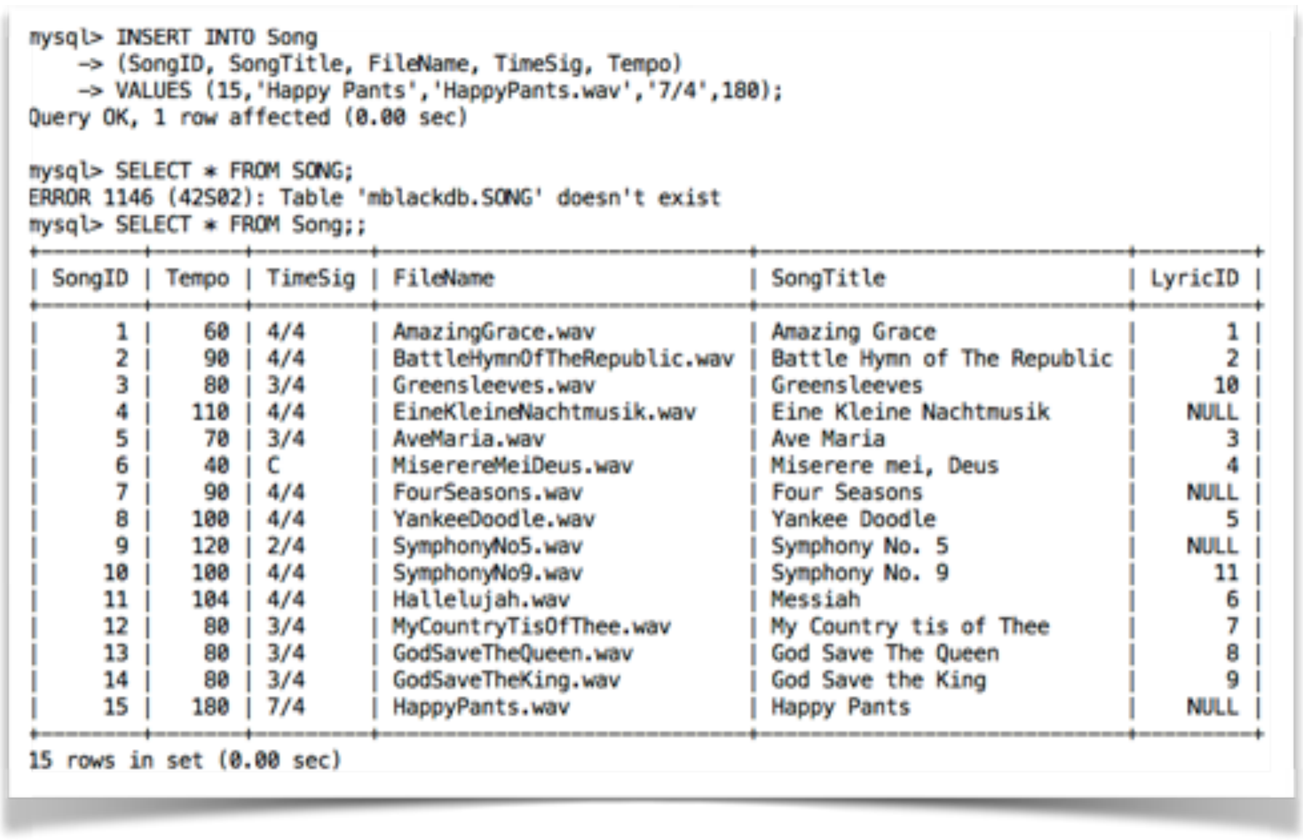
Note: A song may exist without lyrics (an instrumental).

Syntax

INSERT INTO Song
(SongID, SongTitle, FileName, TimeSig, Tempo)
VALUES (<New ID>,<Song Name>,<File Name>,<Time Signature>,<Tempo>,<Lyric ID (if present)>);

Note: If a lyric exists for this song, add the LyricID while inserting the row to link the song to the lyric.

Example

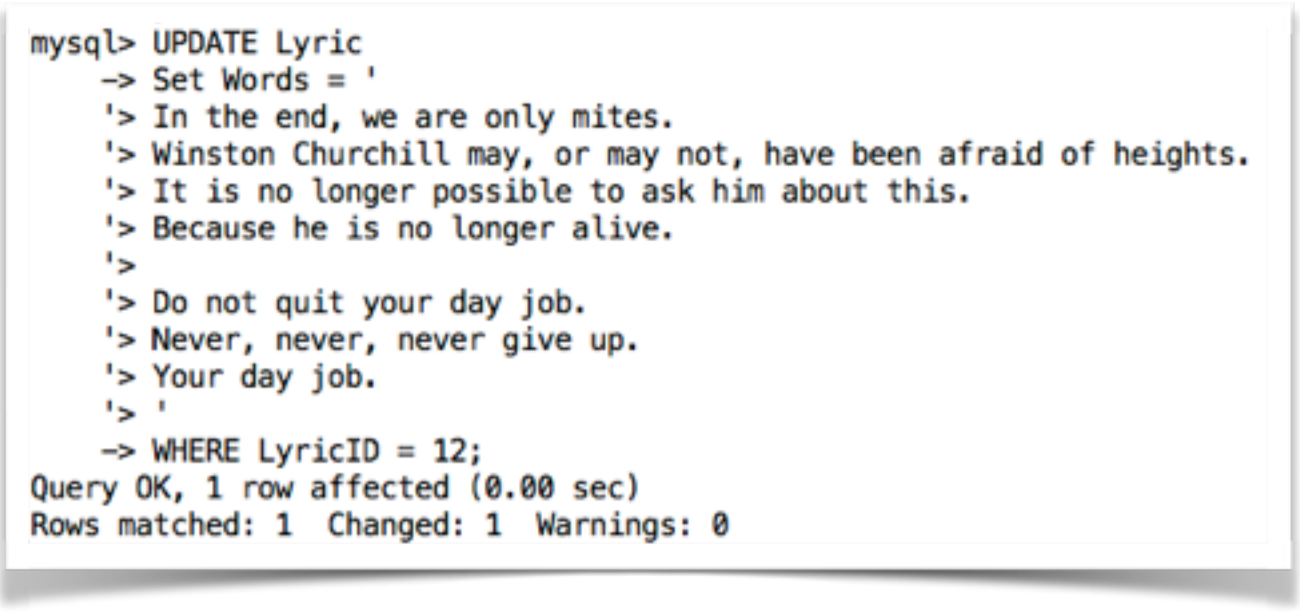


```
mysql> INSERT INTO Song
    -> (SongID, SongTitle, FileName, TimeSig, Tempo)
    -> VALUES (15,'Happy Pants','HappyPants.wav','7/4',180);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM SONG;
ERROR 1146 (42S02): Table 'mblackdb.SONG' doesn't exist
mysql> SELECT * FROM Song;;
+--------+-------+---------+-----------------------------+----------------------------+---------+
| SongID | Tempo | TimeSig | FileName                    | SongTitle                  | LyricID |
+--------+-------+---------+-----------------------------+----------------------------+---------+
|      1 |    60 | 4/4     | AmazingGrace.wav            | Amazing Grace              |       1 |
|      2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav | Battle Hymn of The Republic |      2 |
|      3 |    80 | 3/4     | Greensleeves.wav            | Greensleeves               |      10 |
|      4 |   110 | 4/4     | EineKleineNachtmusik.wav    | Eine Kleine Nachtmusik     |    NULL |
|      5 |    70 | 3/4     | AveMaria.wav                | Ave Maria                  |       3 |
|      6 |    40 | C       | MiserereMeiDeus.wav         | Miserere mei, Deus         |       4 |
|      7 |    90 | 4/4     | FourSeasons.wav             | Four Seasons               |    NULL |
|      8 |   100 | 4/4     | YankeeDoodle.wav            | Yankee Doodle              |       5 |
|      9 |   120 | 2/4     | SymphonyNo5.wav             | Symphony No. 5             |    NULL |
|     10 |   100 | 4/4     | SymphonyNo9.wav             | Symphony No. 9             |      11 |
|     11 |   104 | 4/4     | Hallelujah.wav              | Messiah                    |       6 |
|     12 |    80 | 3/4     | MyCountryTisOfThee.wav      | My Country tis of Thee     |       7 |
|     13 |    80 | 3/4     | GodSaveTheQueen.wav         | God Save The Queen         |       8 |
|     14 |    80 | 3/4     | GodSaveTheKing.wav          | God Save the King          |       9 |
|     15 |   180 | 7/4     | HappyPants.wav              | Happy Pants                |    NULL |
+--------+-------+---------+-----------------------------+----------------------------+---------+
15 rows in set (0.00 sec)
```

Syntax

UPDATE Lyric
Set Words = '
<Updated Lyric/Words>
'
WHERE LyricID = <Lyric ID>;

Example



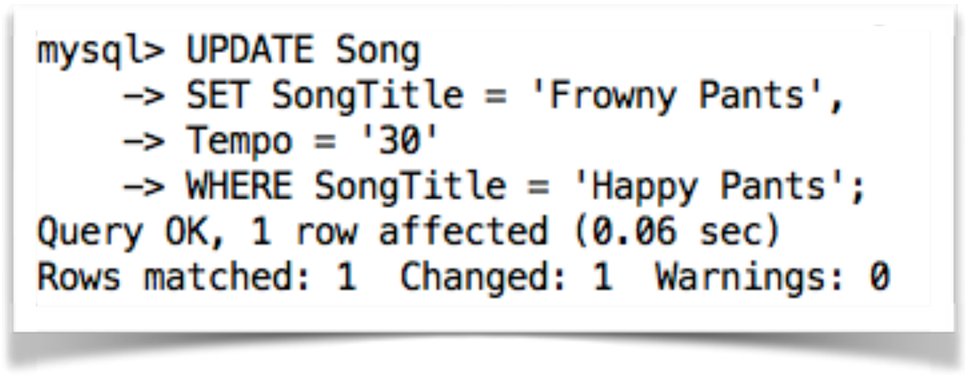
```
mysql> UPDATE Lyric
    -> Set Words = '
    '> In the end, we are only mites.
    '> Winston Churchill may, or may not, have been afraid of heights.
    '> It is no longer possible to ask him about this.
    '> Because he is no longer alive.
    '>
    '> Do not quit your day job.
    '> Never, never, never give up.
    '> Your day job.
    '> '
    -> WHERE LyricID = 12;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Update a song

Syntax

UPDATE Song
SET SongTitle = <New Title>,
Tempo = <New Tempo>,
WHERE SongTitle = <Old Title>;

Example



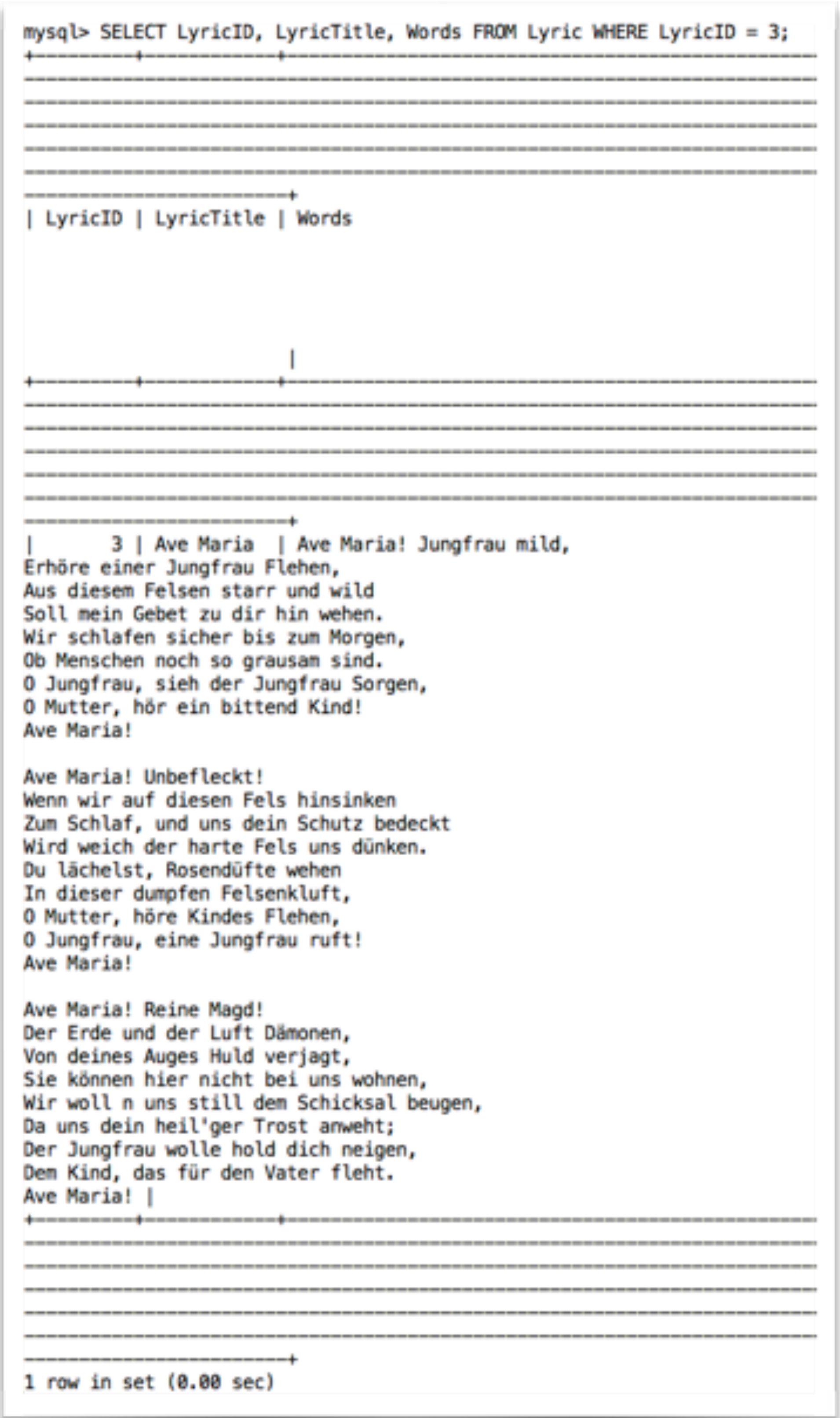
```
mysql> UPDATE Song
    -> SET SongTitle = 'Frowny Pants',
    -> Tempo = '30'
    -> WHERE SongTitle = 'Happy Pants';
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Syntax

```
SELECT LyricID, LyricTitle, Words FROM Lyric
WHERE LyricID = <Lyric ID>;
```

Note: Alternatively, the Lyric Title may be used.

Example (screenshot appears on next page)



```
mysql> SELECT LyricID, LyricTitle, Words FROM Lyric WHERE LyricID = 3;
+---------+------------+-------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------+
| LyricID | LyricTitle | Words





                        |
+---------+------------+-------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------+
|       3 | Ave Maria  | Ave Maria! Jungfrau mild,
Erhöre einer Jungfrau Flehen,
Aus diesem Felsen starr und wild
Soll mein Gebet zu dir hin wehen.
Wir schlafen sicher bis zum Morgen,
Ob Menschen noch so grausam sind.
O Jungfrau, sieh der Jungfrau Sorgen,
O Mutter, hör ein bittend Kind!
Ave Maria!

Ave Maria! Unbefleckt!
Wenn wir auf diesen Fels hinsinken
Zum Schlaf, und uns dein Schutz bedeckt
Wird weich der harte Fels uns dünken.
Du lächelst, Rosendüfte wehen
In dieser dumpfen Felsenkluft,
O Mutter, höre Kindes Flehen,
O Jungfrau, eine Jungfrau ruft!
Ave Maria!

Ave Maria! Reine Magd!
Der Erde und der Luft Dämonen,
Von deines Auges Huld verjagt,
Sie können hier nicht bei uns wohnen,
Wir woll n uns still dem Schicksal beugen,
Da uns dein heil'ger Trost anweht;
Der Jungfrau wolle hold dich neigen,
Dem Kind, das für den Vater fleht.
Ave Maria! |
+---------+------------+-------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------+
1 row in set (0.00 sec)
```

Release a new record

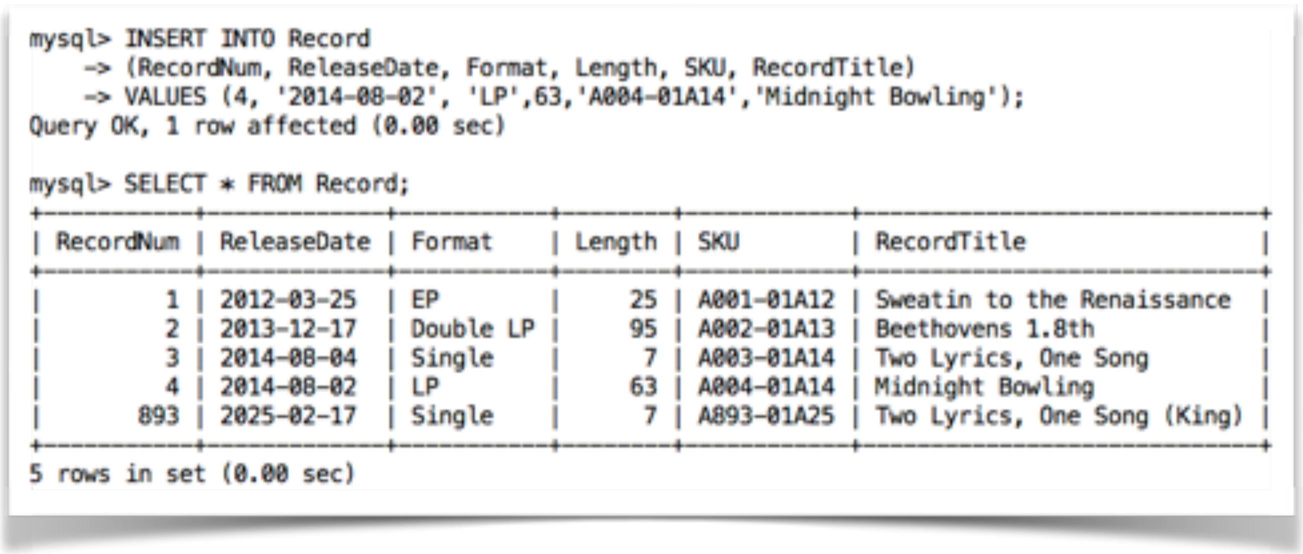
Any of these formats are available:

- **LP (Long Play):** This is a typical album length; the maximum size is usually between 70-80 minutes.
- **Double LP:** A "double-album."
- **Single:** Short recording; it usually has between 1-3 tracks.
- **EP (Extended Play):** This is a length that is in-between an LP and a single. An EP typically contains about 4 songs.
- **Video:** A video format release. This is typically a recording of a live performance, a collection of music videos or a documentary.
- **Box Set:** A large collection of records that is typically sold in a souvenir-like container. A box set will often span an artist's entire career.

Syntax

INSERT INTO Record
(RecordNum, ReleaseDate, Format, Length, SKU, RecordTitle)
VALUES (<Record Number>,<Date>,<Format><Length In Minutes><SKU>,<Title>');

Example



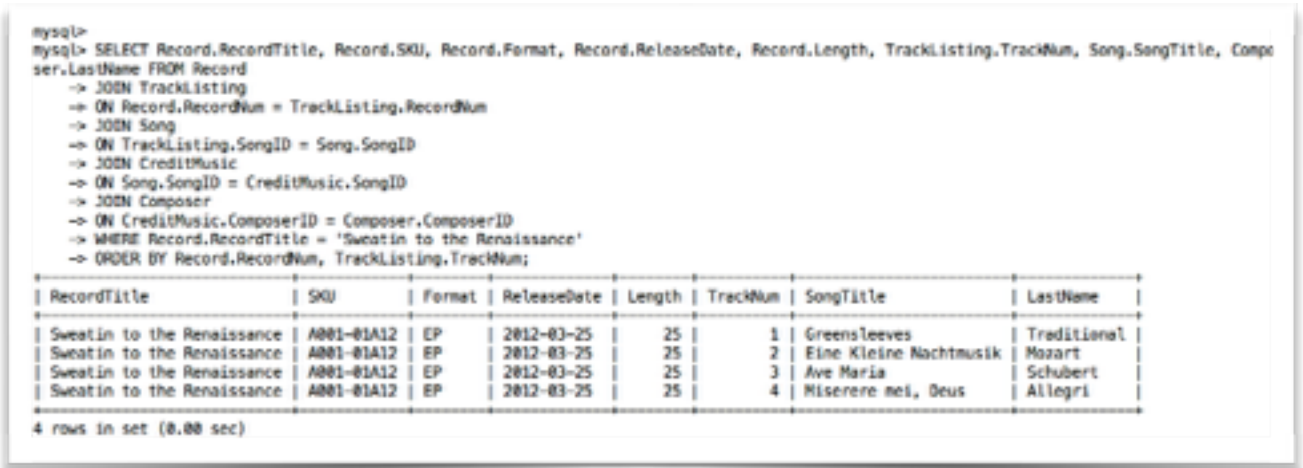
```
mysql> INSERT INTO Record
    -> (RecordNum, ReleaseDate, Format, Length, SKU, RecordTitle)
    -> VALUES (4, '2014-08-02', 'LP',63,'A004-01A14','Midnight Bowling');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Record;
+-----------+-------------+-----------+--------+------------+------------------------------+
| RecordNum | ReleaseDate | Format    | Length | SKU        | RecordTitle                  |
+-----------+-------------+-----------+--------+------------+------------------------------+
|         1 | 2012-03-25  | EP        |     25 | A001-01A12 | Sweatin to the Renaissance   |
|         2 | 2013-12-17  | Double LP |     95 | A002-01A13 | Beethovens 1.8th             |
|         3 | 2014-08-04  | Single    |      7 | A003-01A14 | Two Lyrics, One Song         |
|         4 | 2014-08-02  | LP        |     63 | A004-01A14 | Midnight Bowling             |
|       893 | 2025-02-17  | Single    |      7 | A893-01A25 | Two Lyrics, One Song (King)  |
+-----------+-------------+-----------+--------+------------+------------------------------+
5 rows in set (0.00 sec)
```

Look-up a record by name

Syntax

SELECT Record.RecordTitle, Record.SKU, Record.Format, Record.ReleaseDate, Record.Length, TrackListing.TrackNum, Song.SongTitle, Composer.LastName FROM Record
JOIN TrackListing
ON Record.RecordNum = TrackListing.RecordNum
JOIN Song
ON TrackListing.SongID = Song.SongID
JOIN CreditMusic
ON Song.SongID = CreditMusic.SongID
JOIN Composer
ON CreditMusic.ComposerID = Composer.ComposerID
WHERE Record.RecordTitle = **<ENTER RECORD NAME HERE>**
ORDER BY Record.RecordNum, TrackListing.TrackNum;

Example



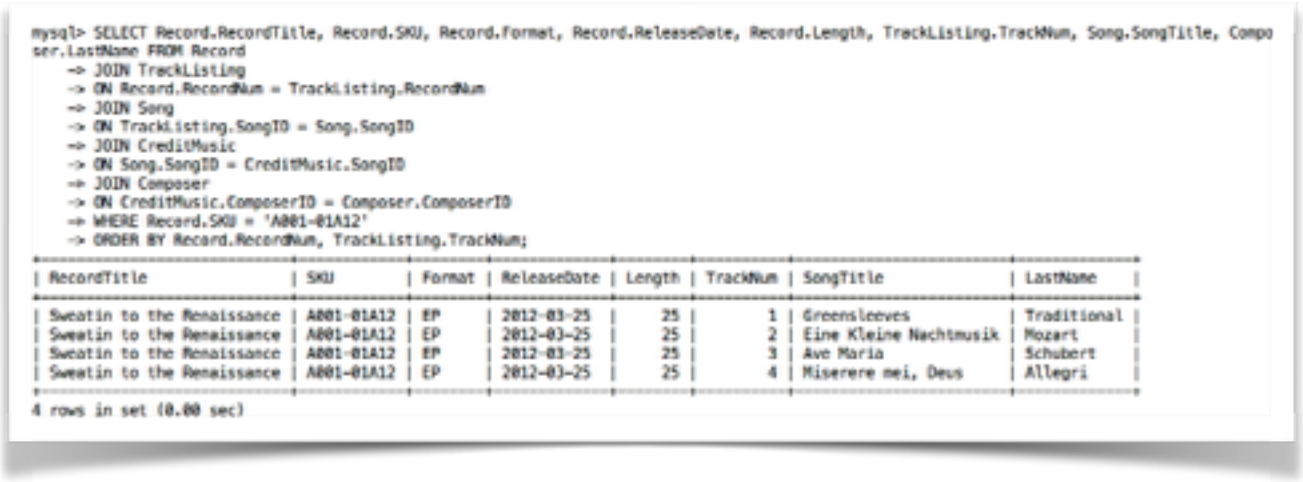
```
mysql>
mysql> SELECT Record.RecordTitle, Record.SKU, Record.Format, Record.ReleaseDate, Record.Length, TrackListing.TrackNum, Song.SongTitle, Compo
ser.LastName FROM Record
    -> JOIN TrackListing
    -> ON Record.RecordNum = TrackListing.RecordNum
    -> JOIN Song
    -> ON TrackListing.SongID = Song.SongID
    -> JOIN CreditMusic
    -> ON Song.SongID = CreditMusic.SongID
    -> JOIN Composer
    -> ON CreditMusic.ComposerID = Composer.ComposerID
    -> WHERE Record.RecordTitle = 'Sweatin to the Renaissance'
    -> ORDER BY Record.RecordNum, TrackListing.TrackNum;
+----------------------------+------------+--------+-------------+--------+----------+-----------------------+-------------+
| RecordTitle                | SKU        | Format | ReleaseDate | Length | TrackNum | SongTitle             | LastName    |
+----------------------------+------------+--------+-------------+--------+----------+-----------------------+-------------+
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        1 | Greensleeves          | Traditional |
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        2 | Eine Kleine Nachtmusik | Mozart     |
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        3 | Ave Maria             | Schubert    |
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        4 | Miserere mei, Deus    | Allegri     |
+----------------------------+------------+--------+-------------+--------+----------+-----------------------+-------------+
4 rows in set (0.00 sec)
```

Look-up a record by SKU

Syntax

SELECT Record.RecordTitle, Record.SKU, Record.Format, Record.ReleaseDate, Record.Length, TrackListing.TrackNum, Song.SongTitle, Composer.LastName FROM Record
JOIN TrackListing
ON Record.RecordNum = TrackListing.RecordNum
JOIN Song
ON TrackListing.SongID = Song.SongID
JOIN CreditMusic
ON Song.SongID = CreditMusic.SongID
JOIN Composer
ON CreditMusic.ComposerID = Composer.ComposerID
WHERE Record.SKU = **<ENTER SKU HERE>**
ORDER BY Record.RecordNum, TrackListing.TrackNum;

Example



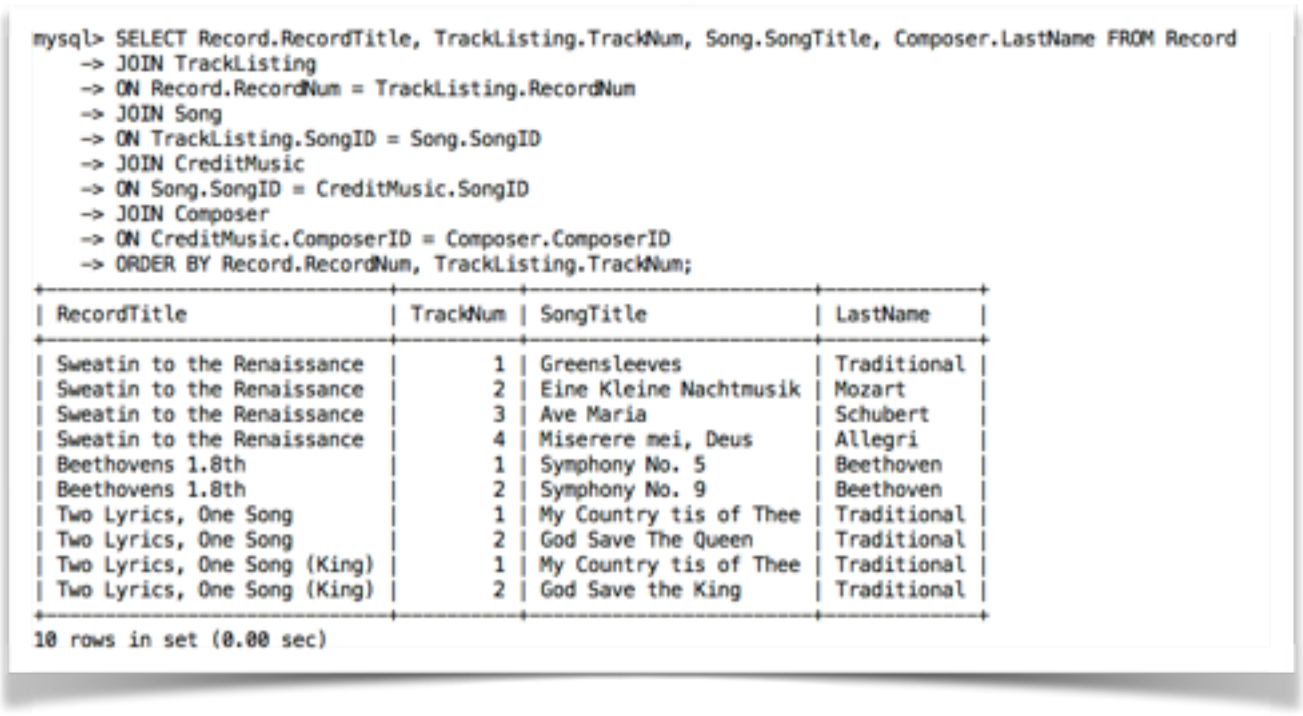
```
mysql> SELECT Record.RecordTitle, Record.SKU, Record.Format, Record.ReleaseDate, Record.Length, TrackListing.TrackNum, Song.SongTitle, Compo
ser.LastName FROM Record
    -> JOIN TrackListing
    -> ON Record.RecordNum = TrackListing.RecordNum
    -> JOIN Song
    -> ON TrackListing.SongID = Song.SongID
    -> JOIN CreditMusic
    -> ON Song.SongID = CreditMusic.SongID
    -> JOIN Composer
    -> ON CreditMusic.ComposerID = Composer.ComposerID
    -> WHERE Record.SKU = 'A001-01A12'
    -> ORDER BY Record.RecordNum, TrackListing.TrackNum;
+---------------------------+------------+--------+-------------+--------+----------+-----------------------+-------------+
| RecordTitle               | SKU        | Format | ReleaseDate | Length | TrackNum | SongTitle             | LastName    |
+---------------------------+------------+--------+-------------+--------+----------+-----------------------+-------------+
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        1 | Greensleeves          | Traditional |
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        2 | Eine Kleine Nachtmusik | Mozart      |
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        3 | Ave Maria             | Schubert    |
| Sweatin to the Renaissance | A001-01A12 | EP     | 2012-03-25  |     25 |        4 | Miserere mei, Deus    | Allegri     |
+---------------------------+------------+--------+-------------+--------+----------+-----------------------+-------------+
4 rows in set (0.00 sec)
```

Complete Track-Listing Report

Syntax

SELECT Record.RecordTitle, TrackListing.TrackNum, Song.SongTitle, Composer.LastName
FROM Record
JOIN TrackListing
ON Record.RecordNum = TrackListing.RecordNum
JOIN Song
ON TrackListing.SongID = Song.SongID
JOIN CreditMusic
ON Song.SongID = CreditMusic.SongID
JOIN Composer
ON CreditMusic.ComposerID = Composer.ComposerID
ORDER BY Record.RecordNum, TrackListing.TrackNum;

Example

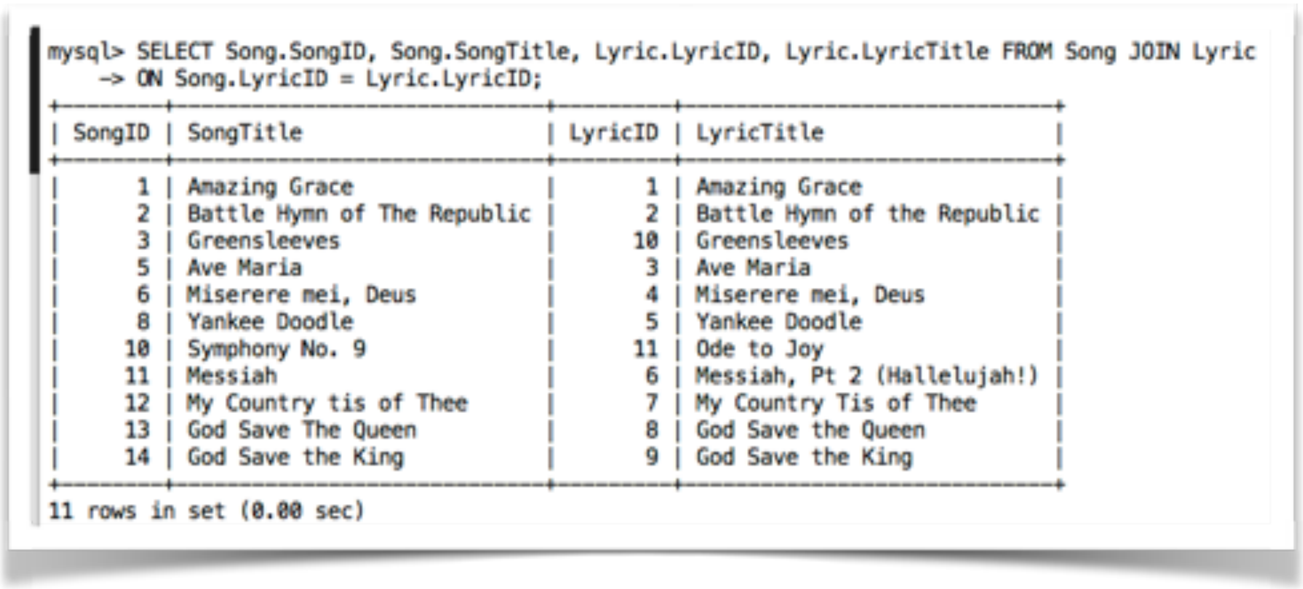


```
mysql> SELECT Record.RecordTitle, TrackListing.TrackNum, Song.SongTitle, Composer.LastName FROM Record
    -> JOIN TrackListing
    -> ON Record.RecordNum = TrackListing.RecordNum
    -> JOIN Song
    -> ON TrackListing.SongID = Song.SongID
    -> JOIN CreditMusic
    -> ON Song.SongID = CreditMusic.SongID
    -> JOIN Composer
    -> ON CreditMusic.ComposerID = Composer.ComposerID
    -> ORDER BY Record.RecordNum, TrackListing.TrackNum;
+------------------------------+----------+------------------------+-------------+
| RecordTitle                  | TrackNum | SongTitle              | LastName    |
+------------------------------+----------+------------------------+-------------+
| Sweatin to the Renaissance   |        1 | Greensleeves           | Traditional |
| Sweatin to the Renaissance   |        2 | Eine Kleine Nachtmusik | Mozart      |
| Sweatin to the Renaissance   |        3 | Ave Maria              | Schubert    |
| Sweatin to the Renaissance   |        4 | Miserere mei, Deus     | Allegri     |
| Beethovens 1.8th             |        1 | Symphony No. 5         | Beethoven   |
| Beethovens 1.8th             |        2 | Symphony No. 9         | Beethoven   |
| Two Lyrics, One Song         |        1 | My Country tis of Thee | Traditional |
| Two Lyrics, One Song         |        2 | God Save The Queen     | Traditional |
| Two Lyrics, One Song (King)  |        1 | My Country tis of Thee | Traditional |
| Two Lyrics, One Song (King)  |        2 | God Save the King      | Traditional |
+------------------------------+----------+------------------------+-------------+
10 rows in set (0.00 sec)
```

Syntax

SELECT Song.SongID, Song.SongTitle, Lyric.LyricID, Lyric.LyricTitle FROM Song
JOIN Lyric
ON Song.LyricID = Lyric.LyricID;

Example

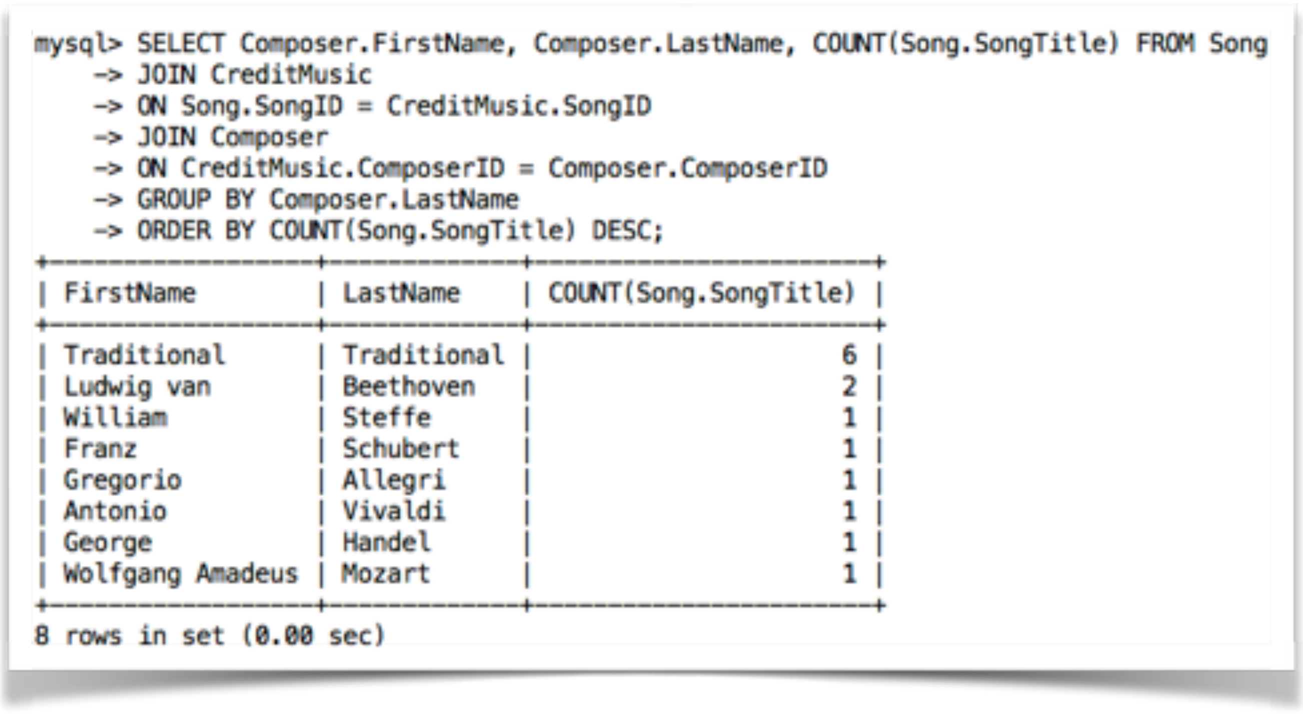


```
mysql> SELECT Song.SongID, Song.SongTitle, Lyric.LyricID, Lyric.LyricTitle FROM Song JOIN Lyric
    -> ON Song.LyricID = Lyric.LyricID;
+--------+------------------------------+---------+------------------------------+
| SongID | SongTitle                    | LyricID | LyricTitle                   |
+--------+------------------------------+---------+------------------------------+
|      1 | Amazing Grace                |       1 | Amazing Grace                |
|      2 | Battle Hymn of The Republic  |       2 | Battle Hymn of the Republic  |
|      3 | Greensleeves                 |      10 | Greensleeves                 |
|      5 | Ave Maria                    |       3 | Ave Maria                    |
|      6 | Miserere mei, Deus           |       4 | Miserere mei, Deus           |
|      8 | Yankee Doodle                |       5 | Yankee Doodle                |
|     10 | Symphony No. 9               |      11 | Ode to Joy                   |
|     11 | Messiah                      |       6 | Messiah, Pt 2 (Hallelujah!)  |
|     12 | My Country tis of Thee       |       7 | My Country Tis of Thee       |
|     13 | God Save The Queen           |       8 | God Save the Queen           |
|     14 | God Save the King            |       9 | God Save the King            |
+--------+------------------------------+---------+------------------------------+
11 rows in set (0.00 sec)
```

Syntax

SELECT Composer.FirstName, Composer.LastName, COUNT(Song.SongTitle) FROM Song
JOIN CreditMusic
ON Song.SongID = CreditMusic.SongID
JOIN Composer
ON CreditMusic.ComposerID = Composer.ComposerID
GROUP BY Composer.LastName
ORDER BY COUNT(Song.SongTitle) DESC;

Example

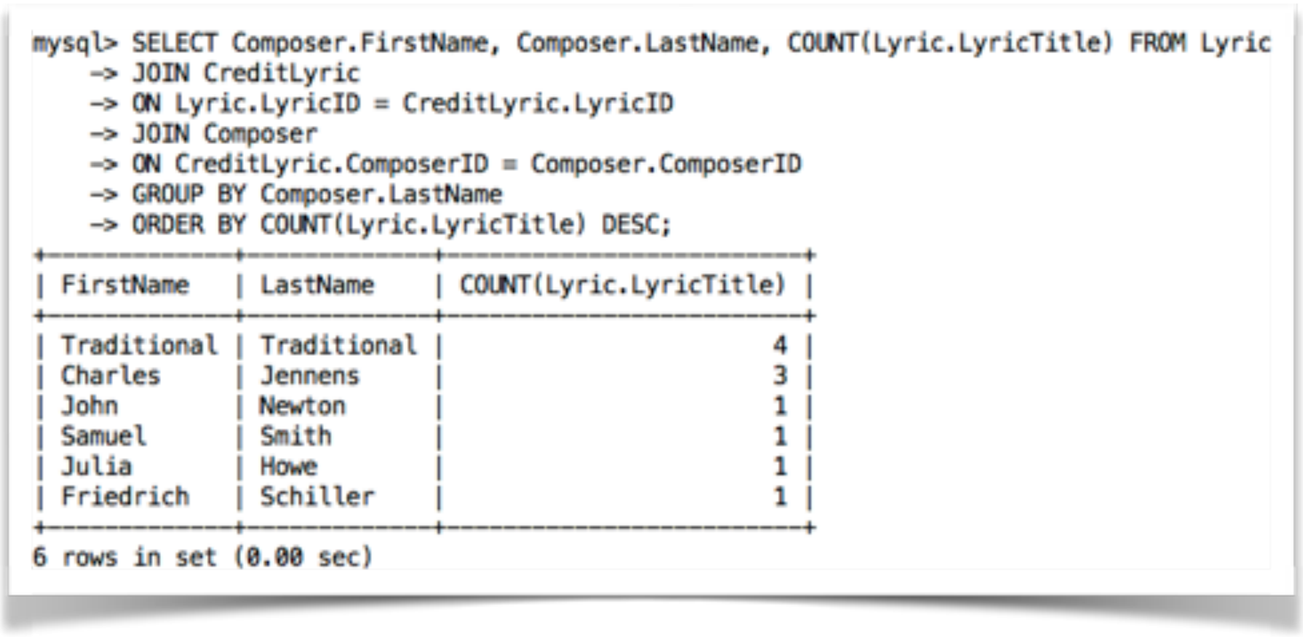


```
mysql> SELECT Composer.FirstName, Composer.LastName, COUNT(Song.SongTitle) FROM Song
    -> JOIN CreditMusic
    -> ON Song.SongID = CreditMusic.SongID
    -> JOIN Composer
    -> ON CreditMusic.ComposerID = Composer.ComposerID
    -> GROUP BY Composer.LastName
    -> ORDER BY COUNT(Song.SongTitle) DESC;
+-----------------+-------------+----------------------+
| FirstName       | LastName    | COUNT(Song.SongTitle) |
+-----------------+-------------+----------------------+
| Traditional     | Traditional |                    6 |
| Ludwig van      | Beethoven   |                    2 |
| William         | Steffe      |                    1 |
| Franz           | Schubert    |                    1 |
| Gregorio        | Allegri     |                    1 |
| Antonio         | Vivaldi     |                    1 |
| George          | Handel      |                    1 |
| Wolfgang Amadeus | Mozart     |                    1 |
+-----------------+-------------+----------------------+
8 rows in set (0.00 sec)
```

Syntax

SELECT Composer.FirstName, Composer.LastName, COUNT(Lyric.LyricTitle) FROM Lyric
JOIN CreditLyric
ON Lyric.LyricID = CreditLyric.LyricID
JOIN Composer
ON CreditLyric.ComposerID = Composer.ComposerID
GROUP BY Composer.LastName
ORDER BY COUNT(Lyric.LyricTitle) DESC;

Example

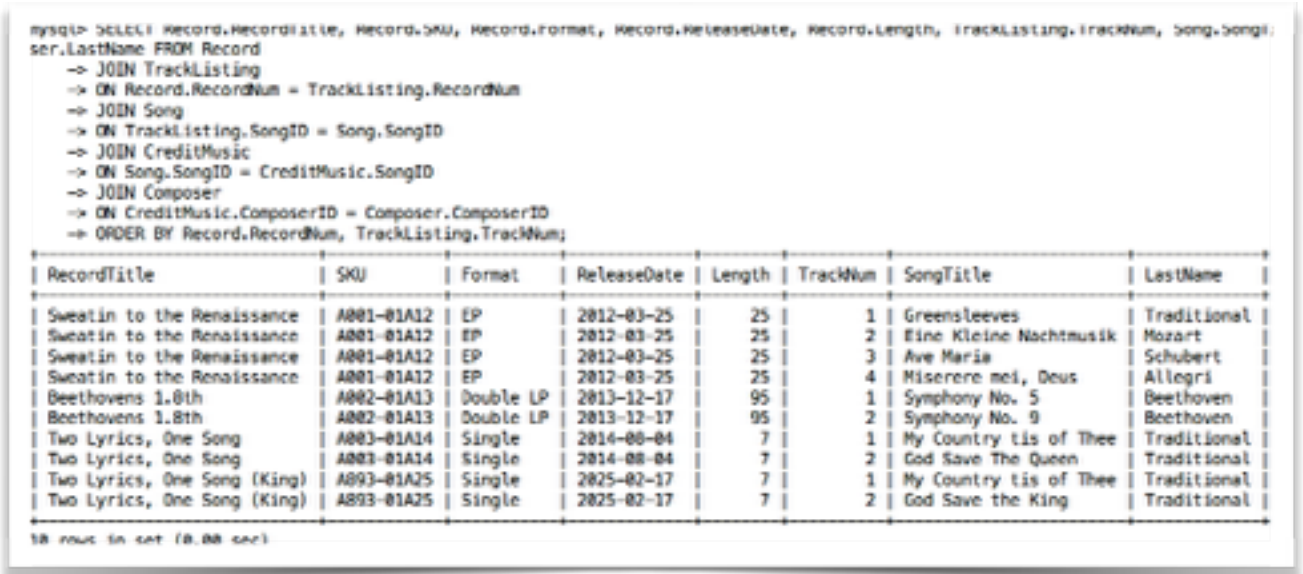


```
mysql> SELECT Composer.FirstName, Composer.LastName, COUNT(Lyric.LyricTitle) FROM Lyric
    -> JOIN CreditLyric
    -> ON Lyric.LyricID = CreditLyric.LyricID
    -> JOIN Composer
    -> ON CreditLyric.ComposerID = Composer.ComposerID
    -> GROUP BY Composer.LastName
    -> ORDER BY COUNT(Lyric.LyricTitle) DESC;
+-------------+-------------+-------------------------+
| FirstName   | LastName    | COUNT(Lyric.LyricTitle) |
+-------------+-------------+-------------------------+
| Traditional | Traditional |                       4 |
| Charles     | Jennens     |                       3 |
| John        | Newton      |                       1 |
| Samuel      | Smith       |                       1 |
| Julia       | Howe        |                       1 |
| Friedrich   | Schiller    |                       1 |
+-------------+-------------+-------------------------+
6 rows in set (0.00 sec)
```

Syntax

SELECT Record.RecordTitle, Record.SKU, Record.Format, Record.ReleaseDate, Record.Length, TrackListing.TrackNum, Song.SongTitle, Composer.LastName FROM Record
JOIN TrackListing
ON Record.RecordNum = TrackListing.RecordNum
JOIN Song
ON TrackListing.SongID = Song.SongID
JOIN CreditMusic
ON Song.SongID = CreditMusic.SongID
JOIN Composer
ON CreditMusic.ComposerID = Composer.ComposerID
ORDER BY Record.RecordNum, TrackListing.TrackNum;

Example



```
mysql> SELECT Record.RecordTitle, Record.SKU, Record.Format, Record.ReleaseDate, Record.Length, TrackListing.TrackNum, Song.SongT
ser.LastName FROM Record
    -> JOIN TrackListing
    -> ON Record.RecordNum = TrackListing.RecordNum
    -> JOIN Song
    -> ON TrackListing.SongID = Song.SongID
    -> JOIN CreditMusic
    -> ON Song.SongID = CreditMusic.SongID
    -> JOIN Composer
    -> ON CreditMusic.ComposerID = Composer.ComposerID
    -> ORDER BY Record.RecordNum, TrackListing.TrackNum;
+----------------------------+-------------+-----------+-------------+--------+----------+-------------------------+-------------+
| RecordTitle                | SKU         | Format    | ReleaseDate | Length | TrackNum | SongTitle               | LastName    |
+----------------------------+-------------+-----------+-------------+--------+----------+-------------------------+-------------+
| Sweatin to the Renaissance | A001-01A12  | EP        | 2012-03-25  |     25 |        1 | Greensleeves            | Traditional |
| Sweatin to the Renaissance | A001-01A12  | EP        | 2012-03-25  |     25 |        2 | Eine Kleine Nachtmusik  | Mozart      |
| Sweatin to the Renaissance | A001-01A12  | EP        | 2012-03-25  |     25 |        3 | Ave Maria               | Schubert    |
| Sweatin to the Renaissance | A001-01A12  | EP        | 2012-03-25  |     25 |        4 | Miserere mei, Deus      | Allegri     |
| Beethovens 1.8th           | A002-01A13  | Double LP | 2013-12-17  |     95 |        1 | Symphony No. 5          | Beethoven   |
| Beethovens 1.8th           | A002-01A13  | Double LP | 2013-12-17  |     95 |        2 | Symphony No. 9          | Beethoven   |
| Two Lyrics, One Song       | A003-01A14  | Single    | 2014-08-04  |      7 |        1 | My Country tis of Thee  | Traditional |
| Two Lyrics, One Song       | A003-01A14  | Single    | 2014-08-04  |      7 |        2 | God Save The Queen      | Traditional |
| Two Lyrics, One Song (King)| A893-01A25  | Single    | 2025-02-17  |      7 |        1 | My Country tis of Thee  | Traditional |
| Two Lyrics, One Song (King)| A893-01A25  | Single    | 2025-02-17  |      7 |        2 | God Save the King       | Traditional |
+----------------------------+-------------+-----------+-------------+--------+----------+-------------------------+-------------+
10 rows in set (0.00 sec)
```

8. Project Log

Note: This is a transcript of the process that I followed while building and troubleshooting this database. I am including this in case there is any question as to decisions that I made, or how I did something. Feel free to skip it.

Begin Log

DDL:

```
CREATE TABLE Music (
musicID INT,
tempo FLOAT,
timeSig CHAR(10),
filename CHAR(80) NOT NULL,
revisionNum FLOAT,

PRIMARY KEY (musicID)
);

//Note: When creating the table, I kept getting syntax errors. I eventually
had to change the "key" attribute, because it is a reserved word. I decided
to omit "key" since most poplar music tends to have multiple keys anyway.


CREATE TABLE Lyric (
lyricID INT,
lyricTitle CHAR(80) NOT NULL,
words TEXT,


PRIMARY KEY (lyricID)
);

// Note: I have merged verse1, verse2, chorus and bridge into "words." Breaking
it up into separate units seems to be making things needlessly complex. I left
it as a TEXT data type, though I will see if there is a better way to do this.



CREATE TABLE Song (
musicID INT,
lyricID INT,
songTitle CHAR(80) NOT NULL,
composerFirstName CHAR(20) NOT NULL,
composerLastName CHAR(20) NOT NULL,
PRIMARY KEY (musicID, lyricID),
FOREIGN KEY (musicID) references Music(musicID),
FOREIGN KEY (lyricID) references Lyric(lyricID)
);
```

// Note: I decided to make words, and the composer names NOT NULL because I am assuming that 1) sometimes, you just have a title in mind without words yet, and 2) this could be used by individual songwriters, who may not want to constantly have to type their own name for every song.

```
CREATE TABLE Record (
releaseID INT,
songTitle CHAR(80) NOT NULL,
releaseYear INT,
format ENUM('LP','EP','Single','Video','BoxSet'),
length INT,

PRIMARY KEY (releaseID),
FOREIGN KEY (songTitle) references Song(songTitle)
);
```

// Note: Apparently, the word "Release" is reserved too! I will change this to "Record." I am avoiding "Album," since I want a word that is inclusive of singles and EPs (Extended Plays).

DML:

```
INSERT INTO Lyric (lyricID, LyricTitle, words)
VALUES (1,'Amazing Grace','Amazing grace! how sweet the sound
That sav\'d a wretch like me!
I once was lost, but now am found,
Was blind, but now I see.');
```

// Note: I had to add an escape character before the apostrophe in "sav'd"

```
INSERT INTO Lyric (lyricID, LyricTitle, words)
VALUES(2,'Battle Hymn of the Republic','Mine eyes have seen the glory of the
coming of the Lord.
He is trampling out the vintage where the grapes of wrath are stored.
He hath loosed the fateful lightning of His terrible swift sword.
His truth is marching on.
Glory, glory, hallelujah!
Glory, glory, hallelujah!
Glory, glory, hallelujah!
His truth is marching on.');


mysql> select * from Lyric
-> ;
+---------+----------------------------
+-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------+
| lyricID | lyricTitle | words |
+---------+----------------------------
+-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------+
| 1 | Amazing Grace | Amazing grace! how sweet the sound
That sav'd a wretch like me!
I once was lost, but now am found,
Was blind, but now I see. |
| 2 | Battle Hymn of the Republic | Mine eyes have seen the glory of the
coming of the Lord.
He is trampling out the vintage where the grapes of wrath are stored.
He hath loosed the fateful lightning of His terrible swift sword.
His truth is marching on.
Glory, glory, hallelujah!
Glory, glory, hallelujah!
Glory, glory, hallelujah!
His truth is marching on. |
+---------+----------------------------
+-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

// Note: I am not sure of the meaning of all of the “----+---“ stuff.

// I am allowing NULLs for many things because I believe they may not
Immediately be defined. This will presumably not be a tremendously large database, so the presence of some NULLS shouldn't be a major problem.

// In several places, I have renamed the ID and Title attributes to make them unique without relying on prefixes.

// A key issue that I need to look into is how to fit multiple instances of an object into a single row. For example, what happens if there is more than one songwriter? Also, I am still not certain of how I will address revision numbers. This would theoretically allow a songwriter to store multiple revision of the same lyric or music under the same ID, but I am not sure if that's even allowed.

// I wanted to rename the ReleaseID to SKU, but MySQL kept returning syntax errors. It is likely that it does not want me to rename a primary key.

// As a work-around, I added a new column and then altered the database to set the new column as a primary key.

```
mysql> ALTER TABLE Record ADD SKU VARCHAR(20);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Record MODIFY COLUMN SKU VARCHAR(20) NOT NULL;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Record
    -> ADD PRIMARY KEY (SKU);
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

// Renamed releaseID to ReleaseNum and reduced the INT size to 3:

```
mysql> ALTER TABLE `Record` CHANGE COLUMN `releaseID` `ReleaseNum` INT(3) NOT NULL;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

// Renamed ReleaseNum to RecordNum

```
mysql> ALTER TABLE Record
    -> CHANGE COLUMN ReleaseNum RecordNum INT(3);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

// Renamed songTitle to TrackTitle

```
mysql> ALTER TABLE Record CHANGE COLUMN songTitle TrackTitle VARCHAR(20);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

// changed releaseYear to ReleaseDate and changed the data type to DATE

```
mysql> ALTER TABLE Record CHANGE COLUMN releaseYear ReleaseDate DATE;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

// Renamed table Song to Track (more inclusive of instrumentals, or non-pop music.

```
mysql> RENAME TABLE Song TO Track;
Query OK, 0 rows affected (0.01 sec)
```

// Dropped the Composer Fields

```
mysql> ALTER TABLE Track DROP composerFirstName;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Track DROP composerLastName;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

// Adding values

```
mysql> INSERT INTO Music (tempo, filename, musicID) VALUES (60,'AmazingGrace.wav',1);
Query OK, 1 row affected (0.00 sec)

mysql> select * from Music;
+---------+-------+---------+------------------+-------------+
| musicID | tempo | timeSig | filename         | revisionNum |
+---------+-------+---------+------------------+-------------+
|       1 |    60 | NULL    | AmazingGrace.wav |        NULL |
+---------+-------+---------+------------------+-------------+
1 row in set (0.00 sec)

mysql> UPDATE Music
    -> SET timeSig = '4/4'
    -> WHERE musicID = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

// To keep things from getting overly complex, I am going to drop revisionNum

```
mysql> ALTER TABLE Music DROP revisionNum;
Query OK, 1 row affected (0.01 sec)
Records: 1  Duplicates: 0  Warnings: 0
```

// I will now attempt to add multiple rows simultaneously

```
mysql> INSERT INTO Music (musicID, tempo, timeSig, filename) VALUES
    -> ('2','90','4/4','BattleHymnOfTheRepublic.wav'),
    -> ('3','80','3/4','Greensleeves.wav'),
    -> ('4','110','4/4','EineKleineNachtmusik.wav'),
    -> ('5','70','3/4','AveMaria.wav'),
    -> ('6','40','C','MiserereMeiDeus.wav'),
    -> ('7','90','4/4',’FourSeasons.wav'),
    -> ('8','100','4/4',’YankeeDoodle.wav'),
    -> ('9','120','2/4',’SymphonyNo5.wav'),
    -> ('10','100','4/4',’SymphonyNo9.wav'),
    -> ('11','104','4/4',’Hallelujah.wav'),
    -> ('12','80','3/4',’MyCountryTisOfThee.wav'),
    -> ('13','80','3/4',’GodSaveTheQueen.wav');
Query OK, 12 rows affected (0.00 sec)
Records: 12  Duplicates: 0  Warnings: 0


mysql> select * from Music;
+---------+-------+---------+-----------------------------+
| musicID | tempo | timeSig | filename                    |
+---------+-------+---------+-----------------------------+
|       1 |    60 | 4/4     | AmazingGrace.wav            |
|       2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav |
|       3 |    80 | 3/4     | Greensleeves.wav            |
|       4 |   110 | 4/4     | EineKleineNachtmusik.wav    |
|       5 |    70 | 3/4     | AveMaria.wav                |
|       6 |    40 | C       | MiserereMeiDeus.wav         |
|       7 |    90 | 4/4     | FourSeasons.wav             |
|       8 |   100 | 4/4     | YankeeDoodle.wav            |
|       9 |   120 | 2/4     | SymphonyNo5.wav             |
|      10 |   100 | 4/4     | SymphonyNo9.wav             |
|      11 |   104 | 4/4     | Hallelujah.wav              |
|      12 |    80 | 3/4     | MyCountryTisOfThee.wav      |
|      13 |    80 | 3/4     | GodSaveTheQueen.wav         |
+---------+-------+---------+-----------------------------+
13 rows in set (0.00 sec)
```

// Yay!

// When I try to insert lyrics, it appears that I need to add escape characters in front of apostrophes. To accomplish this, I used
the search/replace functionality of vim.

```
mysql> INSERT INTO Lyric (lyricID, lyricTitle, words)
    -> VALUES ('4','Ave Maria','Ave Maria! Jungfrau mild,
    '> Erhöre einer Jungfrau Flehen,
    '> Aus diesem Felsen starr und wild
    '> Soll mein Gebet zu dir hin wehen.
    '> Wir schlafen sicher bis zum Morgen,
    '> Ob Menschen noch so grausam sind.
    '> O Jungfrau, sieh der Jungfrau Sorgen,
    '> O Mutter, hör ein bittend Kind!
    '> Ave Maria!
    '>
    '> Ave Maria! Unbefleckt!
    '> Wenn wir auf diesen Fels hinsinken
    '> Zum Schlaf, und uns dein Schutz bedeckt
    '> Wird weich der harte Fels uns dünken.
    '> Du lächelst, Rosendüfte wehen
    '> In dieser dumpfen Felsenkluft,
    '> O Mutter, höre Kindes Flehen,
    '> O Jungfrau, eine Jungfrau ruft!
    '> Ave Maria!
    '>
    '> Ave Maria! Reine Magd!
    '> Der Erde und der Luft Dämonen,
    '> Von deines Auges Huld verjagt,
    '> Sie können hier nicht bei uns wohnen,
    '> Wir woll'n uns still dem Schicksal beugen,
    -> Da uns dein heil'ger Trost anweht;
    '> Der Jungfrau wolle hold dich neigen,
    '> Dem Kind, das für den Vater fleht.
    '> Ave Maria!');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'n uns still dem
Schicksal beugen,
Da uns dein heil'ger Trost anweht;
Der Jungfra' at line 2


// trying this again

mysql> INSERT INTO Lyric (lyricID, lyricTitle, words)
    -> VALUES ('4','Ave Maria',
    -> 'Ave Maria! Jungfrau mild,
    '> Erhöre einer Jungfrau Flehen,
    '> Aus diesem Felsen starr und wild
    '> Soll mein Gebet zu dir hin wehen.
    '> Wir schlafen sicher bis zum Morgen,
    '> Ob Menschen noch so grausam sind.
    '> O Jungfrau, sieh der Jungfrau Sorgen,
    '> O Mutter, hör ein bittend Kind!
    '> Ave Maria!
    '>
    '> Ave Maria! Unbefleckt!
    '> Wenn wir auf diesen Fels hinsinken
    '> Zum Schlaf, und uns dein Schutz bedeckt
    '> Wird weich der harte Fels uns dünken.
    '> Du lächelst, Rosendüfte wehen
    '> In dieser dumpfen Felsenkluft,
    '> O Mutter, höre Kindes Flehen,
    '> O Jungfrau, eine Jungfrau ruft!
    '> Ave Maria!
    '>
    '> Ave Maria! Reine Magd!
    '> Der Erde und der Luft Dämonen,
    '> Von deines Auges Huld verjagt,
    '> Sie können hier nicht bei uns wohnen,
    '> Wir woll n uns still dem Schicksal beugen,
    '> Da uns dein heil\'ger Trost anweht;
    '> Der Jungfrau wolle hold dich neigen,
    '> Dem Kind, das für den Vater fleht.
    '> Ave Maria!');
Query OK, 1 row affected (0.00 sec)

// And now for the rest…

mysql> INSERT INTO Lyric (lyricID, lyricTitle, words)
    -> VALUES ('5','Miserere mei, Deus',
    -> 'Miserere mei, Deus: secundum magnam misericordiam tuam.
    '> Et secundum multitudinem miserationum tuarum, dele iniquitatem meam.
    '> Amplius lava me ab iniquitate mea: et a peccato meo munda me.
    '> Quoniam iniquitatem meam ego cognosco: et peccatum meum contra me est semper.
    '> Tibi soli peccavi, et malum coram te feci: ut justificeris in sermonibus tuis, et vincas cum
judicaris.
    '> Ecce enim in iniquitatibus conceptus sum: et in peccatis concepit me mater mea.
    '> Ecce enim veritatem dilexisti: incerta et occulta sapientiae tuae manifestasti mihi.
    '> Asperges me hysopo, et mundabor: lavabis me, et super nivem dealbabor.
    '> Auditui meo dabis gaudium et laetitiam: et exsultabunt ossa humiliata.
    '> Averte faciem tuam a peccatis meis: et omnes iniquitates meas dele.
    '> Cor mundum crea in me, Deus: et spiritum rectum innova in visceribus meis.
    '> Ne proiicias me a facie tua: et spiritum sanctum tuum ne auferas a me.
    '> Redde mihi laetitiam salutaris tui: et spiritu principali confirma me.
    '> Docebo iniquos vias tuas: et impii ad te convertentur.
    '> Libera me de sanguinibus, Deus, Deus salutis meae: et exsultabit lingua mea justitiam
tuam.
    '> Domine, labia mea aperies: et os meum annuntiabit laudem tuam.
    '> Quoniam si voluisses sacrificium, dedissem utique: holocaustis non delectaberis.
    '> Sacrificium Deo spiritus contribulatus: cor contritum, et humiliatum, Deus, non despicies.
    '> Benigne fac, Domine, in bona voluntate tua Sion: ut aedificentur muri Ierusalem.
    '> Tunc acceptabis sacrificium justitiae, oblationes, et holocausta: tunc imponent
    '> super altare tuum vitulos.');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO Lyric (lyricID, lyricTitle, words)
    -> VALUES ('8','Yankee Doodle',
    -> 'Yankee Doodle went to town
    '> A-riding on a pony,
    '> Stuck a feather in his cap
    '> And called it macaroni .
    '>
    '> Yankee Doodle keep it up,
    '> Yankee Doodle dandy,
    '> Mind the music and the step,
    '> And with the girls be handy.
    '>
    '> Fath r and I went down to camp,
    '> Along with Captain Gooding,
    '> And there we saw the men and boys
    '> As thick as hasty pudding.
    '>
    '> And there we saw a thousand men
    '> As rich as Squire David,
    '> And what they wasted every day,
    '> I wish it could be saved.
    '>
    '> The  lasses they eat it every day,
    '> Would keep a house a winter
    '> They have so much, that I ll be bound,
    '> They eat it when they ve mind ter.
    '>
    '> And there I see a swamping gun
    '> Large as a log of maple,
    '> Upon a deuced little cart,
    '> A load for father s cattle.
    '>
    '> And every time they shoot it off,
    '> It takes a horn of powder,
    '> and makes a noise like father s gun,
    '> Only a nation louder.
    '>
    '> I went as nigh to one myself
    '> As  Siah s inderpinning
    '> And father went as nigh again,
    '> I thought the deuce was in him.
    '>
    '> Cousin Simon grew so bold,
    '> I thought he would have cocked it
    '> It scared me so I shrinked it off
    '> And hung by father s pocket.
    '>
    '> And Cap n Davis had a gun,
    '> He kind of clapt his hand on t
    '> And stuck a crooked stabbing iron
    '> Upon the little end on t
    '>
    '> And there I see a pumpkin shell
    '> As big as mother s bason,
    '> And every time they touched it off
    '> They scampered like the nation.
    '>
    '> I see a little barrel too,
    '> The heads were made of leather
    '> They knocked on it with little clubs
    '> And called the folks together.
    '>
    '> And there was Cap n Washington,
    '> And gentle folks about him
    '> They say he s grown so tarnal proud
    '> He will not ride without em .
    '>
    '> He got him on his meeting clothes,
    '> Upon a slapping stallion
    '> He sat the world along in rows,
    '> In hundreds and in millions.
    '>
    '> The flaming ribbons in his hat,
    '> They looked so tearing fine, ah,
    '> I wanted dreadfully to get
    '> To give to my Jemima.
    '>
    '> I see another snarl of men
    '> A digging graves they told me,
    '> So  tarnal long, so tarnal deep,
    '> They  tended they should hold me.
    '>
    '> It scared me so, I hooked it off,
    '> Nor stopped, as I remember,
    '> Nor turned about till I got home,
    '> Locked up in mother s chamber.'
    -> );
Query OK, 1 row affected (0.00 sec)


// Just realized that “Hallelujah!” is actually just a part of "Messiah, Pt 2"

mysql>
mysql> INSERT INTO Lyric (lyricID, lyricTitle, words)
    -> VALUES ('11','Messiah, Pt 2 (Hallelujah!)',
    -> 'Hallelujah! for the Lord God omnipotent reigneth.
```

```
   '>
   '> . . the kingdoms of this world are become the kingdoms of our Lord, and of His Christ: and
He shall reign for ever and ever.
   '>
   '> . . . KING OF KINGS, LORD OF LORDS.'
   -> );
Query OK, 1 row affected (0.00 sec)


mysql> INSERT INTO Lyric (lyricID, lyricTitle, words)
   -> VALUES ('12','My Country Tis of Thee',
   -> 'My country, tis of thee,
   '> Sweet land of liberty,
   '> Of thee I sing
   '> Land where my fathers died,
   '> Land of the pilgrims pride,
   '> From evry mountainside
   '> Let freedom ring!
   '>
   '> My native country, thee,
   '> Land of the noble free,
   '> Thy name I love
   '> I love thy rocks and rills,
   '> Thy woods and templed hills
   '> My heart with rapture thrills,
   '> Like that above.
   '>
   '> Let music swell the breeze,
   '> And ring from all the trees
   '> Sweet freedoms song
   '> Let mortal tongues awake
   '> Let all that breathe partake
   '> Let rocks their silence break,
   '> The sound prolong.
   '>
   '> Our fathers God to Thee,
   '> Author of liberty,
   '> To Thee we sing.
   '> Long may our land be bright,
   '> With freedoms holy light,
   '> Protect us by Thy might,
   '> Great God our King.');
Query OK, 1 row affected (0.00 sec)


mysql> INSERT INTO Lyric (lyricID, lyricTitle, words)
   -> VALUES ('13','God Save the Queen',
   -> 'God save our gracious Queen,
   '> Long live our noble Queen,
   '> God save The Queen!
   '> Send her victorious,
   '> Happy and glorious,
   '> Long to reign over us,
   '> God save The Queen.
   '>
   '> Thy choicest gifts in store
   '> On her be pleased to pour,
   '> Long may she reign
   '> May she defend our laws,
   '> And ever give us cause
   '> To sing with heart and voice,
   '> God save The Queen!
   '>
   '> God bless our native land,
   '> May heaven s protective hand
   '> Still guard our shore
   '> May peace her power extend,
   '> Foe be transformed to friend,
   '> And Britain s power depend
   '> On war no more.
   '>
   '> May just and righteous laws
   '> Uphold the public cause,
   '> And bless our isle.
   '> Home of the brave and free,
   '> Fair land and liberty,
   '> We pray that still on thee
   '> Kind heaven may smile.
   '>
   '> And not this land alone-
   '> But be thy mercies known
   '> From shore to shore.
   '> Lord, make the nations see
   '> That men should brothers be,
   '> And from one family
   '> The wide world o er.');
Query OK, 1 row affected (0.00 sec)
```

// at this point, I ran into trouble because a select * on the Track table would not return any rows, even though the key is a composite of foreign keys from both Music and Lyric, which both definitely have values. After fretting for a while, I tried running SHOW CREATE TABLE:

```
mysql> SHOW CREATE TABLE Music
    -> ;
+-------
+------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------+
| Table | Create
Table
                                                  |
+-------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------+
| Music | CREATE TABLE `Music` (
  `musicID` int(11) NOT NULL default '0',
  `tempo` float default NULL,
  `timeSig` char(10) default NULL,
  `filename` char(80) NOT NULL,
  PRIMARY KEY  (`musicID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 |
+-------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SHOW CREATE TABLE Track;
+-------
+------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------+
| Table | Create Table

                                                       |
+-------
+------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------+
| Track | CREATE TABLE `Track` (
  `musicID` int(11) NOT NULL default '0',
  `lyricID` int(11) NOT NULL default '0',
  `songTitle` char(80) NOT NULL,
  PRIMARY KEY  (`musicID`,`lyricID`),
  KEY `lyricID` (`lyricID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 |
+-------
+------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

// I do not see any indication of a foreign key. What happened to my foreign keys????

// In the interest of time, I will just populate the Track table manually.

// I just realized that I synchronized by lyricIDs to my musicIDs. While this is fine, I am going to change the lyricIDs to be in strict order, so the musicIDs and lyricIDs don't match exactly (to simulate a more real-world usage).

```
UPDATE Lyric
SET LyricID = 3
WHERE lyricID = 4;


UPDATE Lyric
SET LyricID = 4
WHERE lyricID = 5;

UPDATE Lyric
SET LyricID = 5
WHERE lyricID = 8;


UPDATE Lyric
SET LyricID = 6
WHERE lyricID = 11;


UPDATE Lyric
SET LyricID = 7
WHERE lyricID = 12;


UPDATE Lyric
SET LyricID = 8
WHERE lyricID = 13;


mysql> SELECT lyricID, lyricTitle FROM Lyric;
+---------+-----------------------------+
| lyricID | lyricTitle |
+---------+-----------------------------+
| 1 | Amazing Grace |
| 2 | Battle Hymn of the Republic |
| 3 | Ave Maria |
| 4 | Miserere mei, Deus |
| 5 | Yankee Doodle |
| 6 | Messiah, Pt 2 (Hallelujah!) |
| 7 | My Country Tis of Thee |
| 8 | God Save the Queen |
+---------+-----------------------------+
8 rows in set (0.00 sec)
```

```
INSERT INTO Track
(musicID, lyricID, songTitle)
VALUES (1,1,'Amazing Graces');
```

// So, the next song is just an instrumental of Greensleeves. Since I am doing this manually, I will insert a NULL value for the musicID/

```
INSERT INTO Track
(musicID, lyricID, songTitle)
VALUES (2,2,'Battle Hymn of the Republic'),
(NULL,3,'Greensleeves');


INSERT INTO Track
(musicID, lyricID, songTitle)
VALUES (2,2,'Battle Hymn of the Republic'),
(NULL,3,'Greensleeves');
```

// and now for the rest...

```
mysql> INSERT INTO Track
-> (musicID, lyricID, songTitle)
-> VALUES (NULL,4,'Eine Kleine Nachtmusik'),
-> (3,5,'Ave Maria'),
-> (4,6,'Miserere mei, Deus'),
-> (NULL,7,'Four Seasons'),
-> (5,8,'Yankee Doodle'),
-> (NULL,9,'Symphony No. 5'),
-> (NULL,10,'Symphony No. 9'),
-> (6,11,'Messiah, Pt. 2 (Hallelujah!)'),
-> (7,12,'My Country Tis of Thee'),
-> (8,13,'God Save the Queen');
Query OK, 10 rows affected, 4 warnings (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 4

mysql> select * from Track;
+---------+---------+------------------------------+
| musicID | lyricID | songTitle |
+---------+---------+------------------------------+
| 1 | 1 | Amazing Grace |
| 2 | 2 | Battle Hymn of the Republic |
| 0 | 3 | Greensleeves |
| 0 | 4 | Eine Kleine Nachtmusik |
| 3 | 5 | Ave Maria |
| 4 | 6 | Miserere mei, Deus |
| 0 | 7 | Four Seasons |
| 5 | 8 | Yankee Doodle |
| 0 | 9 | Symphony No. 5 |
| 0 | 10 | Symphony No. 9 |
| 6 | 11 | Messiah, Pt. 2 (Hallelujah!) |
| 7 | 12 | My Country Tis of Thee |
| 8 | 13 | God Save the Queen |
+---------+---------+------------------------------+
13 rows in set (0.00 sec)

// now, let's try some joins:

mysql> SELECT * FROM Music JOIN Track
-> ON Music.musicID = Track.musicID;
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
| musicID | tempo | timeSig | filename | musicID | lyricID | songTitle |
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
| 1 | 60 | 4/4 | AmazingGrace.wav | 1 | 1 | Amazing Grace |
| 2 | 90 | 4/4 | BattleHymnOfTheRepublic.wav | 2 | 2 | Battle Hymn of the Republic |
| 3 | 80 | 3/4 | Greensleeves.wav | 3 | 5 | Ave Maria |
| 4 | 110 | 4/4 | EineKleineNachtmusik.wav | 4 | 6 | Miserere mei, Deus |
| 5 | 70 | 3/4 | AveMaria.wav | 5 | 8 | Yankee Doodle |
| 6 | 40 | C | MiserereMeiDeus.wav | 6 | 11 | Messiah, Pt. 2 (Hallelujah!) |
| 7 | 90 | 4/4 | FourSeasons.wav | 7 | 12 | My Country Tis of Thee |
| 8 | 100 | 4/4 | YankeeDoodle.wav | 8 | 13 | God Save the Queen |
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
8 rows in set (0.01 sec)


mysql> SELECT * FROM Music LEFT JOIN Track ON Music.musicID = Track.musicID;
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
| musicID | tempo | timeSig | filename | musicID | lyricID | songTitle |
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
| 1 | 60 | 4/4 | AmazingGrace.wav | 1 | 1 | Amazing Grace |
| 2 | 90 | 4/4 | BattleHymnOfTheRepublic.wav | 2 | 2 | Battle Hymn of the Republic |
| 3 | 80 | 3/4 | Greensleeves.wav | 3 | 5 | Ave Maria |
| 4 | 110 | 4/4 | EineKleineNachtmusik.wav | 4 | 6 | Miserere mei, Deus |
| 5 | 70 | 3/4 | AveMaria.wav | 5 | 8 | Yankee Doodle |
| 6 | 40 | C | MiserereMeiDeus.wav | 6 | 11 | Messiah, Pt. 2 (Hallelujah!) |
| 7 | 90 | 4/4 | FourSeasons.wav | 7 | 12 | My Country Tis of Thee |
| 8 | 100 | 4/4 | YankeeDoodle.wav | 8 | 13 | God Save the Queen |
| 9 | 120 | 2/4 | SymphonyNo5.wav | NULL | NULL | NULL |
| 10 | 100 | 4/4 | SymphonyNo9.wav | NULL | NULL | NULL |
| 11 | 104 | 4/4 | Hallelujah.wav | NULL | NULL | NULL |
| 12 | 80 | 3/4 | MyCountryTisOfThee.wav | NULL | NULL | NULL |
| 13 | 80 | 3/4 | GodSaveTheQueen.wav | NULL | NULL | NULL |
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
13 rows in set (0.00 sec)

mysql> SELECT * FROM Music RIGHT JOIN Track ON Music.musicID = Track.musicID;
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
| musicID | tempo | timeSig | filename | musicID | lyricID | songTitle |
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
| 1 | 60 | 4/4 | AmazingGrace.wav | 1 | 1 | Amazing Grace |
| 2 | 90 | 4/4 | BattleHymnOfTheRepublic.wav | 2 | 2 | Battle Hymn of the Republic |
| NULL | NULL | NULL | NULL | 0 | 3 | Greensleeves |
| NULL | NULL | NULL | NULL | 0 | 4 | Eine Kleine Nachtmusik |
| 3 | 80 | 3/4 | Greensleeves.wav | 3 | 5 | Ave Maria |
| 4 | 110 | 4/4 | EineKleineNachtmusik.wav | 4 | 6 | Miserere mei, Deus |
| NULL | NULL | NULL | NULL | 0 | 7 | Four Seasons |
| 5 | 70 | 3/4 | AveMaria.wav | 5 | 8 | Yankee Doodle |
| NULL | NULL | NULL | NULL | 0 | 9 | Symphony No. 5 |
| NULL | NULL | NULL | NULL | 0 | 10 | Symphony No. 9 |
| 6 | 40 | C | MiserereMeiDeus.wav | 6 | 11 | Messiah, Pt. 2 (Hallelujah!) |
| 7 | 90 | 4/4 | FourSeasons.wav | 7 | 12 | My Country Tis of Thee |
| 8 | 100 | 4/4 | YankeeDoodle.wav | 8 | 13 | God Save the Queen |
+---------+-------+---------+---------------------------+---------+---------+------------------------------+
13 rows in set (0.01 sec)
```

// received an email from Dr. Lockwood recommending that I change the Table engine to InnoDB

```
mysql> ALTER TABLE Track ENGINE=InnoDB;
Query OK, 13 rows affected (0.06 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> DESC Track;
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| musicID   | int(11)  | NO   | PRI | 0       |       |
| lyricID   | int(11)  | NO   | PRI | 0       |       |
| songTitle | char(80) | NO   |     |         |       |
+-----------+----------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> ALTER TABLE Music ENGINE=InnoDB;
Query OK, 13 rows affected (0.00 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Lyric ENGINE=InnoDB;
Query OK, 8 rows affected (0.09 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Record ENGINE=InnoDB;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

// but, I still can’t create a foreign key. At least it now gives me an actual error:

```
ALTER TABLE Track
    -> ADD FOREIGN KEY (musicID)
    -> references Music(musicID);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`mblackdb/#sql-4df_67d3`, CONSTRAINT `#sql-4df_67d3_ibfk_1` FOREIGN KEY (`musicID`)
REFERENCES `Music` (`musicID`))
mysql> DROP TABLE Track;
Query OK, 0 rows affected (0.03 sec)
```

// dropping and recreating the Track table, with the foreign keys and cascade into built-in

```
CREATE TABLE `Track` (
  `musicID` int(11) NOT NULL default '0',
  `lyricID` int(11) NOT NULL default '0',
  `songTitle` char(80) NOT NULL,
  PRIMARY KEY  (`musicID`,`lyricID`),
  KEY `lyricID` (`lyricID`),
 FOREIGN KEY (musicID) references Music(musicID)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
 FOREIGN KEY (lyricID) references Lyric(lyricID)
  ON DELETE CASCADE
  ON UPDATE CASCADE
)
  ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

// re-inserting the values

```
mysql> INSERT INTO Track
    -> (musicID, lyricID, songTitle)
    -> VALUES
    -> (1,1,'Amazing Grace'),
    -> (2,2,'Battle Hymn of the Republic'),
    -> (NULL,3,'Greensleeves'),
    -> (NULL,4,'Eine Kleine Nachtmusik'),
    -> (3,5,'Ave Maria'),
    -> (4,6,'Miserere mei, Deus'),
    -> (NULL,7,'Four Seasons'),
    -> (5,8,'Yankee Doodle'),
    -> (NULL,9,'Symphony No. 5'),
    -> (NULL,10,'Symphony No. 9'),
    -> (6,11,'Messiah, Pt. 2 (Hallelujah!)'),
    -> (7,12,'My Country Tis of Thee'),
    -> (8,13,'God Save the Queen');
```

```
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`mblackdb/Track`, CONSTRAINT `Track_ibfk_1` FOREIGN KEY (`musicID`) REFERENCES
`Music` (`musicID`) ON DELETE CASCADE ON UPDATE CASCADE)
mysql> SELECT * FROM Track;
Empty set (0.00 sec)
```

// Not sure why it's not liking this, but I think I may have columns out of order in the INSERT statement. Also, maybe it doesn't like NULLs being added to a composite key.

```
 INSERT INTO Track
    -> (lyricID, musicID, songTitle)
    -> VALUES
    -> (1,1,'Amazing Grace'),
    -> (2,2,'Battle Hymn of the Republic'),
    -> (0,3,'Greensleeves'),
    -> (0,4,'Eine Kleine Nachtmusik'),
    -> (3,5,'Ave Maria'),
    -> (4,6,'Miserere mei, Deus'),
    -> (0,7,'Four Seasons'),
    -> (5,8,'Yankee Doodle'),
    -> (0,9,'Symphony No. 5'),
    -> (0,10,'Symphony No. 9'),
    -> (6,11,'Messiah, Pt. 2 (Hallelujah!)'),
    -> (7,12,'My Country Tis of Thee'),
    -> (8,13,'God Save the Queen');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`mblackdb/Track`, CONSTRAINT `Track_ibfk_2` FOREIGN KEY (`lyricID`) REFERENCES
`Lyric` (`lyricID`) ON DELETE CASCADE ON UPDATE CASCADE)
```

// OK, let's try adding the values one-at-a-time.

```
mysql>  INSERT INTO Track
    -> (lyricID, musicID, songTitle)
    -> VALUES
    -> (1,1,'Amazing Grace');
Query OK, 1 row affected (0.05 sec)

mysql>  INSERT INTO Track
    -> (lyricID, musicID, songTitle)
    -> VALUES
    -> (2,2,'Battle Hymn of the Republic'),
    -> (0,3,'Greensleeves');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`mblackdb/Track`, CONSTRAINT `Track_ibfk_2` FOREIGN KEY (`lyricID`) REFERENCES
`Lyric` (`lyricID`) ON DELETE CASCADE ON UPDATE CASCADE)
```

```
mysql>  INSERT INTO Track
    -> (lyricID, musicID, songTitle)
    -> VALUES
    -> (2,2,'Battle Hymn of the Republic');
Query OK, 1 row affected (0.07 sec)

mysql>  INSERT INTO Track
    -> (lyricID, musicID, songTitle)
    -> VALUES
    -> (NULL,3,'Greensleeves');
ERROR 1048 (23000): Column 'lyricID' cannot be null
```

// Oops. Back to 0s

```
mysql> INSERT INTO Track (lyricID, musicID, songTitle) VALUES  (0,3,'Greensleeves');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`mblackdb/Track`, CONSTRAINT `Track_ibfk_2` FOREIGN KEY (`lyricID`) REFERENCES
`Lyric` (`lyricID`) ON DELETE CASCADE ON UPDATE CASCADE)
```

// What if I try only updating the musicID, since Greensleeves is an instrumental here.

```
mysql> INSERT INTO Track (musicID, songTitle) VALUES  (3,'Greensleeves');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`mblackdb/Track`, CONSTRAINT `Track_ibfk_2` FOREIGN KEY (`lyricID`) REFERENCES
`Lyric` (`lyricID`) ON DELETE CASCADE ON UPDATE CASCADE)
```

// maybe what I am trying to add doesn't match up with the referenced foreign keys. let's take a look at the tables:

```
mysql> SELECT musicID, fileName FROM Music;
+---------+-----------------------------+
| musicID | fileName                    |
+---------+-----------------------------+
|       1 | AmazingGrace.wav            |
|       2 | BattleHymnOfTheRepublic.wav |
|       3 | Greensleeves.wav            |
|       4 | EineKleineNachtmusik.wav    |
|       5 | AveMaria.wav                |
|       6 | MiserereMeiDeus.wav         |
|       7 | FourSeasons.wav             |
|       8 | YankeeDoodle.wav            |
|       9 | SymphonyNo5.wav             |
|      10 | SymphonyNo9.wav             |
|      11 | Hallelujah.wav              |
|      12 | MyCountryTisOfThee.wav      |
|      13 | GodSaveTheQueen.wav         |
+---------+-----------------------------+
13 rows in set (0.00 sec)
```

```
mysql> SELECT lyricID, lyricTitle FROM Lyric;
+---------+----------------------------+
| lyricID | lyricTitle                 |
+---------+----------------------------+
|       1 | Amazing Grace              |
|       2 | Battle Hymn of the Republic |
|       3 | Ave Maria                  |
|       4 | Miserere mei, Deus         |
|       5 | Yankee Doodle              |
|       6 | Messiah, Pt 2 (Hallelujah!) |
|       7 | My Country Tis of Thee     |
|       8 | God Save the Queen         |
|      99 | HappyPants                 |
+---------+——————————————+
```

// I think I now see a flaw in my design: I attempted to allow there to exist a lyric that does not have music (or vice versa) but this would require NULL values. But I am not allowed to have NULLS in a key. I need to figure out another way to accomplish this

// I think I should fold "Music" into “Track.” Actually, I will drop Track, and just rename Music to “Song.”

```
mysql> RENAME TABLE Track TO Song;
Query OK, 0 rows affected (0.04 sec)
```

// Adding the columns from Track

```
mysql> ALTER TABLE Song ADD COLUMN SongTitle VARCHAR(20);
Query OK, 13 rows affected (0.13 sec)
Records: 13  Duplicates: 0  Warnings: 0
```

// Now I will add lyricID as a foreign key into Song.
// Note: I am keeping the lyrics in a separate table to allow for the possibility of instrumentals, or lyrics without a song.

```
mysql> ALTER TABLE Song
    -> ADD lyricID int(11);
Query OK, 13 rows affected (0.05 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql>
mysql> ALTER TABLE Song
    -> ADD FOREIGN KEY (lyricID) REFERENCES Lyric(lyricID);
Query OK, 13 rows affected (0.05 sec)
Records: 13  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM Song;
+---------+-------+---------+----------------------------+-----------+---------+
| musicID | tempo | timeSig | filename                   | SongTitle | lyricID |
+---------+-------+---------+----------------------------+-----------+---------+
|       1 |    60 | 4/4     | AmazingGrace.wav           | NULL      |    NULL |
|       2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav | NULL      |    NULL |
|       3 |    80 | 3/4     | Greensleeves.wav           | NULL      |    NULL |
|       4 |   110 | 4/4     | EineKleineNachtmusik.wav   | NULL      |    NULL |
|       5 |    70 | 3/4     | AveMaria.wav               | NULL      |    NULL |
|       6 |    40 | C       | MiserereMeiDeus.wav        | NULL      |    NULL |
|       7 |    90 | 4/4     | FourSeasons.wav            | NULL      |    NULL |
|       8 |   100 | 4/4     | YankeeDoodle.wav           | NULL      |    NULL |
|       9 |   120 | 2/4     | SymphonyNo5.wav            | NULL      |    NULL |
|      10 |   100 | 4/4     | SymphonyNo9.wav            | NULL      |    NULL |
|      11 |   104 | 4/4     | Hallelujah.wav             | NULL      |    NULL |
|      12 |    80 | 3/4     | MyCountryTisOfThee.wav     | NULL      |    NULL |
|      13 |    80 | 3/4     | GodSaveTheQueen.wav        | NULL      |    NULL |
+---------+-------+---------+----------------------------+-----------+---------+
13 rows in set (0.00 sec)

mysql> select lyricID, lyricTitle from Lyric;
+---------+-----------------------------+
| lyricID | lyricTitle                  |
+---------+-----------------------------+
|       1 | Amazing Grace               |
|       2 | Battle Hymn of the Republic |
|       3 | Ave Maria                   |
|       4 | Miserere mei, Deus          |
|       5 | Yankee Doodle               |
|       6 | Messiah, Pt 2 (Hallelujah!) |
|       7 | My Country Tis of Thee      |
|       8 | God Save the Queen          |
|      99 | HappyPants                  |
+---------+-----------------------------+

9 rows in set (0.00 sec)
```

// I need to increase the size of the SongTitle field. It is too small for some of the titles:

```
mysql> ALTER TABLE Song MODIFY SongTitle VARCHAR(80);
Query OK, 13 rows affected (0.04 sec)
Records: 13  Duplicates: 0  Warnings: 0
```

// now I will insert values for lyricID and SongTitle

```
mysql> UPDATE Song
    -> SET lyricID = 1, SongTitle = 'Amazing Grace'
    -> WHERE musicID = 1;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Song
    -> SET lyricID = 2, SongTitle = 'Battle Hymn of The Republic'
    -> WHERE musicID = 2;
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET SongTitle = 'Greensleeves'
    -> WHERE musicID = 3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET SongTitle = 'Eine Kleine Nachtmusik'
    -> WHERE musicID = 4;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET lyricID = 3, SongTitle = 'Ave Maria'
    -> WHERE musicID = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET lyricID =  4, SongTitle = 'Miserere mei, Deus'
    -> WHERE musicID = 6;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET SongTitle = 'Four Seasons'
    -> WHERE musicID = 7;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
```

```
    -> SET lyricID = 5, SongTitle = 'Yankee Doodle'
    -> WHERE musicID = 8;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET SongTitle = 'Symphony No. 5'
    -> WHERE musicID = 9;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET SongTitle = 'Symphony No. 9'
    -> WHERE musicID = 10;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET lyricID = 6, SongTitle = 'Messiah'
    -> WHERE musicID = 11;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET lyricID = 7, SongTitle = 'My Country tis of Thee'
    -> WHERE musicID = 12;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Song
    -> SET lyricID = 8, SongTitle = 'God Save The Queen'
    -> WHERE musicID = 13;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> SELECT * FROM Song;
+---------+-------+---------+-----------------------------+-----------------------------+---------+
| musicID | tempo | timeSig | filename                    | SongTitle                   | lyricID |
+---------+-------+---------+-----------------------------+-----------------------------+---------+
|       1 |    60 | 4/4     | AmazingGrace.wav            | Amazing Grace               |       1 |
|       2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav | Battle Hymn of The Republic |       2 |
|       3 |    80 | 3/4     | Greensleeves.wav            | Greensleeves                |    NULL |
|       4 |   110 | 4/4     | EineKleineNachtmusik.wav    | Eine Kleine Nachtmusik      |    NULL |
|       5 |    70 | 3/4     | AveMaria.wav               | Ave Maria                  |       3 |
|       6 |    40 | C       | MiserereMeiDeus.wav        | Miserere mei, Deus         |       4 |
|       7 |    90 | 4/4     | FourSeasons.wav            | Four Seasons               |    NULL |
|       8 |   100 | 4/4     | YankeeDoodle.wav           | Yankee Doodle              |       5 |
|       9 |   120 | 2/4     | SymphonyNo5.wav            | Symphony No. 5             |    NULL |
|      10 |   100 | 4/4     | SymphonyNo9.wav            | Symphony No. 9             |    NULL |
|      11 |   104 | 4/4     | Hallelujah.wav             | Messiah                    |       6 |
|      12 |    80 | 3/4     | MyCountryTisOfThee.wav     | My Country tis of Thee     |       7 |
|      13 |    80 | 3/4     | GodSaveTheQueen.wav        | God Save The Queen         |       8 |
+---------+-------+---------+----------------------------+----------------------------+---------+
13 rows in set (0.00 sec)


// Now, let's try a LEFT JOIN:

mysql> SELECT musicID, tempo, timeSig, filename, SongTitle, Song.lyricID, lyricTitle  FROM
Song LEFT JOIN Lyric ON Song.lyricID = Lyric.lyricID;
+---------+-------+---------+----------------------------+----------------------------+---------
+----------------------------+
| musicID | tempo | timeSig | filename                   | SongTitle                  | lyricID |
lyricTitle                 |
+---------+-------+---------+----------------------------+----------------------------+---------
+----------------------------+
|       1 |    60 | 4/4     | AmazingGrace.wav           | Amazing Grace              |       1 | Amazing
Grace              |
|       2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav | Battle Hymn of The Republic |       2 |
Battle Hymn of the Republic |
|       3 |    80 | 3/4     | Greensleeves.wav           | Greensleeves               |    NULL |
NULL                       |
|       4 |   110 | 4/4     | EineKleineNachtmusik.wav   | Eine Kleine Nachtmusik     |    NULL |
NULL                       |
|       5 |    70 | 3/4     | AveMaria.wav               | Ave Maria                  |       3 | Ave Maria
                 |
|       6 |    40 | C       | MiserereMeiDeus.wav        | Miserere mei, Deus         |       4 | Miserere
mei, Deus         |
|       7 |    90 | 4/4     | FourSeasons.wav            | Four Seasons               |    NULL |
NULL                       |
|       8 |   100 | 4/4     | YankeeDoodle.wav           | Yankee Doodle              |       5 | Yankee
Doodle              |
|       9 |   120 | 2/4     | SymphonyNo5.wav            | Symphony No. 5             |    NULL |
NULL                       |
|      10 |   100 | 4/4     | SymphonyNo9.wav            | Symphony No. 9             |    NULL |
NULL                       |
|      11 |   104 | 4/4     | Hallelujah.wav             | Messiah                    |       6 | Messiah, Pt 2
(Hallelujah!) |
|      12 |    80 | 3/4     | MyCountryTisOfThee.wav     | My Country tis of Thee     |       7 | My
Country Tis of Thee      |
|      13 |    80 | 3/4     | GodSaveTheQueen.wav        | God Save The Queen         |       8 | God
Save the Queen          |
+---------+-------+---------+---------------------------+---------------------------+---------
+-----------------------------+
13 rows in set (0.00 sec)
```

// I omitted the Lyric.words column because the lyric entries are so humongous. Notice that some of the songs do not have lyrics (I know that this is historically inaccurate for some of these, but I intentionally omitted song lyrics as a proof-of-concept. Also notice that it is possible for the lyrics to have a different title than the song. I feel that maintaining this separation is a plus for songwriters.

// Now from the Lyrics perspective (RIGHT JOIN)

```
mysql> SELECT musicID, tempo, timeSig, filename, SongTitle, Song.lyricID, lyricTitle  FROM
Song RIGHT JOIN Lyric ON Song.lyricID = Lyric.lyricID;
+---------+-------+---------+---------------------------+---------------------------+---------
+-----------------------------+
| musicID | tempo | timeSig | filename                  | SongTitle                 | lyricID |
lyricTitle                  |
+---------+-------+---------+---------------------------+---------------------------+---------
+-----------------------------+
|       1 |    60 | 4/4     | AmazingGrace.wav          | Amazing Grace             |       1 | Amazing
Grace               |
|       2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav | Battle Hymn of The Republic |       2 |
Battle Hymn of the Republic |
|       5 |    70 | 3/4     | AveMaria.wav              | Ave Maria                 |       3 | Ave Maria
                    |
|       6 |    40 | C       | MiserereMeiDeus.wav       | Miserere mei, Deus        |       4 | Miserere
mei, Deus           |
|       8 |   100 | 4/4     | YankeeDoodle.wav          | Yankee Doodle             |       5 | Yankee
Doodle              |
|      11 |   104 | 4/4     | Hallelujah.wav            | Messiah                   |       6 | Messiah, Pt 2
(Hallelujah!) |
|      12 |    80 | 3/4     | MyCountryTisOfThee.wav    | My Country tis of Thee    |       7 | My
Country Tis of Thee      |
|      13 |    80 | 3/4     | GodSaveTheQueen.wav       | God Save The Queen        |       8 | God
Save the Queen          |
+---------+-------+---------+---------------------------+---------------------------+---------
+-----------------------------+
8 rows in set (0.00 sec)
```

// Now, I need to link this to the Record table, but I realize now that I need a new association table to accommodate the many-to-many relationship between a Record and a Song (a record can have multiple songs, and a song may appear on multiple records).

// So, I will drop Record, and create a new table called Release, and an association table, which will be called Record.

```
mysql> CREATE TABLE `Release` (
   ->  `RecordNum` INT(3),
   ->  `ReleaseDate` DATE,
   ->  `format` enum('LP','EP','Single','Video','BoxSet'),
   ->  `length` int(11),
   ->  `SKU` varchar(20) NOT NULL,
   ->  PRIMARY KEY  (SKU)
   -> )
   -> ENGINE=InnoDB DEFAULT CHARSET=latin1;
Query OK, 0 rows affected (0.02 sec)


mysql> DROP TABLE Record;
Query OK, 0 rows affected (0.02 sec)


mysql> CREATE TABLE Record (
   ->  RecordNum INT(3),
   ->  SongTitle VARCHAR(80),
   ->  TrackNum INT(4)
   -> )
   -> ENGINE=InnoDB DEFAULT CHARSET=latin1;
Query OK, 0 rows affected (0.02 sec)
```

// OOPS. Release is a reserved word in MySQL. I will change it to Product:

```
mysql> RENAME TABLE `Release` TO `Product`;
Query OK, 0 rows affected (0.02 sec)
```

// Now to add the foreign keys:

```
mysql> ALTER TABLE Record ADD FOREIGN KEY (RecordNum) references
Product(RecordNum);
ERROR 1005 (HY000): Can't create table './mblackdb/#sql-4df_67d3.frm' (errno: 150)
```

// OK, I think it is objecting because RecordNum is not a primary key for Product. I think this means that I will need to rename Product to Record and then rename the existing association table to TrackListing.

```
mysql> RENAME TABLE Record to TrackListing;
Query OK, 0 rows affected (0.01 sec)

mysql> RENAME TABLE Product to Record;
```

```
Query OK, 0 rows affected (0.00 sec)
```

// It looks like I cannot use SKU as my primary key in Record. I need it to be RecordNum, so I can insert it as a foreign key into TrackListing.

```
mysql> ALTER TABLE Record
    -> DROP PRIMARY KEY,
    -> ADD PRIMARY KEY(RecordNum);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC Record;
+------------+-----------------------------------------+------+-----+---------+-------+
| Field      | Type                                    | Null | Key | Default | Extra |
+------------+-----------------------------------------+------+-----+---------+-------+
| RecordNum  | int(3)                                  | NO   | PRI | 0       |       |
| ReleaseDate| date                                    | YES  |     | NULL    |       |
| format     | enum('LP','EP','Single','Video','BoxSet') | YES  |     | NULL    |       |
| length     | int(11)                                 | YES  |     | NULL    |       |
| SKU        | varchar(20)                             | NO   |     |         |       |
+------------+-----------------------------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> ALTER TABLE TrackListing
    -> ADD FOREIGN KEY (RecordNum) references Release(RecordNum);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near
'Release(RecordNum)' at line 2
mysql> ALTER TABLE TrackListing
    -> ADD FOREIGN KEY (RecordNum) references Record(RecordNum);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE TrackListing
    -> ADD FOREIGN KEY (SongTitle) references Song(SongTitle);
ERROR 1005 (HY000): Can't create table './mblackdb/#sql-4df_67d3.frm' (errno: 150)
```

// Now I’m running into the same problem with the SongTitle attribute. I will need to use musicID instead. I think I will also rename musicID to songID

```
mysql> ALTER TABLE Song
    -> DROP PRIMARY KEY;
Query OK, 13 rows affected (0.03 sec)
Records: 13  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE Song CHANGE musicID SongID INT(11);
Query OK, 13 rows affected (0.05 sec)
Records: 13  Duplicates: 0  Warnings: 0


mysql> ALTER TABLE Song
    -> ADD PRIMARY KEY(SongID);
Query OK, 13 rows affected (0.07 sec)
Records: 13  Duplicates: 0  Warnings: 0



mysql> DESC Song;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| SongID    | int(11)     | NO   | PRI | 0       |       |
| tempo     | float       | YES  |     | NULL    |       |
| timeSig   | char(10)    | YES  |     | NULL    |       |
| filename  | char(80)    | NO   |     |         |       |
| SongTitle | varchar(80) | YES  |     | NULL    |       |
| lyricID   | int(11)     | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)



// Now I need to change SongTitle to SongID in TrackListing

mysql> ALTER TABLE TrackListing CHANGE SongTitle SongID int(11);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC TrackListing;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| RecordNum | int(3)  | YES  | MUL | NULL    |       |
| SongID    | int(11) | YES  |     | NULL    |       |
| TrackNum  | int(4)  | YES  |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
3 rows in set (0.00 sec)


// Now to add the foreign key to TrackListing:

mysql> ALTER TABLE TrackListing ADD FOREIGN KEY (SongID) references Song(SongID)
    -> ;
```

```
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW CREATE TABLE TrackListing
    -> ;
+--------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
----------------------------------+
| Table       | Create
Table

                                                                                    |
+--------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
----------------------------------+
| TrackListing | CREATE TABLE `TrackListing` (
  `RecordNum` int(3) default NULL,
  `SongID` int(11) default NULL,
  `TrackNum` int(4) default NULL,
  KEY `RecordNum` (`RecordNum`),
  KEY `SongID` (`SongID`),
  CONSTRAINT `TrackListing_ibfk_2` FOREIGN KEY (`SongID`) REFERENCES `Song`
(`SongID`),
  CONSTRAINT `TrackListing_ibfk_1` FOREIGN KEY (`RecordNum`) REFERENCES `Record`
(`RecordNum`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+--------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
----------------------------------+
1 row in set (0.00 sec)
```

// Now I am updating the column names to standardize on PascalCasing, instead of camelCasing.

```
mysql> ALTER TABLE Record CHANGE length Length int(11);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0


mysql> ALTER TABLE Record
```

```
    -> CHANGE format Format enum('LP','EP','Single','Video','BoxSet');
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0


mysql> DESC Record;
+------------+-----------------------------------------+------+-----+---------+-------+
| Field      | Type                                    | Null | Key | Default | Extra |
+------------+-----------------------------------------+------+-----+---------+-------+
| RecordNum  | int(3)                                  | NO   | PRI | 0       |       |
| ReleaseDate | date                                   | YES  |     | NULL    |       |
| Format     | enum('LP','EP','Single','Video','BoxSet') | YES  |     | NULL    |       |
| Length     | int(11)                                 | YES  |     | NULL    |       |
| SKU        | varchar(20)                             | NO   |     |         |       |
+------------+-----------------------------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)


mysql> ALTER TABLE Song
    -> CHANGE tempo Tempo Float,
    -> CHANGE timeSig TimeSig char(10),
    -> CHANGE filename FileName char(80);
Query OK, 13 rows affected (0.02 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql>
mysql> DESC Song;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| SongID    | int(11)     | NO   | PRI | 0       |       |
| Tempo     | float       | YES  |     | NULL    |       |
| TimeSig   | char(10)    | YES  |     | NULL    |       |
| FileName  | char(80)    | YES  |     | NULL    |       |
| SongTitle | varchar(80) | YES  |     | NULL    |       |
| lyricID   | int(11)     | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)



// Will it let me rename a foreign key?

mysql> ALTER TABLE Song
    -> CHANGE lyricID LyricID int(11);
Query OK, 13 rows affected (0.01 sec)
Records: 13  Duplicates: 0  Warnings: 0
```

```
mysql> DESC Song;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| SongID    | int(11)     | NO   | PRI | 0       |       |
| Tempo     | float       | YES  |     | NULL    |       |
| TimeSig   | char(10)    | YES  |     | NULL    |       |
| FileName  | char(80)    | YES  |     | NULL    |       |
| SongTitle | varchar(80) | YES  |     | NULL    |       |
| LyricID   | int(11)     | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> SHOW CREATE TABLE Song;
+-------
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------------------------------+
| Table | Create Table


                                                                                 |
+-------
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------------------------------+
| Song  | CREATE TABLE `Song` (
  `SongID` int(11) NOT NULL default '0',
  `Tempo` float default NULL,
  `TimeSig` char(10) default NULL,
  `FileName` char(80) default NULL,
  `SongTitle` varchar(80) default NULL,
  `LyricID` int(11) default NULL,
  PRIMARY KEY  (`SongID`),
  KEY `lyricID` (`LyricID`),
  CONSTRAINT `Song_ibfk_1` FOREIGN KEY (`lyricID`) REFERENCES `Lyric` (`lyricID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-------
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------------------------------+
1 row in set (0.00 sec)
```

// It allowed me to rename the column, but now my foreign key doesn't match. I will need to recreate the foreign key

```
mysql> ALTER TABLE Song DROP FOREIGN KEY Song_ibfk_1;
Query OK, 13 rows affected (0.03 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Song
    -> ADD FOREIGN KEY (LyricID) references Lyric(LyricID);
Query OK, 13 rows affected (0.01 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> DESC Song;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| SongID    | int(11)     | NO   | PRI | 0       |       |
| Tempo     | float       | YES  |     | NULL    |       |
| TimeSig   | char(10)    | YES  |     | NULL    |       |
| FileName  | char(80)    | YES  |     | NULL    |       |
| SongTitle | varchar(80) | YES  |     | NULL    |       |
| LyricID   | int(11)     | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> SHOW CREATE TABLE Song;
+-------
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
----------------------------------+
| Table | Create Table



                                                                        |
+-------
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
----------------------------------+
| Song  | CREATE TABLE `Song` (
  `SongID` int(11) NOT NULL default '0',
  `Tempo` float default NULL,
  `TimeSig` char(10) default NULL,
  `FileName` char(80) default NULL,
  `SongTitle` varchar(80) default NULL,
  `LyricID` int(11) default NULL,
  PRIMARY KEY  (`SongID`),
  KEY `LyricID` (`LyricID`),
  CONSTRAINT `Song_ibfk_1` FOREIGN KEY (`LyricID`) REFERENCES `Lyric` (`lyricID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-------
+-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------+
1 row in set (0.00 sec)
```

// And now to update Lyric

```
mysql> ALTER TABLE Lyric
    -> CHANGE lyricID LyricID int(11),
    -> CHANGE lyricTitle LyricTitle varchar(80),
    -> CHANGE words Word text;
Query OK, 8 rows affected (0.04 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> DESC Lyric;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| LyricID    | int(11)     | NO   | PRI | 0       |       |
| LyricTitle | varchar(80) | YES  |     | NULL    |       |
| Word       | text        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

// it looks like dropping and adding the primary key is not necessary when renaming a column

// Now it’s time to create some records.

Title: Sweatin to the Renaissance
Record Number: 1
Release Date: March 15th 2012
Format: EP
Length: 25 minutes
SKU: A001-01A12
TrackListing:
01 Greensleeves
02 Eine Kleine Nachtmusik
03 Ave Maria
04 Miserere mei, Deus

Title: Beethoven's 1.8th (9/5)th Symphony
Record Number: 2
Release Date: December 17th 2013 (Beethoven's Birthday!)
Format: LP
Length: 95 minutes
SKU: A002-01A13
TrackListing:
01 Symphony No. 5
01 Symphony No. 9

Title: Two Lyrics, One Song
Record Number: 3
Release Date: August 4th, 2014
Format: Single
Length: 7 minutes
SKU: A003-01A14
Tracklisting:
01 My Country tis of Thee
02 God Save The Queen

Title: Two Lyrics, One Song - King Version (theoretical future re-release after the Queen shuffles off of this mortal coil.)
Record Number: 893
Release Date: February 17th, 2025
Format: Single
Length: 7 minutes
SKU: A893-01A25
Tracklisting:
01 My Country tis of Thee
02 God Save The King

// I will add a new Lyric and Song for “God Save the King”

```
mysql> INSERT INTO Lyric
    -> (LyricID, LyricTitle, Words)
    -> VALUES (9, 'God Save the King','
    '> God save our gracious King,
    '> Long live our noble King,
    '> God save The King!
    '> Send him victorious,
    '> Happy and glorious,
    '> Long to reign over us,
    '> God save The King.
    '>
    '> Thy choicest gifts in store
    '> On him be pleased to pour,
    '> Long may he reign
    '> May he defend our laws,
    '> And ever give us cause
    '> To sing with heart and voice,
    '> God save The King!
    '>
    '> God bless our native land,
    '> May heaven s protective hand
    '> Still guard our shore
    '> May peace him power extend,
    '> Foe be transformed to friend,
    '> And Britain s power depend
    '> On war no more.
    '>
    '> May just and righteous laws
    '> Uphold the public cause,
    '> And bless our isle.
    '> Home of the brave and free,
    '> Fair land and liberty,
    '> We pray that still on thee
    '> Kind heaven may smile.
    '>
    '> And not this land alone-
    '> But be thy mercies known
    '> From shore to shore.
    '> Lord, make the nations see
    '> That men should brothims be,
    '> And from one family
    '> The wide world o er.');
Query OK, 1 row affected (0.02 sec)

mysql> DESC Lyric;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| LyricID    | int(11)     | NO   | PRI | 0       |       |
| LyricTitle | varchar(80) | YES  |     | NULL    |       |
| Words      | text        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> SELECT LyricID, LyricTitle FROM Lyric;
+---------+----------------------------+
| LyricID | LyricTitle                 |
+---------+----------------------------+
|       1 | Amazing Grace              |
|       2 | Battle Hymn of the Republic |
|       3 | Ave Maria                  |
|       4 | Miserere mei, Deus         |
|       5 | Yankee Doodle              |
|      6 | Messiah, Pt 2 (Hallelujah!) |
|      7 | My Country Tis of Thee      |
|      8 | God Save the Queen          |
|      9 | God Save the King           |
+---------+----------------------------+
9 rows in set (0.00 sec)
```

// Now adding God Save the King to the Song table:

```
mysql> INSERT INTO Song
    -> (SongID, Tempo, TimeSig, FileName, SongTitle, LyricID)
    -> VALUES (14,80,'3/4','GodSaveTheKing.wav','God Save the King',9);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Song;
+--------+-------+---------+---------------------------+---------------------------+---------+
| SongID | Tempo | TimeSig | FileName                  | SongTitle                 | LyricID |
+--------+-------+---------+---------------------------+---------------------------+---------+
|      1 |    60 | 4/4     | AmazingGrace.wav          | Amazing Grace             |       1 |
|      2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav | Battle Hymn of The Republic |       2 |
|      3 |    80 | 3/4     | Greensleeves.wav          | Greensleeves              |    NULL |
|      4 |   110 | 4/4     | EineKleineNachtmusik.wav  | Eine Kleine Nachtmusik    |    NULL |
|      5 |    70 | 3/4     | AveMaria.wav              | Ave Maria                 |       3 |
|      6 |    40 | C       | MiserereMeiDeus.wav       | Miserere mei, Deus        |       4 |
|      7 |    90 | 4/4     | FourSeasons.wav           | Four Seasons              |    NULL |
|      8 |   100 | 4/4     | YankeeDoodle.wav          | Yankee Doodle             |       5 |
|      9 |   120 | 2/4     | SymphonyNo5.wav           | Symphony No. 5            |    NULL |
|     10 |   100 | 4/4     | SymphonyNo9.wav           | Symphony No. 9            |    NULL |
|     11 |   104 | 4/4     | Hallelujah.wav            | Messiah                   |       6 |
|     12 |    80 | 3/4     | MyCountryTisOfThee.wav    | My Country tis of Thee    |       7 |
|     13 |    80 | 3/4     | GodSaveTheQueen.wav       | God Save The Queen        |       8 |
|     14 |    80 | 3/4     | GodSaveTheKing.wav        | God Save the King         |       9 |
+--------+-------+---------+---------------------------+---------------------------+---------+
14 rows in set (0.00 sec)
```

// I just realized that I don’t actually have a Title column in my Record Table.

```
mysql> ALTER TABLE Record
    -> ADD COLUMN RecordTitle varchar(80);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC Record;
+-------------+----------------------------------------+------+-----+---------+-------+
| Field       | Type                                   | Null | Key | Default | Extra |
+-------------+-----------------------------------------+------+-----+---------+-------+
| RecordNum   | int(3)                                  | NO   | PRI | 0       |       |
| ReleaseDate | date                                    | YES  |     | NULL    |       |
| Format      | enum('LP','EP','Single','Video','BoxSet') | YES  |     | NULL    |       |
| Length      | int(11)                                 | YES  |     | NULL    |       |
| SKU         | varchar(20)                             | NO   |     |         |       |
| RecordTitle | varchar(80)                             | YES  |     | NULL    |       |
+-------------+-----------------------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)



// Now inserting the records:

mysql> INSERT INTO Record
    -> (RecordNum, ReleaseDate, Format, Length, SKU, RecordTitle)
    -> VALUES (1, '2012/03/25','EP',25,'A001-01A12','Sweatin to the Renaissance');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Record
    -> (RecordNum, ReleaseDate, Format, Length, SKU, RecordTitle)
    -> VALUES (2, '2013/12/17','LP',95,'A002-01A13','Beethovens 1.8th
    '> Symphony'),
    -> (3, '2014/08/04','Single',7,'A003-01A14','Two Lyrics, One Song'),
    -> (893, '2025/02/17','Single',7,'A893-01A25','Two Lyrics, One Song (King)');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0


// Now, I need to do the track listings for each record.

mysql> INSERT INTO TrackListing
    -> (RecordNum, SongID, TrackNum)
    -> VALUES
    -> (1, 3, 1),
    -> (1, 4, 2),
    -> (1, 5, 3),
    -> (1, 6, 4),
    -> (2, 9, 1),
    -> (2, 10, 2),
    -> (3, 12, 1),
    -> (3, 13, 2),
    -> (893, 12, 1),
    -> (893, 14, 2);
Query OK, 10 rows affected (0.05 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM TrackListing;
+-----------+--------+----------+
| RecordNum | SongID | TrackNum |
+-----------+--------+----------+
|         1 |      3 |        1 |
|         1 |      4 |        2 |
|         1 |      5 |        3 |
|         1 |      6 |        4 |
|         2 |      9 |        1 |
|         2 |     10 |        2 |
|         3 |     12 |        1 |
|         3 |     13 |        2 |
|       893 |     12 |        1 |
|       893 |     14 |        2 |
+-----------+--------+----------+
10 rows in set (0.00 sec)
```

// Now that all the data has been entered, I will try a 3-way JOIN to link the Record, the Tracklisting and Song tables together

```
mysql> SELECT RecordTitle, TrackNum, SongTitle FROM Record JOIN TrackListing ON
Record.RecordNum = TrackListing.RecordNum JOIN Song On TrackListing.SongID =
Song.SongID WHERE Record.RecordNum = 1;
+---------------------------+----------+-----------------------+
| RecordTitle               | TrackNum | SongTitle             |
+---------------------------+----------+-----------------------+
| Sweatin to the Renaissance |        1 | Greensleeves          |
| Sweatin to the Renaissance |        2 | Eine Kleine Nachtmusik |
| Sweatin to the Renaissance |        3 | Ave Maria             |
| Sweatin to the Renaissance |        4 | Miserere mei, Deus    |
+---------------------------+----------+————————————+
4 rows in set (0.00 sec)
```

// this time, with the SKU:

```
mysql> SELECT SKU, RecordTitle, TrackNum, SongTitle FROM Record JOIN TrackListing ON
Record.RecordNum = TrackListing.RecordNum JOIN Song On TrackListing.SongID =
Song.SongID WHERE Record.RecordNum = 2;
+-----------+-----------------+----------+---------------+
| SKU       | RecordTitle     | TrackNum | SongTitle     |
+-----------+-----------------+----------+---------------+
| A002-01A13 | Beethovens 1.8th |        1 | Symphony No. 5 |
| A002-01A13 | Beethovens 1.8th |        2 | Symphony No. 9 |
+-----------+-----------------+----------+---------------+
2 rows in set (0.00 sec)
```

// Now I will try adding the composers into the database. I expect that this will require three more tables: 1 for the composers, and then 2 more intersection tables; 1 for the music and 1 for the lyrics.

```
mysql> CREATE TABLE Composer (
    -> ComposerID int(7),
    ->   FirstName varchar(20),
    ->   LastName varchar(20),
    -> PRIMARY KEY (ComposerID)
    -> ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Query OK, 0 rows affected (0.03 sec)

mysql> DESC Composer;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| ComposerID | int(7)      | NO   | PRI | 0       |       |
| FirstName  | varchar(20) | YES  |     | NULL    |       |
| LastName   | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)


mysql> CREATE TABLE CreditLyric (
    -> ComposerID int(7),
    -> LyricID int(11),
    -> FOREIGN KEY (ComposerID) REFERENCES Composer (ComposerID),
    -> FOREIGN KEY (LyricID) REFERENCES Lyric (LyricID)
    -> ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Query OK, 0 rows affected (0.04 sec)

mysql> DESC CreditLyric;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| ComposerID | int(7)  | YES  | MUL | NULL    |       |
| LyricID    | int(11) | YES  | MUL | NULL    |       |
+------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> SHOW CREATE TABLE CreditLyric;
+-------------
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------+
| Table       | Create
Table
```

```
                                               |
+-------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
-----------+
| CreditLyric | CREATE TABLE `CreditLyric` (
  `ComposerID` int(7) default NULL,
  `LyricID` int(11) default NULL,
  KEY `ComposerID` (`ComposerID`),
  KEY `LyricID` (`LyricID`),
  CONSTRAINT `CreditLyric_ibfk_1` FOREIGN KEY (`ComposerID`) REFERENCES
`Composer` (`ComposerID`),
  CONSTRAINT `CreditLyric_ibfk_2` FOREIGN KEY (`LyricID`) REFERENCES `Lyric` (`LyricID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
-----------+
1 row in set (0.00 sec)

mysql> CREATE TABLE CreditMusic (
    -> ComposerID int(7),
    -> SongID int(11),
    -> FOREIGN KEY (ComposerID) REFERENCES Composer (ComposerID),
    -> FOREIGN KEY (SongID) REFERENCES Song (SongID)
    -> ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Query OK, 0 rows affected (0.04 sec)

mysql> DESC CreditMusic;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| ComposerID | int(7)  | YES  | MUL | NULL    |       |
| SongID     | int(11) | YES  | MUL | NULL    |       |
+------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)


ysql> SHOW CREATE TABLE CreditMusic
    -> ;
+-------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
-----+
| Table       | Create
Table

                                                       |
+-------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
-----+
| CreditMusic | CREATE TABLE `CreditMusic` (
  `ComposerID` int(7) default NULL,
  `SongID` int(11) default NULL,
  KEY `ComposerID` (`ComposerID`),
  KEY `SongID` (`SongID`),
  CONSTRAINT `CreditMusic_ibfk_1` FOREIGN KEY (`ComposerID`) REFERENCES
`Composer` (`ComposerID`),
  CONSTRAINT `CreditMusic_ibfk_2` FOREIGN KEY (`SongID`) REFERENCES `Song`
(`SongID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-------------
+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------
-----+
1 row in set (0.00 sec)



// Now I will insert the composers


mysql> INSERT INTO Composer
    -> (ComposerID, FirstName, LastName)
    -> VALUES ('1','Traditional','Traditional'),
    -> ('2','Ludwig van','Beethoven'),
    -> ('3','George','Handel'),
    -> ('4','John','Newton'),
    -> ('5','Julia','Howe'),
    -> ('6','William','Steffe'),
    -> ('7','Gregorio','Allegri'),
    -> ('8','Charles','Jennens'),
    -> ('9','Henry','Carey'),
    -> ('10','Samuel','Smith');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Composer;
+------------+-------------+-------------+
| ComposerID | FirstName   | LastName    |
+------------+-------------+-------------+
|          1 | Traditional | Traditional |
|          2 | Ludwig van  | Beethoven   |
|          3 | George      | Handel      |
|          4 | John        | Newton      |
|          5 | Julia       | Howe        |
|          6 | William     | Steffe      |
|          7 | Gregorio    | Allegri     |
|          8 | Charles     | Jennens     |
|          9 | Henry       | Carey       |
|         10 | Samuel      | Smith       |
+------------+-------------+-------------+
10 rows in set (0.00 sec)


// OOPS. Forgot about Mozart, Schubert & Vivaldi

mysql> INSERT INTO Composer
    -> (ComposerID, FirstName, LastName)
    -> VALUES ('11','Wolfgang Amadeus','Mozart');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Composer
    -> (ComposerID, FirstName, LastName)
    -> VALUES ('12','Franz','Schubert');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Composer
    -> (ComposerID, FirstName, LastName)
    -> VALUES ('13','Antonio','Vivaldi');
Query OK, 1 row affected (0.00 sec)


// Inserting the Music Credits:

mysql> INSERT INTO CreditMusic
    -> (ComposerID, SongID)
    -> VALUES (2,9),
    -> (2,10),
    -> (3,11),
    -> (6,2),
    -> (7,6),
    -> (1,1),
    -> (1,3),
    -> (11,4),
    -> (12,5),
    -> (13,7),
    -> (1,8),
    -> (1,12),
    -> (1,13),
    -> (1,14);
Query OK, 14 rows affected (0.00 sec)
Records: 14  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM CreditMusic;
+------------+--------+
| ComposerID | SongID |
+------------+--------+
|          2 |      9 |
|          2 |     10 |
|          3 |     11 |
|          6 |      2 |
|          7 |      6 |
|          1 |      1 |
|          1 |      3 |
|         11 |      4 |
|         12 |      5 |
|         13 |      7 |
|          1 |      8 |
|          1 |     12 |
|          1 |     13 |
|          1 |     14 |
+------------+--------+
14 rows in set (0.00 sec)
```

// I realize that I should have a value for “Instrumental” for cases where there are no lyrics. This will reduce NULLs.

```
mysql> INSERT INTO Composer
    -> (ComposerID, FirstName, LastName)
    -> VALUES ('14','Instrumental','Instrumental');
Query OK, 1 row affected (0.00 sec)
```

// Also, forgot about Frederich Schiller

```
mysql> INSERT INTO Composer
    -> (ComposerID, FirstName, LastName)
    -> VALUES ('15','Friedrich','Schiller');
Query OK, 1 row affected (0.00 sec)
```

// Inserting the lyrics for Greensleeves and Ode to Joy

```
mysql> INSERT INTO Lyric
    -> (LyricID, LyricTitle, Words)
    -> VALUES (10, 'Greensleeves','
    '> Alas, my love, you do me wrong,
    '> To cast me off discourteously.
    '> For I have loved you well and long,
    '> Delighting in your company.
    '>
    '> Greensleeves was all my joy
    '> Greensleeves was my delight,
    '> Greensleeves was my heart of gold,
    '> And who but my lady greensleeves.
    '>
    '> Your vows you ve broken, like my heart,
    '> Oh, why did you so enrapture me?
    '> Now I remain in a world apart
    '> But my heart remains in captivity.
    '>
    '> I have been ready at your hand,
    '> To grant whatever you would crave,
    '> I have both wagered life and land,
    '> Your love and good-will for to have.
    '>
    '> If you intend thus to disdain,
    '> It does the more enrapture me,
    '> And even so, I still remain
    '> A lover in captivity.
    '>
    '> My men were clothed all in green,
    '> And they did ever wait on thee
    '> All this was gallant to be seen,
    '> And yet thou wouldst not love me.
    '>
    '> Thou couldst desire no earthly thing,
    '> but still thou hadst it readily.
    '> Thy music still to play and sing
    '> And yet thou wouldst not love me.
    '>
    '> Well, I will pray to God on high,
    '> that thou my constancy mayst see,
    '> And that yet once before I die,
    '> Thou wilt vouchsafe to love me.
    '>
    '> Ah, Greensleeves, now farewell, adieu,
    '> To God I pray to prosper thee,
    '> For I am still thy lover true,
    '> Come once again and love me.
    '>
    '> ');
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO Lyric
```

```
    -> (LyricID, LyricTitle, Words)
    -> VALUES (11, 'Ode to Joy',
    -> '
    '> Joy of beautiful God s sparks
    '> Daughter of the Elysium
    '> Drunken of fire we will enter
    '> Your holy shrine, Heavenly daughter!
    '>
    '> Your miracles will connect again
    '> Whatever fashion had strictly devided
    '> All men will become brothers
    '> Wherever your tender wing remains.
    '>
    '> Whoever had the good luck
    '> To be a friend s friend
    '> Whoever married a beautiful woman
    '> Should mix in his cheers!
    '>
    '> Yes, whoever calls a single soul
    '> To be his on this earth s ball
    '> And whoever never managed should
    '> leave this brotherhood crying!
    '> ');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT LyricID, LyricTitle FROM Lyric;
+---------+-----------------------------+
| LyricID | LyricTitle                  |
+---------+-----------------------------+
|       1 | Amazing Grace               |
|       2 | Battle Hymn of the Republic |
|       3 | Ave Maria                   |
|       4 | Miserere mei, Deus          |
|       5 | Yankee Doodle               |
|       6 | Messiah, Pt 2 (Hallelujah!) |
|       7 | My Country Tis of Thee      |
|       8 | God Save the Queen          |
|       9 | God Save the King           |
|      10 | Greensleeves                |
|      11 | Ode to Joy                  |
+---------+-----------------------------+
11 rows in set (0.00 sec)


// Also need to update Song with this information:

mysql> UPDATE Song
    -> SET LyricID = 10
    -> WHERE SongID = 3;
```

```
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Song
    -> SET LyricID = 11
    -> WHERE SongID = 10;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Song;
+--------+-------+---------+-----------------------------+-----------------------------+---------+
| SongID | Tempo | TimeSig | FileName                    | SongTitle                   | LyricID |
+--------+-------+---------+-----------------------------+-----------------------------+---------+
|      1 |    60 | 4/4     | AmazingGrace.wav            | Amazing Grace               |       1 |
|      2 |    90 | 4/4     | BattleHymnOfTheRepublic.wav | Battle Hymn of The Republic |       2 |
|      3 |    80 | 3/4     | Greensleeves.wav            | Greensleeves                |      10 |
|      4 |   110 | 4/4     | EineKleineNachtmusik.wav    | Eine Kleine Nachtmusik      |    NULL |
|      5 |    70 | 3/4     | AveMaria.wav                | Ave Maria                   |       3 |
|      6 |    40 | C       | MiserereMeiDeus.wav         | Miserere mei, Deus          |       4 |
|      7 |    90 | 4/4     | FourSeasons.wav             | Four Seasons                |    NULL |
|      8 |   100 | 4/4     | YankeeDoodle.wav            | Yankee Doodle               |       5 |
|      9 |   120 | 2/4     | SymphonyNo5.wav             | Symphony No. 5              |    NULL |
|     10 |   100 | 4/4     | SymphonyNo9.wav             | Symphony No. 9              |      11 |
|     11 |   104 | 4/4     | Hallelujah.wav              | Messiah                     |       6 |
|     12 |    80 | 3/4     | MyCountryTisOfThee.wav      | My Country tis of Thee      |       7 |
|     13 |    80 | 3/4     | GodSaveTheQueen.wav         | God Save The Queen          |       8 |
|     14 |    80 | 3/4     | GodSaveTheKing.wav          | God Save the King           |       9 |
+--------+-------+---------+-----------------------------+-----------------------------+---------+
14 rows in set (0.00 sec)



// Adding the lyric credits…

mysql> INSERT INTO CreditLyric
    -> (ComposerID, LyricID)
    -> VALUES (4,1),
    -> (5,2),
    -> (1,3),
    -> (1,4),
    -> (1,5),
    -> (8,6),
    -> (10,7),
    -> (8,8),
    -> (8,9),
    -> (1,10),
    -> (15,10);
Query OK, 11 rows affected (0.00 sec)
Records: 11  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM CreditLyric;
+------------+---------+
| ComposerID | LyricID |
+------------+---------+
|          4 |       1 |
|          5 |       2 |
|          1 |       3 |
|          1 |       4 |
|          1 |       5 |
|          8 |       6 |
|         10 |       7 |
|          8 |       8 |
|          8 |       9 |
|          1 |      10 |
|         15 |      10 |
+------------+---------+
11 rows in set (0.00 sec)
```

// Now I will try a JOIN to match the lyrics with their composers:

```
mysql> SELECT Lyric.LyricID, Lyric.LyricTitle, Composer.FirstName, Composer.LastName
    -> FROM Lyric
    -> JOIN CreditLyric
    -> ON Lyric.LyricID = CreditLyric.LyricID
    -> JOIN Composer
    -> ON CreditLyric.ComposerID = Composer.ComposerID
    -> WHERE Lyric.LyricID IS NOT NULL;
+---------+----------------------------+-------------+-------------+
| LyricID | LyricTitle                 | FirstName   | LastName    |
+---------+----------------------------+-------------+-------------+
|       1 | Amazing Grace              | John        | Newton      |
|       2 | Battle Hymn of the Republic | Julia       | Howe        |
|       3 | Ave Maria                  | Traditional | Traditional |
|       4 | Miserere mei, Deus         | Traditional | Traditional |
|       5 | Yankee Doodle              | Traditional | Traditional |
|       6 | Messiah, Pt 2 (Hallelujah!) | Charles     | Jennens     |
|       7 | My Country Tis of Thee     | Samuel      | Smith       |
|       8 | God Save the Queen         | Charles     | Jennens     |
|       9 | God Save the King          | Charles     | Jennens     |
|      10 | Greensleeves               | Traditional | Traditional |
|      10 | Greensleeves               | Friedrich   | Schiller    |
+---------+----------------------------+-------------+-------------+
11 rows in set (0.00 sec)
```

// OOPS, looks like I assigned Schiller to Greensleeves, by accident.

```
mysql> SELECT * FROM CreditLyric;
+------------+---------+
| ComposerID | LyricID |
+------------+---------+
|          4 |       1 |
|          5 |       2 |
|          1 |       3 |
|          1 |       4 |
|          1 |       5 |
|          8 |       6 |
|         10 |       7 |
|          8 |       8 |
|          8 |       9 |
|          1 |      10 |
|         15 |      10 |
+------------+---------+
11 rows in set (0.00 sec)
```

// Looks I added two entries for LyricID 10 (Greensleeves).

```
mysql> UPDATE CreditLyric
    -> SET LyricID = 11
    -> WHERE ComposerID = 15;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT Lyric.LyricID, Lyric.LyricTitle, Composer.FirstName, Composer.LastName
FROM Lyric JOIN CreditLyric ON Lyric.LyricID = CreditLyric.LyricID JOIN Composer ON
CreditLyric.ComposerID = Composer.ComposerID WHERE Lyric.LyricID IS NOT NULL;
+---------+----------------------------+-------------+-------------+
| LyricID | LyricTitle                 | FirstName   | LastName    |
+---------+----------------------------+-------------+-------------+
|       1 | Amazing Grace              | John        | Newton      |
|       2 | Battle Hymn of the Republic | Julia       | Howe        |
|       3 | Ave Maria                  | Traditional | Traditional |
|       4 | Miserere mei, Deus         | Traditional | Traditional |
|       5 | Yankee Doodle              | Traditional | Traditional |
|       6 | Messiah, Pt 2 (Hallelujah!) | Charles     | Jennens     |
|       7 | My Country Tis of Thee     | Samuel      | Smith       |
|       8 | God Save the Queen         | Charles     | Jennens     |
|       9 | God Save the King          | Charles     | Jennens     |
|      10 | Greensleeves               | Traditional | Traditional |
|      11 | Ode to Joy                 | Friedrich   | Schiller    |
+---------+----------------------------+-------------+-------------+
11 rows in set (0.00 sec)
```

// Much Better! Now let’s try Music Credits. These should be somewhat different…

```
mysql> SELECT Song.SongID, Song.SongTitle, Composer.FirstName, Composer.LastName
    -> FROM Song
    -> JOIN CreditMusic
    -> ON Song.SongID = CreditMusic.SongID
    -> JOIN Composer
    -> ON CreditMusic.ComposerID = Composer.ComposerID
    -> WHERE Song.SongID IS NOT NULL;
+--------+----------------------------+------------------+-------------+
| SongID | SongTitle                  | FirstName        | LastName    |
+--------+----------------------------+------------------+-------------+
|      1 | Amazing Grace              | Traditional      | Traditional |
|      2 | Battle Hymn of The Republic | William         | Steffe      |
|      3 | Greensleeves               | Traditional      | Traditional |
|      4 | Eine Kleine Nachtmusik     | Wolfgang Amadeus | Mozart      |
|      5 | Ave Maria                  | Franz            | Schubert    |
|      6 | Miserere mei, Deus         | Gregorio         | Allegri     |
|      7 | Four Seasons               | Antonio          | Vivaldi     |
|      8 | Yankee Doodle              | Traditional      | Traditional |
|      9 | Symphony No. 5             | Ludwig van       | Beethoven   |
|     10 | Symphony No. 9             | Ludwig van       | Beethoven   |
|     11 | Messiah                    | George           | Handel      |
|     12 | My Country tis of Thee     | Traditional      | Traditional |
|     13 | God Save The Queen         | Traditional      | Traditional |
|     14 | God Save the King          | Traditional      | Traditional |
+--------+----------------------------+------------------+-------------+
14 rows in set (0.00 sec)
```