Owen Dix

9/25/2015

Normalization, Keys, Relationships, Joins

3.1. Referential integrity is the constraint ensuring that any non-null foreign key in a child table has a matching primary key in the parent table. This protects the relationship between the two tables from:

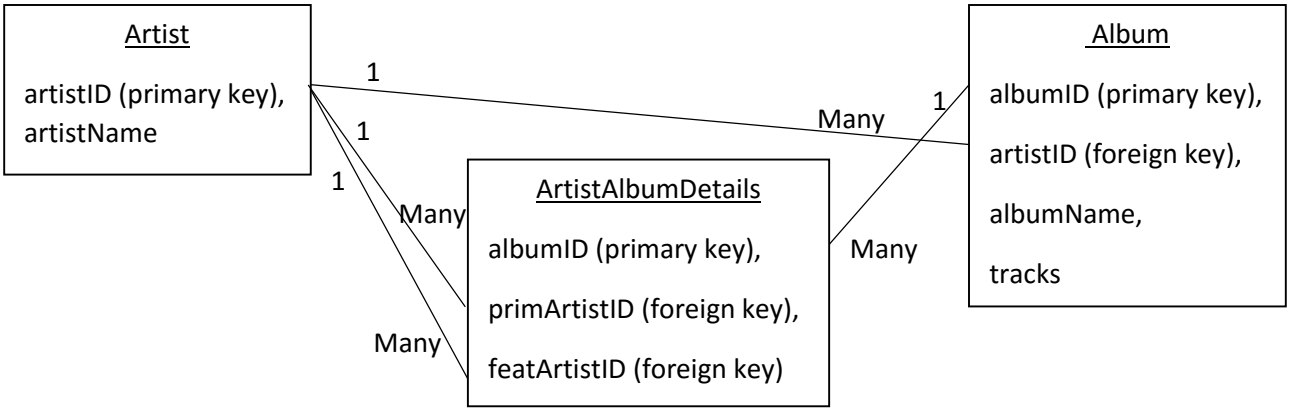
- adding a row with a foreign key value in the child table that doesn't match any corresponding primary key value in the parent table, and from
- deleting or updating a primary key value in the parent table with a matching foreign key value in the child table.

3.2. Since the problem statement says that Employment table's EmployeeNum and PositionID fields make up the (composite) primary key for that table, then the Employment table contains an entity integrity constraint, a type of primary key constraint, which says that the primary key cannot contain a null value. The null value is in the PositionID field on the second tuple for EmployeeNum=4519: (4519, null, 11/11/2007).

The relationship between the Employee table and the Employment table contains a referential integrity constraint, which says that each nonnull foreign key value must match a primary key value in the primary table. Here, the EmployeeNum field in the Employment table contains a value (9876) which does not match any entry in the primary key (EmployeeNum) field in the Employee table.

6.

- Can an artist have one album or many albums?
    - Answer: An artist can have many albums.
- Can an album belong to one artist or many artists?
    - Answer: An album can sometimes include more than one artist.
    - More complete answer: An album usually belongs to one artist but occasionally includes a "featured" artist. I could ignore this but I will allow 1 primary artist and 1 featured artist for a challenge (I will limit the two options to 1 artist each).
    - Since I expect, in practice, few albums have featured artists, including a separate column in the Album table for featured artists would have a lot of null values, I will choose to make a separate table for details about only those albums with featured artists called albumArtistDetails.
    - Note I am not going to breakdown which track has that featured artist in it, just whether they are in the album.
- Can an albumArtistDetails entry have more than one album?
    - Answer: No it has exactly 1 corresponding album.
- Can an albumArtistDetails entry have more than one artist?
    - Answer: Yes it has precisely two artists: one primary artist and one featured artist. This means I need two fields to identify them, a primArtistID and a featArtistID.
- Can an album have more than one albumArtistDetails entry?
    - Answer: No an album has zero or 1 albumArtistDetails entry.
- Can an artist have more than one albumArtistDetails entry?
    - Answer: Yes, an artist can have zero, 1 or many albumArtistDetails entries, either in the primary artist field or featured artist field (or both).



7. Need to create albumArtistDetails:

```
mysql> create table albumArtistDetails(
    -> albumID int unsigned not null,
    -> primArtistID mediumint unsigned not null,
    -> featArtistID mediumint unsigned not null,
    -> PRIMARY KEY albumArtistDetailsPKIdx (albumID),
    -> FOREIGN KEY albumPKIdx (albumID) REFERENCES album (albumID),
    -> FOREIGN KEY primArtistPKIdx (primArtistID) REFERENCES artist (artistID),
    -> FOREIGN KEY featArtistPKIdx (featArtistID) REFERENCES artist (artistID));
Query OK, 0 rows affected (0.17 sec)

mysql>
```

Found that Album's definition of artistID column is not an unsigned mediumint. This is the data type it needs to have to be a foreign key to Artist's primary key artistID. I also need to create the foreign key and want to change the name of a field from trackList to tracks, which is shorter and simpler.

```
mysql> ALTER TABLE Album
    -> CHANGE trackList tracks varchar(200) not null,
    -> MODIFY artistID mediumint unsigned not null,
    -> ADD FOREIGN KEY artistPKIdx (artistID) REFERENCES artist (artistID);
Query OK, 7 rows affected (0.27 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql>
```

8.

```
mysql> desc artist;
+------------+----------------------+------+-----+---------+----------------+
| Field      | Type                 | Null | Key | Default | Extra          |
+------------+----------------------+------+-----+---------+----------------+
| artistID   | mediumint(8) unsigned | NO   | PRI | NULL    | auto_increment |
| artistName | varchar(60)          | NO   |     | NULL    |                |
+------------+----------------------+------+-----+---------+----------------+
2 rows in set (0.02 sec)
```

```
mysql> desc album;
+-----------+----------------------+------+-----+---------+----------------+
| Field     | Type                 | Null | Key | Default | Extra          |
+-----------+----------------------+------+-----+---------+----------------+
| albumID   | int(10) unsigned     | NO   | PRI | NULL    | auto_increment |
| artistID  | mediumint(8) unsigned | NO   | MUL | NULL    |                |
| albumName | varchar(60)          | NO   |     | NULL    |                |
| tracks    | varchar(200)         | NO   |     | NULL    |                |
+-----------+----------------------+------+-----+---------+----------------+
4 rows in set (0.09 sec)
```

```
mysql> desc albumArtistDetails;
+--------------+----------------------+------+-----+---------+-------+
| Field        | Type                 | Null | Key | Default | Extra |
+--------------+----------------------+------+-----+---------+-------+
| albumID      | int(10) unsigned     | NO   | PRI | NULL    |       |
| primArtistID | mediumint(8) unsigned | NO   | MUL | NULL    |       |
| featArtistID | mediumint(8) unsigned | NO   | MUL | NULL    |       |
+--------------+----------------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

9.

```
mysql> insert into albumArtistDetails
    -> SELECT albumID, artistID, 1
    -> FROM Album
    -> WHERE artistID = 2 OR artistID = 3 OR artistID = 4;
Query OK, 3 rows affected (0.13 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> insert into albumArtistDetails
    -> SELECT albumID, artistID, 6
    -> FROM Album
    -> WHERE artistID = 5 OR artistID = 1;
Query OK, 2 rows affected (0.05 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

10.

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

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

mysql> select * from albumArtistDetails;
+---------+--------------+--------------+
| albumID | primArtistID | featArtistID |
+---------+--------------+--------------+
|       1 |            1 |            6 |
|       2 |            2 |            1 |
|       3 |            3 |            1 |
|       4 |            4 |            1 |
|       5 |            5 |            6 |
+---------+--------------+--------------+
5 rows in set (0.00 sec)
```

11. I tried adding an entry into albumArtistDetails that had a featured artist which was not in the Artist table (artistID = 7 is not in Artist). This tests the relationship between the featArtistID field (foreign key) in the

albumArtistDetails table and the artistID field (primary key) in the Artist table. I was denied because of a foreign key constraint, saying I cannot add or update a child row:

```
mysql> insert into albumArtistDetails
    -> SELECT albumID, artistID, 7
    -> FROM Album
    -> WHERE albumID=7;
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`itunesdb`.`al
bumartistdetails`, CONSTRAINT `albumartistdetails_ibfk_3` FOREIGN KEY (`featArtistID`) REFERENCES `a
rtist` (`artistID`))
mysql>
```

I tried adding an entry into albumArtistDetails with an albumID (foreign key) that does not exist in the Album table's albumID field (primary key). I was denied because of a foreign key constraint, saying I cannot add or update a child row:

```
mysql> insert into albumArtistDetails (albumID, primArtistID, featArtistID) values (8, 1, 2);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`itunesdb`.`al
bumartistdetails`, CONSTRAINT `albumartistdetails_ibfk_1` FOREIGN KEY (`albumID`) REFERENCES `album`
 (`albumID`))
mysql>
```

I tried deleting from Album a row with a corresponding entry in the AlbumArtistDetails table. I was denied by a foreign key constraint saying I cannot delete or update a parent row:

```
mysql> delete from Album where albumID=4;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`itunesdb`
.`albumartistdetails`, CONSTRAINT `albumartistdetails_ibfk_1` FOREIGN KEY (`albumID`) REFERENCES `al
bum` (`albumID`))
mysql>
```

I added a row to Artist and Album, where the artist had no entry in AlbumArtistDetails (no featured or featuring albums). This way I can test just the relationship between Artist and Album via the artistID field. I then tried to delete from Artist this newly added entry with artistID=7 and was denied by the foreign key constraint saying I cannot delete or update a parent row:

```
mysql> select * from artist;
+----------+-------------------------+
| artistID | artistName              |
+----------+-------------------------+
|        1 | aarons, The             |
|        2 | aardvarks, The          |
|        3 | applesauce metalheads   |
|        4 | badger barbers          |
|        5 | camel captains          |
|        6 | doggy destroyers        |
|        7 | elephant eruptions, the |
+----------+-------------------------+
7 rows in set (0.00 sec)

mysql> insert into Album (artistID, albumName, tracks) values (7, 'Kaboomage', 'Blow the nose, swat some flies, st
omp stomp stomp, eat and drink');
Query OK, 1 row affected (0.09 sec)

mysql> select * from album;
+---------+----------+--------------------+-----------------------------------------------------------------+
| albumID | artistID | albumName          | tracks                                                          |
+---------+----------+--------------------+-----------------------------------------------------------------+
|       1 |        1 | Hi Aaron           | aaron1, aaron2, aaron3                                          |
|       2 |        2 | Rolling Strong     | we roll, we rock, we eat stuff                                  |
|       3 |        3 | The Spicy Kind     | motts, umm, some other applesauce...with metal in it            |
|       4 |        4 | Grrr               | hairy mammal, i eat, i mate, i reproduce, im angry              |
|       5 |        5 | The Two Hump Kind  | no water, we spit                                               |
|       6 |        6 | We Rock            | bark, ruff, scratch, lick, eat, smile                           |
|       7 |        6 | We Roll...in Stuff | happy to see you - single                                       |
|       8 |        7 | Kaboomage          | Blow the nose, swat some flies, stomp stomp stomp, eat and drink |
+---------+----------+--------------------+-----------------------------------------------------------------+
8 rows in set (0.00 sec)

mysql> delete from artist where artistID=7;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`itunesdb`.`album`, CONS
TRAINT `album_ibfk_1` FOREIGN KEY (`artistID`) REFERENCES `artist` (`artistID`))
mysql>
```

12. Questions to ask of database:

1. How many times are the doggy destroyers featured artists on another album?
2. How many times do the badger barbers feature another artist on their albums?
3. What are the names of the albums that feature another artist?
4. What are the names of the albums that do not feature another artist?
5. What are the artist names and album names of “single” albums: albums with just one track (tracks are separated by commas so the tracks field will have no commas in it)?

13.

1. How many times are the doggy destroyers featured artists on another album?

```
mysql> SELECT count(*)
    -> FROM artist a, albumArtistDetails aad
    -> WHERE a.artistID = aad.featArtistID AND a.artistName LIKE 'doggy destroyers%';
+----------+
| count(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)
```

2. How many times do the badger barbers feature another artist on their albums?

```
mysql> SELECT count(*)
    -> FROM artist a, albumArtistDetails aad
    -> WHERE a.artistID = aad.primArtistID AND a.artistName LIKE 'badger barbers%';
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)
```

3. What are the names of the albums that feature another artist?

```
mysql> SELECT a.albumName
    -> FROM album a, albumArtistDetails aad
    -> WHERE a.artistID = aad.primArtistID;
+-------------------+
| albumName         |
+-------------------+
| Hi Aaron          |
| Rolling Strong    |
| The Spicy Kind    |
| Grrr              |
| The Two Hump Kind |
+-------------------+
5 rows in set (0.00 sec)
```

4. What are the names of the albums that do not feature another artist?

```
mysql> SELECT albumName
    -> FROM album
    -> WHERE albumID NOT IN (SELECT albumID FROM albumArtistDetails);
+--------------------+
| albumName          |
+--------------------+
| We Rock            |
| We Roll...in Stuff |
| Kaboomage          |
+--------------------+
3 rows in set (0.11 sec)
```

5. What are the artist names and album names of “single” albums: albums with just one track (tracks are separated by commas so the tracks field will have no commas in it)?

```
mysql> SELECT artistName, albumName
    -> FROM artist, album
    -> WHERE artist.artistID = album.artistID AND tracks NOT LIKE '%,%';
+------------------+--------------------+
| artistName       | albumName          |
+------------------+--------------------+
| doggy destroyers | We Roll...in Stuff |
+------------------+--------------------+
1 row in set (0.00 sec)
```

14.

4. What are the names of the albums that do not feature another artist?

```
mysql> SELECT albumName
    -> FROM album a
    -> LEFT JOIN albumArtistDetails aad
    -> ON a.albumID = aad.albumID
    -> WHERE aad.albumID IS NULL;
+--------------------+
| albumName          |
+--------------------+
| We Rock            |
| We Roll...in Stuff |
| Kaboomage          |
+--------------------+
3 rows in set (0.00 sec)
```

15. N/A

16. In this activity, I learned how to better set up a database so it avoids anomalies and conforms to the first three normal forms. I learned about how to establish relationships implementing foreign key constraints. I learned how to join tables using the simple query standards and Standard SQL join syntax. I think this is extremely important to the database world. The ability to efficiently design and retrieve information from a database is critical to using that data effectively.