9/19/2019

Kevin Vogel

Record Collection

IST 659 Final Project

Table of Contents

**Project summary ………………………………………………………………………………………………………………………………. 2**

**Business Case ……..…………………………………………………………………………………………………………………………. 2**

**Stakeholders …………………………………………………………………………………………………………………………………. 2**

**Goal ………………………………………………………………………………………………………………………………………………. 2**

**Reflection …………………………………………………………………………………………………………………………………………. 2**

**Conceptual Model ……………………………………………………………………………………………………………………………. 3**

**Conceptual Model (table) .……………………………………………………………………………………………………………. 3**

**Normalized Logical Model ………………………………………………………………………………………………………………… 4**

**Normalized Logical Model (table) .………………………………………………………………………………………………… 5**

**Explanation of Necessary Changes .………………………………………………………………………………………………. 5**

**Physical Database Design ………….……………………………………………………………………………………………………… 6**

**Table Creation ………………………….…………………………………………………………………………………………………… 6**

**Data Entry .……………………………………………………………………………………………………………………………………. 8**

**Completed Tables ………………………………………………………………………………………………………………………... 22**

**Insert, Update, Delete Statements …………………………………………………………………………………………………… 26**

**Created queries, views, etc. ….…………………………………………………………………………………………………………. 29**

**Security …………………………….….…………………………………………………………………………………………………………. 36**

**Implementation ………………………………………………………………………………………………………………………………. 38**

Summary:

My father has what he calls “a small record collection.” He has been collecting – buying and selling – vinyl records since he was a child. He has amassed a significant number of records; over 6,000 in total. They are arranged on shelves alphabetically by artist – this is the extent of the cataloguing he has done.

Some of the albums he owns are worth considerable amounts of money; the value of others is entirely intrinsic to him. For these reasons and more, it is necessary and prudent to catalogue his collection in a database. As he continues to purchase and occasionally sell records, he attempts to minimize duplicate copies while ensuring he does not sell his only copy of a record – this has proven a difficult task.

Providing him with the utility of searching his personal vinyl record database **before** a purchase or sale is what creates value for him in this project.

Should he pass away before creating a functional database, whomever he leaves his record collection to will have little to no understanding of the value of the records. I’m generally not a fan of probate court so it would be fantastic if we already had this squared away.

Business Rules:

* Artists may perform both as an individual and as part of a group.
* Just because an artist is featured on a track on an album does not mean that they are an artist on the album. The album artist for this dataset is dictated by discogs.com.

Stakeholders:

* My father
  + While not technically savvy, he can manage simple data entry as long as tasks require minimal competence. GUI will be required for use.
* His heir(s) if he does not liquidate the collection before his death
  + No additional requirements are currently being made on behalf of other stakeholders.

Goal:

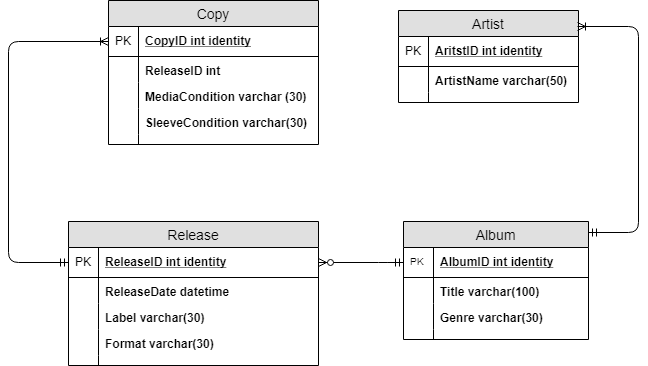
Upon completion of this project, a query should determine:

1. How many copies of an album exist in his collection
2. The condition of a piece of media
3. Whether or not the original sleeve for the media exists and its condition
4. Summary information by artist, genre, record label, etc.

Reflection:

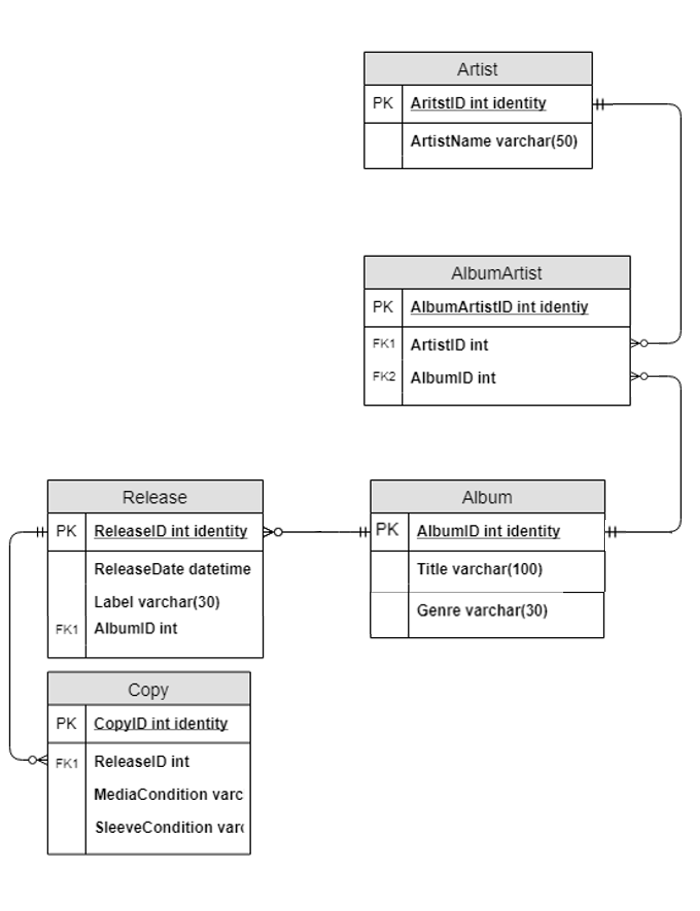
* This project has been useful in that it has taught me the basics of database management. Coming in to this course, I had never used SQL and now I am able to create tables, create useful queries (that actually query the information that I want them to, store those queries as procedures, and create a front end in the form of an Access form for the end user to be able to interact with the data.
* With more time, everything would be set up in Access. For now, I believe this is minimally viable for adding his collection to the database.

Conceptual model:

****

Conceptual Model (table):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Column Name** | **Description** | **Data** | **Length** | **Key** | **Unique** |
| Artist | ArtistID | ID of the artist | Int, ID |  | PK | True |
|  | ArtistName | Name of the artist | Text | 50 |  | False |
| Album | AlbumID | ID of the album | Int, ID |  | PK | True |
|  | AlbumName | Name of the Album | Text | 100 |  | False |
|  | Genre | Genre of the album | Text | 30 |  | False |
| Copy | CopyID | ID of the copy | Int, ID |  | PK | True |
|  | ReleaseID | What release is this a copy of | Int |  |  | False |
|  | MediaCondition | Condition of the media | Text | 30 |  | False |
|  | SleeveCondition | Condition of the sleeve | Text | 30 |  | False |
| Release | ReleaseID | ID of the release | Int, ID |  |  | True |
|  | Label | Label that released that album | Text | 30 |  | False |
|  | ReleaseDate | Title release date of an album release | DateTime |  |  | False |
|  | Format | The format that the media is in | Text | 30 |  | False |

Normalized Logical Model: 

Normalized Logical Model (table):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity** | **Column Name** | **Description** | **Data** | **Length** | **Key** | **Unique** |
| Artist | ArtistID | ID of the artist | Int, ID |  | PK | True |
|  | ArtistName | Name of the artist | Text | 50 |  | False |
| Album | AlbumID | ID of the album | Int, ID |  | PK | True |
|  | Title | Name of the Album | Text | 100 |  | False |
|  | Genre | Genre of the album | Text | 30 |  | False |
| Copy | CopyID | ID of the copy | Int, ID |  | PK | True |
|  | ReleaseID | What release is this a copy of | Int |  |  | False |
|  | MediaCondition | Condition of the media | Text | 30 |  | False |
|  | SleeveCondition | Condition of the sleeve | Text | 30 |  | False |
| Release | ReleaseID | ID of the release | Int, ID |  |  | True |
|  | Label | Label that released that album | Text | 30 |  | False |
|  | ReleaseDate | Title release date of an album release | DateTime |  |  | False |

Explanation of Necessary Changes:

In order to move from the conceptual model to the logical model I had to create additional entities. These entities were many to many relationships, so bridge relationships had to be created as well. It is possible that many of his albums he may only have one release of an album, but he has it in multiple formats (reel to reel, LP, etc.). Thus, this needed to be its own entity. I have since decided that this will database will only be for LP’s, so this is no longer necessary. Separating release from album as its own entity was necessary as well. If my father ever finished cataloguing his albums, then he might move on to adding song titles. As you well know, a bonus track version of an album or a rerelease of an album may have an additional song or two on it. Rereleases also have very different values, so if he wanted to add value in the form of a book price or a sale price into this database, it is well set up to do this.

Physical Database Design:

--THIS SECTION WILL BE USED TO MAKE TABLES

-- Creating the Album Table

Create Table Album (

-- Columns for the Album table

AlbumID int identity,

Title varchar(100) not null,

Genre varchar(30) not null,

-- Constraints on the Album table

CONSTRAINT PK\_Album PRIMARY KEY (AlbumID)

)

--End creating the Album table

-- Creating the Release Table

Create Table Release (

-- Columns for the Release table

ReleaseID int identity,

ReleaseDate datetime not null,

Label varchar(30) not null,

AlbumID int not null,

-- Constraints on the Release table

CONSTRAINT PK\_Release PRIMARY KEY (ReleaseID),

CONSTRAINT FK1\_Release FOREIGN KEY (AlbumID) REFERENCES Album(AlbumID)

)

--End creating the Release table

-- Creating the Copy Table

Create Table Copy (

-- Columns for the Copy table

CopyID int identity,

ReleaseID int not null,

MediaCondition varchar(30) not null,

SleeveCondition varchar(30) not null,

-- Constraints on the Copy table

CONSTRAINT PK\_Copy PRIMARY KEY (CopyID),

CONSTRAINT FK1\_Copy FOREIGN KEY (ReleaseID) REFERENCES Release(ReleaseID)

)

--End creating the Copy table

-- Creating the Artist Table

Create Table Artist (

-- Columns for the Artist table

ArtistID int identity,

ArtistName varchar(50) not null,

-- Constraints on the Album table

CONSTRAINT PK\_Artist PRIMARY KEY (ArtistID)

)

--End creating the Artist table

--Creating the AlbumArtist table

Create table AlbumArtist (

--columns for the AlbumArtist table

AlbumArtistID int identity,

ArtistID int not null,

AlbumID int not null,

--constraints for the AlbumArtist table

CONSTRAINT PK\_AlbumArtist Primary Key (AlbumArtistID),

CONSTRAINT U1\_AlbumArtist Unique (ArtistID, AlbumID),

CONSTRAINT FK1\_AlbumArtist Foreign Key (ArtistID) References Artist(ArtistID),

CONSTRAINT FK2\_AlbumArtist Foreign Key (AlbumID) References Album(AlbumID)

)

--end creating the AlbumArtist table

Data entry:

--THIS SECTION WILL BE USED TO ENTER DATA INTO THE ALBUM TABLE

Insert into Album (Title, Genre)

Values ('The Album', 'Pop')

Insert into Album (Title, Genre)

Values ('Voulez-Vous', 'Pop')

Insert into Album (Title, Genre)

Values ('Arrival', 'Pop')

Insert into Album (Title, Genre)

Values ('Dirty Deeds Done Dirt Cheap', 'Rock')

Insert into Album (Title, Genre)

Values ('Who Made Who', 'Rock')

Insert into Album (Title, Genre)

Values ('Amazing', 'Rock')

Insert into Album (Title, Genre)

Values ('Aerosmith Greatest Hits', 'Rock')

Insert into Album (Title, Genre)

Values ('Old Flame', 'Country')

Insert into Album (Title, Genre)

Values ('Born Under a Bad Sign', 'Blues')

Insert into Album (Title, Genre)

Values ('Lovejoy', 'Blues')

Insert into Album (Title, Genre)

Values ('Billion Dollar Babies', 'Rock')

Insert into Album (Title, Genre)

Values ('Muscle of Love', 'Rock')

Insert into Album (Title, Genre)

Values ('Killer', 'Rock')

Insert into Album (Title, Genre)

Values ('Schools Out', 'Rock')

Insert into Album (Title, Genre)

Values ('Live Concert at the Forum', 'Easy Listening')

Insert into Album (Title, Genre)

Values ('Guilty', 'Easy Listening')

Insert into Album (Title, Genre)

Values ('The Way We Were', 'Easy Listening')

Insert into Album (Title, Genre)

Values ('The Broadway Album', 'Easy Listening')

Insert into Album (Title, Genre)

Values ('Main Course', 'Pop')

Insert into Album (Title, Genre)

Values ('Trafalgar', 'Pop')

Insert into Album (Title, Genre)

Values ('Turnstiles', 'Pop')

Insert into Album (Title, Genre)

Values ('Songs in the Attic', 'Pop')

Insert into Album (Title, Genre)

Values ('The Stranger', 'Pop')

Insert into Album (Title, Genre)

Values ('Keeping the Faith', 'Pop')

Insert into Album (Title, Genre)

Values ('Glass Houses', 'Pop')

Insert into Album (Title, Genre)

Values ('Streetlife Serenade', 'Pop')

Insert into Album (Title, Genre)

Values ('52nd Street', 'Pop')

Insert into Album (Title, Genre)

Values ('Piano Man', 'Pop')

Insert into Album (Title, Genre)

Values ('Biograph', 'Folk')

Insert into Album (Title, Genre)

Values ('Highway 61 Revisited', 'Folk')

Insert into Album (Title, Genre)

Values ('A Fire I Cant Put Out', 'Country')

Insert into Album (Title, Genre)

Values ('Greatest Hits', 'Country')

Insert into Album (Title, Genre)

Values ('Does Fort Worth Ever Cross Your Mind', 'Country')

Insert into Album (Title, Genre)

Values ('Life to Legend', 'Country')

Insert into Album (Title, Genre)

Values ('The Essential Hank Williams', 'Country')

Insert into Album (Title, Genre)

Values ('The Very Best of Hank Williams', 'Country')

--THIS SECTION WILL BE USED TO ENTER DATA INTO THE ARTIST TABLE

Insert into Artist (ArtistName)

Values

('ABBA'),

('AC/DC'),

('Aerosmith'),

('Alabama'),

('Albert King'),

('Alice Cooper'),

('Barbara Streisand'),

('Bee Gees'),

('Billy Joel'),

('Bob Dylan'),

('George Strait'),

('Hank Williams')

--INSERTING INTO ALBUMARTIST TABLE

INSERT INTO AlbumArtist (ArtistID, AlbumID)

VALUES

((SELECT ArtistID FROM Artist WHERE ArtistName='ABBA'),

(SELECT AlbumID FROM Album WHERE Title='The Album')),

((SELECT ArtistID FROM Artist WHERE ArtistName='ABBA'),

(SELECT AlbumID FROM Album WHERE Title='Voulez-Vous')),

((SELECT ArtistID FROM Artist WHERE ArtistName='ABBA'),

(SELECT AlbumID FROM Album WHERE Title='Arrival')),

((SELECT ArtistID FROM Artist WHERE ArtistName='AC/DC'),

(SELECT AlbumID FROM Album WHERE Title='Dirty Deeds Done Dirt Cheap')),

((SELECT ArtistID FROM Artist WHERE ArtistName='AC/DC'),

(SELECT AlbumID FROM Album WHERE Title='Who Made Who')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Aerosmith'),

(SELECT AlbumID FROM Album WHERE Title='Amazing')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Aerosmith'),

(SELECT AlbumID FROM Album WHERE Title='Aerosmith Greatest Hits')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Alabama'),

(SELECT AlbumID FROM Album WHERE Title='Old Flame')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Albert King'),

(SELECT AlbumID FROM Album WHERE Title='Born Under A Bad Sign')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Albert King'),

(SELECT AlbumID FROM Album WHERE Title='Lovejoy')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Alice Cooper'),

(SELECT AlbumID FROM Album WHERE Title='Schools Out')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Alice Cooper'),

(SELECT AlbumID FROM Album WHERE Title='Billion Dollar Babies')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Alice Cooper'),

(SELECT AlbumID FROM Album WHERE Title='Muscle Of Love')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Alice Cooper'),

(SELECT AlbumID FROM Album WHERE Title='Killer')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Barbara Streisand'),

(SELECT AlbumID FROM Album WHERE Title='Live Concert At The Forum')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Barbara Streisand'),

(SELECT AlbumID FROM Album WHERE Title='Guilty')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Barbara Streisand'),

(SELECT AlbumID FROM Album WHERE Title='The Way We Were')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Barbara Streisand'),

(SELECT AlbumID FROM Album WHERE Title='The Broadway Album')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Bee Gees'),

(SELECT AlbumID FROM Album WHERE Title='Main Course')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Bee Gees'),

(SELECT AlbumID FROM Album WHERE Title='Trafalgar')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='Turnstiles')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='Songs In The Attic')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='The Stranger')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='Keeping The Faith')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='Glass Houses')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='Streetlife Serenade')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='52nd Street')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Billy Joel'),

(SELECT AlbumID FROM Album WHERE Title='Piano Man')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Bob Dylan'),

(SELECT AlbumID FROM Album WHERE Title='Biograph')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Bob Dylan'),

(SELECT AlbumID FROM Album WHERE Title='Highway 61 Revisited')),

((SELECT ArtistID FROM Artist WHERE ArtistName='George Strait'),

(SELECT AlbumID FROM Album WHERE Title='A Fire I Cant Put Out')),

((SELECT ArtistID FROM Artist WHERE ArtistName='George Strait'),

(SELECT AlbumID FROM Album WHERE Title='Greatest Hits')),

((SELECT ArtistID FROM Artist WHERE ArtistName='George Strait'),

(SELECT AlbumID FROM Album WHERE Title='Does Fort Worth Ever Cross Your Mind')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Hank Williams'),

(SELECT AlbumID FROM Album WHERE Title='Life To Legend')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Hank Williams'),

(SELECT AlbumID FROM Album WHERE Title='The Essential Hank Williams')),

((SELECT ArtistID FROM Artist WHERE ArtistName='Hank Williams'),

(SELECT AlbumID FROM Album WHERE Title='The Very Best Of Hank Williams'))

-------------------------------------------------------------------------------

-- MAKING SURE THAT I DID THAT CORRECTLY

Select

Artist.ArtistName,

Album.Title,

Album.Genre

From AlbumArtist

JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

Order By Artist.ArtistName, Album.Title

-- I DID, WOOT.

-------------------------------------------------------------------------------

-- INSERTING INTO RELEASE

INSERT INTO Release (ReleaseDate, Label, AlbumID)

VALUES

(('1976'),

('Atlantic'),

(SELECT AlbumID FROM Album WHERE Title='Arrival'))

-- TRYING OUT INSERT

SELECT Year(ReleaseDate) AS Year from Release

-- WOOT

-- NOW TO DO THE REST

INSERT INTO Release (ReleaseDate, Label, AlbumID)

VALUES

(('1976'),

('Atlantic'),

(SELECT AlbumID FROM Album WHERE Title='Arrival')),

(('1977'),

('Atlantic'),

(SELECT AlbumID FROM Album WHERE Title='The Album')),

(('1979'),

('Atlantic'),

(SELECT AlbumID FROM Album WHERE Title='Voulez-Vous')),

(('1981'),

('Atlantic'),

(SELECT AlbumID FROM Album WHERE Title='Dirty Deeds Done Dirt Cheap')),

(('1986'),

('Atlantic'),

(SELECT AlbumID FROM Album WHERE Title='Who Made Who')),

(('1980'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Aerosmith Greatest Hits')),

(('1993'),

('Geffen Records'),

(SELECT AlbumID FROM Album WHERE Title='Amazing')),

(('1981'),

('RCA'),

(SELECT AlbumID FROM Album WHERE Title='Old Flame')),

(('1998'),

('Sundazed Music'),

(SELECT AlbumID FROM Album WHERE Title='Born Under a Bad Sign')),

(('2016'),

('Stax'),

(SELECT AlbumID FROM Album WHERE Title='Lovejoy')),

(('1973'),

('Warner Bros. Records'),

(SELECT AlbumID FROM Album WHERE Title='Billion Dollar Babies')),

(('2016'),

('Warner Bros. Records'),

(SELECT AlbumID FROM Album WHERE Title='Billion Dollar Babies')),

(('2012'),

('Friday Music'),

(SELECT AlbumID FROM Album WHERE Title='Killer')),

(('1973'),

('Warner Bros. Records'),

(SELECT AlbumID FROM Album WHERE Title='Muscle of Love')),

(('2013'),

('Friday Music'),

(SELECT AlbumID FROM Album WHERE Title='Muscle of Love')),

(('1972'),

('Warner Bros. Records'),

(SELECT AlbumID FROM Album WHERE Title='Schools Out')),

(('1973'),

('Warner Bros. Records'),

(SELECT AlbumID FROM Album WHERE Title='Schools Out')),

(('1980'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Guilty')),

(('1972'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Live Concert at the Forum')),

(('1985'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='The Broadway Album')),

(('1974'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='The Way We Were')),

(('1975'),

('RSO'),

(SELECT AlbumID FROM Album WHERE Title='Main Course')),

(('1971'),

('ATCO Records'),

(SELECT AlbumID FROM Album WHERE Title='Trafalgar')),

(('1978'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='52nd Street')),

(('1980'),

('CBS'),

(SELECT AlbumID FROM Album WHERE Title='Glass Houses')),

(('1984'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Keeping the Faith')),

(('1976'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Piano Man')),

(('1981'),

('CBS'),

(SELECT AlbumID FROM Album WHERE Title='Songs in the Attic')),

(('1974'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Streetlife Serenade')),

(('1977'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='The Stranger')),

(('1976'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Turnstiles')),

(('1985'),

('Columbia'),

(SELECT AlbumID FROM Album WHERE Title='Biograph')),

(('2018'),

('Sony Records International'),

(SELECT AlbumID FROM Album WHERE Title='Highway 61 Revisited')),

(('1982'),

('MCA Records'),

(SELECT AlbumID FROM Album WHERE Title='A Fire I Cant Put Out')),

(('1984'),

('MCA Records'),

(SELECT AlbumID FROM Album WHERE Title='Does Fort Worth Ever Cross Your Mind')),

(('1985'),

('MCA Records'),

(SELECT AlbumID FROM Album WHERE Title='Greatest Hits')),

(('1970'),

('MGM Records'),

(SELECT AlbumID FROM Album WHERE Title='Life to Legend')),

(('1969'),

('MGM Records'),

(SELECT AlbumID FROM Album WHERE Title='The Essential Hank Williams')),

(('1976'),

('Polydor'),

(SELECT AlbumID FROM Album WHERE Title='The Very Best of Hank Williams'))

-------------------------------------------------------------------------------

-- INSERTING INTO COPY

INSERT INTO Copy(ReleaseID, MediaCondition, SleeveCondition)

VALUES

((SELECT ReleaseID FROM Release WHERE ReleaseID='1'),

('VG'),

('NO')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='2'),

('NM'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='3'),

('M'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='4'),

('VG+'),

('G')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='5'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='6'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='7'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='8'),

('NM'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='9'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='10'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='11'),

('VG'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='12'),

('VG'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='13'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='14'),

('NM'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='15'),

('M'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='16'),

('VG+'),

('NO')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='17'),

('NM'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='18'),

('VG'),

('G')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='19'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='20'),

('VG+'),

('G')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='21'),

('M'),

('M')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='22'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='23'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='24'),

('NM'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='25'),

('VG'),

('G')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='26'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='27'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='28'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='29'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='30'),

('NM'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='31'),

('VG'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='32'),

('VG+'),

('G')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='33'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='34'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='35'),

('VG+'),

('G')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='36'),

('VG+'),

('NO')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='37'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='38'),

('VG+'),

('VG+')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='39'),

('VG+'),

('G')),

((SELECT ReleaseID FROM Release WHERE ReleaseID='40'),

('NM'),

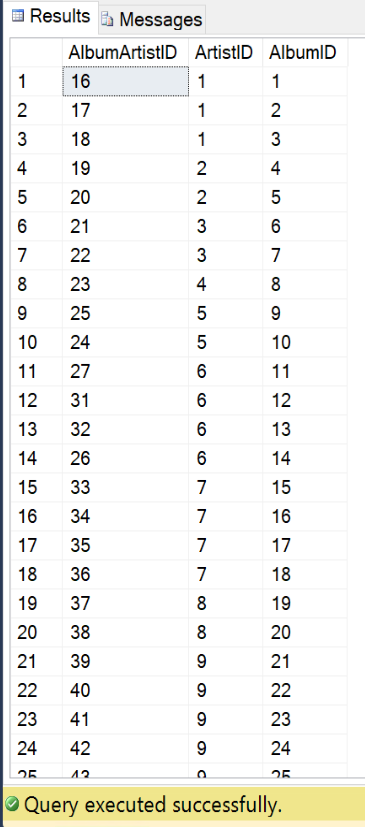
('M'))

Created Tables

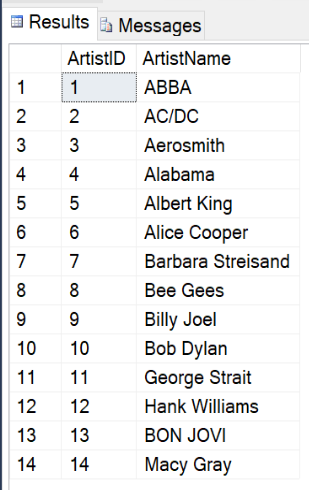
album



AlbumArtist



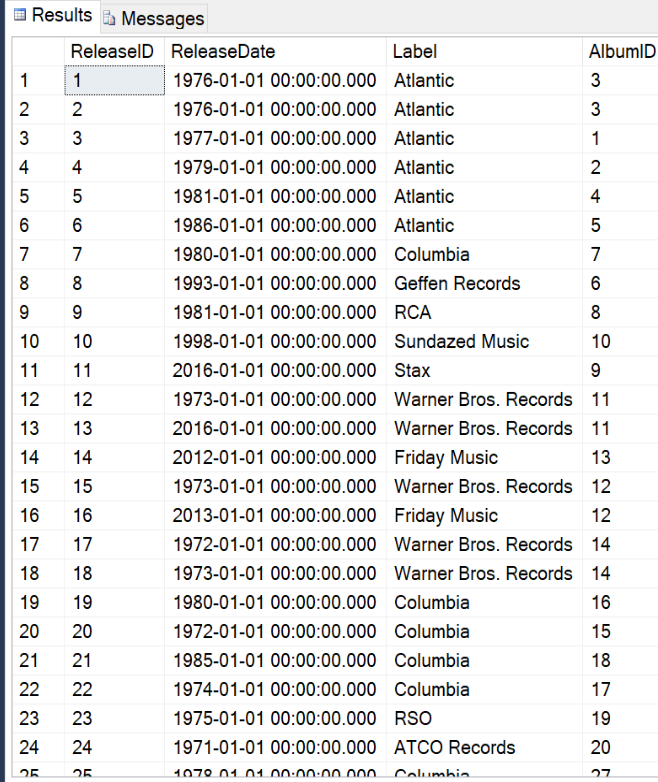
Artist



copy



Release



INSERT

CREATE PROCEDURE AddArtist(@ArtistName varchar(50)) AS

BEGIN

IF NOT EXISTS (SELECT 1 FROM Artist

WHERE ArtistName = @ArtistName)

BEGIN

INSERT INTO Artist(ArtistName)

Values (@ArtistName)

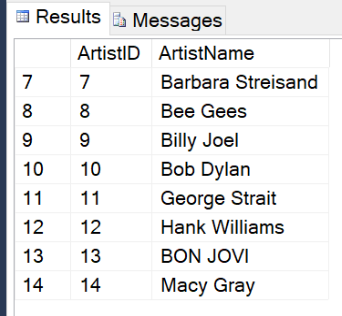
RETURN @@IDENTITY

END

END

EXEC AddArtist 'BON JOVI'

select\* from artist



-----------------------------------------------------------

CREATE PROCEDURE AddAlbum(@Title varchar(100), @Genre varchar(30)) AS

BEGIN

IF NOT EXISTS (SELECT 1 FROM Album

WHERE Title = @Title)

BEGIN

INSERT INTO Album(Title, Genre)

Values (@Title, @Genre)

RETURN @@IDENTITY

END

END

Exec AddAlbum 'NewTitle', 'NewAge'

select \* from album



CREATE PROCEDURE AddRelease(@ReleaseDate datetime, @Label varchar(30), @Title varchar(100)) AS

BEGIN

INSERT INTO Release (ReleaseDate, Label, AlbumID)

VALUES

((@ReleaseDate),

(@Label),

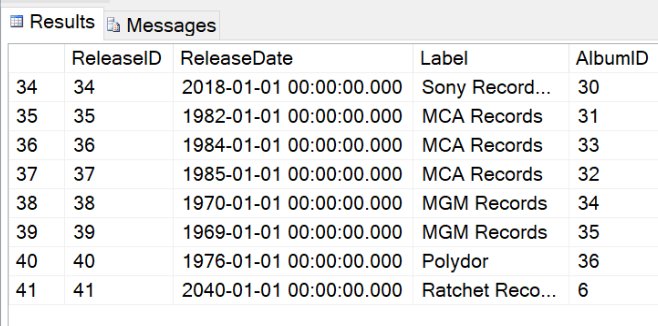
(SELECT AlbumID FROM Album WHERE Title= @Title))

RETURN @@IDENTITY

END

Exec AddRelease '2040', 'Ratchet Records', 'Amazing'

select \* from Release



DELETE

--made a delete statement for artist

Exec AddArtist 'xyz'

Select\* from Artist

delete from Artist where ArtistName = 'xyz'

Select\* from Artist



--deleting NewTitle from Albums

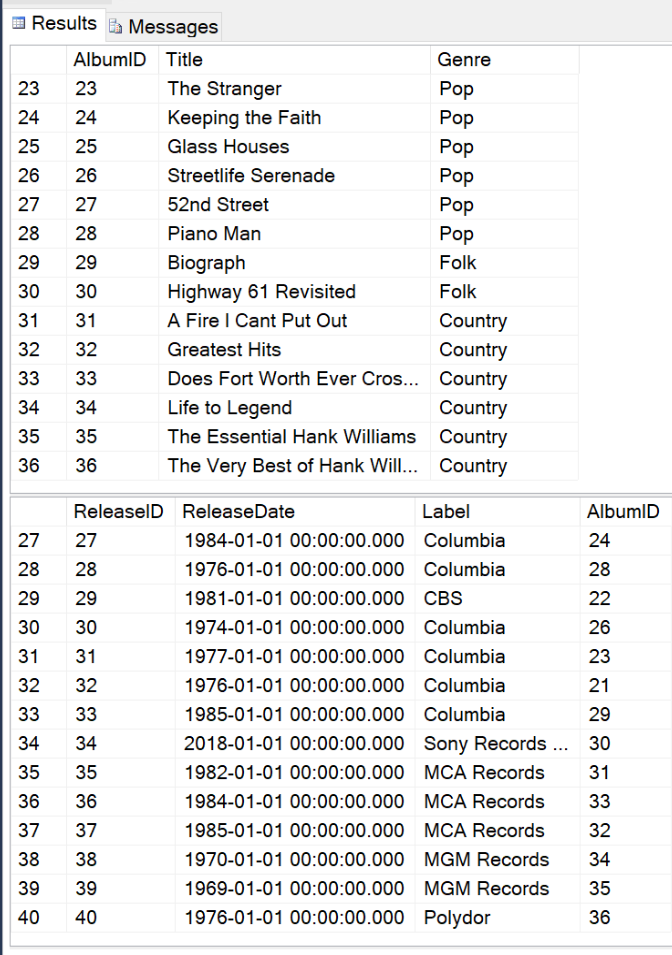
Delete Album where Title = 'NewTitle'

Select \* From Album

--deleting ratchet records release from the future

Delete Release where ReleaseID = 41

Select \* from Release



Update

--Correcting the case on bon jovi

UPDATE Artist

SET ArtistName = 'Bon Jovi'

WHERE Artist.ArtistName = 'BON JOVI'



Created queries, views, etc:

-------------------------------------------------------------------------------

--Selects ERRTHANG

Select

Artist.ArtistName AS Artist,

Album.Title,

Album.Genre,

Year(Release.ReleaseDate) AS Year,

Release.Label,

Copy.MediaCondition,

Copy.SleeveCondition

From AlbumArtist

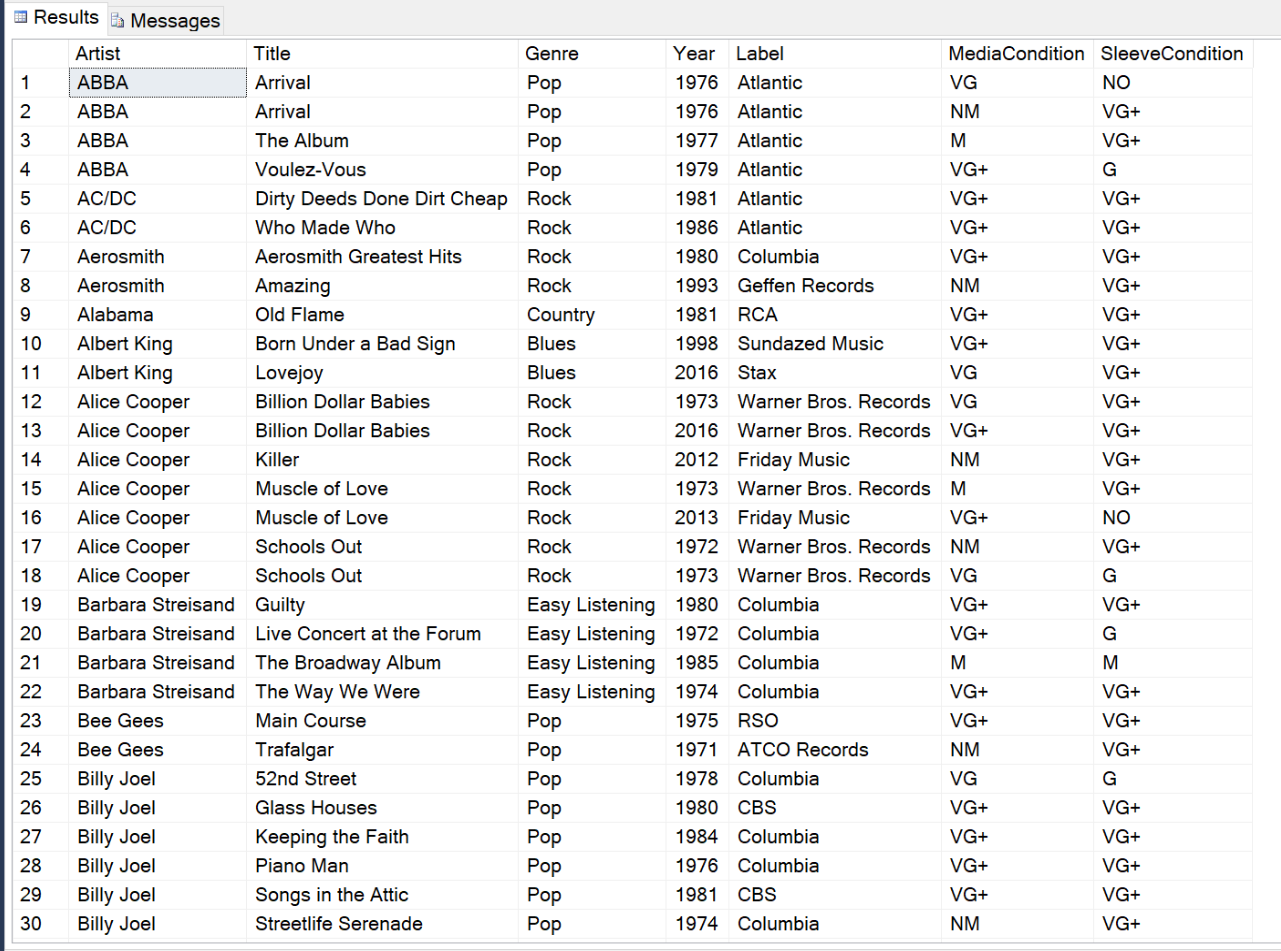
JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

RIGHT JOIN Release ON Album.AlbumID = Release.AlbumID

JOIN Copy ON Release.ReleaseID = Copy.ReleaseID

Order By Artist.ArtistName, Album.Title



-------------------------------------------------------------------------------

--Creates View that Selects ERRTHANG

CREATE VIEW ERRTHANG AS

Select

Artist.ArtistName AS Artist,

Album.Title AS Title,

Album.Genre,

Year(Release.ReleaseDate) AS Year,

Release.Label,

Copy.MediaCondition,

Copy.SleeveCondition

From AlbumArtist

JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

RIGHT JOIN Release ON Album.AlbumID = Release.AlbumID

JOIN Copy ON Release.ReleaseID = Copy.ReleaseID

GO

SELECT \* FROM ERRTHANG

Order By Artist, Title



-------------------------------------------------------------------------------

--Creates view that shows CopyCountByArtist

Create VIEW CopyCountByArtist AS

Select

Artist.ArtistName AS Artist,

Count(Album.Title) AS NumberOfCopies

From AlbumArtist

JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

RIGHT JOIN Release ON Album.AlbumID = Release.AlbumID

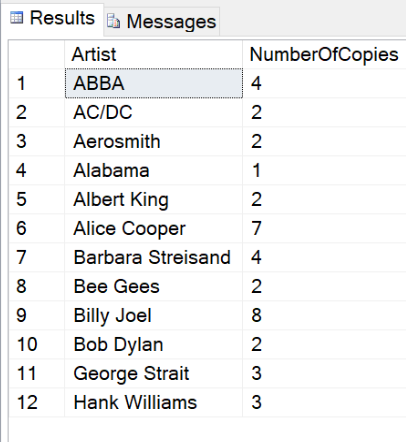
JOIN Copy ON Release.ReleaseID = Copy.ReleaseID

group by ArtistName

GO

SELECT \* FROM CopyCountByArtist

Order By Artist



-------------------------------------------------------------------------------

--Creates view that shows AlbumCountByArtist

Create VIEW AlbumCountByArtist AS

Select

Artist.ArtistName AS Artist,

Count(Album.Title) AS NumberOfAlbums

From AlbumArtist

JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

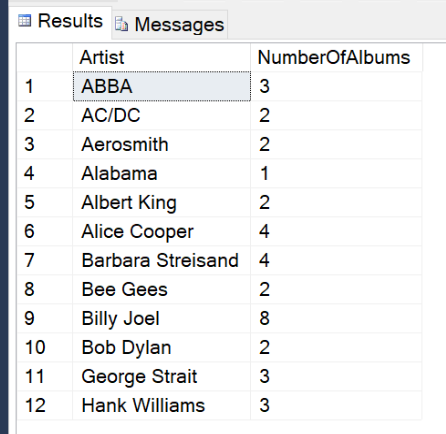
JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

group by ArtistName

GO

SELECT \* FROM AlbumCountByArtist

Order By Artist



-------------------------------------------------------------------------------

--Creates view that shows CopyCountByAlbumAndYear

Create VIEW CopyCountByAlbumAndYear AS

Select

Artist.ArtistName AS Artist,

Album.Title as Title,

Album.Genre,

Year(Release.ReleaseDate) AS Year,

Count(Album.Title) AS NumberOfCopies

From AlbumArtist

JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

RIGHT JOIN Release ON Album.AlbumID = Release.AlbumID

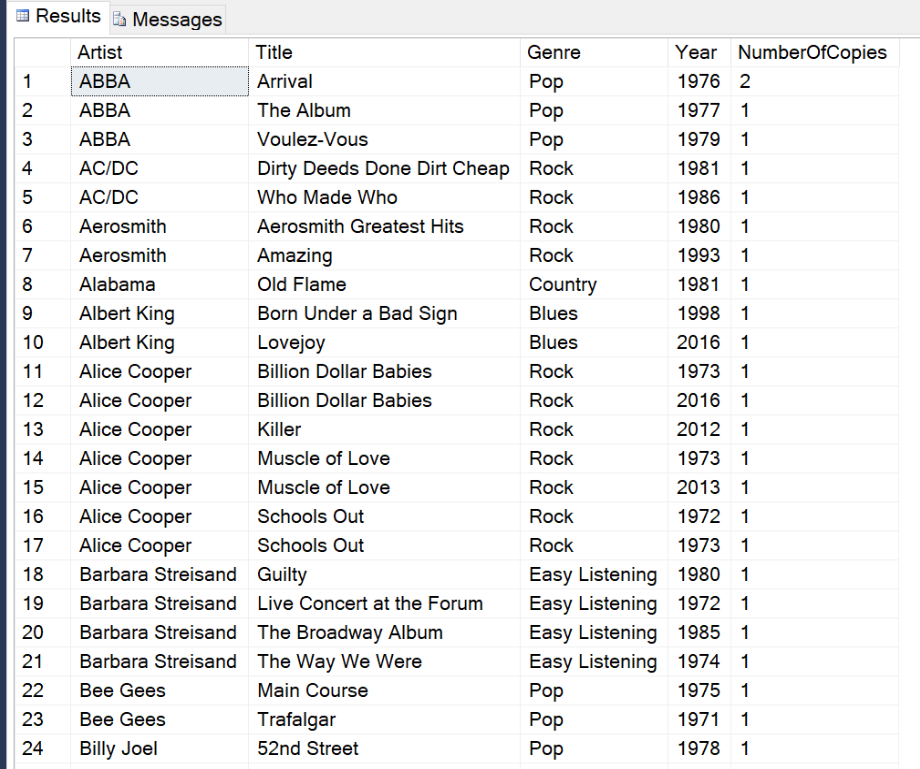
JOIN Copy ON Release.ReleaseID = Copy.ReleaseID

group by Artist.ArtistName, Album.Title, Album.Genre, Release.ReleaseDate

GO

SELECT \* FROM CopyCountByAlbumAndYear

Order By Artist



------------------------------------------------------------------------------

--Creates view that shows CopyCountByAlbum

Alter VIEW CopyCountByAlbum AS

Select

Artist.ArtistName AS Artist,

Album.Title as Title,

Count(Album.Title) AS NumberOfCopies

From AlbumArtist

JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

RIGHT JOIN Release ON Album.AlbumID = Release.AlbumID

JOIN Copy ON Release.ReleaseID = Copy.ReleaseID

group by Artist.ArtistName, Album.Title

GO

SELECT \* FROM CopyCountByAlbum

Order By Artist



-------------------------------------------------------------------------------

--Function to retrieve ERRTHANG by a given artist

--Selects ERRTHANG

CREATE PROCEDURE ERRTHANG\_ByArtist(@ArtistName varchar(50))

AS

BEGIN

Select

Artist.ArtistName AS Artist,

Album.Title,

Album.Genre,

Year(Release.ReleaseDate) AS Year,

Release.Label,

Copy.MediaCondition,

Copy.SleeveCondition

From AlbumArtist

JOIN Artist ON AlbumArtist.ArtistID = Artist.ArtistID

JOIN Album ON AlbumArtist.AlbumID = Album.AlbumID

RIGHT JOIN Release ON Album.AlbumID = Release.AlbumID

JOIN Copy ON Release.ReleaseID = Copy.ReleaseID

Where Artist.ArtistName = (@ArtistName)

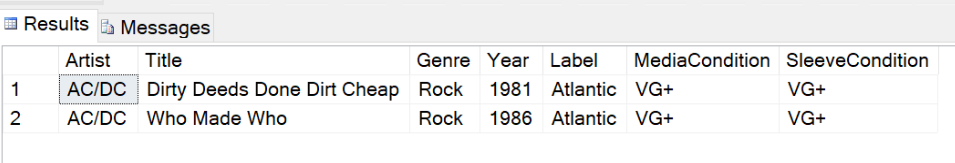
Order By Artist.ArtistName, Album.Title

END

GO

EXEC ERRTHANG\_ByArtist ‘AC/DC

-- iT wErKeD –



------------------------------------------------------------------------------

CREATE PROCEDURE AddArtist(@ArtistName varchar(50)) AS

BEGIN

IF NOT EXISTS (SELECT 1 FROM Artist

WHERE ArtistName = @ArtistName)

BEGIN

INSERT INTO Artist(ArtistName)

Values (@ArtistName)

RETURN @@IDENTITY

END

DECLARE @NewArtistID int

EXEC @NewArtistID = AddArtist 'Macy Gray'

-----------------------------------------------------------------------------------

CREATE PROCEDURE AddAlbum(@Title varchar(100), @Genre varchar(30)) AS

BEGIN

IF NOT EXISTS (SELECT 1 FROM Album

WHERE Title = @Title)

BEGIN

INSERT INTO Album(Title, Genre)

Values (@Title, @Genre)

RETURN @@IDENTITY

END

END

Exec AddAlbum 'NewTitle', 'NewAge'

select \* from album

---------------------------------------------------------------

CREATE PROCEDURE AddRelease(@ReleaseDate datetime, @Label varchar(30), @Title varchar(100)) AS

BEGIN

INSERT INTO Release (ReleaseDate, Label, AlbumID)

VALUES

((@ReleaseDate),

(@Label),

(SELECT AlbumID FROM Album WHERE Title= @Title))

RETURN @@IDENTITY

END

Exec AddRelease '2040', 'Ratchet Records', 'Amazing'

select \* from Release

----------------------------------------------------------------------------------------------------------------------------

Security

Create user Dad for Login Dad

Grant Select on Album, AlbumArtist, Artist, Copy, Release to Dad

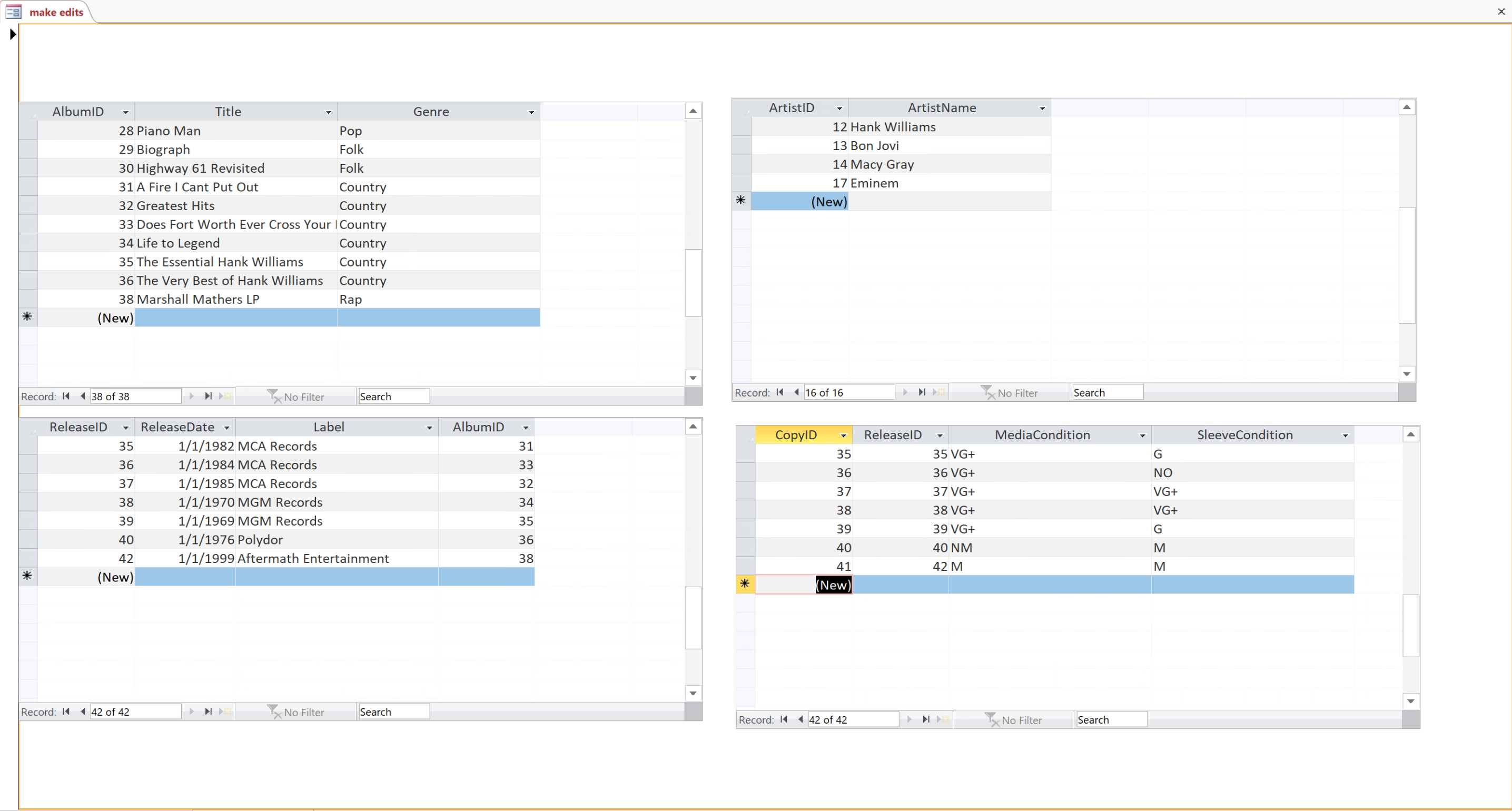
Grant Update on Album, AlbumArtist, Artist, Copy, Release to Dad

Grant Delete on Album, AlbumArtist, Artist, Copy, Release to Dad

Grant Insert on Album, AlbumArtist, Artist, Copy, Release to Dad

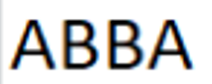
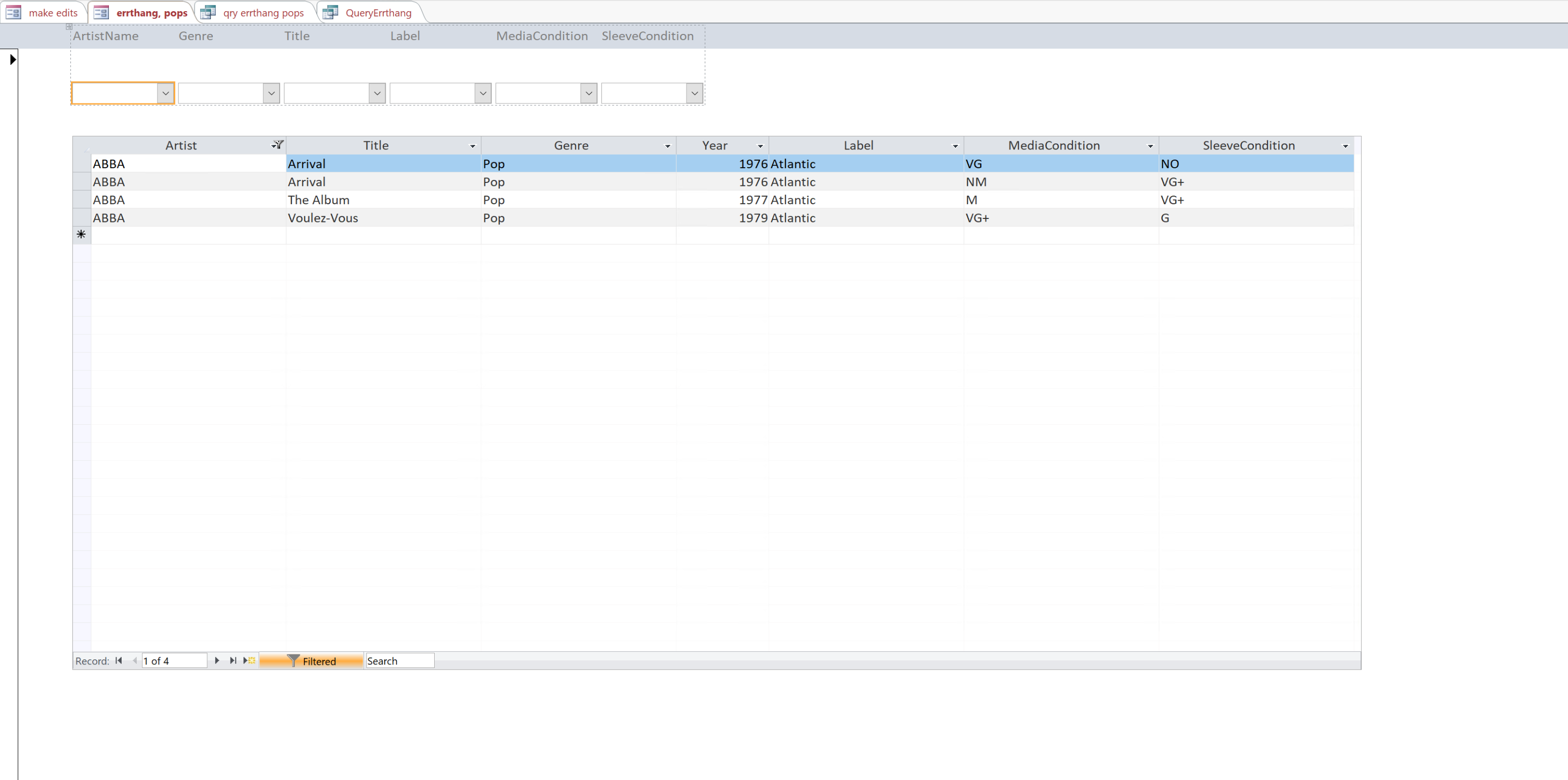
implementation

The below access form can be used to enter data directly into the database. This is not eloquent; but, for now this will work. Additional user testing will be necessary to determine how he will use this. Again, for now, this gives him the ability to enter data. As you can see, he inserted the Marshall Mathers LP by Eminem into the database. Now, this will show up on the queries that are in another form.

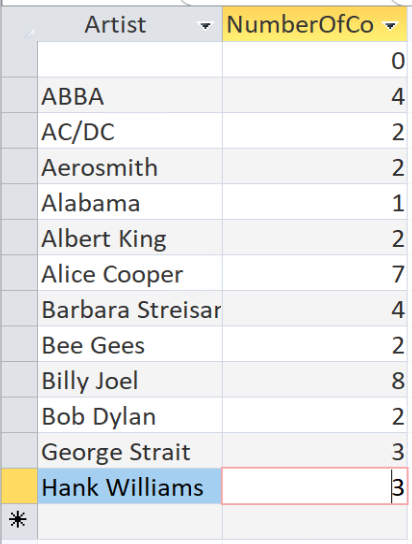


Review all the fields in one searchable form here (allows you to know if an album exists and in what shape it is in:

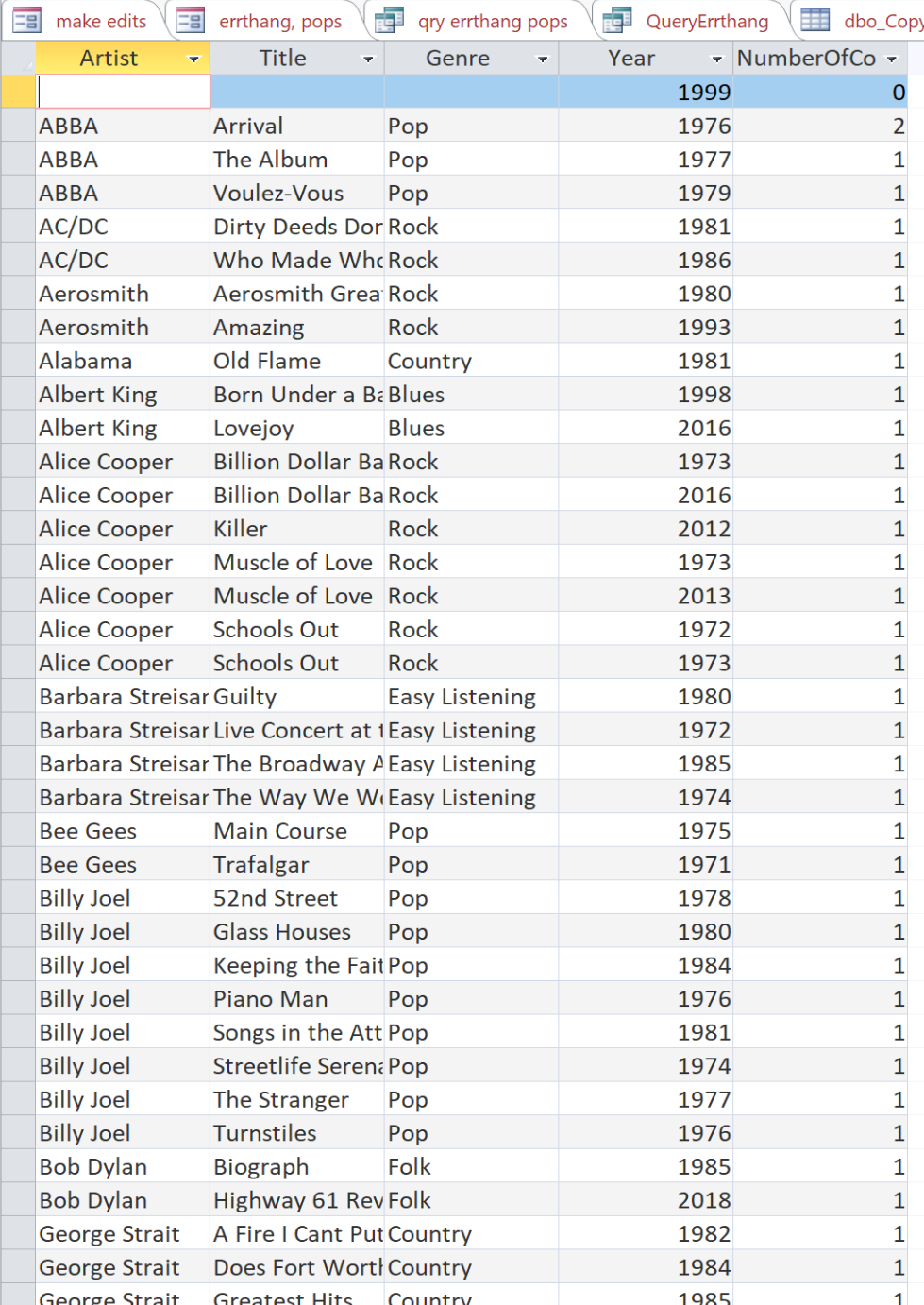




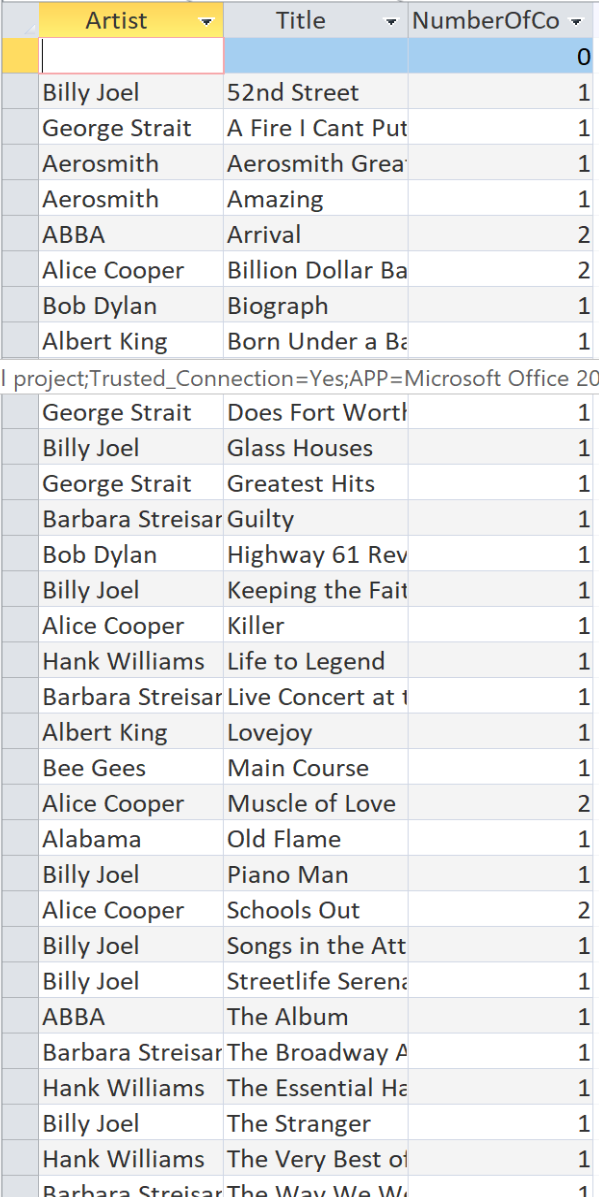
Copy count by artist



Copy count by artist and year



Copy count by album



**Album count by artist**

