Database Design and Joins

Ying Xu
Assistant Professor
Engineering Systems and Design (ESD)
Singapore University of Technology and Design

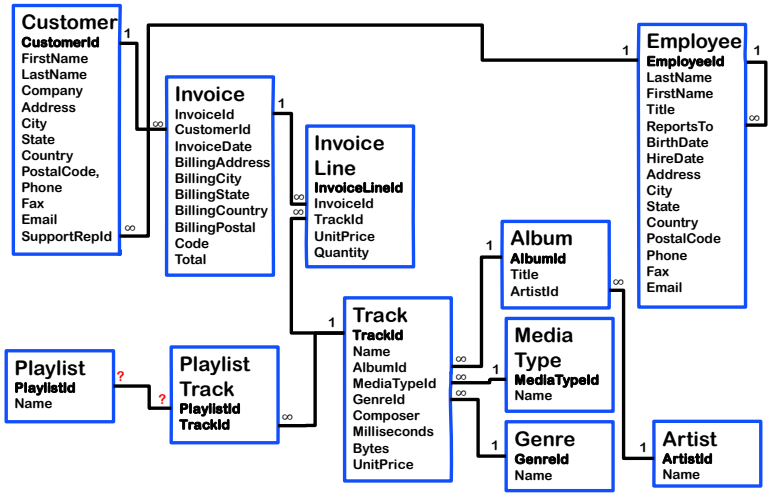
Edited based on Peter Jackson's slides on "Database Design and Joins" of DBA 2017

Overview

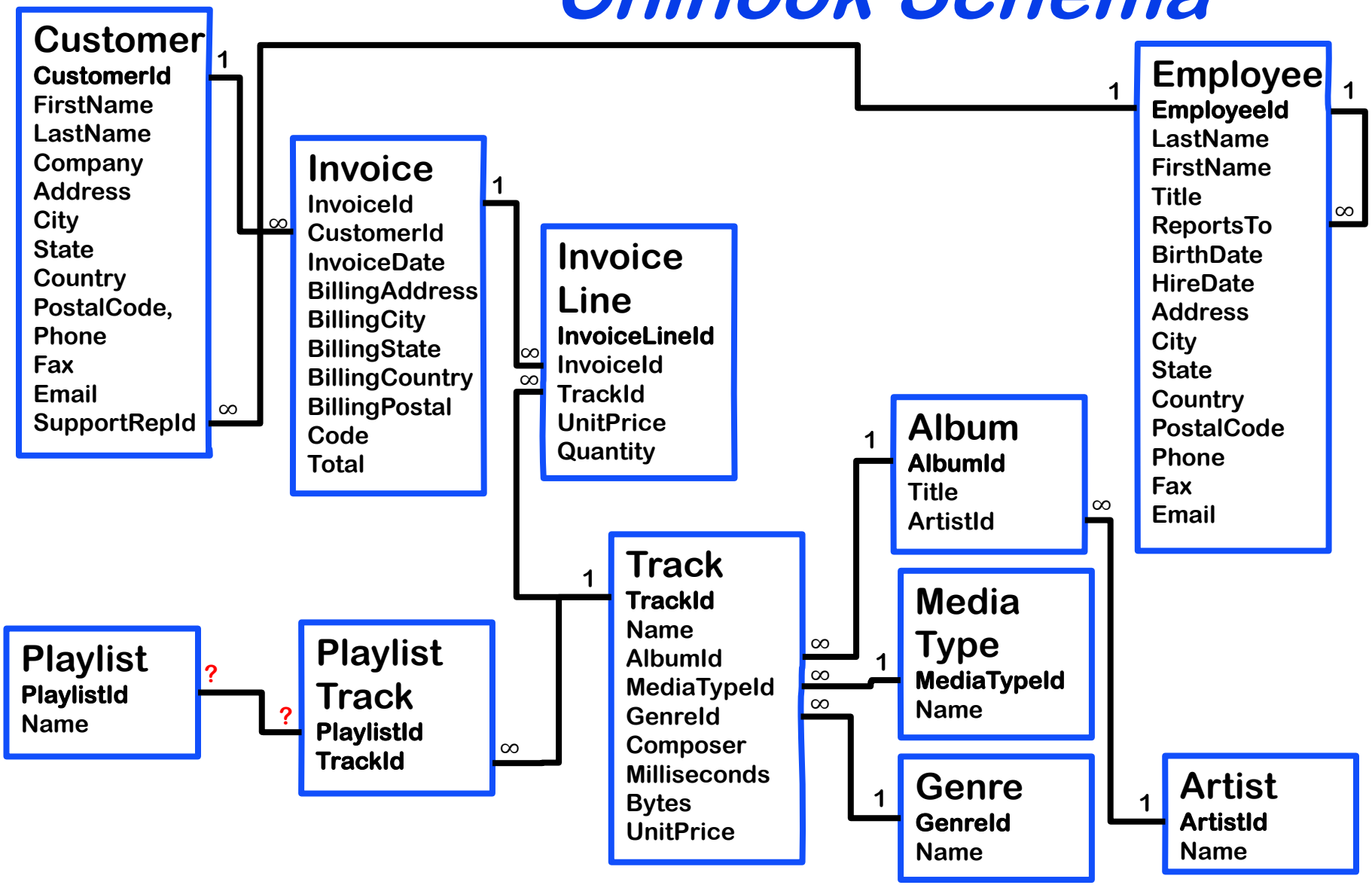
- *Database Design*
- *Database Joins*

Database Schema

- *Schema: map of tables, fields, and relationships in a relational database*
- *Schema = database design*

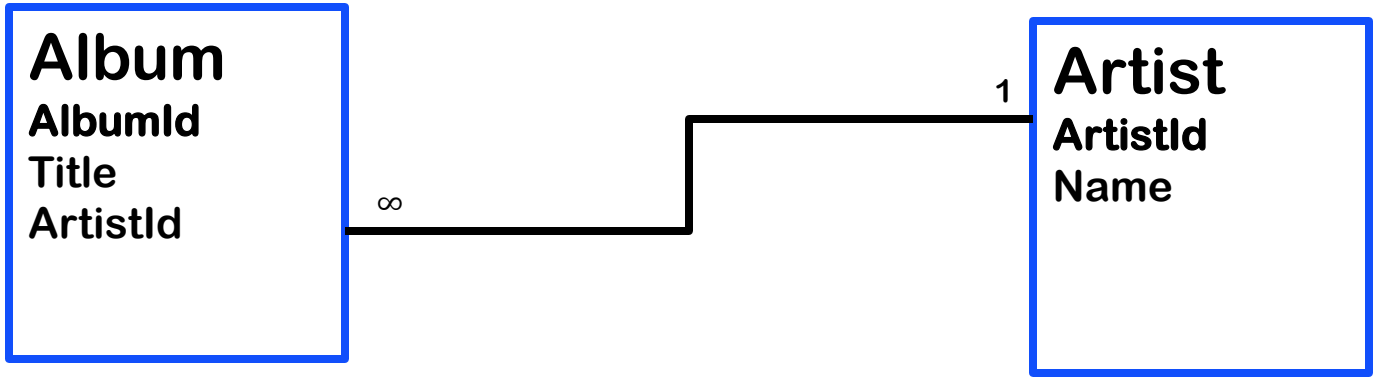


Chinook Schema



One-to-Many Relationship

One album is owned by one artist,
but one artist might own many albums.



That's why we could add ArtistId into the table Album as a foreign key

Question: could we add AlbumId into the table Artist as foreign key?

Many-to-Many Relations?

PlaylistId	TrackId
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10

Playlist
PlaylistId
Name

Track
TrackId
Name
AlbumId
MediaTypeId
GenreId
Composer
Milliseconds
Bytes
UnitPrice

PlaylistId	TrackId
1	1
8	1
17	1
1	2
8	2
17	2
1	3
5	3
8	3
17	3

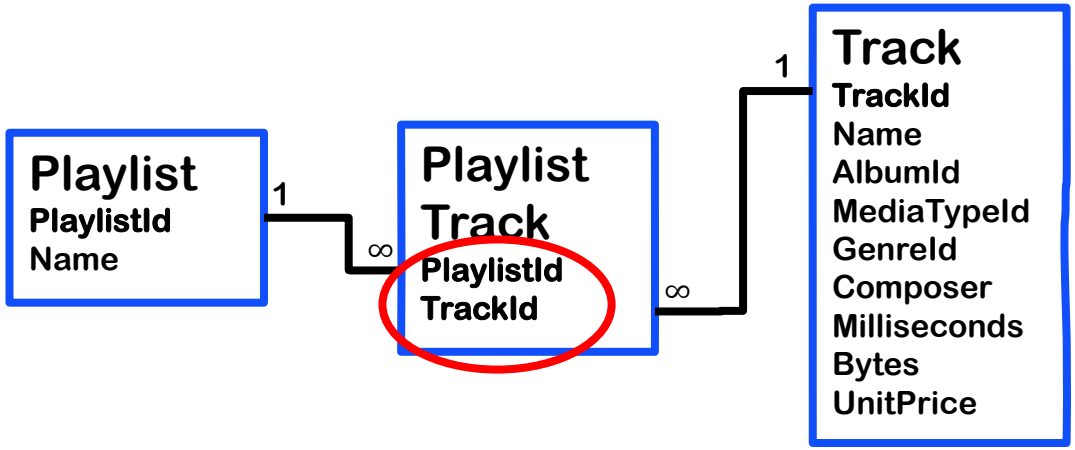
A playlist can consist of many tracks.

A track can appear on many playlists.

How do we capture many-to-many relations?

Membership Lists

In general, if you have many-to-many relations, you will need an extra table



One primary key has two fields: no one field by itself is unique in the table

Table name: PlaylistTrack ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	PlaylistId	INTEGER							NULL
2	TrackId	INTEGER							NULL

PlaylistTrack Table

Sorted by PlaylistId

PlaylistId	TrackId
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10

```
SELECT PlaylistId,
       TrackId
  FROM PlaylistTrack order by PlayListID;
```

Sorted by TrackId

PlaylistId	TrackId
1	1
8	1
17	1
1	2
8	2
17	2
1	3
5	3
8	3
17	3

```
SELECT PlaylistId,
       TrackId
  FROM PlaylistTrack order by TrackID;
```

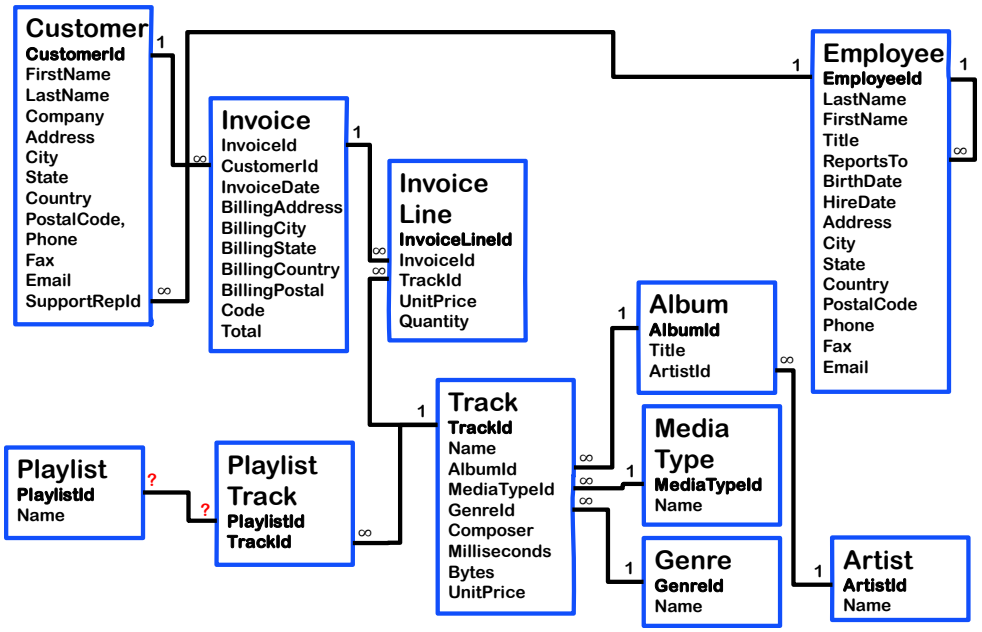
Examples of Membership Lists

- *Class list = StudentId+CourseId*
- *Fifth Row Club list = StudentId+ClubId*
- *Surgery Schedule = PatientId + SurgeonId + OperatingRoomId+Date+Time*
- *Library Loans = StudentId+BookId*

In general, if you have many-to-many relations, you will need an extra table whose primary key consists of multiple fields

Design Limits Analysis

- *The schema determines what questions you can answer using the database*



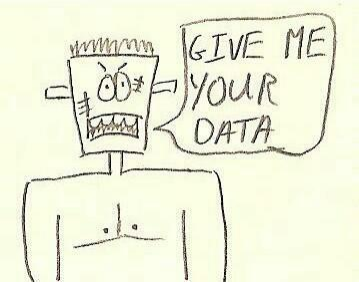
Not every question I am interested in can be answered using this database.

Which Questions Can We (Can We Not) Answer Using This DB?

- Which albums does the artist named "Queen" owns?
- Which employee serves the most number of customers?
- Which artist has the greatest sales?
- What is the most popular genre of music in Texas?
- Which invoice caused the most number of customer support calls?

Project Database

- When you get or request data from your client, you should be building a database schema in your mind
  - What data do you need to answer the client's question?
  - How will you break up the data into tables?
  - What fields are in each table?
  - How are the tables linked?
- Don't just say "Give me your data."

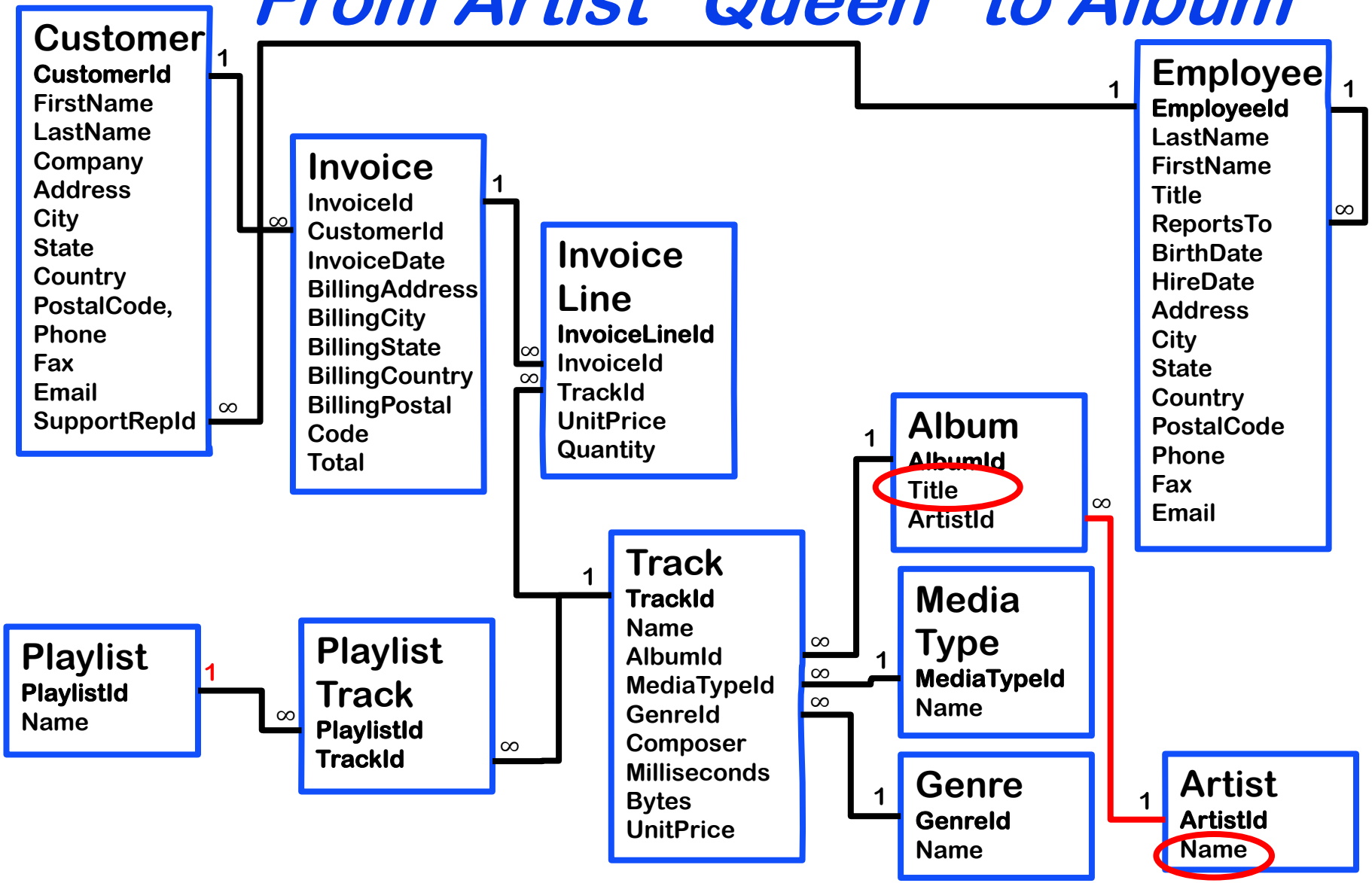


http://lawschooltuitionbubble.files.wordpress.com/2011/03/2011-03-12-give-me-your-data.jpg

Which Questions Can We (Can We Not) Answer Using This DB?

- Which albums does the artist named "Queen" owns?
- Which employee serves the most number of customers?
- Which artist has the greatest sales?
- What is the most popular genre of music in Texas?
- Which invoice caused the most number of customer support calls?

From Artist "Queen" to Album



Joins

- *Good database design separates related data into tables so that the data are not needlessly repeated*

- *So, how do we put it back together?*

All Possible Combinations

```
SELECT * FROM Album, Artist
```

“,” means all combinations

Total rows loaded: 95425

	AlbumId	Title	ArtistId	ArtistId:1	Name
1	1	For Those About To Rock We Salute You	1	1	AC/DC
2	1	For Those About To Rock We Salute You	1	2	Accept
3	1	For Those About To Rock We Salute You	1	3	Aerosmith
4	1	For Those About To Rock We Salute You	1	4	Alanis Morissette
5	1	For Those About To Rock We Salute You	1	5	Alice In Chains
6	1	For Those About To Rock We Salute You	1	6	Antônio Carlos Jobim
7	1	For Those About To Rock We Salute You	1	7	Apocalyptica
8	1	For Those About To Rock We Salute You	1	8	Audioslave
9	1	For Those About To Rock We Salute You	1	9	BackBeat
10	1	For Those About To Rock We Salute You	1	10	Billy Cobham
11	1	For Those About To Rock We Salute You	1	11	Black Label Society
12	1	For Those About To Rock We Salute You	1	12	Black Sabbath
13	1	For Those About To Rock We Salute You	1	13	Body Count
14	1	For Those About To Rock We Salute You	1	14	Bruce Dickinson
15	1	For Those About To Rock We Salute You	1	15	Buddy Guy
16	1	For Those About To Rock We Salute You	1	16	Caetano Veloso
17	1	For Those About To Rock We Salute You	1	17	Chico Buarque
18	1	For Those About To Rock We Salute You	1	18	Chico Science & Nação Zumbi
19	1	For Those About To Rock We Salute You	1	19	Cidade Negra
20	1	For Those About To Rock We Salute You	1	20	Cláudio Zoli
21	1	For Those About To Rock We Salute You	1	21	Various Artists

Note: “ArtistId” is ambiguous

We Want Match Album and Artist Based on 'ArtistId'

```
SELECT * FROM Album, Artist
WHERE Artist.ArtistId = Album.ArtistId
```

Total rows loaded: 347

	AlbumId	Title	ArtistId	ArtistId:1	Name
1	1	For Those About To Rock We Salute You	1	1	AC/DC
2	2	Balls to the Wall	2	2	Accept
3	3	Restless and Wild	2	2	Accept
4	4	Let There Be Rock	1	1	AC/DC
5	5	Big Ones	3	3	Aerosmith
6	6	Jagged Little Pill	4	4	Alanis Morissette
7	7	Facelift	5	5	Alice In Chains
8	8	Warner 25 Anos	6	6	Antônio Carlos Jobim
9	9	Plays Metallica By Four Cellos	7	7	Apocalyptica
10	10	Audioslave	8	8	Audioslave
11	11	Out Of Exile	8	8	Audioslave
12	12	BackBeat Soundtrack	9	9	BackBeat
13	13	The Best Of Billy Cobham	10	10	Billy Cobham
14	14	Alcohol Fueled Brewtality Live! [Disc 1]	11	11	Black Label Society
15	15	Alcohol Fueled Brewtality Live! [Disc 2]	11	11	Black Label Society

Note: "ArtistId" is matched

An Equivalent Query: INNER JOIN

```
SELECT * FROM Album INNER JOIN Artist
ON Album.ArtistId = Artist.ArtistId
```

Replace "WHERE" with "ON"

Replace "," with "INNER JOIN"

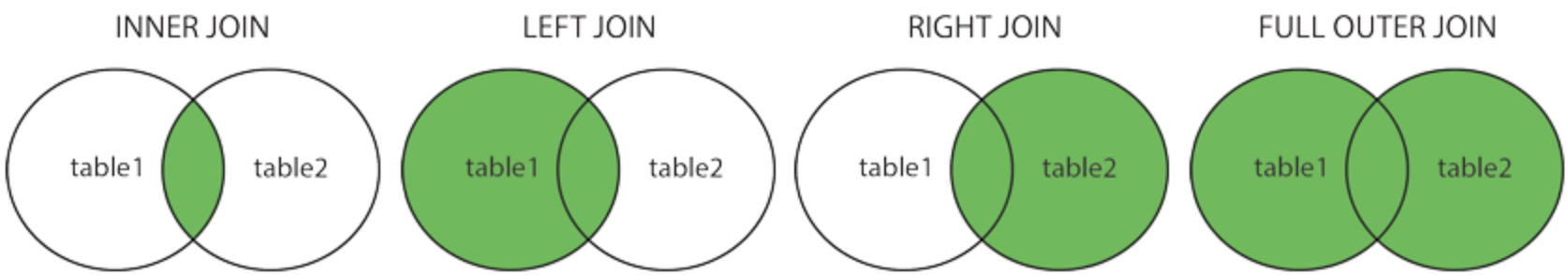
Total rows loaded: 347

	AlbumId	Title	ArtistId	ArtistId:1	Name
1	1	For Those About To Rock We Salute You	1	1	AC/DC
2	2	Balls to the Wall	2	2	Accept
3	3	Restless and Wild	2	2	Accept
4	4	Let There Be Rock	1	1	AC/DC
5	5	Big Ones	3	3	Aerosmith
6	6	Jagged Little Pill	4	4	Alanis Morissette
7	7	Facelift	5	5	Alice In Chains
8	8	Warner 25 Anos	6	6	Antônio Carlos Jobim
9	9	Plays Metallica By Four Cellos	7	7	Apocalyptica
10	10	Audioslave	8	8	Audioslave
11	11	Out Of Exile	8	8	Audioslave
12	12	BackBeat Soundtrack	9	9	BackBeat
13	13	The Best Of Billy Cobham	10	10	Billy Cobham
14	14	Alcohol Fueled Brewtality Live! [Disc 1]	11	11	Black Label Society
15	15	Alcohol Fueled Brewtality Live! [Disc 2]	11	11	Black Label Society

Joins are more readable, controllable than WHERE clauses…

Different Types of SQL JOINs

- ***INNER JOIN: Returns records that have matching values in both tables***
- ***LEFT JOIN: Return all records from the left table, and the matched records from the right table***
- ***RIGHT JOIN: Return all records from the right table, and the matched records from the left table***
- ***FULL OUTER JOIN: Return all records when there is a match in either left or right table (not supported by SQLite or MS excess)***



Which Questions Can We (Can We Not) Answer Using This DB?

- Which albums does the artist named "Queen" own?
- Which employee serves the most number of customers?
- Which artist has the greatest sales?
- What is the most popular genre of music in Texas?
- Which invoice caused the most number of customer support calls?

Which albums does the artist named "Queen" own?

```
SELECT * FROM Album INNER JOIN Artist
ON Album.ArtistId = Artist.ArtistId
WHERE Artist.Name = 'Queen'
```

Query results

	AlbumId	Title	ArtistId	ArtistId:1	Name
1	36	Greatest Hits II	51	51	Queen
2	185	Greatest Hits I	51	51	Queen
3	186	News Of The World	51	51	Queen

Note: "ArtistId" not needed anymore

```
SELECT Artist.Name, Album.Title FROM Album INNER JOIN Artist
ON Album.ArtistId = Artist.ArtistId
WHERE Artist.Name = 'Queen'
```

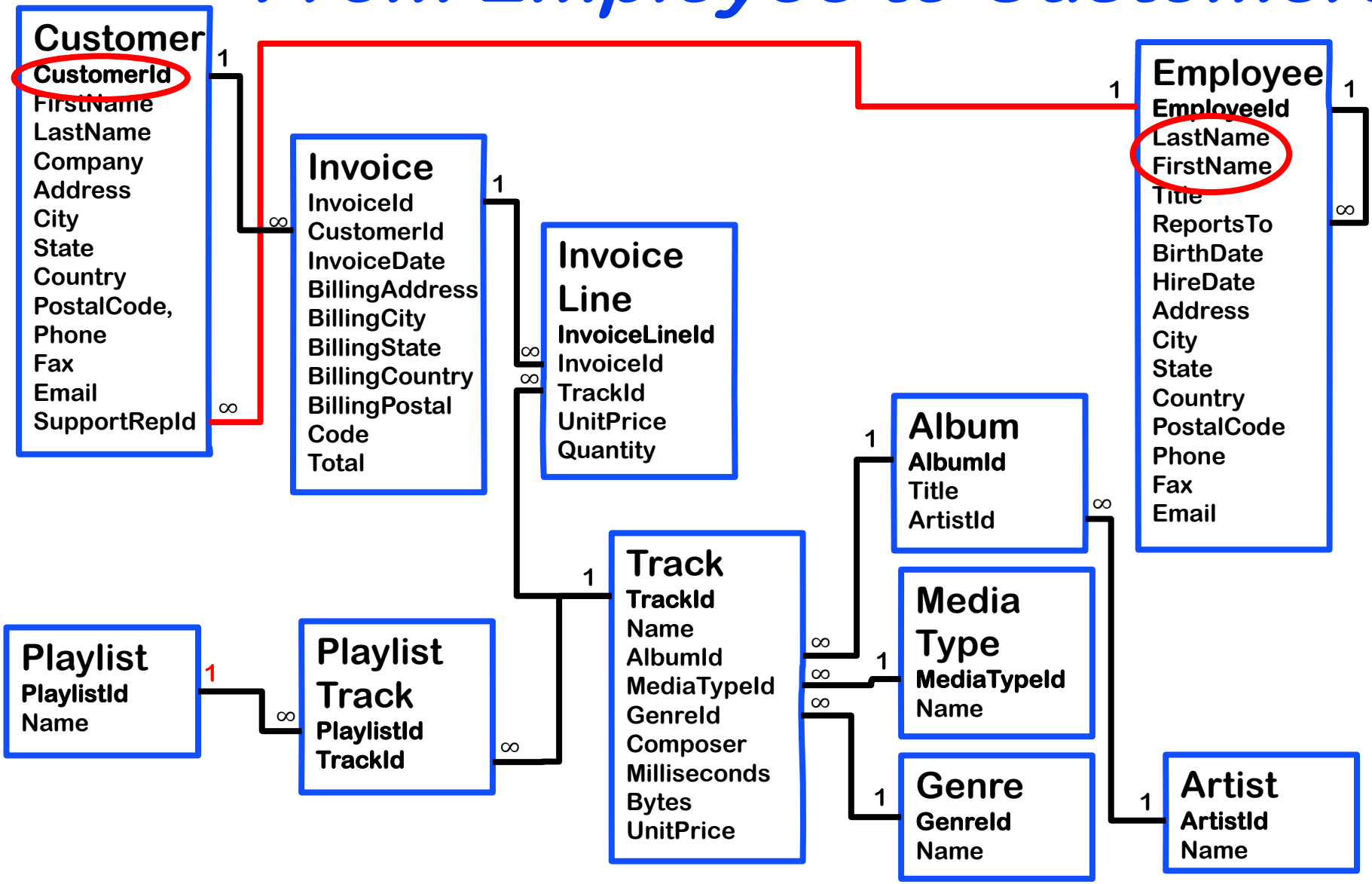
Query results

	Name	Title
1	Queen	Greatest Hits II
2	Queen	Greatest Hits I
3	Queen	News Of The World

Which Questions Can We (Can We Not) Answer Using This DB?

- Which albums does the artist named "Queen" own?
- Which employee serves the most number of customers?
- Which artist has the greatest sales?
- What is the most popular genre of music in Texas?
- Which invoice caused the most number of customer support calls?

From Employee to Customers



Match Employee and Customers Based on 'EmployeeId' (SupportRepId)

```
SELECT Employee.FirstName, Employee.LastName, Customer.CustomerId
FROM Employee INNER JOIN Customer
ON Employee.EmployeeId = Customer.SupportRepId
ORDER BY Employee.FirstName,Employee.LastName,Customer.CustomerId
```

Query results

Total rows loaded: 59

	FirstName	LastName	CustomerId
1	Jane	Peacock	1
2	Jane	Peacock	3
3	Jane	Peacock	12
4	Jane	Peacock	15
5	Jane	Peacock	18
6	Jane	Peacock	19
7	Jane	Peacock	24
8	Jane	Peacock	29
9	Jane	Peacock	30
10	Jane	Peacock	33
11	Jane	Peacock	37
12	Jane	Peacock	38
13	Jane	Peacock	42
14	Jane	Peacock	43
15	Jane	Peacock	44
16	Jane	Peacock	45
17	Jane	Peacock	46
18	Jane	Peacock	52
19	Jane	Peacock	53
20	Jane	Peacock	58
21	Jane	Peacock	59
22	Margaret	Park	4

We are able to count the number of customers served by each employee using keyword COUNT and GROUP BY

Which employee serves the most number of customers?

```
SELECT Employee.FirstName,Employee.LastName,
COUNT(Customer.CustomerId) AS CustomerNo
FROM Employee INNER JOIN Customer
ON Employee.EmployeeId=Customer.SupportRepId
GROUP BY Employee.EmployeeId
ORDER BY CustomerNo DESC
```

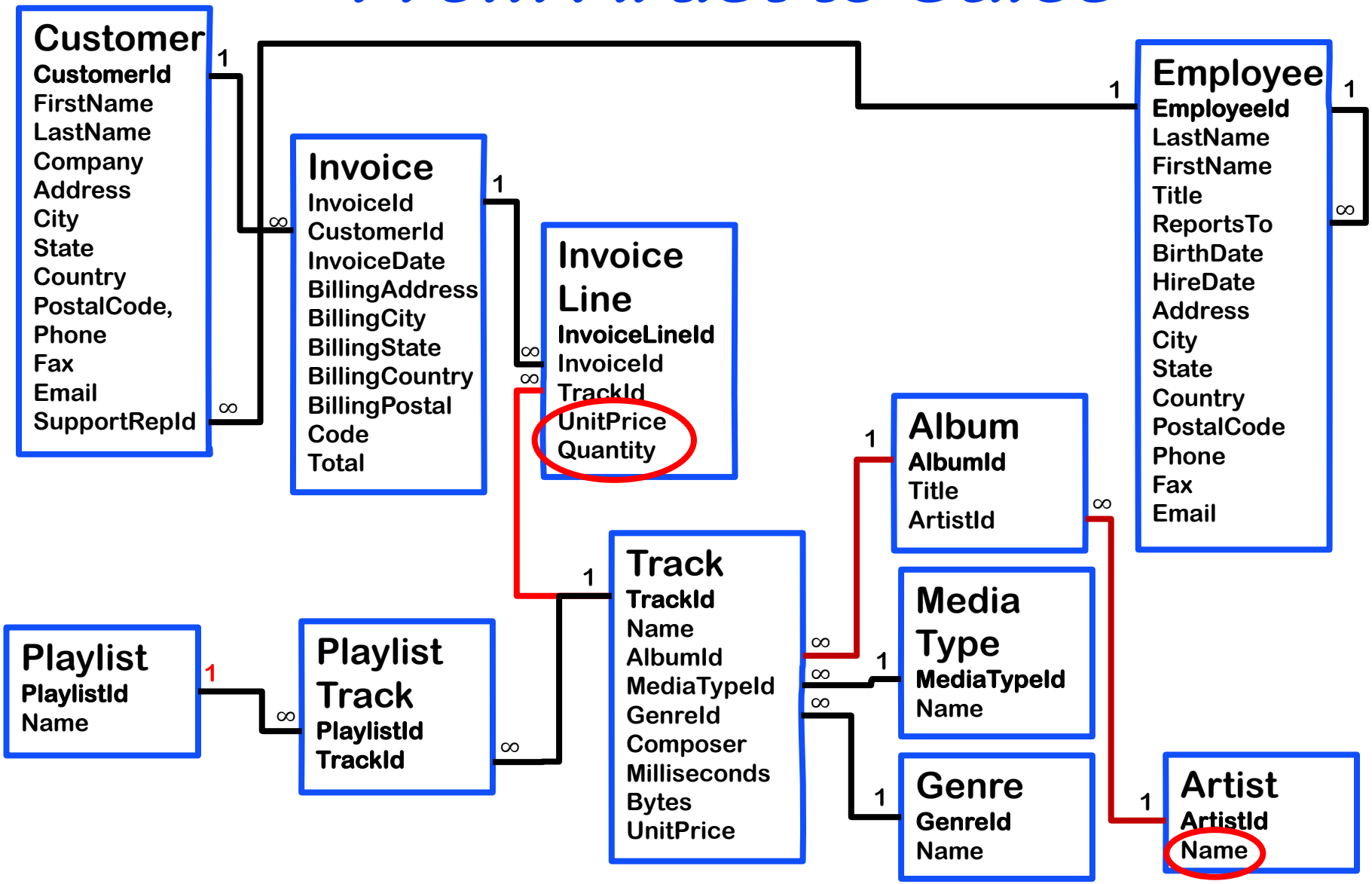
Query results

	FirstName	LastName	CustomerNo
1	Jane	Peacock	21
2	Margaret	Park	20
3	Steve	Johnson	18

Which Questions Can We (Can We Not) Answer Using This DB?

- Which albums does the artist named "Queen" own?
- Which employee serves the most number of customers?
- Which artist has the greatest sales?
- What is the most popular genre of music in Texas?
- Which invoice caused the most number of customer support calls?

From Artist to Sales



Which artist has the greatest sales?

1. The sales of each track?
2. The sales of each track in each album?
3. The sales of each track of each artist?
4. The total sales of each artist?

The sales of each track?

```
SELECT InvoiceLine.TrackId,
InvoiceLine.UnitPrice * InvoiceLine.Quantity AS TrackSales
FROM InvoiceLine
```

Query results

Total rows loaded: 2240

	TrackId	TrackSales
1	2	0.99
2	4	0.99
3	6	0.99
4	8	0.99
5	10	0.99
6	12	0.99
7	16	0.99
8	20	0.99
9	24	0.99
10	28	0.99
11	32	0.99
12	36	0.99
13	42	0.99
14	48	0.99
15	54	0.99
16	60	0.99

In-Class Activity 1

The sales of each track in each album?

```
SELECT InvoiceLine.TrackId,Track.AlbumId,
InvoiceLine.UnitPrice * InvoiceLine.Quantity AS TrackSales
FROM InvoiceLine INNER JOIN Track
ON InvoiceLine.TrackId = Track.TrackId
```

Query results

Total rows loaded: 2220

	TrackId	AlbumId	TrackSales
1	2	2	0.99
2	4	3	0.99
3	6	1	0.99
4	8	1	0.99
5	10	1	0.99
6	12	1	0.99
7	16	4	0.99
8	20	4	0.99
9	24	5	0.99
10	28	5	0.99
11	32	5	0.99
12	36	5	0.99
13	42	6	0.99
14	48	6	0.99
15	54	7	0.99

The sales of each track of each artist?

```
SELECT Album.ArtistId,
InvoiceLine.UnitPrice * InvoiceLine.Quantity AS TrackSales
FROM ( InvoiceLine INNER JOIN Track
ON InvoiceLine.TrackId = Track.TrackId )
INNER JOIN Album ON Track.AlbumId = Album.AlbumId
```

Query results

Total rows loaded: 2240

	ArtistId	TrackSales
1	2	0.99
2	2	0.99
3	1	0.99
4	1	0.99
5	1	0.99
6	1	0.99
7	1	0.99
8	1	0.99
9	3	0.99
10	3	0.99
11	3	0.99
12	3	0.99
13	4	0.99
14	4	0.99

The sales of each tracks of each artist (with name)?

```
SELECT Artist.Name,
InvoiceLine.UnitPrice * InvoiceLine.Quantity AS TrackSales
FROM ((InvoiceLine INNER JOIN Track ON
InvoiceLine.TrackId = Track.TrackId )
INNER JOIN Album ON Track.AlbumId = Album.AlbumId)
INNER JOIN Artist ON Album.ArtistId=Artist.ArtistId
```

Query results

Total rows loaded: 2240

	Name	TrackSales
1	Accept	0.99
2	Accept	0.99
3	AC/DC	0.99
4	AC/DC	0.99
5	AC/DC	0.99
6	AC/DC	0.99
7	AC/DC	0.99
8	AC/DC	0.99
9	Aerosmith	0.99
10	Aerosmith	0.99
11	Aerosmith	0.99
12	Aerosmith	0.99

The total sales of each artist?

```
SELECT Artist.Name,
SUM(InvoiceLine.UnitPrice * InvoiceLine.Quantity) AS ArtistSales
FROM ( ( InvoiceLine INNER JOIN Track ON InvoiceLine.TrackId =
Track.TrackId )
INNER JOIN Album ON Track.AlbumId = Album.AlbumId )
INNER JOIN Artist ON Album.ArtistId = Artist.ArtistId
GROUP BY Artist.Name
ORDER BY ArtistSales DESC
```

Query results

Total rows loaded: 165

	Name	ArtistSales
1	Iron Maiden	138.5999999999998
2	U2	105.92999999999982
3	Metallica	90.0899999999999
4	Led Zeppelin	86.12999999999992
5	Lost	81.58999999999997
6	The Office	49.75000000000001
7	Os Paralamas Do Sucesso	44.55
8	Deep Purple	43.56
9	Faith No More	41.58
10	Eric Clapton	39.50000000000000

Which artist has the greatest sales?

1. The sales of each track?
2. The sales of each track in each album?
3. The sales of each track of each artist?
4. The total sales of each artist?

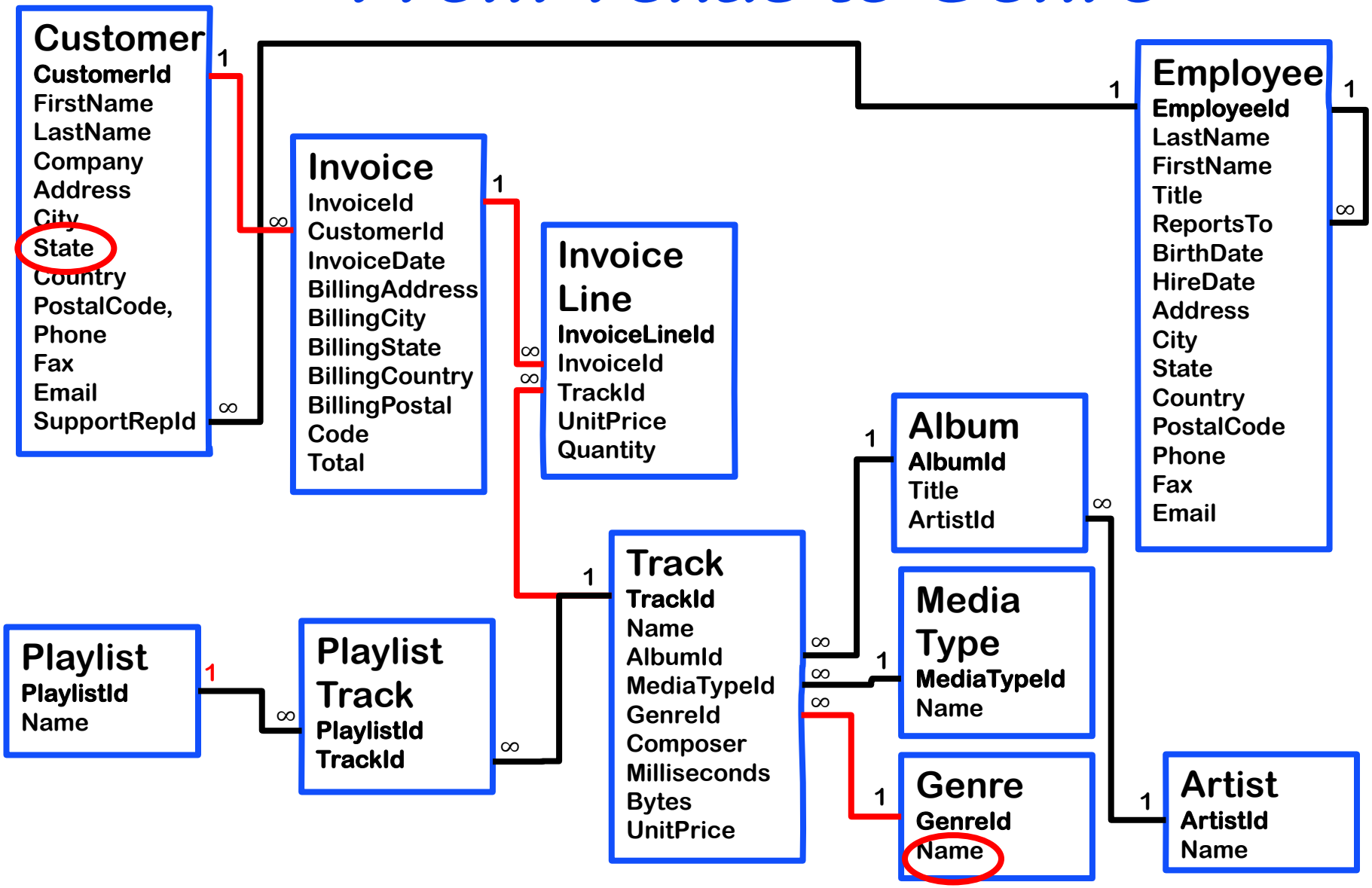
After-class question: why not calculate the total sales of each album first as follows?

1. The sales of each track?
2. The total sales of each album?
3. The total sales of each artist?

Which Questions Can We (Can We Not) Answer Using This DB?

- Which albums does the artist named "Queen" own?
- Which employee serves the most number of customers?
- Which artist has the greatest sales?
- What is the most popular genre of music in Texas?
- Which invoice caused the most number of customer support calls?

From Texas to Genre



What Is The Most Popular Genre in Texas?

```
SELECT COUNT(InvoiceLine.Quantity) AS GenreNo,Genre.Name
FROM (((Genre INNER JOIN Track ON Genre.GenreID=Track.GenreID)
INNER JOIN InvoiceLine ON Track.TrackID=InvoiceLine.TrackId)
INNER JOIN Invoice ON InvoiceLine.InvoiceId=Invoice.InvoiceId)
INNER JOIN Customer ON Invoice.CustomerId=Customer.CustomerId
WHERE Customer.State="TX" GROUP BY Genre.GenreId ORDER BY GenreNo DESC
```

Query results

	GenreNo	Name
1	14	Rock
2	7	Latin
3	6	Drama
4	5	Alternative & Punk
5	3	TV Shows
6	2	Metal
7	1	Sci Fi & Fantasy

Which Questions Can We (Can We Not) Answer Using This DB?

- Which employee serves the most number of customers?
- Which artist has the greatest sales?
- What is the most popular genre of music in Texas?
- Which invoice caused the most number of customer support calls?
  - No. We cannot answer this. The database does not record support calls made by invoice.

New Challenge

Okay, people! Let's start with this: what type of music does Eduardo listen to?

Customer -> Invoice-> Line -> Track-> Genre

http://www.zuguide.com/#David-Strathairn

Overview

- Database Design
- Database Joins