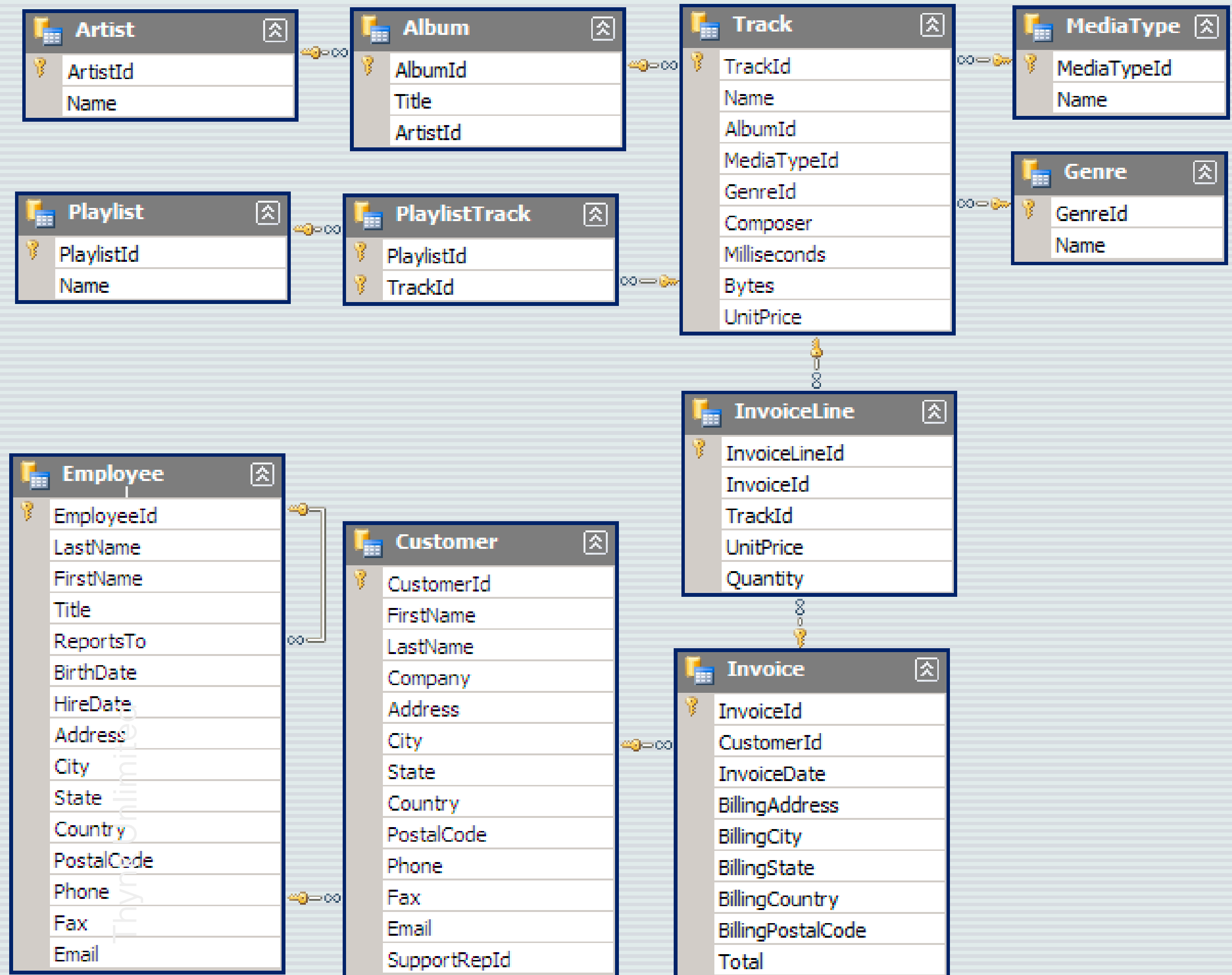


The background of the slide features a light blue gradient. On the left, there is a faint, semi-transparent image of a pair of white over-ear headphones. On the right, there is a smaller, more detailed image of a black vinyl record spinning on a turntable with a blue center label.

SQL PROJECT- MUSIC STORE DATA ANALYSIS

SCHEMA

01



Who is the senior most employee based on job title?

```
select * from employee
order by levels desc
limit 1
```

employee_id	last_name	first_name	title	reports_to	levels	birthdate	hire_date
1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00	14-08-2016 00:00

Which countries have the most Invoices?

```
select count(*), billing_country as total from invoice
group by billing_country
order by billing_country desc
```

	count(*)	total
▶	131	USA
	28	United Kingdom
	10	Sweden
	11	Spain
	29	Portugal
	10	Poland
	9	Norway
	10	Netherlands

What are top 3 values of total invoice?

```
select * from invoice
order by total desc
limit 3
```

	invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
►	183	42	2018-02-09 00:00:00	9, Place Louis Barthou	Bordeaux	None	France	33000	23.759999999999998
	92	32	2017-07-02 00:00:00	696 Osborne Street	Winnipeg	MB	Canada	R3L 2B9	19.8
	526	5	2020-06-08 00:00:00	Klanova 9/506	Prague	None	Czech Republic	14700	19.8

Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals

```
select sum(total) as invocie_total, billing_city as NAME_OF_CITY from invoice
group by billing_city
order by invocie_total desc
```

	invocie_total	NAME_OF_CITY
►	273.240000000000007	Prague
	169.29	Mountain View
	166.32	London
	158.4	Berlin
	151.47	Paris
	129.69	SÃ£o Paulo
	114.839999999999997	Dublin
	111.869999999999999	Delhi

Who is the best customer? The customer who has spent the most money will be declared the best customer. Write a query that returns the person ID who has spent the most money

```
SELECT C.CUSTOMER_ID,  
       SUM(I.TOTAL)  
FROM CUSTOMER C  
JOIN INVOICE I ON C.CUSTOMER_ID = I.CUSTOMER_ID  
GROUP BY 1  
ORDER BY 2 DESC
```

	CUSTOMER_ID	SUM(I.TOTAL)
►	5	144.54000000000002
	6	128.7
	46	114.83999999999997
	58	111.86999999999999
	1	108.89999999999998
	13	106.91999999999999
	34	102.96000000000001
	3	99.99

Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A

```
SELECT DISTINCT C.first_name, C.last_name, C.email, (G.name)genre_name
FROM customer C
INNER JOIN invoice I
ON C.customer_id = I.customer_id
INNER JOIN invoice_line IL
ON I.invoice_id = IL.invoice_id
INNER JOIN track T
ON IL.track_id = T.track_id
INNER JOIN genre G
ON T.genre_id = G.genre_id
WHERE G.name = 'Rock'
ORDER BY C.email;
```

	first_name	last_name	email	genre_name
►	Aaron	Mitchell	aaronmitchell@yahoo.ca	Rock
	Alexandre	Rocha	alero@uol.com.br	Rock
	Astrid	Gruber	astrid.gruber@apple.at	Rock
	Björn	Hansen	bjorn.hansen@yahoo.no	Rock
	Camille	Bernard	camille.bernard@yahoo.fr	Rock
	Daan	Peeters	daan_peeters@apple.be	Rock
	Diego	Gutiérrez	diego.gutierrez@yahoo.ar	Rock
	Dan	Miller	dmiller@comcast.com	Rock

Let's invite the artists who have written the most rock music in our dataset.
Write a query that returns the Artist name and total track count of the top 10 rock bands

```
1 • SELECT AR.NAME,  
2       COUNT(T.NAME)  
3     FROM TRACK T  
4    JOIN GENRE G ON T.GENRE_ID = G.GENRE_ID  
5    JOIN ALBUM AL ON AL.ALBUM_ID = T.ALBUM_ID  
6    JOIN ARTIST AR ON AR.ARTIST_ID = AL.ARTIST_ID  
7   WHERE G.NAME = 'Rock'  
8  GROUP BY 1  
9  ORDER BY 2 DESC
```

	NAME	COUNT(T.NAME)
▶	AC/DC	18
	Aerosmith	15
	Audioslave	14
	Led Zeppelin	14
	Alanis Morissette	13
	Alice In Chains	12
	Frank Zappa & Captain Beefheart	9
	Accept	4

Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first

```
1
2      #FIRST FIND OUT AVG MILLISECOND
3 •    SELECT avg(milliseconds) as avgmille
4      from track;
5      # SECOND QUERY
6 •    select name, milliseconds
7      from track
8      where milliseconds > 251177
```

	name	milliseconds
►	For Those About To Rock (We Salute You)	343719
	Balls to the Wall	342562
	Restless and Wild	252051
	Princess of the Dawn	375418
	Evil Walks	263497
	Breaking The Rules	263288
	Spellbound	270863
	Go Down	331180

Thank You