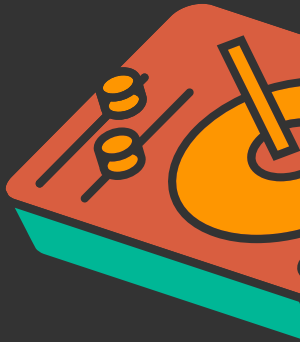



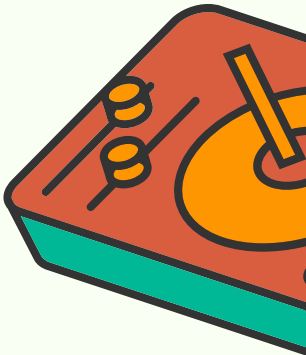
MUSIC STORE

ANALYSIS





OBJECTIVE

- 
- The primary objective of the music store is to achieve sustainable business growth while addressing existing challenges.
 - We need to examine dataset with SQL and help the music store understand its business growth by answering simple questions.
- 

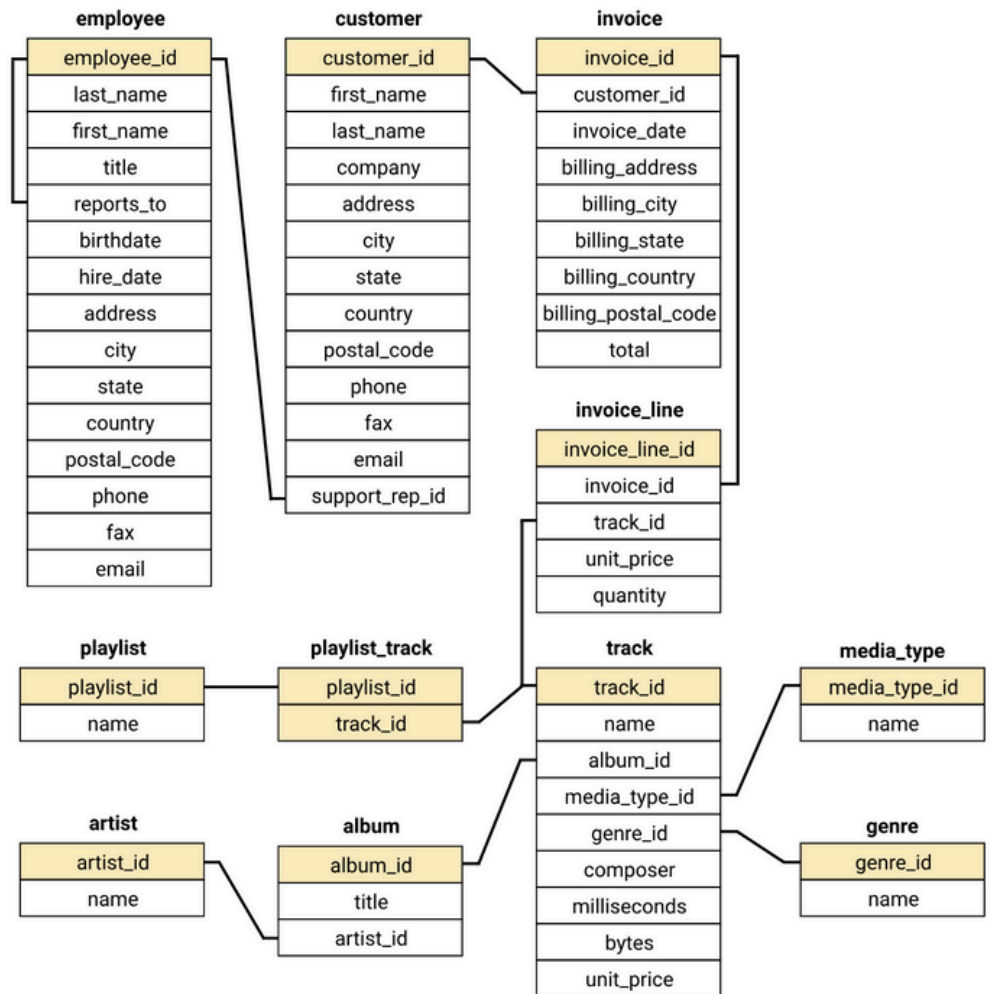


QUESTION MODES

- Easy-Queries include: SELECT, GROUP BY, ORDER BY, LIMIT, DESC
- Moderate-Queries Include: JOINS, GROUP BY, ORDER BY, LIMIT.
- Advance Queries Include: CTE (COMMON TABLE EXPRESSION).



MUSIC DATABASE



► LEVEL - EASY

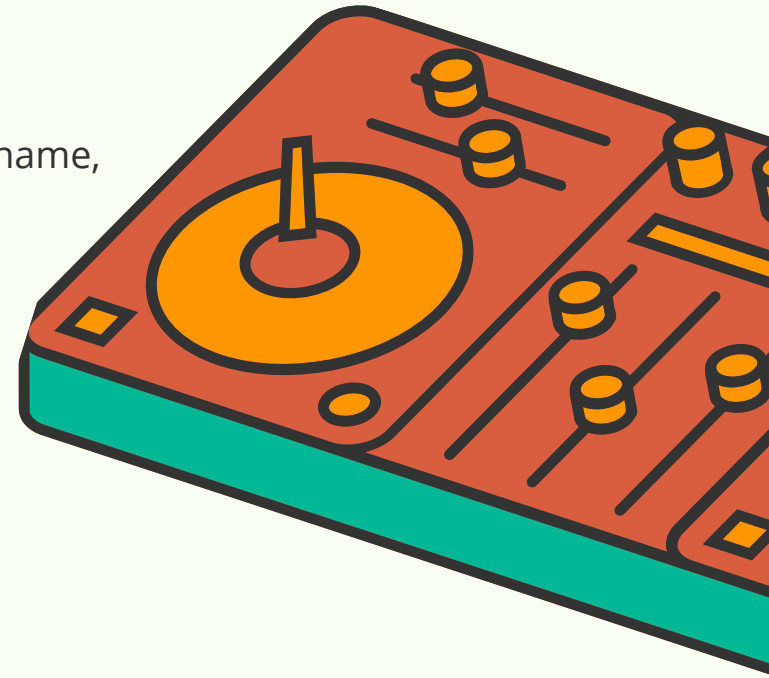
Q1: WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

- **Input :**

```
SELECT title,(first_name || ' ' || last_name) AS full_name,  
levels, email, city, country FROM employee  
ORDER BY levels DESC  
LIMIT 1
```

- **Output :**

	title	full_name	levels	email	city	country
►	Senior General Manager	Mohan Madan	L7	madan.mohan@chinookcorp.com	Edmonton	Canada



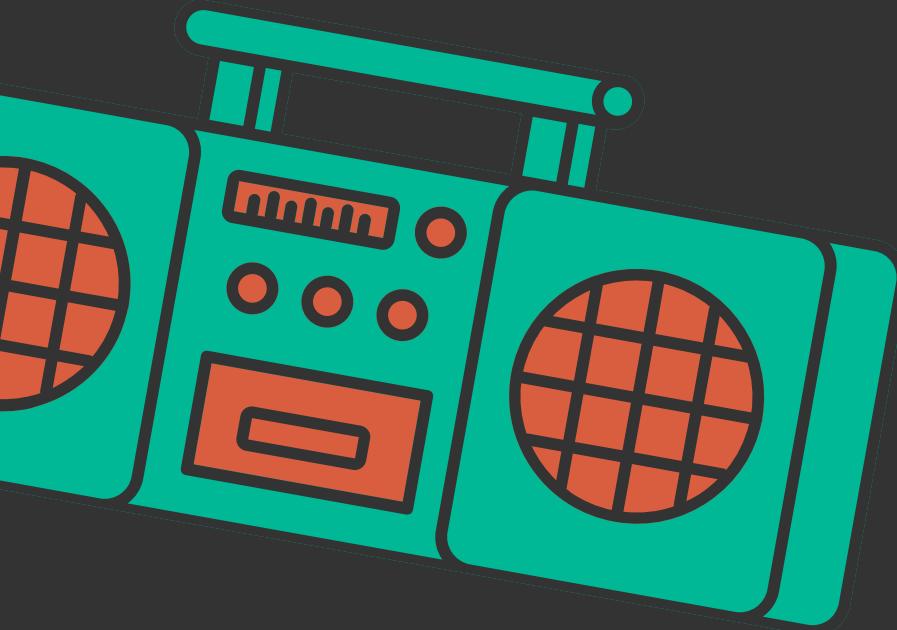
Q2: WHICH COUNTRIES HAVE THE MOST INVOICES?

- **Input :**

```
SELECT COUNT(*) AS c, billing_country  
FROM invoice  
GROUP BY billing_country  
ORDER BY c DESC
```

- **Output :**

	invoice_counts	billing_country
▶	131	USA
	76	Canada
	61	Brazil
	50	France
	41	Germany
	30	Czech Republic
	29	Portugal
	28	United Kingdom
	21	India
	13	Ireland
	13	Chile
	11	Finland
	11	Spain
	10	Poland
	10	Denmark



Q3: WHAT ARE THE TOP 3 VALUES OF TOTAL INVOICE?

- **Input :**

```
SELECT customer_id, total  
FROM invoice  
ORDER BY total DESC  
LIMIT 3
```

- **Output :**

	customer_id	total
▶	42	23.759999999999998
	32	19.8
	3	19.8



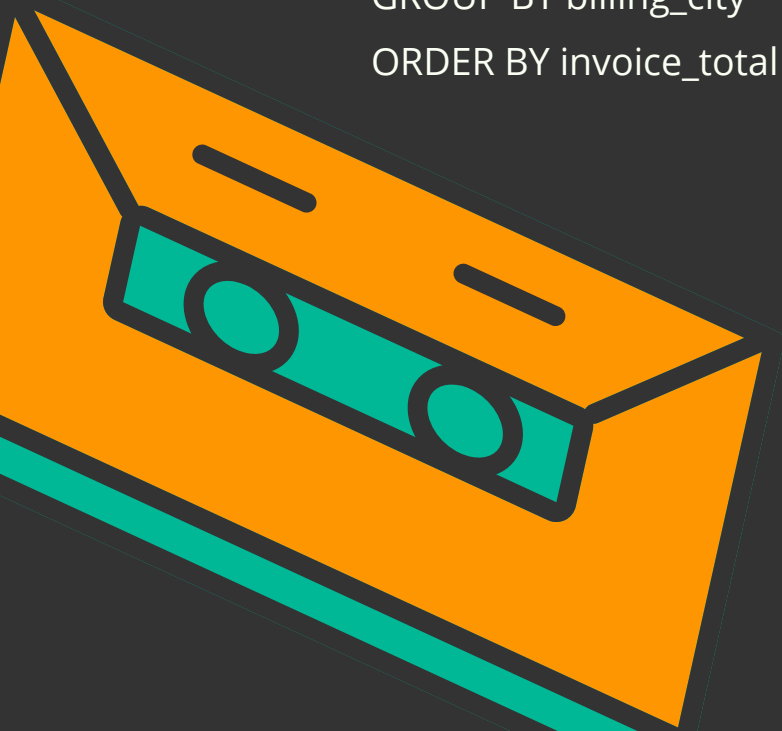
Q4: WHICH CITY HAS THE BEST CUSTOMERS? WE WOULD LIKE TO THROW A PROMOTIONAL MUSIC FESTIVAL IN THE CITY WE MADE THE MOST MONEY. WRITE A QUERY THAT RETURNS ONE CITY THAT HAS THE HIGHEST SUM OF INVOICE TOTALS. RETURN BOTH THE CITY NAME & SUM OF ALL INVOICE TOTALS.

- **Input :**

```
SELECT SUM(total) as invoice_total,  
billing_city  
FROM invoice  
GROUP BY billing_city  
ORDER BY invoice_total DESC
```

- **Output :**

	invoice_total	billing_city
▶	273.24000000000007	Prague
	169.29	Mountain View
	166.32	London
	158.4	Berlin
	151.47	Paris
	129.69	São Paulo
	114.83999999999997	Dublin
	111.86999999999999	Delhi
	108.89999999999998	São José dos Campos



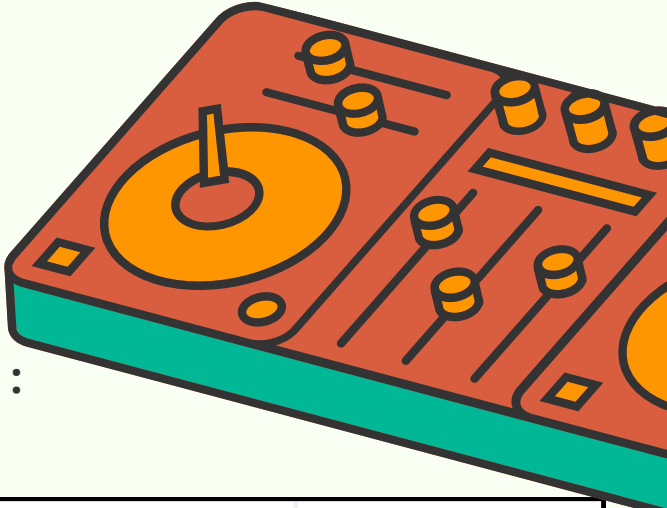
► LEVEL - MODERATE

Q1: 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.

- **Input :**

```
SELECT DISTINCT email, CONCAT(first_name, ' ', last_name)
AS customer_name
FROM customer
JOIN invoice ON customer.customer_id =
invoice.customer_id
JOIN invoice_line ON invoice.invoice_id =
invoice_line.invoice_id
WHERE track_id IN(
SELECT track_id FROM track
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name LIKE 'Rock')
ORDER BY email;
```

- **Output :**



	email	customer_name
►	aaronmitchell@yahoo.ca	Aaron Mitchell
	alero@uol.com.br	Alexandre Rocha
	astrid.gruber@apple.at	Astrid Gruber
	bjorn.hansen@yahoo.no	Björn Hansen
	camille.bernard@yahoo.fr	Camille Bernard
	daan_peeters@apple.be	Daan Peeters
	diego.gutierrez@yahoo.ar	Diego Gutiérrez
	dmiller@comcast.com	Dan Miller
	dominiquelefebvre@gmail.com	Dominique Lefebvre
	edfrancis@yahoo.ca	Edward Francis
	eduardo@woodstock.com.br	Eduardo Martins

Q2: 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.

- **Input :**

```
SELECT a.name,COUNT(t.track_id) AS tracks
FROM artist AS a
JOIN album AS al ON a.artist_id = al.artist_id
JOIN track AS t ON al.album_id = t.album_id
JOIN genre AS g ON t.genre_id = g.genre_id
WHERE g.name LIKE 'Rock'
GROUP BY a.name
ORDER BY tracks DESC
LIMIT 10
```

- **Output :**

	name	tracks
▶	AC/DC	18
	Aerosmith	15
	Audioslave	14
	Led Zeppelin	14
	Alanis Morissette	13
	Alice In Chains	12
	Frank Zappa & Captain Beefheart	9
	Accept	4



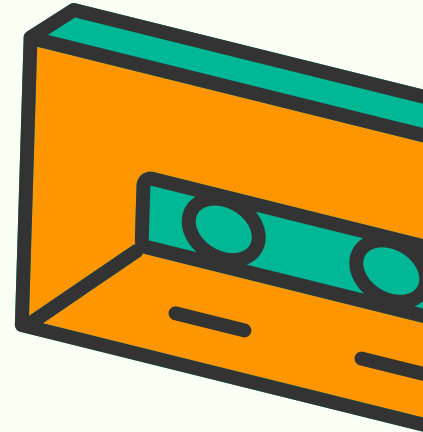
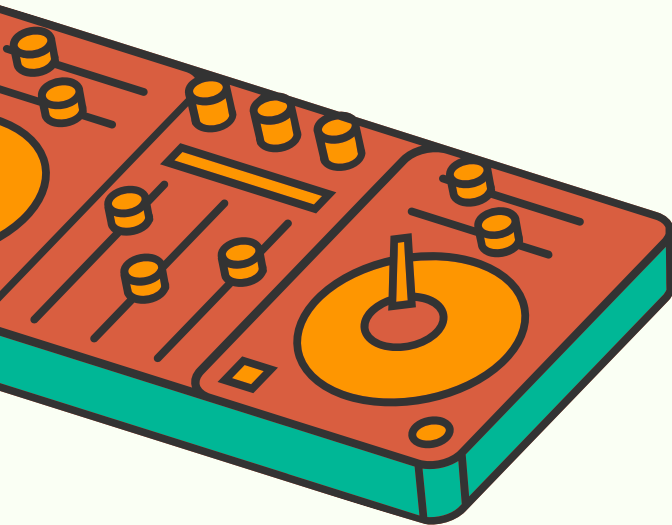
Q3: RETURN ALL THE TRACK NAMES THAT HAVE A SONG LENGTH LONGER THAN THE AVERAGE SONG LENGTH. RETURN THE NAME AND MILLISECONDS FOR EACH TRACK. ORDER BY THE SONG LENGTH WITH THE LONGEST SONGS LISTED FIRST.

- **Input :**

```
SELECT name, milliseconds
FROM track
WHERE milliseconds > (
  SELECT AVG(milliseconds) AS avg_track_length
  FROM track )
ORDER BY milliseconds DESC;
```

- **Output :**

	name	milliseconds
▶	How Many More Times	711836
	Advance Romance	677694
	Sleeping Village	644571
	You Shook Me(2)	619467
	Talkin' 'Bout Women Obviously	589531
	Stratus	582086
	No More Tears	555075
	The Alchemist	509413
	Wheels Of Confusion / The Straightener	494524
	Book Of Thel	494393
	You Oughta Know (Alternate)	491885



LEVEL - ADVANCE

Q1: FIND HOW MUCH AMOUNT SPENT BY EACH CUSTOMER ON ARTISTS? WRITE A QUERY TO RETURN CUSTOMER NAME, ARTIST NAME AND TOTAL SPENT.

- **Input :**

```
WITH best_selling_artist AS(
SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
FROM invoice_line
JOIN track ON track.track_id = invoice_line.track_id
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
GROUP BY 1 ORDER BY 3 DESC LIMIT 1)
SELECT c.customer_id, CONCAT(c.first_name,' ', c.last_name) AS customer,
bsa.artist_name,
SUM(il.unit_price*il.quantity) AS amount_spent FROM invoice i
JOIN customer c ON c.customer_id = i.customer_id
JOIN invoice_line il ON il.invoice_id = i.invoice_id
JOIN track t ON t.track_id = il.track_id
JOIN album alb ON alb.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
GROUP BY 1, 2, 3 ORDER BY 4 DESC;
```

- **Output :**




	customer_id	customer	artist_name	amount_spent
▶	54	Steve Murray	AC/DC	17.82
	53	Phil Hughes	AC/DC	10.89
	21	Kathy Chase	AC/DC	10.89
	49	Stanisław Wójcik	AC/DC	9.9
	1	Luís Gonzães	AC/DC	7.92
	24	Frank Ralston	AC/DC	7.92
	31	Martha Silk	AC/DC	3.96
	16	Frank Harris	AC/DC	2.97
	42	Wyatt Girard	AC/DC	2.97
	6	Helena Holm	AC/DC	2.97

Q2: WE WANT TO FIND OUT THE MOST POPULAR MUSIC GENRE FOR EACH COUNTRY. WE DETERMINE THE MOST POPULAR GENRE AS THE GENRE WITH THE HIGHEST AMOUNT OF PURCHASES. WRITE A QUERY THAT RETURNS EACH COUNTRY ALONG WITH THE TOP GENRE. FOR COUNTRIES WHERE THE MAXIMUM NUMBER OF PURCHASES IS SHARED RETURN ALL GENRES.

- **Input :**

```
WITH popular_genre AS (  
  SELECT COUNT(invoice_line.quantity) AS purchases,  
         customer.country, genre.name, genre.genre_id,  
         ROW_NUMBER() OVER(PARTITION BY customer.country  
                             ORDER BY COUNT(invoice_line.quantity) DESC) as RowNo  
  FROM invoice_line  
  JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id  
  JOIN customer ON customer.customer_id =  
    invoice.customer_id  
  JOIN track ON track.track_id = invoice_line.track_id  
  JOIN genre ON genre.genre_id = track.genre_id  
  GROUP BY 2,3,4  
  ORDER BY 2 ASC, 1 DESC)  
SELECT * FROM popular_genre WHERE RowNo <=1
```

- **Output :**



	purchases	country	name	genre_id	RowNo
▶	1	Argentina	Rock	1	1
	18	Australia	Rock	1	1
	6	Austria	Rock	1	1
	5	Belgium	Rock	1	1
	26	Brazil	Rock	1	1
	57	Canada	Rock	1	1
	7	Chile	Rock	1	1
	14	Czech Republic	Rock	1	1
	6	Denmark	Rock	1	1
	6	Finland	Rock	1	1
	26	France	Rock	1	1

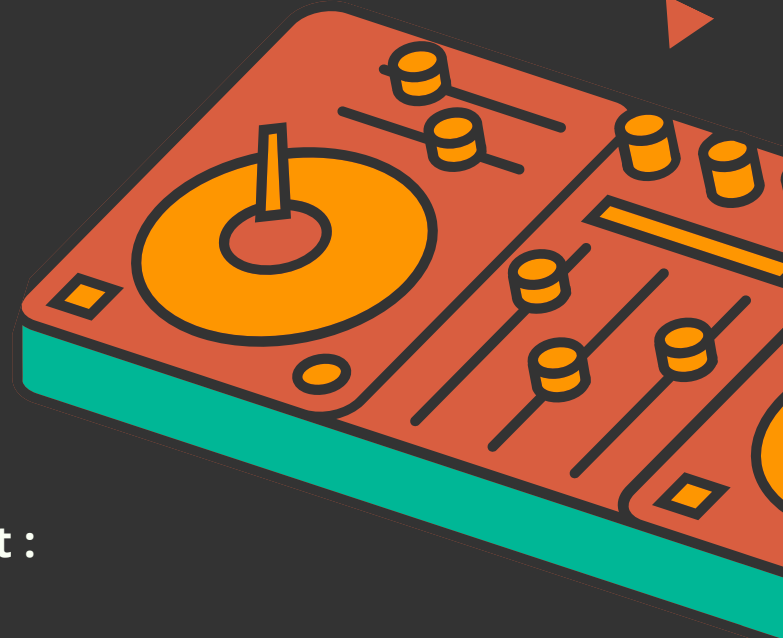
Q3: WRITE A QUERY THAT DETERMINES THE CUSTOMER THAT HAS SPENT THE MOST ON MUSIC FOR EACH COUNTRY. WRITE A QUERY THAT RETURNS THE COUNTRY ALONG WITH THE TOP CUSTOMER AND HOW MUCH THEY SPENT. FOR COUNTRIES WHERE THE TOP AMOUNT SPENT IS SHARED, PROVIDE ALL CUSTOMERS WHO SPENT THIS AMOUNT.

- **Input :**

```
WITH customer_with_country AS (
SELECT customer.customer_id, CONCAT(first_name,' ',
last_name) AS customer_name, billing_country, SUM(total)
AS total_spending,
ROW_NUMBER() OVER(PARTITION BY billing_country ORDER
BY SUM(total) DESC) AS RowNo
FROM invoice
JOIN customer ON customer.customer_id =
invoice.customer_id
GROUP BY 1,2,3
ORDER BY 3 ASC, 4 DESC)
SELECT * FROM customer_with_country WHERE RowNo <= 1
```

- **Output :**

	customer_id	customer_name	billing_country	total_spending	RowNo
▶	56	Diego Guti��rrez	Argentina	39.6	1
	55	Mark Taylor	Australia	81.18	1
	7	Astrid Gruber	Austria	69.3	1
	8	Daan Peeters	Belgium	60.38999999999999	1
	1	Lu��s Gon��salves	Brazil	108.89999999999998	1
	3	Fran��ois Tremblay	Canada	99.99	1
	57	Luis Rojas	Chile	97.02000000000001	1
	5	Franti��ek Wichterlov��i	Czech Republic	144.54000000000002	1
	9	Kara Nielsen	Denmark	37.61999999999999	1
	44	Terhi H��m��inen	Finland	79.2	1
	42	Wyatt Girard	France	99.99	1





RESULTS:

1. THE MOST POPULAR GENRE: ROCK
2. THE MOST POPULAR ARTIST: AC/DC
3. THE LONGEST SONG IN THE PLAYLIST: "HOW
MANY MORE TIMES"
4. THE AVERAGE ALBUM PRICE: \$1
5. THE TOP COUNTRY FOR MUSIC PURCHASES:
UNITED STATES



*THANK
YOU!*

BY JATIN KUMAR

