SQL PROJECT

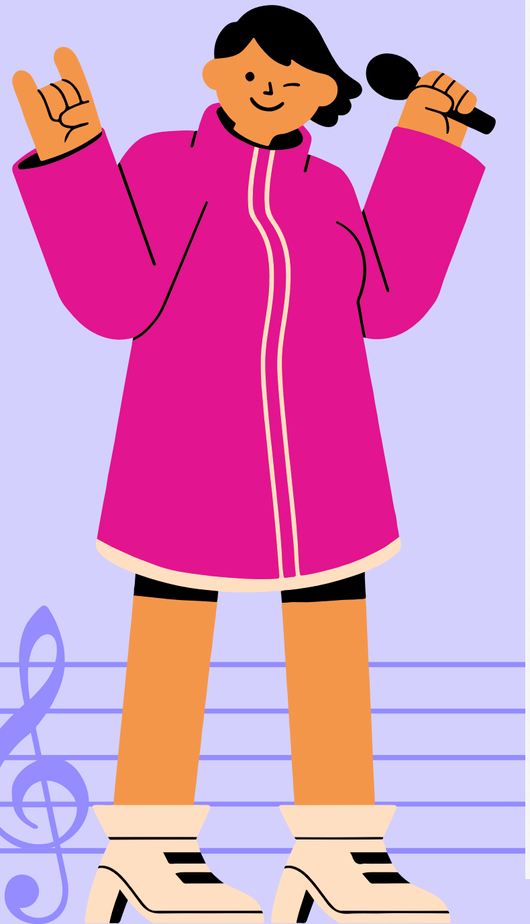
Music Store Analysis

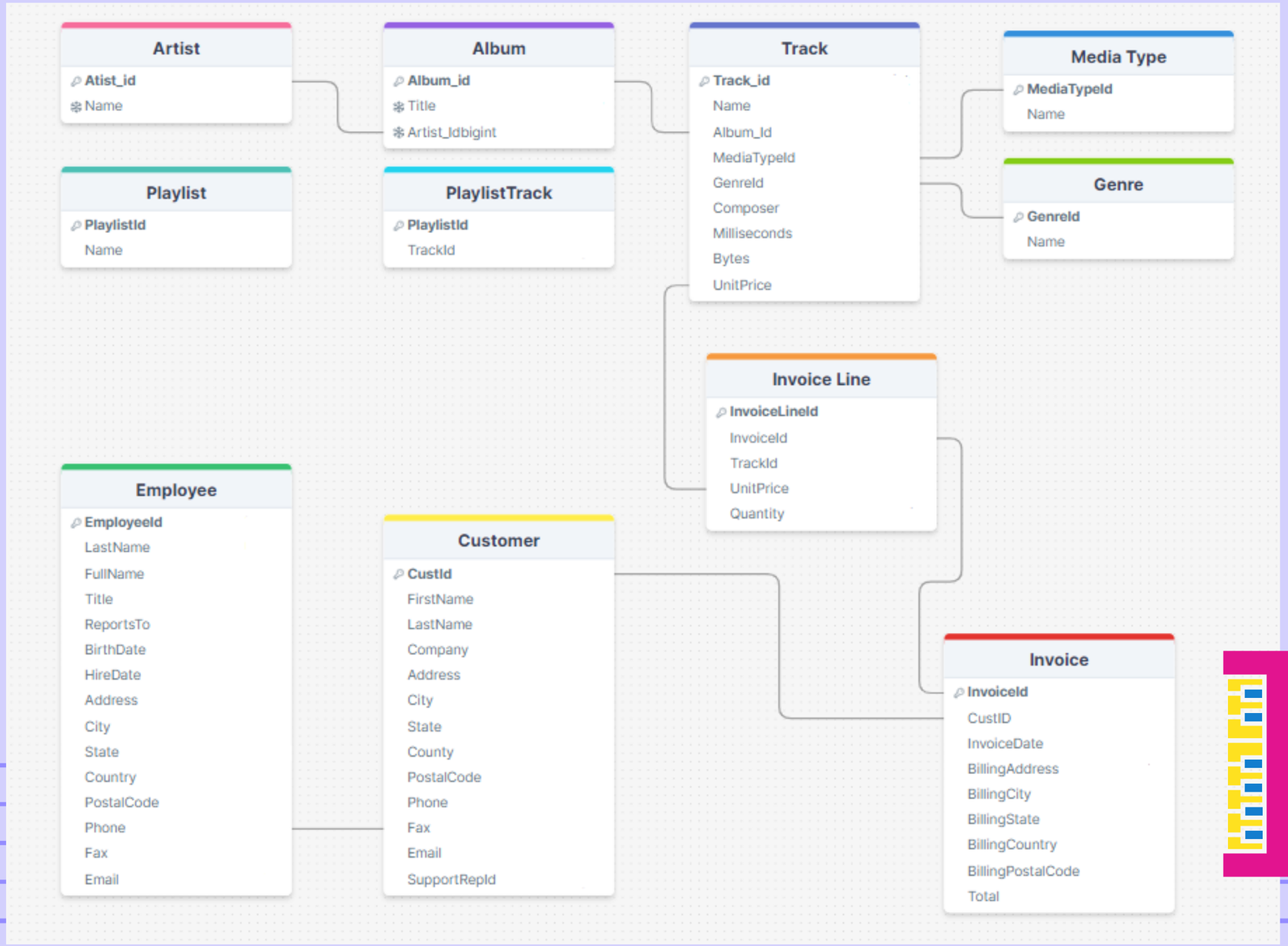
-Naman Kamra

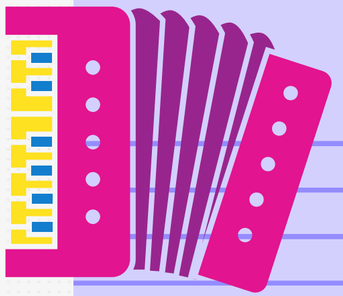
Objective

- THE PROJECT AIMS TO ANALYZE A DIGITAL MUSIC STORE DATABASE USING SQL, FOCUSING ON PROVIDING STAKEHOLDERS WITH VALUABLE INSIGHTS FOR EFFECTIVE DECISION-MAKING.
- THROUGH SQL QUERIES, THE PROJECT ADDRESSES A RANGE OF QUESTIONS, INCLUDING GEOGRAPHICAL GROWTH, PURCHASE POWER, TOTAL REVENUE, GENRE PERFORMANCE, AND MUSIC BAND POPULARITY.
- BY ANSWERING BOTH SIMPLE AND COMPLEX QUESTIONS, THE ANALYSIS OFFERS ACTIONABLE RECOMMENDATIONS TO DRIVE BUSINESS GROWTH AND OPTIMIZE PERFORMANCE.

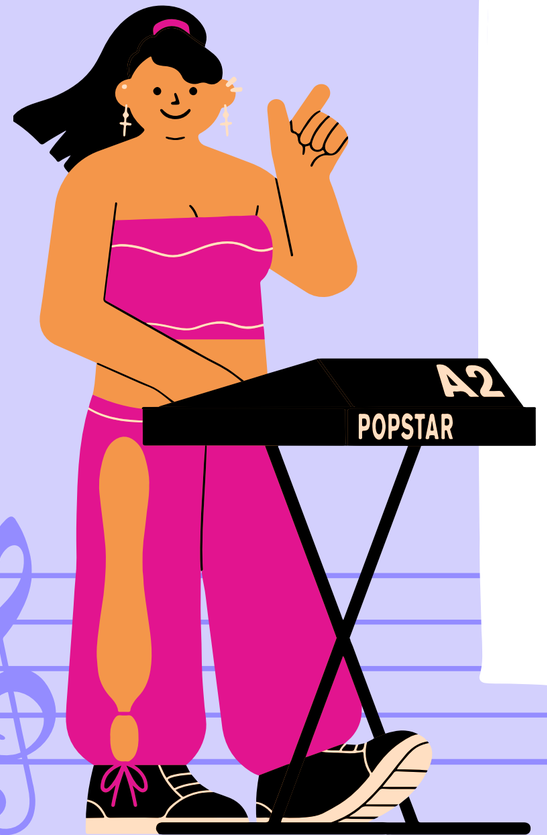
Database Schema







Level Of Queries



EASY

Includes:

Select, Group By, Order By , Limit, Desc/Asc

MODERATE

Includes:

Joins, Order By, Group By, Limits

ADVANCED

Includes:

CTE(Common Table Expression)

EASY

Query

```
SELECT*FROM employee
ORDER BY levels desc
limit 1
```

Who is the senior most employee based on job title?

Output

	title character varying (50)	last_name character	first_name character
1	Senior General Manager	Madan	Mohan

EASY

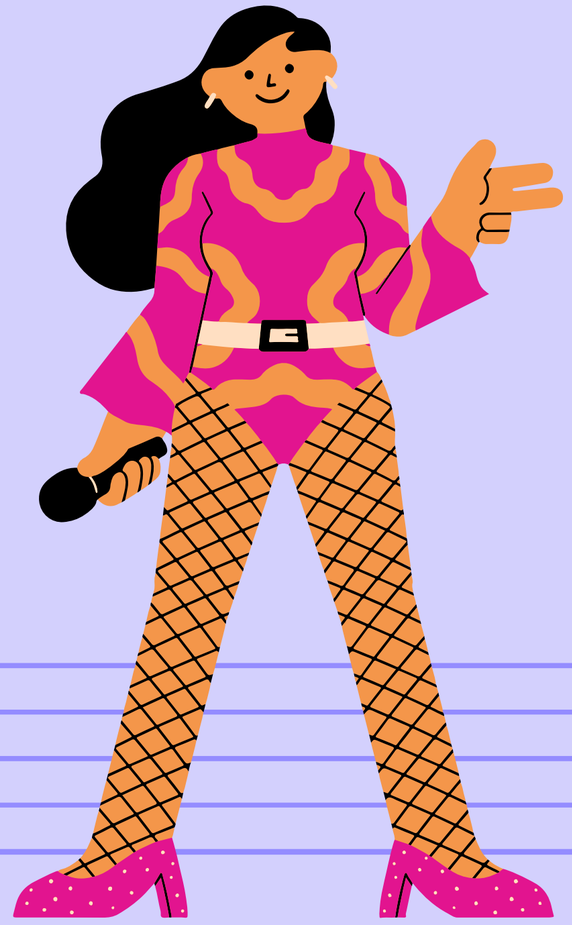
Query

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

Which countries have the most Invoices?

Output

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile
11	13	Ireland
12	11	Spain
13	11	Finland
14	10	Australia



EASY

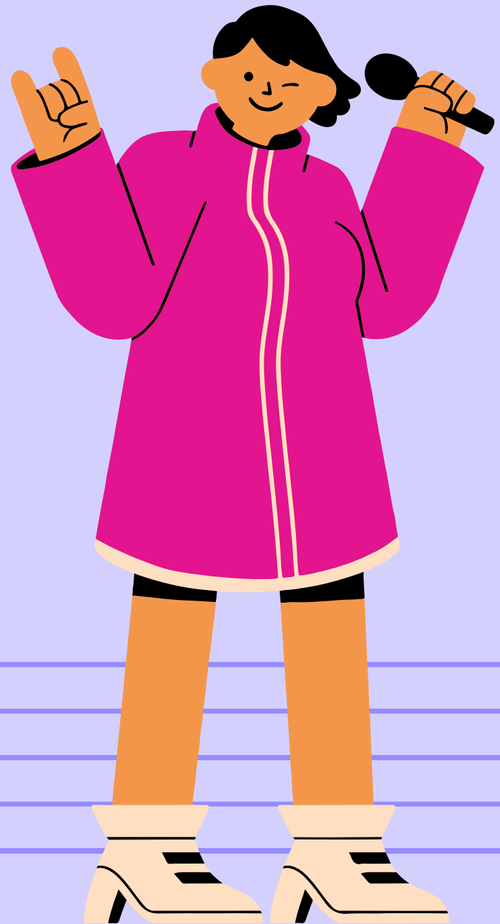
What are top 3 values of total invoice?

Query

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

Output

	total double precision
1	23.759999999999998
2	19.8
3	19.8



EASY

Query

```
SELECT SUM(total) AS invoice_total, billing_city
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC
LIMIT 1
```

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

Output

	invoice_total double precision	billing_city character varying (30)
1	273.24000000000007	Prague

EASY

Query

```
SELECT customer.customer_id,customer.first_name,customer.last_name,
SUM(invoice.total)AS TOTAL
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
Group by customer.customer_id
ORDER BY TOTAL DESC
LIMIT 1
```

1. **Who is the best customer? (The customer who has spent the most money will be declared the best customer)**
2. **Write a query that returns the person who has spent the most money.**

Output

Add row

	customer_id [K] integer	first_name character	last_name character	total double precision
1	5	R ...	Madhav	144.54000000000002

MODERATE

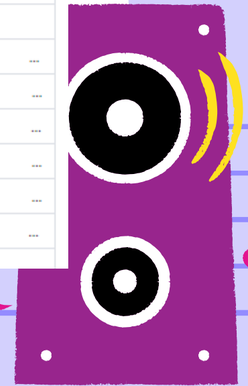
Query

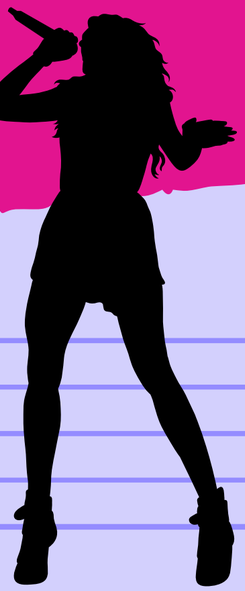
```
SELECT DISTINCT email,first_name,last_name
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoice_line on invoice.invoice_id = invoice_line.invoice_id
JOIN track on invoice_line.track_id = track.track_id
JOIN genre on track.genre_id = genre.genre_id
WHERE genre.name = 'Rock'
ORDER BY email;
```

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

Output

	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan
13	emma_jones@hotmail.com	Emma	Jones
14	enrique_munoz@yahoo.es	Enrique	Muñoz
15	fernadaramos4@uol.com.br	Fernanda	Ramos





MODERATE

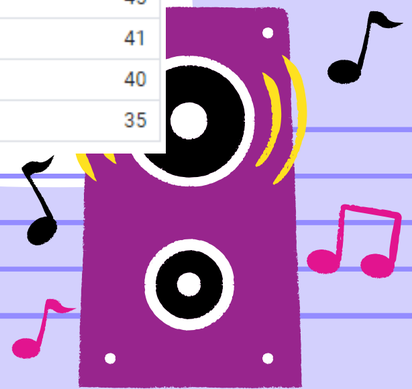
Query

```
SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS number_of_songs
FROM track
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
GROUP BY artist.artist_id
ORDER BY number_of_songs DESC
LIMIT 10;
```

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

Output

	artist_id [PK] character varying (50)	name character varying (120)	number_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35



MODERATE

Query

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

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

Output

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677

ADVANCED

Query

```
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, c.first_name, c.last_name, 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 1,2,3,4
ORDER BY 5 DESC;
```

Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

Output

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.830000000000002
5	53	Phil	Hughes	Queen	11.88
6	41	Marc	Dubois	Queen	11.88
7	47	Lucas	Mancini	Queen	10.89
8	33	Ellie	Sullivan	Queen	10.89
9	20	Dan	Miller	Queen	3.96
10	5	R	Madhav	Queen	3.96

ADVANCED

Query

```
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
```

- 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.

Output

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character vary	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1

ADVANCED

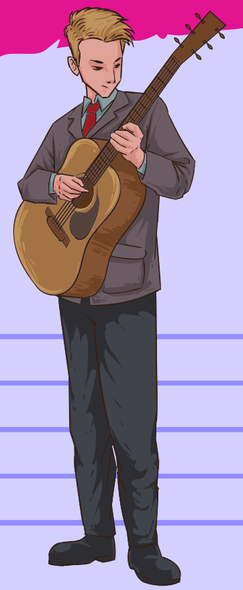
Query

```
WITH Customter_with_country AS (
        SELECT customer.customer_id,first_name,last_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 1,2,3,4
        ORDER BY 4 ASC,5 DESC)
SELECT * FROM Customter_with_country WHERE RowNo <= 1
```

- 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.

Output

	customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spending double precision	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999999999	1
5	1	Luís	Gonçalves	Brazil	108.8999999999999	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000001	1
8	5	R	Madhav	Czech Republic	144.5400000000000	1
9	9	Kara	Nielsen	Denmark	37.61999999999999	1
10	44	Terhi	Hämäläine...	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmerma...	Germany	94.05000000000001	1



Thank you!

nkamra12345@gmail.com