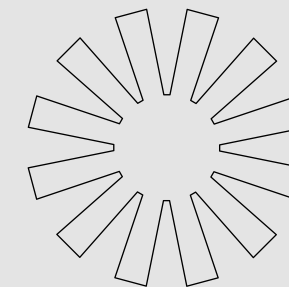
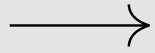


THE
MUSIC
STORE
ANALYSIS

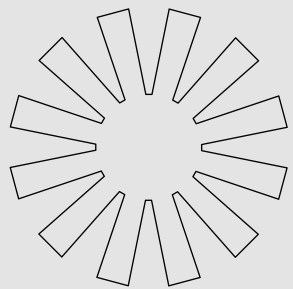


CREATED BY MAYUR GOYAL



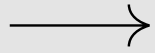
INTRODUCTION

This is our music store analysis. This presentation will explore current trends, market dynamics, and consumer behavior in the music retail sector. We'll provide key insights and data to help stakeholders make informed decisions. Join us as we uncover the evolving landscape of music stores.





03



AMOUNT SPENT BY EACH CUSTOMER

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

```
SELECT
  c.customer_id,
  c.first_name,
  c.last_name,
  a.name AS 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
  album2 alb ON alb.album_id = t.album_id
JOIN
  artist a ON a.artist_id = alb.artist_id
GROUP BY
  c.customer_id, c.first_name, c.last_name, a.name
ORDER BY
  amount_spent DESC;
```

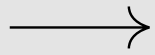
QUERY

OUTPUT

customer_id	first_name	last_name	artist_name	amount_spent
54	Steve	Murray	AC/DC	17.82
15	Jennifer	Peterson	Aerosmith	14.8500000000000001
55	Mark	Taylor	Aerosmith	14.8500000000000001
13	Fernanda	Ramos	Antônio Carlos Jobim	13.8600000000000001
2	Leonie	Köhler	Audioslave	13.8600000000000001



04



BEST CUSTOMER

```
-- Who is the best customer?

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,
    customer.first_name,
    customer.last_name
ORDER BY
    total DESC
LIMIT 1;
```

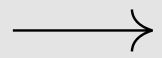
QUERY

OUTPUT

customer_id	first_name	last_name	total
5	František	Wichterlovský	144.54000000000002



05



CITY OF BEST CUSTOMERS

```
-- Which city has the best customers?
```

```
SELECT
```

```
    SUM(total) AS invoice_total,  
    billing_city
```

```
FROM
```

```
    invoice
```

```
GROUP BY
```

```
    billing_city
```

```
ORDER BY
```

```
    invoice_total DESC
```

```
LIMIT 1;
```

QUERY

OUTPUT

invoice_total

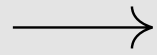
billing_city

273.2400000000000007

Prague



06



MOST INVOICES COUNTRY

```
-- Which countries have the most invoices?
```

```
SELECT
```

```
    COUNT(*) AS c,  
    billing_country
```

```
FROM
```

```
    invoice
```

```
GROUP BY
```

```
    billing_country
```

```
ORDER BY
```

```
    c DESC;
```

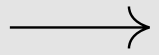
QUERY

OUTPUT

c	billing_country
131	USA
76	Canada
61	Brazil
50	France
41	Germany



07



MOST POPULAR GENRE

```
-- 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).

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
    customer.country, genre.name, genre.genre_id
  ORDER BY
    customer.country ASC, purchases DESC
)

SELECT * FROM popular_genre WHERE RowNo <= 1;
```

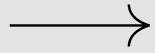
QUERY

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



08

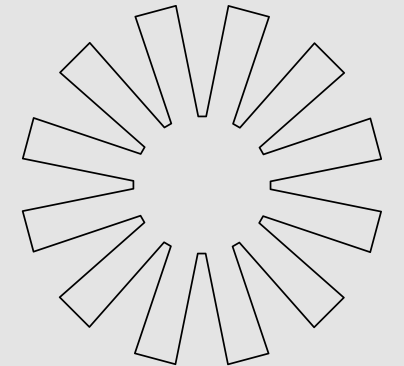


MOST SPENDING CUSTOMERS BY COUNTRY

```
-- 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.
```

```
WITH RECURSIVE Customer_with_country AS (  
  SELECT  
    customer.customer_id,  
    first_name,  
    last_name,  
    billing_country,  
    SUM(total) AS total_spending  
  FROM  
    invoice  
  JOIN  
    customer ON customer.customer_id = invoice.customer_id  
  GROUP BY  
    customer.customer_id, first_name, last_name, billing_country  
  ORDER BY  
    customer.customer_id, total_spending DESC  
)  
Country_max_spending AS (  
  SELECT  
    billing_country,  
    MAX(total_spending) AS max_spending  
  FROM  
    Customer_with_country  
  GROUP BY  
    billing_country
```

```
SELECT  
  cc.billing_country,  
  cc.total_spending,  
  cc.first_name,  
  cc.last_name  
FROM  
  Customer_with_country cc  
JOIN  
  Country_max_spending ms ON cc.billing_country = ms.billing_country  
WHERE  
  cc.total_spending = ms.max_spending  
ORDER BY  
  cc.billing_country;
```



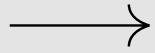
QUERY

OUTPUT

billing_country	total_spending	first_name	last_name
Argentina	39.6	Diego	Gutiérrez
Australia	81.18	Mark	Taylor
Austria	69.3	Astrid	Gruber
Belgium	60.38999999999999	Daan	Peeters
Brazil	108.89999999999998	Luís	Gonçalves



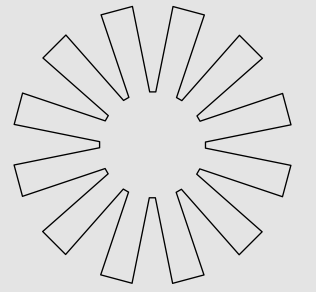
09



ROCK GENRE

```
-- 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  
  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  
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;
```



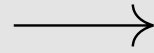
QUERY

OUTPUT

email	first_name	last_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



10



— SENIOR MOST EMPLOYEE

```
-- Who is the senior most employee based on job title?
```

```
SELECT
```

```
    *
```

```
FROM
```

```
    employee
```

```
ORDER BY
```

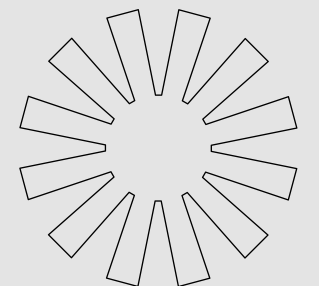
```
    levels DESC
```

```
LIMIT 1;
```

QUERY

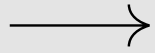
OUTPUT

employee_id	last_name	first_name	title	reports_to	levels	birthdate	hire_date	address	city	state	country	postal_code	phone	fax	email
1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00	14-08-2016 00:00	11120 Jasper Ave NW	Edmonton	AB	Canada	T5K 2N1	+1 (780) 428-9482	+1 (780) 428-3457	andrew@chinookcorp.com





11



— TOP 3 OF TOTAL VALUES

```
-- What are top 3 values of total invoice?
```

```
SELECT
```

```
    total
```

```
FROM
```

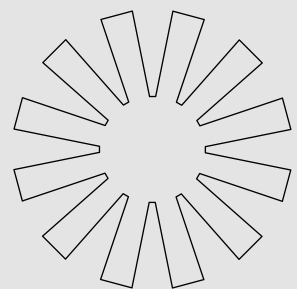
```
    invoice
```

```
ORDER BY
```

```
    total DESC
```

```
LIMIT 3;
```

QUERY

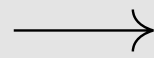


OUTPUT

total
23.759999999999998
19.8
19.8



12



TOP 10 ROCK BANDS

```
-- Write a query that returns the artist name and total track count of the top 10 rock bands.
```

```
SELECT
  artist.artist_id,
  artist.name,
  COUNT(artist.artist_id) AS number_of_songs
FROM
  track
  JOIN album2 ON album2.album_id = track.track_id
  JOIN artist ON artist.artist_id = album2.artist_id
  JOIN genre ON genre.genre_id = track.genre_id
WHERE
  genre.name LIKE 'Rock'
GROUP BY
  artist.artist_id,
  artist.name
ORDER BY
  number_of_songs DESC
LIMIT 10;
```

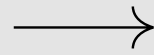
QUERY

artist_id	name	number_of_songs
58	Deep Purple	7
90	Iron Maiden	5
21	Various Artists	4
27	Gilberto Gil	3
88	Guns N' Roses	3
1	AC/DC	2
2	Accept	2
19	Cidade Negra	2
6	Antônio Carlos Jobim	2
16	Caetano Veloso	2

OUTPUT



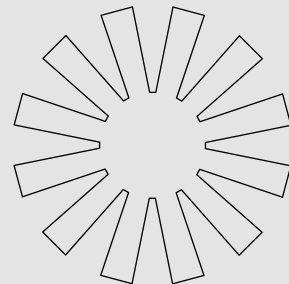
13



TRACKS LENGTH

```
-- 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.
```

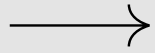
```
SELECT  
    name,  
    milliseconds  
FROM  
    track  
WHERE  
    milliseconds > (  
        SELECT  
            AVG(milliseconds) AS avg_track_length  
        FROM  
            track  
    )  
ORDER BY  
    milliseconds DESC;
```



QUERY

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



CONCLUSION

In summary, the music store industry is evolving rapidly with new trends and technologies. Our analysis provides valuable insights to help stakeholders navigate these changes and seize opportunities. Thank you for your attention, and we welcome your questions.

