**Project: SQL Analysis of Music store**

**Project Overview**

This project involves analyzing an invoice dataset using SQL queries. The goal is to extract meaningful insights from the data, such as identifying the best customers, most popular music genres by country, and best-selling artists. The analysis is performed using PostgreSQL.

**Exploratory Data Analysis (EDA)**

**1. Senior Most Employee Based on Job Title**

SELECT \*

FROM employee

ORDER BY levels DESC

LIMIT 1;

|  |  |  |  |
| --- | --- | --- | --- |
| employee\_id | last\_name | first\_name | title |
| 9 | Madan | Mohan | Senior General Manager |

This query retrieves the most senior employee based on job title levels.

**2. Country with the Most Invoices**

SELECT billing\_country, COUNT(billing\_country) AS count

FROM invoice

GROUP BY billing\_country

ORDER BY count DESC

LIMIT 1;

|  |  |
| --- | --- |
| billing\_country | count |
| USA | 131 |

This helps identify the country with the highest number of invoices.

**3. Top 3 Invoice Amounts**

SELECT billing\_city, total

FROM invoice

ORDER BY total DESC

LIMIT 3;

|  |  |
| --- | --- |
| billing\_city | total |
| Bordeaux | 23.76 |
| Winnipeg | 19.8 |
| MontrÃ©al | 19.8 |

This provides the top 3 invoice values along with the corresponding billing cities.

**4. City with the Best Customers (Highest Revenue)**

SELECT billing\_city, SUM(total) AS totall

FROM invoice

GROUP BY billing\_city

ORDER BY totall DESC

LIMIT 1;

|  |  |
| --- | --- |
| billing\_city | totall |
| Prague | 273.24 |

This query identifies the city that has generated the most revenue.

**5. Best Customer (Highest Spender)**

SELECT c.customer\_id, c.first\_name, c.last\_name, SUM(i.total) AS total

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY c.customer\_id

ORDER BY total DESC

LIMIT 1;

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_id | first\_name | last\_name | total |
| 5 | R | Madhav | 144.54 |

This retrieves the customer who has spent the most money on invoices.

**6. Rock Music Listeners**

SELECT DISTINCT c.email, c.first\_name, c.last\_name, g.name

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

JOIN track t ON il.track\_id = t.track\_id

JOIN genre g ON t.genre\_id = g.genre\_id

WHERE g.name LIKE 'Rock'

ORDER BY c.email;

|  |  |  |  |
| --- | --- | --- | --- |
| email | first\_name | last\_name | name |
| aaronmitchell@yahoo.ca | Aaron | Mitchell | Rock |
| alero@uol.com.br | Alexandre | Rocha | Rock |
| astrid.gruber@apple.at | Astrid | Gruber | Rock |
| bjorn.hansen@yahoo.no | BjÃ¸rn | Hansen | Rock |
| camille.bernard@yahoo.fr | Camille | Bernard | Rock |
| daan\_peeters@apple.be | Daan | Peeters | Rock |
| diego.gutierrez@yahoo.ar | Diego | GutiÃ©rrez | Rock |
| dmiller@comcast.com | Dan | Miller | Rock |
| dominiquelefebvre@gmail.com | Dominique | Lefebvre | Rock |

This query lists all customers who listen to rock music.

**7. Top 10 Rock Artists by Track Count**

SELECT a.name, COUNT(a.artist\_id) AS total\_track

FROM artist a

JOIN album al ON a.artist\_id = al.artist\_id

JOIN track t ON t.album\_id = al.album\_id

JOIN genre g ON g.genre\_id = t.genre\_id

WHERE g.name LIKE 'Rock'

GROUP BY a.artist\_id

ORDER BY total\_track DESC

LIMIT 10;

|  |  |  |
| --- | --- | --- |
| name | total\_track | artist\_id |
| Led Zeppelin | 114 | 22 |
| U2 | 112 | 150 |
| Deep Purple | 92 | 58 |
| Iron Maiden | 81 | 90 |
| Pearl Jam | 54 | 118 |
| Van Halen | 52 | 152 |
| Queen | 45 | 51 |
| The Rolling Stones | 41 | 142 |
| Creedence Clearwater Revival | 40 | 76 |
| Kiss | 35 | 52 |

This helps identify the top 10 rock music artists in terms of the number of tracks.

**8. Songs Longer Than Average Length**

SELECT name, milliseconds AS song\_length

FROM track

WHERE milliseconds > (SELECT AVG(milliseconds) FROM track)

ORDER BY song\_length DESC;

|  |  |
| --- | --- |
| name | song\_length |
| Occupation / Precipice | 5286953 |
| Through a Looking Glass | 5088838 |
| Greetings from Earth, Pt. 1 | 2960293 |
| The Man With Nine Lives | 2956998 |
| Battlestar Galactica, Pt. 2 | 2956081 |
| Battlestar Galactica, Pt. 1 | 2952702 |
| Murder On the Rising Star | 2935894 |
| Battlestar Galactica, Pt. 3 | 2927802 |
| Take the Celestra | 2927677 |

Retrieves all songs that are longer than the average track length.

**9. Amount Spent by Customers on Best-Selling Artist**

WITH best\_selling\_artist AS (

SELECT a.artist\_id, a.name, SUM(il.unit\_price \* il.quantity) AS artist\_total

FROM artist a

JOIN album al ON a.artist\_id = al.artist\_id

JOIN track t ON al.album\_id = t.album\_id

JOIN invoice\_line il ON t.track\_id = il.track\_id

GROUP BY 1

ORDER BY 3 DESC

LIMIT 1

)

SELECT c.customer\_id, c.first\_name, c.last\_name, bsa.name, SUM(il.unit\_price \* il.quantity) AS amount\_spent

FROM customer c

JOIN invoice i 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 al ON al.album\_id = t.album\_id

JOIN artist a ON a.artist\_id = al.artist\_id

JOIN best\_selling\_artist bsa ON bsa.artist\_id = al.artist\_id

GROUP BY 1,2,3,4

ORDER BY 5 DESC;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| customer\_id | first\_name | last\_name | name | amount\_spent |
| 46 | Hugh | O'Reilly | Queen | 27.72 |
| 38 | Niklas | SchrÃ¶der | Queen | 18.81 |
| 3 | FranÃ§ois | Tremblay | Queen | 17.82 |
| 34 | JoÃ£o | Fernandes | Queen | 16.83 |
| 53 | Phil | Hughes | Queen | 11.88 |
| 41 | Marc | Dubois | Queen | 11.88 |

This identifies how much each customer has spent on the best-selling artist.

**10. Most Popular Music Genre in Each Country**

WITH most\_popular\_genre AS (

SELECT c.country, g.name, COUNT(il.quantity) AS total\_genre,

ROW\_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) AS rownum

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

JOIN track t ON il.track\_id = t.track\_id

JOIN genre g ON t.genre\_id = g.genre\_id

GROUP BY c.country, g.name

)

SELECT \*

FROM most\_popular\_genre

WHERE rownum <= 1;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| country | name | genre\_id | total\_genre | rownum |
| USA | Rock | 1 | 561 | 1 |
| Canada | Rock | 1 | 333 | 1 |
| France | Rock | 1 | 211 | 1 |
| Brazil | Rock | 1 | 205 | 1 |
| Germany | Rock | 1 | 194 | 1 |
| United Kingdom | Rock | 1 | 166 | 1 |
| Czech Republic | Rock | 1 | 143 | 1 |
| Portugal | Rock | 1 | 108 | 1 |
| India | Rock | 1 | 102 | 1 |
| Ireland | Rock | 1 | 72 | 1 |

This query finds the most popular music genre in each country based on purchase count.

**11. Highest-Spending Customer in Each Country**

WITH top\_customer AS (

SELECT c.customer\_id, c.first\_name, c.last\_name, i.billing\_country, SUM(i.total) AS total\_spending,

ROW\_NUMBER() OVER(PARTITION BY i.billing\_country ORDER BY SUM(i.total) DESC) AS rownum

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

GROUP BY 1,2,3,4

)

SELECT \*

FROM top\_customer

WHERE rownum <= 1;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| customer\_id | first\_name | last\_name | billing\_country | total\_spending | rownum |
| 56 | Diego | GutiÃ©rrez | Argentina | 39.6 | 1 |
| 55 | Mark | Taylor | Australia | 81.18 | 1 |
| 7 | Astrid | Gruber | Austria | 69.3 | 1 |
| 8 | Daan | Peeters | Belgium | 60.39 | 1 |
| 1 | LuÃ­s | GonÃ§alves | Brazil | 108.9 | 1 |
| 3 | FranÃ§ois | Tremblay | Canada | 99.99 | 1 |
| 57 | Luis | Rojas | Chile | 97.02 | 1 |
| 5 | R | Madhav | Czech Republic | 144.54 | 1 |
| 9 | Kara | Nielsen | Denmark | 37.62 | 1 |

This retrieves the top-spending customer in each country.

This project provides a comprehensive SQL-based analysis of an invoice dataset, extracting key business insights and actionable recommendations. 🎵📊