Studio Shodwe

SQL Project

data analtics project on music store

Studio Shodwe

Objective

This project is for biginners and will teach us how to analyze the music playlist database. we can examine the dataset with SQL and help the store understand its business growth by answering simple questions

Learn More

Studio Shodwe

who is the senior most employee based on job title

```sql
SELECT
    *
FROM
    employee
ORDER BY levels DESC
LIMIT 1;
```

WHICH COUNTRIES HAVE THE MOST INVOICE

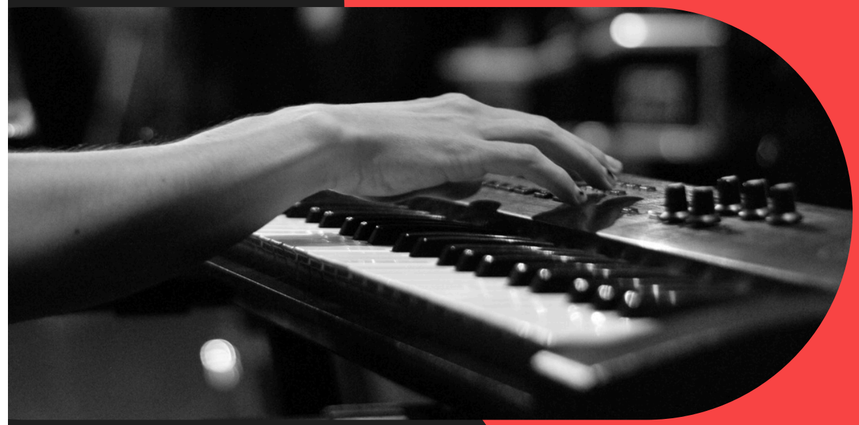
```sql
SELECT
    COUNT(*) AS count_invoice, billing_country
FROM
    invoice
GROUP BY billing_country
ORDER BY count_invoice DESC;
```

WHAT ARE TOP 3 VALUES OF TOTAL INVOICE

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

Studio Shodwe

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 WHO HAS SPENT THE MOST MONEY.



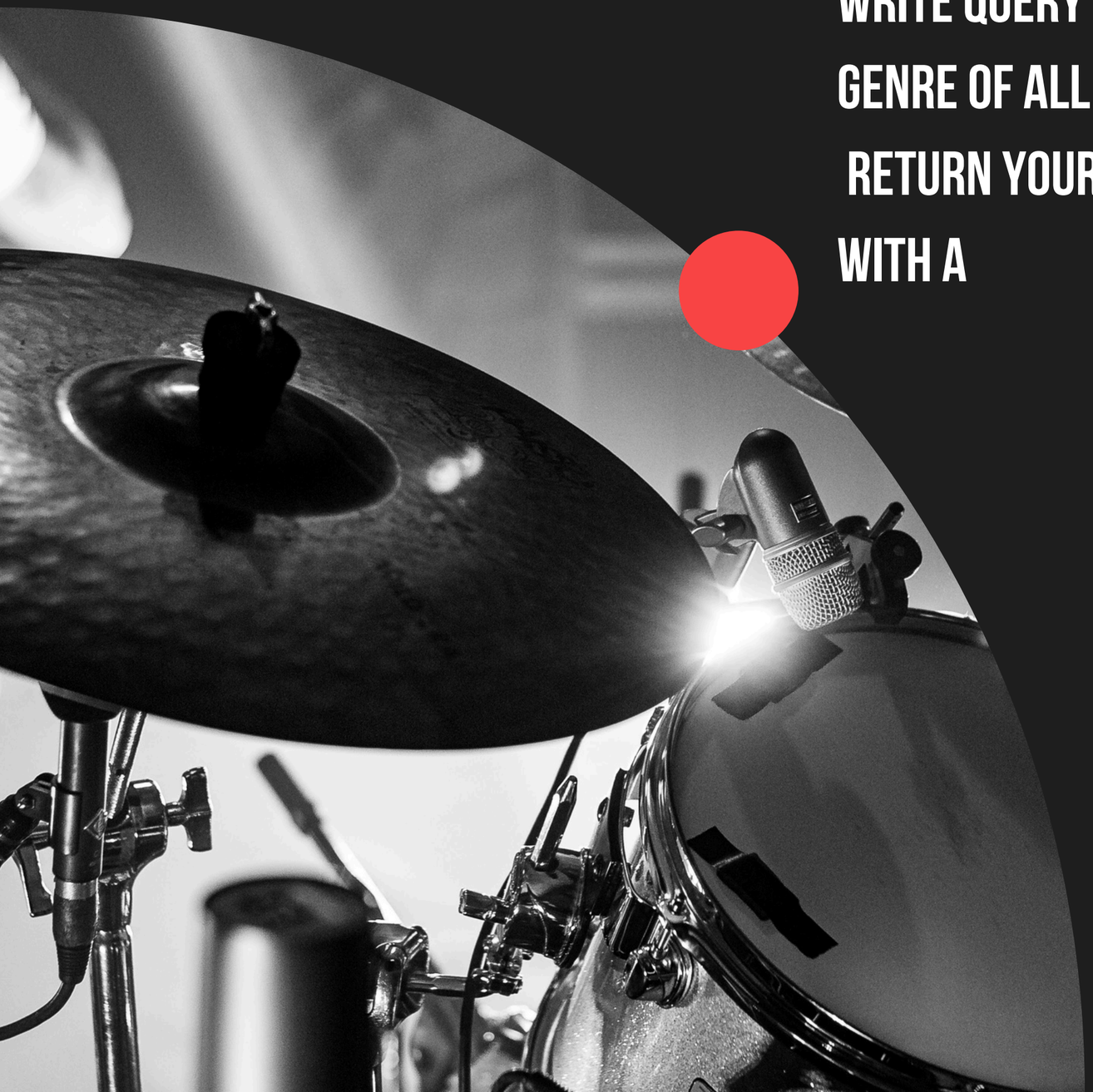
```sql
SELECT
    customer.customer_id,
    customer.first_name,
    customer.last_name,
    SUM(invoice.total) AS total
FROM
    customer
        JOIN
    invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id , customer.first_name , customer.last_name
ORDER BY total DESC
LIMIT 1;
```

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 HIGEST SUM OF INVOICE TOTAL. RETURN BOTH THE CITY NAME AND SUM OF ALL INVOICE TOTALS

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

Studio Shodwe



WRITE QUERY TO RETURN THE EMAIL, FIRST NAME , LAST NAME AND GENRE OF ALL ROCK MUSIC LISTENERS.

RETURN YOUR LIST ORDERED ALPHABETIALLY BY EMAIL STARTING WITH A

```sql
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;
```

LET'S INVITE THE ARTISTS WHO HAVE WRITTEN THE MOST ROCK MUSIC IN OUR DATASE.
WRITE A QUERY THAT RETURN THE ARTIST NAME AND TOTAL TRACK COUNT OF THE TOP 10 ROCK BANDS

```sql
SELECT
    artist.artist_id,
    artist.name,
    COUNT(artist.artist_id) AS number_of_songs
FROM
    track
        JOIN
    album2 ON album2.album_id = track.album_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
ORDER BY number_of_songs DESC
LIMIT 10;
```

return all the the track names that havea song length longer than the avg song length. return the name and millisecond for each track
order by the song length with the longest songs listed first

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