SQL PROJECT

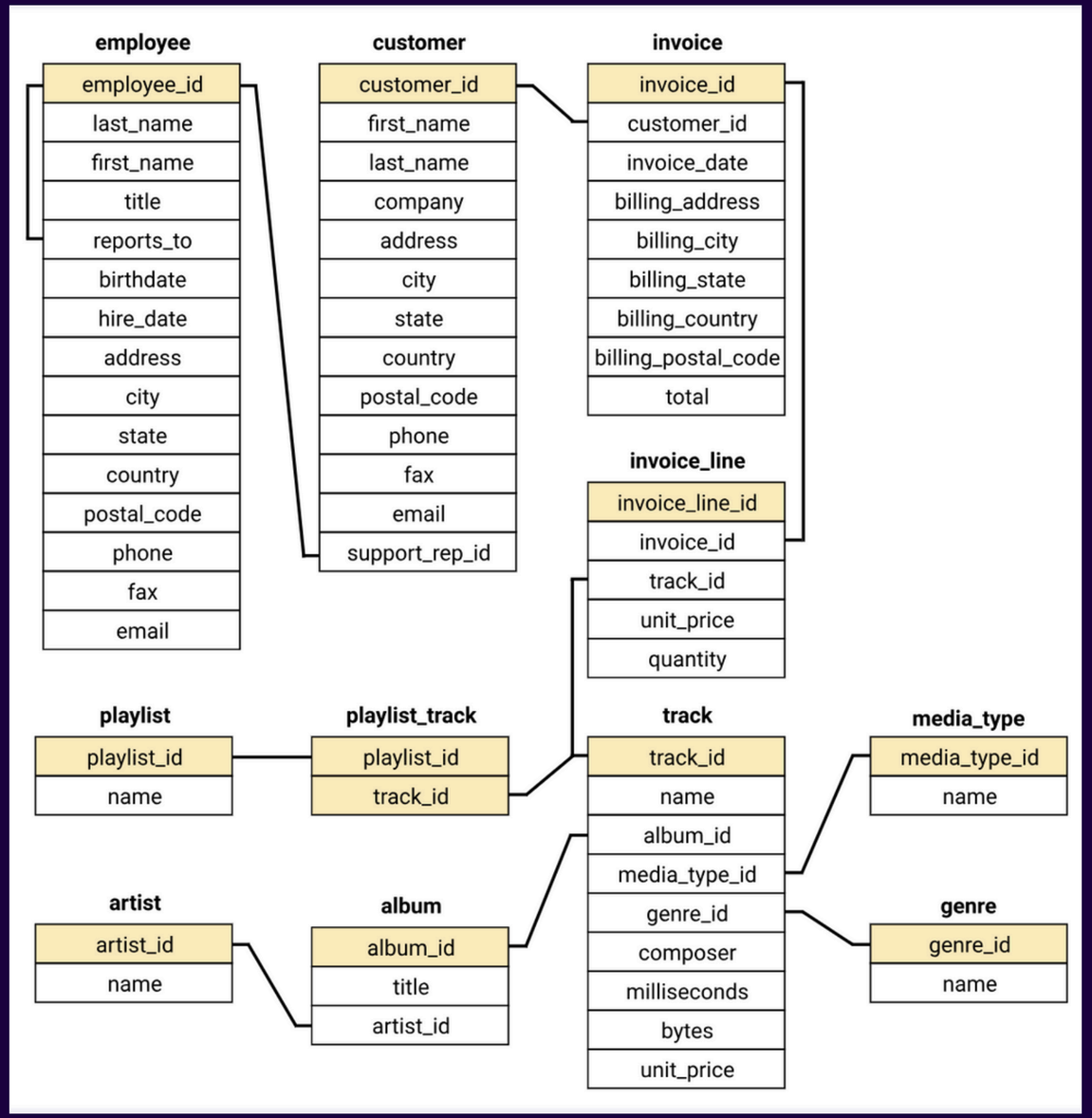
MUSIC STORE DATABASE

INTRODUCTION

This project focuses on analyzing a Music Store Database using SQL to uncover meaningful business insights. The database represents a digital music store that records customer purchases, invoices, artists, albums, and tracks. By applying SQL queries, we can answer critical business questions.

The objective of this project is to transform raw data into actionable insights that can help improve marketing strategies, customer engagement, and overall sales performance.

DATABASE SCHEMA



SQL QUERIES & BUSINESS QUESTIONS

Easy Queries

- Senior-most employee based on job title.
- Country with the most invoices.
- Top 3 highest invoice totals.
- City with the best customers.
- The customer who spent the most money.

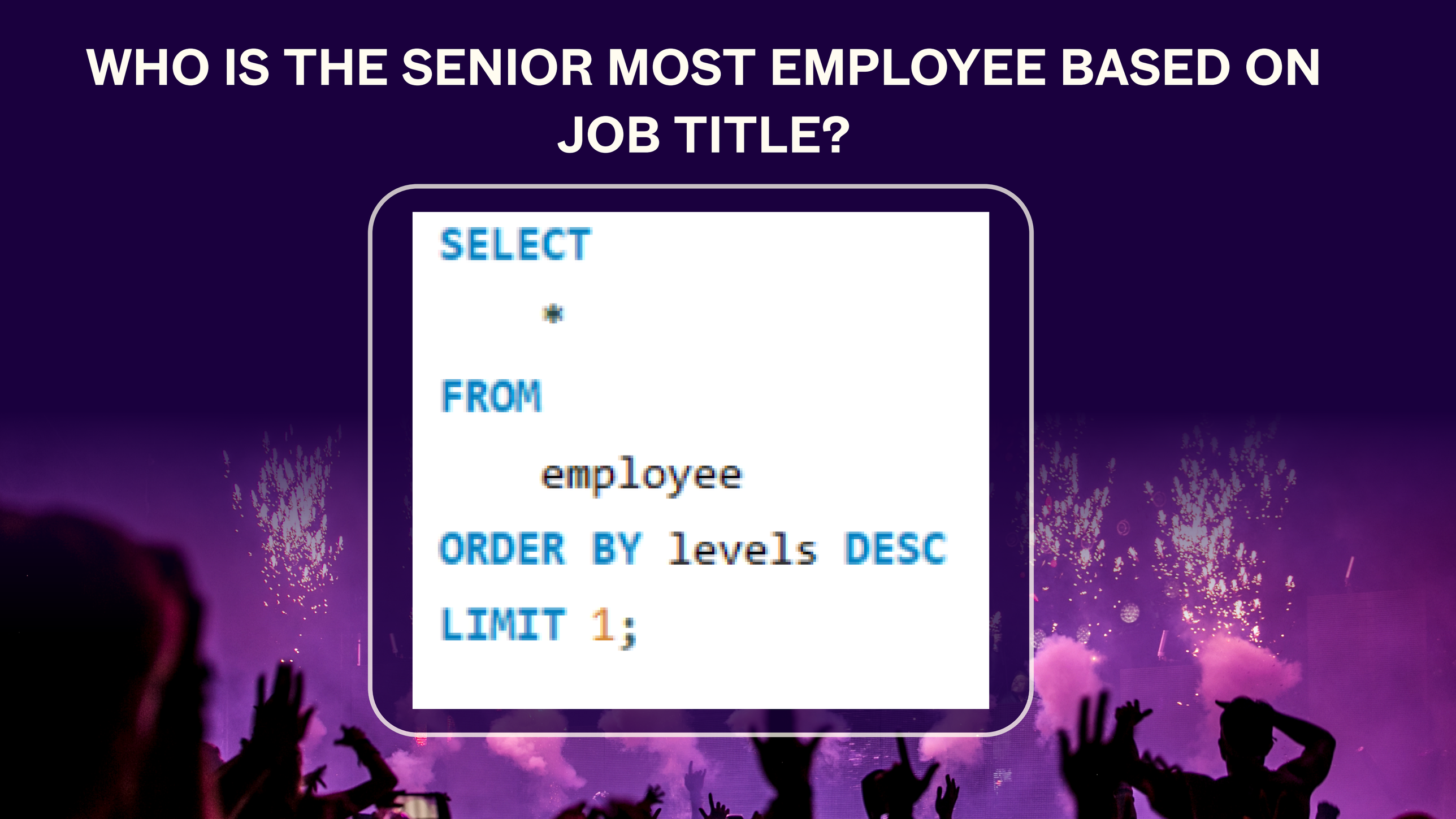
Moderate Queries

- Rock music listeners (emails, names).
- Top 10 artists with the most rock tracks.
- Tracks longer than the average song length.

Advanced Queries

- Amount spent by each customer on top artists.
- Most popular genre for each country.
- Top customer per country by total spending.

WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

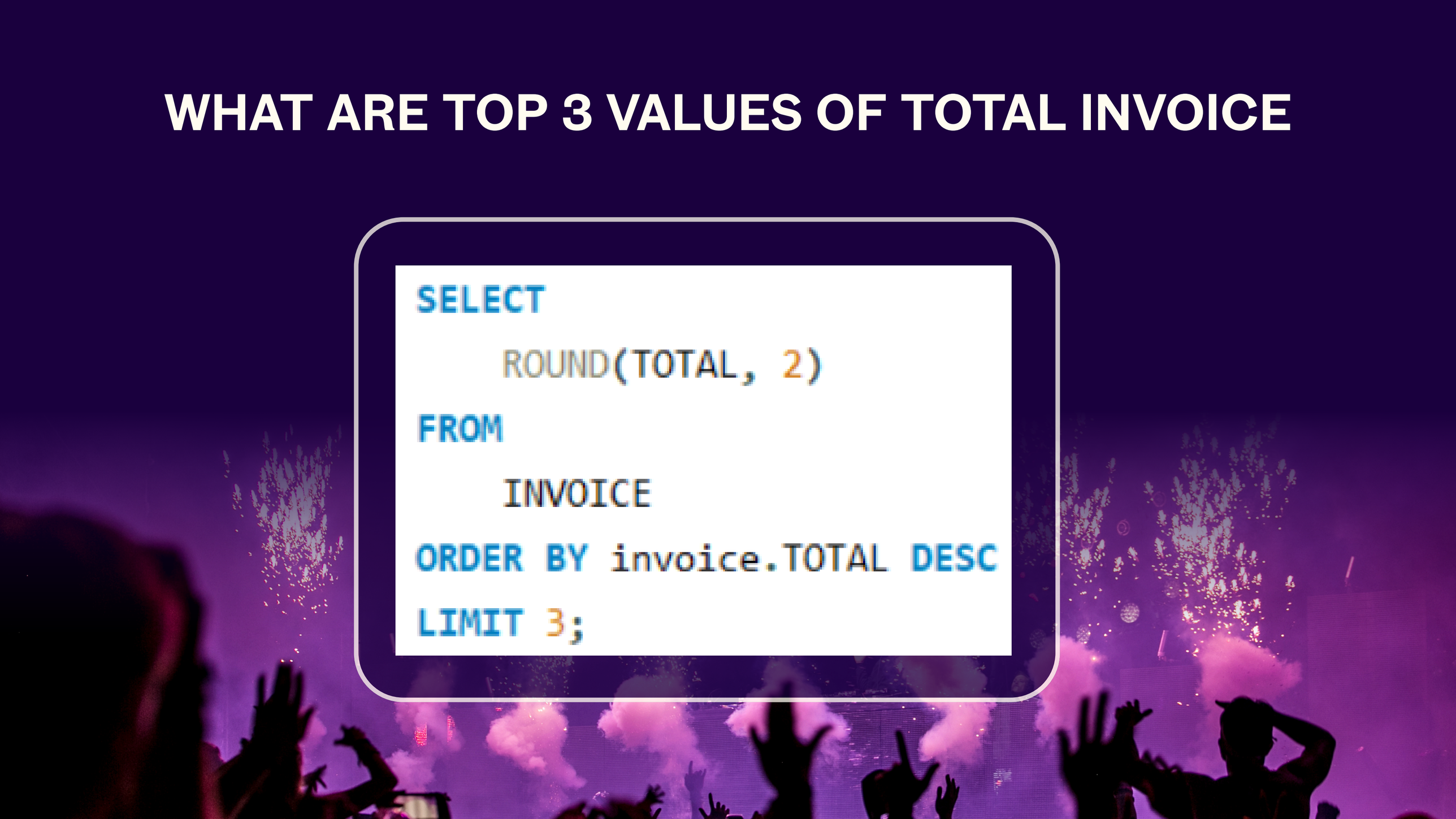


```sql
SELECT
    *
FROM
    employee
ORDER BY levels DESC
LIMIT 1;
```

WHICH COUNTRIES HAVE THE MOST INVOICES?

```sql
SELECT
    billing_country, COUNT(invoice_id) total_invoices
FROM
    invoice
GROUP BY billing_country
ORDER BY COUNT(invoice_id) DESC
LIMIT 1;
```

WHAT ARE TOP 3 VALUES OF TOTAL INVOICE

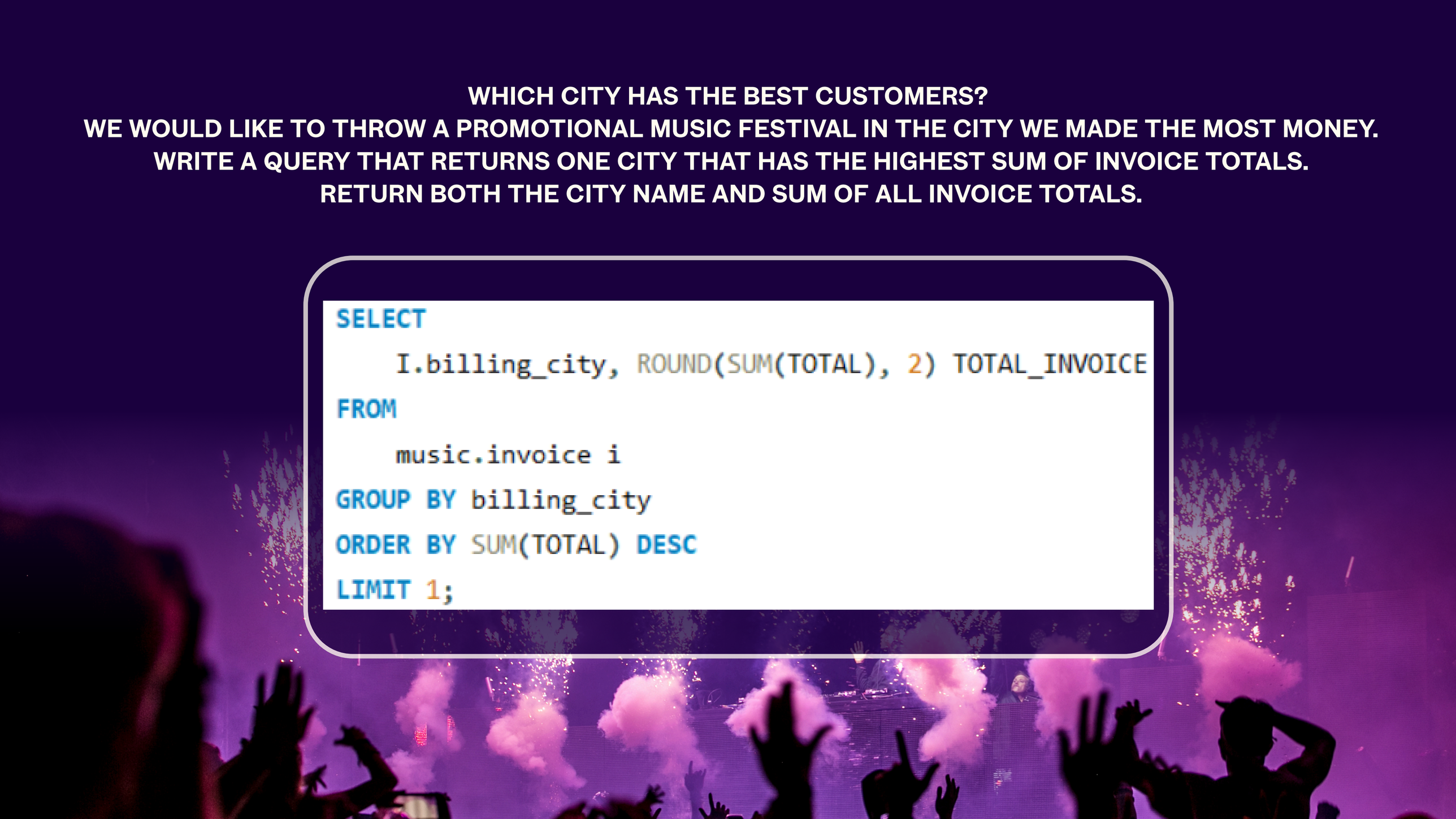


```sql
SELECT
    ROUND(TOTAL, 2)
FROM
    INVOICE
ORDER BY invoice.TOTAL DESC
LIMIT 3;
```

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 AND SUM OF ALL INVOICE TOTALS.

```sql
SELECT
    I.billing_city, ROUND(SUM(TOTAL), 2) TOTAL_INVOICE
FROM
    music.invoice i
GROUP BY billing_city
ORDER BY SUM(TOTAL) DESC
LIMIT 1;
```



WHO IS THE BEST CUSOTMER?

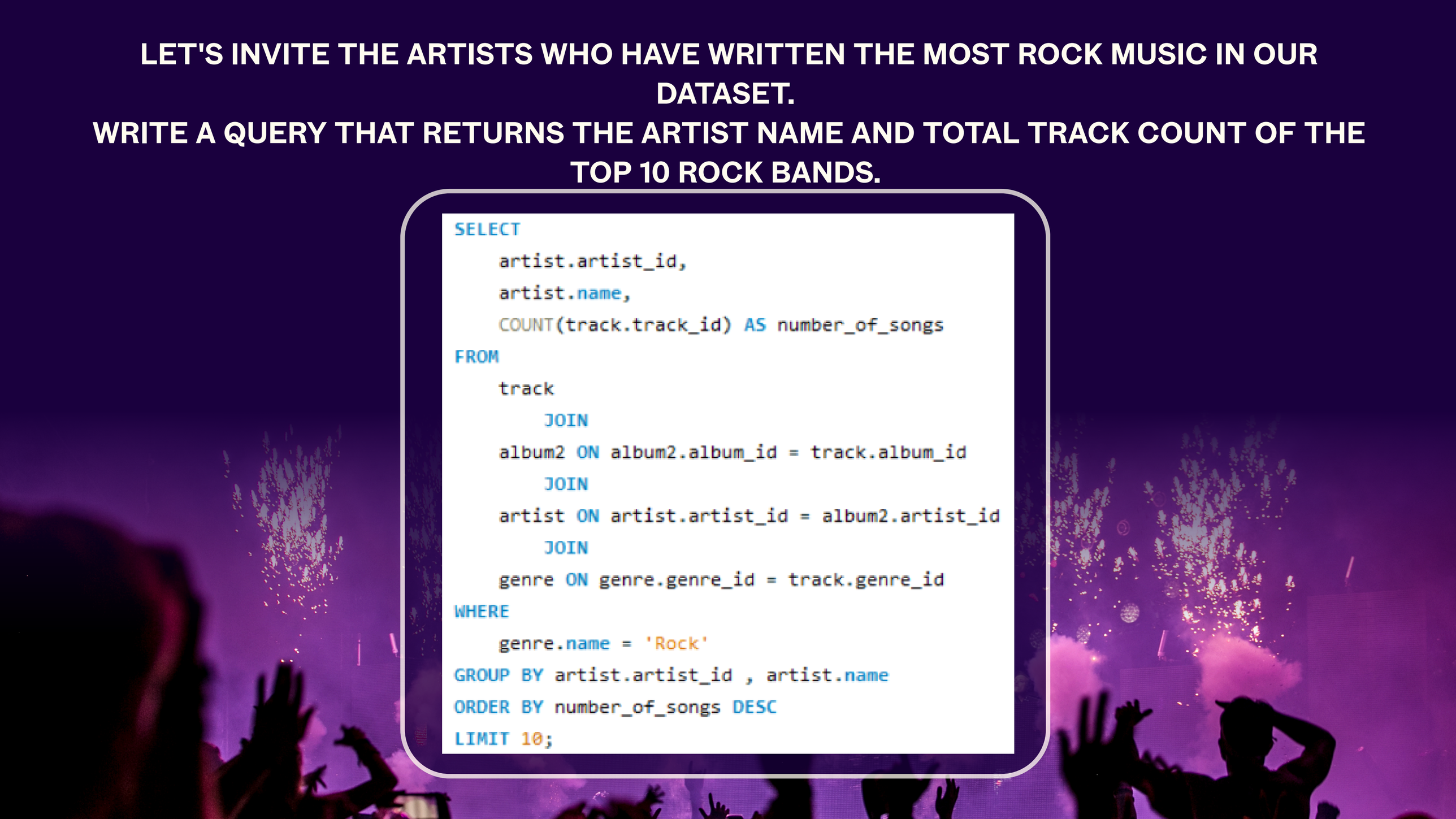
THE CUSTOMER WHO HAS SPENT THE MOST MONEY WILL BE DECLARED THE BEST CUSTOMER.

WRITE A QUERY THAT RETURNS THE PERSON WHO HAS SPENT THE MOST MONEY.

```sql
SELECT
    CONCAT(C.FIRST_NAME, ' ', C.LAST_NAME) FULL_NAME,
    ROUND(SUM(TOTAL), 2) TOTAL_SPEND
FROM
    CUSTOMER C
        LEFT JOIN
    INVOICE I ON C.CUSTOMER_ID = I.CUSTOMER_ID
GROUP BY CONCAT(C.FIRST_NAME, ' ', C.LAST_NAME)
ORDER BY SUM(TOTAL) DESC
LIMIT 1;
```

WRITE QUERY TO RETURN THE EMAIL,FIRST NAME, LAST NAME AND GENRE OF ALL ROCK MUSIC LISTENERS.
RETURN YOUR LIST ORDERED ALPHABETICALLY BY EMAIL STARTING WITH A.

```sql
SELECT DISTINCT
    (C.EMAIL), C.FIRST_NAME, C.LAST_NAME
FROM
    CUSTOMER C
        LEFT JOIN
    invoice I ON C.customer_id = I.CUSTOMER_ID
        LEFT JOIN
    invoice_line IL ON IL.invoice_id = I.invoice_id
WHERE
    track_id IN (SELECT
            track_id
        FROM
            track T
                LEFT JOIN
            genre G ON G.genre_id = T.genre_id
        WHERE
            G.NAME = 'ROCK')
ORDER BY email;
```



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.

```sql
SELECT
    artist.artist_id,
    artist.name,
    COUNT(track.track_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 = 'Rock'
GROUP BY artist.artist_id , artist.name
ORDER BY number_of_songs DESC
LIMIT 10;
```

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.

```sql
SELECT
    NAME, MILLISECONDS
FROM
    TRACK
WHERE
    MILLISECONDS > (SELECT
            ROUND(AVG(milliseconds), 2) AVG_MILLISECONDS
        FROM
            TRACK)
ORDER BY MILLISECONDS DESC
LIMIT 1;
```

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

```sql
WITH BEST_SELLING_ARTIST AS (
    SELECT
        ar.artist_id,
        ar.name AS artist_name,
        SUM(il.unit_price * il.quantity) AS total_sales
    FROM invoice_line il
    JOIN track t ON t.track_id = il.track_id
    JOIN album al ON al.album_id = t.album_id
    JOIN artist ar ON ar.artist_id = al.artist_id
    GROUP BY ar.artist_id, ar.name
    ORDER BY total_sales 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 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 album al
    ON t.album_id = al.album_id
JOIN BEST_SELLING_ARTIST bsa
    ON bsa.artist_id = al.artist_id
GROUP BY c.customer_id, c.first_name, c.last_name, bsa.artist_name
ORDER BY amount_spent DESC;
```

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 PURCHASE.
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.

```sql
with popular_genre as
(
select row_number() over(partition by c.country order by count(il.quantity) desc) as rownumber,
 c.country,g.name, g.genre_id,count(il.quantity) as total_purchases
from customer c
left join invoice i
on c.customer_id = i.invoice_id
left join invoice_line il
on i.invoice_id = il.invoice_id
left join track t
on il.track_id = t.track_id
left join genre g
on t.genre_id = g.genre_id
group by c.country,g.name, g.genre_id
order by 2 asc, 5 desc
)
select rownumber,country,total_purchases from popular_genre where rownumber <=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 CUSTOMER WHO SPENT THIS AMOUNT.

```sql
with customer_with_country 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 billing_country order by sum(total) desc) as rownumber
from customer c
left join invoice i
on c.customer_id = i.customer_id
group by 1,2,3,4
order by 4 asc,5 desc
)
select * from customer_with_country where rownumber<=1;
```

CONCLUSION

Through this project, we uncovered who the best customers are, which countries and cities drive the most revenue, and which music genres dominate global sales. These insights reflect the real-world value of SQL in guiding business strategies.

The analysis doesn’t just explain what happened—it also builds the foundation for predicting future trends and shaping more personalized, customer-focused strategies.


THANK YOU
by: Hritik Mishra