



DATA ANALYSIS

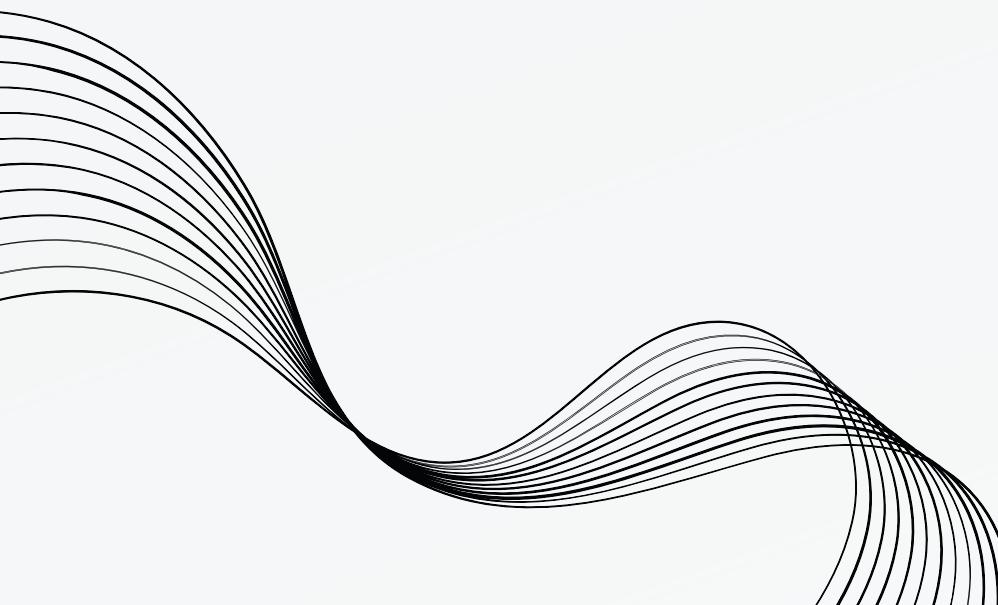
MUSIC STORE

[HTTPS://GITHUB.COM/WORKWITHSHREESH](https://github.com/WorkWithShreesh)

WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

```
select * from employee;
```

```
select * from employee  
    order by levels desc  
limit 1;
```



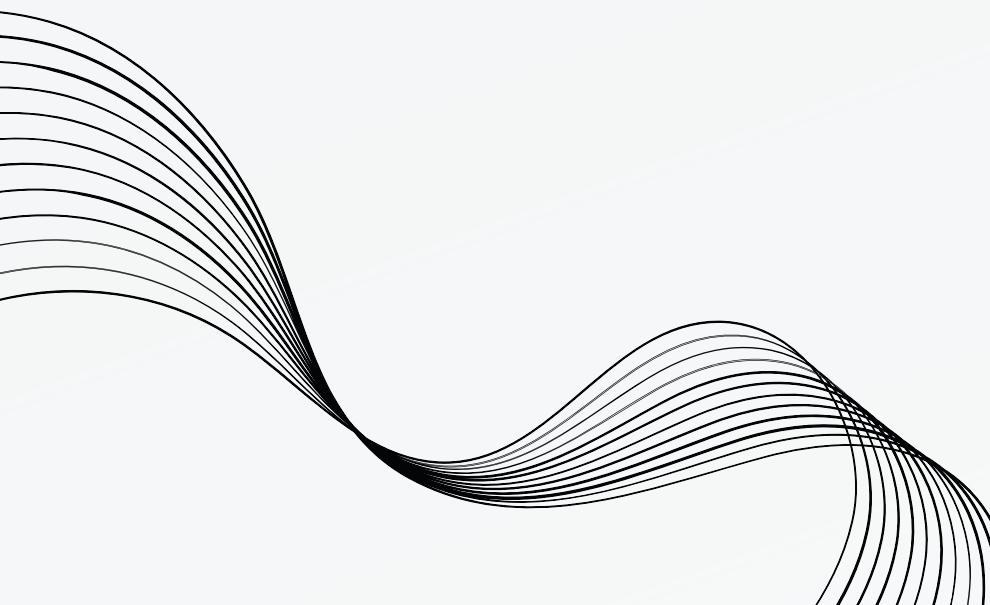
WHICH COUNTRIES HAVE THE MOST INVOICES?

```
select * from invoice;  
  
select count(*) as coun, billing_country from  
invoice group by billing_country  
order by coun desc;
```



WHAT ARE TOP 3 VALUES OF TOTAL INVOICE?

```
select * from invoice;  
  
select *, round(total::numeric,2) as total_round from invoice  
order by 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 & SUM OF ALL INVOICE TOTALS

```
select * from invoice;  
|  
select sum(round(total::numeric,2)) as sum_total, billing_city  
      from invoice  
      group by billing_city  
      order by sum_total desc limit 3;
```

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

```
SELECT (c.first_name || ' ' || c.last_name) as
       customer_name, i.total
  FROM customer c
 JOIN invoice i ON c.customer_id = i.customer_id
 ORDER BY i.total DESC
 LIMIT 1;
```

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;
```

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

```
SELECT artist.name, COUNT(track.track_id) AS total_rock_tracks
FROM artist
JOIN album ON artist.artist_id = album.artist_id
JOIN track ON album.album_id = track.album_id
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name = 'Rock'
GROUP BY artist.name
ORDER BY total_rock_tracks 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

```
select name, milliseconds
from track where milliseconds >
(select avg(milliseconds) as avg_track_length
from track) order by
milliseconds desc;
```

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

```
SELECT
    customer.first_name || ' ' || customer.last_name AS customer_name,
    artist.name AS artist_name,
    SUM(invoice_line.unit_price * invoice_line.quantity) AS total_spent
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 album ON track.album_id = album.album_id
JOIN artist ON album.artist_id = artist.artist_id
GROUP BY customer_name, artist_name
ORDER BY customer_name, total_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 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

```
WITH GenrePurchaseCount AS (SELECT customer.country,
genre.name AS genre_name,COUNT(invoice_line.invoice_line_id)
AS purchase_count 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
GROUP BY customer.country, genre.name
),MaxGenrePurchaseCount AS (SELECT
country, MAX(purchase_count) AS max_purchase_count
FROM GenrePurchaseCount GROUP BY country)
SELECT gpc.country, gpc.genre_name, gpc.purchase_count
FROM GenrePurchaseCount gpc
JOIN MaxGenrePurchaseCount mgpc ON gpc.country = mgpc.country
AND gpc.purchase_count = mgpc.max_purchase_count
ORDER BY gpc.country, gpc.genre_name;
```

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 CustomerSpending AS (
    SELECT customer.customer_id, customer.country,
    customer.first_name || ' ' || customer.last_name AS customer_name,
    SUM(invoice.total) AS total_spent FROM customer
    JOIN invoice ON customer.customer_id = invoice.customer_id
    GROUP BY customer.customer_id, customer.country, customer_name
), MaxSpending AS (SELECT country, MAX(total_spent) AS max_spent
    FROM CustomerSpending GROUP BY country )
SELECT cs.country, cs.customer_name, cs.total_spent
    FROM CustomerSpending cs JOIN MaxSpending ms ON
    cs.country = ms.country AND cs.total_spent = ms.max_spent
    ORDER BY cs.country, cs.customer_name;
```

THANK YOU FOR YOUR ATTENTION

- Want to explore the project further?
- Visit our GitHub repository for the full code and documentation.
- Access SQL queries, datasets, and project details.
- We welcome your contributions and feedback!

github.com/workwithshreesh



LARANA, INC.