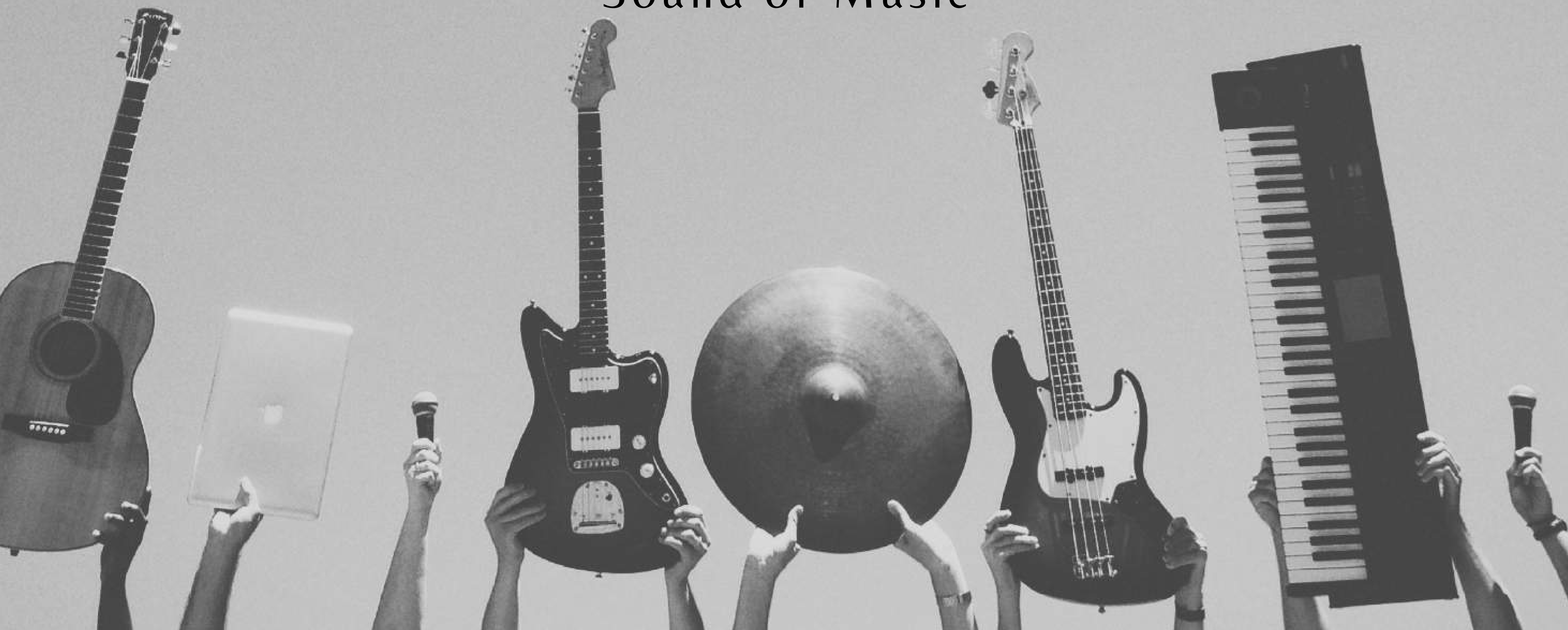
WELCOME TO MUSIC WORLD

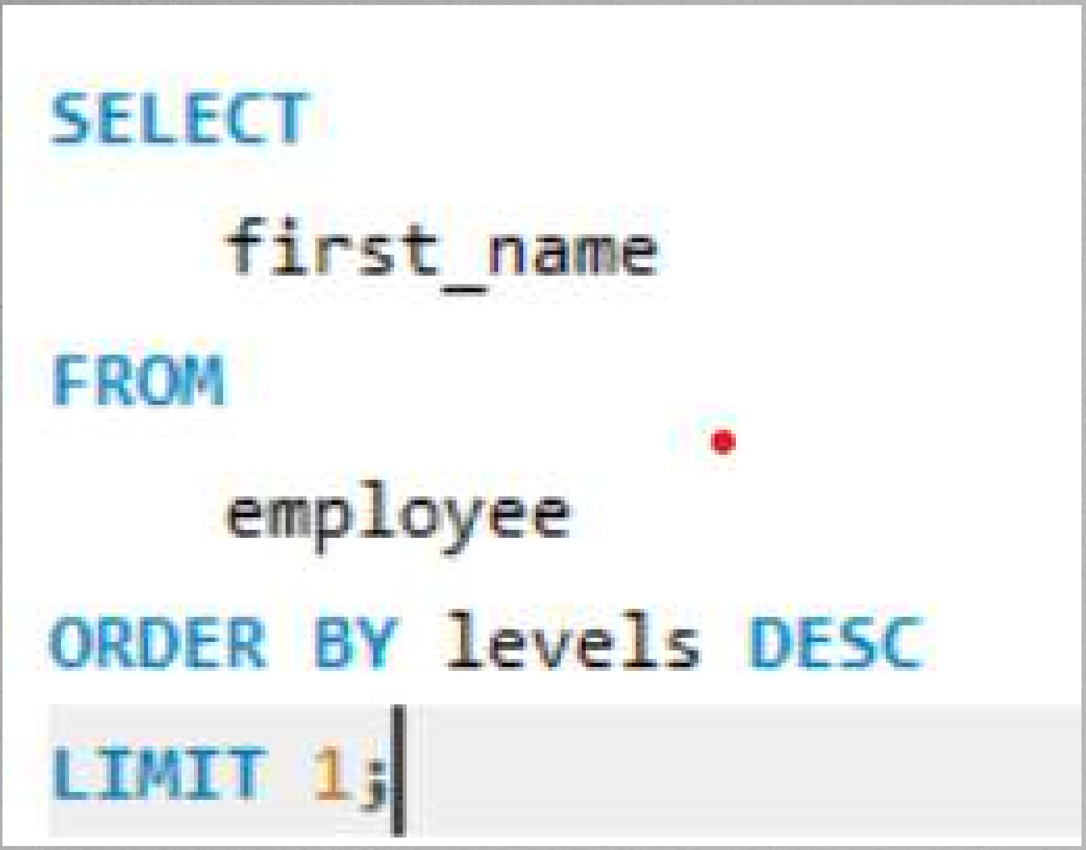
Love Yourself in the
Sound of Music



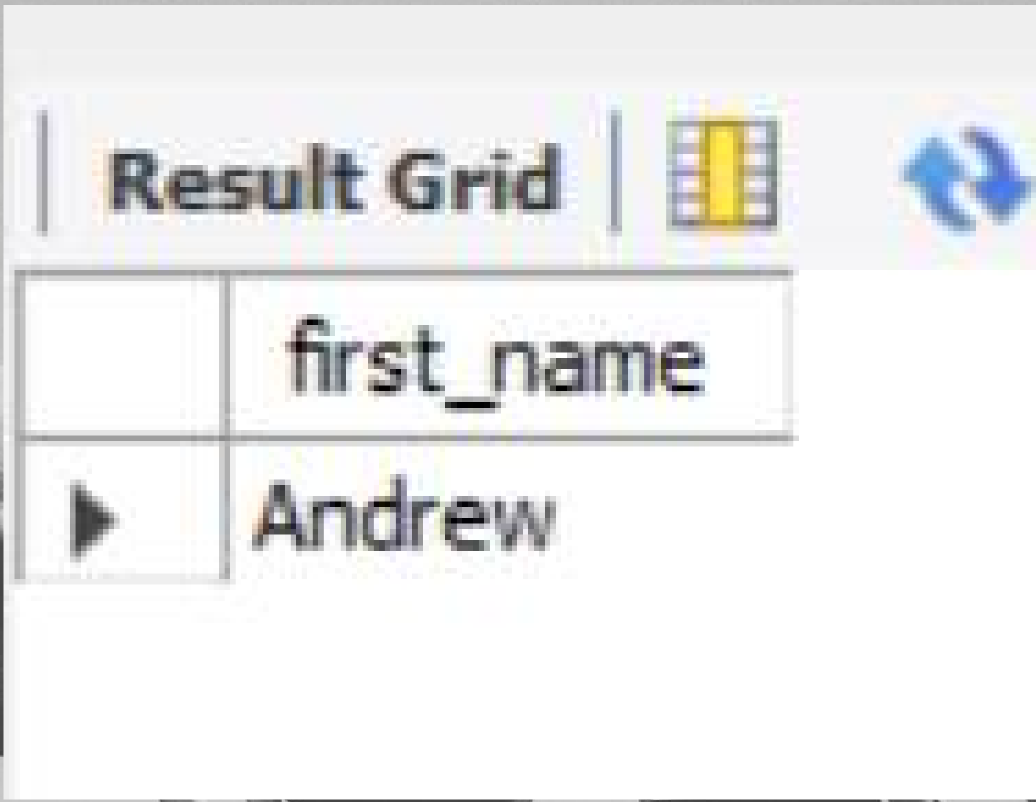
HELLO!

"IN THIS ANALYSIS, WE EXPLOIT SQL TO GAIN INSIGHT INTO THE SELLING DATA OF OUR MUSIC STORE. WE WILL TRY AND UNDERSTAND THE TRENDS OF CUSTOMERS' PURCHASE BEHAVIOR, THE TOP-SELLING MUSIC GENRES AND ARTISTS, AND SEASONAL SALES PATTERNS. ALL THESE RECOMMENDATIONS SHALL FEED INTO INVENTORY AND PROMOTION STRATEGIES FOR BETTER CUSTOMER SATISFACTION AND REVENUE GENERATION."

WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

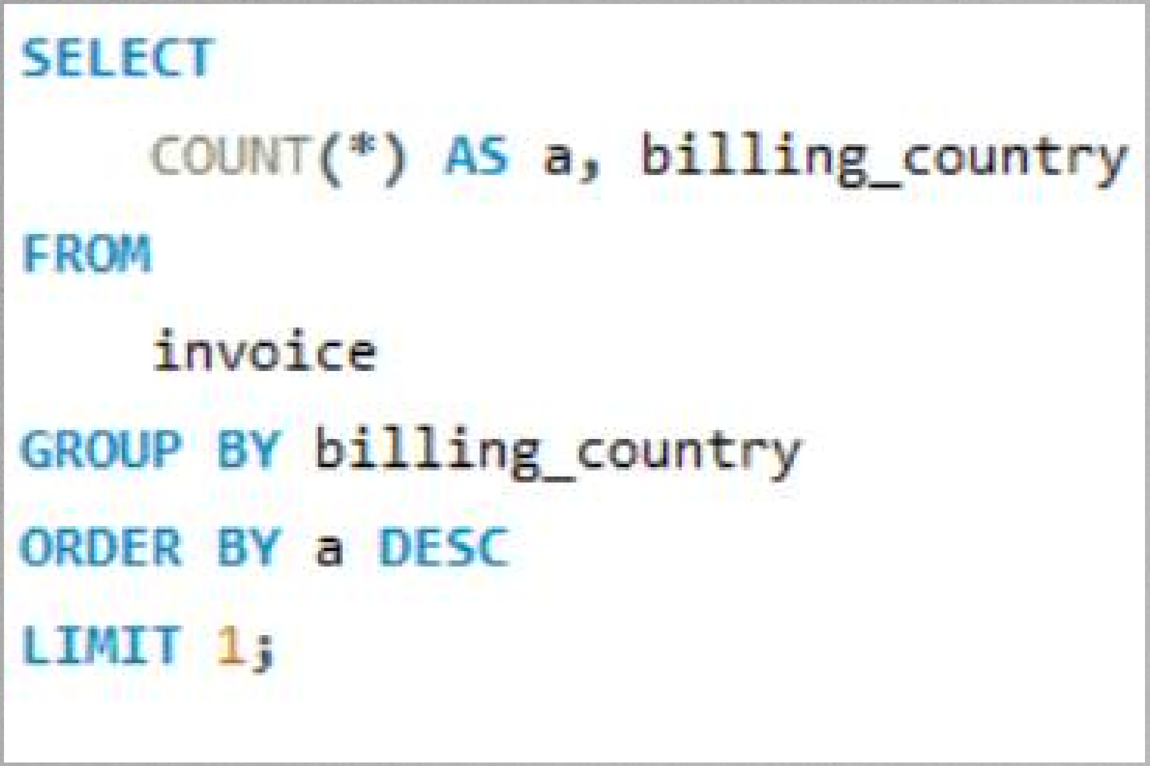


```
SELECT
    first_name
FROM
    employee
ORDER BY levels DESC
LIMIT 1;
```

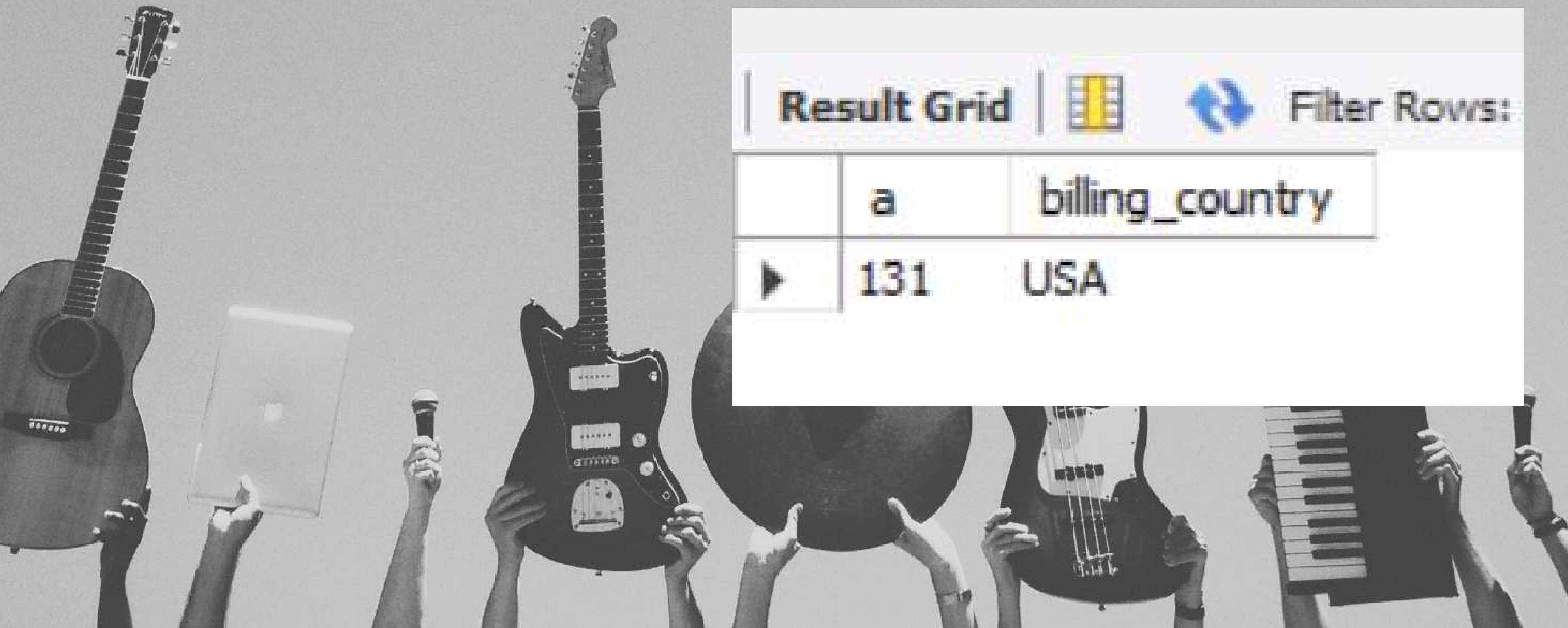


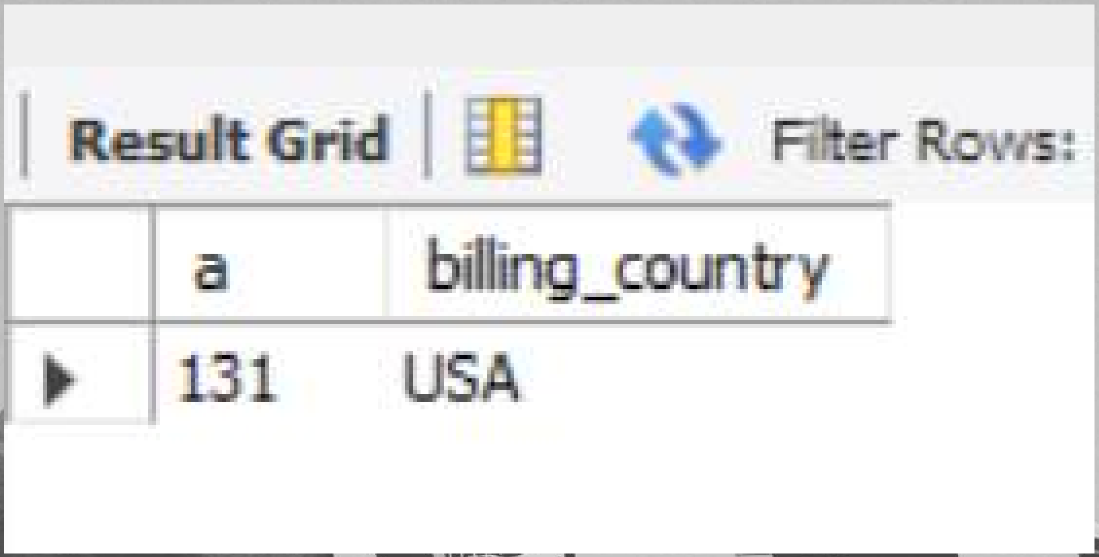
```
first_name
▶ Andrew
```

WHICH COUNTRY HAVE THE MOST INVOICES?



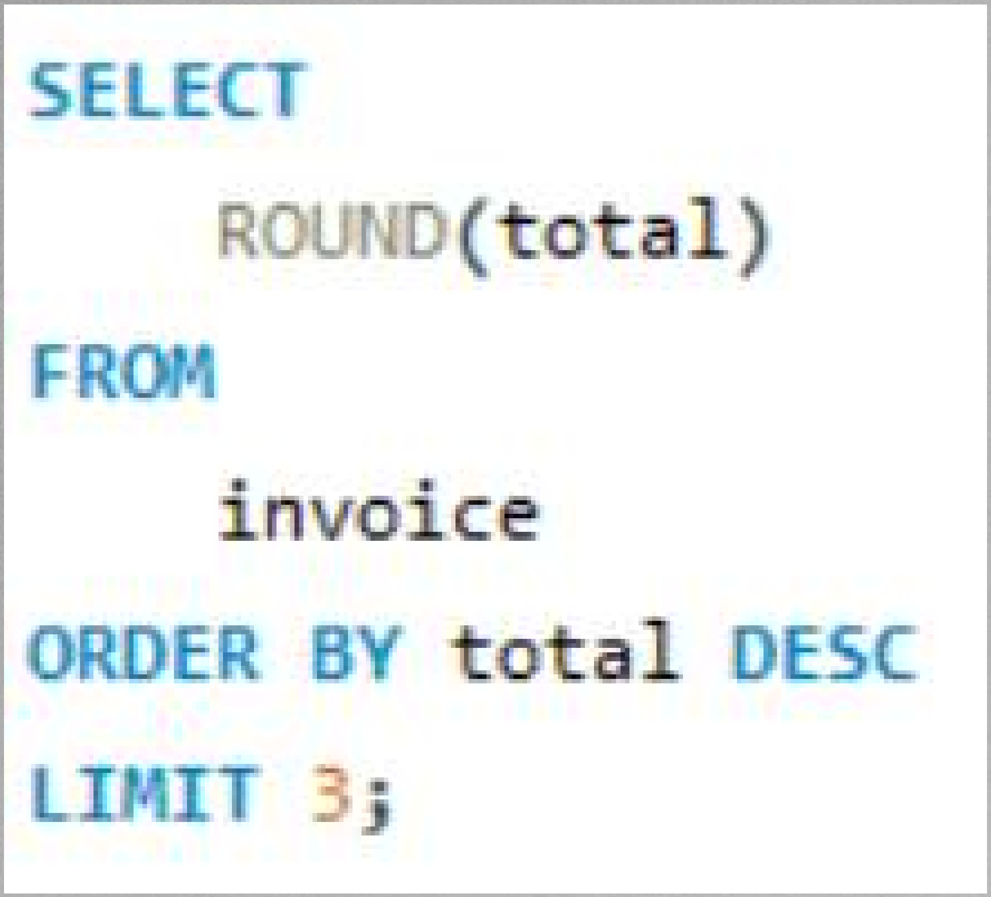
```
SELECT
    COUNT(*) AS a, billing_country
FROM
    invoice
GROUP BY billing_country
ORDER BY a DESC
LIMIT 1;
```



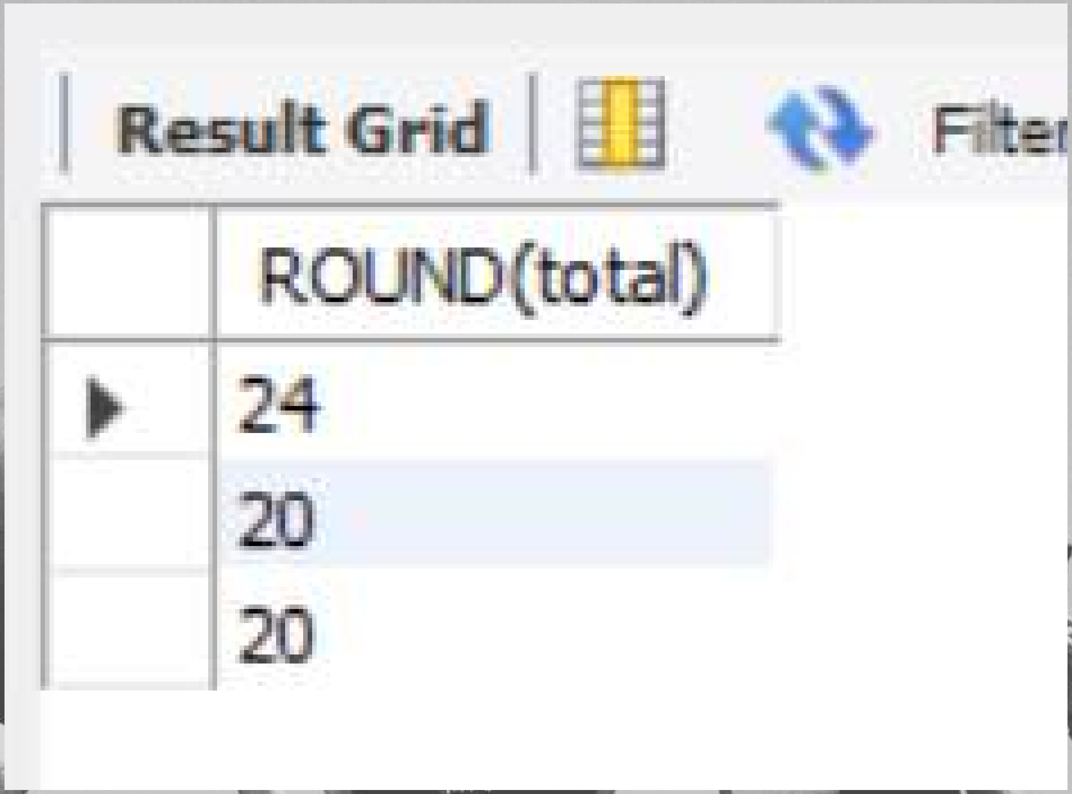


```
    a      billing_country
▶   131    USA
```

WHAT ARE TOP 3 VALUES OF TOTAL INVOICE?

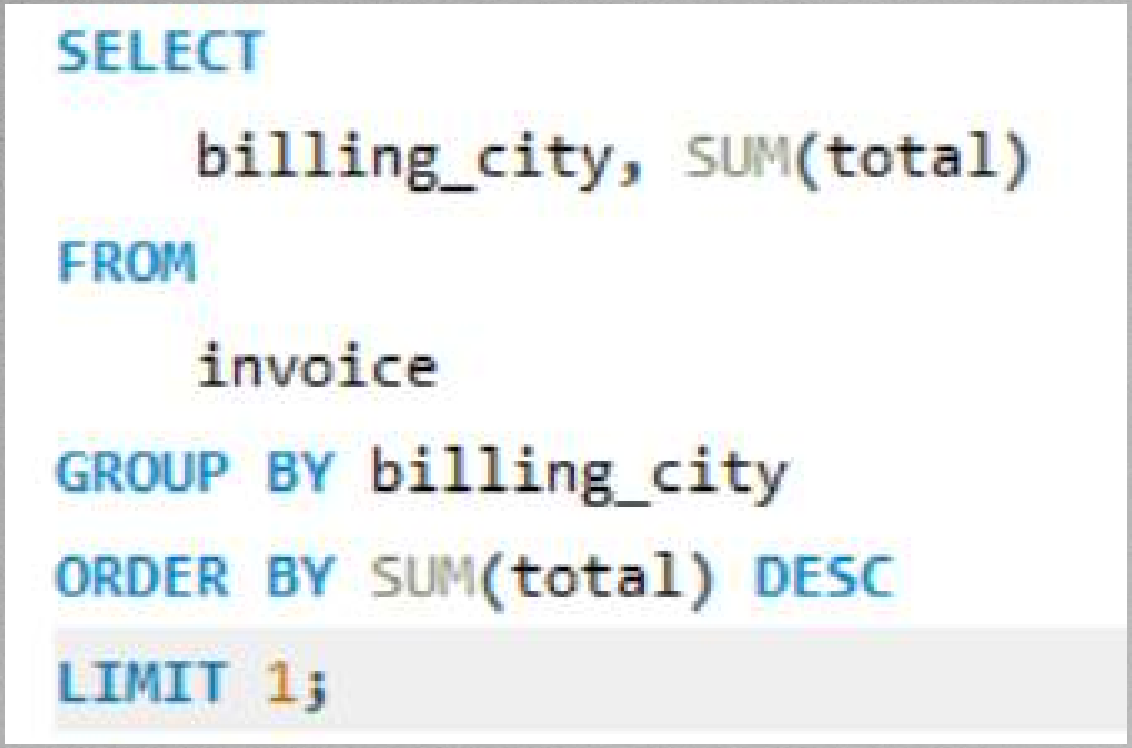


```
SELECT
    ROUND(total)
FROM
    invoice
ORDER BY total DESC
LIMIT 3;
```

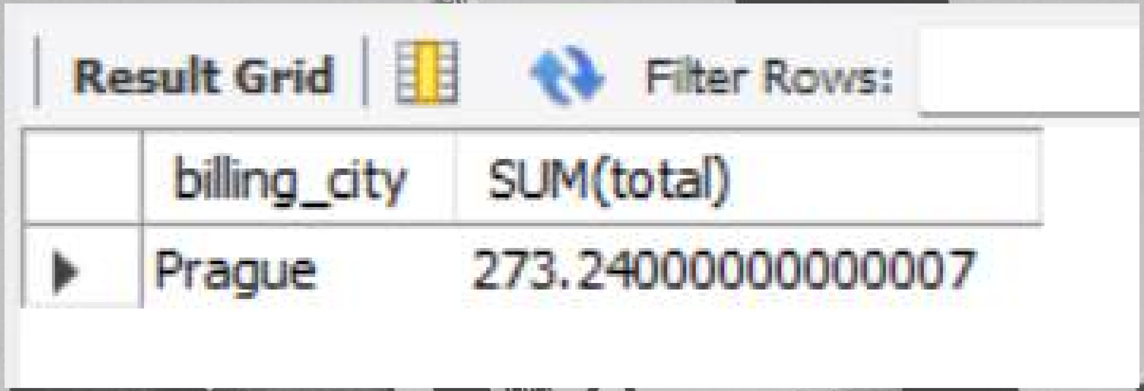


```
ROUND(total)
24
20
20
```

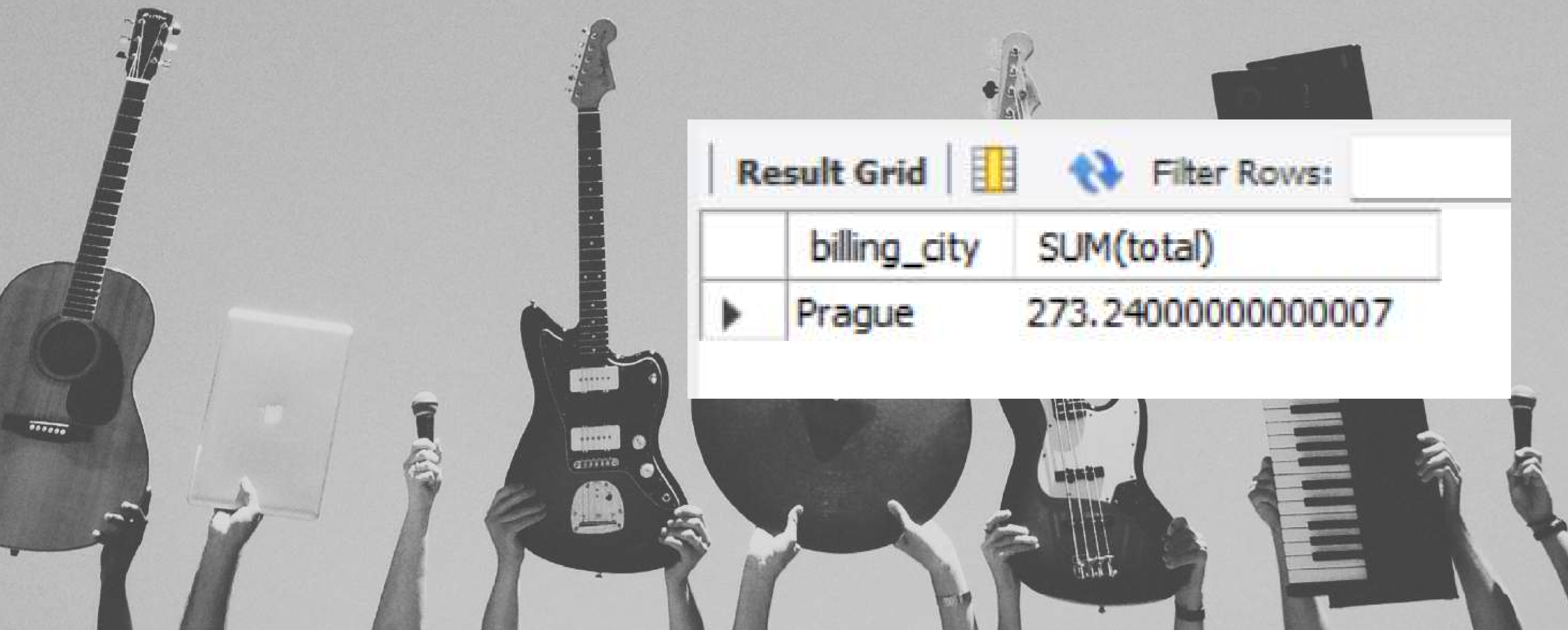
WHICH CITY HAS THE BEST CUSTOMER? WE WOULD LIKE TO THROW A PROMOTIONAL MUSIC FESTIVAL IN THE CITY WE MADE THE MOSTMONEY.WRITE A QUERY THAT RETURNS ONE CITY THAT HAS THEHIGHEST SUM OF INVOICE TOTALS. RETURN BOTH THE CITY NAMEAND SUM ALL THE INVOICES TOTALS.



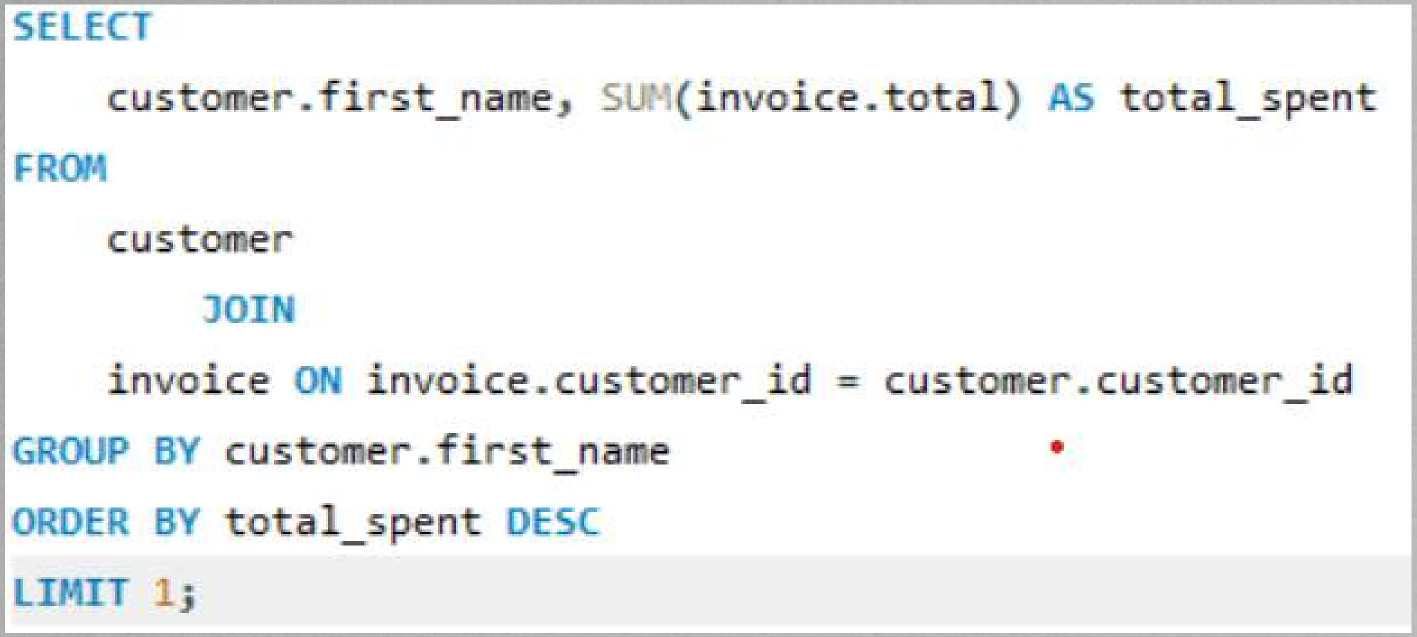
```sql
SELECT
    billing_city, SUM(total)
FROM
    invoice
GROUP BY billing_city
ORDER BY SUM(total) DESC
LIMIT 1;
```



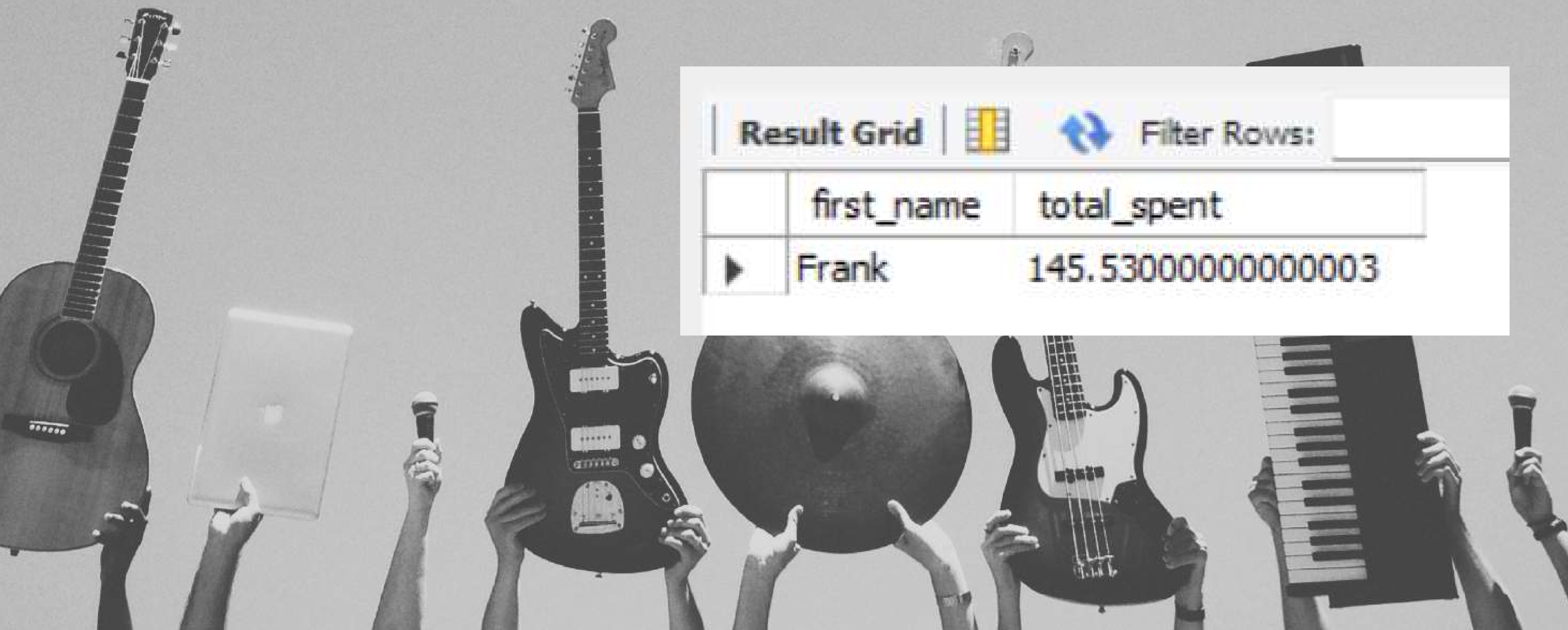
```
billing_city   SUM(total)
Prague         273.2400000000007
```

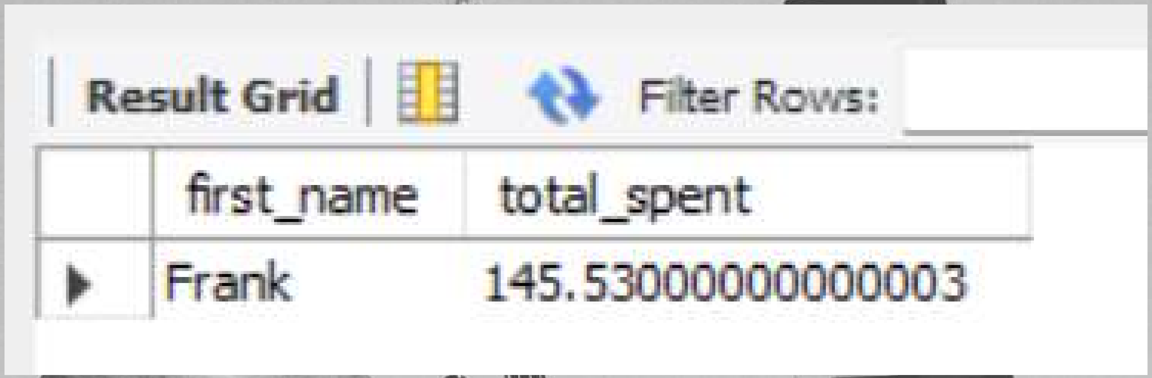


WHO IS THE BEST CUSTOMER? THE CUSTOMER WHO HAS SPENT THE MOST MONEY WILL BE DECLARED AS THE BEST CUSTOMER. WRITE A QUERY THAT RETURN THE PERSON WHO HAS SPENT THE MOST MONEY.



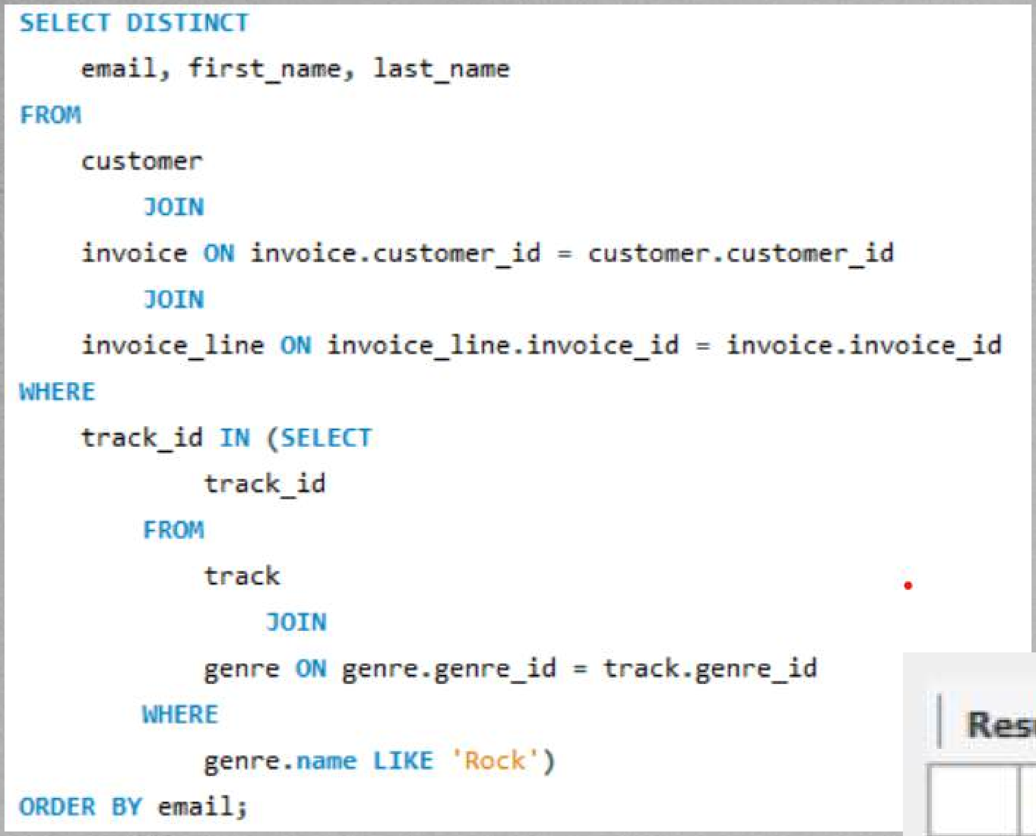
```
SELECT
    customer.first_name, SUM(invoice.total) AS total_spent
FROM
    customer
        JOIN
    invoice ON invoice.customer_id = customer.customer_id
GROUP BY customer.first_name
ORDER BY total_spent DESC
LIMIT 1;
```



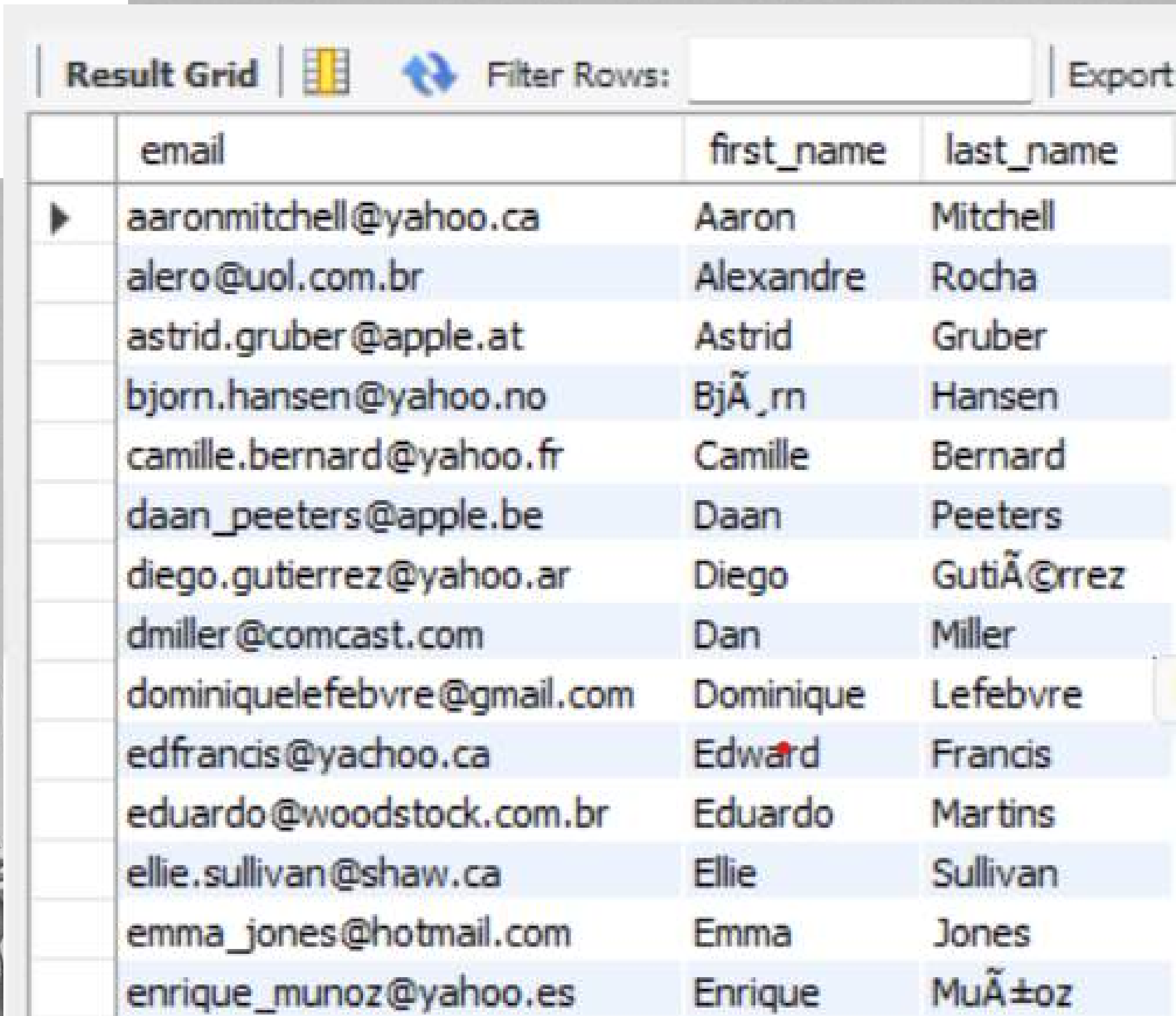


```
first_name   total_spent
Frank        145.5300000000003
```

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

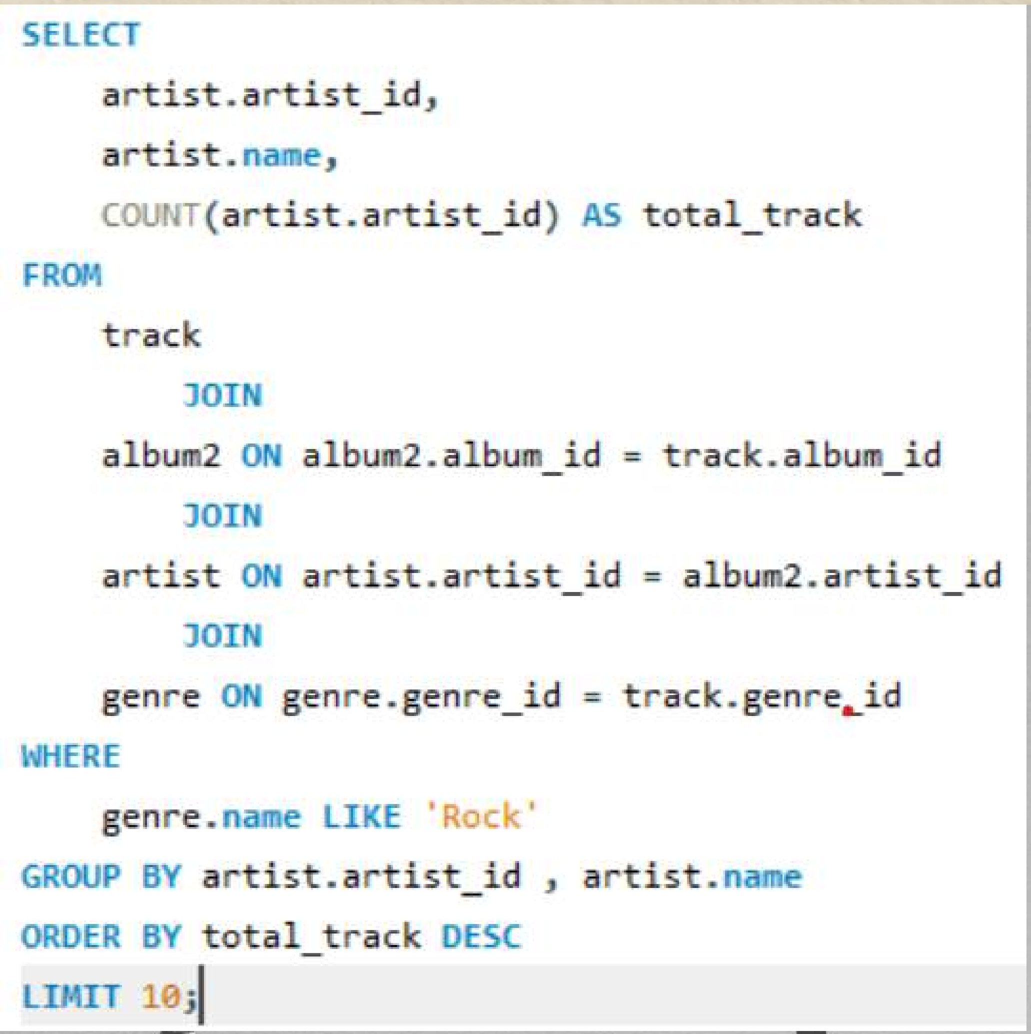


```sql
SELECT DISTINCT
    email, first_name, last_name
FROM
    customer
        JOIN
    invoice ON invoice.customer_id = customer.customer_id
        JOIN
    invoice_line ON invoice_line.invoice_id = invoice.invoice_id
WHERE
    track_id IN (SELECT
            track_id
        FROM
            track
                JOIN
            genre ON genre.genre_id = track.genre_id
        WHERE
            genre.name LIKE 'Rock')
ORDER BY email;
```

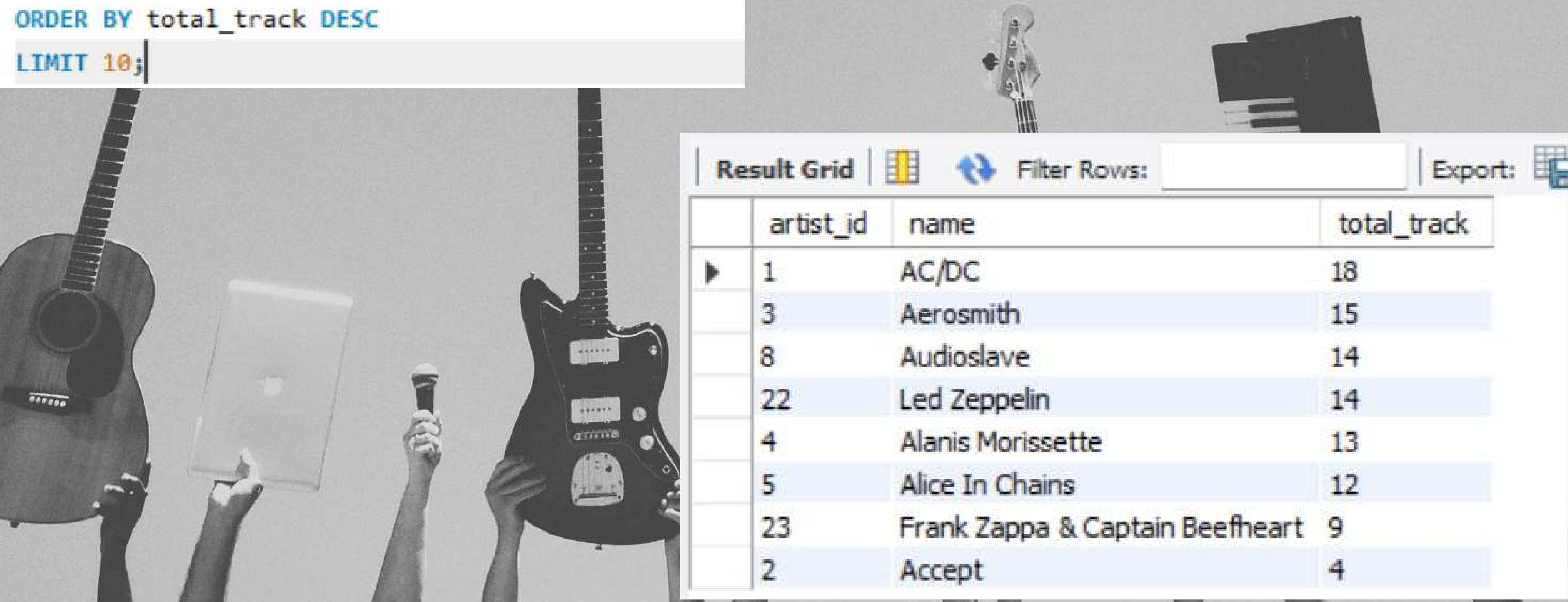


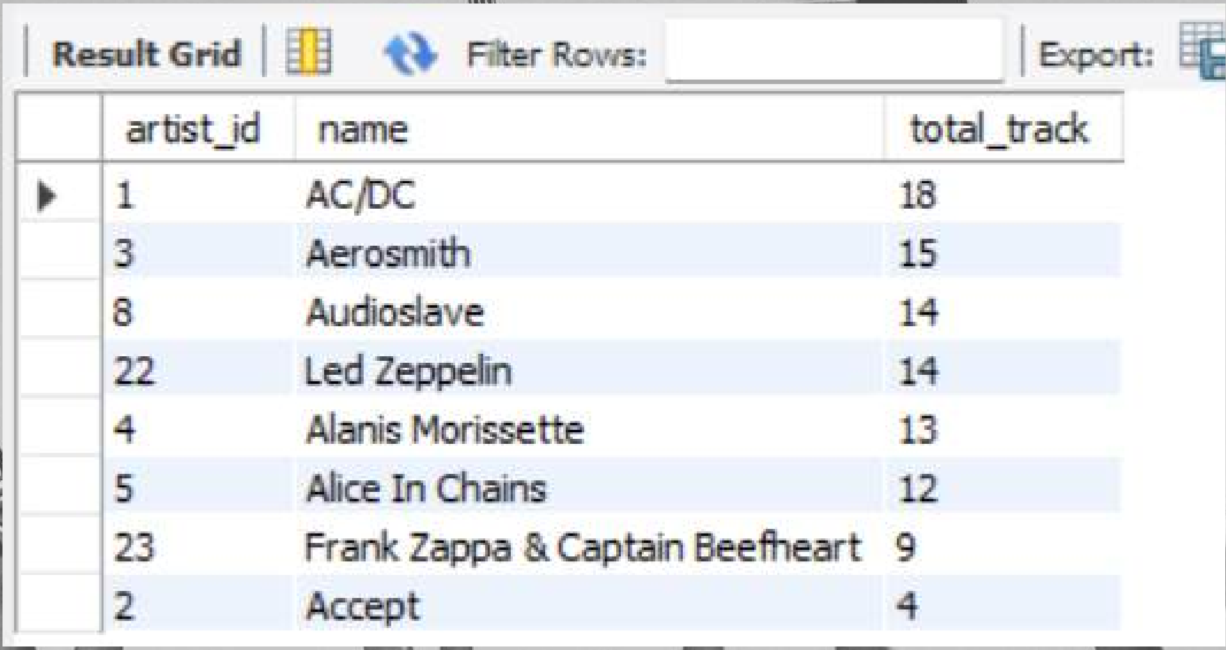
email	first_name	last_name
aaronmitchell@yahoo.ca	Aaron	Mitchell
alero@uol.com.br	Alexandre	Rocha
astrid.gruber@apple.at	Astrid	Gruber
bjorn.hansen@yahoo.no	BjÃ¸rn	Hansen
camille.bernard@yahoo.fr	Camille	Bernard
daan_peeters@apple.be	Daan	Peeters
diego.gutierrez@yahoo.ar	Diego	GutiÃ©rrez
dmiller@comcast.com	Dan	Miller
dominiquelefebvre@gmail.com	Dominique	Lefebvre
edfrancis@yachoo.ca	Edward	Francis
eduardo@woodstock.com.br	Eduardo	Martins
ellie.sullivan@shaw.ca	Ellie	Sullivan
emma_jones@hotmail.com	Emma	Jones
enrique_munoz@yahoo.es	Enrique	MuÃ±oz

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



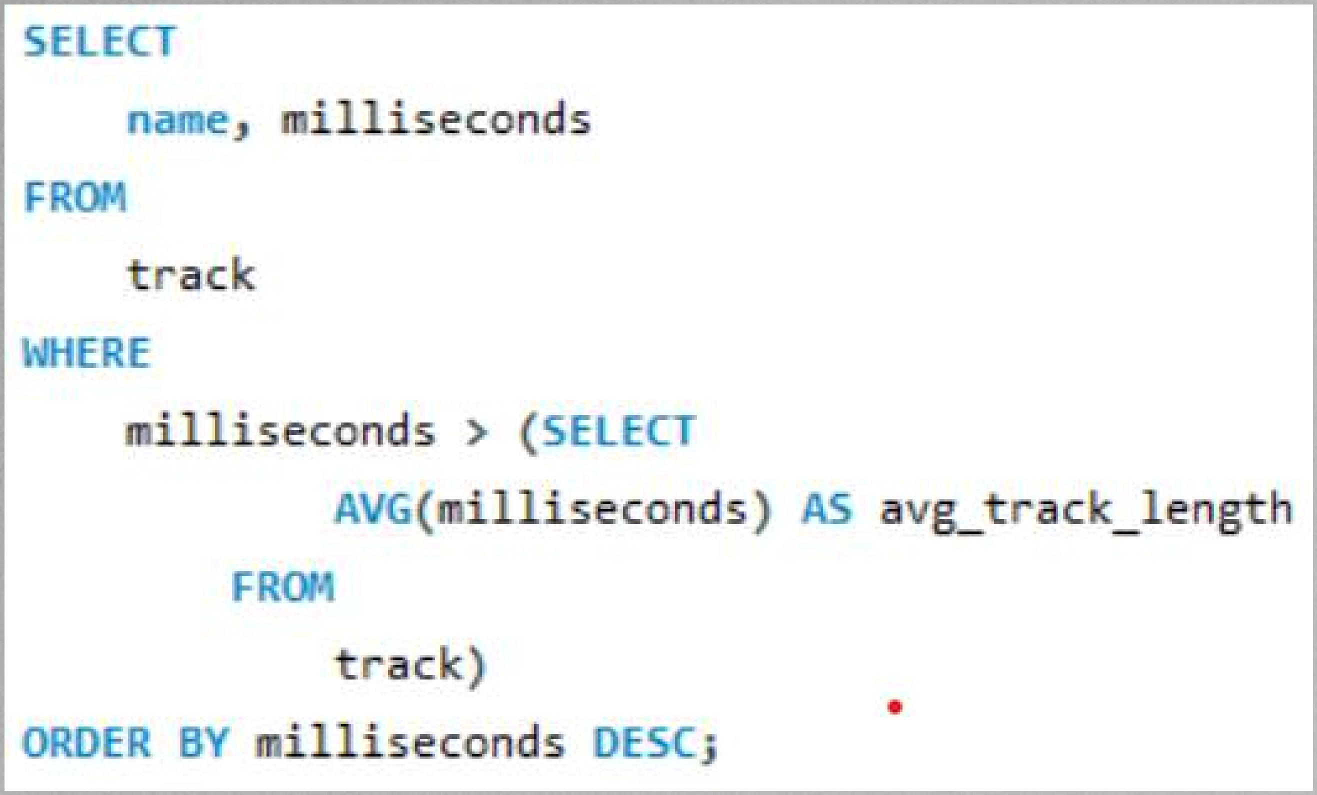
```sql
SELECT
    artist.artist_id,
    artist.name,
    COUNT(artist.artist_id) AS total_track
FROM
    track
        JOIN
    album2 ON album2.album_id = track.album_id
        JOIN
    artist ON artist.artist_id = album2.artist_id
        JOIN
    genre ON genre.genre_id = track.genre_id
WHERE
  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



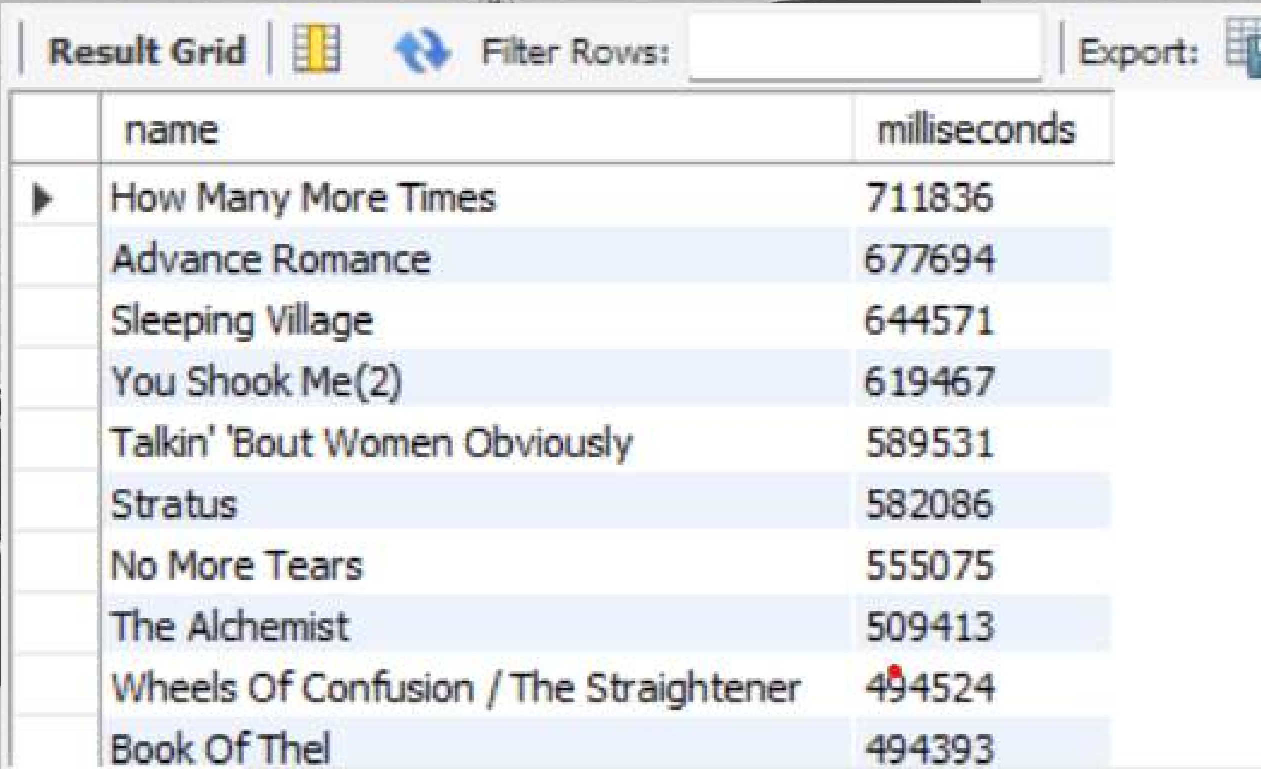


artist_id	name	total_track
1	AC/DC	18
3	Aerosmith	15
8	Audioslave	14
22	Led Zeppelin	14
4	Alanis Morissette	13
5	Alice In Chains	12
23	Frank Zappa & Captain Beefheart	9
2	Accept	4

RETURN ALL THE TRACK NAMES THAT HAVE A SONG LENGTH LONGER THAN THE AVERAGE SONG LENGTH. RETURN THE NAME AND MILLISECONDS. EACH TRACK. ORDER BY THE SONG LENGTH WITH THE LONGEST SONG LIST EVER



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



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

THANK YOU