

SQL Project Presentation

By Tasnim Ahmad Mumu

Project Objective

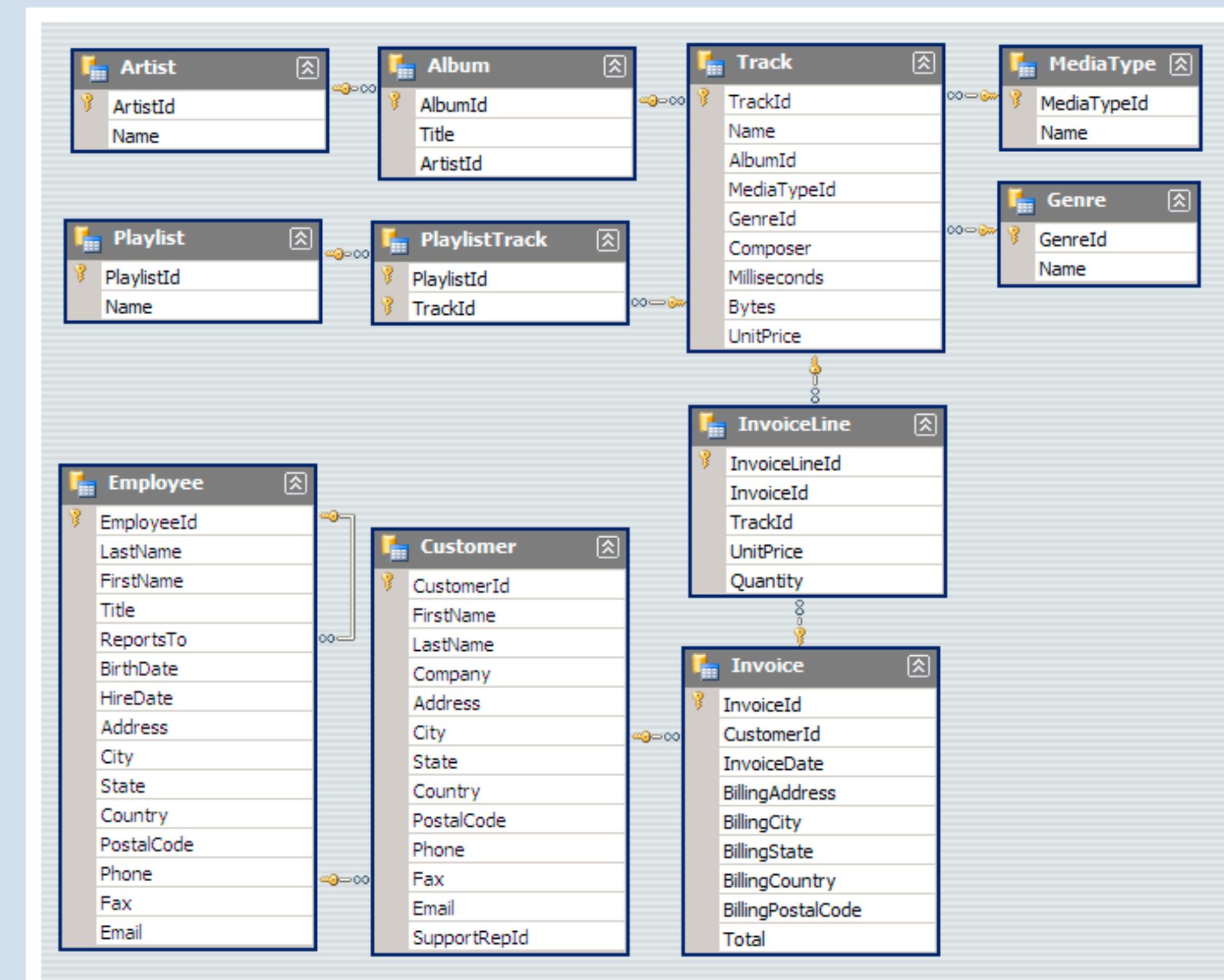
The objective of this project is to analyze a music playlist database using SQL to gain insights into the store's business growth and performance.

By examining key metrics and trends within the dataset, we aim to provide actionable insights that can help the store understand its customer preferences, optimize its offerings, and drive business growth. This beginner-level project demonstrates foundational data analysis skills, including data extraction, transformation, and visualization, while providing valuable business intelligence.

PROJECT GOALS

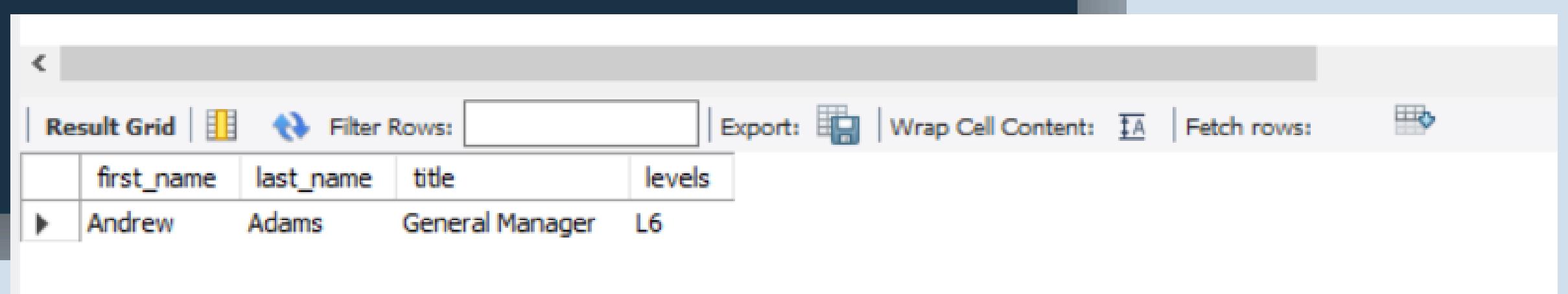
- Data Extraction and Cleaning
- Business Performance Analysis
- Customer Insights
- Trend Analysis
- Actionable Recommendations

Schema - Music Store Database



Q1: Who is the senior most employee based on job title?

```
● ● ●  
1 select  
2   first_name,  
3   last_name,  
4   title,  
5   levels  
6 from  
7   employee  
8 order by  
9   levels desc  
10 limit  
11  1
```



The screenshot shows a database query results grid. The grid has a header row with columns: first_name, last_name, title, and levels. Below the header, there is one data row: Andrew Adams, General Manager, L6. The grid includes standard navigation buttons (first, last, previous, next) and export options (Result Grid, Filter Rows, Export, Wrap Cell Content, Fetch rows).

	first_name	last_name	title	levels
▶	Andrew	Adams	General Manager	L6

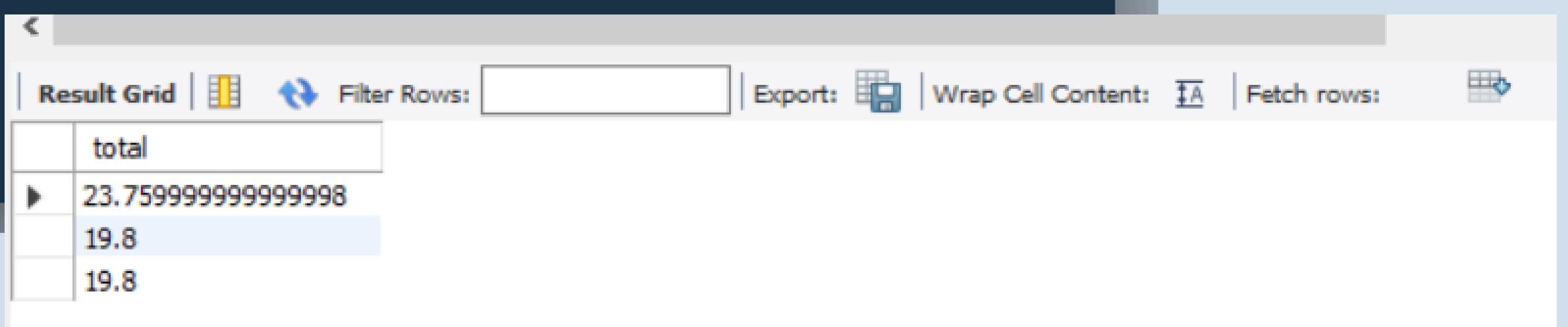
Q2: Which countries have the most Invoices?

```
1 SELECT
2     COUNT(*) AS c,
3     billing_country
4 FROM
5     invoice
6 GROUP BY
7     billing_country
8 ORDER BY
9     c DESC
```

	c	billing_country
131	USA	
76	Canada	
61	Brazil	
50	France	
41	Germany	
30	Czech Republic	
29	Portugal	
28	United Kingdom	
21	India	
13	Ireland	
13	Chile	
11	Finland	
11	Spain	
10	Poland	
10	Denmark	
10	Australia	
10	Hungary	
10	Sweden	
10	Netherlands	
9	Norway	
9	Italy	

Q3: What are top 3 values of total invoice?

```
● ● ●  
1 select  
2     total  
3 from  
4     Invoice  
5 order by  
6     total desc  
7 limit  
8     3
```



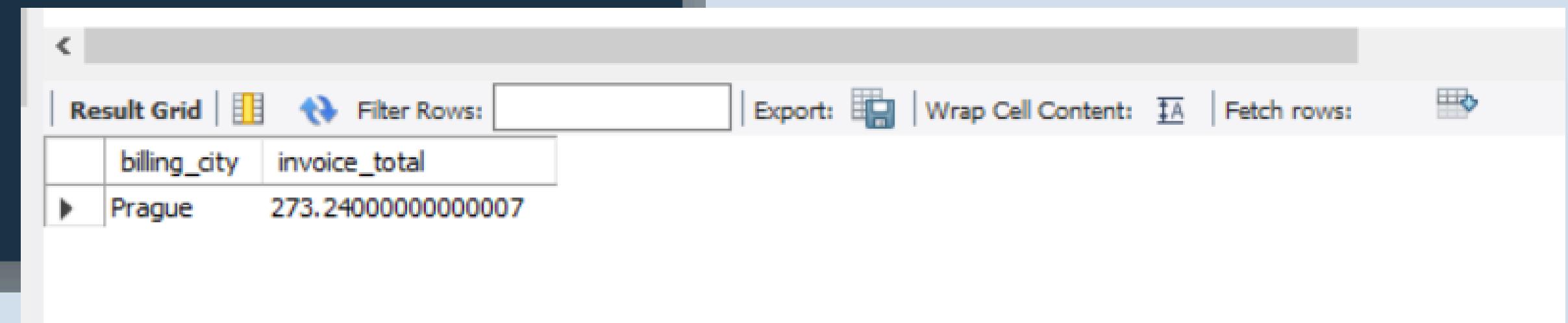
The screenshot shows a database query results grid. The grid has a single column labeled 'total'. The data rows are: 23.759999999999998, 19.8, and 19.8. The grid includes standard database navigation and export tools at the top.

	total
▶	23.759999999999998
▶	19.8
▶	19.8

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

Q4: 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 .

```
1  SELECT
2      billing_city,
3      sum(total) as invoice_total
4  FROM
5      invoice
6  group by
7      billing_city
8  order by
9      invoice_total desc
10 limit
11  1
```



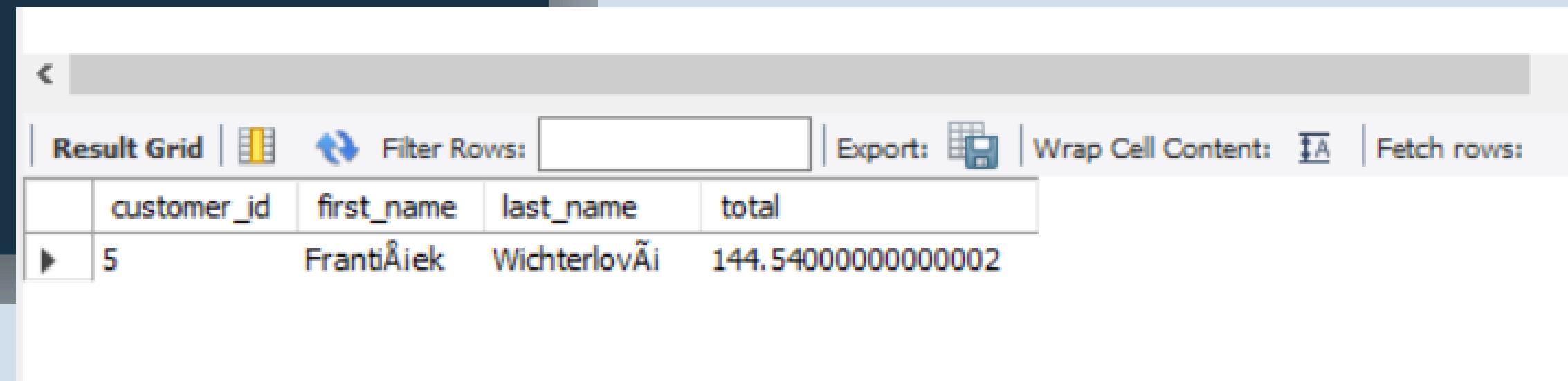
A screenshot of a database query results window. The window has a toolbar at the top with buttons for 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Contents', and 'Fetch rows'. The results grid shows a single row of data with two columns: 'billing_city' and 'invoice_total'. The data row is: Prague | 273.24000000000007.

	billing_city	invoice_total
▶	Prague	273.24000000000007

Q5: 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.

```
● ● ●

1  SELECT
2      c.customer_id,
3      c.first_name,
4      c.last_name,
5      sum(i.total) as total
6  FROM
7      customer c
8      join invoice i on c.customer_id = i.customer_id
9  group by
10     c.customer_id,
11     c.first_name,
12     c.last_name
13  order by
14      total desc
15  limit
16      1
```



	customer_id	first_name	last_name	total
▶	5	František	Wichterlová	144.54000000000002

Q6: 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.

```
1  SELECT distinct
2    email,
3    first_name,
4    last_name
5  from
6    customer
7  join invoice on customer.customer_id = invoice.customer_id
8  join invoice_line on invoice.invoice_id = invoice_line.invoice_id
9  where
10   track_id in (
11     select
12       track_id
13     from
14       track
15     join genre on track.genre_id = genre.genre_id
16     where
17       genre.name like "rock"
18   )
19  order by
20    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	Bjørn	Hansen
	camille.bernard@yahoo.fr	Camille	Bernard
	daan_peeters@apple.be	Daan	Peeters
	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
	dmiller@comcast.com	Dan	Miller
	dominiquelefebvre@gmail.com	Dominique	Lefebvre
	edfrancis@yahoo.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	Muñoz
	fernadaramos4@uol.com.br	Fernanda	Ramos
	fharris@google.com	Frank	Harris
	fralston@gmail.com	Frank	Ralston
	frantisekw@jetbrains.com	František	Wichterlová
	ftremblay@gmail.com	François	Tremblay
	fzimmermann@yahoo.de	Fynn	Zimmermann
	hannah.schneider@yahoo.de	Hannah	Schneider
	hholy@gmail.com	Helena	Holá
	hleacock@gmail.com	Heather	Leacock
	hughoreilly@apple.ie	Hugh	O'Reilly
	isabelle_mercier@apple.fr	Isabelle	Mercier

Q7: 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.

```
1 select
2     artist.artist_id,
3     artist.name,
4     count(artist.artist_id) as number_of_songs
5 from
6     artist
7     join album_1 on artist.artist_id = album_1.artist_id
8     join track on album_1.album_id = track.album_id
9     join genre on genre.genre_id = track.genre_id
10 where
11     genre.name like "rock"
12 group by
13     artist.artist_id,
14     artist.name
15 order by
16     number_of_songs desc
17 limit
18     10
```

Result Grid | Filter Rows: Export: Wrap Cell Content

	artist_id	name	number_of_songs
▶	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

Q8: 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.

```
1  SELECT
2      name,
3      milliseconds
4  FROM
5      track
6  where
7      milliseconds > (
8          select
9              avg(milliseconds) as avg_value
10         from
11             track
12     )
13  order by
14      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
	You Oughta Know (Alternate)	491885
	Terra	482429
	Snoopy's search-Red baron	456071
	Sozinho (Hitmakers Classic Mix)	436636
	Master Of Puppets	436453
	Stone Crazy	433397
	Snowblind	420022
	Computadores Fazem Arte	404323
	Jerusalem	402390

Q9: FIND HOW MUCH AMOUNT SPENT BY EACH CUSTOMER ON ARTISTS? WRITE A QUERY TO RETURN CUSTOMER NAME, ARTIST NAME AND TOTAL SPENT

```
1  with
2    best_selling_artists as (
3      select
4        artist.artist_id as artist_id,
5        artist.name as artist_name,
6        sum(invoice_line.unit_price * invoice_line.quantity) as total_sales
7      from
8        invoice_line
9        join track on track.track_id = invoice_line.track_id
10       join album_1 on album_1.album_id = track.album_id
11       join artist on artist.artist_id = album_1.artist_id
12     group by
13       1,
14       2
15     order by
16       3 desc
17     limit
18       1
19   )
20  select
21    c.customer_id,
22    c.first_name,
23    c.last_name,
24    bsa.artist_name,
25    sum(il.unit_price * il.quantity) as amount_spent
26  from
27    customer c
28    join invoice i on c.customer_id = i.customer_id
29    join invoice_line il on il.invoice_id = i.invoice_id
30    join track t on t.track_id = il.track_id
31    join album_1 a on a.album_id = t.album_id
32    join best_selling_artists bsa on bsa.artist_id = a.artist_id
33  group by
34    1,
35    2,
36    3,
37    4
38  order by
39    5 desc
```

	customer_id	first_name	last_name	artist_name	amount_spent
54	Steve	Murray	AC/DC	17.82	
53	Phil	Hughes	AC/DC	10.89	
21	Kathy	Chase	AC/DC	10.89	
49	Stanisław	Wąjcik	AC/DC	9.9	
1	Luís	Gonçalves	AC/DC	7.9200000000000001	
24	Frank	Ralston	AC/DC	7.9200000000000001	
31	Martha	Silk	AC/DC	3.96	
16	Frank	Harris	AC/DC	2.9699999999999998	
42	Wyatt	Girard	AC/DC	2.9699999999999998	
6	Helena	Holáš	AC/DC	2.9699999999999998	
38	Niklas	Schräder	AC/DC	2.9699999999999998	
35	Madalena	Sampaio	AC/DC	2.9699999999999998	
44	Terhi	Härmälä...	AC/DC	2.9699999999999998	
9	Kara	Nielsen	AC/DC	1.98	
34	João	Fernandes	AC/DC	1.98	
57	Luis	Rojas	AC/DC	1.98	
27	Patrick	Gray	AC/DC	1.98	
20	Dan	Miller	AC/DC	1.98	
30	Edward	Francis	AC/DC	1.98	
5	František	Wichterlová	AC/DC	1.98	
47	Lucas	Mancini	AC/DC	0.99	
43	Isabelle	Mercier	AC/DC	0.99	
19	Tim	Goyer	AC/DC	0.99	
39	Camille	Bernard	AC/DC	0.99	

Q10: 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 REURN ALL GENRES.

```
1  with
2      popular_genre as (
3          select
4              g.genre_id as genre_id,
5              g.name as genre_name,
6              i.billing_country as country,
7              count(il.quantity) as highest_purchase,
8              row_number() over (
9                  partition by
10                     i.billing_country
11                     order by
12                         count(il.quantity) desc
13             ) as row_no
14         from
15             genre g
16             join track t on t.genre_id = g.genre_id
17             join invoice_line il on il.track_id = t.track_id
18             join invoice i on i.invoice_id = il.invoice_id
19         group by
20             1,
21             2,
22             3
23         order by
24             4 desc
25     )
26 select
27     *
28 from
29     popular_genre
30 where
31     row_no <= 1
```

	genre_id	genre_name	country	highest_purchase	row_no
▶	1	Rock	USA	70	1
	1	Rock	Canada	57	1
	1	Rock	United Kingdom	47	1
	1	Rock	Germany	28	1
	1	Rock	Brazil	26	1
	1	Rock	France	26	1
	1	Rock	Portugal	23	1
	1	Rock	Australia	18	1
	1	Rock	Poland	14	1
	1	Rock	Czech Republic	14	1
	1	Rock	India	13	1
	1	Rock	Chile	7	1
	1	Rock	Austria	6	1
	1	Rock	Netherlands	6	1
	1	Rock	Denmark	6	1
	1	Rock	Finland	6	1
	1	Rock	Belgium	5	1
	1	Rock	Sweden	5	1
	3	Metal	Spain	4	1
	1	Rock	Hungary	4	1

Q11: 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.

```
● ● ●
1  with
2    customer_spent as (
3      select
4        c.customer_id,
5        c.first_name,
6        c.last_name,
7        i.billing_country,
8        sum(i.total) as total_spent,
9        row_number() over (
10          partition by
11            i.billing_country
12            order by
13              sum(i.total)
14        ) as row_no
15      from
16        customer c
17        join invoice i on i.customer_id = c.customer_id
18      group by
19        i.billing_country,
20        c.customer_id,
21        c.first_name,
22        c.last_name
23      order by
24        total_spent desc
25    )
26    select
27      *
28    from
29      customer_spent
30    where
31      row_no <= 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	customer_id	first_name	last_name	billing_country	total_spent	row_no
▶	6	Helena	HolÃ½	Czech Republic	128.7	1
	46	Hugh	O'Reilly	Ireland	114.8399999999997	1
	50	Enrique	MuÃ±oz	Spain	98.01	1
	57	Luis	Rojas	Chile	97.0200000000001	1
	35	Madalena	Sampaio	Portugal	82.17	1
	55	Mark	Taylor	Australia	81.18	1
	44	Terhi	HÃ¤mÃ¤nen	Finland	79.2	1
	45	Ladislav	KovÃ¡cs	Hungary	78.21	1
	49	StanisÅaw	WÃ³jciech	Poland	76.2299999999999	1
	51	Joakim	Johansson	Sweden	75.24	1
	38	Niklas	SchrÃ¶der	Germany	73.2599999999999	1
	4	BjÃ¸rn	Hansen	Norway	72.2700000000001	1
	59	Rishabh	Mishra	India	71.28	1
	7	Astrid	Gruber	Austria	69.3	1
	52	Emma	Jones	United Kingdom	68.31	1
	48	Johannes	Van der Berg	Netherlands	65.34	1
	41	Marc	Dubois	France	64.35	1
	10	Eduardo	Martins	Brazil	60.39	1
	8	Daan	Peeters	Belgium	60.3899999999999	1
	19	Tim	Goyer	USA	54.4499999999996	1
	47	Lucas	Mancini	Italy	50.49	1
	56	Diego	GutiÃ©rrez	Argentina	39.6	1
	9	Kara	Nielsen	Denmark	37.6199999999999	1

**THANK
you**