

MUSIC DATABASE PROJECT USING SQL



INTRODUCTION

Overview:

- The music database project aims to develop a comprehensive database system for managing and querying music-related data.
- The project focuses on efficiently organizing information about artists, albums, tracks, and genres.

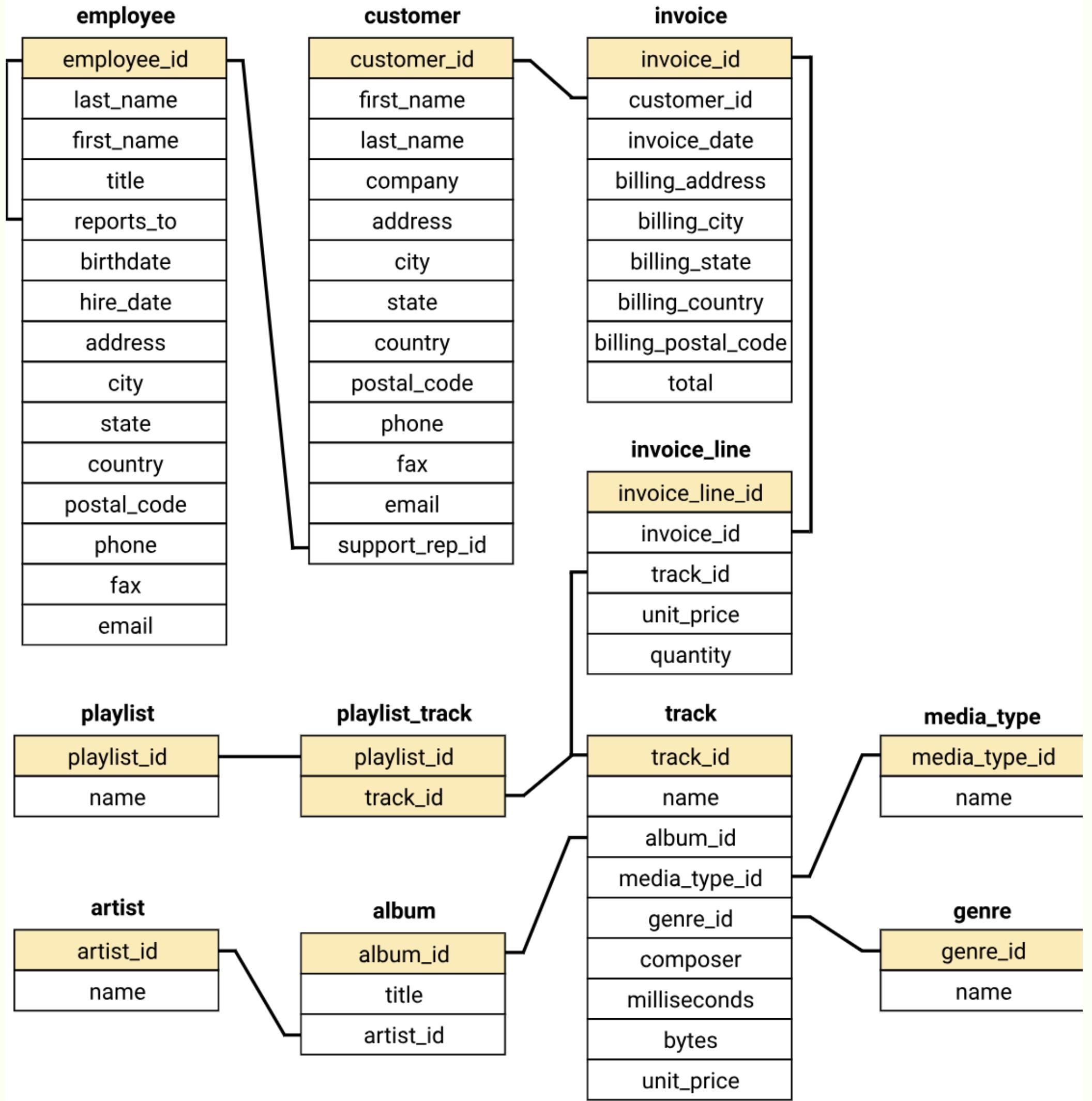
Purpose:

- To provide a robust platform for storing, retrieving, and analyzing music data.
- To enable users to perform complex queries that yield meaningful insights into the music data.

OBJECTIVES

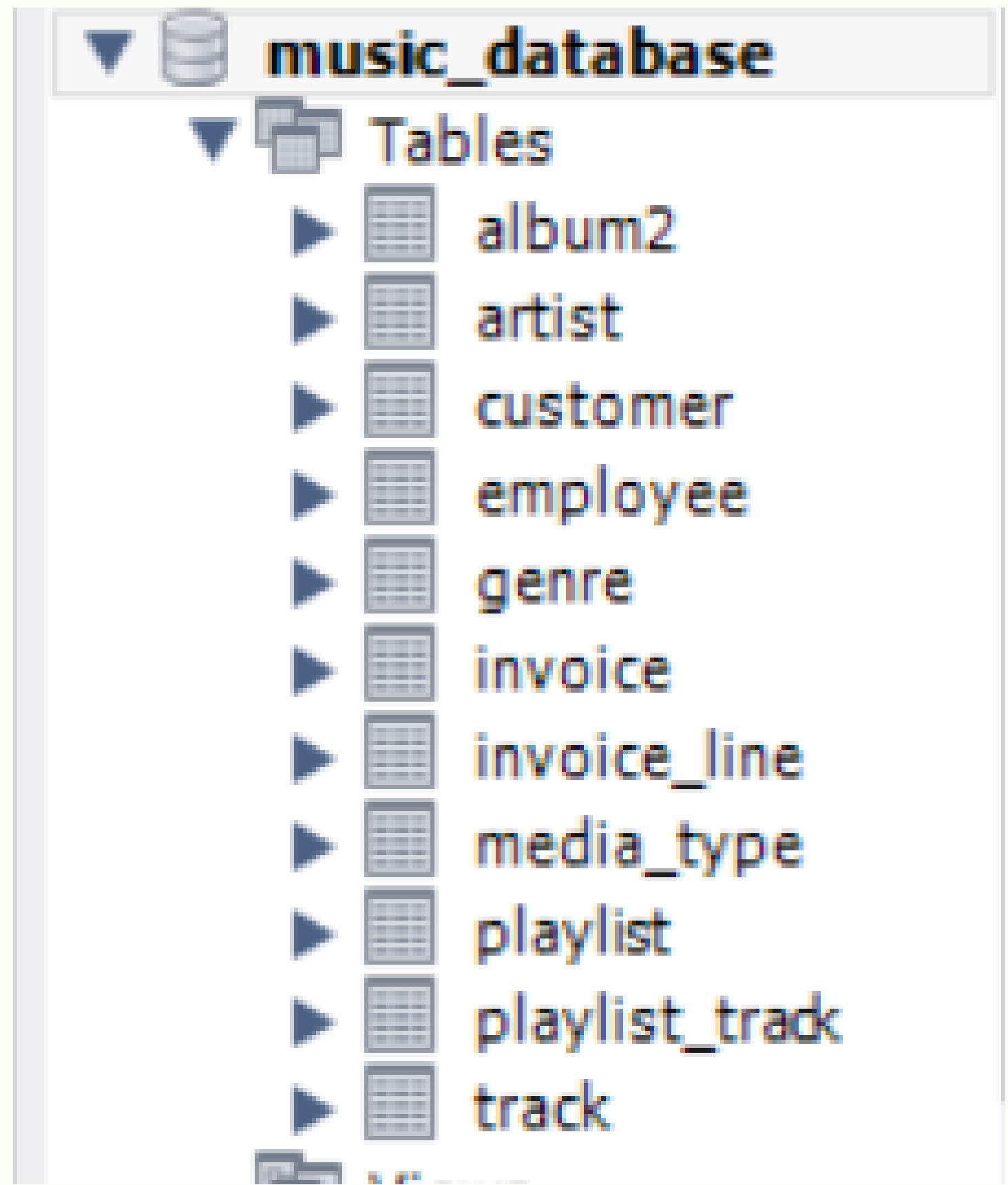
- Efficient Data Storage: Design and implement a database schema that ensures efficient storage and management of music-related data, including information about artists, albums, tracks, and genres.
- Complex Query Facilitation: Develop SQL queries that allow users to retrieve and analyze data in various ways, such as finding the most popular tracks, filtering music by genre, or aggregating data to show total tracks per album.
- Data Integrity and Scalability: Ensure the database maintains data integrity through normalization and indexing, and is scalable to handle large datasets as the volume of music data grows.

DATABASE



SCHEMA

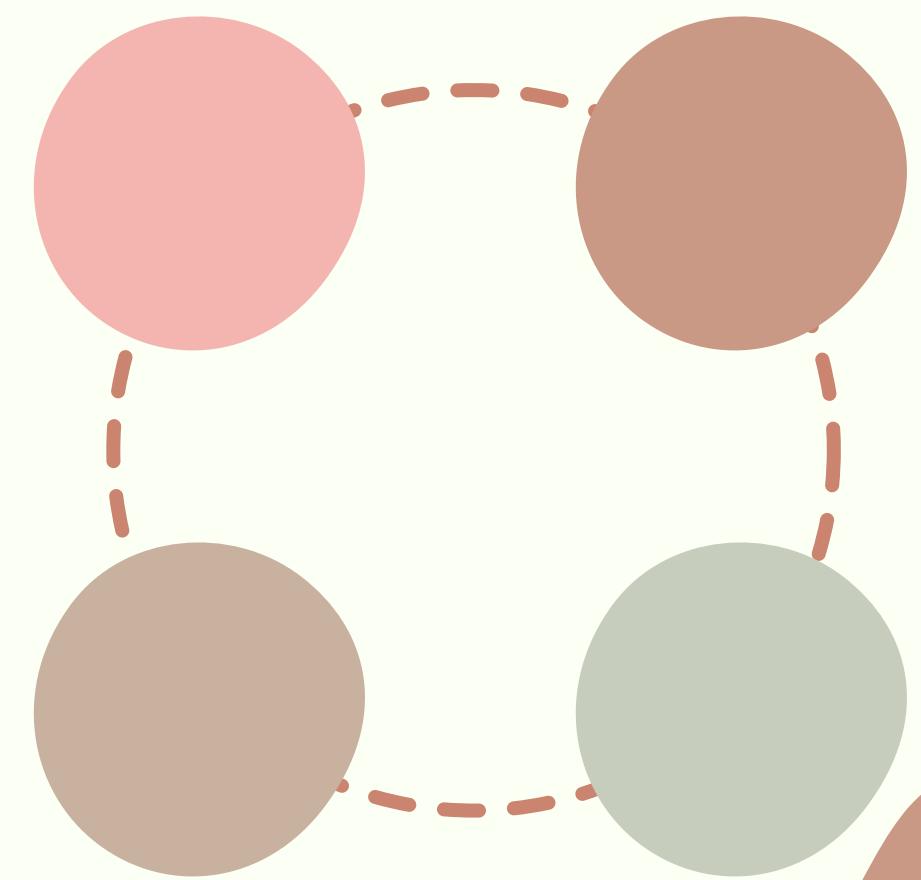
TABLES OVERVIEW



DATA IMPORT

DATABASE IMPORTED USING BELOW LINK :

[HTTPS://BIT.LY/3WYYP88](https://bit.ly/3WYYP88)



QUESTIONS

EASY LEVEL

- 1) Who is the senior most employee based on job title?
- 2) Which countries have the most invoices?
- 3) What are top 3 values of total invoices?
- 4) 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 invoice totals.
- 5) Who is the best customer? The customer who has spent the most money will be declared as the best customer. Write the query to returns the person who has spent the most money.

QUESTIONS

MODERATE LEVEL

- 1) Write query to return the email, first name, last name and genre of all rock music listeners . Return your list order alphabetically by email starting with A ?
- 2) Let's invite the artist who have written the most rock music in our dataset. Write a query that return Artist name and total track count of the top 10 rock bands.
- 3) Return all the track names that have song length longer the average song length. Return Name and Milliseconds for each track. Order by song length with the longest songs listed first.

QUESTIONS

ADVANCED LEVEL

- 1) Find how much amount spent by each customer on artists? Write a query to return customer name , artist name and total spent.
- 2) We want to find out most popular Genre music of each country. We determine most popular genre as Genre with the highest amount of purchases. Write a query to return each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genres.
- 3) 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 customers and how they spent . For countries where the top amount spent is shared , provide all customers who spent this amount.

SOLUTIONS

EASY

1

-- QUESTION 1) Who is the senior most employee based on job title?

-- SOLUTION -

```
select employee_id , last_name,first_name,levels from employee  
order by levels desc  
limit 1;
```

Result Grid | Filter Rows:

	employee_id	last_name	first_name	levels
▶	1	Adams	Andrew	L6

SOLUTIONS

EASY

2

-- QUESTION 2) Which countries have the most invoices?

-- SOLUTION -

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

Result Grid | Filter Rows:

	number_of_invoices	billing_country
▶	131	USA
	76	Canada
	61	Brazil
	50	France
	41	Germany
	30	Czech Republic
	29	Portugal
	28	United Kingdom
	21	India

SOLUTIONS

EASY

3

-- QUESTION 3) What are top 3 values of total invoices?

-- SOLUTION --

```
select total as total_invoices from invoice  
order by total_invoices desc  
limit 3;
```

Result Grid | Filter

	total_invoices
▶	23.759999999999998
	19.8
	19.8

SOLUTIONS

4

EASY

-- QUESTION 4) 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 invoice totals.

-- SOLUTION -

```
select sum(total) as invoice_totals ,billing_city  
from invoice  
group by billing_city  
order by invoice_totals desc;
```

	invoice_totals	billing_city
▶	273.24000000000007	Prague
	169.29	Mountain View
	166.32	London
	158.4	Berlin
	151.47	Paris
	129.69	São Paulo
	114.83999999999997	Dublin
	111.86999999999999	Delhi
	108.89999999999998	São José dos Campos
	108.89999999999998	Buenos Aires

5

SOLUTIONS

EASY

-- QUESTION 5) Who is the best customer?

-- The customer who has spent the most money will be declared as the best customer.

-- Write the query to returns the person who has spent the most money.

-- SOLUTION -

```
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 ;
```

The screenshot shows a MySQL query results grid. The grid has four columns: customer_id, first_name, last_name, and total. There is one row of data: customer_id 5, first_name František, last_name Wichterlová, and total 144.54000000000002. The grid includes standard database navigation buttons (first, previous, next, last) and export options (CSV, XML, JSON).

	customer_id	first_name	last_name	total
▶	5	František	Wichterlová	144.54000000000002

SOLUTIONS

MODERATE

1

```
-- QUESTION 1)Write query to return the email, first name, last name and genre of all rock music listeners  
-- Return your list order alphabetically by email starting with A ?  
-- SOLUTION -  
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 = '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@yahoo.ca	Edward	Frands
	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
	fernadaramos4@uol.com.br	Fernanda	Ramos
	fharris@google.com	Frank	Harris
	fralston@gmail.com	Frank	Ralston
	frantisekw@jetbrains.com	FrantiÅiek	WichterlovÃ¡
	ftremblay@gmail.com	FranÃ§ois	Tremblay
	fzimmermann@yahoo.de	Fynn	Zimmermann
	hannah.schneider@yahoo.de	Hannah	Schneider
	hholy@gmail.com	Helena	HolÃ½

SOLUTIONS

MODERATE

-- QUESTION 2) Let's invite the artist who have written the most rock music in our dataset.
-- Write a query that return Artist name and total track count of the top 10 rock bands.

2
--- SOLUTION ---

```
select artist.name, artist.artist_id , 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.name, artist.artist_id
order by number_of_songs desc
limit 10;
```

2

Result Grid Filter Rows:Export:

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

SOLUTIONS

MODERATE

3

```
-- QUESTION 3)Return all the track names that have song length longer the average song length.  
-- Return Name and Milliseconds for each track.  
-- Order by song lenth with the longest songs listed first.  
-- SOLUTION -
```

```
SELECT
```

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

Result Grid



Filter Rows:

E

	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
	Dazed and Confused	401920
	The Winner Loses	392254

3

SOLUTIONS ADVANCED

1

```
-- QUESTION 1) Find how much amount spent by each customer on artists?  
-- Write a query to return customer name , artist name and total spent  
-- SOLUTION -  
  
with best_selling_artist as  
(  
    Select artist.artist_id as artist_id ,  
    artist.name as artist_name ,  
    sum(invoice_line.unit_price*invoice_line.quantity) as total_sales  
    from invoice_line  
    join track on track.track_id = invoice_line.track_id  
    join album2 on album2.album_id = track.album_id  
    join artist on artist.artist_id = album2.artist_id  
    group by 1,2  
    order by 3 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 invoice i  
join customer c on c.customer_id = i.customer_id  
join invoice_line il on il.invoice_id = i.invoice_id  
join track t on t. track_id = il.track_id  
join album2 alb on alb.album_id = t.album_id  
join best_selling_artist bsa on bsa.artist_id = alb.artist_id  
group by 1,2,3,4  
order by 5 desc;
```

1

	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	StanisÅaw	WÄjciÄ	AC/DC	9.9
	1	LuÃ-s	GonÃsalves	AC/DC	7.92000000000001
	24	Frank	Ralston	AC/DC	7.92000000000001
	31	Martha	Silk	AC/DC	3.96
	16	Frank	Harris	AC/DC	2.969999999999998
	42	Wyatt	Girard	AC/DC	2.969999999999998
	6	Helena	HolÃ½	AC/DC	2.969999999999998
	38	Niklas	SchrÃ¶der	AC/DC	2.969999999999998
	35	Madalena	Sampaio	AC/DC	2.969999999999998
	44	Terhi	HÃ¤mÃ¤lÃ¤	AC/DC	2.969999999999998
	9	Kara	Nielsen	AC/DC	1.98
	34	JoÃ£o	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	FrantiÅiek	WichterlovÃ¡	AC/DC	1.98
	47	Lucas	Mancini	AC/DC	0.99
	43	Isabelle	Merlier	AC/DC	0.99
	19	Tim	Goyer	AC/DC	0.99
	26	Emilia	Pons	AC/DC	0.99

SOLUTIONS

ADVANCED

2

- QUESTION 2) 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 return each country along with the top genre .
- For countries where the maximum number of purchases is shared return all Genre.
- SOLUTION -

```
WITH popular_genre as (
    select count(invoice_line.quantity)as purchases , customer.country , genre.name, genre.genre_id,
    row_number() OVER (partition by customer.country order by count(invoice_line.quantity) desc) AS row_no
    from invoice_line
    join invoice on invoice.invoice_id = invoice_line.invoice_id
    join customer on customer.customer_id = invoice.customer_id
    join track on track.track_id = invoice_line.track_id
    join genre on genre.genre_id = track. genre_id
    group by 2,3,4
    order by 1 desc
)
select * from popular_genre WHERE row_no = 1;
```

2

Result Grid | Filter Rows: Export:

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

SOLUTIONS

ADVANCED

3

-- QUESTION 3)- 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 customers and how they spent .

-- For countries where the top amount spent is shared , provide all customers who spent this amount.

-- SOLUTION -

```
WITH customer_with_country as (
select customer.customer_id , first_name , last_name , billing_country , sum(total) as total_spending,
row_number() over(partition by billing_country order by sum(total) desc) AS Rowno
from invoice
join customer on customer.customer_id = invoice.customer_id
group by 1,2,3,4
order by 4 asc, 5 desc )
select * from customer_with_country where Rowno <= 1;
```

3

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

	customer_id	first_name	last_name	billing_country	total_spending	Rowno
▶	56	Diego	Gutiérrez	Argentina	39.6	1
	55	Mark	Taylor	Australia	81.18	1
	7	Astrid	Gruber	Austria	69.3	1
	8	Daan	Peeters	Belgium	60.38999999999999	1
	1	Luís	Gonçalves	Brazil	108.8999999999998	1
	3	François	Tremblay	Canada	99.99	1
	57	Luis	Rojas	Chile	97.0200000000001	1
	5	František	Wichterlová	Czech Republic	144.5400000000002	1
	9	Kara	Nielsen	Denmark	37.61999999999999	1
	44	Terhi	Härmäläinen	Finland	79.2	1
	42	Wyatt	Girard	France	99.99	1
	37	Fynn	Zimmermann	Germany	94.0500000000001	1
	45	Ladislav	Kovács	Hungary	78.21	1
	58	Manoj	Pareek	India	111.8699999999999	1
	46	Hugh	O'Reilly	Ireland	114.8399999999997	1
	47	Lucas	Mandini	Italy	50.49	1
	48	Johannes	Van der Berg	Netherlands	65.34	1
	4	Bjørn	Hansen	Norway	72.2700000000001	1
	49	Stanisław	Wąsik	Poland	76.2299999999999	1
	34	João	Fernandes	Portugal	102.9600000000001	1
	50	Enrique	Muñoz	Spain	98.01	1
	51	Joakim	Johansson	Sweden	75.24	1
	53	Phil	Hughes	United Kingdom	98.01	1
	17	Jack	Smith	USA	98.01	1

CONTACT SHWETA DHIMAN



9821218938



LinkedIn - www.linkedin.com/in/shweta-dhiman-4869b6126



dhimanshweta0303@gmail.com



**THANK
YOU**