

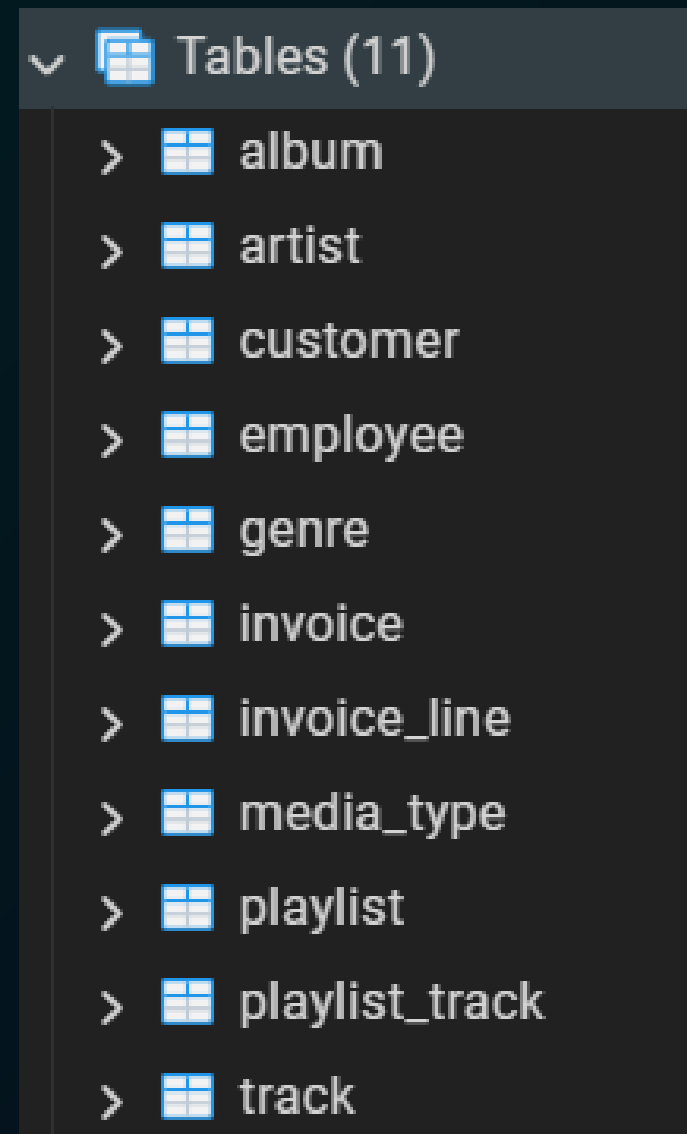
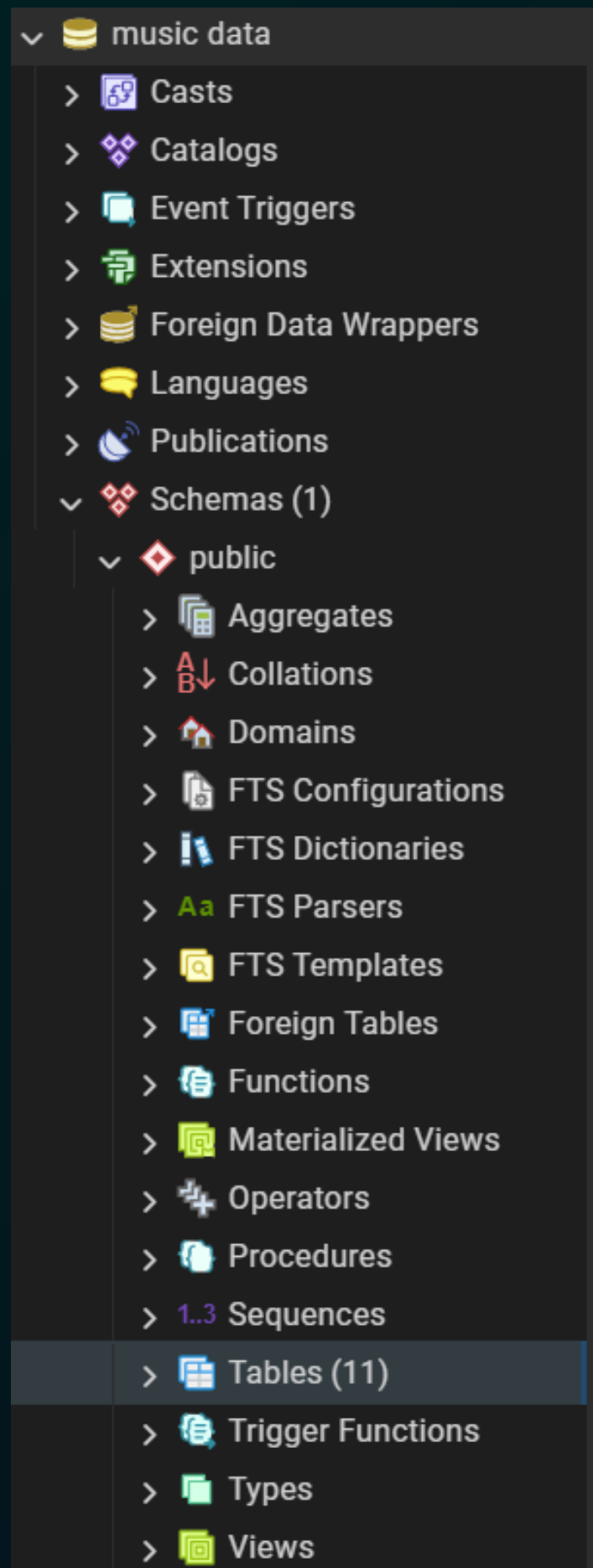


# MUSIC STORE ANALYSIS



# Intro:

The database ("music data") contains different tables related to Artists, their Albums, Genres, Listeners, Invoice etc..



We are required to perform analysis to answer the questions asked in order to manage the database and perform data-driven decision making.



```
-- who is the senior most employee based on job title
```

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

Data Output Messages Notifications							
	employee_id [PK] character varying (50)	last_name character (50)	first_name character (50)	title character varying (50)	reports_to character varying (30)	levels character varying (10)	birthdate timestamp without time zone
1	9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 00:00:00

```
--Which country have the most invoices
```

```
select billing_country, count(billing_country) as c  
from invoice  
group by billing_country  
order by c desc  
limit 1
```

Data Output Messages Notifications		
	billing_country character varying (30)	c bigint
1	USA	131

```
--what are top 3 values of total invoice
```

```
select customer_id, total
from invoice
order by total desc
limit 3
```

Data Output

Messages

Notifications

SQL

	<div>customer_id</div> <div>integer</div> <div></div>	<div>total</div> <div>double precision</div> <div></div>	
1	42	23.759999999999998	
2	32	19.8	
3	3	19.8	

```
-- Which city with highest sum of invoice
```

```
select billing_city, sum(total) as s
from invoice
group by billing_city
order by s desc
limit 1
```

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	<b>billing_city</b> character varying (30) <div>🔒</div>	<b>s</b> double precision <div>🔒</div>
1	Prague	273.240000000000007

-- which customer spent most money

```
select i.customer_id, c.first_name, c.last_name, sum(i.total) as s
from invoice as i
join customer as c on i.customer_id=c.customer_id
group by i.customer_id, c.first_name, c.last_name
order by s desc
limit 1
```

Data Output Messages Notifications				
<div><div><div>≡+</div><div>📄</div><div>▼</div><div>📋</div><div>▼</div><div>🗑</div><div>🗄</div><div>⬇</div><div>📈</div><div>SQL</div></div></div>				
	customer_id integer	first_name character (50)	last_name character (50)	s double precision
1	5	R	Madhav	144.540000000000002

--email, first , last name, genre of all rock music listeners, order by email alphabetically

```
select --g.genre_id, g.name, t.track_id, il.invoice_id, i.customer_id,
distinct c.email, c.first_name, c.last_name
from track as t
join genre as g on t.genre_id=g.genre_id
join invoice_line as il on t.track_id=il.track_id
join invoice as i on il.invoice_id=i.invoice_id
join customer as c on i.customer_id=c.customer_id
where g.name like 'Rock'
order by email
```

Data Output Messages Notifications				
<div><div><div>≡+</div><div>📄</div><div>▼</div><div>📋</div><div>▼</div><div>🗑</div><div>🗄</div><div>⬇</div><div>📈</div><div>SQL</div></div></div>				
	email character varying (50)	first_name character (50)	last_name character (50)	
1	aaronmitchell@yahoo.ca	Aaron	Mitchell	...
2	alero@uol.com.br	Alexandre	Rocha	...
3	astrid.gruber@apple.at	Astrid	Gruber	...
4	bjorn.hansen@yahoo.no	Bjørn	Hansen	...
5	camille.bernard@yahoo.fr	Camille	Bernard	...
6	daan_peeters@apple.be	Daan	Peeters	...
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	...
Total rows: 59 of 59    Query complete 00:00:00.794    Ln 55, Col 1				

--artist\_name and total track count of the top 10 rock bands

```
select al.artist_id, ar.name, count(track_id) as c
from track as t
join genre as g on t.genre_id=g.genre_id
join album as al on t.album_id=al.album_id
join artist as ar on al.artist_id=ar.artist_id
where g.name like 'Rock'
group by al.artist_id, ar.name
order by c desc
limit 10
```

Data Output				Messages	Notifications
	artist_id	name	c		
	character varying (30)	character varying (120)	bigint		
1	22	Led Zeppelin	114		
2	150	U2	112		
3	58	Deep Purple	92		
4	90	Iron Maiden	81		
5	118	Pearl Jam	54		
6	152	Van Halen	52		
7	51	Queen	45		
Total rows: 10 of 10				Query complete 00:00:00.415 Ln 68, Col 1	

--track names and length in milliseconds for the songs with length longer than avg length, order by song length desc

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

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑

🗄

⬇

📈

SQL

	<div>name</div> <div>character varying (150)</div>	<div>milliseconds</div> <div>integer</div>
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894

Total rows: 494 of 494

Query complete 00:00:00.432

Ln 76, Col 1

```
--amount spent by each customer on artist, return customer name , artist name and total spent

WITH most_popular_genre AS
(
    SELECT COUNT(il.quantity) AS purchases, c.country, g.name, g.genre_id,
    ROW_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) as RowNo
    FROM invoice_line as il
    JOIN invoice as i ON i.invoice_id = il.invoice_id
    JOIN customer as c ON c.customer_id = i.customer_id
    JOIN track as t ON t.track_id = il.track_id
    JOIN genre as g ON g.genre_id = t.genre_id
    GROUP BY 2,3,4
    ORDER BY 2 ASC, 1 DESC
)
SELECT * FROM most_popular_genre WHERE RowNo <= 1
```

Data OutputMessagesNotifications

SQL

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
Total rows: 24 of 24    Query complete 00:00:01.471    Ln 80, Col 1					

--genre with highest amount of purchase, each country along with top genre, if max purchase equal for any country , return all

```
select i.billing_country,g.name, sum(il.quantity) as amount_of_purchase
from invoice_line as il
join track as t on il.track_id = t.track_id
join invoice as i on il.invoice_id=i.invoice_id
join genre as g on t.genre_id=g.genre_id
group by 1,2
order by 1
```

Data Output Messages Notifications

SQL

	billing_country character varying (30)	name character varying (120)	amount_of_purchase double precision
1	Argentina	Blues	2
2	Argentina	Easy Listening	1
3	Argentina	Alternative	1
4	Argentina	Latin	2
5	Argentina	Metal	2
6	Argentina	Rock	11
7	Argentina	Reggae	1



--customer who spent most on music for each country

```
With rank as (  
With spent as  
(select i.billing_country, c.first_name, sum(il.unit_price*il.quantity) as total_spent  
from invoice as i  
join customer as c on i.customer_id=c.customer_id  
join invoice_line as il on il.invoice_id=i.invoice_id  
group by i.billing_country, c.first_name  
)  
select spent.billing_country, spent.first_name, total_spent,  
rank() over (partition by billing_country order by total_spent desc) as ranks  
from spent  
)  
select rank.billing_country, rank.first_name, total_spent, ranks  
from rank  
where ranks=1
```

Data OutputMessagesNotifications

	billing_country character varying (30)	first_name character (50)	total_spent double precision	ranks bigint
1	Argentina	Diego	39.599999999999994	1
2	Australia	Mark	81.179999999999995	1
3	Austria	Astrid	69.300000000000001	1
4	Belgium	Daan	60.3900000000000036	1
5	Brazil	Luís	108.89999999999998	1
6	Canada	François	99.989999999999985	1
7	Chile	Luis	97.019999999999987	1
Total rows: 24 of 24    Query complete 00:00:01.240    Ln 121, Col 10				