



ROCK



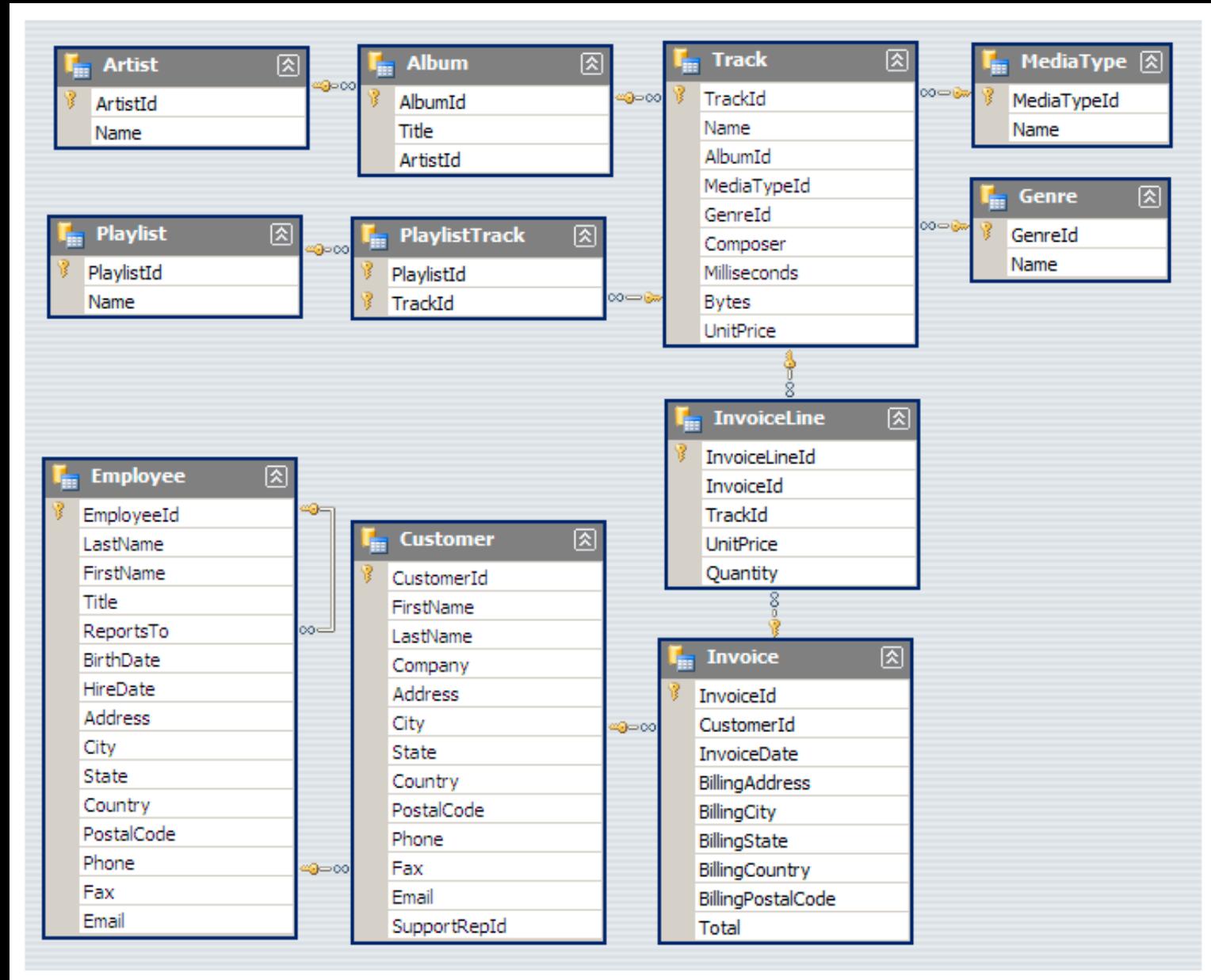
MUSIC STORE DATA ANALYSIS

SQL PROJECT (CASE STUDY)



BY:- AMIT PAWAR

SCHEMA DIGRAM



Question Set 1 – Easy

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

The screenshot shows a database query interface with the following details:

Query History: The current query is:

```
13  
14 select * from employee  
15 order by levels desc  
16 limit 1  
17  
18  
19
```

Output: The results of the query are displayed in a table:

	employee_id	last_name	first_name	title	reports_to	levels
1	9	Madan	Mohan	Senior General Manager	[null]	L7

A large black arrow points from the word "OUTPUT" to the table.

Question Set 1 – Easy

2) Which countries have the most invoices?

The screenshot shows a database query interface with the following details:

Toolbar: Includes icons for file operations, search, filters, and various database management functions.

Query History: Shows the history of queries run, with the current query labeled #2.

Query Editor: Displays the SQL code for the current query:18 #2)
19 select count(billing_country), billing_country from invoice
20 group by billing_country
21 order by billing_country desc
22 limit 5

Data Output: A large black arrow points from the word "OUTPUT" in the editor to this tab. It contains a table showing the results of the query:

	count bigint	billing_country character varying (30)
1	131	USA
2	28	United Kingdom
3	10	Sweden
4	11	Spain
5	29	Portugal

Question Set 1 – Easy

3)What are top 3 values of total invoice?

The screenshot shows a database query interface with the following details:

- Toolbar:** Includes icons for file operations, search, filters, and various database management functions.
- Query History:** Shows "Query History" tab.
- Query Editor:** Displays the following SQL code:

```
23
24  #3)
25  select total from invoice
26  order by total desc
27  limit 3
```
- Output Section:** Labeled "OUTPUT" with a large black arrow pointing down to the results table.
- Data Output:** Shows tabs for "Data Output", "Messages", and "Notifications".
- Result Table:** A table showing the top 3 invoice totals.

	total
1	23.759999999999998
2	19.8
3	19.8

Question Set 1 – Easy

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 & sum of all invoice totals

The screenshot shows a PostgreSQL database interface with the following details:

- Tables (11):** A sidebar listing tables: album, artist, customer, employee, genre, invoice, invoice_line, media_type, playlist, playlist_track, track, Trigger Function, Types, Views, Subscriptions, postgres, Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrapper.
- Query History:** A tab labeled "Query History" is active.
- Query:** The query is numbered "#4)" and is as follows:

```
28
29
30
31
32
33
34
#4)
select sum(total) as total_invoice, billing_city from invoice
group by billing_city
order by total_invoice desc
```
- OUTPUT:** A large table titled "OUTPUT" displays the results of the query. A black arrow points from the question text above to this table.
- Data Output:** A tab labeled "Data Output" is active.
- Messages:** A tab labeled "Messages" is present.
- Notifications:** A tab labeled "Notifications" is present.
- Table Headers:** The table has two columns: "total_invoice" (double precision) and "billing_city" (character varying (30)).
- Table Data:** The table contains 9 rows of data:

	total_invoice	billing_city
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris
6	129.69	São Paulo
7	114.8399999999997	Dublin
8	111.8699999999999	Delhi
9	108.8999999999998	São José dos Campos
- Total rows:** 53 of 53
- Query complete:** 00:00:00.057
- Ln 34, Col 1:** Status bar at the bottom right.

Question Set 1 – Easy

1. 5) 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

The screenshot shows a MySQL Workbench interface. On the left, there's a sidebar with tabs for Sequences, Tables (11), Trigger Function, Types, and Views. The Tables tab is selected. The main area has a toolbar at the top with various icons for file operations, search, and database management. Below the toolbar, the Query tab is active, showing a numbered query:

```
33  
34 #5)  
35 select customer.first_name,customer.last_name, sum(invoice.total) as total from customer  
36 join invoice on customer.customer_id = invoice.customer_id  
37 group by customer.customer_id  
38 order by total desc  
39 limit 1
```

Below the query, there are tabs for Data Output, Messages, and Notifications. A large black arrow points from the word "OUTPUT" to the Data Output tab. Under Data Output, there are icons for creating new tables, opening files, and other database operations. The results are displayed in a table:

	first_name	last_name	total
1	R	... Madhav	144.54000000000002

Question Set 2 – Moderate

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

The screenshot shows a database interface with a sidebar containing schema navigation and a main area for writing and executing queries.

Query History:

```
#1) LONG QUESTION
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;
```

Data Output:

OUTPUT (indicated by a large black arrow pointing down)

	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	aleroluo@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

Question Set 2 – Moderate

2) 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

The screenshot shows a PostgreSQL database interface with the following details:

- Sidebar:** A tree view of database objects including: album, artist, customer, employee, genre, invoice, invoice_line, media_type, playlist, playlist_track, track, Trigger Function, Types, Views, Subscriptions, postgres, Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrapper, and Languages.
- Query Tab:** The current tab is "Query". The query itself is:

```
12
13 #2) LONG Question
14 select artist.artist_id, artist.name, count(artist.artist_id) as number_of_song from artist
15 join album on artist.artist_id = album.artist_id
16 join track on album.album_id = track.album_id
17 join genre on track.genre_id = genre.genre_id
18 where genre.name like 'Rock'
19 group by artist.artist_id
20 order by number_of_song desc
21 limit 10
```
- Output Tab:** The tab is labeled "OUTPUT" and contains:
 - Data Output (selected)
 - Messages
 - NotificationsA large black arrow points from the "Data Output" tab to the table below.
- Data Output Table:** A table showing the results of the query:

	artist_id [PK] character varying (50)	name character varying (120)	number_of_song 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

Question Set 2 – Moderate

1. 3) 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

The screenshot shows a PostgreSQL database interface with the following details:

- Sidebar:** A tree view of database objects including employee, genre, invoice, invoice_line, media_type, playlist, playlist_track, and track.
- Query Editor:** A code editor window containing the following SQL query:

```
24
25 #3) Long Question
26 select track.track_id, track.name, track(milliseconds) from track
27 where milliseconds > (
28     select avg(milliseconds) as average_song_length from track
29 )
30 order by milliseconds desc
31
32
33
```
- Output Area:** A large table titled "OUTPUT" showing the results of the query. The table has three columns: track_id, name, and milliseconds. The data is as follows:

	track_id	name	milliseconds
5	3227	Battlestar Galactica, Pt. 2	2956081
6	3226	Battlestar Galactica, Pt. 1	2952702
7	3243	Murder On the Rising Star	2935894
8	3228	Battlestar Galactica, Pt. 3	2927802
9	3248	Take the Celestra	2927677
10	3239	Fire In Space	2926593

Question Set 3 – Advance

1) Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

The screenshot shows a PostgreSQL database interface with a sidebar containing a tree view of schema objects and a main area for running queries.

Query History:

```
#1) Very Long Question
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 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 1
    order by 3 desc
    limit 1
)
select customer.customer_id, customer.first_name, customer.last_name, best_selling_artist.artist_name,
sum(invoice_line.unit_price * invoice_line.quantity) as amount_spent
from invoice
join customer on invoice.customer_id = customer.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 best_selling_artist on album.artist_id = best_selling_artist.artist_id
group by 1,2,3,4
order by 5 desc;
```

Data Output:

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.71999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.83000000000002
5	53	Phil	Hughes	Queen	11.88

OUTPUT ↗

Question Set 3 – Advance

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 returns each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genres

The screenshot shows a database interface with a sidebar containing a navigation tree and a main panel for writing and executing SQL queries.

Navigation Tree:

- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (11)
 - album
 - artist
 - customer
 - employee
 - genre
 - invoice
 - invoice_line
 - media_type
 - playlist
 - playlist_track
 - track
- Trigger Functions
- Types
- Views
- Subscriptions
- postgres

Main Panel:

Query History: #2) Very Long Question

```
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 RowNo
    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 2 ASC, 1 DESC
)
```

Data Output: OUTPUT

	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
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1

Your device needs to restart to install updates.
Select a time to restart.

Question Set 3 – Advance

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 customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount

Query History

```
18 #3)Very Long Question
19 WITH Customer_with_country AS (
20     SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,
21         ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
22     FROM invoice
23     JOIN customer ON customer.customer_id = invoice.customer_id
24     GROUP BY 1,2,3,4
25     ORDER BY 4 ASC,5 DESC)
26     SELECT * FROM Customer_with_country WHERE RowNo <= 1
27
```

Data Output Messages Notifications

OUTPUT ↘

	customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spending double precision	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999999999	1
5	1	Luís	Gonçalves	Brazil	108.8999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000001	1
8	5	R	Madhav	Czech Republic	144.54000000000002	1
9	9	Kara	Nielsen	Denmark	37.61999999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	27	Eugenio	Zelený	Croatia	94.95000000000001	1

THANK
YOU

NOT
ONE
I
-D
YI
S
OR
DXQ