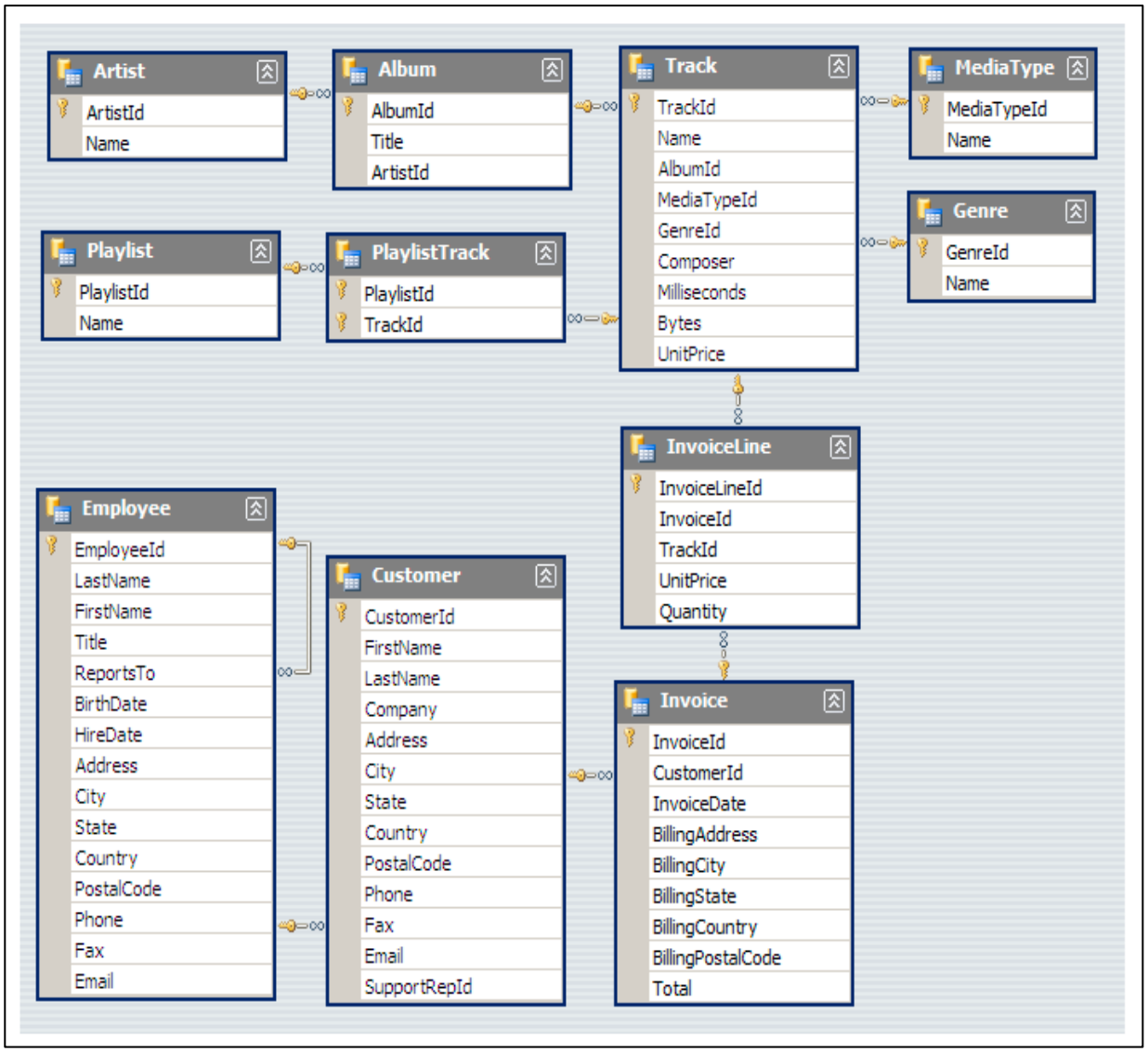
# End-to-End SQL Project: Music Store Analysis

by Mahendralal Prajapati

# Schema:


Artist
ArtistId
Name
Album
AlbumId
Title
ArtistId
Track
TrackId
Name
AlbumId
MediaTypeId
GenreId
Composer
Milliseconds
Bytes
UnitPrice
MediaType
MediaTypeId
Name
Genre
GenreId
Name
Playlist
PlaylistId
Name
PlaylistTrack
PlaylistId
TrackId
InvoiceLine
InvoiceLineId
InvoiceId
TrackId
UnitPrice
Quantity
Employee
EmployeeId
LastName
FirstName
Title
ReportsTo
BirthDate
HireDate
Address
City
State
Country
PostalCode
Phone
Fax
Email
Customer
CustomerId
FirstName
LastName
Company
Address
City
State
Country
PostalCode
Phone
Fax
Email
SupportRepId
Invoice
InvoiceId
CustomerId
InvoiceDate
BillingAddress
BillingCity
BillingState
BillingCountry
BillingPostalCode
Total

# Beginner

**Q.1) Who is the senior employee based on job title?**

```
SELECT *
FROM EMPLOYEE
ORDER BY LEVELS DESC
LIMIT 1;
```

**Output:**

Data Output | Messages | Notifications

| | employee_id [PK] character varying (50) | last_name character | first_name character | title character varying (50) | reports_to character varying (30) | levels character varying (10) |
|---|---|---|---|---|---|---|
| 1 | 9 | Madan | Mohan | Senior General Manager | [null] | L7 |

**Q.2) Which countries have the most Invoices?**

```
SELECT COUNT(*) AS C,
    BILLING_COUNTRY
FROM INVOICE
GROUP BY BILLING_COUNTRY
ORDER BY C DESC;
```

**Output:**

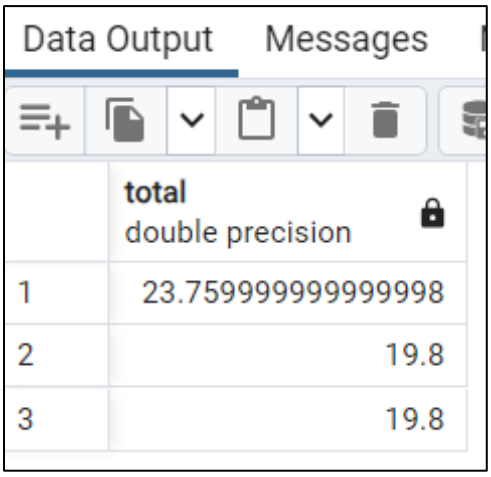
Data Output | Messages | Notifications

| | c bigint | billing_country character varying (30) |
|---|---|---|
| 1 | 131 | USA |
| 2 | 76 | Canada |
| 3 | 61 | Brazil |
| 4 | 50 | France |
| 5 | 41 | Germany |
| 6 | 30 | Czech Republic |

**Q.3) What are top 3 values of total invoice?**

```
SELECT TOTAL
FROM INVOICE
ORDER BY TOTAL DESC
LIMIT 3;
```

**Output:**




| | total<br>double precision |
|---|---|
| 1 | 23.759999999999998 |
| 2 | 19.8 |
| 3 | 19.8 |

**Q.4) Which city has the best customers? We would like to throw a promotional Music festival in the city we made that most money, write a query that returns one city that has highest sum of invoice totals. Return both the city name and sum of all invoices.**

```
SELECT SUM(TOTAL) AS INVOICE_TOTAL,
    BILLING_CITY
FROM INVOICE
GROUP BY BILLING_CITY
ORDER BY INVOICE_TOTAL DESC;
```

**Output:**

Data Output | Messages | Notifications

| | invoice_total<br>double precision | billing_city<br>character varying (30) |
|---|---|---|
| 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 |

**Q.5) Who is the one of 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.**

```
SELECT CUSTOMER.CUSTOMER_ID,
    CUSTOMER.FIRST_NAME,
    CUSTOMER.LAST_NAME,
    SUM(INVOICE.TOTAL) AS TOTAL_SPENT
FROM CUSTOMER
JOIN INVOICE ON CUSTOMER.CUSTOMER_ID = INVOICE.CUSTOMER_ID
GROUP BY CUSTOMER.CUSTOMER_ID
ORDER BY TOTAL_SPENT DESC
LIMIT 1;
```

**Output:**

Data Output | Messages | Notifications

| | customer_id [PK] integer | first_name character | last_name character | total_spent double precision |
|---|---|---|---|---|
| 1 | 5 | R ... | Madhav | 144.54000000000002 |

# Intermediate

**Q.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.**

```
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 like 'Rock' )
ORDER BY EMAIL;
```

**Output:**

Data Output | Messages | Notifications

| | email character varying (50) | first_name character | last_name character |
|---|---|---|---|
| 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 |

**Q.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.**

```
SELECT ARTIST.ARTIST_ID,
    ARTIST.NAME,
    COUNT(ARTIST.ARTIST_ID) AS NUMBER_OF_SONGS
FROM TRACK
JOIN ALBUM ON ALBUM.ALBUM_ID = TRACK.ALBUM_ID
JOIN ARTIST ON ARTIST.ARTIST_ID = ALBUM.ARTIST_ID
JOIN GENRE ON GENRE.GENRE_ID = TRACK.GENRE_ID
WHERE GENRE.NAME like 'Rock'
GROUP BY ARTIST.ARTIST_ID
ORDER BY NUMBER_OF_SONGS DESC
LIMIT 10;
```

**Output:**

Data Output | Messages | Notifications

| | artist_id [PK] character varying (50) | name character varying (120) | number_of_songs 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 |
| 8 | 142 | The Rolling Stones | 41 |
| 9 | 76 | Creedence Clearwater Revival | 40 |
| 10 | 52 | Kiss | 35 |

**Q.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.**

```
SELECT NAME,
    MILLISECONDS
FROM TRACK
WHERE MILLISECONDS >
        (SELECT AVG(MILLISECONDS)
            FROM TRACK)
ORDER BY MILLISECONDS DESC;
```

**Output:**

Data Output | Messages | Notifications

| | name<br>character varying (150) | milliseconds<br>integer |
|---|---|---|
| 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 |
| 8 | Battlestar Galactica, Pt. 3 | 2927802 |
| 9 | Take the Celestra | 2927677 |
| 10 | Fire In Space | 2926593 |

# Advance

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

```
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 ALBUM ON ALBUM.ALBUM_ID = TRACK.ALBUM_ID
       JOIN ARTIST ON ARTIST.ARTIST_ID = ALBUM.ARTIST_ID
       GROUP BY 1
       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 ALBUM 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;
```

**Output:**

Data Output | Messages | Notifications

| | customer_id integer | first_name character | last_name character | artist_name character varying (120) | amount_spent double precision |
|---|---|---|---|---|---|
| 1 | 46 | Hugh | O'Reilly | Queen | 27.719999999999985 |
| 2 | 38 | Niklas | Schröder | Queen | 18.81 |
| 3 | 3 | François | Tremblay | Queen | 17.82 |
| 4 | 34 | João | Fernandes | Queen | 16.830000000000002 |
| 5 | 53 | Phil | Hughes | Queen | 11.88 |
| 6 | 41 | Marc | Dubois | Queen | 11.88 |
| 7 | 47 | Lucas | Mancini | Queen | 10.89 |
| 8 | 33 | Ellie | Sullivan | Queen | 10.89 |
| 9 | 20 | Dan | Miller | Queen | 3.96 |
| 10 | 5 | R | Madhav | Queen | 3.96 |

**Q.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.**

```
WITH POPULAR_GENRE AS
    (SELECT COUNT(INVOICE_LINE.QUANTITY) AS PURCHASE,
            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)
SELECT *
FROM POPULAR_GENRE
WHERE ROWNO <= 1;
```

**Output:**

Data Output | Messages | Notifications

| | purchase 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 |
| 10 | 46 | Finland | Rock | 1 | 1 |

**Q.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.**

```
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;
```

**Output:**

Data Output | Messages | Notifications

| | 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 | 144.5400000000002 | 1 |
| 9 | 9 | Kara | Nielsen | Denmark | 37.61999999999999 | 1 |
| 10 | 44 | Terhi | Hämäläinen | Finland | 79.2 | 1 |