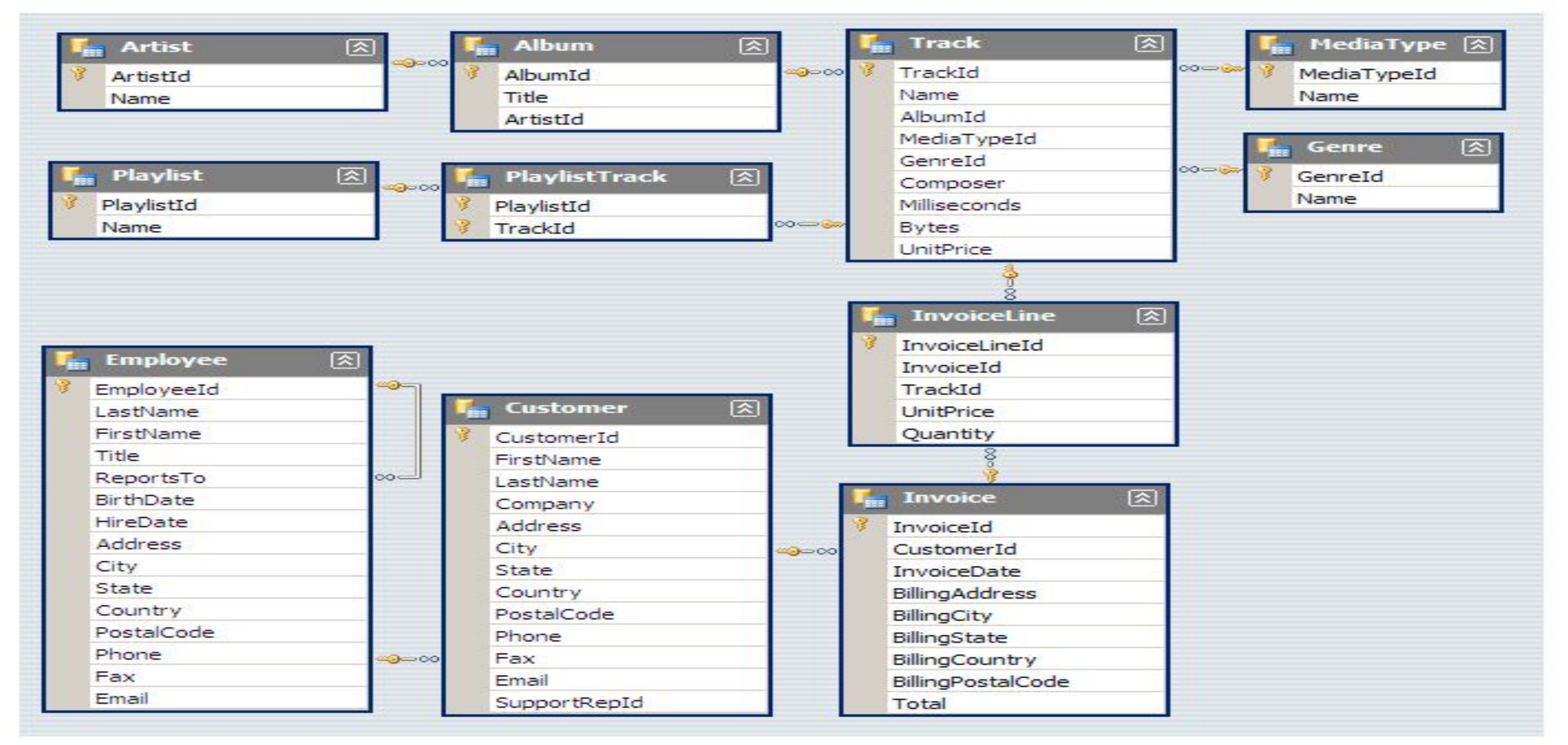
MUSIC STORE DATA ANALYSIS

VIJAY KUMAR

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

Who is the senior most employee based on job title?

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

Which countries have the most Invoices?

```
SELECT BILLING_COUNTRY, COUNT(*) AS
NUMBER_OF_INVOICES FROM INVOICE

GROUP BY BILLING_COUNTRY

ORDER BY NUMBER_OF_INVOICES DESC

LIMIT 3;
```

What are top 3 values of total invoice?

```
SELECT TOTAL FROM INVOICE

ORDER BY 1 DESC

LIMIT 3;
```

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.

```
SELECT BILLING_CITY,SUM(TOTAL) AS INVOICETOTAL

FROM INVOICE

GROUP BY BILLING_CITY

ORDER BY INVOICETOTAL DESC

LIMIT 1;
```

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

```
SELECT CONCAT(CUS.FIRST_NAME,' ',CUS.LAST_NAME) AS FULL_NAME,
SUM(INVO.TOTAL) AS GROUP_TOTAL FROM CUSTOMER CUS

JOIN INVOICE INVO ON

INVO.CUSTOMER_ID = CUS.CUSTOMER_ID

GROUP BY FULL_NAME

ORDER BY GROUP_TOTAL DESC

LIMIT 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  CUS.EMAIL, CUS.FIRST_NAME, CUS.LAST_NAME, GENRE.NAME

FROM CUSTOMER CUS JOIN INVOICE ON

INVOICE.CUSTOMER_ID = CUS.CUSTOMER_ID

JOIN INVOICE_LINE ON INVOICE.INVOICE_ID = INVOICE_LINE.INVOICE_ID

JOIN TRACK ON INVOICE_LINE.TRACK_ID = TRACK .TRACK_ID

JOIN GENRE ON TRACK.GENRE_ID =GENRE.GENRE_ID

WHERE GENRE.NAME LIKE 'ROCK'

ORDER BY 1;
```

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.NAME, COUNT(ARTIST.NAME) AS NUMBER_OF_SONGS FROM ARTIST

JOIN ALBUM ON ARTIST.ARTIST_ID = ALBUM.ARTIST_ID

JOIN TRACK ON ALBUM.ALBUM_ID = TRACK.ALBUM_ID

JOIN GENRE ON GENRE.GENRE_ID = TRACK.GENRE_ID

WHERE GENRE.NAME LIKE 'ROCK'

GROUP BY ARTIST.NAME

ORDER BY NUMBER_OF_SONGS DESC

LIMIT 10;
```

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) AS AVG_TRACK_LENGTH FROM TRACK )

ORDER BY MILLISECONDS DESC;
```

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;
```

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 CTE AS (

SELECT GENRE.NAME, CUSTOMER.COUNTRY,COUNT(INVOICE_LINE.QUANTITY) AS MOST_POPULAR_GENRE,

ROW_NUMBER() OVER(PARTITION BY CUSTOMER.COUNTRY ORDER BY

COUNT(INVOICE_LINE.QUANTITY) DESC) AS ROWNO

FROM CUSTOMER

JOIN INVOICE ON INVOICE.CUSTOMER_ID =CUSTOMER.CUSTOMER_ID

JOIN INVOICE_LINE ON INVOICE_LINE.INVOICE_ID = INVOICE.INVOICE_ID

JOIN TRACK ON TRACK.TRACK_ID = INVOICE_LINE.TRACK_ID

JOIN GENRE ON GENRE.GENRE_ID = TRACK.GENRE_ID

GROUP BY 1,2

ORDER BY 2 ASC ,3 DESC)

SELECT COUNTRY, NAME, MOST_POPULAR_GENRE FROM CTE

WHERE ROWNO <= 1
```

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 CTE AS (

SELECT CUSTOMER.FIRST_NAME,CUSTOMER.LAST_NAME ,CUSTOMER.COUNTRY,
SUM(INVOICE.TOTAL) AS TOTAL_SPEND,

ROW_NUMBER()OVER(PARTITION BY CUSTOMER.COUNTRY ORDER BY
SUM(INVOICE.TOTAL) DESC ) AS ROWNO

FROM CUSTOMER

JOIN INVOICE ON CUSTOMER.CUSTOMER_ID = INVOICE.CUSTOMER_ID

GROUP BY 1 ,2,3

ORDER BY 3 ASC, 4 DESC )

SELECT FIRST_NAME, COUNTRY, TOTAL_SPEND FROM CTE

WHERE ROWNO <= 1
```