Music store database analysis with SQL

Gordon Ma

2023-01-15

Introduction

The Chinook database is a publicly available database that provides a comprehensive overview of a music store's operations. Containing a wealth of information about employees, digital media, and more, this database offers a unique opportunity for data analysis and exploration. In this project, I leveraged the power of SQL comments to delve into the intricacies of the Chinook database, uncovering valuable insights and gaining a deeper understanding of its structure and relationships.

Database Structure

Before diving into the analysis, it's essential to understand the underlying structure of the Chinook database. The following diagram illustrates the database's schema, highlighting the various tables and their relationships:

As shown in the diagram, the Chinook database consists of multiple tables, each containing specific information about the music store's operations. The tables are interconnected, with relationships established through primary and foreign keys. This structure allows for efficient querying and analysis of the data.

There are a total of 11 tables in the Chinook sample database:

- `Employees` table stores employee data such as employee ID, last name, first name, etc. It also has a field named `ReportsTo` to specify who reports to whom.
- `Customers` table stores customer data.
- `Invoices` & `Invoice_items` tables: these two tables store invoice data. The `Invoices` table stores invoice header data and the `Invoice_items` table stores the invoice line items data.
- `Artists` table stores artists' data. It is a simple table that contains only the artist's ID and name.
- `Album` table stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums.
- `Media_types` table stores media types such as MPEG audio and AAC audio files.
- `Genres` table stores music types such as rock, jazz, metal, etc.
- `Tracks` table stores the data of songs. Each track belongs to one album.
- `Playlists` & `Playlist_track` tables: `Playlists` table stores data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The `Playlist_track` table is used to reflect this relationship.

Problem statements

We can ask problems such as:

- Which genres of music are the most popular among the customers?
- Which employers made the most sales? Which are the most loyal customers?

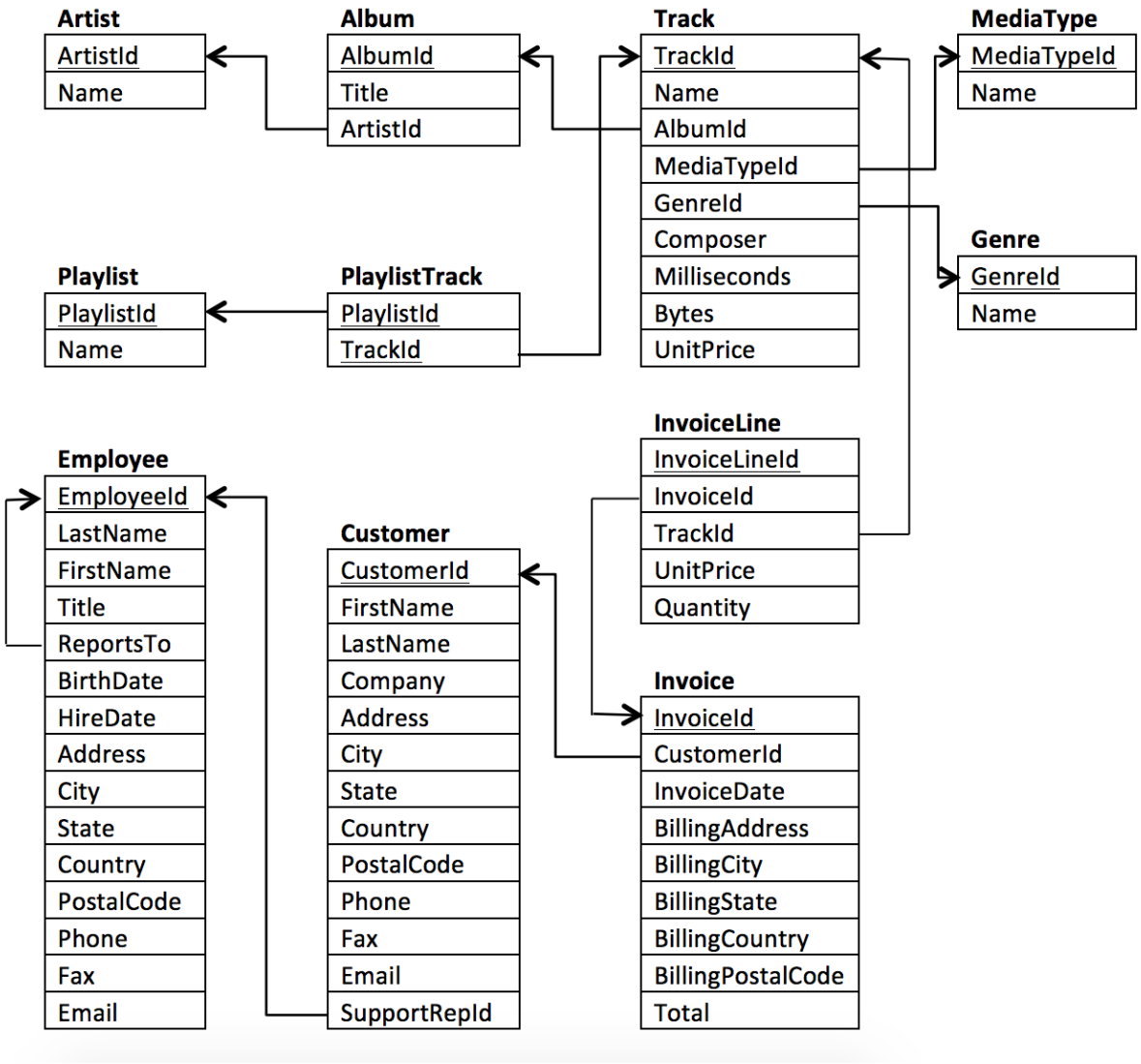


Figure 1: Chinook Database structure

- Which is the most successful artist?
- etc.

Data exploration

To analyze the database, I employed SQL to create a series of queries that extracted and manipulated the data. We first load the required packages in R and connect to the database file.

```
#We first load the required packages in R
library(RSQLite)
#Connect to your SQLite database file and list all the tables
db_conn <- dbConnect(SQLite(), dbname = "chinook.db")
dbListTables(db_conn)
```

```
##  [1] "albums"          "artists"         "customers"       "employees"
##  [5] "genres"          "invoice_items"   "invoices"        "media_types"
##  [9] "playlist_track"  "playlists"       "sqlite_sequence" "sqlite_stat1"
## [13] "tracks"
```

Let us look at the `employees` table to find out who works at the company.

```
query <- "SELECT *
          FROM employees
          "
result <- dbSendQuery(db_conn, query)
dbFetch(result)
```

```
##   EmployeeId LastName FirstName               Title ReportsTo
## 1          1    Adams    Andrew     General Manager        NA
## 2          2  Edwards     Nancy       Sales Manager         1
## 3          3  Peacock      Jane Sales Support Agent         2
## 4          4     Park  Margaret Sales Support Agent         2
## 5          5  Johnson     Steve Sales Support Agent         2
## 6          6 Mitchell   Michael          IT Manager         1
## 7          7     King    Robert            IT Staff         6
## 8          8 Callahan     Laura            IT Staff         6
##             BirthDate            HireDate                    Address
## 1 1962-02-18 00:00:00 2002-08-14 00:00:00         11120 Jasper Ave NW
## 2 1958-12-08 00:00:00 2002-05-01 00:00:00                825 8 Ave SW
## 3 1973-08-29 00:00:00 2002-04-01 00:00:00               1111 6 Ave SW
## 4 1947-09-19 00:00:00 2003-05-03 00:00:00            683 10 Street SW
## 5 1965-03-03 00:00:00 2003-10-17 00:00:00                7727B 41 Ave
## 6 1973-07-01 00:00:00 2003-10-17 00:00:00        5827 Bowness Road NW
## 7 1970-05-29 00:00:00 2004-01-02 00:00:00 590 Columbia Boulevard West
## 8 1968-01-09 00:00:00 2004-03-04 00:00:00                 923 7 ST NW
##         City State Country PostalCode             Phone               Fax
## 1   Edmonton    AB  Canada    T5K 2N1 +1 (780) 428-9482 +1 (780) 428-3457
## 2    Calgary    AB  Canada    T2P 2T3 +1 (403) 262-3443 +1 (403) 262-3322
## 3    Calgary    AB  Canada    T2P 5M5 +1 (403) 262-3443 +1 (403) 262-6712
## 4    Calgary    AB  Canada    T2P 5G3 +1 (403) 263-4423 +1 (403) 263-4289
## 5    Calgary    AB  Canada    T3B 1Y7  1 (780) 836-9987  1 (780) 836-9543
## 6    Calgary    AB  Canada    T3B 0C5 +1 (403) 246-9887 +1 (403) 246-9899
## 7 Lethbridge    AB  Canada    T1K 5N8 +1 (403) 456-9986 +1 (403) 456-8485
## 8 Lethbridge    AB  Canada    T1H 1Y8 +1 (403) 467-3351 +1 (403) 467-8772
##                      Email
```

```
## 1   andrew@chinookcorp.com
## 2    nancy@chinookcorp.com
## 3     jane@chinookcorp.com
## 4 margaret@chinookcorp.com
## 5    steve@chinookcorp.com
## 6  michael@chinookcorp.com
## 7   robert@chinookcorp.com
## 8    laura@chinookcorp.com
```

To find the employee who made the most sales, we need to join the tables `employees` and `Invoice` using the "JOIN BY" function. We then group the table by the employees' ID and count the total number of sales of each employee.

```
query <- "SELECT m.FirstName ||' '|| m.LastName AS ManagerName, E.FirstName ||' '|| E.LastName AS Emplo
  FROM employees AS E
  JOIN Customers AS C ON E.EmployeeId = C.SupportRepId
  JOIN invoices AS I ON I.CustomerId = C.CustomerId
  JOIN invoice_items AS II on I.InvoiceId = II.InvoiceId
  JOIN Employees m ON e.ReportsTo = m.EmployeeID
  GROUP BY E.EmployeeId
  ORDER BY total_sold DESC;
          "
result <- dbSendQuery(db_conn, query)
dbFetch(result)
```

```
##     ManagerName  EmployeeName              Title total_sold   value
## 1 Nancy Edwards  Jane Peacock Sales Support Agent        796 7427.06
## 2 Nancy Edwards Margaret Park Sales Support Agent        760 6931.40
## 3 Nancy Edwards Steve Johnson Sales Support Agent        684 6490.16
```

The sales support agent Jane Peacock made the highest sales with a total of 796 units.

How many customers are in each country? And how much sales are in each country?

We can use the `COUNT` function to count the distinct customers and `SUM` to sum up the total sales.

```
query <- "SELECT C.COUNTRY, COUNT(DISTINCT C.CUSTOMERID) AS total_user_count, SUM(I.total) AS total_sal
  FROM customers AS C
  JOIN invoices AS I
  GROUP BY 1
  ORDER BY 3 DESC;
          "
result <- dbSendQuery(db_conn, query)
(users <- dbFetch(result))
```

```
##           Country total_user_count total_sales
## 1             USA               13     30271.8
## 2          Canada                8     18628.8
## 3          France                5     11643.0
## 4          Brazil                5     11643.0
## 5         Germany                4      9314.4
## 6  United Kingdom                3      6985.8
## 7        Portugal                2      4657.2
## 8           India                2      4657.2
## 9  Czech Republic                2      4657.2
## 10         Sweden                1      2328.6
```

```
## 11          Spain                1      2328.6
## 12         Poland                1      2328.6
## 13         Norway                1      2328.6
## 14    Netherlands                1      2328.6
## 15          Italy                1      2328.6
## 16        Ireland                1      2328.6
## 17        Hungary                1      2328.6
## 18        Finland                1      2328.6
## 19        Denmark                1      2328.6
## 20          Chile                1      2328.6
## 21        Belgium                1      2328.6
## 22        Austria                1      2328.6
## 23      Australia                1      2328.6
## 24      Argentina                1      2328.6
```

The USA has the highest user count of 13 and the highest total sales.

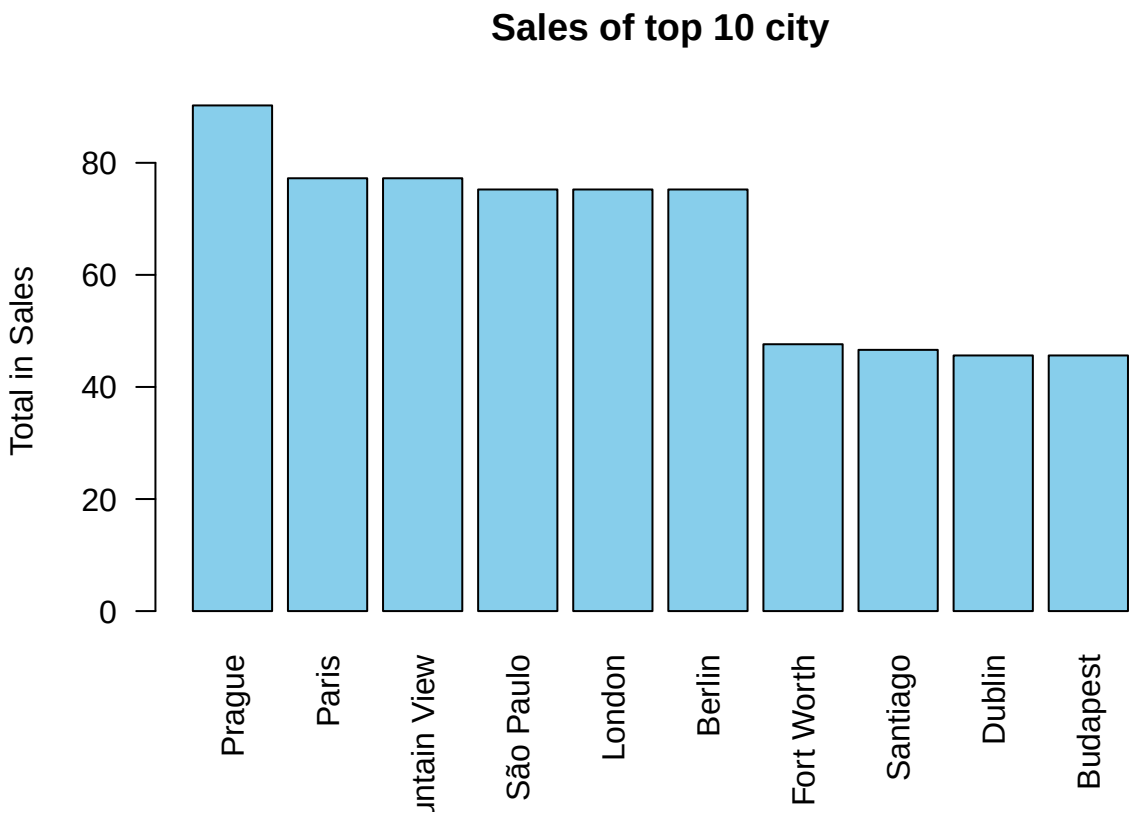
Which city has the most sales?

```
query <- "SELECT BILLINGCITY AS City,
  SUM(TOTAL) AS profits
  FROM invoices
  GROUP BY 1
  ORDER BY 2 DESC;
          "
result <- dbSendQuery(db_conn, query)
(city_sales <- dbFetch(result))
```

```
##                   City profits
## 1               Prague   90.24
## 2                Paris   77.24
## 3        Mountain View   77.24
## 4            São Paulo   75.24
## 5               London   75.24
## 6               Berlin   75.24
## 7           Fort Worth   47.62
## 8             Santiago   46.62
## 9               Dublin   45.62
## 10            Budapest   45.62
## 11      Salt Lake City   43.62
## 12           Frankfurt   43.62
## 13             Chicago   43.62
## 14              Vienne   42.62
## 15             Madison   42.62
## 16            Helsinki   41.62
## 17               Dijon   40.62
## 18           Amsterdam   40.62
## 19 São José dos Campos   39.62
## 20             Redmond   39.62
## 21                Oslo   39.62
## 22             Orlando   39.62
## 23            Montréal   39.62
## 24              Lisbon   39.62
## 25            Bordeaux   39.62
```

```
## 26           Vancouver   38.62
## 27           Stockholm   38.62
## 28               Delhi   38.62
## 29           Cupertino   38.62
## 30         Yellowknife   37.62
## 31            Winnipeg   37.62
## 32              Warsaw   37.62
## 33              Tucson   37.62
## 34             Toronto   37.62
## 35           Stuttgart   37.62
## 36              Sidney   37.62
## 37                Rome   37.62
## 38      Rio de Janeiro   37.62
## 39                Reno   37.62
## 40               Porto   37.62
## 41              Ottawa   37.62
## 42            New York   37.62
## 43              Madrid   37.62
## 44                Lyon   37.62
## 45             Halifax   37.62
## 46           Edmonton   37.62
## 47          Edinburgh   37.62
## 48          Copenhagen   37.62
## 49        Buenos Aires   37.62
## 50            Brussels   37.62
## 51            Brasília   37.62
## 52              Boston   37.62
## 53           Bangalore   36.64
```

```r
#Visualization of each city's profits
barplot(city_sales$profits[1:10], names.arg = city_sales$City[1:10], col = "skyblue", main = "Sales of
```

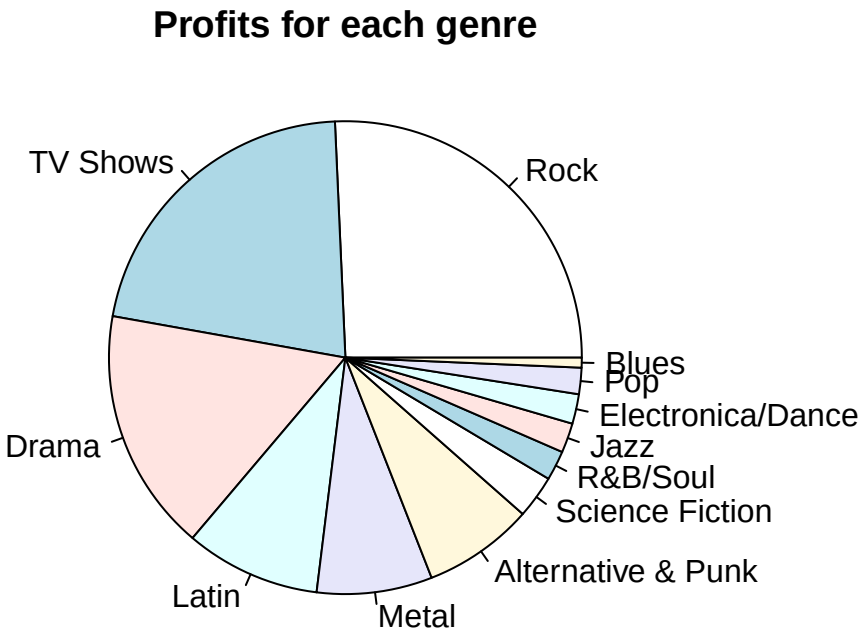


Prague has the highest number of sales. Let us find out the genres of music people enjoy in this city.

```
query <- "SELECT G.Name, SUM(TOTAL) as profits
  FROM invoices AS I
  JOIN invoice_items AS II ON I.InvoiceId = II.InvoiceId
  JOIN tracks AS T ON T.TrackId = II.TrackId
  JOIN genres AS G ON T.GenreId = G.GenreId
  WHERE I.billingCity = 'Prague'
  GROUP BY G.Name
  ORDER BY profits DESC;
          "
result <- dbSendQuery(db_conn, query)
(genres_profits <- dbFetch(result) )
```

```
##                  Name profits
## 1                Rock  226.02
## 2            TV Shows  188.88
## 3               Drama  146.16
## 4               Latin   81.27
## 5               Metal   69.36
## 6  Alternative & Punk   66.45
## 7     Science Fiction   25.86
## 8             R&B/Soul   17.82
## 9                Jazz   17.82
## 10  Electronica/Dance   17.82
## 11                Pop   15.84
## 12              Blues    5.94
```

```
#Create a pie chart to visualize the profits of each generes
pie(genres_profits$profits, genres_profits$Name, radius = 1.0, main = "Profits for each genre")
```



So Rock music is the most popular in Prague. Let us now find out which artists are known for this genres and are most popular. The following code returns the list of artists who have at least 20 rock music, and order by the total sales.

```
query <- "SELECT AR.Name, COUNT(T.Name) AS Num_of_RockMusic, SUM(I.Total) AS Profits_made
  FROM tracks AS T
  JOIN genres AS G ON T.GENREID = G.GENREID
  JOIN albums AS AL ON AL.ALBUMID = T.ALBUMID
  JOIN artists AS AR ON AR.ARTISTID = AL.ARTISTID
  JOIN invoice_items AS II ON II.TrackId = T.TrackId
  JOIN invoices AS I ON I.InvoiceId = II.InvoiceId
  WHERE G.Name = 'Rock'
  GROUP BY AR.Name HAVING Num_of_RockMusic >= 20
  ORDER BY 3 DESC
           "
result <- dbSendQuery(db_conn, query)
dbFetch(result)
```

```
##                          Name Num_of_RockMusic Profits_made
## 1                          U2               91       773.65
## 2              Led Zeppelin               87       620.73
## 3               Deep Purple               44       550.44
## 4               Iron Maiden               54       473.22
## 5                 Van Halen               29       336.82
```

```
## 6                     Pearl Jam               26       335.61
## 7                         Queen               37       256.41
## 8  Creedence Clearwater Revival               37       215.82
## 9                          Kiss               31       211.86
## 10               Guns N' Roses               26       142.56
```

The band U2 had the highest number of rock music and made the most profits.

What is the best-selling musical albums by U2?

```
query <- "SELECT AL.title, SUM(Quantity) as total_sold
          FROM albums AS AL
          JOIN tracks AS T ON T.AlbumId = AL.AlbumId
          JOIN invoice_items as II ON II.trackid = T.trackid
          WHERE artistid IN (
          SELECT artistid
          FROM artists
          WHERE name = 'U2'
          )
          GROUP BY al.title
          ORDER BY total_sold DESC
          "
result <- dbSendQuery(db_conn, query)
dbFetch(result)
```

```
##                                                                Title total_sold
## 1                                                    Rattle And Hum         17
## 2  Instant Karma: The Amnesty International Campaign to Save Darfur         16
## 3                                                                War         11
## 4                                           The Best Of 1980-1990         11
## 5                                               B-Sides 1980-1990         11
## 6                                                              Pop         10
## 7                                   How To Dismantle An Atomic Bomb         10
## 8                                                         Zooropa          9
## 9                                   All That You Can't Leave Behind          6
## 10                                                   Achtung Baby          6
```

Which playlist contain U2's tracks?

```
query <- "SELECT P.Name AS name_of_the_playlist, COUNT(DISTINCT T.Name) as total_U2_song
          FROM playlists as P
          JOIN playlist_track as PT ON P.PlaylistId = PT.PlaylistId
          JOIN tracks AS T ON T.trackid = PT.trackid
          WHERE AlbumId IN (
            SELECT AlbumId
            FROM artists AS AR
            JOIN albums AS AL ON AR.artistid = AL.artistid
            WHERE AR.name = 'U2'
          )
          GROUP BY P.PlaylistId
          ORDER BY total_U2_song DESC;
          "
result <- dbSendQuery(db_conn, query)
dbFetch(result)
```

```
##   name_of_the_playlist total_U2_song
```

```
## 1               Music          124
## 2               Music          124
## 3          90's Music           62
```

```
#Explore the 90's Music playlist
query <- "SELECT T.*
          FROM tracks AS T
          JOIN playlist_track as PT ON T.trackid = PT.trackid
          JOIN playlists as PL on PL.playlistid = PT.playlistid
          WHERE PL.name = '90's Music'
          LIMIT 10;
          "
result <- dbSendQuery(db_conn, query)
dbFetch(result)
```

```
##    TrackId                     Name AlbumId MediaTypeId GenreId
## 1        3          Fast As a Shark       3           2       1
## 2        4        Restless and Wild       3           2       1
## 3        5     Princess of the Dawn       3           2       1
## 4       23            Walk On Water       5           1       1
## 5       24      Love In An Elevator       5           1       1
## 6       25                Rag Doll       5           1       1
## 7       26            What It Takes       5           1       1
## 8       27 Dude (Looks Like A Lady)       5           1       1
## 9       28        Janie's Got A Gun       5           1       1
## 10      29                   Cryin'       5           1       1
##                                                                Composer
## 1                     F. Baltes, S. Kaufman, U. Dirkscneider & W. Hoffman
## 2  F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. Dirkscneider & W. Hoffman
## 3                                                 Deaffy & R.A. Smith-Diesel
## 4                       Steven Tyler, Joe Perry, Jack Blades, Tommy Shaw
## 5                                             Steven Tyler, Joe Perry
## 6                     Steven Tyler, Joe Perry, Jim Vallance, Holly Knight
## 7                                  Steven Tyler, Joe Perry, Desmond Child
## 8                                  Steven Tyler, Joe Perry, Desmond Child
## 9                                             Steven Tyler, Tom Hamilton
## 10                                 Steven Tyler, Joe Perry, Taylor Rhodes
##    Milliseconds    Bytes UnitPrice
## 1        230619  3990994      0.99
## 2        252051  4331779      0.99
## 3        375418  6290521      0.99
## 4        295680  9719579      0.99
## 5        321828 10552051      0.99
## 6        264698  8675345      0.99
## 7        310622 10144730      0.99
## 8        264855  8679940      0.99
## 9        330736 10869391      0.99
## 10       309263 10056995      0.99
```

Key Findings and Insights

The analysis of the Chinook database using SQL queries revealed several key findings and insights. For example, I discovered that the music store's sales were mostly in the US, followed by Canada, France, Brazil and Germany. This analysis can be performed at the city level, where we find the city of Prague has the

highest sales. This insight has important implications for the store's marketing strategy.

I also perform analysis to discover the most popular types of music and artists. Using this information, I can recommend similar music to users based on their preferences. This can assist the store in developing effective marketing strategies and providing better services to its customers.

Furthermore, the analysis revealed that certain employees were more likely to generate high sales volumes. This finding has important implications for employee training and development, suggesting that targeted training programs could help to boost sales performance.