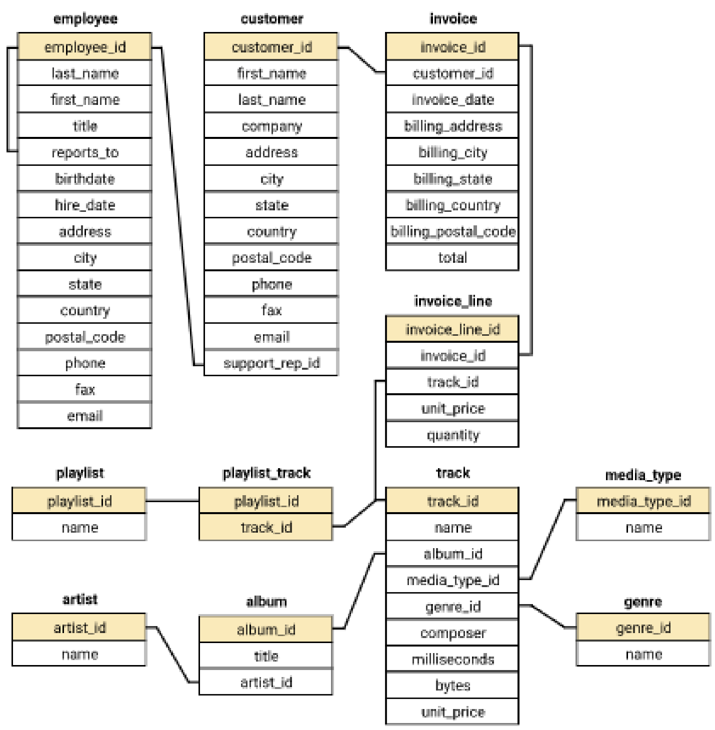
Chinook Store Analysis

Introduction

The chinook database is a sample database available for a variety of SQL flavours and it offers a good opportunity to practice SQL. In this project, we are going to be querying the chinook database to answer certain hypothetical business questions. The questions are as follows:

- We want to figure out which 3 album out of 4 to add to our store based on popular genres in the USA.
- We want to evaluate sales employee performance based on how much sales they have made.
- We want to find out the most valuable country for our business.
- We want to compare what percentages of our sales are from album sales and track sales.
- We want to find out the most popular artists in playlists.
- We want to find out how many tracks have been purchased and not purchased.
- We want to find out if the range of tracks in the store is reflective of their sales popularity.
- We want to find out if protected or unprotected tracks have any effect on track popularity.

Below is the schema for the chinook database




```r
# loading required packages
library(tidyverse)
library(RSQLite)
library(DBI)
library(kableExtra)
```

Creating Functions

We are going to create 3 functions, one to render tables in our pdf output, the other two we will use to interact with our database and run SQL queries and display the tables in our database respectively.

```
# function to render tibbles as  pdf tables
render_table <- function(table, scale_down=F){
  if(scale_down == T){
    rendered_table <- kbl(table) %>% kable_styling(
      latex_options = c("stripe", "HOLD_position", "scale_down")
    )
  } else{
    rendered_table <- kbl(table) %>% kable_styling(
      latex_options = c("stripe", "HOLD_position")
    )
  }
  return(rendered_table)
}
```

```
# function to run SQL queries
run_query <- function(query){
  conn <- dbConnect(SQLite(),"chinook.db")
  result <- dbGetQuery(conn, query)
  dbDisconnect(conn)
  return(as_tibble(result))
}

# function to show tables in the database

show_table <- function(){
  query <- "SELECT
    name,
    type
  FROM sqlite_master
 WHERE type IN ('table', 'view');"

  return(run_query(query))
}
```

Let's look at the list of all the tables in our database.

```
show_table() %>% render_table()
```

name	type
album	table
artist	table
customer	table
employee	table
genre	table
invoice	table
invoice_line	table
media_type	table
playlist	table
playlist_track	table
track	table

Case 1

The Chinbook store has just signed a deal with a new record label that specialises with artist from the US and we are tasked with finding out the three albums out of four to add to our store. All four artists have no tracks in our store and each specialise in different genre of music. We are going to pick 3 out of the 4 artists based on which of their genres generate more sales in the US. Below is a table showing the artist name and their genre of music.

Artist	Genre
Regal	Hip-Hop
Red Tone	Punk
Meteor and the Girls	Pop
Slim Jim Bites	Blues

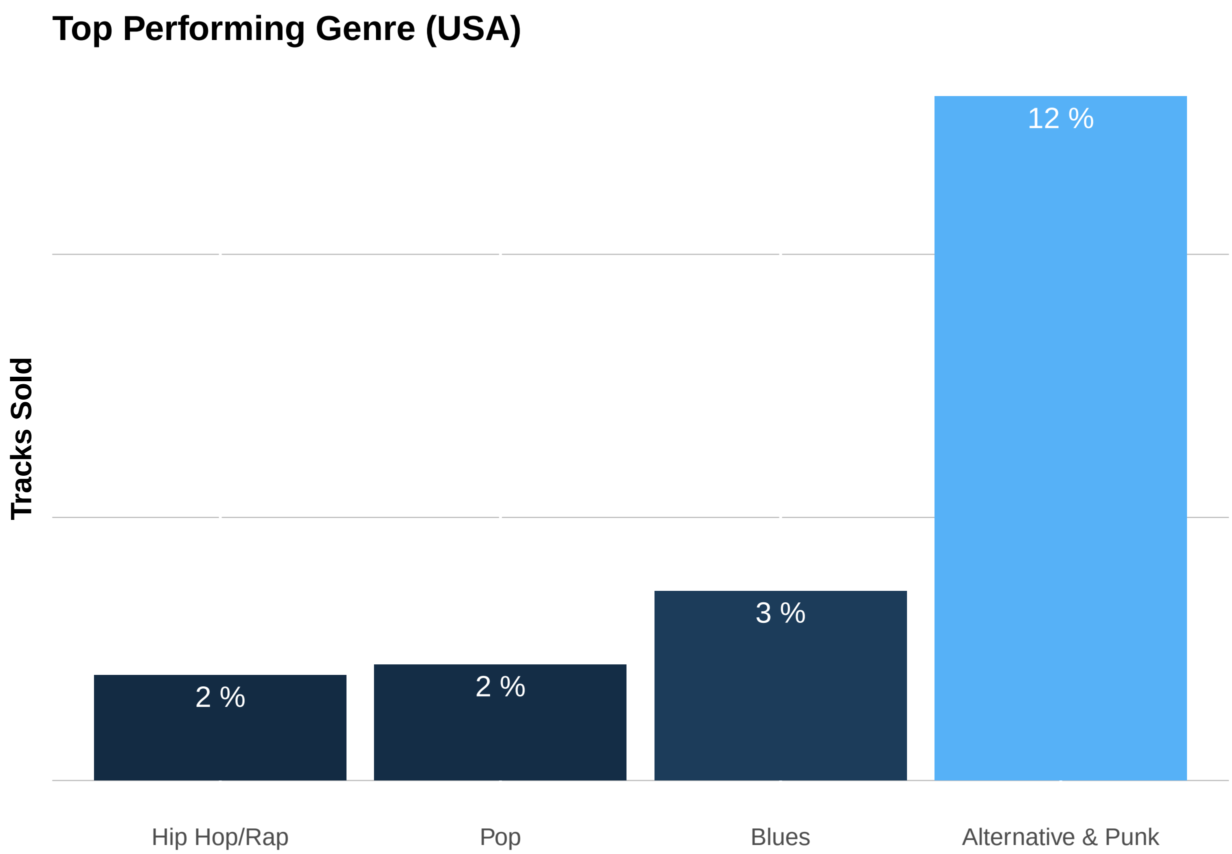
```
query1 <-  "WITH us_records AS
     (SELECT
            c.country,
            il.track_id
        FROM customer AS c
        LEFT JOIN invoice AS i
          ON i.customer_id = c.customer_id
        LEFT JOIN invoice_line AS il
          ON i.invoice_id = il.invoice_id
        WHERE country = 'USA'
     ),


us_genre_records AS
    (SELECT
          g.name AS genre,
          COUNT(*) AS tracks_sold

      FROM us_records AS ur
      LEFT JOIN track AS t
        ON t.track_id = ur.track_id
      LEFT JOIN genre AS g
        ON g.genre_id = t.genre_id
     GROUP BY genre
    )

SELECT
      *,
      ROUND(CAST(tracks_sold AS FLOAT) / (SELECT
                                              SUM(tracks_sold)
                                          FROM us_genre_records
                                         ), 3) AS percentage_sold
  FROM us_genre_records
 ORDER BY tracks_sold DESC ;"
```

```
top_US_genre <- run_query(query1)
top_US_genre %>% render_table()
```

genre	tracks_sold	percentage_sold
Rock	561	0.534
Alternative & Punk	130	0.124
Metal	124	0.118
R&B/Soul	53	0.050
Blues	36	0.034
Alternative	35	0.033
Latin	22	0.021
Pop	22	0.021
Hip Hop/Rap	20	0.019
Jazz	14	0.013
Easy Listening	13	0.012
Reggae	6	0.006
Electronica/Dance	5	0.005
Classical	4	0.004
Heavy Metal	3	0.003
Soundtrack	2	0.002
TV Shows	1	0.001

Above we can see all of the top genres in the US. We are going to filter the table and select only the genres of the four artists in the record label we just signed a deal with and see how they compare.

```r
required_genre <- c("Hip Hop/Rap", "Alternative & Punk", "Pop", "Blues")
top_artist_genre <- top_US_genre %>% filter(genre %in% required_genre)
top_artist_genre %>% mutate(
  genre = factor(genre, levels = genre) # converts the genre column to a categorical column
)  %>% ggplot(
  aes(x= genre, y = tracks_sold, fill = tracks_sold)
  ) +
  scale_x_discrete(limits=rev) + # reverses the order of the bar plot

  geom_bar(stat = "identity", show.legend = F) +
  labs(
    title = "Top Performing Genre (USA)",
    y = "Tracks Sold"
  ) +
  geom_text(aes(label = paste(round(percentage_sold * 100), "%")),
              vjust = 1.5, color="white") +
  theme(
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.ticks = element_blank(),
    axis.text.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_blank(),
    panel.grid.major.y = element_line(colour = "gray", linewidth = 0.2)
  )
```



Alternative/Punk accounted for 12% of all the tracks sold in the US so Red Tone's music should be the first one in our store. Blues sold 3% while Hip Hop and pop both sold 3% in the US. So any 2 of the remaining 3 artists can complete the list. In this case we will be going with Red Tone, Meteor and the Girls and then Regal.

Case 2

We want to evaluate the sales employees performance at Chinook based on how much sale each employee has generated.

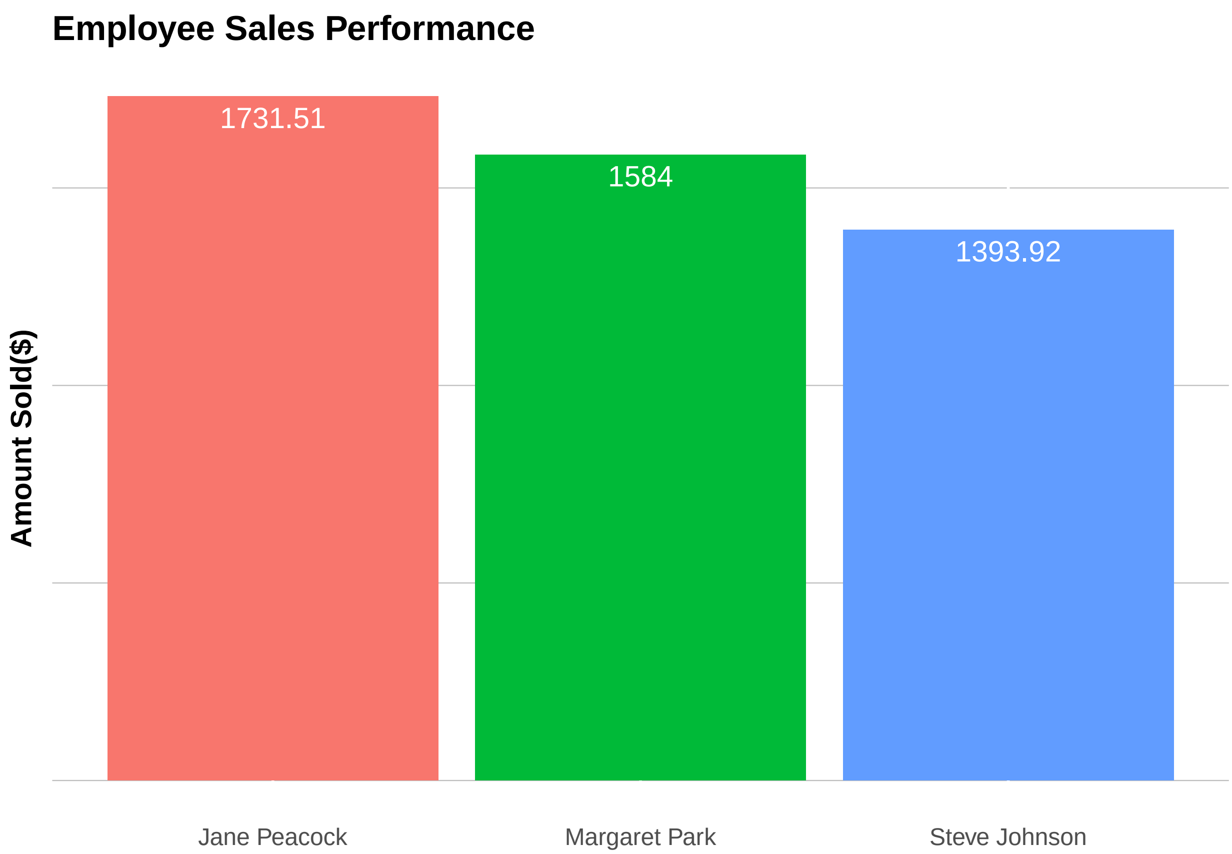
```
query2 <- "WITH invoice_details AS
        (SELECT *,
             SUM(il.quantity) AS total_quantity_sold
        FROM invoice_line AS il
        INNER JOIN invoice AS i  on il.invoice_id = i.invoice_id
        GROUP BY il.invoice_id
        ),

customer_invoice AS
        (SELECT *,
                SUM(total) AS customer_total,
                SUM(total_quantity_sold) AS total_quantity_purchased
           FROM invoice_details AS iv
          INNER JOIN customer AS c
             ON c.customer_id = iv.customer_id
          GROUP BY c.customer_id
        )
```

```
SELECT e.first_name || ' ' || e.last_name AS employee_name,
       e.title,
       e.hire_date,
       ROUND(SUM(ci.customer_total), 2) AS total_amount_sold,
       SUM(ci.total_quantity_purchased) AS total_quantity_sold
  FROM customer_invoice AS ci
 INNER JOIN employee AS e
    ON e.employee_id = ci.support_rep_id
 GROUP BY e.employee_id ;"
```

```
employee_perf <- run_query(query2)
employee_perf %>% render_table()
```

employee_name	title	hire_date	total_amount_sold	total_quantity_sold
Jane Peacock	Sales Support Agent	2017-04-01 00:00:00	1731.51	1749
Margaret Park	Sales Support Agent	2017-05-03 00:00:00	1584.00	1600
Steve Johnson	Sales Support Agent	2017-10-17 00:00:00	1393.92	1408

```
employee_perf %>% ggplot(
  aes(x=employee_name, y = total_amount_sold, fill = employee_name) ) +

  geom_bar(stat = "identity", show.legend = F) +
  labs(
    title = "Employee Sales Performance",
    y = "Amount Sold($)"
  ) +
  geom_text(aes(label = total_amount_sold),
            vjust = 1.5, colour = "white") +
  theme(
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.ticks = element_blank(),
    axis.title.x = element_blank(),
    axis.text.y = element_blank(),
    panel.background = element_blank(),
    panel.grid.major.y = element_line(colour = "gray", linewidth = 0.2)
  )
```



Jane Peacock is our top performing sales employee, followed by Margaret Park and then Steve Johnson. From the table showing the employee performance, you'll notice that the employees with more sales were hired before other employees, so the reason behind the difference in their sales performance is principally because of how long each employee has spent in the sales department.

Case 3

We want to find out which country is our most valuable market by finding out the average customer value and total track sales in these countries. Countries with only 1 customer will be grouped as others.

```
query3 <- "WITH country_or_other AS (
  SELECT il.*,
         c.customer_id,
         CASE WHEN(
          SELECT COUNT(*)
          FROM customer
          WHERE country = c.country
          GROUP BY country
          ) = 1
          THEN 'Others'
          ELSE c.country
          END AS country
  FROM invoice_line il
  JOIN invoice i ON i.invoice_id = il.invoice_id
  JOIN customer c ON c.customer_id = i.customer_id)

  SELECT country,
```

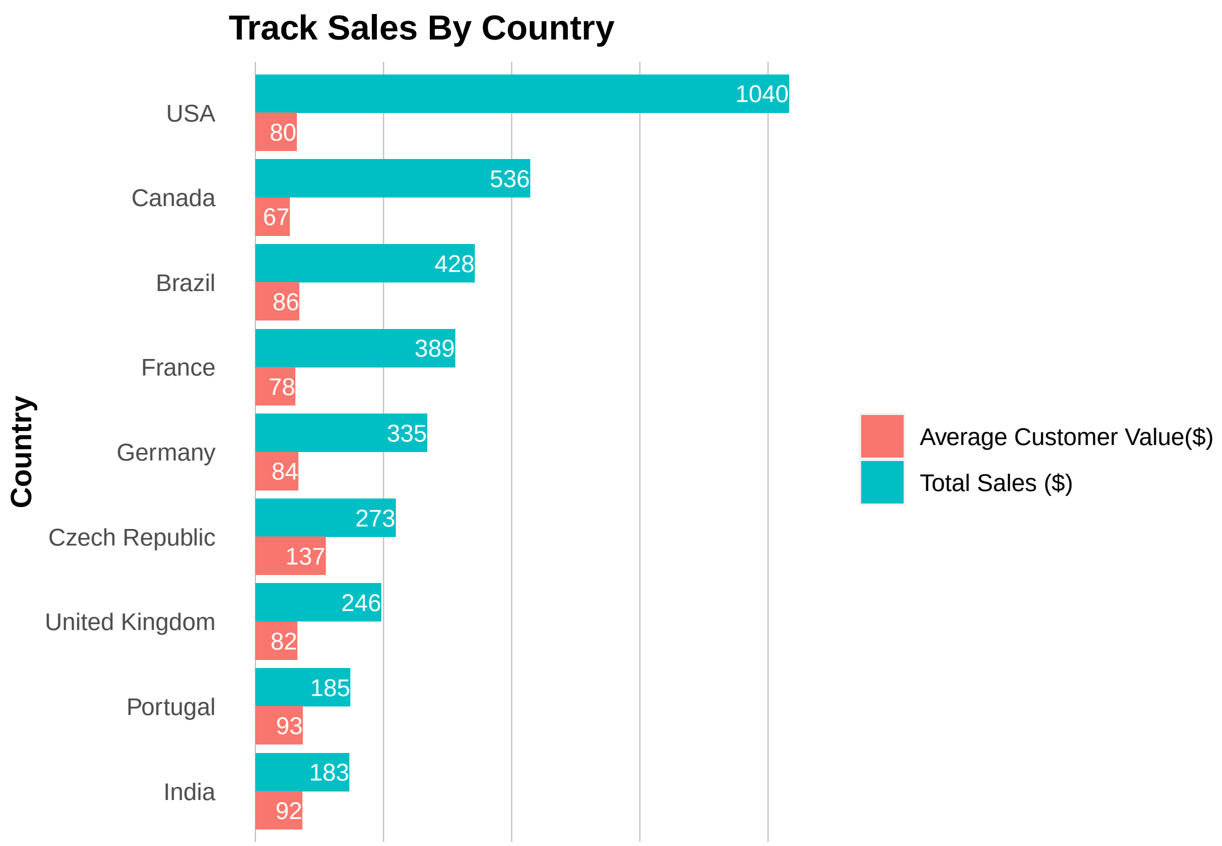
```
         customers,
         average_order_value,
         average_customer_value,
         total_sales
  FROM (
        SELECT country,
               count(DISTINCT customer_id) AS customers,
               ROUND(SUM(unit_price), 2) AS total_sales,
               ROUND(SUM(unit_price) / COUNT(DISTINCT customer_id), 2) AS average_customer_value,
               ROUND(SUM(unit_price) / COUNT(DISTINCT invoice_id), 2) AS average_order_value,
               CASE
                  WHEN country = 'Others'
                  THEN 1
                  ELSE 0
                  END AS sort
         FROM country_or_other
         GROUP BY country
         ORDER BY sort, total_sales DESC) ;"
```

```
country_sales <- run_query(query3)
country_sales %>% render_table()
```

country	customers	average_order_value	average_customer_value	total_sales
USA	13	7.94	80.04	1040.49
Canada	8	7.05	66.95	535.59
Brazil	5	7.01	85.54	427.68
France	5	7.78	77.81	389.07
Germany	4	8.16	83.66	334.62
Czech Republic	2	9.11	136.62	273.24
United Kingdom	3	8.77	81.84	245.52
Portugal	2	6.38	92.57	185.13
India	2	8.72	91.58	183.15
Others	15	7.45	73.00	1094.94

```
country_sales %>% filter(country != "Others") %>% mutate(
  country = factor(country, levels = country)
) %>% pivot_longer(
  cols = c("total_sales", "average_customer_value"),
  names_to = "sales",
  values_to = "value") %>% ggplot(
    aes(x = value, y = country, fill = sales ) ) +

  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Track Sales By Country",
    y = "Country"
  ) +
  scale_y_discrete(limits=rev) +
  scale_fill_discrete(labels = c("Average Customer Value($)",
                                 "Total Sales ($)")
                    ) +
  geom_text(aes(label = round(value)), # labelling the bars
             colour = "white", size = 3,
```

```
hjust = 1, position = position_dodge(.9)) +

theme(
  plot.title = element_text(face = "bold"),
  axis.title = element_text(face = "bold"),
  axis.ticks = element_blank(),
  axis.title.x = element_blank(),
  axis.text.x = element_blank(),
  legend.title  = element_blank(),
  panel.background = element_blank(),
  panel.grid.major.x = element_line(colour = "gray", linewidth = 0.2)
)
```



Based on the tracks sold, the United States, Canada and Brazil are our most valuable markets but when we look at the average customer value then the Czech republic, Portugal and India are our most valuable market.

Case 4

The management are currently considering changing their purchasing strategy to save money. The strategy they are considering is to purchase only the most popular tracks from each album from record companies, instead of purchasing every track from an album.

We have been asked to find out what percentage of purchases are individual tracks vs whole albums, so that management can use this data to understand the effect this decision might have on overall revenue.

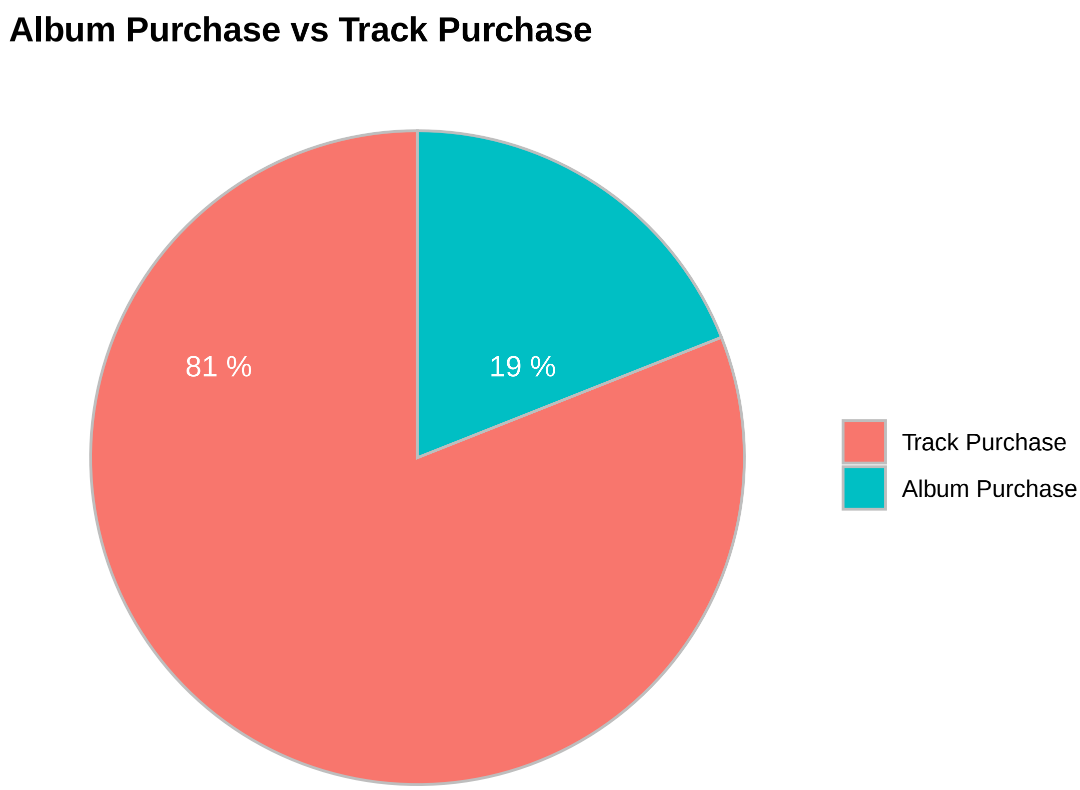
```
query4 <- "WITH invoice_track_info AS
     (SELECT
```

```
            invoice_id,
            MAX(track_id) AS track_id
        FROM invoice_line
       GROUP BY invoice_id
     )

SELECT
     album_purchase,
     COUNT(invoice_id) AS invoices,
     ROUND(CAST(COUNT(invoice_id) AS FLOAT) / (SELECT COUNT(*)
             FROM invoice), 2) AS percentage
  FROM (SELECT it.*,
               CASE WHEN (SELECT t.track_id
                            FROM track AS t
                           WHERE album_id = (
                               SELECT t2.album_id FROM track AS t2
                                WHERE t2.track_id = it.track_id
                          )

                          EXCEPT

                          SELECT il2.track_id
                             FROM invoice_line AS il2
                            WHERE il2.invoice_id = it.invoice_id

                        ) IS NULL

                    AND
                        (SELECT il2.track_id
                           FROM invoice_line AS il2
                          WHERE il2.invoice_id = it.invoice_id

                 EXCEPT
                        SELECT t.track_id
                          FROM track AS t
                         WHERE t.album_id = (
                         SELECT t2.album_id FROM track AS t2
                          WHERE t2.track_id = it.track_id
                         )
                        ) IS NULL

                         THEN 'Yes'
                         ELSE 'No'
                         END AS album_purchase
          FROM invoice_track_info AS it
       )
 GROUP BY album_purchase ;"
```

```
album_v_track <- run_query(query4)
album_v_track %>% render_table()
```

album_purchase	invoices	percentage
No	500	0.81
Yes	114	0.19

```
album_v_track %>% ggplot(
  aes(x="", y = percentage, fill = album_purchase)
) +
  geom_bar(stat = "identity", color = "gray") +
  labs(title = "Album Purchase vs Track Purchase") +
  scale_fill_discrete(label = c("Track Purchase", "Album Purchase")) +
  coord_polar("y", start = 0) + # turns the bar chart into a pie chart
  geom_text(aes(label = paste(percentage * 100, "%")),
            hjust = 1.2, vjust = -1, colour = "white") +
  theme_void() +
  theme(legend.title = element_blank(),
        plot.title = element_text(face = "bold"))
```



Album purchases accounts for only 19% of the sale, it makes no sense to throw away 195 of our revenue stream. It also wouldn't be a good strategy to purchase only the popular tracks from albums, this overall will reduce the amount of tracks we have to sell and will lead to a drop in revenue. The current strategy of purchasing all the tracks in albums is good as it is.

Case 5

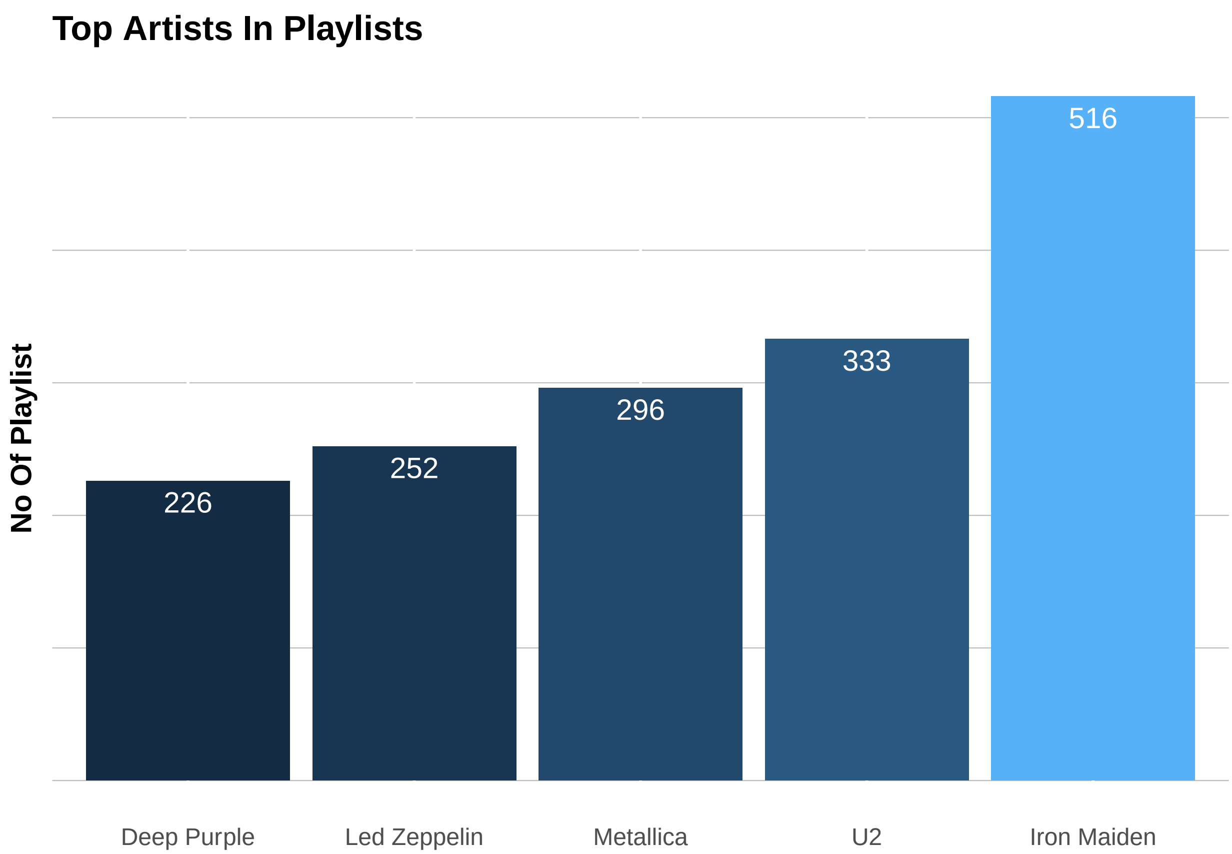
We want to find out which artists are the most common in playlists. We are going to look at the top 5 artists that can be found in customer's playlist.

```r
query5 <- "SELECT
      a.name AS artist_name,
      COUNT(pt.playlist_id) AS number_of_playlist
 FROM artist a
INNER JOIN album ab
   ON ab.artist_id  = a.artist_id
INNER JOIN track t
   ON ab.album_id = t.album_id
INNER JOIN playlist_track pt
   ON pt.track_id = t.track_id
GROUP BY a.name
ORDER BY number_of_playlist DESC
LIMIT 5 ;"
```

```r
top_artist_playlist <- run_query(query5)
top_artist_playlist %>% render_table()
```

artist_name	number_of_playlist
Iron Maiden	516
U2	333
Metallica	296
Led Zeppelin	252
Deep Purple	226

```r
top_artist_playlist %>% mutate(
  artist_name = factor(artist_name, levels = artist_name)
) %>% ggplot(
  aes(x = artist_name, y = number_of_playlist, fill = number_of_playlist) ) +

  geom_bar(stat = "identity", show.legend = F) +
  labs(
    title = "Top Artists In Playlists",
    y = "No Of Playlist"
  ) +
  scale_x_discrete(limits = rev) +
  geom_text(aes(label = number_of_playlist),
          vjust = 1.5, color = "white") +
  theme(
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.ticks = element_blank(),
    axis.title.x = element_blank(),
    axis.text.y = element_blank(),
    panel.background = element_blank(),
    panel.grid.major.y = element_line(colour = "gray", linewidth = 0.2)
  )
```



Case 6

We want to find out how many tracks in our store have been purchased and how many tracks have not been purchased.

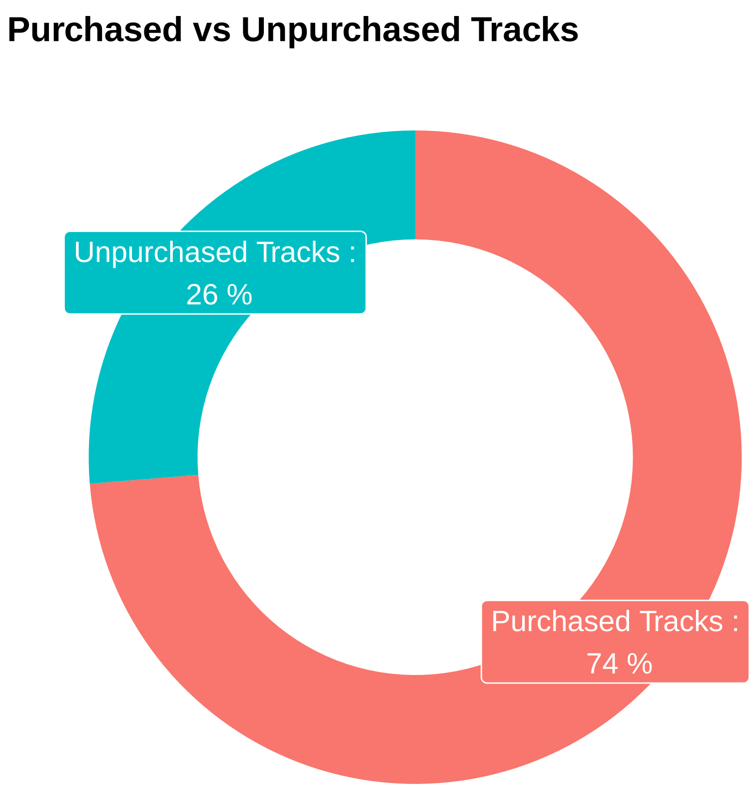
```
query6 <- " SELECT COUNT(il.track_id) purchased_tracks,
                   COUNT(t.track_id) - COUNT(il.track_id) unpurchased_tracks
              FROM track t
              LEFT JOIN invoice_line il ON il.track_id = t.track_id ;"
```

```
purchased_v_unpurchased <- run_query(query6)
purchased_v_unpurchased %>% render_table()
```

purchased_tracks	unpurchased_tracks
4757	1697

```
purchased_v_unpurchased %>% pivot_longer(
  cols = c("purchased_tracks", "unpurchased_tracks")
) %>% mutate(
  percentage = value / sum(value)
) %>% mutate(
  ymax = cumsum(percentage), # creating ymin and ymax values for rectangle plot
  ymin = c(0, percentage[1])
) %>% mutate(
  label = c("Purchased Tracks", "Unpurchased Tracks")
) %>% ggplot(
```

```
  aes(ymax = ymax, ymin = ymin, xmax=4, xmin=3, fill = name)
) +
  geom_rect() +
  labs(title = "Purchased vs Unpurchased Tracks") +
  coord_polar("y") + # turns rectangle plot to a donut chart
  geom_label(x=3.5, aes(y = (ymax + ymin)/2, label = paste(label, ":\n",round(percentage * 100), "%")),
  xlim(c(1, 4)) +
  theme_void() +
  theme(legend.position = "none",
        plot.title = element_text(face = "bold")
        )
```



74% of the tracks in the store have been purchased while 26% are unpurchased.

Case 7

We want to fnd out if the range of tracks that we have in our store is reflective of their sales popularity.

```
query7 <- "WITH pop_genre AS (
SELECT g.name genre,
       g.genre_id genre_id,
       COUNT(t.track_id) no_of_tracks
  FROM track t
  JOIN genre g ON g.genre_id = t.genre_id
 GROUP BY 1
 ORDER BY 2 DESC
)
```

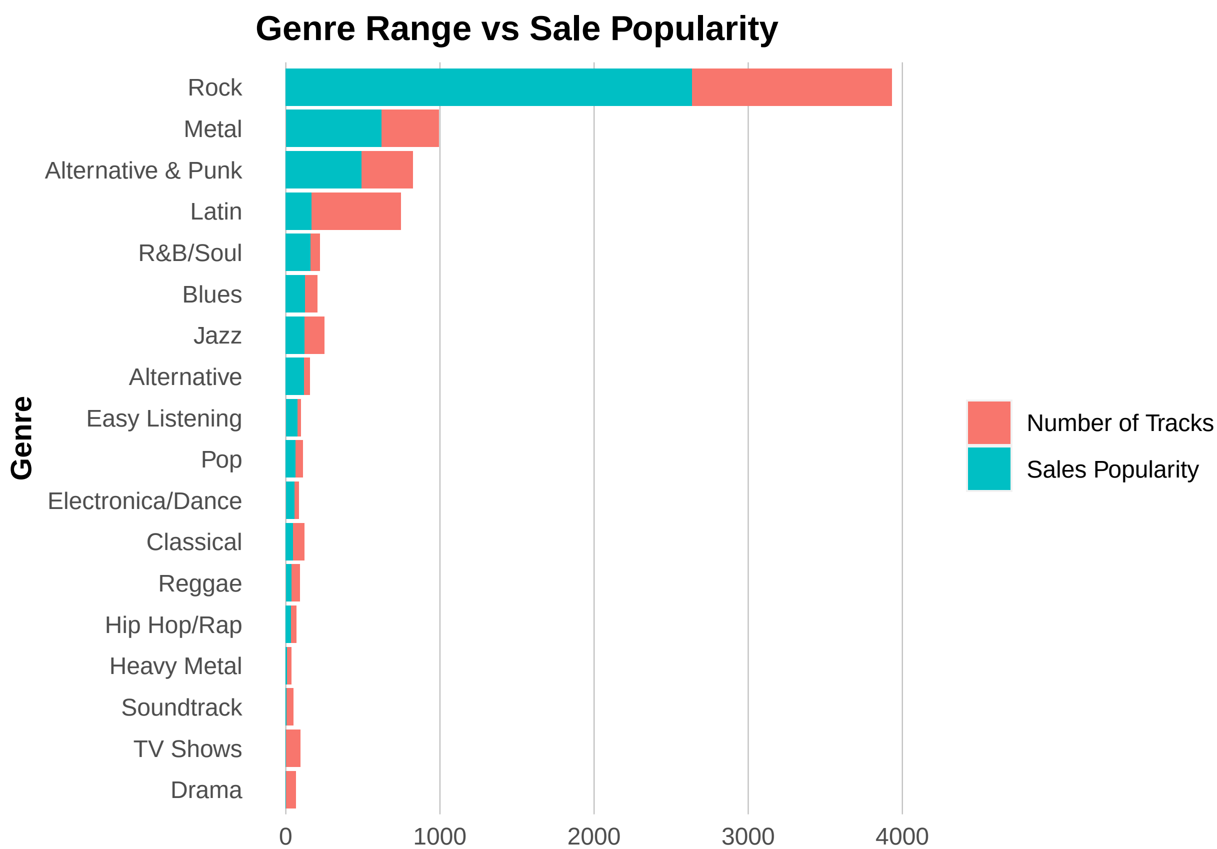
```
SELECT pg.genre genre,
       pg.no_of_tracks no_of_tracks,
       COUNT(il.track_id) tracks_sold
  FROM invoice_line il
  JOIN track t ON il.track_id = t.track_id
  JOIN pop_genre pg ON t.genre_id = pg.genre_id
 GROUP BY 1
 ORDER BY 3 DESC, 2"
```

```
track_range_sales <- run_query(query7)
track_range_sales %>% render_table()
```

genre	no_of_tracks	tracks_sold
Rock	1297	2635
Metal	374	619
Alternative & Punk	332	492
Latin	579	167
R&B/Soul	61	159
Blues	81	124
Jazz	130	121
Alternative	40	117
Easy Listening	24	74
Pop	48	63
Electronica/Dance	30	55
Classical	74	47
Reggae	58	35
Hip Hop/Rap	35	33
Heavy Metal	28	8
Soundtrack	43	5
TV Shows	93	2
Drama	64	1

```
track_range_sales %>% mutate(
  genre = factor(genre, levels = genre)
) %>%  pivot_longer(
  cols = c("no_of_tracks", "tracks_sold")
) %>% ggplot(
  aes(x=value, y=genre, fill=name)
) +
  geom_bar(stat ="identity", position = "stack") +
  labs(title = "Genre Range vs Sale Popularity",
       y = "Genre") +
  scale_y_discrete(limits = rev) +
  scale_fill_discrete(labels = c("Number of Tracks", "Sales Popularity")
  ) +

  theme(
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.ticks = element_blank(),
    axis.title.x = element_blank(),
    legend.title  = element_blank(),
```

```r
    panel.background = element_blank(),
    panel.grid.major.x = element_line(colour = "gray", linewidth = 0.2)
  )
```



For a lot of the common genre available in the store, they are reflective of their sales popularity. The genres that sell the most are the genres with the most tracks in the store.

Case 8

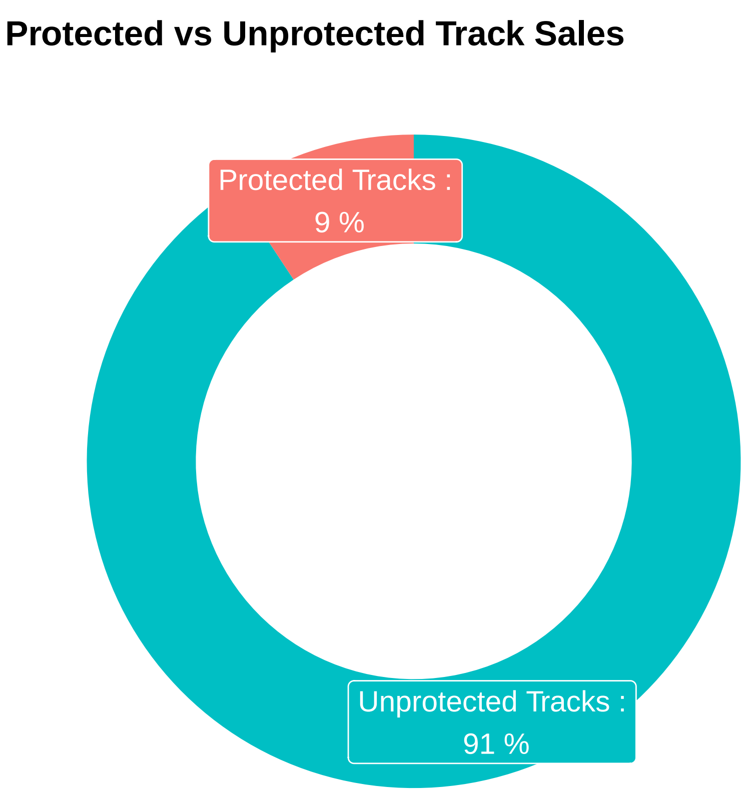
Does a track being protected or unprotected drive sales of track?

```r
query8 <- "SELECT CASE WHEN mt.name LIKE '%Protected%'
                  THEN 'protected'
                  ELSE 'unprotected'
                  END AS media_type,
                  COUNT(il.track_id) tracks_sold
             FROM invoice_line il
             JOIN track t ON t.track_id = il.track_id
             JOIN media_type mt ON mt.media_type_id = t.media_type_id
            GROUP BY 1
            ORDER BY 2 DESC ;"
```

```r
protected_v_uprotected <- run_query(query8)
protected_v_uprotected %>% render_table()
```

media_type	tracks_sold
unprotected	4315
protected	442

```
protected_v_uprotected %>% mutate(
   percentage = tracks_sold / sum(tracks_sold)
 ) %>% mutate(
   ymax = cumsum(percentage),
   ymin = c(0, percentage[1])
 ) %>% mutate(
  label = c("Unprotected Tracks", "Protected Tracks")
  ) %>% ggplot(
   aes(ymax = ymax, ymin = ymin, xmax=4, xmin=3, fill = media_type)
 ) +
   geom_rect() +
   labs(title = "Protected vs Unprotected Track Sales") +
   coord_polar("y") +
   geom_label(x=3.5, aes(y = (ymax + ymin)/2, label = paste(label, ":\n", round(percentage * 100), "%")
   xlim(c(1, 4)) +
   theme_void() +
   theme( plot.title = element_text(face = "bold"),
          legend.position = "none"
   )
```



Only 9% of the tracks sold are protected, so protected tracks have no influence on the track sales.

Conclusion

We simulated the role of a data analyst to answer business question querying a SQLite database from R. This has enabled us to be able to visualise the results of our queries using ggplot. There other R packages such as RMySQL and RPostgreSQL which can be used to interact with other SQL datbases besides SQLite.