The chinook DB

Answering Business Questions using SQL

Emmanuel Messori

08/09/2021

```
library(RSQLite)
library(DBI)
source("functions.R")
```

Tables

```
show_tables()
```

```
##              name  type
## 1           album table
## 2          artist table
## 3        customer table
## 4        employee table
## 5           genre table
## 6         invoice table
## 7    invoice_line table
## 8      media_type table
## 9        playlist table
## 10 playlist_track table
## 11          track table
## 12         n_cust  view
```

New records

The Chinook record store has just signed a deal with a new record label, and you're in charge of choosing the first three albums to be added to the store. There are four albums to choose from, and all four are by artists who don't have any tracks in the store right now. Below is the list of artist names and the genre of music they produce:

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

To aid in selecting albums, we're interested in finding out which genres sell the best in the USA.

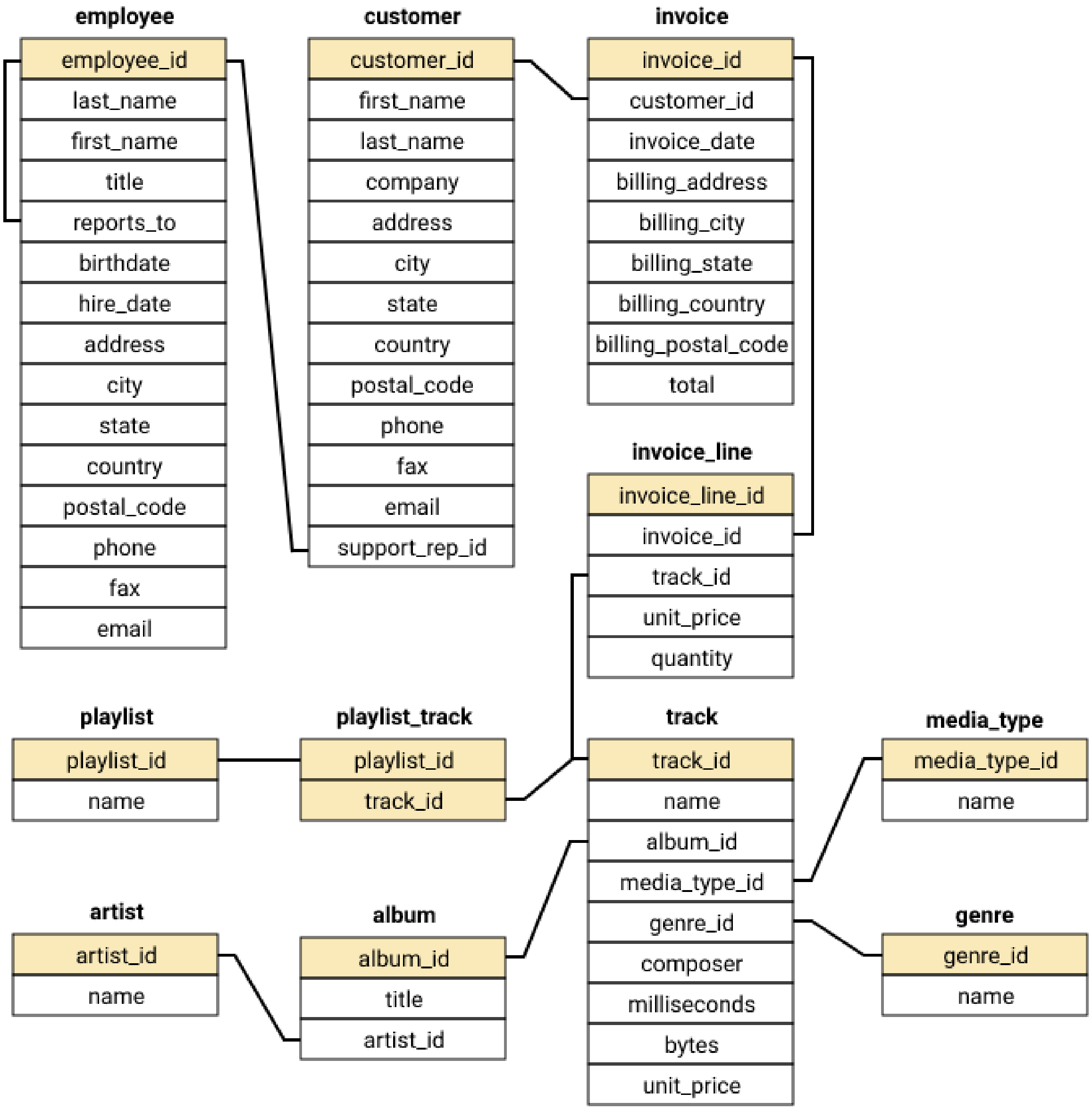
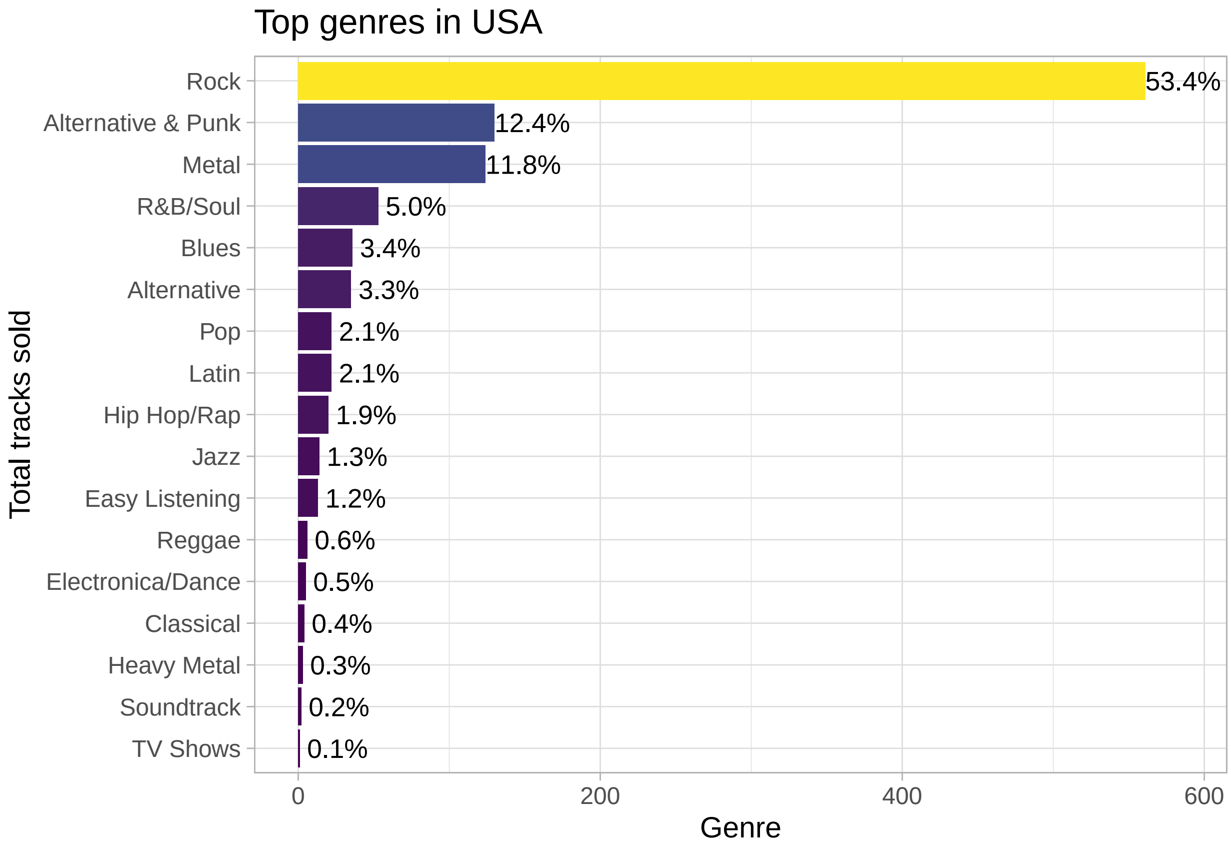


Figure 1: The database relational schema

```
query <- "WITH usa_sold AS (
SELECT il.*
FROM invoice_line il
INNER JOIN invoice i USING(invoice_id)
INNER JOIN customer c USING(customer_id)
WHERE c.country = \"USA\"
)
SELECT g.name as genre,
       COUNT(invoice_line_id) as total,
       CAST(SUM(quantity) AS FLOAT) / (
       SELECT SUM(quantity) FROM usa_sold
       ) perc_sold
FROM usa_sold
INNER JOIN track USING(track_id)
INNER JOIN genre g USING(genre_id)
GROUP BY genre
ORDER BY total DESC"
usa_genres_most_sold <- run_query(query)
usa_genres_most_sold
```

```
##                 genre total    perc_sold
## 1                Rock   561 0.5337773549
## 2  Alternative & Punk   130 0.1236917222
## 3               Metal   124 0.1179828735
## 4            R&B/Soul    53 0.0504281637
## 5               Blues    36 0.0342530923
## 6         Alternative    35 0.0333016175
## 7                 Pop    22 0.0209324453
## 8               Latin    22 0.0209324453
## 9         Hip Hop/Rap    20 0.0190294957
## 10               Jazz    14 0.0133206470
## 11     Easy Listening    13 0.0123691722
## 12             Reggae     6 0.0057088487
## 13  Electronica/Dance     5 0.0047573739
## 14          Classical     4 0.0038058991
## 15        Heavy Metal     3 0.0028544244
## 16         Soundtrack     2 0.0019029496
## 17           TV Shows     1 0.0009514748
```

```
library(tidyverse)
library(scales)
theme_set(theme_light())
ggplot(usa_genres_most_sold, aes(fct_reorder(genre,
    total), total)) + geom_col(aes(fill = total),
    show.legend = FALSE) + coord_flip() +
    labs(title = "Top genres in USA", x = "Total tracks sold",
        y = "Genre") + geom_text(aes(label = sprintf("%1.1f%%",
    round(perc_sold, 4) * 100)), size = 3.5,
    nudge_y = 25) + scale_fill_viridis_c()
```



According to the barplot, Rock is by far the highest selling genre in the database, followed by Alternative & Punk and Metal. Among the recordings proposed by the record label, we have four genres: Punk, Hip-Hop, Blues and Pop. Following the chart, we would then exclude the Hip-Hop artist.

Sales support performance

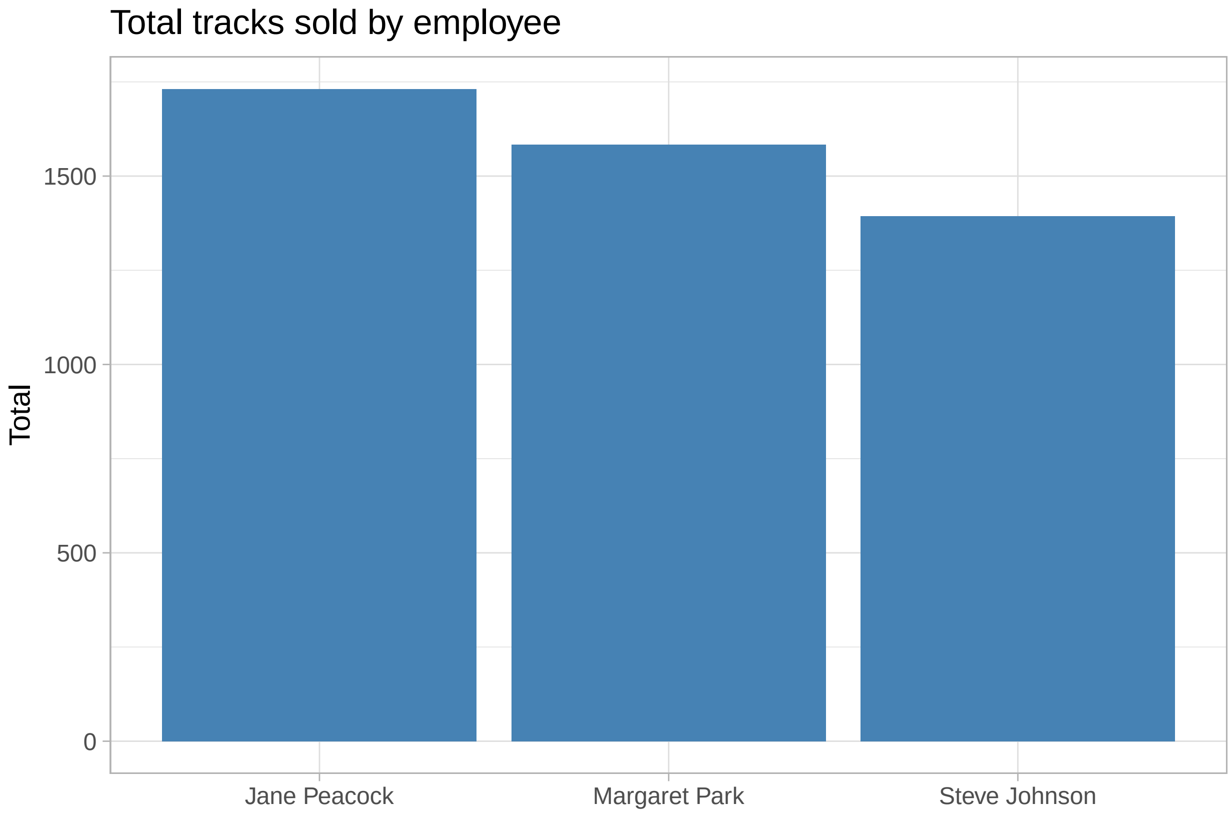
Each customer for the Chinook store gets assigned to a sales support agent within the company when they first make a purchase. You have been asked to analyze the purchases of customers belonging to each employee to see if any sales support agent is performing either better or worse than the others.

```
ss_performance <- "SELECT e.employee_id as id,

                          e.first_name || \" \" || e.last_name as name,
                          e.title as title,
                          e.hire_date,
                          COUNT (DISTINCT c.customer_id) as n_customers,
                          sum(i.total) AS total_sold,
                          sum(i.total)  / COUNT (DISTINCT (c.customer_id)) as tot_sold_per_cust

                   FROM employee e

                   INNER JOIN customer c ON employee_id = support_rep_id
                   INNER JOIN invoice i USING(customer_id)

                   GROUP BY id
```

```
                    ORDER BY total_sold DESC"

ss_perf <- run_query(ss_performance)
ss_perf
```

```
##   id          name               title           hire_date n_customers
## 1  3  Jane Peacock Sales Support Agent 2017-04-01 00:00:00          21
## 2  4 Margaret Park Sales Support Agent 2017-05-03 00:00:00          20
## 3  5 Steve Johnson Sales Support Agent 2017-10-17 00:00:00          18
##   total_sold tot_sold_per_cust
## 1    1731.51          82.45286
## 2    1584.00          79.20000
## 3    1393.92          77.44000
```

```
ggplot(ss_perf, aes(name, total_sold)) +
    geom_col(fill = "steelblue") + labs(title = "Total tracks sold by employee",
    x = "", y = "Total")
```



The three Sales Support have been hired at different times. That could explain the difference in the performance. Also each employee has a different number of customers assigned. If we divide the total sold by the number of customer we find the mean sold per customer. Of course to have a more precise view we could standardize the data and apply an ANOVA.

International sales data

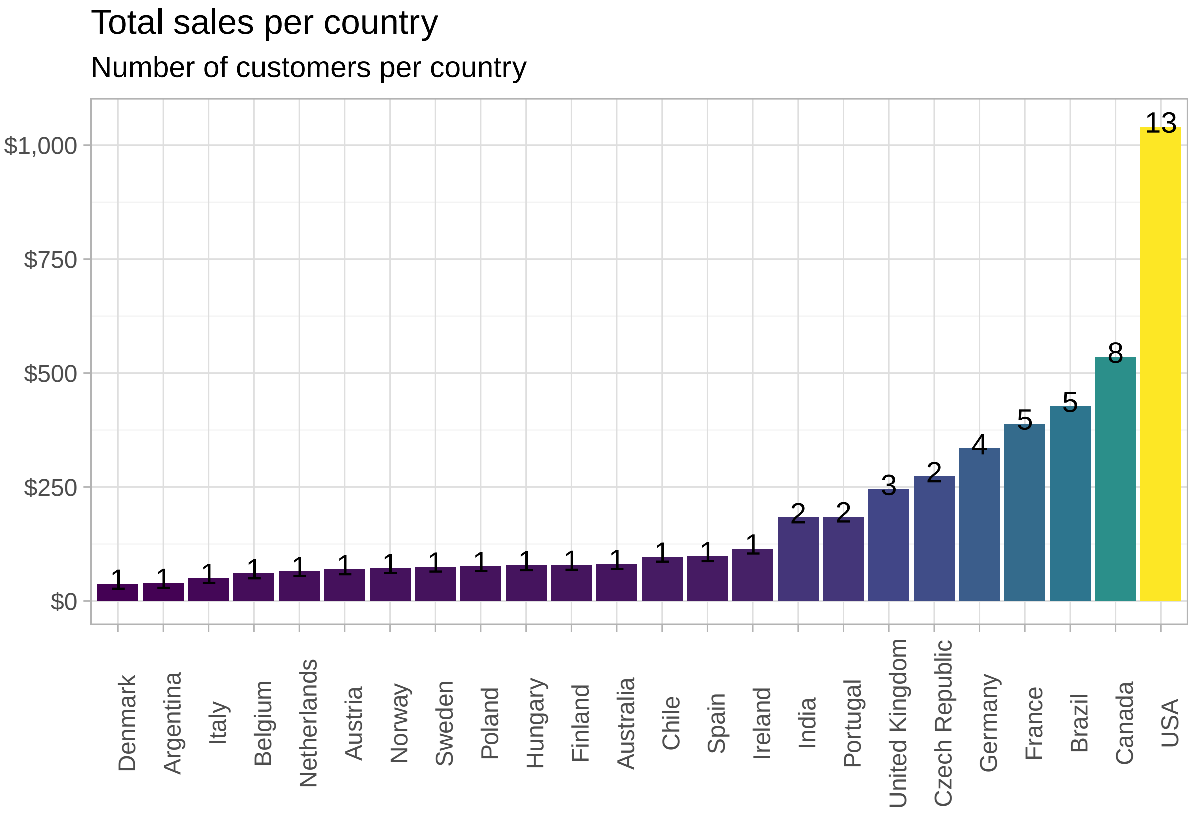
```
run_statement("DROP VIEW IF EXISTS n_cust")

view <- "CREATE VIEW n_cust AS
         SELECT country, COUNT(customer_id) as tot_cust FROM customer
         GROUP BY country"

country_sales <- "WITH sub AS
                  (
                  SELECT nc.*,
                  CASE
                      WHEN tot_cust > 1 THEN \"Normal\"
                      ELSE \"Other\"
                      END
                      AS c_group
                  FROM n_cust nc
                  )
                  SELECT sub.country,
                         sub.tot_cust,
                         SUM(i.total) as tot_sold,
                         SUM(i.total) / COUNT(DISTINCT(i.customer_id)) as tot_sales_per_cust,
                         SUM(i.total) / COUNT(invoice_id) as avg_invoice,
                         c_group
                  FROM customer c
                  INNER JOIN invoice i USING(customer_id)
                  INNER JOIN sub USING (country)
                  GROUP BY country
                  ORDER BY tot_sold DESC"


run_statement(view)
show_tables()
```

```
##              name  type
## 1           album table
## 2          artist table
## 3        customer table
## 4        employee table
## 5           genre table
## 6         invoice table
## 7    invoice_line table
## 8      media_type table
## 9        playlist table
## 10 playlist_track table
## 11          track table
## 12         n_cust  view
```

```
country_sales <- run_query(country_sales)
country_sales
```

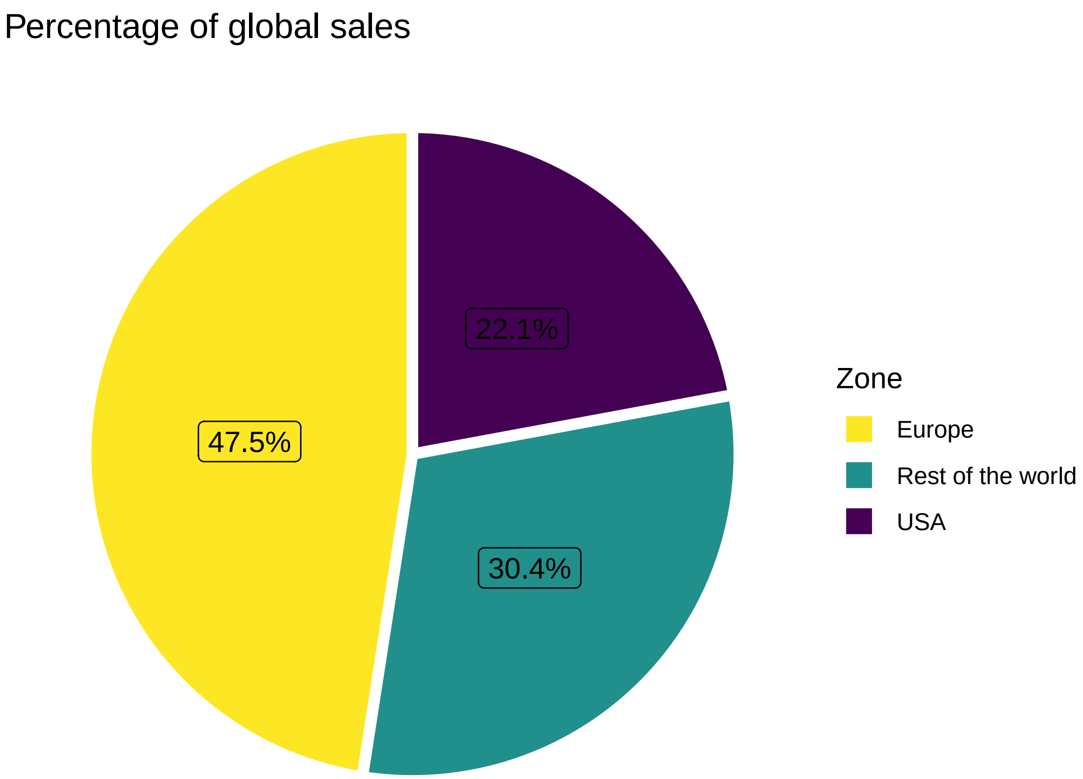
```
##            country tot_cust tot_sold tot_sales_per_cust avg_invoice c_group
## 1              USA       13  1040.49           80.03769    7.942672  Normal
## 2           Canada        8   535.59           66.94875    7.047237  Normal
## 3           Brazil        5   427.68           85.53600    7.011148  Normal
## 4           France        5   389.07           77.81400    7.781400  Normal
## 5          Germany        4   334.62           83.65500    8.161463  Normal
## 6  Czech Republic        2   273.24          136.62000    9.108000  Normal
## 7  United Kingdom        3   245.52           81.84000    8.768571  Normal
## 8         Portugal        2   185.13           92.56500    6.383793  Normal
## 9            India        2   183.15           91.57500    8.721429  Normal
## 10        Ireland        1   114.84          114.84000    8.833846   Other
## 11          Spain        1    98.01           98.01000    8.910000   Other
## 12          Chile        1    97.02           97.02000    7.463077   Other
## 13      Australia        1    81.18           81.18000    8.118000   Other
## 14        Finland        1    79.20           79.20000    7.200000   Other
## 15        Hungary        1    78.21           78.21000    7.821000   Other
## 16         Poland        1    76.23           76.23000    7.623000   Other
## 17         Sweden        1    75.24           75.24000    7.524000   Other
## 18         Norway        1    72.27           72.27000    8.030000   Other
## 19        Austria        1    69.30           69.30000    7.700000   Other
## 20    Netherlands        1    65.34           65.34000    6.534000   Other
## 21        Belgium        1    60.39           60.39000    8.627143   Other
## 22          Italy        1    50.49           50.49000    5.610000   Other
## 23      Argentina        1    39.60           39.60000    7.920000   Other
## 24        Denmark        1    37.62           37.62000    3.762000   Other
```

```
ggplot(country_sales) + geom_col(mapping = aes(reorder(country,
    tot_sold), tot_sold, fill = tot_sold),
    show.legend = FALSE) + scale_y_continuous(labels = label_dollar()) +
    theme(axis.text.x = element_text(angle = 90)) +
    scale_fill_viridis_c() + labs(title = "Total sales per country",
    subtitle = "Number of customers per country",
    y = "", x = "") + geom_text(aes(x = reorder(country,
    tot_sold), y = tot_sold, label = tot_cust),
    nudge_y = 10)
```

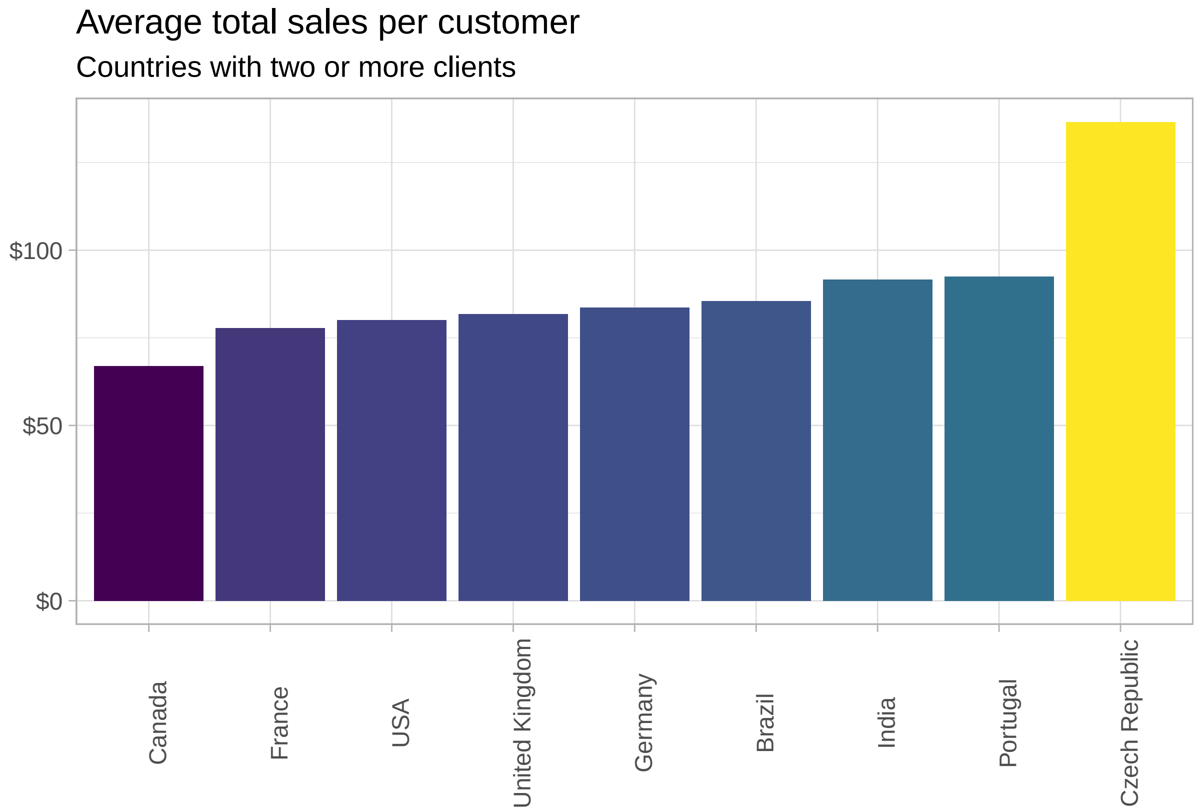


```
country_sales %>%
    mutate(Zone = case_when(country == "USA" ~
        "USA", country %in% c("Austria",
        "Belgium", "Czech Republic", "Denmark",
        "Finland", "France", "Germany", "Hungary",
        "Ireland", "Italy", "Netherland",
        "Norway", "Poland", "Portugal", "Spain",
        "Sweden", "United Kingdom") ~ "Europe",
        TRUE ~ "Rest of the world"), ) %>%
    group_by(Zone) %>%
    summarise(perc_sales = sum(tot_sold)) %>%
    ungroup() %>%
    mutate(perc_sales = round(perc_sales/sum(perc_sales) *
        100, 2)) -> perc_sales
```

```
perc_sales %>%
    ggplot(aes(x = 1, perc_sales, fill = Zone)) +
    geom_col(color = "white", size = 2) +
    coord_polar("y") + theme_void() + scale_fill_viridis_d(direction = -1) +
    geom_label(aes(label = sprintf("%1.1f%%",
        perc_sales)), position = position_stack(vjust = 0.5),
        show.legend = FALSE) + ggtitle("Percentage of global sales")
```



```
ggplot(filter(country_sales, c_group == "Normal"),
    aes(reorder(country, tot_sales_per_cust),
        tot_sales_per_cust, fill = tot_sales_per_cust)) +
    geom_col(show.legend = FALSE) + scale_fill_viridis_c() +
    theme(axis.text.x = element_text(angle = 90)) +
    scale_y_continuous(labels = label_dollar()) +
    labs(title = "Average total sales per customer",
        subtitle = "Countries with two or more clients",
        x = "", y = "")
```



USA is by far the most solid market, which constitutes almost 1/4 of the total sales. Anyway we have a lot of emerging tendencies from countries like Czech Republic, Portugal, India which, though have few users have higher sales per customer. These countries could be a good target for a promotional campaign. Canada is the second global market, which should be stimulated : the number of clients is good but they're mean expenses are quite low. In general, an European promotional campaign would be very beneficial.

Album purchases vs single tracks

The sales department is investigating whether is still profitable to buy and entire albums from record companies or to just focus on the potential most popular tracks.

```
albums_vs_tracks <- 'WITH invoice_first_track AS
    (
     SELECT
         il.invoice_id invoice_id,
         MIN(il.track_id) first_track_id
     FROM invoice_line il
     GROUP BY 1
    )
SELECT
    album_purchase,
    COUNT(invoice_id) number_of_invoices,
    CAST(count(invoice_id) AS FLOAT) / (
                                         SELECT COUNT(*) FROM invoice
                                      ) percent
```

```
FROM
    (
    SELECT
        ifs.*,
        CASE
            WHEN
                 (
                  SELECT t.track_id FROM track t
                  WHERE t.album_id = (
                                      SELECT t2.album_id FROM track t2
                                      WHERE t2.track_id = ifs.first_track_id
                                     )
                  EXCEPT
                  SELECT il2.track_id FROM invoice_line il2
                  WHERE il2.invoice_id = ifs.invoice_id
                 ) IS NULL
             AND
                 (
                  SELECT il2.track_id FROM invoice_line il2
                  WHERE il2.invoice_id = ifs.invoice_id
                  EXCEPT
                  SELECT t.track_id FROM track t
                  WHERE t.album_id = (
                                      SELECT t2.album_id FROM track t2
                                      WHERE t2.track_id = ifs.first_track_id
                                     )
                 ) IS NULL
             THEN "yes"
             ELSE "no"
         END AS "album_purchase"
     FROM invoice_first_track ifs
    )
GROUP BY album_purchase'

run_query(albums_vs_tracks)
```

```
##   album_purchase number_of_invoices   percent
## 1             no                500 0.8143322
## 2            yes                114 0.1856678
```

Album purchases account for almost a quarter of the total sales, so it is inadvisable to change strategy to just purchase the most popular tracks.