

Chinook Music Store

The chinook record store database can be taken as a blueprint for a digital music shop, containing details about artists, songs, albums, customers and purchases. It works on the principle of itunes, where it keeps an organized data about everything into eleven tables. It serves as a valuable resource for music fans and data enthusiasts, rendering insightful knowledge on how a music store operates digitally.

```
In [1]: import pandas as pd
import numpy as np
import sqlite3 as sql
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: db = 'chinook.db'

def run_query (q):
    with sql.connect(db) as conn:
        return pd.read_sql_query(q, conn)
```

Genre analysis for album selection:

Identifying Top-selling Genre in the USA

Chinook record store wants to select three albums to add from a list of four new artists, each belonging to different Genres. The Genres include Hip-Hop, Punk, Pop and Blues. The record label focuses on US artists and plans to advertise in the USA.

To decide which albums to choose, we need to find out which Genres are most popular in the USA. We'll do this by querying the database to see which Genres sell the most tracks in the USA. Based on this information. We'll make recommendations for the three artists whose album chinook should purchase.

```
In [3]: q = '''
WITH usa_tracks AS
  (SELECT t.genre_id AS genre_id, il.invoice_line_id
   FROM track AS t
   INNER JOIN invoice_line AS il ON il.track_id = t.track_id
   INNER JOIN invoice AS i ON i.invoice_id = il.invoice_id
   WHERE i.billing_country = 'USA'
  )
SELECT g.name AS genre_name,
       COUNT(usa.genre_id) AS num_purchases,
       ROUND((CAST(COUNT(usa.genre_id) AS float) / (SELECT COUNT(genre_id)
       FROM usa_tracks))*100,2) AS percentage_sold

FROM usa_tracks AS usa
INNER JOIN genre AS g ON g.genre_id = usa.genre_id
```

```
GROUP BY g.name
ORDER BY num_purchases DESC;'''
```

```
df = run_query(q)
df
```

Out[3]:

	genre_name	num_purchases	percentage_sold
0	Rock	561	53.38
1	Alternative & Punk	130	12.37
2	Metal	124	11.80
3	R&B/Soul	53	5.04
4	Blues	36	3.43
5	Alternative	35	3.33
6	Pop	22	2.09
7	Latin	22	2.09
8	Hip Hop/Rap	20	1.90
9	Jazz	14	1.33
10	Easy Listening	13	1.24
11	Reggae	6	0.57
12	Electronica/Dance	5	0.48
13	Classical	4	0.38
14	Heavy Metal	3	0.29
15	Soundtrack	2	0.19
16	TV Shows	1	0.10

In [4]: `import matplotlib.pyplot as plt`

```
df = df.sort_values(by='percentage_sold')
```

```
album_options = ['Hip Hop/Rap', 'Alternative & Punk', 'Blues', 'Pop']
cmap = df['genre_name'].apply(lambda x: '#0064AB' if x in album_options else '#BABABA')
```

```
fig, ax = plt.subplots(figsize=(8, 5))
bars = ax.barh(df['genre_name'], df["percentage_sold"], height=0.8, color=cmap, alpha=0.8)
```

```
for bar, percentage in zip(bars, df["percentage_sold"]):
    ax.text(bar.get_width() + 0.5, bar.get_y() + bar.get_height() / 2, f'{percentage}%',
            va='center', ha='left', fontsize=10, color='#C28499') # Change text color
```

```
#customizing and layout
```

```
ax.set_yticklabels(df['genre_name'], fontsize=10.5, color='#B0151B')
```

```
ax.set_xticks([])
```

```
plt.text(0.2, 1.07, 'Best Selling Genre In The USA', fontsize=20, fontweight='bold', color='red')
```

```

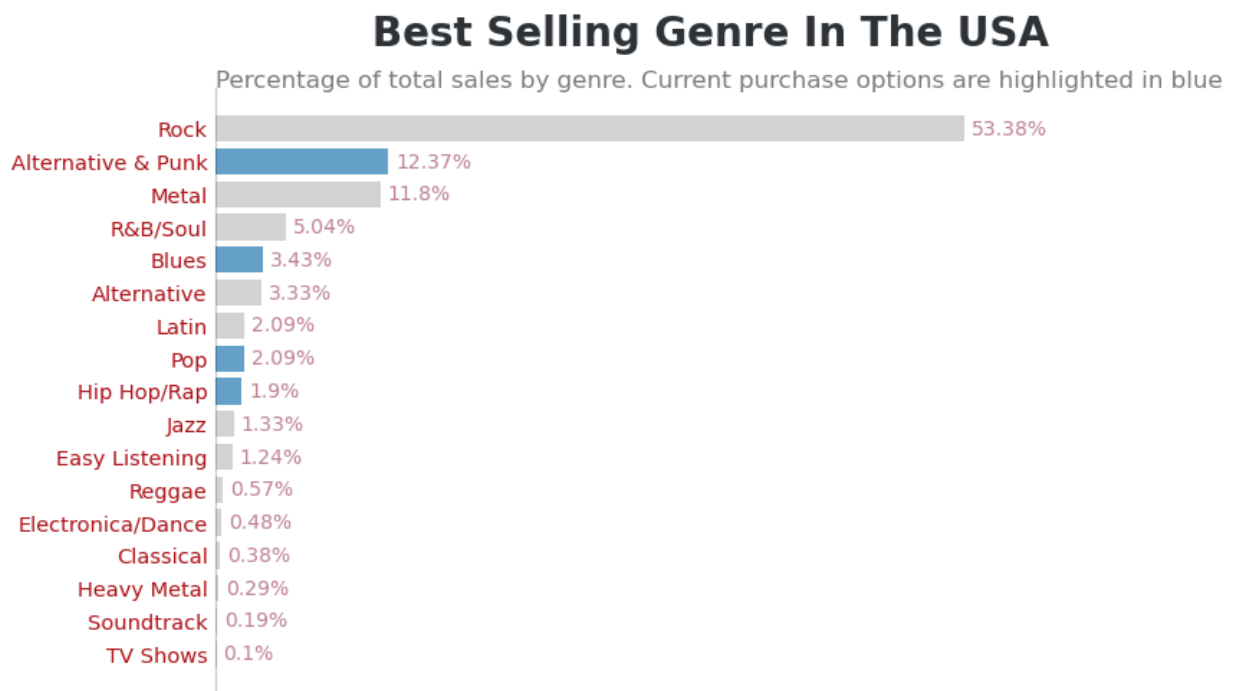
transform=plt.gca().transAxes)
plt.text(0.0, 1.00, 'Percentage of total sales by genre. Current purchase options are
        fontsize=12, color='grey', transform=plt.gca().transAxes)

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)

#adjusting left spines opacity
ax.spines['left'].set_color('#000000')
ax.spines['left'].set_alpha(0.2)
ax.tick_params(axis='y', which='both', length=0)
plt.tight_layout()
plt.show()

```

C:\Users\Admin\AppData\Local\Temp\ipykernel_17308\3526699267.py:19: UserWarning: FixedFormatter should only be used together with FixedLocator
ax.set_yticklabels(df['genre_name'], fontsize=10.5, color='#B0151B')



Result

The bar chart represents a distribution of music genre sales percentages in the USA. The "Rock" genre dominates at 53.38%. Interestingly, "Alternative & Punk" and "Metal" also have a significant share, with 12.37% and 11.8% respectively. They are followed by genres such as "R&B/Soul" and "Blues", holding smaller shares of the market at 5.04% and 3.43% respectively. The genres "Alternative," "Latin," and "Pop" each make up just over 2% of sales. "Hip Hop/Rap" is just behind at 1.9%. It is evident from the chart that the "Rock," "Alternative & Punk," and "Metal", highlighted in blue, are the genres which are good options for purchase.

Employee sales performance Analysis

The situation

After an initial purchase, each chinook customer is assigned to sales support agent. The company asked us to analyze purchases for customers belonging to each sales agent. They want to know if some agents are performing better or worse than others.

Analysis

To answer this question, we need to extract the names of the sales executives, their hire dates, the number of accounts they handle, and the total purchases made by these accounts. We will also compute the average sales for each account that the sales executives manage.

```
In [5]: q="""
        With t1 as (SELECT em.first_name || '|' || em.last_name AS sales_rep_name,
                           em.hire_date,
                           COUNT(cu.customer_id) AS num_invoices,
                           CAST(SUM(iv.total) AS Integer) AS total_sales
                           FROM employee em
                           JOIN customer cu
                           ON em.employee_id = cu.support_rep_id
                           JOIN invoice iv
                           ON iv.customer_id = cu.customer_id
                           GROUP BY 1
                           ORDER BY 4 DESC)
        SELECT *,
               Round(CAST(total_sales AS Float) / num_invoices, 2)
               AS sales_per_customer
        From t1;
        """
df = run_query(q)
df
```

```
Out[5]:
```

	sales_rep_name	hire_date	num_invoices	total_sales	sales_per_customer
0	JanePeacock	2017-04-01 00:00:00	212	1731	8.17
1	MargaretPark	2017-05-03 00:00:00	214	1584	7.40
2	SteveJohnson	2017-10-17 00:00:00	188	1393	7.41

```
In [6]: import matplotlib.pyplot as plt

#Data
sales_rep_name = df["sales_rep_name"].values
total_sales = df["total_sales"].values
sales_per_customer = df["sales_per_customer"].values

#create subplots
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 8))

#total sales chart
axes[0].bar(sales_rep_name, total_sales, color = "#F1C232", width=0.6)
axes[0].bar(sales_rep_name[0], total_sales[0], color = "#3D85C6", alpha=0.3, width=0.6)
#total sales chart label
```

```

axes[0].text(x=-0.1, y=2150, s="Total_sales", size=11, fontweight="bold", alpha=0.6, f

#total sales annotations
for sales,index in zip(total_sales,range(3)):
    axes[0].text(x=index,y=sales+100,s='${:,}''.format(sales),ha='center', \
        fontweight='bold', alpha=0.6, size=12)

#main chart title
axes[0].text(x=-0.2, y=2500, ha='center', s="Employee Sales Performance",size=16,fontw
alpha=0.8)

#average sales chart
axes[1].bar(sales_rep_name,sales_per_customer, color = "#9BDCED",width=0.6)
axes[1].bar(sales_rep_name[0],sales_per_customer[0],color = "#1D22A3", alpha=0.3, widt

#average sales chart Label
axes[1].text(x=-0.2, y=10,s="Average Sales",size=11, fontweight="bold",alpha=0.6, ha='

#average sales annotations
for sales,index in zip(sales_per_customer,range(3)):
    axes[1].text(x=index,y=sales+0.5,s='${:,.2f}''.format(sales),ha='center', \
        fontweight="bold",alpha=0.6, size=12)

#footnote
axes[1].text(x=-0.1, y=-3.5, s="Jane Joined in April,Margaret in May,Steve in september
size=11, alpha=0.9)

#common attributes to both charts
for x in axes:

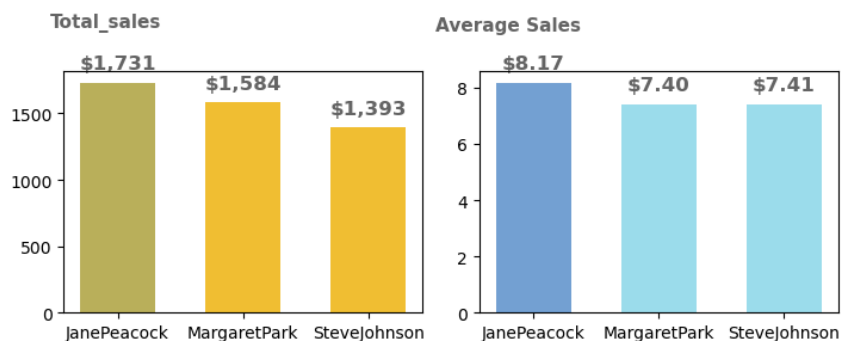
    ax.spines['right'].set_visible(False)
    ax.spines['left'].set_color('#DDD')
    ax.spines['left'].set_alpha(0.5)
    ax.spines['bottom'].set_color('#DDD')
    ax.spines['bottom'].set_alpha(0.5)

    ax.tick_params(left=False,bottom=False,labelsiz=10,labelcolor='grey')
    ax.grid(axis='y',linestyle='--',alpha=0.3)

plt.tight_layout(rect=[0,0.03,1,0.95])
plt.show()

```

Employee Sales Performance



Jane Joined in April,Margaret in May,Steve in september 2017.

Result

- Of the three sales employees, Jane achieved the highest total sales of 1,731 dollars. Margaret Park comes second, and Steve occupies the last place with 1,393 dollars. This difference in sales is understandable, considering that Jane and Margaret were employed about five months before Steve.
- Jane's average sales numbers support the leading position. She has achieved 8.17 dollars in sales per customer, the highest the three employees. Steve also performs marginally better than Margaret despite the difference in their employment dates.

Sales by Country

Scenario

Chinook wants to understand how sales are distributed across different countries. The company intends to identify countries with growth potential and may even run advertising campaigns in these countries.

Analysis

To answer this question, we will write a query that collates data on purchases from different countries. For each country, we will include that total number of customers, total sales value, average sales per customer, and the average order value. Where a country has only one customer, we will collect it into an "other" group:

```
In [7]: q = """
-- Collate the number of customers in each country
WITH t1 AS ( SELECT country, COUNT(customer_id) AS num_customers
              FROM customer
              GROUP BY country
            ),

-- Collate the total sales in each country
t2 AS ( SELECT cu.country,
              ROUND(SUM(iv.total), 2) AS total_sales,
              COUNT(iv.invoice_id) AS num_sales
        FROM
            customer cu
        JOIN
            invoice iv
        ON
            cu.customer_id = iv.customer_id
        GROUP BY
            cu.country
      ),

-- Group countries with only one customer as 'others'
t3 AS (
  SELECT
    CASE
      WHEN t1.num_customers = 1 THEN 'others'
      ELSE t1.country
    
```

```

        END AS countries,
        SUM(t1.num_customers) AS num_customers,
        SUM(t2.total_sales) AS total_sales,
        SUM(t2.num_sales) AS num_sales
    FROM t1
    JOIN t2
    ON t1.country = t2.country
    GROUP BY 1
),

-- Calculate relevant sales metrics
t4 AS (
    SELECT
        countries,
        num_customers,
        total_sales,
        ROUND(total_sales / num_sales, 2) AS avg_order_value,
        ROUND(total_sales / num_customers, 2) AS sales_per_customer,
        CASE
            WHEN countries = 'others' THEN 1
            ELSE 0
        END AS sort
    FROM
        t3
)

SELECT
    countries,
    num_customers,
    total_sales,
    avg_order_value,
    sales_per_customer
FROM
    t4
ORDER BY
    sort, num_customers DESC;
"""
df = run_query(q)
df

```

Out[7]:

	countries	num_customers	total_sales	avg_order_value	sales_per_customer
--	-----------	---------------	-------------	-----------------	--------------------

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

```
In [8]: avg_cust_purchase = df.sales_per_customer.mean()
print(avg_cust_purchase)
df['pcent_customers'] = round(100*df.num_customers/df.num_customers.sum(),1)
df['pcent_sales'] = round(100*df.total_sales/df.total_sales.sum(),1)
df['cust_purchases_diff'] = round(100*(df.sales_per_customer - avg_cust_purchase) /
                                avg_cust_purchase, 2)
```

86.96099999999998

In [9]: df

Out[9]:

	countries	num_customers	total_sales	avg_order_value	sales_per_customer	pcent_customers	pcent
--	-----------	---------------	-------------	-----------------	--------------------	-----------------	-------

0	USA	13	1040.49	7.94	80.04	22.0	
1	Canada	8	535.59	7.05	66.95	13.6	
2	Brazil	5	427.68	7.01	85.54	8.5	
3	France	5	389.07	7.78	77.81	8.5	
4	Germany	4	334.62	8.16	83.66	6.8	
5	United Kingdom	3	245.52	8.77	81.84	5.1	
6	Czech Republic	2	273.24	9.11	136.62	3.4	
7	India	2	183.15	8.72	91.58	3.4	
8	Portugal	2	185.13	6.38	92.57	3.4	
9	others	15	1094.94	7.45	73.00	25.4	

```
In [10]: import seaborn as sns
y_labs = df.countries.values
y_axes = np.arange(df.countries.size)

fig = plt.figure(figsize = (5, 6))
```



```

plt.barh(y_axes-0.3, df.pcent_customers, height = 0.2, color = '#3C8DF6')
plt.barh(y_axes-0.1, df.pcent_sales, height = 0.2, color = '#acadad')
plt.barh(y_axes+0.1, df.cust_purchases_diff, height = 0.2, color = '#FFFF00', alpha=0.8)

color_map = ['', '', '', '', '', '', 'yes', 'yes', 'yes', '']

#Annotate y-axis ticks
for loc, label, color in zip(y_axes, y_labs, color_map):
    if color == 'yes':
        plt.text(x=0.01, y=loc-0.25, s=label, ha='right', color='#0064AB', \
                alpha=0.9, size=10, fontweight='bold')
    else:
        plt.text(x=0.01, y=loc-0.25, s=label, ha='right', \
                alpha=1, size=10, fontweight='bold')

#set plot legend
plt.text(x=60, y=8.7, s='Customer Spending\nvs Average Customer Spending', color=
        alpha=1.0, size=10, fontweight='bold')
plt.text(x=60, y=8.3, s='Percentage Sales', color='#BABABA', alpha=1.0, size=10, fontweight='bold')
plt.text(x=60, y=7.9, s='Customer Percentage', color='#0064AB', size=10, fontweight='bold')

#set plot title
plt.text(x=30, y=11, s='Please Approve A Marketing Campaign In\nCzech Republic, I', \
        alpha=0.7, size=14, fontweight='bold')

for ax in fig.get_axes():
    plt.sca(ax)
    sns.despine(left=True, bottom=True, top=False)
    ax.tick_params(left=False, bottom=False, color='#ddd')
    ax.xaxis.set_ticks_position('top')
    ax.spines['top'].set_color('#DDD')
    plt.yticks([])
    plt.xticks([-20, 0, 20, 40, 60], ['-20', '0', '20', '40', '60'], \
            alpha=0.3, size=10, fontweight='bold')

```

Please Approve A Marketing Campaign In Czech Republic, Inida and Portugal



Result

The chart displays a comparative analysis of customer spending by country against average customer spending worldwide, percentage sales and customer percentage in various countries.

At the top, the United States stands out with the highest customer spending, significantly surpassing the average. This is accompanied by the highest percentage sales and customer percentage among the listed nations. Following the US, Canada exhibits substantial customer spending, with both percentage sales and customer percentage figures. Brazil, France and Germany exhibit comparable customer spending and percentage sales.

The point to be noted is that the Czech Republic shows customer spending and percentage sales which are just above the average, and a customer spending that is significantly above the global customer spending average. India and Portugal show similar trends, with both countries recording lower customer spending vs average customer spending than the Czech Republic.

Finally, the 'others' category, which likely shows popular markets with the values across all metrics comparable to those of USA or Canada.

How many tracks never sell?

In order to answer this question, we have to distinguish between entire inventory of tracks in track table and distinct tracks from invoice_line table.

```
In [11]: q = '''
WITH all_and_purchased AS (
    SELECT
        t.track_id AS all_tracks,
        il.track_id AS purchased_tracks
    FROM
        track AS t
    LEFT JOIN
        invoice_line AS il ON il.track_id = t.track_id
)
SELECT
    COUNT(DISTINCT a.all_tracks) total_tracks,
    COUNT(DISTINCT a.purchased_tracks) tracks_purchased,
    COUNT(DISTINCT a.all_tracks) - COUNT(DISTINCT a.purchased_tracks) not_purchased,
    ROUND(COUNT(DISTINCT a.purchased_tracks) / CAST(COUNT(DISTINCT a.all_tracks) AS float), 2) percent_purchased
FROM
    all_and_purchased AS a; '''

purchased = run_query(q)
purchased
```

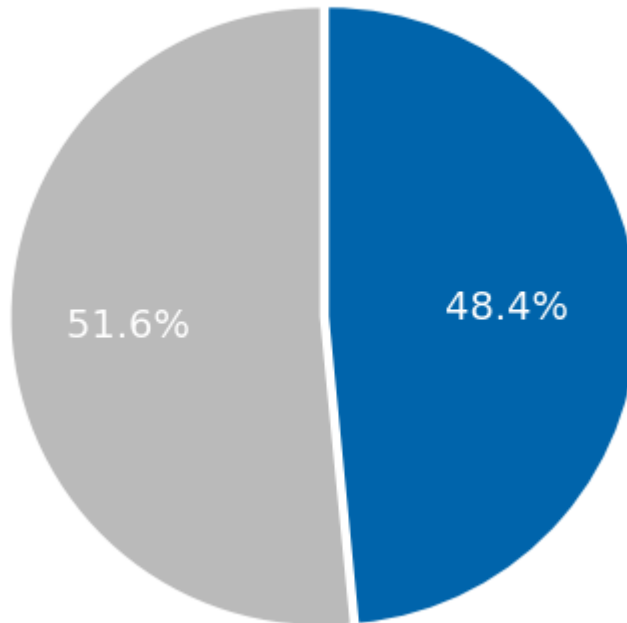
```
Out[11]:
```

	total_tracks	tracks_purchased	not_purchased	percent_purchased
0	3503	1806	1697	0.52

```
In [12]: purchased = purchased.T
purchased = purchased.iloc[[1,2],0 ]
purchased_list = purchased.tolist()
```

```
In [13]: plt.pie(purchased_list ,
    explode = (0,0.03),
    startangle = 90,
    autopct = '%1.1f%% ',
    textprops ={'fontsize': 14 , 'color': 'white'},
    colors = ('#BABABA' , '#0064AB'))
plt.title('TRACKS PURCHASED VS NOT PURCHASED', fontsize = 15 , color = 'black')
fig = plt.gcf()
fig.set_size_inches(5,5)
plt.show()
```

TRACKS PURCHASED VS NOT PURCHASED



Observation

The pie chart from my database analysis on Chinook presents a nearly even distribution between purchased and not purchased tracks, with a small majority of tracks being purchased (51.6%). This suggests that while over half of the music tracks offered by Chinook have been sold, a considerable portion (48.4%) remains unsold. Such a distribution prompts further investigation into the causes behind the lackluster performance of a significant part of the inventory. It may be beneficial to examine customer purchasing patterns, the genres of unsold tracks, or possibly the marketing efforts to determine how to better align Chinook's offerings with its customers' preferences. Additionally, this could be an opportunity to explore promotional strategies to increase the sales of the currently unsold tracks.

```
In [14]: q = '''
SELECT ar.name AS artist_name,
       g.name AS genre,
       COUNT(il.track_id) units_sold
FROM track t
LEFT JOIN invoice_line il ON il.track_id = t.track_id
INNER JOIN album al ON al.album_id = t.album_id
INNER JOIN artist ar ON ar.artist_id = al.artist_id
INNER JOIN genre AS g ON g.genre_id = t.genre_id
GROUP BY artist_name
HAVING units_sold = 0
ORDER BY units_sold;
'''
run_query(q)
```

Out[14]:

	artist_name	genre	units_sold
0	Aaron Copland & London Symphony Orchestra	Classical	0
1	Academy of St. Martin in the Fields Chamber En...	Classical	0
2	Academy of St. Martin in the Fields, John Birc...	Classical	0
3	Academy of St. Martin in the Fields, Sir Nevil...	Classical	0
4	Adrian Leaper & Doreen de Feis	Classical	0
...
69	The Office	TV Shows	0
70	The Tea Party	Alternative & Punk	0
71	Ton Koopman	Classical	0
72	Toquinho & Vinícius	Bossa Nova	0
73	Various Artists	Pop	0

74 rows × 3 columns

Observation

74 artist have not sold any units, with most of these tracks belonging to be classical music genre. Half of the company's inventory remains unsold, potentially tying up working capital without generating returns. Depending on the payment arrangement with record labels, there are two scenarios to consider:

- If chinook pays a fixed fee to host these tracks. It might be wise to focus on more popular genres and discontinue signing less successful artists.
- If chinook pays the record labels based on sales percentage, there is little downside to keeping the tracks in the store.

Regardless of the scenario, chinook should explore ways to promote these low-selling artists. Suggestions could be integrated into the purchasing process or displayed on the website's cart page to increase exposure.

Album vs Individual tracks

Chinook allows customers to purchase either a single track or an entire album. When customers purchase an album, they are charged the same price as if they had purchased single tracks individually.

In [15]:

```
q = '''
WITH invoice_data AS
(
  SELECT invoice_id , MIN(track_id) track_id
  FROM invoice_line
  GROUP BY 1
)
```

```

    ),
Album_purchased AS
(
    SELECT invoice_id ,
    CASE
        When
        (
            SELECT t2.track_id
            FROM track t1
            JOIN track t2
            ON
            t1.album_id = t2.album_id
            WHERE t1.track_id = invd.track_id

            EXCEPT

            SELECT il.track_id
            FROM invoice_line il
            WHERE il.invoice_id = invd.invoice_id

        )IS NULL

    AND
    (
        SELECT il.track_id
        FROM invoice_line il
        WHERE il.invoice_id = invd.invoice_id

        EXCEPT

        SELECT t2.track_id
        FROM track t1
        JOIN track t2
        ON
        t1.album_id = t2.album_id
        WHERE t1.track_id = invd.track_id

    )IS NULL

    then 'YES'
    ELSE 'NO'
    END AS Purchased_Album
    FROM invoice_data invd

)
SELECT Purchased_Album,
       COUNT(invoice_id) AS no_of_invoices,
       CAST(COUNT(invoice_id)AS float)*100/
       (SELECT COUNT(*)FROM Album_Purchased)AS percent
FROM Album_Purchased
GROUP BY 1
'''

run_query(q)

```

Out[15]:	Purchased_Album	no_of_invoices	percent
	0	NO	500 81.433225
	1	YES	114 18.566775

Result

Out of the total invoices recorded, 81.43% or 500 invoices were for transactions where no album was purchased. In contrast, 18.57% or 114 invoices were for transactions that included an album purchase. This data suggests that individual track purchases or other items are more common than album purchases among the invoices sampled.

Conclusion

The analysis indicates a predominant customer preference for purchasing single items, with only 18.57% of invoices including album purchases. This suggests that customers are more selective and possibly only interested in specific tracks rather than entire albums.

Recommendations

Based on these findings, the following strategies could be beneficial:

1. Selection of new albums based on Genre:

The above analysis suggests that chinook should focus more towards albums which are more popular among customers, rather than focusing on the less popular albums.

2. Employees' performance:

Jane Peacock came out as the top performer among the sales representatives. While Steve Johnson appears to have lower sales. It should be noted that Steve joined the team later, so he may require some support or training to deliver his maximum potential.

3. Modify Marketing Strategies based on sales analysis by country:

Marketing needs to be strategized based on sales in individual countries. As customers from countries like Czech Republic, India and Portugal spend more than average. Targeted marketing could render more potential customers and increased profit subsequently.

4. Purchasing practice of customer:

Chinook's plan to focus solely on single tracks may seem appealing, but it risks alienating customers who prefer to purchase entire album. It should engage with customers through surveys and gathering feedback to understand customer preferences.

