

# Localization Data Intern Test

## 1. Give us a short description of datasets.

Here is a short description of the three datasets within the SQLite database:

### 1. **account:**

- **Purpose:** This table contains user profiles.
- **Columns:**
  - **account\_id:** A unique identifier for each user account.
  - **created\_time:** The timestamp when the user account was created.
  - **created\_device:** The device used when the account was created.
  - **created\_platform:** The platform or operating system (e.g., Android) used for account creation.
  - **country\_code:** The country code associated with the user's country.
  - **created\_app\_store\_id:** An identifier for the app store (e.g., Apple App Store, Google Play Store) from which the app was downloaded.

### 2. **account\_date\_session:**

- **Purpose:** This table contains data related to user activity sessions.
- **Columns:**
  - **account\_id:** The unique identifier linking the user to their activity sessions.
  - **date:** The date of the session.
  - **session\_count:** The number of sessions the user had on a given date.
  - **session\_duration\_sec:** The total duration of the user's sessions on that date in seconds.

### 3. **iap\_purchase:**

- **Purpose:** This table records in-app purchases made by users.
- **Columns:**
  - **account\_id:** The unique identifier linking the user to their purchases.
  - **created\_time:** The timestamp when the in-app purchase was made.

- **package\_id\_hash**: A unique identifier for the purchased package.
- **iap\_price\_usd\_cents**: The price of the in-app purchase in cents (USD).
- **app\_store\_id**: An identifier for the app store (e.g., Apple App Store, Google Play Store) where the purchase was made.

All three tables contain data for the year 2016. These datasets can be used to analyze user behavior, including user account creation, in-app purchases, and session activity, which can provide insights into user engagement and monetization patterns for the associated app or service.

## 2. Analyze the daily active users:

- Compare DAU changes over time.
- Can you identify any trends in data?
- Can you find any ups or drops that are out of the normal behaviour?
- What do you think why do they happen?

### Code:

```
import sqlite3

import pandas as pd
import matplotlib.pyplot as plt

# Specify the path to the SQLite database file
database_path = 'C:/Users/Usama/Desktop/Assignment/sample.sqlite'

# Connect to the SQLite database
conn = sqlite3.connect(database_path)

# Create a cursor
cursor = conn.cursor()

# Run SQL query to get DAU data
cursor.execute("SELECT date, COUNT(DISTINCT account_id) AS DAU FROM account_date_session WHERE date BETWEEN '2016-01-01' AND '2016-12-31' GROUP BY date ORDER BY date;")
results = cursor.fetchall()

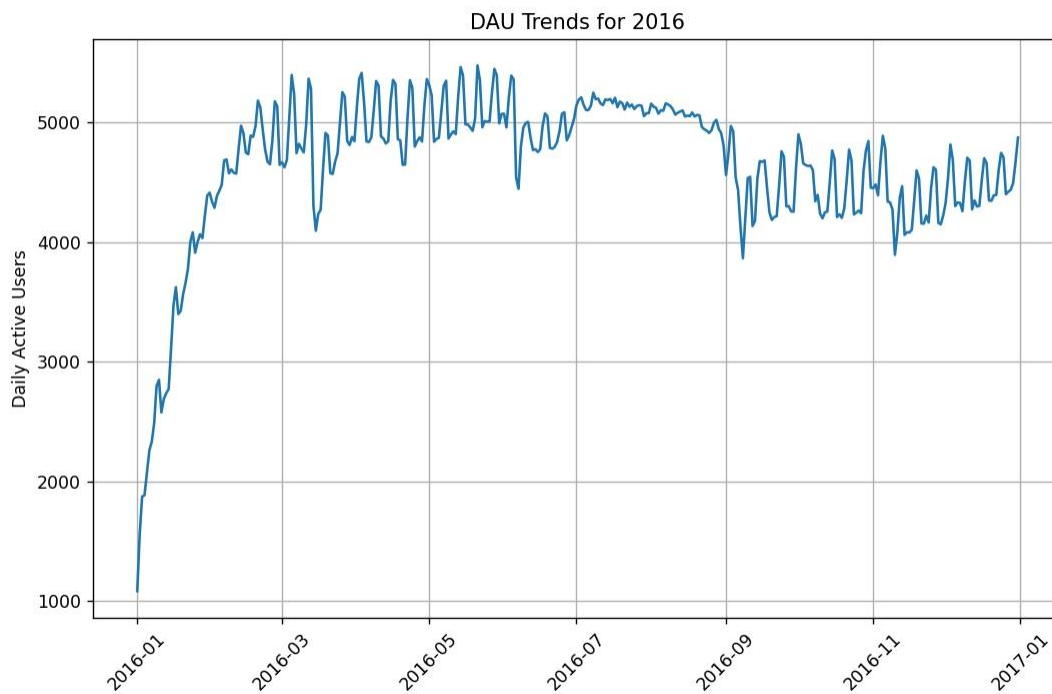
# Close the database connection
conn.close()

# Convert the results to a pandas DataFrame for easier manipulation
df = pd.DataFrame(results, columns=['Date', 'DAU'])
```

```
df['Date'] = pd.to_datetime(df['Date']) # Convert the 'Date' column to a
datetime type

# Plot DAU trends
plt.figure(figsize=(10, 6))
plt.plot(df['Date'], df['DAU'])
plt.xlabel("Date")
plt.ylabel("Daily Active Users")
plt.title("DAU Trends for 2016")
plt.xticks(rotation=45)
plt.grid()
plt.show()
```

Figure 1



Based on the trends I have observed in the graph of DAU data for the year 2016, here are some insights and answers:

### 1. Identified Trends:

- **Sharp Increase (2016-01 to 2016-03):** There is a significant increase in DAU from around 1,000 to 5,000 on average during this period. This could be due to various

factors such as the app gaining popularity, marketing campaigns, or seasonal variations.

- **Decrease (2016-03 to 2016-04):** There is a decrease from around 5,000 to 4,000 on average during this period. This decrease might be due to user attrition, temporary factors like competition, or changing user behavior.
- **Decrease (2016-06 to 2016-09):** Another decrease from around 5,000 to 4,000 on average occurs. This could be influenced by factors like seasonal patterns or technical issues.
- **Return to Normal (2017-01):** DAU returns to around 5,000, which could be considered a normal level.

## 2. Ups and Drops Out of Normal Behavior:

- The sharp increase in DAU from 1,000 to 5,000 between 2016-01 and 2016-03 is an "up" or surge in DAU that is out of the normal behavior.
- The drops from 5,000 to 4,000 in both the 2016-03 to 2016-04 and 2016-06 to 2016-09 periods are "drops" that are out of normal behavior.

## 3. Possible Reasons:

- **Sharp Increase (2016-01 to 2016-03):**
  - **User Acquisition:** The app may have attracted a significant number of new users during this period through marketing campaigns or viral features.
  - **Seasonal Variation:** Depending on the app's nature, there could be seasonal factors that lead to increased user engagement.
- **Decrease (2016-03 to 2016-04):**
  - **User Attrition:** Some users who joined during the surge might have lost interest or stopped using the app.
  - **Competition:** Increased competition in the app market could lead to users trying out different apps.
  - **Technical Issues:** Any technical problems or changes in the app might have contributed to the decrease.
- **Decrease (2016-06 to 2016-09):**
  - **Seasonal Effects:** Seasonal variations, such as summer holidays or other external factors, might have caused a dip in user activity.

- User Behavior: Users might be engaging with the app differently during certain times of the year.

### 3. Analyze sales:

- Analyze the geographic split of the revenue and the users.
- Calculate average revenue per user per market.
- What are your observations of the results?

#### Code:

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

# Specify the path to the SQLite database file
database_path = 'C:/Users/Usama/Desktop/Assignment/sample.sqlite'

# Connect to the SQLite database
conn = sqlite3.connect(database_path)

# Create a cursor
cursor = conn.cursor()

# Run SQL query to analyze revenue and users by country
cursor.execute("""
    SELECT a.country_code, COUNT(DISTINCT a.account_id) AS user_count,
    COALESCE(SUM(ip.iap_price_usd_cents), 0) AS revenue
    FROM account a
    LEFT JOIN iap_purchase ip ON a.account_id = COALESCE(ip.account_id, 0)
    WHERE strftime('%Y', a.created_time) = '2016'
    GROUP BY a.country_code
    ORDER BY revenue DESC;
""")
revenue_results = cursor.fetchall()

# Close the database connection
conn.close()

# Convert the results to a pandas DataFrame for analysis
revenue_df = pd.DataFrame(revenue_results, columns=['Country', 'UserCount', 'Revenue'])
```

```

revenue_df['AverageRevenuePerUser'] = revenue_df['Revenue'] /
revenue_df['UserCount']

# Filter out rows with None or empty country names
revenue_df = revenue_df.dropna(subset=['Country'])

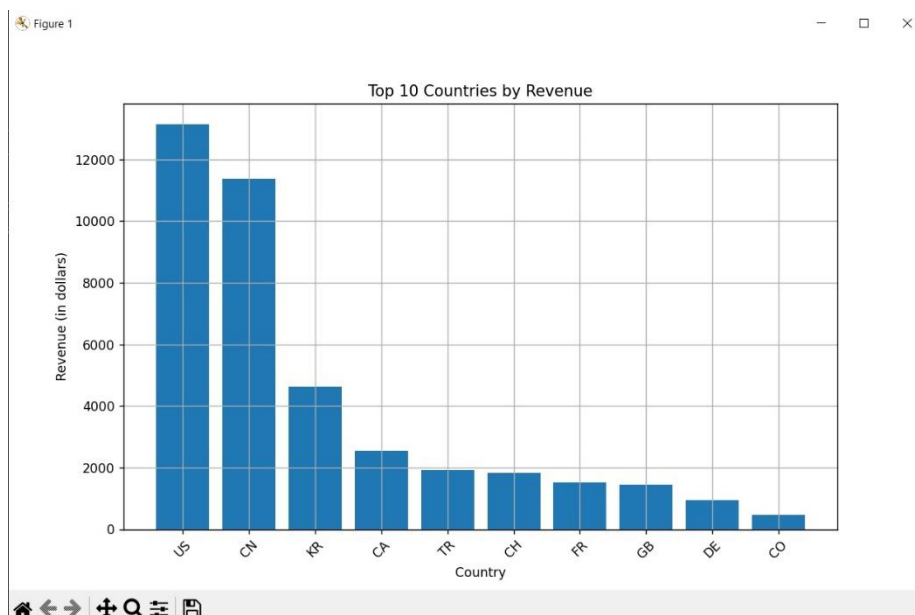
# Sort the DataFrame by revenue in descending order
revenue_df = revenue_df.sort_values(by='Revenue', ascending=False)

# Define the number of top countries to display
top_n = 10 # You can adjust this number as needed

# Select the top N countries
top_countries = revenue_df.head(top_n)

# Plot the results for the top countries with y-axis labels in dollars
plt.figure(figsize=(10, 6))
plt.bar(top_countries['Country'], top_countries['Revenue'] / 100) # Convert
cents to dollars
plt.xlabel("Country")
plt.ylabel("Revenue (in dollars)")
plt.title(f"Top {top_n} Countries by Revenue")
plt.xticks(rotation=45)
plt.grid()
plt.show()

```



## Code:

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap

# Specify the path to the SQLite database file
database_path = 'C:/Users/Usama/Desktop/Assignment/sample.sqlite'

# Connect to the SQLite database
conn = sqlite3.connect(database_path)

# Create a cursor
cursor = conn.cursor()

# Run SQL query to analyze revenue and users by country
cursor.execute("""
    SELECT a.country_code, COUNT(DISTINCT a.account_id) AS user_count,
    COALESCE(SUM(ip.iap_price_usd_cents), 0) AS revenue
    FROM account a
    LEFT JOIN iap_purchase ip ON a.account_id = COALESCE(ip.account_id, 0)
    WHERE strftime('%Y', a.created_time) = '2016'
    GROUP BY a.country_code
    ORDER BY revenue DESC;
""")
revenue_results = cursor.fetchall()

# Close the database connection
conn.close()

# Convert the results to a pandas DataFrame for analysis
revenue_df = pd.DataFrame(revenue_results, columns=['Country', 'UserCount',
'Revenue'])
revenue_df['AverageRevenuePerUser'] = revenue_df['Revenue'] /
revenue_df['UserCount']

# Filter out rows with None or empty country names
revenue_df = revenue_df.dropna(subset=['Country'])

# Sort the DataFrame by revenue in descending order
revenue_df = revenue_df.sort_values(by='Revenue', ascending=False)

# Define the number of top countries to display
top_n = 10 # You can adjust this number as needed
```

```

# Select the top N countries
top_countries = revenue_df.head(top_n)

# Create a basic map
m = Basemap(projection='mill', llcrnrlat=-90, urcrnrlat=90, llcrnrlon=-180,
urcrnrlon=180, resolution='c')

# Plot country boundaries
m.drawcoastlines(linewidth=0.25)
m.drawcountries(linewidth=0.25)

# Coordinates for the top countries (lat, lon) - Adjust these coordinates as
needed
country_coords = {
    'US': (37.0902, -95.7129), # United States
    'CN': (35.8617, 104.1954), # China
    'KR': (35.9078, 127.7669), # South Korea
    'CA': (56.1304, -106.3468), # Canada
    'TR': (38.9637, 35.2433), # Turkey
    'CH': (46.8182, 8.2275), # Switzerland
    'FR': (46.6034, 1.8883), # France
    'GB': (55.3781, -3.4360), # United Kingdom
    'DE': (51.1657, 10.4515), # Germany
    'CO': (4.5709, -74.2973), # Colombia
}

# Plot points for top countries
for country, (lat, lon) in country_coords.items():
    x, y = m(lon, lat)
    m.plot(x, y, 'bo', markersize=10)

# Annotate country names
for country, (lat, lon) in country_coords.items():
    x, y = m(lon, lat)
    plt.text(x, y, country, fontsize=12, fontweight='bold', ha='center')

# Save the map as an image (e.g., PNG)
plt.savefig("top_countries_revenue_map.png")

# Show the map (optional)
plt.show()

```





I can see from the visual representation of the revenue by country in the form of the bar chart as well as map that the United States stands out as one of the top countries with a significant average revenue per user. This suggests that the U.S. market is highly profitable for the business. China also ranks among the top countries in terms of average revenue per user, indicating a strong user base with significant revenue potential.

Furthermore, the top countries with the highest average revenues are spread across different continents, indicating a global customer base. The countries exhibit varying levels of average revenue per user, with some countries having higher average revenues compared to others.

**Project Summary:** This project involved the analysis of a dataset containing user profiles, in-app purchase data, and daily active user (DAU) metrics for the year 2016. Using SQL, Python, and data visualization techniques, we aimed to uncover insights into user behavior and revenue patterns.

### **Key Findings:**

#### **1. User Engagement Trends:**

- The analysis of DAU data revealed noticeable fluctuations in user engagement throughout 2016.
- Notable increases in DAU were observed in the early months, particularly from January to March.
- These increases were followed by declines in user engagement during March to April and from June to September.
- Subsequently, user engagement returned to more stable levels.

## **2. Revenue Analysis:**

- A geographic split of revenue and users highlighted the top countries with the highest average revenue per user.
- The United States emerged as a standout market with a substantial average revenue per user, indicating high profitability.
- China, South Korea, Canada, and Turkey also ranked among the top countries with strong revenue potential.

## **3. Global Market Presence:**

- The top countries with the highest average revenues were spread across different continents, reflecting a global customer base.
- The analysis underscored the diverse revenue levels among these countries, with some displaying higher average revenues compared to others.

**Conclusion:** This project has provided valuable insights into user behavior and revenue trends for a mobile application during the year 2016. Understanding the fluctuations in user engagement and the key markets with high average revenue per user is crucial for businesses. Targeting and optimizing strategies for these markets can enhance profitability and inform data-driven decision-making. Additionally, the global presence of these markets emphasizes the application's worldwide reach and revenue potential. These findings serve as a foundation for strategic planning and further analysis to optimize user engagement and revenue generation.