Stakeholder Identification:

The target audience consists of aspiring app developers seeking to make informed decisions during the app development process.

Data Relevance:

The data serves the purpose of providing data-driven insights to aid in selecting the most suitable app category for development.

Data Utilization:

The acquired data assists in determining popular app categories, establishing an optimal pricing strategy, and devising methods to enhance user ratings.

Key Recommendations:

1. Consider offering paid apps for potentially higher ratings compared to free apps.

2. Favor apps supporting a wide range of languages (10-30) for improved ratings.

3. Explore app development in genres such as finance and books, which show lower ratings but signify an unmet user need.

4. Acknowledge the significance of well-crafted app descriptions in achieving higher ratings.

5. Aim for a target rating above the app store's average of 3.5.

6. Acknowledge the competitive landscape of games and entertainment apps, while recognizing the substantial demand within these genres.

CODE:

CREATE TABLE appleStore\_description\_combined AS

SELECT \* FROM appleStore\_description1

UNION ALL

SELECT \* FROM appleStore\_description2

UNION ALL

SELECT \* FROM appleStore\_description3

UNION ALL

SELECT \* FROM appleStore\_description4

\*\*EXPLORATORY DATA ANALYSIS\*\*

--check the number of unique apps in both tables

--Comparing the unique IDs of both data sets to ensure a match

SELECT COUNT(DISTINCT id) AS UniqueAppIDs

FROM AppleStore

SELECT COUNT(DISTINCT id) AS UNIQUEAppIDs

FROM appleStore\_description\_combined

--Check for any missing values in key fields

SELECT COUNT(\*) AS MissingValues

FROM AppleStore

WHERE track\_name IS null OR user\_rating IS null OR prime\_genre IS null

SELECT COUNT(\*) AS MissingValues

FROM appleStore\_description\_combined

WHERE app\_desc IS null

--Find out the number of apps per genre

SELECT prime\_genre, COUNT(\*) AS NumApps

FROM AppleStore

GROUP BY prime\_genre

ORDER BY NumApps DESC

--Get an overview of the apps ratings

SELECT min(user\_rating) AS MinRating,

max(user\_rating) AS MaxRating,

avg(user\_rating) AS AvgRating

FROM AppleStore

\*\*DATA ANALYSIS

--Determine whether paid apps have higher ratings than free apps

SELECT CASE

WHEN price > 0 THEN 'Paid'

ELSE 'FREE'

END AS App\_Type,

avg(user\_rating) AS Avg\_Rating

FROM AppleStore

GROup by App\_Type

--Check if apps with more aupported languages have higher ratings

SELECt CASE

WHEN lang\_num < 10 THEN '<10 languages'

WHEN lang\_num BETWEEN 10 and 30 THEN '10-30 languages'

ELSE '>30 languages'

END AS language\_bucket,

avg(user\_rating) AS Avg\_Rating

FROM AppleStore

GROUP BY language\_bucket

ORDER BY Avg\_Rating DESC

--Check genres with low ratings

SELECT prime\_genre,

avg(user\_rating) AS Avg\_Rating

FROM AppleStore

Group by prime\_genre

Order by Avg\_Rating ASC

LIMIT 10

--Check if there is correlation between the length of the app descrption and the user rating

SELECT CASE

WHEN length(AppleStoreDescription.app\_desc) <500 THEN 'Short'

When length(AppleStoreDescription.app\_desc) BETWEEN 500 And 1000 THEN 'Medium'

ELSE 'Long'

END AS Description\_length,

avg(AppleStoreMain.user\_rating) AS average\_rating

FROM

AppleStore As AppleStoreMain

JOIN

appleStore\_description\_combined AS AppleStoreDescription

ON

AppleStoreMain.id = AppleStoreDescription.id

GROUP BY description\_length

Order BY average\_rating DESC

--Check the top-rated apps for each genre

SELECT

prime\_genre,

track\_name,

user\_rating

FROM (

SELECT

prime\_genre,

track\_name,

user\_rating,

RANK () OVER(PARTITION BY prime\_genre ORder BY user\_rating DESC, rating\_count\_tot DESC) AS rank

FROM AppleStore

) AS AppleRank

WHERE AppleRank.rank = 1