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 table AppleStore

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 fieldsAppleStore

Select count(\*) as MissingValues

From AppleStore

where track\_name is null or user\_rating is null or prime\_genre is null

Select count(distinct id) 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 apps' rating

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 appsAppleStore

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 supported 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 the genre 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 a correlation between the length of the app description and user ratingAppleStore

Select CASE

When length(b.app\_desc) <500 then 'Short'

When length(b.app\_desc) BETWEEN 500 and 1000 then 'Medium'

Else 'Long'

End as description\_length\_bucket,

avg(user\_rating) as Avg\_Rating

From

AppleStore AS A

JOIN

appleStore\_description\_combined as b

on

a.id = b.id

GROUP by description\_length\_bucket

order by Avg\_Rating Desc

--Check the top rated apps fpr each genreAppleStore

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 a

Where

a.rank = 1