**GOOGLE PLAY STORE SQL QUERIES**

**Creating a Database**

create database Google\_playstore\_db;

use Google\_playstore\_db;

**Data Cleaning and Processing**

select \*

from googleplaystore

where App is null

or Category is null

or Rating is null

or Reviews is null

or Size is null

or Installs is null

or Type is null

or Price is null

or Content\_Rating is null

or Genres is null

or Last\_Updated is null

or Current\_Ver is null

or Android\_Ver is null;

-- **Removing Null Values**

delete from googleplaystore

where App is null

or Category is null

or Rating is null

or Reviews is null

or Size is null

or Installs is null

or Type is null

or Price is null

or Content\_Rating is null

or Genres is null

or Last\_Updated is null

or Current\_Ver is null

or Android\_Ver is null;

**Overview of Dataset**

select \*

from googleplaystore;

**KPI’s**

-- **Total App**

select

count(distinct App) as Total\_Apps

from googleplaystore;

-- **Total Category**

select

count(distinct Category) as Total\_Categories

from googleplaystore;

**CHARTS**

-- **App Category Distribution**

select top 10 Category,

count(distinct App) as Total\_Apps

from googleplaystore

group by Category

order by Total\_Apps desc;

-- **Top-rated Free Apps**

select top 10 App,

Category,

Rating,

Reviews

from googleplaystore

where Type = 'Free' and Rating <> 'NaN'

order by Rating desc;

-- **Top Reviewed Apps**

select top 10 App,

max(try\_cast(Reviews as int)) as Total\_Review

from googleplaystore

group by App

order by Total\_Review desc;

-- **Average Rating by Category**

select top 10 Category,

AVG(try\_cast(Rating as float)) as Avg\_Rating

from googleplaystore

group by Category

order by Avg\_Rating desc;

-- **App Update Frequency Analysis**

select

App, year(try\_cast(Last\_Updated as date)) as Last\_Updated\_Year

from googleplaystore;

-- **Sentiment Spectrum Across Categories**

select top 10 Category,

avg(try\_cast(Sentiment\_Polarity as float)) as Avg\_Sentiment\_Polarity

from googleplaystore

join googleplaystore\_user\_reviews

on googleplaystore.App = googleplaystore\_user\_reviews.App

group by Category

order by Avg\_Sentiment\_Polarity desc;

-- **Sentiment Spread by Category**

select top 10 Category,

Sentiment,

count(\*) as total\_sentiment

from googleplaystore

join googleplaystore\_user\_reviews

on googleplaystore.App = googleplaystore\_user\_reviews.App

where Sentiment <> 'Nan'

group by Category, Sentiment

order by total\_sentiment desc;