```sql
create database RgAnalytics;


CREATE or REPLACE TABLE googleplaystore (
    App VARCHAR(194) NOT NULL,
    Category VARCHAR(19) NOT NULL,
    Rating DECIMAL(38, 9) NOT NULL,
    Reviews DECIMAL(38, 0) NOT NULL,
    Size VARCHAR(18) NOT NULL,
    Installs DECIMAL(38, 0) NOT NULL,
    Type VARCHAR(4) NOT NULL,
    Price DECIMAL(38, 2) NOT NULL,
    Content_Rating VARCHAR(15) NOT NULL,
    Android_Ver VARCHAR(18) NOT NULL
);

 SELECT * FROM googleplaystore;

---#1) Distinct Categories-------------------------------------------------
 Select count(Distinct(Category)) from googleplaystore;
 33 Categories;

 Select Distinct(Category) from googleplaystore;

---#2) Top 10 Installed Apps-----------------------------------------------

 Select app, installs from googleplaystore order by installs desc limit 10

---#3) Top 3 Most Popular Category-----------------------------------------------

select category, sum(installs) as total_inst from googleplaystore group by category
order by total_inst desc limit 3
   ---GAME, COMMUNICATION, TOOLS

select category, round(sum(installs)/1000000000,2) as Total_Installs_Billion from
googleplaystore
group by category order by Total_Installs_Billion desc;

---#4) Top 1/2 Apps per Category-------------------------------------------------

with cte as (
select app, category, installs, dense_rank() over (partition by category order by
installs desc) ranks
from googleplaystore)

select * from cte where ranks<=2;

---#5) Free apps vs Paid apps(%)-------------------------------------------------
select * from googleplaystore;

with cte as (
select
sum(case when type = 'Free' then 1 else 0 end) as FreeCount,
sum(case when type = 'Paid' then 1 else 0 end) as PaidCount from googleplaystore)

select
Concat(round((select freecount from cte)/count(*)*100,2), '%') as free_per,
Concat(round((select paidcount from cte)/count(*)*100,2), '%') as paid_per
from googleplaystore

---#7) Revenue generated by paid apps based on installation.(no. of installation *
Charges)----------------------------

select app, installs*price as revenue from googleplaystore where type='Paid' order by
revenue desc;

---#8) Lowest Rated Apps(installation
wise)-------------------------------------------------------------------------

```

```
select app, installs, rating from googleplaystore where rating=(select min(rating) from googleplaystore);

---#10) Content rating (waterfall/Funnel)----------
SELECT * FROM googleplaystore;

select Content_Rating, count(app) as app_count from googleplaystore group by 1 order by 2 desc ;

---#11) Overall avg Rating (Cards)----------
select round(avg(rating),2) from googleplaystore;

---#12) Avg app rating based on types(Free/Paid)----------
select type, round(avg(rating),2) avg_rating from googleplaystore group by 1;

---#13) Revenue generated acc to Content rating----------
select content_rating, sum(installs*price) as revenue from googleplaystore
where type= 'Paid' group by content_rating order by revenue desc;

---#14) Total Downloads Per Category (Top 10)----------
select category, sum(installs) as total_downloads from googleplaystore group by 1 order by total_downloads desc limit 10;

select category, count(*) as total_downloads from googleplaystore group by 1 order by total_downloads desc limit 10;

----#15) Top Expensive apps /(Category wise)----------

select category, app, price from (
select category, app, price,
dense_rank() over(partition by category order by price desc) as Ranks
from googleplaystore where type = 'Paid') a where ranks =1 order by price desc

-----#16) Top 10 Rated apps----------
SELECT * FROM googleplaystore;

select app, rating, installs from googleplaystore order by 2 desc limit 10;

-----#17) Type vs Rating wise----------
select * from
(select type, app, rating,
dense_rank() over(partition by type order by rating desc) ranks
from googleplaystore ) a where ranks=1

-----#18) Total revenue of paid apps----------
select sum(installs*price) from googleplaystore

----#19) Top Revenue created by apps /(Category wise)----------
with cte as
(select category, app, installs*price as revenue,
dense_rank() over(partition by category order by revenue desc) ranks
from googleplaystore where type = 'Paid')

select  category, app, revenue from cte where ranks =1 order by category
```

```
114
115  ----#20)Top 10 rated apps, which has more than 1M
     downloads----------------------------------------------------------------
116  select app, rating, installs from googleplaystore where installs > 1000000 order by 2
     desc,3 desc limit 10;
```