SQL Queries for Mobile Game Company Project

QUERY: 30-Day Retention & Growth Rate

```
select
    dayofjoin ,
    fractional_retention,
    pastfractionrate,
    round(safe_divide ((fractional_retention - pastfractionrate)/ pastfractionrate),4)
as  growth_rate
from  (
    select
        dayofjoin ,
        fractional_retention,
        lag(fractional_retention,1) over (order by dayofjoin ) as  pastfractionrate,
from(
Select
    joined as dayofjoin ,
    count (distinct(player_id)) as totalplayer,
    countif (retention_status= 1) as totalretained,
    round(((countif (retention_status= 1)) / count (distinct(player_id))),2) as
fractional_retention

  from(
   select
    p.player_id,
    joined,
     if ((max(day)>= joined + 30), 1, 0) as retention_status,

  from
    `my-first-project-329514.Project1.player_info` as p
    join `my-first-project-329514.Project1.matches_info` as m
    on
    p.player_id = m.player_id
    group by joined, p.player_id)

group by 1))
order by 1
```

QUERY: Average Purchase By Retained and Non-Retained Group

```
with  player_info_retention_stat as (
    select
        distinct  p.player_id,
        p.joined,
        if(max(day) over (partition by p.player_id) >= joined+30, 1, 0) as
retention_status,
    from `my-first-project-329514.Project1.player_info` p
    left join  `my-first-project-329514.Project1.matches_info` m
    ON p.player_id = m.player_id)
--calculate the amount spent by each player to determine average amount purchased by
reteined and non-retined players
select
    retention_status,
    round(avg(total_spent),2) as avg_spent
from (
    select
        distinct  pr.player_id,
        retention_status,
        sum(price) over (partition by   pr.player_id) as total_spent
    from  player_info_retention_stat as pr
    join `my-first-project-329514.Project1.purchase info` pi
    on pr.player_id = pi.player_id
    join `my-first-project-329514.Project1.item_info` i
    on pi.item_id = i.item_id)
group by retention_status
```