# Game Analysis 1 Write your MySQL query statement below

select player\_id, min(event\_date) first\_login from Activity group by player\_id

# department vs. company salary Write your MySQL query statement below

select department\_salary.pay\_month, department\_id

case

when department\_avg>company\_avg then 'higher'

when department\_avg<company\_avg then 'lower'

else 'same'

end as comparison from(

    select distinct date\_format(pay\_date,'%Y-%m') as pay\_month, department\_id, avg(amount)

    over(partition by date\_format('%Y-%m') as pay\_month,department\_id) as department\_avg from Salary s join Employee e on s.employee\_id=e.employee\_id) as department\_salary join

    (select distinct date\_format(pay\_date,'%Y-%m') as pay\_month, department\_id, avg(amount)

    over(partition by date\_format('%Y-%m') as pay\_month) as company\_avg from Salary) as company\_salary on department\_salary.pay\_month=company\_salary.pay\_month

# student report by geography Write your MySQL query statement below

select America, Asia, Europe from(

(SELECT @am := 0, @as := 0, @eu := 0) t,

(SELECT @as := @as + 1 as asid, name as Asia from student where continent = 'Asia' order by Asia) as t1 right join

(SELECT @am := @am + 1 as amid, name as America from student where continent = 'America' order by America) as t2 on t1.asid = t2.amid  left join

(SELECT @eu := @eu + 1 as euid, name as Europe from student where continent = 'Europe' order by Europe) as t3 on t3.euid = t2.amid)

# Report Contiguos Dates Write your MySQL query statement below

with interim as (select 'failed' as period\_state,fail\_date as considered\_date, rank() over(order\_by fail\_date) as rnk from Failed where year(fail\_date)=2019

union all

select 'succeeded' as period\_state,success\_date as considered\_date, rank() over(order\_by success\_date) as rnk from Succeeded where year(success\_date)=2019)

select period\_state,min(considered\_date) start\_date,max(considered\_date) end\_date from

(select \*,(rank() over(order by considered\_date))-rnk as group\_rank from interim) final group by period\_state, group\_rank order by 2