# MYSQL

## MYSQL II

* number of calls between two person

select from\_id as person1, to\_id as person2,count(duration) as call\_count,sum(duration) as total\_duration

from

(

select \* from Calls

union all

select to\_id, from\_id,duration from Calls

) as new\_table

where from\_id < to\_id

group by person1, person2;

* average selling price

select p.product\_id, round(sum(units\*price)/sum(units),2) as average\_price

from UnitsSold u

join Prices p

using(product\_id)

where datediff(u.purchase\_date, p.start\_date) >=0 and

datediff(u.purchase\_date, p.end\_date)<=0

group by 1;

* warehouse manager

SELECT name AS warehouse\_name, SUM(volume) AS volume

FROM

(SELECT name, w.product\_id ,Width\*Length\*Height\*units AS volume

FROM Warehouse w LEFT JOIN Products p

ON w.product\_id = p.product\_id) l

GROUP BY name

* apple & orrange

select a.sale\_date as sale\_date, a.sold\_num-o.sold\_num as diff

from

(select \* from Sales where fruit = 'apples') a

join

(select \* from Sales where fruit ='oranges') o

on a.sale\_date = o.sale\_date;

select s.sale\_date,

sum(if (s.fruit ='apples', s.sold\_num, -s.sold\_num)) as diff

from Sales s

group by s.sale\_date;

6/21

* monthly transaction

select date\_format(trans\_date,"%Y-%m") as month, country,

count(id) as trans\_count,

sum(case when state ='approved' then 1 else 0 end) as approved\_count,

sum(amount) as trans\_total\_amount,

sum(case when state ='approved' then amount else 0 end ) as approved\_total\_amount

from

Transactions as t

group by country, month(trans\_date);

* percentage of users attend

# Write your MySQL query statement below

select contest\_id, round(user\_count/count(user\_id.Users))as percentage

from

(select contest\_id, count(user\_id) as user\_count

from Register r

group by r.contest\_id) c

order by percentage desc, contest\_id asc;

* immediate food delivery

## MYSQL III

### Day1

* find the team size

# Write your MySQL query statement below

select country

from

(

select \*

from

(select sum(duration) as sum\_duration, count(\*) as sum\_calls, sum(duration) /count(\*) as avg\_duration, c.name as country

from

(select \*

from Calls

union all

select callee\_id, caller\_id, duration from Calls) as t1

left join person p on p.id = t1.caller\_id

left join country c on c.country\_code = left(p.phone\_number,3)

group by c.name) as t2

)

where avg\_duration >sum(sum\_duration)/sum(sum\_calls)

limit 5;

select gender, day,

sum(score\_points) over (partition by gender order by day) as total

from scores

group by gender,day

order by gender, day asc;

order by gender, day asc;

* countries you can safely invest in

# Write your MySQL query statement below

select country

from

(select sum(duration) as sum\_duration, count(\*) as sum\_calls, sum\_duration/sum\_calls as avg\_duration c.name as country

from

(select \*

from Calls

union all

select callee\_id, caller\_id, duration from Calls) t1

left join person p on p.id = t1.caller\_id

left join country c on c.country\_code = left(p.phone\_number,3)

group by c.name) t2

where t2.avg\_dutation > sum(sum\_duration)/sum(sum\_calls)

order by avg\_duration desc

limit 1;

select c.name as country from Country c

inner join Person p on c.country\_code=substr(p.phone\_number,1,3)

inner join Calls cl

on p.id in (cl.caller\_id,callee\_id)

group by c.name having avg(cl.duration)>(select avg(duration) from calls);

### Day2

* Project employees III

select project\_id, employee\_id

from

(select p.project\_id, p.employee\_id, e.experience\_years, max(e.experience\_years) over(partition by p.project\_id) as max\_year

from Project p

left join employee e

on p.employee\_id = e.employee\_id) t1

where t1.experience\_years = t1.max\_year;

* most recent order for each product

select product\_name, product\_id, order\_id,order\_date

from(

select p.product\_name, p.product\_id, o.order\_id, o.order\_date,

rank() over (partition by p.product\_id order by o.order\_date desc) as rn

from products p

inner join orders o on p.product\_id = o.product\_id

) t1

where t1.rn =1

order by t1.product\_name asc, t1.product\_id asc, t1.order\_id asc;

SELECT x.product\_name,x.product\_id,x.order\_id,x.order\_date

FROM(

SELECT p.product\_name,p.product\_id,o.order\_id,o.order\_date

,RANK() OVER (PARTITION BY p.product\_id ORDER BY o.order\_date DESC) as rn

FROM Orders o

INNER JOIN Products p ON o.product\_id=p.product\_id

)x

WHERE x.rn=1

ORDER BY x.product\_name,x.product\_id,x.order\_id

* find the start and end number of a continuous range

with cte as

(

select log\_id,

row\_number() over (order by log\_id) as row\_num

from Logs

)

select min(log\_id) as start\_id,

max(log\_id) as end\_id

from cte

group by log\_id - row\_num

* The Most Frequently Ordered Products for Each Customer

select a.customer\_id,a.product\_id,p.product\_name

from Products p inner join

(select customer\_id,product\_id,

RANK()OVER(partition by customer\_id ORDER BY count(product\_id) desc) as RNK

from Orders

group by customer\_id,product\_id)a

on a.product\_id=p.product\_id

where RNK=1;

### Day3 join

* Sellers With No Sales

select seller\_name

from seller s

where s.seller\_id not in

(select seller\_id from orders o where sale\_date like "2020%")

order by seller\_name asc;

* Biggest Single Number

select t1.num

from

(select num, count(\*) as count\_num

from MyNumbers n

group by n.num) t1

where t1.count\_num =1

order by t1.num desc

limit 1;

* Highest Grade For Each Student

select t1.student\_id, t1.course\_id, t1.grade

from

(select student\_id, course\_id, grade,

rank() over (partition by student\_id order by grade desc, course\_id asc) as rnk

from enrollments e) t1

where rnk = 1

order by t1.student\_id asc;

* Customers Who Bought Products A and B but Not C

with cte1 as (

select o.customer\_id

from Orders o

join (select customer\_id from Orders where product\_name = 'B') c on c.customer\_id=o.customer\_id

where o.product\_name = 'A'),

cte2 as (

select c1.customer\_id

from cte1 c1

where c1.customer\_id not in (select customer\_id from Orders where product\_name = 'C'))

select c.customer\_id, c.customer\_name

from Customers c

where c.customer\_id in (select customer\_id from cte2)

order by c.customer\_id

### Day4 join

* Evaluate Boolean Expression

# 