# 1.Pads

SELECT CONCAT(Name, '(',LEFT(Occupation,1),')') FROM OCCUPATIONS

ORDER BY Name ASC;

SELECT

CONCAT("There are a total of ", COUNT(Occupation), " ", LOWER(Occupation), "s.") FROM OCCUPATIONS

GROUP BY Occupation

ORDER BY Count(Occupation) ASC;

# 2.Occupations

SELECT

CASE WHEN OCCUPATION = 'Doctor' THEN NAME END AS DOCTOR,

CASE WHEN OCCUPATION = 'Professor' THEN NAME END AS PROF,

CASE WHEN OCCUPATION = 'Singer' THEN NAME END AS SINGER,

CASE WHEN OCCUPATION = 'Actor' THEN NAME END AS ACTOR,

ROW\_NUMBER() OVER(PARTITION BY OCCUPATION ORDER BY NAME) AS RN

FROM OCCUPATIONS)

SELECT MAX(DOCTOR), MAX(PROF), MAX(SINGER), MAX(ACTOR)

FROM CTE

GROUP BY RN;

# 3.Binary Tree Nodes

SELECT bs1.N, case when p is NULL then 'Root' when (select count(bs2.N) from bst bs2 where bs1.n in (select bs3.P from bst bs3)) != 0 then 'Inner' else 'Leaf' end from bst bs1 order by N asc;

SELECT e.COMPANY\_CODE , c.founder , count(DISTINCT e.lead\_manager\_code), count(DISTINCT e.senior\_manager\_code), count(DISTINCT e.manager\_code), count(DISTINCT e.employee\_code) FROM EMPLOYEE e JOIN Company c on e.company\_code=c.company\_code group by e.company\_code, c.founder order by e.company\_code

# 4.New Companies

SELECT e.COMPANY\_CODE , c.founder , count(DISTINCT e.lead\_manager\_code), count(DISTINCT e.senior\_manager\_code), count(DISTINCT e.manager\_code), count(DISTINCT e.employee\_code) FROM EMPLOYEE e JOIN Company c on e.company\_code=c.company\_code group by e.company\_code, c.founder order by e.company\_code

# 5.Weather Observation Station 18

SELECT ABS(ROUND((a-c)+(b-d),4))

FROM(

SELECT MIN(LAT\_N) as a,

MIN(LONG\_W) as b,

MAX(LAT\_N) as c,

MAX(LONG\_W) as d

FROM STATION

)sub

# 6.Weather Observation Station 19

SELECT ROUND(SQRT(POWER(MAX(LAT\_N)-MIN(LAT\_N),2)+(POWER(MAX(LONG\_W)-MIN(LONG\_W), 2))), 4)

FROM Station;

# 7.Weather Observation Station 20

with t1 as(select LAT\_N,

row\_number() over (order by LAT\_N) as rn

from station),

t2 as (select count(LAT\_N) as co

from station)

select distinct

case when mod(t2.co,2)!=0 then (select round(LAT\_N,4) from t1,t2 where rn=(t2.co+1)/2)

when mod(t2.co,2)=0 then (select round(avg(LAT\_N),4) from t1,t2 where rn=t2.co/2 or rn=t2.co/2+1)

end as median

from t1, t2;

# 8.The Report

SELECT

CASE WHEN G.Grade >=8 THEN S.Name ELSE NULL END,

G.Grade,

S.Marks

FROM Students AS S

LEFT JOIN Grades AS G

ON S.Marks BETWEEN G.Min\_Mark AND G.Max\_Mark

ORDER BY

G.Grade DESC,

IF(G.Grade>= 8, S.Name,NULL),

IF(G.Grade< 8, S.Marks,NULL) ASC;

# 9.Top Competitors

/\*

Enter your query here.

\*/SELECT hacker\_id,

name\_hacker

FROM (

SELECT COUNT(\*) AS challenge\_with\_full\_score,

hacker\_id,

name\_hacker

FROM (

SELECT submission\_id,

hacker\_id,

(SELECT

(SELECT

(SELECT score

FROM difficulty

WHERE challenges.difficulty\_level = difficulty.difficulty\_level)

FROM challenges

WHERE challenges.challenge\_id = submissions.challenge\_id))

AS total\_score\_challenge,

(SELECT name

FROM hackers

WHERE submissions.hacker\_id = hackers.hacker\_id)

AS name\_hacker,

score

FROM submissions

) AS table\_parsed

WHERE score = total\_score\_challenge

GROUP BY hacker\_id, name\_hacker

) AS table\_re\_parsed

WHERE challenge\_with\_full\_score > 1

ORDER BY challenge\_with\_full\_score DESC,

hacker\_id ASC

;

# 10.Ollivander's Inventory

W.id, WP.age, W.coins\_needed, W.power

from wands W join Wands\_Property WP on W.code = WP.code

where W.coins\_needed = (select min(W1.coins\_needed)

from wands W1 join Wands\_Property WP1

on W1.code = WP1.code

where W.power = W1.power and

WP.age = WP1.age and WP1.is\_evil = 0)

order by W.power desc, WP.age desc