**Identify duplicates**

drop table email\_id;

create table email\_id

(id int, email varchar(255));

insert into email\_id(id, email)

values (3, 'john@example.com'),

(2, 'bob@example.com'),

(1, 'john@example.com');

select \* from email\_id;

with cte as (select \*,

row\_number() over (partition by email

  order by id) as rn

from email\_id)

select distinct email from email\_id

where email in (select email from cte where rn>1);

**Delete duplicate rows**

with cte as (select \*,

row\_number() over (partition by email

  order by id) as rn

from email\_id)

delete from email\_id

where id in (select id from cte where rn>1);

**Check for consecutive numbers**

with cte as

(select id, num,

     lead(num) over (order by id) as lead1,

lead(num, 2) over (order by id) as lead2

from Logs)

select distinct num as "ConsecutiveNums" from Logs

where id in (select id from cte where num=lead1 and num=lead2);

**Select Nth highest salary**

CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT

BEGIN

DECLARE M INT;

SET M=N-1;

  RETURN (

    # Write your MySQL query statement below.

    SELECT DISTINCT Salary FROM Employee Group by Salary ORDER BY Salary DESC LIMIT M ,1

  );

END

**Select second highest salary**

Type 1

with cte as (select distinct "Salary",

dense\_rank() over (order by "Salary" desc) as "Rank"

from "Employee")

SELECT ISNULL ((select "Salary" from cte where "Rank"= 2), null) as SCH ;

Type 2

with cte as (

select salary, dense\_rank() over (order by salary desc) as rank from employee

)

select COALESCE((select top 1 salary from cte where rank = 2),null) as SecondHighestSalary;

**Customers who didnt purchase anything**

create table Customers

(Id int, Name varchar(255));

insert into Customers(Id, Name)

values (1, 'Joe'), (2, 'Henry'), (3, 'Sam'), (4, 'Max');

create table orders

(Id int, CustomerId int);

insert into orders (Id, CustomerId)

values (1,3), (2,1);

select \* from Orders;

select Customers.name customers from customers

left join orders on customers.id = orders.customerid

where orders.id is null;

**Combine first name, last name, suffix in format**

insert into names

values ('Jess', 'Ces', 'Ms'), ('Jeb', 'Kan', 'Mr'), ('Mar', 'Feb',NULL),('Day',NULL,'K');

select \* from names;

select coalesce(firstname,'')||' '|| coalesce(lastname, '')||' '|| coalesce('('|| surname ||')', '')

from names

Ans: Jes Ces (Ms)

**Select highest salary by department**

create table employee

(Id int, Name varchar(255), Salary int, DepartmentId int);

insert into employee

values (1, 'Joe', 70000,1), (2, 'Jim', 90000, 1), (3, 'Henry', 80000,2), (4, 'Sam', 60000, 2), (5, 'Max', 90000, 1);

drop table department;

create table department

(Id int, Name varchar(255));

insert into department

values(3, 'IT'),(2,'Sales');

with cte as

(select employee.name Employee, employee.salary, department.name Department,

rank() over (partition by department.name order by employee.salary desc) R

from employee join department

on employee.departmentid = department.id)

select Department, Employee, salary from cte

where R=1;

**Rank scores without gaps**

create table Scores

(Id int, Score float);

Insert into Scores

values(1,3.50),(2,3.65),(3,4.00),(4,3.85),(5,4.00),(6,3.65);

select score,

dense\_rank() over(order by score desc) "Rank"

from Scores;

**Employee’s salary more than manager’s (self join)**

create table employee1

(Id int, Name varchar(255), Salary int, ManagerId int);

insert into employee1

values(1, 'Joe', 70000, 3), (2, 'Henry', 80000,4), (3, 'Sam', 60000,NULL), (4,'Max', 90000,NULL);

select \* from employee1;

with cte as

(select e1.id, e1.name, e1.salary ESalary, e1.managerid, e2.salary MSalary

from employee1 e1 join employee1 e2 on

e1.managerid = e2.id)

select name from cte where Esalary>Msalary;

**Exchange student name (odd or even)**

create table mary (id int, name varchar(255));

insert into mary

values(1,'Abbot'),(2,'Doris'),(3,'Emerson'),(4,'Green'),(5,'Jeams');

with cte as

(select id, name,

lag(name) over(order by id), lead(name) over(order by id)

from mary)

select id,

case when id%2 !=0 AND id != (select count(id) from cte) then lead

     when id%2=0 then lag

else name end as student

from cte;

**Compare temp to previous date**

Solution 1

with cte as (select \*, lag(recorddate) over (order by recorddate) ND, lag(temperature) over (order by recorddate) from weather)

select id from cte where lag < temperature and recorddate-ND=1 ;

Solution 2

select w1.id from weather w1

join (select \*, DATEADD(DAY, 1, (Recorddate)) lastday from weather) w2

on w1.recorddate = w2.lastday

where w1.temperature > w2.temperature