1)Write an SQL query to report the first name, last name, city, and state of each person in the Person

table. If the address of a personId is not present in the Address table, report null instead.

select firstname,lastname,city,state from person left join address on person.personId=Address.personid

2)Write an SQL query to report the second highest salary from the Employee table. If there is no second highest salary, the query should report null.

Select max(salary) as SecondHighestSalary from Employee where salary not in (select max(salary) from Employee)

SELECT MAX(Salary) From Employee e WHERE Salary < ( SELECT Max(Salary) FROM Employee);

3)Write an SQL query to report the nth highest salary from the Employee table. If there is no nth highest salary, the query should report null.

SELECT DISTINCT salary AS nth\_highest\_salary FROM Employee ORDER BY salary DESC LIMIT 1 OFFSET (N - 1);

Eg:

SELECT DISTINCT salary

FROM Employee

ORDER BY salary DESC

LIMIT 1 OFFSET 4;

4)Write an SQL query to rank the scores. The ranking should be calculated according to the

following rules1

\_x0019\_ The scores should be ranked from the highest to the lowest+

\_x0019\_ If there is a tie between two scores, both should have the same ranking+

\_x0019\_ After a tie, the next ranking number should be the next consecutive integer value. In other

words, there should be no holes between ranks.

select Score, dense\_rank() over(order by Score desc) as 'Rank' from Scores

5)Write an SQL query to find all numbers that appear at least three times consecutively.

SELECT DISTINCT t1.num

FROM logs t1

JOIN logs t2 ON t1.num = t2.num

JOIN logs t3 ON t2.num = t3.num

JOIN logs t4 ON t3.num = t4.num

WHERE t1.id = t2.id - 1 AND t2.id = t3.id - 1 AND t3.id= t4.id-1

6)Write an SQL query to find the employees who earn more than their managers.

Select e.Name as Employee from Employee e inner join Employee m

on e.ManagerId = m.Id where e.Salary > m.Salary

7)Write an SQL query to report all the duplicate emails. Note that it's guaranteed that the email

field is not NULL.

select Email

from Person

group by Email

having count(Email) > 1;

8)Write an SQL query to report all customers who never order anything.

select

Name as Customers

from

Customers

where

Id

not in(

select CustomerId from Orders

);

9)Write an SQL query to find employees who have the highest salary in each of the departments.

SELECT d.name AS Department, e.name AS Employee, e.Salary

FROM Employee e

INNER JOIN Department d ON e.departmentId = d.id

WHERE (e.departmentId, e.salary) IN (

SELECT departmentId, MAX(salary)

FROM Employee

GROUP BY departmentId

);

SELECT d.Name as Department, e.Name as Employee, e.Salary FROM Employee e

    left join Department d on e.DepartmentId=d.Id

    GROUP BY d.Name

    order by e.Salary desc limit 1

10)Department Top Three Salaries

Question. 10

A company's executives are interested in seeing who earns the most money in each of the

company's departments. A high earner in a department is an employee who has a salary in the

top three unique salaries for that department.

Write an SQL query to find the employees who are high earners in each of the departments.

SELECT

d.Name AS 'Department', e1.Name AS 'Employee', e1.Salary

FROM

Employee e1

JOIN

Department d ON e1.DepartmentId = d.Id

WHERE

3 > (SELECT

COUNT(DISTINCT e2.Salary)

FROM

Employee e2

WHERE

e2.Salary > e1.Salary

AND e1.DepartmentId = e2.DepartmentId

)

11)Write an SQL query to delete all the duplicate emails, keeping only one unique email with the

smallest id. Note that you are supposed to write a DELETE statement and not a SELECT one.

After running your script, the answer shown is the Person table. The driver will first compile and

run your piece of code and then show the Person table. The final order of the Person table does

not matter.

delete A from Person A, Person B where A.id > B.id and A.email=B.email;

12)Write an SQL query to find all dates' Id with higher temperatures compared to its previous dates

(yesterday).

SELECT

weather.id AS 'Id'

FROM

weather

JOIN

weather w ON DATEDIFF(weather.recordDate, w.recordDate) = 1

AND weather.Temperature > w.Temperature

13)The cancellation rate is computed by dividing the number of canceled (by client or driver)

requests with unbanned users by the total number of requests with unbanned users on that

day.

Write a SQL query to find the cancellation rate of requests with unbanned users (both client and

driver must not be banned) each day between "2013-10-01" and "2013-10-03". Round

Cancellation Rate to two decimal points.

SELECT t.request\_at AS 'Day', ROUND(SUM(

IF(t.status LIKE 'ca%', 1, 0)

)/COUNT(\*),2) AS "Cancellation Rate"

FROM Trips t

JOIN Users clients

ON t.client\_id = clients.users\_id

AND clients.banned = 'No'

JOIN Users drivers

ON t.driver\_id = drivers.users\_id

AND drivers.banned = 'No'

WHERE t.request\_at

BETWEEN '2013-10-01'

AND '2013-10-03'

GROUP BY t.request\_at;

14)Write an SQL query to report the names of the customer that are not referred by the

customer with id = 2.

select name from Customer where referee\_id!=2 or referee\_id is null

15)Write an SQL query to find the customer\_number for the customer who has placed the largest

number of orders.

The test cases are generated so that exactly one customer will have placed more orders than any

other customer.

select customer\_number from

(

select customer\_number, count(\*) as cnt from orders group by customer\_number

) as e

order by e.cnt desc

limit 1

16)A country is big if,

\* it has an area of at least three million (i.e., 3000000 km2), o6

\* it has a population of at least twenty-five million (i.e., 25000000).

Write an SQL query to report the name, population, and area of the big countries.

select name,population,area from world where area>=3000000 or population>=25000000

17)Write an SQL query to report all the classes that have at least five students.

select class from Courses group by class having count(class)>= 5

18)Write an SQL query to display the records with three or more rows with consecutive id's, and the

number of people is greater than or equal to 100 for each.

select distinct t1.\*

from stadium t1, stadium t2, stadium t3

where t1.people >= 100 and t2.people >= 100 and t3.people >= 100

and

(

(t1.id - t2.id = 1 and t1.id - t3.id = 2 and t2.id - t3.id =1) -- t1, t2, t3

or

(t2.id - t1.id = 1 and t2.id - t3.id = 2 and t1.id - t3.id =1) -- t2, t1, t3

or

(t3.id - t2.id = 1 and t2.id - t1.id =1 and t3.id - t1.id = 2) -- t3, t2, t1

)

order by t1.id