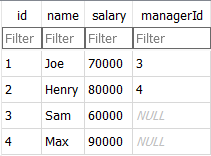
**SQL Project**

**Problem 1** SQL Schema

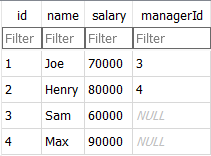
Table: Employee • id is the primary key column for this table. • Each row of this table indicates the ID of an employee, their name, salary, and the ID of their manager.

[](https://user-images.githubusercontent.com/133694651/248578557-3a1b4f6d-083f-44e6-a4e9-a20266c85354.png)

*select e.name as Employee from Employee e join Employee m on e.managerId=m.id where e.salary >m.salary;*

**Problem 2:** SQL Schema

Table: Employee • id is the primary key column for this table. • Each row of this table indicates the ID of an employee, their name, salary, and the ID of their manager.

[](https://user-images.githubusercontent.com/133694651/248578575-fc9719b7-73a6-4796-9f4f-5d0417f6772f.png)

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 DISTINCT Salary AS SecondHighestSalary FROM Employee ORDER BY Salary DESC LIMIT 1 OFFSET 1*

**Problem 3:** SQL Schema

Table: (i)

Person • personId is the primary key column for this table. • This table contains information about the ID of some persons and their first and last names.

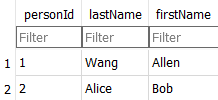
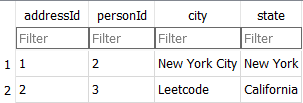
[](https://user-images.githubusercontent.com/133694651/248578269-b613b7bd-4c3d-48b6-b8e0-a2a52d8daa71.png)

Table: (ii)

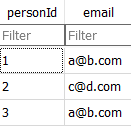
Address • addressId is the primary key column for this table. • Each row of this table contains information about the city and state of one person with ID = personId.

[](https://user-images.githubusercontent.com/133694651/248578287-2b2e090c-6a98-4e93-80a0-a93d9889a5b4.png)

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;*

**Problem 4:** SQL Schema Table: Person • id is the primary key column for this table. • Each row of this table contains an email. The emails will not contain uppercase letters.

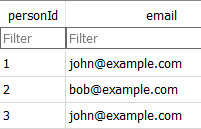
[](https://user-images.githubusercontent.com/133694651/248578356-77dacc06-b16f-47a9-8289-c9da7c0139fc.png)

Write an SQL query to report all the duplicate emails.

*select Email from Person group by Email having count(Email)>1;*

**Problem 5:** SQL Schema

Table: Person • id is the primary key column for this table. • Each row of this table contains an email. The emails will not contain uppercase letters.

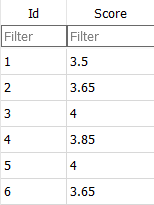
[](https://user-images.githubusercontent.com/133694651/248578392-dc62f77d-f42d-4b39-bf6a-85051a12ddc3.png)

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.

*DELETE p2 FROM Person p1 JOIN Person p2 ON p1.email = p2.email where p1.personId< p2.personId; select \* from person;*

**Problem 6:** SQL Schema

Table: Person • id is the primary key column for this table. • Each row of this table contains an ID and Score for the ID.

[](https://user-images.githubusercontent.com/133694651/248578440-a725f3a8-2363-4fbe-90f8-abeeb9854451.png)

Write a SQL query to rank scores. • If there is a tie between two scores, both should have the same ranking. • Note that 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 Person;*

**Problem 7:** SQL Schema

Table: Person • personId is the primary key column for this table. • This table contains information about the ID of some persons and their first and last names.

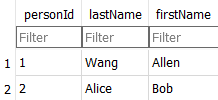
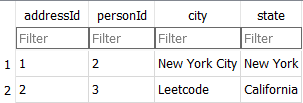
[](https://user-images.githubusercontent.com/133694651/248578479-5aecb71d-9d95-4480-b142-9e45ae56d112.png)

Table: Address • addressId is the primary key column for this table. • Each row of this table contains information about the city and state of one person with ID = personId. [](https://user-images.githubusercontent.com/133694651/248578506-e5583a5f-617c-43c2-a6a4-fd6fda5f2e28.png)

Write a SQL query for a report that provides the following information for each person in the Person table, regardless if there is an address for each of those people: FirstName, LastName, City, State

*select FirstName, LastName, City, State from Person P left join Address A on P.PersonId = A.PersonId*