C# and SQL Server Problems

[Instructions 2](#_Toc31976794)

[Unique Email Addresses 3](#_Toc31976795)

[Advanced Tic-Tac-Toe 4](#_Toc31976796)

[Duplicate Email Addresses 5](#_Toc31976797)

[Department Highest Salaries 6](#_Toc31976798)

[Camp Site Tracking 7](#_Toc31976799)

# Instructions

Feel free to make any reasonable assumptions (although noting those would be beneficial via comments). Please check this code into a public or private github repo and send us the link when you’re done. If it’s private, please share the repository with [Michael.Vrooman@extendhealth.com](mailto:Michael.Vrooman@extendhealth.com), [Ronald.Hollberg@extendhealth.com](mailto:Ronald.Hollberg@extendhealth.com), [Joel.Searcy@extendhealth.com](mailto:Joel.Searcy@extendhealth.com), and [Justin.Mayhew@extendhealth.com](mailto:Justin.Mayhew@extendhealth.com). Our hope is that this is simple enough it won’t take too much of your time.

# Unique Email Addresses

Email addresses consist of a local-part and a domain name separated by the @ symbol. In addition to numbers and lower case letters, the local-part may contain ‘.’s or ‘+’s.

When a period is placed between characters in the name, the email is delivered to the same address as if the periods were not included. For example an email sent to [first.m.last@somewhere.com](mailto:first.m.last@somewhere.com) will be delivered to the same address as [firstmlast@somewhere.com](mailto:firstmlast@somewhere.com).

When a plus is placed in the local name, everything after the plus symbol will be ignored. This would allow for additional filtering by the recipients.

These rules only apply to the local-part and do not apply to the domain name. The domain name can consist of lowercase characters and the ‘.’ symbol.

Example: team1@somwhere.com , [team.1+bob@somewhere.com](mailto:team.1+bob@somewhere.com) and [team1+jill+bob@somewhere.com](mailto:team1+jill+bob@somewhere.com) would all go to the same email address. [team2@somewhere.com](mailto:team2@somewhere.com) and [team2@some.where.com](mailto:team2@some.where.com) would go to different addresses,

Given a list of email addresses, return the number of unique email addresses in C#.

The method stub for this method is as follows:

public class Solution

{

public static int NumberOfUniqueEmailAddresses(string[] emails)

{

Dictionary<string, int> addressParts = new Dictionary<string, int>();

int i = 0;

foreach (var item in emails)

{

var idx = item.IndexOf('@');

var left = item.Substring(0, idx);

if (left.Contains('.'))

left = Regex.Replace(left, @"[.]", "");

var right = item.Substring(idx+1);

if (left.Contains('+'))

left = left.Substring(0, left.IndexOf('+'));

var address = left + '@' + right;

if (!addressParts.ContainsKey(address))

addressParts.Add(address, 1);

i++;

}

return addressParts.Count;

}

}

# Advanced Tic-Tac-Toe

Design a Tic-Tac-Toe game that can be played on an *n x n* grid by two players.

It can be assumed that all inputs into the game will be valid modes. After a winning condition is reached, no more moves will be allowed. The winning condition is to place n pieces either horizontally, vertically or diagonally.

The structure of this game will be as follows:

public class TicTacToe

{

/// <summary>

/// Created a Tic Tac Tow game board

/// </summary>

/// <param name="n">nxn dimension for the game board</param>

public TicTacToe(int n)

{

}

/// <summary>

/// Place a piece on the game board

/// </summary>

/// <param name="row">row to place a piece</param>

/// <param name="col">column to place a piece</param>

/// <param name="player">the player (1 or 2) the piece is for</param>

/// <returns>0 = no winner, 1 = player 1 won, 2 = player 2 won</returns>

public int PlacePiece(int row, int col, int player)

{

}

}

# Duplicate Email Addresses

Write a SQL query that will return duplicate email addresses in an ‘Employee’ Table

Example:

Employee

|  |  |
| --- | --- |
| employeeId | email |
| 1 | bob.smith@somwhere.com |
| 2 | eric.jones@somewhere.com |
| 3 | Jill.brown@somewhere.com |
| 4 | bob.smith@somwhere.com |

The query for this table would result in

|  |
| --- |
| email |
| bob.smith@somwhere.com |

SELECT email

From Employee

Group BY email

Having Count(email) > 1

# Department Highest Salaries

Write a query that will find the employees that have the highest salary for their department. If multiple employees share the same salary, the result should include both employees. The result should include the department name, employee name and the salary.

The Employee table stores an employeeId, name, salary and departmentId

The Department table stores a departmentId and a name.

Example:

Employee

|  |  |  |  |
| --- | --- | --- | --- |
| employeeId | name | salary | departmentId |
| 1 | Eric | 85000 | 1 |
| 2 | Jill | 95000 | 1 |
| 3 | Bob | 76000 | 2 |
| 4 | Emily | 76000 | 2 |
| 5 | Sam | 75000 | 2 |

Department

|  |  |
| --- | --- |
| departmentId | name |
| 1 | Sales |
| 2 | Marketing |

The query would result in:

|  |  |  |
| --- | --- | --- |
| Department | Employee | Salary |
| Sales | Jill | 95000 |
| Marketing | Bob | 76000 |
| Marketing | Emily | 76000 |

WITH maxSalary as (

select departmentid, MAX(salary) as salary from Employee group by departmentid

)

SELECT

d.name as 'Department',

e.name as 'Employee',

e.salary as 'Salary'

FROM Employee e

CROSS APPLY

(SELECT \* FROM Department where departmentid = e.departmentid) d

WHERE e.salary in (select salary from maxSalary)

# Camp Site Tracking

Millcreek Canyon management has asked you to create a table to store campsites, existing reservations, and available reservation dates for each campsite, and they would also like to know the number of visitors to the canyon each day.

* Please use relational tables to track this information
* Populate the tables with a small amount of dummy data
  + The data does not have to be accurate to Millcreek canyon real world specifications
* Create stored procedure(s) to add or cancel a reservation
  + Don’t worry about updating information on an existing reservation for this problem
* Create a view to show available campsite reservation dates
* Create a function that shows the most popular day to visit the canyon