



C++ "Diamond Problem" of Multiple Inheritance

Home

Privacy Policy GDPR

Java

If You Put System.exit(0) on Try or Catch block, Will Finally Block Execute?

JVM platform dependent?

Method overriding vs overloading

Why have a private constructor

Difference between object and class

How copy constructors work

Final modifier

Finally block

Java: Will Finally run after return?

Java Protected Example

Java serializable class example

Multiple Inheritance

How to Create a Thread in Java

Interface vs. Abstract Class

Thread States in Java

Arithmetic Exception

Overridden Method

Dynamic Binding

Can constructors be synchronized in Java?

Does Java pass by reference or by value?

Difference between a primitive type and a class type?

Does Java have pointers?

Downcasting in Java

Java: Diamond Problem

Java: Can an interface extend another interface?

Java: Are objects of the same type as the interface implemented?

Java: Can an interface be instantiated?

Find First Nonrepeated Character

Java: What's the difference between equals() and ==?

Find trailing zeros in factorial

Java Reflection Example

Bit Manipulation Interview Questions and Answers

XOR in Java

Java Inner Class Example

When to use inner classes in Java

Inner vs nested class

Java Anonymous Class Example

Anonymous Class Interface

Argument Defined Anonymous Inner Class

C/C++

Introduction

Function Overloading

Suppose that you are given the following simple database table called Employee that has 2 columns named Employee ID and Salary:

Employee	
Employee ID	Salary
3	200
4	800
7	450

Write a SQL query to get the second highest salary from the table above. Also write a query to find the nth highest salary in SQL, where n can be any number.

The easiest way to start with a problem like this is to ask yourself a simpler question first. So, let's ask ourselves how can we find the **highest** salary in a table? Well, you probably know that is actually really easy – we can just use the MAX aggregate function:

```
select MAX(Salary) from Employee;
```

Remember that SQL is based on set theory

You should remember that SQL uses **sets** as the foundation for most of its queries. So, the question is how can we use set theory to find the 2nd highest salary in the table above? Think about it on your own for a bit – even if you do not remember much about sets, the answer is very easy to understand and something that you might be able to come up with on your own.

Figuring out the answer to find the 2nd highest salary

What if we try to **exclude** the highest salary value from the result set returned by the SQL that we run? If we remove the highest salary from a group of salary values, then we will have a new group of values whose highest salary is actually the 2nd highest in the *original* Employee table.

So, if we can somehow select the highest value from a result set that **excludes** the highest value, then we would actually be selecting the 2nd highest salary value. Think about that carefully and see if you can come up with the actual SQL yourself before you read the answer that we provide below. Here is a small hint to help you get started: you will have to use the **"NOT IN"** SQL operator.

Solution to finding the 2nd highest salary in SQL

Now, here is what the SQL will look like:

```
SELECT MAX(Salary) FROM Employee
WHERE Salary NOT IN (SELECT MAX(Salary) FROM Employee )
```

Running the SQL above would return us "450", which is of course the 2nd highest salary in the Employee table.

[Subscribe to our newsletter for more free interview questions.](#)

An explanation of the solution

The SQL above first finds the highest salary value in the Employee table using "(select MAX(Salary) from Employee)". Then, adding the "WHERE Salary NOT IN" in front basically creates a **new set of Salary values** that does not include the highest Salary value. For instance, if the highest salary in the Employee table is 200,000 then that value will be excluded from the results using the "NOT IN" operator, and all values except for 200,000 will be retained in the results.

This now means that the highest value in this *new* result set will actually be the 2nd highest value in the Employee table. So, we then select the max Salary from the new result set, and that gives us 2nd highest Salary in the Employee table. And that is how the query above works.

An alternative solution using the not equals SQL operator

We can actually use the not equals operator – the "<>" – instead of the NOT IN operator as an alternative solution to this problem. This is what the SQL would look like:

How to Read Complex C Declarations
C++ abstract keyword
Pure Virtual Function
Lvalues and Rvalues in C
Inline vs. Macro
Diamond Problem
How Vtables Work
Virtual Destructors
Friend Classes in C++
How do you call C functions from C++?
What is a memory leak in C++?
What is the difference between delete and delete[]?
What's the difference between a class variable and an instance variable?
Can static function access non-static members of class?
Execution order of constructor and destructor in inheritance
Does C++ support multiple inheritance?
Can you change the "this" pointer?
Function Templates in C++ example
C++: The compiler and function templates
C++: Function template with more than one type parameter
C++: #include " " vs. #include <>
C++: Name Hiding
C++ Ellipsis Catch Handler
What happens if a thrown exception is not handled?
C++ namespace example
C++ const pointer examples

Databases/SQL

SQL Interview Questions
UNION vs. UNION ALL
Inner vs. Outer joins
SQL Key Definition
Differences between Primary and Foreign Keys
Natural Key In Database
Secondary Key
Simple key in SQL
Superkey Example
What is Referential Integrity
Having vs. Where clause
How do database indexes work?
What is a self join?
Example of DISTINCT in SQL
Retrieve unique rows without DISTINCT
Practice Interview Question 1
Practice Interview Question 1 continued
Practice Interview Question 1 continued
Practice Interview Question 2
Advanced SQL Interview Questions and Answers
Advanced SQL Interview Questions and Answers Part 2
Practice Interview Question 2 continued
Data Mining vs. Data Warehousing
Ternary/Three-valued Logic in SQL
Find Maximum Value Without Using Aggregate
SQL Injection Example and Tutorial

```
select MAX(Salary) from Employee
WHERE Salary <> (select MAX(Salary) from Employee )
```

How would you write a SQL query to find the Nth highest salary?

What we did above was write a query to find the 2nd highest Salary value in the Employee table. But, another commonly asked interview question is how can we use SQL to find the **Nth** highest salary, where N can be any number whether it's the 3rd highest, 4th highest, 5th highest, 10th highest, etc? This is also an interesting question – try to come up with an answer yourself before reading the one below to see what you come up with.

The answer and explanation to finding the nth highest salary in SQL

Here we will present one possible answer to finding the nth highest salary first, and the explanation of that answer after since it's actually easier to understand that way. Note that the first answer we present is actually not optimal from a performance standpoint since it uses a subquery, but we think that it will be interesting for you to learn about because you might just learn something new about SQL. If you want to see the more optimal solutions first, you can skip down to the sections that says "Find the nth highest salary without a subquery" instead.

The SQL below will give you the correct answer – but you will have to plug in an actual value for N of course. This SQL to find the Nth highest salary should work in SQL Server, MySQL, DB2, Oracle, Teradata, and almost any other RDBMS:

```
SELECT * /*This is the outer query part */
FROM Employee Emp1
WHERE (N-1) = ( /* Subquery starts here */
SELECT COUNT(DISTINCT(Emp2.Salary))
FROM Employee Emp2
WHERE Emp2.Salary > Emp1.Salary)
```

How does the query above work?

The query above can be quite confusing if you have not seen anything like it before – pay special attention to the fact that "Emp1" appears in both the subquery (also known as an inner query) and the "outer" query. The outer query is just the part of the query that is not the subquery/inner query – both parts of the query are clearly labeled in the comments.

The subquery is a correlated subquery

The subquery in the SQL above is actually a specific type of subquery known as a **correlated** subquery. The reason it is called a correlated subquery is because the the subquery uses a value from the outer query in it's WHERE clause. In this case that value is the Emp1 table alias as we pointed out earlier. A normal subquery can be run **independently** of the outer query, but a **correlated** subquery can **NOT** be run independently of the outer query. If you want to read more about the differences between correlated and uncorrelated subqueries you can go here: [Correlated vs Uncorrelated Subqueries](#).

The **most** important thing to understand in the query above is that the subquery is evaluated each and every time a *row* is processed by the outer query. In other words, the inner query can not be processed independently of the outer query since the inner query uses the Emp1 value as well.

Finding nth highest salary example and explanation

Let's step through an actual example to see how the query above will actually execute step by step. Suppose we are looking for the 2nd highest Salary value in our table above, so our N is 2. This means that the query will look like this:

```
SELECT *
FROM Employee Emp1
WHERE (1) = (
SELECT COUNT(DISTINCT(Emp2.Salary))
FROM Employee Emp2
WHERE Emp2.Salary > Emp1.Salary)
```

You can probably see that Emp1 and Emp2 are just aliases for the same Employee table – it's like we just created 2 separate clones of the Employee table and gave them different names.

Understanding and visualizing how the query above works

SQL Injection Prevention

Blind SQL Injection Example

Parameterized Queries vs Prepared Statements

Prepared Statement Example

Difference between a full join and an inner join?

Difference between a left outer join and right outer join?

Difference between a left join and a left outer join?

SQL: Having vs Group By

SQL: Group By with multiple columns

SQL – Select Distinct and Order By

SQL – Order By default sort order

Derived table vs subquery

Correlated vs Uncorrelated Subquery

Find nth highest salary – SQL

Cardinality in SQL

Selectivity in SQL Databases

Cardinality versus Selectivity

Clustered vs. Non Clustered Index

Page versus block

Database Locking

Lock Escalation

Database Deadlock Example

What is a database transaction?

SQL Server Transaction

Oracle Transaction

MySQL Transaction

DB2 Transaction

Concurrent Update Problem

How to Tune Database Performance

Database Full Table Scan

What is cost based optimization?

How to tune SQL queries

SQL Index Performance

What is a bitmap index?

Oracle Indexes Examples

System privileges vs. object privileges

SQL Grant

SQL Revoke

SQL Create User

Database Roles

SQL CASE Statement

SQL Searched CASE Statement

SQL Inline View

RANK() versus DENSE_RANK()

Javascript

Javascript Interview Questions

Javascript Outside the Browser

JavaScript Strict Mode

Javascript: What is ECMAScript?

How to print a Javascript object

Javascript: how "var" works

Javascript: Block scope

Javascript Function Scope

Javascript: Example of Hoisting

What is Javascript minification?

Minification vs Obfuscation

Let's assume that we are using this data:

Employee	
Employee ID	Salary
3	200
4	800
7	450

For the sake of our explanation, let's assume that N is 2 – so the query is trying to find the 2nd highest salary in the Employee table. The first thing that the query above does is process the very first row of the Employee table, which has an alias of Emp1.

The salary in the first row of the Employee table is 200. Because the subquery is correlated to the outer query through the alias Emp1, it means that when the first row is processed, the query will essentially look like this – note that all we did is replace Emp1.Salary with the value of 200:

```
SELECT *
FROM Employee Emp1
WHERE (1) = (
  SELECT COUNT(DISTINCT(Emp2.Salary))
  FROM Employee Emp2
  WHERE Emp2.Salary > 200)
```

So, what exactly is happening when that first row is processed? Well, if you pay special attention to the subquery you will notice that it's basically searching for the count of salary entries in the Employee table that are greater than 200. Basically, the subquery is trying to find how many salary entries are greater than 200. Then, that count of salary entries is checked to see if it equals 1 in the outer query, and if so then everything from that particular row in Emp1 will be returned.

Note that Emp1 and Emp2 are both aliases for the same table – Employee. Emp2 is only being used in the subquery to compare *all* the salary values to the current salary value chosen in Emp1. This allows us to find the number of salary entries (the count) that are greater than 200. And if this number is equal to N-1 (which is 1 in our case) then we know that we have a winner – and that we have found our answer.

But, it's clear that the subquery will return a 2 when Emp1.Salary is 200, because there are clearly 2 salaries greater than 200 in the Employee table. And since 2 is not equal to 1, the salary of 200 will clearly not be returned.

So, what happens next? Well, the SQL processor will move on to the next row which is 800, and the resulting query looks like this:

```
SELECT *
FROM Employee Emp1
WHERE (1) = (
  SELECT COUNT(DISTINCT(Emp2.Salary))
  FROM Employee Emp2
  WHERE Emp2.Salary > 800)
```

Since there are no salaries greater than 800, the query will move on to the last row and will of course find the answer as 450. This is because 800 is greater than 450, and the count will be 1. More precisely, the entire row with the desired salary would be returned, and this is what it would look like:

EmployeeID	Salary
7	450

It's also worth pointing out that the reason DISTINCT is used in the query above is because there may be duplicate salary values in the table. In that scenario, we only want to count repeated salaries just once, which is exactly why we use the DISTINCT operator.

A high level summary of how the query works

Let's go through a high level summary of how someone would have come up with the SQL in the first place – since we showed you the answer first without really going through the thought process one would use to arrive at that answer.

Think of it this way – we are looking for a pattern that will lead us to the answer. One way to look at it is that the 2nd highest salary would have just one salary that is greater than it. The 4th highest salary would have 3 salaries that are greater than it. In more general terms, ***in order to find the Nth highest salary, we just find the salary that has exactly N-1 salaries greater than itself.*** And that is exactly what

Falsy values in Javascript
Javascript Method Chaining
Javascript Nested Function Example
Javascript Global Object
Wrapper Objects in Javascript
PHP
Find a Client IP Address
PHP Interview Questions
Difference between == and === in PHP?
How would you parse HTML in PHP?
PHP: What are magic methods?
PHP: Example of the __autoload function
PHP: self vs \$this
PHP: self vs. static
Find if string contains another string – php
How to delete an element from an array in php?
PHP toString
Return JSON from PHP
How to remove warnings in PHP?
Advanced PHP Interview Questions And Answers
Advanced PHP Interview Questions And Answers Part 2
Advanced PHP Interview Questions And Answers Part 3
How to return an array from a PHP function
How to delete cookies in PHP
PHP cookie versus session
Can sessions work without cookies?
Data Structures
Data Structure Interview Questions and Answers
Big O Notation
Big O versus Big Omega notations
DFS vs. BFS
Hash Tables versus Binary Search Trees
How to find if a linked list is circular has a cycle or ends
Preorder Traversal Algorithm
Inorder Traversal
Postorder Traversal
Difference between stack and heap
Find nth to last element in a linked list
Delete a node in the middle of a singly linked list
Reverse a linked list
Design Pattern Questions
Design Pattern Interview Questions and Answers
What Is GOF?
Design Pattern Interview Question 1
Design Pattern Interview Question Part 2
Design Pattern Interview Question part 3
Decorator pattern versus Inheritance
Thread Safety in Singleton
Excel Interview Questions
Excel Interview Questions
Cut and paste versus copy and paste in Excel

the query above accomplishes – it simply finds the salary that has N-1 salaries greater than itself and returns that value as the answer.

Find the nth highest salary using the TOP keyword in SQL Server

We can also use the TOP keyword (for databases that support the TOP keyword, like SQL Server) to find the nth highest salary. Here is some fairly simply SQL that would help us do that:

```
SELECT TOP 1 Salary
FROM (
    SELECT DISTINCT TOP N Salary
    FROM Employee
    ORDER BY Salary DESC
) AS Emp
ORDER BY Salary
```

To understand the query above, first look at the subquery, which simply finds the N highest salaries in the Employee table and arranges them in descending order. Then, the outer query will actually rearrange those values in ascending order, which is what the very last line "ORDER BY Salary" does, because of the fact that the [ORDER BY Default](#) is to sort values in ASCENDING order. Finally, that means the Nth highest salary will be at the top of the list of salaries, which means we just want the first row, which is exactly what "SELECT TOP 1 Salary" will do for us!

Find the nth highest salary without using the TOP keyword

There are many other solutions to finding the nth highest salary that do not need to use the TOP keyword, one of which we already went over. Keep reading for more solutions.

Find the nth highest salary in SQL without a subquery

The solution we gave above actually does *not* do well from a performance standpoint. This is because the use of the subquery can really slow down the query. With that in mind, let's go through some different solutions to this problem for different database vendors. Because each database vendor (whether it's MySQL, Oracle, or SQL Server) has a different SQL syntax and functions, we will go through solutions for specific vendors. But keep in mind that the solution presented above using a subquery should work across different database vendors.

Find the nth highest salary in MySQL

In MySQL, we can just use the LIMIT clause along with an offset to find the nth highest salary. If that doesn't make sense take a look at the MySQL-specific SQL to see how we can do this:

```
SELECT Salary FROM Employee
ORDER BY Salary DESC LIMIT n-1,1
```

Note that the DESC used in the query above simply arranges the salaries in descending order – so from highest salary to lowest. Then, the key part of the query to pay attention to is the "LIMIT N-1, 1". The LIMIT clause takes two arguments in that query – the first argument specifies the offset of the first row to return, and the second specifies the maximum number of rows to return. So, it's saying that the offset of the first row to return should be N-1, and the max number of rows to return is 1. What exactly is the offset? Well, the offset is just a numerical value that represents the number of rows from the very first row, and since the rows are arranged in descending order we know that the row at an offset of N-1 will contain the (N-1)th highest salary.

Find the nth highest salary in SQL Server

In SQL Server, there is no such thing as a LIMIT clause. But, we can still use the offset to find the nth highest salary without using a subquery – just like the solution we gave above in MySQL syntax. But, the SQL Server syntax will be a bit different. Here is what it would look like:

```
SELECT Salary FROM Employee
ORDER BY Salary DESC OFFSET N-1 ROW(S)
FETCH FIRST ROW ONLY
```

Note that I haven't personally tested the SQL above, and I believe that it will only work in SQL Server 2012 and up. Let me know in the comments if you notice anything else about the query.

Find the nth highest salary in Oracle using rownum

How to link cells in Excel

Absolute references in Excel

Excel Range Formula

Excel Entire Column

How to move an entire column in excel?

How to add a dropdown list to Excel

HTML5

HTML5 Interview Questions and Answers

What is a CSS Box Model?

History of HTML5

Void elements HTML5

New Features in HTML5 Forms

HTML5 email form field

HTML5 Polyfill Example

What is a shiv in HTML5?

HTML5 Video Codecs

Containers versus codecs

Codec Hardware Support

Networking

Networking Interview Questions

Why Use Mesh Topology for My Network?

Baud rate vs bit rate

What is a MAC address?

What happens when you visit a website?

Operating Systems

Operating System Interview Questions

How Virtual Memory Works

Paging and Page Faults

Purpose of Swapping

Thread vs. Process

Swapping Versus Paging

Vmware Fusion come with Windows?

Does Parallels Desktop come with Windows?

Monitors vs Semaphores

Example of Threading and Synchronization

User Threads vs. System Threads

Recursion

Recursion Interview Questions

Explanation of Recursion

Can every recursive function be made iterative?

Recursion Versus Iteration

Permutations of A String

Factorial Iterative and Non Recursive

Find nth Fibonacci number

Tail Recursion

Tail Call Optimization

Apache Interview Questions

Introduction

Purpose of ASF's DocumentRoot Directive

Is a restart to Apache required after a change to .htaccess file?

How to password protect a file in .htaccess

How to disable password protection in .htaccess

Oracle syntax doesn't support using an offset like MySQL and SQL Server, but we can actually use the row_number analytic function in Oracle to solve this problem. Here is what the Oracle-specific SQL would look like to find the nth highest salary:

```
select * from (
    select Emp.*,
    row_number() over (order by Salary DESC) rownumb
    from Employee Emp
)
where rownumb = n; /*n is nth highest salary*/
```

The first thing you should notice in the query above is that inside the subquery the salaries are arranged in descending order. Then, the row_number analytic function is applied against the list of descending salaries. Applying the row_number function against the list of descending salaries means that each row will be assigned a row number starting from 1. And since the rows are arranged in descending order the row with the highest salary will have a 1 for the row number. Note that the row number is given the alias rownumb in the SQL above.

This means that in order to find the 3rd or 4th highest salary we simply look for the 3rd or 4th row. The query above will then compare the rownumb to n, and if they are equal will return everything in that row. And that will be our answer!

Find the nth highest salary in Oracle using RANK

Oracle also provides a RANK function that just assigns a ranking numeric value (with 1 being the highest) for some sorted values. So, we can use this SQL in Oracle to find the nth highest salary using the RANK function:

```
select * FROM (
    select EmployeeID, Salary
    ,rank() over (order by Salary DESC) ranking
    from Employee
)
WHERE ranking = N;
```

The rank function will assign a ranking to each row starting from 1. This query is actually quite similar to the one where we used the row_number() analytic function, and works in the same way as well.

We've now gone through many different solutions in different database vendors like Oracle, MySQL, and SQL Server. Hopefully now you understand how to solve a problem like this, and you have improved your SQL skills in the process! Be sure to leave a comment if you have any questions or observations.

Find number of squares chessboard - Programmer and Software Interview...

How would you find the number of squares on a chessboard? Clearly there are 64 of the little square...

Programmer and Software Interview Questions and Answers

Hiring? Job Hunting? Post a JOB or your RESUME on our JOB BOARD >>

Subscribe to our newsletter for more free interview questions.

Follow @programmerintw

Previous...

Next...