USING RANK & ROW NUMBER

```
CREATE DATABASE ChrisWorkDB
GO
IF OBJECT_ID('dbo.ChrisWorkDB', 'U') IS NOT NULL
 DROP TABLE [dbo].[ChrisWorkDB];
 GO
 CREATE TABLE Employee
 ( EmployeeId INT
  ,EmployeeName VARCHAR(50)
  ,AnnualSalary INT
 )

 USE ChrisWorkDB;
 --truncate table Employee

 INSERT INTO Employee
 VALUES(1, 'John', 35000),
       (2, 'Mary', 60000),
        (3, 'Mark', 70000),
        (4, 'Joe', 90000),
        (5, 'Chris', 30000),
        (6, 'Paul', 33000),
        (7, 'Eric', 65000),
        (8, 'Steve', 65400),
        (9, 'Brian', 101000),
        (10, 'Jennifer', 95000),
        (11, 'Mike', 82000);

USE [ChrisWorkDB]

--Checking the table...
SELECT * FROM [dbo].[Employee]
GO

/*
QUESTION: Display only the second highest salary (salary is unique).

The trick to this question is to remember that this is nothing less
than a sub-query, that I was told can be done in four different ways.
I am showing these four ways that are the most common. Note: Use Rank
if only the AnnualSalary is unique, if not unique, then use the
DENSE_RANK() function. The DENSE_RANK function returns every row
consecutively and is very useful if you have the same values on other
rows.

*/
```

```
--Option 1: Common Table Expression (CTE)
With cte_Ranking_From_Highest
  AS
     (SELECT EmployeeId, EmployeeName, AnnualSalary,
            ROW_NUMBER() OVER(ORDER BY AnnualSalary DESC) AS Ranking
        FROM Employee
        )

SELECT EmployeeId, EmployeeName, AnnualSalary, Ranking
FROM cte_Ranking_From_Highest
WHERE Ranking = 2
GO


--Option 2: Subquery: Note, You can Use Rank() or Row_Number().
SELECT EmployeeId, EmployeeName, AnnualSalary, Ranking
FROM (SELECT EmployeeId, EmployeeName, AnnualSalary,
           RANK() OVER(ORDER BY AnnualSalary DESC) AS Ranking
        FROM Employee) AS t
WHERE Ranking = 2
GO


--Option 3: Subquery using INNER JOIN. Note: INNER JOIN’s are
                                                     typically faster.
SELECT e1.EmployeeId, e1.AnnualSalary, e2.Ranking
FROM Employee AS e1
     INNER JOIN (SELECT EmployeeId, AnnualSalary,
                         RANK() OVER(ORDER BY AnnualSalary DESC) AS
Ranking
                  FROM Employee) AS e2
      ON e1.EmployeeId = e2.EmployeeId
WHERE e2.Ranking = 2


--Option 4: If you know the second highest, then you can directly get
            it this way (simple). Note: This does not need Ranking.
SELECT EmployeeId, EmployeeName, AnnualSalary
FROM Employee
WHERE AnnualSalary = 95000 -- This is the second highest.
ORDER BY AnnualSalary DESC


--This is the result of all four ways, the forth not needing Ranking.
```

```
Result:
EmployeeId   EmployeeName      AnnualSalary  Ranking
-----------------------------------------------------------------------------
10               Jennifer              95000       2
```