/****************/
/* SQL SERVER   */
/* INTERVIEW Q. */
/*   INDEX   */
/****************/

--001-- FIND NTH HIGHEST SALARY IN SQL SERVER
--002-- SQL QUERY TO GET ORGANIZATION HIERARCHY
--003-- HOW DOES A RECURSIVE CTE WORK (CONTINUATION OF 002)
--004-- DELETE DUPLICATE ROWS IN SQL
--005-- SQL QUERY TO FIND EMPLOYEES HIRED IN THE LAST N MONTHS
--006-- TRANSFORM ROWS INTO COLUMNS IN SQL SERVER
--007-- SQL QUERY TO FIND ROWS THAT CONTAIN ONLY NUMERICAL DATA
--008- SQL QUERY TO FIND THE DEPARTMENT WITH HIGHEST NUMBER OF EMPLOYEES
--009-- DIFFERENCE BETWEEN INNER JOIN AND LEFT JOIN

--001-- FIND NTH HIGHEST SALARY IN SQL SERVER

This is a very common SQL Server Interview Question. There are several ways of finding the nth highest salary.

By the end of this video, we will be able to answer all the following questions as well.
How to find nth highest salary in SQL Server using a Sub-Query
How to find nth highest salary in SQL Server using a CTE
How to find the 2nd, 3rd or 15th highest salary

Lets use the following Employees table for this demo

Use the following script to create Employees table

```
Create table Employees
(
ID int primary key identity,
FirstName nvarchar(50),
LastName nvarchar(50),
Gender nvarchar(50),
Salary int
)
GO
```

```
Insert into Employees values ('Ben', 'Hoskins', 'Male', 70000)
Insert into Employees values ('Mark', 'Hastings', 'Male', 60000)
Insert into Employees values ('Steve', 'Pound', 'Male', 45000)
Insert into Employees values ('Ben', 'Hoskins', 'Male', 70000)
Insert into Employees values ('Philip', 'Hastings', 'Male', 45000)
Insert into Employees values ('Mary', 'Lambeth', 'Female', 30000)
Insert into Employees values ('Valarie', 'Vikings', 'Female', 35000)
Insert into Employees values ('John', 'Stanmore', 'Male', 80000)
```

GO

To find the highest salary it is straight forward. We can simply use the Max() function as shown below.
Select Max(Salary) from Employees

To get the second highest salary use a sub query along with Max() function as shown below.
Select Max(Salary)
from Employees
where Salary < (Select Max(Salary) from Employees)

```
--To find 2nd highest (or any number high) salary using Sub-Query
SELECT TOP 1 SALARY
FROM ( SELECT DISTINCT TOP 2 SALARY
          FROM EMPLOYEES
          ORDER BY SALARY DESC ) RESULT
ORDER BY SALARY


--To find nth highest salary using CTE
WITH RESULT AS -- WITH KEYWORD WILL MAKE THE PARENTHESIS CONTENT A
COMMON TABLE EXPRESSION
(
        SELECT SALARY,
        DENSE_RANK() -- IF YOU USE ROW_NUMBER() FUNCTION YOU WILL GET THE
THIRD HIGHEST SALARY FOR DUPLICATE DATA WHERE TWO SAME RECORDS ARE
CONSIDERED DIFFERENT
        OVER (ORDER BY SALARY DESC) AS DENSERANK
        FROM EMPLOYEES
)

SELECT TOP 1 SALARY
FROM RESULT
WHERE DENSERANK = N -- N WILL EQUAL WHATEVER N HIGHEST SALARY WE WANT

-- To find 2nd highest salary we can use any of the above queries.
-- Simple replace N with 2.

-- Similarly, to find 3rd highest salary, simple replace N with 3.
```

Please Note: On many of the websites, you may have seen that,
the following query can be used to get the nth highest salary.
The below query will only work if there are no duplicates.

```
WITH RESULT AS
(
        SELECT SALARY, ROW_NUMBER()
        OVER (ORDER BY SALARY DESC) AS ROWNUMBER
        FROM EMPLOYEES
)
SELECT SALARY
```

```
FROM RESULT
WHERE ROWNUMBER = 3
```

--002-- SQL QUERY TO GET ORGANIZATION HIERARCHY

Here is the problem definition:
1. Employees table contains the following columns a) EmployeeId, b) EmployeeName c) ManagerId
2. If an EmployeeId is passed, the query should list down the entire organization hierarchy i.e who is the manager of the EmployeeId passed and who is managers manager and so on till full hierarchy is listed.

For example,
Scenario 1: If we pass David's EmployeeId to the query, then it should display the organization hierarchy starting from David.

Scenario 2: If we pass Lara's EmployeeId to the query, then it should display the organization hierarchy starting from Lara.

We will be Employees table for this demo. SQL to create and populate Employees table with test data

```
Create table Employees
(
EmployeeID int primary key identity,
EmployeeName nvarchar(50),
ManagerID int foreign key references Employees(EmployeeID)
)
GO

Insert into Employees values ('John', NULL)
Insert into Employees values ('Mark', NULL)
Insert into Employees values ('Steve', NULL)
Insert into Employees values ('Tom', NULL)
Insert into Employees values ('Lara', NULL)
Insert into Employees values ('Simon', NULL)
Insert into Employees values ('David', NULL)
Insert into Employees values ('Ben', NULL)
Insert into Employees values ('Stacy', NULL)
Insert into Employees values ('Sam', NULL)
GO

Update Employees Set ManagerID = 8 Where EmployeeName IN ('Mark', 'Steve', 'Lara')
Update Employees Set ManagerID = 2 Where EmployeeName IN ('Stacy', 'Simon')
Update Employees Set ManagerID = 3 Where EmployeeName IN ('Tom')
Update Employees Set ManagerID = 5 Where EmployeeName IN ('John', 'Sam')
Update Employees Set ManagerID = 4 Where EmployeeName IN ('David')
GO
```

Here is the SQL that does the job

```
Declare @ID int ;
Set @ID = 7;

WITH EmployeeCTE AS
(
	Select EmployeeId, EmployeeName, ManagerID
	From Employees
	Where EmployeeId = @ID

	UNION ALL

	Select Employees.EmployeeId , Employees.EmployeeName, Employees.ManagerID
	From Employees
	JOIN EmployeeCTE -- WE ARE JOINING THE CTE WITH ITSELF, SO IT IS RECURSIVE
	ON Employees.EmployeeId = EmployeeCTE.ManagerID
)

Select E1.EmployeeName, ISNULL(E2.EmployeeName, 'No Boss') as ManagerName
From EmployeeCTE E1
LEFT Join EmployeeCTE E2 -- we use LEFT JOIN BECAUSE WITHOUT 'LEFT' WE WOULD NOT GET BEN IN THE HIERARCHY
ON E1.ManagerID = E2.EmployeeId
```

--003-- HOW DOES A RECURSIVE CTE WORK (CONTINUATION OF 002)

A lot of you have asked to explain, how a recursive CTE work line by line.
If you have not watched Part 2 already, I strongly recommend to watch that
video first before proceeding.

In Part 2 of SQL Server Interview questions and answers video series we discussed
recursive CTE to retrieve the organization hierarchy.

Lets now discuss how the CTE executes line by line.
Step 1: Execute the anchor part and get result R0

```
-- ANCHOR
Select EmployeeId, EmployeeName, ManagerID
From Employees
Where EmployeeId = @ID
```

Step 2: Execute the recursive member using R0 as input and generate result R1

```
--RECURSIVE
Select Employees.EmployeeId , Employees.EmployeeName, Employees.ManagerID
From Employees
JOIN EmployeeCTE -- R0 WILL BE THE RESULT OF EMPLOYEECTE
```

ON Employees.EmployeeId = EmployeeCTE.ManagerID

Step 3: Execute the recursive member using R1 as input and generate result R2
Step 4: Recursion goes on until the recursive member output becomes NULL
Step 5: Finally apply UNION ALL on all the results to produce the final output

UNION ALL

--004-- DELETE DUPLICATE ROWS IN SQL

In this video, we will discuss deleting all duplicate rows except one from a sql server table.

SQL Script to create Employees table

```
Create table Employees
(
ID int,
FirstName nvarchar(50),
LastName nvarchar(50),
Gender nvarchar(50),
Salary int
)
GO

Insert into Employees values (1, 'Mark', 'Hastings', 'Male', 60000)
Insert into Employees values (1, 'Mark', 'Hastings', 'Male', 60000)
Insert into Employees values (1, 'Mark', 'Hastings', 'Male', 60000)
Insert into Employees values (2, 'Mary', 'Lambeth', 'Female', 30000)
Insert into Employees values (2, 'Mary', 'Lambeth', 'Female', 30000)
Insert into Employees values (3, 'Ben', 'Hoskins', 'Male', 70000)
Insert into Employees values (3, 'Ben', 'Hoskins', 'Male', 70000)
Insert into Employees values (3, 'Ben', 'Hoskins', 'Male', 70000)
```

The delete query should delete all duplicate rows except one.

Here is the SQL query that does the job. PARTITION BY divides the query result set into partitions.

```
WITH EmployeesCTE AS
(
        SELECT *, ROW_NUMBER()OVER(PARTITION BY ID ORDER BY ID) AS RowNumber
        FROM Employees
)

-- THE EMPLOYEES CTE 'SELECT * FROM EMPLOYEESCTE'
-- We partition the row numbers by ID

DELETE FROM EmployeesCTE WHERE RowNumber > 1 -- this will leave just one record of
```

-- each kind.

--005-- SQL query to find employees hired in last n months

-- Replace N with number of months
Select *
FROM Employees
Where DATEDIFF(MONTH, HireDate, GETDATE()) Between 1 and N

--006-- Transform rows into columns in sql server

This is another common sql server interview question. We will be using Countries table in this example.

SQL to create the table
Create Table Countries
(
Country nvarchar(50),
City nvarchar(50)
)
GO

Insert into Countries values ('USA','New York')
Insert into Countries values ('USA','Houston')
Insert into Countries values ('USA','Dallas')

Insert into Countries values ('India','Hyderabad')
Insert into Countries values ('India','Bangalore')
Insert into Countries values ('India','New Delhi')

Insert into Countries values ('UK','London')
Insert into Countries values ('UK','Birmingham')
Insert into Countries values ('UK','Manchester')

Here is the interview question.
Write a sql query to transpose rows to columns.

Using PIVOT operator we can very easily transform rows to columns.

```
Select Country, City1, City2, City3
From
(
        Select Country, City, 'City'+ cast(
                                                        row_number() over(

        partition by Country

        order by Country
```

```
                                                                                ) as
varchar(10)
                                                                        ) ColumnSequence
        from Countries
) Temp
PIVOT -- WILL MAKE THE ROWS INTO COLUMNS
(
max(City) -- AGGREGATE FUNCTION COLUMN WILL BE CITY, WHICH IS WHAT WELL
PRIVOT
for ColumnSequence in (City1, City2, City3) -- COLUMNSEQUENCE WILL BE THE NAME
-- OF THE COLUMN THAT WILL CONTAIN THE VALUES TO BE PIVOTED
) Piv -- THIS IS THE ALIAS

-- WELL GET ONE ROW FOR EACH COUNTRY, AND THE CITIES HAVE BECOME COLUMNS
```

--007-- SQL QUERY TO FIND ROWS THAT CONTAIN ONLY NUMERICAL DATA

Let me explain the scenario mentioned in one of the sql server interview.
We have the following table.

ID	Value
1	123
2	ABC
3	DEF
4	901
5	JKL

Write a SQL query to retrieve rows that contain only numerical data.

SQL Script to create the TestTable

```
Create Table TestTable
(
        ID int identity primary key,
        Value nvarchar(50)
)

Insert into TestTable values ('123')
Insert into TestTable values ('ABC')
Insert into TestTable values ('DEF')
Insert into TestTable values ('901')
Insert into TestTable values ('JKL')
```

This is very easy to achieve. If you have used ISNUMERIC() function in SQL Server,
then you already know the answer. Here is the query

```
SELECT Value
FROM TestTable
WHERE ISNUMERIC(Value) = 1
```

ISNUMERIC function returns 1 when the input expression evaluates to a valid numeric data type, otherwise it returns 0. For the list of all valid numeric data types in SQL Server please visit the following MSDN link.

--008-- SQL QUERY TO FIND THE DEPARTMENT WITH HIGHEST NUMBER OF EMPLOYEES

Scenario asked in the SQL Server Interview
Based on the above two tables write a SQL Query to get the name of the Department that has got the maximum number of Employees. To answer this question it will be helpful if you the knowledge of JOINS & GROUP BY in SQL Server. We discusses these in Parts 11 & 12 of SQL Server Tutorial video series.

SQL query that retrieves the department name with maximum number of employees

```
SELECT TOP 1 DepartmentName
FROM Employees
JOIN Departments
ON Employees.DepartmentID = Departments.DepartmentID
GROUP BY DepartmentName
ORDER BY COUNT(*) DESC -- THIS WILL MAKE THE DEPARTMENT WITH THE HIGHEST NUMBER
-- OF EMPLOYEES TO APPEAR FIRST
```

--009-- DIFFERENCE BETWEEN INNER JOIN AND LEFT JOIN

This is one of the very common sql server interview question. Different JOINS in SQL Server are discussed in detail in Part 12 of SQL Server tutorial for beginners video series.

Lets understand the difference with an example.

INNER JOIN returns only the matching rows between the tables involved in the JOIN. Notice that, Pam employee record which does not have a matching DepartmentId in departments table is eliminated from the result-set.

```
SELECT EmployeeName, DepartmentName
FROM Employees
INNER JOIN Departments
ON Employees.DepartmentID = Departments.DepartmentID
```

LEFT JOIN returns all rows from left table including non-matching rows. Notice that, Pam employee record which does not have a matching DepartmentId in departments table is also included in the result-set.

```
SELECT EmployeeName, DepartmentName
FROM Employees
LEFT JOIN Departments
```

ON Employees.DepartmentID = Departments.DepartmentID

In general there could be several questions on JOINS in a sql server interview.
If we understand the basics of JOINS properly, then answering any JOINS related
questions should be a cakewalk.

What is the difference between INNER JOIN and RIGHT JOIN
INNER JOIN returns only the matching rows between the tables involved in the JOIN,
where as RIGHT JOIN returns all the rows from the right table including the NON-MATCHING rows.

What is the difference between INNER JOIN and FULL JOIN
FULL JOIN returns all the rows from both the left and right tables including the NON-MATCHING rows.

What is the Difference between INNER JOIN and JOIN
There is no difference they are exactly the same. Similarly there is also no difference between
LEFT JOIN and LEFT OUTER JOIN
RIGHT JOIN and RIGHT OUTER JOIN
FULL JOIN and FULL OUTER JOIN