

SQL Interview Questions

* Find the Nth Highest Salary in table :

ID	FirstName	LastName	Gender	Salary
1	Ben	Hoskins	Male	70000
2	Mark	Hastings	Male	60000
3	Steve	Pound	Male	45000
4	Ben	Hoskins	Male	70000
5	Philip	Hastings	Male	45000
6	Mary	Lambeth	Female	30000
7	Valarie	Vikings	Female	35000
8	John	Stanmore	Male	80000

- هيكون عن طريق نوعين من الـ Query :

(1) عن طريق الـ Max (Salary) :
- هتجيب الـ Maximum Value فى الـ Salary Column

Select Max(Salary) from Employees

- طب لو عاوز أجيب الـ 2nd هستخدم فكرة الـ Subquery

(الفكرة إن هقوله عاوز الـ Max فى الـ result set اللى أقل من الـ max الأساسى)

(يعنى هشىل الـ 8000 من الـ table وأقول هات الـ max على كذا "اللى هو تانى أكبر Salary")

Select Max(Salary) from Employees

where Salary < (Select Max(Salary) from Employees)

- عاوز الـ 3rd هعمل اللى فات جوا Subquery وهكذا

(2) عن طريق الـ Select Top (1) salary مع الـ order by salary Desc

- هيجيب الـ Top record فى الـ result set اللى معمولها order by salary بـ Descending

```
SELECT TOP 1 SALARY
FROM (
    SELECT DISTINCT TOP N SALARY
    FROM EMPLOYEES
    ORDER BY SALARY DESC
) RESULT
ORDER BY SALARY
```

- الفكرة هنا إن إستخدمت الـ sub query فى إنه :

(1) يرتب الـ Table بالـ Salary Desc

(2) يختار عدد منهم العدد دا يكون الـ salary المطلوب لو عاوز الـ 3rd بالتالى هحط مكان الـ N تبقى 3

(3) كدا هو معاه أول 3 Salaries فى الـ result set اللى فى الـ sub query

(4) هعمل Select top 1 فى الـ 3 Salaries بس هخلى الـ order ascending بالتالى كدا الـ 3rd هيبقى الـ top 1

(3) عن طريق الـ Dense Rank function :

- هعمل ranking لـ table بالـ salary descending (كل Salary ليه rank)

- هستخدم الـ dense عشان لو فيه Duplicates يستخدم نفس الـ Rank ليهم

- كدا الـ rank الاول هو أعلى salary والثانى هو تانى أعلى salary وهكذا

- هستخدم الـ CTE للتسهيل (Common Table Expression) : هو عبارة عن Command عبارة عن إنه بيخلي الناتج من query معين كـ Temp result set عشان أستخدامه بعد كدا فى نفس الـ batch وهو للتسهيل مش أكثر وإن الكود يبقى readable

بيتم عن طريق With Temp_CTE_Name as (Query)

WITH RESULT AS

```
(  
    SELECT SALARY,  
           DENSE_RANK() OVER (ORDER BY SALARY DESC) AS DENSERANK  
    FROM EMPLOYEES  
)
```

```
SELECT TOP 1 SALARY  
FROM RESULT  
WHERE DENSERANK =N
```

- لو عاوز أى rank هستبدل الـ N بالـ rank اللى عاوزه

* SQL Query to get Organization Hierarchy :

- الـ pre-requisite :

Self Join (1

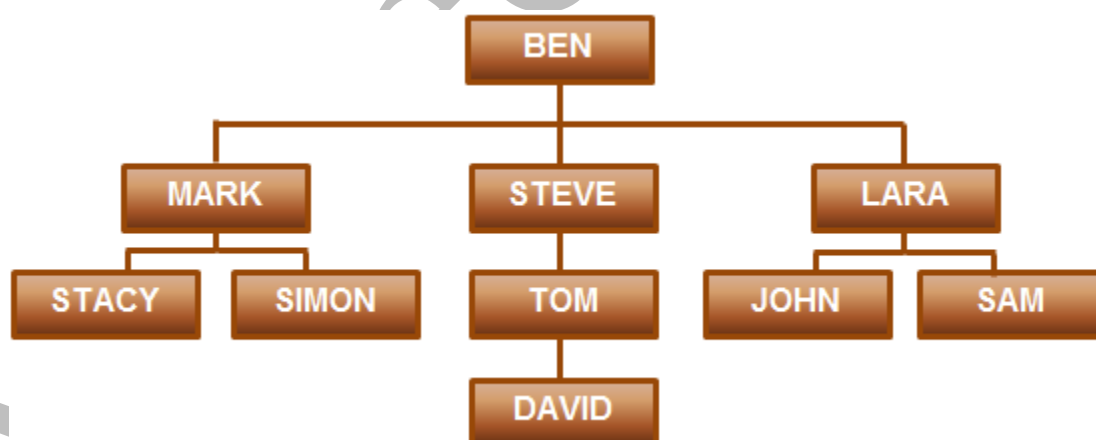
CTE (2

Recursive CTE (3

- عندى Table الـ Employee فيه :

EmployeeID	EmployeeName	ManagerID
1	John	5
2	Mark	8
3	Steve	8
4	Tom	3
5	Lara	8
6	Simon	2
7	David	4
8	Ben	NULL
9	Stacy	2
10	Sam	5

- الـ Hierarchy الناتج من الـ Data :



- المطلوب إن لما أعرض موظف معين أعرض معاه مين الـ Managers اللى عليه لحد ما أوصل لـ Super boos مثلا David

EmployeeName	ManagerName
David	Tom
Tom	Steve
Steve	Ben
Ben	No Boss

- فكرة إن أعمل Sequence لحد ما أوصل للأخر هنطبقها بال recursive CTE

- فكرة ال Recursive CTE هيا نفس ال CTE + ال Recursion (هنادى على ال CTE جواها)

```
Declare @ID int;
```

```
Set @ID = 7;
```

```
WITH EmployeeCTE AS
```

```
(
```

```
-- Anchor
```

```
Select EmployeeId, EmployeeName, ManagerID
```

```
From Employees
```

```
Where EmployeeId = @ID
```

```
UNION ALL
```

```
-- Recursive Member
```

```
Select Employees.EmployeeId , Employees.EmployeeName, Employees.ManagerID
```

```
From Employees
```

```
JOIN EmployeeCTE
```

```
ON Employees.EmployeeId = EmployeeCTE.ManagerID
```

```
)
```

```
Select E1.EmployeeName, ISNULL(E2.EmployeeName, 'No Boss') as ManagerName
```

```
From EmployeeCTE E1
```

```
LEFT Join EmployeeCTE E2
```

```
ON E1.ManagerID = E2.EmployeeId
```

(1) فى البداية جزء ال Anchor هيتنفذ بالتالى ال Result منه (4 – David – 7)

(2) هيتم تنفيذ ال Recursive Number (هو عبارة عن إنك بت join ال CTE مع ال Table)

فالناتج هيكون بيانات الموظف رقم 4 (3 – Tom – 4)

(3) نفس الخطوة هتتعد و هتطلع بيانات رقم 3 (8 – Steve – 3)

(4) هتتعد الخطوة و تطلع بيانات الموظف رقم 8 (8 – Ben – NULL)

(5) وصلت لـ Null هيبدا يعمل UNION ALL للنواتج

- بالتالى ناتج ال Recursive CTE هو :

Employee ID	Employee Name	Manager ID
7	David	4
4	Tom	3
3	Steve	8
8	Ben	NULL

- هاند ناتج ال Recursive CTE ككل وأعمل Self join
(ال PK يبقى ال Manager ID وال FK يبقى ال Employee ID)

- بكدا لما أدخل ID لأي موظف يجيب إسمه وإسم مديره وكمان إسم مدير مديره لحد ما يوصل لـ
Super Boss

* Delete Duplicates in table :

- الفكرة هنا (عاوز أمسح ال duplicates من كل record و أسيب نسخه واحدة بس)

- هستخدم ال Row_Number Function معاها ال Partition

هـ Partition ال table بالـ Column وبالتالي هيبقى عندي rank لكل Partition وأصلا كل partition هو عبارة عن Record وال Duplicates بتاعته

فهمسح الكل ماعدا اللي ال Row Number بتاعه 1 أو أي رقم (بس واحد أضمن لأنك ضامن إن فيه في كل Partition رقم 1 اللي هو أول Rank ولو مفيش Duplicate لـ Record فـ Still هو 1 برضو)

ID	FirstName	LastName	Gender	Salary
1	Mark	Hastings	Male	60000
1	Mark	Hastings	Male	60000
1	Mark	Hastings	Male	60000
2	Mary	Lambeth	Female	30000
2	Mary	Lambeth	Female	30000
3	Ben	Hoskins	Male	70000
3	Ben	Hoskins	Male	70000
3	Ben	Hoskins	Male	70000

WITH EmployeesCTE AS

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

)
DELETE FROM EmployeesCTE WHERE RowNumber > 1

ID	FirstName	LastName	Gender	Salary
1	Mark	Hastings	Male	60000
2	Mary	Lambeth	Female	30000
3	Ben	Hoskins	Male	70000

* SQL Query to find Employees hired in last n months :

- الفكرة فى استخدام built in Function وهى الـ DATEDIFF ()

DATEDIFF(datepart, startdate, enddate)

datepart → The part of the date to calculate the difference in

(e.g., day, month, year).

startdate → The starting date.

enddate → The ending date.

Select *

FROM Employees

Where DATEDIFF(MONTH, HireDate, GETDATE()) Between 1 and N

* SQL query to find rows that contain only numerical data :

Input Table :

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

output result :

Value
123
901

- الفكرة هنا إن استخدم Built in function وهى الـ IsNumeric ودى بت target الـ Numeric Values فى الـ Column

- بت 1 return لو كان الـ Numeric Value Column

- 0 لو مش Numeric

- الـ Numeric Data types :

Numeric - int – tinyint – smallint – bigint – decimal – money – smallmoney – float – real

Query :

Select Value from TestTable Where ISNUMERIC(Value) = 1