MS SQL Tips & Tricks

Sergei Tkachuk

About this webinar

This webinar
is being recorded

You were automatically muted on entry

If you have a question, use the Q&A
or chat functions

We will post the recordings and the presentations
on Veeam ProPartner Portal

© 2021 Veeam Software. Confidential information. All rights reserved. All trademarks are the property of their respective owners.

R&D Offices :

83%
of Fortune 500

35+
Countries where Veeam has offices

400,000+
Customers worldwide

4400+
Employees worldwide

3.5 times
Higher customer satisfaction

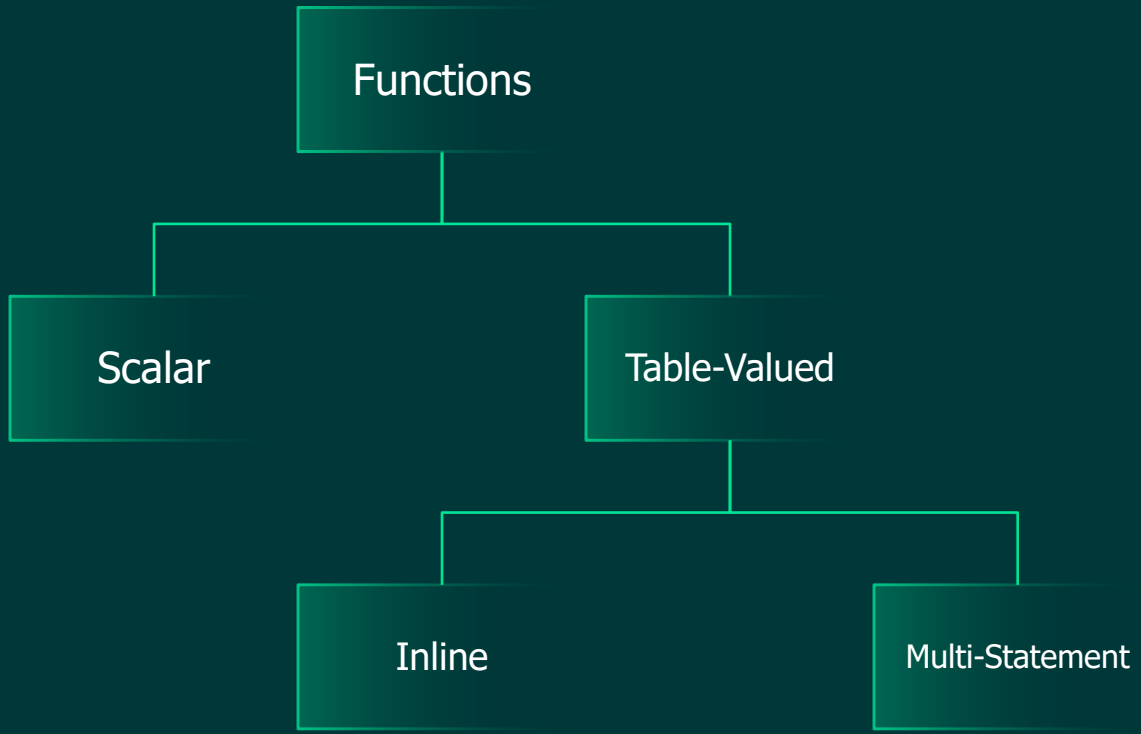
$1+ billion
Revenue in 2019

Agenda

- User-Defined Functions
- Temporary Tables vs Table Variables
- Float Data Type
- Window Functions

© 2021 Veeam Software. Confidential information. All rights reserved. All trademarks are the property of their respective owners.

User-Defined Functions



© 2021 Veeam Software. Confidential information. All rights reserved. All trademarks are the property of their respective owners.

Temporary Tables vs Table Variables

Temporary Tables

```
create table #Temp
  ( Column1   int
  , Column2   int );
```

Table Variables

```
declare @Temp
  ( Column1   int
  , Column2   int );
```

© 2021 Veeam Software. Confidential information. All rights reserved. All trademarks are the property of their respective owners.

Float Data Type

float [(n)]

n	Precision	Size
1 - 24	7 digits	4 bytes
25 - 53	15 digits	8 bytes

decimal (p, s)

Precision	Size
1 - 9	5 bytes
10 - 19	9 bytes
20 - 28	13 bytes
29 - 38	17 bytes

© 2021 Veeam Software. Confidential information. All rights reserved. All trademarks are the property of their respective owners.

Window Functions

```
select SalesOrderID, SalesOrderDetailID, ProductID, OrderQty, UnitPrice, LineTotal
     , GrandTotal = sum( LineTotal ) over( partition by SalesOrderID order by SalesOrderDetailID )
  from Sales.SalesOrderDetail
  order by SalesOrderID, SalesOrderDetailID;
```

SalesOrderID	SalesOrder DetailID	…	LineTotal	GrandTotal
43659	1		2 024.99	2 024.99
43659	2		6 074.98	8 099.98
43659	3		2 024.99	10 124.97
43659	4		2 039.99	12 164.96
43660	5		419.46	419.46
43660	6		874.79	1 294.25
43661	7		809.76	809.76
43661	8		714.70	1 524.46

1. The PARTITION BY defines the windows, one for each SalesOrderID
2. The ORDER BY sorts the rows within the window
3. The SUM calculates accumulative total value

© 2021 Veeam Software. Confidential information. All rights reserved. All trademarks are the property of their respective owners.

Thank you!

VEEAM

Experienced C# Developer
(with possible relocation to Prague)

Fullstack C# Developer

C# API Developer

C++ Developer
(with possible relocation to Prague)