

# Can We Get Good Performances from SQL Server?

---



**Rudi Bruchez**

SQL SERVER CONSULTANT AND TRAINER

[www.linkedin.com/in/rudibruchez/](http://www.linkedin.com/in/rudibruchez/) [www.babaluga.com](http://www.babaluga.com)



# Summary



Think declaratively

Think set-oriented

Think relational

Think mathematics

Real-world solutions

Production challenges

Think in SQL



# How to Think in SQL?



**Think declarative**  
**Speak to SQL Server**



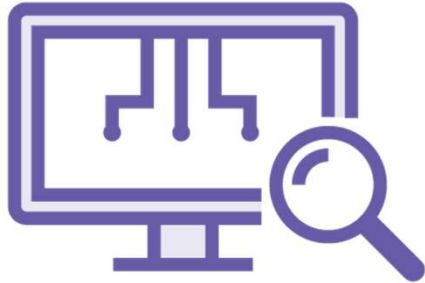
**Learn the syntax**  
**There are many T-SQL  
enhancements**



**Find solutions**  
**Search online and  
spend some time  
thinking**



# Don't Give up Too Quickly



**Profile**  
Ask SQL Server  
what it receives



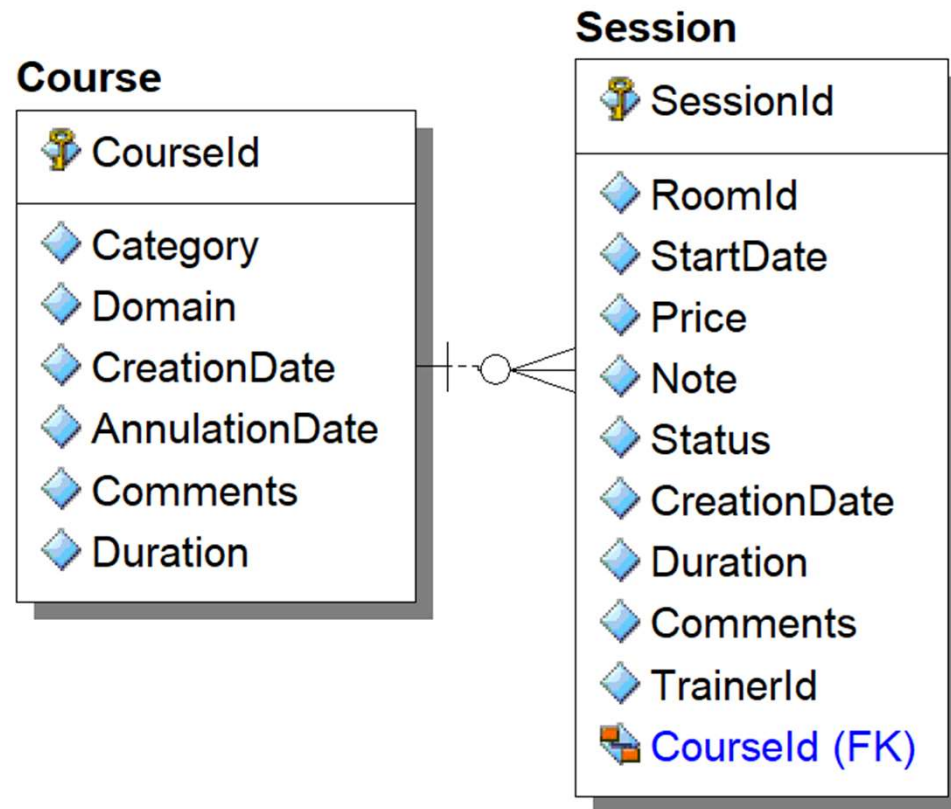
**Don't trust the layers**  
Go to SQL



**Help the optimizer**  
Look at the query plan



# The Bad Session Problem



# Session Duration

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
StartDate				Duration 5					



# Session Duration

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
StartDate				Duration 5					

5 day session



# How to Find Bad Sessions?

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Session

Session





## Date and Time Functions

`DATEPART(WeekDay, StartDate)`

`DATEADD(day, -5, StartDate)`



# Move the Sessions

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

**5 day session**

**3 day session**



# How to Calculate the Start Date?

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

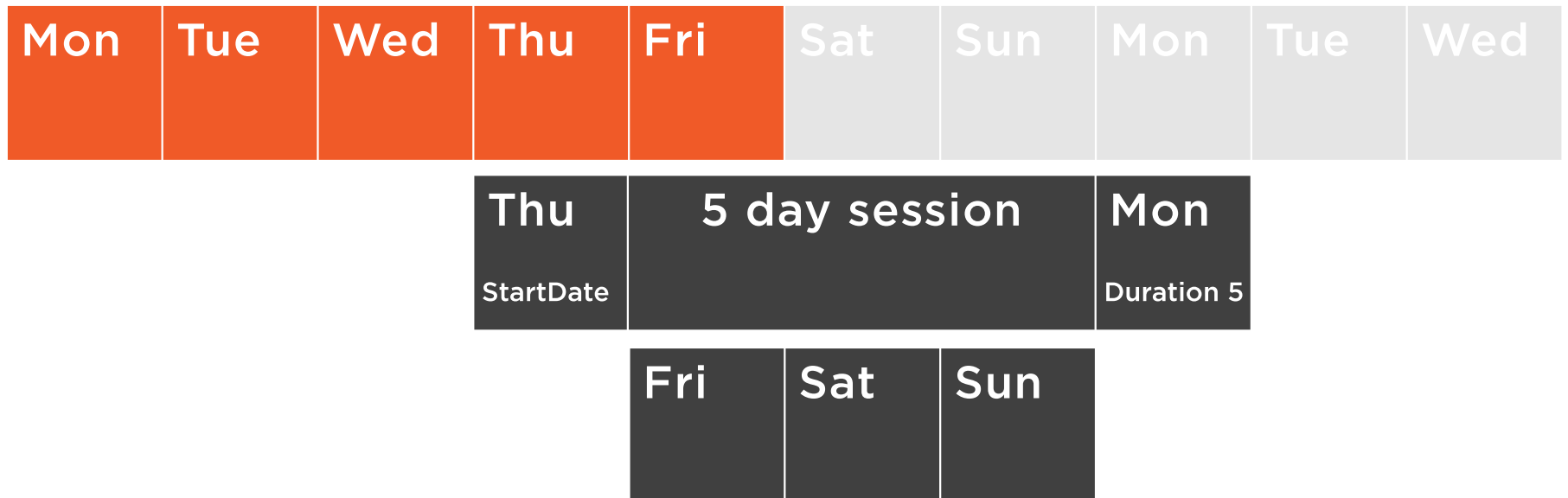
5 day session

3 day session

1	2	3	4	5	6	7
---	---	---	---	---	---	---



Is It Possible to Do Find It Declaratively?



## Join on a Tally Table

FROM Course.Session

Thu	Fri	Sat	Sun	Mon
StartDate				Duration 5



JOIN Tools.TallyDate

Thu	Fri	Sat	Sun	Mon
-----	-----	-----	-----	-----

ON d.[Day] BETWEEN s.StartDate  
AND DATEADD(DAY, Duration + 1, StartDate)



# The Concatenation Challenge

Course	Schedule
Firebird 2.0 Administration	1/11/2017, 1/12/2017, 4/11/2017, 4/18/2017, 6/14/2017, 7/6/2017, 8/29/2017, 10/9/2017, 1/1/2018
Firebird 2.0 Advanced	2/2/2017, 5/30/2017, 8/3/2017, 10/24/2017, 6/26/2018
Firebird 2.0 Basics	3/9/2017, 5/3/2017, 6/5/2017, 9/5/2017, 11/24/2017, 11/29/2017, 2/1/2018
Firebird 2.0 Essential	9/26/2017
Firebird 2.0 Expert	5/30/2017, 9/8/2017, 1/23/2018
Firebird 2.0 Performances	2/2/2017, 2/22/2017, 4/5/2017, 6/7/2017, 6/21/2017, 12/18/2017, 1/29/2018
Firebird 2.0 Programming	3/22/2017, 10/24/2017, 2/21/2018, 4/17/2018
Firebird 2.0 Upgrade	12/27/2017, 1/18/2018, 2/12/2018, 3/8/2018



# The Concatenation Challenge

```
SELECT cl.Title as Course, s.StartDate
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
ORDER BY Title, s.StartDate;
```

```
SELECT cl.Title as Course --, ... ?
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
GROUP BY cl.Title
ORDER BY Title;
```

Course	StartDate
Firebird 2.0 Administration	2017-07-06
Firebird 2.0 Administration	2017-04-11
Firebird 2.0 Administration	2017-01-11
Firebird 2.0 Administration	2017-01-12
Firebird 2.0 Administration	2017-04-18
Firebird 2.0 Administration	2017-08-29
Firebird 2.0 Administration	2018-01-01
Firebird 2.0 Administration	2017-10-09
Firebird 2.0 Administration	2017-06-14
Firebird 2.0 Advanced	2018-06-26
Firebird 2.0 Advanced	2017-10-24
Firebird 2.0 Advanced	2017-08-03
Firebird 2.0 Advanced	2017-05-30
Firebird 2.0 Advanced	2017-02-02
Firebird 2.0 Basics	2017-05-03
Firebird 2.0 Basics	2017-11-29
Firebird 2.0 Basics	2017-09-05
Firebird 2.0 Basics	2018-02-01
Firebird 2.0 Basics	2017-06-05



# The Concatenation Challenge

```
SELECT cl.Title as Course, s.StartDate
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
ORDER BY Title, s.StartDate;
```

```
SELECT cl.Title as Course, MIN(StartDate)
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
GROUP BY cl.Title
ORDER BY Title;
```

Course	StartDate
Firebird 2.0 Administration	2017-07-06
Firebird 2.0 Administration	2017-04-11
Firebird 2.0 Administration	2017-01-11
Firebird 2.0 Administration	2017-01-12
Firebird 2.0 Administration	2017-04-18
Firebird 2.0 Administration	2017-08-29
Firebird 2.0 Administration	2018-01-01
Firebird 2.0 Administration	2017-10-09
Firebird 2.0 Administration	2017-06-14
Firebird 2.0 Advanced	2018-06-26
Firebird 2.0 Advanced	2017-10-24
Firebird 2.0 Advanced	2017-08-03
Firebird 2.0 Advanced	2017-05-30
Firebird 2.0 Advanced	2017-02-02
Firebird 2.0 Basics	2017-05-03
Firebird 2.0 Basics	2017-11-29
Firebird 2.0 Basics	2017-09-05
Firebird 2.0 Basics	2018-02-01
Firebird 2.0 Basics	2017-06-05





# The Concatenation Challenge

```
SELECT cl.Title as Course, s.StartDate
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
ORDER BY Title, s.StartDate;
```

```
SELECT cl.Title as Course, MAX(StartDate)
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
GROUP BY cl.Title
ORDER BY Title;
```

Course	StartDate
Firebird 2.0 Administration	2017-07-06
Firebird 2.0 Administration	2017-04-11
Firebird 2.0 Administration	2017-01-11
Firebird 2.0 Administration	2017-01-12
Firebird 2.0 Administration	2017-04-18
Firebird 2.0 Administration	2017-08-29
Firebird 2.0 Administration	2018-01-01
Firebird 2.0 Administration	2017-10-09
Firebird 2.0 Administration	2017-06-14
Firebird 2.0 Advanced	2018-06-26
Firebird 2.0 Advanced	2017-10-24
Firebird 2.0 Advanced	2017-08-03
Firebird 2.0 Advanced	2017-05-30
Firebird 2.0 Advanced	2017-02-02
Firebird 2.0 Basics	2017-05-03
Firebird 2.0 Basics	2017-11-29
Firebird 2.0 Basics	2017-09-05
Firebird 2.0 Basics	2018-02-01
Firebird 2.0 Basics	2017-06-05



# The Concatenation Challenge

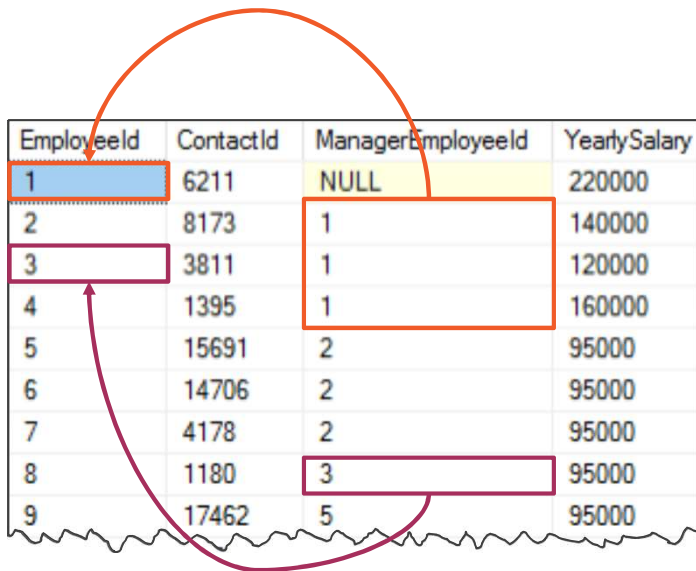
```
SELECT cl.Title as Course, s.StartDate
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
ORDER BY Title, s.StartDate;
```

```
SELECT cl.Title as Course, STRING_AGG(StartDate)
FROM Course.CourseLanguage AS cl
JOIN Course.Session as s
    ON cl.CourseId = s.CourseId
    AND cl.LanguageCd = s.LanguageCd
WHERE Title LIKE 'Firebird%'
GROUP BY cl.Title
ORDER BY Title;
```

Course	StartDate
Firebird 2.0 Administration	2017-07-06
Firebird 2.0 Administration	2017-04-11
Firebird 2.0 Administration	2017-01-11
Firebird 2.0 Administration	2017-01-12
Firebird 2.0 Administration	2017-04-18
Firebird 2.0 Administration	2017-08-29
Firebird 2.0 Administration	2018-01-01
Firebird 2.0 Administration	2017-10-09
Firebird 2.0 Administration	2017-06-14
Firebird 2.0 Advanced	2018-06-26
Firebird 2.0 Advanced	2017-10-24
Firebird 2.0 Advanced	2017-08-03
Firebird 2.0 Advanced	2017-05-30
Firebird 2.0 Advanced	2017-02-02
Firebird 2.0 Basics	2017-05-03
Firebird 2.0 Basics	2017-11-29
Firebird 2.0 Basics	2017-09-05
Firebird 2.0 Basics	2018-02-01
Firebird 2.0 Basics	2017-06-05



# The Recursive Headache



EmployeeId	ContactId	ManagerEmployeeId	YearlySalary
1	6211	NULL	220000
2	8173	1	140000
3	3811	1	120000
4	1395	1	160000
5	15691	2	95000
6	14706	2	95000
7	4178	2	95000
8	1180	3	95000
9	17462	5	95000

ManagerEmployeeId **int** NULL **FOREIGN KEY**  
**REFERENCES** Contact.Employee (EmployeeId)

```
SELECT *  
FROM Contact.Employee e1  
JOIN Contact.Employee e2  
    ON e1.EmployeeId = e2.ManagerEmployeeId  
WHERE e1.ManagerEmployeeId IS NULL;
```

```
SELECT *  
FROM Contact.Employee e1  
JOIN Contact.Employee e2  
    ON e1.EmployeeId = e2.ManagerEmployeeId  
JOIN Contact.Employee e3  
    ON e2.EmployeeId = e3.ManagerEmployeeId  
...
```



# The Window Functions Treasure



## The Logical Evaluation Order

```
var cts = 1 from cn in contacts  
          2 where cn.LastName == "Smith"  
          3 select new {  
              FirstName,  
              LastName };
```




## The Logical Evaluation Order

```
5 SELECT FirstName, LastName  
1 FROM Contact.Contact  
2 WHERE LastName LIKE 'S%'  
3 GROUP BY FirstName, LastName  
4 HAVING COUNT(*) = 1  
6 ORDER BY FirstName, LastName;
```



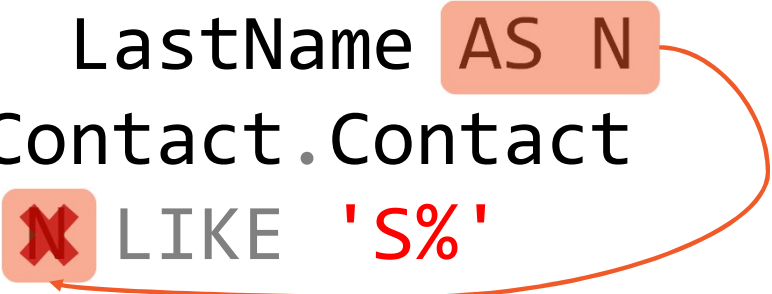
## The Logical Evaluation Order

```
SELECT FirstName,  
       LastName AS N  
FROM Contact.Contact  
WHERE LastName LIKE 'S%'  
GROUP BY FirstName, LastName  
HAVING COUNT(*) = 1  
ORDER BY FirstName, N;
```



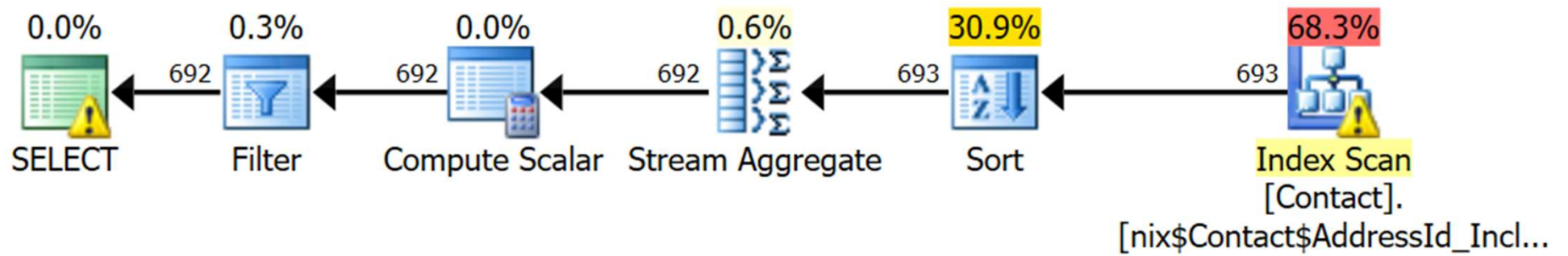
## The Logical Evaluation Order

```
SELECT FirstName,  
        LastName AS N  
FROM Contact.Contact  
WHERE X LIKE 'S%'  
GROUP BY FirstName, LastName  
HAVING COUNT(*) = 1  
ORDER BY FirstName, N ;
```

A diagram illustrating the logical evaluation order of the SQL query. It consists of two orange rectangular boxes. The first box contains the text 'AS N' and is positioned over the 'AS N' part of the 'LastName AS N' clause. The second box contains a red 'X' over the 'WHERE' clause. An orange curved arrow originates from the 'AS N' box and points to the 'WHERE' clause box, indicating that the alias 'N' is evaluated before the 'WHERE' clause.



# Query Plan



## Grouping

```
SELECT FirstName  
FROM Contact.Contact  
GROUP BY FirstName  
ORDER BY FirstName;
```



Hash Match  
(Aggregate)  
Cost: 61 %

FirstName
Abel
Abel
Abel
Abel
Abel
Abel
Abel
Abel
Abel
Achille
Achille
Achille
Achille
Achille
Achille

FirstName
Abel
Achille



# The Parts of the OVER Clause T-SQL Window Functions

by Kathi Kellenberger

Learn to use the versatile T-SQL functions called "window functions" to solve complex queries.

Renew Subscription



Bookmark



Add to Channel



Live mentoring

Course author



Kathi Kellenberger

Kathi Kellenberger is a SQL Server MVP and a Teammate with Linchpin People. She is the author or co-author of several SQL Server books, including Beginning T-SQL from Apress, and frequently speaks...

Course info

Level Intermediate

Rating ★★★★★ (206)

Duration 2h 31m

Released 8 Jul 2015

Share course

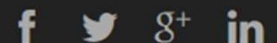


Table of contents Description Exercise files Recommended

Expand all

▶ Getting Started	✓	🔖	10m 48s	▼
▶ Understanding the OVER Clause	✓	🔖	11m 11s	▼
▶ Using Ranking Functions	✓	🔖	25m 34s	▼
▶ Using Window Aggregate Functions	✓	🔖	26m 34s	▼
▶ Understanding Framing	✓	🔖	13m 58s	▼

# The Window Functions Treasure

```
SELECT InvoiceId, InvoiceDate, Amount
FROM Enrollment.Invoice
WHERE InvoiceDate = '2017-06-26';
```

InvoiceId	InvoiceDate	Amount
F003611	2017-06-26	2990.00
F003612	2017-06-26	2990.00
F003613	2017-06-26	2990.00
F003614	2017-06-26	1794.00
F003615	2017-06-26	2990.00

```
SELECT InvoiceId, InvoiceDate, Amount,
       MIN(Amount) AS MinAmount
FROM Enrollment.Invoice
WHERE InvoiceDate = '2017-06-26';
```

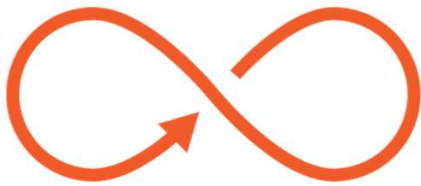
Msg 8120, Level 16, State 1, Line 21  
Column 'Enrollment.Invoice.InvoiceId' is  
invalid in the select list because it is not  
contained in either an aggregate function or  
the GROUP BY clause.

```
SELECT InvoiceId, InvoiceDate, Amount,
       MIN(Amount)
       OVER (PARTITION BY InvoiceDate)
       AS MinAmount
FROM Enrollment.Invoice
WHERE InvoiceDate = '2017-06-26';
```

F003621	2017-06-26	2990.00
F003622	2017-06-26	1794.00
F003623	2017-06-26	2990.00
F003624	2017-06-26	1794.00
F003625	2017-06-26	2990.00
F003626	2017-06-26	2990.00
F003627	2017-06-26	2990.00
F003628	2017-06-26	2990.00
F003629	2017-06-26	16146.00



# Conclusion



Join the no-loop  
movement



SQL tricks are our  
little treasures



SQL is a limited  
language



SQL syntax is a  
delicacy!



Have we said  
“declarative”?



Learn window  
functions

