

SQL Server Is the best  
RDBMS

I love you, will you  
marry me?

# Time Passes

SQL Server, You're being  
kind of immature

SQL Server, Why are you  
running up our credit  
card?

SQL Server, what is with  
your friend Sharepoint?

Maybe I should see what  
else is out there

SQL Server, I'm filing for  
divorce

# Intruder Alert

- You can call me Rob
- SQL Server DBA
- Dictator-Base Administrator
- Douche-Bag Administrator
- Better than the SAN Admin



You better  
have a license  
for that DB Rob



# Religious Debates

- Microsoft / Oracle both have people called “Evangelists”
- Your backend isn’t a religion



# Which is best?

## SQL Server vs Postgres

- Not the question we are trying to answer
- Less emotion, more pragmatism
- They are just databases man

# But Postgres is open “sores”



# Visualization Helps

Postgres



SQL Server



# Blinding Speed of Closed

- 3 year or so release cycle for SQL Server
- Oh you need something? Don't call us, we'll call you

# Why I choose Windows

- I hate freedom and filesystem choices
- I need a Browser, Silverlight and Games folder on my DB Server
- RAM grows on trees
- What is an “SSH” ?



# Installing SQL Server is easy

- The installer has an installer
- Only 22 simple screens
- Postgres has Schemaverse... SQL Server has its installer

# Installing Postgres is Hard

- `apt-get install -y postgres`
- Press Return (slang for Enter)

# Microsoft Licensing

- Disclaimer - Confusing Slide Ahead

# I've got 99 Invoices but Postgres Ain't One

- 16 Logical Core box
- (# of cores \* Core Factor Table) / 2 for core pack pricing
- $(16 * 1) / 2 * \$13,300 = 106,400$
- \$1,525 Windows Server license
- \$107,925 - O.M.F.G

# But wait, there's more

- Now we buy 3rd party tools to remove the suck

# MS - It's your (de)fault

- Defaults to 2048TBs RAM usage
- Won't compress backups
- Send alerts to your Pager!
- Send alerts via Netsend (deprecated)

# Functions and Indexes

- Index only Queries
- Index on a function

# Hold the Line

- I have to manually update stats
- I have to manually defrag indexes
- How I deal with indexes depends on how much I have spent on SQL Server
- And I have little control over all of this

# Backup the Truck

SQL Server:

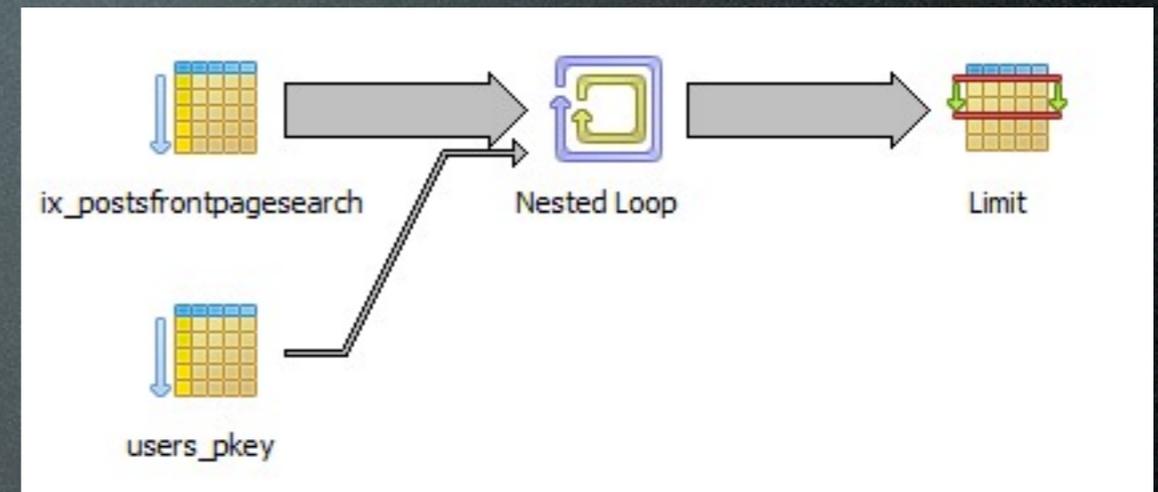
```
BACKUP DATABASE [pgcon]  
TO DISK = 'NULL'  
WITH FORMAT, COMPRESSION
```

Postgres:



# Query Tune Postgres

```
previous queries  
explain select p.postid, p.answercount,p.viewcount,p.title  
,p.tags,u.userid,u.displayname,u.reputation  
from posts p inner join users u on p.owneruserid = u.userid  
where p.posttypeid = 1  
order by p.creationdate desc  
limit 20
```

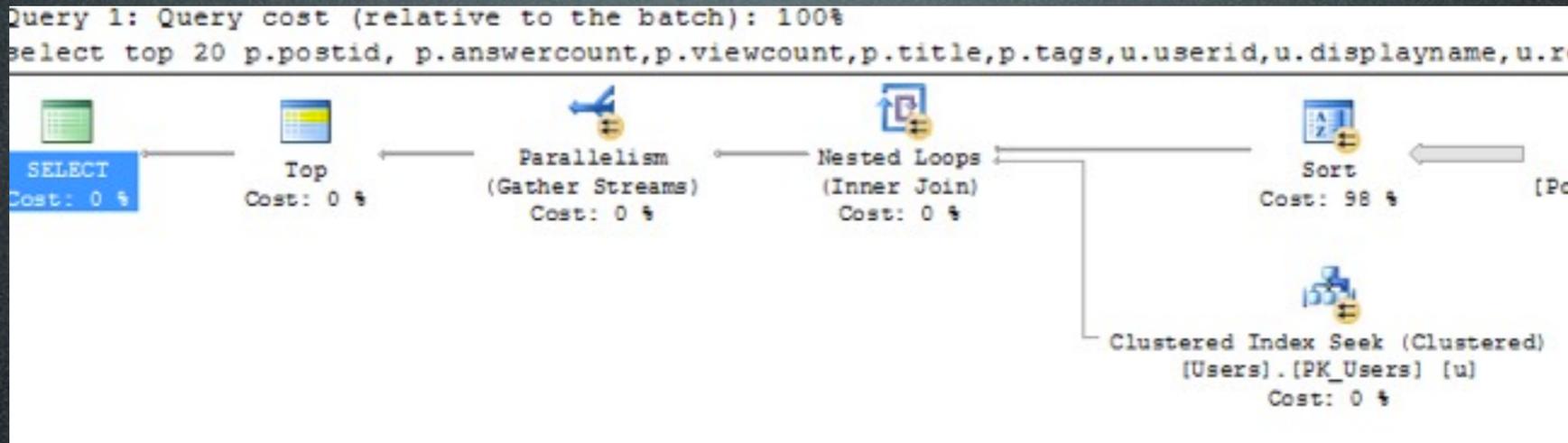


Output pane

Data Output Explain Messages History

	QUERY PLAN text
1	Limit (cost=0.00..232.36 rows=20 width=115)
2	-> Nested Loop (cost=0.00..17074899.15 rows=1469699 width=115)
3	-> Index Scan Backward using ix postsfrontpagesearch on posts p (cost=0.00..5073782.42 rows=1492737 width=115)
4	Index Cond: (posttypeid = 1)
5	-> Index Scan using users pkey on users u (cost=0.00..8.03 rows=1 width=15)
6	Index Cond: (userid = p.owneruserid)

# Query Tune SQL Server



```
--Top (TOP EXPRESSION: ((20)))  
|--Parallelism(Gather Streams, ORDER BY: ([p].[CreationDate] DESC))  
  |--Nested Loops (Inner Join, OUTER REFERENCES: ([p].[OwnerUserId], [Expr1004]) WITH O  
    |--Sort (ORDER BY: ([p].[CreationDate] DESC))  
      |--Index Seek (OBJECT: ([stackoverfaux].[dbo].[Posts].[IX_Posts_PostTypeID_  
        |--Clustered Index Seek (OBJECT: ([stackoverfaux].[dbo].[Users].[PK_Users] AS [u
```

```
<Statements>  
<StmtSimple StatementText="&#xD;&#xA;select top 20 p.postid, p.  
<StatementSetOptions QUOTED_IDENTIFIER="true" ARITHABORT="true"  
<QueryPlan CachedPlanSize="40" CompileTime="10" CompileCPU="10"  
<RelOp NodeId="0" PhysicalOp="Top" LogicalOp="Top" Estimated  
<OutputList>  
<ColumnReference Database="[stackoverfaux]" Schema="[dbo]">  
</OutputList>  
<Top RowCount="0" IsPercent="0" WithTies="0">  
<TopExpression>  
<ScalarOperator ScalarString="(20)">  
<Const ConstValue="(20)" />  
</ScalarOperator>  
</TopExpression>  
<RelOp NodeId="1" PhysicalOp="Parallelism" LogicalOp="G  
<OutputList>  
<ColumnReference Database="[stackoverfaux]" Schema="[dbo]">  
<ColumnReference Database="[stackoverfaux]" Schema="[dbo]">  
<ColumnReference Database="[stackoverfaux]" Schema="[dbo]">  
<ColumnReference Database="[stackoverfaux]" Schema="[dbo]">
```

# Are all tables created Equally?

Config Params

Session Params

Postgres adapts as my data adapts

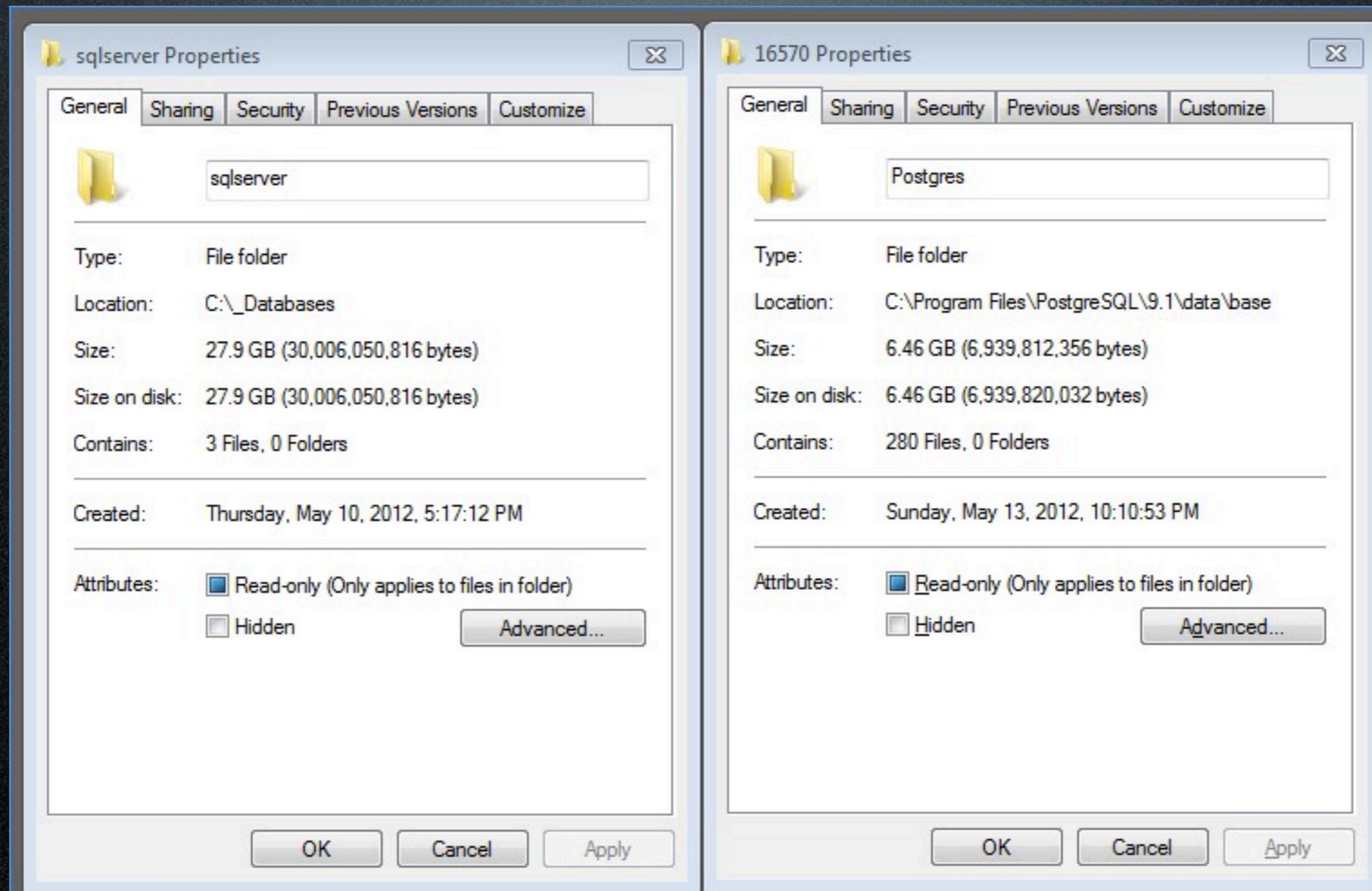
# Modern Development

- Postgres 9.2 has a JSON Type
- Postgres has V8 Module
- This is hot and sexy in the fast world of App-Devery

# What edition of SQL Server has V8 or JSON?

- Upcoming feature in SQL Server - “Nein”

# Hungry for TOAST!



# Data Types!

## Postgres Timestamp

```
create table blah (rowid int, ummm timestamp default now());
insert into blah (rowid) values (1);
Select * from blah;
```

rowid	ummm
integer	timestamp without time zone
1	2012-05-17 09:38:47.0839

## SQL Server Timestamp

```
SQLQuery3.sql - lo...ex\rsullivan (54)* SQLQuery2.sql - lo...ex\rsullivan
create table blah (rowid int, ummm timestamp)
insert into blah (rowid) values (1)
Select * from blah
```

rowid	ummm
1	1
	0x000000000000009C43

Born on the 4th of  
0x0000000000000000000000009C43!

# Speaking of Timestamps

Infinity... WHOA!

All Balls... WHAT?!

# But Rob, SQL Server is a “Data Platform”

- SQL Server Engine
- ETL Tools
- Reporting Tools (SSRS)
- Analysis Tools (SSAS)

# Tech World Moves Fast

- FDW can help with ETL
- Reporting is a fairly solvable problem
- SSRS is not all that

# Analysis is interesting

- We need distributed, parallel, fast execution on gigantic aggregates
- Google CitusData, email Umur and call it a day!

# What about our app?

- Drivers / ORMs and other gotchas
- How hard can it be to go to Linux?

Shall we look at code?

What else makes Postgres  
Great?

YOU

Thank you sponsors!

@datachomp