SQL
KONFERENZ
2026



Questioning my SQL Server Faith

... so you don't have to

Gianluca Sartori



Founder of

QUANTUMDATIS

Independent SQL Server consultant

Data platform MVP

Works with SQL Server since 7.0

DBA @ Scuderia Ferrari

@spaghettidba.bsky.social

spaghettidba.com

quantumdatis.com

SQL KONFERENZ 2026



AGENDA

- A BIT OF HISTORY
- THE CHALLENGE
- SQL SERVER FIGHTS BACK
- CONCLUSIONS



SQL
KONFERENZ
2026

A BIT OF HISTORY

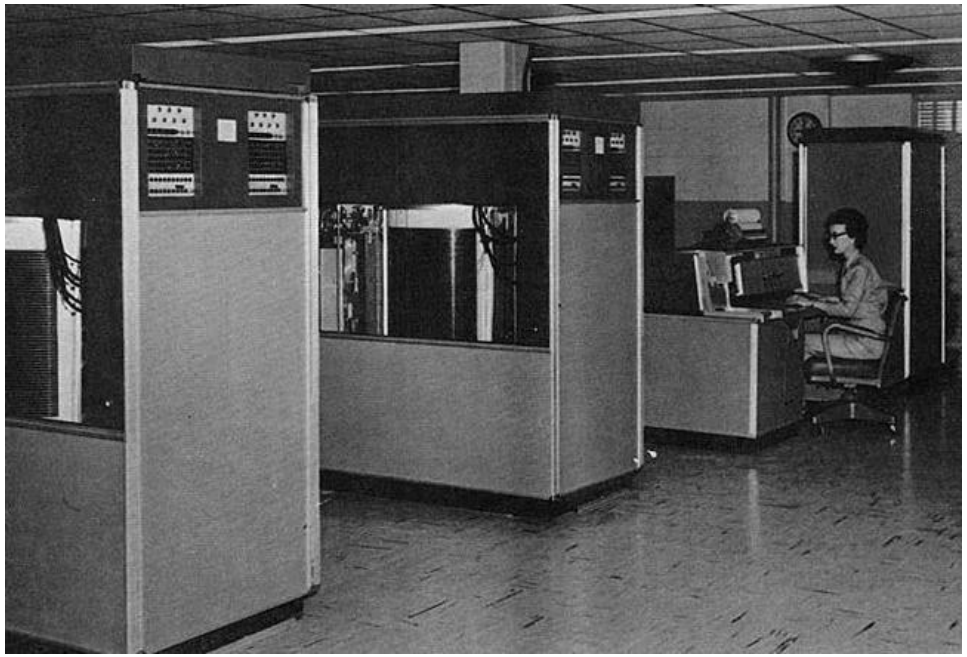


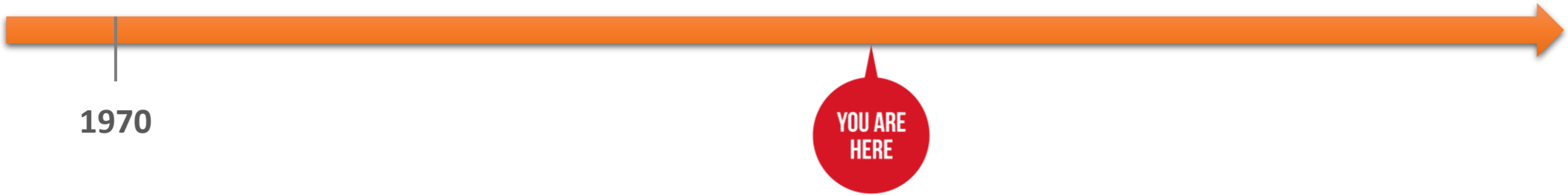


A BIT OF HISTORY

The seventies

- Early databases were ridden with inconsistencies and hard to work with
- Hardware was bulky and expensive
- Edgar Codd creates the Relational Model





A BIT OF HISTORY

The eighties

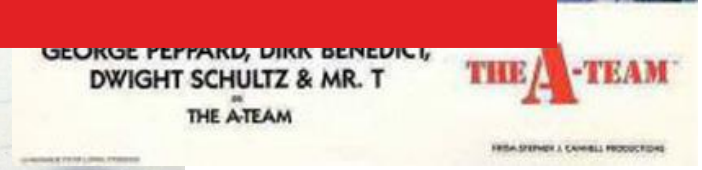
- Oracle dominates the scene
- Other players are slowly entering the market





ORACLE®



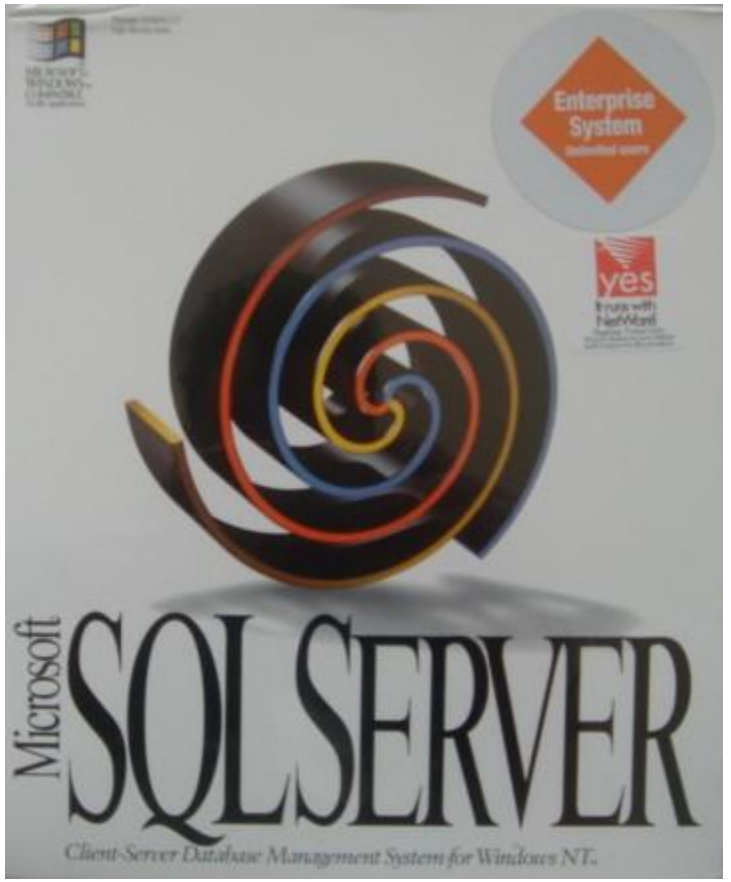


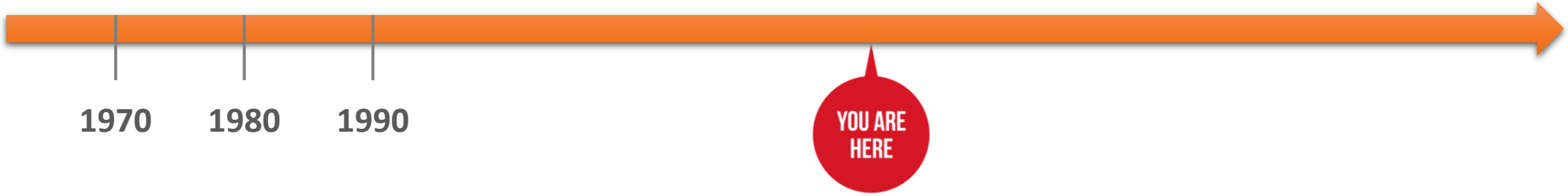
1970 1980

A BIT OF HISTORY

The nineties

- 1989 – SQL Server 1.0
- 1992 – SQL Server 4.2 on WINNT
- 1994 – Postgres95
- 1995 - MySQL
- 1998 – SQL Server 7
- Oracle still dominates the market

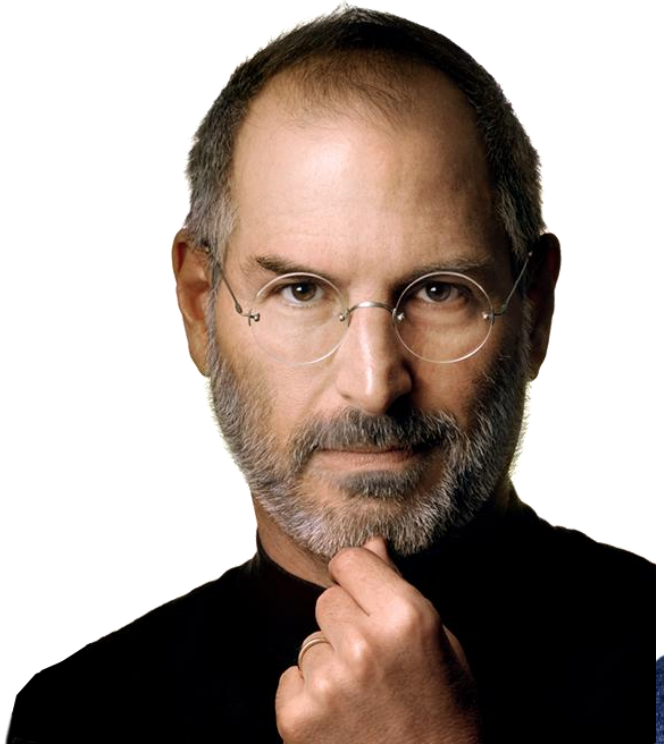


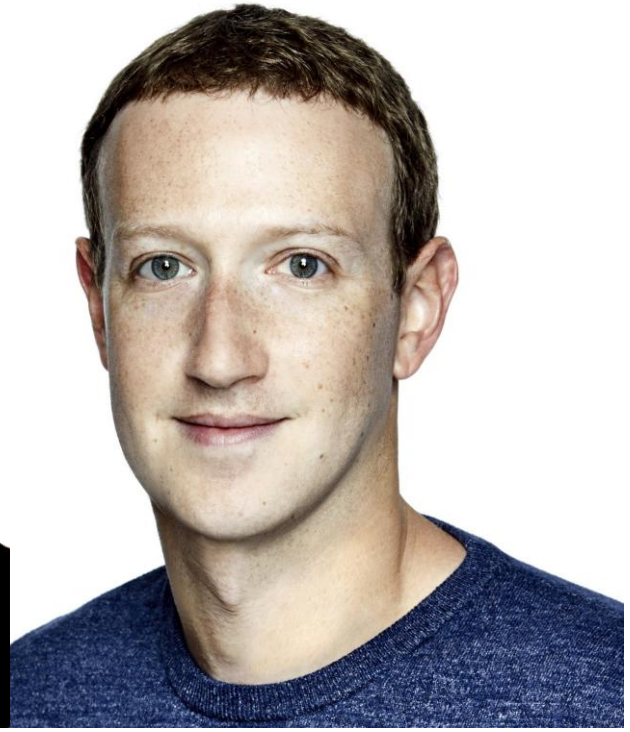


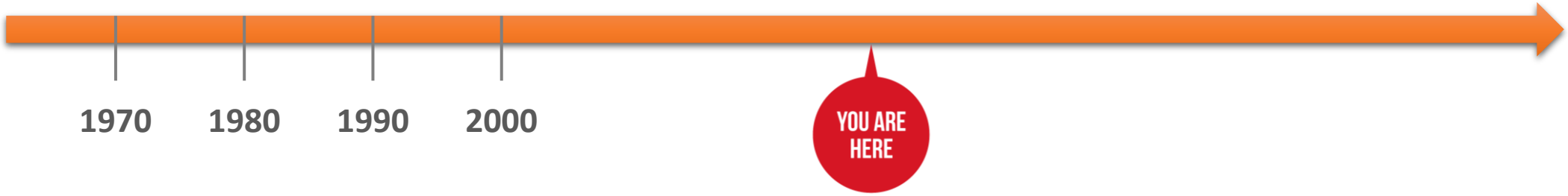
A BIT OF HISTORY

The 2000s

- 1998 – Google
- 2003 - Myspace
- 2003 – Facebook
- 2005 - YouTube
- 2006 – Twitter
- 2007 - iPhone
- 2009 - WhatsApp





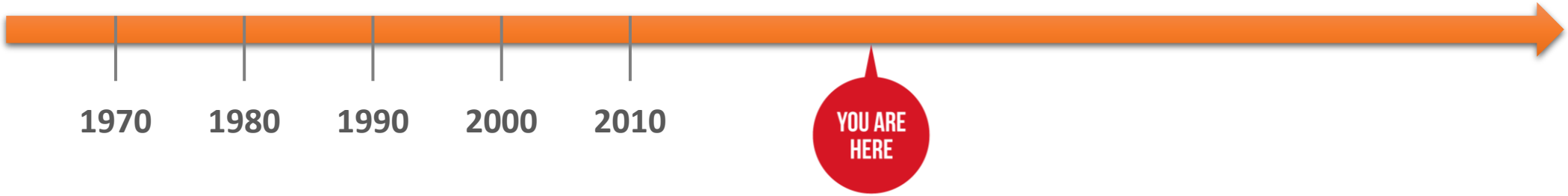


A BIT OF HISTORY

The 2010s

- NoSQL databases emerge as contenders to relational databases
- SQL databases start to incorporate NoSQL features
- The cloud makes enterprise grade features available to companies of all sizes



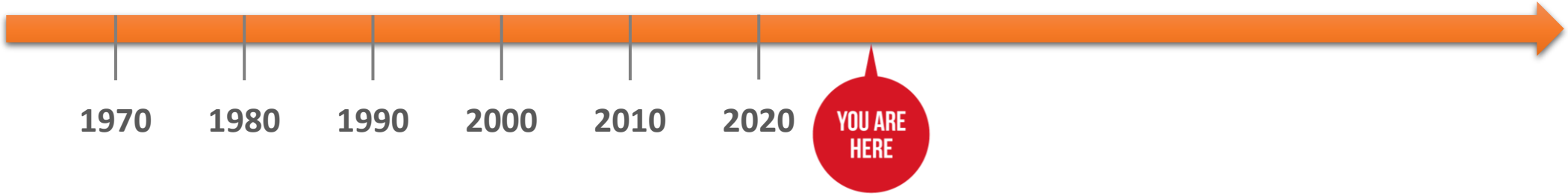


A BIT OF HISTORY

The 2020s

- Scale of data continues to grow
- AI on the rise
- SQL databases incorporate AI features
  - vector embeddings
  - AI driven self-tuning





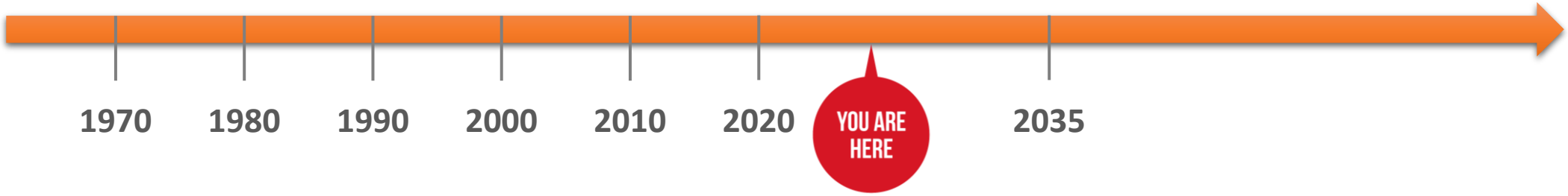
A BIT OF HISTORY

The Future

What are we looking for in a database:

- Cloud-native
- Seamlesss horizontal scaling
- AI integration
- Real-time analytics







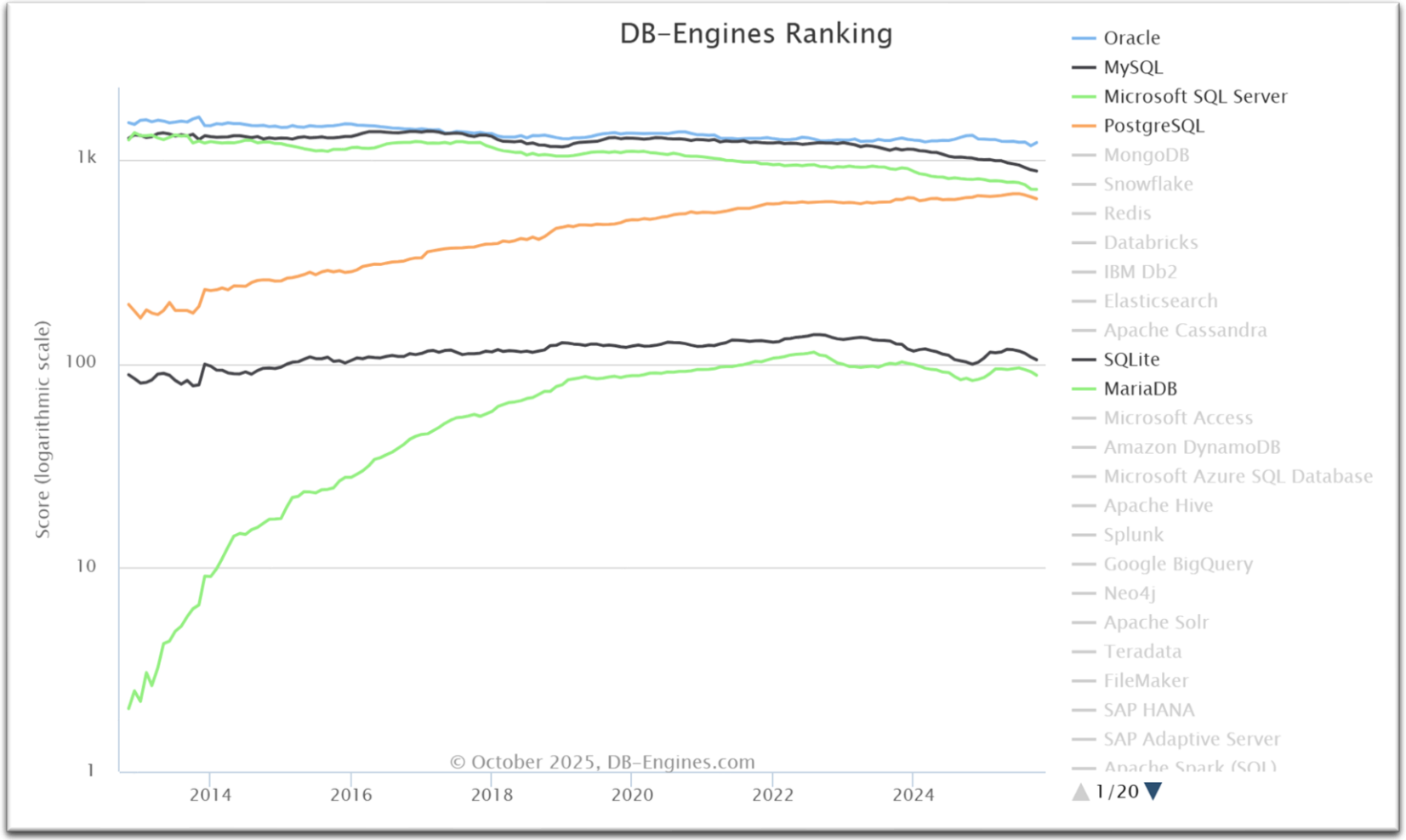
CURRENT TRENDS

Is SQL Server's dominance declining?

Rank			DBMS	Database Model	Score		
Oct 2025	Sep 2025	Oct 2024			Oct 2025	Sep 2025	Oct 2024
1.	1.	1.	Oracle	Relational, Multi-model	1212.77	+42.15	-96.67
2.	2.	2.	MySQL	Relational, Multi-model	879.66	-12.11	-143.09
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	715.05	-2.27	-87.04
4.	4.	4.	PostgreSQL	Relational, Multi-model	643.20	-13.97	-8.96
5.	5.	5.	MongoDB	Document, Multi-model	368.01	-12.49	-37.20
6.	6.	↑ 7.	Snowflake	Relational	198.65	+8.46	+58.05
7.	7.	↓ 6.	Redis	Key-value, Multi-model	142.33	-2.84	-7.30
8.	↑ 9.	↑ 14.	Databricks	Multi-model	128.80	+4.74	+43.21
9.	↓ 8.	9.	IBM Db2	Relational, Multi-model	122.37	-1.81	-0.40
10.	10.	↓ 8.	Elasticsearch	Multi-model	116.67	-1.59	-15.18

CURRENT TRENDS

Is SQL Server's dominance declining?



SQL
KONFERENZ
2026

THE CHALLENGE



THE ELEPHANT IN THE ROOM

PostgreSQL's plans to rule the world

- Fastest growing open-source database
- DBMS of the year in 2023
- Offers enterprise-grade features
- Rich set of extensions
- Modern SQL feature (JSON, JSONB,RegExp...)
- Cloud-Native Ecosystem
- Vendor Neutrality
- ... will it rule the world?



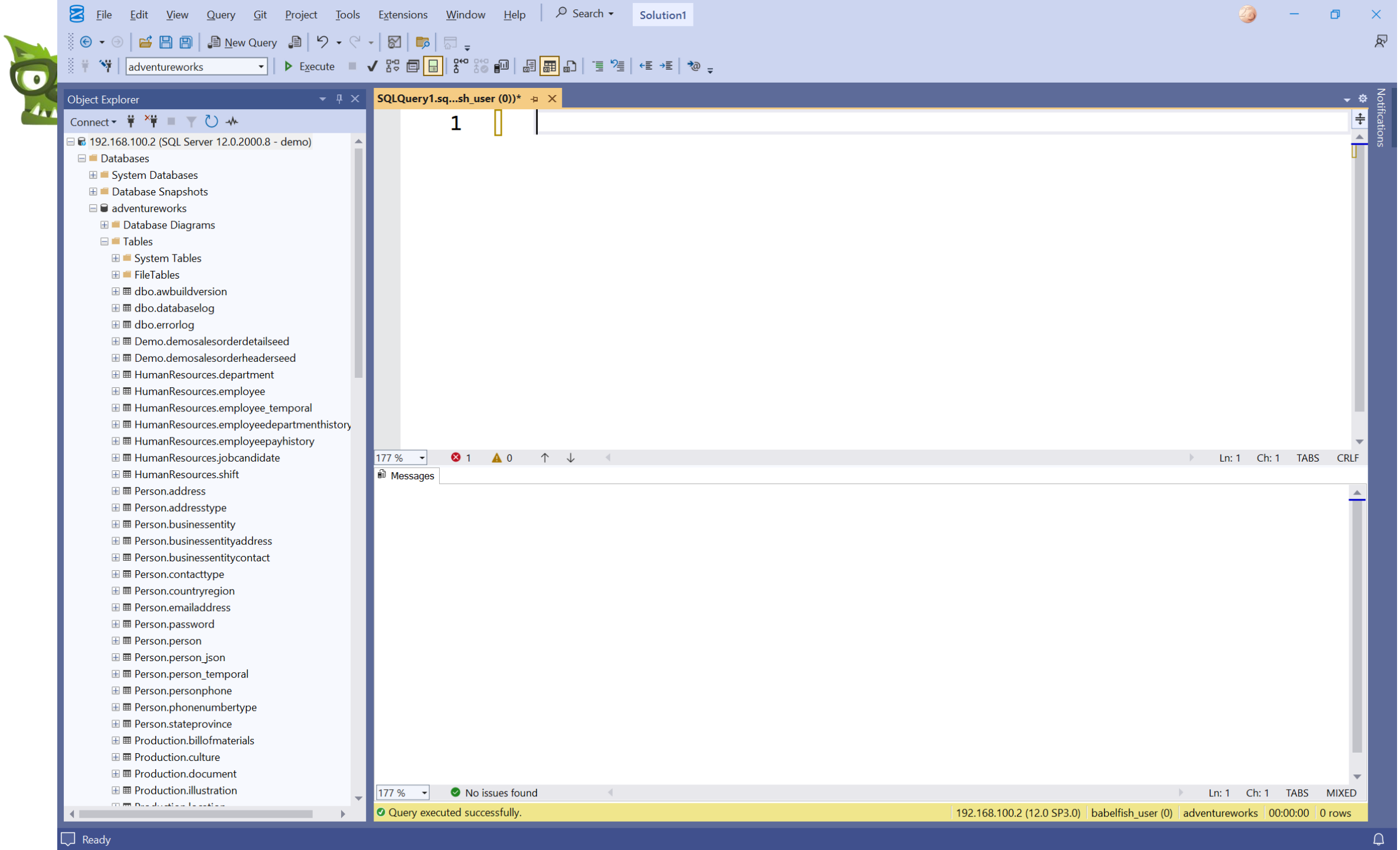
FRIENDS TURN TO FOES

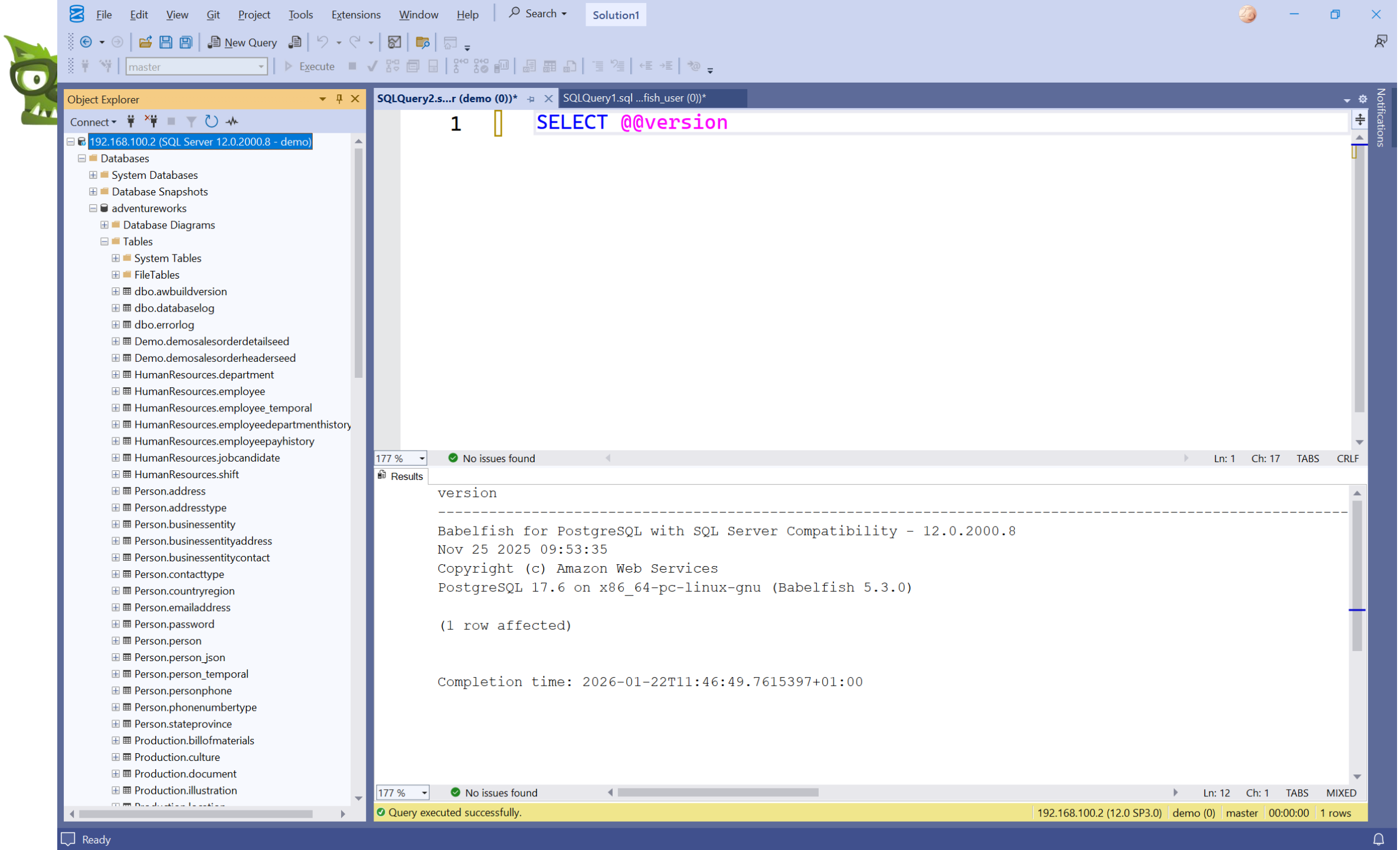









File Edit View Query Git Project Tools Extensions Window Help
Search
Solution1
New Query
adventureworks
Execute
Object Explorer
Connect
192.168.100.2 (SQL Server 12.0.2000.8 - demo)
Databases
System Databases
Database Snapshots
adventureworks
Database Diagrams
Tables
System Tables
FileTables
dbo.awbuildversion
dbo.databaselog
dbo.errorlog
Demo.demosalesorderdetailseed
Demo.demosalesorderheaderseed
HumanResources.department
HumanResources.employee
HumanResources.employee_temporal
HumanResources.employeedepartmenthistory
HumanResources.employeepayhistory
HumanResources.jobcandidate
HumanResources.shift
Person.address
Person.addresstype
Person.businessentity
Person.businessentityaddress
Person.businessentitycontact
Person.contacttype
Person.countryregion
Person.emailaddress
Person.password
Person.person
Person.person_json
Person.person_temporal
Person.personphone
Person.phonenumbertype
Person.stateprovince
Production.billofmaterials
Production.culture
Production.document
Production.illustration
SQLQuery1.sq...sh_user (0))*
1
177 %
1
0
Ln: 1 Ch: 1 TABS CRLF
Messages
177 %
No issues found
Ln: 1 Ch: 1 TABS MIXED
Query executed successfully.
192.168.100.2 (12.0 SP3.0)
babelfish_user (0)
adventureworks
00:00:00
0 rows
Notifications
Ready


File Edit View Git Project Tools Extensions Window Help
Search
Solution1
New Query
master
Execute
Object Explorer
Connect
192.168.100.2 (SQL Server 12.0.2000.8 - demo)
Databases
System Databases
Database Snapshots
adventureworks
Database Diagrams
Tables
System Tables
FileTables
dbo.awbuildversion
dbo.databaselog
dbo.errorlog
Demo.demosalesorderdetailseed
Demo.demosalesorderheaderseed
HumanResources.department
HumanResources.employee
HumanResources.employee_temporal
HumanResources.employeedepartmenthistory
HumanResources.employeepayhistory
HumanResources.jobcandidate
HumanResources.shift
Person.address
Person.addresstype
Person.businessentity
Person.businessentityaddress
Person.businessentitycontact
Person.contacttype
Person.countryregion
Person.emailaddress
Person.password
Person.person
Person.person_json
Person.person_temporal
Person.personphone
Person.phonenumbertype
Person.stateprovince
Production.billofmaterials
Production.culture
Production.document
Production.illustration
SQLQuery2.s...r (demo (0))*
SQLQuery1.sql ...fish_user (0))*
1
SELECT @@version
177 %
No issues found
Ln: 1 Ch: 17 TABS CRLF
Results
version
Babelfish for PostgreSQL with SQL Server Compatibility - 12.0.2000.8
Nov 25 2025 09:53:35
Copyright (c) Amazon Web Services
PostgreSQL 17.6 on x86_64-pc-linux-gnu (Babelfish 5.3.0)
(1 row affected)
Completion time: 2026-01-22T11:46:49.7615397+01:00
177 %
No issues found
Ln: 12 Ch: 1 TABS MIXED
Query executed successfully.
192.168.100.2 (12.0 SP3.0)
demo (0)
master
00:00:00
1 rows
Notifications
Ready



A DIRECT THREAT TO SQL SERVER

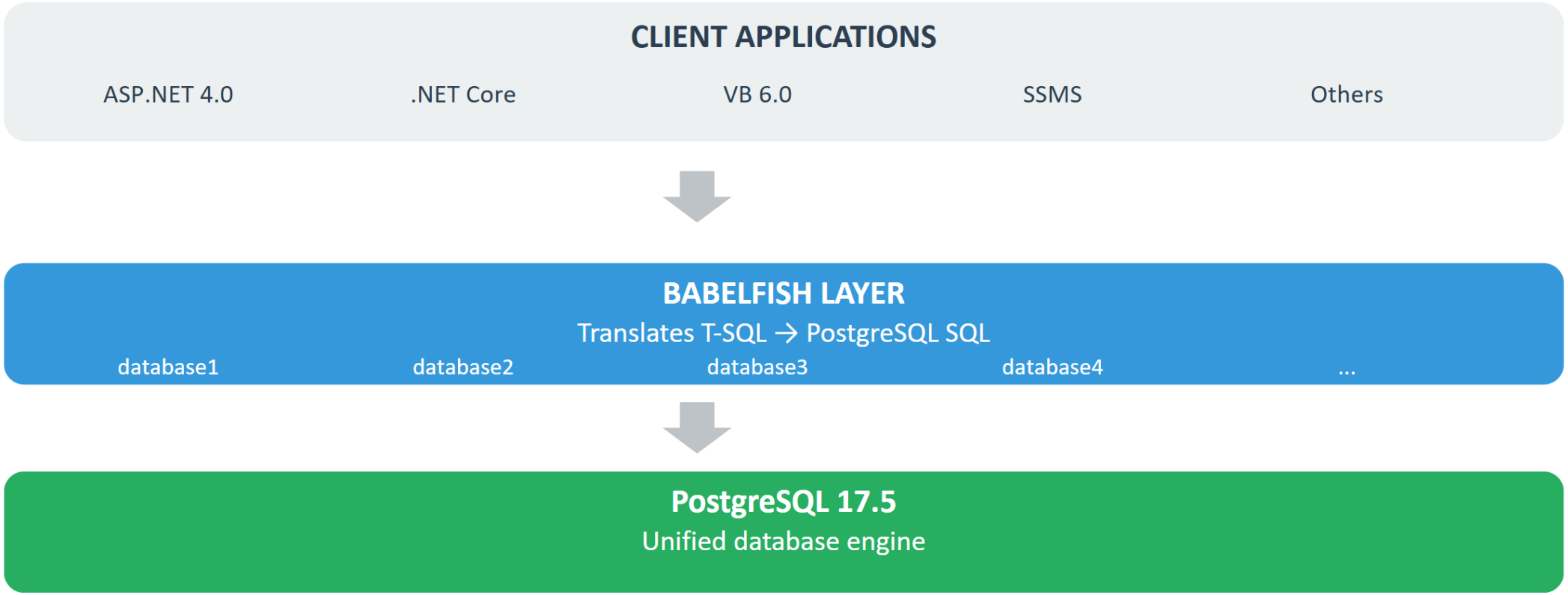
Babelfish promises drop-in replacement

- PostgreSQL extension
- Developed ad maintained by AWS
- Open-source, active development
- Speaks T-SQL over TDS
- Offers low-friction migration path from SQL Server to PostgreSQL
- Provides migration tools and docs
- Not a silver bullet



BABELFISH ABSTRACTION LAYER

You speak T-SQL, it speaks PGSQL





BABELFISH MULTI-DB MODE

Many databases in SQL Server, many schemas in PostgreSQL

T-SQL View (port 1433)	PostgreSQL View (port 5432)
Database Structure: **Multiple Databases:** • database1 (with dbo, sales, hr) • database2 (with dbo, sales, hr) • database3 (with dbo, sales, hr) **Example Query:** USE database1; SELECT * FROM dbo.customers; SELECT * FROM sales.orders; USE database2; SELECT * FROM dbo.customers;	**Database Structure:** **Database: babelfish_db (single!)** Schemas (prefixed with DB name): • database1_dbo • database1_sales • database1_hr • database2_dbo • database2_sales • database2_hr **Example Query:** SELECT * FROM database1_dbo.customers; SELECT * FROM database1_sales.orders; SELECT * FROM database2_dbo.customers;



BABELFISH SINGLE-DB MODE

One database in SQL Server, one database in PostgreSQL

T-SQL View (port 1433)	PostgreSQL View (port 5432)
Database Structure: **Databases: MyDatabase** **Schemas:** • dbo • sales • hr **Example Query:** USE database1; SELECT * FROM dbo.customers; SELECT * FROM sales.orders; SELECT * FROM hr.employees;	**Database Structure:** **Database: babelfish_db** **Schemas:** • dbo • sales • hr **Example Query:** SELECT * FROM dbo.customers; SELECT * FROM sales.orders; SELECT * FROM hr.employees;



ALL THAT GLITTERS IS GOLD?

Unsupported statements

ALTER DATABASE

ALTER DATABASE SCOPED CONFIGURATION

ALTER DATABASE SCOPED CREDENTIAL

ALTER DATABASE SET HADR

ALTER INDEX

ALTER PARTITION FUNCTION

ALTER PARTITION SCHEME

ALTER SCHEMA

ALTER SERVER CONFIGURATION

ALTER SERVICE, BACKUP/RESTORE SERVICE MASTER KEY clause

BEGIN CONVERSATION TIMER

BEGIN DISTRIBUTED TRANSACTION

BEGIN DIALOG CONVERSATION

BULK INSERT

CREATE COLUMNSTORE INDEX

CREATE EXTERNAL FILE FORMAT

CREATE EXTERNAL TABLE

CREATE, ALTER, DROP APPLICATION ROLE

CREATE, ALTER, DROP ASSEMBLY

CREATE, ALTER, DROP ASYMMETRIC KEY

CREATE, ALTER, DROP CREDENTIAL

CREATE, ALTER, DROP CRYPTOGRAPHIC PROVIDER

CREATE, ALTER, DROP ENDPOINT

CREATE, ALTER, DROP EVENT SESSION

CREATE, ALTER, DROP EXTERNAL LANGUAGE

CREATE, ALTER, DROP EXTERNAL RESOURCE POOL

CREATE, ALTER, DROP FULLTEXT CATALOG

CREATE, ALTER, DROP FULLTEXT INDEX

CREATE, ALTER, DROP FULLTEXT STOPLIST

CREATE, ALTER, DROP MESSAGE TYPE

CREATE, ALTER, DROP, OPEN/CLOSE, BACKUP/RESTORE MASTER KEY

CREATE, ALTER, DROP QUEUE

CREATE, ALTER, DROP RESOURCE GOVERNOR

CREATE, ALTER, DROP RESOURCE POOL

CREATE, ALTER, DROP ROUTE

CREATE, ALTER, DROP SEARCH PROPERTY LIST

CREATE, ALTER, DROP SECURITY POLICY

CREATE, ALTER, DROP SELECTIVE XML INDEX clause

CREATE, ALTER, DROP SERVICE

CREATE, ALTER, DROP SPATIAL INDEX

CREATE, ALTER, DROP TYPE

CREATE, ALTER, DROP XML INDEX

CREATE, ALTER, DROP XML SCHEMA COLLECTION

CREATE/DROP RULE

CREATE, DROP WORKLOAD CLASSIFIER

CREATE, ALTER, DROP WORKLOAD GROUP

ALTER TRIGGER

CREATE TABLE... GRANT clause

CREATE TABLE... IDENTITY clause

CREATE USER

DENY

END, MOVE CONVERSATION

EXECUTE with AS LOGIN or AT option

GET CONVERSATION GROUP

GROUP BY ALL clause

GROUP BY CUBE clause

GROUP BY ROLLUP clause

INSERT... DEFAULT VALUES

MERGE

READTEXT

REVERT

SELECT TOP x PERCENT WHERE x <> 100

SELECT TOP... WITH TIES

SELECT... FOR BROWSE

SELECT... FOR XML AUTO

SELECT... FOR XML EXPLICIT

SELECT... FOR XML PATH

SEND

SET DATEFORMAT

SET DEADLOCK_PRIORITY

SET FMTONLY

SET FORCEPLAN

SET NUMERIC_ROUNDABORT ON

SET OFFSETS

SET REMOTE_PROC_TRANSACTIONS

SET SHOWPLAN_TEXT

SET SHOWPLAN_XML

SET STATISTICS

SET STATISTICS PROFILE

SET STATISTICS TIME

SET STATISTICS XML

SHUTDOWN statement

UPDATE STATISTICS

UPDATETEXT

Using EXECUTE to call a SQL function

VIEW... CHECK OPTION clause

VIEW... VIEW_METADATA clause

WAITFOR DELAY

WAITFOR TIME

WAITFOR, RECEIVE

WITH XMLNAMESPACES construct

WRITETEXT

XPATH expressions

SQL
KONFERENZ
2026



SQL SERVER FIGHTS BACK

WHAT MAKES SQL SERVER SPECIAL?

The challenge

Cost

Support

Development

Maintenance

Performance

Enterprise Features



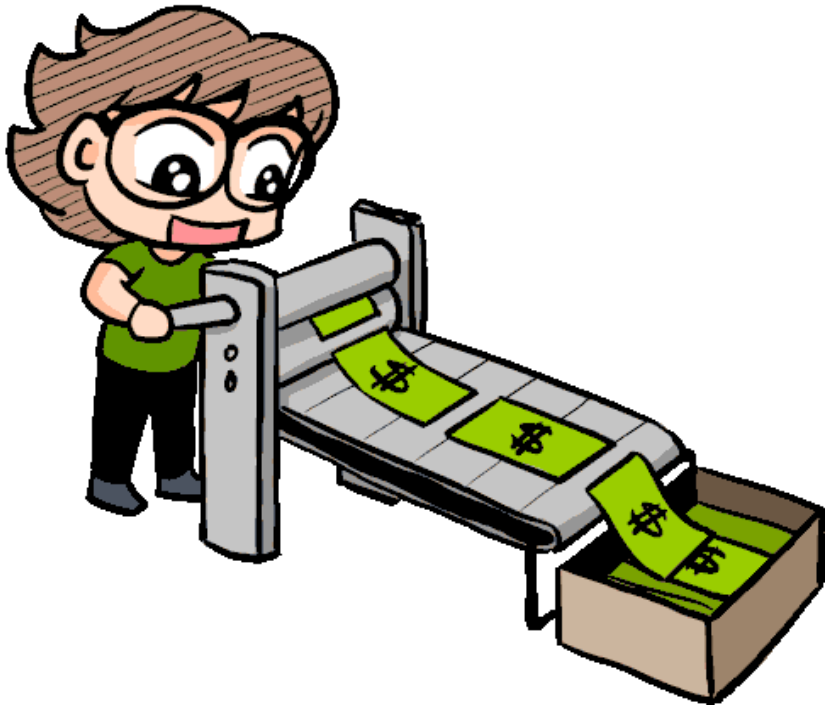


COST

Commercial vs. Open-Source

Analysis Services
Integration Services
Reporting Services
Master Data Services
Machine Learning Services

- SQL Server on-prem requires a license
  - Server + CAL — € 1.000 per server + € 250 per CAL
  - Standard Edition — € 2.000 per core
  - Enterprise Edition — € 7.000 per core
- Oracle lives in a different universe
  - Standard Edition — € 8.050 per processor
  - Enterprise Edition — € 42.500 per processor
- PostgreSQL is free
  - Open source





CHEAP OUT ON FREE EDITION

Free as in beer

- SQL Server
  - Express Edition – 1 CPU + 1 GB RAM + ~~10 GB DB~~ 50 GB in 2025!!
  - Developer Edition – Unlimited for non-production use
  - Free tier on Azure SQL
- Oracle
  - Express Edition – 2 CPU + 2 GB RAM + 20 GB DB
  - Personal Edition – not free but inexpensive for dev use
  - Oracle Cloud Free Tier
- PostgreSQL
  - Everything is for free

THE “OPEN-SOURCE TAX”

- Lack of official support
  - Community-centric support – AKA “RTFM”
- Sparse or outdated documentation
- Dependency hell
  - “Works on my machine” syndrome
- DIY Troubleshooting
  - StackOverflow, Reddit, GitHub issues, Source Code
- Mitigated by the cloud
  - All is taken care by pros





EVERY CLOUD HAS A SILVER LINING

… but can cost you serious money

Platform	Engine	vCPU	RAM	Storage	DB Cost/ month	Storage/ month	License included?	Main notes
AWS RDS for SQL Server Standard	SQL Server	4	16 GB	128 GB	~$839	~$15	Yes	MS support, higher cost
AWS RDS for PostgreSQL	PostgreSQL	4	16 GB	128 GB	~$532	~$23	Yes (open-source)	Much cheaper
Azure SQL Managed Instance (GP)	SQL Server	4	16 GB	128 GB	~$800	~$11	Yes	Enterprise features, MS support
Azure Database for PostgreSQL Flex.	PostgreSQL	4	16 GB	128 GB	~$260	~$14	Yes (open-source)	Replica, HA, Azure integration



SUPPORT

SQL Server

- Official Microsoft Support
  - Several options available
- Cloud Support
- Outstanding documentation
- Partners and consultants
- Legacy product support
  - Mainstream support
  - Extended support
- Community

PostgreSQL

- No official support channel
  - EnterpriseDB (EDB) is the leader
- Cloud Support
- Good documentation
- Partners and consultants
- No legacy support
  - Partners
  - Community
- Community



COMMUNITY

Sharing is caring

SQL Server

- Very Active
- MVPs, bloggers, speakers
- International and regional events
- Forums and support channels
- Sharing culture, friendly attitude

PostgreSQL

- Very Active
- Very technical
- International and regional events
- Forums and mailing lists
- Rigorous but unfriendly attitude – “RTFM” prone answers



DEVELOPMENT TOOLS

SQL Server

- Visual Studio Database Projects
  - Versioning
  - Schema compare
  - Refactoring
  - Automated deployment
  - Unit testing
  - CI/CD integration
  - Top notch IDE experience

PostgreSQL

- Nothing like Visual Studio
  - pgAdmin ERD Tool
  - pgModeler
- Liquibase/Flyway for CI/CD
- Command-line tools

DEVELOPMENT

A true Swiss Army Knife



PostgreSQL has unique development features

- Countless supported **languages** (SQL, pgSQL, python, java, PHP, R, Lua, Rust, Ruby, Perl…)
- Native types, **table types** (with inheritance!)
- **JSON/JSONB** native data type for NoSQL features
- Fulltext search, RegExp, advanced text manipulation
- **Several extensions***
  - TimescaleDB
  - pgVector
  - Apache AGE (graph)
  - pgML
  - postgREST
  - pgTap unit testing
  - pg_graphql
  - omnigres (containers)
  - pg_doom

***open-source tax applies**



MANAGEMENT TOOLS

SQL Server

- Management Studio (SSMS)
- Visual Studio Code Extension
- Sqlcmd
- More “visual”

PostgreSQL

- pgAdmin
- Dbeaver / DataGrip / HeidiSQL
- psql
- More “scripting”

Scripts are good for automation

GUI helps with troubleshooting, especially with inexperienced DBAs or when an outage hits



PERFORMANCE

Indexing

SQL Server

- Columnstore indexes
- Filtered Indexes / Indexed Views
- In-memory OLTP
- Parallelism
- Advanced Partitioning

PostgreSQL

- GIN (General Inverted Indexes)
- GiST (Generalized Search Tree)
- Hash indexes
- Partitioning
- Columnstore and in-mem available with extensions



PERFORMANCE

Performance tuning

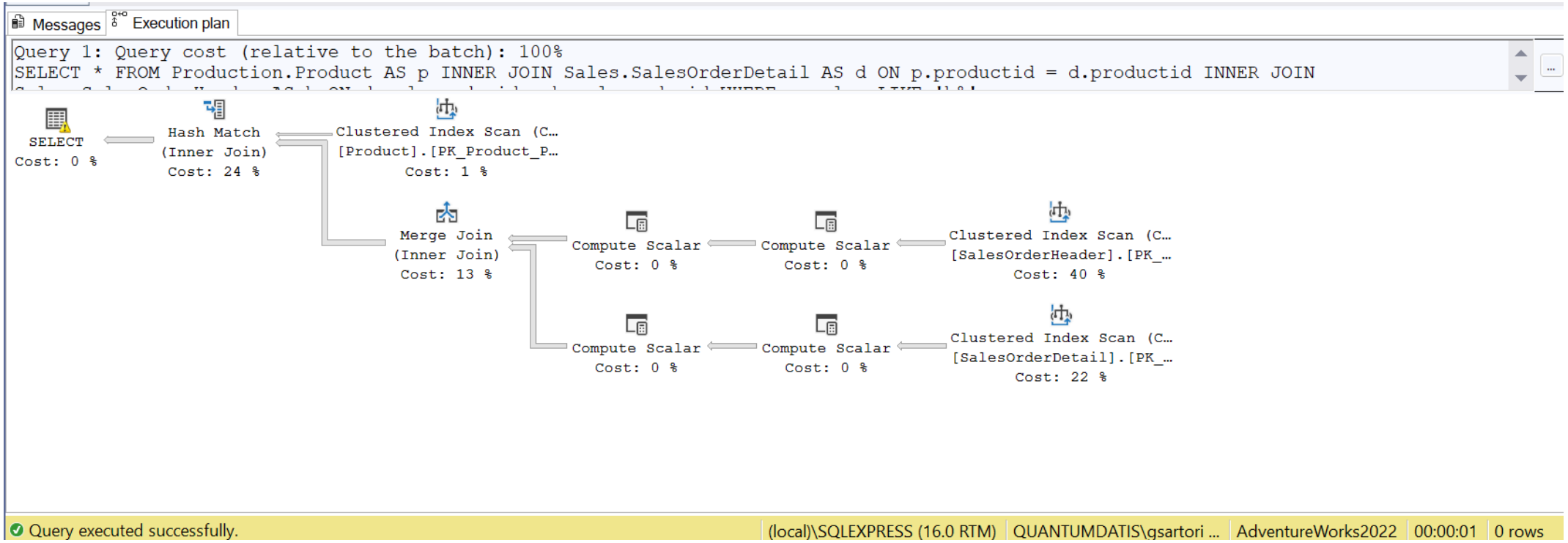
SQL Server superior performance tuning tools

- Rich query execution plans
- Live execution plans
- Query Store
- Extended Events
- Automatic tuning
- Countless DMVs to observe every possible entity



EXECUTION PLANS

Rich, graphical execution plans





EXECUTION PLANS

XML execution plans loaded with information

Execution plan.xml | SQLQuery3.sql...sartori (83))* | SQLQuery2.sql...ks (demo (0))* | SQLQuery1.sql...ks (demo (0))*

```
<?xml version="1.0" encoding="utf-16"?>
<ShowPlanXML xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="
  <BatchSequence>
    <Batch>
      <Statements>
        <StmtSimple StatementCompId="1" StatementEstRows="52540.3" St
          <StatementSetOptions ANSI_NULLS="true" ANSI_PADDING="true"
          <QueryPlan NonParallelPlanReason="NoParallelPlansInDesktopO
            <Warnings>
              <PlanAffectingConvert ConvertIssue="Cardinality Estimat
            </Warnings>
            <MemoryGrantInfo SerialRequiredMemory="1024" SerialDesire
            <OptimizerHardwareDependentProperties EstimatedAvailableM
            <TraceFlags IsCompileTime="true">
              <TraceFlag Value="8017" Scope="Global" />
            </TraceFlags>
            <RelOp AvgRowSize="664" EstimateCPU="0.624811" EstimateIO
              <OutputList>
                <ColumnReference Database="[AdventureWorks2022]" Sche
                <ColumnReference Database="[AdventureWorks2022]" Sche
```



EXECUTION PLANS

Feels like SQL Server 6.5

Grid Text Record

```
QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------------------------------
Nested Loop  (cost=19.46..408.87 rows=246 width=494) (actual time=1.147..20.208 rows=1218 loops=1)
  Output: p.productid, p.name, p.productnumber, p.makeflag, p.finishedgoodsflag, p.color, p.safetystocklevel, p.reorderpoint, p.standardcost, p.l
  Inner Unique: true
  ->  Hash Join  (cost=19.17..157.17 rows=246 width=267) (actual time=1.095..7.284 rows=1218 loops=1)
        Output: p.productid, p.name, p.productnumber, p.makeflag, p.finishedgoodsflag, p.color, p.safetystocklevel, p.reorderpoint, p.standardcos
        Inner Unique: true
        Hash Cond: (d.productid = p.productid)
        ->  Seq Scan on sales.salesorderdetail d  (cost=0.00..126.28 rows=4428 width=122) (actual time=0.045..1.772 rows=5998 loops=1)
              Output: d.salesorderid, d.salesorderdetailid, d.carriertrackingnumber, d.orderqty, d.productid, d.specialofferid, d.unitprice, d.un
        ->  Hash  (cost=18.82..18.82 rows=28 width=145) (actual time=1.030..1.031 rows=119 loops=1)
              Output: p.productid, p.name, p.productnumber, p.makeflag, p.finishedgoodsflag, p.color, p.safetystocklevel, p.reorderpoint, p.stand
              Buckets: 1024  Batches: 1  Memory Usage: 30kB
              ->  Seq Scan on production.product p  (cost=0.00..18.82 rows=28 width=145) (actual time=0.038..0.959 rows=119 loops=1)
                    Output: p.productid, p.name, p.productnumber, p.makeflag, p.finishedgoodsflag, p.color, p.safetystocklevel, p.reorderpoint, p
                    Filter: (((p.color)::text ~~* 'b%'::text COLLATE sys.bbf_unicode_cp1_cs_as) AND (p.color >= 'b'::sys.nvarchar) AND (p.color <
                    Rows Removed by Filter: 385
  ->  Index Scan using pk_salesorderheader_salesorderid on sales.salesorderheader h  (cost=0.29..1.02 rows=1 width=227) (actual time=0.008..0.008
        Output: h.salesorderid, h.revisionnumber, h.orderdate, h.duedate, h.shipdate, h.status, h.onlineorderflag, h.salesordernumber, h.purchase
        Index Cond: (h.salesorderid = d.salesorderid)
Planning Time: 1.102 ms
Execution Time: 20.584 ms
```



ENTEPRISE FEATURES

Security

Feature	SQL Server	PostgreSQL
Encryption at rest	Transparent Data Encryption (TDE)	pgcrypto (extension), or via filesystem/disk
Row-level security	Yes, native	Yes, native (since v9.5+)
Data masking	Yes, native	Plugin/extension (e.g. pg_masking)
Auditing	Yes, very advanced	Basic (log statement), external plugins (pgaudit)
Always Encrypted	Yes, native (client-side encryption)	Needs external tool or custom implementation



ENTERPRISE FEATURES

Compression & Storage

Feature	SQL Server	PostgreSQL
Table/Index compression	Row/Page/Columnstore compression, native	TOAST (automatic), extensions for compression
Columnstore index	Yes, native, very advanced	Extensions (e.g. cstore_fdw), less integrated
Partitioning	Yes, native, advanced	Yes, native (since v10)
FileStream/FileTable	Yes, manage BLOBs/files in DB or filesystem	BLOBs via bytea or Large Object, less flexible



ENTEPRISE FEATURES

High Availability and Disaster Recovery

Feature	SQL Server	PostgreSQL
High Availability	Always On AG, FCI, log shipping, mirroring	Streaming replication, Patroni, repmgr
Advanced backup/restore	Yes, integrated, point-in-time, offloaded backup	Yes, via CLI tools (pg_dump, basebackup)
Geo-replica	Yes, native	Yes, complex configuration
Online restore	Yes, partial (page/partial restore)	More limited, often full restore

- PostgreSQL lacks automatic failover and automated client routing
- PostgreSQL requires external tools and/or extensions:
  - Patroni + etcd/consul/ZooKeeper
  - haproxy
  - repmgr
  - pgBouncer



ENTERPRISE FEATURES

Other features

Feature	SQL Server	PostgreSQL
Change Data Capture (CDC)	Yes, native	Extension (pgoutput, wal2json, Debezium)
Temporal tables	Yes, native	Extension (temporal_tables)
Data masking	Yes, native	Extension
Service Broker (messaging)	Yes, native	External plugins (PGQ, others)
JSON support	Yes	Yes, very advanced
GIS/Spatial	Yes, native (SQL Server Spatial)	Yes, PostGIS (very advanced)

SQL
KONFERENZ
2026

CONCLUSIONS





KEEP THE FAITH

- SQL Server is still the winner for Enterprise Features
- PostgreSQL provides many features as extensions
- The cloud reduces friction and “open-source tax”
- Development and Maintenance are better in SQL Server
- PostgreSQL offers more possibilities, requires more tinkering

Do not attempt a migration!

Tools like Babelfish make it look easy. It’s a trap.

IS SQL SERVER STILL MY FAVORITE CHILD?





FEEDBACK PLEASE!



THANKS!