SQL Server on Linux

Dev Space Conference
14 October 2017

Allison Benneth

Allison@sqltran.org

@SQLTran

www.sqltran.org

DevSpace would like to thank our sponsors

RESULT STACK

novation systems

Progress

CODEROCKET

COMPLETE DEVELOPER PODCAST

RED HAT OPENSHIFT Container Platform

PLURALSIGHT

Dynetics

ROCKET CITY ·DIGITAL·

U.S. Space & Rocket Center

TOYOTA MOTOR MANUFACTURING ALABAMA

Nodevember

PHOCAZ

Agenda

- A little history
- SQL Server on Linux
  - Installation walkthrough
- Limitations
  - Express Edition
  - Restore from Windows to Linux
- New(ish) Features
  - Graph database
  - Temporal tables

Ancient History

- July 2011 - Hal Berenson (former MS) on why SQL won’t be ported to *Nix
  - Analyzed feasibility of port about 2000
  - Reasons why not
    - Core engine would be relatively simple; everything else very hard
    - Too many Windows-specific components (CLR, for example)
    - Concern over undercutting Microsoft’s Windows business
    - Concern over how third-parties would sell it
    - Concern over mixed support scenarios

https://hal2020.com/2011/07/27/porting-microsoft-sql-server-to-linux/

More Recent History

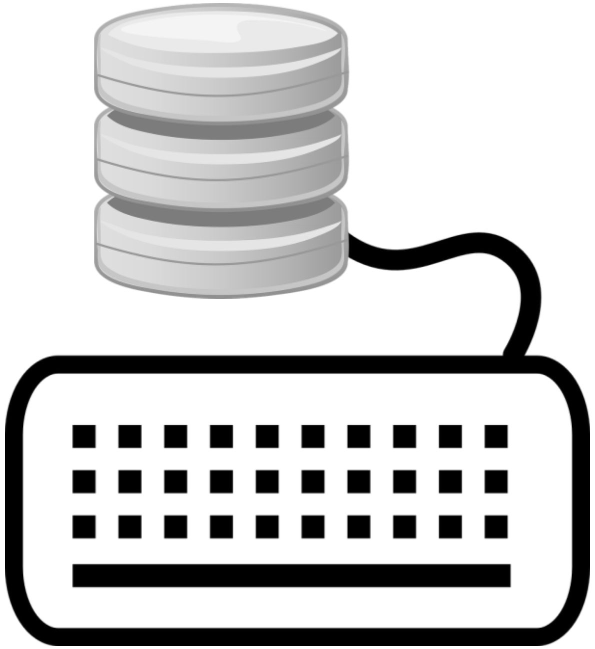
- 7 March 2016 – MS announces SQL Linux
- Secret recipe: SQLPAL
- Slava Oks on overcoming Hal’s limitations
  - https://blogs.msdn.microsoft.com/slavao/2017/09/24/what-made-porting-microsoft-sql-server-to-linux-to-be-a-right-move-after-all/
- General Availability on 2 October 2017

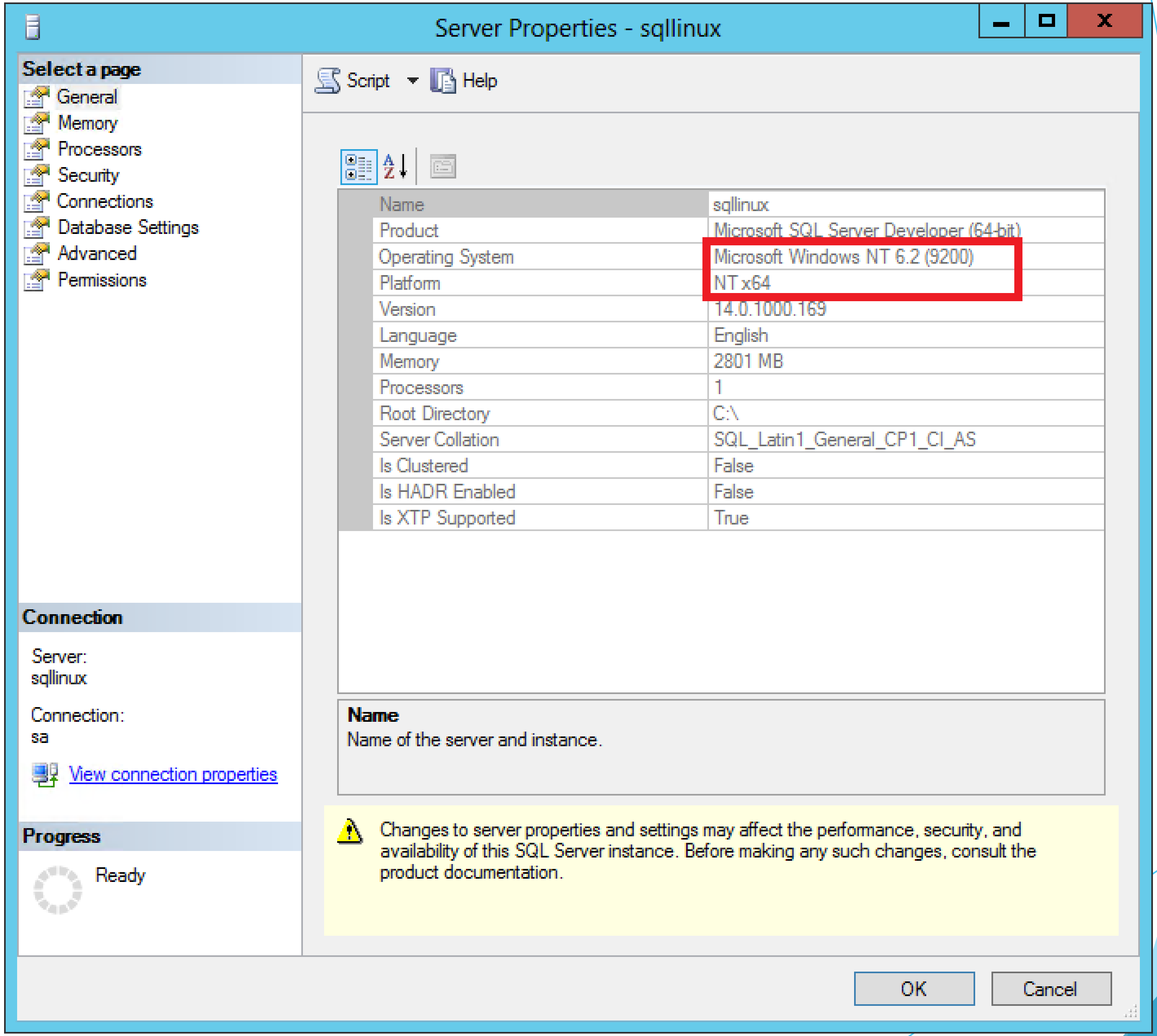
SQL Server 2017 Platforms

- Red Hat Enterprise Linux 7.3 or 7.4
- SUSE Linux Enterprise Server, v12 SP2
- Ubuntu 16.04
- Docker 1.8+
- Windows Server 2012+

Installing SQL Server

DEMO





Name	sqllinux
Product	Microsoft SQL Server Developer (64-bit)
Operating System	Microsoft Windows NT 6.2 (9200)
Platform	NT x64
Version	14.0.1000.169
Language	English
Memory	2801 MB
Processors	1
Root Directory	C:\
Server Collation	SQL_Latin1_General_CP1_CI_AS
Is Clustered	False
Is HADR Enabled	False
Is XTP Supported	True

Name
Name of the server and instance.

Connection

Server:
sqllinux

Connection:
sa

View connection properties

Progress

Ready

⚠ Changes to server properties and settings may affect the performance, security, and availability of this SQL Server instance. Before making any such changes, consult the product documentation.

SQL on Linux Components

- Database engine
- SQL agent
- Configuration script
- bcp
- sqlcmd
- dtexec

Configuration Tool (mssql-conf)

- Collation
- Default DB Mail Profile
- Default data/log/backup/dump/audit directories
- Dump type
- Availability Group
- Memory limit
- Locale
- TCP Port / TLS
- Startup trace flags
- Telemetry

Limitations (SQL on Linux)

- No transactional/merge replication
- No buffer pool extensions
- No StretchDB, Polybase, system extended SPs, Filetable
- Safe CLR assemblies only
- T-SQL only in agent jobs
- No agent alerts, log reader, change data capture, mirroring
- No browser, R services, Analysis services, Reporting services
- Limited SSIS (package execution only, no catalog)
- And much more …

Cumulative Updates

- Bug fixes specific to a SQL Server version and service pack
- New servicing model for SQL 2017
  - New CU issued by Microsoft every month for first year
  - Every three months thereafter
  - No service packs
- Are “cumulative,” so only need the most recent update
- Recent CUs go through more rigorous testing; MS recommends applying them by default

For the First Time Ever …

- Install Linux OS on server
- Install SQL Server, choose Express Edition

- Net result = $0 licensing

Limitations on Express Edition

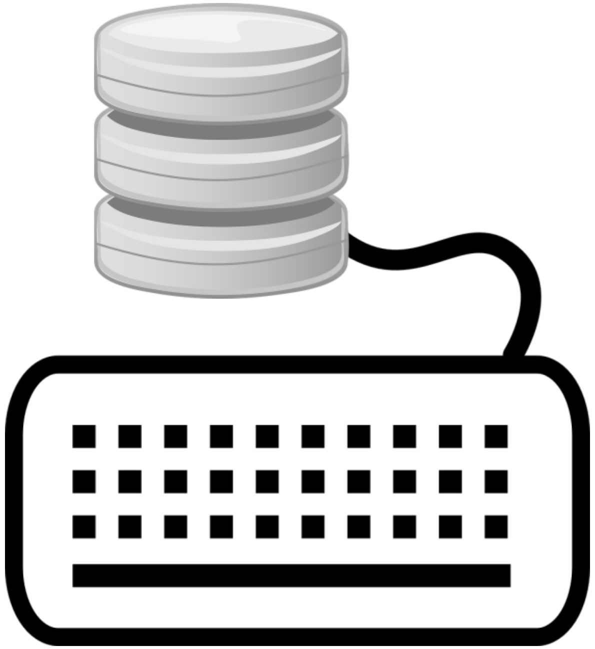
- Performance
  - One CPU / four cores – per instance
  - 1.4 GB RAM (buffer pool) – per instance
  - 350 MB for in-memory tables – per instance, not counted toward buffer pool limit – single-threaded only
  - 350 MB for columnstore data – per database, not counted toward buffer pool limit – single-threaded only

Limitations on Express Edition

- Functionality
  - 10 GB per database
  - No SQL Agent (service installed, but cannot be started)
    - Schedule backups and other jobs via another SQL Agent or OS scheduler (sqlcmd or PowerShell)
- Feature Limitations
  - Availability Groups
  - No SSIS, SSAS, R Server, etc.
- Beware! Mandatory telemetry

Restore Database to SQL Linux

DEMO



Key New SQL 2017 Features

- Adaptive query processing
- Resumable online index rebuilds
- Availability Groups on unmanaged clusters
- String functions: CONCAT_WS, TRANSLATE, TRIM, STRING_AGG
- Graph databases

TRIM

- Mostly equivalent to LTRIM(RTRIM(…))
- Can specify chars to be trimmed

```
select trim('*#!' from '#A proper trim
function!***');
```

TRANSLATE

- Multi-character version of REPLACE
- Syntactic sugar for REPLACE(REPLACE(REPLACE(…)))

```
select translate('string to be replaced',
'abc', 'xyz');
```

CONCAT_WS (Concatenate with separator)

- CONCAT_WS(separator, value1, value2, value3)

```
select concat_ws(',', 'field1', 'field2',
'field3');
```

STRING_AGG

- Aggregate function to concatenate strings

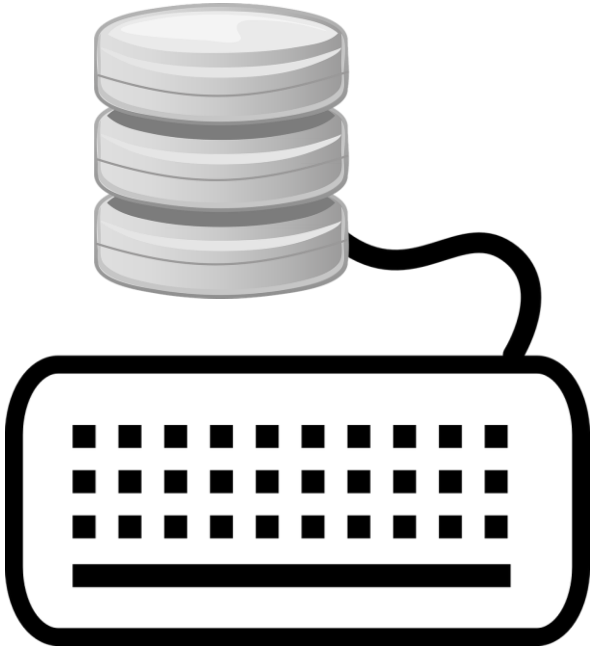
```
select c.City, c.State,
string_agg(cast(c.FirstName as varchar(max)), ',
')
within group (order by c.FirstName) as
CustomerList
from CorpDB.dbo.Customer c
where c.State = 'AL'
group by c.City, c.State
order by c.City, c.State;
```

Graph Databases

- Model many-to-many relationships
- Useful for hierarchies, social media systems, parts explosions, etc.
- Graph = nodes + edges
  - Node = object / entity
  - Edge = relationship
- Directed (one-way) graphs only at this time
- Match operator

Graph Databases

DEMO

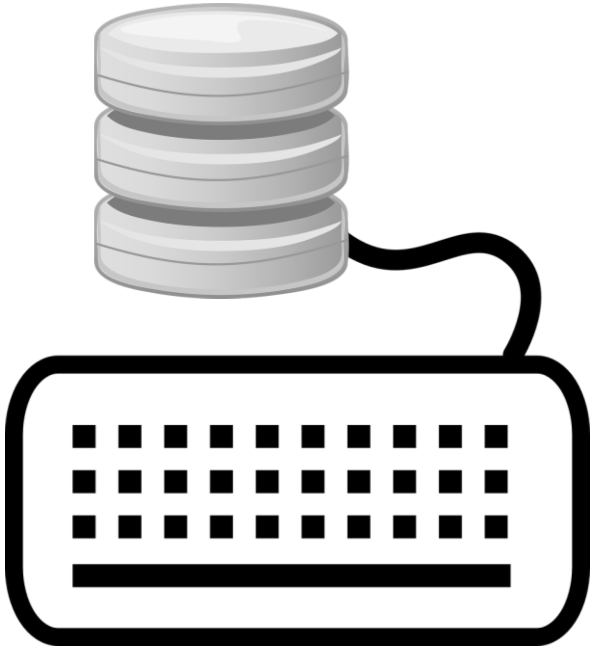


Temporal Tables (2016 Feature)

- Most applications / databases inherently contain a temporal element
- If temporal components are tracked, traditionally done with triggers or change detection
- Temporal tables handle tracking automatically
- Allows greatly simplified point-in-time querying
- Requires additional columns on source table and requires history table
- Schema changes in source table are reflected in the history table

Temporal Tables

DEMO



Temporal Tables

Temporal querying:	FROM TableName FOR SYSTEM_TIME _____
Point in time	AS OF '2017-02-06 11:30:00'
Full history	ALL
Between (‘start’ < EndTime AND ‘end’ >= StartTime)	BETWEEN '2017-01-11 18:55:04' AND '2017-05-06 11:30:00'
From (‘start’ < EndTime AND ‘end’ > StartTime)	FROM '2017-01-11 18:55:04' TO '2017-05-06 11:30:00'
Contained in (‘start’ >= EndTime AND ‘end’ <= StartTime)	CONTAINED IN ('2017-01-11 18:55:04', '2017-05-06 11:30:00')

Temporal Tables

- Performance
  - Insert operations – no difference than non-temporal tables
  - Update operations – overhead due to writes to both source and history tables
  - Read operations – Default clustered index on history table usually not helpful – consider changing it

Temporal Tables

- Beware of v1 (and v2!) limitations!
  - Dropping a column in the source table will drop the column in the history table – all history is lost!
  - Cannot add a non-nullable column to the source table
  - Pruning history is an offline operation
- SQL 2017 Enhancements
  - Can define retention policy

```
CREATE TABLE(…
WITH(… HISTORY_RETENTION_PERIOD = 6 MONTHS));
```

Resources

- SQL Server 2017 release notes for Linux (with links to installation instructions) docs.microsoft.com/en-us/sql/linux/sql-server-linux-release-notes
- Companion blog page to this session www.sqltran.org/sql-linux

Allison Benneth

Allison@sqltran.org

@SQLTran

www.sqltran.org