

Archiving via DB conversion



















Spectral Core




































Damir Bulic • dbulic@spectralcore.com • @BulicDamir

Databases supported

Over 40 formats

- File-based
- Relational
- Cloud
- Data lakes

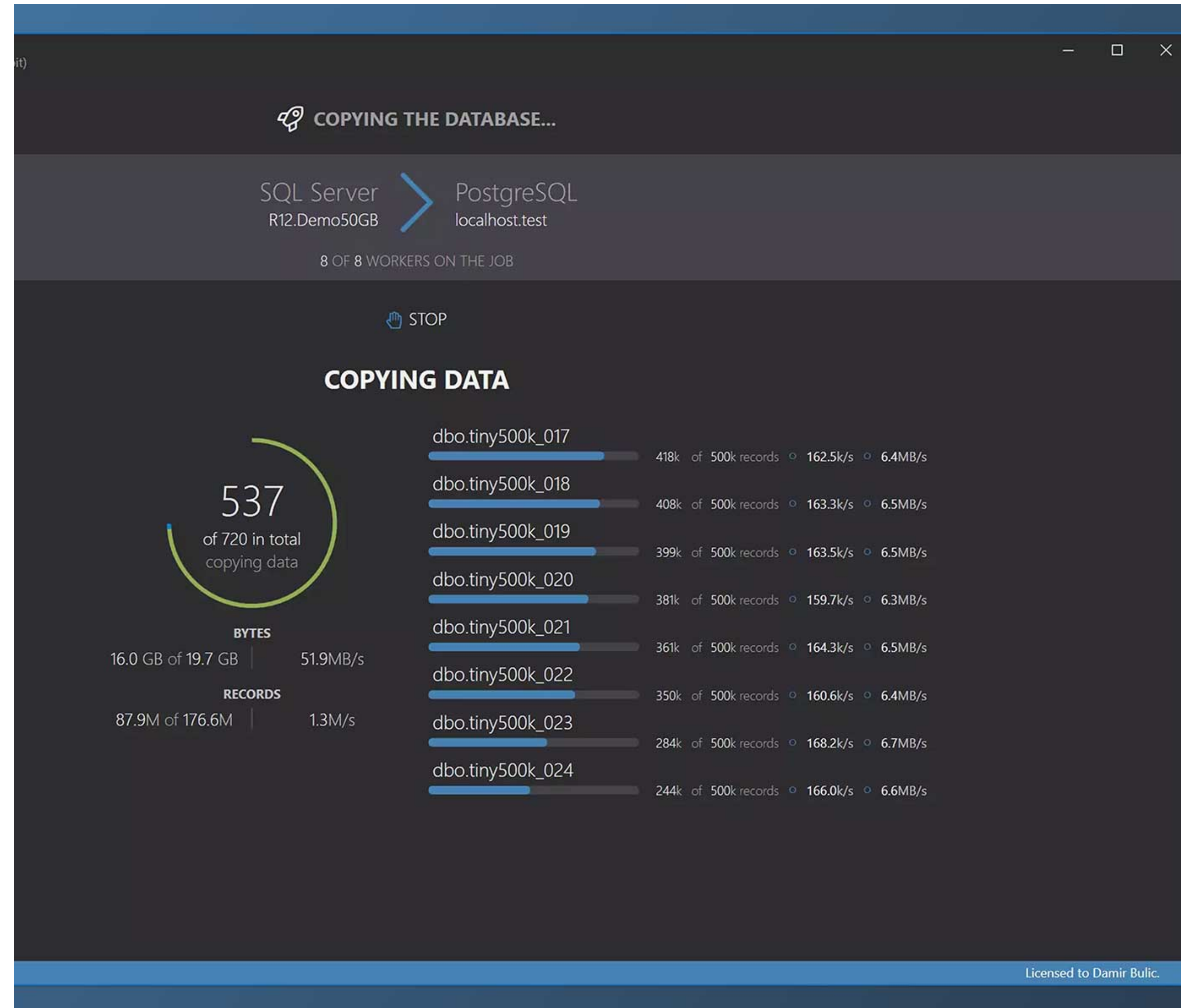
 ANALYTICS PLATFORM SYSTEM	 AZURE SQL DATABASE	 AZURE SYNAPSE ANALYTICS	 CSV
 IBM DB2 LUW	 IBM DB2 ISERIES	 IBM DB2 Z/OS	 INGRES
 MARIADB	 MYSQL	 ODBC	 ORACLE
 PARQUET	 POSTGRESQL	 SQL SERVER	 SYBASE IQ
 TERADATA	 VERTICA		

 ACCESS	 AMAZON RDS	 CSV	 IBM DB2 LUW	 IBM DB2 Z/OS	 IBM DB2 AS/400	 DBASE	 EXCEL
 FIREBIRD	 GREENPLUM	 INGRES	 INTERBASE	 LOCALDB	 MARIADB	 MYSQL	 NEXUSDB
 ODBC	 ORACLE	 PARADOX	 PERSVASIVE	 POSTGRESQL	 OPENEDGE	 SIARD	 SQL AZURE
 SQL SERVER	 SQL CE	 SQLBASE	 SQLITE	 SYBASE ASE	 SYBASE ADVANTAGE	 SQL ANYWHERE	 TERADATA
 VERTICA	 VISTADB	 VISUAL FOXPRO					

Full Convert

spectralcore.com/fullconvert

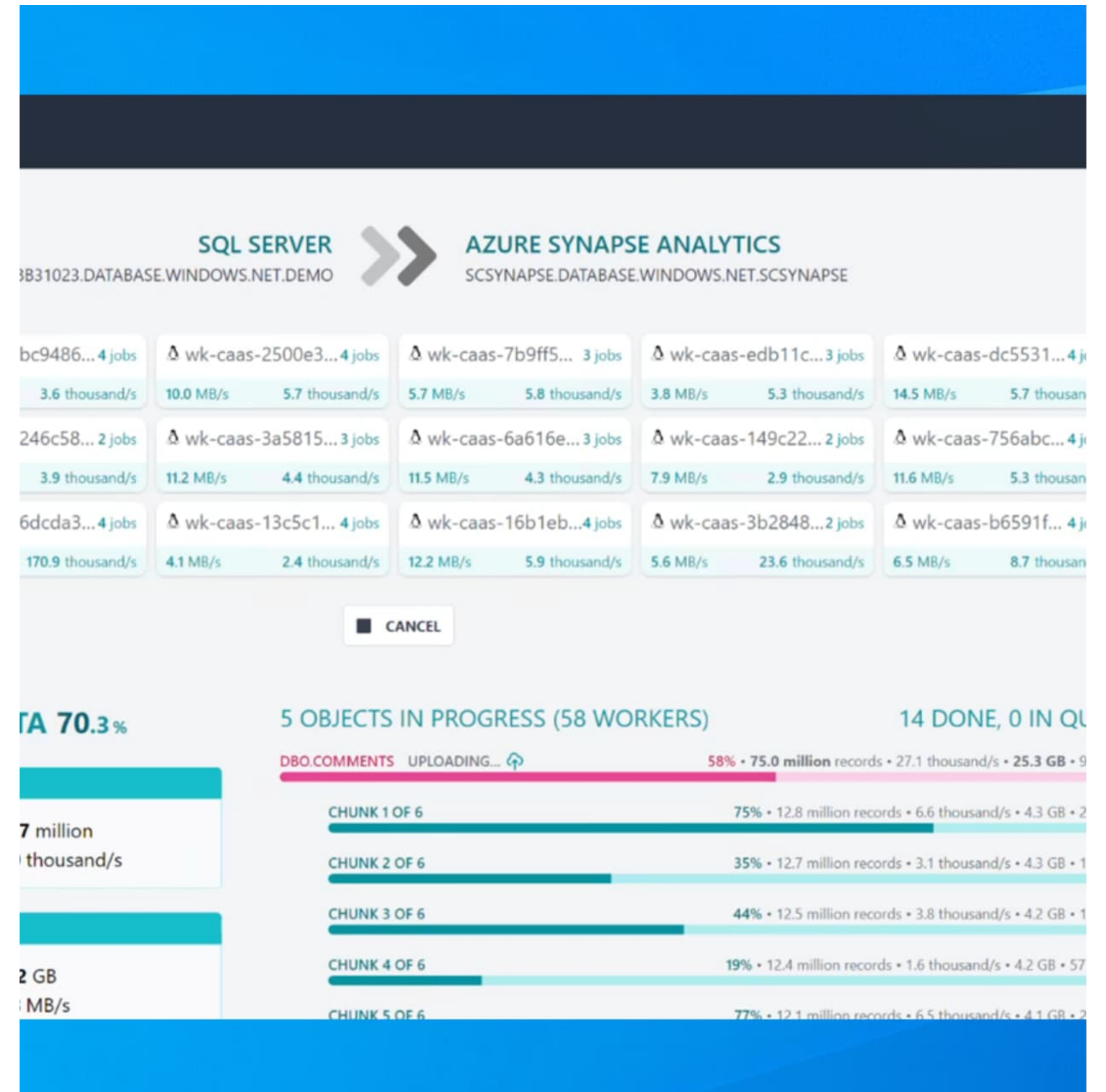
- Single-click migration
- Drivers built-in, no external dependencies
- Scheduler built-in for recurring migrations
- Console included for easy scripting
- Customers in more than 90 countries
- Many success stories
- **Full SIARD support!**



Omni Loader

Distributed migration cluster

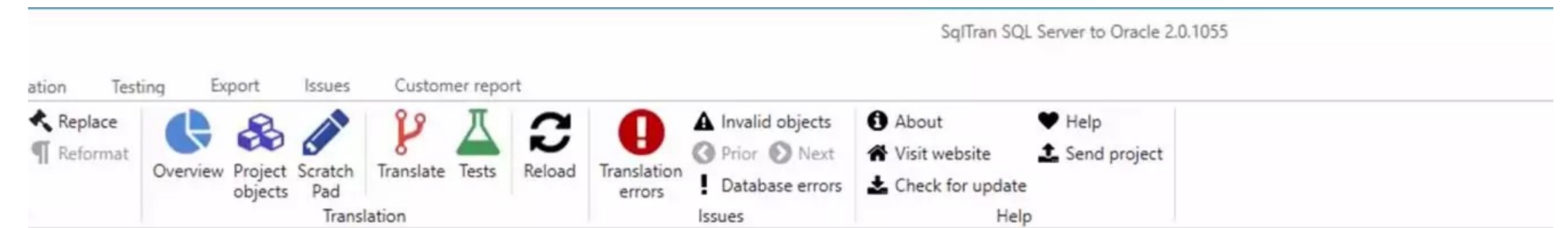
- Ideal solution for migrating databases on premises to the cloud
- Handles hundreds of terabytes easily
- Same ease of use as Full Convert, for the basic use-case
- Works on Windows and Linux
- Fetches data from several source databases via hundreds of connections from several agents, compresses and encrypts in-flights, ingests to the target
- Repartitions data on the fly



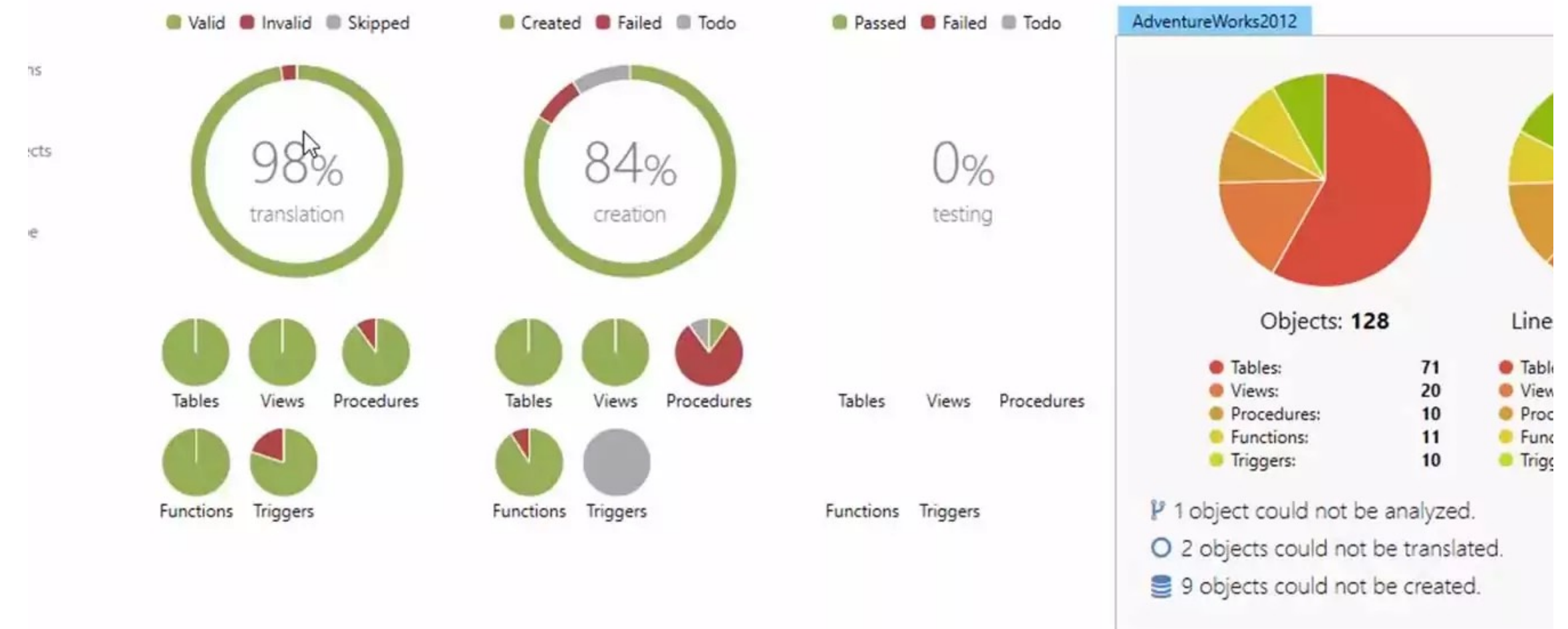
SQL Tran

DDL translation of very complex schema

- Custom parser generator engine
- 150x faster translation than closest competitor
- Goal of >98% fully automated translation
- Manual translation available for code that needs complete rewrite
- Static analysis used to generate tests
- Uses OmniLoader engine for data migration



Adventure Works



Documenter

Database schema documentation

- Reads full database metadata (same engine as SQL Tran)
- Builds a HTML website with the navigable schema
- **Discontinued!** To be included into upcoming database manager.

HumanResources.Employee x +

File | C:/Users/Damir/Documents/Spectral%20Core/Documenter/Database%20documentation/R12.AdventureWorks2017/_Table_HumanResource...

R12.AdventureWorks2017

Assemblies | Data Types | Roles | Schemas | Tables | Views | Procedures | Functions | Triggers

R12.AdventureWorks2017 Database

HumanResources.**Employee** Table

Information

Name	Employee
Schema	HumanResources
Row Count	0
Data Size	
Index Size	
Reserved Size	
Unused Size	
Created	27-Oct-17 14:33:01
Modified	27-Oct-17 14:33:14

Columns

PK	Key	Identity	Name	Data Type	Allow Nulls	Collation	References	Default	Computed	Compute Expression
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	BusinessEntityID	int	<input type="checkbox"/>				<input type="checkbox"/>	
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NationalIDNumber	nvarchar(15)	<input type="checkbox"/>	SQL_Latin1_General_CP1_CI_AS			<input type="checkbox"/>	
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	LoginID	nvarchar(256)	<input type="checkbox"/>	SQL_Latin1_General_CP1_CI_AS			<input type="checkbox"/>	
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	OrganizationNode	hierarchyid	<input checked="" type="checkbox"/>				<input type="checkbox"/>	
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	OrganizationLevel	smallint	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	([OrganizationNode].[GetLevel]())
	<input type="checkbox"/>	<input type="checkbox"/>	JobTitle	nvarchar(50)	<input type="checkbox"/>	SQL_Latin1_General_CP1_CI_AS			<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	BirthDate	date	<input type="checkbox"/>				<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	MaritalStatus	nchar(1)	<input type="checkbox"/>	SQL_Latin1_General_CP1_CI_AS			<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	Gender	nchar(1)	<input type="checkbox"/>	SQL_Latin1_General_CP1_CI_AS			<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	HireDate	date	<input type="checkbox"/>				<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	SalariedFlag	bit	<input type="checkbox"/>			1	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	VacationHours	smallint	<input type="checkbox"/>			0	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	SickLeaveHours	smallint	<input type="checkbox"/>			0	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	CurrentFlag	bit	<input type="checkbox"/>			1	<input type="checkbox"/>	
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	rowguid	uniqueidentifier	<input type="checkbox"/>			newid()	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	ModifiedDate	datetime	<input type="checkbox"/>			getdate()	<input type="checkbox"/>	

Indexes

Archival needs today

We got you covered

- Full Convert supports SIARD for reading and writing
- Wide range of databases can be easily converted to SIARD with minimal work
- SIARD can be extracted to wide range of databases with minimal work
- Full customization available (renaming, mapping, BLOB placement)
- High performance reader and writer (limited by inherent limitations of SIARD format)

SIARD challenges

It's XML. Zipped XML.

- We can handle up to 16 connections to speed things up, but...
- ZIP writing can't be parallelized. I have a 12-core CPU. We generate as much as possible using all cores. In the end, just one core is used to compress data.
- XML is verbose and clunky
- XML is not well suited to handle a lot of data. We can spill BLOB data outside, but it's a clunky solution.
- Reasonably, SIARD is useful only for small datasets
- SIARD already can't handle much of what we see in real world today. Datasets are growing at extreme pace and I expect SIARD to be less applicable in the future.
- That's why SIARD is not supported in Omni Loader, our most powerful database migration software.

Looking into the future

Horizontal scaling

- Vertical scaling is dead. Moore's law is dead.
- Vertical scaling of a single server today is in reality a horizontal scaling - more cores, not faster cores.
- Looking at cloud, it is all about horizontal, on-demand scaling. This trend will continue.
- Workflows are moving away from personal computers. Developers are often remoting from laptops to remote development environments with less limitations (network, CPU).
- Data is streamed from many sources (IoT, heterogeneous databases) into specialized storage (data lakes, OLTP, OLAP, NoSQL)
- While we can expect relational engines to stay with us for a very long time, we can also expect new engines to appear. We have graph data, hierarchical data, custom data types, CLR types, spatial data etc.

Future proofing archives

A new archival data format

- As described in an email I sent
- In short:
 - Separate structure for data and schema
 - Compressed chunks of columnar data (we use that in Omni Loader already, similar to Parquet)
 - Extensible data types
 - Schema can be written in a single SQLite file (even though JSON would be good enough)
 - Data should be written in one file, or many files on a local network, or anywhere in the world. Instead of referencing data separately for each value, we should be referencing a chunk of data.
 - Data chunk can contain from a few records all the way to billions of records.

Discussion time.