diffst - A tool for Data Integration Testing

Francois Marais

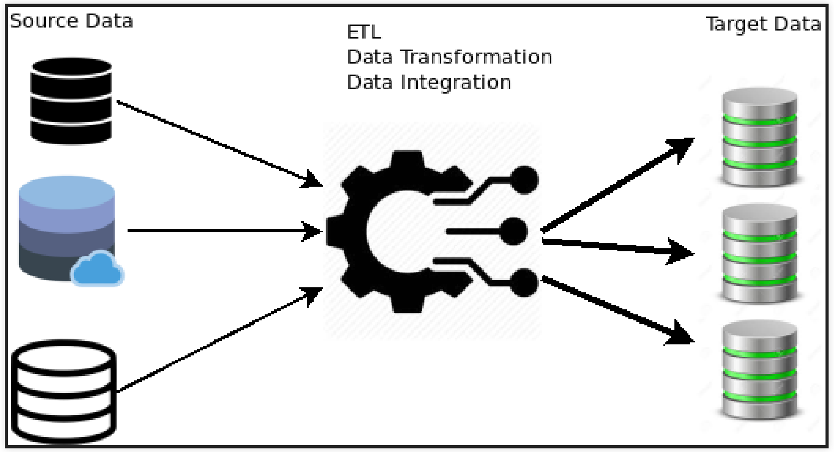
francois@busii.com

Outline

- Data Integration Applications
- Functional Testing Aproaches
- Solution provided by Market Leader
- Demo
- diffst - Features and limitations
- Questions and Discussion

Data Integration/Migration Applications

- Once-off projects
  - Data Migration - hardware, software, database upgrade/relocation
  - Application Migration (CRM/ERP)
  - Take-on of new data sets (eg corporate mergers/acquisitions)
- Systems and Applications
  - Data Warehousing
  - Business Intelligence, Big Data,...
  - Customer Loyalty Programme Engines



Data Integration Applications - Typical Risks

- Data not extracted from Source as expected
  - Wrong time period
  - Not all expected data extracted
  - Extracted at wrong time
- Expected business rules not applied
  - Fields truncated
  - Data Types misread
- Target data Incomplete/Duplicated/Inconsistent..
- Wrong grain

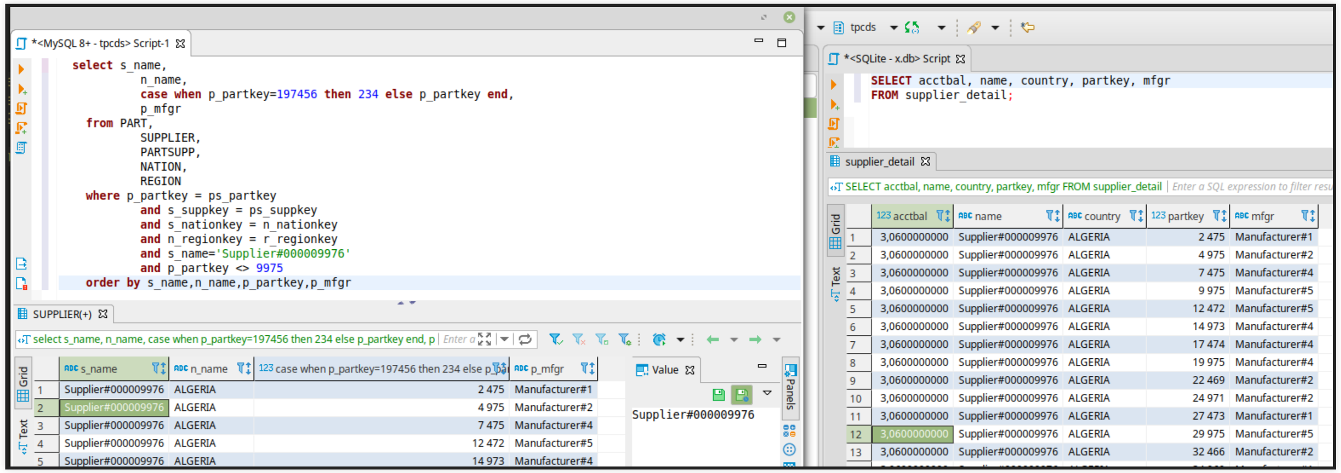
Coverage Challenges

- Large actual data volumes and
- Business logic complexity - per column
- Implies many actual data points available for testing
- ... and they need to be tested - more important than synthetic test data

Common functional testing checks

- SELECT data from Source
- Apply business rules
- SELECT expected data from Target
- Compare and evaluate
- ⇒ Automation opportunity

Sampling: Comparing individual data items



*<MySQL 8+ - tpcds> Script-1

```
select s_name,
          n_name,
          case when p_partkey=197456 then 234 else p_partkey end,
          p_mfgr
  from PART,
          SUPPLIER,
          PARTSUPP,
          NATION,
          REGION
  where p_partkey = ps_partkey
          and s_suppkey = ps_suppkey
          and s_nationkey = n_nationkey
          and n_regionkey = r_regionkey
          and s_name='Supplier#000009976'
          and p_partkey <> 9975
  order by s_name,n_name,p_partkey,p_mfgr
```

SUPPLIER(+)

select s_name, n_name, case when p_partkey=197456 then 234 else p_partkey end, p

	s_name	n_name	case when p_partkey=197456 then 234 else p_pa	p_mfgr
1	Supplier#000009976	ALGERIA	2 475	Manufacturer#1
2	Supplier#000009976	ALGERIA	4 975	Manufacturer#2
3	Supplier#000009976	ALGERIA	7 475	Manufacturer#4
4	Supplier#000009976	ALGERIA	12 472	Manufacturer#5
5	Supplier#000009976	ALGERIA	14 973	Manufacturer#4

Value

```
Supplier#000009976
```

*<SQLite - x.db> Script

```
SELECT acctbal, name, country, partkey, mfgr
FROM supplier_detail;
```

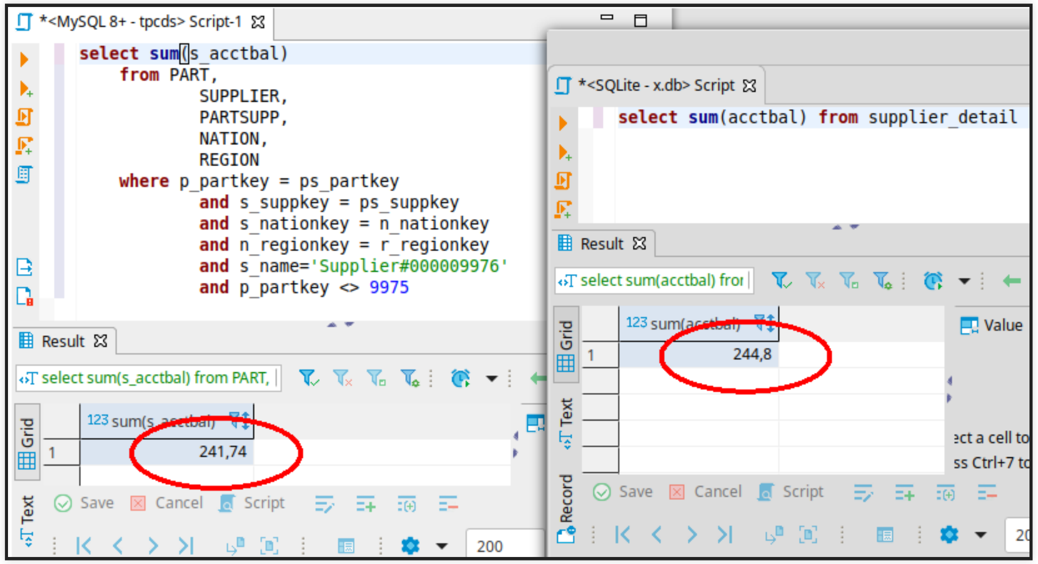
supplier_detail

SELECT acctbal, name, country, partkey, mfgr FROM supplier_detail

	acctbal	name	country	partkey	mfgr
1	3,0600000000	Supplier#000009976	ALGERIA	2 475	Manufacturer#1
2	3,0600000000	Supplier#000009976	ALGERIA	4 975	Manufacturer#2
3	3,0600000000	Supplier#000009976	ALGERIA	7 475	Manufacturer#4
4	3,0600000000	Supplier#000009976	ALGERIA	9 975	Manufacturer#5
5	3,0600000000	Supplier#000009976	ALGERIA	12 472	Manufacturer#5
6	3,0600000000	Supplier#000009976	ALGERIA	14 973	Manufacturer#4
7	3,0600000000	Supplier#000009976	ALGERIA	17 474	Manufacturer#4
8	3,0600000000	Supplier#000009976	ALGERIA	19 975	Manufacturer#4
9	3,0600000000	Supplier#000009976	ALGERIA	22 469	Manufacturer#2
10	3,0600000000	Supplier#000009976	ALGERIA	24 971	Manufacturer#2
11	3,0600000000	Supplier#000009976	ALGERIA	27 473	Manufacturer#1
12	3,0600000000	Supplier#000009976	ALGERIA	29 975	Manufacturer#5
13	3,0600000000	Supplier#000009976	ALGERIA	32 466	Manufacturer#2

- Useful for testing specific data items - Exploratory value
- But not effective in testing large data volumes
- How to build into regression, test suites?

Aggregated SQL



```sql
select sum(s_acctbal)
    from PART,
            SUPPLIER,
            PARTSUPP,
            NATION,
            REGION
    where p_partkey = ps_partkey
            and s_suppkey = ps_suppkey
            and s_nationkey = n_nationkey
            and n_regionkey = r_regionkey
            and s_name='Supplier#000009976'
            and p_partkey <> 9975
```

	sum(s_acctbal)
1	241,74

```sql
select sum(acctbal) from supplier_detail
```

	sum(acctbal)
1	244,8

- Can only test few columns at a time
- Slow to investigate source-target discrepancies

MINUS Queries

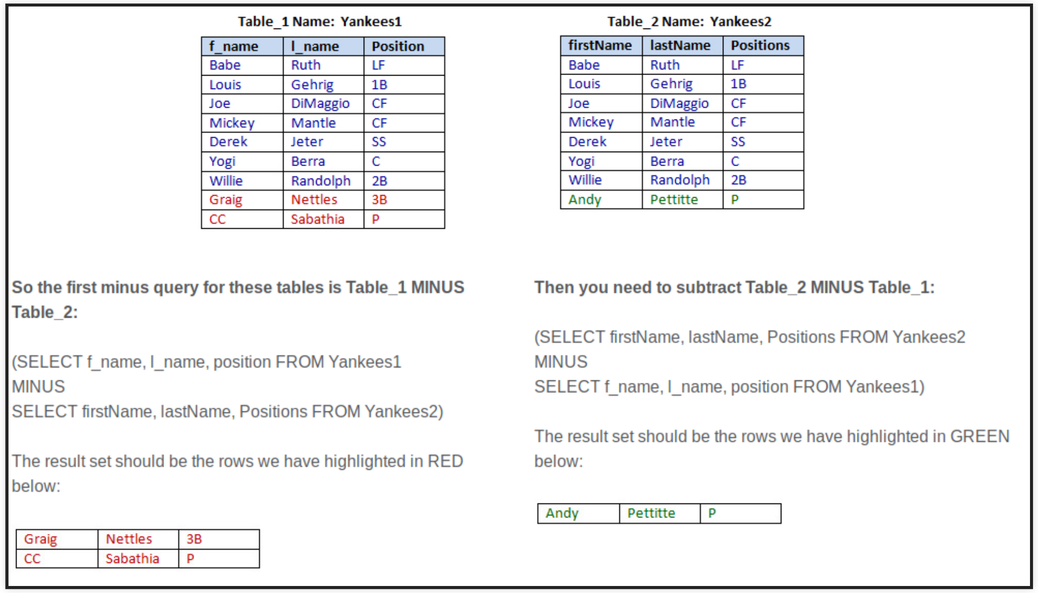


Table_1 Name: Yankees1

f_name	l_name	Position
Babe	Ruth	LF
Louis	Gehrig	1B
Joe	DiMaggio	CF
Mickey	Mantle	CF
Derek	Jeter	SS
Yogi	Berra	C
Willie	Randolph	2B
Graig	Nettles	3B
CC	Sabathia	P

Table_2 Name: Yankees2

firstName	lastName	Positions
Babe	Ruth	LF
Louis	Gehrig	1B
Joe	DiMaggio	CF
Mickey	Mantle	CF
Derek	Jeter	SS
Yogi	Berra	C
Willie	Randolph	2B
Andy	Pettitte	P

So the first minus query for these tables is Table_1 MINUS Table_2:

```
(SELECT f_name, l_name, position FROM Yankees1
MINUS
SELECT firstName, lastName, Positions FROM Yankees2)
```

The result set should be the rows we have highlighted in RED below:

Graig	Nettles	3B
CC	Sabathia	P

Then you need to subtract Table_2 MINUS Table_1:

```
(SELECT firstName, lastName, Positions FROM Yankees2
MINUS
SELECT f_name, l_name, position FROM Yankees1)
```

The result set should be the rows we have highlighted in GREEN below:

Andy	Pettitte	P

- Slow - have to run a query on each side separately - then compare

MINUS Queries

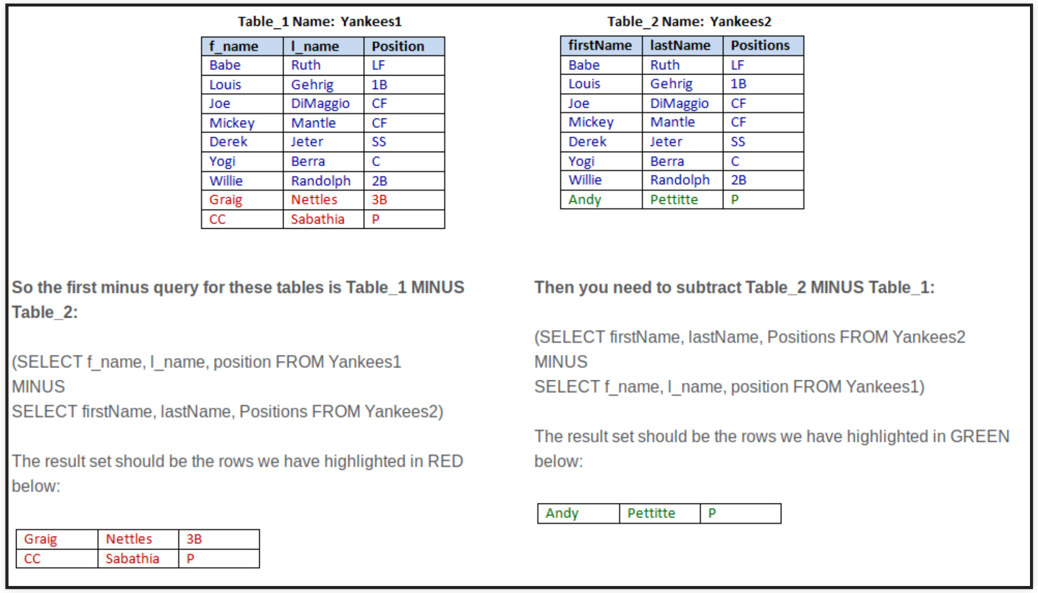


Table_1 Name: Yankees1

f_name	l_name	Position
Babe	Ruth	LF
Louis	Gehrig	1B
Joe	DiMaggio	CF
Mickey	Mantle	CF
Derek	Jeter	SS
Yogi	Berra	C
Willie	Randolph	2B
Graig	Nettles	3B
CC	Sabathia	P

Table_2 Name: Yankees2

firstName	lastName	Positions
Babe	Ruth	LF
Louis	Gehrig	1B
Joe	DiMaggio	CF
Mickey	Mantle	CF
Derek	Jeter	SS
Yogi	Berra	C
Willie	Randolph	2B
Andy	Pettitte	P

So the first minus query for these tables is Table_1 MINUS Table_2:

```
(SELECT f_name, l_name, position FROM Yankees1
MINUS
SELECT firstName, lastName, Positions FROM Yankees2)
```

The result set should be the rows we have highlighted in RED below:

Graig	Nettles	3B
CC	Sabathia	P

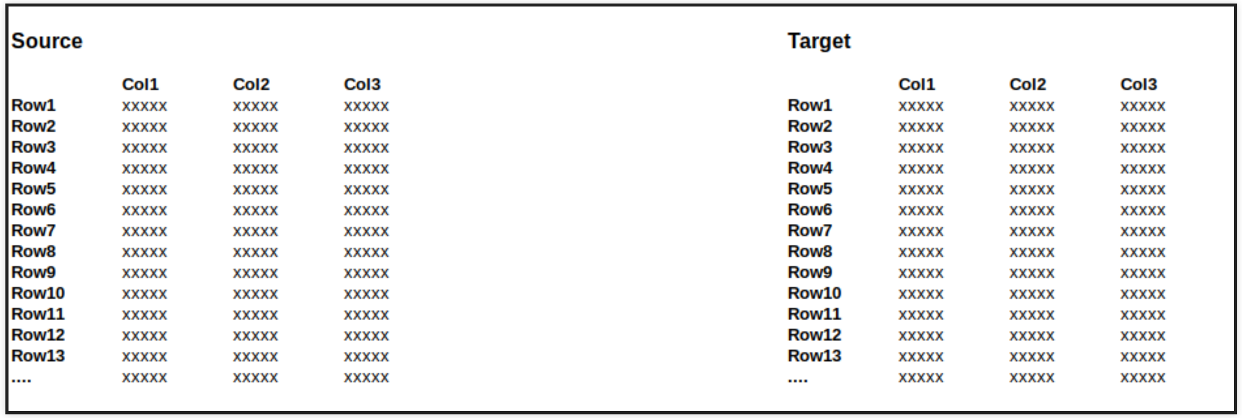
Then you need to subtract Table_2 MINUS Table_1:

```
(SELECT firstName, lastName, Positions FROM Yankees2
MINUS
SELECT f_name, l_name, position FROM Yankees1)
```

The result set should be the rows we have highlighted in GREEN below:

Andy	Pettitte	P

- Source and Target must be on same database
- Slow - have to run a query on each side separately - then compare



Source

	Col1	Col2	Col3
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx

Target

	Col1	Col2	Col3
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx

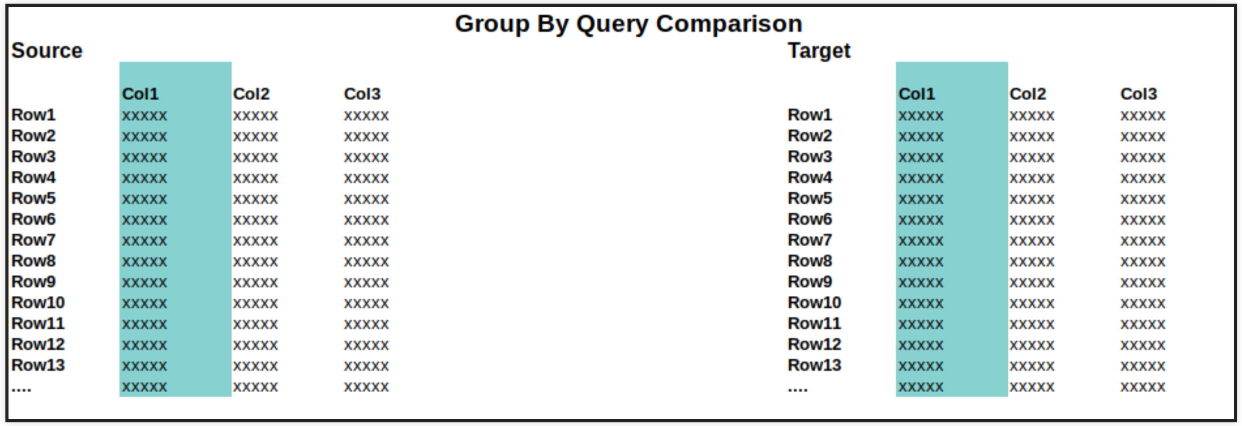
Manual Detailed Comparison

Source

	Col1	Col2	Col3
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx

Target

	Col1	Col2	Col3
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx



Group By Query Comparison

Source

	Col1	**Col2**	**Col3**
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx

Target

	Col1	**Col2**	**Col3**
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx

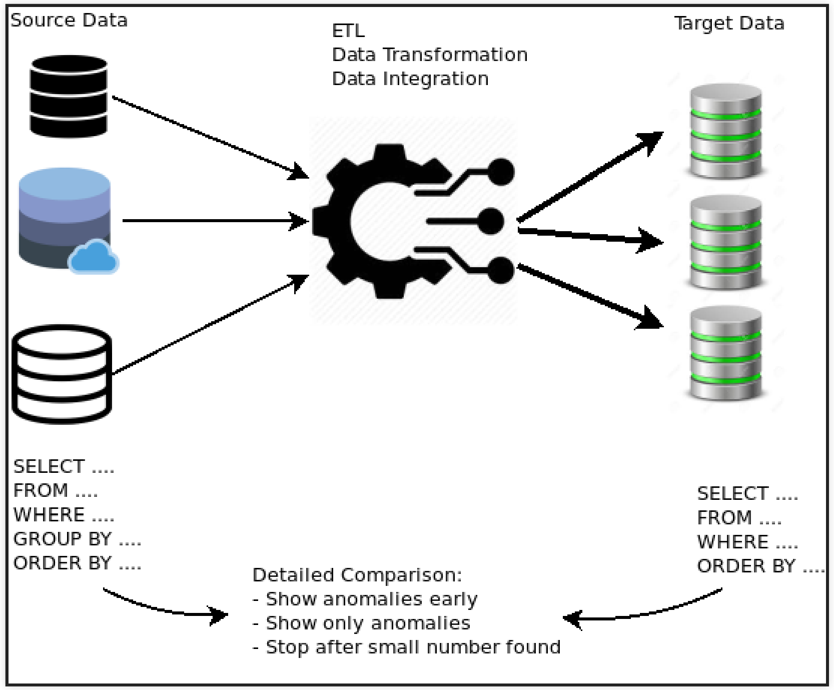
Test Tool – Detailed Comparison of entire data sets

Source

	Col1	Col2	Col3
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx

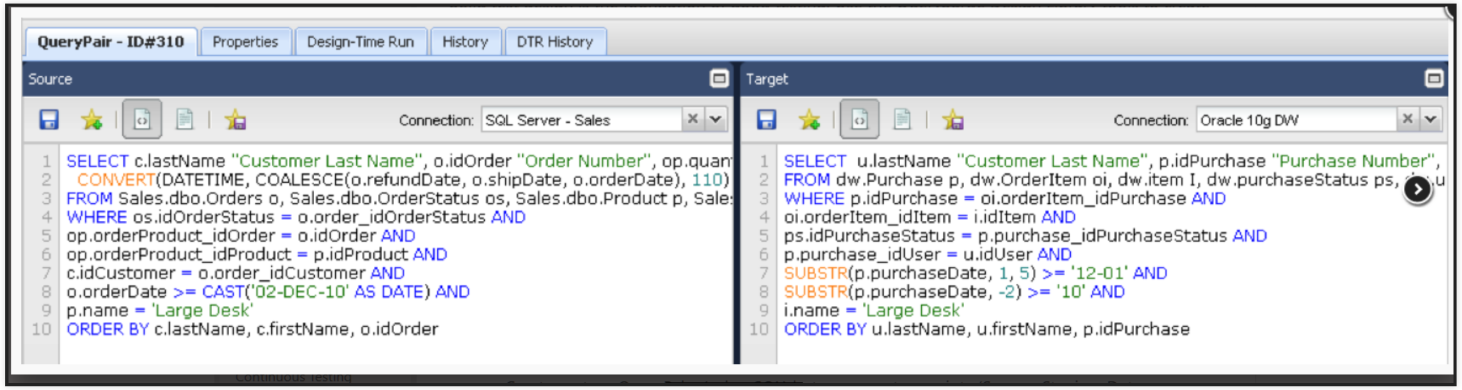
Target

	Col1	Col2	Col3
Row1	xxxxx	xxxxx	xxxxx
Row2	xxxxx	xxxxx	xxxxx
Row3	xxxxx	xxxxx	xxxxx
Row4	xxxxx	xxxxx	xxxxx
Row5	xxxxx	xxxxx	xxxxx
Row6	xxxxx	xxxxx	xxxxx
Row7	xxxxx	xxxxx	xxxxx
Row8	xxxxx	xxxxx	xxxxx
Row9	xxxxx	xxxxx	xxxxx
Row10	xxxxx	xxxxx	xxxxx
Row11	xxxxx	xxxxx	xxxxx
Row12	xxxxx	xxxxx	xxxxx
Row13	xxxxx	xxxxx	xxxxx
....	xxxxx	xxxxx	xxxxx



Requirement for Test Tool

The Market Leader: QuerySurge



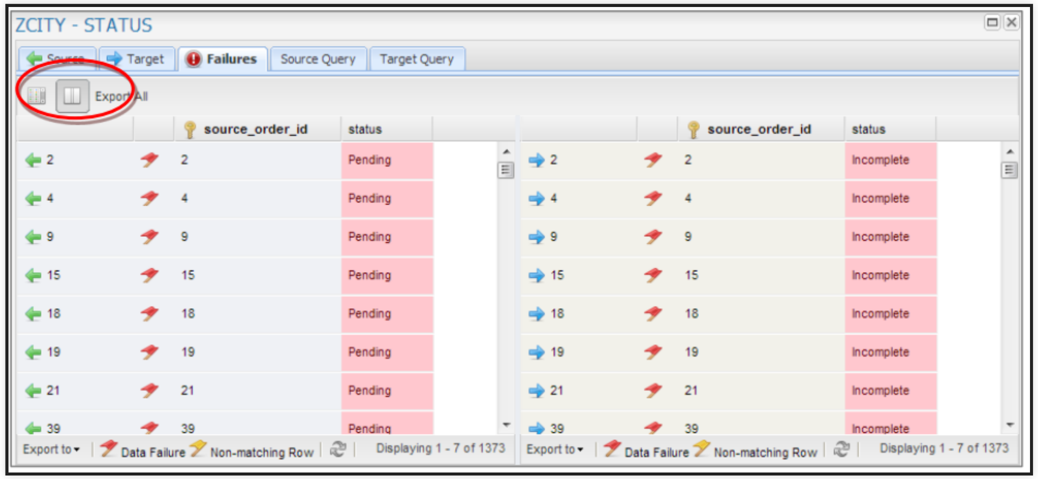
Connection: SQL Server - Sales

```
SELECT c.lastName "Customer Last Name", o.idOrder "Order Number", op.quan
  CONVERT(DATETIME, COALESCE(o.refundDate, o.shipDate, o.orderDate), 110)
FROM Sales.dbo.Orders o, Sales.dbo.OrderStatus os, Sales.dbo.Product p, Sale
WHERE os.idOrderStatus = o.order_idOrderStatus AND
op.orderProduct_idOrder = o.idOrder AND
op.orderProduct_idProduct = p.idProduct AND
c.idCustomer = o.order_idCustomer AND
o.orderDate >= CAST('02-DEC-10' AS DATE) AND
p.name = 'Large Desk'
ORDER BY c.lastName, c.firstName, o.idOrder
```

Connection: Oracle 10g DW

```
SELECT  u.lastName "Customer Last Name", p.idPurchase "Purchase Number",
FROM dw.Purchase p, dw.OrderItem oi, dw.item I, dw.purchaseStatus ps,
WHERE p.idPurchase = oi.orderItem_idPurchase AND
oi.orderItem_idItem = i.idItem AND
ps.idPurchaseStatus = p.purchase_idPurchaseStatus AND
p.purchase_idUser = u.idUser AND
SUBSTR(p.purchaseDate, 1, 5) >= '12-01' AND
SUBSTR(p.purchaseDate, -2) >= '10' AND
i.name = 'Large Desk'
ORDER BY u.lastName, u.firstName, p.idPurchase
```

QuerySurge - Results



ZCITY - STATUS

Source | Target | Failures | Source Query | Target Query

Export All

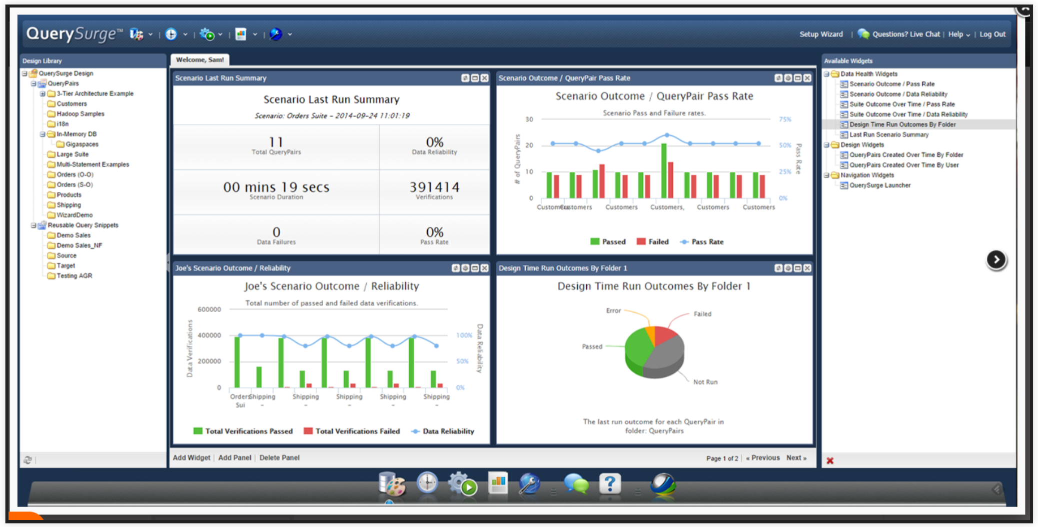
		source_order_id	status
2		2	Pending
4		4	Pending
9		9	Pending
15		15	Pending
18		18	Pending
19		19	Pending
21		21	Pending
39		39	Pending

Export to | Data Failure | Non-matching Row | Displaying 1 - 7 of 1373

		source_order_id	status
2		2	Incomplete
4		4	Incomplete
9		9	Incomplete
15		15	Incomplete
18		18	Incomplete
19		19	Incomplete
21		21	Incomplete
39		39	Incomplete

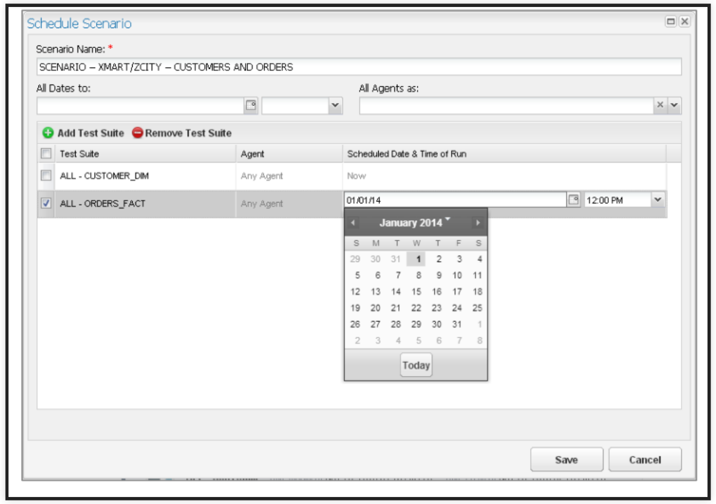
Export to | Data Failure | Non-matching Row | Displaying 1 - 7 of 1373

It is All-Singing, All-Dancing



QuerySurge
Setup Wizard | Questions? Live Chat | Help | Log Out
Design Library
Welcome, Sam!
Scenario Last Run Summary
Scenario: Orders Suite - 2014-09-24 11:01:19
11
Total QueryPairs
0%
Data Reliability
00 mins 19 secs
Scenario Duration
391414
Verifications
0
Data Failures
0%
Pass Rate
Scenario Outcome / QueryPair Pass Rate
Scenario Pass and Failure rates.
Passed
Failed
Pass Rate
Joe's Scenario Outcome / Reliability
Total number of passed and failed data verifications.
Total Verifications Passed
Total Verifications Failed
Data Reliability
Design Time Run Outcomes By Folder 1
Error
Failed
Passed
Not Run
The last run outcome for each QueryPair in folder: QueryPairs
Available Widgets
Data Health Widgets
Scenario Outcome / Pass Rate
Scenario Outcome / Data Reliability
Suite Outcome Over Time / Pass Rate
Suite Outcome Over Time / Data Reliability
Design Time Run Outcomes By Folder
Last Run Scenario Summary
Design Widgets
QueryPairs Created Over Time By Folder
QueryPairs Created Over Time By User
Navigation Widgets
QuerySurge Launcher
Add Widget | Add Panel | Delete Panel
Page 1 of 2 | « Previous Next »

Has a Scheduler



Schedule Scenario

Scenario Name: *
SCENARIO – XMART/ZCITY – CUSTOMERS AND ORDERS

All Dates to:

All Agents as:

Add Test Suite · Remove Test Suite

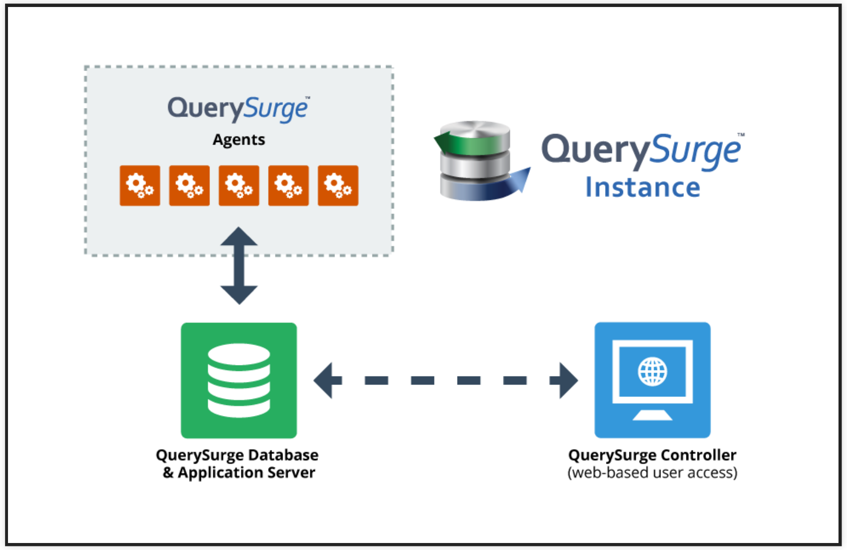
	Test Suite	Agent	Scheduled Date & Time of Run
☐	ALL - CUSTOMER_DIM	Any Agent	Now
☑	ALL - ORDERS_FACT	Any Agent	01/01/14 12:00 PM

January 2014

S	M	T	W	T	F	S
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

- Timing/Scheduling is a major cause of flakiness in Data Testing
- BUT - need to use inter-application/enterprise scheduler

QuerySurge Architecture

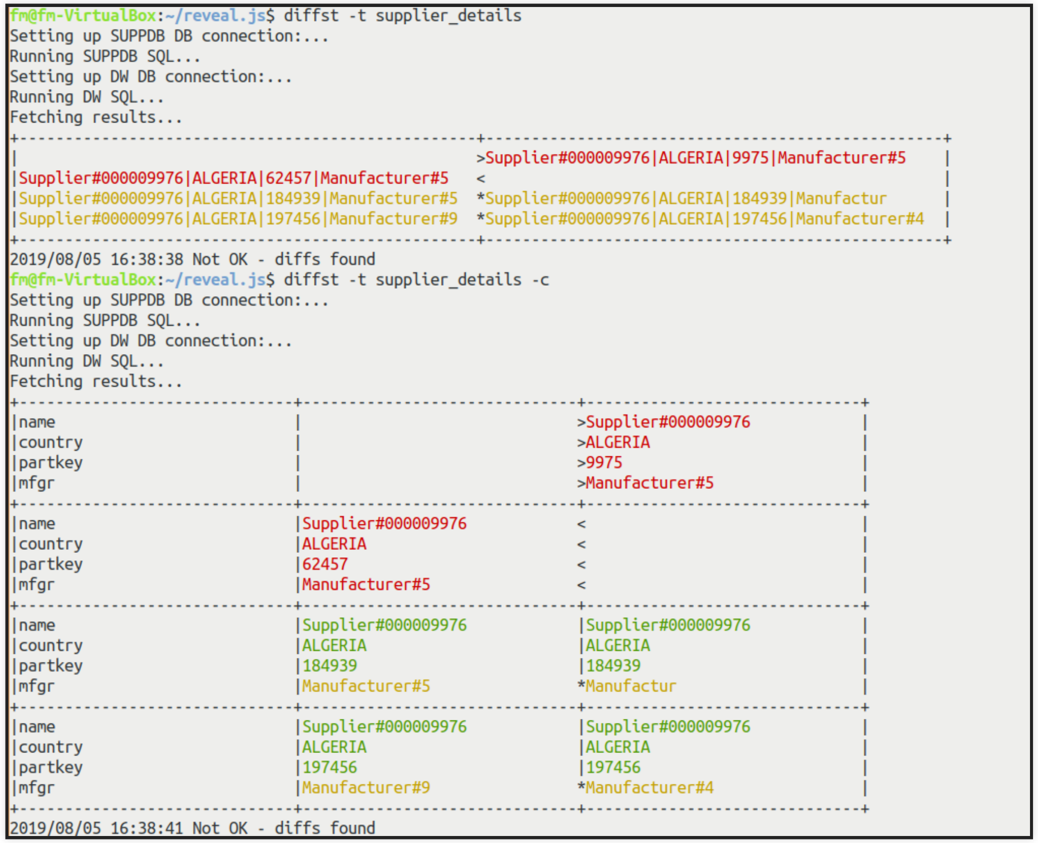


ready
ready MACHINERY MOVERS

Alternative Solution - Based on Unix Philosophy

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.

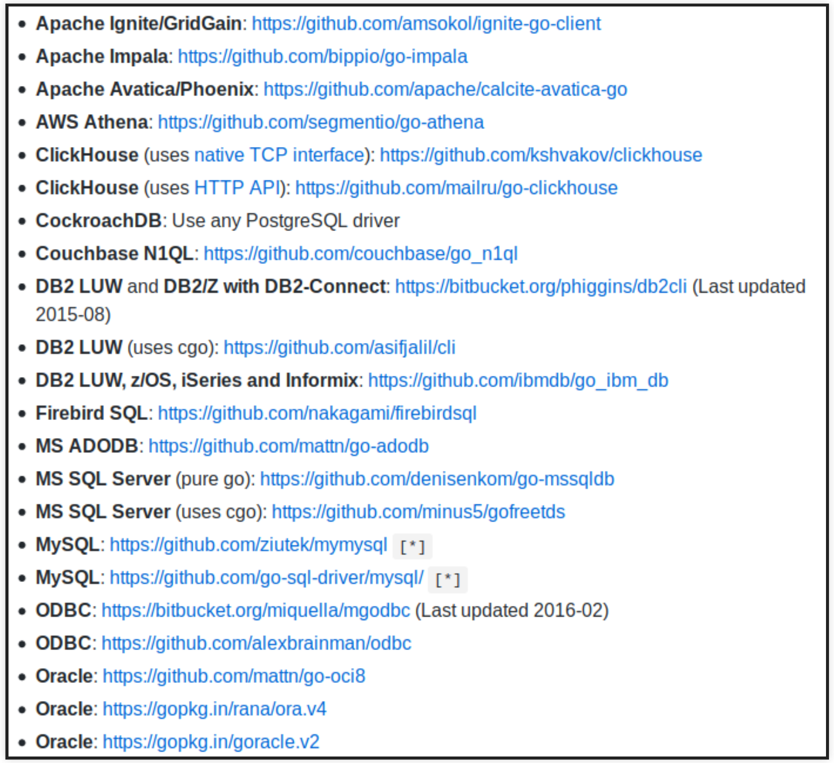
Peter H. Salus, A Quarter-Century of Unix (1994)



```
fm@fm-VirtualBox:~/reveal.js$ diffst -t supplier_details
Setting up SUPPDB DB connection:...
Running SUPPDB SQL...
Setting up DW DB connection:...
Running DW SQL...
Fetching results...
+--------------------------------------------------+--------------------------------------------------+
|                                                  >Supplier#000009976|ALGERIA|9975|Manufacturer#5    |
|Supplier#000009976|ALGERIA|62457|Manufacturer#5   <                                                  |
|Supplier#000009976|ALGERIA|184939|Manufacturer#5  *Supplier#000009976|ALGERIA|184939|Manufactur      |
|Supplier#000009976|ALGERIA|197456|Manufacturer#9  *Supplier#000009976|ALGERIA|197456|Manufacturer#4  |
+--------------------------------------------------+--------------------------------------------------+
2019/08/05 16:38:38 Not OK - diffs found
fm@fm-VirtualBox:~/reveal.js$ diffst -t supplier_details -c
Setting up SUPPDB DB connection:...
Running SUPPDB SQL...
Setting up DW DB connection:...
Running DW SQL...
Fetching results...
+------------------------------+------------------------------+------------------------------+
|name                          |                              >Supplier#000009976            |
|country                       |                              >ALGERIA                       |
|partkey                       |                              >9975                          |
|mfgr                          |                              >Manufacturer#5                |
+------------------------------+------------------------------+------------------------------+
|name                          |Supplier#000009976            <                              |
|country                       |ALGERIA                       <                              |
|partkey                       |62457                         <                              |
|mfgr                          |Manufacturer#5                <                              |
+------------------------------+------------------------------+------------------------------+
|name                          |Supplier#000009976            |Supplier#000009976            |
|country                       |ALGERIA                       |ALGERIA                       |
|partkey                       |184939                        |184939                        |
|mfgr                          |Manufacturer#5                *Manufactur                    |
+------------------------------+------------------------------+------------------------------+
|name                          |Supplier#000009976            |Supplier#000009976            |
|country                       |ALGERIA                       |ALGERIA                       |
|partkey                       |197456                        |197456                        |
|mfgr                          |Manufacturer#9                *Manufacturer#4                |
+------------------------------+------------------------------+------------------------------+
2019/08/05 16:38:41 Not OK - diffs found
```

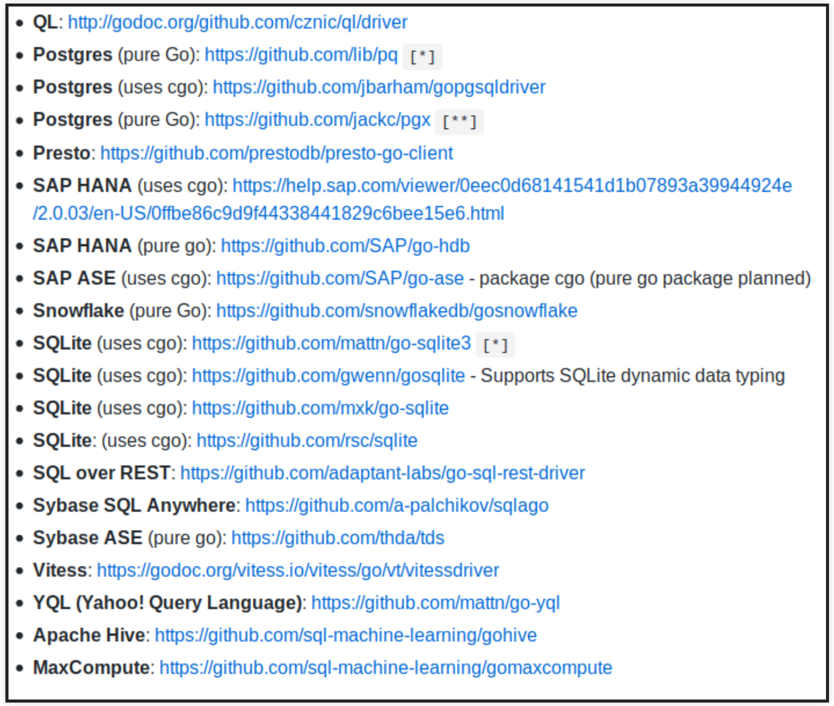
Limitations

- SQL-accessibility
- Limited by DB Drivers



- **Apache Ignite/GridGain**: https://github.com/amsokol/ignite-go-client
- **Apache Impala**: https://github.com/bippio/go-impala
- **Apache Avatica/Phoenix**: https://github.com/apache/calcite-avatica-go
- **AWS Athena**: https://github.com/segmentio/go-athena
- **ClickHouse** (uses native TCP interface): https://github.com/kshvakov/clickhouse
- **ClickHouse** (uses HTTP API): https://github.com/mailru/go-clickhouse
- **CockroachDB**: Use any PostgreSQL driver
- **Couchbase N1QL**: https://github.com/couchbase/go_n1ql
- **DB2 LUW** and **DB2/Z with DB2-Connect**: https://bitbucket.org/phiggins/db2cli (Last updated 2015-08)
- **DB2 LUW** (uses cgo): https://github.com/asifjalil/cli
- **DB2 LUW, z/OS, iSeries and Informix**: https://github.com/ibmdb/go_ibm_db
- **Firebird SQL**: https://github.com/nakagami/firebirdsql
- **MS ADODB**: https://github.com/mattn/go-adodb
- **MS SQL Server** (pure go): https://github.com/denisenkom/go-mssqldb
- **MS SQL Server** (uses cgo): https://github.com/minus5/gofreetds
- **MySQL**: https://github.com/ziutek/mymysql `[*]`
- **MySQL**: https://github.com/go-sql-driver/mysql/ `[*]`
- **ODBC**: https://bitbucket.org/miquella/mgodbc (Last updated 2016-02)
- **ODBC**: https://github.com/alexbrainman/odbc
- **Oracle**: https://github.com/mattn/go-oci8
- **Oracle**: https://gopkg.in/rana/ora.v4
- **Oracle**: https://gopkg.in/goracle.v2

Source and Target database options - 1



- **QL**: http://godoc.org/github.com/cznic/ql/driver
- **Postgres** (pure Go): https://github.com/lib/pq `[*]`
- **Postgres** (uses cgo): https://github.com/jbarham/gopgsqldriver
- **Postgres** (pure Go): https://github.com/jackc/pgx `[**]`
- **Presto**: https://github.com/prestodb/presto-go-client
- **SAP HANA** (uses cgo): https://help.sap.com/viewer/0eec0d68141541d1b07893a39944924e/2.0.03/en-US/0ffbe86c9d9f44338441829c6bee15e6.html
- **SAP HANA** (pure go): https://github.com/SAP/go-hdb
- **SAP ASE** (uses cgo): https://github.com/SAP/go-ase - package cgo (pure go package planned)
- **Snowflake** (pure Go): https://github.com/snowflakedb/gosnowflake
- **SQLite** (uses cgo): https://github.com/mattn/go-sqlite3 `[*]`
- **SQLite** (uses cgo): https://github.com/gwenn/gosqlite - Supports SQLite dynamic data typing
- **SQLite** (uses cgo): https://github.com/mxk/go-sqlite
- **SQLite**: (uses cgo): https://github.com/rsc/sqlite
- **SQL over REST**: https://github.com/adaptant-labs/go-sql-rest-driver
- **Sybase SQL Anywhere**: https://github.com/a-palchikov/sqlago
- **Sybase ASE** (pure go): https://github.com/thda/tds
- **Vitess**: https://godoc.org/vitess.io/vitess/go/vt/vitessdriver
- **YQL (Yahoo! Query Language)**: https://github.com/mattn/go-yql
- **Apache Hive**: https://github.com/sql-machine-learning/gohive
- **MaxCompute**: https://github.com/sql-machine-learning/gomaxcompute

Source and Target database options - 2

diffst - About

- Developed in Go (golang)
- Go programs compile into statically linked executables for Windows/Linux/Mac/...
- ⇒ No runtime required

Advantages of bespoke tool

- Extends the reach of manual testing
- Bug lifecycle simpler, faster, better
- CLI, numeric exit status, text input and output supports:
  - Job scheduling
  - Interface with test management tool via REST/SOAP API
  - Customizable test suites with scripting
  - Deployability - tester laptop or server
  - Test case management and maintainability thru version control

Questions/Comments?