

A Regression Testing Framework for Financial Time-Series Databases

An Effective Combination of FitNesse, Scala, and
KDB/Q

Roberto Salama – Morgan Stanley, New York, NY

[Project Goals]

- Develop a single database to store both tick by tick data and end-of-day data
- Retire two relational time-series databases.
- Offer a scalable platform that can cope with increasing volumes
- Currently at 1PB footprint expect to double in the by 2013

[KDB / Q]

- KDB is a column-based database
- Operates in both in-memory and disk based modes
- Q is a vector based language with roots in APL.
- Simple type system:
scalars, lists, dictionaries, and tables.

[Testing infrastructure]

- Chose FitNesse (Bob Martin, Ward Cunningham)
- Wrote fixtures in scala which executed dynamic code in Q
- Exposed results through FitNesse wiki tables.
- Tests execute queries in legacy and new database.
- FitNesse tables provide an efficient way to inspect pass/fail conditions

[Q basics]

- Create a list

q)l:til 10

0 1 2 3 4 5 6 7 8 9

- 10 random numbers [0..2]

q)rv:10?2.0

1.566737 0.8199122 1.221763 0.9952983
0.8175089 0.8994621 0.02784152 1.429756...

- **q)bp:10 + rv**

11.56674 10.81991 11.22176 10.9953 10.81751
10.89946 10.02784 11.42976 10.3893 10.18118

[Q basics]

- Build a dictionary with our vectors
- `q)d:`vd`id`ap`bp!(vd;id;ap;bp)`

```
vd| 2011.01.01 2011.01.02 2011.01.03 2011.01.04 2011.01.05 2011.01.06 2011.01.07 2011.01.08 2011.01.09 2011.01.10
```

```
id| aa    aa    aa    aa    aa    aa    aa    aa    aa    aa
```

```
bp| 11.56674 10.81991 11.22176 10.9953 10.81751 10.89946 10.02784 11.42976 10.3893 10.18118
```

```
ap| 10.32553 11.37695 11.63551 11.50402 10.21736 11.91979 10.07337 11.2862 11.34175 11.35782
```

- Average of vector at key bp:
`q)avg d`bp`
10.83488
- `avg d`bp + avg`ap`

[Q basics]

- Alternate syntax to create a table:

1. q)t:([vd:vd; id:id]; ap:ap; bp:bp)
2. q)t:2!([[] vd: 2011.01.01+ til 10;

```
id: 10#`aa;  
bp: 10 + 10?2.0;  
ap: 10 + 10?2.0)
```

vd	id	bp	ap
2011.01.01	aa	11.56674	10.32553
2011.01.02	aa	10.81991	11.37695
2011.01.03	aa	11.22176	11.63551
2011.01.04	aa	10.9953	11.50402
2011.01.05	aa	10.81751	10.21736
2011.01.06	aa	10.89946	11.91979
2011.01.07	aa	10.02784	10.07337
2011.01.08	aa	11.42976	11.2862
2011.01.09	aa	10.3893	11.34175
2011.01.10	aa	10.18118	11.35782

[Q basics]

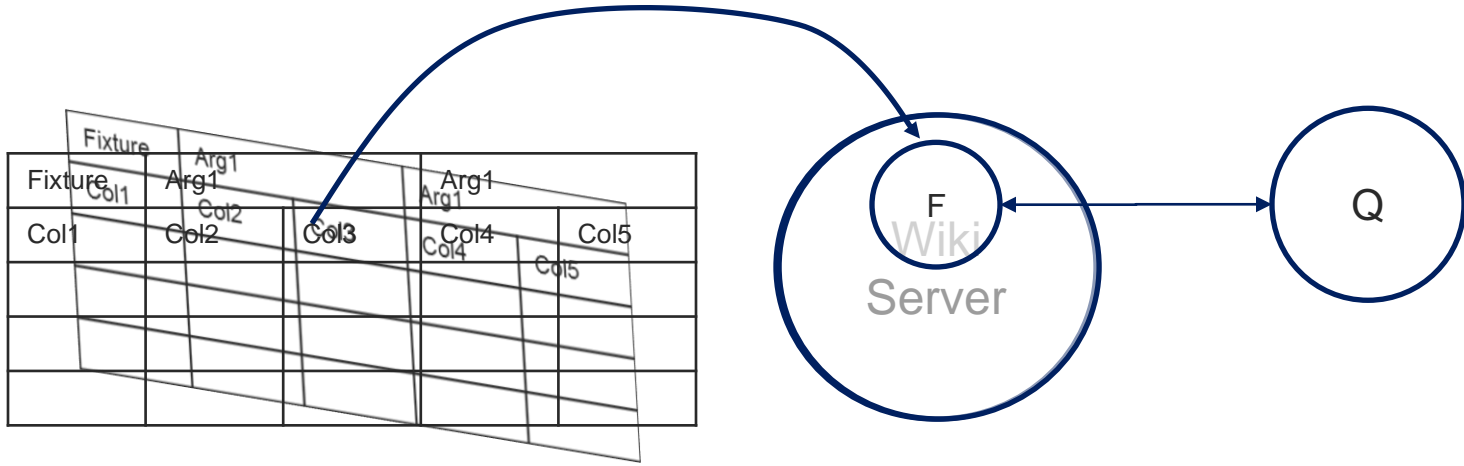
- q)select from t where ap>bp

vd	id	bp	ap
2011.01.01	aa	11.56674	10.32553
2011.01.05	aa	10.81751	10.21736
2011.01.08	aa	11.42976	11.2862

[Q basics]

- The test framework executes:
passFunction(evaluationFunction(args))
- Evaluation function – create a table:
**q)ev: {[n] 2!([] vd: vd: 2011.01.01+ til n; id: n#`a;
bp: 10 + n?2.0; ap: 10 + n?2.0)}**
- Pass function – at least one row where $ap > bp$
q)pass: {[t] 0 < count select from t where ap < bp}
- Validate is just a function composition:
q)pass ev n => pass[ev[n]]
1b

[Simple architecture]



How does it work?

The screenshot shows a web browser displaying test results for a class named `com.ms.HorizonFixture`. The page is titled "Test Results: HorizonFitness.HorizonFitness.HorizonCommandTestSuite.DemoTest".

Annotations on the page include:

- A box labeled "Class" pointing to `com.ms.HorizonFixture`.
- A box labeled "Constructor arguments" pointing to the "Q" column header.
- A box labeled "Methods" pointing to the "pass?" column header.
- A box labeled "execute()" pointing to the "true" result in the "pass?" column.
- A box labeled "reset()" pointing to the "reset()" method name in the "eval" column.

The main content area is titled "FUNCTIONS TO TEST BASIC CONNECTIVITY" and contains a table with the following data:

name	eval	pass	pass?
TIME	z N	{-16h-type x}	true
PROCESS	{z N; z i; z K} {[y] 8/[1h, (-16h-type x);x:0];x:2 63} @' y}		true

Below this is a section titled "TEST THAT DBIDX / DBINFO MATCH" with a table:

name	eval	pass	pass?	msgs
Tables	DBmatchFitNesse['devsys]	{0-count x}	[false] expected [true]	.ResultsTables
Functions	DBfuncMatchFitNesse['devsys]	{0-count x}	true	.ResultsFunctions
HDBDateRanges	DBhdbDateRanges['devsys]	{0b}	[false] expected [true]	.ResultsHdbDateRanges

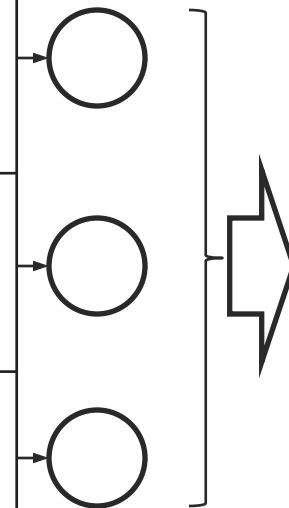
At the bottom is a section titled "HDB TEST - OCEAN / KDB COMPARISON" with a table:

name	eval	pass	pass?	msgs
			[false]	

Testing across two databases

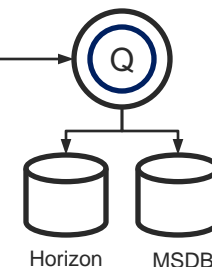
Actor Fixture	qa	Async.q	40	
name	pass?	msgs	eval	pass
S1	>98	S1	hthPullData[`684 1.F;`GFLR.NS; 2011.02.11 2011.03.18	{round[2]x `pass}
S2	=100	S2	hthPullData[`NA RA.DF;`MS.N; 2011.02.11 2011.03.18	{round[2]x `pass}
S3	=100	S3	hthPullData[`NA. F;`APPL..O; 2011.02.11 2011.03.18	{round[2]x `pass}

Actors



Statistics
Collector

S10	=100	S10	hthPullData[`I BM.N;`GE..N; 2011.02.11 2011.03.18	{round[2] x`pass}
-----	------	---------------------	--	----------------------



Horizon

MSDB

[Comparing tables]

t		
vd	tp	vss
4/14/2010	131.25	2018700
4/15/2010	130.89	1688100
4/16/2010	130.63	3163800
4/19/2010	132.23	2195400
4/20/2010	129.69	3605000
4/21/2010	128.99	1952600
4/22/2010	129.23	1720400

u		
vd	tp	vss
4/14/2010	131.35	2018700
4/15/2010	130.89	1688100
4/16/2010	130.63	3163800
4/19/2010	132.23	2195400
4/20/2010	129.69	3605000
4/21/2010	128.99	1952800
4/22/2010	129.23	1720400

cmpTbl: t,"u

cmpTbl		
vd	tp	vss
4/14/2010	131.25	2018700
	131.35	2018700
4/15/2010	130.89	1688100
	130.89	1688100
4/16/2010	130.63	3163900
	130.63	3163800
4/19/2010	132.23	2195400
	132.23	2195400
4/20/2010	129.69	3605000
	129.69	3605000
4/21/2010	128.99	1952600
	128.99	1952800
4/22/2010	129.13	1720400
	129.23	1720400

[Comparing tables]

cmpTbl		
vd	tp	vss
4/14/2010	131.25	2018700
	131.35	2018700
4/15/2010	130.89	1688100
	130.89	1688100
4/16/2010	130.63	3163900
	130.63	3163800
4/19/2010	132.23	2195400
	132.23	2195400
4/20/2010	129.69	3605000
	129.69	3605000
4/21/2010	128.99	1952600
	128.99	1952800
4/22/2010	129.13	1720400
	129.23	1720400



truthTbl: ?[cmpTbl;();0b;f!(each;{**cmp** x}),/:
f:cols value cmpTbl]

cmp:{all (first x) {sp[type x]. (x;y)}/: x}



-9| {1e-5>abs x-y}
-6| {x=y}

truthTbl		
vd	tp	vs
4/14/2010	0	1
4/15/2010	1	1
4/16/2010	1	1
4/19/2010	1	1
4/20/2010	1	1
4/21/2010	1	0
4/22/2010	1	1



htmlTbl			
vd	tp	vss	
4/14/2010	131.25 131.35	2018700	2018700
4/15/2010	130.89 130.89	1688100	1688100
4/16/2010	130.63 130.63	3163900	3163900
4/19/2010	132.23 132.23	2195400	2195400
4/20/2010	129.69 129.69	3605000	3605000
4/21/2010	128.99 128.99	1952600	1952800
4/22/2010	129.13 129.13	1720400	1720400

[Output]

Statistics Fixture			
n?	avg?	sdev?	exceptions?
17	99.96	0.15	0

Summary Fixture	mismatches	num	url
spre ad5	2532	146	[T1@0_6,T1@0_8,...]
shrout	1002	26	[T1@1_4,T1@1_6,...]
expVol	936	24	[T1@1_4,T1@1_6,...]
currencyCode	858	25	[T2@0_2,T2@0_4,...]
volatility	538	23	[T1@2_4,T2@0_2,T2@0_3,...]

[Output]

By Ticker Fixture	num	tickers
spre ad5	359	`012340.KQ`015390.KQ....
rawbp_filter	76	`ACNB.OB`ADMZ0....
rawap_filter	74	`ACNB.OB`ADMZ0...
shrout	51	`0303.DE`0303.MU`035620.KQ...
expVol	49	`0303.DE`0303.MU`035620.KQ`036000.KQ`...

By Test Name Fixture	num	tickers
T1@0_6	1	`012340.KQ
T1@0_8	1	`015390.KQ
T1@1_0	1	`017510.KQ
T1@1_4	4	`0303.DE`0303.MU,...
T1@1_5	5	`033110.KQ`033290.KQ,...E
T1@1_6	7	`035620.KQ`035760.KQ`036000.KQ`036010.KQ...

[Conclusions]

- Created a testing framework leveraging existing technologies.
- Scalable test framework
- Instantly made the QA team more productive.
- Currently manage over 10,000 regression tests.

[Questions / Remarks]
