ORMBattle.NET test results

Visit http://ormbattle.net for details

LINQ implementation

Test name	Minimum	Maximum	SqlClient	Entity Framework	LINQ to SQL	BLToolkit	DataObjects.Net	Lightspeed	NHibernate	OpenAccess	Subsonic	Unit
LINQ implementation												
Aggregates	0	5	n/a	0	0	4	0	3	3	2	n/a	f,a
All/Any/Contains	0	6	n/a	3	1	6	0	6	4,2	4	n/a	f,a
Complex	0	6	n/a	1	0	6	0	6	6	5	n/a	f,a
Element operations	0	9	n/a	4	2	6	0	6	6	5	n/a	f,a
Filtering	0	12	n/a	4,2	2,2	1	0	5,5	6,1	2	n/a	f,a
Grouping	0	10	n/a	1	1	6	0	10	10,2	5	n/a	f,a
Join	0	4	n/a	1	0	4	0	4	4	2	n/a	f,a
Ordering	0	8	n/a	3,2	2	1	0	5,1	6	3	n/a	f,a
Projections	0	13	n/a	2	1	8	0	9,2	6,1	3	n/a	f,a
References	0	4	n/a	0	0	4	0	4	3	1	n/a	f,a
Set operations	0	9	n/a	0	0	5,1	0	5,1	6,2	4,1	n/a	f,a
Standard functions	0	21	n/a	9	1	13	0	8	16	6,1	n/a	f,a
Take/Skip	0	5	n/a	1	0	1	0	2,1	2,1	2	n/a	f,a
Type casts	0	5	n/a	1	1	3	0	3,1	4	2	n/a	f,a
LINQ Implementation total:												
Performed	0	117	n/a	117	117	117	117	117	117	117	n/a	#
Passed	0	117	n/a	87	106	49	117	41	35	71	n/a	#
Failed	0	117	n/a	30	11	68	0	76	82	46	n/a	#
Properly	0	117	n/a	26	9	67	0	65	73	44	n/a	#
Asserted	0	117	n/a	4	2	1	0	11	9	2	n/a	#
Score	0	100	n/a	74,4	90,6	41,9	100	35	29,9	60,7	n/a	%

Color bar — Worst result … Best result

Units:	
f/a	total count of failed tests [, count of tests failed with assertion], less is better (0 is ideal)
#	count
%	percentage (% of passed tests), more is better

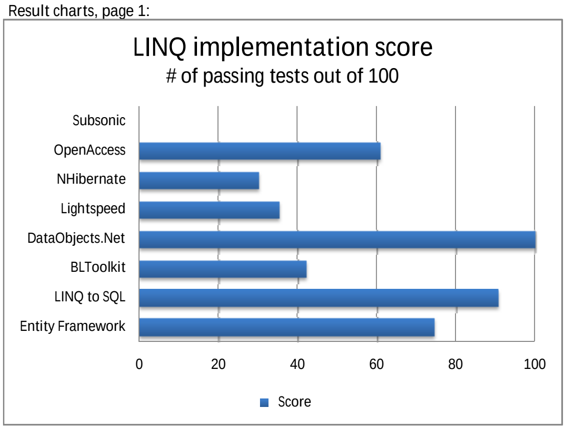
1000 item sequence

Test name	Minimum	Maximum	SqlClient	Entity Framework	LINQ to SQL	BLToolkit	DataObjects.Net	Lightspeed	NHibernate	OpenAccess	Subsonic	Unit
CRUD Performance:												
Fetch	0		21 854	4634	9280	13001	9433	10444	7289	6611	8765	op/s
Single Operation:												
Create Instance	0		19 043	4566	3670	11832	7441	4851	3741	9848	4167	op/s
Update Instance	0		18 749	3898	1493	11507	9948	4955	871	9609	4099	op/s
Remove Instance	0		19 334	5955	1670	12047	10785	4776	1440	10936	4354	op/s
CUD Average	0		18 655	4641	1942	11777	9147	4859	1421	9959	4105	op/s
Multiple Operations:												
Create Instance	0		61 981	7915	4879	22748	14988	12649	20587	10197	4010	op/s
Update Instance	0		68 567	6199	1687	64451	33320	27537	17259	15497	4154	op/s
Remove Instance	0		82 155	9365	1841	79514	42499	26199	18551	24022	4472	op/s
CUD Average	0		69 274	6997	2232	41583	24729	19538	18623	14579	4167	op/s
Data Access Performance:												
Query:												
LINQ Query	0		n/a	450	835	8749	1559	1290	1156	4168	351	queries/s
Compiled LINQ Query	0		n/a	6501	9114	13222	8046	1290	1156	4269	351	queries/s
Native Query	0		18 645	9728	9114	17187	10335	10424	5494	6590	7312	queries/s
Paging (LINQ only):												
Get Small Page (10 items)	0		n/a	5009	7736	10625	6199	1105	907	3458	322	pages/s
Get Average Page (20 items)	0		n/a	4494	7014	9009	5051	1175	726	3130	298	pages/s
Get Large Page (50 items)	0		n/a	3445	5551	6132	3410	1043	564	2447	262	pages/s
Get Huge Page (100 items)	0		n/a	2493	4108	4732	2236	892	369	1803	208	pages/s
Materialization:												
LINQ Materialize	0		n/a	257884	462149	451752	276556	390747	41555	279126	47665	objects/s
Native Materialize	0		1 276 161	476553	462149	752841	350029	437196	55515	343666	56430	objects/s
File sizes (in bytes):												
Original .cs + mapping files			n/a	58 188	15 521	9 631	8 275	14 762	32 008	61 621	n/a	bytes
Generated .cs files			n/a	126 774	92 043	0	0	32 121	0	31 190	n/a	bytes
Runtime libraries (.dlls)			n/a	2 879 448	684 032	1 224 704	3 126 272	354 304	2 210 816	3 999 232	280 064	bytes

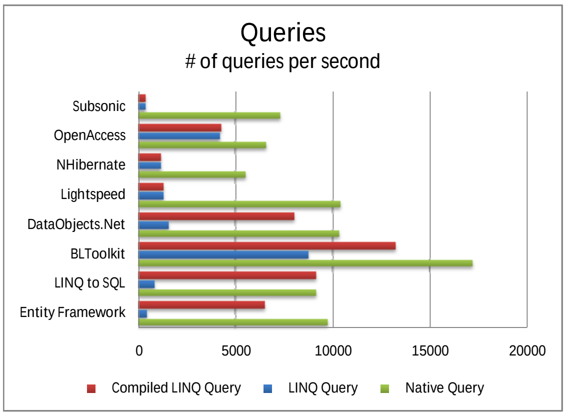
Color bar — Worst result … Best result

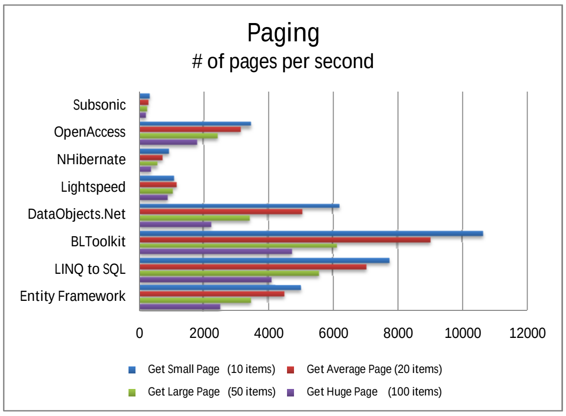
Units:	
op/s	operations per second, more is better
queries/s	queries per second, more is better
pages/s	pages per second, more is better
objects/s	# of materialized objects per second, more is better

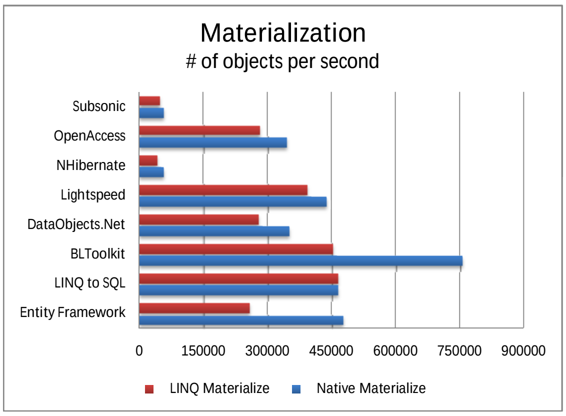
Result charts, page 1:

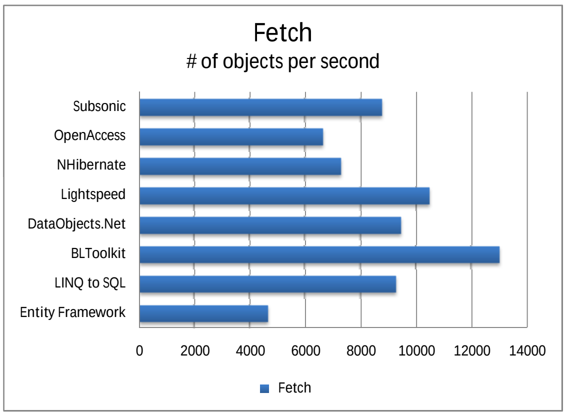


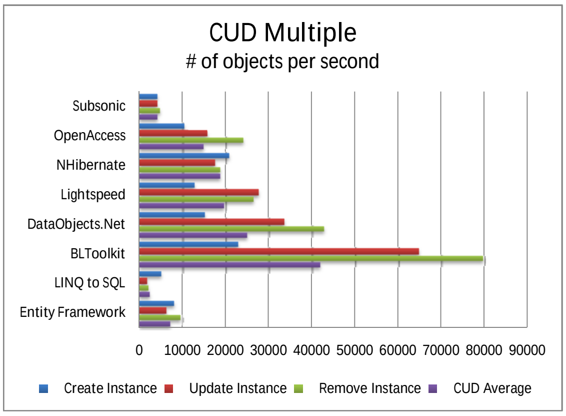
Larger = better!

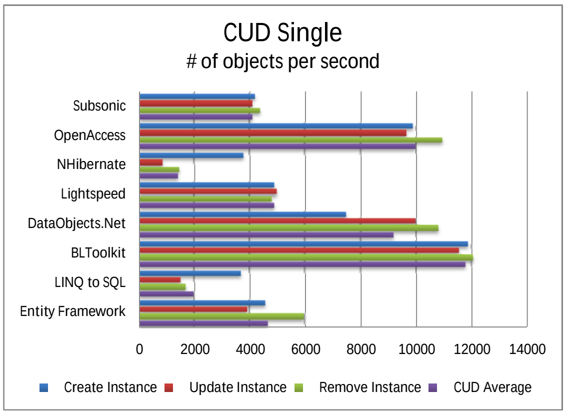












Cell: R46C7
Comment: Alex Yakunin:
Specialized API:
SqlQuery<T>.Insert(..., int batchSize, IEnumerable<T> sequence)

Cell: R46C10
Comment: Alex Yakunin:
Specialized API: ISessionFactory.OpenStatelessSession is used in this test.

Cell: R47C7
Comment: Alex Yakunin:
Specialized API:
SqlQuery<T>.Update(..., int batchSize, IEnumerable<T> sequence)

Cell: R47C10
Comment: Alex Yakunin:
Specialized API: ISessionFactory.OpenStatelessSession is used in this test.

Cell: R48C7
Comment: Alex Yakunin:
Specialized API:
SqlQuery<T>.Delete(..., int batchSize, IEnumerable<T> sequence)

Cell: R48C10
Comment: Alex Yakunin:
Specialized API: ISessionFactory.OpenStatelessSession is used in this test.

Cell: R54C9
Comment: Alex Yakunin:
Complied queries are not supported, result is copied from above cell.

Cell: R54C10
Comment: Alex Yakunin:
Complied queries are not supported, result is copied from above cell.

Cell: R54C12
Comment: Alex Yakunin:
Complied queries are not supported, result is copied from above cell.

Cell: R55C6
Comment: Alex Yakunin:
No native queries, result is copied from above cell.

Cell: R63C6
Comment: Alex Yakunin:
No native queries, result is copied from above cell.