Evaluation of queries

I. Results when N = 10,000

Data (N = 10,000)

I will show you at first the running time that I got when N=10000. All these data are the average of 3 times « warm RAM » and 1 time « cool RAM ».

PostgreSQL

	Running time (ms)
Query1	63
Query2	29,5
Query3	131
Query4	115,25
Query5	84,5
Query6	177,5
Query7	764,75

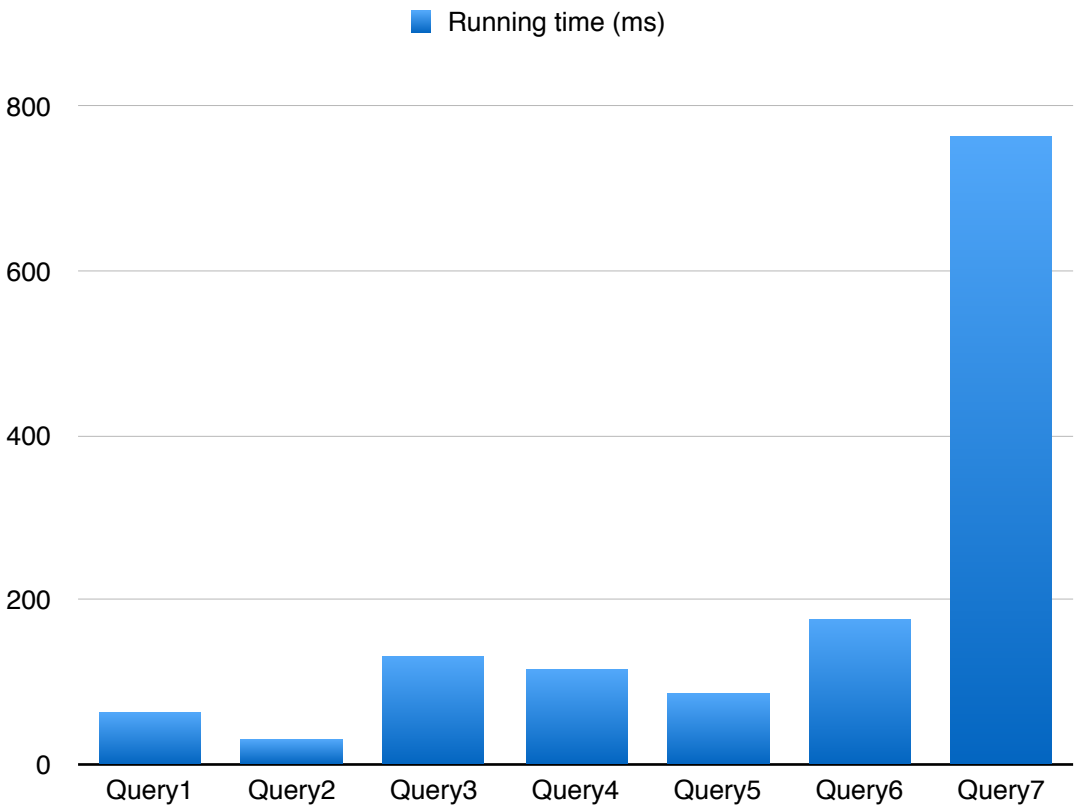
Oracle NoSQL

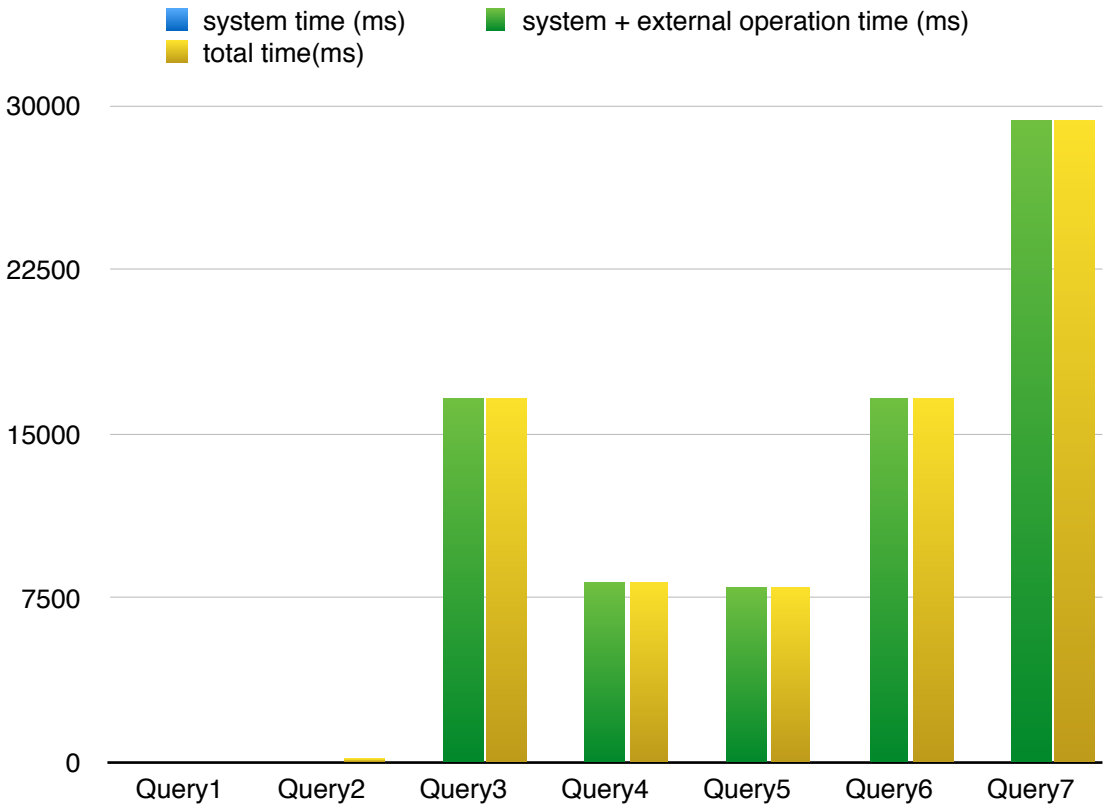
Oracle Nosql	system time (ms)	system + external operation time (ms)	total time (ms)
Query1	4	4	4
Query2	50	50	50
Query3	2654	2712	2712
Query4	3305	3326	3428
Query5	1575	1593	1608
Query6	2845	2971	2971
Query7	4693	5295	5294

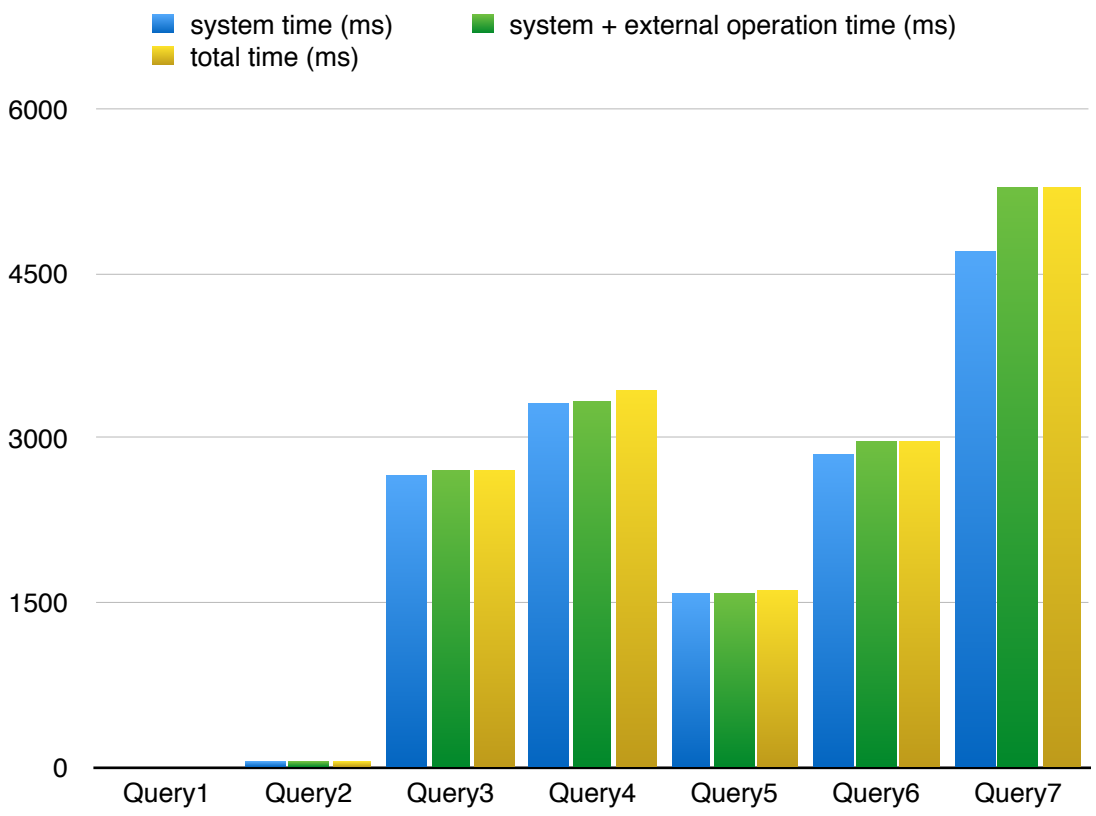
HBase

Hbase	system time (ms)	system + external operation time (ms)	total time(ms)
Query1	2	2	3
Query2	3	3	129
Query3	1	16668	16668
Query4	1	8186	8186
Query5	1	7967	7967
Query6	2	16599	16601
Query7	1	29251	29251

Graphs


Running time (ms)
800
600
400
200
0
Query1
Query2
Query3
Query4
Query5
Query6
Query7


system time (ms)
system + external operation time (ms)
total time(ms)
30000
22500
15000
7500
0
Query1
Query2
Query3
Query4
Query5
Query6
Query7



As you can see in these graphs above :

1. For the first query : the profile of user such that id = « usr50 », the NoSQL database system works better than the relational database system.
   In PostgreSQL : 63ms
   In Oracle NoSQL : 4ms
   In HBase : 3ms
2. For the second query : all the ages of users who live in « Fr ». The running time of these 3 systems is almost the same.
   In PostgreSQL : 29,5ms
   In Oracle NoSQL : 50ms
   In HBase : 129ms
3. For the others queries, the speed of PostgreSQL is much faster than that of Oracle NoSQL and HBase.

So, we can conclude : (in case of N=10,000).

1. For the very simple queries (like query 1 and query 2), NoSQL database system works better than relational database system.
2. For the complex queries like the classification queries, the queries with join, etc, relational database system works better than NoSQL database system.

The reason is that the operation is faster when implemented within the server than when all the data needs to be taken out and processing performed in the application.

P.S. We can define the « simple query » with the definition :

The query use only « get » operation and need no information processing (external operation) or need few information processing (external operation) after « get ».

If we only talk about NoSQL database systems :

1. The query 1 and query 2, their running time in Oracle NoSQL and HBase is almost the same.
2. For the others queries, Oracle Nosql run them much faster than HBase.

We can conclude :

1. Concerning these 7 queries, overall, Oracle NoSQL is faster than HBase.
2. But the system running time of HBase is much faster than that of Oracle NoSQL. As we can see from the first part , HBase use only 2ms or 3ms for each query, but Oracle NoSQL use thousands of ms.
3. HBase uses a lot of time to treat the data after getting them from system but Oracle NoSQL does not.

The reason why we get these conclusions :

1. Why the system running time of HBase is much faster than that of Oracle NoSQL ?

   At first, we should know that for the queries 3, 4, 5, 6, 7, we need to « scan » the tables at least one time for each query (two times scan for query 6 and query 7). So the running time of « scan table» is an important thing which affects the system running time.
   Because HBase is a column-oriented database, we can scan all data of one table by using a single operation « scan ».
   But for Oracle NoSQL, which is a key-value database, we can not scan all data of one table by using a single operation « scan » as we do in HBase. So, if we want to « scan » the table of Oracle NoSQL, we should do « get » operations as many as the number of tuples of the table that we want to scan.
   So this is why the system running time of HBase is much faster than that of Oracle NoSQL.

2. Why the external operations running time of Oracle NoSQL is much faster than that of HBase ?
   Another time, Oracle NoSQL is a key-value database. With the help of index (the keys), the external operations run very fast.
   But for HBase, which is column-oriented database, without the help of index, the external operations run very slowly.
   In the level of java code :
   1. We use only one « for » loop to treat the result returned by system. The complexity is O(n).
   2. We have to use two « for » (one in another) to treat the result returned by system. The complexity is O(n2).
   3. Especially, for the 6th and 7th queries which need to perform « HashJoin », HBase doesn't support stocking arrays, lists, maps in database, but Oracle NoSQL does, so we can use the nested solution of vip2p in Oracle NoSQL, but we cannot use it in HBase. Nested solution is proved more efficient than the unnested one, so this is another reason why the external operations running time of Oracle NoSQL is much faster than that of HBase.

What can we conclude based on these facts above :

1. If one query don't need external operations, HBase is an good idea. For example, the age of all users who live in « Fr ». Because the result returned by system is all we need to get (no external operations). But, if we want to run the query : the average age of usrs who live in « Fr », HBase might not be a good idea, because we should calculate its average value (it is an external operation which will take a lot of time in HBase).
2. If one query need to do a lot of external operations, Oracle NoSQL is better than HBase.

II. Results when N = 100,000 (Medium size)

Data (N = 100,000)

This time I try with N = 100,000 (medium size) which means we have :

entity	nb of tuples
users	100,000
shops	100,000
items	100,000
addr	100,000
friend	500,000
orders	1,000,000
itemline	1,000,000
visit	1,000,000

I got the data concerning running time of queries as below :

- **PostgreSQL**

	Running time (ms)
Query1	246.5
Query2	109.5
Query3	5,573.75
Query4	2,783.75
Query5	2,826.75
Query6	7,631
Query7	10,975.5

- **HBase**

	system running time (ms)	system + execution time (ms)	total time (ms)
Query1	22.25	22.25	38.5
Query2	39.75	39.75	171.25
Query3	1	161,930.25	161,930.25
Query4	10.25	76,175	76,175
Query5	1	75,376.5	75,376.5
Query6	31.5	163,535.5	163,666
Query7	98.25	300,115.75	300,222

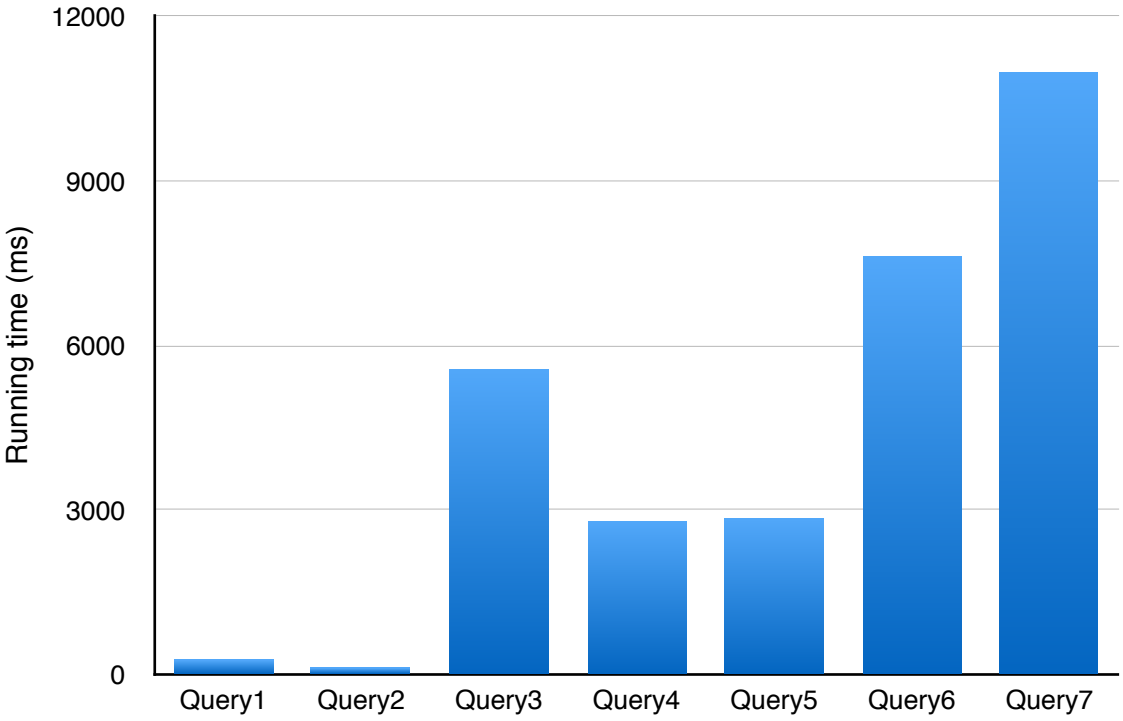
- **Oracle NoSQL**

	system running time (ms)	system + execution time (ms)	total time (ms)
Query1	14	15.25	15.25
Query2	80.75	86	91.25
Query3	13,060.25	13,205	13,205
Query4	22,538.25	22,622	23,028.75
Query5	15,423.25	15,502.5	15,592.5
Query6	53,594.75	54,161	54,161
Query7	91,399.5	93,616.75	93,960.5

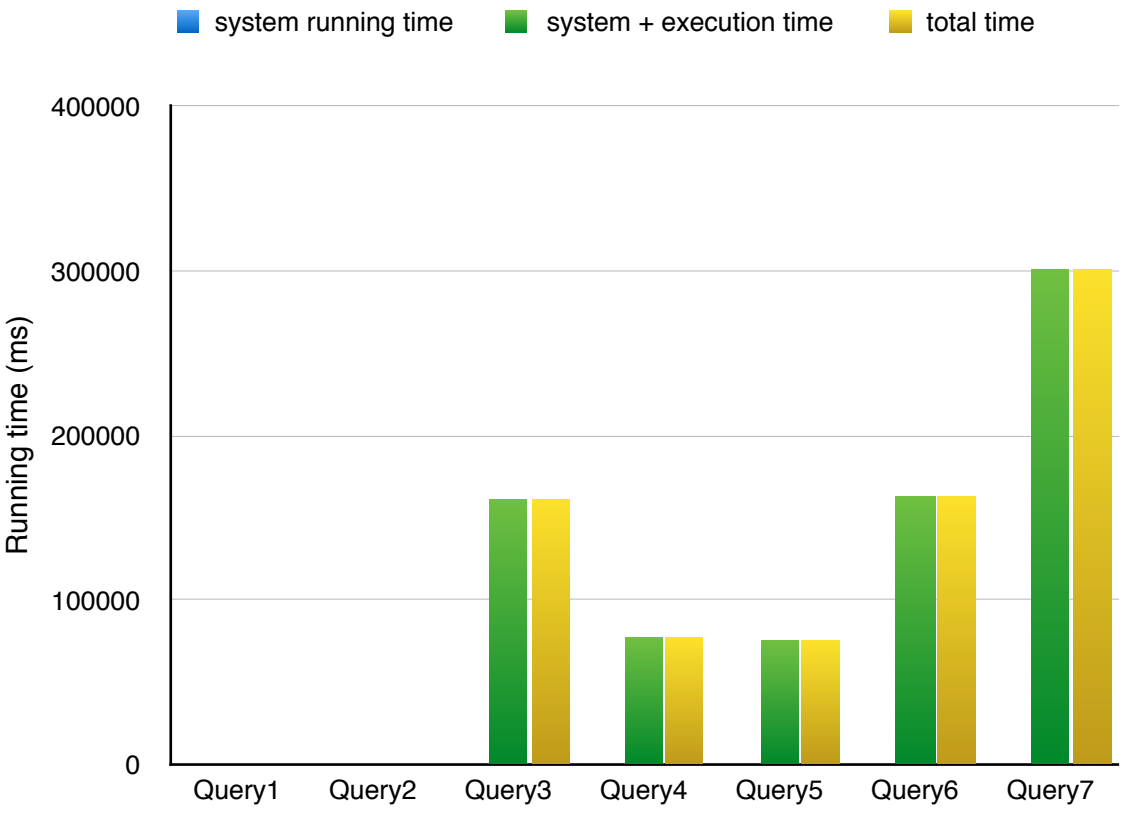
Graphs

These are the graphs that represent the data above :

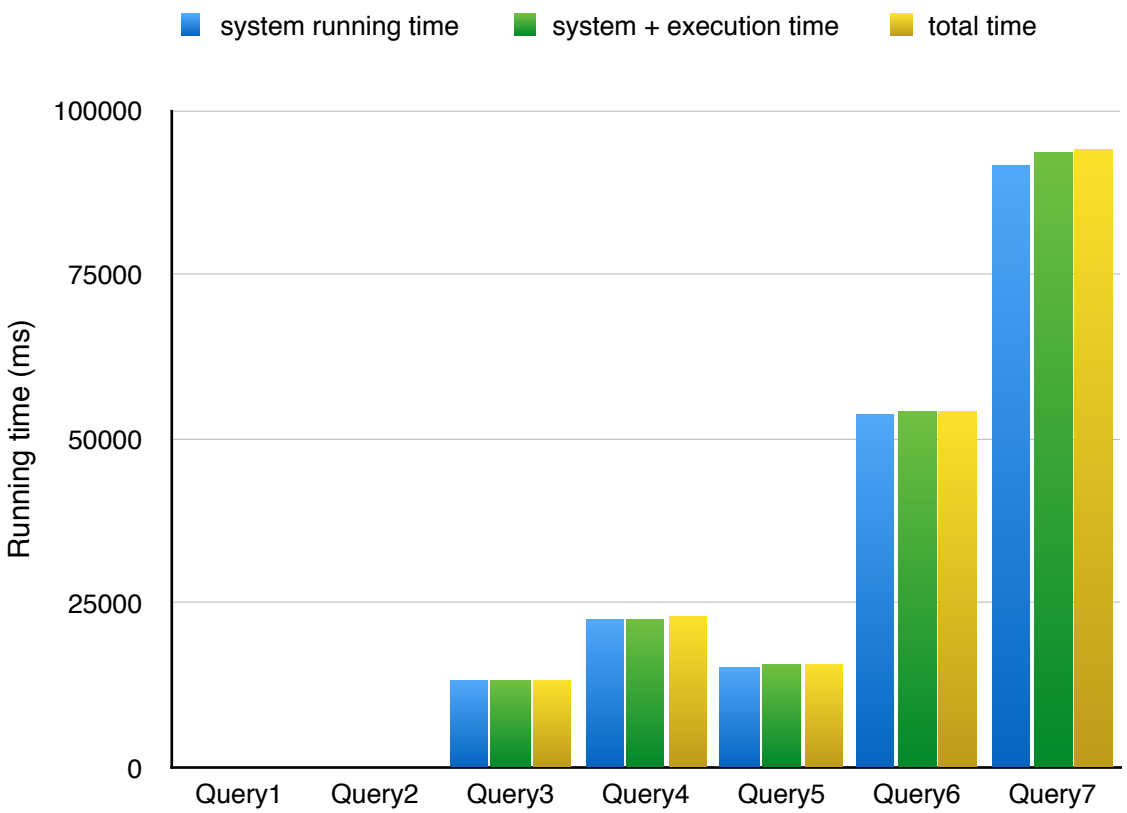
- **PostgreSQL**



- **HBase**



- **Oracle NoSQL**

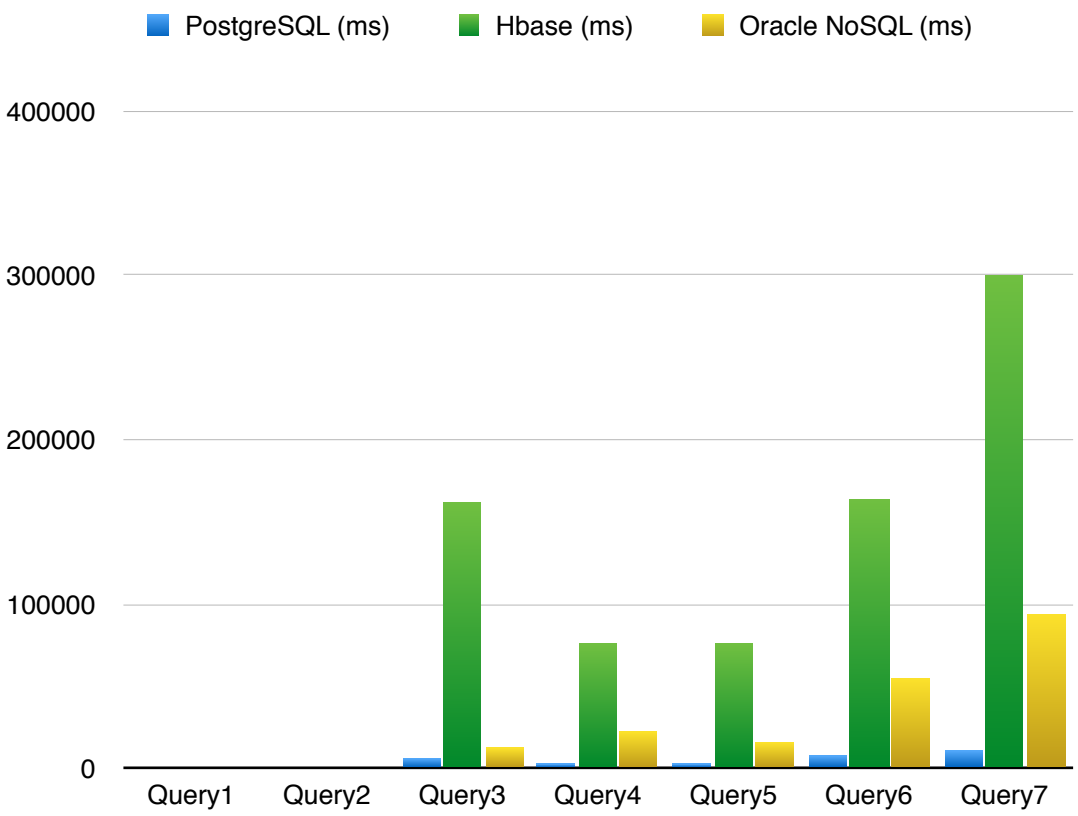


If we put the « total time » of running queries of these 3 systems together :
And we can get the data and the graph as following :

- **Data**

	PostgreSQL	Hbase	Oracle NoSQL
Query1	246.5	38.5	15.25
Query2	109.5	171.25	91.25
Query3	5,573.75	161,930.25	13,205
Query4	2,783.75	76,175	23,028.75
Query5	2,826.75	75,376.5	15,592.5
Query6	7631	163,666	54,161
Query7	10,975.5	300,222	93,960.5

- **Graph**



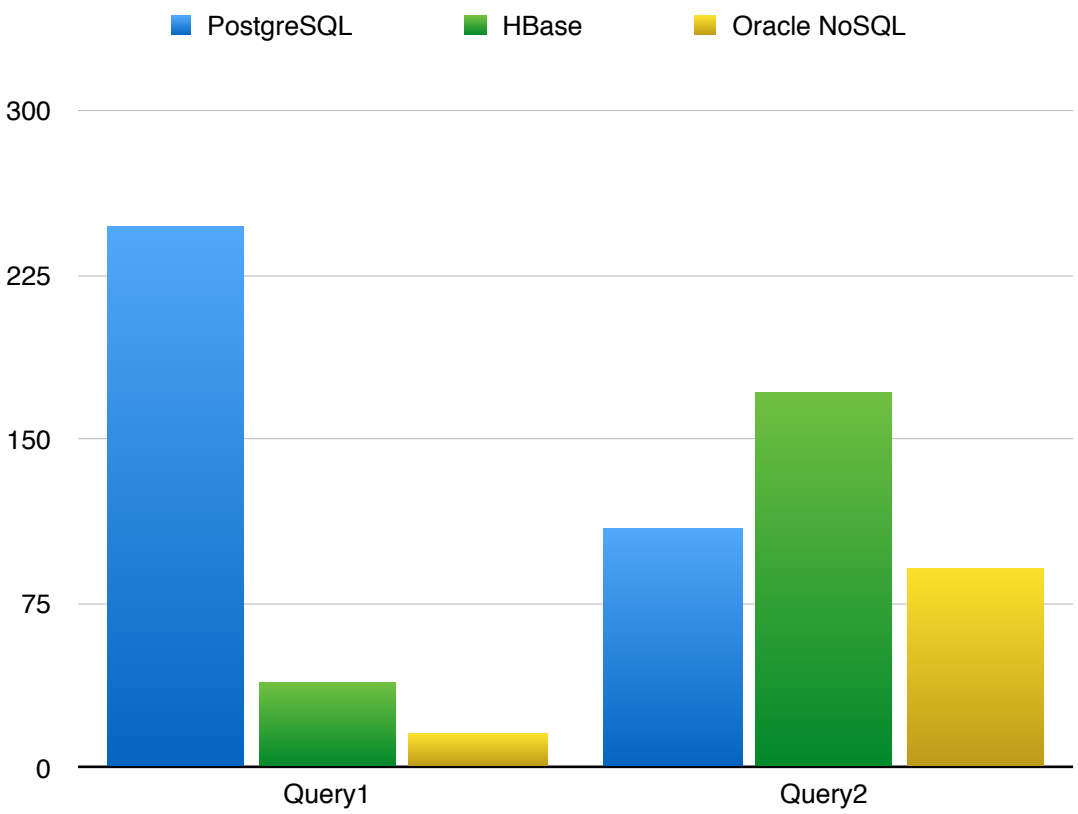
Since the result of running time for the first 2 queries are not very clear from the graph above, I made another data table and graph only for the first 2 queries :

And I got the new data table :

Running time of the first 2 queries

	PostgreSQL	Hbase	Oracle NoSQL
Query1	246.5	38.5	15.25
Query2	109.5	171.25	91.25

- **Graph**



We can see that Oracle NoSQL has the best performance for the first query (the profile of the use s.t id=« usr50 ») and the second query (the ages of all users who live in « Fr »).

This time, we get almost the same results and, obviously, we can get the same conclusion when N = 10,000.

III. Evaluation of new queries

With the advise of Francesca, we decide to add 3 new queries to evaluate the performance of the 3 systems because that we need different types of queries.
The queries :
Query 8 : the item with the highest price.
Query 9 : the medium price of item in one order.
Query 10 : the name and price of item in one shop.

This time I tried the new queries above to evaluate the performance of 3 systems with N=10,000 (I will try N=100,000 later) and I got the new results as following :

- **PostgreSQL**

	Running time
Query 8	42,75
Query 9	28
Query 10	5,25

- **HBase**

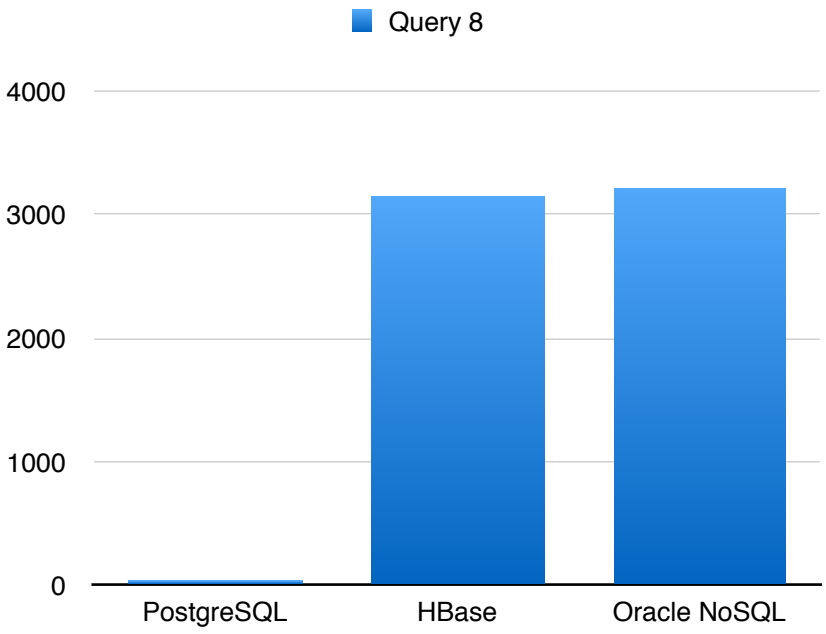
	system running time	system + external operation time	total time
Query 8	2	3153	3154
Query 9	1	3	3
Query 10	1	1	1

- **Oracle NoSQL**

	system running time	system + external operation time	total time
Query 8	3211	3216	3216
Query 9	29	29	29
Query 10	4	4	4

The results of query 8 :

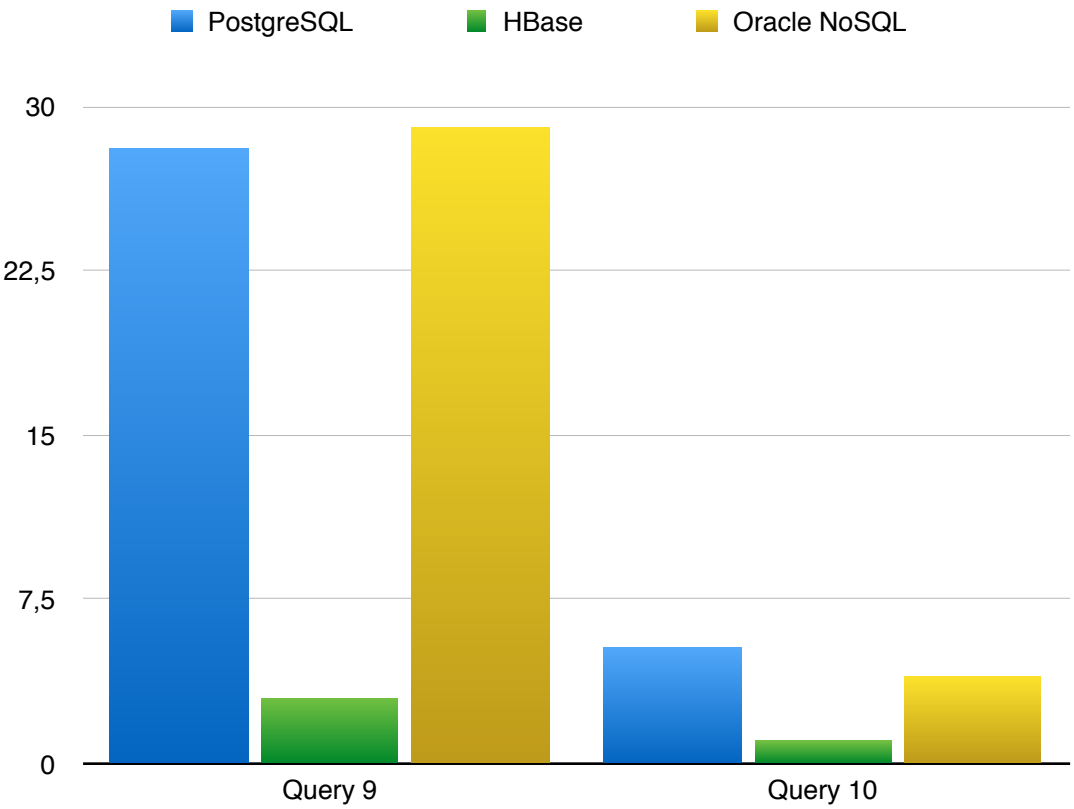
	PostgreSQL	HBase	Oracle NoSQL
Query 8	42,75	3145	3216



Obviously, the running time of PostgreSQL is the fastest. the reason is that calculating the max price of items is very costly in NoSQL database system because we need to get all the data and perform processing in application (not in server which is faster).

The results of query 9 and query 10 :
This is the data about the total time of running query 9 and query 10 :

	Query 9	Query 10
PostgreSQL	28	5,25
HBase	3	1
Oracle NoSQL	29	4



As we can see from the graph, HBase is the fastest system to run both query 9 and query 10. And PostgreSQl and Oracle NoSQL have almost the same running time.

The reason is that :
for these 2 queries, external operation is very simple because query 9 and query 10 concern only one order or only one shop. So the total running time depend only on the system time and we have proven (the first report I send you) that HBase is the fastest one between these 3 systems in the system time. So HBase is the best one to run these 2 queries.

IV. Suggestion of a query across two or more systems

We need to see at first which system has the best performance when running different queries.
Oracle NoSQL has the best performance when :
Query 1 : the profile of user such that id=« usr1 ».
Query 2 : all the ages of users from « Fr ».

HBase has the best performance when :
Query 9 : the medium price of item in one order.
Query 10 : the name and price of item in one shop.

PostgreSQL has the best performance when :
Query 3 : the number of each item to be bought
Query 4 : top 10 best selling items.
Query 5 : top 10 worst selling items.
Query 6 : top 100 most visited item
Query 7 : the friends who bought same item.
Query 8 : the item with the highest price.

According to the results above, I can make some suggestion as following :

1. For the complex queries like the across table queries, classification queries, and the queries using aggregation function, etc, PostgreSQL (relational database system) has the best performance because the data stored in database conform one unique relational model that guarantee efficiency of performing external operation like « join », « classification ».
2. For the very simple queries, NoSQL database system is more efficient than relational database system. Simple queries mean that queries need no external operation or few external information
3. If a query has no external operation, for example , a query that only return one tuple or several tuples and no external operation, HBase is the most efficient system because its system running time is the most efficient between these 3 systems.
4. If a query has few external operation, it's very hard to chose between HBase and Oracle NoSQL because we need to know the balance point between system running time and external operation running time. If the external operation time is « relative » expensive, Oracle NoSQL is better and if the external operation time is « relative » not expensive, HBase is better. But « relative » is really hard to define.