What is Claimed is:

1. A computer-implemented method for analysis of data from multiple data sources, the computer-implemented method comprising:

receiving a query instance;

processing the query instance to generate one or more relational query instances based upon, at least in part, the query instance;

distributing the generated one or more relational query instances among one or more nodes;

processing each of the generated one or more relational query instances by at least one node of one or more nodes to extract data from at least one data source of multiple data sources in communication therewith corresponding to the one or more relational query instances;

generating one or more insights based on the extracted data; and

storing the one or more insights generated in a relevant cache to be retrieved therefrom.

2. The method according to claim 1 further comprising identifying entities in the query instance and attributes associated therewith.

3. The method according to claim 2 further comprising:

retrieving historic data from at least one data source of the multiple data sources related to one or more of the entities and attributes identified;

training at least one model based on the historic data retrieved;

associating tags with the at least one trained model indicative of corresponding one or more of the entities and attributes; and

storing the at least one trained model with associated tags in at least one data source of the multiple data sources, to be retrieved therefrom.

4. The method according to claim 3 further comprising:

monitoring the at least one data source of the multiple data sources for updates in the historic data;

determining entities and attributes associated with the updates in the historic data; and

re-training the at least one model, with associated tags related to the determined entities and attributes, based on the updates in the historic data.

5. The method according to claim 1 further comprising:

determining a query context for the query instance; and

linking a cached query instance to the query context based on a matching algorithm.

6. The method according to claim 1 further comprising:

generating a hash value for the query instance received; and

using the hash value to find a query cache that matches the hash value.

7. The method according to claim 1 further comprising:

generating a query plan based on the processing of the query instance; and

processing another related query instance based on the query plan.

8. A computing system for analysis of data from multiple data sources, comprising:

an enterprise query base configured to communicate with at least one data source of the multiple data sources to access data therefrom, the enterprise query base is configured to:

receive a query instance;

process the query instance to generate one or more relational query instances based upon, at least in part, the query instance, with each of the one or more relational query instances to be processed by at least one node of one or more nodes, wherein the one or more relational query instances are distributed among the one or more nodes to extract data from the at least one data source in communication therewith corresponding to the one or more relational query instances; and

receive extracted data from each of the one or more nodes queried;

and

an analytic service in communication with the enterprise query base, wherein the analytic service is configured to:

generate one or more insights based on the extracted data; and

store the one or more insights generated in at least one data source of the multiple data sources, to be retrieved therefrom.

9. The system according to claim 15, wherein the enterprise query base is further configured to identify entities in the query instance and attributes associated therewith.

10. The system according to claim 16, wherein the enterprise query base is further configured to:

retrieve historic data from the at least one data source of the multiple data sources related to one or more of the entities and attributes identified;

train at least one model based on the historic data retrieved;

associate tags with the at least one trained model indicative of corresponding one or more of the entities and attributes; and

store the at least one trained model with associated tags in at least one data source of the multiple data sources, to be retrieved therefrom.

11. The system according to claim 17, wherein the enterprise query base is further configured to:

monitor the at least one data source of the multiple data sources for updates in the historic data;

determine entities and attributes associated with the updates in the historic data; and

re-train the at least one model, with associated tags related to the determined entities and attributes, based on the updates in the historic data.

12. The system according to claim 8, wherein the enterprise query base is further configured to determine a query context for the query instance and link a cached query instance to the determined query context based on a matching algorithm.

13. The system according to claim 15, wherein the enterprise query base is further configured to generate a hash value for the query instance received, wherein the enterprise query base is configured to use the hash value to find a query cache that most closely matches the hash value.

14. The system according to claim 15, wherein the enterprise query base is further configured to generate a query plan based on the processing of the query instance, and wherein the enterprise query base is further configured to process another related query instance based on the query plan.

15. A computer program product residing on a computer readable storage medium having a plurality of instructions stored thereon which, when executed across one or more processors, causes at least a portion of the one or more processors to perform operations comprising:

receiving a query instance;

processing the query instance to generate one or more relational query instances based upon, at least in part, the query instance;

distributing the generated one or more relational query instances among one or more nodes;

processing each of the generated one or more relational query instances by at least one node of one or more nodes to extract data from at least one data source of multiple data sources in communication therewith corresponding to the one or more relational query instances;

generating one or more insights based on the extracted data; and

storing the one or more insights generated in a relevant cache to be retrieved therefrom.

16. The computer program product according to claim 15 further comprising identifying entities in the query instance and attributes associated therewith.

17. The computer program product according to claim 16 further comprising:

retrieving historic data from at least one data source of the multiple data sources related to one or more of the entities and attributes identified;

training at least one model based on the historic data retrieved;

associating tags with the at least one trained model indicative of corresponding one or more of the entities and attributes; and

storing the at least one trained model with associated tags in at least one data source of the multiple data sources, to be retrieved therefrom.

18. The computer program product according to claim 17 further comprising:

monitoring the at least one data source of the multiple data sources for updates in the historic data;

determining entities and attributes associated with the updates in the historic data; and

re-training the at least one model, with associated tags related to the determined entities and attributes, based on the updates in the historic data.

19. The computer program product according to claim 15 further comprising:

determining a query context for the query instance; and

linking a cached query instance to the query context based on a matching algorithm.

20. The computer program product according to claim 15 further comprising:

generating a hash value for the query instance received; and

using the hash value to find a query cache that matches the hash value.