DIRA: Data Integration to Return Ranked Alternatives

Reham I. Abdel Monem
Information Systems Department,
Faculty of Computers and Information,
Cairo University, Giza, Egypt
reham@fci-cu.edu.eg

Ali H. El-Bastawissy
Faculty of Computer Science, MSA
University, Giza, Egypt
alibasta@fci-cu.edu.eg

Mohamed M. Elwakil
The Software Engineering Laboratory,
Innopolis University, Innopolis, Russia
m.elwakil@innopolis.ru

ABSTRACT
Data integration (DI) is the process of collecting data needed for answering a query from distributed and heterogeneous data sources and providing users with a unified form of this data. Data integration is strictly tied with data quality due to two main data integration challenges first, providing user with high qualitative query results second, identifying and solving values conflicts on the same real-world objects efficiently and in the shortest time. In our work, we focus on providing user with high qualitative query results.

The quality of a query result can be enhanced by evaluating the quality of the data sources and retrieving results from the significant ones only. Data quality measures are used not only for determining the significant data sources but also in ranking data integration results according to user-required quality and presenting them in a reasonable time. In this paper, we perform an experiment that shows a mechanism to calculate and store a set of quality measures on different granularities through new data integration framework called data integration to return ranked alternatives (DIRA). These quality measures are used in selecting the most significant data sources and producing top-k query results according to query types that we proposed. DIRA validation using the transaction processing performance council (TPC) benchmark version called TPC-DI will show how our framework improves the returned query results.

1. INTRODUCTION
Data integration system (DIS) is a system where query results are combined from different and autonomous data sources. These query results may be found in one source or distributed among many sources [1].

Data integration may face three types of heterogeneity: technological heterogeneities because products are used by different vendors, applied in different categories of information and communication infrastructures, schema heterogeneities due to the use of different data models and different data representations and instance heterogeneities where the same object from different data sources represented by different data values. In our work, we focused on instance heterogeneities where data quality problems become very evident and as they have big effect on the query processing in data integration.

Such systems have different architectures but virtual integration and data warehousing architectures are the most commonly used ones. In this work, we use the virtual integration architecture where many local data sources are combined together to form a single virtual data source, data are stored in local data sources and are accessed through global schema which is the presentation where users send their queries to a data integration system.

Data sources quality changes frequently so it is important to store some data sources quality measures to use them during query planning.

In our previous work (DIRA) [2], we presented data integration framework called Data Integration to return Ranked Alternatives (DIRA) that uses data quality (DQ) in data integration systems to improve their performance and query results quality. This framework adds quality system components and data quality assessment module to any data integration system.

In this paper, we report on an experiment on the DIRA framework that uses the data integration benchmark TPC_DI [3].

In this paper data quality measures and the way of their assessment are introduced in section 2. Section 3 presents the DIRA framework. Experiments on DIRA framework using the data integration benchmark TPC_DI [3] are presented in section 4. Conclusion and future work are introduced in section 5.

2. DATA QUALITY MEASURES USED IN DATA INTEGRATION
Data quality is fitness for use or the ability to meet user's needs [4]. There are a lot of measures to assess data quality called data quality measures. They are classified according to many aspects [5] and table 1 presents one of their classifications.

Table 1. Information quality measures classification for data integration [6]

Data Integration Components	IQ criteria
Data Source	Reputation, Verifiability, Availability and Response Time.
Schema	Schema Completeness, Minimalism and Type Consistency.
Data	Data Completeness, Data Timeliness, Data Accuracy and Data Validity.

In DIRA, we focused on data quality measures important for user and related to data in the integration process.

Quality measures can assess the quality of information better if we use a combination of metrics, subjective ratings and qualitative description of issues. We use that in DIRA assessment module.

There are different levels of granularities that data quality measures can be calculated according to them and table 2 presents our selected data quality measures and the granularity level for each selected measure.

Table 2. Data quality measures selected by DIRA and the granularity for each measure

Data Quality Measures	Measures Granularities		
	Data Source Level	Relation Level	Attribute Level
Completeness			✓
Validity			✓
Accuracy			✓
Timeliness		✓	

Following sub-sections explain our selected data quality measures definitions and the equations for their assessment:

2.1 Data completeness

Data completeness categorized into two types: Null-Completeness and Population-Completeness. Null-Completeness represents the extent to which data set contains missing values. Population-Completeness represents the extent to which all needed data by the user is available [5].

In DIRA, Fact-Completeness was introduced and is defined as an inferred and accurate type of completeness which value is assessed using Null-Completeness and Population-Completeness.

Completeness types assessment at attribute level will be concluded as follows (Where a_m (r) is attribute number m in relation r) [7]:

- Null-Completeness assessment (C_{Null}): It is the percentage of existing values (non-null values) to the whole number of values in the universal relation.

$$C_{Null}(a_m\ (\mathrm{r})) \frac{\text{Number of non}-\text{null values } a_m\,(\mathrm{r})}{\text{Total number of values in the universal relation}} (1)$$

- Population-Completeness assessment ($C_{Population}$): It is the percentage of actually presented rows in a relation r to the number of rows in ref(r) where ref(r), is the relation of all rows that satisfy r relational schema.

$$C_{Population}(a_m\ (\mathrm{r})) = \frac{Cardinality\ of\ a_m\,(\mathrm{r})}{Cardinality\ of\ ref(r)} \quad (2)$$

- Fact-Completeness assessment (C_{fact}): It is subtraction of the number of missing values from the total number of existing values divided by the whole number of values in ref(r).

  If reference relation isn't available, fact-completeness will equal null-completeness.

$$InC_{Null}(a_m\,(\mathrm{r})) = \frac{\text{Number of null values for } a_m\,(r)}{Cardinality\ of\ ref(r)} \quad (3)$$

$$C_{fact}(a_m\,(\mathrm{r})) = C_{Population}\ (a_m\,(\mathrm{r})) - InC_{Null}\ (a_m\ (\mathrm{r})) \quad (4)$$

- Data completeness scaled aggregate value (C_{Type}) for attributes required by user in query

$$\text{Scaled Total } (C_{Type}) = \frac{\sum_{m=1}^{M} C_{Type}(a_m(r))}{M} \quad (5)$$

  Where M represents the number of attributes that required by the user in the query

2.2 Data validity

Data validity is the extent to which attribute value within specified domain [5].

Validity assessment at attribute level concluded as follows (Where a_m (r) is attribute number m in relation r) [7]:

- Data validity assessment (l): It is the percentage between of valid values and the whole number of values in the universal relation.

$$l(a_m\ (\mathrm{r})) = \frac{\text{Number of valid values } a_m\,(\mathrm{r})}{\text{Total number of values in the universal relation}} (6)$$

- Data validity scaled aggregate value (L) for attributes required by user in query

$$\text{Scaled Total } (L) = \frac{\sum_{m=1}^{M} l(a_m(r))}{M} \quad (7)$$

Where M represents the number of attributes that required by the user in the query.

2.3 Data accuracy

Data accuracy is divided into two types: semantic accuracy and (0 or 1) accuracy. Semantic accuracy represents the gap between recorded value v and correct value v'. (0 or 1) accuracy represents the ratio between data values considered accurate (they don't conflict with real-world values) and the total number of values in the universal relation. (0 or 1) accuracy was the type used in our work [5].

Since a reference relation is almost always missing, costly and time consuming, we compare the value of each attribute to its domain of allowed values (it will consider as validity) but if reference relation is available and user not care about cost or time; accuracy will be calculated and can be improved by complaints and domain experts' feedback that identify erred data and a correction for them.

Accuracy assessment at attribute level will be concluded as follows (Where a_m (r) is attribute number m in relation r) [7]:

Data accuracy assessment (a): It is the percentage of accurate values and the whole number of values in the universal relation.

$$a(a_m\ (\mathrm{r})) = \frac{\text{Number of accurate values } a_m\ (\mathrm{r})}{\text{Total number of values in the universal relation}} (8)$$

- Data Accuracy scaled aggregate value (A) for attributes required by user in query

$$\text{Scaled Total } (A) = \frac{\sum_{m=1}^{M} a(a_m(r))}{M} \quad (9)$$

Where M represents the number of attributes that required by user in query

2.4 Data timeliness

Data timeliness is the extent to which data is up-to-date [5].In DIRA, we judge how far the data is modern using insertion time and volatility that we store in DIRA metadata structure.

Timeliness assessment at relation level is concluded as follows [7]:

Data timeliness assessment (t): Timeliness for relation r can be calculated using currency and volatility variables that will be presented in the following equations 10 and 11.

Currency = Age + (DeliveryTime – InputTime) (10)

Where:

Currency: It reflects how far the data is modern.[8]

Age: It reflects how old the data is when it is delivered.

DeliveryTime: Data delivery time to user.

InputTime: The data obtaining time.

$$\text{Timeliness (t (r))} = \max\left\{0,1 - \frac{Currency}{Volatility}\right\} \quad (11)$$

Where:

Volatility: The data validity lifetime.

In our work, we suppose that DeliveryTime = InputTime (no time between obtaining data and delivering it to the user) so Currency = Age

- Data timeliness aggregate value (T) for attributes required by user in query

$$Total(T) = \text{Maximum (t (r))} \quad (12)$$

3. DATA INTEGRATION TO RETURN RANKED ALTERNATIVES (DIRA) FRAMEWORK

DIRA framework consists of data quality assessment module and data quality system components.

3.1 DIRA data quality assessment module

There are many components to assess data quality in DIRA assessment module; these components are [9]:

- Assessment metrics. They are procedures for assessing data quality measures assessment scores using a scoring function.
- Aggregation metrics**.** They are procedures for assessing aggregated score. This aggregate score is calculated from individual assessment scores using aggregation functions like sum, count, and average functions.
- Data quality measures. They are metadata for describing how data is suitable for a specific task.
- Scoring functions. They are the way for assessing data quality measures. They may be simple comparisons, set function, aggregation function and complex statistical function.

3.2 DIRA quality system components

Quality system components consist of data quality acquisition and user input, these components are added to integration systems to enhance query results.

3.2.1 Data quality acquisition

Data quality acquisition stores data sources columns, tables and data quality assessment module results in the metadata store.

Hierarchical quality framework [10] that is introduced in DIRA is used in building DIRA metadata store entities presented in figure 1.

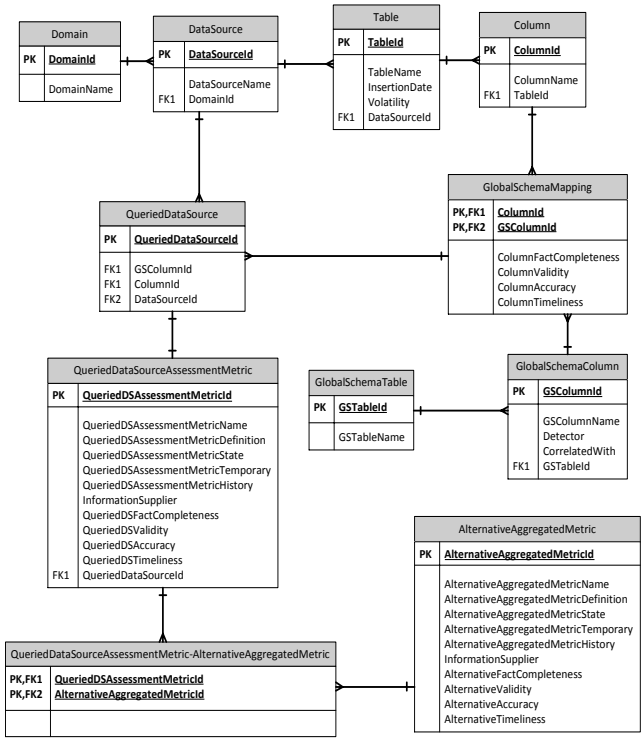

Figure 1. DIRA metadata structure [2]

3.2.2 User input

Qualified results can be returned to the user using some quality measures in a user query. SQL can include some quality measures as we present in the following query [1]:

```
Select A1… Ak
From G
Where < selection condition >
With < data quality goal >

Where A1, A2, Ai are global attributes of G
```

3.3 DIRA workflow components

DIRA consists of some basic components that can clarify how DIRA works. These components are

- Data quality assessment metrics of data sources attributes (columns). They are responsible for calculating the chosen data quality measures scores for all data sources attributes (columns) with its associated columns in the global schema. These scores are stored in global schema mapping entity.
- Data quality assessment metrics for queried data sources. Queried data sources are data sources that participate in query answering by attributes. In these assessment metrics, the aggregate data quality score for all attributes that the data source will participate with will be assessed for each measure.
- Alternatives formation. Alternative is one or more queried data source. There are two types of alternatives: qualified alternatives and not qualified alternatives. Alternative

considers qualified if it satisfies the specified quality level in the user query and not qualified otherwise.

In our framework, query type specifies the way of alternatives formation. Queries devoid of any quality constraint, combinations of all queried data sources are formed to build alternatives and all are qualified alternatives. Queries have one or more quality constraint, qualified alternatives of single queried data source that satisfies the specified quality level are built and other data sources are pruned from forming alternatives (First Pruning) then alternatives from more than one queried data sources are built from combinations of all queried data sources, alternatives that don't satisfy the quality constraint are pruned (Second Pruning) and the rest are qualified alternatives.

- Alternatives aggregated metrics. They are responsible for assessing aggregated scores of alternatives containing more than one queried data source.

  Alternative aggregated score is single representative value for alternative queried data sources assessment scores that assessed using aggregate function.

  There are different aggregate functions to represent single value for collection of values like average (mean), mode and median. In DIRA, we use simple average (arithmetic mean) aggregate function.

  Simple average is aggregate function that provides accurate description of entire data and uses every value in data so it considers good representative for data. Following in table 3 we present what the best aggregate function is with respect to the different types of variables.

Table 3. The best aggregate function with respect to the different types of variables[11][12]

Type of variable	The best aggregate function
Nominal	Mode
Ordinal	Median
Interval/Ratio	Mean or median

These aggregated scores are built from queried data sources assessment metrics according to the following equations [13].

- Alternative data fact completeness

$$C_{DSs} = \sum_{q=1}^{Q} C_{fact}(DS_q)/Q \quad (13)$$

- Alternative data validity

$$L_{DSs} = \sum_{q=1}^{Q} L(DS_q)/Q \quad (14)$$

- Alternative data accuracy

$$A_{DSs} = \sum_{q=1}^{Q} A(DS_q)/Q \quad (15)$$

- Alternative data timeliness

$$T_{DSs} = \text{Maximum}\big(T(DS_q)\big) \quad (16)$$

- Alternatives ranking according to proposed queries types. In this component, alternatives ranking based on different types of queries; No-measure top-k selection query, quantitative measure-feature top-K selection query, qualitative single-measure top-K selection query, quantitative multi-measure top-K selection query and qualitative multi-measure top-K selection query; These types vary in number of quality measures in query condition (from one to four) and the kind of quality measures value (quantitative or qualitative).

  In no-measure top-k selection query, alternatives returned to the user are ranked according to all scope measures and user chooses the most suitable ranking for him.

  In quantitative single-measure top-K selection query, alternatives returned to user are ranked according to specified data quality measure in user query, in qualitative single-measure top-K selection query, DIS receives message as input from user with quality measure score that represents the qualitative value for quality measure in user query and DIS returns alternatives ranked according to required data quality measure in user query.

  In quantitative multi-measure top-K selection query, alternatives returned to user are ranked through ranking algorithm called threshold algorithm [14] according to three case; Case1: user's query contains data quality measures, they are separated with (AND) and all are fulfilled, in this case, threshold algorithm returns alternatives ranked by total score. Case2: user's query contains data quality measures, they are separated with (AND) and the quality level for one or more of data quality measures isn't compatible with quality level for other data quality measures or isn't achieved, in this case, user receives a message to notify him that the quality level for query answering can't be achieved. Case3: user's query contains data quality measures, they are separated with (OR) and the quality level for all data measures fulfilled or the quality level for one or more of data quality measures isn't compatible with the quality level for other data quality measures or can't be achieved, in this case, threshold algorithm ranks alternatives with total score.

  In qualitative multi-measure top-K selection query, DIS returns alternatives ranked by threshold algorithm according to previous three cases but after receiving the message as input from the user with quality measures scores that represent the qualitative values for quality measures in user's query.

Following in figure 2, we present DIRA workflow components that we explained above

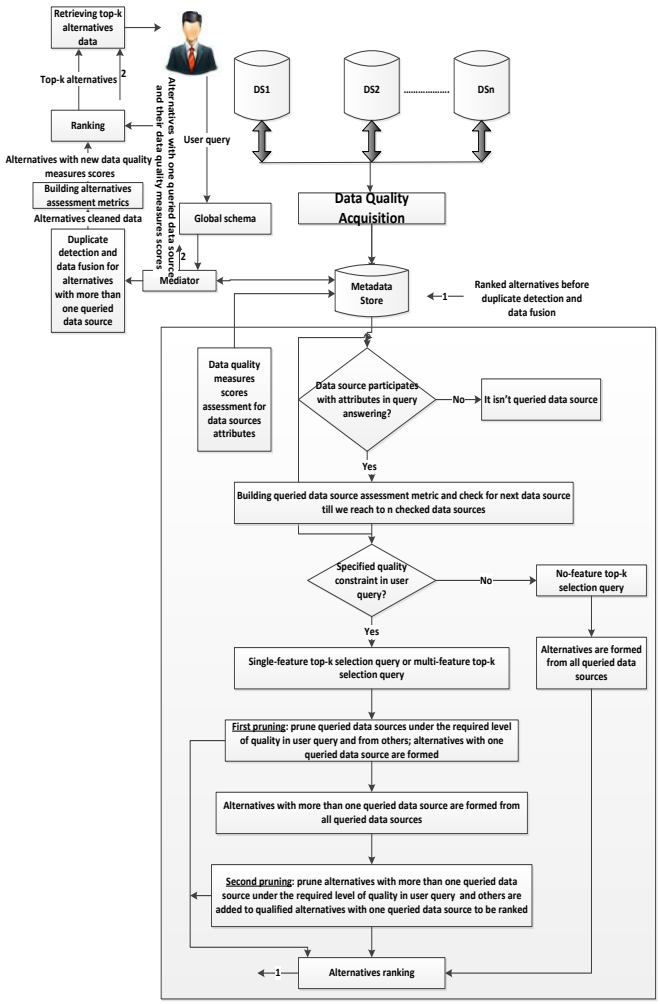

Figure 2. DIRA workflow components [2]

4. EXPERMENTS AND RESULTS

In this section, we clarify how to implement and validate our data integration framework and data quality system components. The experiments aim to calculate the response time and data sources number used to return query results. The following execution paths are the principles of our experiments:

- No-measure top-k selection query. This means, query doesn't contain quality constraints. Top-k alternatives are returned ranked by every scope data quality measure.
- Single-measure top-k selection query. This means, user specifies one data quality measure as a constraint, the data integration system (DIS) retrieves top-k alternatives ranked according to specified data quality measure.
- Multi-measure top-k selection query. This means, user specifies more than one data quality measure as a constraint, the data integration system (DIS) retrieves top-k alternatives ranked according to specified data quality measures together.

We execute the experiments on a laptop with an Intel(R) Core(TM) I5-5200U CPU @ 2.20GHz and 8 GB RAM. The laptop works with Windows 8.1 Enterprise edition. Microsoft SQL Server 2014 and Microsoft Visual Studio Express 2012 C# are tools that we use in our experiments.

4.1 TPC-DI benchmark

Transaction processing performance council (TPC) is an organization that defines benchmarks related to transaction processing and database. TPC benchmarks are TPC-C, TPC-DI, TPC-DS, TPC-E, TPC-H, TPC-VMS, TPCx-HS and TPCx-V, evaluating computer systems performance is their goal [15].

Our scope benchmark is TPC-DI; TPC released the first version of its data integration benchmark in January 2014.TPC-DI uses some tools to estimate the performance of data integration systems. Figure 3 illustrates a conceptual view of TPC-DI benchmark.

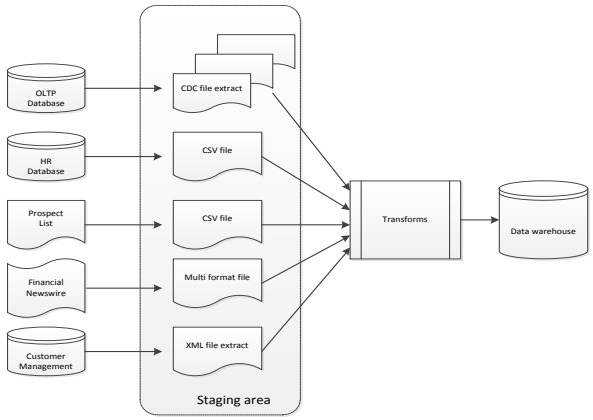

Figure 3. A conceptual view of TPC-DI benchmark [16]

The benchmark defines:

- Many schemas for data sources and file formats.
- The way to generate source data and how to store them.
- The schema for destination source (data warehouse).
- The way to transform and move data from data sources to the data warehouse.

In our experiment, we use only TPC-DI data sources to test our framework but with some modifications in data as TPC-DI benchmark depends on 100% accurate data sources but in our framework we depend on data sources with different levels of data quality. Data modification was done by adding errors in data; we replaced some values with nulls and others with not valid and not accurate values. We will consider the original TPC-DI data as reference data.

TPC-DI data sources that we use in our experiments are an online transaction processing database (OLTP DB) a human resource system (HR) and a customer relationship management system (CRM).

TPC data sources files are created using data integration generator called (DIGen) [16]. DIGen uses parallel data generation framework (PDGF) in data generation. PDGF is a common data generation framework that provides a set of data generation capabilities to generate specific TPC-DI data with specific prosperities [16].

We downloaded TPC-DI tool that contains DIGen file and PDGF folder from TPC website [15]. We downloaded Java SE 8 as Java

Virtual Machine (JVM) with a minimum of Java SE 7 must be used with DIGen to create the source data.

We used some commands to generate source data by DIGen like "java –jar DIGen.jar". The generated source data was in some batches in the form of file.txt, file.xml and file.csv. DIGen also generate statistics file named "digen_report.txt". The statistics file has some information about the way to generate data and number of rows in each batch. The schemas created in a SQL server database called "TPC-DI" and loaded the data into it.

As we mentioned above in section 3.2.1, the process of storing data sources attributes and relations in the metadata store is the responsibility of data quality acquisition component (N/P: In our experiment we will work in relation with population – completeness = 1). It executes also data quality queries on the data sources and stores their results in the metadata store. So, metadata store described in figure 1 was created to include eleven tables in the same database "TPC-DI". We also build a mapping tool to match the global schema columns with the local schema columns.

Table 4 presents the used global schema tables and global schema columns:

Table 4. Global schema tables and global schema columns that will be used in our experiment

Global Schema Tables	Global Schema Columns			
Customer	CustomerId	CLastName	CFirstName	CGender
	CAddressLine	CCity	CState	CPhone
	CCountry	CAge	c_m_name	c_maritalstatus
	c_postalcode	c_income	c_networth	c_numbercards
Account	CA_ID	CA_NAME	CA_STATE	

We use stored procedures to implement data quality queries that run by the data quality acquisition component. Those stored procedures have the equations used to assess the completeness, validity, accuracy and timeliness of the data sources attributes, execute according to schedule job created by SQL server and re-execute as soon as data sources update.

User input is the second component of data quality system components. User input lets the user specifies as optional the quality constraints. He can select between completeness or validity or accuracy or timeliness or any combinations. The user also can specify the quality levels for his chosen quality constraints, in addition to the number of alternatives that he wants to retrieve.

Following tables and response time of data sources used in our experiment are presented in table 5

Table 5. Tables and response time of data sources

Data Sources	Tables	Response Time
OLTP (DS1)	Customer and Account	500 sec
HR (DS2)	Employee	500 sec
CRM (DS3)	Customer and Account	500 sec

Following queried data sources assessment metrics are presented in table 6 (N/P: Required attributes in user query are CFirstName, CLastName, CAddressLine, CPhone and Cage and we assume that user chooses top-1 alternative from the list of ranked alternatives).

Table 6. Queried data sources assessment metric

Queried Data Sources	Retrieved Attributes	Aggregate Completeness for Retrieved Attributes	Aggregate Validity for Retrieved Attributes	Aggregate Accuracy for Retrieved Attributes	Aggregate Timeliness for Retrieved Attributes
DS1	CFirstName CLastName CAddressLine CPhone CAge	0.998	0.982	0.975	0.773
DS2	CFirstName CLastName CAddressLine CPhone CAge	0.987	0.964	0.956	0.628
DS3	CFirstName CLastName CAddressLine CPhone CAge	0.987	0.956	0.948	0.299

Table 7 presents simple form for alternatives aggregated metrics[2] according to scope data quality measures and queried data sources presented in table 6.

Table 7. Alternatives aggregated metrics

Alternative Name	Alternative Queried Data Sources	Alternative Completeness	Alternative Validity	Alternative Accuracy	Alternative Timeliness
Alternative1	DS1	0.998	0.982	0.975	0.773
Alternative2	DS2	0.987	0.964	0.956	0.628
Alternative3	DS3	0.987	0.956	0.948	0.299
Alternative4	DS1,DS2	0.992	0.973	0.966	0.773
Alternative5	DS1,DS3	0.992	0.969	0.962	0.773
Alternative6	DS2,DS3	0.987	0.960	0.952	0.628
Alternative7	DS1,DS2, DS3	0.990	0.967	0.960	0.773

4.1.1 Number of ranked alternatives and the number of accessed data sources in user selected alternative (Example 1)

In example 1, we choose completeness and timeliness from scope data quality measures as quality constraints and table 8 presents the number of returned ranked alternatives according to our framework and the number of accessed data sources in user selected alternative that used to execute query according to different execution paths.

Table 8. Number of ranked alternatives and the number of accessed data sources in user selected alternative

Execution Paths	Retrieved Attributes	Quality Constraint (Optional)	Number of Accessed Data Sources	Number Of Ranked Alternatives
No-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	-	3	7
Single-measure top-k selection query	CFirstName CLastName CAddressLine CPhone Cage	Completeness > 0.992	1	1

Multi-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	Completeness > 0.992 and Timeliness > 0.7	1	1

The results in table 8 and figure 4 show that if no determined quality measures; whole data sources need to be queried by DIS and return all combinations of data sources as alternatives. While adding quality measures reduce the number of accessed data sources to 1 instead of 3 and the number of returned alternatives that satisfy user requirements to 1 instead of 7.

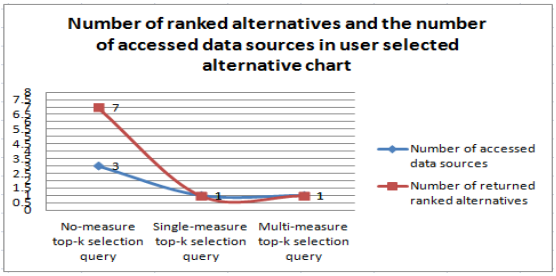

Figure 4. Number of ranked alternatives and the number of accessed data sources in user selected alternative chart

4.1.2 Response time (Example 1)

It is the time between the mediator query submission and receiving complete query answers from data sources.

In our work, we use calibration techniques[17],[18] to measure response time. The standard unit to measure the time interval is seconds. We assume that all data sources have the same capabilities for answering queries, network traffic, the servers' workload, database management system and hardware.

Table 9 shows the response time of our framework according to example 1 in different execution paths.

Table 9. Response time

Execution Paths	Retrieved Attributes	Quality Constraint (Optional)	Response Time (sec)
No-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	-	3.225 sec
Single-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	Completeness > 0.992	2.595 sec
Multi-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	Completeness > 0.992 and Timeliness > 0.7	2.595 sec

Table 9 and figure 5 shows that response time reduced by adding data quality measures.

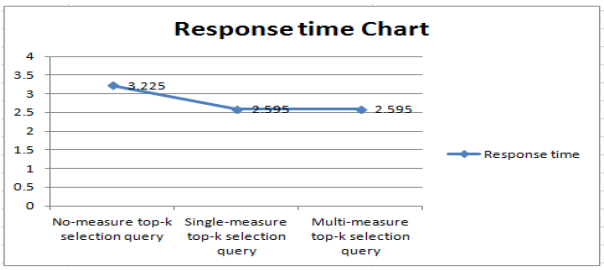

Figure 5. Response time chart

4.1.3 Number of ranked alternatives and the number of accessed data sources in user selected alternative (Example 2)

In example 2, user query become more complex by using completeness, validity, accuracy and timeliness together as quality constraints, the required quality level for them is satisfied by more than one alternative and one of satisfied alternatives consists of more than one queried data source that they integrate to achieve the required quality levels.

Table 10 presents the number of returned ranked alternatives according to our framework and the number of accessed data sources in user selected alternative that used to execute query according to different execution paths.

Table 10. Number of ranked alternatives and the number of accessed data sources in user selected alternative

Execution Paths	Retrieved Attributes	Quality Constraint (Optional)	Number of Accessed Data Sources	Number Of Ranked Alternatives
No-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	-	3	7
Single-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	Completeness > 0.990	1	3
Multi-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	Completeness > 0.990 and validity > 0.960 and accuracy > 0.962 and Timeliness > 0.7	1	2

The results in table 10 and figure 6 show that if no determined quality measures; whole data sources need to be queried by DIS and return all combinations of data sources as alternatives. While adding quality measures reduce the number of accessed data sources to 1 instead of 3 (the number of accessed data source may not reduce in another queries and it is a worth case) and the number of returned alternatives that satisfy user requirements to 2 or 3 instead of 7.

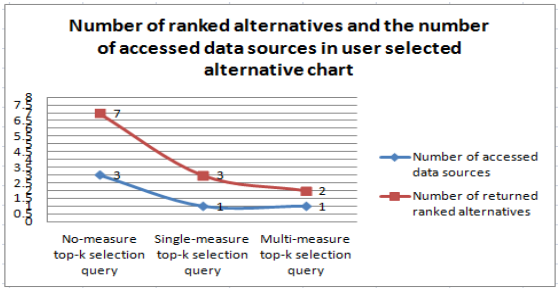

Figure 6. Number of ranked alternatives and the number of accessed data sources in user selected alternative chart

4.1.4 Response time (Example 2)

Table 11 shows the response time of our framework according to example 2 in different execution paths.

Table 11. Response time

Execution Paths	Retrieved Attributes	Quality Constraint (Optional)	Response Time (sec)
No-measure top-k selection query	CFirstName CLastName CAddressLine CPhone	-	3.225 sec

	CAge		
Single-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	Completeness > 0.990	2.595 sec
Multi-measure top-k selection query	CFirstName CLastName CAddressLine CPhone CAge	Completeness > 0.990 and validity > 0.960 and accuracy > 0.962 and Timeliness > 0.7	2.595 sec

Table 11 and figure 7 shows that response time reduced by adding data quality measures.

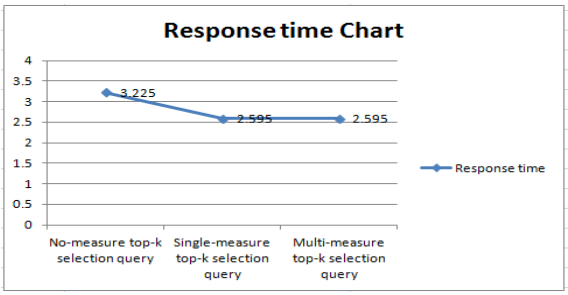

Figure 7. Response time chart

5. CONCULSION AND FUTURE WORK

Query results obtained from data integration system have some problems; they are all returned to the user from all queried data sources without any specified quality level and hence they are not ranked and they take a long time.

In this paper, we present data integration framework called DIRA, this framework improves query results obtained from DIS and generates them in reasonable time by adding quality system components and data quality assessment module to any DIS to retrieve results from only convenient data sources and return these results ranked according to their quality in both cases if quality measures are specified in user query or not.

Our experiments illustrate that our framework can retrieve results with number of data sources less than the original DIS, hence less number of ranked alternatives in a reasonable time.

We can extend our work to include different types of databases like semi-structured and unstructured data sources, use additional data quality measures, use different ranking algorithms, use both as view (BaV) as data integration system mapping technique and use integrity constraints in global schema to make query answers more consistent.

6. REFERENCES

[1] M. S. Abdel-moneim and A. H. El-bastawissy, "Data Quality Based Data Integration Approach," World of Computer Science and Information Technology Journal (WCSIT), vol. 5, no. 10, pp. 155–164, 2015.

[2] Reham I. Abdel Monem, A. H. El-bastawissy, and M. M. Elwakil, "DIRA : A Framework Of Data Integration Using Data Quality," International Journal of Data Mining & Knowledge Management Process (IJDKP), vol. 6, no. 2, pp. 37–58, 2016.

[3] M. Poess, T. Rabl, B. Caufield, and I. Datastage, "TPC-DI : The First Industry Benchmark for Data Integration," the 40th International Conference on Very Large Data Bases, vol. 7, no. 13, pp. 1367–1378, 2014.

[4] F. Sidi, P. Hassany, S. Panahy, L. S. Affendey, M. A. Jabar, H. Ibrahim, and A. Mustapha, "Data Quality : A Survey of Data Quality Dimensions," IEEE, pp. 300–304, 2012.

[5] M. Kaiser, "A Conceptual Approach to Unify Completeness , Consistency , and Accuracy as Quality Dimensions of Data Values," European and Mediterranean Conference on Information Systems, vol. 2010, pp. 1–17, 2010.

[6] C. Moraes and A. C. Salgado, "Information Quality Measurement in Data Integration Schemas," ACM, 2007.

[7] C. Batini and M. Scannapieco, Data Quality concepts,Methodologies and techniques. 2006.

[8] W. Fan, F. Geerts, N. Tang, and W. Yu, "Inferring Data Currency and Consistency for Conflict Resolution," ICDE, 2013.

[9] P. N. Mendes, H. Mühleisen, and C. Bizer, "Sieve: Linked Data Quality Assessment and Fusion," Proceedings of the 2012 Joint EDBT/ICDT Workshops, pp. 116–123, 2012.

[10] I. N. R. Etrieval, "A Flexible Quality Framework For Use Within Information Retrieval," Proceedings of the Eighth International Conference on Information Quality (ICIQ-03), pp. 297–313.

[11] "Measures of central tendency." [Online]. Available: https://statistics.laerd.com/statistical-guides/measures-central-tendency-mean-mode-median.php.

[12] "Fundamentals of statics." [Online]. Available: http://www.usablestats.com/lessons/noir.

[13] P. Angeles and F. García-ugalde, "A Data Quality Practical Approach," International Journal on Advances in Software, vol. 2, no. 2, pp. 259–274, 2009.

[14] I. F. Ilyas, G. Beskales, and M. a. Soliman, "Query Processing Techniques in Relational Database Systems," ACM Computing Surveys, vol. 40, no. 4, pp. 1–58, 2008.

[15] "TPC." [Online]. Available: http://www.tpc.org/information/benchmarks.asp.

[16] S. Specification and A. R. Reserved, "TPC BENCHMARK ™ DI Transaction Processing Performance Council (TPC)," no. November, 2014.

[17] R. M. G. and M. A. L. Jeff C.Gust, "Stopwatch and Timer Calibrations (2009 edition)," 2009.

[18] M. Spiliopoulou, I. Wirtschaftsinformatik, and H. Berlin, "A Calibration Mechanism Identifying the Optimization Technique of a Multidatabase Participant 3 Optimizer Calibration Methodology," Conference on Parallel and Distributed Computing Systems (PDCS), Dijon, France, 1996.