CHAPTER-2

ARCHITECTURE

2.0 Introduction

SIQS allows a user to frame queries in SIQL, the query language of SIQS, without explicitly specifying the structures to be used to answer these queries. The user is provided with a list of semantically meaningful names (SMNs) that may represent relations and/or attributes present in component databases registered with the global system. The user uses these SMNs to construct a query in SIQL. SIQS is responsible for arriving at the appropriate structures, at run time, which are used to answer the user query. SIQS generates multiple queries for a user query and each of these generated queries after execution constitutes a result. The user while viewing the results, one at a time, can save the results of interest to him/her. Once all the queries are generated, the saved results are shown together to the user. The user is allowed to choose the relevant results, for integration, from these saved results. The chosen results are then integrated whereafter the final integrated result is shown to the user.

Unlike the multidatabase language systems, discussed in Chapter-1, which provide access to multiple component databases, SIQS allows the user to specify his/her
intent by posing multiple interrelated queries in SIQL. The results from these multiple user queries are integrated and presented to the user.

The underlying architecture of SIQS can be described in terms of the System architecture. The system architecture describes the key processing elements used and the way they are related in SIQS.

2.1 System Architecture

The system architecture of SIQS comprises of the SIQS system and the component databases registered with the global system. The system architecture is shown in Figure-1.

2.1.1 Component Databases

In SIQS, the component databases that are registered with the global system are autonomous and therefore have the following capabilities

1. The component databases can associate and disassociate from the global system at any time.

2. Each of these component databases has the ability to decide the amount of data it wishes to share with the global system. The component databases may not share all their data, as some data may be private to them.

The relational model is used as the data model in each of these component databases.
Figure-1 - System Architecture of SIQS
The local DBA manages and controls the access to data in the component database present in his/her site. He/She is responsible for registering his/her component database with the global system. For this he/she maintains the mapping information (MI) about the relations and attributes that are associated with the global system, at the local component database site. The mapping information contains the semantically meaningful names that are used to represent the local relations and local attributes. This information is accessed only when relations and attributes in the corresponding component database is referred to in order to answer the user query. The mapping information being stored at the component database site simplifies the task of associating and disassociating the component databases from the global system.

2.1.2 SIQS System

SIQS generates the final integrated result based on the user query. For each SIQL query posed by the user, the SIQS is responsible for the following

- Identifying the component database schemas, the relations and the attributes that are to be used to answer the user query. This requires using the mapping information present at the component database site referred to by the SMNs in the user query.
- Generating multiple generated queries using the structures so identified to answer the user query.
• Executing these generated queries by accessing the data present in the component databases referred to in the generated query.

• Collecting and storing the relevant results, for the user, after executing these generated queries based on user choice.

• Integrating the results for presentation to the user.

As shown in Figure-1, the SIQS has the following processing elements that interact to perform the above mentioned tasks.

• Database Server

• Multidatabase Dictionary

• Graphical User Interface

• Query Generator

• Result Integrator

2.1.2.1 Database Server

SIQS has its own resident database server that is responsible for managing data required for its operations. The database server performs the following tasks.

• It stores the multidatabase dictionary, in the form of a table, which contains the semantic information along with the information about the component databases registered with the global system.
• It stores a temporary table, created by the Query Generator, which contains information about the schemas and the relations that are required to answer the user query. This table is used to generate multiple query resolutions.

• It executes the meta-queries, generated by the query generator for the temporary table stored.

• It stores the query resolutions, generated by executing the meta query, as tables. These query resolutions are used to generate non-redundant query resolutions in future.

• It executes the generated query by accessing one or more component databases referred to in the generated query.

• It stores the results of interest to the user, as tables, for integration based on the user query.

• It executes the integration query, generated by the Result Integrator, to integrate the results chosen by the user.

2.1.2.2 Multidatabase Dictionary

Multidatabase dictionary contains information about the data present in the component databases registered with the global system. This dictionary is maintained by the SIQS and is updated as and when the component databases associate and disassociate from the global system. The multidatabase dictionary is
stored as a table in the database server. The information present in the multidatabase dictionary is as follows

- It contains information about the component databases that are registered with the global system. This information is used by the Query Generator to generate multiple generated queries.

- It contains the semantic information about the relations and attributes that are associated with the global system. The semantic information for relations consist of the SMNs and their description whereas the semantic information for attributes consist of the SMNs, data types and their descriptions. These SMNs are used by the global system to represent the relations and attributes present in the component databases registered with it.

2.1.2.3 Graphical User Interface

SIQS provides the user with a Graphical User Interface in the form of a window. The GUI performs the following tasks

- It provides the user with a query interface to frame a query in SIQL. The interface assists the user by providing a list of semantically meaningful names, present in the multidatabase dictionary, to construct the user query. The user uses these SMNs to construct a query in SIQL. This query is then submitted to the system for execution.

- It displays the multiple results generated for a user query, one at a time, to the user. It allows the user to view the detailed information about the result, which
consists of the generated query and the pictorial representation showing the component database schemas and the relations that are used to answer the generated query. This information is presented to the user through the graphical interface. The user is given an option, through the interface, to save the relevant results for future reference. The GUI stores these saved results, as tables, in the database server.

• It fetches the saved results from the database server after all the results for the user query are generated. These saved results are then shown to the user, all together, through the GUI. The user uses the interface to choose results, from these saved results, for integration. The GUI sends the chosen results to the Result Integrator for Integration.

• It presents to the user the final integrated result, made available by the Result Integrator.

2.1.2.4 Query Generator

The Query Generator accepts the user query in SIQL as input and produces results after executing the result queries, as output. The Query Generator performs the following tasks

• It identifies the relations along with the corresponding component database schemas that are used to answer the user query. For this, it uses the
multidatabase dictionary and the mapping information of the component databases registered with the global system.

- It creates a temporary table in the database server to store the relation names along with their schema names, so identified, to answer the user query.

- It generates meta-queries on this temporary table, one at a time, based on the desired criterion for ordering of results. It then sends the meta query to the database server for execution.

- It creates a table in the database server to store the query resolutions arrived at after execution of a meta-query. This table is used thereafter to generate non-redundant query resolutions.

- It uses the mapping information present in the component databases referred to in the query resolution to transform the query resolution into a generated query.

- It sends the generated query to the database server for execution.

- It collects the generated results, after execution of the generated query, and sends it to the GUI in order to be shown to the user.

2.1.2.5 Result Integrator

The Result Integrator accepts the results chosen for integration, by the user, as input and produces the final integrated result as the output. It performs the following tasks

- It orders the results, for integration, based on a defined result ordering criterion.
• It generates a sequence of integration queries in SQL, based on the defined order.

• It sends these integration queries to the database server for execution. Each of these integration queries integrates a pair of results.

• It collects the final integrated result arrived at after integrating all the chosen results and sends them to the GUI in order to be shown to the user.

2.2 Query Processing

The query processing in SIQS is performed in the following way:

The user is presented with a Graphical User Interface to pose a query in SIQL. The GUI assists the user in constructing the query by providing him/her with a list of semantically meaningful names. These names, assigned by the SIQS, represent the relations and the attributes present in the component databases registered with the global system. The user constructs the query in SIQL and submits it to the system for execution. SIQS accepts the query from the user and sends it to the Query Generator. Based on the user query, the Query Generator uses the multidatabase dictionary and the mapping information present at the participating component database sites to create a temporary table in the Database Server that stores all the relation names with their schema names that can be use to answer the user query. Meta-queries are generated, one at a time, on this table, by the query generator, in a defined order. The meta-query is executed on this table to arrive at a query
A query resolution contains one or more relation names, along with their schema names, that can be used to answer the user query. Query Generator uses the multidatabase dictionary to transform the query resolution into the corresponding generated query. The generated query is executed against the component databases registered with the global system to give a result for the user query. The result is then sent to the GUI in order to be shown to the user.

The GUI presents the result to the user. The GUI also gives the user the option to see the details of the result. The details include a pictorial representation, showing the schema names of the component databases and the relation names that participate to fetch the result. The details also show the generated query of the corresponding result. The user is given an option, in the GUI, to save the result. The GUI stores the result saved by the user, as tables, in the database server. The user can proceed to the next result, if he/she wishes to, using the GUI. If the user chooses to proceed to the next result, the Query Generator generates a result based on a new query resolution generated by the meta-query. This process continues till all the queries to be generated are exhausted.

Once all the queries are generated and executed, the GUI fetches all the saved results from the database server and shows them to the user for integration. Since each of these saved results is stored as a table in the database server, the table names are also shown along with them. The user is given a choice to choose the results, for integration, from these saved results. He/She may choose some or all of the results for integration. The chosen results are sent to the Result Integrator for integration. The Result Integrator orders the chosen results based on a defined result ordering.
criterion. A sequence of integration queries in SQL is generated based on this defined order. Each of these integration queries, on execution, integrates a pair of the chosen results. Once all the chosen results are integrated, the final integrated result is sent to the GUI in order to be shown to the user.

In SIQS, the user is allowed to pose multiple interrelated queries in addition to a single query. The results of the individual queries are first integrated. The integrated results from each of these individual queries are then integrated to give the final result.