Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

Methods to Implement Homogeneous and Heterogeneous Distributed Database

[1]Abbas Atwan Mhawes, [2]Mustafa Ali Hassan

Ministry of Education 10065 A1 Tarbawi,

Collection, Baghdad, 10001, Iraq

Abstract: **Objective of this research is how to implement distributed databases in a distributed environment with both types homogeneous and heterogeneous, where for each of these types there are special methods to deal with it, will be discussing these methods that deal with both types of types of distributed databases, with explain the strategies for each method in detail in this search.**

Keywords: **Homogeneos; Heterogeneous; Distributed Database.**

1. INTRODUCTION

Design in distributed computer systems, including decision-making in the put of data and programs across sites computer network, as well as the network itself may design. In the case of (DDBMS) , the distribution of applications includes two things: the distribution of database management systems (DBMS), and the distribution of application programs They are run on it , The organization of distributed systems can be investigated him through three dimensions [3] :

1. Level of sharing
2. Behavior of access patterns
3. Level of knowledge on access pattern behavior

The first dimension (LEVEL OF SHARING). There are possibilities you may does not have to share all application, and data implement on one site, and there is no contact with any other program or access to any data file at the sites Other Another possibility is having sharing of data and programs, both data and programs may be sharing, this means existing program, on the site can request service from another program exists on the second site, which in turn may have access to the data in the file exist on the third The second pattern is a (BEHAVIOR OF ACCESS PATTERNS) it is possible to identify two patterns[1], for patterns of user access requests, it may be static which does not change with the passage of time, or may be dynamic, which are changing over time It is the third dimension is (LEVEL OF KNOWLEDGE ON ACCESS PATTERN BEHAVIOR) where designers in a distributed environment they may have complete information about the expectations of patterns, access or have partial information about it

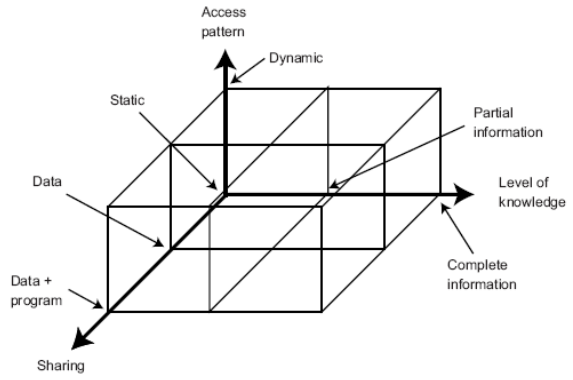

Fig. 3.1 Framework of Distribution

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

In a distributed environment we are dealing with two types of databases, they are heterogeneous databases and Homogeneous databases ,two type of methods have been identified for the design of distributed databases,(top-down method) which is more appropriate to deal with Homogeneous databases, while (bottom-up method) is more appropriate to deal with multiple heterogeneous databases[7-12] .

2. TYPE OF DISTRIBUTED DATABASE

In this section will be identify the types of distributed databases, which is necessary to identify these types, because , our research deals with two types using two methods each method, especially with each of these types, and each method has strategy deal to implement, and design of this type of distributed databases optimally, these types are Homogeneous databases, and multiple heterogeneous database .

2.1. Homogeneous Databases

the databases that distributed on more than server, and database management systems (DBMS) in all distributed sites are the same this mean have the same type of database, in terms of structure as well as database management systems[11-12] .

2.2. Heterogeneous Database

the databases that distributed on more than server ,where in some sites there is a type of database with certain management systems (DBMS) and certain structure, but in other sites there are other types of databases in The another words these databases are diverse and different . And have database management systems different (DBMS) distributed over the sites [8-9] .

3. METHODS FOR IMPLEMENTATION OF DISTRIBUTED DATABASE

we have two types to implement distributed database these types :

3.1 TOP-DOWN-METHOD

General framework for (TOP-DOWN-METHOD) is shown in Figure (3-2) beginning with the Following activities, (REQUIREMENT ANALYSIS) at this stage is identify the system environment as well as the needs of data and processing to all potential database users, either in the (SYSTEM REQUIREMENT OBJECTIVES) Relating to the determining objectives of distributed database systems and these goals represent, performance, reliability, availability, economy, expandability (flexibility). [1] Where considered these requirements is the (INPUT) to two parallel activities and are the (VIEW DESIGN) and (CONCEPTUAL DESIGN) The activity, (VIEW DESIGN) deals with determining the interfaces to the end user, either (CONCEPTUAL DESIGN) In This activity, it is treated, in order to understand and explain the structure of the database through the use of diagrams (ER MODEL) and how to determine the entities and attributes for these entities, also the relations between those entities THE (GLOBAL CONCEPTUAL SCHEMA (GCS)) and (ACCESS PATTERN INFORMATION) is collected as a result of (VIEW DESIGN) as inputs to (DISTRIBUTION DESIGN), where at this stage, which represents the goal of the method (top down method), is designed (LOCA CONCEPTUAL SCHEMAS) (LCS) by the distribution on the sites distributed system To prosessing each entity as a unit in the distribution , (DISTRIBUTION DESIGN) consists of two steps They (FRAGMENTATION AND ALLOCATION) where in a separating (DISTRIBUTION DESIGN) to two steps to deal better with the complexity of the problem, the last step in the design process is (PHYSICAL DESIGN), which translates the expected schemes, to a real database structures (actual In this stage the conceptual design conversion into a real actual database structure, also its known that the design and development of activities in any type, is a continuous, therefore it requires continuous monitoring and adjusting periodically and adjust, so (OBSERVATION AND MONITORING) is considered a key activity in this process [2].

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

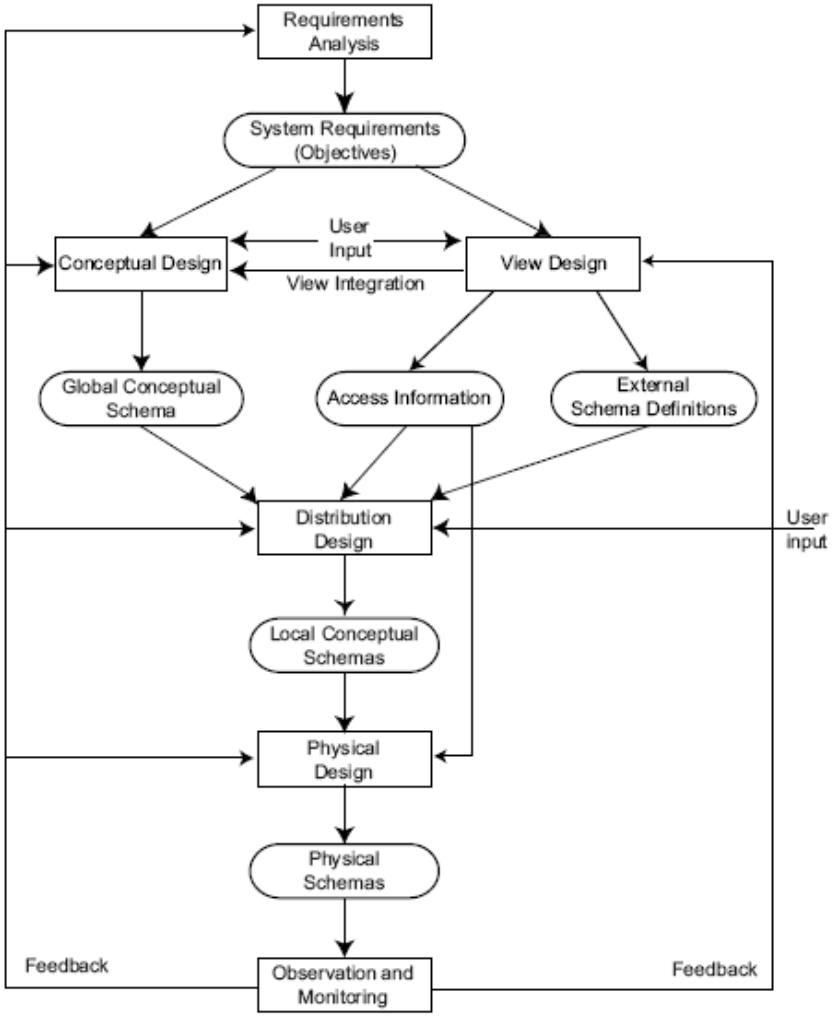

Fig. 3.2 Top-Down Design Process

3.1.1 Distribution Design

In (Distribution Design) are designed (local conceptual schema) by distribution entities on the sites in the distributed system, and this is done by two main factors which are (fragmentation and allocation) [5-6]

3.1.1.1 Fragmentation

It is a form of transparency in distributed databases, which is a division of the database to the smaller fragmentation where stores each fragment at the site different, and treated as an object separate database, and is usually done for performance reasons, availability, reliability, and transparency, etc,[5] in this section we will talk about some of the important issues in the fragmentation, in addition to existence of some of the different strategies, to implement fragmentation They (Horizontal fragmentation) and (Vertical fragmentation), and moreover, interference may occur, in these two types, so what is known (hybrid fragmentation)

3.1.1.1.1 Reasons for Fragmentation

Data can be stored in different computers, by fragmenting, the entire database into several pieces called fragments, each fragment stored in the different site, fragments are units of logical data stored in different sites in a distributed database system, before we discuss fragmentation in detail there are reasons for fragmentation the relation they are :

The first, usage in general is that the applications work with a part of relation rather than the entire relation, so the distribution of the data, it appears appropriate to work with subsets of the relation as a unit in the distribution

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

The second, efficiency where the data is stored close to where they are frequently used in addition to, are not taken to store data, which are not needed, the local applications

Third, parallel with fragments as the unit distribution, the transaction can be divided into several sub-queries that run on fragments, this will increase the degree of synchronization, or parallelism in the system and therefore allows for transactions be implemented in parallel [6].

3.1.1.1.2 Correctness Rule of Fragmentation

Here we will impose three rules through fragmentation and that working side by side, to ensure that the database is not getting them any semantic change through fragmentation:

1. Completeness : If the relation (R) decomposed to fragments where FR = {R1, R2, ...RN} Each data item which can be found in the relation (R) must show in one or more of the fragmentation, this is important in the fragmentation because it ensures that the data in the global relation without any loss

2. Reconstruction : If the relation (R) decomposed to fragmentation FR = {R1, R2, ...RN} must be possible to identify a relational process that would restore Reconstruction (R) of fragments where the Reconstruction for (horizontal fragmentation) is the process of (union) and the process (join) is for (vertical fragmentation)

3. Disjointness : If the relation (R) is the decomposed horizontally fragmentation FR = {R1, R2, ... RN} and the data item (DI) appeared in the fragments (RI) this data item must does not appear in any fragments Other, this It ensures that Horizontal fragmentation is Disjoint, as well as if the relation (R) decomposed vertically fragmentation the primary key is usually repeated in every fragments for Reconstruction.

3.1.1.1.3 Information requirements of fragmentation

In this stage when we want to make fragmentation we need to understand the structure of the database correctly to avoid loss of data as well as to ensure the return of Reconstruction fragments to the normal state, in the database that was before the fragments work, so it requires in this aspect that note how the relations in the database are associated with each other. in the relational model described these relations where be linked between the tables, and in another form, such as (ER MODEL) these relationships are described between the database objects which are known entities, where these entities are associated with each other through Relationships So understand this information and the manner of connect between tables is to help in the distribution design [5-6] .

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Fig. 3.3 Modified Example Database

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

we note of the figure (3.3) that the relationship between the table (PAY) and table (EMP) is (one to many) where each (title) there are several employees while the relation between the table (emp) and table (PROJ) are represented (many to many), because each employee may be assigned to him several projects as well as for each project may be there are several employees . So there should be another table which is (ASG), which has in turn two Forign Key these Keys (PNO) in a table (PROJ), and the column (ENO), which represents Primary Key in the table (EMP) the purpose of understanding of this information in the database structure to helps in distribution design and specifically at the make fragmentation .

3.1.1.1.4 Type of Fragmentation

In this section will be the focus of the types of fragments to the database where there are several types based on the need for access to data by the user, and these types are: [5]

1. Horizontal Fragmentation : Consists of a subset of the (records), where in this type of fragmentation is divided the relation depending on the rows, where each fragments, will have a subset of rows of the relation, where its determines by using (where clause) in statement of (sql), there are versions of Horizontal Fragmentation they are : (primary) and (derived) In the (primary), in this type of Horizontal Fragmentation the relation will be division based on several records that represents as conditions in statement of (sql), where these records are exist in the same original table, and will be Explanation for this type of fragmentation by using this version of the relational database, which contains four tables as shown in Figure (3.3)

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Fig. 3.3 Modified Example Database

From the Figure (3.3) will be the division of the relation (PROJ) into several fragments (PROJ1 , PROJ2 , PROJ3) depending on the field (loc) in a table (PROJ) This is done as follows :

PROJ1 = Select * from PROJ where LOC = "Montreal"

PROJ2 = Select * from PROJ where LOC = "new yorkl"

PROJ3 = Select * from PROJ where LOC = "paris"

The rezult of fragmentation is :

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

$PROJ_1$

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal

$PROJ_2$

PNO	PNAME	BUDGET	LOC
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York

$PROJ_3$

PNO	PNAME	BUDGET	LOC
P4	Maintenance	310000	Paris

Fig. 3.8 Primary Horizontal Fragmentation of Relation PROJ

The second type of Horizontal Fragmentation is (derived) in this type we will make the fragments for the relation depending on several records that represents as conditions in clause (sql), but these records exist in another table, but after connecting tables by relation will be make fragments , will be explain by using the same structure of database that we have referred to in the previous example but this time want to make fragments to the table (EMP) depending on the field (SAL) in a table (PAY) to (EMP1 , EMP2) this is done as follows :

EMP1 = select * from EMP
Where Emp.title = PAY.title AND sal <= 30000

EMP2 = select * from EMP
Where Emp.title = PAY.title AND sal > 30000

The result of fragmentation is :

EMP_1

ENO	ENAME	TITLE
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E7	R. Davis	Mech. Eng.

EMP_2

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E8	J. Jones	Syst. Anal.

Fig. 3.11 Derived Horizontal Fragmentation of Relation EMP

2. Vertical Fragmentation : This type of Fragmentation consists of a subset of columns,where in this type of fragmentation we will make fragments for the relation depending on the columns where each fragments will have a subset of the columns, where it determines by using (select clause) in statement of (sql), and to explain that we will use the same as of the previous structure of the database for Explanation Vertical Fragmentation, here we will make fragments to the table (PROJ) and dividing it into (PROJ1 , PROJ2) as the follows:

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

PROJ1 : select PNO , BUDGET FROM PROJ
PROJ2 : select PNO , PNAME , LOC FROM PROJ

The result of fragmentation is :

$PROJ_1$

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000

$PROJ_2$

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Develop.	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris

Fig. 3.5 Example of Vertical Partitioning

3. Hybrid Fragmentation : In most cases of Horizontal Fragmentation or Vertical Fragmentation in the database schema, will not be a sufficient to satisfy user's application requirements in this case Vertical Fragmentation may be follow the Horizontal Fragmentation or vice versa (Figure 3.19) shows, these two types of strategies of fragmentation vertical and horizontal that apply one after the other, and make another type of fragmentations called this type of fragmentation (Hybrid Fragmentation), also called (mixed fragmentation)

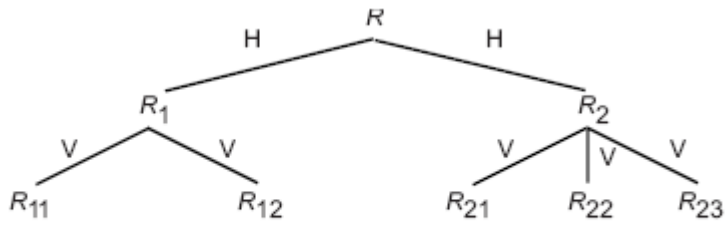

Fig. 3.19 Hybrid Fragmentation

3.1.1.2 Allocation

Here and after the completion of the work fragmentation of the database became necessary, how is the distribution of these fragments across the sites in a distributed environment, this section is interested in allocating a place for these resources across the network, to ensure finding the optimum distribution of the distribution fragments across the sites,we first need to identify the problem of allocation in more accurately, and to explain this, we suppose that ,there is a group of fragments They are F = {f1, f2, ... fn} and distribution system consists of sites S = {s1, s2, ... sm} and there is applications runing on these sites these applications Q = {q1, q2, ... qa} problem of allocation involves finding the optimal distribution in the distribution of (F) on the (S), where it can definition of optimal and that relates with two Factors are the :[1]

1. Minimal Cost: Function cost consists of the cost of storage for each fragments on the site, cost of queries at each site, and cost of the update for fragments at all sites where they are stored, in addition to the cost of data communications, problem of allocation trying to find (allocation scheme) which It reduces the costs for issues that mentioned above.

2. Performance : Strategy of allocation is designed to maintain of performance at work, two things are known, which reduce the response time, as well as maximize the productivity of the system at each site . Most of the models that have been proposed until now to make this distinction from ideal and therefore when one is really looking for the problem in

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

depth, it is clear, it must include the ideal that are measured by both, performance and cost factors In other words, one should be looking for scheme allocation, Meaning the answer to the requests of the user in the shortest possible time, as well as to maintain the cost of trnsactions at a minimum.

3.2 Bottom – up Method

This method is appropriate in cases of Systems Multi database, which are most likely Heterogeneous database, where in this case the number of databases are already exists and tasks of the design include the integration of these databases, where in simply the process consists of integration (local database) with (local schemas) (LCS) to (global database) with (global schemas) (GCS), in other words, the conversion from (LCS) to (GCS) for the formation of multiple database coherent , (Bottom - up Method) occur in two general steps Figure (4.3) shows those steps , The first step is called (schema translation) or (simply translation) and the second is called (schema generation)

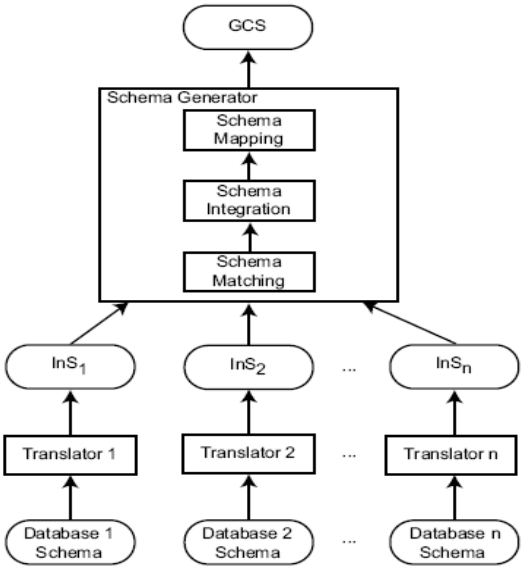

Fig. 4.3 Database Integration Process

3.2.1 Schema Translation

In this step the components of databases are translated to the common intermediate canonical representation,canonical representation is a design pattern used to communicate between different data format, where the use of (canonical representation) facilitates the translation process by reducing the number of translations that need to be written, the selection of (canonical model) is important,in principle, should be one of the expressions is Sufficient to incorporate the concepts available in every databases that will be later integrated where there are alternatives used for this purpose include (entity relation ship model), or (object oriented model) or simply be (graph tree model), where he became the very model common, which clearly above,the step of translation is only necessary if the database components It is heterogeneous and (local schemas) identified by using different data model [2].

3.2.2 Schema Generation

In this step from (Bottom up Method) the (intermediate schemas) are used to generate (GCS) process Schema Generation consists of the following steps:

1. Schema matching to determine the syntactic and semantic correspondences among the translated LCS elements or between individual LCS elements and the pre-defined GCS elements

2. Integration of the common schema elements into a global conceptual (medi- ated)schema if one has not yet been defined

3. Schema mapping that determines how to map the elements of each LCS to the other elements of the GCS .

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 7, Issue 2, pp: (8-19), Month: May - August 2020, Available at: www.noveltyjournals.com

3.2.2.1 Schema Matching

In this step is determined, (syntactic) or (semantic) matched between translated (LCS) or between schemes different (LCS) that have been identified

3.2.2.2 Schema Integration

After the completion of stage of (Schema Matching) to find correspondence between schemes (LCS) different, the next step is to create (GCS) and this step refers to the (Schema Integration) In this stage is generated (GCS) as a result from integration schemes (LCS) different depending on the matches that have been identified through stage of (Schema Matching) [2], and to implementation of our discussion at this stage we provide a simple example to explain how to generate (GCS) based on the matches that resulted from schemes (LCS) to explain the example will be to deal with the two schemes (LCS), which was used in the beginning of the search and are as follows :

EMP(ENO, ENAME, TITLE)
PROJ(PNO, PNAME, BUDGET, LOC, CNAME)
ASG(ENO, PNO, RESP, DUR)
PAY(TITLE, SAL)

Fig. 4.4 Relational Engineering Database Representatio **LCS 1**

WORKER(WNUMBER, NAME, TITLE, SALARY)
PROJECT(PNUMBER, PNAME, BUDGET)
CLIENT(CNAME, ADDRESS)
WORKS_IN(WNUMBER, PNUMBER, RESPONSIBILITY, DURATION)
CONTRACTED_BY(PNUMBER, CNAME, CONTRACTNO)

LCS 2

Fig. 4.6 Relational Mapping of E-R Schema

Employee(ENUMBER, ENAME, TITLE)
Pay(TITLE, SALARY)
Project(PNUMBER, PNAME, BIDGET, LOCATION)
Client(CNAME, ADDRESS, CONTRACTNO, PNUMBER)
Works(ENUMBER, PNUMBER, RESP, DURATION)

Fig. 4.9 Example Integrated GCS **GCS = LCS 1 + LCS 2**

Integration methodologies that can be divided into (binary or nary mechanisms) depending on the pattern loca schemas (LCS), (Figure 4.10) shows Integration methodologies, the (binary Integration methodologies), includes dealing with two schemas at a time,and this can happen in two modes, the first is the (stepwise (ladder) fashion) or (pure binary fashion) (Figure 4.11) shows this In the first mode (stepwise) where creation the intermediate schemas to integrate with subsequent

schemas, either in the (pure binary) where each schema is integrated with the other one, is created intermediate schemas to integrate with the intermediate schemas Other.

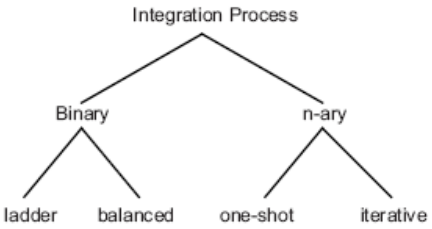

Fig. 4.10 Taxonomy of Integration Methodologies

(Naray integration mechanisms) here integration is for more than two schemas at each iteration and also occur in two modes, the first (one pass integration) This mode occurs when each schemas integrated once and produce the global conceptual schemas (GCS) after each repetition, the benefits of this approach includes the full availability of information about all the databases at the time of integration, there is no priority to arrange integration schemes, and handle like the best representation of the data elements, in addition to all these benefits there are difficulties with this approach, which includes, increase adherence, and the difficulty automation

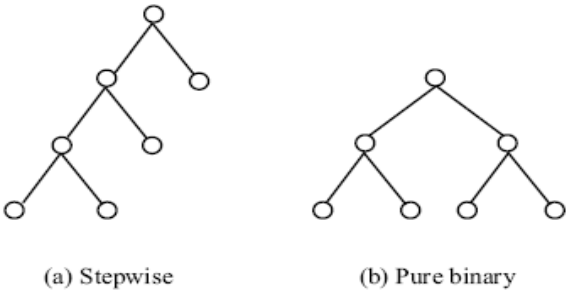

Fig. 4.11 Binary Integration Methods

The second mode which is (iterative nary integration) (Figure 4.12) shows this approach, this mode provide more flexible, typically more information is provided, which is more general, the number of schemas can be a diverse depending on the preference of integration, (Binary approaches) is special case from (iterative nary), which minimizes the probability of the complexity of integration and lead to automation techniques, the integration by nary process can integration to perform operations on more than two schemas For practical reasons, most systems are used (binary methodology), but a number of researchers prefer (nary approaches), because the information available is complete

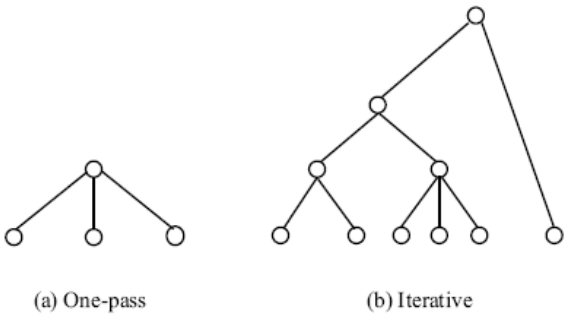

Fig. 4.12 *N*ary Integration Methods

3.2.2.3 Schema Mapping

After the completion of the identification (GCS) (global conceptual schema) it would be necessary to determine how the data from each local database can be mapped to (GCS) (target), while maintaining the consistency semantic as identified by both the source and the target, in spite of that schema matching identified matches between (LCS) and (GCS), but not

be a clearly identified, how to get the global database from local ones this is what trying to achieve schema Mapping, there are two types of issues related to the Schema Mapping They mapping creation and mapping maintenance Mapping creation is the creation Processes by clear queries that map data from local databases to global data but the mapping maintenance is the detection and correction of the contradictions that result from Schema Mapping [9]

1. mapping creation : its creation Processes a clear queries that map data from local databases to global data at this stage we need to point out some things to help clarify understanding, sources (LCS), which represents a group of local conceptual schema and we refer that it S = {S1, S2, ... Sm} as well as (GCS) global conceptual schema that has been generated as a result of the integration (LCS) in the second stage and is the schema integration we will refer to him T= {T1,T2,…TN} , In addition to (M) consists of a set of schema match that have been generated in the first step which is has matches between (LCS), now we need to implement the four steps to implement the map to get the global data based on existing data [11-10]

The first step : is the generation of a set containing matches between the (T), which represents a set (GCS) and between a set (M) that generated from the stage schema match called set generated candidate sets

The second step : is an analysis on the attributes in the candidate sets to find join path in every candidate sets using foregn key

The third step : is make cover to the candidate sets, the cover, is a subset of the candidate sets that has at least one match

The fourth step : and the last is built query for each candidate sets in cover selected in the previous step, the Union for all those queries is the result the final mapping, and generating steps for built queries are as follows:

SELECT CLAUSE : Include all correspondences for each candidate set of the selected cover in step3

FROM CLAUSE : Source relation

WHERE FILTER AND JOIN CONDITION : From join path in step2

For the entire selected cover

Compute the UNION of the SELECT blocks

4. CONCLUSION

- Distributed DataBases either homogeneous or heterogeneous
- Are two Methods to implement both types TOP-DOWN-METHOD and BOTTOM – UO METHOD
- TOP-DOWN-METHOD Appropriate for implementation of homogeneous databases and consists of:
  - ✓ Fragmentation
  - ✓ Allocation
- BOTTOM – UP METHOD Appropriate for implementation of heterogeneous databases and consists of:
  - ✓ Schema Translation
  - ✓ Schema Generation (Integration)
  - ✓ Schema Mapping

REFERENCES

[1] Özsu, M. Tamer, and Patrick Valduriez. *Principles of distributed database systems*. Vol. 2. Englewood Cliffs: Prentice Hall, 1999.

[2] Chomicki, Jan. "Data Integration: Schema Mapping." (2007).

[3] Ma, Hui. Distribution design for complex value databases: a dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Information Systems at Massey University. Diss. Massey University, 2007.

[4] Gamper, Johann. "Distributed Databases." SRV College of Education, Tamil Nadu (2009).

[5] Khan, Shahidul Islam, and A. S. M. L. Hoque. "A new technique for database fragmentation in distributed systems." International Journal of Computer Applications 5.9 (2010): 20-24.

[6] Hauglid, Jon Olav, Norvald H. Ryeng, and Kjetil Nørvåg. "DYFRAM: dynamic fragmentation and replica management in distributed database systems." Distributed and Parallel Databases 28.2-3 (2010): 157-185.

[7] Bodamer, Roger, Eric Voss, and Jacco Draaijer. "Apparatus and method for accessing foreign databases in a heterogeneous database system." U.S. Patent No. 6,236,997. 22 May 2001.

[8] Tian, Qing, and Songcan Chen. "Cross-heterogeneous-database age estimation through correlation representation learning." Neurocomputing 238 (2017): 286-295.

[9] Coden, Anna Rosa, JoAnn Piersa Brereton, and Michael Stephen Schwartz. "System and method for performance complex heterogeneous database queries using a single SQL expression." U.S. Patent No. 6,341,277. 22 Jan. 2002.

[10] Mack, Robert. "Method and apparatus for converting heterogeneous databases into standardized homogeneous databases." U.S. Patent No. 8,554,801. 8 Oct. 2013.

[11] Adlassnig, K. P. "A semantic approach for the homogeneous identification of events in eight patient databases: a contribution to the European eu-ADR project." Medical Informatics in a United and Healthy Europe: Proceedings of MIE 2009, the XXII International Congress of the European Federation for Medical Informatics. Vol. 150. Ios PressInc, 2009.

[12] Mishra, Ms Anju, Ms Gunjan Nehru, and Mr Ashish Pandey. "Dynamic programming solution for query optimization in homogeneous distributed databases." International Journal of Engineering 1 (2012).

[13] http://ynucc.yu.ac.kr/~hrcho/Courses/DDB/Chap3.pdf