A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective

Y. Richard Wang and Stuart E. Madnick

Composite Information Systems Laboratory, Room E53-320, Sloan School of Management
Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139 USA

ABSTRACT

This paper studies heterogeneous database systems from the multiple *(poly)* source *(gen)* perspective. It aims at addressing issues such as "where is the data from" and "which intermediate data sources were used to arrive at that data" -- issues which are critical to many users in utilizing information composed from multiple sources. Specifically, it presents a polygen model for resolving the *Data Source Tagging* and *Intermediate Source Tagging* problems. Secondly, it presents a data-driven query translation mechanism for mapping a polygen query into a set of local queries dynamically. A concrete example is also provided to exemplify polygen query processing.

The significance of this paper lies not only in a precise characterization of a practical problem and a solution per se, but also in the establishment of a foundation for resolving many other critical research issues such as domain mismatch, semantic reconciliation, and data conflict amongst data retrieved from different sources. In a federated database environment with hundreds of databases, all of these issues are critical to their effective use.

I. Introduction

The increasingly globalized economy has driven many corporations to expand business beyond their traditional geographic and organizational boundaries. It is widely recognized today that many important application systems require access to and integration of multiple heterogeneous database systems both within and across organizational boundaries [4, 10, 15, 53, 66]. These types of application systems have been referred to as *Federated Database Systems* [23, 26, 32, 41, 51], *Multidatabase Systems* [49] or *Composite Information Systems* (CIS) [37, 38, 52, 79-83, 85].

This paper presents a *polygen model*[1] to study heterogeneous database systems from the multiple *(poly)* source *(gen)* perspective. It aims at addressing issues such as "where is the data from" and "which intermediate data sources were used to arrive at that data" -- issues which are critical to many users in utilizing information composed from multiple sources. To the best of our knowledge, these issues have not been addressed before[2]. To date, heterogeneous (distributed) database systems strive to encapsulate the heterogeneity of the underlying databases in order to produce an illusion that all information originates from a single source, often referred to as *location transparency*. In our field studies of actual needs, we have found that although the users want the simplicity of making a query as if it were a single large database, they also want the ability to know the source of each piece of data retrieved.

The significance of this paper lies not only in a precise characterization of a practical problem and a solution per se, but also in the establishment of a foundation for resolving many other critical research issues. For example, knowing the data source will enable a user or a query processor to interpret the data semantics more accurately, and knowing the data source credibility will enable the user or the query processor to further resolve potential conflicts amongst the data retrieved from different sources. Moreover, the polygen model has been developed as a direct extension of the Relational Model to the multiple database setting with source tagging capabilities, thus it enjoys all of the strengths of the traditional Relational Model.

Proceedings of the 16th VLDB Conference
Brisbane, Australia 1990

1 To highlight the source tagging problems, the phrase "polygen model" will be used in the paper instead of the conventional "global model." By the same token, "polygen query" will be used instead of "global query," and so on, and so forth.

2 In MRDSM [49], an administrator may define for any collection of databases a collective name called a multidatabase name. For instance, the databases Michelin, Kleber, and Gault_M may collectively get the name Rest_guides. However, the focus of such names is to simplify the expression of some commands; otherwise, these commands may require an enumeration of the corresponding databases.

NEED FOR SOURCE TAGGING

Most end-users wish to know the source of their data (e.g., "Source: Reuters' Newstext, Thursday, May 31, 1990) This source knowledge may be important to them for many reasons. For example, it enables them to apply their own judgment to the credibility of the information. We call this the *Data Source Tagging* problem.

A decision maker may need to know not only the sources of information but also the intermediate sources that helped in composing the information. We call this the *Intermediate Source Tagging* problem.

For instance, in preparing a special report on the top ten graduate programs in Information Systems [ComputerWorld, October 30, 1989], Sullivan-Trainor, a ComputerWorld staff, called the top schools to get the names of CEO's who graduated from these schools with an MBA degree. In order to respond to his request, let us assume that the following SQL polygen query

```
SELECT CEO
FROM PORGANIZATION, PALUMNUS
WHERE CEO = ANAME AND DEGREE = "MBA"
```

was created based on a polygen schema derived from an Alumni Database and a Company Database as shown below.

Polygen Schema	Alumni Database (AD): Alumni Schema	Company Database (CD): Company Schema
PORGANIZATION(ONAME, CEO, IND)	BUSINESS(BNAME, IND)	FIRM(FNAME, CEO)
PALUMNUS(AID#, ANAME, DEGREE, MAJOR)	ALUMNUS(AID#, ANAME, DEG, MAJ)	

The query result contains only the names of CEO which originated from the Company Database, but the query processor also needs to access the Alumni Database (an intermediate source) in order to select those CEOs who received an MBA degree. Moreover, the query processor needs to "know" that it has to merge the BUSINESS and the FIRM relations first before joining the CEO attribute with the ANAME attribute. As such, the challenge is to develop not only a polygen model but also a polygen algebra and the algorithms for a polygen query processor capable of resolving the data and intermediate source tagging problems for any arbitrary polygen query. Tagging the Company Database name accurately to the result is referred to as the *Data Source Tagging* problem. Tagging the intermediate use of the Alumni Database accurately is referred to as the *Intermediate Source Tagging* problem.

The data and intermediate source tagging problems have not been dealt with to date. We have reviewed a broad range of literature (see the bibliography) and examined various research prototypes of heterogeneous distributed database systems, for example MULTIBASE in the United States [23-25, 39, 74-75], PRECI* in England [26-27], and MRDSM in France [49-50]. In addition, we have surveyed more than forty U.S. commercial systems offering partial solutions to the heterogeneous distributed database problem, including Data Integration's MERMAID, Cincom's SUPRA, Metaphor's DIS, and TRW's Data Integration Engine [40]. To the best of our knowledge, none of these systems have dealt with these source tagging problems.

RESEARCH AND GOALS

Two related issues, among others, need to be addressed in source tagging: (1) What kind of polygen model should be created in order to tag multiple sources explicitly? (2) What is the relationship between the polygen model and the polygen query processing facility?

Most heterogeneous distributed database systems adopt one of the following four data models [42, 65]: the Relational Model, the Functional Data Model, the Semantic Database Model, or the Entity Relationship Model. Each data model has merits for its intended purposes.[3] We selected the relational model. Based on the relational model, we define a *polygen model* for resolving the data and intermediate source tagging problems.

One of the key activities in formulating composite information is to translate a polygen query into a set of local queries, which in turn are routed to the corresponding local databases. Query translation has been approached through view definition in most heterogeneous distributed database systems [6, 23-24, 45, 49]. A symbolic query transformation technique has also been proposed [21, 69-70] in which a syntax-directed parser converts a polygen query and transformation rules[4] into multiway trees. Through subtree matching,

3 Both the Functional Data Model and the Semantic Database Model are rich in semantics and implemented in operational systems. The Entity Relationship Model is also rich in semantics and is widely accepted as the leading database design tool. The relational model lends itself to a simple structure and an elegant theoretical foundation. Its Relational Data Base Management Systems dominate the database market today. Codd [1979] also extended the relational model to capture semantics such as generalization and aggregation.

4 Each transformation rule contains a source part and a target part. For example,
Source:SELECT attribute-1 FROM relation-1 WHERE condition; Target: Projection ((attribute-1), Selection (condition, (relation-1));

these multiway trees are further translated into local queries, given the specific source and target language syntax descriptions.

As we will discuss later, our query translation mechanism differs from the above mentioned techniques in two important aspects: (1) Instead of the view definition approach which encodes the procedure for translating a polygen query into the corresponding local queries, our mechanism separates the mapping algorithm from the mapping data. As a result, adding a new database to the existing system does not require modifying the existing procedural view definitions. (2) Instead of the symbolic query transformation technique which tackles a broad range of nodal query languages at a higher level, our mechanism focuses on the mapping between a polygen algebraic expression and the corresponding local operations, permitting entities (and attributes) in local databases to overlap one another.[5]

In sum, the first goal of this paper is to present a *polygen model*. Secondly, it presents a data-driven query translation mechanism for mapping a polygen query into a set of local queries dynamically. A concrete example is also provided to exemplify polygen query processing.

RESEARCH BACKGROUND AND ASSUMPTIONS

At the Composite Information Systems Laboratory, Sloan School of Management, MIT, we have evolved a research prototype which has access to three internal MIT databases (the Alumni Database, the Placement Database, and the Student Database) and three external commercial databases (Finsbury's Dataline and I.P. Sharp's Disclosure and Currency, both owned by Reuters Holdings PLC.). These databases provide breadth in data and examples of differences in style, accentuated somewhat by the different origins. For example, Finsbury is based in England, I.P. Sharp in Canada, and the MIT databases in the United States.

The prototype query processor architecture is depicted in Figure 1. Briefly, the Application Query Processor translates an end-user query into a polygen query for the Polygen Query Processor (PQP) based on the user's application schema. The PQP in turn translates the polygen query into a set of local queries based on the corresponding polygen schema, and routes them to the Local Query Processors (LQP). The details of the mapping and communication mechanisms between an LQP and its local data bases is encapsulated in the LQP. To the PQP, each LQP behaves as a local relational system. Upon return from the LQPs, the retrieved data are further processed by the PQP in order to produce the desired composite information.[6]

Many critical problems need to be resolved in order to provide a seamless solution to the end-user. These problems include source tagging, query translation, schema integration [5, 32], inter-database instance matching [82], domain mapping [28, 81], and semantic reconciliation [79]. We focus on the first two problems and make the following assumptions in this paper:

- The local schemata and the polygen schema are all based on the relational model.
- Sources are tagged after data has been retrieved from each database.
- Schema integration has been performed, and the attribute mapping information is stored in the polygen schema.
- The inter-database instance identifier mismatching problem (e.g., IBM vs. I.B.M or social security identification number vs. employee identification number) has been resolved and the information is available for the PQP to use.
- The domain mismatch problem such as unit ($ vs. ¥), scale (in billions vs. in millions), and description interpretation ("expensive" vs. "$$$", "Chinese Cuisine" vs. "Hunan or Cantonese") has been resolved in the schema integration phase and the domain mapping information is also available to the PQP.

Section II defines the polygen model. Polygen query translation is presented in Section III. Section IV provides a detailed example of polygen query processing. Finally, concluding remarks are made in section V.

II. The Polygen Model

We first embellish the scenario described in Section I in order to exemplify the polygen model to be presented in this section. Let us assume that the following three local relational schemata had been chosen instead.

---

5 Katz and Goodman [1981] first explored view definition in MULTIBASE. There it was assumed that the local databases were disjoint (i.e., contains no entities in common). However, the most interesting (and difficult) problems occur when the local databases do overlap. This class of problem was tackled by Dayal and Hwang [1984] using the view definition approach in the context of the Functional Data Model.

6 Hierarchical, network, and other data models can be accommodated by including the necessary data-model translations. The problems of data model translation has been addressed, among others, in MULTIBASE and SCOOP [Spaccapietra et al]. In fact, our prototype's LQP can handle unusual query interfaces, such as I.P. Sharp's proprietary query language and Finsburg's menu-driven interface [Paget, 1989; Wong, 1989].

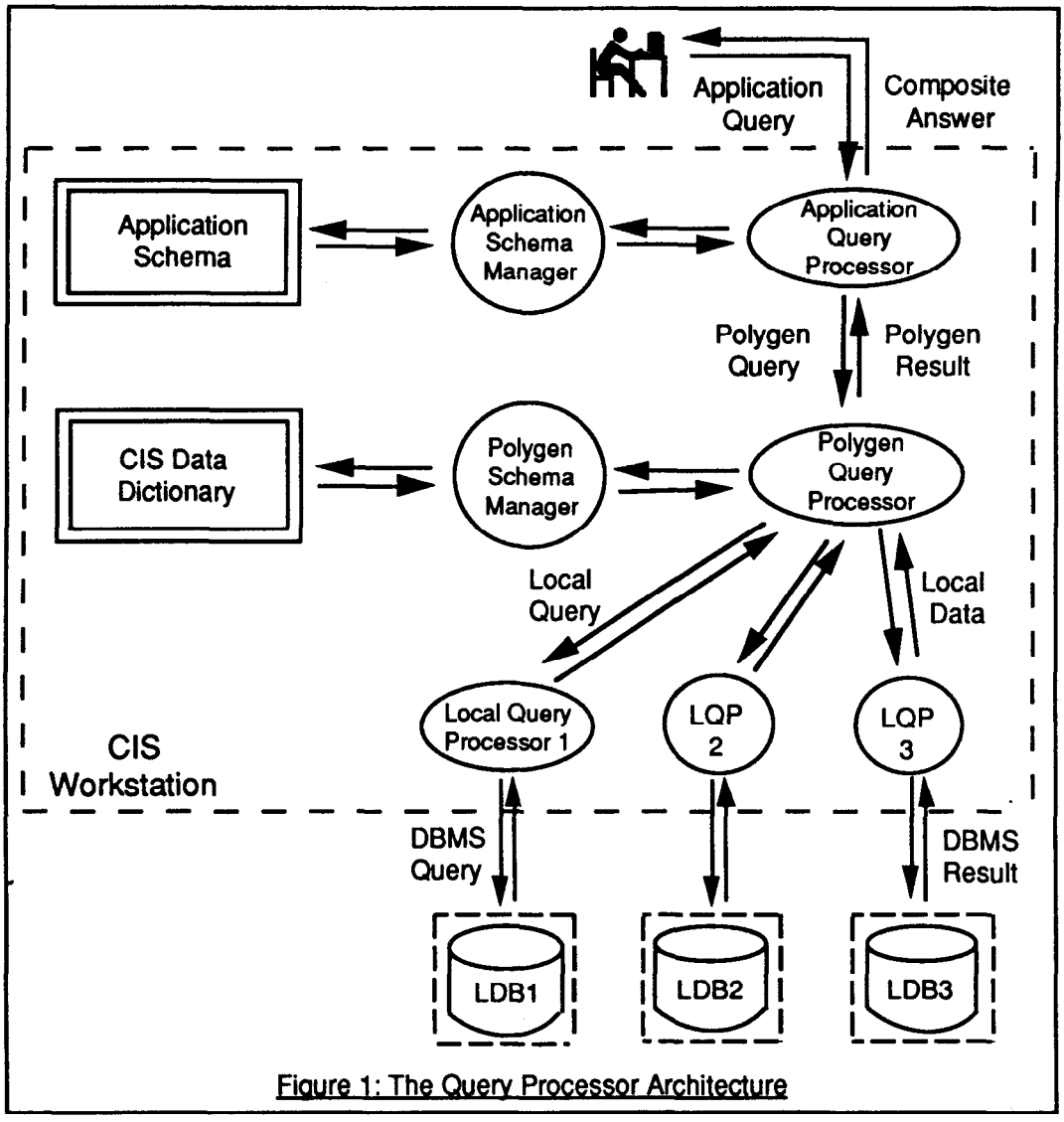

Figure 1: The Query Processor Architecture

Alumni Database (AD)	Placement Database (PD)	Company Database (CD)
ALUMNUS(AID#, ANAME, DEG, MAJ)	STUDENT(SID#, SNAME, GPA, MAJOR)	FIRM(FNAME, CEO, HQ)
CAREER(AID#, BNAME, POS)	INTERVIEW(SID#, CNAME, JOB, LOC)	FINANCE(FNAME, YR, PROFIT)
BUSINESS(BNAME, IND)	CORPORATION (CNAME, TRADE, STATE)	

Each alumnus in the Alumni Database is uniquely identified through an alumnus identification number (AID#). Associated with each alumnus is a name, a degree, and a major. An alumnus may have positions in many businesses. Finally, a business is associated with an industry.

A student in the Placement Database is uniquely identified by a student identifier number, and associated with a name, a GPA, and a major. A student may schedule interviews with many corporations for a job at a certain location. Finally, a corporation is associated with a trade and is headquartered in a state.

A firm in the Company Database has a name, a CEO, and is headquartered in a city. It discloses yearly financial information on profit.

A corresponding polygen schema is shown as follows:
PALUMNUS (AID#, ANAME, DEGREE, MAJOR)
PCAREER (AID#, ONAME, POSITION)
PORGANIZATION (ONAME, INDUSTRY, CEO, HEADQUARTERS)
PSTUDENT (SID#, SNAME, GPA, MAJOR)
PINTERVIEW (SID#, ONAME, JOB, LOCATION)
PFINANCE (ONAME, YEAR, PROFIT)

For expository purposes, we will use the prefix "P" to

denote that it is a *polygen scheme*. Although most of the polygen relations correspond to specific local relations, we note that the PORGANIZATION scheme combines the BUSINESS scheme in the Alumni Database, the CORPORATION scheme in the Placement Database, and the FIRM scheme in the Company Database. The attribute mapping relationships in the form (database, relation, attribute) are shown below.

The PALUMNUS Polygen Scheme

AID#	ANAME	DEGREE	MAJOR
{(AD, ALUMNUS, AID#)}	{(AD, ALUMNUS, ANAME)}	{(AD, ALUMNUS, DEG)}	{(AD, ALUMNUS, MAJ)}

The PCAREER Polygen Scheme

AID#	ONAME	POSITION
{(AD, CAREER, AID#)}	{(AD, CAREER, BNAME)}	{(AD, CAREER, POS)}

The PORGANIZATION Polygen Scheme

ONAME	INDUSTRY	CEO	HEADQUARTERS
{(AD, BUSINESS, BNAME), (PD, CORPORATION, CNAME), (CD, FIRM, FNAME)}	{(AD, BUSINESS, IND), (PD, CORPORATION, TRADE)}	{(CD, FIRM, CEO)}	{(PD, CORPORATION, STATE), (CD, FIRM, HQ)}

The PSTUDENT Polygen Scheme

SID#	SNAME	GPA	MAJOR
{(PD, STUDENT, SID#)}	{(PD, STUDENT, SNAME)}	{(PD, STUDENT, GPA)}	{(PD, STUDENT, MAJOR)}

The PINTERVIEW Polygen Scheme

SID#	ONAME	JOB	LOCATION
{(PD, INTERVIEW, SID#)}	{(PD, INTERVIEW, CNAME)}	{(PD, INTERVIEW, JOB)}	{(PD, INTERVIEW, LOC)}

The PFINANCE Polygen Scheme

ONAME	YEAR	PROFIT
{(CD, FINANCE, FNAME)}	{(CD, FINANCE, YR)}	{(CD, FINANCE, PROFIT)}

We now define the polygen model. Let PA be a polygen attribute in a polygen scheme P, LS a local scheme in a local database LD, and LA a local attribute in LS. For example, ONAME is a polygen attribute in the polygen scheme PORGANIZATION, BUSINESS a local scheme in the local database AD, and BNAME a local attribute in the local scheme BUSINESS.

Let MA be the set of local attributes corresponding to a PA, i.e.,

MA = {(LD, LS, LA) | (LD, LS, LA) denotes a local attribute to the corresponding PA}.

For ONAME in the PORGANIZATION polygen scheme, MA = {(AD, BUSINESS, BNAME), (PD, CORPORATION, CNAME), (CD, FIRM, FNAME)}.

A polygen scheme P is defined as
$P = \langle (PA_1, MA_1), \ldots, (PA_n, MA_n) \rangle$ where n is the number of attributes in P.

For the polygen scheme PORGANIZATION in the above scenario,

PORGANIZATION = ⟨(ONAME, {(AD, BUSINESS, BNAME), (PD, CORPORATION, CNAME), (CD, FIRM, FNAME)}), (INDUSTRY, {(AD, BUSINESS, IND), (PD, CORPORATION, TRADE)}), (CEO, {(CD, FIRM, CEO)}), (HEADQUARTERS, {(PD, CORPORATION, STATE), (CD, FIRM, HQ)})⟩

A polygen schema is defined as a set $\{P_1, \ldots, P_N\}$ of N polygen schemes. In the above scenario, the polygen schema consists of the following schemes:

{PALUMNUS, PCAREER, PORGANIZATION, PSTUDENT, PINTERVIEW, PFINANCE}

A polygen domain is defined as a set of ordered triplets. Each triplet consists of three elements: the first is a datum drawn from a simple domain in an LQP. The second is a set of LDs denoting the local databases from which the datum originates. The third is a set of LDs denoting the intermediate local databases whose data led to the selection of the datum.

A polygen relation p of degree n is a finite set of time-varying n-tuples, each n-tuple having the same set of attributes drawing values from the corresponding polygen domains. A cell in a polygen relation is an ordered triplet c=(c⟨d⟩, c⟨o⟩, c⟨i⟩) where c⟨d⟩ denotes the datum portion, c⟨o⟩ the originating portion, and c⟨i⟩ the intermediate source portion. Two polygen relations are union-compatible if their corresponding attributes are defined on the same polygen domain.

Note that P contains the mapping information between a polygen scheme and the corresponding local relational schemes. In contrast, p contains the actual time-varying data and their originating sources. Occasionally, a polygen scheme and a polygen relation may be used synonymously without confusion. The data and intermediate source tags for p are updated along the way as polygen algebraic operations are performed.

THE POLYGEN ALGEBRA

Let attrs(p) denote the set of attributes of p. For each tuple t in a polygen relation p, let t⟨d⟩ denote the data portion, t⟨o⟩ the originating source portion, and t⟨i⟩ the intermediate source portion. If $x \in attrs(p)$, $X = \{x_1 \ldots, x_j, \ldots, x_J\}$ is a sublist of attrs(p), then let p[x] be the column in p corresponding to attribute x, let p[X] be the columns in p corresponding the sublist of attributes X, let t[x] be the cell in t corresponding to attribute x, and let t[X] be the cells in t corresponding to the sublist of attributes X. As such, p[x]⟨o⟩ denotes the originating source portion of the column corresponding to attribute x in polygen relation p while t[X]⟨i⟩ denotes the intermediate source portion of the cells corresponding to the sublist of attributes X in tuple t. On the other hand, p[x] denotes the column corresponding to attribute x in polygen relation p inclusive of the data, originating source, and intermediate source portions while t[X] denotes the cells corresponding to the sublist of attributes X in tuple t inclusive of the data, originating source, and intermediate source portions.

The five orthogonal algebraic primitive operators [16-20, 47] in the polygen model are defined as follows:

Project. If p is a polygen relation, and $X = \{x_1 \ldots, x_j, \ldots, x_J\}$ is a sublist of attrs(p), then

$p[X] = \{t' \mid t' = t[X]$ if $t \in p \wedge t[X]\langle d\rangle$ is unique;
$t'\langle d\rangle = t_i[X]\langle d\rangle$, $t'[x_j]\langle o\rangle = t_i[x_j]\langle o\rangle \cup \ldots \cup$
$t_k[x_j]\langle o\rangle \; \forall \; x_j \in X$, $t'[x_j]\langle i\rangle = t_i[x_j]\langle i\rangle \cup \ldots \cup t_k[x_j]\langle i\rangle \; \forall \; x_j \in X$
if $t_i, \ldots t_k \in p \wedge t_i[X]\langle d\rangle = \ldots = t_k[X]\langle d\rangle\}$.

Cartesian product. If p_1 and p_2 are two polygen relations, then

$(p_1 \times p_2) = \{t_1 \circ t_2 \mid t_1 \in p_1$ and $t_2 \in p_2$ where $\circ$ denotes concatenation$\}$.

Restrict. If p is a polygen relation, $x \in attrs(p)$, $y \in$ attrs(p), and θ is a binary relation, then

$p[x \; \theta \; y] = \{t' \mid t'\langle d\rangle = t\langle d\rangle, t'\langle o\rangle = t\langle o\rangle, t'[w]\langle i\rangle = t[w]\langle i\rangle \cup$
$t[x]\langle o\rangle \cup t[y]\langle o\rangle \; \forall \; w \in attrs(p)$,
if $t \in p \wedge t[x]\langle d\rangle \; \theta \; t[y]\langle d\rangle\}$.

Union. If p_1 and p_2 are two polygen relations and both have degree n, $t_1 \in p_1, t_2 \in p_2$, then

$(p_1 \cup p_2) = \{t' \mid t' = t_1$ if $t_1\langle d\rangle \in p_1 \wedge t_1\langle d\rangle \notin p_2$;
$t' = t_2$ if $t_2\langle d\rangle \notin p_1 \wedge t_2\langle d\rangle \in p_2$;
$t'\langle d\rangle = t_1\langle d\rangle$, $t'\langle o\rangle = t_1\langle o\rangle \cup t_2\langle o\rangle$, $t'\langle i\rangle = t_1\langle i\rangle \cup t_2\langle i\rangle$ if
$t_1\langle d\rangle = t_2\langle d\rangle\}$

Difference. Let p⟨o⟩ denote the union of all the t⟨o⟩ sets in p, and p⟨i⟩ denote the union of all the t⟨i⟩ sets in p. If p_1 and p_2 are two polygen relations and both have degree *n*, then

$(p_1 - p_2) = \{t' \mid t'\langle d\rangle = t\langle d\rangle, t'\langle o\rangle = t\langle o\rangle, t'[w]\langle i\rangle = t[w]\langle i\rangle \cup$
$p_2\langle o\rangle \cup p_2\langle i\rangle \; \forall \; w \in attrs(p)$, if $t \in p_1$ and $t\langle d\rangle \notin p_2\}$.

The intermediate source portion, t⟨i⟩, is updated by *Restrict* and *Difference*. The *Restrict* operation selects the tuples in a polygen relation which satisfies the [x θ y] condition. As such, the originating local databases of the x and y attribute values are added to the t⟨i⟩ set in order to signify their mediating role. Since *Select* and *Join* are defined through *Restrict*, they also update t⟨i⟩.

Difference selects a tuple in p_1 to be a tuple in $(p_1 - p_2)$ if the data portion of the tuple in p_1 is not identical to those of the tuples in p_2. Since each tuple in p_1 needs to be compared with all the tuples in p_2, it follows that all the originating sources of the data in p_2 should be included in the intermediate source set of $(p_1 - p_2)$, as $t'\langle i\rangle = t\langle i\rangle \cup p_2\langle o\rangle \cup p_2\langle i\rangle$ denotes.

In contrast, *Project*, *Cartesian Product*, and *Union* do not involve intermediate local databases as the mediating sources. Other traditional operators can be defined in terms of the above five operators. The most common are *Join*, *Select*, and *Intersection*. *Join* and *Select* are defined as the restriction of a *Cartesian product*. *Intersection* is defined as the project of a join over all the attributes in each of the relations involved in the Intersection.

In order to process a polygen query, we also need to introduce the following new operators to the polygen model: *Retrieve*, *Coalesce*, *Outer Natural Primary Join*, *Outer Natural Total Join*, and *Merge*.

A local database relation needs to be retrieved from a local database to the PQP first before it is considered as a *PQP base relation*. This is required in the polygen model because a polygen operation may require data from multiple local databases. Although a *PQP base relation* can be materialized dynamically like a view in the conventional database system, for conceptual purposes,

we define it to reside physically in the PQP.[7] The *Retrieve* operation can be defined as an LQP *Restrict* operation without any restricting condition.

Coalesce and *Outer Natural Join* have been informally introduced by Date to handle a surprising number of practical applications. *Coalesce* takes two columns as input, and coalesce them into one column. An *Outer Natural Join* is an outer join with the join attributes coalesced [22].

We define an *Outer Natural Primary Join* as an *Outer Natural Join* on the primary key of a polygen relation. For example, the *Outer Natural Primary Join* for PORGANIZATION is an *Outer Natural Join* on ONAME. An *Outer Natural Total Join* is an *Outer Natural Primary Join* with all the other polygen attributes in the polygen relation coalesced as well. In the PORGANIZATION example, an *Outer Natural Total Join* would perform an *Outer Natural Primary Join* on ONAME followed by a number of *Coalesce* operations on INDUSTRY, CEO, and HEADQUARTERS. *Merge* extends *Outer Natural Total Join* to include more than two polygen relations. It can be shown that the order in which *Outer Natural Total Join* are performed over a set of polygen relations in a *Merge* is immaterial.

Since *Coalesce* can be used in conjunction with the other polygen algebraic operators to define the *Outer Natural Primary Join, Outer Natural Total Join,* and *Merge,* we define *Coalesce* as the sixth orthogonal primitive of the polygen model.

Coalesce. Let © denote the coalesce operator. If p is a polygen relation, x ∈ attrs(p), y ∈ attrs(p), z = attrs(p) - {x, y}, and w is the coalesced attribute of x and y, then p[x © y:w] =

{t' | t'[z]=t[z], t'[w]⟨d⟩=t[x]⟨d⟩, t'[w]⟨o⟩ =t[x]⟨o⟩ ∪t[y]⟨o⟩, t'[w]⟨i⟩ =t[x]⟨i⟩ ∪t[y]⟨i⟩, if t[x]⟨d⟩=t[y]⟨d⟩;

t'[z]=t[z], t'[w]⟨d⟩=t[x]⟨d⟩, t'[w]⟨o⟩ =t[x]⟨o⟩, t'[w]⟨i⟩ =t[x]⟨i⟩, if t[y]⟨d⟩=nil;

t'[z]=t[z], t'[w]⟨d⟩=t[y]⟨d⟩, t'[w]⟨o⟩ =t[y]⟨o⟩, t'[w]⟨i⟩ =t[y]⟨i⟩, if t[x]⟨d⟩=nil}.

Note that in a heterogeneous distributed environment, the values to be coaleased may be inconsistent. That issue is beyond the scope of this paper, we have assumed that inter-database instance mismatching problems will be resolved before the coalease operation is performed [82, 86].

We have presented the polygen model and the polygen algebra. The algebra will be used in Section IV to compose information with *data source tags* and *intermediate source tags*. In order to do that, it is necessary to know the process of translating a polygen query into a query execution plan. This process is presented below.

III. Polygen Query Translation

To illustrate how a polygen query would be processed, we now describe a possible polygen query translator algorithm. For brevity of exposition, this algorithm does not employ any sophisticated optimization techniques. As a simpler example, let us assume that the following SQL polygen query was submitted to the PQP in order to respond to Sullivan-Trainor's request:

```
SELECT   ONAME, CEO
FROM     PORGANIZATION,PALUMNUS
WHERE    CEO = ANAME  AND  ONAME IN
           (SELECT ONAME FROM
           PCAREER WHERE  AID# IN
             (SELECT AID#  FROM
             PALUMNUS  WHERE
             DEGREE = "MBA"))
```

A corresponding polygen algebraic expression for the SQL polygen query is as follows:

((((PALUMNUS [DEGREE = "MBA"]) [AID#=AID#] PCAREER) [ONAME = ONAME] PORGANIZATION) [CEO = ANAME]) [ONAME, CEO]

In this expression, those alumni with an MBA degree are selected from the PALUMNUS relation. The result is joined with the PCAREER relation on AID#, which in turn is joined with PORGANIZATION on ONAME, followed by a restriction on "CEO=ANAME" followed by a projection on ONAME and CEO.

In general, the PQP takes a polygen algebraic expression as an input and produces a query execution plan for retrieving data from the local databases and formulating composite information. Three components are involved in this process: the *Algebraic Analyzer*, the *Polygen Operation Interpreter*, and the *Query Optimizer*, as shown in Figure 2.

---

7 This approach simplifies the Polygen Operation Interpreter, to be presented in Section III.

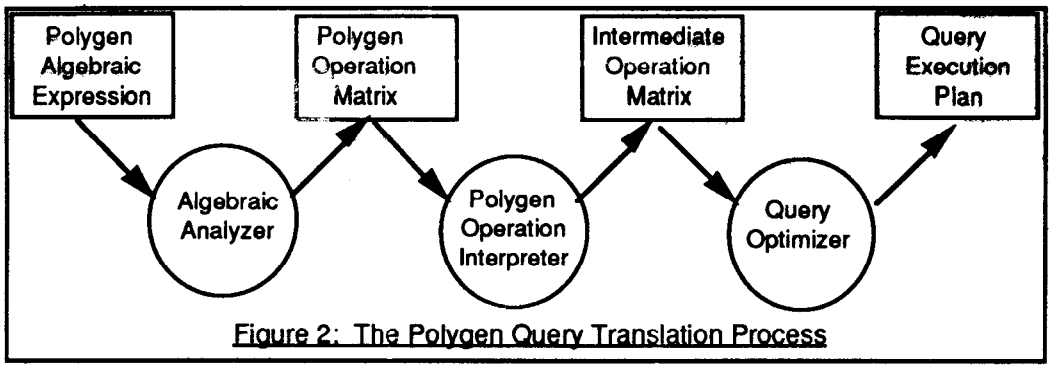

Figure 2: The Polygen Query Translation Process

The *Algebraic Analyzer* parses a polygen algebraic expression and generates a *Polygen Operation Matrix*. For example, the *Polygen Operation Matrix* for the example polygen algebraic expression is presented in Table 1 below. The first row indicates that a *Select* operation should be performed on the Left-Hand Relation (LHR) PALUMNUS using the θ relation "=" between the Left-Hand Attribute (LHA) DEGREE and the Right-Hand Attribute (RHA) "MBA." In this case, there is no need for a Right-Hand Relation (RHR). The result is denoted by R(1), a Polygen Relation (PR). Details of the *Algebraic Analyzer* is beyond the scope of this paper.

Table 1: The Polygen Operation Matrix for the Example Polygen Algebraic Expression

PR	OP	LHR	LHA	θ	RHA	RHR
R(1)	Select	PALUMNUS	DEGREE	=	"MBA"	nil
R(2)	Join	R(1)	AID#	=	AID#	PCAREER
R(3)	Join	R(2)	ONAME	=	ONAME	PORGANIZATION
R(4)	Restrict	R(3)	CEO	=	ANAME	nil
R(5)	Project	R(4)	ONAME, CEO	nil	nil	nil

Next the *Polygen Operation Interpreter* expands the *Polygen Operation Matrix* and generates an *Intermediate Operation Matrix*. In addition to the *Polygen Operation Matrix*, the *Polygen Operation Interpreter* takes the polygen schema as an input in order to produce the *Intermediate Operation Matrix*. For clarity, a two-pass *Polygen Operation Interpreter*, pass one dealing with the left-hand side and pass two the right-hand side of polygen operations, is presented below.

The input to pass one is a *Polygen Operation Matrix* as Table 1 exemplifies and an empty *Intermediate Operation Matrix*. The output from pass one (and input to pass two) is a half-processed *Intermediate Operation Matrix*, as shown in Table 2. The *execution location* (EL) of an operation depends on where the data resides. Note that when the execution location is an LQP (e.g., AD in the first row of Table 2), it is also used as the *originating source tag* for each of the cell, c(o), of the polygen base relation (R(1) in this case).

Table 2: A Half-Processed IOM Generated by Pass One of the POI Algorithm

PR	OP	LHR	LHA	θ	RHA	RHR	EL
R(1)	Select	ALUMNUS	DEG	=	"MBA"	nil	AD
R(2)	Join	R(1)	AID#	=	AID#	PCAREER	PQP
R(3)	Join	R(2)	ONAME	=	ONAME	PORGANIZATION	PQP
R(4)	Restrict	R(3)	CEO	=	ANAME	nil	PQP
R(5)	Project	R(4)	ONAME, CEO	nil	nil	nil	PQP

In this example, pass one recognizes that the first row of Table 1 contains the polygen relation PALUMNUS whose attribute DEGREE corresponds to {(AD,ALUMNUS,DEG)}. Thus, LS=ALUMNUS, LA=DEG, LD=AD, and the tuple ⟨R(1), Select, ALUMNUS, DEG, =, "MBA", nil, AD⟩ is inserted into the first row of Table 2 which is empty initially. The second to the fifth row of Table 1 are mapped into Table 2 without any change, and the PQP is assigned as the execution location because the left-hand relations, R(1) through R(4), reside in the PQP.

Continuing with the example, pass two processes the right-hand side of Table 2 and produces Table 3 below. In general, the *left-hand relation* is either a relation defined by the polygen schema or a R(#) denoting a polygen base relation (or a polygen relation derived from other polygen base relations). In the first case, the *left-hand relation* may correspond to either one or multiple local relations. If only one local relation exists, then the polygen operation is mapped into the local operation, and the corresponding LQP is assigned as the execution location. If multiple local relations exist, then these relations are retrieved and merged first before the requested operation is performed by the PQP.[8] The second case involves an update of the R(#) from the Polygen Operation Matrix to the corresponding R(#) in the half-processed Intermediate Operation Matrix. In addition, the PQP is assigned as the execution location because R(#) resides in the PQP. The pass one algorithm is presented in Figure 3 below.[9]

Continuing with the example, pass two processes the right-hand side of Table 2 and produces Table 3 .

```
k=1;  /* start from the first row of the Polygen Operation Matrix (POM) */
n=1;  /* start from the first row of the initially empty matrix H */
while k ≤ Cardinality(POM)  /* for each row in the POM, process the left-hand side */
begin
if POM(k,LHR)∈ P ∧ POM(k,LHA)=PA_i then  /* case: the LHR is defined in the polygen schema */
  if MA_i ={(LD,LS,LA)} then /* case: MA_i has a single element */
      begin
        H(n)=(R(n), POM(k,OP), LS, LA, POM(k,θ), POM(k,RHA), POM(k,RHR), LD); n=n+1;
      end
  else begin /* case: MA_i ={(LD_1,LS_1,LA_1),..., (LD_J,LS_J,LA_J)}*/
        j=1; while j ≤ J begin H(n)=(R(n), Retrieve, LS_j, nil, nil, nil, nil, LD_j); j=j+1; n=n+1; end
        H(n)=(R(n), Merge, {R(n-1),..., R(n-J)}, nil, nil, nil, nil, "PQP"); n=n+1;
        H(n)=(R(n), POM(k,OP), R(n-1), POM(k,LHA), POM(k,θ),
           POM(k,RHA), POM(k,RHR), "PQP"); n=n+1;
      end
else  begin/* case: R(#) */
        H(n)=(R(n), POM(k,OP), R(map(POM(k,LHR)))[10], POM(k,LHA),
                              POM(k,θ), POM(k,RHA), POM(k,RHR), "PQP"); n=n+1;
      end
k=k+1;  /* point to the next row of POM */
end
```

Figure 3: The Pass One Algorithm for the Polygen Operation Interpreter

Table 3: An Intermediate Operation Matrix for the Example Polygen Algebraic Expression

PR	OP	LHR	LHA	θ	RHA	RHR	EL
R(1)	Select	ALUMNUS	DEG	=	"MBA"	nil	AD
R(2)	Retrieve	CAREER	nil	nil	nil	nil	AD
R(3)	Join	R(1)	AID#	=	AID#	R(2)	PQP
R(4)	Retrieve	BUSINESS	nil	nil	nil	nil	AD
R(5)	Retrieve	CORPORATION	nil	nil	nil	nil	PD
R(6)	Retrieve	FIRM	nil	nil	nil	nil	CD
R(7)	Merge	R(4), R(5), R(6)	nil	nil	nil	nil	PQP
R(8)	Join	R(3)	ONAME	=	ONAME	R(7)	PQP
R(9)	Restrict	R(8)	CEO	=	ANAME	nil	PQP
R(10)	Project	R(9)	ONAME, CEO	nil	nil	nil	PQP

8 We do not consider any optimization here.

9 assuming that the Algebraic Analyzer has insured that a POM represents a legal polygen query.

10 The *map* function converts the R(#) denoted by POM(k,LHR) to the corresponding R(#) in H.

The first row of Table 2 is copied over to Table 3 directly because the *right-hand relation* is non-existent (nil) and no other mapping is required. The second row of Table 2 is a *join* of a polygen relation, R(1), with PCAREER which corresponds to a single local scheme CAREER. As such, the local relation CAREER is retrieved first (Row 2, Table 3) followed by a *Join* (Row 3, Table 3). The third row of Table 2 is a *Join* between R(2) and PORGANIZATION

```
k=1;  /* start from the first row of the half-processed Intermediate Operation Matrix (H) */
n=1;  /* start from the first row of the initially empty Intermediate Operation Matrix (IOM) */
while k ≤ Cardinality(H)  /* for each row in the Half-processed IOM, process the right-hand side */
begin
if H(k,RHR)∈ P ∧ H(k,RHA)=PA_i then  /* case: the RHR is defined in the polygen schema */
  if  MA_i ={(LD,LS,LA)} then /* case: MA_i has a single element */
      if H(k,EL)="PQP" then  /* Case: LHR already an R(#) */
        begin
           IOM(n)=⟨R(n), Retrieve, LS, nil, nil, nil, nil, LD⟩; n=n+1;
           IOM(n)=⟨R(n), H(k,OP), R(map(H(k,LHR))), H(k,LHA), H(k,θ), H(k,RHA),
                    R(n-1), "PQP"⟩; n=n+1;
        end
      else /* case: LHR and RHR both as defined in the polygen schema */
           begin
           IOM(n)=⟨R(n), H(k,OP), H(k,LHR), nil, nil, nil, nil, H(k,EL)⟩; n=n+1;
           IOM(n)=⟨R(n), Retrieve, LS, nil, nil, nil, nil, LD⟩; n=n+1;
           IOM(n)=⟨R(n), H(k,OP), R(n-2), PA(H(k,LHR), H(k,LHA))[11], H(k,θ), H(k,RHA),
                     R(n-1), "PQP"⟩; n=n+1;
           end
  else begin /* case: MA_i ={(LD_1,LS_1,LA_1),..., (LD_J,LS_J,LA_J)}*/
       j=1;
       while j ≤ J
             begin
                IOM(n)=⟨R(n), Retrieve, LS_j, nil, nil, nil, nil, LD_j⟩;
                j=j+1; n=n+1;
             end
       IOM(n)=⟨R(n), Merge, {R(n-1),..., R(n-J)}, nil, nil, nil, nil, "PQP"⟩; n=n+1;
       if H(k,EL)="PQP" then  /* Case: LHR already an R(#) */
                              begin
                              IOM(n)=⟨R(n), H(k,OP), R(map(H(k,LHR))), H(k,LHA),
                              H(k,θ), H(k,RHA), R(n-1), "PQP"⟩; n=n+1;
                              end
       else /* case: LHR and RHR both as defined in the polygen schema */
            begin
       IOM(n)=⟨R(n), Retrieve, (H(k,LHR)), nil, nil, nil, nil, H(k,EL)⟩; n=n+1;
       IOM(n)=⟨R(n), H(k,OP), R(n-1), PA(H(k,LHR), H(k,LHA)), H(k,θ), H(k,RHA), R(n-2), "PQP"⟩; n=n+1;
            end
    end
else  begin       /* case: R(#) or nil*/
      IOM(n)=⟨R(n), H(k,OP), R(map(H(k,LHR))), H(k,LHA),
      H(k,θ), H(k,RHA), R(map(H(k,RHR))), H(k,EL)⟩; n=n+1;
    end k=k+1;  /* point to the next row of H */
end
```

Figure 4: The Pass Two Algorithm of the Polygen Operation interpreter

[11] Return the corresponding polygen attribute given a pair of local scheme and attribute. This is needed to undo the pass one work in this case.

which corresponds to three local relations -- BUSINESS, CORPORATION, and FIRM. As such, these three local relations are retrieved (Row 4-6, Table 3), merged (Row 7, Table 3), and followed by a *Join* with R(2) of Table 2 -- which maps to R(3) of Table 3. Finally, the fourth and fifth row of Table 2 maps to the ninth and tenth row of Table 3.

In general, three possibilities exist for the *right-hand relation:* (1) a relation defined by the polygen schema (2) a R(#) denoting a polygen base relation or a polygen relation derived from other polygen base relations, and (3) non-existent (nil). The second and third cases follow the second case of pass one closely. The first case is also similar to pass one unless both the left- and right-hand sides require LQP operations. For example, the scenario presented in Section I has a join between PORGANIZATION and PALUMNUS, both requiring LQP operations first. That being the condition, separate LQP operations need to be performed first before the requested polygen operation is performed. The pass two algorithm is shown in Figure 4.

Finally, the *Query Optimizer* examines the *Intermediate Operation Matrix* and generates a query execution plan. Details of the *Query Optimizer* is also beyond the scope of this paper. Note also that the local database systems will most likely have their own high-level query languages, such as SQL, with their own optimization methods. As such, the algebraic expressions could be synthesized before sending to the corresponding local database systems.

IV. Example Source Tagging in the PQP

We now illustrate the processing of the example polygen query assuming the following local relations.

The Alumnus Relation (AD)

AID#	ANAME	DEG	MAJ
012	John McCauley	MBA	IS
123	Bob Swanson	MBA	MGT
234	Stu Madnick	MBA	IS
345	James Yao	BS	EECS
456	Dave Horton	MBA	IS
567	John Reed	MBA	MGT
678	Bob Horton	SF	MGT
789	Ken Olsen	MS	EE

The Career Relation (AD)

AID#	BNAME	POS
012	Citicorp	MIS Director
123	Genentech	CEO
234	Langley Castle	CEO
345	Oracle	Manager
456	Ford	Manager
567	Citicorp	CEO
678	BP	CEO
789	DEC	CEO
234	MIT	Professor

The Business Relation (AD)

BNAME	IND
Langley Castle	Hotel
IBM	High Tech
MIT	Education
CitiCorp	Banking
Oracle	High Tech
Ford	Automobile
DEC	High Tech
BP	Energy
Genentech	High Tech

The Student Relation (PD)

SID#	SNAME	GPA	MAJOR
01	Forea Wang	3.5	Math
12	Yeuk Yuan	3.9	EECS
23	Rich Bolsky	3.2	Finance
34	John Smith	3.9	Finance
45	Mike Lavine	3.7	IS

The Interview Relation (PD)

SID#	CNAME	JOB
01	IBM	System Analyst
12	Oracle	Product Manager
23	Banker's Trust	CFO
34	Citicorp	Far East Manager

The Corporation Relation (PD)

CNAME	TRADE	STATE
Apple	High Tech	CA
Oracle	High Tech	CA
AT&T	High Tech	NY
IBM	High Tech	NY
Citicorp	Banking	NY
DEC	High Tech	MA
Banker's Trust	Finance	NY

The Firm Relation (CD)

FNAME	CEO	HQ
AT&T	Robert Allen	NY, NY
Langley Castle	Stu Madnick	Cambridge, MA
Banker's Trust	Charles Sanford	NY, NY
CitiCorp	John Reed	NY, NY
Ford	Donald Peterson	Dearborn, MI
IBM	John Ackers	Armonk, NY
Apple	John Sculley	Cupertino, CA
Oracle	Lawrence Ellison	Belmont, CA
DEC	Ken Olsen	Maynard, MA
Genentech	Bob Swanson	So. San Francisco, CA

The Finance Relation (CD)

FNAME	YR	PROFIT
AT&T	1989	-1.7 bil
Langley Castle	1989	1 mil
Banker's Trust	1989	648 mil
CitiCorp	1989	1.7 bil
Ford	1989	5.3 bil
IBM	1989	5.5 bil
Apple	1989	400 mil
Oracle	1989	43 mil
DEC	1989	1.3 bil
Genentech	1989	21 mil

Let us assume that Table 3 is used as a query execution plan (i.e., without further optimization). The first row of Table 3 indicates that the operation ALUMNUS[DEG = "MBA"] should be executed by the Alumni Database LQP and the result is shown in Table 4. Note that the data source cell is the set {AD} which is taken directly from the EL cell of the first row, Table 3. The intermediate source is an empty set.

Table 4: Result of the Operation of Row 1, Table 3

AID#	ANAME	DEG	MAJ
012, {AD}, {}	John McCauley, {AD}, {}	MBA, {AD}, {}	IS, {AD}, {}
123, {AD}, {}	Bob Swanson, {AD}, {}	MBA, {AD}, {}	MGT {AD}, {}
234, {AD}, {}	Stu Madnick, {AD}, {}	MBA, {AD}, {}	IS, {AD}, {}
456, {AD}, {}	Dave Horton, {AD}, {}	MBA, {AD}, {}	IS, {AD}, {}
567, {AD}, {}	John Reed, {AD}, {}	MBA, {AD}, {}	MIT, {AD}, {}

The second row of Table 3 indicates that the CAREER relation should be retrieved from the Alumni Database and joined (Row 3, Table 3) with Table 4. The result is shown in Table 5. The *Join* requires that the intermediate source cells to be {AD} although in this case it appears to be redundant.

Table 5: Result of the Operations of Row 2 and 3, Table 3

AID#	ANAME	DEG	MAJ	BNAME	POS
012, {AD}, {AD}	John McCauley, {AD}, {AD}	MBA, {AD}, {AD}	IS, {AD}, {AD}	Citicorp, {AD}, {AD}	MIS Director, {AD}, {AD}
123, {AD}, {AD}	Bob Swanson, {AD}, {AD}	MBA, {AD}, {AD}	MGT, {AD}, {AD}	Genentech, {AD}, {AD}	CEO, {AD}, {AD}
234, {AD}, {AD}	Stu Madnick, {AD}, {AD}	MBA, {AD}, {AD}	IS, {AD}, {AD}	Langley Castle, {AD}, {AD}	CEO, {AD}, {AD}
456, {AD}, {AD}	Dave Horton, {AD}, {AD}	MBA, {AD}, {AD}	IS, {AD}, {AD}	Ford, {AD}, {AD}	Manager, {AD}, {AD}
567, {AD}, {AD}	John Reed, {AD}, {AD}	MBA, {AD}, {AD}	MIT, {AD}, {AD}	Citicorp, {AD}, {AD}	CEO, {AD}, {AD}
234, {AD}, {AD}	Stu Madnick, {AD}, {AD}	MBA, {AD}, {AD}	IS, {AD}, {AD}	MIT, {AD}, {AD}	Professor, {AD}, {AD}

Next the BUSINESS, CORPORATION, and FIRM relations are retrieved from the Alumni Database, the Placement Database, and the Company Database respectively, then merged in the PQP. The result is shown in Table 6. The *Outer Natural Primary Join*, *Outer Natural Total Join*, and *Coalesce* operations for generating Table 6 is shown in the Appendix A.

Table 6: Result of the Operation of Row 4 through 7, Table 3

ONAME	INDUSTRY	HEADQUARTERS	CEO
Langley Castle, {AD, CD},{AD, CD}	Hotel, {AD},{AD, CD}	MA, {CD},{AD, CD}	Stu Madnick, {CD},{AD, CD}
IBM, {AD, PD, CD}, {AD, PD, CD}	High Tech, {AD, PD},{AD, PD, CD}	NY, {PD, CD},{AD, PD, CD}	John Ackers, {CD},{AD, PD, CD}
MIT, {AD},{AD}	Education, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
CitiCorp, {AD, PD, CD},{AD, PD, CD}	Banking, {AD, PD},{AD, PD, CD}	NY, {PD, CD},AD, PD, CD}	John Reed, {CD},{AD, PD, CD}
Oracle, {AD, PD, CD},{AD, PD, CD}	High Tech, {AD, PD},{AD, PD, CD}	CA, {PD, CD},{AD, PD, CD}	Lawrence Ellison, {CD},{AD, PD, CD}
Ford, {AD, CD},{AD, CD}	Automobile, {AD},{AD, CD}	MI, {CD},{AD, CD}	Donald Peterson, {CD},{AD, CD}
DEC, {AD, PD, CD}, {AD, PD, CD}	High Tech, {AD, PD}, {AD, PD, CD}	MA, {PD, CD},{AD, PD, CD}	Ken Olsen, {CD}, {AD, PD, CD}
BP, {AD},{AD}	Energy, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
Genentech, {AD, CD}, {AD, CD}	High Tech, {AD}, {AD, CD}	CA, {CD}, {AD, CD}	Bob Swanson, {CD}, {AD, CD}
Apple, {PD, CD},{PD, CD}	High Tech, {PD},{PD, CD}	CA, {PD, CD,{PD, CD}	John Sculley, {CD},{PD, CD}
AT&T, {PD, CD},{PD, CD}	High Tech, {PD}, {PD, CD}	NY, {PD, CD},{PD, CD}	Robert Allen, {CD},{PD, CD}
Banker's Trust, {PD, CD}, {PD, CD}	Finance, {PD}, {PD, CD}	NY, {PD, CD},{PD, CD}	Charles Sanford, {CD}, {PD, CD}

The PQP now joins Table 5 with Table 6 and produces Table 7

Table 7: Result of the Operation of Row 8, Table 3

AID#	ANAME	DEG	MAJ	ONAME	POS	INDUSTRY	Headquarters	CEO
012, {AD}, {AD, PD, CD}	John McCauley, {AD}, {AD, PD, CD}	MBA, {AD}, {AD, PD, CD}	IS, {AD}, {AD, PD, CD}	Citicorp, {AD, PD, CD}, {AD, PD, CD}	MIS Director, {AD}, {AD, PD, CD}	Banking, {AD, PD} {AD, PD, CD}	NY, {PD, CD}, {AD, PD, CD}	John Reed, {CD},{AD, PD, CD}
123, {AD}, {AD, CD}	Bob Swanson, {AD}, {AD, CD}	MBA, {AD}, {AD, CD}	MGT, {AD}, {AD, CD}	Genentech, {AD, CD}, {AD, CD}	CEO, {AD}, {AD, CD}	High Tech, {AD}, {AD, CD}	CA, {CD}, {AD, CD}	Bob Swanson, {CD}, {AD, CD}
234, {AD}, {AD, CD}	Stu Madnick, {AD}, {AD, CD}	MBA, {AD}, {AD, CD}	IS, {AD}, {AD, CD}	Langley Castle, {AD, CD}, {AD, CD}	CEO, {AD}, {AD, CD}	Hotel, {AD}, {AD, CD}	MA, {CD}, {AD, CD}	Stu Madnick, {CD}, {AD, CD}
456, {AD}, {AD, CD}	Dave Horton, {AD}, {AD, CD}	MBA, {AD}, {AD, CD}	IS, {AD}, {AD, CD}	Ford, {AD, CD}, {AD, CD}	Manager, {AD}, {AD, CD}	Automobile, {AD}, {AD, CD}	MI, {CD}, {AD, CD}	Don Peterson, {CD}, {AD, CD}
567, {AD}, {AD, PD, CD}	John Reed, {AD}, {AD, PD, CD}	MBA, {AD}, {AD, PD, CD}	MIT, AD}, {AD, PD, CD}	Citicorp, {AD, PD, CD}, {AD, PD, CD}	CEO, {AD}, {AD, PD, CD}	Banking, {AD, PD},{AD, PD, CD}	NY, {PD, CD}, {AD, PD, CD}	John Reed, {CD},{AD, PD, CD}
234, {AD}, {AD}	Stu Madnick, {AD}, {AD}	MBA, {AD}, {AD}	IS, {AD}, {AD}	MIT, {AD}, {AD}	Professor, {AD}, {AD}	Education, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}

Table 7 is restricted to produce Table 8 .

Table 8: Result of the Operation of Row 9, Table 3

AID#	ANAME	DEG	MAJ	ONAME	POS	Industry	Headquarters	CEO
123, {AD}, {AD, CD}	Bob Swanson, {AD}, {AD, CD}	MBA, {AD}, {AD, CD}	MGT, {AD}, {AD, CD}	Genentech, {AD, CD}, {AD, CD}	CEO, {AD}, {AD, CD}	High Tech, {AD}, {AD, CD}	CA, {CD}, {AD, CD}	Bob Swanson, {CD}, {AD, CD}
234, {AD}, {AD, CD}	Stu Madnick, {AD}, {AD, CD}	MBA, {AD}, {AD, CD}	IS, {AD}, {AD, CD}	Langley Castle, {AD, CD}, {AD, CD}	CEO, {AD}, {AD, CD}	Hotel, {AD}, {AD, CD}	MA, {CD}, {AD, CD}	Stu Madnick, {CD}, {AD, CD}
567, {AD}, {AD, PD, CD}	John Reed, {AD}, {AD, PD, CD}	MBA,{AD}, {AD, PD, CD}	MIT,{AD}, {AD, PD, CD}	Citicorp, {AD, PD, CD}, {AD, PD, CD}	CEO, {AD}, {AD, PD, CD}	Banking, {AD, PD}, {AD, PD, CD}	NY, {PD, CD}, {AD, PD, CD}	John Reed, {CD}, {AD, PD, CD}

Finally, Table 8 is projected to form Table 9 which contains only those organizations and their CEOs who graduated from MIT's Sloan School of Management with an MBA degree.

Table 9: Result of the Operation of Row 10, Table 3

ONAME	CEO
Genentech, {AD, CD}, {AD, CD}	Bob Swanson, {CD},{AD, CD}
Langley Castle, {AD, CD}, {AD, CD}	Stu Madnick, {CD},{AD, CD}
Citicorp, {AD, PD, CD}, {AD, PD, CD}	John Reed, {CD},{AD, PD, CD}

Several observations can be made based on the source tagging information:

(1) The information of Genentech is from the Alumni Database and Company Database, and only from these two databases. On the other hand, the information that Genentech's CEO is Bob Swanson came from the Company Database, and the Alumni Database has served as an intermediate source in obtaining the information.

(2) The information about Citicorp is available from all three databases, but the information about its CEO, John Reed, is available only in the Company Database.

(3) From the polygen schema and the information of (ONAME, {AD, CD}), the polygen query processor can derive the information that Genentech is from the BNAME column, BUSINESS relation in the Alumni Database and from the FNAME column, FIRM relation in the Company Database. This information can be shown to the user upon request with a simple mapping.

In a federated database environment with hundreds of databases, the data source and intermediate source information can be very valuable to the user as well as the polygen query processor in formulating cost-effective, customized, and credible composite information.

V. Concluding Remarks

We have presented a polygen model for resolving the *Data Source Tagging* and *Intermediate Source Tagging* problems. The *polygen model* research addresses issues in heterogeneous distributed database systems from the *"where "* perspective - a perspective that, to the best of our knowledge, has not been studied to date. Furthermore, we have presented a data-driven query translation mechanism for mapping a polygen algebraic expression into a set of intermediate polygen operations dynamically. A Prototype, called System P, is currently being developed [86] to realize the polygen model and the polygen query processing capability presented in this paper.

This research has provided us with a theoretical foundation for further investigation of many other critical research issues in heterogeneous distributed systems, for example the cardinality inconsistency problem which is inherent in heterogeneous database systems.[12] It also enable us to interpret information from different sources more accurately. By storing the metadata about each of the data sources in the PQP, many domain mismatch, semantic reconciliation, and data conflict problems can be resolved systematically using the data and intermediate source tags.

Furthermore, this research serves as a departure point for developing polygen models for heterogeneous distributed database systems based on the Entity Relationship Model, the Functional Data Model, and the more recent object-oriented models [71]. We believe that further research in this important area will not only contribute to the academic discipline but also benefit the business community in the foreseeable future.

Appendix A: The Operations that Generate Table 6.

The 4th, 5th, and 6th row of Table 3 indicates that the BUSINESS, CORPORATION, and FIRM relations should be retrieved from the Alumni Database, the Placement Database, and the Company Database respectively. As such, the corresponding data source cells are the set {AD}, {PD}, and {CD} respectively, as shown in Table A1, A2, and A3 below. The intermediate source is an empty set because no other data sources have been involved in obtaining these relations.

Table A1: The Business Relation

BNAME	IND
Langley Castle, {AD},{}	Hotel, {AD},{}
IBM, {AD},{}	High Tech, {AD},{}
MIT, {AD},{}	Education, {AD},{}
CitiCorp, {AD},{}	Banking, {AD},{}
Oracle, {AD},{}	High Tech, {AD},{}
Ford, {AD},{}	Automobile, {AD},{}
DEC, {AD},{}	High Tech, {AD},{}
BP, {AD},{}	Energy, {AD},{}
Genentech, {AD},{}	High Tech, {AD},{}

Table A2: The Corporation Relation

CNAME	TRADE	STATE
Apple, {PD},{}	High Tech, {PD},{}	CA, {PD},{}
Oracle, {PD}, {}	High Tech, {PD},{}	CA, {PD},{}
AT&T, {PD},{}	High Tech, {PD},{}	NY, {PD},{}
IBM, {PD}, {}	High Tech, {PD},{}	NY, {PD},{}
Citicorp, {PD},{}	Banking, {PD},{}	NY, {PD},{}
DEC, {PD},{}	High Tech, {PD},{}	MA, {PD},{}
Banker's Trust, {PD},{}	Finance, {PD},{}	NY, {PD},{}

Table A3: The Firm Relation

FNAME	CEO	HQ
AT&T, {CD},{}	Robert Allen, {CD},{}	NY, {CD},{}
Langley Castle, {CD},{}	Stu Madnick, {CD},{}	MA, {CD},{}
Banker's Trust, {CD},{}	Charles Sanford, {CD},{}	NY, {CD},{}
CitiCorp, {CD},{}	John Reed, {CD},{}	NY, {CD},{}
Ford, {CD},{}	Donald Peterson, {CD},{}	MI, {CD},{}
IBM, {CD},{}	John Ackers, {CD},{}	NY, {CD},{}
Apple, {CD},{}	John Sculley, {CD},{}	CA, {CD},{}
Oracle, {CD},{}	Lawrence Ellison, {CD},{}	CA, {CD},{}
DEC, {CD},{}	Ken Olsen, {CD},{}	MA, {CD},{}
Genentech, {CD},{}	Bob Swanson, {CD},{}	CA, {CD},{}

[12] Under the relational assumption, the cardinality inconsistency problem exists in heterogeneous database systems because the referential integrity is not enforceable over multiple pre-existing databases which have been developed and administered independently and are likely to remain so.

Table A4: The Outer join of Table A1 and Table A2

BNAME	IND	CNAME	TRADE	STATE
Langley Castle, {AD},{AD}	Hotel, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}	nil, {}, {AD}
IBM, {AD},{AD, PD}	High Tech, {AD},{AD, PD}	IBM, {PD},{AD, PD}	High Tech, {PD},{AD, PD}	NY, {PD},{AD, PD}
MIT, {AD},{AD}	Education,{AD},{AD}	nil, {}, {AD}	nil, {}, {AD}	nil, {}, {AD}
CitiCorp, {AD},{AD, PD}	Banking, {AD},{AD, PD}	Citicorp, {PD},{AD, PD}	Banking, {PD},{AD, PD}	NY, {PD},{AD, PD}
Oracle, {AD},{AD, PD}	High Tech, {AD},{AD, PD}	Oracle, {PD},{AD, PD}	High Tech, {PD},{AD, PD}	CA, {PD},{AD, PD}
Ford, {AD},{AD}	Auto, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}	nil, {}, {AD}
DEC, {AD},{AD, PD}	High Tech, {AD},{AD, PD}	DEC, {PD},{AD, PD}	High Tech, {PD},{AD, PD}	MA, {PD},{AD, PD}
BP, {AD},{AD}	Energy, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}	nil, {}, {AD}
Genentech, {AD},{AD}	High Tech, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}	nil, {}, {AD}
nil, {}, {PD}	nil, {}, {PD}	Apple, {PD},{PD}	High Tech, {PD},{PD}	CA, {PD},{PD}
nil, {}, {PD}	nil, {}, {PD}	AT&T, {PD},{PD}	High Tech, {PD},{PD}	NY, {PD},{PD}
nil, {}, {PD}	nil, {}, {PD}	Banker's Trust, {PD},{PD}	Finance, {PD},{PD}	NY, {PD},{}

(2) A *Coalesce* of the BNAME and CNAME columns into the ONAME column. The result is shown in Table A5.

As we defined in Section II, step one and two together are called an *Outer Natural Primary Join*.

Table A5: The Outer Natural Primary Join of Table A1 and Table A2

ONAME	IND	TRADE	STATE
Langley Castle, {AD},{AD}	Hotel, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
IBM, {AD, PD},{AD, PD}	High Tech, {AD},{AD, PD}	High Tech, {PD},{AD, PD}	NY, {PD},{AD, PD}
MIT, {AD},{AD}	Education, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
CitiCorp, {AD, PD},{AD, PD}	Banking, {AD},{AD, PD}	Banking, {PD},{AD, PD}	NY, {PD},{AD, PD}
Oracle, {AD, PD},{AD, PD}	High Tech, {AD},{AD, PD}	High Tech, {PD},{AD, PD}	CA, {PD},{AD, PD}
Ford, {AD},{AD}	Automobile, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
DEC, {AD, PD},{AD, PD}	High Tech, {AD},{AD, PD}	High Tech, {PD},{AD, PD}	MA, {PD},{AD, PD}
BP, {AD},{AD}	Energy, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
Genentech, {AD},{AD}	High Tech, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
Apple, {PD},{PD}	nil, {}, {PD}	High Tech, {PD},{PD}	CA, {PD},{PD}
AT&T, {PD},{PD}	nil, {}, {PD}	High Tech, {PD},{PD}	NY, {PD},{PD}
Banker's Trust, {PD},{PD}	nil, {}, {PD}	Finance, {PD},{PD}	NY, {PD},{PD}

(3) A *Coalesce* of the IND and TRADE columns into the INDUSTRY column, and a mapping of the local attribute STATE into the polygen attribute HEADQUARTERS. The result is shown in Table A6.

Table A6: The Outer Natural Total Join of Table A1 and Table A2

ONAME	INDUSTRY	HEADQUARTERS
Langley Castle, {AD},{AD}	Hotel, {AD},{AD}	nil, {}, {AD}
IBM, {AD, PD},{AD, PD}	High Tech, {AD, PD},{AD, PD}	NY, {PD},{AD, PD}
MIT, {AD},{AD}	Education, {AD},{AD}	nil,{}, {AD}
CitiCorp, {AD, PD},{AD, PD}	Banking, {AD, PD},{AD, PD}	NY, {PD},{AD, PD}
Oracle, {AD, PD},{AD, PD}	High Tech, {AD, PD},{AD, PD}	CA,{PD},{AD, PD}
Ford, {AD},{AD}	Automobile, {AD},{AD}	nil, {}, {AD}
DEC,{AD, PD},{AD, PD}	High Tech, {AD, PD},{AD, PD}	MA, {PD},{AD, PD}
BP, {AD},{AD}	Energy,{AD},{AD}	nil, {}, {AD}
Genentech, {AD},{AD}	High Tech, {AD},{AD}	nil, {}, {AD}
Apple, {PD},{PD}	High Tech, {PD},{PD}	CA, {PD},{PD}
AT&T, {PD},{PD}	High Tech, {PD},{PD}	NY, {PD},{PD}
Banker's Trust, {PD},{PD}	Finance, {PD},{PD}	NY, {PD},{PD}

The *Outer Natural Total Join* of Table A6 and Table A3 is shown in Table A7, A8, and A9 following the same procedure. The resulting Table A9 is shown as Table 6 in the body of the paper.

Table A7: The Outer Join of Table A6 and Table A3

ONAME	INDUSTRY	HEADQUARTERS	FNAME	CEO	HQ
Langley Castle, {AD},{AD}	Hotel, {AD},{AD}	nil, {}, {AD}	Langley Castle, {CD},{}	Stu Madnick, {CD},{}	MA, {CD},{}
IBM, {AD, PD},{AD, PD}	High Tech, {AD, PD},{AD, PD}	NY, {PD},{AD, PD}	IBM, {CD},{}	John Ackers, {CD},{}	NY, {CD},{}
MIT, {AD},{AD}	Education, {AD},{AD}	nil, {}, {AD}	nil, { }, {AD}	nil, { }, {AD}	nil, { }, {AD}
CitiCorp, {AD, PD},{AD, PD}	Banking, {AD, PD},{AD, PD}	NY, {PD},{AD, PD}	CitiCorp, {CD},{}	John Reed, {CD},{}	NY, {CD},{}
Oracle, {AD, PD},{AD, PD}	High Tech, {AD, PD},{AD, PD}	CA, {PD},{AD, PD}	Oracle, {CD},{}	Lawrence Ellison, {CD},{}	CA, {CD},{}
Ford, {AD},{AD}	Automobile, {AD},{AD}	nil, {}, {AD}	Ford, {CD},{}	Donald Peterson, {CD},{}	MI, {CD},{}
DEC, {AD, PD},{AD, PD}	High Tech, {AD, PD},{AD, PD}	MA, {PD},{AD, PD}	DEC, {CD},{}	Ken Olsen, {CD},{}	MA, {CD},{}
BP, {AD},{AD}	Energy, {AD},{AD}	nil, {}, {AD}	nil, { }, {AD}	nil, { }, {AD}	nil, { }, {AD}
Genentech, {AD},{AD}	High Tech, {AD},{AD}	nil, {}, {AD}	Genentech, {CD},{}	Bob Swanson, {CD},{}	CA, {CD},{}
Apple, {PD},{PD}	High Tech, {PD},{PD}	CA, {PD},{PD}	Apple, {CD},{}	John Sculley, {CD},{}	CA, {CD},{}
AT&T, {PD},{PD}	High Tech, {PD},{PD}	NY, {PD},{PD}	AT&T, {CD},{}	Robert Allen, {CD},{}	NY, {CD},{}
Banker's Trust, {PD},{PD}	Finance, {PD},{PD}	NY, {PD},{PD}	Banker's Trust, {CD},{}	Charles Sanford, {CD},{}	NY, {CD},{}

Table A8: The Outer Natural Primary Join of Table A6 and Table A3

ONAME	INDUSTRY	HEADQUARTERS	CEO	HQ
Langley Castle, {AD, CD},{AD, CD}	Hotel, {AD},{AD, CD}	nil, {}, {AD, CD}	Stu Madnick, {CD},{AD, CD}	MA, {CD},{AD, CD}
IBM, {AD, PD, CD},{AD, PD,CD}	High Tech, {AD, PD},{AD, PD, CD}	NY, {PD},{AD, PD, CD}	John Ackers, {CD},{AD, PD, CD}	NY, {CD},{AD, PD, CD}
MIT, {AD},{AD}	Education, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}	nil, {}, {AD}
CitiCorp, {AD, PD, CD},{AD, PD}	Banking, {AD, PD},{AD, PD, CD}	NY, {PD},{AD, PD, CD}	John Reed, {CD},{AD, PD, CD}	NY, {CD},{AD, PD, CD}
Oracle, {AD, PD, CD},{AD, PD, CD}	High Tech, {AD, PD},{AD, PD, CD}	CA, {PD},{AD, PD, CD}	Lawrence Ellison, {CD},{AD, PD, CD}	CA, {CD},{AD, PD, CD}
Ford, {AD, CD},{AD, CD}	Automobile, {AD},{AD, CD}	nil, {}, {AD, CD}	Donald Peterson, {CD},{AD, CD}	MI, {CD},{AD, CD}
DEC, {AD, PD, CD},{AD, PD, CD}	High Tech, {AD, PD},{AD, PD, CD}	MA, {PD},{AD, PD, CD}	Ken Olsen, {CD},{AD, PD, CD}	MA, {CD},{AD, PD, CD}
BP, {AD},{AD}	Energy, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}	nil, {}, {AD}
Genentech, {AD, CD},{AD, CD}	High Tech, {AD},{AD, CD}	nil, {}, {AD, CD}	Bob Swanson, {CD},{AD, CD}	CA, {CD},{AD, CD}
Apple, {PD, CD},{PD, CD}	High Tech, {PD},{PD, CD}	CA, {PD},{PD, CD}	John Sculley, {CD},{PD, CD}	CA, {CD},{PD, CD}
AT&T, {PD, CD},{PD, CD}	High Tech, {PD},{PD, CD}	NY, {PD},{PD, CD}	Robert Allen, {CD},{PD, CD}	NY, {CD},{PD, CD}
Banker's Trust, {PD, CD},{PD, CD}	Finance, {PD},{PD, CD}	NY, {PD},{PD, CD}	Charles Sanford, {CD},{PD, CD}	NY, {CD},{PD, CD}

Table A9: The Outer Natural Total Join of Table A6 and Table A3

ONAME	INDUSTRY	HEADQUARTERS	CEO
Langley Castle, {AD, CD},{AD, CD}	Hotel, {AD},{AD, CD}	MA, {CD},{AD, CD}	Stu Madnick, {CD},{AD, CD}
IBM, {AD, PD, CD},{AD, PD, CD}	High Tech, {AD, PD},{AD, PD, CD}	NY, {PD, CD},{AD, PD, CD}	John Ackers, {CD},{AD, PD, CD}
MIT, {AD},{AD}	Education, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
CitiCorp, {AD, PD, CD},{AD, PD, CD}	Banking, {AD, PD},{AD, PD, CD}	NY, {PD, CD},{AD, PD, CD}	John Reed, {CD},{AD, PD, CD}
Oracle, {AD, PD, CD},{AD, PD, CD}	High Tech, {AD, PD},{AD, PD, CD}	CA, {PD, CD},{AD, PD, CD}	Lawrence Ellison, {CD},{AD, PD, CD}
Ford, {AD, CD},{AD, CD}	Automobile, {AD},{AD, CD}	MI, {CD},{AD, CD}	Donald Peterson, {CD},{AD, CD}
DEC, {AD, PD, CD},{AD, PD, CD}	High Tech, {AD, PD},{AD, PD, CD}	MA, {PD, CD},{AD, PD, CD}	Ken Olsen, {CD},{AD, PD, CD}
BP, {AD},{AD}	Energy, {AD},{AD}	nil, {}, {AD}	nil, {}, {AD}
Genentech, {AD, CD},{AD, CD}	High Tech, {AD},{AD, CD}	CA, {CD},{AD, CD}	Bob Swanson, {CD},{AD, CD}
Apple, {PD, CD},{PD, CD}	High Tech, {PD},{PD, CD}	CA, {PD, CD,{PD, CD}	John Sculley, {CD},{PD, CD}
AT&T, {PD, CD},{PD, CD}	High Tech, {PD},{PD, CD}	NY, {PD, CD},{PD, CD}	Robert Allen, {CD},{PD, CD}
Banker's Trust, {PD, CD},{PD, CD}	Finance, {PD},{PD, CD}	NY, {PD, CD},{PD, CD}	Charles Sanford, {CD},{PD, CD}

Bibliography

1. Abiteboul, S. & Hull, R. IFO : a formal semantic database model. *ACM Transactions on Database System 12*, 4 (1987), pp. 525-565.
2. Al-Fedaghi, S. & Scheuermann, P. Mapping considerations in the design of schemas for the relational model. *IEEE Transactions on Software Engineering SE-7*, 1 (January).
3. Atzeni, P. & Chen, P. P. Completeness of query languages for the entity-relationship model. In P. P. Chen ed., *Information Modeling and Analysis*, ER Institute, 1981, pp. 111-124.
4. Barrett, S. Strategic alternatives and inter-organizational systems implementations: an overview. *Journal of Management Information System. 3*, 3 (Winter 1986-87), pp. 3-16.
5. Batini, C., Lenzirini, M., & Navathe, S. B. A comparative analysis of methodologies for database schema integration. *ACM Computing Survey. 18*, 4 (December 1986), pp. 323 - 364.
6. Breitbart, Y., Olson, P. L., & Thompson, G. R. Database integration in a distributed heterogeneous database system. In *Proceedings of the 2nd International Conference on Data Engineering*, (Los Angeles, CA, February, 1986), pp. 301-310.
7. Brill, D., Templeton, M. & Yu, C. Distributed query processing strategies in MERMAID, a frontend to data management systems. In *Proceedings First International Conference on Data Engineering*, (Los Angeles, CA, 1984), pp. 310-310.
8. Brodie, M. & Mylopoulos, J. ed., *On Knowledge Base Management Systems*, Springer-Verlag, 1986.
9. Casanova, M. & Vidal, M. Towards a sound view integration methodology. In *Proceedings of the 2nd ACM SIGACT/SIGMOD Conference on Principles of Database Systems* (Atlanta, GA, Mar. 21-23). ACM, New York, pp. 36-47.
10. Cash, J. I. & Konsynski, B. R. IS redraws competitive boundaries. *Harvard Business Review*, (March-April 1985), pp. 134-142.
11. Ceri, S. & Pelagatti, G. *Distributed Databases Principles & Systems*, McGraw-Hill, 1984.
12. Chen, P. P. *A Preliminary Framework for Entity-Eelationship Model*, in P.P. Chen, ed., *Information Modeling and Analysis*, ER Institute, (1981), pp. 19-28.
13. Chen, P. P. An algebra for a directional binary entity-relationship model. In *Proceedings of the 1st IEEE International Conference on Data Engineering*, (Los Angeles, CA, 1984), pp. 37-40.
14. Chen, P. P. The entity-relationship model - toward a unified view of data. *Transactions on Data Base Systems. 1*, 1 (1976), pp. 166-193.
15. Clemons, E. K. & McFarlan, F. W. Telecom: hook up or lose out. *Harvard Business Review*, (July-August, 1986).
16. Codd, E. F. A relational model of data for large shared data banks. *Communications of the ACM. 13*, 6 (1970), pp. 377-387.
17. Codd, E. F. An evaluation scheme for database management systems that are claimed to be relational. Keynote address. *Proceedings of the International Conference of Data Engineering*, (1985), pp. 720-729.
18. Codd, E. F. Extending the database Relational model to capture more meaning. *ACM Transactions on Database Systems, 4* (4), (1979), pp. 397-434.
19. Codd, E. F. Relational completeness of data base sublanguages. R. Rustin, ed., *Data Base Systems*, Prentice-Hall, (1972), pp. 65-96.
20. Codd, E. F. Relational database: A practical foundation for productivity. In *Introduction to AI and Databases*, (1982), pp. 60-68.
21. Czejdo, B., Rusinkiewicz, M., & Embley, D. An approach to schema integration and query formulation in federated database systems. In *Proceedings of the 3rd International Conference on Data Engineering*, (Los Angeles, CA, February, 1987), pp. 477-484.
22. Date, C. J. The outer join. In *Proceedings of the 2nd International Conference on Databases* (Cambridge, England, September, 1983), pp. 76-106.
23. Dayal, U. & Hwang, K. View definition and generalization for database integration in multidatabase system. *IEEE Transactions on Software Engineering, SE-10*, 6 (November 1984), pp. 628-644.
24. Dayal, U. Processing queries over generalization hierarchies in a multidatabase system. In *Proceedings of the 9th International Conference, VLDB*. (Computer Corporation of America, 4 Cambridge Center, Cambridge, MA 02142), pp. 342-353.
25. Dayal, U., Hwang, H., Manola, F., Rosenthal, A. S., & Smith, J. M. Knowledge-oriented database management: final technical report, phase I. *Computer Corporation of America*, (Cambridge, MA., 1984).
26. Deen, S. M., Amin, R. R., & Taylor M. C. Data integration in distributed databases. *IEEE Transactions on Software Engineering, SE-13*, 7 (July 1987), pp. 860-864.
27. Deen, S. M., Amin, R. R., & Taylor, M. C. Implementation of a prototype for PRECI*. *Computer Journal, 30*, 2 (1987b), pp. 157-162.
28. DeMichiel, L. G. Performing operations over mismatched domains. In *Proceedings of the Fifth International Conference on Data Engineering* (Los Angeles, CA., February 1989).
29. Devor, C. C., & Weeldreyer, J. A. DDTS: A testbed for distributed database research. In *Proceedings of the ACM Pacific Conference (1980), pp. 86-94.*
30. Devor, C., Elmasri, R., & Rahimi, S. The design of DDTS: A testbed for reliable distributed database management. In *Proceedings of the 2nd IEEE Symposium on Reliability in Distributed Systems*, (Pittsburgh. PA, July, 1982), pp. 150-162.

31. dos Santos, C. S., Neuhold, E. J., & Furtado, A. L. A Data Type Approach to the Entity-Relationship Model. In P. P. Chen, ed., *Entity-Relationship Approach to Systems Analysis and Design*, North-Holland Publishing Co., 1980, pp. 103-119.
32. Elmasri R., Larson J., & Navathe, S. Schema integration algorithms for federated databases and logical database design. Submitted for Publication (1987).
33. Elmasri, R. & Wiederhold, G. GORDAS: A Formal High-Level Query Language for the Entity-Relationship Model. In P. P. Chen, ed., *Information Modeling and Analysis*, ER Institute, 1981, pp. 49-72.
34. Embley, D. Programming with data frames for everyday data items. 1980. National Computer Conference, 1980, pp.301-305.
35. Estrin, D. Inter-organizational networks: stringing wires across administrative boundaries. *Computer Networks and ISDN Systems 9*. North-Holland (1985).
36. Ferrier, A. & Strangret, C. Heterogenity in the distributed database management systems Sirius-Delta. In *Proceedings of the 8th VLDB International Conference*, (Mexico City, Mexico, September, 1982).
37. Frank, W., Madnick, S., & Wang, Y. R. A conceptual model for integrated autonomous processing: an international bank's experience with large databases. In *Proceedings of the 8th International Conference on Information Systems* (ICIS, December, 1987).
38. Godes, D. B. Use of heterogeneous data sources: three case studies. *WP # CIS-89-02*. (Sloan School of Management, MIT, Cambridge, MA., 1989). CISL Project.
39. Goldhirsch, D., Landers, T., Rosenberg, R., & Yedwab, L. MULTIBASE: System administrator's guide. *Computer Corporation of America*. (November 1984).
40. Gupta, A., Madnick, S., Poulsen, C., & Wingfield, T. An architecture comparison of contemporary approaches and products for integrating heterogeneous information systems. *IFSRC # 110-89*. (Sloan School of Management, MIT, Cambridge, MA 02139).
41. Heimbigner, D. & McLeod, D. A Federated architecture for information management. *ACM Transactions on Office Information Systems, 3*, 3 (1985), pp. 253-278.
42. Hull, R. & King, R. Semantic database modeling: survey, applications, and research issues. *ACM Computing Surveys, 19*, 3 (September 1987), pp. 201-259
43. Hwang, H. & Dayal, U. Using the entity-relationship model for implementing multi-model database systems. In P. Chen, ed., *Entity Relationship Approach to Information Modeling and Analysis* (California, E. R. Institute). pp. 237-258.
44. Ives, B. & Learmonth, G. P. The information system as a competitive weapon. *Communications of the ACM, 27*, 12 (December 1984), pp. 1193-1201.
45. Katz, R. H. & Goodman, N. View processing in MULTIBASE, a heterogeneous database system. In P. P. Chen ed., Entity-Relationship Approach to Information Modeling and Analysis, ER Institute, 1981, pp. 259-279
46. Kerschberg, L., ed. Expert Database Systems, *Proceedings from the First International Workshop*. The Benjamin/Cummings Publishing Company 1986.
47. Klug, A. Equivalence of relational algebra and relational calculus query languages having aggregate functions. *Journal of the Association for Computing Machinery. 29*, 3 (July, 1982), pp. 699-717.
48. Lien, Y. E., Shopiro, J. E., & Tsur, S. DSIS - A database system with interrelational semantics. In *Proceedings of the 7th International VLDB Conference* (1981), pp. 465-477.
49. Litwin, W. & Abdellatif, A. Multidatabase interoperability. *IEEE Computer* (December 1986), pp. 10-18.
50. Litwin, W., Boudenant, J., Esculier, C., Ferrier, A., Glorieux, A. M., LaChimia, J., Kabbaj, K., Moulinoux, C., Rolin, P. & Strangret, C. SIRIUS system for distributed data management. In *Proceedings of International Symposium on Distributed Databases*, (Berlin, West Germany, 1982), pp. 311-366.
51. Lyngbaek, P. & McLeod, D. An approach to object sharing in distributed database systems. In *The Proceedings of the 9th International Conf. on VLDB* (October, 1983).
52. Madnick, S. & Wang, Y. R. Integrating disparate databases for composite answers. In *Proceedings of the 21st Annual Hawaii International Conference on System Sciences* (January, 1988).
53. Madnick, S., ed., *The Strategic Use of Information Technology*. Oxford University Press, 1987.
54. Manola, F. & Dayal, U. PDM: An object-oriented data model. In *Proceedings of the International Workshop on Object-Oriented Database Systems* (Pacific Grove, CA., September, 1986), pp. 18-25.
55. Manola, F. Applications of object-oriented database technology in knowledge-based integrated information systems. In paper prepared for the *CRAI School on Recent Techniques for Integrating Heterogeneous Databases*, (Venezia University, April 10-14, 1989).
56. Manola, F. Distributed object management technology. In *Technical Memorandum, GTE Laboratories, TM-0014-06-88-165* (1988).
57. Markowitz, V. M. & Raz, Y. A Modified relational algebra and its use in an entity-relationship environment. In C. G. Davis, S. Jajodia, P. A. Ng, & R. T. Yeh, eds., *Entity-relationship approach to software engineering*. Elsevier Science Publishers, 1983, pp. 315-328.
58. Markowitz, V. M. & Shoshani, A. Abbreviated query interpretation in extended entity-relationship oriented databases. In *Proceedings of the 8th International Conference* (Toronto, Canada, 1989), pp.

40-58.
59. Markowitz, V. M. and Shoshani, A. On the correctness of representing extended entity-relationship structures in the relational model. *Proceedings of the 1989 SIDMOD Conference, SIDMOD Record 18*, 2, (June 1989), pp. 430-439.
60. McLeod, D. J. High level domain definition in a relational data base system. *IBM Research Laboratory* (San Jose, CA., 1976).
61. Navathe, S. B., Sashidhar, T., & Elmasri, R. Relationship merging in schema integration. In *Proceedings of the 10th International Conference on Very Large Data Bases*, (Singapore, August, 1984).
62. Paget, M. L. A knowledge-based approach toward integrating international on-line databases. *WP # CIS-89-01* (Sloan School of Management, MIT, Cambridge, MA. 1989). CISL Project.
63. Parent, C. & Spaccapietra, S. An algebra for a general entity-relationship model. *IEEE Transactions on Software Engineering, SE-11*, 7 (1985), pp. 634-643.
64. Parent, C., Rolin, H., Yetongnon, K., & Spaccapietra, S. An ER calculus for the entity-relationship complex model. In *Proceedings of the 8th International Conference* (Toronto, Canada, 1989). pp. 75-98.
65. Peckham, J. & Maryanski, F. Semantic data models. *ACM Computing Surveys, 20*, 3 (1988), pp. 153-189.
66. Porter, M. & Millar, V. E., How information gives you competitive advantages. *Harvard Business Review* (July-August, 1985), pp. 149-160.
67. Qian, X. & Wiederhold, G. Knowledge-based integrity constraint validation. In *Proceedings of the Twelfth International Conference on Very Large Data Bases* (1986), pp. 3-12.
68. Rockart, J. The line takes the leadership: IS management in a wired society. *Sloan Management Review, MIT, 29*, 4 (Spring 1988), pp. 57-64.
69. Rusinkiewicz, M. & Czejdo, B. Query transformation in heterogeneous distributed database systems. In *Proceedings of the 5th International Conference on Distributed Computing*, (Denver, CO, 1985).
70. Rusinkiewicz, M., Emasri, R., Czejdo, B., Georgakopoulos, D., Karabatis, G., Jamoussi, A., Loa, K., Li, Y., Gilbert, J., & Musgrove, R. Query processing in omnibase - a loosely coupled multi-database system. University of Houston *Technical Report #UH-CS-88-05*. (1988).
71. Shaw, G. & Zdonik, S. B. A query algebra for object-oriented databases. In *Proceedings of the 6th International Conference on Data Engineering* (February, 1990).
72. Shaw, G. & Zdonik, S. B. Object-oriented queries: equivalence & optimization . In *Proceedings of the 1st International Conference on Deductive & Object-Oriented Databases* , (Kyoto, Japan, December, 1989).
73. Shin, D. G. Semantics for handling queries with missing information. In *Proceedings of the Ninth International Conference on Information Systems 9* (1988), pp. 161 - 167.
74. Shipman, D. W. Functional data model and the data language DAPLEX. *ACM Transactions on Database Systems,6*, 1 (March, 1981), pp. 140-173.
75. Smith, J. M., Bernstein, P. A., Dayal, U., Goodman, N., Landers, T., Lin, K. W. T., & Wong, E. *Multibase - Integrating Heterogeneous Distributed Database Systems*. 1981 National Computer Conference, 1981, pp. 487-499.
76. Stonebraker, M. Inclusion of new types in relational data base systems. *IEEE*, (1986), pp. 480-487.
77. Templeton, M., Brill, D., Hwang, A., Kameny, I., & Lund, E. An Overview of the MERMAID system - a frontend to heterogeneous databases. In *Proceedings IEEE EASCON*, (Washington, DC, September, 1983).
78. Teorey, T. J., Yang, D., & Fry, J. P. A logical design methodology for relational databases using the extended entity-relationship model. *Computing Surveys, 18*, 2 (1986), pp. 197-222.
79. Wang, Y. R. & Madnick, S. Connectivity among information systems. *Composite Information Systems (CIS) Project 1* (1988), 141 pages.
80. Wang, Y. R. & Madnick, S. E. Evolution towards strategic applications of databases through composite information systems. *Journal of Management Information System* (Fall, 1988), pp. 5-22.
81. Wang, Y. R. & Madnick, S. Facilitating connectivity in composite information systems, To appear in *The ACM, Database* (1989a).
82. Wang, Y. R. & Madnick, S. The inter-database instance identification problem in integrating autonomous systems. In *Proceedings of the Fifth International Conference on Data Engineering* (February 6-10, 1989).
83. Wang, Y. R. & Madnick, S. A polygen model for Heterogeneous Database Systems: The Source Tagging Perspective. *WP # 3119-90 MSA*. (Sloan School of Management, MIT, Cambridge, MA, January 1990).
84. Weller D. L. & York, B. W. A relational representation of an abstract type system. *IEEE Transactions on Software Engineering SE-10*, 3 (May, 1984), pp. 303-309.
85. Wong, T. K. Data connectivity for the composite information system/tool kit. *WP # CIS-89-03* (Sloan School of Management, MIT, Cambridge, MA. 1989), CISL Project.
86. Yuan, Y. The design and implementation of system P: A polygen database masnagement system, *WP # CIS-90-15*. (Sloan School of Management, MIT, Cambridge, MA. 1990), CISL Project.
87. Zaniolo, C. The database language GEM. *Readings in Object-Oriented Database Systems* (1990), pp. 449-469.