U.S. DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

Analytical Data-Management Systems: 2. Normalization, Atomicity, and Structure of Geosciences Data

by
J.T. O'Connor[1]

Open-File Report 94-406

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards (or with the North American Stratigraphic Code). Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

1. Box 25046 MS 939
Denver Federal Center
Denver, CO 80225

ANALYTICAL DATA-MANAGEMENT SYSTEMS: 2. NORMALIZATION, ATOMICITY, AND STRUCTURE OF GEOSCIENCES DATA

by J.T. O'Connor

ABSTRACT

Normalization of geosciences data for use in a database management system is important for the effective utilization of that system. The normalization process consists of making data values as independent of each other as possible by removing most of the intradataset dependencies. Proper normalization facilitates querying of a database from a one-to-many, a many-to-one, or a many-to-many basis. Atomicity is an important uniqueness aspect of data in a data-storage system; it is often violated in encoded quantitative data. Normalization and atomicity are not independent properties of a dataset although both may be context-dependent and somewhat subjective in application. Virtual relations in a database management system greatly facilitate the user's ease of query and data entry but may be costly in terms of processing time.

INTRODUCTION

Scientists rely on data although their treatment of data is often rather cavalier. Chemical data resides in isolated databases, in paper files in filing cabinets, or in piles of paper on desks; field notes are scattered through stiff little books, on water-stained note pads, and on pin-pricked photos; petrographic analyses hide in manila folders and on sheets of paper that have slid under the wordprocessor. With the advent of massive electronic data collection systems and data base management systems (DBMS or RDBMS, if relational) such treatment is no longer practicable or even tolerable. Distributed databases permit utilization of data by persons with varied or limited backgrounds in the data, a variation assured by public usage of scientific data. Applications and attempted applications not even contemplated by the original collector will be commonplace with such databases. The form of data storage has been something that most scientists have taken for granted - it has been whatever form they were used to. But the increased mass of data and its widened application have rendered this tolerance obsolete as well; the form of data storage has become critical for scientific communication.. The change in the breadth and depth of scientific data places an increased burden on the modern scientist. In order to take advantage of the potential of modern data systems, and to avoid the stigma and reality of social and scientific irrelevancy, the scientist should participate in the design of the data storage system as well as the collection and interpretation of the data. This note is intended to outline some of the form criteria for analytical data storage and to present some of the advantages and disadvantages of different choices.

NORMALIZATION

Normalization of a database is a process that, like its mathematical counterpart, decomposes a data set into more effective subsets. Like mathematical normalization, data system normalization enhances some aspects of the system while detracting from others. Normalization of datasets is generally accepted as necessary by database managers for dataset utilization within a RDBMS (Date, 1982), although the concept is resisted by many scientists who prefer highly-coded data units with which they are personally familiar. Several purposes are achieved by normalization including the avoidance of data redundancy, data conservancy (*i.e.* the prevention of loss of parts of a data set when other related parts are deleted from the set), and the deterrence of ambiguity in data set updating. For the everyday database user, to whom the following outline may appear eclectic and inapplicable to everyday problems, a rule of thumb is offered: A fully normalized database is one in which the elements (variables and tables) have been simplified and separated so that they are truly independent data entities within the context of the expected database use.

The process of database normalization is based on the mutual dependency of elements of the data set (with respect to data processing). More formally, normal forms of data (degrees of normalization) are defined by the state of functional dependence (including transitive dependence) of its attributes. Although normalization is discussed in analytical terms and algorithms exist for normalizing data sets, there is a fair amount of fuzziness even in the definition of the process. For instance, a relation scheme *R* is in the first normal form if the values in *dom(A)* are atomic for every attribute *A* in *R* (Maier, 1983). However, a value that is atomic in one application may not be atomic in another (*e.g.* a geochemist may be intereted in the variation in absolute Na values given by atomic absorbtion sectrometers and xray fluorescence spectrometers whereas the geochemist's administrative officer may be interested only in the total number of analyses completed. A coded labnumber that contained the method as part of the code would be atomic for the administrative officer but not for the geochemist). Atomicity is defined in this report as a value that is nondecomposable so far as the system is concerned (Date, 1982); therefore, the context of the database must be considered and its uses anticipated.

The discussion of normal forms below will concentrate on the advantages and disadvantages of different normal forms using an existing geosciences database

schema (O'Connor, 1994). In general the discussion will follow that of Date (*see* Date, 1982, for references and for a more rigorous derivation of normal forms).

The first normal form requires only atomicity of values with respect to the data system (data context). The value and necessity of the first normal form lies in the separation of autonomous values (e.g. pointid, state, and county attributes in Table 1) so that one is not lost if the other is deleted or not entered. If Formation and System were combined, for instance, retrieval of the last entry by a stratigraphic query would be complicated because of a partially missing value. Some disadvantage ensues from this restriction of the common practice of encoding related, but not dependent, values into one attribute. The NCRDS database USCHEM used with the PACER DBMS was a first normal form [1] consisting of one tuple for each analysis containing every attribute for that sample.

USSTRAT (partial) - SINGLE RELATION					
POINTID	**STATE**	**COUNTY**	**UNIT**	**FORMATION**	**SYSTEM**
wx603	CO	JF	1	Idaho Springs	PreCambrian
wx603	CO	JF	2	Idaho Springs	PreCambrian
wx605	CO	FR	1	Idaho Springs	PreCambrian
jto33	VA	LDN	1	Antietam	Cambrian
jto33	VA	LDN	2	Antietam	Cambrian
jto43	MD	FDK	1	Antietam	Cambrian
jto44	MD	FDK	5	.....	Cambrian

Table 1. Tabulation of a relation from the NCRDS DBMS USSTRAT in the first normal form. This model is currently found in many familiar databases (e.g. RASS, NCRDS, DBASE (suggested models), RBASE (suggested models), LOTUS databases, SPSS databases, and more).

The second normal form requires first normal form plus every nonkey attribute must be fully dependent upon the primary key. A relation such as USSTRAT (Figure 1) is not of second normal form because values in the tuples (Unit, Formation,

1. Actually less than first normal because of the use of encoded qualifiers in recording chemical analysis results.

and System) are not uniquely defined by the primary key (Pointid). This relation can be decomposed into two relations that are of second normal form by placing only the functionally dependent attributes in the same relations (Figure 2). In a geo-

LOCATION		
POINTID	STATE	COUNTY
wx603	CO	JF
wx605	CO	FR
jto33	VA	LDN
jto43	MD	FDK
jto44	MD	FDK

UNIT DATA			
POINTID	UNIT	FORMATION	SYSTEM
wx603	1	Idaho Springs	PreCambrian
wx603	2	Idaho Springs	PreCambrian
wx605	1	Idaho Springs	PreCambrian
jto33	1	Antietam	Cambrian
jto33	2	Antietam	Cambrian
jto43	1	Antietam	Cambrian
jto43	5	.....	Cambrian

Table 2. Second normal form resulting from decomposition of the original single relation database into two relations makes the State and County attributes uniquely dependent upon the Pointid but leave ambiguity and redundancy in the UNITDATA relation.

sciences database the two problems arising from not realizing a second normal form are redundancey and poor error-correction efficiency. The State and/or County value may be repeated many times for each Pointid in the less normalized relation. An error in State or County assignment to Pointid in any of the tuples will propagate and could lead to difficult-to-identify data losses during queries of the unnormalized USSTRAT relation. A similar error in data entry in the second normal form of Figure 2 would result in all the State or County assignments for that Pointid being incorrect but this result is much less likely to occur because of the single entry for Pointid, State, and County. If it does occur, this error is much easier to detect by a database user (even if not familiar with the stratigraphic section at that location) because of the simple form of the relation. There maybe hundreds of unit numbers for a particular Pointid (if that point represents a logged well or measured section), so the redundancy reduction and error correction efficiency are significant.

Although the second normal form is an advance over the previous, it still suffers from a transitive dependence. Examining the UNITDATA relation in Figure 2 we

can see that the System value can be determined through the Formation value for any Pointid[1]. Such transitive dependence implies that information about a formation might not reside in a subset of the data unless Pointid's within that formation were part of the subset. Such disparities could affect the construction of schematic stratigraphic sections from a database subset. If we wanted to construct such a section for Frederick County, MD, for instance, and no Pointid's occurred within the Cambrian rocks of the county, we could miss the entire Cambrian part of the section.

The third normal form reduces the problem by requiring second normal form plus every nonkey attribute must be nontransitively dependent on the primary key (Date, 1982, p. 248). The third normal form (or the more restrictive BNF form - Date, 1982, p. 249) of these relationships is shown in Table 3 . With a composite

1. In an ideal stratigraphic world. In the real world, the State value is also necessary because of interstate nomenclature problems. Ambiguities may still exist at this level of refinement but such ambiguities are part of the stratigraphic nomenclature world and will not be resolved by database managers.

LOCATION				UNITDATA		
POINTID	STATE	COUNTY		POINTID	UNIT	FORMATION
wx603	CO	JF		wx603	1	Idaho Springs
wx605	CO	FR		wx603	2	Idaho Springs
jto33	VA	LDN		wx605	1	Idaho Springs
jto43	MD	FDK		jto33	1	Antietam
jto44	MD	FDK		jto33	2	Antietam
				jto43	1	Antietam
				jto44	5	.....

FORMATIONS		
FORMATION	STATE	SYSTEM
Idaho Springs	CO	PreCambrian
Antietam	VA	Cambrian
Antietam	MD	Cambrian

Table 3. Third and fourth normal (BNF) forms resulting from decomposition of the original single relation database produces three relations in which transitive dependencies are removed. A more complex dataset (e.g. coalescing formational names; merging and splitting litholoies; different levels of reporting of details; and other complexities) may have more complex dependencies that are not as easily resolved as in this example.

key of Pointid/Unit the projection of Table 3 is also in the fourth normal form - no multivalued dependencies exist (Date, 1982, p. 259) although Formation must be a mandatory field if data for the last entry (jto44) is not to be lost.

There is at least one more level of decomposition, that of the fifth normal form, which depends upon removing join-dependencies. Join-dependencies involve projections of relations from a single relation to three [or more] new relations joined through several attributes (Date, 1982, p. 262). This form may be difficult to recognize (Date, 1982, p. 263). All projections to fourth and fifth normal forms are not necessarily good decompositions. The selection of good projections is, to some extent, intuitive but the rule-of-thumb is to make the projections as independent of each other as possible within the context of the database. For a proposed U.S. Geol.

Survey database structure (O'Connor, 1994), some relations are shown in Table 4. .

LOCATION	UNITDATA	STRATIG.	FORMATION	FIELD DET	SEDIMEN-
Point ID *	Point ID *	Point ID	Formation *	Point ID *	Point ID *
Field ID	Unit ID *	Unit ID	State *	Unit ID *	Unit ID *
X Coord.	Lithology	Formation	Group	Subunit ID *	Subunit ID *
Y Coord.	Thickness	Member	System	Feature	Bed Forms
XY Code	Data Type	Bed	Series	Description	Grain Forms
Surface Elev.	Data Quality				Clast Data
State	Data Source				Contact Data
County					Color
Zipcode					Comment

FIELD / LAB CORRELATION	SAMPLE EVALUATION
Point ID *	Point ID *
Unit ID *	Laboratory Number *
Subunit ID *	Analysis Type
Laboratory Number *	Sample Description
Laboratory Code	Sample Representation
	Sample Reliability
	Sample Source
	Analysis Date
	Collection Date

Table 4. Partial schema for a proposed U.S.Geol. Survey database. Normalization of the data has been balanced with the normal usage patterns of geosciences data and with the complexity of deriving some transitively dependent balues (e.g. surface elevation from X and Y coordinates).

QUANT. ANAL.	QUAL. ANAL.	PETROGRAPHY	PALEO. DATA
Lab Number *	Lab Number *	Lab Number *	Lab Number *
Measurement	Component	Rock Type	Species ID
Species	Tech./Proc.	Primary Fabric	Abundance
Units	Quality	Secondary Fabric	Age/Environ. ID
Qualifier	Comment	Alteration Fabric	Comment
Quality		Comment	
		Image ID	

Table 5. Partial Schema (continued) for a proposed U.S. Geol. Survey database. Normalization of the data has been balanced with the normal usage patterns of geosciences data and with the complexity of deriving some transitiely dependent values (e.g. surface elevation from X and Y coordinates).

Note that transitive dependencies exist between the X Coordinate and Y Coordinate and the State, County, and Elevation in the LOCATION table. However, the dependency is not easily calculated in most areas and these attributes can be considered independent. Parallel relations (tables) are proposed for quantitative chemical and mineralogic analyses - not to satisfy normalcy requirements but to lower the quantity of data searched in queries for either petrographic or chemical values - typically separate operations.

ATOMICITY

Atomicity was briefly mentioned in the discussion of the first normal form in the previous section. Many attempts to compress data for storage in databases result in composite attributes with non-atomic values. Because atomicity is a relative concept - relative to the database context and the perceptions of the users - there are no absolute definitions to follow. The following examples are offered to illustrate the types of consideration used for determining atomicity in the proposed U.S. Geological Survey database. Quantitative chemical values are sometimes listed as qualified values (e.g. 1.023g - where g indicates "greater than". This form was not considered an atomic value in the database because of the two separate (if related) uses of the numneric and alpha parts of the composite entry. Hence, numbers and qualifiers

were entered separately. In a sedimentology data relation, however, bedding forms are suggested to be entered as coded values (Table 6) which would be decoded by a lookup table

BEDDING-FORM CODES FOR DATA ENTRY IN SEDIMENTOLOGY RELATION			
CODE	BEDDING-FORM DESCRIPTION	CODE	BEDDING-FORM DESCRIPTION
I	External Form (EF) unobserved	A	Internal Organization (IO) unobserved
II	(EF) equal/subequal thickness, laaterally uniform, continuous.	B	(IO) massive
III	(EF) unequal thickness, laterally uniform, continuous.	C	(IO) bedded (parallel)
IV	(EF) unequal thickness, laterally variable, continuous.	D	(IO) laminated (normal)
V	(EF) unequal thickness, laterally variable, discontinuous.	E	(IO) graded
		F	(IO) imbricated or oriented
		G	(IO) growth structures (primary)
		H	(IO) flaser bedded
		J	(IO) fining upward
		K	(IO) fining downward
		L	(IO) cross-bedded
		M	(IO) channel-formed
		N	ripple-laminated
		O	combination or other, see comment.
CODE	BEDDING-FORM DESCRIPTION	CODE	BEDDING-FORM DESCRIPTION
1	Bedding Size (BS) unmeasured	a	Bedding Deformation (BD) unobserved
2	(BS) very thick (> 1m)	b	(BD) load clasts, ball and pillow structures
3	(BS) thickly bedded (30 - 100 cm)	c	(BD) convolute bedding
4	(BS) medium bedded (10 - 30 cm)	d	(BD) slump structures
5	(BS) thinly bedded (3 - 10 cm)	e	(BD) injection structures
6	(BS) very thinly bedded (1 - 3 cm)	f	(BD) root zones
7	(BS) thickly laminated (0.3 - 1 cm)	g	(BD) bioturbation
8	(BS) thinly laminated (< 0.3 cm)	h	(BD) combination of others, see comment.

CODE	BEDDING-FORM DESCRIPTION	CODE UTILIZATION
1)	Bedding Plane Markings (BPM) unobserved	Enter: I,A,1,a,1) for no observations or measurements.
2)	(BPM) scour or tool marks	
3)	(BPM) trace fossils	Substitute appropriate dominant feature code in corresponding category.
4)	(BPM) other organic remains	
5)	(BPM) parting lineations	Include commas in entry.
6)	(BPM) erosional markings (rill marks, current crescents)	Additions or edits must be made to the Beddingforms Table.
7)	(BPM) pits or bubble markings (rain prints)	Use Comment field for more detail and odd features
8)	(BPM) mud cracks and casts	
9)	(BPM) combinations or others, see comment	

Table 6. Proposed bedding form codes that may be entered into the bedding-form attribute of the sedimentology relation are modified from Folk, 1980. This form can exist as a look-up table in the DBMS so that a query would produce the text, rather than the code

Although the values represented by the parts of the code could be entered separately, common usage of the definitions ofTable 5 (Folk 1980) prompts the use of the composite variable in this instance.

A host of potential coded variables were considered by the author in proposing the above cited schema. Among those rejected as nonatomic were:

1. Composite UNITS, TECHNIQUE, QUALIFIER, and QUALITY codes for chemical analyses
2. Composite codes for SOURCE and LABORATORY EVALUATION.
3. Composite codes for LITHOLOGY, COLOR, and GRAIN SIZE.OTHER FACTORS IN RELATION ORGANIZATION

Even for relations for which each nonkey value of a tuple is fully dependent upon the [composite] key, it is easliy possible for the tuples to be too large. For example, a quantitative mineralogic data relation could have several hundreds of

possible entries in a schema (see e.g. Table 7 for a partial listing of mineral types).

DATABASE MNEMONIC	MINERAL SPECIES REPRESENTED
QZMO	Monocrystalline quartz (igneous?)
QZPL	Polycrystalline quartz (metamorphic?)
QZCY	Recycled quartz (sedimentary origin?)
CHRT	Chert (microcrystalline quartz)
QZFS	Quartz replacing feldspar
QZRX	Quartz replacing rock-fragment clasts
KSPR	K-feldspar (undifferentiated)
PLAG	Plagioclase feldspar (undifferentiated)
FSPR	Feldspar (undifferentiated)
ILLT	Illite (as matrix or void filling)
KLNT	Kaolinite (as matrix or void filling)
CLAM	Mixed-layer clay minerals
CLAY	Clay minerals (after rock-fragments)
CLFS	Clay minerals (replacing feldspar)
RXFN	Rock fragments (very-fine-grained)
RXCS	Rock fragments ("coarse" grained)
CLCT	Calcite (as matrix or in voids)
CTFS	Calcite (replacing feldspar)
CTRX	Calcite (replacing rock fragments)
SIDT	Siderite (as matrix of in voids)
SIDX	Siderite (replacing rock fragments)
SIDF	Siderite (replacing feldspar)

Table 7. Mineralogic species represented in the database can be listed in a lookup table as above. Such a table can be modified by the user or restricted to system-administration modification. Reports written using the data within the Petrography Table can reference either the mnemonic or the mineral species name (or a nickname if it is also entered).

Virtual tables can hide this complexity from the user but the creation of a virtual table usually involves extensive hidden database operations to satisfy a query, a data entry, or an edit. Instead, we have chosen to employ a tuple that includes only one Measurement along with its Units, Technique, Qualifier, and Quality. This choice results in a large number of tuples in the relation, a result that influences the time for queries (e.g. a query for total quartz in a sample will require retrieval of 6 tuples and addition of the measurements). In some cases, because the tuples are small and key searches are used, this procedure will be faster than retrieving one very large tuple, sorting the measurements, and adding six separate attributes. This choice of form gains versatility by allowing the species attribute to be a variable. Data entry is greatly facilitated by identifying the species at entry time (including preidentification of common species through forms). The efficacy of this choice can only be determined by testing with large quantities of data, a procedure that is currently under way using the INGRES DBMS.

Common usage prompted the inclusion of the transitive dependent State, County, and Zipcode attributes within the LOCATION relation. Standard county codes are unique but normal data-entry uses a non-unique common countyname that is not transitively dependent upon State. Zipcode is join-dependent upon X-Coordinate and Y-Coordinate and could replace State and County in the LOCATION table or be in an ancillary table. However, the calculations required for determining such dependencies seemed unjustified to the author for the avoidance of small redundancies or the strict adherence to normalcy.

LOOKUP TABLES AND VIRTUAL TABLES

Lookup tables allow abbreviated or mnemonic attributes to be stored wheras [decoded] long descriptions may appear in reports generated by queries to the coded attributes. Two examples are the bedding-form codes of Figure 5 and the mineral abbreviations of Figure 6. Care must be taken in forming such tables that atomicity of the attributes is maintained and that functional dependencies are not compromised.

Virtual tables (views) are of great assistance to the database user who does not want to see the acutal database structure but does need to utilize a subset (usually a well-defined subset) of the data. For queries, report writing, or data entry, virtual tables can supply the user with the appropriate data subset in a format of his/her choice.

In some DBMS implementations the subset must be supplied through a form based on a join definition - this part of the implementation should be transparent to the user. Such virtual tables should not be confused with truly smaller datasets. Because they are products of relational language constructs, they carry the baggage of overall database construction with them. In preliminary testing of the cited database, only small savings in query time were realized by the definition of views and their specifying the appropriate relation joins and attribute selection. The author would speculate that the saving in time was due to the greater efficiency of the DBMS (than the query writer) in optimizing the search pattern. Reports generated from virtual tables that invlove complex calculations (Table 6) should be expected

Original Data

Laboratory Number	Measurement	Oxide Species	Units	Qualifier	Quality
test-1a	22.93	Al2O3	pct	-	10
test-1a	9.74	MgO	pct	-	10
test-1a	30.98	FeO	pct	-	10
test-1a	34.81	Cr2O3	pct	-	10
test-1a	0.22	MnO	pct	-	8
test-1a	0.56	TiO2	pct	-	8
test-1a	8000	SiO2	ppm	gt	6
	99.34	Sum (S1) in percent			
Lookup Table					

Atomic Species	Atomic Wt	Oxide Species	Calculation	Mole Number
O	15.999	Al2O3	22.93/(2*26.982+3*15.999)	0.2249
Al	26.9815	MgO	9.74/(24.312+15.999)	0.2416
Mg	24.312	FeO	30.98(55.847+15.999)	0.4312
Fe	55.847	Cr2O3	34.81/(2*51.996+3*15.999)	0.2290
Cr	51.996	MnO	0.22/(54.938+15.999)	0.0031
Mn	54.938	TiO2	0.56/(47.90+2*15.999)	0.0070
Ti	47.90	SiO2	0.08/(28.086+2*15.999)	0.0017
Si	28.086		Sum (S2)	1.1382

Oxide Species	Calculation	Mole Percent	Calculation (I1)	No. atoms 24*(I1)/S3
Al2O3	S1*0.2249/S2	20.8667	20.8667*2	6.78
MgO	S1*0.3026/S2	22.4187	22.4187	3.64
FeO	S1*0.3384/S2	40.01	40.01	6.50
Cr2O3	S1*0.3198/S2	21.25	21.25*2	6.90
MnO	S1*0.0113/S2	0.29	0.29	0.05
TiO2	S1*0.0050/S2	0.65	0.65	0.11
SiO2	S1*0.0017/S2	0.12	0.12	0.02
			Sum (S3)	24.00

Oxidation Calculation			Excess Charge 64 - (S4)		No. Atoms
Al	3*6.78	20.34	2.06	Al	6.78
Mg	2*3.64	7.28		Mg	3.64
Fe(+2)	2*6.50	13.00		Fe(+2)	4.44
Cr	3*6.90	20.71		Fe(+3)	2.06
Mn	2*0.05	0.09		Cr	6.90
Ti	4*0.11	0.42		Mn	0.05
Si	4*0.02	0.08		Ti	0.11
Sum of Positive Charges (S4)		61.94		Si	0.02
Virtual Table Using Calculations					
Lab Number		R1 = Mg/(Mg + Fe(+2))	R2 = Cr/(Al + Cr + Fe(+3))	R3 = Fe(+3)/ (Al + Cr + Fe(+3))	
Test-1a		0.451	0.438	0.131	
(more samples)		.....	.....	.....	

Table 8. Example of a virtual table derived from chemical data (chromite analyses) by calculations performed within the database. This type of virtual table could call for the commonly used ratios (R1, R2, and R3) for all chromite analyses defined by any location or other geologic criterion but the query time may not be very fast and user defined programs that access the database through the DBMS would probably be quicker and more efficient.

to be relatively slow in execution. It may be far quicker to perform such calculations with a program written for the purpose utilizing programmatic internal calls to the database. The user must decide, in coordination with the database administrator and experienced programming assistance, if the savings gained by externally programmed retrieval are worth the effort of finding or writing such programs.

SUMMARY

The utilization of a relational DBMS as part of a scientific investigation places aspects of data-formatting and integrity into consideration that have heretofore been largely ignored. A database constructed during such an investigation will often be used by personis uninvolved in its construction or only peripherally familliar with many parts of it. It is, therfore, important for its effective use that data redundancies and omissions be minimized. The only possible means for such optimization is the involvement of both the scientist and the data specialist in the design of the data-

base. The scientist must be involved because of the meaning of elements of a database within the context of the scientific study. The data specialist must be invloved because of the part that database semantics play in effective querying, report writhing, and data entry. It is extremely important for management personnel to realize that the expansion of task effectiveness contributed by operating in a database environment can only be realized as a result of investment of time, staff, and money in proper system design. Withoutsuch involvement, database operations will always remain purely demonstration projects with potential but no real-world return.

REFERENCES

Date, C. J., 1981, An Introduction to Database Systems; Volume 1; Addison-Wesley Publishing Company, Reading, MA, 574 p.

Flores, I., 1981, Data Base Architecture: Van Nostrand Rheinhold Co, New York, 396 p.

Folk, R.L., 1980, Petrology of Sedimentary Rocks: Hemphil Publ. Co., Austin, Texas, 184 p.

Maier, D., 1983, The Theory of Relational Databases: Computer Science Press, Rockville, MD., 637 p.

O'Connor, J.T., 1994, Analytical Data-management Systems; 1. Data Management Strategy for Geo-Sciences Databases: U.S. Geological Survey Open File Report No. 94-405.