LOGICAL OPTIMIZATION FOR DATABASE UNIFORMIZATION

Technical Report

Principal Investigator: Dr. John Grant
Department of Computer and Information Sciences
Towson State University
Towson, MD 21204

Project Period: September 1, 1983 - August 31, 1984

Grant No. NAG 5-267
1. Introduction

Database uniformization refers to the building of a common user interface facility to support uniform access to any or all of a collection of distributed heterogeneous databases. Such a system should enable a user, situated anywhere along a set of distributed databases, to access all of the information in the databases without having to learn the various data manipulation languages. Furthermore, such a system should leave intact the component databases, and in particular, their already existing software. A survey of various aspects of the database uniformization problem and a proposed solution can be found in [2].

The proposed solution involves a global data manager. The components are illustrated in Figure 1.1 (the older version is given in Figure 1.2). The global data manager includes a global model which provides a uniform front-end for all users to interact with the databases. The global model is based on database logic and includes the global data manipulation language for user interaction with the global data manager. The global dictionary contains all of the individual databases in the network as well as previously created external views. The user consults the global dictionary to extract the databases of interest.

In our previous report ([1]) we described a portion of the external-to-conceptual translation component of the global data
manager. That component translates the user's queries on his/her global view to queries on the underlying view. In general, the translated queries tend to be in a highly complex form. In the old version of the global data manager, logical optimization was one component. In the new version the external-to-conceptual translation component includes logical optimization whose purpose is to simplify logical formulas. We continue to use the sample NASA GSFC databases with typical GSFC applications from the previous report.

In the database literature logical optimization is usually discussed in terms of minimization of tableaux ([4]). A tableau is a tabular representation of a query for a relational database schema. A query is logically optimized by obtaining an equivalent tableau with the minimum possible number of rows. The process involves the deletion of some subformulas in the original query without modifying the answer for any database (which is a model of the appropriate view). This optimization process is also called folding. In our case we need to generalize folding to heterogeneous (i.e. relational, hierarchic, and network) databases via database logic. The logical optimization process also needs to apply the constraints of the view, i.e. the nonlogical axioms. This generalizes the minimization process for chased tableaux.

The outline of this report is as follows. In Section 2 we review the database logic views for the NASA databases ERB-ORAC, ERB-SEED and PCDB, as well as for various external views
constructed in [1]. We note that the original definitions for these databases as well as some of the queries we use are taken from [3]. In Section 3 we describe two logical optimization algorithms for interpreted queries over a single conceptual view. The first one can be used for existential conjunctive queries, while the second one allows local disjunctions also. We then apply logical optimization to the interpretation of four queries. In Section 4 we describe the generalization of the logical optimization algorithm for interpreted queries over a distributed conceptual view. We also apply the generalized logical optimization algorithm to an example. We summarize the report in Section 5.

2. Database Views and Interpretations

In this section we review views and interpretations which we constructed and used in [1]. A database view is comprised of a schema which describes the structure of the database, a language which is used for making assertions about the database, and a set of constraints that every instance of the database must obey. In database logic an external view is constructed by formulating an interpretation of the external view into the conceptual view. An interpretation consists of a coding section which indicates the transformation of types, a defining formula section which indicates the transformation of predicates and functions, and a constant transformation section.
Appendix 2.1 contains \( V(ERB-ORAC) \), the Earth Radiation Budget database maintained under ORACLE; while Figure 2.1 contains an instance of this database. Appendix 2.2 contains \( V(ERB-SEED) \), the Earth Radiation Budget database maintained under SEED; while Figure 2.2 contains an instance. The database view \( V(PCDB) \) for the Pilot Climate Data Base which is maintained under ORACLE is given in Appendix 2.3; an instance is given in Figure 2.3.

In [1] we constructed 2 views over a single conceptual view: \( V(ERB-R) \), a relational view, over \( V(ERB-SEED) \); and \( V(ERB-N) \), a network view, over \( V(ERB-ORAC) \). We give \( V(ERB-R) \) in Appendix 2.4 and \( V(ERB-N) \) in Appendix 2.5. The construction of \( V(ERB-R) \) as an external view over \( V(ERB-SEED) \) is the \( I(ERB-R;ERB-SEED) \) of Appendix 2.6. The induced \( ERB-R \) database instance for the \( ERB-SEED \) instance of Figure 2.2 is given in Figure 2.4. Similarly, we give the construction of \( V(ERB-N) \) over \( V(ERB-ORAC) \) as \( I(ERB-N;ERB-ORAC) \) in Appendix 2.7, and the corresponding induced \( ERB-N \) instance for the \( ERB-ORAC \) instance of Figure 2.1 in Figure 2.5.

In [1] we also constructed views over distributed databases. The \( V(GLOBE) \), given in Appendix 2.8, is the union of \( V(ERB-ORAC), V(ERB-SEED), \) and \( V(PCDB) \). It is not a true union, however, but is oriented towards \( ERB-ORAC \) rather than \( ERB-SEED \). Figure 2.6 contains an instance of \( GLOBE \) based on the instances of \( ERB-ORAC, ERB-SEED, \) and \( PCDB \) from Figures 2.1-2.3. Additionally, we constructed a view, \( V(GLOBE-R) \), given in
Appendix 2.9, for a user wishing to see the distributed database as a single relational view. \( V(\text{GLOBE-R}) \) is given as an external view over \( V(\text{GLOBE}) \) by the interpretation \( I(\text{GLOBE-R}; \text{GLOBE}) \) of Appendix 2.10. We present the GLOBE-R database instance induced from the GLOBE database instance of Figure 2.6 in Figure 2.7.

3. Logical Optimization of Queries over a Single Conceptual View

Recall that a user writes a query on his/her external view. Such a query is transformed to a query on the conceptual view by the interpretation. Often, such an interpreted query is in a highly complex form whose straightforward evaluation would be very costly. The process of logical optimization is used to generate a query which yields the identical answer as the interpreted query but which is simpler in form and hence less costly to execute.

We first deal with existential conjunctive queries in database logic since they are in the appropriate form for many (interpreted) queries. We can write such a query as

\[
Q(<R;>) = (E)<R'>((P1(<R;>,<c>,<R'>)) & \ldots & Pn(<R;>,<c>,<R'>))
\]

where \( <R;> \) are the free (target) variables, \( <c> \) are the constants, and \( <R'> \) are the existentially quantified variables of the query. Each conjunct \( P_i \) is a cluster. We assume that function symbols may appear only in equalities. An interpreted query is initially often not in this form but can be placed into
this form by moving the quantifiers to the front.

We logically optimize an existential conjunctive query by getting rid of superfluous conjuncts. The algorithm is given in Appendix 3.1. It consists of three parts. In the first part we delete conjuncts which are superfluous equalities. In the second part we identify objects forced to be equal by functional dependency constraints. In the third part we delete subsumed clusters. Q(i)(〈R; 〉) is the optimized query at the end.

We demonstrate the logical optimization algorithm on some examples. We take two queries on V(ERB-R), interpret them in V(ERB-SEED), and optimize the interpreted queries. The first query is given in [1]; it was originally given in [3] as a sample query on the ERB database maintained by Oracle. We present each query in English first and then in DBL on V(ERB-R). Then we give the interpreted DBL queries on V(ERB-SEED) and show the steps in the optimization. We also present the answers to the queries.

The first query is given in Appendix 3.2. The interpreted query is first placed in prenex conjunctive normal form. In the first three steps the three superfluous equalities are eliminated. In the next three steps the functional dependency RTAPE:NOTAPE→TAPETYPE,PLAYBACK,RTITLE is used to identify three pairs of existentially quantified variables. The optimized query is then obtained by deleting a subsumed conjunct. The answer to query 1 on the database of Figure 2.4 is given in Figure 3.1. This is identical to the answer to the interpreted (and optimized) query on the database.
of Figure 2.2.

The second query is given in Appendix 3.3. The interpreted query is again first placed in prenex conjunctive normal form. In this case the optimization is achieved in one step by renaming existentially quantified variables and deleting a subsumed conjunct. The answer to query 2 on the database of Figure 2.4 is given in Figure 3.2; it is identical to the answer to the interpreted (and optimized) query on the database of Figure 2.2.

Next we consider two queries (the third and fourth) on V(ERB-N), interpret them in V(ERB-ORAC) and optimize the queries. However, in these cases, the interpreted queries are not purely conjunctive as one conjunct contains disjunctions. We write such queries as above but allow a Pi to be a disjunction of clusters. We also modify our algorithm to try to delete some of the disjuncts. This modified algorithm, for an existential query with local disjunctions, is given in Appendix 3.4. We then apply the new algorithm to these two queries.

The third query is given in Appendix 3.5. It is taken from C13; it was originally given in [3] as a sample query on the ERB database maintained by SEED. The optimization consists of two steps: in each step a disjunct is deleted from the third conjunct. The fourth query is given in Appendix 3.6. The first two steps in the optimization are the same as for the third query. But then an extra step is used to rename some existentially quantified variables and delete the subsumed conjunct. The answers to these two queries on the database of
4. Logical Optimization of Queries over a Distributed Conceptual View

In the previous section we gave two algorithms to logically optimize queries. The first algorithm can be applied to purely conjunctive queries; while the second algorithm can be applied to queries with local disjunctions, typically disjunctions in a single conjunct. However, when we deal with queries over a distributed conceptual view, there may be disjunctions in many conjuncts. Recall, for example, \( I(\text{GLOBE-R}; \text{GLOBE}) \), given in Appendix 2.10, where GLOBE is a distributed database. Note that the interpretations for several predicates contain disjunctions in addition to conjunctions. In contrast, \( I(\text{ERB-R}; \text{ERB-SEED}) \), given in Appendix 2.6, contains only conjunctions; while \( I(\text{ERB-N}; \text{ERB-ORAC}) \), given in Appendix 2.7, contains local disjunctions in one conjunct only. Therefore, in dealing with distributed views, we must optimize existential disjunctive queries.

The logical optimization of such an interpreted query consists of several phases. We first place the query in prenex conjunctive normal form. This allows the deletion of superfluous
equalities and subsumed local disjuncts as in the logical optimization of a query with local disjunctions. Then we place the query in prenex disjunctive form. We logically optimize each disjunct separately as in the logical optimization of a conjunctive query. Finally we delete subsumed disjuncts. Note that while a subsumed conjunct is a subcluster of another cluster, a subsumed disjunct contains enough clusters to subsume all the clusters of another disjunct. Also, in the case of conjuncts, existentially quantified variables are identified with other objects to obtain subsumptions; while in the case of disjuncts, various objects are identified with existential quantifiers to obtain subsumptions. The algorithm is given in Appendix 4.1.

We end this section by demonstrating the logical optimization algorithm on an example query taken from [1]. We modify the example slightly to remove a function symbol from inside a predicate in order to apply the logical optimization algorithm. The query, which we call query 5, is logically equivalent to the original query. The complete example is given in Appendix 4.2. We present the query in English first and then in DBL for V(GLOBE-R). We also give the interpreted query for V(GLOBE). We place this query in prenex conjunctive normal form. In this case there are no equations or local subsumed disjuncts. Then we place the query in prenex disjunctive normal form. As we go through the conjuncts separately we find that the RECINFO predicate appears twice in the first and fifth conjunct.
The extra appearances are deleted. Finally we find that four of the eight disjuncts are subsumed. We obtain the logically optimized query by deleting the subsumed disjuncts. We present the answer to query 5 on the database of Figure 2.7 in Figure 4.1. This is identical to the answer to the interpreted (and optimized) query on the database of Figure 2.6.

5. Summary

In this report we demonstrated the logical optimization portion of the external-to-conceptual translator component of the data manager for database uniformization. A user of the proposed system will write queries on his/her external view. The external-to-conceptual translator will translate such a query to the conceptual view. After some additional steps the new query will be processed and the answer returned to the user. We gave three logical optimization algorithms: one for existential conjunctive queries, another one for queries with local disjunctions, and a general algorithm for disjunctive queries. The first two algorithms can be used to logically optimize queries over a single conceptual view, while the last one can be used for a distributed conceptual view. To illustrate heterogeneity and to demonstrate NASA applications we used the relational databases ERB-ORAC and PCDB and the network database ERB-SEED for our five examples.
References


Appendix 2.1  V(ERB-ORAC)

V(ERB-ORAC): VIEW DEFINITION
S(ERB-ORAC): SCHEMA DEFINITION

TABLE TAPEINFO = (NOTAPE, TAPETYPE, TITLE1, TITLE2, TITLE3)
TABLE FILEINFO = (PB, FILE, NOTAPE)
TABLE RECINFO = (DATE_TIME, DATE, TIME, LON, LAT, ALT, ZEN, PB,
QUALITY, ELECTR, ILLUMIN, CALIB, SCAN)

L(ERB-ORAC): LANGUAGE DEFINITION
T(ERB-ORAC): TYPING DEFINITION

TYPE NUMBER = (NOTAPE, PB, FILE, DATE_TIME, DATE, TIME, LON,
LAT, ALT, ZEN, QUALITY) ASCII NUM(22)
TYPE CHAR4 = (TAPETYPE) ASCII CHAR(4)
TYPE CHAR115 = (TITLE1, TITLE2, TITLE3) ASCII CHAR(115)
TYPE CHAR3 = (ELECTR, CALIB, SCAN) ASCII CHAR(3)
TYPE CHAR8 = (ILLUMIN) ASCII CHAR(8)

NONLOGICAL SYMBOLS DEFINITION

PREDICATES

TAPEINFO: (NUMBER, CHAR4, CHAR115, CHAR115, CHAR115) - PREDICATE
FILEINFO: (NUMBER, NUMBER, NUMBER) - PREDICATE
RECINFO: (NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER,
NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3, CHAR3)
- PREDICATE

FUNCTIONS

CONC: (NUMBER, NUMBER; NUMBER) - PREDEFINED FUNCTION

C(ERB-ORAC): CONSTRAINTS DEFINITION
C(ERB-ORAC, 1): CONSTRAINT TAPEINFO: NOTAPE -> TAPETYPE, TITLE1,
TITLE2, TITLE3
C(ERB-ORAC, 2): CONSTRAINT FILEINFO: PB, NOTAPE -> FILE
C(ERB-ORAC, 3): CONSTRAINT RECINFO: DATE_TIME -> DATE, TIME, LON,
LAT, ALT, ZEN, PB, QUALITY, ELECTR, ILLUMIN, CALIB, SCAN
C(ERB-ORAC, 4): CONSTRAINT RECINFO: DATE_TIME = CONC(DATE, TIME)
Appendix 2.2  V(ERB-SEED)

V(ERB-SEED): VIEW DEFINITION
S(ERB-SEED): SCHEMA DEFINITION
TABLE RTAPE = (NOTAPE, TAPETYPE, PLAYBACK, RTITLE)
TABLE PLAYBACK = (PB, NOFILE, CATALOG)
TABLE RTITLE = (TITLE)
TABLE CATALOG = (LON, LAT, ALT, ZEN, TIME1, QUALITY, ELECTR, ILLUMIN, CALIB, SCAN)
TABLE RDATES = (DATE1, CATALOG)
TABLE RLOM = (ILON, CATALOG)
TABLE RALT = (ILAT, CATALOG)
TABLE RALT = (IALT, CATALOG)
TABLE RZEN = (IZEN, CATALOG)

L(ERB-SEED): LANGUAGE DEFINITION
T(ERB-SEED): TYPING DEFINITION
TYPE INT4 = (NOTAPE, NOFILE, DATE1, TIME1) EBCDIC INTEGER(4)
TYPE CHAR4 = (TAPETYPE) EBCDIC CHAR(4)
TYPE CHAR115 = (TITLE) EBCDIC CHAR(115)
TYPE REAL4 = (PB, LON, LAT, ALT, ZEN) EBCDIC REAL(4)
TYPE INT2 = (ILON, ILAT, IALT, IZEN, QUALITY)
    EBCDIC INTEGER(2)
TYPE CHAR3 = (ELECTR, CALIB, SCAN) EBCDIC CHAR(3)
TYPE CHAR8 = (ILLUMIN) EBCDIC CHAR(8)

NONLOGICAL SYMBOLS DEFINITION
PREDICATES
RDATES-CATALOG: (INT4, CATALOG, REAL4, REAL4, REAL4, REAL4, REAL4, INT4, INT4, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RLOM-CATALOG: (INT2, CATALOG, REAL4, REAL4, REAL4, REAL4, INT4, INT2, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RLAT-CATALOG: (INT2, CATALOG, REAL4, REAL4, REAL4, REAL4, INT4, INT2, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RALT-CATALOG: (INT2, CATALOG, REAL4, REAL4, REAL4, REAL4, INT4, INT2, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RZEN-CATALOG: (INT2, CATALOG, REAL4, REAL4, REAL4, REAL4, INT4, INT2, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RTAPE-PLAYBACK-CATALOG-RTITLE: (INT4, CHAR4, PLAYBACK, REAL4, INT4, CATALOG, REAL4, REAL4, REAL4, REAL4, INT4, INT2, CHAR3, CHAR8, CHAR3, CHAR3, RTITLE, CHAR115) - FULL CLUSTER PREDICATE

FUNCTIONS
TITLE1: (INT4; CHAR115) - FUNCTION
TITLE2: (INT4; CHAR115) - FUNCTION
TITLE3: (INT4; CHAR115) - FUNCTION
INT: (REAL4; INT2) - PREDEFINED FUNCTION
C(ERB-SEED):CONSTRAINTS DEFINITION

C(ERB-SEED,1): CONSTRAINT RTAPE:NOTAPE->TAPETYPE,PLAYBACK, RTITLE
C(ERB-SEED,2): CONSTRAINT RTAPE-PLAYBACK:NOTAPE,PB->Nofile, CATALOG
C(ERB-SEED,3): CONSTRAINT RDATES:DATE1->CATALOG
C(ERB-SEED,4): CONSTRAINT RLON:ILON->CATALOG
C(ERB-SEED,5): CONSTRAINT RLAT:ILAT->CATALOG
C(ERB-SEED,6): CONSTRAINT RALT:IALT->CATALOG
C(ERB-SEED,7): CONSTRAINT RZEN:IZEN->CATALOG
C(ERB-SEED,8): CONSTRAINT RLON-CATALOG:ILON=INT(LON)
C(ERB-SEED,9): CONSTRAINT RLAT-CATALOG:ILAT=INT(LAT)
C(ERB-SEED,10):CONSTRAINT RALT-CATALOG:IALT=INT(ALT)
C(ERB-SEED,11):CONSTRAINT RZEN-CATALOG:IZEN=INT(ZEN)
V(PCDB): VIEW DEFINITION

S(PCDB): SCHEMA DEFINITION

TABLE TAPE = (TAPEID,MISSION,SENSOR,FORMAT,PROJNUM,GENDATE, INVDATE,ARCHIVER,NUMFILES,TPFIRSTORB,
TPLASTORB,TPSTART,TPSTOP,TPALGORITHM,COORDSYS,
SYNOPSTART,SYNOPSTOP)

TABLE FILE = (TAPEID,FILENUM,FLFIRSTORB,FLLASTORB,FLSTART,
FLSTOP,FLALGORITHM,NUMITEMS,FLLEN)

TABLE ITEM = (TAPEID,FILENUM,ITEM,ITSTART,ITSTOP,RECNUM,
ITALGORITHM,ITLEN)

TABLE CAT = (TAPEID,FILENUM,ITEM,CAT,FUNCTION,CATEGORY)

TABLE DESCR = (ITEM,NAME)

L(PCDB): LANGUAGE DEFINITION

T(PCDB): TYPING DEFINITION

TYPE NUMBER = (NUMFILES,TPFIRSTORB,TPLASTORB,FILENUM,
FLFIRSTORB,FLLASTORB,NUMITEMS,FLLEN,RECNUM,
ITLEN) ASCII NUM(22)

TYPE CHAR15 = (TAPEID,MISSION,PROJNUM,COORDSYS) ASCII
CHAR(15)

TYPE CHAR10 = (SENSOR,FORMAT) ASCII CHAR(10)

TYPE CHAR12 = (GENDATE,ARCHIVER,TPSTART,TPSTOP,
SYNOPSTART,SYNOPSTOP,FLSTART,FLSTOP,
ITSTART,ITSTOP) ASCII CHAR(12)

TYPE CHAR8 = (INVDATE) ASCII CHAR(8)

TYPE CHAR5 = (TPALGORITHM,FLALGORITHM,ITEM,ITALGORITHM,
CAT) ASCII CHAR(5)

TYPE CHAR50 = (FUNCTION) ASCII CHAR(50)

TYPE CHAR30 = (CATEGORY,NAME) ASCII CHAR(30)

NONLOGICAL SYMBOLS DEFINITION

PREDICATES

TAPE: (CHAR15,CHAR15,CHAR10,CHAR10,CHAR15,CHAR12,CHAR8,
CHAR12,NUMBER,NUMBER,NUMBER,NUMBER,CHAR12,CHAR12,CHAR5,
CHAR15,CHAR12,CHAR12) - PREDICATE

FILE: (CHAR15,NUMBER,NUMBER,NUMBER,CHAR12,CHAR12,CHAR5,
NUMBER,NUMBER) - PREDICATE

ITEM: (CHAR15,NUMBER,CHAR5,CHAR12,CHAR12,NUMBER,CHAR5,
NUMBER) - PREDICATE

CAT: (CHAR15,NUMBER,CHAR5,CHAR5,CHAR5,CHAR50,CHAR30) - PREDICATE

DESCR: (CHAR5,CHAR30) - PREDICATE

FUNCTIONS

NONE

-15-
C(PCDB): CONSTRAINTS DEFINITION
  C(PCDB,1): CONSTRAINT TAPE: TAPEID -> MISSION, SENSOR, FORMAT, PROJNUM, GENDATE, INVDATE, ARCHIVER, NUMFILES, TPFIRSTORB, TPLASTORB, TSTART, TSTOP, TPALGORITHM, COORDSYS, SYNOPSTART, SYNOPSTOP
  C(PCDB,2): CONSTRAINT FILE: TAPEID, FILENUM -> FLFIRSTORB, FLLASTORB, FLSTART, FLSTOP, FLALGORITHM, NUMITEMS, FLLEN
  C(PCDB,3): CONSTRAINT ITEM: TAPEID, FILENUM, ITEM, RECNUM -> ITSTART, ITSTOP, ITALGORITHM, ITLEN
  C(PCDB,4): CONSTRAINT CAT: TAPEID, FILENUM, ITEM -> CAT, FUNCTION, CATEGORY
  C(PCDB,5): CONSTRAINT DESCRIPT: ITEM -> NAME
Appendix 2.4  V(ERB-R)

V(ERB-R): VIEW DEFINITION
S(ERB-R): SCHEMA DEFINITION
   TABLE TAPEINFO = (NOTAPE, TAPETYPE, TITLE1, TITLE2, TITLE3)
   TABLE FILEINFO = (PB, FILE, NOTAPE)
   TABLE RECINFO = (DATE, TIME, LON, LAT, ALT, ZEN, PB, QUALITY, ELECTR, ILLUMIN, CALIB, SCAN)

L(ERB-R): LANGUAGE DEFINITION
T(ERB-R): TYPING DEFINITION
   TYPE INT4 = (NOTAPE, FILE, DATE, TIME) ASCII INTEGER(4)
   TYPE INT2 = (QUALITY) ASCII INTEGER(2)
   TYPE REAL4 = (PB, LON, LAT, ALT, ZEN) ASCII REAL(4)
   TYPE CHAR4 = (TAPETYPE) ASCII CHAR(4)
   TYPE CHAR115 = (TITLE1, TITLE2, TITLE3) ASCII CHAR(115)
   TYPE CHAR3 = (ELECTR, CALIB, SCAN) ASCII CHAR(3)
   TYPE CHAR8 = (ILLUMIN) ASCII CHAR(8)

NONLOGICAL SYMBOLS DEFINITION
   PREDICATES
      TAPEINFO: (INT4, CHAR4, CHAR115, CHAR115, CHAR115) - PREDICATE
      FILEINFO: (REAL4, INT4, INT4) - PREDICATE
      RECINFO: (INT4, INT4, REAL4, REAL4, REAL4, REAL4, REAL4, INT2, CHAR3, CHAR8, CHAR3, CHAR3) - PREDICATE

   FUNCTIONS
      NONE

C(ERB-R): CONSTRAINTS DEFINITION
   C(ERB-R, 1): CONSTRAINT TAPEINFO: NOTAPE -> TAPETYPE, TITLE1, TITLE2, TITLE3
   C(ERB-R, 2): CONSTRAINT FILEINFO: PB, NOTAPE -> FILE
Appendix 2.5 V(ERB-N)

V(ERB-N): VIEW DEFINITION
S(ERB-N): SCHEMA DEFINITION
  TABLE RTAPE = (NOTAPE, TAPETYPE, PLAYBACK, RTITLE)
  TABLE PLAYBACK = (PB, NOFILE, CATALOG)
  TABLE RTITLE = (TITLE)
  TABLE CATALOG = (LON, LAT, ALT, ZEN, TIME1, QUALITY, ELECTR, ILLUMIN, CALIB, SCAN)
  TABLE RDATES = (DATE1, CATALOG)

L(ERB-N): LANGUAGE DEFINITION
T(ERB-N): TYPING DEFINITION
  TYPE INT4 = (NOTAPE, NOFILE, DATE1, TIME1) EBCDIC INTEGER(4)
  TYPE CHAR4 = (TAPETYPE) EBCDIC CHAR(4)
  TYPE CHAR115 = (TITLE) EBCDIC CHAR(115)
  TYPE REAL4 = (PB, LON, LAT, ALT, ZEN) EBCDIC REAL(4)
  TYPE INT2 = (QUALITY) EBCDIC INTEGER(2)
  TYPE CHAR3 = (ELECTR, CALIB, SCAN) EBCDIC CHAR(3)
  TYPE CHAR8 = (ILLUMIN) EBCDIC CHAR(8)

NONLOGICAL SYMBOLS DEFINITION
PREDICATES
  RDATES-CATALOG: (INT4, CATALOG, REAL4, REAL4, REAL4, REAL4, INT4, INT2, CHAR3, CHAR8, CHAR3, CHAR3, CHAR3) - FULL
  CLUSTER PREDICATE
  RTAPE-PLAYBACK-CATALOG-RTITLE: (INT4, CHAR4, PLAYBACK, REAL4, INT4, CHAR115) - FULL

FUNCTIONS
  TITLE1: (INT4, CHAR115) - FUNCTION
  TITLE2: (INT4, CHAR115) - FUNCTION
  TITLE3: (INT4, CHAR115) - FUNCTION

C(ERB-N): CONSTRAINTS DEFINITION
  C(ERB-N, 1): CONSTRAINT RTAPE: NOTAPE -> TAPETYPE, PLAYBACK, RTITLE
  C(ERB-N, 2): CONSTRAINT RTAPE-PLAYBACK: NOTAPE, PB -> NOFILE, CATALOG
  C(ERB-N, 3): CONSTRAINT RDATES: DATE1 -> CATALOG
Appendix 2.6 \( I(ERB-R; ERB-SEED) \)

\( I(ERB-R; ERB-SEED) \): INTERPRETATION DEFINITION
EXTERNAL VIEW IS \( V(ERB-R) = <S(ERB-R), L(ERB-R), C(ERB-R)> \)
CONCEPTUAL VIEW IS \( V(ERB-SEED) = <S(ERB-SEED), L(ERB-SEED), C(ERB-SEED)> \)

CODING SECTION
CODE FOR INT4 IS INT4;
CODE FOR INT2 IS INT2;
CODE FOR REAL4 IS REAL4;
CODE FOR CHAR4 IS CHAR4;
CODE FOR CHAR15 IS CHAR15;
CODE FOR CHAR3 IS CHAR3;
CODE FOR CHAR8 IS CHAR8;

DEFINING FORMULA SECTION
PREDICATE: TAPEINFO
ARGUMENTS ARE: (NOTAPE:1, TAPETYPE:1, TITLE:1, TITLE:2, TITLE:3)
IS DEFINED BY
(\( E \))RTITLE:1(\( E \))PLAYBACK:1
( RTAPE(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, RTITLE:1) & TITLE:1 = TITLE1(NOTAPE:1) & TITLE:2 = TITLE2(NOTAPE:1) & TITLE:3 = TITLE3(NOTAPE:1) )

PREDICATE: FILEINFO
ARGUMENTS ARE: (PB:1, NOFILE:1, NOTAPE:1)
IS DEFINED BY
(\( E \))TAPETYPE:1(\( E \))PLAYBACK:1(\( E \))CATALOG:1(\( E \))RTITLE:1
RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, NOFILE:1, CATALOG:1, RTITLE:1)

PREDICATE: RECINFO
ARGUMENTS ARE: (DATE1:1, TIME1:1, LON:1, LAT:1, ALT:1, ZEN:1, PB:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1)
IS DEFINED BY
(\( E \))NOTAPE:1(\( E \))TAPETYPE:1(\( E \))PLAYBACK:1(\( E \))NOFILE:1
(\( E \))CATALOG:1(\( E \))RTITLE:1(\( E \))CATALOG:2
( RTAPE-PLAYBACK-CATALOG(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, NOFILE:1, CATALOG:1, LON:1, LAT:1, ALT:1, ZEN:1, TIME1:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1) & RDATES-CATALOG(DATE1:1, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1, TIME1:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1) )

CONSTANT TRANSFORMATION SECTION

/*CODE converts from ASCII to EBCDIC;
DECODE converts from EBCDIC to ASCII.*/

-19-
Appendix 2.7  I(ERB-N;ERB-ORAC)

I(ERB-N;ERB-ORAC): INTERPRETATION DEFINITION
EXTERNAL VIEW IS V(ERB-N) = <S(ERB-N),L(ERB-N),C(ERB-N)>
CONCEPTUAL VIEW IS V(ERB-ORAC) = <S(ERB-ORAC),L(ERB-ORAC),
C(ERB-ORAC)>

CODING SECTION
CODE FOR INT4 IS NUMBER;
CODE FOR INT2 IS NUMBER;
CODE FOR REAL4 IS NUMBER;
CODE FOR CHAR4 IS CHAR4;
CODE FOR CHAR115 IS CHAR115,CHAR!15,CHAR115;
CODE FOR CHAR3 IS CHAR3;
CODE FOR CHAR8 IS CHAR8;
CODE FOR PLAYBACK IS NUMBER;
CODE FOR RTITLE IS NUMBER;
CODE FOR CATALOG IS NUMBER,NUMBER;

DEFINING FORMULA SECTION
PREDICATE: RDATES-CATALOG
ARGUMENTS ARE: (DATE:1,DATE:1,LON:1,ALT:1,ALT:1,TIME:1,
QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,SCAN:1)
IS DEFINED BY
(E)DATE_TIME:1(E)PB:1
RECINFO(DATE_TIME:1,DATE:1,TIME:1,LON:1,ALT:1,ZEN:1,
PB:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,SCAN:1)
PREDICATE: RTAPE-PLAYBACK-CATALOG-RTITLE
ARGUMENTS ARE: (NOTAPE:1,TAPETYPE:1,NOTAPE:1,PB:1,FILE:1,
NOTAPE:1,PB:1,LON:1,ALT:1,ZEN:1,TIME:1,
QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,SCAN:1,
NOTAPE:1,TITLE1:1,TITLE2:1,TITLE3:1)
IS DEFINED BY
(E)DATE_TIME:1(E)DATE:1(E)TITLE2:2(E)TITLE3:2(E)TITLE1:3
(E)TITLE3:3(E)TITLE1:4(E)TITLE2:4
( FILEINFO(PB:1,FILE:1,NOTAPE:1)
& RECINFO(DATE_TIME:1,DATE:1,TIME:1,LON:1,ALT:1,
ZEN:1,PB:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,
SCAN:1)
& ( TAPEINFO(NOTAPE:1,TAPETYPE:1,TITLE1:1,TITLE2:2,
TITLE3:2)
  v TAPEINFO(NOTAPE:1,TAPETYPE:1,TITLE1:3,TITLE2:1,
TITLE3:3)
  v TAPEINFO(NOTAPE:1,TAPETYPE:1,TITLE1:4,TITLE2:4,
TITLE3:1) ) )
FUNCTION: TITLE1
ARGUMENTS ARE: (NOTAPE:1,TITLE1:1,TITLE2:1,TITLE3:1)
IS DEFINED BY
  (E)TAPETYPE:2(E)TITLE2:2(E)TITLE3:2
  TAPEINFO(NOTAPE:1,TAPETYPE:2,TITLE1:1,TITLE2:2,TITLE3:2)
FUNCTION: TITLE2
ARGUMENTS ARE: (NOTAPE:1,TITLE1:1,TITLE2:1,TITLE3:1)
IS DEFINED BY
  (E)TAPETYPE:2(E)TITLE1:2(E)TITLE3:2
  TAPEINFO(NOTAPE:1,TAPETYPE:2,TITLE1:2,TITLE2:1,TITLE3:2)
FUNCTION: TITLES
ARGUMENTS ARE: (NOTAPE:1,TITLE1:1,TITLE2:1,TITLES:1)
IS DEFINED BY
  (E)TAPETYPE:2(E)TITLE1:2(E)TITLE2:2
  TAPEINFO(NOTAPE:1,TAPETYPE:2,TITLE1:2,TITLE2:2,TITLES:1)

CONSTANT TRANSFORMATION SECTION

/*CODE converts -from EBCDIC to ASCII -
Additionally, CODE multiplies LON, LAT, ALT, and
ZEN entries by 100 and PB entries by 10;
DECODE converts -from ASCII to EBCDIC -
Additionally, DECODE divides LON, LAT, ALT, and
ZEN entries by 100 and PB entries by 10.*/
Appendix 2.8 V(GLOBE)

V(GLOBE): VIEW DEFINITION
S(GLOBE): SCHEMA DEFINITION

TABLE TAPEINFO = (NOTAPE,TAPETYPE,TITLE1,TITLE2,TITLE3)
TABLE FILEINFO = (PB,FILE,NOTAPE)
TABLE RECINFO = (DATE_TIME,DATE,TIME,LON,LAT,ALT,ZEN,PB,
QUALITY,ELECTR,ILLUMIN,CALIB,SCAN)
TABLE RTAPE = (NOTAPE,TAPETYPE,PLAYBACK,RTITLE)
TABLE PLAYBACK = (PB,FILE,CATALOG)
TABLE RTITLE = (TITLE)
TABLE CATALOG = (LON,LAT,ALT,ZEN,TIME,QUALITY,ELECTR,
ILLUMIN,CALIB,SCAN)
TABLE RDATES = (DATE,CATALOG)
TABLE RLON = (ILON,CATALOG)
TABLE RLAT = (ILAT,CATALOG)
TABLE RALT = (IALT,CATALOG)
TABLE RZEN = (IZEN,CATALOG)
TABLE TAPE = (TAPEID,MISSION,SENSOR,FORMAT,PROJNUM,GENDATE,
INVDATE,ARCHIVER,NUMFILES,TPFIRSTORB,
TPLASTORB,TPSTART,TPSTOP,TPALGORITHM,COORDSYS,
SYNOPSTART,SYNOPSTOP)
TABLE FILE = (TAPEID,FILE,FLFIRSTORB,FLLASTORB,FLSTART,
FLSTOP,FLALGORITHM,NUMITEMS,FLLEN)
TABLE ITEM = (TAPEID,FILE,ITEM,ITSTART,ITSTOP,RECNUM,
ITALGORITHM,ITLEN)
TABLE CAT = (TAPEID,FILE,ITEM,CAT,FUNCTION,CATEGORY)
TABLE DESCR = (ITEM,NAME)

L(GLOBE): LANGUAGE DEFINITION
T(GLOBE): TYPING DEFINITION

TYPE NUMBER = (NOTAPE,PB,FILE,DATE_TIME,DATE,TIME,LON,
LAT,ALT,ZEN,ILON,ILAT,IALT,IZEN,QUALITY,
NUMFILES,TPFIRSTORB,TPLASTORB,TPFIRSTORB,
TPLASTORB,NUMITEMS,FLLEN,RECNUM,ITLEN)
ASCII NUM(22)

TYPE CHAR4 = (TAPETYPE) ASCII CHAR(4)
TYPE CHAR115 = (TITLE1,TITLE2,TITLE3,TITLE) ASCII CHAR(115)
TYPE CHAR3 = (ELECTR,CALIB,SCAN) ASCII CHAR(3)
TYPE CHAR8 = (ILLUMIN,INVDATE) ASCII CHAR(8)
TYPE CHAR15 = (TAPEID,MISSION,PROJNUM,COORDSYS) ASCII CHAR(15)
TYPE CHAR10 = (SENSOR,FORMAT) ASCII CHAR(10)
TYPE CHAR12 = (GENDATE,ARCHIVER,TPSTART,TPSTOP,
SYNOPSTART,SYNOPSTOP,FLSTART,FLSTOP,
ITSTART,ITSTOP) ASCII CHAR(12)
TYPE CHAR5 = (TPALGORITHM,FLALGORITHM,ITEM,ITALGORITHM,
CAT) ASCII CHAR(5)
TYPE CHAR50 = (FUNCTION) ASCII CHAR(50)
TYPE CHAR30 = (CATEGORY,NAME) ASCII CHAR(30)
NONLOGICAL SYMBOLS DEFINITION

PREDICATES

TAPEINFO: (NUMBER, CHAR4, CHAR115, CHAR115, CHAR115) - PREDICATE
FILEINFO: (NUMBER, NUMBER, NUMBER) - PREDICATE
RECINFO: (NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3) - PREDICATE
RDATES-CATALOG: (NUMBER, CATALOG, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RLON-CATALOG: (NUMBER, CATALOG, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RLAT-CATALOG: (NUMBER, CATALOG, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RALT-CATALOG: (NUMBER, CATALOG, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RZEN-CATALOG: (NUMBER, CATALOG, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3) - FULL CLUSTER PREDICATE
RTAPE-PLAYBACK-CATALOG-RTITLE: (NUMBER, CHAR4, PLAYBACK, NUMBER, NUMBER, CATALOG, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3, RTITLE, CHAR115) - FULL CLUSTER PREDICATE

TAPE: (CHAR15, CHAR15, CHAR10, CHAR10, CHAR15, CHAR12, CHAR8, CHAR12, NUMBER, NUMBER, NUMBER, CHAR15, CHAR12, CHAR12) - PREDICATE
FILE: (CHAR15, NUMBER, NUMBER, NUMBER, CHAR12, CHAR12, CHAR5, NUMBER, NUMBER) - PREDICATE
ITEM: (CHAR15, NUMBER, NUMBER, NUMBER, CHAR12, CHAR12, CHAR5, NUMBER, NUMBER) - PREDICATE
CAT: (CHAR15, NUMBER, NUMBER, CHAR5, CHAR5, CHAR50, CHAR30) - PREDICATE

DESCRIPTORS

FUNCTIONS

CONC: (NUMBER, NUMBER, NUMBER) - PREDEFINED FUNCTION
TITLE1: (NUMBER; CHAR115) - FUNCTION
TITLE2: (NUMBER; CHAR115) - FUNCTION
TITLE3: (NUMBER; CHAR115) - FUNCTION
NUM: (CHAR15; NUMBER) - PREDEFINED FUNCTION
C(GLOBE): CONSTRAINTS DEFINITION
C(GLOBE,1): CONSTRAINT TAPEINFO: NOTAPE -> TAPETYPE, TITLE1, TITLE2, TITLE3
C(GLOBE,2): CONSTRAINT FILEINFO: PB, NOTAPE -> FILE
C(GLOBE,3): CONSTRAINT RECINFO: DATE_TIME -> DATE, TIME, LON, LAT, ALT, ZEN, PB, QUALITY, ELECTR, ILLUMIN, CALIB, SCAN
C(GLOBE,4): CONSTRAINT RECINFO: DATE_TIME = CONC(DATE, TIME)
C(GLOBE,5): CONSTRAINT RTAPE: NOTAPE -> TAPETYPE, PLAYBACK, RTITLE
C(GLOBE,6): CONSTRAINT RTAPE-PLAYBACK: NOTAPE, PB -> FILE, CATALOG
C(GLOBE,7): CONSTRAINT RDATES: DATE -> CATALOG
C(GLOBE,8): CONSTRAINT RLON: ILON -> CATALOG
C(GLOBE,9): CONSTRAINT RLAT: ILAT -> CATALOG
C(GLOBE,10): CONSTRAINT RALT: IALAT -> CATALOG
C(GLOBE,11): CONSTRAINT RZEN: IZEN -> CATALOG
C(GLOBE,12): CONSTRAINT RLON-CATALOG: ILON=INT(LON)
C(GLOBE,13): CONSTRAINT RLAT-CATALOG: ILAT=INT(LAT)
C(GLOBE,14): CONSTRAINT RALT-CATALOG: IALAT=INT(ALT)
C(GLOBE,15): CONSTRAINT RZEN-CATALOG: IZEN=INT(ZEN)
C(GLOBE,16): CONSTRAINT NOTAPE IN TAPEINFO -> NOTAPE NOT IN RTAPE
C(GLOBE,17): CONSTRAINT NOTAPE, PB IN FILEINFO -> NOTAPE, PB NOT IN RTAPE-PLAYBACK
C(GLOBE,18): CONSTRAINT TAPE: TAPEID -> MISSION, SENSOR, FORMAT, PROJNUM, GENDATE, INVDATE, ARCHIVER, NUMFILES, TPFIRSTORB, TPALASTORB, TPSTART, TPSTOP, TPALGORITHM, COORDSYS, SYNOPSTART, SYNOPSTOP
C(GLOBE,19): CONSTRAINT FILE: TAPEID, FILE -> FLFIRSTORB, FLLASTORB, FLSTART, FLSTOP, FLALGORITHM, NUMITEMS, FLEN
C(GLOBE,20): CONSTRAINT ITEM: TAPEID, FILE, ITEM, RECNUM -> ITSTART, ITSTOP, ITALGORITHM, ITLEN
C(GLOBE,21): CONSTRAINT CAT: TAPEID, FILE, ITEM -> CAT, FUNCTION, CATEGORY
C(GLOBE,22): CONSTRAINT DESCR: ITEM -> NAME

-24-
Appendix 2.9 \( V(GLOBE-R) \)

\( V(GLOBE-R) \): VIEW DEFINITION

\( S(GLOBE-R) \): SCHEMA DEFINITION

| TABLE | TAPEINFO = (NOTAPE, TAPETYPE, TITLE1, TITLE2, TITLE3) |
| TABLE | FILEINFO = (PB, FILE, NOTAPE) |
| TABLE | RECINFO = (DATE, TIME, LON, LAT, ALT, ZEN, PB, QUALITY, ELECTR, ILLUMIN, CALIB, SCAN) |
| TABLE | TAPE = (TAPEID, MISSION, SENSOR, FORMAT, PROJNUM, GENDATE, INVDATE, ARCHIVER, NUMFILES, TPFIRSTORB, TPLASTORB, TPSTART, TPSTOP, TPFIRSTORB, TPLASTORB, TPSTART, TPSTOP, TPALGORITHM, COORDSYS, SYNOPSTART, SYNOPSTOP) |
| TABLE | FILE = (TAPEID, FILE, FLFIRSTORB, FLLASTORB, FLSTART, FLSTOP, FLAGORITHM, NUMITEMS, FLLEN) |
| TABLE | ITEM = (TAPEID, FILE, ITEM, ITSTART, ITSTOP, RECNUM, ITALGORITHM, ITLEN) |
| TABLE | CAT = (TAPEID, FILE, ITEM, CAT, FUNCTION, CATEGORY) |
| TABLE | DESCR = (ITEM, NAME) |

\( L(GLOBE-R) \): LANGUAGE DEFINITION

\( T(GLOBE-R) \): TYPING DEFINITION

| TYPE | NUMBER = (NOTAPE, PB, FILE, DATE, TIME, LON, LAT, ALT, ZEN, QUALITY, NUMFILES, TPFIRSTORB, TPLASTORB, FLFIRSTORB, FLLASTORB, NUMITEMS, FLLEN, RECNUM, ITLEN) ASCII NUM(22) |
| TYPE | CHAR4 = (TAPETYPE) ASCII CHAR(4) |
| TYPE | CHAR115 = (TITLE1, TITLE2, TITLE3) ASCII CHAR(115) |
| TYPE | CHAR3 = (ELECTR, CALIB, SCAN) ASCII CHAR(3) |
| TYPE | CHAR8 = (ILLUMIN, INVDATE) ASCII CHAR(8) |
| TYPE | CHAR15 = (TAPEID, MISSION, PROJNUM, COORDSYS) ASCII CHAR(15) |
| TYPE | CHAR10 = (SENSOR, FORMAT) ASCII CHAR(10) |
| TYPE | CHAR12 = (GENDATE, ARCHIVER, TPSTART, TPSTOP, SYNOPSTART, SYNOPSTOP, FLSTART, FLSTOP, ITSTART, ITSTOP) ASCII CHAR(12) |
| TYPE | CHAR5 = (TPALGORITHM, FLALGORITHM, ITEM, ITALGORITHM, CAT) ASCII CHAR(5) |
| TYPE | CHAR50 = (FUNCTION) ASCII CHAR(50) |
| TYPE | CHAR30 = (CATEGORY, NAME) ASCII CHAR(30) |
NONLOGICAL SYMBOLS DEFINITION

PREDICATES

TAPEINFO: (NUMBER, CHAR4, CHAR15, CHAR15, CHAR15, CHAR15) - PREDICATE
FILEINFO: (NUMBER, NUMBER, NUMBER) - PREDICATE
RECINFO: (NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, NUMBER, CHAR3, CHAR8, CHAR3, CHAR3) - PREDICATE
TAPE: (CHAR15, CHAR15, CHAR15, CHAR10, CHAR10, CHAR15, CHAR12, CHAR8, CHAR12, NUMBER, NUMBER, NUMBER, CHAR12, CHAR12, CHAR12, CHAR12, CHAR12) - PREDICATE
FILE: (CHAR15, NUMBER, NUMBER, NUMBER, CHAR12, CHAR12, NUMBER, NUMBER) - PREDICATE
ITEM: (CHAR15, NUMBER, CHAR5, CHAR12, CHAR12, NUMBER, CHAR5, NUMBER) - PREDICATE
CAT: (CHAR15, NUMBER, CHAR5, CHAR5, CHAR5, CHAR50, CHAR30) - PREDICATE
DESCR: (CHAR5, CHAR30) - PREDICATE

FUNCTIONS

NUM: (CHAR15; NUMBER) - PREDEFINED FUNCTION

C(GLOBE-R): CONSTRAINTS DEFINITION

C(GLOBE-R,1): CONSTRAINT TAPEINFO: NOTAPE -> TAPETYPE, TITLE1, TITLE2, TITLE3
C(GLOBE-R,2): CONSTRAINT FILEINFO: PB, NOTAPE -> FILE
C(GLOBE-R,3): CONSTRAINT TAPE: TAPEID -> MISSION, SENSOR, FORMAT, PROJNUM, GENDATE, INVDATE, ARCHIVER, NUMFILES, TFIRSTORB, TPLASTORB, TPSTART, TPSTOP, TPALGORITHM, COORDSYS, SYNOPSTART, SYNOPSTOP
C(GLOBE-R,4): CONSTRAINT FILE: TAPEID, FILE -> FLFIRSTORB, FLASTORB, FLSTART, FLSTOP, FLALGORITHM, NUMITEMS, FLLEN
C(GLOBE-R,5): CONSTRAINT ITEM: TAPEID, FILE, ITEM, RECNUM -> ITSTART, ITSTOP, ITALGORITHM, ITLEN
C(GLOBE-R,6): CONSTRAINT CAT: TAPEID, FILE, ITEM -> CAT, FUNCTION, CATEGORY
C(GLOBE-R,7): CONSTRAINT DESCR: ITEM -> NAME

-26-
Appendix 2.10  I(GLOBE-R;GLOBE)

I(GLOBE-R;GLOBE): INTERPRETATION DEFINITION
EXTERNAL VIEW IS V(GLOBE-R) = <S(GLOBE-R), L(GLOBE-R),
C(GLOBE-R)>
CONCEPTUAL VIEW IS V(GLOBE) = <S(GLOBE), L(GLOBE), C(GLOBE)>
CODING SECTION
CODE FOR NUMBER IS NUMBER;
CODE FOR CHAR4 IS CHAR4;
CODE FOR CHAR115 IS CHAR115;
CODE FOR CHAR3 IS CHAR3;
CODE FOR CHAR8 IS CHAR8;
CODE FOR CHAR15 IS CHAR15;
CODE FOR CHAR10 IS CHAR10;
CODE FOR CHAR12 IS CHAR12;
CODE FOR CHAR5 IS CHAR5;
CODE FOR CHAR50 IS CHAR50;
CODE FOR CHAR30 IS CHAR30;

DEFINING FORMULA SECTION
PREDICATE: TAPEINFO
ARGUMENTS ARE: (NOTAPE:1, TAPETYPE:1, TITLE:1, TITLE:2, TITLE:3)
IS DEFINED BY
TAPEINFO(NOTAPE:1, TAPETYPE:1, TITLE:1, TITLE:2, TITLE:3)
\[ \land (RTAPE(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, RTITLE:1)
\land TITLE:1 = TITLE1(NOTAPE:1)
\land TITLE:2 = TITLE2(NOTAPE:1)
\land TITLE:3 = TITLE3(NOTAPE:1) \]
PREDICATE: FILEINFO
ARGUMENTS ARE: (PB:1, FILE:1, NOTAPE:1)
IS DEFINED BY
FILEINFO(PB:1, FILE:1, NOTAPE:1)
\[ \land (RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1,
FILE:1, CATALOG:1, RTITLE:1) \]
PREDICATE: RECINFO
ARGUMENTS ARE: (DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1, PB:1, QUALITY:i, ELECTR:i, ILLUMIN:i, CALIB:1, SCAN:1)
IS DEFINED BY
(E) DATE_TIME:1
RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1, PB:1, QUALITY:i, ELECTR:i, ILLUMIN:i, CALIB:1, SCAN:1)
\(\land \) RDATES-CATALOG(DATE:1, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1, TIME:1, QUALITY:i, ELECTR:i, ILLUMIN:i, CALIB:1, SCAN:1, RTITLE:1)

PREDICATE: TAPE
ARGUMENTS ARE: (TAPEID:1, MISSION:1, SENSOR:1, FORMAT:1, PROJNUM:1, GENDATE:1, INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1, COORDSYS:1, SYNOPSTART:i, SYNOPSTOP:i)
IS DEFINED BY
TAPE(TAPEID:1, MISSION:1, SENSOR:1, FORMAT:1, PROJNUM:1, GENDATE:1, INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1, COORDSYS:1, SYNOPSTART:i, SYNOPSTOP:i)

PREDICATE: FILE
ARGUMENTS ARE: (TAPEID:1, FILE:1, FLFIRSTORB:1, FLLASTORB:1, FLSTART:1, FLSTOP:1, FLALGORITHM:1, NUMITEMS:1, FLLEN:1)
IS DEFINED BY
FILE(TAPEID:1, FILE:1, FLFIRSTORB:1, FLLASTORB:1, FLSTART:1, FLSTOP:1, FLALGORITHM:1, NUMITEMS:1, FLLEN:1)

PREDICATE: ITEM
ARGUMENTS ARE: (TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, ITALGORITHM:1, ITLEN:1)
IS DEFINED BY
ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, ITALGORITHM:1, ITLEN:1)

PREDICATE: CAT
ARGUMENTS ARE: (TAPEID:1, FILE:1, ITEM:1, CAT:1, FUNCTION:1, CATEGORY:1)
IS DEFINED BY
CAT(TAPEID:1, FILE:1, ITEM:1, CAT:1, FUNCTION:1, CATEGORY:1)

PREDICATE: DESCR
ARGUMENTS ARE: (ITEM:1, NAME:1)
IS DEFINED BY
DESCR(ITEM:1, NAME:1)
FUNCTION: NUM
ARGUMENTS ARE: (TAPEID:1)
   IS DEFINED BY
      NUM(TAPEID:1)

CONSTANT TRANSFORMATION SECTION

   /*CODE and DECODE are the identity maps.*/
Appendix 3.1
Logical optimization algorithm for a conjunctive query

Begin
Place the query in prenex conjunctive normal form: Q(0);
Set i := 0;
For j := 1 to n Do
  If Pj is an equality that gives the value of a function as an existentially quantified variable that does not appear in any other conjunct Then
    Begin
      Obtain Q(i+1) from Q(i) as follows:
      Begin
        Delete Pj;
        Omit the superfluous quantifiers from the prefix
      End;
      Set i := i + 1
    End;
  End;
While a functional dependency constraint may be applied to identify different objects in Q(i) Do
Begin
  Pick an applicable functional dependency constraint FD;
  Set S := the set of objects identified by FD;
  Set a := the total number of target variables in S;
  Set b := the total number of constants in S;
  If a+b=0 Then
    Obtain Q(i+1) from Q(i) as follows:
    Begin
      Pick one object from S;
      Change all objects in S to the chosen object;
      Omit the superfluous quantifiers from the prefix
    End
  Else
    If a+b=1 Then
      Obtain Q(i+1) from Q(i) as follows:
      Begin
        Change all existentially quantified variables in S to the target variable or constant in S;
        Omit the superfluous quantifiers from the prefix
      End
    Else
      \[
      \text{End} 
      \]
Else
   If b=0
      Then
      Obtain Q(i+1) from Q(i) as follows:
      Begin
      Pick one target variable, say X, in S;  
      For every other target variable, say Y, in 
      S, add the conjunct Y = X;  
      Change all other variables in S to X 
      except in the just added equality 
      conjuncts; 
      Omit the superfluous quantifiers from the 
      prefix 
      End
      Else
      If b=1
      Then
      Obtain Q(i+1) from Q(i) as follows:
      Begin
      For every target variable, say Y, in S  
      add the conjunct Y = c  
      (c is the constant in S); 
      Change all the variables in S to c  
      except in the just added equality 
      conjuncts; 
      Omit the superfluous quantifiers from 
      the prefix 
      End
      Else Begin
      Print "The query has no answers";
      Exit procedure
      End;
      Set i := i+1
      End;
End;
Let P := the set of pairs of conjuncts <C1,C2> where C1 is 
structurally a subcluster of C2;
While P≠empty Do
Begin
Pick a pair <C1,C2> from P;
Delete <C1,C2> from P;
Obtain C1' from C1 as follows:
   Rename all existentially quantified variables in C1 to 
   match the corresponding objects in C2 (if possible);
If $C_1'$ is a subcluster of $C_2$ and every existentially quantified variable which was renamed to a target variable or a constant does not appear in another conjunct
Then
Begin
Obtain $Q(i+1)$ from $Q(i)$ as follows:
Begin
Delete the conjunct $C_1$;
Rename all existentially quantified variables according to the formation of $C_1'$ from $C_1$;
Omit the superfluous quantifiers from the prefix
End;
Delete every pair from $P$ which includes $C_1$;
Set $i := i+1$
End
End
End.
Appendix 3.2

Query 1 transformation from ERB-R to ERB-SEED and optimization

Query 1 in English:
What are the tape numbers, the tape types, the playback numbers, and the file numbers in the data inventory?

Query 1 in DBL for ERB-R:
GET W(NOTAPE:1,TAPETYPE:1,PB:1,FILE:1):
    (E)TITLE1:1(E)TITLE2:1(E)TITLE3:1
    ( TAPEINFO(NOTAPE:1,TAPETYPE:1,TITLE1:1,TITLE2:1,TITLE3:1)
      & FILEINFO(PB:1,FILE:1,NOTAPE:1) )

Interpreted query 1 in DBL for ERB-SEED:
GET W(DECODE(NOTAPE:1),DECODE(TAPETYPE:1),DECODE(PB:1),
      DECODE(NOFILE:1)):
    (E)TITLE:1 (E)TITLE:2 (E)TITLE:3
    ( (E)RTITLE:i (E)PLAYBACK:i
      ( RTAPE(NOTAPE:i,TAPETYPE:i,PLAYBACK:i,RTITLE:i)
        & TITLE:i = TITLE1(NOTAPE:i) & TITLE:i = TITLE2(NOTAPE:i) & TITLE:i = TITLE3(NOTAPE:i) ) &
      (E)TAPETYPE:2 (E)PLAYBACK:2 (E)CATALOG:i (E)RTITLE:2
      RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:2,PLAYBACK:2,PB:1,
        NOFILE:1,CATALOG:i,RTITLE:2) )

Place interpreted query 1 in prenex conjunctive normal form — Obtain Q(0):
GET W(DECODE(NOTAPE:1),DECODE(TAPETYPE:1),DECODE(PB:1),
      DECODE(NOFILE:1)):
    (E)TITLE:1 (E)TITLE:2 (E)TITLE:3 (E)RTITLE:1 (E)PLAYBACK:1
    (E)TAPETYPE:2 (E)PLAYBACK:2 (E)CATALOG:1 (E)RTITLE:2
    ( RTAPE(NOTAPE:1,TAPETYPE:1,PLAYBACK:1,RTITLE:1)
      & TITLE:1 = TITLE1(NOTAPE:1) & TITLE:2 = TITLE2(NOTAPE:1) & TITLE:3 = TITLE3(NOTAPE:1) ) &
    (E)RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:2,PLAYBACK:2,PB:1,
      NOFILE:1,CATALOG:1,RTITLE:2) )

Delete the first equality — Obtain Q(1):
GET W(DECODE(NOTAPE:1),DECODE(TAPETYPE:1),DECODE(PB:1),
      DECODE(NOFILE:1)):
    (E)TITLE:2 (E)TITLE:3 (E)RTITLE:1 (E)PLAYBACK:1 (E)TAPETYPE:2
    (E)PLAYBACK:2 (E)CATALOG:1 (E)RTITLE:2
    ( RTAPE(NOTAPE:1,TAPETYPE:1,PLAYBACK:1,RTITLE:1)
      & TITLE:2 = TITLE2(NOTAPE:1) & TITLE:3 = TITLE3(NOTAPE:1) &
    (E)RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:2,PLAYBACK:2,PB:1,
      NOFILE:1,CATALOG:1,RTITLE:2) )
Delete the next equality —

Obtain Q(2):

GET W(DECODE(NOTAPE:1), DECODE(TAPETYPE:1), DECODE(PB:1),
DECODE(NOFILE:1)):
(E) TITLE:3 (E) RTITLE:1 (E) PLAYBACK:1 (E) TAPETYPE:2 (E) PLAYBACK:2
(E) CATALOG:1 (E) RTITLE:2
  ( RTAPE(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, RTITLE:1)
  & TITLE:3 = TITLE3(NOTAPE:1)
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:2, PLAYBACK:2, PB:1,
  NOFILE:1, CATALOG:1, RTITLE:2) )

Delete the last equality —

Obtain Q(3):

GET W(DECODE(NOTAPE:1), DECODE(TAPETYPE:1), DECODE(PB:1),
DECODE(NOFILE:1)):
(E) RTITLE:1 (E) PLAYBACK:1 (E) TAPETYPE:2 (E) PLAYBACK:2
(E) CATALOG:1 (E) RTITLE:2
  ( RTAPE(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, RTITLE:1)
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:2, PLAYBACK:2, PB:1,
  NOFILE:1, CATALOG:1, RTITLE:2) )

Apply C(ERB-SEED,1) to identify TAPETYPE —

Obtain Q(4):

GET W(DECODE(NOTAPE:1), DECODE(TAPETYPE:1), DECODE(PB:1),
DECODE(NOFILE:1)):
(E) RTITLE:1 (E) PLAYBACK:1 (E) TAPETYPE:2 (E) PLAYBACK:2
(E) CATALOG:1 (E) RTITLE:2
  ( RTAPE(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, RTITLE:1)
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:2, PLAYBACK:2, PB:1,
  NOFILE:1, CATALOG:1, RTITLE:2) )

Apply C(ERB-SEED,1) to identify PLAYBACK —

Obtain Q(5):

GET W(DECODE(NOTAPE:1), DECODE(TAPETYPE:1), DECODE(PB:1),
DECODE(NOFILE:1)):
(E) RTITLE:1 (E) PLAYBACK:1 (E) PLAYBACK:2 (E) CATALOG:1 (E) RTITLE:2
  ( RTAPE(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, RTITLE:1)
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1,
  NOFILE:1, CATALOG:1, RTITLE:2) )

Apply C(ERB-SEED,1) to identify RTITLE —

Obtain Q(6):

GET W(DECODE(NOTAPE:1), DECODE(TAPETYPE:1), DECODE(PB:1),
DECODE(NOFILE:1)):
(E) RTITLE:1 (E) PLAYBACK:1 (E) CATALOG:1
  ( RTAPE(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, RTITLE:1)
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1,
  NOFILE:1, CATALOG:1, RTITLE:2) )
Delete subsumed conjunct –

Obtain Q(7) (the optimized query):

\[
\text{GET W(DECODE(NOTAPE:1), DECODE(TAPETYPE:1), DECODE(PB:1), DECODE(NOFILE:1))}
\]

\[
\text{(E)RTITLE:1(E)PLAYBACK:1(E)CATALOG:1}
\]

\[
\text{RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, NOFILE:1, CATALOG:1, RTITLE:1)}
\]
Appendix 3.3
Query 2 transformation from ERB-R to ERB-SEED and optimization

Query 2 in English:
What are the dates, the times, and the illuminations for all
the data in the inventory where the playback number is 81261.4
and is associated with some file number?

Query 2 in DBL for ERB-R:
GET W(DATE:1,TIME:1,ILLUMIN:1):
(E)FILE:1(E)NOTAPE:1(E)LON:1(E)LAT:1(E)ALT:1(E)ZEN:1
(E)QUALITY:1(E)ELECTR:1(E)CALIB:1(E)SCAN:1
(FILEINFO(81261.4,FILE:1,NOTAPE:1)
& RECINFO(DATE:1,TIME:1,LON:1,LAT:1,ALT:1,ZEN:1,81261.4,
QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,SCAN:1)-

Interpreted query 2 in DBL for ERB-SEED:
GET W(DECODE(DATE1:1),DECODE(TIME1:1),DECODE(ILLUMIN:1)):
(E)NOFILE:1(E)NOTAPE:1(E)LON:1(E)LAT:1(E)ALT:1(E)ZEN:1
(E)QUALITY:1(E)ELECTR:1(E)CALIB:1(E)SCAN:1
(E)TAPETYPE:1(E)PLAYBACK:1(E)CATALOG:1(E)RTITLE:1
RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:1,PLAYBACK:1,81261.4,
NOFILE:1,CATALOG:1,RTITLE:1)
& (E)NOTAPE:2(E)TAPETYPE:2(E)PLAYBACK:2(E)NOFILE:2
(E)CATALOG:2(E)RTITLE:2(E)CATALOG:3
(RTAPE-PLAYBACK-CATALOG(NOTAPE:2,TAPETYPE:2,PLAYBACK:2,
81261.4,NOFILE:2,CATALOG:2,LON:1,LAT:1,ALT:1,ZEN:1,
TIME:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,
SCAN:1,RTITLE:2)
& RDATES-CATALOG(DATE1:1,CATALOG:3,LON:1,LAT:1,ALT:1,ZEN:1,
TIME:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,
SCAN:1)

Place interpreted query 2 in prenex conjunctive normal form —
Get Q(0):
GET W(DECODE(DATE1:1),DECODE(TIME1:1),DECODE(ILLUMIN:1)):
(E)NOFILE:1(E)NOTAPE:1(E)LON:1(E)LAT:1(E)ALT:1(E)ZEN:1
(E)QUALITY:1(E)ELECTR:1(E)CALIB:1(E)SCAN:1(E)TAPETYPE:1
(E)PLAYBACK:1(E)CATALOG:1(E)RTITLE:1(E)NOTAPE:2(E)TAPETYPE:2
(E)PLAYBACK:2(E)NOFILE:2(E)CATALOG:2(E)RTITLE:2(E)CATALOG:3
(RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:1,PLAYBACK:1,81261.4,
NOFILE:1,CATALOG:1,RTITLE:1)
& RTAPE-PLAYBACK-CATALOG(NOTAPE:2,TAPETYPE:2,PLAYBACK:2,
81261.4,NOFILE:2,CATALOG:2,LON:1,LAT:1,ALT:1,ZEN:1,
TIME:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,
SCAN:1,RTITLE:2)
& RDATES-CATALOG(DATE1:1,CATALOG:3,LON:1,LAT:1,ALT:1,ZEN:1,
TIME:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,
SCAN:1)

-36-
Rename variables and delete subsumed conjunct -
Obtain Q(1) (the optimized query):

GET W(DECODE(DATE1:1), DECODE(TIME1:1), DECODE(ILLUMIN:1)):
  (E)LON:1 (E)LAT:1 (E)ALT:1 (E)ZEN:1 (E)QUALITY:1 (E)ELECTR:1
  (E)CALIB:1 (E)SCAN:1 (E)NOTAPE:2 (E)TAPETYPE:2 (E)PLAYBACK:2
  (E)NOFILE:2 (E)CATALOG:2 (E)RTITLE:2 (E)CATALOG:3
  (RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2,
    81261.4, NOFILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1,
    TIME1:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1,
    RTITLE:2))

& RDATES-CATALOG(DATE1:1, CATALOG:3, LON:1, LAT:1, ALT:1, ZEN:1,
    TIME1:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1,
    SCAN:1) )
Appendix 3.4
Logical optimization for a query with local disjunctions

Begin
Place the query in prenex conjunctive normal form: \( Q(0) \); 
Set \( i := 0 \); 
For \( j := 1 \) to \( n \) Do
  If \( P_j \) is an equality that gives the value of a function as an
  existentially quantified variable that does not appear in
  any other conjunct
  Then
    Begin
      Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
      Begin
        Delete \( P_j \);
        Omit the superfluous quantifiers from the prefix
      End;
      Set \( i := i+1 \)
    End 
  Else
    Begin
      Set \( H_j := \) the set of pairs of disjuncts \( <D_1,D_2> \) in \( P_j \)
      such that \( D_1 \) is structurally a subcluster
      of \( D_2 \)
      While \( H_j \neq \) empty Do
        Begin
          Pick a pair \( <D_1,D_2> \) from \( H_j \);
          Delete \( <D_1,D_2> \) from \( H_j \);
          Obtain \( D_2' \) from \( D_2 \) as follows:
            Rename all objects in \( D_2 \) to match the
            corresponding existentially quantified
            variables in \( D_1 \) (if possible);
          If \( D_1 \) is a subcluster of \( D_2' \)
            Then
              Begin
                Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
                  Begin
                    Delete the disjunct \( D_2 \) from \( P_j \);
                    Omit the superfluous quantifiers from the
                    prefix
                  End;
                  Delete every pair from \( H_j \) which includes \( D_2 \);
                Set \( i := i+1 \)
              End
        End 
    End;
While a functional dependency constraint may be applied to conjuncts which are clusters to identify different objects in $Q(i)$, do

Begin
Pick an applicable functional dependency constraint $FD$;
Set $S :=$ the set of objects identified by $FD$;
Set $a :=$ the total number of target variables in $S$;
Set $b :=$ the total number of constants in $S$;
If $a+b=0$
Then
Obtain $Q(i+1)$ from $Q(i)$ as follows:
Begin
Pick one object from $S$;
Change all objects in $S$ to the chosen object;
Omit the superfluous quantifiers from the prefix
End
Else
If $a+b=1$
Then
Obtain $Q(i+1)$ from $Q(i)$ as follows:
Begin
Change all existentially quantified variables in $S$ to the target variable or constant in $S$;
Omit the superfluous quantifiers from the prefix
End
Else
If $b=0$
Then
Obtain $Q(i+1)$ from $Q(i)$ as follows:
Begin
Pick one target variable, say $X$, in $S$;
For every other target variable, say $Y$, in $S$, add the conjunct $Y = X$;
Change all other variables in $S$ to $X$ except in the just added equality conjuncts;
Omit the superfluous quantifiers from the prefix
End
Else
  If b=1
    Then
      Obtain Q(i+1) from Q(i) as follows:
      Begin
        For every target variable, say Y, in S
        add the conjunct Y = c
        (c is the constant in S);
        Change all the variables in S to c
        except in the just added equality
        conjuncts;
        Omit the superfluous quantifiers from
        the prefix
      End
    Else Begin
      Print "The query has no answers";
      Exit procedure
    End;
  Set i := i+1
End;
Let P := the set of pairs of conjuncts <C1,C2> where C1 is
structurally a subcluster of C2;
While P is not empty Do
  Begin
    Pick a pair <C1,C2> from P;
    Delete <C1,C2> from P;
    Obtain C1' from C1 as follows:
    Rename all existentially quantified variables in C1 to
    match the corresponding objects in C2 (if possible);
    If C1' is a subcluster of C2 and every existentially
    quantified variable which was renamed to a target
    variable or a constant does not appear in another
    conjunct
    Then
      Begin
        Obtain Q(i+1) from Q(i) as follows:
        Begin
          Delete the conjunct C1;
          Rename all existentially quantified variables
          according to the formation of C1' from C1;
          Omit the superfluous quantifiers from the prefix
        End;
        Delete every pair from P which includes C1;
        Set i := i+1
      End
    End
  End
End.
Appendix 3.5
Query 3 transformation from ERB-N to ERB-ORAC and optimization

Query:
For tape number = 1003 what are the tape types, the playback numbers and the file numbers?

DBL Query:
GET W(TAPETYPE:1, PB:1, NOFILE:1):
E PLAYBACK:1, RTITLE:1, CATALOG:1
RTAPE-PLAYBACK(1003, TAPETYPE:1, PLAYBACK:1, PB:1, NOFILE:1, CATALOG:1, RTITLE:1)

Interpreted DBL Query (already in normal form) -
Obtain Q(0):
GET W(DECODE(TAPETYPE:1), DECODE(PB:1), DECODE(FILE:1)):
E DATE_TIME:1, DATE:1, LON:1, LAT:1, ALT:1, ZEN:1
E TIME:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1
E SCAN:1, TITLE1:1, TITLE2:1, TITLE3:1
E TITLE3:2, TITLE1:3, TITLE3:3, TITLE1:4, TITLE2:4
(FILEINFO(PB:1, FILE:1, 1003)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1, PB:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1)
& (TAPEINFO(1003, TAPETYPE:1, TITLE1:1, TITLE2:2, TITLE3:2)
v TAPEINFO(1003, TAPETYPE:1, TITLE1:3, TITLE2:1, TITLE3:3)
v TAPEINFO(1003, TAPETYPE:1, TITLE1:4, TITLE2:4, TITLE3:1))

Delete a subsumed disjunct -
Obtain Q(1):
GET W(DECODE(TAPETYPE:1), DECODE(PB:1), DECODE(FILE:1)):
E DATE_TIME:1, DATE:1, LON:1, LAT:1, ALT:1, ZEN:1
E TIME:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1
E SCAN:1, TITLE1:1, TITLE3:1, TITLE2:2, TITLE3:2
E TITLE1:4, TITLE2:4
(FILEINFO(PB:1, FILE:1, 1003)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1, PB:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1)
& (TAPEINFO(1003, TAPETYPE:1, TITLE1:1, TITLE2:2, TITLE3:2)
v TAPEINFO(1003, TAPETYPE:1, TITLE1:3, TITLE2:1, TITLE3:3)
v TAPEINFO(1003, TAPETYPE:1, TITLE1:4, TITLE2:4, TITLE3:1))

-41-
Delete another subsumed disjunct -

Obtain \( Q(2) \) (the optimized query):

\[
\text{GET } W(\text{DECODE(TAPETYPE:1), DECODE(PB:1), DECODE(FILE:1)}):
\]

\[
(\text{E)DATE\_TIME:1(\text{E)DATE:1(\text{E)LON:1(\text{E)LAT:1(\text{E)ALT:1(\text{E)ZEN:1(\text{E)TIME:1(\text{E)QUALITY:1(\text{E)ELECTR:1(\text{E)ILLUMIN:1(\text{E)CALIB:1(\text{E)SCAN:1(\text{E)TITLE1:1(\text{E)TITLE2:2(\text{E)TITLE3:2(\text{( FILEINFO(PB:1,FILE:1,1003)
\& \text{RECINFO(DATE\_TIME:1,DATE:1,TIME:1,LON:1,LAT:1,ALT:1,
\text{ZEN:1,PB:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,
\text{SCAN:1)
\& \text{TAPINFO(1003,TAPETYPE:1,TITLE1:1,TITLE2:2,TITLE3:2) }}
\]

-42-
Appendix 3.6
Query 4 transformation from ERB-N to ERB-ORAC and optimization

Query 4 in English:
What are the tape numbers, dates, and times where longitude < 30 and latitude > 10? Display tape number, longitude, and latitude together with date and time.

Query 4 in DEL for ERB-N:
GET W (NOTAPE: i, DATE1: l, TIME1: l, LON: i, LAT: i):
  (E) TAPETYPE: 1 (E) PLAYBACK: 1 (E) PB: 1 (E) NOFILE: 1 (E) CATALOG: 1
  (E) RTITLE: 1 (E) ALT: 1 (E) ZEN: 1 (E) QUALITY: 1 (E) ELECTR: 1
  (E) ILLUMIN: 1 (E) CALIB: 1 (E) SCAN: 1 (E) CATALOG: 2
  (RTAPE-PLAYBACK-CATALOG (NOTAPE: 1, TAPETYPE: 1, PLAYBACK: 1, PB: 1,
NOFILE: 1, CATALOG: 1, LON: i, LAT: i, ALT: 1, ZEN: 1, TIME1: 1,
QUALITY: 1, ELECTR: 1, ILLUMIN: 1, CALIB: 1, SCAN: 1, RTITLE: 1)
& RDATES-CATALOG (DATE1: 1, CATALOG: 2, LON: 1, LAT: 1, ALT: 1, ZEN: 1,
  TIME1: 1, QUALITY: 1, ELECTR: 1, ILLUMIN: 1, CALIB: 1, SCAN: 1)
& LON: 1 < 30
& LAT: 1 > 10 )

Interpreted query 4 in DBL for ERB-ORAC:
GET W (DECODE (NOTAPE: 1), DECODE (DATE: 1), DECODE (TIME: 1),
  DECODE (LON: 1), DECODE (LAT: 1)):
  (E) TAPETYPE: 1 (E) PB: 1 (E) FILE: 1 (E) ALT: 1 (E) ZEN: 1 (E) QUALITY: 1
  (E) ELECTR: 1 (E) ILLUMIN: 1 (E) CALIB: 1 (E) SCAN: 1
  (E) DATE_TIME: 1 (E) DATE: 2 (E) TITLE2: 2 (E) TITLE3: 2 (E) TITLE1: 3
  (E) TITLE3: 3 (E) TITLE1: 4 (E) TITLE2: 4 (E) TITLE1: 1 (E) TITLE2: 1
  (E) TITLE3: 1
  (FILEINFO (PB: 1, FILE: 1, NOTAPE: 1)
& RECINFO (DATE_TIME: 1, DATE: 2, TIME: 1, LON: 1, LAT: 1, ALT: 1,
  ZEN: 1, PB: 1, QUALITY: 1, ELECTR: 1, ILLUMIN: 1, CALIB: 1,
  SCAN: 1)
& ( TAPEINFO (NOTAPE: 1, TAPETYPE: 1, TITLE1: 1, TITLE2: 2,
  TITLE3: 2) 
  v TAPEINFO (NOTAPE: 1, TAPETYPE: 1, TITLE1: 3, TITLE2: 1,
  TITLE3: 3) 
  v TAPEINFO (NOTAPE: 1, TAPETYPE: 1, TITLE1: 4, TITLE2: 4,
  TITLE3: 1) ) )
& (E) DATE_TIME: 2 (E) PB: 2
RECINFO (DATE_TIME: 2, DATE: 1, TIME: 1, LON: 1, LAT: 1, ALT: 1, ZEN: 1,
  PB: 2, QUALITY: 1, ELECTR: 1, ILLUMIN: 1, CALIB: 1, SCAN: 1)
& LON: 1 < 3000
& LAT: 1 > 1000 )
Place interpreted query 4 in prenex conjunctive normal form -

Obtain Q(0):

GET W(DECODE(NOTAPE:1), DECODE(DATE:1), DECODE(TIME:1),
    DECODE(LON:1), DECODE(LAT:1)):

(E) TAPETYPE:1 (E) PB:1 (E) FILE:1 (E) ALT:1 (E) ZEN:1 (E) QUALITY:1
( E ) ELECTR:1 ( E ) ILLUMIN:1 ( E ) CALIB:1 ( E ) SCAN:1 ( E ) DATE_TIME:1
( E ) DATE:2 (E) TITLE2:2 (E) TITLE3:2 (E) TITLE1:3 (E) TITLE3:3
( E ) TITLE1:4 (E) TITLE2:4 (E) TITLE1:1 (E) TITLE2:1 (E) TITLE3:1
( E ) DATE_TIME:2 (E) PB:2
  ( FILEINFO(PB:1,FILE:1,NOTAPE:1)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1,
    PB:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1)
& ( TAPEINFO(NOTAPE:1, TAPETYPE:1, TITLE1:1, TITLE2:2, TITLE3:2)
   v TAPEINFO(NOTAPE:1, TAPETYPE:1, TITLE1:3, TITLE2:1, TITLE3:3)
   v TAPEINFO(NOTAPE:1, TAPETYPE:1, TITLE1:4, TITLE2:4,
     TITLE3:1) )
& RECINFO(DATE_TIME:2, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1,
    PB:2, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1)
& LON:1 < 3000
& LAT:1 > 1000

Delete a subsumed disjunct from the third conjunct -

Obtain Q(1):

GET W(DECODE(NOTAPE:1), DECODE(DATE:1), DECODE(TIME:1),
    DECODE(LON:1), DECODE(LAT:1)):

(E) TAPETYPE:1 (E) PB:1 (E) FILE:1 (E) ALT:1 (E) ZEN:1 (E) QUALITY:1
( E ) ELECTR:1 ( E ) ILLUMIN:1 ( E ) CALIB:1 ( E ) SCAN:1 ( E ) DATE_TIME:1
( E ) DATE:2 (E) TITLE2:2 (E) TITLE3:2 (E) TITLE1:4 (E) TITLE2:4
( E ) TITLE1:1 (E) TITLE3:1 (E) DATE_TIME:2 (E) PB:2
  ( FILEINFO(PB:1,FILE:1,NOTAPE:1)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1,
    PB:1, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1)
& ( TAPEINFO(NOTAPE:1, TAPETYPE:1, TITLE1:1, TITLE2:2, TITLE3:2)
   v TAPEINFO(NOTAPE:1, TAPETYPE:1, TITLE1:4, TITLE2:4,
     TITLE3:1) )
& RECINFO(DATE_TIME:2, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1,
    PB:2, QUALITY:1, ELECTR:1, ILLUMIN:1, CALIB:1, SCAN:1)
& LON:1 < 3000
& LAT:1 > 1000

-44-
Delete the subsumed disjunct from the third conjunct -

Obtain Q(2):

\[
\text{GET } \text{W(DECODE(NOTAPE:1),DECODE(DATE:1),DECODE(TIME:1),}
\]
\[
\text{DECODE(LON:1),DECODE(LAT:1))}:
\]
\[
\text{(E)TAPETYPE:1(E)PB:1(E)FILE:1(E)ALT:1(E)ZEN:1(E)QUALITY:1}
\]
\[
\text{(E)ELECTR:1(E)ILLUMIN:1(E)CALIB:1(E)SCAN:1(E)DATE_TIME:1}
\]
\[
\text{(E)DATE:2(E)TITLE2:2(E)TITLE3:2(E)TITLE1:1(E)DATE_TIME:2}
\]
\[
\text{(E)PB:2}
\]
\[
\text{( FILEINFO(PB:1,FILE:1,NOTAPE:1)
}\]
\[
\& \text{RECINFO} (\text{DATE_TIME:1,DATE:2,TIME:1,LON:1,LAT:1,ALT:1,ZEN:1,}
\]
\[
\text{PB:1,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,SCAN:1})
\]
\[
\& \text{TAPEINFO} (\text{NOTAPE:1,TAPETYPE:1,TITLE1:1,TITLE2:2,TITLE3:2})
\]
\[
\& \text{RECINFO} (\text{DATE_TIME:2,DATE:1,TIME:1,LON:1,LAT:1,ALT:1,ZEN:1,}
\]
\[
\text{PB:2,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,SCAN:1})
\]
\[
\& \text{LON:1} < 3000
\]
\[
\& \text{LAT:1} > 1000
\]

Rename variables and delete subsumed conjunct -

Obtain Q(3) (the optimized query):

\[
\text{GET } \text{W(DECODE(NOTAPE:1),DECODE(DATE:1),DECODE(TIME:1),}
\]
\[
\text{DECODE(LON:1),DECODE(LAT:1))}:
\]
\[
\text{(E)TAPETYPE:1(E)FILE:1(E)ALT:1(E)ZEN:1(E)QUALITY:1(E)ELECTR:1}
\]
\[
\text{(E)ILLUMIN:1(E)CALIB:1(E)SCAN:1(E)DATE_TIME:1(E)TITLE2:2}
\]
\[
\text{(E)DATE:2(E)TITLE1:1(E)DATE_TIME:2(E)PB:2}
\]
\[
\text{( FILEINFO(PB:1,FILE:1,NOTAPE:1)
}\]
\[
\& \text{TAPEINFO} (\text{NOTAPE:1,TAPETYPE:1,TITLE1:1,TITLE2:2,TITLE3:2})
\]
\[
\& \text{RECINFO} (\text{DATE_TIME:2,DATE:1,TIME:1,LON:1,LAT:1,ALT:1,ZEN:1,}
\]
\[
\text{PB:2,QUALITY:1,ELECTR:1,ILLUMIN:1,CALIB:1,SCAN:1})
\]
\[
\& \text{LON:1} < 3000
\]
\[
\& \text{LAT:1} > 1000
\]
Appendix 4.1

Logical optimization algorithm for a disjunctive query

Begin
Place the query in prenex conjunctive normal form: \( Q(0) \);
Set \( i := 0 \);
For \( j := 1 \) to \( n \) Do
If \( P_j \) is an equality that gives the value of a function as an
existentially quantified variable that does not appear in
any other conjunct
Then
Begin
Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
Begin
Delete \( P_j \);
Omit the superfluous quantifiers from the
prefix
End;
Set \( i := i+1 \)
End
Else
Begin
Set \( H_j := \) the set of pairs of disjuncts <\( D_1, D_2 \)> in \( P_j \)
such that \( D_1 \) is structurally a subcluster
of \( D_2 \)
While \( H_j \neq \emptyset \) Do
Begin
Pick a pair <\( D_1, D_2 \)> from \( H_j \);
Delete <\( D_1, D_2 \)> from \( H_j \);
Obtain \( D_2' \) from \( D_2 \) as follows:
Rename all objects in \( D_2 \) to match the
corresponding existentially quantified
variables in \( D_1 \) (if possible);
If \( D_1 \) is a subcluster of \( D_2' \)
Then
Begin
Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
Begin
Delete the disjunct \( D_2 \) from \( P_j \);
Omit the superfluous quantifiers from the
prefix
End;
Delete every pair from \( H_j \) which includes \( D_2 \);
Set \( i := i+1 \)
End
End;
End;
Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
Place in prenex disjunctive normal form;
Set \( i := i+1; \)

For \( j := 1 \) to \( k \) Do (\( D_1, \ldots, D_k \) are the disjuncts in \( Q(i) \))

Begin

While a functional dependency constraint may be applied to identify different objects in \( D_j \) Do

Begin

Pick an applicable functional dependency constraint \( FD \);
Set \( S_j := \) the set of objects identified by \( FD \) in \( D_j \);
Set \( a_j := \) the total number of target variables in \( S_j \);
Set \( b_j := \) the total number of constants in \( S_j \);
If \( a_j+b_j=0 \)
Then
Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
Begin
Pick one object from \( S_j \);
Change all objects in \( S_j \) to the chosen object for \( D_j \);
Omit the superfluous quantifiers from the prefix
End
Else
If \( a_j+b_j=1 \)
Then
Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
Begin
Change all existentially quantified variables in \( S_j \) for \( D_j \) to the target variable or constant in \( S_j \);
Omit the superfluous quantifiers from the prefix
End
Else
If \( b_j=0 \)
Then
Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
Begin
Pick one target variable, say \( X \), in \( S_j \);
For every other target variable, say \( Y \), in \( S_j \), add the conjunct \( Y = X \) to \( D_j \);
Change all other variables in \( S_j \) to \( X \) for \( D_j \) except in the just added equality conjuncts;
Omit the superfluous quantifiers from the prefix
End
Else
  If \( b = 1 \)
  Then
    Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
    Begin
      For every target variable, say \( Y \), in \( S_j \), add the conjunct \( Y = c \) to \( D_j \) (\( c \) is the constant in \( S_j \));
      Change all the variables in \( S_j \) to \( c \) in \( D_j \) except in the just added equality conjuncts;
      Omit the superfluous quantifiers from the prefix
    End
  Else
    Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
    Delete \( D_j \);
    Set \( i := i + 1 \)
  End;
Let \( P_j := \) the set of pairs of conjuncts \( <C_1, C_2> \) where \( C_1 \)
is structurally a subcluster of \( C_2 \);
While \( P_j \neq \) empty Do
  Begin
    Pick a pair \( <C_1, C_2> \) from \( P_j \);
    Delete \( <C_1, C_2> \) from \( P_j \);
    Obtain \( C_1' \) from \( C_1 \) as follows:
      Rename all existentially quantified variables in \( C_1 \)
to match the corresponding objects in \( C_2 \) (if possible);
    If \( C_1' \) is a subcluster of \( C_2 \) and every existentially quantified variable which was renamed to a target variable or a constant does not appear in another conjunct of \( D_j \)
    Then
      Begin
        Obtain \( Q(i+1) \) from \( Q(i) \) as follows:
        Begin
          Delete the conjunct \( C_1 \);
          Rename all existentially quantified variables in \( D_j \) according to the formation of \( C_1' \) from \( C_1 \);
          Omit the superfluous quantifiers from the prefix
        End;
        Delete every pair from \( P_j \) which includes \( C_1 \);
        Set \( i := i + 1 \)
      End
  End
End
Let \( P := \) the set of pairs of disjuncts \( \langle D_1, D_2 \rangle \) where every cluster of \( D_1 \) is structurally a subcluster of \( D_2 \).

While \( P \neq \emptyset \) Do

Begin

Pick a pair \( \langle D_1, D_2 \rangle \) from \( P \);
Delete \( \langle D_1, D_2 \rangle \) from \( P \);
Obtain \( D'_2 \) from \( D_2 \) as follows:

- Rename all objects in \( D_2 \) to match the corresponding existentially quantified variables in \( D_1 \) (if possible);
- If every cluster of \( D_1 \) is a subcluster of \( D'_2 \)

  Then

  Begin

  Obtain \( Q(i+1) \) from \( Q(i) \) as follows:

  Begin

  Delete the disjunct \( D_2 \);
  Omit the superfluous quantifiers from the prefix

  End;

  Delete every pair from \( P \) which includes \( D_2 \);

  Set \( i := i + 1 \)

  End

End

End.
Appendix 4.2
Query 5 transformation from GLOBE-R to GLOBE and optimization

Query 5 in English:
What are the project numbers, the description names and the altitudes for all ERB files that represent information corresponding to the electronic status being ON and the scanning mode being OFF?

Query 5 in DBL for GLOBE-R:
GET W(PROJNUM:1, NAME:1, ALT:1):
(E)TAPEID:1 (E)MISSION:1 (E)FORMAT:1 (E)GENDATE:1 (E)INVDATE:1
(E)ARCHIVER:1 (E)NUMFILES:1 (E)TPFIRSTORB:1 (E)TPLASTORB:1
(E)TPSTART:1 (E)TPSTOP:1 (E)TPALGORITHM:1 (E)COORDSYS:1
(E)SYNOPSTART:1 (E)SYNOPSTOP:1 (E)FILE:1 (E)ITEM:1 (E)ITSTART:1
(E)ITSTOP:1 (E)RECU:1 (E)ITALGORITHM:1 (E)ITLEN:1 (E)PB:1
(E)DATE:1 (E)TIME:1 (E)LON:1 (E)LAT:1 (E)ZEN:1 (E)QUALITY:1
(E)ILLUMIN:1 (E)CALIB:1 (E)NOTAPE:1
(E)TAPE(TAPEID:1,MISSION:1,ERB,FORMAT:1,PROJNUM:1,GENDATE:1, INVDATE:1,ARCHIVER:1,NUMFILES:1,TPFIRSTORB:1,
TPLASTORB:1,TPSTART:1,TPSTOP:1,TPALGORITHM:1,
COORDSYS:1,SYNOPSTART:1,SYNOPSTOP:1)
& ITEM(TAPEID:1,FILE:1,ITEM:1,ITSTART:1,ITSTOP:1,RECU:1,
ITALGORITHM:1,ITLEN:1)
& DESCR(ITEM:1,NAME:1)
& FILEINFO(PB:1,FILE:1,NOTAPE:1)
& RECINFO(DATE:1,TIME:1,LON:1,LAT:1,ALT:1,ZEN:1,PB:1,
QUALITY:1,ON,ILLUMIN:1,CALIB:1,OFF)
& NOTAPE:1 = NUM(TAPEID:1) )
Interpreted query 5 in DL for GLOBE:

GET W(DECODE(PROJNUM:1), DECODE(NAME:1), DECODE(ALT:1)):

(E) TAPEID:1 (E) MISSION:1 (E) FORMAT:1 (E) GENDATE:1 (E) INVDATE:1
(E) ARCHIVER:1 (E) NUMFILES:1 (E) TPFIRSTORB:1 (E) TPLASTORB:1
(E) TPSTART:1 (E) TPSTOP:1 (E) TPALGORITHM:1 (E) COORDSYS:1
(E) SYNOPSTART:1 (E) SYNOPSTOP:1 (E) FILE:1 (E) ITEM:1 (E) ITSTART:1
(E) ITSTOP:1 (E) RECNUM:1 (E) ITALGORITHM:1 (E) ITLEN:1 (E) PB:1
(E) DATE:1 (E) TIME:1 (E) LAT:1 (E) ZEN:1 (E) QUALITY:1
(E) ILLUMIN:1 (E) CALIB:1 (E) NOTAPE:1

( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
    INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
    TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1,
    COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)

& ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1,
    ITALGORITHM:1, ITLEN:1)

& DESCR(ITEM:1, NAME:1)

& ( FILEINFO(PB:1, FILE:1, NOTAPE:1)
  v (E) TAPETYPE:1 (E) PLAYBACK:1 (E) CATALOG:1 (E) RTITLE:1
    RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1,
    FILE:1, CATALOG:1, RTITLE:1)

& ( E) DATE_TIME:1

  RECFINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,
    ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)

  v (E) NOTAPE:2 (E) TAPETYPE:2 (E) PLAYBACK:2 (E) FILE:2
    (E) CATALOG:2 (E) RTITLE:2 (E) CATALOG:3
      ( RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2,
        PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1,
        TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF,
        RTITLE:2)

  & RDATES-CATALOG(DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1,
    ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)

& NOTAPE:1 = NUM(TAPEID:1) )
Place interpreted query 5 in prenex conjunctive normal form -

Obtain Q(0):

\[ \text{GET W(DECODE(PROJNUM:1), DECODE(NAME:1), DECODE(ALT:1))}: \]

\[ \begin{align*}
& \quad (E) \text{TapeID:1}(E) \text{Mission:1}(E) \text{GenDate:1}(E) \text{INVDate:1} \\
& \quad (E) \text{Archiver:1}(E) \text{NumFiles:1}(E) \text{TPFirstOrb:1}(E) \text{TPLastOrb:1} \\
& \quad (E) \text{TPStart:1}(E) \text{TPStop:1}(E) \text{TPAlgorithm:1}(E) \text{CoordSys:1} \\
& \quad (E) \text{SynOpStart:1}(E) \text{SynOpStop:1}(E) \text{File:1}(E) \text{Item:1}(E) \text{ITStart:1} \\
& \quad (E) \text{ITStop:1}(E) \text{RecNum:1}(E) \text{ITAlgorithm:1}(E) \text{ITLen:1}(E) \text{PB:1} \\
& \quad (E) \text{Date:1}(E) \text{Time:1}(E) \text{Lon:1}(E) \text{Lat:1}(E) \text{Zen:1}(E) \text{Quality:1} \\
& \quad (E) \text{Illumin:1}(E) \text{Calib:1}(E) \text{Notape:1}(E) \text{TapeType:1}(E) \text{Playback:1} \\
& \quad (E) \text{Catalog:1}(E) \text{RTitle:1}(E) \text{Date_Time:1}(E) \text{Notape:2}(E) \text{TapeType:2} \\
& \quad (E) \text{Playback:2}(E) \text{File:2}(E) \text{Catalog:2}(E) \text{RTitle:2}(E) \text{Catalog:3} \\
& \quad (E) \text{Tape}(\text{TapeID:1}, \text{Mission:1}, \text{ERB}, \text{Format:1}, \text{PROJNUM:1}, \text{GenDate:1}, \\
& \quad \text{INVDate:1}, \text{Archiver:1}, \text{NumFiles:1}, \text{TPFirstOrb:1}, \\
& \quad \text{TPLastOrb:1}, \text{TPStart:1}, \text{TPStop:1}, \text{TPAlgorithm:1}, \\
& \quad \text{CoordSys:1}, \text{SynOpStart:1}, \text{SynOpStop:1}) \\
\& \text{Item}(\text{TapeID:1}, \text{File:1}, \text{Item:1}, \text{ITStart:1}, \text{ITStop:1}, \text{RecNum:1}, \\
& \quad \text{ITAlgorithm:1}, \text{ITLen:1}) \\
\& \text{Descr}(\text{Item:1}, \text{Name:1}) \\
\& \quad (\text{FileInfo}(\text{PB:1}, \text{File:1}, \text{Notape:1}) \\
& \quad \text{v RaPe-PLAYBACK(Notape:1, TapeType:1, Playback:1, PB:1,} \\
& \quad \text{File:1, Catalog:1, RTitle:1)}) \\
\& \quad (\text{RecInfo}(\text{DATE_TIME:1}, \text{DATE:1}, \text{TIME:1}, \text{LON:1}, \text{LAT:1}, \text{ALT:1}, \\
& \quad \text{Zen:1}, \text{PB:1}, \text{QUALITY:1}, \text{ON}, \text{ILLUMIN:1}, \text{CALIB:1}, \text{OFF}) \\
& \quad \text{v RaPe-PLAYBACK-CATALOG(Notape:2, TapeType:2, Playback:2,} \\
& \quad \text{PB:1, File:2, Catalog:2, LON:1, LAT:1, ALT:1, Zen:1,} \\
& \quad \text{TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF,} \\
& \quad \text{RTitle:2)}) \\
\& \quad (\text{RecInfo}(\text{DATE_TIME:1}, \text{DATE:1}, \text{TIME:1}, \text{LON:1}, \text{LAT:1}, \text{ALT:1}, \\
& \quad \text{Zen:1}, \text{PB:1}, \text{QUALITY:1}, \text{ON}, \text{ILLUMIN:1}, \text{CALIB:1}, \text{OFF}) \\
& \quad \text{v RDates-CATALOG(\text{DATE:1}, \text{Catalog:3}, \text{LON:1}, \text{LAT:1}, \text{ALT:1}, \\
& \quad \text{Zen:1}, \text{TIME:1}, \text{QUALITY:1}, \text{ON}, \text{ILLUMIN:1}, \text{CALIB:1}, \text{OFF})}) \\
\& \text{Notape:1 = Num(\text{TapeID:1})})
Place in prenex disjunctive normal form:

Obtain Q(1):

\[
\text{GET } W(\text{DECODE(PROJNUM:1)}, \text{DECODE(NAME:1)}, \text{DECODE(ALT:1)}):
\]

\[
\begin{align*}
(\text{E}) & \text{TapeID:1 (E) MISSION:1 (E) FORMAT:1 (E) GENDATE:1 (E) INVDATE:1} \\
(\text{E}) & \text{ARCHIVER:1 (E) NUMFILES:1 (E) TPFIRSTORB:1 (E) TPLASTORB:1} \\
(\text{E}) & \text{TPSTART:1 (E) TPSTOP:1 (E) TPALGORITHM:1 (E) COORDSYS:1} \\
(\text{E}) & \text{SYNOPSTART:1 (E) SYNOPSTOP:1 (E) FILE:1 (E) ITEM:1 (E) ITSTART:1} \\
(\text{E}) & \text{ITSTOP:1 (E) RECNUM:1 (E) ITALGORITHM:1 (E) ITLEN:1 (E) PB:1} \\
(\text{E}) & \text{DATE:1 (E) TIME:1 (E) LON:1 (E) LAT:1 (E) ZEN:1 (E) QUALITY:1} \\
(\text{E}) & \text{ILLUMIN:1 (E) CALIB:1 (E) NOTAPE:1 (E) TAPETYPE:1 (E) PLAYBACK:1} \\
(\text{E}) & \text{CATALOG:1 (E) RTITLE:1 (E) DATE_TIME:1 (E) NOTAPE:2 (E) TAPETYPE:2} \\
(\text{E}) & \text{PLAYBACK:2 (E) FILE:2 (E) CATALOG:2 (E) RTITLE:2 (E) CATALOG:3} \\
\end{align*}
\]

\[
\text{( } \text{TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,}
\text{ INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,}
\text{ TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1,}
\text{ COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)}
\]

\[
\& \text{ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1,}
\text{ ITALGORITHM:1, ITLEN:1)}
\]

\[
\& \text{DESCR(ITEM:1, NAME:1)}
\]

\[
\& \text{FILENAME(PB:1, FILE:1, NOTAPE:1)}
\]

\[
\& \text{RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,}
\text{ ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)}
\]

\[
\& \text{RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,}
\text{ ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)}
\]

\[
\& \text{NOTAPE:1 = NUM(TAPEID:1)}
\]

\[
\lor \text{( } \text{TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,}
\text{ INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,}
\text{ TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1,}
\text{ COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)}
\]

\[
\& \text{ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1,}
\text{ ITALGORITHM:1, ITLEN:1)}
\]

\[
\& \text{DESCR(ITEM:1, NAME:1)}
\]

\[
\& \text{FILENAME(PB:1, FILE:1, NOTAPE:1)}
\]

\[
\& \text{RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,}
\text{ ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)}
\]

\[
\& \text{RDATES-CATALOG(DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1,}
\text{ ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)}
\]

\[
\& \text{NOTAPE:1 = NUM(TAPEID:1)}
\]

-53-
Delete subsumed conjunct in first disjunct -
Obtain Q(2):

GET W(DECODE(PROJNUM:1), DECODE(NAME:1), DECODE(ALT:1)):

(E) TAPEID:1 (E) MISSION:1 (E) FORMAT:1 (E) GENDATE:1 (E) INVDATE:1
(E) ARCHIVER:1 (E) NUMFILES:1 (E) TPFIRSTORB:1 (E) TPLASTORB:1
(E) SYNOPSIS:1 (E) TPSTOP:1 (E) TPALGORITHM:1 (E) COORDSYS:1
(E) ITSTART:1 (E) ISTOP:1 (E) RECU:1 (E) ITALGORITHM:1 (E) ITLEN:1 (E) PB:1
(E) DATE:1 (E) TIME:1 (E) LON:1 (E) LAT:1 (E) ZEN:1 (E) QUALITY:1
(E) ILLUMIN:1 (E) CALIB:1 (E) NOTAPE:1 (E) TAPETYPE:1 (E) PLAYBACK:1
(E) CATALOG:1 (E) RTITLE:1 (E) DATE_TIME:1 (E) NOTAPE:2 (E) TAPETYPE:2
(E) PLAYBACK:2 (E) FILE:2 (E) CATALOG:2 (E) RTITLE:2 (E) CATALOG:3

( ( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1,
COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ISTOP:1, RECU:1,
ITALGORITHM:1, ITLEN:1)
& DESCR(ITEM:1, NAME:1)
& FILEINFO(PB:1, FILE:1, NOTAPE:1)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,
ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM(TAPEID:1) )

v ( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1,
COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ISTOP:1, RECU:1,
ITALGORITHM:1, ITLEN:1)
& DESCR(ITEM:1, NAME:1)
& FILEINFO(PB:1, FILE:1, NOTAPE:1)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,
ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& RDATES-CATALOG(DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1,
ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM(TAPEID:1) )

v ( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALGORITHM:1,
COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ISTOP:1, RECU:1,
ITALGORITHM:1, ITLEN:1)
& DESCR(ITEM:1, NAME:1)
& FILEINFO(PB:1, FILE:1, NOTAPE:1)
& RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2,
PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1,
ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF,
RTITLE:2)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,
ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM(TAPEID:1) )

-56-
Delete subsumed conjunct in fifth disjunct –

Obtain Q(3):

GET W(DECODER (PROJNUM: 1), DECODE (NAME: 1), DECODE (ALT: 1)):

(E) TAPEID: 1 (E) MISSION: 1 (E) FORMAT: 1 (E) GENDATE: 1
(E) INVDATE: 1 (E) ARCHIVER: 1 (E) NUMFILES: 1
(E) TPFIRSTORB: 1 (E) TPLASTORB: 1 (E) TPSTART: 1
(E) TPSTOP: 1 (E) TPALGORITHM: 1 (E) COORDSYS: 1
(E) SYNOPSTART: 1 (E) SYNOPSTOP: 1 (E) FILE: 1
(E) ITEM: 1 (E) ITSTART: 1 (E) ITSTOP: 1
(E) RECU: 1 (E) ITLEN: 1 (E) PB: 1
(E) DATE: 1 (E) TIME: 1 (E) LON: 1 (E) LAT: 1
(E) ZEN: 1 (E) QUALITY: 1 (E) ILLUMIN: 1 (E) CALIB: 1
(E) NOTAPE: 1 (E) TAPEETYPE: 1 (E) PLAYBACK: 1
(E) CATALOG: 1 (E) RTITLE: 1 (E) DATE_TIME: 1
(E) NOTAPE: 2 (E) TAPEETYPE: 2 (E) PLAYBACK: 2
(E) FILE: 2 (E) CATALOG: 2 (E) RTITLE: 2 (E) CATALOG: 3

(TAPE(TAPEID: 1, MISSION: 1, ERB, FORMAT: 1, PROJNUM: 1, GENDATE: 1,
INVDATE: 1, ARCHIVER: 1, NUMFILES: 1, TPFIRSTORB: 1,
TPLASTORB: 1, TPSTART: 1, TPSTOP: 1, TPALGORITHM: 1,
COORDSYS: 1, SYNOPSTART: 1, SYNOPSTOP: 1)
& ITEM(TAPEID: 1, FILE: 1, ITEM: 1, ITSTART: 1, ITSTOP: 1, RECU: 1,
ITLEN: 1)
& DESCR(ITEM: 1, NAME: 1)
& FILEINFO(PB: 1, FILE: 1, NOTAPE: 1)
& RECU: 1 (DATE_TIME: 1, DATE: 1, TIME: 1, LON: 1, LAT: 1, ALT: 1,
ZEN: 1, PB: 1, QUALITY: 1, ON, ILLUMIN: 1, CALIB: 1, OFF)
& NOTAPE: 1 = NUM(TAPEID: 1) )

(TAPE(TAPEID: 1, MISSION: 1, ERB, FORMAT: 1, PROJNUM: 1, GENDATE: 1,
INVDATE: 1, ARCHIVER: 1, NUMFILES: 1, TPFIRSTORB: 1,
TPLASTORB: 1, TPSTART: 1, TPSTOP: 1, TPALGORITHM: 1,
COORDSYS: 1, SYNOPSTART: 1, SYNOPSTOP: 1)
& ITEM(TAPEID: 1, FILE: 1, ITEM: 1, ITSTART: 1, ITSTOP: 1, RECU: 1,
ITLEN: 1)
& DESCR(ITEM: 1, NAME: 1)
& FILEINFO(PB: 1, FILE: 1, NOTAPE: 1)
& RECU: 1 (DATE_TIME: 1, DATE: 1, TIME: 1, LON: 1, LAT: 1, ALT: 1,
ZEN: 1, PB: 1, QUALITY: 1, ON, ILLUMIN: 1, CALIB: 1, OFF)
& RDATES–CATALOG(DATE: 1, CATALOG: 3, LON: 1, LAT: 1, ALT: 1,
ZEN: 1, TIME: 1, QUALITY: 1, ON, ILLUMIN: 1, CALIB: 1, OFF)
& NOTAPE: 1 = NUM(TAPEID: 1) )
Delete subsumed second disjunct -

Obtain $Q_4$:

\[
\begin{aligned}
\text{GET } & W(\text{DECODE}(\text{PROJNUM}:1), \text{DECODE}(\text{NAME}:1), \text{DECODE}(\text{ALT}:1)): \\
& (\text{E})\text{TapeID}:1 (\text{E})\text{Mission}:1 (\text{E})\text{Format}:1 (\text{E})\text{Gendate}:1 (\text{E})\text{Invdate}:1 \\
& (\text{E})\text{Archiver}:1 (\text{E})\text{Numfiles}:1 (\text{E})\text{Tpfirstorb}:1 (\text{E})\text{Tplastorb}:1 \\
& (\text{E})\text{Tpstart}:1 (\text{E})\text{Tpstop}:1 (\text{E})\text{Tpalgorithm}:1 (\text{E})\text{Coordsys}:1 \\
& (\text{E})\text{Synopstart}:1 (\text{E})\text{Synopstop}:1 (\text{E})\text{File}:1 (\text{E})\text{Item}:1 (\text{E})\text{ITstart}:1 \\
& (\text{E})\text{ITstop}:1 (\text{E})\text{Recnum}:1 (\text{E})\text{Ialgorithm}:1 (\text{E})\text{ITlen}:1 (\text{E})\text{PB}:1 \\
& (\text{E})\text{Date}:1 (\text{E})\text{Time}:1 (\text{E})\text{Lon}:1 (\text{E})\text{Lat}:1 (\text{E})\text{Zen}:1 (\text{E})\text{Quality}:1 \\
& (\text{E})\text{Illumin}:1 (\text{E})\text{Calib}:1 (\text{E})\text{Notape}:1 (\text{E})\text{Tapetype}:1 (\text{E})\text{Playback}:1 \\
& (\text{E})\text{Catalog}:1 (\text{E})\text{Rtitle}:1 (\text{E})\text{Date_time}:1 (\text{E})\text{Notape}:2 (\text{E})\text{Tapetype}:2 \\
& (\text{E})\text{Playback}:2 (\text{E})\text{File}:2 (\text{E})\text{Catalog}:2 (\text{E})\text{Rtitle}:2 (\text{E})\text{Catalog}:3 \\
& \quad (\text{TAPE}(\text{TapeID}:1, \text{Mission}:1, \text{ERB}, \text{Format}:1, \text{PROJNUM}:1, \text{Gendate}:1, \\
& \quad \text{Invdate}:1, \text{Archiver}:1, \text{Numfiles}:1, \text{Tpfirstorb}:1, \\
& \quad \text{Tplastorb}:1, \text{Tpstart}:1, \text{Tpstop}:1, \text{Tpalgorithm}:1, \\
& \quad \text{Coordsys}:1, \text{Synopstart}:1, \text{Synopstop}:1) \\
& \quad \& \text{ITEM}(\text{TapeID}:1, \text{File}:1, \text{Item}:1, \text{ITstart}:1, \text{ITstop}:1, \text{Recnum}:1, \\
& \quad \text{Ialgorithm}:1, \text{ITlen}:1) \\
& \quad \& \text{Descr}(\text{Item}:1, \text{Name}:1) \\
& \quad \& \text{Fileinfo}(\text{PB}:1, \text{File}:1, \text{Notape}:1) \\
& \quad \& \text{Recinfo}(\text{Date_time}:1, \text{Date}:1, \text{Time}:1, \text{Lon}:1, \text{Lat}:1, \text{Alt}:1, \\
& \quad \quad \text{Zen}:1, \text{PB}:1, \text{Quality}:1, \text{ON, Illumin}:1, \text{Calib}:1, \text{OFF}) \\
& \quad \& \text{Notape}:1 = \text{NUM}(\text{TapeID}:1) \\
& \quad \text{v} (\text{TAPE}(\text{TapeID}:1, \text{Mission}:1, \text{ERB}, \text{Format}:1, \text{PROJNUM}:1, \text{Gendate}:1, \\
& \quad \text{Invdate}:1, \text{Archiver}:1, \text{Numfiles}:1, \text{Tpfirstorb}:1, \\
& \quad \text{Tplastorb}:1, \text{Tpstart}:1, \text{Tpstop}:1, \text{Tpalgorithm}:1, \\
& \quad \text{Coordsys}:1, \text{Synopstart}:1, \text{Synopstop}:1) \\
& \quad \& \text{ITEM}(\text{TapeID}:1, \text{File}:1, \text{Item}:1, \text{ITstart}:1, \text{ITstop}:1, \text{Recnum}:1, \\
& \quad \text{Ialgorithm}:1, \text{ITlen}:1) \\
& \quad \& \text{Descr}(\text{Item}:1, \text{Name}:1) \\
& \quad \& \text{Fileinfo}(\text{PB}:1, \text{File}:1, \text{Notape}:1) \\
& \quad \& \text{RTape-Playback-Catalog}(\text{Notape}:2, \text{Tapetype}:2, \text{Playback}:2, \\
& \quad \text{PB}:1, \text{File}:2, \text{Catalog}:2, \text{Lon}:1, \text{Lat}:1, \text{Alt}:1, \text{Zen}:1, \\
& \quad \text{Time}:1, \text{Quality}:1, \text{ON, Illumin}:1, \text{Calib}:1, \text{OFF,} \\
& \quad \text{Rtitle}:2) \\
& \quad \& \text{Recinfo}(\text{Date_time}:1, \text{Date}:1, \text{Time}:1, \text{Lon}:1, \text{Lat}:1, \text{Alt}:1, \\
& \quad \quad \text{Zen}:1, \text{PB}:1, \text{Quality}:1, \text{ON, Illumin}:1, \text{Calib}:1, \text{OFF}) \\
& \quad \& \text{Notape}:1 = \text{NUM}(\text{TapeID}:1) 
\end{aligned}
\]
( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1, INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, TPLASTORB:1, TPSTART:1, TPSTOP:1, TPAUTHORITY:1, COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1) & ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, ITALGORITHM:1, ITLEN:1) & DESCR(ITEM:1, NAME:1) & FILEINFO(PB:1, FILE:1, NOTAPE:1) & RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2, PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF, RTITLE:2) & RDATES-CATALOG(DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1, ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) & NOTAPE:1 = NUM(TAPEID:1) )

( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1, INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, TPLASTORB:1, TPSTART:1, TPSTOP:1, TPAUTHORITY:1, COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1) & ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, ITALGORITHM:1, ITLEN:1) & DESCR(ITEM:1, NAME:1) & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, FILE:1, CATALOG:1, RTITLE:1) & RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) & NOTAPE:1 = NUM(TAPEID:1) )

( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1, INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, TPLASTORB:1, TPSTART:1, TPSTOP:1, TPAUTHORITY:1, COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1) & ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, ITALGORITHM:1, ITLEN:1) & DESCR(ITEM:1, NAME:1) & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, FILE:1, CATALOG:1, RTITLE:1) & RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) & RDATES-CATALOG(DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1, ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) & NOTAPE:1 = NUM(TAPEID:1) )

-63-
\texttt{v \ (TAPE(TAPEID:1,MISSION:1,ERB,FORMAT:1,PROJNUM:1,GENDATE:1,}
\texttt{INVDATE:1,ARCHIVER:1,NUMFILES:1,TFFIRSTORB:1,}
\texttt{TPLASTORB:1,TPSTART:1,TPSTOP:1,TPALGORITHM:1,}
\texttt{COORDSYS:1,SYNOPSTART:1,SYNOPSTOP:1)}
\texttt{\& ITEM(TAPEID:1,FILE:1,ITEM:1,ITSTART:1,ITSTOP:1,RECNUM:1,}
\texttt{ITALGORITHM:1,ITLEN:1)}
\texttt{\& DESCRIPT(ITEM:1,NAME:1)}
\texttt{\& RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:1,PLAYBACK:1,PB:1,}
\texttt{FILE:1,CATALOG:1,RTITLE:1)}
\texttt{\& RTAPE-PLAYBACK-CATALOG(NOTAPE:2,TAPETYPE:2,PLAYBACK:2,}
\texttt{PB:1,FILE:2,CATALOG:2,LON:1,Lat:1,ALT:1,ZEN:1,}
\texttt{TIME:1,QUALITY:1,ON,IILLUMIN:1,CALIB:1,OFF,}
\texttt{RTITLE:2)}
\texttt{\& RECINFO(DATE_TIME:1,DATE:1,TIME:1,LON:1,Lat:1,ALT:1,}
\texttt{ZEN:1,PB:1,QUALITY:1,ON,IILLUMIN:1,CALIB:1,OFF)}
\texttt{\& NOTAPE:1 = NUM(TAPEID:1) )
\texttt{v \ (TAPE(TAPEID:1,MISSION:1,ERB,FORMAT:1,PROJNUM:1,GENDATE:1,}
\texttt{INVDATE:1,ARCHIVER:1,NUMFILES:1,TFFIRSTORB:1,}
\texttt{TPLASTORB:1,TPSTART:1,TPSTOP:1,TPALGORITHM:1,}
\texttt{COORDSYS:1,SYNOPSTART:1,SYNOPSTOP:1)}
\texttt{\& ITEM(TAPEID:1,FILE:1,ITEM:1,ITSTART:1,ITSTOP:1,RECNUM:1,}
\texttt{ITALGORITHM:1,ITLEN:1)}
\texttt{\& DESCRIPT(ITEM:1,NAME:1)}
\texttt{\& RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:1,PLAYBACK:1,PB:1,}
\texttt{FILE:1,CATALOG:1,RTITLE:1)}
\texttt{\& RTAPE-PLAYBACK-CATALOG(NOTAPE:2,TAPETYPE:2,PLAYBACK:2,}
\texttt{PB:1,FILE:2,CATALOG:2,LON:1,Lat:1,ALT:1,ZEN:1,}
\texttt{TIME:1,QUALITY:1,ON,IILLUMIN:1,CALIB:1,OFF,}
\texttt{RTITLE:2)}
\texttt{\& RDATES-CATALOG(DATE:1,CATALOG:3,LON:1,Lat:1,ALT:1,}
\texttt{ZEN:1,TIME:1,QUALITY:1,ON,IILLUMIN:1,CALIB:1,OFF)}
\texttt{\& NOTAPE:1 = NUM(TAPEID:1) } \) \)
Delete subsumed second disjunct - 

Obtain Q(5):

GET W (DECODE (PROJNUM: 1), DECODE (NAME: 1), DECODE (ALT: 1)):

(E) TAPEID: 1 (E) MISSION: 1 (E) FORMAT: 1 (E) GENDATE: 1 (E) INVDATE: 1
(E) ARCHIVER: 1 (E) NUMFILES: 1 (E) TPFIRSTORB: 1 (E) TPALGORITHM: 1 (E) COORDSYS: 1
(E) SYNOPSTART: 1 (E) SYNOPSTOP: 1 (E) ITEM: 1 (E) ITSTART: 1
(E) ITSTOP: 1 (E) RECNUM: 1 (E) ITALGORITHM: 1 (E) ITLEN: 1 (E) PB: 1
(E) DATE: 1 (E) TIME: 1 (E) LON: 1 (E) LAT: 1 (E) ZEN: 1 (E) QUALITY: 1
(E) ILLUMIN: 1 (E) CALIB: 1 (E) NOTAPE: 1 (E) TAPETYPE: 1 (E) PLAYBACK: 1
(E) CATALOG: 1 (E) RTITLE: 1 (E) DATE_TIME: 1 (E) NOTAPE: 2 (E) TAPETYPE: 2
(E) PLAYBACK: 2 (E) FILE: 2 (E) CATALOG: 2 (E) RTITLE: 2 (E) CATALOG: 3
( ( TAPE (TAPEID: 1, MISSION: 1, ERB, FORMAT: 1, PROJNUM: 1, GENDATE: 1,
INVDATE: 1, ARCHIVER: 1, NUMFILES: 1, TPFIRSTORB: 1,
TPALGORITHM: 1, COORDSYS: 1, SYNOPSTART: 1, SYNOPSTOP: 1)
& ITEM (TAPEID: 1, FILE: 1, ITEM: 1, ITSTART: 1, ITSTOP: 1, RECNUM: 1,
ITALGORITHM: 1, ITLEN: 1)
& DECR (ITEM: 1, NAME: 1)
& FILEINFO (PB: 1, FILE: 1, NOTAPE: 1)
& RECINFO (DATE_TIME: 1, DATE: 1, TIME: 1, LON: 1, LAT: 1, ALT: 1,
ZEN: 1, PB: 1, QUALITY: 1, ON, ILLUMIN: 1, CALIB: 1, OFF)
& NOTAPE: 1 = NUM (TAPEID: 1) )
)

v ( TAPE (TAPEID: 1, MISSION: 1, ERB, FORMAT: 1, PROJNUM: 1, GENDATE: 1,
INVDATE: 1, ARCHIVER: 1, NUMFILES: 1, TPFIRSTORB: 1,
TPALGORITHM: 1, COORDSYS: 1, SYNOPSTART: 1, SYNOPSTOP: 1)
& ITEM (TAPEID: 1, FILE: 1, ITEM: 1, ITSTART: 1, ITSTOP: 1, RECNUM: 1,
ITALGORITHM: 1, ITLEN: 1)
& DECR (ITEM: 1, NAME: 1)
& FILEINFO (PB: 1, FILE: 1, NOTAPE: 1)
& RTAPE-PLAYBACK-CATALOG (NOTAPE: 2, TAPETYPE: 2, PLAYBACK: 2,
PB: 1, FILE: 2, CATALOG: 2, LON: 1, LAT: 1, ALT: 1, ZEN: 1,
TIME: 1, QUALITY: 1, ON, ILLUMIN: 1, CALIB: 1, OFF,
RTITLE: 2)
& RDATES-CATALOG (DATE: 1, CATALOG: 3, LON: 1, LAT: 1, ALT: 1,
ZEN: 1, TIME: 1, QUALITY: 1, ON, ILLUMIN: 1, CALIB: 1, OFF)
& NOTAPE: 1 = NUM (TAPEID: 1) )
)

v ( TAPE (TAPEID: 1, MISSION: 1, ERB, FORMAT: 1, PROJNUM: 1, GENDATE: 1,
INVDATE: 1, ARCHIVER: 1, NUMFILES: 1, TPFIRSTORB: 1,
TPALGORITHM: 1, COORDSYS: 1, SYNOPSTART: 1, SYNOPSTOP: 1)
& ITEM (TAPEID: 1, FILE: 1, ITEM: 1, ITSTART: 1, ITSTOP: 1, RECNUM: 1,
ITALGORITHM: 1, ITLEN: 1)
& DECR (ITEM: 1, NAME: 1)
& RTAPE-PLAYBACK (NOTAPE: 1, TAPETYPE: 1, PLAYBACK: 1, PB: 1,
FILE: 1, CATALOG: 1, RTITLE: 1)
& RECINFO (DATE_TIME: 1, DATE: 1, TIME: 1, LON: 1, LAT: 1, ALT: 1,
ZEN: 1, PB: 1, QUALITY: 1, ON, ILLUMIN: 1, CALIB: 1, OFF)
& NOTAPE: 1 = NUM (TAPEID: 1) )
( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1, 
  INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, 
  TPSTARTB:1, TSTOPB:1, TPALGORITHM:1, 
  COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1, 
  & ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, 
    ITALGORITHM:1, ITLEN:1) 
  & DESCRIPT(ITEM:1, NAME:1) 
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, 
    FILE:1, CATALOG:1, RTITLE:1) 
  & RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, 
    ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) 
  & RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2, 
    PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1, 
    TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF, 
    RTITLE:2) 
  & RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, 
    ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) 
  & NOTAPE:1 = NUM(TAPEID:1) ) )

( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1, 
  INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, 
  TPSTARTB:1, TSTOPB:1, TPALGORITHM:1, 
  COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1, 
  & ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, 
    ITALGORITHM:1, ITLEN:1) 
  & DESCRIPT(ITEM:1, NAME:1) 
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, 
    FILE:1, CATALOG:1, RTITLE:1) 
  & RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2, 
    PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1, 
    TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF, 
    RTITLE:2) 
  & RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, 
    ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) 
  & NOTAPE:1 = NUM(TAPEID:1) ) )

( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1, 
  INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1, 
  TPSTARTB:1, TSTOPB:1, TPALGORITHM:1, 
  COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1, 
  & ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1, 
    ITALGORITHM:1, ITLEN:1) 
  & DESCRIPT(ITEM:1, NAME:1) 
  & RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1, 
    FILE:1, CATALOG:1, RTITLE:1) 
  & RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2, 
    PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1, 
    TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF, 
    RTITLE:2) 
  & RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1, 
    ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF) 
  & NOTAPE:1 = NUM(TAPEID:1) ) )

-66-
Delete subsumed fourth disjunct –

Obtain Q(6):

GET W(DECODER(Projenum:1), DECODER(NAME:1), DECODER(ALT:1)):

(E) TAPEID:1(E) MISSION:1(E) FORMAT:1(E) GENERATE:1(E) INVDATE:1
(E) ARCHIVER:1(E) NUMFILES:1(E) TPFIRSTORB:1(E) TLASTORB:1
(E) TPSTART:1(E) TPSTOP:1(E) TPALEGGH:1(E) COORDSYS:1
(E) SYNOPSIS:1(E) SYNOPS:1(E) FILE:1(E) ITEM:1(E) ITSTART:1
(E) ITSTOP:1(E) REFCNUM:1(E) ITALGORITHM:1(E) ITLEN:1(E) PB:1
(E) DATE:1(E) TIME:1(E) LON:1(E) LAT:1(E) ZEN:1(E) QUALITY:1
(E) ILUMIN:1(E) CALIB:1(E) NOTAPE:1(E) TAPETYPE:1(E) PLAYBACK:1
(E) CATALOG:1(E) RTITLE:1(E) DATE_TIME:1(E) NOTAPE:2(E) TAPETYPE:2
(E) PLAYBACK:2(E) FILE:2(E) CATALOG:2(E) RTITLE:2(E) CATALOG:3

( ( TAPE(TAPEID:1,MISSION:1,ERB,FORMAT:1,PROJNUM:1,GENDATE:1, INVDATE:1,ARCHIVER:1,NUMFILES:1,TPFIRSTORB:1, TLASTORB:1,TPSTART:1,TPSTOP:1,TPALGORITHM:1, COORDSYS:1,SYNOPSIS:1,SYNOPS:1)
& ITEM(TAPEID:1,FILE:1,ITEM:1,ITSTART:1,ITSTOP:1,RECNUM:1, ITALGORITHM:1,ITLEN:1)
& DESCRIPTOR(ITEM:1,NAME:1)
& FILEINFO(PB:1,FILE:1,NOTAPE:1)
& RECINFO(DATE_TIME:1,DATE:1,TIME:1,LON:1,LAT:1,ALT:1, ZEN:1,PB:1,QUALITY:1,ON,ILLUMIN:1,CALIB:1,OFF)
& NOTAPE:1 = NUM(TAPEID:1) )

v ( TAPE(TAPEID:1,MISSION:1,ERB,FORMAT:1,PROJNUM:1,GENDATE:1, INVDATE:1,ARCHIVER:1,NUMFILES:1,TPFIRSTORB:1, TLASTORB:1,TPSTART:1,TPSTOP:1,TPALGORITHM:1, COORDSYS:1,SYNOPSIS:1,SYNOPS:1)
& ITEM(TAPEID:1,FILE:1,ITEM:1,ITSTART:1,ITSTOP:1,RECNUM:1, ITALGORITHM:1,ITLEN:1)
& DESCRIPTOR(ITEM:1,NAME:1)
& FILEINFO(PB:1,FILE:1,NOTAPE:1)
& RTAPE-PLAYBACK-CATALOG(NOTAPE:2,TAPETYPE:2,PLAYBACK:2, PB:1,FILE:2,CATALOG:2,LON:1,LAT:1,ALT:1,ZEN:1, TIME:1,QUALITY:1,ON,ILLUMIN:1,CALIB:1,OFF, RTITLE:2)
& RDATES-CATALOG(DATE:1,CATALOG:3,LON:1,LAT:1,ALT:1, ZEN:1,TIME:1,QUALITY:1,ON,ILLUMIN:1,CALIB:1,OFF)
& NOTAPE:1 = NUM(TAPEID:1) )

v ( TAPE(TAPEID:1,MISSION:1,ERB,FORMAT:1,PROJNUM:1,GENDATE:1, INVDATE:1,ARCHIVER:1,NUMFILES:1,TPFIRSTORB:1, TLASTORB:1,TPSTART:1,TPSTOP:1,TPALGORITHM:1, COORDSYS:1,SYNOPSIS:1,SYNOPS:1)
& ITEM(TAPEID:1,FILE:1,ITEM:1,ITSTART:1,ITSTOP:1,RECNUM:1, ITALGORITHM:1,ITLEN:1)
& DESCRIPTOR(ITEM:1,NAME:1)
& RTAPE-PLAYBACK(NOTAPE:1,TAPETYPE:1,PLAYBACK:1,PB:1, FILE:1,CATALOG:1,RTITLE:1)
& RECINFO(DATE_TIME:1,DATE:1,TIME:1,LON:1,LAT:1,ALT:1, ZEN:1,PB:1,QUALITY:1,ON,ILLUMIN:1,CALIB:1,OFF)
& NOTAPE:1 = NUM(TAPEID:1) )
v ( TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
 INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
 TPLASTORB:1, TPSTART:1, TSTOP:1, TALKGORITHM:1,
 COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1,
 ITALGORITHM:1, ITLEN:1)
& DESCR(ITEM:1, NAME:1)
& RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1,
 FILE:1, CATALOG:1, RTITLE:1)
& RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2,
 PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1,
 TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF,
 RTITLE:2)
& RECINFO(DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,
 ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM(TAPEID:1) 
v ( TAPE(TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
 INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
 TPLASTORB:1, TPSTART:1, TSTOP:1, TALKGORITHM:1,
 COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM(TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1,
 ITALGORITHM:1, ITLEN:1)
& DESCR(ITEM:1, NAME:1)
& RTAPE-PLAYBACK(NOTAPE:1, TAPETYPE:1, PLAYBACK:1, PB:1,
 FILE:1, CATALOG:1, RTITLE:1)
& RTAPE-PLAYBACK-CATALOG(NOTAPE:2, TAPETYPE:2, PLAYBACK:2,
 PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1,
 TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF,
 RTITLE:2)
& RDATES-CATALOG(DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1,
 ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM(TAPEID:1) ) )
Delete subsumed fourth disjunct -

Obtain Q(7) (the optimized query):

GET W(DECODER(PROJNUM:1), DECODER(NAME:1), DECODER(ALT:1)):
(E) TAPEID:1 (E) MISSION:1 (E) FORMAT:1 (E) GENDATE:1 (E) INVDATE:1
(E) ARCHIVER:1 (E) NUMFILES:1 (E) TPFIRSTORB:1 (E) TPLASTORB:1
(E) TPSTART:1 (E) TPSTOP:1 (E) TPALEGORITHM:1 (E) COORDSYS:1
(E) SYNOPSTART:1 (E) SYNOPSTOP:1 (E) FILE:1 (E) ITEM:1 (E) ITSTART:1
(E) ITSTOP:1 (E) RECN:1 (E) ITALGORITHM:1 (E) ITFILENAME:1 (E) PB:1
(E) DATE:1 (E) TIME:1 (E) LON:1 (E) LAT:1 (E) ZEN:1 (E) QUALITY:1
(E) ILLUMIN:1 (E) CALIB:1 (E) NOTAPE:1 (E) TAPE:1 (E) PLAYBACK:1
(E) CATALOG:1 (E) RTITLE:1 (E) DATE_TIME:1 (E) NOTAPE:2 (E) TAPE_TYPE:2
(E) PLAYBACK:2 (E) FILE:2 (E) CATALOG:2 (E) RTITLE:2 (E) CATALOG:3

( T APE (TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALEGORITHM:1,
COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM (TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECN:1,
ITALGORITHM:1, ITFILENAME:1)
& DESCR (ITEM:1, NAME:1)
& FILEINFO (PB:1, FILE:1, NOTAPE:1)
& RECINFO (DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,
ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM (TAPEID:1)
)

v ( T APE (TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALEGORITHM:1,
COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM (TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECN:1,
ITALGORITHM:1, ITFILENAME:1)
& DESCR (ITEM:1, NAME:1)
& FILEINFO (PB:1, FILE:1, NOTAPE:1)
& RTAPE-PLAYBACK-CATALOG (NOTAPE:2, TAPE_TYPE:2, PLAYBACK:2,
PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1,
TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF,
RTITLE:2)
& RDATES-CATALOG (DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1,
ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM (TAPEID:1)
)

v ( T APE (TAPEID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
TPLASTORB:1, TPSTART:1, TPSTOP:1, TPALEGORITHM:1,
COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM (TAPEID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECN:1,
ITALGORITHM:1, ITFILENAME:1)
& DESCR (ITEM:1, NAME:1)
& RTAPE-PLAYBACK (NOTAPE:1, TAPE_TYPE:1, PLAYBACK:1, PB:1,
FILE:1, CATALOG:1, RTITLE:1)
& RECINFO (DATE_TIME:1, DATE:1, TIME:1, LON:1, LAT:1, ALT:1,
ZEN:1, PB:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF)
& NOTAPE:1 = NUM (TAPEID:1)
)
v ( TAPE(TMPSID:1, MISSION:1, ERB, FORMAT:1, PROJNUM:1, GENDATE:1,
    INVDATE:1, ARCHIVER:1, NUMFILES:1, TPFIRSTORB:1,
    TLASTORB:1, TPSSTART:1, TPSSTOP:1, TPALGORITHM:1,
    COORDSYS:1, SYNOPSTART:1, SYNOPSTOP:1)
& ITEM(TMPSID:1, FILE:1, ITEM:1, ITSTART:1, ITSTOP:1, RECNUM:1,
    ITALGORITHM:1, ITLEN:1)
& DESCRTMPSITEM:1, NAME:1)
& RTAPE-PLAYBACK (NOTAPE:1, TAPEID:1, PLAYBACK:1, PB:1,
    FILE:1, CATALOG:1, RTITLE:1)
& RTAPE-PLAYBACK-CATALOG (NOTAPE:1, TAPEID:2, PLAYBACK:2,
    PB:1, FILE:2, CATALOG:2, LON:1, LAT:1, ALT:1, ZEN:1,
    TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1, OFF,
    RTITLE:2)
& RDATES-CATALOG (DATE:1, CATALOG:3, LON:1, LAT:1, ALT:1,
    ZEN:1, TIME:1, QUALITY:1, ON, ILLUMIN:1, CALIB:1,
    OFF)
& NOTAPE:1 = NUM(TMPSID:1) } )
Figure 1.1 The global data manager — new version
Figure 1.2 The global data manager - old version

Global DML query on Global View: EXTERNAL-TO LOGICAL
and Global View: CONCEPTUAL OPTIMIZATION

Uniform DML query:

Decomposed DML query:

User's GLOBAL DML query: LOCAL DBMS's

Global View: MODEL EXECUTION

User: GENERATION

Local DML subqueries:

Subquery:

GLOBAL Execution CALL DIRECTORY plan and code: SUBQUERY TRANSLATION EXECUTION

Local Results RETURN from subquery CALL results

RESULTS INTEGRATION

Final Results

GLOBAL DATA MANAGER
Figure 2.1 An instance of the ERB-ORAC database

<table>
<thead>
<tr>
<th>TAPEINFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTAPE</td>
</tr>
<tr>
<td>1003</td>
</tr>
<tr>
<td>1004</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FILEINFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB</td>
</tr>
<tr>
<td>174000</td>
</tr>
<tr>
<td>174010</td>
</tr>
<tr>
<td>174570</td>
</tr>
<tr>
<td>174590</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECINFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE_TIME</td>
</tr>
<tr>
<td>QUALITY</td>
</tr>
<tr>
<td>790103124549</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>790105145629</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>790117110204</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>790107105349</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>
Figure 2.2 An instance of the ERB-SEED database

<table>
<thead>
<tr>
<th>RTAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO TAPE</td>
</tr>
<tr>
<td>TAPE TYPE</td>
</tr>
<tr>
<td>PB NO CATALOG FILE</td>
</tr>
<tr>
<td>LON LAT ALT ZEN</td>
</tr>
<tr>
<td>TIME1 QUALITY ELECTR</td>
</tr>
<tr>
<td>ILLUMIN CALIB SCAN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLAYBACK</th>
<th>RTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010:SDT</td>
<td>e(PLAYBACK):1</td>
</tr>
<tr>
<td>81261.4</td>
<td>4</td>
</tr>
<tr>
<td>-133.69</td>
<td>-55.62</td>
</tr>
<tr>
<td>1012076</td>
<td>1</td>
</tr>
<tr>
<td>TWILIGHT</td>
<td>NO</td>
</tr>
<tr>
<td>DATA</td>
<td></td>
</tr>
<tr>
<td>2010:SDT</td>
<td>e(PLAYBACK):2</td>
</tr>
<tr>
<td>81261.0</td>
<td>3</td>
</tr>
<tr>
<td>155.04</td>
<td>-80.07</td>
</tr>
<tr>
<td>1021349</td>
<td>0</td>
</tr>
<tr>
<td>DAY</td>
<td>NO</td>
</tr>
<tr>
<td>DATA</td>
<td></td>
</tr>
<tr>
<td>DATE1</td>
<td>CATALOG</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>LON</td>
<td>LAT</td>
</tr>
<tr>
<td>-133.69</td>
<td>-55.62</td>
</tr>
<tr>
<td>TWILIGHT: NO</td>
<td>OFF</td>
</tr>
<tr>
<td>-134.25</td>
<td>-56.47</td>
</tr>
<tr>
<td>TWILIGHT: NO</td>
<td>OFF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>790201</th>
<th>e(CATALOG):2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LON</td>
<td>LAT</td>
</tr>
<tr>
<td>155.04</td>
<td>-80.07</td>
</tr>
<tr>
<td>DAY</td>
<td>NO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RDATEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE1</td>
</tr>
<tr>
<td>LON</td>
</tr>
<tr>
<td>-133</td>
</tr>
<tr>
<td>TWILIGHT: NO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>790201</th>
<th>e(CATALOG):2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LON</td>
<td>LAT</td>
</tr>
<tr>
<td>155.04</td>
<td>-80.07</td>
</tr>
<tr>
<td>DAY</td>
<td>NO</td>
</tr>
<tr>
<td>LON</td>
<td>LAT</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>133.69</td>
<td>-55.62</td>
</tr>
<tr>
<td>134.25</td>
<td>-56.47</td>
</tr>
<tr>
<td>155.04</td>
<td>-80.07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LON</th>
<th>LAT</th>
<th>ALT</th>
<th>ZEN</th>
<th>TIME1</th>
<th>QUALITY</th>
<th>ELECTR</th>
<th>ILLUMIN</th>
<th>CALIB</th>
<th>SCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>133.69</td>
<td>-55.62</td>
<td>1123.70</td>
<td>98.71</td>
<td>012076</td>
<td>1</td>
<td>ON</td>
<td>TWILIGHT: NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>134.25</td>
<td>-56.47</td>
<td>1123.90</td>
<td>97.82</td>
<td>012109</td>
<td>1</td>
<td>ON</td>
<td>TWILIGHT: NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>155.04</td>
<td>-80.07</td>
<td>1126.80</td>
<td>66.65</td>
<td>021349</td>
<td>0</td>
<td>ON</td>
<td>DAY: NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>RZEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IZEN:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATALOG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LON</td>
<td>LAT</td>
<td>ALT</td>
<td>ZEN</td>
<td>TIME1</td>
<td>QUALITY</td>
<td>ELECTR</td>
<td>ILLUMIN</td>
<td>CALIB</td>
<td>SCAN</td>
</tr>
<tr>
<td>98</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e(CATALOG): 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-133.69</td>
<td>-55.62</td>
<td>1123.70</td>
<td>98.71</td>
<td>012076</td>
<td>1</td>
<td>ON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TWILIGHT</td>
<td>NO</td>
<td>I/OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e(CATALOG): 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-134.25</td>
<td>-56.47</td>
<td>1123.90</td>
<td>97.82</td>
<td>012109</td>
<td>1</td>
<td>ON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TWILIGHT</td>
<td>NO</td>
<td>I/OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e(CATALOG): 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>155.04</td>
<td>-80.07</td>
<td>1126.80</td>
<td>166.65</td>
<td>021349</td>
<td>0</td>
<td>ON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>NO</td>
<td>I/OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONS TITLE1, TITLE2, TITLE3

<table>
<thead>
<tr>
<th>NOTAPE</th>
<th>TITLE1</th>
<th>TITLE2</th>
<th>TITLE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010</td>
<td>NIMBUS6...</td>
<td>BY...</td>
<td>DATA...</td>
</tr>
<tr>
<td>2010</td>
<td>NIMBUS7...</td>
<td>BY...</td>
<td>DATA...</td>
</tr>
</tbody>
</table>
Figure 2.3 An instance of the PCDB database

<table>
<thead>
<tr>
<th>TAPE</th>
<th>TAPEID</th>
<th>MISSION</th>
<th>SENSOR</th>
<th>FORMAT</th>
<th>PROJNUM</th>
<th>GENDATE</th>
<th>INVDATE</th>
<th>ARCHIVER</th>
<th>NUMFILES</th>
<th>TPFIRSTORB</th>
<th>TPLASTORB</th>
<th>TPSTART</th>
<th>TPSTOP</th>
<th>TPALGORITHM</th>
<th>COORDSYS</th>
<th>SYNOPSTART</th>
<th>SYNOPSTOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>NIMBUS6</td>
<td>ERB</td>
<td>FARM-SS</td>
<td>00080</td>
<td>198208130000</td>
<td>19830406</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMITH</td>
<td>9</td>
<td>10000</td>
<td>12000</td>
<td>197812020000</td>
<td>197812020000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>197905310000</td>
<td>1</td>
<td>GEOMAC</td>
<td>197841500000</td>
<td>197841500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>NIMBUS6</td>
<td>ERB</td>
<td>FARM-LD</td>
<td>00082</td>
<td>198002100000</td>
<td>19830406</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMITH</td>
<td>10</td>
<td>8000</td>
<td>9000</td>
<td>197812042100</td>
<td>197812042100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>197812082100</td>
<td>2</td>
<td>GEODETIC</td>
<td>197851600000</td>
<td>197851600000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FILE</th>
<th>TAPEID</th>
<th>FILENUM</th>
<th>FLFIRSTORB</th>
<th>FLLASTORB</th>
<th>FLSTART</th>
<th>FLSTOP</th>
<th>FLALGORITHM</th>
<th>NUMITEMS</th>
<th>FLLLEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>35</td>
<td>38</td>
<td>197711500000</td>
<td>197711500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>10</td>
<td>104800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>36</td>
<td>39</td>
<td>197812000000</td>
<td>197812000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>12</td>
<td>106800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITEM</th>
<th>TAPEID</th>
<th>FILE</th>
<th>ITEM</th>
<th>ITSTART</th>
<th>ITSTOP</th>
<th>RECI</th>
<th>ITAL</th>
<th>ITLEN</th>
<th>NUM</th>
<th>NUMALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>WMON</td>
<td>197711500000</td>
<td>197711800000</td>
<td>43</td>
<td>1</td>
<td>3000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>MAP</td>
<td>197812000000</td>
<td>197812200000</td>
<td>31</td>
<td>3</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAPEID</td>
<td>FILENUM</td>
<td>ITEM</td>
<td>CAT: FUNCTION</td>
<td>CATEGORY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>------</td>
<td>---------------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1003</td>
<td>2</td>
<td>WGMON: SST</td>
<td>YOU MAY...</td>
<td>SEA SURFACE TEMP...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>MAP</td>
<td>SST</td>
<td>YOU MAY...</td>
<td>SEA SURFACE TEMP...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGMON</td>
<td>WORLD GRID</td>
</tr>
<tr>
<td>MAP</td>
<td>MONTHLY MAP</td>
</tr>
</tbody>
</table>
Figure 2.4 The induced ERB-R database instance

```
TAPEINFO
NOTAPE: TAPETYPE: TITLE1 | TITLE2 | TITLE3
1010 | SDT | NIMBUS6... | BY... | DATA...
2010 | SDT | NIMBUS7... | BY... | DATA...

FILEINFO
PB !FILE! NOTAPE:
81261.4! 4 | 1010 |
81261.0! 3 | 2010 |

RECINFO
DATE | TIME | LON | LAT | ALT | ZEN | PB | QUALITY
| ELECTR | ILLUMIN | CALIB | SCAN |
790101:012076!-133.69!-55.62!1123.70!98.71!81261.4! 1
| ON | TWILIGHT: NO | OFF |
790101:012109!-134.25!-56.47!1123.90!97.82!81261.4! 1
| ON | TWILIGHT: NO | OFF |
790201:021349!155.04!-80.07!1126.80!66.65!81261.0! 0
| ON | DAY | NO | OFF |
```
Figure 2.5 The induced ERB-N database instance

<table>
<thead>
<tr>
<th>RTAPE</th>
<th>TAPE</th>
<th>PLAYBACK</th>
<th>RTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO TAPE PLAYBACK</td>
<td>TAPE</td>
<td>CATALOG</td>
<td>TITLE</td>
</tr>
<tr>
<td>PB</td>
<td>NO</td>
<td>FILE</td>
<td>LON</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1003</th>
<th>SDT</th>
<th>e(PLAYBACK):3</th>
<th>e(RTITLE):1</th>
</tr>
</thead>
<tbody>
<tr>
<td>17400.0</td>
<td>2</td>
<td>e(CATALOG):5</td>
<td>NIMBUS6...</td>
</tr>
<tr>
<td>158.57</td>
<td>-1.00</td>
<td>1112.50</td>
<td>153.63</td>
</tr>
<tr>
<td>124549</td>
<td>0</td>
<td>ON</td>
<td>NIGHT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17401.0</th>
<th>3</th>
<th>e(CATALOG):6</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.82</td>
<td>10.84</td>
<td>1103.00</td>
</tr>
<tr>
<td>105349</td>
<td>8</td>
<td>OFF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1004</th>
<th>SDT</th>
<th>e(PLAYBACK):4</th>
<th>e(RTITLE):2</th>
</tr>
</thead>
<tbody>
<tr>
<td>17457.0</td>
<td>2</td>
<td>e(CATALOG):7</td>
<td>NIMBUS6...</td>
</tr>
<tr>
<td>2.75</td>
<td>10.43</td>
<td>1105.00</td>
<td>34.11</td>
</tr>
<tr>
<td>110204</td>
<td>0</td>
<td>ON</td>
<td>NIGHT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17459.0</th>
<th>3</th>
<th>e(CATALOG):8</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.90</td>
<td>-0.96</td>
<td>1112.30</td>
</tr>
<tr>
<td>145629</td>
<td>0</td>
<td>ON</td>
</tr>
</tbody>
</table>
FUNCTIONS TITLE1, TITLE2, TITLE3

-82-
Figure 2.6 An instance of the GLOBE database

```
TAPETYPE: TITLE1 TITLE2 TITLE3
1003: SDT: NIMBUS6... BY... DATA...
1004: SDT: NIMBUS6... BY... DATA...

FILEINFO
PB: FILE: NOTAPE
174000: 2: 1003
174010: 3: 1003
174570: 2: 1004
174590: 3: 1004

RECINFO
DATE_TIME DATE TIME LON LAT ALT ZEN PB QUALITY ELECTR ILLUM CALIB SCAN
0 ON NIGHT NO OFF
0 ON NIGHT NO OFF
0 ON NIGHT NO OFF
8 OFF DAY YES ON
```
<table>
<thead>
<tr>
<th>RTAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1010</td>
</tr>
<tr>
<td>1812614</td>
</tr>
<tr>
<td>13369</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>13425</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>DATA...</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>2010</td>
</tr>
<tr>
<td>1812610</td>
</tr>
<tr>
<td>15504</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>BY...</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>DATA...</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>DATE</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>LON</td>
</tr>
<tr>
<td>790101</td>
</tr>
<tr>
<td>-13369</td>
</tr>
<tr>
<td>-13425</td>
</tr>
<tr>
<td>790201</td>
</tr>
<tr>
<td>15504</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LON</th>
<th>CATALOG</th>
</tr>
</thead>
<tbody>
<tr>
<td>LON</td>
<td>LAT</td>
</tr>
<tr>
<td>-133</td>
<td>e(CATALOG):3</td>
</tr>
<tr>
<td>-13369</td>
<td>-5562</td>
</tr>
<tr>
<td>-134</td>
<td>e(CATALOG):4</td>
</tr>
<tr>
<td>-13425</td>
<td>-5647</td>
</tr>
<tr>
<td>155</td>
<td>e(CATALOG):2</td>
</tr>
<tr>
<td>15504</td>
<td>-8007</td>
</tr>
</tbody>
</table>
### RZEN

IZEN: CATALOG

- **LON**: LAT | ALT | ZEN | TIME | QUALITY | ELECTR | ILLUMIN | CALIB | SCAN |

<table>
<thead>
<tr>
<th>98</th>
<th>(CATALOG):3</th>
</tr>
</thead>
<tbody>
<tr>
<td>-13369</td>
<td>-5562</td>
</tr>
<tr>
<td>TWILIGHT: NO</td>
<td>OFF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>97</th>
<th>(CATALOG):4</th>
</tr>
</thead>
<tbody>
<tr>
<td>-13425</td>
<td>-5647</td>
</tr>
<tr>
<td>TWILIGHT: NO</td>
<td>OFF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>66</th>
<th>(CATALOG):2</th>
</tr>
</thead>
<tbody>
<tr>
<td>15504</td>
<td>-8007</td>
</tr>
<tr>
<td>DAY: NO</td>
<td>OFF</td>
</tr>
</tbody>
</table>

---

### TAPE

- **TAPEID**: MISSION: SENSOR: FORMAT: PROJNUM: GENDATE: INVDATE |

| 1003: NIMBUS6: ERB: PARM-SS: 00080 | 198208130000 | 19830406 |
| SMITH: 9 | 10000 | 12000 | 197812020000 |
| 197905310000 | 1: GEDMAC | 197841500000 | 197841500000 |

| 1010: NIMBUS6: ERB: PARM-LD: 00082 | 198002100000 | 19830406 |
| SMITH: 10 | 8000 | 9000 | 197812042100 |
| 197812082100 | 2: GEDDETEC | 197851600000 | 197851600000 |

---

-87-
### FILE

<table>
<thead>
<tr>
<th>TAPEID</th>
<th>FILE</th>
<th>FLFirstOrRb</th>
<th>FLLastOrRb</th>
<th>FLStart</th>
<th>FLStop</th>
<th>FLAlgorithm</th>
<th>NUMITEMS</th>
<th>FLLEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>35</td>
<td>38</td>
<td>197711500000</td>
<td>197711800000</td>
<td>1</td>
<td>10</td>
<td>104800</td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>36</td>
<td>39</td>
<td>197812000000</td>
<td>197812200000</td>
<td>3</td>
<td>12</td>
<td>106800</td>
</tr>
</tbody>
</table>

### ITEM

<table>
<thead>
<tr>
<th>TAPEID</th>
<th>FILE</th>
<th>ITEM</th>
<th>ITStart</th>
<th>ITStop</th>
<th>REC</th>
<th>ITAL</th>
<th>ITLEN</th>
<th>NUM</th>
<th>ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>WGMON</td>
<td>197711500000</td>
<td>197711800000</td>
<td>43</td>
<td>1</td>
<td>3000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>MAP</td>
<td>197812000000</td>
<td>197812200000</td>
<td>31</td>
<td>3</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CAT

<table>
<thead>
<tr>
<th>TAPEID</th>
<th>FILE</th>
<th>ITEM</th>
<th>CAT</th>
<th>FUNCTION</th>
<th>CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>WGMON</td>
<td>SST</td>
<td>YOU MAY...</td>
<td>SEA SURFACE TEMP...</td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>MAP</td>
<td>SST</td>
<td>YOU MAY...</td>
<td>SEA SURFACE TEMP...</td>
</tr>
</tbody>
</table>

### DESCR

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGMON</td>
<td>WORLD GRID</td>
</tr>
<tr>
<td>MAP</td>
<td>MONTHLY MAP</td>
</tr>
</tbody>
</table>
FUNCTIONS TITLE1, TITLE2, TITLE3

<table>
<thead>
<tr>
<th>NOTAPE</th>
<th>TITLE1</th>
<th>TITLE2</th>
<th>TITLE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>NIMBUS6...</td>
<td>BY...</td>
<td>DATA...</td>
</tr>
<tr>
<td>1004</td>
<td>NIMBUS6...</td>
<td>BY...</td>
<td>DATA...</td>
</tr>
<tr>
<td>1010</td>
<td>NIMBUS6...</td>
<td>BY...</td>
<td>DATA...</td>
</tr>
<tr>
<td>2010</td>
<td>NIMBUS7...</td>
<td>BY...</td>
<td>DATA...</td>
</tr>
</tbody>
</table>
Figure 2.7 The induced GLOBE-R database instance

---

TAPEINFO

NOTAPE: TAPETYPE: TITLE1 : TITLE2 : TITLE3

1003 : SDT : NIMBUS6... : BY... : DATA...

1004 : SDT : NIMBUS6... : BY... : DATA...

1010 : SDT : NIMBUS6... : BY... : DATA...

2010 : SDT : NIMBUS7... : BY... : DATA...

---

FILEINFO

PB : FILE : NOTAPE

174000: 2 : 1003

174010: 3 : 1003

174570: 2 : 1004

174590: 3 : 1004

812614: 4 : 1010

812610: 3 : 2010

---
### RECINFO

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>LON</th>
<th>LAT</th>
<th>ALT</th>
<th>ZEN</th>
<th>PB</th>
<th>QUALITY</th>
<th>ELECTR</th>
<th>ILLUMIN</th>
<th>CALIB</th>
<th>SCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>790103</td>
<td>124549</td>
<td>-100</td>
<td>111250</td>
<td>15363</td>
<td>174000</td>
<td>0</td>
<td>ON</td>
<td>NIGHT</td>
<td>NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>790105</td>
<td>145629</td>
<td>-96</td>
<td>111230</td>
<td>15397</td>
<td>174590</td>
<td>0</td>
<td>ON</td>
<td>NIGHT</td>
<td>NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>790117</td>
<td>110204</td>
<td>1043</td>
<td>110500</td>
<td>3411</td>
<td>174570</td>
<td>0</td>
<td>ON</td>
<td>NIGHT</td>
<td>NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>790107</td>
<td>105349</td>
<td>1084</td>
<td>110300</td>
<td>3559</td>
<td>174010</td>
<td>8</td>
<td>OFF</td>
<td>DAY</td>
<td>YES</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>790101</td>
<td>012076</td>
<td>-13369</td>
<td>-5562</td>
<td>112370</td>
<td>19871</td>
<td>812614</td>
<td>1</td>
<td>TWILIGHT</td>
<td>NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>790101</td>
<td>012109</td>
<td>-13425</td>
<td>-5647</td>
<td>112390</td>
<td>19782</td>
<td>812614</td>
<td>1</td>
<td>TWILIGHT</td>
<td>NO</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>790201</td>
<td>021349</td>
<td>-8007</td>
<td>112680</td>
<td>6665</td>
<td>1812610</td>
<td>0</td>
<td>ON</td>
<td>DAY</td>
<td>NO</td>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>

### TAPE

<table>
<thead>
<tr>
<th>TAPEID</th>
<th>MISSION</th>
<th>SENSOR</th>
<th>FORMAT</th>
<th>PROJNUM</th>
<th>GENDATE</th>
<th>INVDATE</th>
<th>ARCHIVER</th>
<th>NUMFILES</th>
<th>TPFIRSTORB</th>
<th>TPLASTORB</th>
<th>TPSTART</th>
<th>TPSTOP</th>
<th>TPALGORITHM</th>
<th>COORDSYS</th>
<th>SYNOPSTART</th>
<th>SYNOPSTOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>NIMBUS6</td>
<td>ERB</td>
<td>PARM-SS</td>
<td>00080</td>
<td>1982081300000</td>
<td>19830406</td>
<td>SMITH</td>
<td>9</td>
<td>10000</td>
<td>12000</td>
<td>1978120200000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1979053100000</td>
<td>1</td>
<td>GEOMAC</td>
<td>197841500000</td>
<td>197841500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>NIMBUS6</td>
<td>ERB</td>
<td>PARM-LD</td>
<td>00082</td>
<td>1980021000000</td>
<td>19830406</td>
<td>SMITH</td>
<td>10</td>
<td>8000</td>
<td>9000</td>
<td>1978120420000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1978120821000</td>
<td>2</td>
<td>GEODETIC</td>
<td>197851600000</td>
<td>197851600000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### FILE

<table>
<thead>
<tr>
<th>TAPEID</th>
<th>FILE</th>
<th>FLFIRSTORB</th>
<th>FLLASTORB</th>
<th>FLSTART</th>
<th>FLSTOP</th>
<th>FLALGORITHM</th>
<th>NUMITEMS</th>
<th>FLLEN</th>
<th>FILEID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>35</td>
<td>38</td>
<td>1</td>
<td>10</td>
<td>13</td>
<td>104800</td>
<td></td>
<td>1003</td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>36</td>
<td>39</td>
<td>3</td>
<td>12</td>
<td>3</td>
<td>106800</td>
<td></td>
<td>1010</td>
</tr>
</tbody>
</table>

### ITEM

<table>
<thead>
<tr>
<th>TAPEID</th>
<th>FILE</th>
<th>ITEM</th>
<th>ITSTART</th>
<th>ITSTOP</th>
<th>RECITAL</th>
<th>ITLEN</th>
<th>NUMALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>WGMON</td>
<td>197711500000</td>
<td>197711800000</td>
<td>43</td>
<td>1</td>
<td>3000</td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>MAP</td>
<td>197812000000</td>
<td>197812200000</td>
<td>31</td>
<td>3</td>
<td>2000</td>
</tr>
</tbody>
</table>

### CAT

<table>
<thead>
<tr>
<th>TAPEID</th>
<th>FILE</th>
<th>ITEM</th>
<th>CAT</th>
<th>FUNCTION</th>
<th>CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>2</td>
<td>WGMON</td>
<td>SST</td>
<td>YOU MAY...</td>
<td>SEA SURFACE TEMP...</td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
<td>MAP</td>
<td>SST</td>
<td>YOU MAY...</td>
<td>SEA SURFACE TEMP...</td>
</tr>
</tbody>
</table>

### DESCR

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGMON</td>
<td>WORLD GRID</td>
</tr>
<tr>
<td>MAP</td>
<td>MONTHLY MAP</td>
</tr>
</tbody>
</table>

-92-
Figure 3.1  Answer to query 1 on the database of Figure 2.4

<table>
<thead>
<tr>
<th>NOTAPE</th>
<th>TAPETYPE</th>
<th>PB</th>
<th>FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010</td>
<td>SDT</td>
<td>81261.4</td>
<td>4</td>
</tr>
<tr>
<td>2010</td>
<td>SDT</td>
<td>81261.0</td>
<td>3</td>
</tr>
</tbody>
</table>

-93-
Figure 3.2 Answer to query 2 on the database of Figure 2.4

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>ILLUMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>790101:012076</td>
<td>TWILIGHT</td>
<td></td>
</tr>
<tr>
<td>790101:012109</td>
<td>TWILIGHT</td>
<td></td>
</tr>
<tr>
<td>790101:012109</td>
<td>TWILIGHT</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.3 Answer to query 3 on the database of Figure 2.5

<table>
<thead>
<tr>
<th>TAPETYPE:</th>
<th>PB</th>
<th>NOFILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDT</td>
<td>17400.0</td>
<td>2</td>
</tr>
<tr>
<td>SDT</td>
<td>17401.0</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 3.4 Answer to query 4 on the database of Figure 2.5

<table>
<thead>
<tr>
<th>NOTAPE</th>
<th>DATE1</th>
<th>TIME1</th>
<th>LON</th>
<th>LAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>790107</td>
<td>105349</td>
<td>4.82</td>
<td>10.84</td>
</tr>
<tr>
<td>1004</td>
<td>790117</td>
<td>110204</td>
<td>2.75</td>
<td>10.43</td>
</tr>
</tbody>
</table>
Table 4.1 Answer to query 5 on the database of Figure 2.7

<table>
<thead>
<tr>
<th>PROJNUM</th>
<th>NAME</th>
<th>ALT</th>
</tr>
</thead>
<tbody>
<tr>
<td>00080</td>
<td>WORLD GRID</td>
<td>111250</td>
</tr>
<tr>
<td>00082</td>
<td>MONTHLY MAP</td>
<td>112370</td>
</tr>
<tr>
<td>00082</td>
<td>MONTHLY MAP</td>
<td>112390</td>
</tr>
</tbody>
</table>