

# LDC Data Warehousing Technical Options

## Deliverable 12.2

Prepared for

Office of Wetlands, Oceans and Watersheds, US  
Environmental Protection Agency

August 8, 2002

GSA Delivery Order Number: OW-1472-NBLX  
Contract Number: GS-035F-4797H  
AMS Project Number: DD451

## Final

Prepared by



American Management Systems, Inc.  
12601 Fair Lakes Circle  
Fairfax, Virginia 22033

# Table of Contents

---

1	Introduction.....	1-1
1.1	Purpose of this Document.....	1-1
1.2	Background.....	1-2
1.3	Methodology.....	1-3
1.4	Assumptions.....	1-4
2	Required Data Elements.....	2-1
2.1	Application Analysis.....	2-1
2.2	Required Elements Summary.....	2-2
2.3	Web Application Components .....	2-5
2.3.1	Procedures .....	2-5
2.3.2	Report Templates .....	2-6
3	Dimensional Modeling Options .....	3-1
3.1	Introduction to Dimensional Modeling.....	3-1
3.1.1	Definitions.....	3-1
3.1.2	Advantages .....	3-2
3.2	Star versus Snowflake Schema.....	3-2
3.3	De-normalization Roadmap.....	3-4
3.3.1	Introduction.....	3-4
3.3.2	RESULT Table.....	3-6
3.3.3	STATION Table.....	3-7
3.3.4	SAMPLE Table.....	3-7
3.3.5	Summary .....	3-7
3.4	Aggregates.....	3-8
4	Oracle Warehousing Tools .....	4-1
4.1	Materialized Views.....	4-1
4.2	Bitmap Indexes and Performance Tuning .....	4-2
4.3	Constraints.....	4-3

4.4	Oracle Extraction, Transformation, and Loading Tools .....	4-3
4.5	Oracle Warehouse Builder .....	4-3
5	Data Model Recommendation.....	5-1
5.1	Logical Data Model.....	5-1
5.2	Data Elements.....	5-4
5.3	Data Element Transformation.....	5-9
6	Transitional Architecture and Data Archiving Scheme .....	6-1
6.1	Data Transition.....	6-1
6.2	Long-Term Data Storage .....	6-2
7	Next Steps.....	7-1
7.1	Final Document Production.....	7-1
7.2	Implementation Scripts .....	7-1
7.3	Application Design Options Paper.....	7-2
7.4	Re-engineering of LDC Web Retrievals .....	7-2
Appendix A: LDC Warehouse Data Dictionary.....		1
Data Elements and Definitions.....		1

# List of Figures

---

Figure 1-1	OLTP and Data Warehouse Comparison.....	1-3
Figure 2-2	Required Elements Summary .....	2-2
Figure 3-1	Star and Snowflake Schema .....	3-3
Figure 3-2	De-normalization Roadmap .....	3-6
Figure 5-1	LDC Warehouse Data Model .....	5-3
Figure 5-2	LDC Warehouse Data Elements and Mappings.....	5-5



# Introduction

---

*This section provides contextual information about the project, the value of this document, and the assumptions and methodology used in developing this document*

## 1.1 Purpose of this Document

---

The STORET Legacy Data Center (LDC) Data Warehousing Technical Options paper explores data warehousing techniques appropriate for the size, usage patterns, and architecture of the LDC. Data modeling options are analyzed as well as several Oracle warehousing technologies. Final database design recommendations are made following this analysis.

This paper is the first deliverable of the LDC Data Warehousing project that is targeted towards improving the performance of LDC web retrievals, reducing the overhead maintenance of the LDC database, and simplifying the process of creating new retrieval functionality and report templates.

The database design recommendations outlined in this document will serve as the first step in achieving these project goals.

Once the database has been reengineered, the Web application will be modified to leverage the performance improvements made possible through warehousing technology. These application modifications are not explored in this document, but will be addressed in the forthcoming STORET LDC Application Design Options paper (see Chapter 7, Next Steps).

Specifically, this document addresses the following topics:

*Introduction*—provides contextual information about the project, the value of this document, and the assumptions and methodology used in developing this document

*Required Data Elements*—provides an overview of data elements required by the current LDC web application. This includes data elements used to

support search and retrieval functionality, as well as those required by the existing report templates. The data warehouse must support all data elements identified in this section

*Dimensional Modeling Options*—provides a detailed analysis of how the current LDC data model should be transformed to a dimensional structure. Special attention is given to key decision points in the dimensional modeling process

*Oracle Warehousing Tools*—provides a detailed analysis of the Oracle data warehousing technologies that may be employed by the LDC. These include: materialized views, bitmap indexes, constraints, ETL tools, and the Oracle Warehouse Builder

*Data Model Recommendation*— summarizes the findings of the previous chapters and provides an initial dimensional model based on the recommendations made in this document

*Transitional Architecture and Data Archiving Scheme*— provides a guide for transitioning between the existing LDC database and the planned data warehouse. Special attention is given to the archiving of data that may be excluded from the data warehouse

*Next Steps*—discusses the deliverables that will be produced by AMS subsequent to this report and feedback received from EPA

*Appendix A*—lists the Attribute Name, Attribute Definition, Column Name, and Table Usage for each data element in the LDC data warehouse

## 1.2 Background

---

The STORET LDC houses data that was originally stored in Legacy STORET (a pre-relational mainframe system). Legacy STORET data was manipulated and transformed by both AMS and EPA to conform to a traditional normalized relational structure. The static nature of the LDC and advances in data warehousing technology have caused EPA to revisit the database design of the LDC in an attempt to improve performance and simplify maintenance and application development. Key differences between a data warehouse and traditional relational system (often referred to as an online transaction processing [OLTP] system) are listed in Figure 1-1.

Figure 1-1

OLTP and Data Warehouse Comparison

OLTP		Data Warehouse
Complex data structures (3NF databases)		Multidimensional data structures
Few	<b>Indexes</b>	Many
Many	<b>Joins</b>	Some
Normalized Table Structure	<b>Duplicate Data</b>	Denormalized Table Structure
Rare	<b>Derived data and Aggregates</b>	Common

By definition, a data warehouse is a data repository for an enterprise and is generally used for research and decision support. By comparison, an OLTP system is used to deal with the everyday running of one aspect of an enterprise. Operational systems are usually not designed for optimal performance for complex queries and report generation.

The major differences between OLTP and Data Warehouse systems lie in designing for update performance versus designing for query performance. Since the LDC contains only static data, a data warehouse design is the logical path of evolution for this system.

### 1.3 Methodology

Recommendations in this report are based on industry standards for data warehousing projects, AMS best practices, and AMS's knowledge of the STORET LDC system and the needs of its user base. Meetings were held between EPA and AMS to determine the scope of this project. This paper is



focused on exploring major decision points that were identified in these meetings.

## 1.4 Assumptions

---

The following assumptions were made in conducting this analysis:

The CANYON and ZION versions of the LDC database are identical

No alterations have been made to the structure of the LDC database during the preparation of this report

This document assumes the reader is knowledgeable of the data structure of the LDC database and is familiar with relational database concepts

## 2

# Required Data Elements

---

*This section provides an overview of data elements required by the current LDC web application. This includes data elements used to support search and retrieval functionality, as well as those required by the existing report templates. The data warehouse must support all data elements identified in this section*

## 2.1 Application Analysis

---

The current LDC web application was analyzed to determine the data elements that must be supported by the new dimensional model. The data elements highlighted in Figure 2-1 represent the data requirements of the new LDC data warehouse. It should be noted that some of the highlighted data elements are redundant since they exist in multiple tables (this is due to the migration of primary keys for the purpose of creating relationships). This redundancy will largely be eliminated through the de-normalization of the data model and subsequent table consolidation.

EPA has decided that the first version of the data warehouse will only contain data required by the LDC web application and the report templates currently offered by the LDC. This approach reduces the overall complexity of the initial project and guarantees that the warehouse size is held in check. Later versions of the data warehouse may expand to include additional tables and data elements (perhaps the full data content of the current LDC). For more details concerning the archiving of the LDC data elements not highlighted in Figure 2-1, see Chapter 6 (Transitional Architecture and Data Archiving Scheme).

Figure 2-1: Required LDC Data Elements

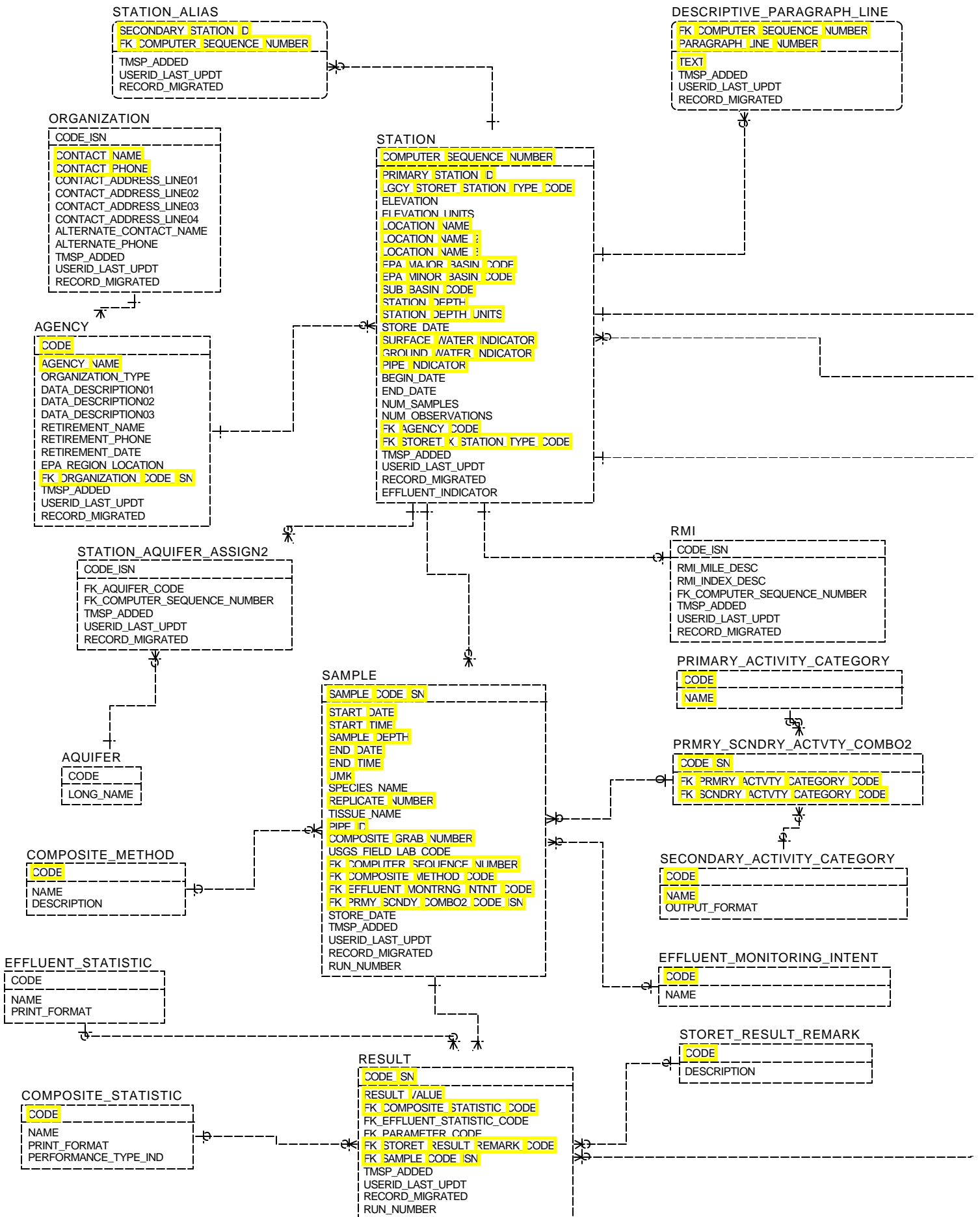


Figure 2-1: Required LDC Data Elements

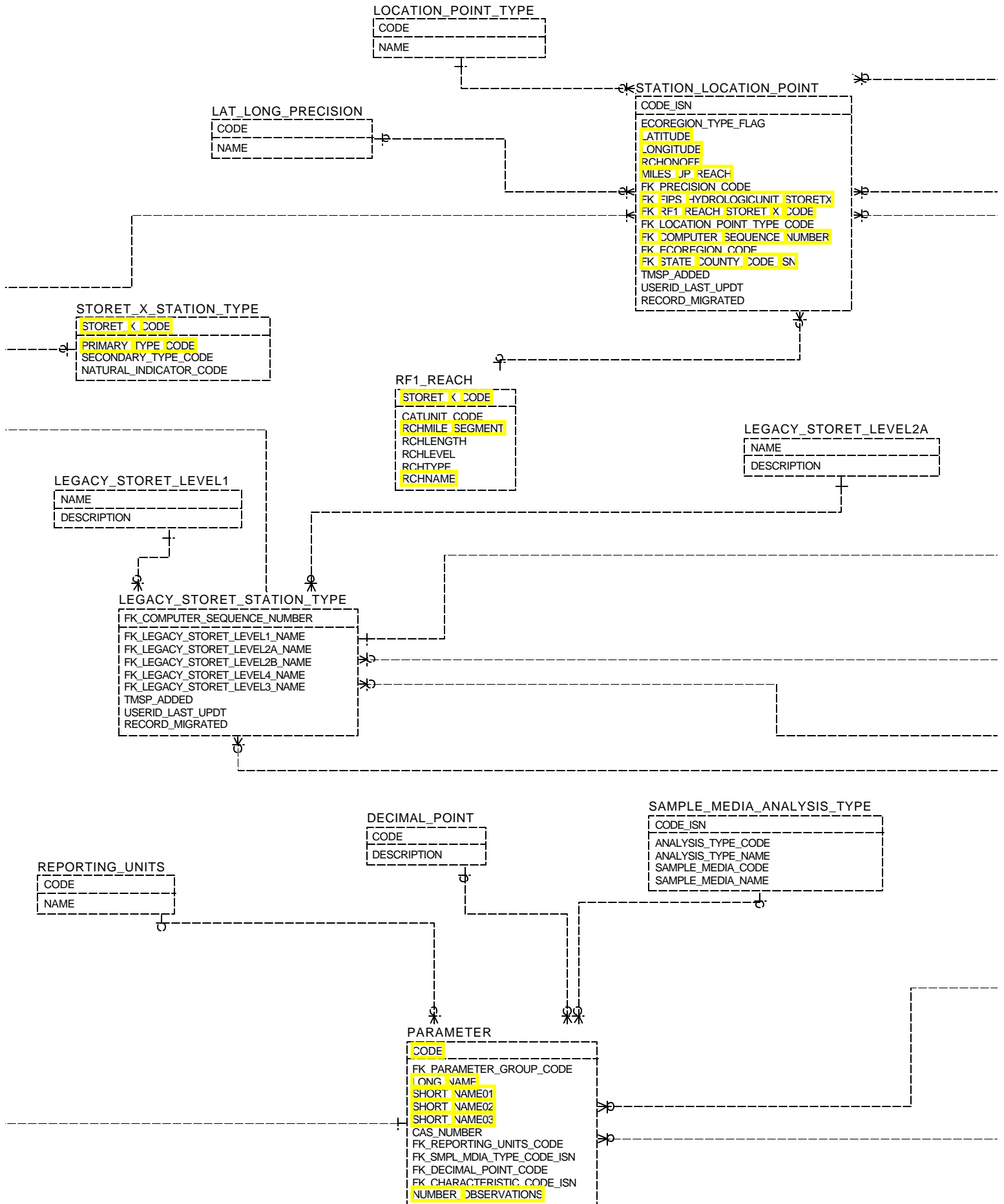
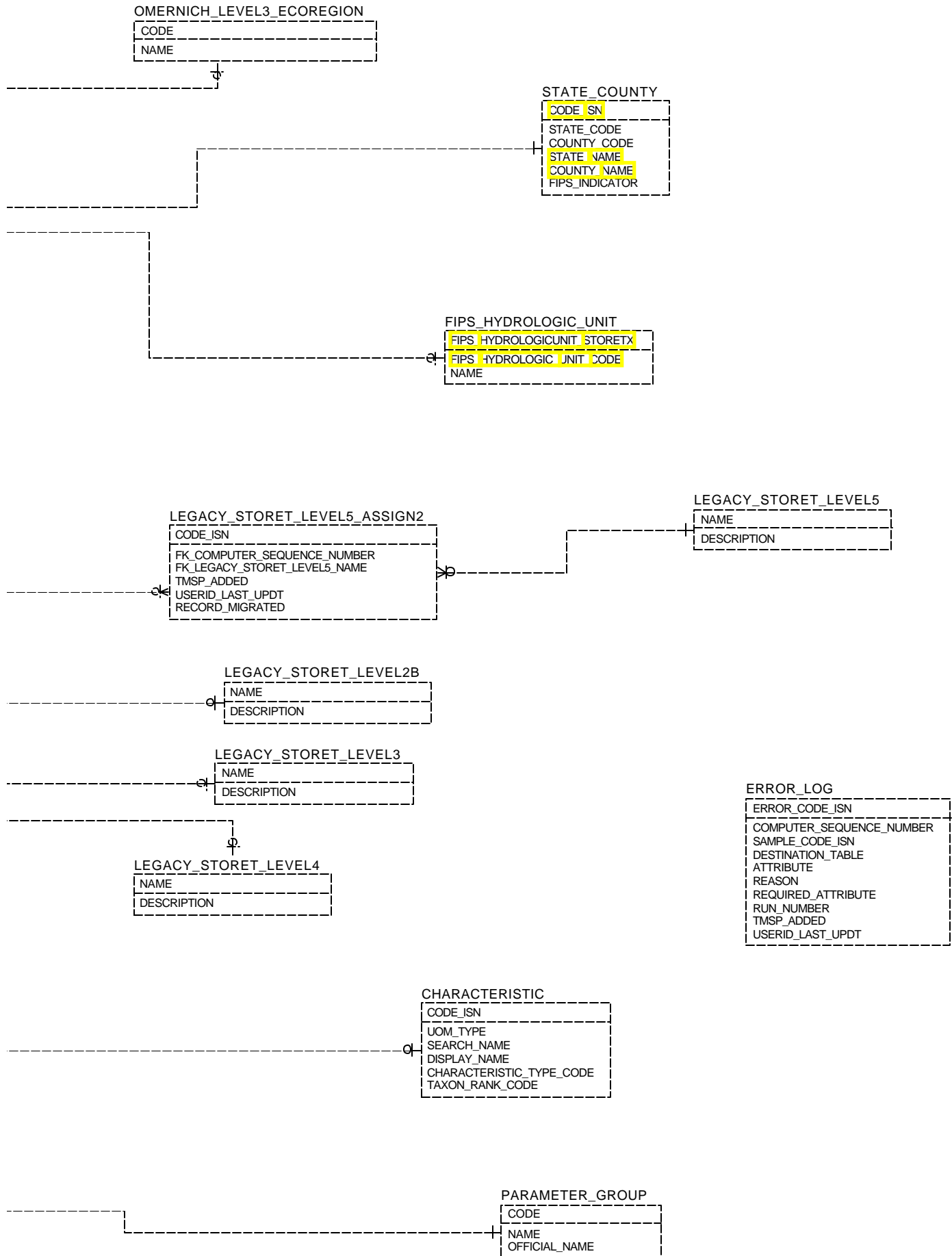


Figure 2-1: Required LDC Data Elements



## 2.2 Required Elements Summary

The required data elements are summarized in the following chart (Figure 2-2). The elements are organized by table name and are described in terms of data type and optionality. In addition, primary keys and foreign keys are identified.

Figure 2-2

Required Elements Summary

Table/Column Name	Data Type	Optionality	Key
PARAMETER			
code	CHAR (5)	NOT NULL	PRIMARY
long_name	VARCHAR (50)	NULL	
short_name01	CHAR (8)	NULL	
short_name02	CHAR (8)	NULL	
short_name03	CHAR (8)	NULL	
number_observations	NUMBER (12)	NULL	
RESULT			
code_isn	NUMBER (12)	NOT NULL	PRIMARY
result_value	VARCHAR2(12)	NULL	
fk_composite_statistic_code	CHAR(1)	NULL	FOREIGN
fk_parameter_code	CHAR(5)	NULL	FOREIGN
fk_storet_result_remark_code	CHAR(1)	NULL	FOREIGN
SAMPLE			
sample_code_isn	NUMBER (12)	NOT NULL	PRIMARY
start_date	DATE	NULL	
start_time	NUMBER (4)	NULL	
sample_depth	NUMBER (8,2)	NULL	
end_date	DATE	NULL	
end_time	NUMBER (4)	NULL	
umk	CHAR (8)	NULL	
replicate_number	NUMBER (12)	NULL	

Table/Column Name	Data Type	Optionality	Key
pipe_id	NUMBER (12)	NULL	
composite_grab_number	NUMBER (2)	NULL	
fk_composite_method_code	CHAR (1)	NULL	FOREIGN
fk_effluent_montrng_intnt_code	CHAR (2)	NULL	FOREIGN
fk_prmy_scndy_combo2_code_isn	NUMBER (12)	NULL	FOREIGN
fk_computer_sequence_number	NUMBER (12)	NOT NULL	FOREIGN
STATION			
computer_sequence_number	NUMBER (12)	NOT NULL	PRIMARY
primary_station_id	CHAR (15)	NOT NULL	
lgcy_storet_station_type_code	VARCHAR2 (65)	NULL	
location_name	VARCHAR2 (48)	NULL	
location_name_2	CHAR (60)	NULL	
location_name_3	CHAR (60)	NULL	
epa_major_basin_code	CHAR (2)	NULL	
epa_minor_basin_code	CHAR (2)	NULL	
sub_basin_code	CHAR (2)	NULL	
station_depth	NUMBER (3)	NULL	
station_depth_units	VARCHAR2 (20)	NULL	
surface_water_indicator	CHAR (1)	NULL	
ground_water_indicator	CHAR (1)	NULL	
pipe_indicator	CHAR (1)	NULL	
fk_agency_code	NUMBER (12)	NOT NULL	FOREIGN
fk_storet_x_station_type_code	NUMBER (12)	NULL	FOREIGN
STATION_ALIAS			
fk_computer_sequence_number	NUMBER (12)	NOT NULL	PRIMARY/ FOREIGN
secondary_station_id	CHAR (12)	NOT NULL	PRIMARY
DESCRIPTIVE_PARAGRAPH_LINE			
fk_computer_sequence_number	NUMBER (12)	NOT NULL	PRIMARY/ FOREIGN

Table/Column Name	Data Type	Optionality	Key
paragraph_line_number	NUMBER (2)	NOT NULL	PRIMARY
text	CHAR (72)	NULL	
PRIMARY_ACTIVITY_CATEGORY			
name	VARCHAR2 (60)	NULL	
SECONDARY_ACTIVITY_CATEGORY			
name	VARCHAR2 (60)	NULL	
STATION_LOCATION_POINT			
latitude	NUMBER (8,6)	NULL	
longitude	NUMBER (9,6)	NULL	
rchonoff	CHAR (3)	NULL	
miles_up_reach	FLOAT	NULL	
fk_rf1_reach_storet_x_code	NUMBER (12)	NULL	FOREIGN
fk_computer_sequence_number	NUMBER (12)	NOT NULL	FOREIGN
fk_state_county_code_isn	NUMBER (12)	NOT NULL	FOREIGN
fk_fips_hydrologicunit_storetx	NUMBER (12)	NULL	FOREIGN
FIPS_HYDROLOGIC_UNIT			
fips_hydrologicunit_storetx	NUMBER (12)	NOT NULL	PRIMARY
fips_hydrologic_unit_code	CHAR (8)	NULL	
STATE_COUNTY			
code_isn	NUMBER (12)	NOT NULL	PRIMARY
state_name	CHAR (30)	NULL	
county_name	CHAR (30)	NULL	
RF1_REACH			
storet_x_code	NUMBER (12)	NOT NULL	PRIMARY
rchmile_segment	NUMBER (3)	NULL	
rchname	VARCHAR2 (60)	NULL	
STORET_X_STATION_TYPE			
storet_x_code	NUMBER (12)	NOT NULL	PRIMARY
primary_type_cde	VARCHAR2 (30)	NULL	
AGENCY			



Table/Column Name	Data Type	Optionality	Key
code	CHAR (8)	NOT NULL	PRIMARY
agency_name	VARCHAR2 (25)	NULL	
ORGANIZATION			
contact_name	VARCHAR2 (25)	NOT NULL	
contact_phone	CHAR (13)	NOT NULL	

---

These data elements serve as the foundation for the dimensional model presented in Chapter 5.

## 2.3 Web Application Components

---

The following list contains the names of the LDC application components (procedures and report templates) that interact directly with the LDC database. The analysis of these components produced the list of required data elements identified in this chapter.

### 2.3.1 Procedures

---

proc\_aggregate\_data  
 proc\_count\_adv  
 proc\_count\_samples  
 proc\_count\_simple  
 proc\_count\_stations  
 proc\_list\_contacts  
 proc\_select\_pcodes\_dates  
 proc\_generte\_pcodes\_name  
 proc\_generate\_pcodes\_code  
 proc\_pass\_counties

proc\_pass\_pcodes\_search

proc\_pass\_stations

proc\_list\_huc\_codes

proc\_list\_counties

proc\_list\_stations

### 2.3.2 Report Templates

---

Station Description Report

Data Summary Report

Sample Data Report

Note: Two separate versions exist for each of these report types. One that produces HTML and PDF reports and another that produces comma-separated text reports. Both report versions were analyzed for each of the report types to ensure that no data elements were missed.

## 3

# Dimensional Modeling Options

---

*This section provides a detailed analysis of how the current LDC data model should be transformed to a dimensional structure. Special attention is given to key decision points in the dimensional modeling process*

## 3.1 Introduction to Dimensional Modeling

### 3.1.1 Definitions

---

Dimensional modeling is a logical design technique that seeks to present the data in a standard framework that is intuitive and allows for high-performance access. It is inherently dimensional and adheres to a discipline that uses the relational model with some important restrictions. Every dimensional model is composed of one central table with several foreign keys (called the fact table) and a set of smaller tables (called dimension tables). Each dimension table has a single part primary key that corresponds exactly to one of the foreign keys in the fact table. This relationship is often referred to as a 'star join'. At a more functional level, the roles of fact tables and dimension tables can be described as follows.

**Fact table**—the primary table in a dimensional model used to contain measurements of the enterprise

**Dimension table**—one of a set of companion tables associated with the fact table. Dimension tables are used for constraining and grouping within data warehouse queries

### 3.1.2 Advantages

---

The fundamental differences between entity-relationship models and dimensional models are parallel to the differences between data warehouses and OLTP systems (see Section 1.2). The advantages to using a dimensional model can be summarized as follows.

Performance is improved by decreasing the number of tables that must be scanned and eliminating complex joins that span numerous tables

The process of creating additional queries and report templates is simplified by the elimination of complex joins

Dimensional models are easily expanded to accept new types of data as the needs of the warehouse (and data provided from source systems) changes over time

Numerous decision support COTS (commercial off-the-shelf) tools can easily interface with a dimensional model (particularly a star-schema) due to the predictable, standard framework

## 3.2 Star versus Snowflake Schema

---

Much of the performance improvement realized by data warehousing is achieved through the use of de-normalization. However, it should be noted that not all data warehouses have a de-normalized table structure. A significant but smaller number of data warehouses use third-normal-form (3NF) schemas, or other schemas that are more normalized than star schemas (described below). These 3NF data warehouses are typically very large data warehouses, which are used primarily for loading data and for feeding data marts. These data warehouses are not typically used for heavy end-user query workloads.

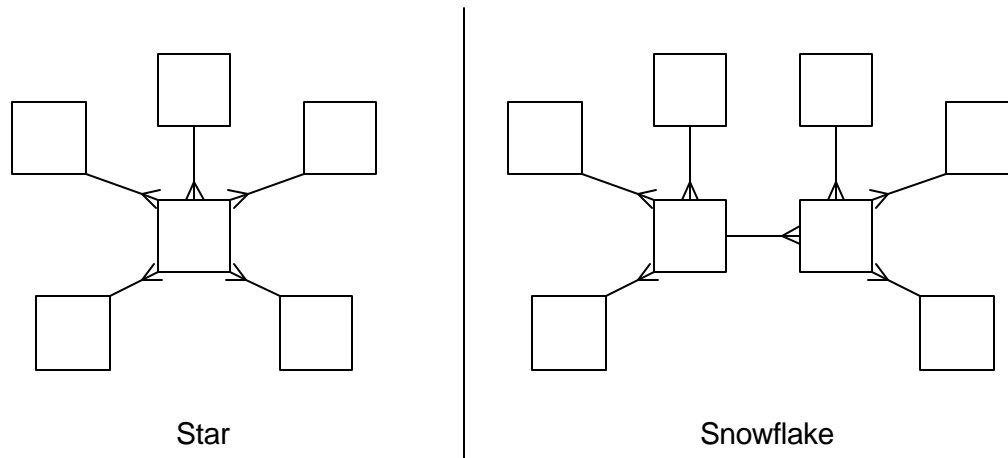
Since performance enhancement is the primary goal of the LDC warehousing project, only de-normalized structures should be considered. The level of de-normalization needs to be determined, however. Two main options exist—the star schema and the snowflake schema. Idealized views of both models are given in Figure 3-1. Definitions for the two types of data models are as follows:

**Star Schema**—one centralized fact table surrounded by several dimension tables (usually joined in a one-to-many relationship)

Snowflake Schema—similar to the star schema, but less symmetrical in nature. Dimension tables may have relationships with additional tables (not just the centralized fact table)

Figure 3-1

Star and Snowflake Schema



The current LDC application has two basic levels of focus—station information and sample/result information. It is logical to merge sample and result data around one central fact table since the two are always retrieved together and ‘results’ represent the lowest level of granularity housed in the LDC. However, station information can be retrieved independently of sample/result data (via the LDC’s Station Description report) and station selection is one of the primary drivers of data filtering employed by the LDC. Also, the station dimension is much larger and more complex than the LDC’s other dimensions.

For these reasons the benefits of maintaining a certain degree of normalization with regards to station information (and thereby using a snowflake schema design) must be explored. A snowflake design offers the following advantages when compared to a pure star-schema structure:

The more normalized structure of a snowflake design will reduce the overall size of the data warehouse

Maintainability is improved since data updates on station information would be easier to implement (i.e.—if organizational contact information

is stored at the STATION level, it will need to be update in several locations if that contact information changes over time)

Two special STATION data elements ('station alias' and 'descriptive paragraph') are currently stored in separate tables that have a one-to-many relationship with the STATION table. A snowflake design allows this structure to be maintained

Many of these considerations are reduced in importance by the unique nature of the LDC. Since only legacy data is housed, mass data alterations are not expected. For this reason the benefits of simplified data maintenance are minimal. Also, EPA has said that database size is less of a concern then performance when discussing goals for the warehouse.

The star schema offers the following advantages for the LDC:

- Maximized performance enhancement by minimizing the number of complex table joins

- Simplified code creation and maintenance by minimizing the number of complex table joins thereby simplifying SQL statements

Because of the static nature of the LDC and the current focus on performance enhancement, it is recommended that a star-schema version of the LDC be implemented. However, moderate 'snowflaking' of the STATION table is acceptable when there is a direct reason for it in the way the application is used. For instance, agency contact information is retrieved independently of station and sample/result information. For this reason, a strong argument can be made for separating out agency contact information in its own table structure (see Chapter 5 for more detail).

## 3.3 De-normalization Roadmap

### 3.3.1 Introduction

---

Figure 3-2 illustrates the logical de-normalization of the current LDC data model to a star schema. This process has created a structure comprised of one fact table and five dimension tables. The purpose of each of these tables is described below.

- RESULT— contains recorded results. This is the fact table of the LDC data warehouse and is associated with all five dimension tables described below

STATION—contains the recorded location where monitoring activities (e.g., collection of samples and measurements, observations) occur. This is the most complex dimension in the LDC data warehouse

PARAMETER— contains the legacy STORET parameter codes and names used to identify the "thing" being measured. Legacy STORET parameter codes also have a non-result usage. (Examples of non-result parameter codes include: sample qualifiers, result qualifiers, lab descriptors, people descriptors, and location descriptors)

SAMPLE—contains information about monitoring activity (e.g., ambient samples, measurements, observations, QC samples). Monitoring activities are performed at a specific date, time, and location in order to characterize the environment

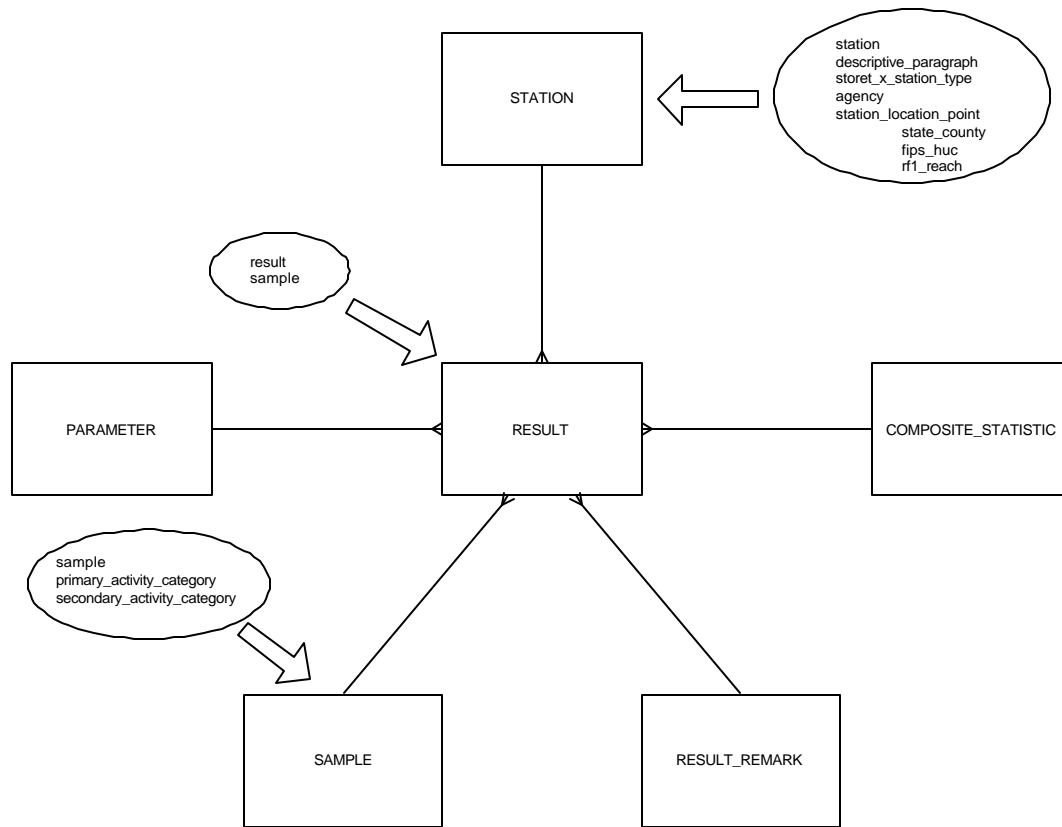
RESULT\_REMARK—contains a list of valid STORET remark codes and definitions used to further qualify a result.

COMPOSITE\_STATISTIC—contains composite statistic types and composite performance types for non-effluent samples. This table contains the following: 1. A list of statistics that STORET may store for a (non-effluent) composite of several grabs (e.g., average, max, min) 2. A list of performance types that STORET may store for a (non-effluent) composite of several grabs (e.g., accuracy, precision)

The three major focus areas for de-normalization are at the RESULT, STATION, SAMPLE levels. These are described in detail in the following sections.

Figure 3-2

### De-normalization Roadmap



### 3.3.2 RESULT Table

The **RESULT** table has been streamlined by the removal of unused data elements. The most significant change in the **RESULT** table from the original LDC design is that a direct relationship now exists between **STATION** and **RESULT**. This eliminates the need to join through **SAMPLE** when selecting **RESULT**s based on location.



### 3.3.3 STATION Table

---

All location information is consolidated and moved to the STATION table (eliminating the need for 4 additional tables and several complex joins). This will greatly speed queries that search for data based on location (which are the most common type of query supported by the LDC). This consolidation will require only one table to be searched for performing STATION counts and the majority of Station Description Reports. Also, only one table join will be required when searching for RESULTS based on STATION location.

Additional support tables such as DESCRIPTIVE\_PARAGRAPH\_LINE are also rolled-up to the STATION level. Even though this relationship is currently a one-to-many in the LDC, the maximum length of a station description is 15 lines of 72 characters. This is small enough to easily be housed in one column using a varchar2 data type (see Section 5.3 for more information on data transformation).

### 3.3.4 SAMPLE Table

---

The SAMPLE table has been streamlined to include only data elements and relations that are utilized by the LDC web application.

ACTIVITY\_CATEGORY information has been de-normalized and incorporated into the SAMPLE data for faster retrieval. SAMPLEs are primarily queried by date and depth. Data elements related to these topics are consolidated in the SAMPLE table.

### 3.3.5 Summary

---

The process of de-normalization has also eliminated the need for certain tables (associative entities) and data elements (internal keys) that are used by the current LDC web application (see Section 2.2).

This figure does not represent all of the tables recommended for the LDC data warehouse. It instead focuses on the key tables that are involved in the central 'star-schema' portion of the data model. Additional lookup tables that are not essential to report generation or complex database searched are shown in Chapter 5—Data Model Recommendations. The current section is focused on the de-normalization of the central LDC tables to form the dimensional portion of the new LDC data warehouse.

## 3.4 Aggregates

---

One of the advantages of a data warehouse system is the ability to pre-calculate expected queries and save that information in the database. This information is then made available to users very quickly without the need for recalculating values for each specific request. These 'pre-calculations' often take the form of aggregates. Oracle utilizes a special technology dedicated to supporting aggregates and other pre-calculated database representations known as Materialized Views. See Section 4.1 for more information on Materialized Views and their potential role in the LDC data warehouse.

# Oracle Warehousing Tools

---

*This section provides a detailed analysis of the Oracle data warehousing technologies that may be employed by the LDC. These include: materialized views, bitmap indexes, constraints, ETL tools, and the Oracle Warehouse Builder*

## 4.1 Materialized Views

---

Materialized views are sometimes used in warehousing to increase the speed of database queries. Materialized views improve query performance by pre-calculating expensive join and aggregation operations on the database prior to execution time and storing these results in the database. The query optimizer can make use of materialized views by automatically recognizing when an existing materialized view can and should be used to satisfy a request. It then rewrites the request to use the materialized view and queries are directed to the materialized view instead of the underlying tables or views (this process is transparent to the user of the system). Rewriting queries to use materialized views rather than detail relations can result in significant performance gain.

Materialized views are often used to calculate and store aggregate information (such as record counts) for logical blocks of data (i.e.—calculate the number of samples for each station in monthly intervals). The problem with incorporating these types of pre-calculated aggregates at the current time is that the application does not enforce standard interval blocks of time when soliciting user queries.

It is unlikely that materialized views can significantly improve the performance of the LDC since all results in the LDC are stored in text fields and unit standards are not enforced. This makes it almost impossible to deliver aggregate results in a quick and meaningful way. The one situation where materialized views may be leveraged is on the LDC's count pages that are used to inform a user of the relative size of the report that will be produced. These count pages are also important for determining whether a report will run in real-time or in batch. While performance improvements are possible, it is doubtful that the average user will see a substantial performance

increase for the majority of counts. For this reason it is recommended that materialized views not be utilized in the initial version of the LDC data warehouse. They may be explored in the future in response to query types that produce special performance complications.

## 4.2 Bitmap Indexes and Performance Tuning

---

Bitmap indexes were included in Oracle for the primary purpose of data warehousing. They are specifically utilized in one-to-many relationships where one of the tables is quite small in relation to the other. In data warehousing, a bitmap index is often used to connect each dimension table to the fact table when using a star-schema structure.

One of the advantages of bitmap indexes is the ability to perform rapid binary searches on a fact table based on values selected from multiple dimension tables. The additive nature of the query logic allows for rapid performance and is leveraged by the Oracle query optimizer which is designed to recognize a data warehouse based on the distribution of table size (one extremely large table [the fact table] joined so several much smaller tables [dimension tables]).

It is recommended that a bitmap index be created for each of the foreign keys of the LDC's fact table (RESULT). It is also recommended that all location information be indexed due to the frequency of user queries that are driven by the geographic location of stations. The majority of these fields were previously housed in the STATION\_LOCATION\_POINT table. However, these elements have been consolidated and migrated to the STATION table as part of the new LDC data warehouse design (see Chapter 3).

It is also recommended that the Oracle initialization parameter `STAR_TRANSFORMATION_ENABLED` be set to 'true' to optimize the speed of star-queries. In addition, the cost-based optimizer should be used for all data warehouse systems. These changes to the Oracle environment will maximize the performance of the star-schema and bitmap indexes.

Note: In Oracle 8i, bitmap indexes are only available with the Enterprise Edition of the DBMS. This is true of many of the data warehousing features of Oracle 8i.

## 4.3 Constraints

---

Many Oracle 7 and Oracle 8 based data warehouses lacked constraints due to concerns over how constraint enforcement would affect performance. However, Oracle 8i provides constraint functionality designed to address these issues specifically for data warehouse systems. It is now possible to manage when and how a constraint is enforced. This is useful in many situations. For instance, a constraint can be turned on during data import to ensure that referential integrity is enforced but later disabled to save DBMS overhead during periods when the database is not undergoing updates. This is especially useful for the LDC since it houses legacy data and will seldom (if ever) be updated.

## 4.4 Oracle Extraction, Transformation, and Loading Tools

---

Oracle provides several products for the populating and maintenance of data warehouses. These tools are largely designed for complex data warehouses that have a number of 'living' source systems that provide data to the data warehouse on a fixed batch update cycle. Extraction, Transformation, and Loading (ELT) tools are used to migrate data from source systems to the data warehouse.

Since the LDC data warehouse has only one source system (the existing LDC database) and only one data migration is planned (the initial population of the LDC data warehouse), ETL tools provide little benefit to the current project. However, these tools should be considered if a decision is made to incorporate additional data from new source systems to the LDC data warehouse in the future.

## 4.5 Oracle Warehouse Builder

---

Oracle also provides a special case tool for modeling and managing data warehouses. The primary benefits of the tool are the ability to track the data model changes made to source systems and incorporate those changes to the ETL cycle for data warehouse updates. This maintains the data mappings between the source systems and data warehouse as the individual systems evolve over time. Since the LDC data warehouse has only one source database

and only one data migration is planned, the Oracle Warehouse Builder product is not currently essential for warehouse development or maintenance.

ERwin is being recommended as the CASE tool for creating and maintaining the LDC data warehouse design. Although ERwin is not capable of producing all of scripts required to build the LDC data warehouse (such as the creation of bitmap indexes), it is a useful tool for generating the CREATE TABLE scripts, managing data element types and sizes, generating dimensional model diagrams, and documenting the LDC data warehouse design. ERwin was selected because the existing LDC data model is maintained in ERwin and it is possible to port the legacy system data dictionary information to the new data warehouse design.

In the future, EPA may decide to switch from ERwin to an Oracle case tool (such as Oracle Designer or Oracle Data Warehouse Builder). This would bring consistency to the CASE tools used by the STORET team. Migration to a new CASE tool would be relatively straightforward due to the reverse-engineering capabilities offered by these products.

# Data Model Recommendation

---

*This section summarizes the findings of the previous chapters and provides an initial dimensional model based on the recommendations made in this document*

## 5.1 Logical Data Model

---

The following data model incorporates the recommendations made throughout this document. It follows the design of a classic star-schema with a few exceptions:

**CONTACT\_INFORMATION**—this table is used to house the contact information that is available independently of the rest of the LDC’s data. Required information from the ORGANIZATION and AGENCY tables has been consolidated in the CONTACT\_INFORMATION table

**STATE**—this reference table is used to allow for the quick look-up of county lists for a selected state. This eliminates the need to perform a costly ‘select unique’ statement on the STATION table in order to produce a specific county list

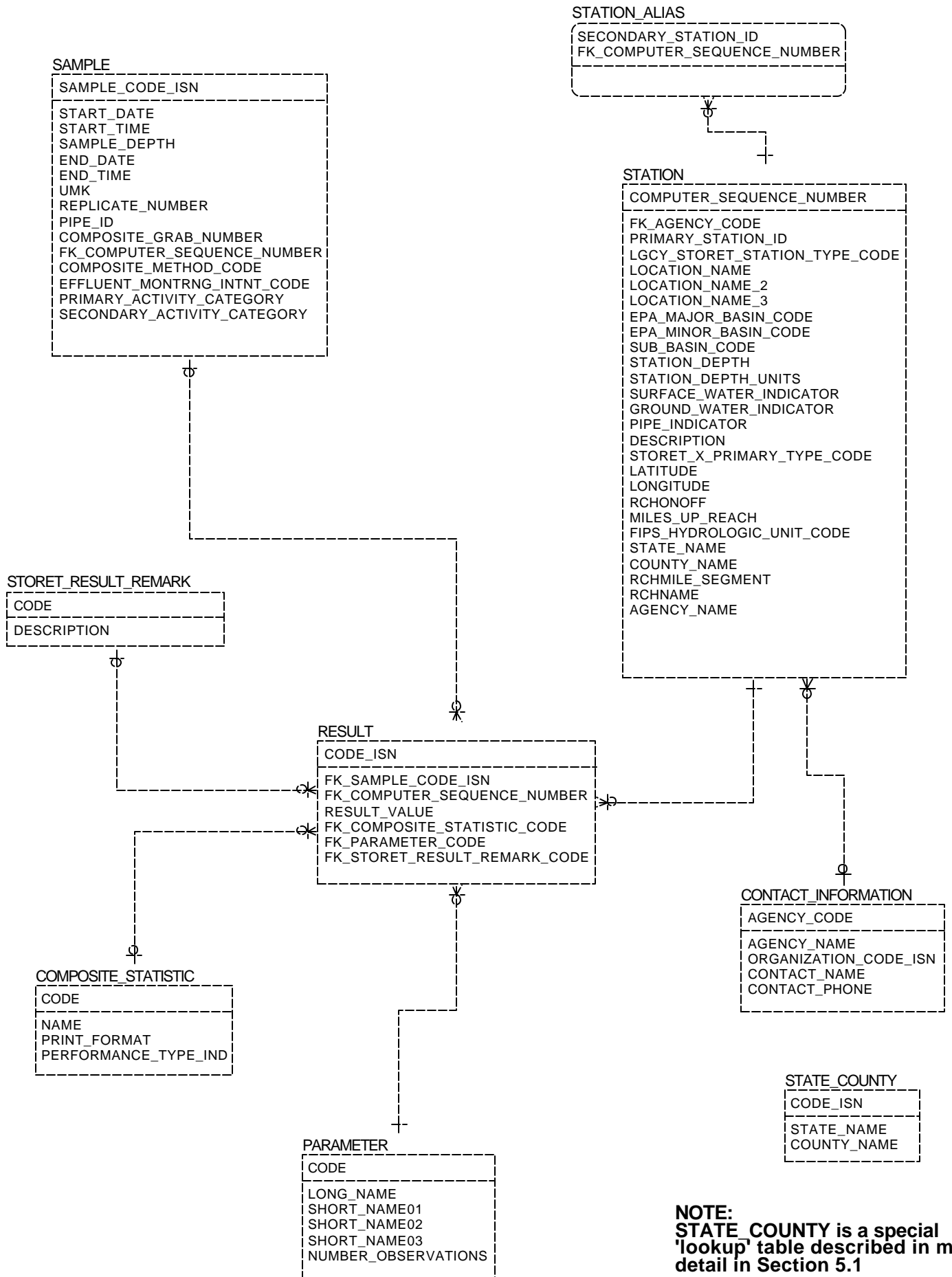
**STATION\_ALIAS**—this table ‘snowflakes’ off the STATION table and is used to house alternate names for a STATION. It was determined that the de-normalization of the STATION\_ALIAS table would provide little performance enhancement since the table is never queried against and is only used by one of the report templates (the HTML/PDF version of the Station Description Report). The elimination of the STATION\_ALIAS table would have required either concatenating all aliases for a given station into one string or creating three additional fields in the STATION table to store potential aliases (the maximum number of station aliases in the LDC is three)

SAMPLE—this table still contains a reference to the STATION table (through `fk_computer_sequence_number`). This is used to speed SAMPLE counts when querying by STATION

Figure 5-1 shows the currently proposed LDC Warehouse Data Model.



Figure 5-1: LDC Warehouse Data Model



## 5.2 Data Elements

---

The required data elements identified in Chapter 2 have all been incorporated into the new data warehouse structure. However, many of these elements have been relocated, consolidated, or transformed. Figure 5-2 lists the data elements that comprise the LDC data warehouse and maps these elements to the LDC source system. This chart will serve as a roadmap for constructing the tables of the LDC data warehouse and developing the data import scripts that will populate the warehouse structure.

Figure 5-2 LDC Warehouse Data Elements and Mappings

Table/Column Name	Data Type	Optionality	Key	Source Table Name	Source Column Name
PARAMETER					
code	CHAR (5)	NOT NULL	PRIMARY	PARAMETER	code
long_name	VARCHAR (50)	NULL		PARAMETER	long_name
short_name01	CHAR (8)	NULL		PARAMETER	short_name01
short_name02	CHAR (8)	NULL		PARAMETER	short_name02
short_name03	CHAR (8)	NULL		PARAMETER	short_name03
number_observations	NUMBER (12)	NULL		PARAMETER	number_observations
RESULT					
code_isn	NUMBER (12)	NOT NULL	PRIMARY	RESULT	code_isn
fk_sample_code_isn	NUMBER (12)	NOT NULL	FOREIGN	RESULT	fk_sample_code_isn
fk_computer_sequence_number	NUMBER (12)	NOT NULL	FOREIGN	SAMPLE	fk_computer_sequence_number
result_value	VARCHAR2(12)	NULL		RESULT	result_value
fk_composite_statistic_code	CHAR(1)	NULL	FOREIGN	RESULT	fk_composite_statistic_code
fk_parameter_code	CHAR(5)	NULL	FOREIGN	RESULT	fk_parameter_code
fk_storet_result_remark_code	CHAR(1)	NULL	FOREIGN	RESULT	fk_storet_result_remark_code
SAMPLE					
sample_code_isn	NUMBER (12)	NOT NULL	PRIMARY	SAMPLE	sample_code_isn
start_date	DATE	NULL		SAMPLE	start_date

Table/Column Name	Data Type	Optionality	Key	Source Table Name	Source Column Name
start_time	NUMBER (4)	NULL		SAMPLE	start_time
sample_depth	NUMBER (8,2)	NULL		SAMPLE	sample_depth
end_date	DATE	NULL		SAMPLE	end_date
end_time	NUMBER (4)	NULL		SAMPLE	end_time
umk	CHAR (8)	NULL		SAMPLE	umk
replicate_number	NUMBER (12)	NULL		SAMPLE	replicate_number
pipe_id	NUMBER (12)	NULL		SAMPLE	pipe_id
composite_grab_number	NUMBER (2)	NULL		SAMPLE	composite_grab_number
fk_computer_sequence_number	NUMBER (12)	NOT NULL	FOREIGN	SAMPLE	fk_computer_sequence_number
composite_method_code	CHAR (1)	NULL		SAMPLE	fk_composite_method_code
primary_activity_category	VARCHAR2 (60)	NULL		PRIMARY_ACTIVITY_CATEGORY	name
secondary_activity_category	VARCHAR2 (60)	NULL		SECONDARY_ACTIVITY_CATEGORY	name
effluent_montrng_intnt_code	CHAR (2)	NULL		SAMPLE	fk_effluent_montrng_intnt_code
STATION					
computer_sequence_number	NUMBER (12)	NOT NULL	PRIMARY	STATION	computer_sequence_number
primary_station_id	CHAR (15)	NOT NULL		STATION	primary_station_id
lgcy_storet_station_type_code	VARCHAR2 (65)	NULL		STATION	lgcy_storet_station_type_code
location_name	VARCHAR2 (48)	NULL		STATION	location_name
location_name_2	CHAR (60)	NULL		STATION	location_name_2
location_name_3	CHAR (60)	NULL		STATION	location_name_3
epa_major_basin_code	CHAR (2)	NULL		STATION	epa_major_basin_code
epa_minor_basin_code	CHAR (2)	NULL		STATION	epa_minor_basin_code

Table/Column Name	Data Type	Optionality	Key	Source Table Name	Source Column Name
sub_basin_code	CHAR (2)	NULL		STATION	sub_basin_code
station_depth	NUMBER (3)	NULL		STATION	station_depth
station_depth_units	VARCHAR2 (20)	NULL		STATION	station_depth_units
surface_water_indicator	CHAR (1)	NULL		STATION	surface_water_indicator
ground_water_indicator	CHAR (1)	NULL		STATION	ground_water_indicator
pipe_indicator	CHAR (1)	NULL		STATION	pipe_indicator
description	VARCHAR2 (100)	NULL		DESCRIPTIVE_PARAGRAPH_LINE	text
storet_x_primary_type_code	VARCHAR2 (30)	NULL		STORET_X_STATION_TYPE	primary_type_code
latitude	NUMBER (8,6)	NULL		STATION_LOCATION_POINT	latitude
longitude	NUMBER (9,6)	NULL		STATION_LOCATION_POINT	longitude
rchonoff	CHAR (3)	NULL		STATION_LOCATION_POINT	rchonoff
miles_up_reach	FLOAT	NULL		STATION_LOCATION_POINT	miles_up_reach
fips_hydrologic_unit_code	CHAR (8)	NULL		FIPS_HYDROLOGIC_UNIT	fips_hydrologic_unit_code
state_name	CHAR (30)	NOT NULL		STATE_COUNTY	state_name
county_name	CHAR (30)	NOT NULL		STATE_COUNTY	county_name
rchmile_segment	NUMBER (3)	NULL		RF1_REACH	rchmile_segment
rchname	VARCHAR2 (60)	NULL		RF1_REACH	rchname
agency_name	VARCHAR2 (25)	NULL		AGENCY	agency_name
fk_agency_code	NUMBER (12)	NOT NULL	FOREIGN	STATION	fk_agency_code
STATION_ALIAS					
fk_computer_sequence_number	NUMBER (12)	NOT NULL	PRIMARY/ FOREIGN	STATION_ALIAS	fk_computer_sequence_number

Table/Column Name	Data Type	Optionality	Key	Source Table Name	Source Column Name
secondary_station_id	CHAR (12)	NOT NULL	PRIMARY	STATION_ALIAS	secondary_station_id
COMPOSITE_STATISTIC					
code	CHAR (1)	NOT NULL	PRIMARY	COMPOSITE_STATISTIC	code
name	VARCHAR2 (60)	NULL		COMPOSITE_STATISTIC	name
print_format	CHAR (3)	NULL		COMPOSITE_STATISTIC	print_format
performance_type_ind	CHAR (1)	NULL		COMPOSITE_STATISTIC	performance_type_ind
STORET_RESULT_REMARK					
code	CHAR (1)	NOT NULL	PRIMARY	STORET_RESULT_REMARK	code
description	VARCHAR2 (254)	NULL		STORET_RESULT_REMARK	description
STATE_COUNTY					
code_isn	NUMBER (12)	NOT NULL	PRIMARY	STATE_COUNTY	code_isn
state_name	CHAR (30)	NULL		STATE_COUNTY	state_name
county_name	CHAR (30)	NULL		STATE_COUNTY	county_name
CONTACT_INFORMATION					
agency_code	CHAR (8)	NOT NULL	PRIMARY	AGENCY	code
agency_name	VARCHAR2 (25)	NULL		AGENCY	agency_name
organization_code_isn	NUMBER (12)	NULL		AGENCY	fk_organization_code_isn
contact_name	VARCHAR2 (25)	NULL		ORGANIZATION	contact_name
contact_phone	CHAR (13)	NULL		ORGANIZATION	contact_phone

## 5.3 Data Element Transformation

---

All data elements can be populated from the existing LDC database using simple SQL joins and INSERT statements. The one exception is the 'description' column in STATION. This column will be populated by a PL/SQL procedure that concatenates the values of each 'text' element in the table DESCRIPTIVE\_PARAGRAPH\_LINE into one text string for a given STATION. All other data elements will be imported to the warehouse in the same form they exist in the current LDC data model.





# 6

## Transitional Architecture and Data Archiving Scheme

---

*This section provides a guide for transitioning between the existing LDC database and the planned data warehouse. Special attention is given to the archiving of data that may be excluded from the data warehouse*

### 6.1 Data Transition

---

The current LDC data model contains many elements that are not used by the LDC web application. The decision has been made to exclude these elements from the data warehouse for the time being for the following reasons:

The de-normalized structure of the data warehouse requires that database size be held in check

The complexity of the data warehouse is reduced by narrowing the scope to include only essential data elements

The current LDC database tables will not be eliminated in the short term (allowing for a phased approach to data migration)

The initial version of the LDC data warehouse will focus solely on supplying the data required by the current LDC application. During the development, deployment, and testing of the LDC data warehouse, it may be necessary to maintain copies of both the data warehouse and current LDC data tables. While these tables can co-exist on the same database instance, it is recommended that separate instance be maintained for performance testing. This allows for appropriate environmental variables to be set for each type of environment enabling more accurate performance comparisons. It will also eliminate potential confusion over naming conflicts between the two data models.

## 6.2 Long-Term Data Storage

---

It is recommended that the existing LDC tables be deleted from the public (Mountain) version of the LDC once the data warehouse has been successfully tested and deployed in that environment. This will free server space, simplify maintenance, and make appropriate performance tuning as straightforward as possible.

It is recommended that the intranet version of the LDC (canyon) be used to archive the existing LDC tables. This will simplify regression testing and protect the complete data content of the LDC. This information may be required by individual studies or incorporated into future LDC report templates (this would require expanding the LDC data warehouse). Once the LDC data warehouse has been successfully implemented and testing completed, existing LDC data tables can be moved to a more cost effective server or stored on back-up medium depending on perceived future use.

Eventually, it may be determined that archival tables should be incorporated into the data warehouse structure. These tables would not be de-normalized or optimized for maximum performance since they would not be directly used by the core application. These tables could relate to the data warehouse by assigning consistent key values. This would allow for the deletion of the current LDC database thereby reducing storage and maintenance overhead.

## Next Steps

---

*This section discusses the deliverables that will be produced by AMS subsequent to this report and feedback received from EPA*

### 7.1 Final Document Production

---

Following EPA feedback, a final version of this document will be produced. A meeting is tentatively planned for the week following the delivery of this report to review the recommendations made and resolve any issues that may surface. A one-week revision period is planned to incorporate any necessary changes to this paper.

### 7.2 Implementation Scripts

---

Once the final version of the LDC Data Warehouse Options Paper has been approved, development will begin on the Oracle scripts to establish the database environment and instantiate the new warehouse design. These scripts will initially be developed at AMS and then ported to an NT server at STORET headquarters. Instructions will be supplied with the scripts that detail the appropriate installation, maintenance and use of the code.

Once the database environment has been established, data will be transformed and loaded from EPA's servers at Research Triangle Park (RTP). This will result in a test database that will house the necessary data content of the LDC in the new warehouse structure. Performance measures will be conducted and appropriate tuning refinements made.

## 7.3 Application Design Options Paper

---

Once the data warehouse design has been completed and an initial testing environment established, work will begin on identifying the application modifications necessary to make the current LDC web application compatible with the new warehouse structure. The first step in this process will be the development of the Application Design Options document that identifies dependencies between data elements and the functionality of the LDC.

This paper will serve as a roadmap for the application modifications that will follow the delivery of this paper. Recommendations will be made concerning development options for updating the LDC web application. Once feedback has been received concerning these recommendations, the re-engineering of the LDC application will begin (see Section 7.4).

## 7.4 Re-engineering of LDC Web Retrievals

---

All PL/SQL dynamic web pages and report templates of the LDC will be modified to work with the data warehouse (these are listed in Section 2.3). These modifications will be made in a manner consistent with the recommendations made in the Application Design Options paper and feedback received from EPA. The application modifications will be tested on internal EPA servers and then migrated to the public production environment.

The new application components will be delivered as part of an implementation package that includes instructions for installation, maintenance and use.

# Appendix A: LDC Warehouse Data Dictionary

---

*This section lists the Attribute Name, Attribute Definition, Column Name, and Table Usage for each data element in the LDC data warehouse*

## Data Elements and Definitions

---

The following data dictionary information was imported from the current STORET LDC ERwin data model. This list was formatted to include only data elements that are included in the new LDC data warehouse design. In cases where column names or table usage has changed, the dictionary reflects the values as they exist in the data warehouse design.

**Column Name:** FK\_AGENCY\_CODE

**Attribute Name:** Agency Code

**Definition:** Code identifying the agency responsible for the data. It identifies the owner of the data and is used to provide authorization for updating or deleting data for that specific agency. Agency codes were user defined prior to 1977. Since 1977, agency codes have been assigned by the STORET system manager. The combination of an agency code and a station identifier uniquely identifies a site. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** AGENCY\_NAME

**Attribute Name:** Agency Name

**Definition:** Name identifying the agency responsible for the data. It identifies the owner of the data and is used to provide authorization for updating or deleting data for specific agencies. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** CONTACT\_INFORMATION

**Column Name:** AGENCY\_NAME

**Attribute Name:** Agency Name

**Definition:** Name identifying the agency responsible for the data. It identifies the owner of the data and is used to provide authorization for updating or deleting data for specific agencies. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** AGENCY\_CODE

**Attribute Name:** Code

**Definition:** Code identifying the agency responsible for the data. It identifies the owner of the data and is used to provide authorization for updating or deleting data for that specific agency. Agency codes were user defined prior to 1977. Since 1977, agency codes have been assigned by the STORET system manager. The combination of an agency code and a station identifier uniquely identifies a site. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** CONTACT\_INFORMATION

**Column Name:** CODE

**Attribute Name:** Code

**Definition:** Code identifying the composite statistic types and composite performance types for non-effluent samples. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** COMPOSITE\_STATISTIC

**Column Name:** FK\_COMPOSITE\_STATISTIC\_CODE

**Attribute Name:** FK\_Composite\_Statistic\_Code

**Definition:** Code identifying the composite statistic types and composite performance types for non-effluent samples. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** RESULT

**Column Name:** FK\_STORET\_RESULT\_REMARK\_CODE

**Attribute Name:** FK\_STORET\_Result\_Remark\_Code

**Definition:** Code used to identify a result remark. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** RESULT

**Column Name:** CODE

**Attribute Name:** Code

**Definition:** Code used to identify a result remark. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STORET\_RESULT\_REMARK

**Column Name:** FK\_ORGANIZATION\_CODE\_ISN

**Attribute Name:** FK Organization Code ISN

**Definition:** System generated unique identifier for an organization.  
REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** CONTACT\_INFORMATION

**Column Name:** CODE\_ISN

**Attribute Name:** Code ISN

**Definition:** System generated unique identifier for a state and county combination. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATE\_COUNTY

**Column Name:** COMPOSITE\_GRAB\_NUMBER

**Attribute Name:** Composite Grab Number

**Definition:** Numeric value representing the number of grabs in the composite sample. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** SAMPLE

**Column Name:** COMPOSITE\_METHOD\_CODE

**Attribute Name:** Composite\_Method\_Code

**Definition:** Code indicating a composite method or sampling method for a qualified composite (e.g., C, G, B). REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** SAMPLE

**Column Name:** FK\_COMPUTER\_SEQUENCE\_NUMBER

**Attribute Name:** Computer Sequence Number

**Definition:** A non-intelligent unique key used to identify stations in legacy STORET. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** RESULT

**Column Name:** FK\_COMPUTER\_SEQUENCE\_NUMBER

**Attribute Name:** FK\_Station\_Computer\_Sequence\_Number

**Definition:** A non-intelligent unique key used to identify stations in legacy STORET. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** SAMPLE

**Column Name:** COMPUTER\_SEQUENCE\_NUMBER

**Attribute Name:** Computer Sequence Number

**Definition:** A non-intelligent unique key used to identify stations in legacy STORET. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** FK\_COMPUTER\_SEQUENCE\_NUMBER

**Attribute Name:** FK\_Station\_Computer\_Sequence\_Number

**Definition:** A non-intelligent unique key used to identify stations in legacy STORET. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION\_ALIAS

**Column Name:** CONTACT\_NAME

**Attribute Name:** Contact Name

**Definition:** Name (last/first) of contact person at agency. REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** CONTACT\_INFORMATION

**Column Name:** CONTACT\_PHONE

**Attribute Name:** Contact Phone

**Definition:** Commercial telephone number of contact person. REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** CONTACT\_INFORMATION

**Column Name:** COUNTY\_NAME

**Attribute Name:** County Name

**Definition:** Name used to identify the county that the station is located in.  
REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATE\_COUNTY

**Column Name:** COUNTY\_NAME

**Attribute Name:** County Name

**Definition:** Name used to identify the county that the station is located in.  
REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** DESCRIPTION

**Attribute Name:** Description

**Definition:** Result remark text. REFERENCE: STORET Y Data Model  
Validation (1997)

**Table Usage:** STORET\_RESULT\_REMARK

**Column Name:** DESCRIPTION

**Attribute Name:** Description

**Definition:** Station remark text. REFERENCE: STORET Y Data Model  
Validation (1997)

**Table Usage:** STATION

**Column Name:** EPA\_MAJOR\_BASIN\_CODE

**Attribute Name:** EPA Major Basin Code

**Definition:** Code representing archaic EPA major basin developed as part of  
a drainage area delineation scheme. REFERENCE:  
STORET/ODES/BIOS Modernization (09/94)

**Table Usage:** STATION

**Column Name:** EPA\_MINOR\_BASIN\_CODE

**Attribute Name:** EPA Minor Basin Code



**Definition:** Code representing archaic EPA minor basin developed as part of a drainage area delineation scheme. REFERENCE: STORET/ODES/BIOS Modernization (09/94)

**Table Usage:** STATION

**Column Name:** END\_DATE

**Attribute Name:** End Date

**Definition:** Composite sample end date. Greater of two dates when two dates are present. Not present if UMK present. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** SAMPLE

**Column Name:** END\_TIME

**Attribute Name:** End Time

**Definition:** Composite sample end time (24-hour clock). This is an optional attribute. For effluent samples in legacy STORET, the end date field contains the pipe id. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** SAMPLE

**Column Name:** FIPS\_HYDROLOGIC\_UNIT\_CODE

**Attribute Name:** Fips HUC

**Description:** Code used to identify the hydrologic unit of the station location based on Federal Information Processing Standards (FIPS). REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** GROUND\_WATER\_INDICATOR

**Attribute Name:** Ground Water Indicator

**Table Usage:** STATION

**Column Name:** LATITUDE

**Attribute Name:** Latitude

**Definition:** North Latitude expressed in degrees, minutes, seconds and tenth of a second (e.g., DD MM SS.S) REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** LGCY\_STORET\_STATION\_TYPE\_CODE

**Attribute Name:** Legacy STORET Station Type Code

**Definition:** Text field that preserves the encoded legacy station type field that can be decoded to determine the legacy station type codes. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** LOCATION\_NAME

**Attribute Name:** Location Name

**Definition:** Text originally used to describe location. REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** LOCATION\_NAME\_2

**Attribute Name:** Location Name 2

**Definition:** Text which often contains the Major Basin name but can also contain other information such as addresses. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** LOCATION\_NAME\_3

**Attribute Name:** Location Name 3

**Definition:** Text which often contains the Minor Basin name but can also contain other information such as addresses. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** LONG\_NAME

**Attribute Name:** Long Name

**Definition:** Long name of parameter, units, medium, etc. REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** PARAMETER

**Column Name:** LONGITUDE

**Attribute Name:** Longitude

**Definition:** West Longitude expressed in degrees, minutes, seconds and tenth of a second (e.g., DDD MM SS.S). Only positive values are stored. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** MILES\_UP\_REACH

**Attribute Name:** Miles Up Reach

**Description:** This value is used to identify a station location by indicating the number of miles up the reach. In legacy STORET, the Miles Up Reach is stored as characters 4-7 and character 9 of the STORET ReachMile field that has the following format: SSSMMMM.MF REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** NAME

**Attribute Name:** Name

**Definition:** Name identifying the composite statistic types and composite performance types for non-effluent samples. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** COMPOSITE\_STATISTIC

**Column Name:** NUMBER\_OBSERVATIONS

**Attribute Name:** Number Observations

**Table Usage:** PARAMETER

**Column Name:** CODE

**Attribute Name:** Parameter\_Code

**Definition:** Numeric code identifying a STORET parameter. REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** PARAMETER

**Column Name:** FK\_PARAMETER\_CODE

**Attribute Name:** FK\_Parameter\_Code

**Definition:** Numeric code identifying a STORET parameter. REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** RESULT

**Column Name:** PERFORMANCE\_TYPE\_IND

**Attribute Name:** Performance\_Type\_Ind

**Definition:** Indicator of whether the code and name represents a composite  
statistic type or a composite performance type for non-effluent samples.  
REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** COMPOSITE\_STATISTIC

**Column Name:** PIPE\_ID

**Attribute Name:** Pipe ID

**Definition:** Pipe identifier for an effluent sample. REFERENCE:  
STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** SAMPLE

**Column Name:** PIPE\_INDICATOR

**Attribute Name:** Pipe Indicator

**Table Usage:** STATION

**Column Name:** PRIMARY\_ACTIVITY\_CATEGORY

**Attribute Name:** Primary Activity Category

**Definition:** Code used to describe a secondary activity category.  
REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** SAMPLE

**Column Name:** PRIMARY\_STATION\_ID

**Attribute Name:** Primary Station ID

**Definition:** Identifier used by the legacy system to denote the primary station  
identifier. In addition to the primary station identifier, legacy STORET  
allows users to define up to three secondary station identifiers or station  
aliases. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** PRINT\_FORMAT

**Attribute Name:** Print Format

**Definition:** Code identifying the print format for composite statistic types and composite performance types for non-effluent samples.

REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** COMPOSITE\_STATISTIC

**Column Name:** RCHMILE\_SEGMENT

**Attribute Name:** Rchmile segment

**Definition:** Reach segment portion of reach index system. In legacy STORET, the ReachMile segment is the first three characters of the STORET

ReachMile field that has the following format: SSSMMMM.MF

REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** RCHNAME

**Attribute Name:** Rchname

**Definition:** Name used to identify the reach unit of the station location.

REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** RCHONOFF

**Attribute Name:** Rchonoff

**Definition:** Flag indicating if station is on an indexed search. In legacy STORET, the ReachOnOff indicator is the last three character of the

STORET ReachMile field that has the following format: SSSMMMM.MF

REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION

**Column Name:** REPLICATE\_NUMBER

**Attribute Name:** Replicate Number

**Definition:** BIOS sample replicate number REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** SAMPLE

**Column Name:** RESULT\_VALUE

**Attribute Name:** Result Value

**Definition:** Data value for sample result or a code representing an observation (e.g., WMO codes). Result values can be numeric or

alphanumeric values. Non-detects are remarked values in the legacy system. In STORET X the following text values can be used as a result

value: "Not Detected" "Detected But Not Quantified" "Detected Above Quantification Limit" "Detected Below Quantification Limit" "Detected

And Quantified" REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** RESULT

**Column Name:** CODE\_ISN

**Attribute Name:** Result\_Code\_ISN

**Definition:** System generated unique identifier for a result. REFERENCE:  
STORET Y Data Model Validation (1997)

**Table Usage:** RESULT

**Column Name:** SAMPLE\_DEPTH

**Attribute Name:** Sample Depth

**Definition:** Sample depth in feet ( not station depth). This is an optional field. If a depth is being stored for a water or water related sample, specific codes are used to store the depth (length) and the depth qualifier followed by a comma (e.g., D999=Sample depth where 999 is the depth in feet, DM999=Sample depth where the 999 is the depth in meters). In legacy STORET if a depth is not present SMK free text is stored for the sample. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** SAMPLE

**Column Name:** FK\_SAMPLE\_CODE\_ISN

**Attribute Name:** Sample\_Code\_ISN

**Definition:** System generated unique identifier for a sample. REFERENCE:  
STORET Y Data Model Validation (1997)

**Table Usage:** RESULT

**Column Name:** SAMPLE\_CODE\_ISN

**Attribute Name:** Sample\_Code\_ISN

**Definition:** System generated unique identifier for a sample. REFERENCE:  
STORET Y Data Model Validation (1997)

**Table Usage:** SAMPLE

**Column Name:** SECONDARY\_ACTIVITY\_CATEGORY

**Attribute Name:** Secondary Activity Category

**Definition:** Code used to describe a secondary activity category.  
REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** SAMPLE

**Column Name:** SECONDARY\_STATION\_ID

**Attribute Name:** Secondary Station ID

**Definition:** Alias station identifier. Each station may have up to three secondary station identifiers. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATION\_ALIAS

**Column Name:** SHORT\_NAME01

**Attribute Name:** Short Name01

**Definition:** First shortened name of parameter, units, medium, etc. Each parameter can have up to three short names to be used as column headers

in legacy STORET. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** PARAMETER

**Column Name:** SHORT\_NAME02

**Attribute Name:** Short Name02

**Definition:** Second shortened name of parameter, units, medium, etc. Each parameter can have up to three short names to be used as column headers in legacy STORET. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** PARAMETER

**Column Name:** SHORT\_NAME03

**Attribute Name:** Short Name03

**Definition:** Third shortened name of parameter, units, medium, etc. Each parameter can have up to three short names to be used as column headers in legacy STORET. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** PARAMETER

**Column Name:** START\_DATE

**Attribute Name:** Start Date

**Definition:** Date which indicates the date on which a grab sample was taken or the start date of a composite. Lesser of two dates when two dates are present. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** SAMPLE

**Column Name:** START\_TIME

**Attribute Name:** Start Time

**Definition:** Time which indicates the time at which a grab sample was taken or the start time of a composite. This is an optional attribute based on a 24-hour clock. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** SAMPLE

**Column Name:** STATE\_NAME

**Attribute Name:** State Name

**Definition:** Name used to identify the state that the station is located in, or in the case of an ocean station, is close to. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** STATE\_COUNTY

**Column Name:** STATE\_NAME

**Attribute Name:** State Name

**Table Usage:** STATION

**Column Name:** STATION\_DEPTH

**Attribute Name:** Station Depth

**Definition:** The usual or typical depth of water at the station measured in feet. REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** STATION\_DEPTH\_UNITS

**Attribute Name:** Station Depth Units

**Definition:** Units used to describe the station depth. Station depth is always stored as "feet". REFERENCE: STORET Y Data Model Validation (1997)

**Table Usage:** STATION

**Column Name:** STORET\_X\_PRIMARY\_TYPE\_CODE

**Attribute Name:** Storet X Type Code

**Table Usage:** STATION

**Column Name:** SUB\_BASIN\_CODE

**Attribute Name:** Sub Basin Code

**Definition:** Code representing user defined sub basin developed as part of a drainage area delineation scheme. REFERENCE: STORET/ODES/BIOS Modernization (09/94)

**Table Usage:** STATION

**Column Name:** SURFACE\_WATER\_INDICATOR

**Attribute Name:** Surface Water Indicator

**Table Usage:** STATION

**Column Name:** UMK

**Attribute Name:** UMK

**Definition:** User multi-purpose key (free text). Not present if End Data present. REFERENCE: STORET/BIOS/ODES Modernization (09/94)

**Table Usage:** SAMPLE