PB99-113011

TransLink® Information Database Requirements

TX-98/2988-2F

TransLink®

Innovative Surface Transportation Management

Texas Transportation Institute
The Texas A&M University System
College Station, Texas

REPRODUCED BY: NTIS
U.S. Department of Commerce
National Technical Information Service
Springfield, Virginia 22161

Texas Department of Transportation

Technical Report Documentation Page

1. Report No. TX-98/2988-2F	PB99-113011
3. Recipient's Catalog No.	
4. Title and Subtitle TRANSLINK™ INFORMATION DATABASE REQUIREMENTS	5. Report Date October 1996
	6. Performing Organization Code
7. Author(s) Tom L. Stout, Jyh-Charn S. Liu, Robert E. Brydia, and Christopher M. Poe	8. Performing Organization Report No. Research Report 2988-2F
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, TX 77843-3135	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. Study No. 7-2988
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Transfer Office P. O. Box 5080 Austin, Texas 78763-5080	13. Type of Report and Period Covered Final: October 1995 -August 1996
	14. Sponsoring Agency Code

15. Supplementary Notes

Research performed in cooperation with the Texas Department of Transportation. Research Study Title: TransLink™ Research Program

16. Abstract

One aspect of the TransLink™ research program is to have a laboratory where the ITS architecture and technologies can be explored, integrated, demonstrated, and showcased. An integral part of this laboratory is a data warehouse where historical and real-time data can reside. This report examines the best method for managing this data. After a brief introduction, a listing of some of the various types of data that could be seen as a result of the Intelligent Transportation Infrastructure (ITI) is presented. The report then proceeds to examine the best method to handle those data needs. Recommendations to accomplish the data management needs consist of the implementation of an Oracle database. The report concludes with a chapter on implementation, which describes the efforts under way to accomplish the goal of this task of the research program.

17. Key Words ITI, ITS Architecture, TransLink™, Database, Client Server	18. Distribution Statement No restrictions. This document is available to the public through NTIS: National Technical Information Service 5285 Port Royal Road Springfield, VA 22161

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	52	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

TRANSLINK™ INFORMATION DATABASE REQUIREMENTS

by

Tom L. Stout
Research Specialist

Jyh-Charn S. Liu. Ph.D.
Associate Research Scientist

Robert E. Brydia
Assistant Research Scientist

and

Christopher M. Poe. P. E.
Associate Research Engineer

Research Report 2988-2F
Research Study Number 7-2988
Research Study Title: TransLink™ Research Program

Sponsored by the
Texas Department of Transportation

October 1996

TEXAS TRANSPORTATION INSTITUTE
The Texas A&M University System
College Station, Texas 77843-3135

PROTECTED UNDER INTERNATIONAL COPYRIGHT
ALL RIGHTS RESERVED.
NATIONAL TECHNICAL INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE

IMPLEMENTATION STATEMENT

This report recommends the use of Oracle software as a means to establish a relational client-server database model for replicating the data flows that are present in a Transportation Management Center (TMC). The physical database will be housed in the Computer Science Department of The Texas A&M University with the applications residing in the TransLink™ research laboratory housed at the Texas Transportation Institute.

The results of this report will be important in establishing connections to the existing TMC's for the purpose of obtaining real-world data for research efforts in Intelligent Transportation Systems (ITS) and in the historical warehousing of such data, again for the purpose of research. Specific implementation recommendations are listed in section 6.4 of this report.

DISCLAIMER

The contents of the report reflect the views of the authors who are responsible for the opinions, findings, and conclusions presented herein. The contents do not necessarily reflect the official views or policies of the Texas Department of Transportation. This report does not constitute a standard, specification, or regulation, nor is it meant for construction, bidding, or permit purposes. The engineer in charge of the project was Mr. Christopher M. Poe, P.E. # 70345

ACKNOWLEDGMENT

This research was sponsored by the Texas Department of Transportation. The research effort is conducted under the TransLink™ Partnership between the Metropolitan Transit Authority of Harris County, Rockwell International, the Texas Department of Transportation, the Texas Transportation Institute, and the U.S. Department of Transportation, Federal Highway Administration

TABLE OF CONTENTS

Page

LIST OF TABLES xi
SUMMARY xiii

1.0 INTRODUCTION AND SCOPE OF REPORT 1
1.1 INTRODUCTION 1
1.2 SCOPE OF REPORT 2

2.0 DATA REQUIREMENTS OF THE INTELLIGENT TRANSPORTATION INFRASTRUCTURE (ITI) 3
2.1 THE INTELLIGENT TRANSPORTATION INFRASTRUCTURE 3
2.2 DATA REQUIREMENTS OF ITI COMPONENTS 3
2.2.1 Freeway Management 4
2.2.2 Traffic Signal Control 5
2.2.3 Transit Management 6
2.2.4 Incident Management 6
2.2.5 Emergency Management Services 7
2.2.6 Railroad Grade Crossings 8
2.2.7 Regional Multi modal Travel Information 8
2.2.8 Electronic Fare Payment 9
2.2.9 Electronic Toll Collection 10

3.0 DATA STORAGE REQUIREMENTS 11
3.1 DATA STORAGE METHODS 11
3.2 EVALUATION OF RDBMS PRODUCTS 13
3.2.1 Architecture 14
3.2.2 Performance 14
3.2.3 Capacity 14
3.2.4 Interface 14
3.2.5 Network Routing 15
3.2.6 Location Transparency 15
3.2.7 Protocol Transparency 15
3.2.8 Stored Procedures 15
3.2.9 Triggers 16
3.2.10 Event Alerters 16

TABLE OF CONTENTS (continued)

Page

4.0 A REVIEW OF RELATIONAL DATABASE MANAGEMENT SYSTEMS 17
4.1 PRODUCT REVIEW 17
4.1.1 Oracle 17
4.1.2 Sybase 18
4.1.3 Informix 20
4.1.4 Ingres 21
4.2 SUMMARY 21

5.0 ALTERNATIVE ACCESS MECHANISMS FOR TRANSPORTATION RELATED DATABASES 23
5.1 REMOTE DATABASE ACCESS 23
5.2 REMOTE DATABASE ACCESS PROTOCOLS AND STANDARDS 23
5.3 ACCESSING TRANSPORTATION DATABASES VIA THE WORLD WIDE WEB 27
5.3.1 Servicing a Web Browser Database Query 28
5.3.2 Critical Database Access Elements 29
5.3.3 Availability of Access 30
5.3.4 User Interface 30
5.3.5 Data Integrity and Security 32
5.3.6 Conclusion 33

6.0 IMPLEMENTATION OF ORACLE DATABASE FOR TransLink™ 34
6.1 REVIEW OF CONCLUSIONS AND RECOMMENDATIONS 34
6.2 DESIGN OF DATA PROCESSING 34
6.3 CURRENT STATUS 35
6.4 IMPLEMENTATION REQUIREMENTS 36

LIST OF TABLES

Page

Table 1 - Summary of Relational Database Features 22

SUMMARY

One aspect of the TransLink™ research program is to have a laboratory where the ITS architecture and technologies can be explored, integrated, demonstrated and showcased. An integral part of this laboratory is a data warehouse where historical and real-time data can reside. This report examines the best method for managing this data. After a brief introduction, a listing of various types of data that could be seen as a result of the Intelligent Transportation Infrastructure (ITI) is presented. The researchers then proceed to examine the best method to handle those data needs. The conclusion of this report focuses on implementation, which describes the efforts underway to accomplish the goal of the research program.

1.0 INTRODUCTION AND SCOPE OF REPORT

1.1 INTRODUCTION

There are currently two Transportation Management Centers (TMC) operating within the state of Texas: TranStar in Houston and TransGuide in San Antonio. In addition to these facilities, several more TMC's are being planned for areas within the state, including Dallas, Fort Worth, El Paso, Austin, and Laredo. With each center making its own decisions regarding system design, core technologies, and database management, the integration and effective utilization of the data from these centers becomes a complex task.

As described in the project agreement (Project 7-2988) between the Texas Transportation Institute (TTI) and the Texas Department of Transportation (TxDOT), one aspect of the TransLink™ research program is to have a laboratory where the ITS architecture and technologies can be explored, integrated, demonstrated, and showcased. The TransLink™ laboratory replicates some functions of a TMC and serves as a research tool for the next generation of TMC's. In order to accomplish this research, data from the existing and planned facilities is necessary. However, since very little data is currently being stored, TransLink™ requires a data warehouse and a data management scheme. The use of a database to store data from these systems would allow for research to take place without impacting the day-to-day operations, and would also allow the existing data to be used for future planning and evaluation of the benefits of these centers and ITS.

Associated with the laboratory phase of this contract are three tasks:

Task 1-1. Expand Capability and Functionality of TransLink™ laboratory.

Task 1-2. Establish Information Database.

Task 1-3. Establish Real-time Connection to an Existing Traffic Management Center.

1.2 SCOPE OF REPORT

This report details the progress to date on Task 1-2, Establish Information Database. The researchers start with a brief introduction, list some of the various types of data that could be seen as a result of the Intelligent Transportation Infrastructure (ITI), and then proceed to examine the best method to handle those data needs. The report concludes with a chapter on implementation, which describes the efforts underway to accomplish the goal of Task 1-2.

2.0 DATA REQUIREMENTS OF THE INTELLIGENT TRANSPORTATION INFRASTRUCTURE (ITI)

2.1 THE INTELLIGENT TRANSPORTATION INFRASTRUCTURE

The Intelligent Transportation Infrastructure (ITI) can be though of as the core elements of a Intelligent Transportation System (ITS). Comprised of nine components, the ITI is really a set of building blocks for achieving an ITS implementation. While each component alone has the capability to provide significant benefits, the implementation of all the components into an integrated and coordinated system has the capability of dramatically improving transportation services. The nine components of the Intelligent Transportation Infrastructure (ITI) are:

- Traffic Signal Control
- Transit Management
- Freeway Management
- Electronic Toll Collection
- Regional Multi modal Travel Information
- Electronic Fare Payment
- Railroad Grade Crossings
- Emergency Management Services
- Incident Management

2.2 DATA REQUIREMENTS OF ITI COMPONENTS

As more and more transportation infrastructure is developed, a large amount of information is available regarding the current status and condition of the infrastructure. While this information could be physical information such as the width of the lanes and the type and condition of pavement, it can also be operational information such as the speed of vehicles using the roadway or the location of an accident. Regardless of the type, collecting and storing this vast amount of information is no small feat. Looking to the future of an intelligent transportation system, one realizes that perhaps even more data will be necessary, especially as each additional

component of the ITI is added into the system. These data can range from the relatively simple, as discussed above, such as traffic volumes on the road, to the complex, such as coordinated video feeds along a corridor where an incident has occurred. Regardless of the type of data or its use, a common theme is that at least some of the data must be collected and stored for later use. This collection and storage area is called a database. While this chapter concludes with an examination of the types of data that might be collected, the remainder of this report examines many aspects of database implementation, access, control and security.

Each section below examines the data requirements for a single component of the ITI. These listings are not meant to be totally inclusive. Indeed, since most components have not yet been implemented in a complete coordinated system, some data requirements may simply not be known at this time. As an additional point, many listings will show duplicated data needs, underscoring the need for a coordinated system that shares data across all components. It should be understood that this discussion of data requirements is for illustrative purposes for this report. TransLink™ is not involved in all nine components of the ITI and as such, the database needs, as discussed in future chapters of this report, are not as extensive as the complete listing below.

2.2.1 Freeway Management

The ability to manage a freeway pro actively instead of merely reacting as conditions develop is considered key to ITS strategies. To perform this task, data needs are varied, including:

- Real time traffic data
  - presence
  - counts
  - speeds
  - queue length
- Video Feeds
- Incident detection

- video cameras
- police reports
- citizen call-ins

- Historical data
  - counts
  - special events

2.2.2 Traffic Signal Control

To be effective, an intelligent deployment of a traffic signal control systems needs to be adaptive, that is, it must be able to react to changing traffic conditions. To accomplish this task, these systems must be able to process real-time traffic data from a variety of sources, as well as access and analyze historical data that might suggest similar patterns of control. When incorporated into an integrated system, traffic signal control systems should be able to give priority to transit and emergency response vehicles. Some of the data required for these systems include:

- Real-time traffic count data
  - inductive loops
  - magnetic sensors
  - video cameras
- Signal phasing
- Signal network progression
- Queue lengths (demand)
- Incident detection
  - video cameras
  - police reports
  - citizen call-ins

- Historical data
  - counts
  - phasing
  - results of timing patterns
  - special events
- Advance notification of transit or emergency vehicle priority needs

2.2.3 Transit Management

Apart from the traditional desire to have one's mode of transportation instantly available, a key factor in public transit ridership has been the on-time performance. Prospective riders will not take buses if they can't be assured the destination arrival time on the schedule is accurate.

To improve its ridership, transit must work smarter, not necessarily harder. By implementing systems that allow buses to be more time efficient and more in touch with the transportation system, transit could improve efficiency and provide better traveler information. Types of data required for this component of ITI include:

- Geographic Information Systems/Geographic Positioning Systems implementation for real-time location tracking (better information on arrival/departure times)

- Route optimization
  - Origin and destination passenger loadings
  - Travel times

2.2.4 Incident Management

The most important factor in an incident management system is to respond rapidly to situations, with the proper personnel and equipment. Not only can this rapid response potentially save the lives of crash victims, it also serves to return the facility to a fully operational status as

soon as possible, thus minimizing delays. However, in order for incident management to be effective, the system needs to be aware of many types of data, such as:

- Real-time traffic data
  - congested routes
  - routes with available signal pre-emption
  - available routes for removal of victims
- Incident data
  - nature of incident
  - type and number of any injuries
  - additional factors, such as the presence of Hazardous Material (HazMat)
  - video feeds showing incident

2.2.5 Emergency Management Services

The backbone of any emergency management system is the rapid response of emergency vehicles to the site where they're needed. In light of this, the data needs are very similar to that of an incident management system. These data needs include:

- Real-time traffic data
  - congested routes
  - routes with available signal pre-emption
  - available routes for removal of victims
- incident data
  - nature of incident
  - type and number of any injuries
  - additional factors, such as the presence of Hazardous Material (HazMat)
  - video feeds showing incident

2.2.6 Railroad Grade Crossings

With a high number of conflict points and potentially heavy traffic volumes, at-grade intersections have long been targeted for performance improvements, as evidenced by railroad grade crossing being included as a major component of the ITI. When a train, which travels at significant speeds and can take several hundred yards to come to a stop, approaches vehicular traffic at a railroad crossing, special difficulties can arise. For this component to function effectively, motorists need to be informed of the oncoming train presence and the crossing hazard. In addition, if the crossing is on a primary emergency access route, details as to the length of time the route will be unavailable are important. The data needs to accomplish these objectives include:

- Real-time train information
  - Speed
  - Position
  - Length
  - Route
- Grade crossing operation information

2.2.7 Regional Multi modal Travel Information

At times, deciding on the travel mode and the route can be difficult. If there is an incident on the freeway, the traveling public may wish to take another route, or to take mass transit. However, many times, this type of information is limited, if not in scope, then in coverage area. The goal behind this component of the IT is to develop a comprehensive regional information system that disseminates timely travel information to those that need it. To accomplish that task, a several types of data are required, including:

- Real-time traffic data

- congested routes
- non-congested routes

- Incident data
  - nature of incident
  - anticipated delay time
- Transit data
  - routes and schedules

2.2.8 Electronic Fare Payment

Focusing on a central theme of convenience to the traveler, electronic fare payment holds promise to reduce some of the headaches associated with traveling, such as the need for exact change or the need for any money at all. Through some sophisticated hardware equipment and software packages, electronic fare payment could be used in mass transit or parking lot applications--wherever the traveling public needs available cash. This reduction in the actual handling of cash would greatly benefit public agencies as well. To date, the data requirements for this component are not well-defined, but certainly include the following:

- Traveler information
  - vehicle
  - owner
  - current location
- Financial information
  - available funds

2.2.9 Electronic Toll Collection

A significant delay point on toll roads occurs at toll plazas. Toll plazas also are also a significant cost to the operating agency. Implementation of a system that would allow for the

automatic collection of tolls would reduce traveler delay and reduce the operating cost of the agency. The data requirements for these systems are similar to electronic fare collection schemes and include the following components:

- Traveler information
  - vehicle
  - owner
  - current location
- Financial information
  - available funds

3.0 DATA STORAGE REQUIREMENTS

3.1 DATA STORAGE METHODS

Perhaps the concept made abundantly clear by the various listings of data requirements in the previous chapter is the tremendous amount of data that not only must be collected and stored, but also organized for effective retrieval and analysis. In addition, there is a need for the data to be retrieved for use in many different applications all at the same time.

As an simple example of the demands placed on the data storage and retrieval mechanisms, consider the data requirements of the freeway management component and the incident management component of a traffic management center. During an incident on the freeway, both of these components share some of the same data, such as video images and information relating to the location, severity, and type of incident. However, their usage of this data is quite different. The freeway management component analyses this information to examine methods of alerting motorists of delays and possibly rerouting traffic to avoid the effects of the incident while the incident management component uses this information to determine the appropriate level of response from emergency service vehicles and if necessary, the best method for getting casualties to treatment facilities. While both components serve very different purposes, they both rely on some of the same data at the same time.

Although the transportation data storage and retrieval mechanisms can be affected from a variety of methods, the amount of data being collected, stored, and retrieved dictates the use of a database management system. Database management systems can be categorized as a flat file database, a CODASYL database management system, a relational database management system, and an object oriented database management system. These are briefly described as follows:

- Flat File Database -- A flat file database is normally created as part of the application development. The advantage of a flat file database is that the data format and the manipulation of the files is totally under the control of the

applications developer. The data file format and access can be optimized for speed by the developer. Flat files are most suitable for applications that have limited storage requirements and few users, such as providing physical configuration information and signal timing parameters to a traffic signal systems. The flat file database's major disadvantage is the difficulty encountered in sharing data among different applications. With a flat file database, all programs have to use the same formats and protocols, and programs to control access and to manage the data must be developed. This adds complexity to the applications programs and other alternatives to flat file database systems are better suited to TransLink™ laboratory's needs.

- CODASYL Database Management System -- CODASYL is an acronym for the conference of data system languages. CODASYL is a standard developed about 30 years ago to define the implementation of databases in a networked data model. CODASYL data base management systems are very fast because the format in which the data is stored, or schema, is fixed and compiled into the application as a library of subroutines that are linked to the application (CODASYL is not a separate system from the application). CODASYL systems do not adapt well to changes to data formats. When there is a need to change the data schema the CODASYL database must be regenerated, which is not a catastrophic event in itself, but none of the previously stored data is accessible to the modified system. A CODASYL database management system is not appropriate for consideration due to the diversity and rapidly changing nature of the ITI data that the TransLink™ laboratory will encounter.

- Relational Database Management System -- A relational database management system (RDBMS) is designed to allow many users to share the same stored data. It is normally a separate system from the application. Data sharing is achieved by the RDBMS hiding the data from the user application and by enforcing access privileges and managing transaction processing for them. A fixed record format is used and groups of data are related to other groups of data through an indexing mechanism that is part of the data itself. The index serves to locate information quickly, similar to the index of a book. Access to data is provided through a standardized query language and advanced programming interfaces. RDBMSs were designed for large, data intensive applications such as those envisioned for the TransLink™ laboratory.

- Object Oriented Database Management Systems -- Object oriented database management systems (OODBMS) are recently developed database management tools, and they reflect contemporary software design philosophy. They combine the CODASYL database management system technology with object-oriented programming, and were developed specifically to support explicit relationships between entities that are not easy accommodated by an RDBMS. OODBMS are

particularly well suited for applications such as computer aided design, geographic information systems, computer aided system engineering, and graphics. However, for most of the applications envisioned for TransLink™ laboratory, an OODBMS performs no better than an RDBMS, and it is generally more complex to use.

From the above, the researchers concluded that the most applicable product to meet the requirements of TransLink™ is a relational, client-server database system. An RDBMS provides the capability for:

- Unlimited storage
- Order imposed on the data
- Ability to simultaneously process multiple requests from various sources
- Ability to simultaneously process multiple requests for the same data

3.2 EVALUATION OF RDBMS PRODUCTS

The purpose of a relational database management system (RDBMS) is to provide an expedient and orderly method of storing transportation data, and subsequently retrieving the data in a prescribed order or relationship such that it is useful for research and applications programs. Although the basic functions of a database are storage and retrieval of data, modern database systems contain numerous tools for convenient report generation, networking, browsing, and user interface development.

Despite the fact that many vendors offer the same basic database features, there are differences between their implementation, as well as their capacity, programming languages, and development tools. In Chapter 4, the researchers present the results of a brief review of four relational database management systems that are applicable for client/server transportation systems. This section discusses some of the important components that should be examined when comparing relational database management systems.

3.2.1. Architecture

In a dedicated-client/server implementation a database server is specifically charted to each client. In a multi-processor system this construct is efficient where the operating system schedules multiple servers over all available processors. However, in a multi-client/server implementation, a single server accommodates multiple client processes which reduces system overhead. All of the RDBMSs investigated are based on the workstation client/server model where the server executes requests from many clients. These RDBMS clients may be either workstations or personal computers.

3.2.2. Performance

The performance of an RDBMS is affected by the database organization, type of database transaction, and storage efficiency. The performance is also affected by the CPU, network traffic, and disk activity. The ability of a RDBMS to monitor performance and to provide a means to tune the system to respond to the application's configuration and requirements is one of the largest factors contributing to the performance of a RDBMS.

3.2.3 Capacity

It is envisioned that some of the transportation databases will be quite large. However, each of the database products reviewed have essentially unlimited storage capabilities when implemented on a workstation.

3.2.4 Interface

Most RDBMSs provide access to data through SQL (Structured Query Language) a standardized, high-level query language. SQL allows users to formulate inquiries of the data in an ad hoc manner using a natural language-like syntax. Data access can also be provided through an

Advanced Programming Interface (API), benefitting applications written in C/C++.

With a macro language API, a program or macro is developed from a series of commands that the application program can invoke and run. The macro can be compiled which increases speed. However, modifications are time consuming as the macro data formats are fixed; and, when changed, it is necessary to verify that the code is correct everywhere the macro is used. With a linkable library API, direct access to RDBMS functions is provided through linkable library calls made by the application. Linkable library calls provide a flexible interface; however, they add to the system overhead since a sequence of instructions must be executed.

3.2.5. Network Routing

An RDBMS that features network routing will automatically select an alternate network route if the primary configured route is inoperable.

3.2.6 Location Transparency

An RDBMS that features location transparency allows development of client applications without considering the location of shared data.

3.2.7 Protocol Transparency

An RDBMS that features protocol transparency is independent of the different network protocols that may exist.

3.2.8 Stored Procedures

SQL commands and program statements can be compiled and stored in the database. They execute quickly and can be shared. A single command can perform multiple tasks, thereby

reducing network traffic.

3.2.9 Triggers

Triggers are a mechanism to invoke the RDBMS to perform an action when certain requests (insert, delete, update, etc.) are made. Triggers are used to enforce integrity rules and to replicate data across multiple databases.

3.2.10 Event Alerters

An RDBMS that features event alerters provides for invoking applications in response to the occurrence of an event. They are used to start an application activity as a result of an event signaled by a transaction.

4.0 A REVIEW OF RELATIONAL DATABASE MANAGEMENT SYSTEMS

4.1 PRODUCT REVIEW

The researchers reviewed four database products for use in managing transportation related data. They were Oracle7, Sybase, Informix, and Ingres. Following is a brief description of the findings for each product.

4.1.1. Oracle

Oracle7 is based on a multi-threaded, multi-server architecture. Databases can be distributed across heterogeneous hardware platforms and communications networks. Oracle has dramatically improved the functionality of the Oracle7 database over its previous versions. The new version, for example, offers a parallel server for use in a clustered environment handling more processes, such as log managers, buffer managers, and dispatchers.

Oracle7 is ANSI X3.135-1992 (SQL-92) compliant. Through the use of Oracle's SQL*Net, network routing capability and protocol transparency features are provided. Oracle7 features database triggers and event alerts, and has stored procedure capabilities. Optimization capabilities include determining the least time-cost query plan for the server to access data and consideration of network costs. Tools include a report generator, database administration, and performance monitoring.

Oracle7 supports larger databases than most of the leading relational database vendors, supporting up to a 2 Terabyte database. Oracle also has strong support for the IBM database environment through SQL*Connect to DB2, RMS, IMS and SQL/DS.

Oracle's own fourth generation language, called PL/SQL, provides Oracle customers with a consistent language for all of Oracle's tools. Initially, PL/SQL was designed as a procedure language. However, due to customer demand, Oracle added some fourth generation constructs such as loops and conditionals. Oracle's client/server tools strategy is to take the various components that have evolved and integrate them into what it calls CDE (Cooperative Development Environment). CDE includes many existing components, several of which (Oracle Forms, Report, and Graphics) have been rewritten, while others have simply been packaged as part of CDE.

4.1.2 Sybase

Sybase was the first RDBMS vendor to position itself in the client/server market. It was designed to support both on-line transactions and decision support requirements. Sybase was also one of the first vendors to publish the interfaces to the gateways and tools within their environment. Third-party developers can now easily create client/server tools for Sybase.

From a technology perspective, Sybase's strengths include its support of user-defined exits within stored procedures and its ability to implement distributed query processing. Sybase's multi-server hierarchy can be configured to pass particular requests or data elements to other servers. This can be used to let one server service time critical processes and other servers service non-time critical functions. In addition, Sybase supports event alerters and triggers.

The Sybase RDBMS is a multi-threaded, multi-server architecture that supports distributed, mobile databases. When a database at a location becomes full or overworked, the database is divided and distributed between multiple locations and computing platforms.

Sybase has a full featured API, and offers a macro language interface and a linkable library with most of the capabilities of the SQL macros. However, Sybase is not compliant with ANSI

SQL-92. Report generation, database administration, and performance monitoring tools are provided.

Sybase recently begun to develop and market its own set of client/server application development tools. The only product Sybase had for reporting and query development was the Apt Workbench, which was character-oriented and, therefore, lacked the ease of use and depth of functionality of emerging graphical development tools. In the past year, Sybase began building its presence in the graphical development tools business, primarily through acquisitions. These acquisitions included the DEFT design tool; the Gain multimedia development system; and Wybern, a company that provided Sybase with a graphical development environment. These are the foundation of Sybase's forthcoming graphical development tools family, called Momentum.

Within this family, GainMomentum will be available to build GUI based applications such as information kiosks. GainMomentum has its own fourth generation language called GEL.

BuildMomentum will be the primary application development tool for mission critical applications. Users of the AptWorkbench language tool will have a migration path to BuildMomentum, and the BuildMomentum language will be merged with the AptWorkbench fourth generation language. The emerging language is a visual development language called Object Script. Sybase intends to use this language as a supervisor language to manage several different languages simultaneously, but this is not yet fully developed.

A third product is called EnterpriseMomentum, a models-based development environment. This environment will include an object-oriented repository and a methodology for developing large-scale client/server applications. Sybase, therefore, intends to offer a repository-based model including process flow, data relationships, and business rules.

4.1.3 Informix

Informix is one of the fastest relational database systems. Its price has attracted third-party value added resellers. As a result, there is a huge portfolio of niche applications written for Informix.

Informix-OnLine 5.0 is an ANSI SQL-92 compliant RDBMS with distributed, open client-server capabilities. It provides location transparency. Its distributed computing capabilities are implemented through Informix-Star 5.0, which implements multi-site reads, joins, and updates. Two-phase commits are handled automatically to coordinate participating servers involved in a transaction.

Informix features stored procedures and triggers, but does not support event alerters. Database transaction flows are monitored to identify bottlenecks. Report generation, configuration and management, and performance monitoring tools are available. Optimization capabilities include determining the least time-cost query plan for the server to access data, and consideration of network costs. A hypertext-based on-line help mechanism is provided.

One of Informix's greatest strengths is its fourth generation language. Its two products, 4GL and 4GL/GX, are considered excellent development tools. They compile into a pseudo-code that enables developers to incrementally compile code. Informix offers a second language called Hyperscript, which was initially part of its WingZ spreadsheet product. Recently, Informix has announced a MS-Windows version of its fourth generation language as a graphical development environment. This will allow developers to do their development work in a Windows environment. Informix has built strong relationships with CASE tool vendors and has licensed Soft Bench, Hewlett Packard's CASE messaging system.

4.1.4 Ingres

Ingres, now part of the Ask Company, offers a complete RDBMS with open, client/server system architecture. The multi-threaded, multi-server architecture of Ingress and its transparent protocol conversion allows applications and data to be distributed across multiple hardware platforms. Global database consistency is provided through a transparent two-phase commit access to heterogeneous databases, and a distributed data dictionary. Ingres' network routing capability selects an alternative route automatically if the primary route is not available. Ingres/Star and Ingres/Net provide the mechanisms for implementing distributed computing. Ingress has plans to offer a peer-to-peer replication capability.

Ingress provides a subset of ANSI SQL-92, but is not fully compliant. Ingress does provide the features of stored procedures, triggers, and alerters, and a query optimizer that enhances performance of operations across heterogeneous computer platforms and databases. Tools are provided for report generation, database administration, and performance monitoring.

Ingres has focused on its distributed database capabilities and integration of its clientbased tools. Ingres' Windows fourth generation language was the first RDBMS graphical development environment offered. Ingres also offers Vision Pro for developing character oriented applications, and tools are available to migrate Vision Pro applications to Ingres' Windows fourth generation language, Vision Pro. Applications can be moved between Unix and Windows.

4.2 SUMMARY

A summary of the relational databases reviewed is shown in Table 1. While all of the products have various strengths and weaknesses, none is clearly dominant to the point of total exclusion of all the other products. However, given the fact that the Houston TranStar TMC utilizes Oracle and that the Computer Science department at The Texas A&M University System is quite familiar with Oracle and finally that the design of TTI's TransLink™ laboratory utilizes

the Computer Science Department as a remote data repository, the selection of Oracle for use in the TransLink™ laboratory for the management of the transportation is the best choice.

		Database		
Feature	Oracle	Sybase	Informix	Ingres
Location Transparency	X	X		
Protocol Transparency	X		X	
Alt. Network Routing	X		X	
Stored Procedures	X	X	X	X
Triggers	X	X	X	X
Event Alerters	X	X		X
Multi-phase Commit	X		X	X
ANSI SQL	X		X	SUBSET
C/C ++ API		X		
Optimizers	X		X	X
Database Administration	X	X	X	X
Report Generation	X	X	X	X
Performance Monitoring	X	X	X	X
On-line Help	X	X	X	X

Table 1 - Summary of Relational Database Features

5.0 ALTERNATIVE ACCESS MECHANISMS FOR TRANSPORTATION RELATED DATABASES

5.1 REMOTE DATABASE ACCESS

Part of the allure of warehousing a large amount of data from transportation systems is not only its availability for use within the TMC, but also the ability to disseminate the data in a variety of formats. The dissemination of transportation data is useful for many purposes. Perhaps the most useful purpose is to provide the general public access to the data.

Given that data is published and presented in the appropriate manner, the general public can and will use the information to its advantage. One application where this has been proven many times is the availability across the Internet of freeway speed maps in various cities. Houston and Seattle have long had a publicly available speed map that shows speeds and congested links on many important commuter routes. Other areas of the country have video feeds and/or snapshots available for obtaining visual information about what the ride home or into work might offer.

The data needs for these resources are all served in one way or another by a database. The planning for using the data in these methods should be incorporated into the database design from the beginning. Section 5.2 discusses some of the protocols and standards associated with remote database access (RDA), while section 5.3 addresses some of the issues that arise when allowing access to databases across the Internet.

5.2 REMOTE DATABASE ACCESS PROTOCOLS AND STANDARDS

As discussed in Section 3.1, the use of a relational client/server architecture for transportation systems databases is most appropriate due to the distributed nature of systems used for transportation purposes. This architecture is more efficient as it permits applications and data

to be located where they are used most frequently, yet the data and the results of the application process are available to other locations when needed. This configuration reduces network data traffic and allows applications to process more quickly. However, in some situations, the efficiency of the client/server architecture is accompanied with the difficulty of accessing databases remotely due to data integrity, format, and protocol issues.

Some of these issues are resolved by embracing remote database access (RDA) as one of the standards used during the system's development. RDA is a communications protocol for remote database access that has been adopted as an international standard by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). It has also been adopted as an American National Standard by the American National Standards Institute (ANSI) and as a Federal Information Processing Standard (FIPS) for the U.S. federal government. RDA was published as a combined ANSI/ISO/IEC standard in 1993. It consists of two parts as follows:

- Part 1 — Generic RDA: ANSI/ISO/IEC Standard 9579-1:1993; specifies the generic model, service, and protocol for arbitrary database connection.
- Part 2 — SQL Specialization: ANSI/ISO/IEC Standard 9579-2:1993; specifies additional protocols for connecting databases conforming to Structured Query Language (SQL).

RDA provides standard protocols for establishing a remote connection between a database client and a database server. The goal is to promote the interconnection of database applications among heterogeneous environments.

An RDA client is an application-process, within an open system, that requests database services from another application-process called an RDA database server. An RDA database server interfaces to a process that controls data transfers to and from a database, within the same or another open system, that supplies database storage facilities and provides, through Open Systems Interconnect (OSI) communication, database services to RDA clients. The RDA client

has the ability to initiate RDA service requests, while the RDA server can only issue RDA service responses to reply to such requests.

The RDA standard specifies an RDA service interface to an RDA communication element that exists both at the client and at the server. The RDA communication element converts RDA service requests into transaction processing (TP) service requests as part of an OSI interconnection.

The RDA service interface consists of service elements for 1) establishing an association between the client and server remote sites and managing connections to specific databases at the server site, 2) transferring database operations and parameters from client to server, 3) transferring the resulting data from server to client, and 4) managing transactions. Client database operation requests are sent as character strings conforming to the SQL language. Resulting data and/or errors and exceptions are described and represented by the server using the ISO Abstract Syntax Notation One (ASN.1) standard.

An RDA transaction dialogue is uniquely identified within the scope of the OSI environment, and all RDA operations occur within the bounds of an RDA dialogue. An RDA client and database server communicate by means of RDA services. RDA operations enable an RDA client to request any of five types of RDA services:

- RDA dialogue management services, to start and end RDA dialogues;
- RDA transaction management services, to start and end RDA transactions;
- DA control services, to report the status or cancel existing operations;
- Resource handing services, to enable or disable access by RDA clients to data resources; and
- Database language services, to access and modify data resources.

An RDA client may request services without waiting for the results of previously requested RDA operations. Thus, an RDA server may have several RDA operations outstanding for a particular RDA dialogue.

An RDA dialogue exists only in the context of an established application-association and ceases to exist if the association is released. A failed RDA dialogue cannot be recovered. The process of recovery after a failure is beyond the scope of the 1993 RDA standard, and recovery actions by the application are generally necessary. In the event of dialogue failure, it is a requirement that all changes made to data resources by any RDA transaction that has not already terminated when the dialogue failure occurred be rolled back by the database server during its recovery process. If an RDA dialogue is terminating when an RDA dialogue failure occurs, then it may either be committed or rolled back.

An RDA transaction is a complete set (unit) of processing as determined by the RDA client. During an RDA transaction, the execution of a sequence of database access services that change data resources enables the set of changes to be handled as an atomic unit. When the RDA transaction is terminated, either the whole set of changes is applied to the data resources or no changes are applied. The RDA client requests termination of an RDA transaction by requesting the RDA server either to commit or to roll back the complete set of changes of that transaction. Changes made to the data content of data resources during an RDA transaction are not made available to other RDA clients until that RDA transaction is terminated at the RDA server.

RDA provides a choice of two application-contexts for managing RDA transactions: 1) a basic application-context for one-phase commitment, and 2) a TP application-context for two-phase commitment. The RDA protocol for the basic application-context is completely specified in the RDA standard, whereas the protocol for the TP application context is dependent upon the ISO/IEC distributed transaction processing standard (ISO/IEC 10026).

RDA is appropriate for remote access to a database in any context where lower layer transport protocols have already been established. RDA protocols have been shown to work properly in both Open Systems Interconnect (OSI) and Internet communications environments. The Internet RFC 1006 is the guide used for executing RDA over a TCP/IP connection. It is expected that RDA will become the basis for all interconnection among SQL database management products from different vendors. Interconnection among database products from the same vendor will likely continue to use vendor specific communication and interchange forms.

The existing RDA/SQL Specialization standard does not yet support all of the facilities in the SQL-92 (ISO 9075:1992) language standard. Instead, it more closely approximates support for just the entry SQL level of SQL-92. Work is in progress on an amendment to the RDA/SQL Specialization to support all SQL-92 features. Adoption as an international standard (ISO/IEC 9579-2) is expected by 1997.

5.3 ACCESSING TRANSPORTATION DATABASES VIA THE WORLD WIDE WEB

Currently, there are several public and private agencies that are experimenting with the use of the portion of Internet known as the World Wide Web (WWW) to disseminate traveler information. The principal use has been to display graphically the vehicle speed information obtained from freeway surveillance systems and to provide general traveler information such as bus schedules, construction advisories, agency information, etc. This new usage is an example of the web's increasing acceptance by both the general public and the technical elite.

As the web's popularity grows, its use will be gradually shaped and specialized for specific applications. Those applications that lend themselves to the strengths of the WWW will find widespread use while those that are inhibited by the web's shortcomings will exist only as applied experiments. As new transportation applications are developed, many of them will utilize the web technology for user interaction. One application of interest is the access of transportation related databases via the WWW.

To examine database access via the web, the concept of what is meant by "database access" must be defined. It could be argued, for example, that the entire base of knowledge contained within the WWW is one huge database and that one is effectively engaged in database access by browsing any page on the web. Hypertext Markup Language (HTML) formatted documents would be analogous to database records, and the client browser would simply be the user interface to those records. According to this definition, the web is certainly a suitable tool for database access, but this is too general a definition for our purposes. Certainly, web browsing can be considered within the realm of information access, but it is not what one thinks of as a typical database. Therefore, the researchers will limit the definition of database access to the concept of accessing a set of data with well-defined fields.

The degree of control over the data implied when the database is accessed is another point of ambiguity. For the context of this discussion, the range of access control is defined as follows:

- Search Level -- Search level access allows a user to query a database for a particular item or a small set of items.
- Browse Level -- Browse level access goes a step further and allows the user to view the entire set of data or large subsections of the data.
- Write Level -- Write level access represents the highest degree of control in which a user can actually change the data.

5.3.1 Servicing a Web Browser Database Query

To fully appreciate the behind the scenes complexity of using database information on the WWW, it is perhaps important to understand how the web server services a database query.

A user requests information by interacting with a web browser, which is usually a graphic interface. The web server receives the request and links the information received to a Common Gateway Interface (CGI) script. CGI is the standard tool by which browsers access other information and programs that are stored on a web server. The stored CGI script is a program to

invoke action by the server. This action leads to a query (usually in the format of embedded Standard Query Language) to the database server. The database engine receives the query and processes it. The result is returned to the web server through another CGI script operation and then transformed into an HTML format. The web browser displays the resulting information.

Of particular relevance to the discussion of access control is the web's handling of browse level database access. While it is simple to manage a small amount of information (search level) the massive amounts of output that could be retrieved from a browse level access is often difficult to manage over the web. The best formatting method for large amounts of data is probably a table, however there are limitations in the implementation of tables in the current HTML specifications that make displays of this type still somewhat problematic. Another limiting aspect of database browsing is the somewhat limited connections speeds. Retrieving a large amount of data via the WWW is often a very time consuming process.

5.3.2 Critical Database Access Elements

The current implementations of transportation database access via the web falls largely within the category of search level access. The critical question to be answered is whether the WWW is a viable tool for browse and write level access to transportation databases as well.

To answer this question a consideration of the critical database access elements and a comparison of these elements to features available on the WWW is necessary. In this analysis the following three main database access elements were considered

- Availability of access;
- User interface; and
- Data integrity and security.

5.3.3 Availability of Access

Without a doubt, the most attractive feature of providing database access on the web is availability. It is a fascinating concept to consider that through a relatively small amount of effort in setting up a WWW server, database access can be provided to the entire world. The concept of client-server databases is taken to the extreme where a server can be any computer on the Internet and the client is simply any other computer on the Internet.

The only other method of database access that provides a comparable level of access availability is Interactive Voice Response (IVR) systems where database access is provided via the telephone (banking-by-phone is a typical IVR application). However, the use of a telephone keypad as the user interface is significantly limiting.

The benefit from wide spread information availability provided by the web strongly depends on the nature of the database. For example, an employee database for a single-office company would not fully benefit from a web database since it is likely to be accessed only by people within that location. However, a customer database for an international corporation with many offices would greatly benefit from the web access. Duplicate storage and data entry requirements at different offices would be eliminated, and each location would have available access to the same data at the same time. This concept readily applies to transportation information, especially in applications that have already been discussed, such as the visual display of roadway speeds for the commute to or from work.

5.3.4 User Interface

Information systems and database access have progressed over the years from a mainframe-based, central-repository system to today's much touted client-server model. One of the causes of this transition is the improved user interface provided by the client-server system. In a client-server system, the client has its own processing capabilities which allow for a much better

user interface. Today's Graphical User Interface (GUI) windowing environments for personal computer and workstation applications allow users to access the data they need with sharply reduced learning curves as compared to older character-based interfaces that are often associated with central-repository systems.

An interesting aspect of the WWW is its diverse rank of clients. It is possible to traverse the web using a text based browser such as Lynx. However, graphical browsers such as Mosaic and Netscape are much more common. Assuming that a graphical browser is being used, the user interface for database access is fairly good. Another interesting factor about the WWW user interface is the rapid speed with which improvements to browsers are made. With freely available beta versions of the latest browsers available, it is quite possible to stay on the leading edge of user-interface technology when utilizing the WWW.

HTML, in most of its varieties, provides support for forms -- graphic elements for accepting user input. Available form elements include text boxes, radio buttons, drop down lists, scrolling style lists, buttons, and check boxes. These provide for a wide range of methods for accepting user input, allowing the developer to match the appropriate form element to the most intuitive way of gathering input from the user.

The graphic web browser environment is a marked improvement over plain text-based database management systems where the user had to memorize cryptic prompt line commands to access data. However, graphical web browsers do not allow for all user interface tools, such as drop down menus and dialog boxes that are common in GUI client-server environments. Database development tools, like Powersoft's Power builder, are often combined with back end data processors, such as Oracle. This combination gives the developer more flexibility in designing an intuitive user interface than is currently possible in even the most advanced web browsers.

With regard to the user interface, it is fair to classify data access via the web as acceptable. It is a definite improvement over command line interfaces, but it is not as good as a client-server GUI interface. The user interface is an essential component of a useful database access. However, the web interface provided through form-based graphical browsers is neither bad enough nor good enough in itself to make or break the viability of full database access via the web.

5.3.5 Data Integrity and Security

While widespread accessibility is a great advantage of the WWW, it is ironic that it is also the source of a number of its most serious drawbacks, namely in the areas of data integrity and security. There are numerous web search applications, but very few web applications that allow for the alteration of data. This limitation results from the essence of how the WWW exists as a stateless medium where the server is not required to track potential clients. The web server has no way of controlling who is modifying its data. The server gives out a copy of a record when the client requests it. The server then modifies the actual database only after the client has modified the record (outside the servers knowledge) and submitted it back. If the server gives out copies of the same record to multiple users, each user has no way of knowing whether their changes will be overwritten by a subsequent update by another user.

Security is an obvious concern occurring from wide accessibility. A database of sensitive employee salary records would not lend itself to use on the WWW unless access was password protected. Even password protection is only a minimal deterrence given the open broadcast nature of Internet communication. Complicated encryption schemes would be required for truly private database information. Although progress is being made in this area, the security issue significantly complicates the process of providing access to secure database information via the web. However, it should be noted that many aspects of transportation data, such as speeds, incidents, transit arrival and departure times, and the like are not considered secure information and do not require password protection.

5.3.6 Conclusion

Given the three main criteria of availability of access, user interface, and data integrity and security for evaluating an effective medium for providing database access, the WWW gets mixed results. Its greatest asset is its level of wide-spread accessibility; but, for that very reason, falls short on issues of security and data integrity. On the issue of user interface, the web has both good points and bad points.

It is possible to argue that if there is true interest in determining how useful the WWW is as a means of disseminating transportation database information and access, we should then look to its current uses and draw our conclusions from that. The trend in providing database access on the web is definitely on an increase, but the typical application is a simple query mechanism and does not fit our previous definition of full database access that includes browse and write level access. However, the WWW is still a relatively new technology and new methods of handling problems are conceived each day. There are already several companies that offer packages that remove the burden of CGI scripting and offer significant enhancements to security measures.

As the researchers suggested earlier, it is quite difficult for the current web servers to handle vast amount of real-time data for control purposes. The major difficulty lies in HTML, which currently has not implemented sufficient control mechanisms to act as a powerful language. Sun Java is a C/C++ type language, yet it has built-in hypertext functionality, object-oriented programming capability. In this sense, Java might be more suitable for web access of transportation applications, though it is also far from maturity.

Full database access via the WWW, whether through change in the medium or change in its use, is a critical step in its evolution. Without changing the base components, the limits have been pushed in terms of what can be done with simple informational displays. Full database capabilities will push the usefulness of the WWW to new levels.

6.0 IMPLEMENTATION OF ORACLE DATABASE FOR TransLink™

The TransLink™ research laboratory has been under physical expansion for much of the past year. That current state of the lab is documented in the technical memorandum for task 1-1, " Expand Capability and Functionality of TransLink™ laboratory." of project number 7-2988 between the Texas Transportation Institute (TTI) and the Texas Department of Transportation (TxDOT). Task 1-2, the subject of this report, goes beyond the physical expansion of the lab and seeks to define the underlying mechanism by which all applications in use in the lab will obtain and interact with the data from the transportation system.

6.1 REVIEW OF CONCLUSIONS AND RECOMMENDATIONS

The researchers have examined some of basic the data needs for much of the Intelligent Transportation Infrastructure. As discussed in Chapter 2, the possible sources of these data are greatly varied in source, size, type, update frequency, and availability. The complexity of gathering the data should not be underestimated. Chapter 3 examined possible methods of storing the data, which resulted in the conclusion that a relational client-server database is the best model to pursue for this application. Accordingly, Chapter 4 reviewed the applicable software packages and recommended the adoption of Oracle for the implementation platform. With the idea of increasing the utility of any system, Chapter 5 examined issues regarding remote database access and explored the use of the Internet (via the World Wide Web) as a platform for making transportation information available.

6.2 DESIGN OF DATA PROCESSING

The TransLink™ research laboratory is poised to begin implementation of the recommendations from this report. The Computer Sciences Department of The Texas A&M University System will be the host site for the Oracle database. The use of the Computer Sciences Department is strategic since it nicely replicates both the client-server model and the distributed

nature of the transportation data sources and the database. The TransLink™ laboratory housed at the Texas Transportation Institute will continue to function as the main applications warehouse, with data requests processed and served from the Computer Science Department.

6.3 CURRENT STATUS

It must be realized that the database designed for TransLink™ operations is not a static environment. The database design and structure will be driven by the needs of the research and in fact will change over time as various research needs are identified and undertaken. As with the listing of the ITI data requirements in Chapter 2, all of the data requirements for this research program are not presently known. Therefore, it is not possible to specify a complete database design that is all inclusive of the TransLink™ research needs over the next several years.

What is possible, however, is to design the starting point. The work to date has given TransLink™ that starting point. The Oracle database in the Computer Science Department is installed, debugged, and running efficiently. Tests have been conducted as to the feasibility of transferring real-time data between the sites which have proved to be successful. The initial design of the database is suitable for research on ITI applications.

The researchers are sending a letter to the Texas Department of Transportation asking for the details of the core database structure used at the Houston TranStar TMC. This information will serve as reference to determine some broad categories of data needs. It is not anticipated however, that the TranStar database design can simply be implemented in TransLink™ and be applicable to all needs over the life of the project.

In addition to the Houston data, efforts have also been focused on acquiring the archived data from the San Antonio TransGuide TMC. This data will allow the researchers to move forward on implementing a small scale database design until the Houston structure can be evaluated for use in the TransLink™ environment. The San Antonio data also provides a

historical archive of the San Antonio freeway system, which provides for some excellent research opportunities.

6.4 IMPLEMENTATION RECOMMENDATIONS

The researchers make the following specific recommendations:

1. Continue to develop the TransLink™ laboratory database capabilities in an open architecture format to be able to interface with or migrate to other database programs, such as Sybase, Informix, or others.

2. Utilize the existing Oracle database installation at the Computer Science Department at The Texas A&M University within the TransLink™ research program to function as the data repository and warehouse.

3. Utilize the San Antonio historical data to design and populate a database for initial research purposes.

4. Prepare a letter to the Texas Department of Transportation asking for the design details of the Houston TranStar TMC and the San Antonio TransGuide TMC.

5. Implement a scaled-down version of the Houston TranStar TMC database design within the Oracle program that also features the capability to interact with the San Antonio database.

6. As real-time data feeds from Houston and San Antonio are implemented, populate the database with data from the different cities to build an historical archive.