TECHNOLOGY DEVELOPMENT AND SYSTEM INTEGRATION FOR A VEHICLE MILES TRAVELED BASED REVENUE COLLECTION SYSTEM PROTOTYPE

Central Database Documentation
Version 1.4

Subtask 4.2

Prepared for
Oregon Department of Transportation

Revision Date: 05/28/04

Prepared by
Department of Industrial and Manufacturing Engineering
Oregon State University

Introduction

The SQL Server 2000 SP3 Central Database (Central DB) was designed and built by the OSU research team to store and manipulate VMT data records from all Oregon vehicles equipped with GPS-based on-vehicle devices. The Central DB is the persistent component of a five-tier application architecture, as depicted in Figure 1.

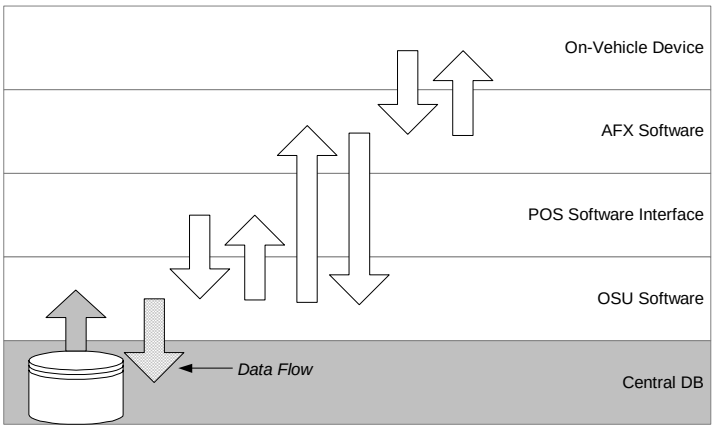

Figure 1. Central DB and Five-Tier Model of Application for the GPS-based Technology Configuration.

The Central DB has the following characteristics:

- *Maintains persistent storage of VMT data records*
  The TRANSACTIONS table stores useful information for enforcement purposes. Important fields in this table are the vehicle identification number (VIN), the station identification number, the VMT-based fee charged, the amount of fuel purchased, and a transaction's time stamp along with other complementary fields.
- *Calculates VMT fee per zone for each transaction*
  When a transaction occurs, the fee for each zone is calculated by the OSU software and then stored in the FEE_CALCULATIONS table.
- *Allows a database administrator to maintain VMT fee information*
  New zones and updated zone prices may be manually entered into the VMT_RATES table.
- *Generates a variety of reports*
  - *Individual vehicle transaction log report*
  - *Compliance and fee verification report*

Platform and Specifications

SQL Server Enterprise Manager (version 8.0) was used as the platform to develop the Central DB. It is a normalized relational database that uses the standard query language (SQL) to issue queries, as defined by the ISO/IEC JTC1/SC32, WorkGroup 3, Database Languages committee (as per Technical Specification 5.5, Kim and Porter, 2003).

For the purpose of demonstrating the prototype GPS-based technology configuration, SQL Server was installed on a DELL PowerEdge 2650 Dual Processor server running Windows Server 2003 Enterprise Edition. The server has five hard drives, divided into four volumes, and has been configured to resemble a typical transaction-based DB server. The four volumes are:

- *System*. The primary system driver.
- *Data*. The DB data files are stored on two mirrored hard drives, providing RAID 1 (redundant array of inexpensive drives) data protection.
- *Transaction*. Stores transaction logs.
- *Temp*. Stores temporary DB.

Entity Relationship Diagram

An entity relationship (E-R) diagram is a tool used to describe the business needs or "rules" that a database is designed to support. An example of a business rule for the VMT-based revenue collection system is that each fueling transaction must involve exactly one gas station and one vehicle. Another business rule is that each VMT zone must be able to charge a different fee per mile, requiring many calculations to determine the total VMT fee charged to a customer during a transaction.

For the VMT based revenue collection system prototype, the data and functional requirements were mapped to a logical model represented by the E-R diagram, depicted in Figure 2.

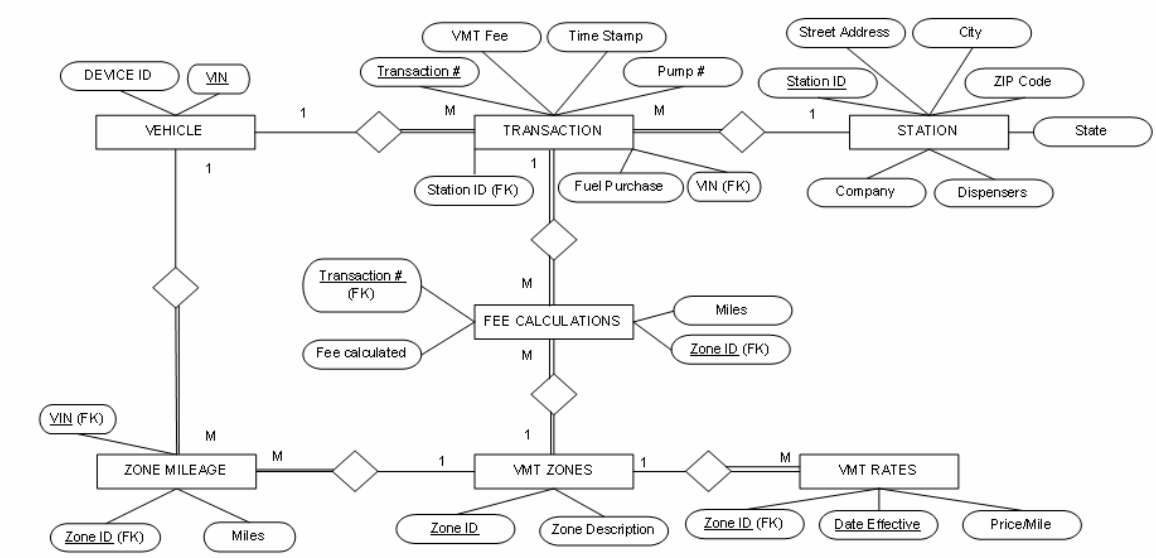

Figure 2. Entity Relationship Diagram for Central DB.

Along with the sample business rules explained above, another requirement for the VMT-based fee revenue system is that every transaction (which involves a vehicle with an on-vehicle device, gas station, and payment of the calculated VMT fee) be recorded in the database. This information is stored in the TRANSACTION table.

Similarly, each time a transaction occurs, the FEE_CALCULATIONS table is updated. This table stores the number of miles traveled in each zone and the respective VMT fee calculated for each of these zones based on this mileage.

One VMT Zone can be assigned many VMT rates over time. However, the latest rate is used to calculate the fee for a zone. This is done because the on-vehicle device is only capable of recording the number of miles per zone but not by zone *and* date. Therefore, the exact miles driven in a zone before and after a rate change cannot be determined. An example of how the appropriate VMT rate is selected is illustrated using sample data records from the VMT_RATES table:

Table 1. Sample Records from the VMT_RATES Table.

Zone ID	Date Effective	Price/Mile
1-1	01/01/2003	0.02
2	01/01/2003	0.03
3	01/01/2003	0.03
2	02/05/2003	0.04
1-1	10/10/2003	0.03
3	10/10/2003	0.05

If a vehicle initiates a transaction at a gas station on September 2, 2003 (09/02/2003) and the time stamp for the last transaction is April 28, 2003 (04/28/2003), then the following rates are used:

- 3 cents/mile for Zone 1-1
- 4 cents/mile for Zone 2
- 5 cents/mile for Zone 3

The ZONE_MILEAGE table stores the mileage accumulated in each zone for each vehicle. Each time a vehicle equipped with an on-vehicle device starts a transaction, the latest stored mileage information for each zone is obtained from this table, and the values are compared to the information obtained from the on-vehicle device. The differences in mileage between the two represent the miles driven in each zone since the last transaction recorded for that vehicle.

Finally, the VEHICLE table stores information (e.g., VIN and the identification number of the installed on-vehicle device) for all vehicles equipped with an on-vehicle device; and the STATION table stores information (e.g., company name, location, and number of dispensers on site where the VMT fee is collected) for all gas stations.

Physical Design

The logical design of the E-R diagram depicted in Figure 2, translates by one-to-one correspondence into the seven-table physical design depicted in Figure 3.

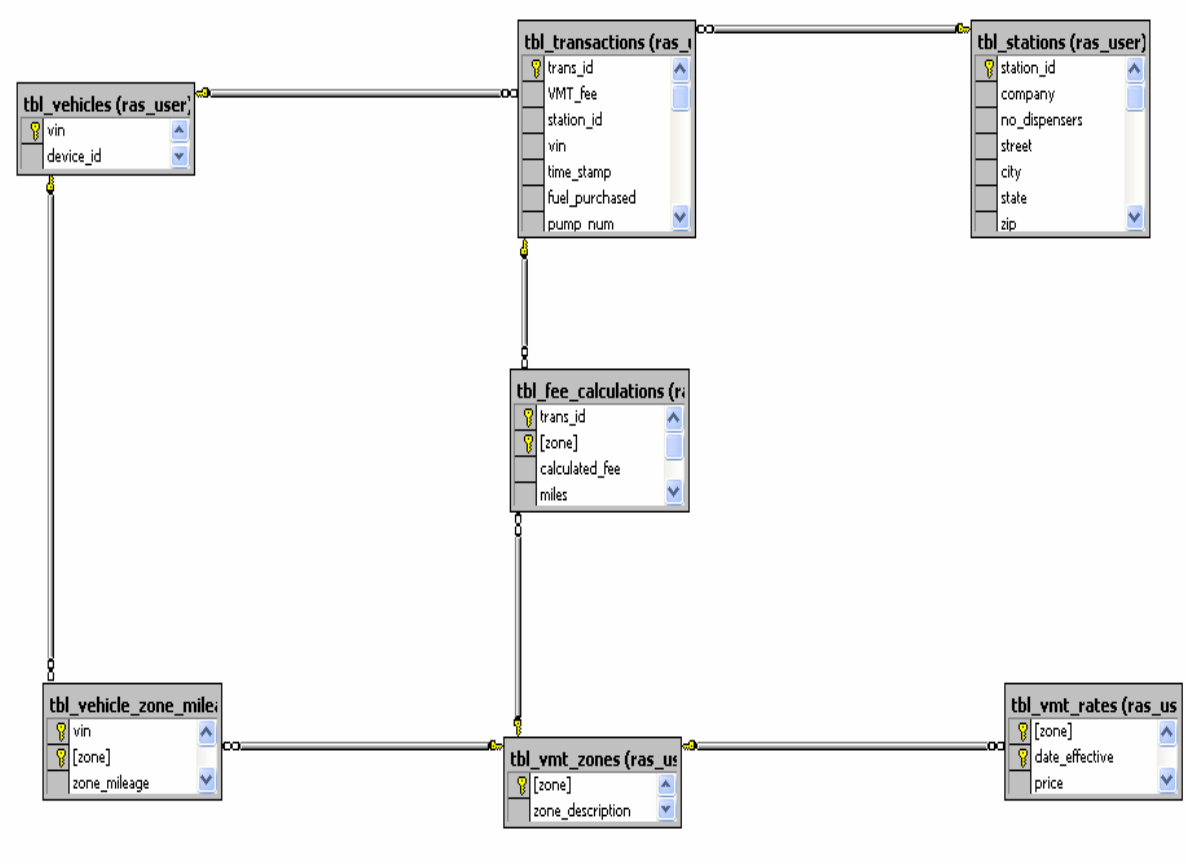

Figure 3. Physical Design of Central DB.

The physical structure of each table, including columns (attributes), characteristics and keys are further described below.

Table 2. Descriptions of Attributes in STATIONS Table.

STATIONS					
Key*	**Attribute**	**Data Type**	**Size**	**Null Values Permitted?**	**Attribute Definition**
PK	station_id	nvarchar	5	No	Gas station ID
	company	nvarchar	50	Yes	Company that owns the gas station
	no_dispensers	integer	4	Yes	Number of pumps at station
	street	nvarchar	100	Yes	
	city	nvarchar	100	Yes	
	state	chararacter	2	Yes	
	zip	nvarchar	25	Yes	

* PK=Primary key.

Table 3. Descriptions of Attributes in TRANSACTIONS Table.

TRANSACTIONS					
Key*	**Attribute**	**Data Type**	**Size**	**Null Values Permitted?**	**Attribute Definition**
PK	trans_id	nvarchar	100	No	Transaction number
	VMT_fee	money	8	Yes	Total VMTfee
FK	station_id	nvarchar	5	Yes	
FK	vin	nvarchar	50	Yes	Vehicle Identification Number
	time_stamp	smalldate	4	Yes	month/day/year time (i.e. 12/28/2003 2:03:00 PM)
	fuel_purchased	float	8	Yes	Amount of fuel purchased
	pump_num	char	2	Yes	Pump number

* PK=Primary key; FK=Foreign key.

Table 4. Descriptions of Attributes in VEHICLES Table.

VEHICLES					
Key*	**Attribute**	**Data Type**	**Size**	**Null Values Permitted?**	**Attribute Definition**
PK	vin	nvarchar	50	No	Vehicle Identification Number
	device_id	nvarchar	50	Yes	Device ID

* PK=Primary key.

Note: The VEHICLE_ZONE_MILEAGES table is a child of the VEHICLES table. Conceptually, it houses the same data that the on-vehicle device records.

Table 5. Descriptions of Attributes in VEHICLE_ZONE_MILEAGES Table.

VEHICLE_ZONE_MILEAGES					
Key*	**Attribute**	**Data Type**	**Size**	**Null Values Permitted?**	**Attribute Definition**
FK	vin	nvarchar	50	No	
FK	zone	nvarchar	5	No	
	zone_mileage	nvarchar	10	Yes	Accumulated mileage per zone

*FK=Foreign key.

Note: The FEE_CALCULATIONS table is a child of the TRANSACTION and VMT_ZONES tables

Table 6. Descriptions of Attributes in FEE_CALCULATIONS Table.

FEE_CALCULATIONS					
Key*	**Attribute**	**Data Type**	**Size**	**Null Values Permitted?**	**Attribute Definition**
FK	trans_id	nvarchar	100	No	Transaction ID
FK	zone	nvarchar	5	No	
	calculated_fee	money	8	Yes	
	miles	float	8	Yes	

*FK=Foreign key.

Table 7. Descriptions of Attributes in VMT_ZONES Table.

VMT_ZONES					
Key*	**Attribute**	**Data Type**	**Size**	**Null Values Permitted?**	**Attribute Definition**
PK	zone	nvarchar	5	No	VMT Zone ID
	zone_description	nvarchar	50	Yes	VMT Zone ID description

* PK=Primary key.

Table 8. Descriptions of Attributes in VMT_RATES Table.

VMT_RATES					
Key*	**Attribute**	**Data Type**	**Size**	**Null Values Permitted?**	**Attribute Definition**
FK Composite	zone	nvarchar	5	No	VMT Zone ID
PK Composite	date_effective	smalldate	4	No	Date effective of Price for the zone.
	price	money	8	Yes	

* PK=Primary key; FK=Foreign key.

Application Design

Input

To reflect the addition or deletion of zones and changes in rates, the database administrator should update the information stored in the VMT_RATES table.

Reports

The following reports can be generated from the data stored in the Central DB.

1. INDIVIDUAL VEHICLE TRANSACTION LOG report. This report provides a transaction log for each vehicle and on-vehicle device.

INDIVIDUAL VEHICLE TRANSACTION LOG
Time Stamp Service Station ID Pump # VIN Device ID Mileage Buckets Miles/Zone 1-1 Miles/Zone 1-2 Miles/Zone 1-3 Miles/Zone 2 Miles/Zone 3 Fee Calculated Fuel Purchased

2. COMPLIANCE AND FEE VERIFICATION REPORT. This report provides the ability to search for vehicles that meet specific criteria. For example, a check for vehicles that have been modified to avoid paying the VMT-based fee may begin with a search for vehicles with unusually low vehicle miles.

COMPLIANCE AND FEE VERIFICATION REPORT
Date 1 Date 2 (where Date 2 > Date 1) Criteria for search VIN Device ID

Graphical User Interface

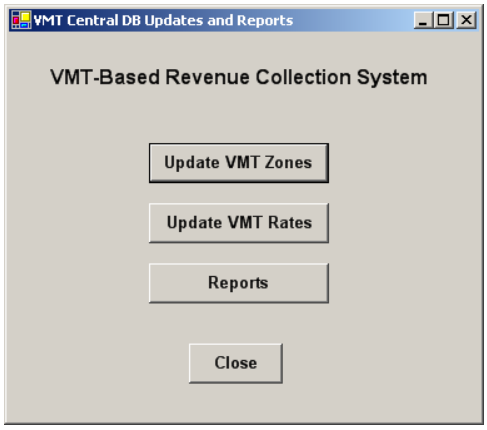

Figure 4. Main Screen Interface.

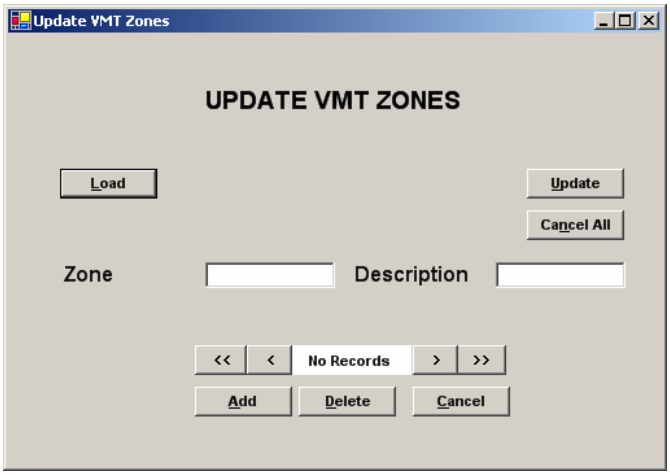

Figure 5. Update VMT Zones Interface.

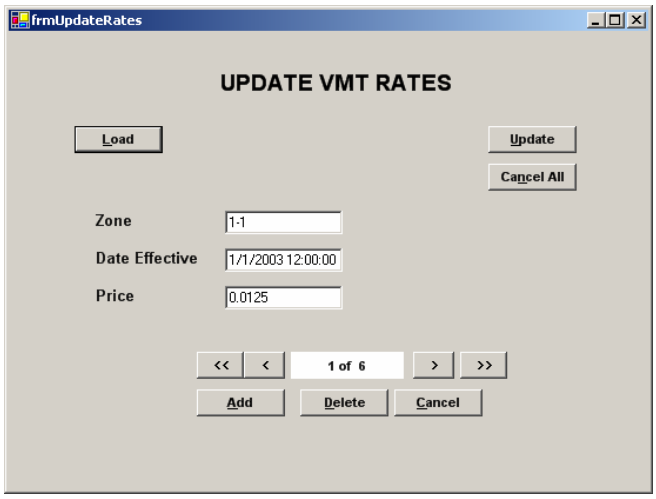

Figure 6. Update VMT Rates Interface.

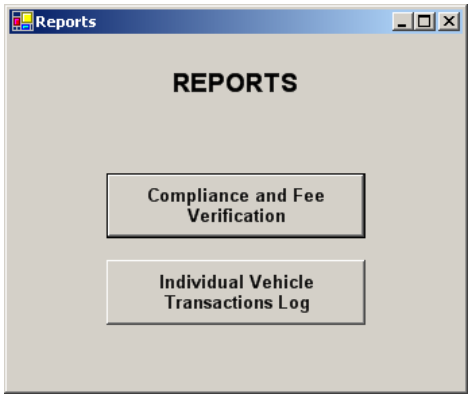

Figure 7. Reports Interface.

References

Kim, DS and JD Porter (2003). *Project Work Plan for Technology Development and System Integration for A Vehicle Miles Traveled Based Revenue Collection System Prototype*. Oregon State University.