2003

A Framework to Utilise Urban Bus Data for Advanced Data Analysis

Markus Hofmann
Technological University Dublin

Brendan Tierney
Technological University Dublin, brendan.tierney@tudublin.ie

Margaret M. O'Mahony
Trinity College Dublin, Ireland

A Framework to Utilise Urban Bus Data for Advanced Data Analysis

Markus Hofmann
Transport Study and Research Group
Department of Civil, Structural & Environmental Engineering
Trinity College Dublin, Ireland
Tel.: +353-1-6082537
Fax: +353-1-608-3072
E-mail: mhofmann@tcd.ie

Dr. Margaret M. O'Mahony
Transport Study and Research Group
Department of Civil, Structural & Environmental Engineering
Trinity College Dublin, Ireland
Tel.: +353-1-6082084
Fax: +353-1-608-3072
E-mail: margaret.omahony@tcd.ie

Brendan Tierney
School of Computing
Dublin Institute of Technology
Kevin Street, Dublin 2, Ireland
Tel.: +353-1-4024348
E-mail: brendan.tierney@comp.dit.ie

ABSTRACT

Most urban bus operators collect detailed data on their respective transportation networks using electronic fare collection systems. However, contrary to the opinion of other service industries that this data is a valuable resource, many bus operators have tended not to fully utilise these resources. International experience suggests using innovative technologies and methodologies such as data warehousing, Online Analytical Processing (OLAP), and data mining, to derive the maximum benefit from this data. Still bus operators tend not to keep the full range of data in a form, which is easy to access or utilise, and therefore, are not able to apply these technologies. The aim of the research project on which this paper reports is to describe the initial data structure of an electronic fare collection system (installed by a public transport operator in Ireland), the storage and enrichment of that data in a relational database, and finally, the representation of the public transport data in a data warehouse. This data warehouse forms the basis of all future data analysis. A 4-phase framework describes the import process leading to a relational database storing the transactional data. The paper concludes with the development of a data warehouse using the star schema.

Keywords: Relational Database Management Systems, Public Urban Transport, Data Warehouse

INTRODUCTION

Most urban bus operators collect detailed data on their respective transportation networks by using electronic fare collection systems. However, contrary to the opinion of other service industries that this data is a valuable resource, many bus operators have tended not to fully utilise these resources [4]. Like many other industries, the public transport industry can be considered data rich but information poor [15]. While international experience suggests using innovative technologies and methodologies such as data warehousing, Online Analytical

Processing (OLAP) and data mining, most bus operators tend not to keep the full range of data in a form, which is easy to access or utilise, and therefore, are not able to optimise the operational information from their data. Within the transport sector mainly the airline industry has implemented advanced analyses and data mining on a large scale [10] in order to improve their management decisions. These enabling technologies can make systems more powerful and facilitate decision makers to react more quickly, intelligently, and efficiently both horizontally and vertically [14] within an urban transport operator. The aim of the research project on which this paper reports is to describe the initial data structure coming from an electronic fare collection system, the storage and enrichment of that data in a relational database and finally, the representation of the public transport data in a data warehouse. This data warehouse can be seen as the foundation for most future data analysis [5, 6, 7, 12]. Future steps in this research project aim to use the data warehouse as foundation for complex data analysis algorithms such as pattern recognition analysis, passenger travel behaviour analysis and origin-destination analysis.

The research concludes with a multi-step framework that explains how to move from flat files that are produced by the electronic fare collection system to an innovative and well structured data warehouse. The partially encoded data from the Wayfarer (Supplier of the electronic fare collection system used by the public transport operator) system has been segmented and analysed so that the semantics of each data item is understood and imported correctly into the relational database. The relational database is shown on a conceptual level, which is then used as a foundation to develop a data warehouse schema facilitating a base for all future analysis applications.

BACKGROUND AND PROJECT DETAILS

The ever-growing need to share data and information throughout an organisation is becoming more and more important [5] to improve operations and decision-making. Accessing and sharing data with internal and external organisations as well as customers can increase productivity and efficiency [14]. Most organisations have to create routine reports with a high frequency that can be time consuming to create [4]. These generally simple reports can be generated without the need of using complex computing techniques or algorithms [14]. Relational databases and data warehouses build a platform for more complex technologies such as multi-dimensional analysis and data mining, which become more and more beneficial to transport related organisations.

The project database is based on data gathered from an Irish public transport operator on its transportation network. Wayfarer provides the electronic fare collection system that is responsible for the collection of this data, which forms the basis of this research project. The vast amount of transactional data from 1998 and 1999 (160 million records) has been moved from text files (one file per day) into a large relational Oracle database. Existing data has been enriched with additional datasets, which contributed considerably to the application and usability of the system. The data structure of the database has been further developed into a data warehouse, which provides a platform for statistical analysis, OLAP and data mining analyses. A sample of partially encoded records of the data files are displayed in Table 1 whereas Table 2 provides a brief description of the data attributes that are available for extraction.

Table 1: Raw Data File Records

```
810D0B0K010704000222 -> Control Record
82148505040000020000 -> Duty Record
8311 101310030105100 -> Journey Start Record
85006610030000000000 -> Stage Record
8B264100020000000000 -> Validation Record
85006710160000000000 -> Stage Record
85006810180000000000 -> Stage Record
0A7C0F5D000000000000 -> Boarding Record
85006910200000000000 -> Stage Record
0A7C0F62000000000000 -> Boarding Record
etc...               -> etc...
```

Table 2: Summary of all available data stored in the data files produced by Wayfarer

Record Identifier	Data Attribute	Description
81 – Control Header	Depot Code	Bus depot codes
81 – Control Header	Machine Code	Is a unique equipment code
82 – Duty Record	Day of the month	Day of the month
82 – Duty record	Month of the year	Month of the year
83 – Journey Start	Route Identifier	Identifies the route number
83 – Journey Start	Scheduled Time	Scheduled bus departure time
83 – Journey Start	Actual Time	Actual bus departure time
83 – Journey Start	Direction	Identifies the direction of the bus journey.
84 – Journey End	Stop Time	Time the bus journey ended
85 – Stage Update	Stage	Indicates the stages the bus has passed
85 – Stage Update	Time of stage update	Time the bus has reached the stage
8B – MCV Record	MCV ID	Unique number of the equipment installed
8B – MCV Record	Successful	Shows the number of successful validations
8B – MCV Record	Retries	Shows the number of retries
8B – MCV Record	Rejection	Shows the number of magnetic card rejections
0–Ticket Transaction	Ticket Type Code	Identifies the type of ticket used to validate
0–Ticket Transaction	ID Card Number	Unique number assigned to each magnetic card
0–Ticket Transaction	Previous Mode	Stores the mode previously used
0–Ticket Transaction	Previous Route	Stores the previous route taken
0–Ticket Transaction	Previous Stage	Stores the previous boarding stage

THE SYSTEM ARCHITECTURE

The rate and speed of accessibility of the data has been improved considerably by implementing a Relational Database Management System (RDBMS). An RDBMS is still the most effective archived data management tool [13] and is more efficient to update, extract and maintain data than storage systems that work with flat files [2]. The technology also allows more reliable and complex reporting as well as a more flexible form of data or result presentation [14].

"First generation" data archiving systems are often systems where data is simply stored in text files and usually remain untouched [15]. Querying and accessing data from these files can only be carried out by computer programmers due to the complexity and quantity of the data. Many data items have been encoded to either enforce security measures or to optimise storage space. In the case of the Wayfarer data the system encodes the records of each boarding in order to get more information into a 20-character string.

The structure of the earlier designed database is shown in Figure 1. The Entity Relationship Diagram (ERD) shows the tables, attributes, primary keys, relationships, optionality and cardinality of the database. The multi-step framework described in the following section will show how the data has been imported into this database.

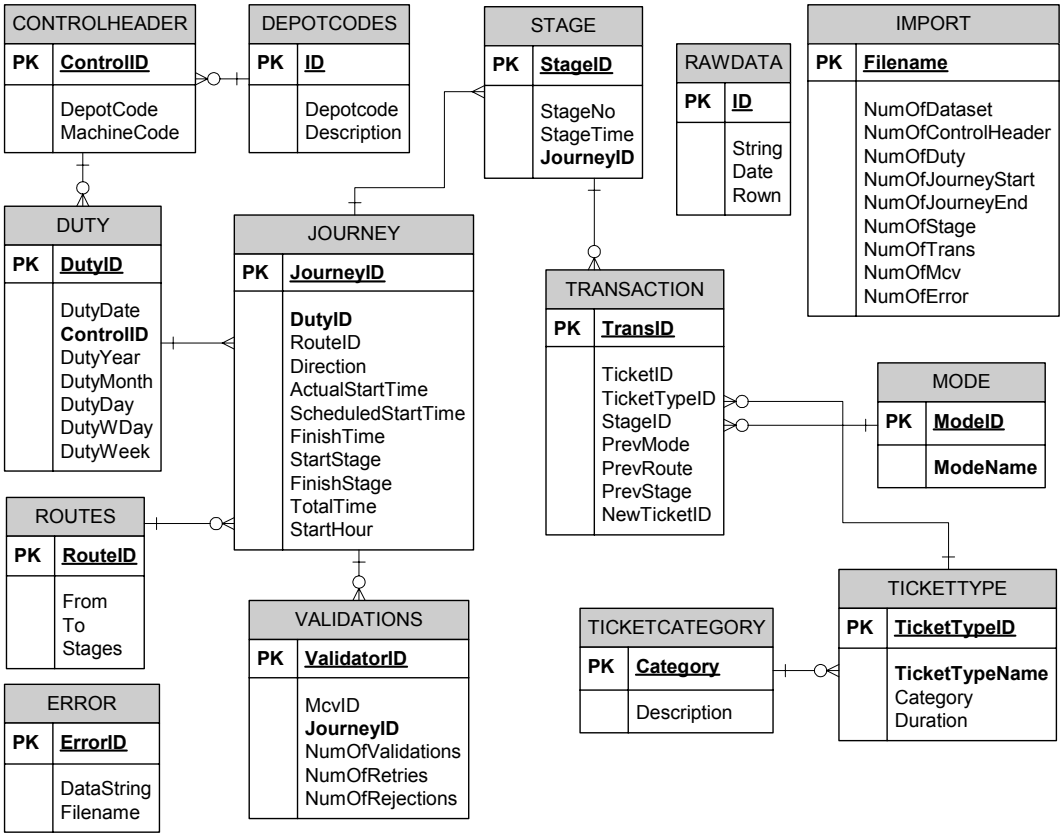

Figure 1: Entity Relationship Diagram (ERD) of the database

MULTI-STEP IMPORT FRAMEWORK

The multi-step import framework describes a 4-phase procedure (see Figure 2) to transfer data stored in flat files into the relational database shown in Figure 1. The primary purpose of the final version of the database is to create a platform from where information can be generated from data. This information can then be used by decision makers and can be called 'actionable information' [14]. The importing procedure has been fragmented into a 4-phase framework. Each of the four phases uses different technologies and/or tools to produce the deliverable for the next stage

Pre-formatting of the Wayfarer data files is necessary to retrieve the information of the record types and data attributes. The goal is to import the data into a relational database so that querying, extracting and further formatting can be carried out with ease. Since relational databases can be manipulated and accessed with Structured Query Language (SQL) and Procedural Structured Query Language (PL/SQL) it has been decided to format the data in such a way that the results can be uploaded directly into the predefined Oracle database allowing the restructuring of the data within the database. The following phases will be introduced:

- Phase 1 – Pre-formatting the raw data
- Phase 2 – Importing data with SQL Loader
- Phase 3 – Initial data model population with PL/SQL
- Phase 4 – Cleaning and extension process

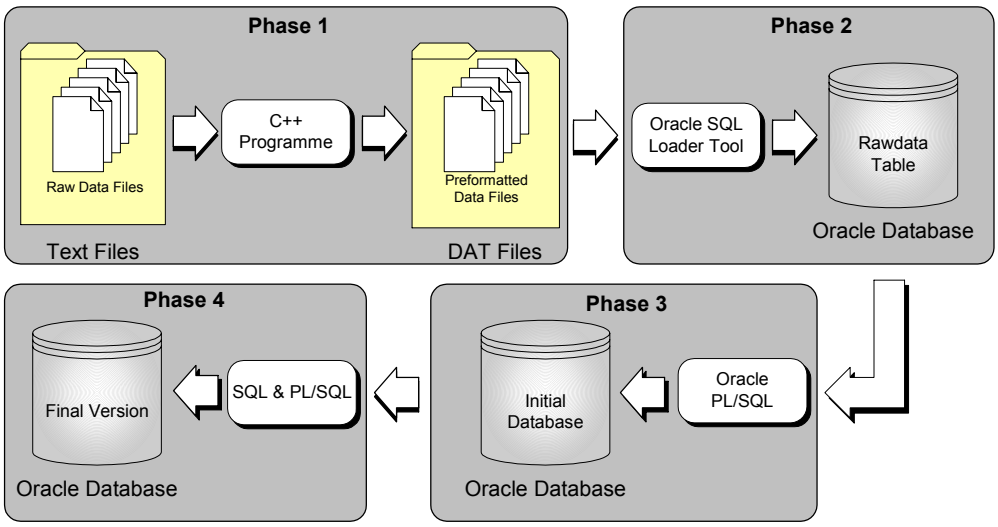

Figure 2: 4-Phase framework of importing Wayfarer data in an Oracle database

PHASE 1 – PRE-FORMATTING THE RAW DATA

The initial task has been defined as pre-formatting the data in such a way that it can be uploaded into one single table stored within the RDBMS. This table, called 'Rawdata' consists of the following four attributes including their data type in brackets:

- ID (Number, 10)
- String (Varchar, 128)
- Date (Date)
- Row Number (Number, 10)

Table 3: Rawdata format after first formatting step

```
1,810C0R0D000408025005,11301999,0,
1,82090430119900010000,11301999,1,
1,8366  07150645010523,11301999,2,
1,85003606450000000000,11301999,3,
1,8B024300040000000000,11301999,4,
1,0-029383-0643-0-0000-00,11301999,5,
2,0-001565-0671-0-0000-00,11301999,6,
3,0-001130-0691-0-0000-00,11301999,7,
4,0-010068-0665-0-0000-00,11301999,8,
5,0-007857-0671-0-0000-00,11301999,9,
6,0-000248-0457-0-0000-00,11301999,10,
7,0-004089-0457-0-0000-00,11301999,11,
8,0-010560-0691-0-0000-00,11301999,12,
9,0-001119-0691-0-0000-00,11301999,13,
10,0-011057-0671-0-0000-00,11301999,14,
```

Table 3 shows the formatting that has been done by the C++ programme. The changes become clearer when comparing Table 3 with Table 1 where the data and each of its 20-character strings is in absolute raw format. The C++ programme made the following changes:

- A new file has been created with a changed naming convention and file format.

- Changes in the 20-character data string are initially carried out in order to identify and rectify errors in the data. Data quality is very important due to the dependency of future analyses. It is therefore important to run quality checks to ensure data quality and integrity. The program identified 3807 bytes as invalid out of a total of 3,187,947,560 bytes. This is equivalent to less than 1.19479E-8 % and can therefore be neglected for all future analyses.
- The format of the 20-character strings beginning with '0' indicates passenger boardings. The structure and semantics of the transaction string is different compared to the other records as the data in the string is encrypted. A decoding procedure was developed due to overlapping bits into previous or next bytes. A technique called bit shifting had to be applied to the data so that the correct information could be extracted.
- The C++ programme further introduced a unique sequential number for each record type.
- The third part of the pre-formatted string shows the date of the record taken from the file name (see Table 3). The format of the date is 'Month – Day – Year' generally known as 'Short Date'.
- The final section is a unique sequential number starting from '0' increasing by '1' for each record to provide a unique identifier for each record.
- At the end of each processed file a summary record is added listing various parameters (such as total error rate, total number of passenger boardings, etc.) that later will be imported into the import table.

The C++ programme carried out these steps in order to pre-format the data in such a way that it complied with the requirements set by the pre-defined database structure and an Oracle tool called SQL Loader, which will be described, in greater detail in phase 2 of the framework. Various error detecting parameters have been introduced in order to determine the quality of the data and to detect any abnormalities.

PHASE 2 – IMPORTING DATA WITH SQL LOADER

The aim of this phase is to import the pre-formatted data produced in phase 1 into a pre-defined database table. SQL Loader version 9.2.0.1.0 is the Oracle utilities used to upload data from flat files (.dat files) into a predefined database structure [11]. The tool transferred nearly 20,000 records per second from the text files into the database table, which stored 160 million records from the years 1998 and 1999.

Figure 3 shows the process summary. The SQL Loader tool requires the actual files that store the data, a control file, which contains the parameters, and a pre-defined Oracle database structure which was in this case the table RAWDATA (see Figure 1 and Table 4). The tool then loads the data into the database table, creates a log file, which contains information about the importing or loading process, and a 'bad' file, which contains information about errors, or failures that occurred throughout the loading process.

Table 4: Sample records of database table structure

ID	String	Date	Rownumber
26291	85006911540000000000	01/12/1999	50004
26292	85007011540000000000	01/12/1999	50005
26293	85007111550000000000	01/12/1999	50006
26294	85007211570000000000	01/12/1999	50007
2139	84120700000000000000	01/12/1999	50008

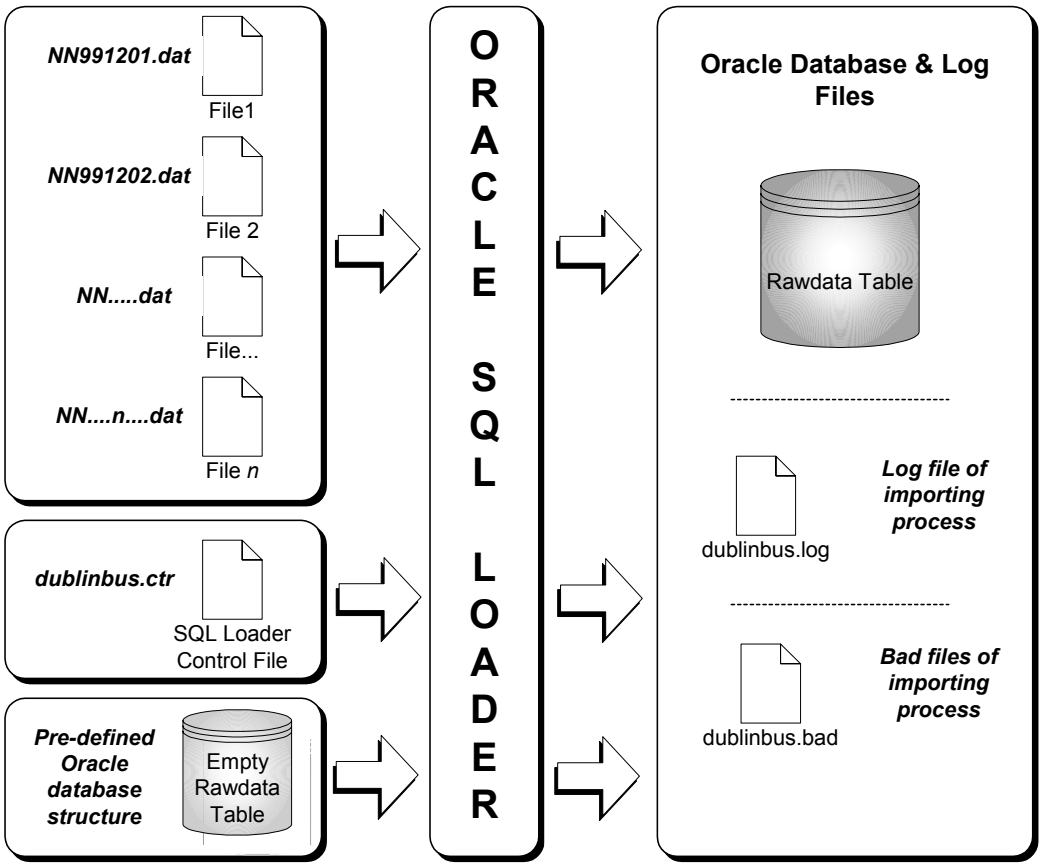

Figure 3: SQL Loader process summary

PHASE 3 – INITIAL DATA MODEL POPULATION WITH PL/SQL

Phase 3 is responsible for restructuring the database and transferring the records into the pre-defined database structure. At this stage the data is only stored in one table called RAWDATA and it was necessary to transfer all records into a relational structure. The PL/SQL code that restructured the data works as follows:

1. Declare all variables that will be used throughout the programme
2. Select the maximum number of records in the table rawdata in order to know how often the loop has to be repeated
3. Initiate the loop
4. Information of the first record is distributed into variables
5. The first 2 characters determine the record type
6. Depending on the record type different PL/SQL statements will be executed after assigning the data Parameters to various variables.
7. The executed SQL statements are committed to the database
8. The loop is repeated until all records have been processed

After finalising phase 3 of the 4-phase procedure all the data is stored in the appropriate tables and are at least in second normal form.

PHASE 4 – CLEANING AND EXTENSION PROCESS

The purpose of this stage is to clean the data and to extend tables by attributes that have been considered as useful for future analysis. This last phase will also ensure that the database is in

third normal form and complies with all relational database rules. This is mainly necessary to improve performance and results of future analyses [3]. The database will however still change over time through changes and extensions of attributes or tables.
The following cleanup or extensions have been introduced:

- Various extensions of date fields such as one field only storing the day, the month, or the year. Further attributes such as number of week or weekday also aim to contribute to the simplicity of querying the database.
- Some of the tables and attributes had to be cleaned up or put into normalised format.
- The total journey time and the start hour of each journey have been calculated.

MOVING TOWARDS A DATA WAREHOUSE

A data warehouse is a time varying and integrated database, which is primarily used for decision support on a management level [1, 5, 6]. It can therefore be seen as an efficient decision support tool for public transport analyses. Due to the volume of records and attribute redundancy the size of data warehouses is generally in the gigabyte and often even in the terabyte region. Structuring and modelling data in that order of magnitude is a complex but crucial venture. From a data modelling point of view a data warehouse is generally not in 3rd normal form and completely denormalised which means that redundancy is increased. This, however, improves the efficiency of querying the dataset [12].

A case study based on a system used in Hampton Roads, Virginia, USA was carried out where query execution speeds were measured [15]. A freeway management database system was tested against a data warehouse, both representing the same data. Ten test cases were completed with the result that the mean query time of the transactional database was 272 seconds whereas the data warehouse design delivered a mean query time of 49 seconds [15]. Considering that the data warehouse has to execute many queries to produce reports or to deliver measures for data mining applications [5, 7], this significant decrease in query time improves analysis efficiency [15]. The conclusion of this study was that transport professionals should be using data warehouses when a complex data analysis is required.

The data modelling and data structure of a data warehouse differs from the one in relational databases. A data warehouse consists of a fact table and a number of surrounding dimension tables [6, 9], which contribute data to the corresponding fact table. The fact table contains the foreign key attributes that build the relation to the adjoining dimension tables. The fact table in the example demonstrated in this paper could be the actual passenger boarding which would be extended by dimension tables such as route-stage dimension, time dimension or customer dimension.

The following section designs and develops the data warehouse for the application described.

SNOWFLAKE SCHEMA VS. STAR SCHEMA

Two main data warehouse schemas exist: star schema and snowflake schema. A star schema has a single fact table with surrounding dimension tables, which form a type of star [5, 6]. The snowflake schema is an extended modelling approach of the star schema and allows dimension tables to facilitate sub-dimensional tables [5, 6].

The Oracle Corporation recommends choosing a star schema over a snowflake schema unless there is a clear reason not to [8]. In the case of the urban public transport sector it is believed that star schemas are more suitable for implementation than snowflake schemas due to simplicity and reduced relationship joins. The main limitation is the available budget, which is generally assigned to the IT departments within such transport organisations.

Because data warehouses are not business critical the necessity of such information systems is more difficult to justify. Nevertheless, there has to be a return on investment. This point can be reached more easily when the project complexity is minimised due to the consequential reduced cost. Star schemas contribute to this due to their simplicity in design and maintenance.

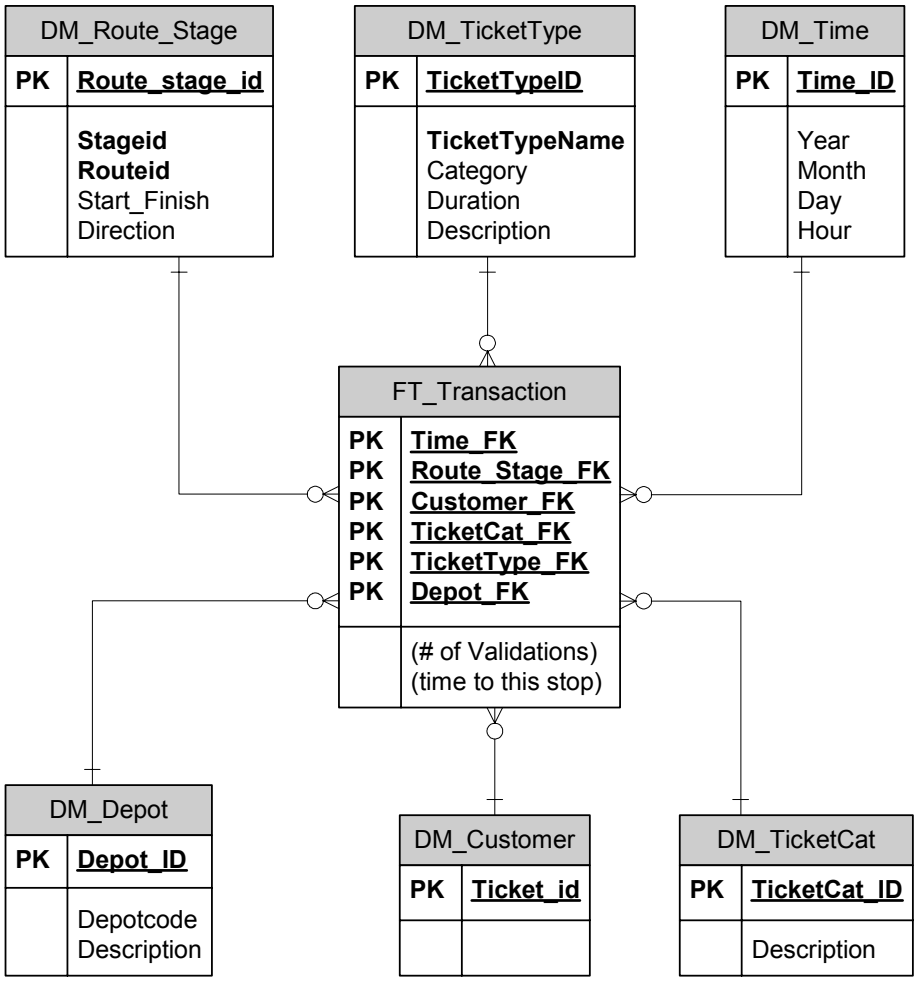

Figure 4: Star schema of the data warehouse of an urban transport operator

This paper provides an example of a star schema approach that has been generated from the public transport operator's database (see Figure 4). The FT_Transaction table is the fact table building the centre of the star schema and is surrounded by the dimension tables DM_Route_Stage, DM_TicketType, DM_Time, DM_Depot, DM_Customer and DM_TicketCat. Sub-dimensional tables as they occur in snowflake schemas generally slow down the query procedure [9] and have therefore been avoided. The simple structure of the star schema has made it one of the most used data warehouse schemas [6]. The schema possesses the following advantages:

- Easy to understand and intuitive design [9]
- Attributes can easily be added to fact tables [9]
- Additional dimension tables can be added without interfering with existing computer applications [9]
- Reduces number of physical joins [8]

- Simplifies the view of the data model [8]
- Allows relative easy maintenance [8]

CONCLUSION

The paper reported on the data import process of transactional data, which has been produced by an electronic fare collection system used by a public transport operator in Ireland. The original files (one per day) have been imported using a 4-phase framework (see Figure 2). The first phase was concerned with the initial cleaning, decoding and pre-formatting of text files. Phase 2 used a tool called SQL Loader to upload the 160 million records into a single table with four attributes stored in an Oracle RDBMS. The third phase transferred the records into a predefined database structure (see Figure 1) using PL/SQL. The data was structured in third normal form according to the rules of the relational model. The final phase four focused on cleaning the database tables and populating additional tables that are used to describe some attributes (e.g. route information). This task was mainly implemented by using SQL and PL/SQL. The result of this 4-phase framework is a relational database storing electronic fare collection data over two years.

The change from a transactional database as described above to a data warehouse can be crucial when the main aim of keeping the data is focused on analysis. A data warehouse optimises the efficiency of executing analysis related algorithms. An introduction to data warehouse schemas leads into the discussion of the most appropriate schema for urban transport operators. The star schema has been identified as more suitable for low budget data warehouse projects. The ERD shown in Figure 1 has been re-structured into a data warehouse model (Figure 4) following the rules of the star model.

ACKNOWLEDGEMENT

The research is supported by the Department of Transport and Higher Education Authority of Ireland under the Transport Research Programme.

REFERENCES

[1] Chaudhuri, S., Dayal, U. (1997) 'An overview of data warehousing and OLAP technology' ACM SIGMOD Record 26, 65-74.

[2] Date, C. J. (2000) *An Introduction to Database Systems* (7th edn), Addison Wesley Longman, Inc., USA.

[3] Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) *'Knowledge Discovery and Data Mining: Towards a Unifying Framework'*, In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, USA.

[4] Furth, P. G. (2000) *'Data Analysis for Bus Planning and Monitoring'*, TCRP Synthesis 34, Transportation Research Board, National Research Council, Washington, D.C., USA.

[5] Inmon, W. H. (2002) Building the Data Warehouse, 3rd Edition, John Wiley & Sons, Chichester, USA.

[6] Kimball, R., Reeves, L., Ross, M., Thornthwaite, W. (1998) The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses, John Wiley & Sons, Chichester, USA.

[7] Klösgen, W., Zytkow, J. M. (2002) *'Handbook of Data Mining and Knowledge Discovery'*, Knowledge Discovery in Databases: The Purpose, Necessity, and Challenges, Oxford University Press, New York, USA.

[8] Lane, P. (2002) Oracle 9i Data Warehousing Guide, Oracle Corporation, Part No. A96520-01.

[9] Levene, M., Loizou, G. (2002) 'Why is the snowflake schema a good data warehouse design?', Journal of Information Systems, Vol. 28, pp.225-240.

[10] Pritscher, F., Feyen, H. (2001) 'Data Mining and Strategic Marketing in the Airline Industry', Conference Proceedings of the ECML/PKDD-01 Workshop, Freiburg, Germany.

[11] Rich, K. (2002) Oracle 9i – Database Utilities, Oracle Corporation, Release 2 (9.2), Part No. A96652-01.

[12] Turban, E., Aronson, J. (2001), Decision Support Systems and Intelligent Systems, Prentice Hall, 6th Edition, New Jersey, USA.

[13] Turner, S. (2002) 'A Simple Approach to Archiving Operations Data: Case Study in Austin, Texas', Conference Proceedings of 81st Annual Meeting, Transportation Research Board, Washington, USA.

[14] Yoder, S., DeLaurentiis, J. and Bacigalupo, R. (2002) 'Framework of Regional Transit Asset Management System', Conference Proceedings of 81st Annual Meeting, Transportation Research Board, Washington, USA.

[15] Smith, B., Lewis, D., Hammond, R. (2003) 'Design of Archival Traffic Databases: A Qualitative Investigation into the Application of Advanced Data Modelling Concepts', Conference Proceedings of 82nd Annual Meeting, Transportation Research Board, Washington, USA.