

**AN ALTERNATIVE TO AN OPERATIONAL SYSTEM  
AND DATA WAREHOUSE SYSTEMS:  
A CONCEPTUAL MODEL**

**Phillip J v Rensburg**

**B.Sc. Honours**

Dissertation submitted in partial fulfillment of the requirements for the degree Magister Scientiae in the School of Modelling Sciences at the University of North West.  
[University of Potchefstroom]

**Supervisor:** Prof. D B Jordaan

Vanderbijlpark  
2003

**NOORDWES UNIVERSITEIT  
VAALDRIEHOEKKAMPUS**

## **Preface**

I wish to dedicate this study to my Heavenly Father who granted me the necessary perception, perseverance, dedication and motivation to complete this task.

I would also like to thank the following people for their contribution to this dissertation:

- My parents and sister for their willingness to help where possible and their support when it was needed.
- My facilitator, Prof. DB. Jordaan, for his input advice and patience.
- Mr. B. Record and Mrs. M. Barnard for assistance in editing (language).
- Staff of the Ferdinand Postma Library for the requesting of publications.

## Table of Contents

<b>TABLE OF CONTENTS .....</b>	<b>3</b>
<b>TABLE OF FIGURES .....</b>	<b>6</b>
<b>GLOSSARY .....</b>	<b>7</b>
<b>AFRIKAANS OPSOMMING VAN DIE STUDIE .....</b>	<b>15</b>
<b>CHAPTER 1: INTRODUCTION .....</b>	<b>17</b>
1.1 DATA .....	17
1.2 INFORMATION .....	17
1.3 DATA AND INFORMATION STORAGE.....	18
1.4 MOTIVATION FOR THIS RESEARCH .....	18
1.4.1 <i>Operational Systems</i> .....	18
1.4.2 <i>Data Warehouse</i> .....	20
1.5 PROBLEM STATEMENT .....	21
1.6 MAIN OBJECTIVE OF THE RESEARCH.....	21
1.7 SIGNIFICANCE OF THE STUDY.....	21
1.8 RESEARCH METHOD.....	22
1.9 BOUNDARIES OF THE STUDY .....	22
<b>CHAPTER 2: OPERATIONAL SYSTEMS .....</b>	<b>23</b>
2.1 CHAPTER OBJECTIVES.....	23
2.2 OPERATIONAL SYSTEM CLASSIFICATION .....	23
2.2.1 <i>Legacy Systems</i> .....	23
2.2.2 <i>OLTP Systems</i> .....	24
2.3 ENTERPRISE RESOURCE PLANNING SYSTEM (ERP).....	25
2.3.1 <i>The Evolution of Information Systems</i> .....	25
2.3.2 <i>The Evolution of ERP Systems</i> .....	27
2.4 FORMAL DEFINITION OF AN ERP SYSTEM .....	28
2.5 VENDORS .....	30
2.5.1 <i>SAP</i> .....	30
2.5.2 <i>Oracle</i> .....	30
2.5.3 <i>JD Edwards</i> .....	31
2.5.4 <i>PeopleSoft</i> .....	31
2.5.5 <i>BAAN</i> .....	32
2.6 MODULES.....	32
2.7 ARCHITECTURE .....	33
2.7.1 <i>The Database Layer</i> .....	33
2.7.2 <i>The Application Layer</i> .....	34
2.7.3 <i>The Presentation Layer</i> .....	34
2.8 HOW ERP WORKS .....	35
2.8.1 <i>Ordering</i> .....	35
2.8.2 <i>Availability</i> .....	36
2.8.3 <i>Production</i> .....	36
2.8.4 <i>Manpower</i> .....	36
2.8.5 <i>Purchasing</i> .....	36
2.8.6 <i>Order Tracking and More Ordering</i> .....	36

2.9	RETURN OF INVESTMENT .....	37
2.10	DESIGN.....	38
2.10.1	Entity Relationship Modelling.....	38
2.10.2	Normalization.....	39
2.11	IMPLEMENTATION PHASES.....	40
2.11.1	Organisational and Conceptual Design.....	40
2.11.2	Detailed Design and Implementation.....	41
2.11.3	Preparations for Production .....	41
2.11.4	Productive Operation.....	41
2.12	STRENGTHS AND WEAKNESSES.....	41
2.12.1	Strengths.....	41
2.12.2	Weaknesses.....	42
2.13	CHAPTER CONCLUSION.....	43
<b>CHAPTER 3: THE DATA WAREHOUSE.....</b>		<b>44</b>
3.1	CHAPTER OBJECTIVES.....	44
3.2	BACKGROUND.....	44
3.3	DECISION SUPPORT SYSTEMS (DSS).....	46
3.4	THE DATA WAREHOUSE USAGE .....	46
3.5	DATA WAREHOUSE USERS.....	47
3.5.1	The Operational User.....	47
3.5.2	The Power User.....	47
3.5.3	The Executive User.....	47
3.6	DATA COVERAGE.....	48
3.7	DATA INTEGRATION.....	49
3.8	DATA QUALITY .....	49
3.9	THE BASIC ELEMENTS OF THE DATA WAREHOUSE .....	50
3.9.1	Source System.....	50
3.9.2	Data Staging Area.....	50
3.9.3	Presentation Server .....	51
3.9.4	Dimensional Model .....	51
3.9.5	Data Mart.....	51
3.9.6	Online Analytic Processing (OLAP).....	52
3.9.7	Relational Online Analytic Processing (ROLAP).....	52
3.9.8	Multi Dimensional Online Analytic Processing (MOLAP) .....	52
3.9.9	End User Application .....	52
3.9.10	End User Data Access Tool.....	52
3.10	EXTRACT TRANSFORM LOAD (ETL).....	53
3.10.1	Extract.....	54
3.10.2	Transform.....	54
3.10.3	Load.....	55
3.11	DATA WAREHOUSE SCOPE.....	55
3.12	DEVELOPMENT METHODOLOGY.....	55
3.13	DATA WAREHOUSE ECONOMIC JUSTIFICATION .....	56
3.14	DATA WAREHOUSE DESIGN.....	57
3.14.1	Star Schema Database Design .....	57
3.14.2	The Benefits of a Star Schema.....	57
3.14.3	Limitations of the Star Schema.....	58
3.14.4	Fact and Dimension Tables.....	58
3.14.5	De-normalization.....	59
3.15	CHARACTERISTICS .....	60
3.16	CHAPTER CONCLUSION.....	62
<b>CHAPTER 4: COMPARISON BETWEEN OPERATIONAL SYSTEMS AND DATA WAREHOUSES .....</b>		<b>63</b>

4.1	CHAPTER OBJECTIVES.....	63
4.2	OPERATIONAL VS ANALYTICAL NEEDS.....	63
4.3	OPERATIONAL AND ANALYTICAL DATA.....	63
4.3.1	<i>Data in the Operational System</i> .....	64
4.3.2	<i>Data in the Data Warehouse</i> .....	65
4.4	UPDATE REGULARITY.....	65
4.5	AVAILABILITY.....	65
4.6	DATA FLEXIBILITY.....	66
4.7	DESIGN OBJECTIVES.....	66
4.8	REQUIREMENTS.....	66
4.9	TUNING.....	67
4.10	CHAPTER CONCLUSION.....	68
<b>CHAPTER 5: THE CONCEPTUAL MODEL.....</b>		<b>69</b>
5.1	CHAPTER OBJECTIVES.....	69
5.2	EXAMPLES OF OPERATIONAL AND DATA WAREHOUSE SYSTEMS.....	69
5.2.1	<i>Operational System</i> .....	69
5.2.2	<i>Data Warehouse</i> .....	73
5.3	AN ALTERNATIVE TO OPERATIONAL AND DATA WAREHOUSE SYSTEMS.....	77
5.3.1	<i>Operational vs Analytical Needs</i> .....	78
5.3.2	<i>Update Regularity</i> .....	79
5.3.3	<i>Availability</i> .....	79
5.3.4	<i>Data Flexibility</i> .....	79
5.3.5	<i>Design Objectives</i> .....	80
5.3.6	<i>Tuning</i> .....	80
5.3.7	<i>Normalization</i> .....	80
5.3.8	<i>Data History</i> .....	81
5.4	THE CONCEPTUAL MODEL: AN EXAMPLE.....	81
<b>CHAPTER 6: SUMMARY AND CONCLUSION.....</b>		<b>85</b>
6.1	CHAPTER OBJECTIVES.....	85
6.2	ADVANTAGES OF THE CONCEPTUAL MODEL.....	85
6.3	DISADVANTAGES OF THE CONCEPTUAL MODEL.....	85
6.4	LIMITATIONS AND FUTURE RESEARCH.....	86
6.4.1	<i>Limitations</i> .....	86
6.4.2	<i>Future Research</i> .....	87
<b>ANNEXURES.....</b>		<b>88</b>
7.1	VARIETIES OF STAR SCHEMAS.....	88
7.1.1	<i>Simple Star Schema</i> .....	88
7.1.2	<i>Multiple Fact Tables</i> .....	89
7.1.3	<i>Outboard Tables</i> .....	90
7.1.4	<i>Multi-Star Schema</i> .....	92
<b>BIBLIOGRAPHY.....</b>		<b>93</b>

## Table of Figures

<b>CHAPTER 2: OPERATIONAL SYSTEMS .....</b>	<b>23</b>
FIGURE 2.1: OPERATIONAL SYSTEMS CLASSIFICATION.....	24
FIGURE 2.2: ERP MARKET SHARE (INFORMATIONWEEK.COM SEPT 1997) .....	29
FIGURE 2.3: LAYERED ARCHITECTURE OF AN ERP SYSTEM.....	35
<b>CHAPTER 3: THE DATA WAREHOUSE.....</b>	<b>44</b>
FIGURE 3.1: THE BASIC ELEMENTS OF THE DATA WAREHOUSE.....	53
FIGURE 3.2: AN EXAMPLE OF A NORMALIZED AND DE-NORMALIZED CUSTOMER RECORD.....	60
<b>CHAPTER 5: THE CONCEPTUAL MODEL.....</b>	<b>69</b>
FIGURE 5.1: AN EXAMPLE OF AN OPERATIONAL SYSTEM.....	70
FIGURE 5.2: AN EXAMPLE OF A DATA WAREHOUSE SYSTEM .....	75
FIGURE 5.3: SALES FACT TABLE .....	76
FIGURE 5.4: AN ALTERNATIVE DATA WAREHOUSE DESIGN.....	77
FIGURE 5.5: THE CONCEPTUAL MODEL .....	82
<b>ANNEXURES .....</b>	<b>88</b>
FIGURE 7.1: A SIMPLE STAR SCHEMA.....	89
FIGURE 7.2: A STAR SCHEMA WITH TWO FACT TABLES .....	89
FIGURE 7.3: A FACT TABLE AS AN ASSOCIATIVE TABLE.....	90
FIGURE 7.4: AN EXAMPLE OF SECONDARY DIMENSION TABLES .....	91
FIGURE 7.5: AN EXAMPLE OF A MULTI-STAR SCHEMA .....	92

## **Glossary**

**Analogs** – Analogs are data elements that have “equivalent” meaning. Analogous data elements often appear to be synonyms but have different shades of differences, which are significant to the business understanding of the data.

**Analytical Processing** – Analytical processing is that processing done to support strategic and management decision-making. Data used in analytical processing is often historical in nature, permitting users to analyse trends and patterns with a large amount of data over wide ranges of time. Analytical processing systems are customarily read-only and do not permit the data to be updated by users as do operational systems.

**Architecture** – An architecture is a set of rules or structures providing the framework for the overall design of the system or product. There are networking architectures, client-server architectures, as well as many others. The data architecture provides this framework by identifying how the data will move throughout the system and how it will be used within the corporation.

**Business Intelligence (BI)** – Originally a term based on data mining but now generalized to encompass all of decision support systems, business intelligence is intended to suggest the connotation of “intelligence”.

**Cartesian Product** – If two tables in a join query have no join condition, Oracle returns their Cartesian product. Oracle combines each row of one table with each row of the other. A Cartesian product always generates many rows and is rarely useful. For example, the Cartesian product of two tables, each with 100 rows, has 10,000 rows.

Always include a join condition unless you specifically need a Cartesian product. If a query joins three or more tables and you do not specify a join condition for a specific pair, the optimizer may choose a join order that avoids producing an intermediate Cartesian product.

**Cube** – An intersection of multiple dimensions, for example, customer-product-time, that determines the key of data warehouse facts. In general cubes are contrasted with the star schema in comparing the methods and results of OLAP engines with relational databases.

**Data Integration** – Within the data warehouse environment, data integration is the process by which the source data's characteristics are changed prior to loading into the data warehouse. Typically, data integration is done when data is extracted from the operational systems and may encompass integrating dissimilar data types, changing codes, and reconciling data definitions. A data analyst generally does data integration.

**Data Mart** – A data mart, often referred to as a subject-oriented data warehouse, represents a subset of the data warehouse comprised of relevant data for a particular business function (e.g. marketing, sales, finance). Similar to the data warehouse, data marts may contain data stored at various levels of granularity, depending on the end-user functions and business requirements. The data warehouse can be build incrementally, by starting off with one data mart and then adding on to it.

**Data Mining** – Methods of directed and undirected knowledge discovery, relying on statistical algorithms neural networks, and optimization research to discover, verify and apply patterns in data to understanding and managing the behavior of customers, products, services, market dynamics and other critical business transactions.

**Data Transformation** – Data transformation is the process by which data in the data warehouse is turned into information, which can be accessed by the end-user community. Data transformation uses the data structures and data content of the data warehouse and transforms these into usable, value added information by allowing the data to be formatted, summarized, and/or viewed in specific ways.

**Data Warehousing** – An architecture that aims to align business imperatives, such as customer knowledge and brand development, with system structures representing the business - usually such constructs such as dimensions, facts and aggregates - addressing decision support questions and issues about customers, products, services and markets and making possible applications that transform dump data into useful business knowledge.

**Decision Support System (DSS)** – A system to assist business professionals – brand managers, marketing specialists and other knowledge workers in making choices about the allocation of resources in promotions, advertising, pricing and planning.

**Dimension** – A data modelling structure exemplified by such entities as customer, product, time period channel, etc. In the context of data modelling, one of the entities constituting the periphery of star schema; those entities that intersect to define the logical position of a fact table.

**Dimensional Data Modelling** – A specific discipline for modelling data that is an alternative to entity relationship (E/R) modelling. A dimensional model contains the same information as an E/R model but packages the data in a symmetric format whose design goals are user understandability, query performance, and resilience to change.

**Enterprise Resource Planning (ERP)** – Software packages designed to run the enterprise from front to end. These are transaction, not data warehousing systems, whose data structures are highly snow-flaked, normalized and optimized for high volume update activities

**Environment for data access** – This includes the front-end data-access tools and technologies allowing users to easily access the data, the training that must take place for users to use these tools, the implementation of meta-data, and the training to navigate through the meta-data. An improper environment, or ill-conceived architecture, for data access will generally cause the data warehouse project to fail.

**Executive Information System (EIS)** – When managers with enough rank realized they had the influence needed to command an intuitive, easy-to-use interface, decision support systems were transformed into executive information systems. Then emphasis is on a simplified executive interface suitable for mouse or touch screen.

**Fact** – In the context of data warehousing, a continuous additive quantity as defined by the intersection of data dimensions. The main body – as opposed to the points – of the star is the star schema. The logical place of intersection of the dimensions - for example, the composite key customer, product, place, and time defines a variety of possible fact structures such as amount of sale or delivery quantity.

**Functional Area** – A functional area is a business function that must be represented by the data warehouse, there are no fixed rules to define a functional area, it can be something like financials, production etc. or National customers, international customers

**Granularity** – Granularity is the level of detail within the data warehouse and it is one of the principal design issues in the data warehouse development. Highly granular data provides a great deal of detailed information and subsequently a large volume of data.

**Hierarchies** – Business hierarchies describe the organisational structure and the logical parent-child relationships in the data. The typical data warehouse will have many hierarchies throughout the dimensions with many levels of calculated or stored aggregate values across the dimensions and hierarchies.

**Homonyms** – Homonyms are data elements that have the same name but represent different business facts.

**Hybrid On-Line Analytical Processing (HOLAP)** – An approach to OLAP which attempts to combine a multidimensional database (MDDB) to build aggregates for desktop slicing and dicing with back-end standard, relational database.

**Legacy** – Can refer to either systems, applications, or data. Those old production systems, applications, or data on which the business depends.

**Meta-Data**– Data about data, indicating the source, use, value or function of data.

**Management Information Systems (MIS)** – MIS can be defined as an information system using formalized procedures to provide managers at all levels in all functions with appropriate information from all relevant sources to enable them to make timely and effective decisions.

**Multidimensional Database (MDDB)** – A database specializing in the aggregating caching, manipulating, and presenting of data cubes to and for OLAP “slicing and dicing” desktop functions.

**Multidimensional On-Line Analytical Processing (MOLAP)** – An approach to OLAP that employs a proprietary database, a multidimensional database.

**Operational Applications or Systems** – The applications that create, update and access operational data.

**Operational Data** – Production files or databases updated by on-line, and batch production programs.

**Operational Data Store (ODS)** – The ODS was meant to serve as the point of integration for operational systems. This was especially important for legacy systems that grew up independently of each other. Banks, for example, typically had several independent systems set up to support different products – loans, checking accounts, savings accounts. The advent of teller support computers and the ATM helped push many banks to create an ODS to integrate current balances and recent history from these separate accounts under one customer number. An ODS is truly an operational system.

**On-line Analytic Processing (OLAP)** – is primarily distinguished from OLTP as a decision support for transactional systems. OLAP applications are responsible for the elaborate development of “slice and dice” functions for data analysis on the desktop.

**On-line Transaction Processing (OLTP)** – The term that is applied to the operational systems in the enterprise that are updated on-line.

**Operational Processing** – Operational processing refers to systems that run the day-to-day business for companies. The emphasis of these systems is to support business functionality by processing transactions accurately and efficiently and is also often referred to as “mission critical” application processing. Common examples of this type of processing are order entry, manufacturing, and general ledger.

**Relational Database Management System (RDMS)** - The presentation of data as mappings of rows and columns in tables, able to be accessed and manipulated by a small set of functions or operations.

**Relational On-Line Analytic Processing (ROLAP)** – An approach to OLAP that emphasizes “reach through” to standard relational databases (not an MDDB) from the desktop on which analytic “slicing and dicing” is occurring.

**Snow-Flaked** – In the context of data warehousing, a data model that is highly normalized with many small entities, taken to fourth or fifth-normal form, such that when laid out visually, the model looks like a snowflake.

**Source and Target Data** – Source data is the data in the various operational databases, files, and segments that run the day-to-day business. The data will be extracted from the source systems and converted to the data in the warehouse. Source data can also come from outside the corporation. The target data is the data that goes into the fields within the data warehouse database.

**Structured Query Language (SQL)** – A database language standard that is used to retrieve, update and load data from databases.

**Star Schema** – A data model particularly applicable to data warehousing, in which a central fact table is surrounded by and joined to multiple data dimensions representing such abstract entities as time and location as well as more concrete entities such as customer, product etc. When laid out visually, the model looks like a star, hence star schema.

**Surrogate Key** – Keep control over record identifiers by generating new keys for the data warehouse. According to the Webster's Unabridged Dictionary, a surrogate is an "artificial or synthetic product that is used as a substitute for a natural product." That's a great definition for the surrogate keys we use in data warehouses. A surrogate key is an artificial or synthetic key that is used as a substitute for a natural key. Actually, a surrogate key in a data warehouse is more than just a substitute for a natural key. In a data warehouse, a surrogate key is a necessary generalization of the natural production key and is one of the basic elements of data warehouse design.

**Synonyms** – Synonyms are data elements that have different names but have the same meaning or represent the same business fact.

**Technical Infrastructures** - Technical infrastructures are closely related to architecture and are the technologies, platforms, databases, gateways and other components necessary to make the architecture functional within the corporation.

## **Afrikaans Opsomming van die studie**

In die informasie era waarin ons vandag leef is dit uiters belangrik om informasie te stoor op die effektiwste manier moontlik. Sonder informasie oor bv. kliente, produkte, mededingers ens. kan besighede nie 'n bestaan maak nie.

Met effektiewe storting van data will organisasies informasie kan stoor in so 'n wyse dat die nodige inligting vir die dag-tot-dag transaksies vinnig gestoor en ontrek kan word, maar ook dat analyses gedoen kan word, deur te kyk na informasie oor 'n langer termyn. Hier spruit 2 behoeftes na vore, en om die 2 behoeftes maksimaal te kan bevredig het ons 2 baie gespesialiseerde informasie sisteme nodig nl. die Operasionele sisteem vir die dag-tot-dag informasie benodighede en die Data pakhuis sisteem (Data warehouse) vir die analise van data.

Om 2 sulke gespesialiseerde sisteme te bedryf is nie net 'n groot taak wat baie arbeid verg nie, maar dit kos ook baie geld, en kan nie altyd bekostig word nie. Die navorsing in die dokument poog om vas te stel of daar 'n mark is vir 'n informasie sisteem wat 'n kompromie is van die 2 bogenoemde sisteme. Die "nuwe" sisteem sal so ontwerp word sodat dit die ander twee sisteme kan vervang, maar dit moet ook soveel moontlik voordele van beide bied. Die nuwe sisteem moet dus vir die dag-tot-dag sovel as die analise informasie behoefte gebruik kan word, die enkel sisteem sal egter sekere voordele inhou bv. Koste is laer as die 2 afsonderlike sisteme, maar ook sekere nadele bv. dit sal tipes nie so gespesialiseerd ontwerp kan word vir 2 behoeftes soos 2 afsonderlike sisteme nie. Die beoogte alhoewel minder gespesialeerde sisteem mag dalk tog 'n mark hê, en die studie poog om dit so te ontwerp om maksimaal voordeel en minimaal nadeel van die ander sisteme in te sluit.

Die studie begin deur die 2 sisteme te beskryf, en daarna 'n vergelyking tussen hulle te trek. Die studie gaan verder en wys hoe sal die ontwerp van so 'n sisteem lyk in vergelyking met die reeds bestaande twee sisteme. Voordele en nadele van die "nuwe" sisteem sovel as die ander twee sisteme word uitgewys, en ook sekere toepassings waar elk beter sal werk.

Daarna sluit die navorsing af met die gevolgtrekking dat so 'n sisteem wel sal kan bestaan en 'n doel dien in kleiner omgewings, maar dat dit tog nie so goed sal vaar vir groot organisasies nie.

## **Chapter 1: Introduction**

The world is filled with facts about almost anything thinkable. The motor industry for example, is filled with facts about different makes of cars, car owners and different production processes. These facts can add a great deal of value to an organisation or industry if they can be organized and managed in an effective way.

Any organisation that wants to survive and thrive in a competitive market has the need for accurate information, which managers can use to help them in making crucial decisions about the organisation. “The real problem being encountered by business leaders is not the shortage of data but an abundance of it. Estimates vary, but a conservative assessment of the growth of data being stored electronically suggests that it is doubling every two years” (Kelly, 1997:3). One needs to distinguish between data and information.

### **1.1 Data**

“Data is represented by isolated facts concerning a subject or groups of subjects stored on fragmented systems and which are measured with reference to the accuracy and integrity of each individual data item” (Kelly, 1997:4). Data lack context. By putting data in context allow analysts to draw conclusions. The data have been transformed into information.

### **1.2 Information**

“Information is data that are suitable for human interpretation, often with the purpose of revealing trends or patterns.” (McFadden and Hoffer, 1994:7) For an organisation to gain the competitive advantage data need to be manipulated to provide information.

### **1.3 Data and Information Storage**

For information to be produced there is a need to store the data that will be used for the information generation and the information that was generated. Data can be captured in different types of storage systems.

“The relational database has proven to be a popular and effective approach to database design, attaining the status of a dominant design in the market.” (Agosta, 1999:63)

The relational database is a single type of database, and it doesn't have to be an electronic means of storage. “A database is a shared collection of logically related data, designed to meet the information needs of multiple users in an organisation.” (McFadden and Hoffer, 1994:8)

The relational database design of an information system can play a major role in the characteristics of the information system. Some designs tend to capture data more effectively than others and some tend to retrieve data faster.

### **1.4 Motivation for this Research**

The need for information has led to a number of different specialised systems within an organisation. This research will focus on two specialised systems namely the operational system and the data warehouse (DWH). These specialised systems are designed to perform specific tasks, and each system has strengths in certain areas and weaknesses in other areas.

#### **1.4.1 Operational Systems**

The operational system is the foundation of an organisation, and serves as the data source for information. “Much of the transactional work of operational systems concerns answering day-to-day questions about service inquiries, product function, delivery schedules, payment schedules, exceptions to expected outcomes and relations with other business entities.” (Agosta, 1999:36)

Operational systems can be updated in one of two methods. In the first method transactions are processed in batches.

For this type of update the transactions will be captured and stored in one place, and in predefined intervals loaded into the operational system. Aspects like integrity and validity will be enforced on the data during this process.

The second type of update is where transactions are processed as they are captured. The integrity, validity and other checks will be performed as the transactions are processed. This data will be available in the operational system immediately after the transaction was captured; there is no need for a batch update. This process is called Online Transaction Processing (OLTP- see Glossary).

OLTP systems were designed to handle updates and not information retrieval. “Anyone who tried to use a major corporate database for both OLTP and querying soon realized some basic truths about systems for getting data in, versus systems for getting data out.” (Kimball, 1995(A):1)

#### 1.4.2 Data Warehouse

The lack of operational systems to satisfy the demand for information retrieval, has led to data warehouses. The data warehouse is designed to handle queries, and therefore outperform operational systems in retrieving data. “A data warehouse is simply a single, complete, and consistent store of data obtained from a variety of sources and made available to end users in a way they can understand and use in a business context.” (Devlin, 1997:20)

It is clear that both operational and data warehouse systems have a very specific task to perform, and that there is a clear definition between the two systems. There are certain drawbacks with more highly specialised systems compared with less specialised systems.

### **1.5 Problem Statement**

The perception that two specialised systems are always needed to provide accurate information for decision support in an organisation, and that there is no room for a single, less specialised system in the market, has led to this research. “The little crack in the database market iceberg is widening very rapidly. A large channel of blue water is now visible between OLTP systems and data warehouse systems. Both IS shops and vendors realize that you can benefit from having two systems that are specialised for each task.” (Kimball, 1995(A):1)

### **1.6 Main Objective of the Research**

The main objective of this study is to develop a single conceptual model as an alternative to an operational system and a data warehouse. This model still needs to satisfy the information needs of the organisation,

The main objective will be supplemented by the following sub-objectives:

- Investigate the main characteristics, design, advantages and disadvantages of operational and data warehouse systems.
- Provide guidelines to decide when to implement a specialised system (operational and/or data warehouse systems) and when to use an alternative system.

### **1.7 Significance of the Study**

Operational and data warehouse systems are both specialised systems designed to meet specific goals. These systems are expensive and the costs to buy and maintain them are ever increasing. “Data warehousing is a costly project with extensive data storage and processing costs, development costs and user query tool and training costs” (Kelly, 1997:17). The “alternative model” might provide some organisations with an alternative solution to their information needs.

Small organisations will definitely benefit from a single system compared to two specialised systems since the costs involved in developing and maintaining these systems are expensive and small organisations do not always have need such specialised information requirements.

### **1.8 Research Method**

In this study the viability of another alternative information system are being researched. This system will be proposed as an alternative to two other information systems. The research will be conducted as follows: Information is needed about the two systems (The Operational System and the Data Warehouse). The study will begin with the exploration of the operational system and after that the focus is moved to the Data warehouse system. This will provide the reader with a good background of the context of the study.

In the next session these 2 systems will be compared to each other, this section will serve as the area where the good and bad of each system are highlighted in comparison with one another. This will also highlight the possible areas where a third system can improve on the existing two.

In the last section an attempt is made to build such a system in theory. If it can be build in theory it should at least be possible to construct a practical example. The best way of knowing if something can exist is to build it and see. The proposed system then tries to solve the problems that comes with running an operational system and data warehouse in an environment, by converting to the “new” system.

### **1.9 Boundaries of the Study**

The research will focus on operational and data warehouse systems. Other types of information systems might be mentioned, but a detail description of them is outside the scope of this study.

## **Chapter 2: Operational Systems**

### **2.1 Chapter Objectives**

Data and information mean nothing to an organisation if it isn't represented and stored in a useful manner. This chapter discusses a certain type of information system that is currently being used in organisations and that was developed out of other systems to help organisations get the most out of their data.

This chapter focuses on operational systems. The particular the reasons why these systems are needed, their strengths and drawbacks. Enterprise Resource Planning (ERP) systems are covered in detail in this chapter

### **2.2 Operational System Classification**

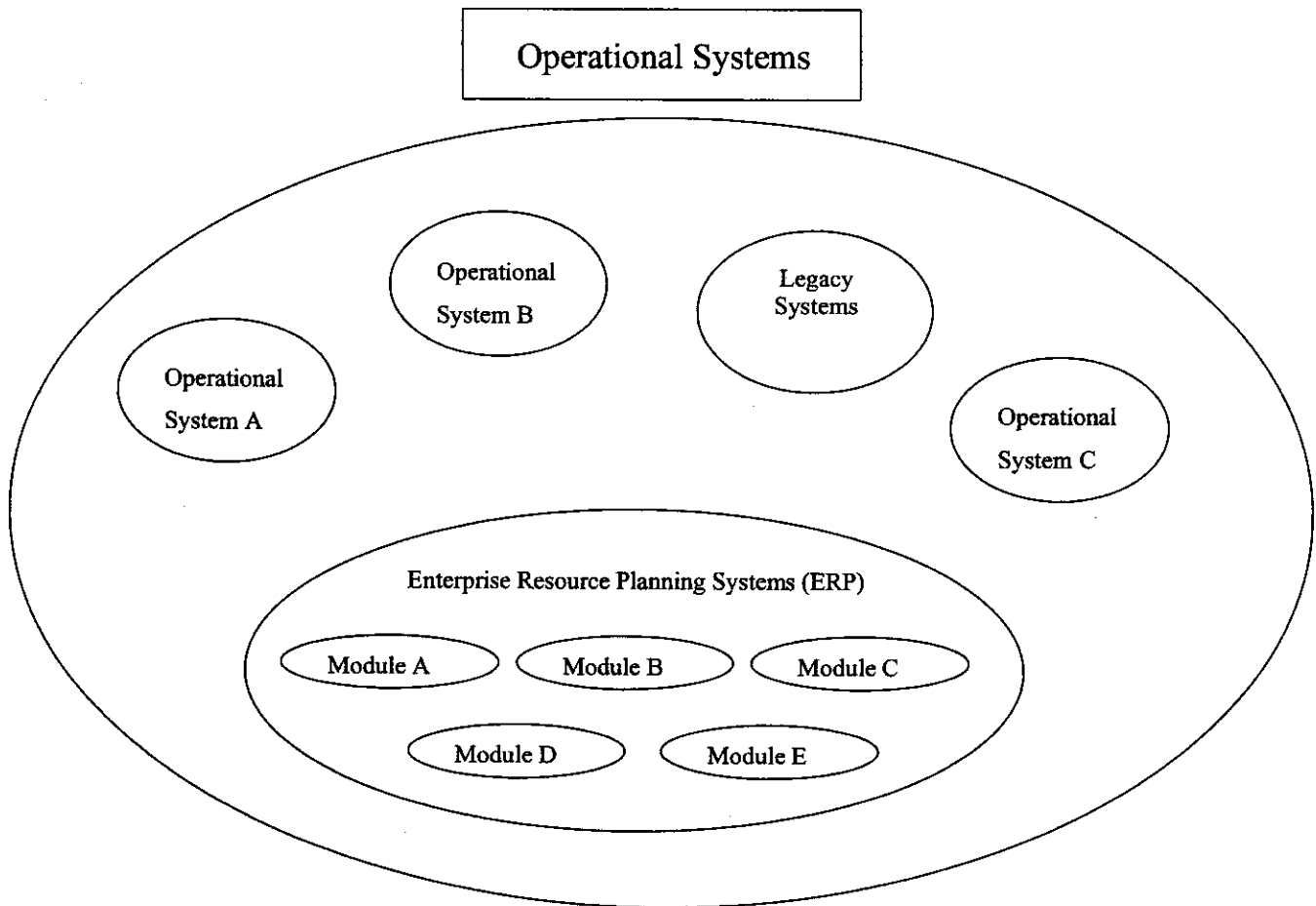
An operational system is the system responsible for running the day-to-day operations of the organisation. The term operational system is however very broad and a number of systems can be classified as operational systems. Figure 2.1 represents one classification of operational systems, and is by no means complete since only operational systems and terms important to this study are shown. They are: OLTP systems, legacy systems and ERP systems.

#### **2.2.1 Legacy Systems**

"Legacy systems refer to software that preceded the software that is now being implemented or considered for implementation. Legacy software typically has been developed by and for a specific firm. Legacy software is often mainframe software. The organisational requirements for legacy software include the staff that can do systems analysis, design and implementation of the software. Typically, there is a substantial maintenance cost of a legacy system as it is updated to meet emerging organisational needs."(O'Leary , 2000:19). A well-known example of a legacy system is the IBM AS400.

### 2.2.2 OLTP Systems

According to the definition (see Glossary), an OLTP system is an operational system that updates data online, therefore the data that are captured will be available for use almost immediately. OLTP systems include a wide range of operational systems, and ERP systems can also be classified as an OLTP systems, since ERP systems are operational and they are updated online (the online updating is not a requirement for an ERP system). OLTP can be seen as a data capturing system rather than an operational system, although it used in certain operational systems.



**FIGURE 2.1: OPERATIONAL SYSTEMS CLASSIFICATION**

## 2.3 Enterprise Resource Planning System (ERP)

### 2.3.1 The Evolution of Information Systems

This discussion of the evolution of information systems is a summary of the work done by O'Leary (2000:13-18).

#### 2.3.1.1 Computing

- **Mainframe Computing** – In this environment all the computing was done on a single computer. Typically this is handled by allowing users to share the computing resource, the mainframe computer. The user would typically be at a terminal with computing capabilities (or a personal computer that emulates a terminal). This was the environment that legacy systems operated in.
- **Client Server Computing** – Over time, the user began to have increasing computing capabilities locally. As a result, computing began to shift some of the processing to the user's computing facilities. This resulted in "client server" computing. This served as a huge advantage since there were a number of computing resources that could do computing instead of sharing only one resource.

The term "client" in "client server" refers to the user's computer while "server" refers to some other computing resource. This led to the need for an information system that can use the benefits of this type of computing.

#### 2.3.1.2 Networks

In the client server environment, there is a network between clients and servers. This network may be a local area network (LAN) or across the Internet. Network capabilities, standards, and security are critical factors to the success of the system.

- **LAN's, Intranets, and Extranets** – Local Area Networks (LAN's) are networks used to link computers over a relatively small geographical area, such as within a

building. This type of network normally has a large transmission capability. Wide Area Networks (WAN's) link computers over a large area. Intranets are typically WAN's that are used by a specific organisation. Extranets are typically WAN's that are used by a specific corporation and its partners.

- **Bandwidth, Standards, and Security** – A network must be able to accommodate the requirements of the applications. Bandwidth is the network's transmission capacity.

Standard protocols allow transmission of information in a common form, facilitating interaction between components. Perhaps the most widespread network standard associated with the Internet is TCP/IP (Transmission Control Protocol / Internet Protocol).

Two of the many approaches to security are firewalls and encryption. Firewalls enforce a site's security by controlling traffic flow. Only certain authenticated users are allowed access through the firewall.

Firewalls are typically placed between the corporations network and the external network. Encryption converts readable information so that only those who can decode the information will be able to read it. This is an important factor in the client server environment since information is sent over a network from a client to a server and vice versa. If the information is not encrypted any outsider will have access to it. For ERP systems it is important that the information is encrypted since this is the information that the organisation relies on to overcome the competition

#### 2.3.1.3 Databases

- **Flat File Databases** – Flat file databases are two-dimensional (rows and columns) files. They are easy to use but are limited in their capability to model enterprise events. Flat file databases are not directly related to other flat files. The main disadvantage of flat file databases is that they may contain substantial data redundancy. Since there is no relation between flat files means that if one captures,

form example sales information, information regarding the will be stored in this flat file. If the same customer makes another sale, all the customer information will be captured again. This is an unnecessary waste of space, and if the customer information changes the problem gets worse since all the duplicated information must be updated. This problem can be solved by the use of relational databases.

- **Relational Databases** - A relational database is a set of tables that are related or linked together. The method of capturing information in a relational database differs from flat files since there will be different tables for sales and customer information. All the relevant customer information will be captured in a customer table. In the sales table only the customer number will be captured every time the customer buys from the organisation. When the customer changes his address only one line needs to be updated in the customer table. The sales records will still be associated with this customer according to the customer number captured in the sales table.

Relational databases eliminate much of the redundancy of flat files. ERP systems, for most of the time, use relational databases because of this advantage.

### 2.3.2 The Evolution of ERP Systems

In the 1960's operational managers began to focus on Inventory Control Systems (Shankarnarayanan 1999:45). This was followed in the 1970's by the material requirement planning in the form of the MRP (Material Requirement Planning) systems, which focussed on scheduling materials in order to maximize production efficiency. This was followed by the MRP II (Manufacturing Resource Planning) systems in the 1980's. MRP II was an extension of MRP to shop floor and distribution management activities. In the 1990's MRP II was further extended to cover areas like Engineering, Finance, Human Resources, Project Management, etc. This included all the activities that were needed in the enterprise and hence the term enterprise resource planning.

“ERP systems developed out of individual and stand-alone computerized business systems that are generically referred to as heritage or legacy systems” (Mohammed 2000:20)

These legacy systems performed business critical tasks, with the main drawback the redundancy of data. Data was re-entered in every stand-alone legacy system in the organisation. This made updating of information just about impossible. “Each of these so-called legacy systems may provide invaluable support for particular business activity. But in combination, they represent one of the heaviest drags on business productivity and performance now in existence.” (Davenport, 1998:123) The costs involved in the storage of redundant data and maintaining it was high. Davenport (1998:123) explains, “ If a company’s sales and ordering system cannot talk with its production-scheduling systems, then its manufacturing productivity and customer responsiveness suffer. If its sales and marketing systems are incompatible with its financial-reporting systems, then management is left to make important decisions by instinct rather than according a detailed understanding of product and customer profitability.”

ERP systems promised to solve these problems. “ERP systems allow organisations whose existing information systems may be highly fragmented to bootstrap themselves into enterprise-wide integrated information systems by adopting a vendor developed solution. These systems incorporate vendors experiences and understandings drawn from business operations in a wide range of industries, in a variety of regulatory environments, and in many different countries around the globe.” (Zeleny 2000:519)

#### **2.4 Formal Definition of an ERP System**

The ERP system serves as the information system responsible for providing all the information that is necessary to run the day-to-day operation of an organisation. This doesn’t only include the sales and production sides of the business but also aspects like human resources.

“Enterprise Resource Planning (ERP) systems represent a new generation of information systems that are designed to process routine business activities across multiple functional areas of large corporate enterprises. ERP systems provide such organisations with highly integrated solutions that rely on the use of common database systems. Linkages between different functional activities within an enterprise and across its organisational boundaries are achieved by maintaining all relevant data in a single database. Thus data corresponding to a business transaction are entered into the system only once, and the effect of this transaction on different business processes and corresponding reports can be determined immediately. Most of the current generation of ERP systems are based on a client-server architecture.” (Zeleny, 2000:516)

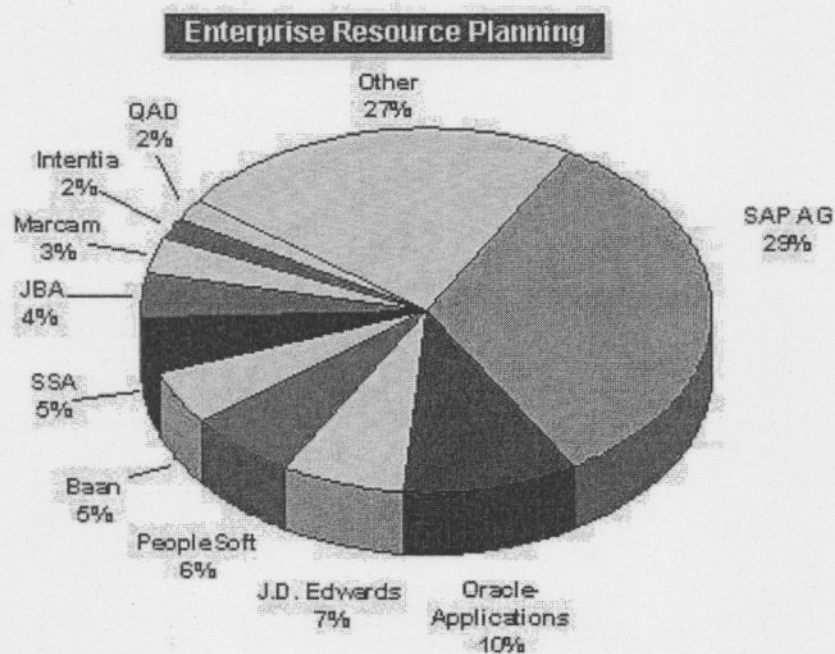


FIGURE 2.2: ERP MARKET SHARE (INFORMATIONWEEK.COM SEPT 1997)

According to (O'Leary, 2000:27) ERP systems must have the following characteristics:

- ERP systems are packaged software designed for a client server environment, whether traditional or web-based.
- ERP systems integrate the majority of a business's processes.
- ERP systems process a large majority of an organisation's transactions.
- ERP systems use an enterprise-wide database that typically stores each piece of data once.
- ERP systems allow access to the data in real time.

## **2.5 Vendors**

In this section a few market leaders in the ERP environment will be discussed. Figure 2.2 represents the market share of each ERP vendor.

### **2.5.1 SAP**

The largest market share for ERP is held by SAP. SAP was founded in 1972 in Walldorf, Germany.

“SAP has a reputation for acquiring firms with features that they are interested in and then completely reprogramming those systems for integration with R/3.” (O'Leary, 2000:29) R/3 is SAP's client-server based ERP system.

“SAP was the first to pursue specific industry versions of their software and the first to be truly international. In addition, they were among the first to pursue developments such as data warehouses.” (O'Leary, 2000:30)

### **2.5.2 Oracle**

Oracle is currently the second largest supplier of software. Oracle was founded in 1977 in the United States and Oracle is better known for their databases than for their ERP systems, but their ERP systems are becoming more popular.

“Oracle’s reputation in ERP systems is for developing a product that can be interfaced with other products in order to construct a best of breed system. Oracle is likely to build software in-house. In addition, possibly because of their basic focus on database management systems, Oracle was the first to provide a data warehouse product and the first to begin to integrate the Internet into their products” (O’Leary, 2000, 29)

### 2.5.3 JD Edwards

JD Edwards distributes an enterprise resource planning product called OneWorld, which is developed for the open systems market, like all the other major enterprise resource planning vendors, according to Buhrmann et al (1999:16).

“JD Edwards recently introduced its multi-platform software, OneWorld, which was designed to gradually replace its previous AS/400 product. Historically JD Edwards has been the leading supplier of AS/400 applications” (O’Leary, 2000:30).

“Recognizing that businesses cannot survive by themselves, JD Edwards has built on the traditional strengths of OneWorld, to provide the means to collaborate with other businesses.

The easier sharing of resources and information with customers and manufacturers creates value for the entire chain” (Mohammed, 2000; 26).

### 2.5.4 PeopleSoft

PeopleSoft is the fourth largest ERP vendor and was founded in 1987, but only went public in 1992. “PeopleSoft has become known for the broadest human resource capability. In many cases, firms have chosen some other ERP system for all other modules and PeopleSoft for human resources. In some cases, the quality of this human resource module led some clients to adopt the rest of Peoplesoft’s ERP modules.” (O’Leary, 2000: 29)

### 2.5.5 BAAN

BAAN was founded in the Netherlands in 1978. "It has successfully developed enterprise resource planning solutions that take advantage of areas where its competitors are weak, allowing it to acquire many blue chip companies." (Mohammed, 2000: 27)

BAAN is famous for its dynamic enterprise modeling architecture. This serves as a buffer between a business and the changes made to its underlying information technology. The actual components used to execute the strategies can be changed as new components are released, allowing the enterprise to leverage the ongoing innovation while minimizing the impact on the business (Mohammed 2000; 27).

### 2.6 Modules

Since SAP is the current market leader in the ERP environment, some of its modules and their functions will be listed. (O'Leary, 2000:31-32)

- AM (fixed asset management), which captures information relating to depreciation, insurance and property values.
- CO (controlling), which includes CCA (cost centre accounting), PC (product cost controlling), and ABC (activity-based accounting).
- FI (financial accounting), which includes GL (general ledger), AR (accounts receivable), AP (accounts payable), and LC (legal consolidation).
- HR (human resources), which includes PA (personnel administration) and PD (planning and development).
- MM (material management), which includes IM (inventory management), IV (invoice verification), and WM (warehouse management).
- PM (plant maintenance), which includes EQM (equipment and technical objects), PRM (preventive maintenance), SMA (service management) and MOM (maintenance order management).
- PP (production planning), which includes SOP (sales and operations planning), MRP (materials requirements planning), and CRP (capacity requirements planning).
- PS (projects system), which includes project tracking and budget management.

- QM (quality management), which includes CA (quality certificate), IM (inspection processing), PT (planning tools) and QN (quality notifications).
- SD (sales and distribution) system.

These modules, and the functions that they are designed for, explain why ERP systems are said to provide organisations with all their information needs for the daily running of the organisation, since all the functions in the organisation are covered by these modules.

Companies that implement ERP systems sometimes prefer the “best of breed” approach. This will happen if their current ERP vendor or the vendor that they decide on doesn’t support a certain function in one of its modules, but some other vendor does. The company will then opt for the implementation of their ERP system that is made up from modules and vendors that best suit their needs.

It is important to note that there might be difficulties in the interface between different modules from different vendors, and therefore it must be ascertained that this scenario of implementation has been tried and tested.

## **2.7 Architecture**

“Most ERP Systems aim to provide a platform-independent suit of application that can work in distributed computing environments as well as in centralized environments. One way that this is achieved is by relying on mainstream relational database systems for organizing and managing data and standard TCP/IP networks for data communication” (Zeleny, 2000:522).

A typical ERP system is consist of layers. The layers are: (Zeleny, 2000:522)

### **2.7.1 The Database Layer**

At the heart of the ERP system is the database management system. Database servers within ERP systems are the most powerful and protected servers. An ERP’s database may be maintained on multiple database servers, offering redundancy, robustness and security. To reduce security over the network as well as disk accesses, database servers

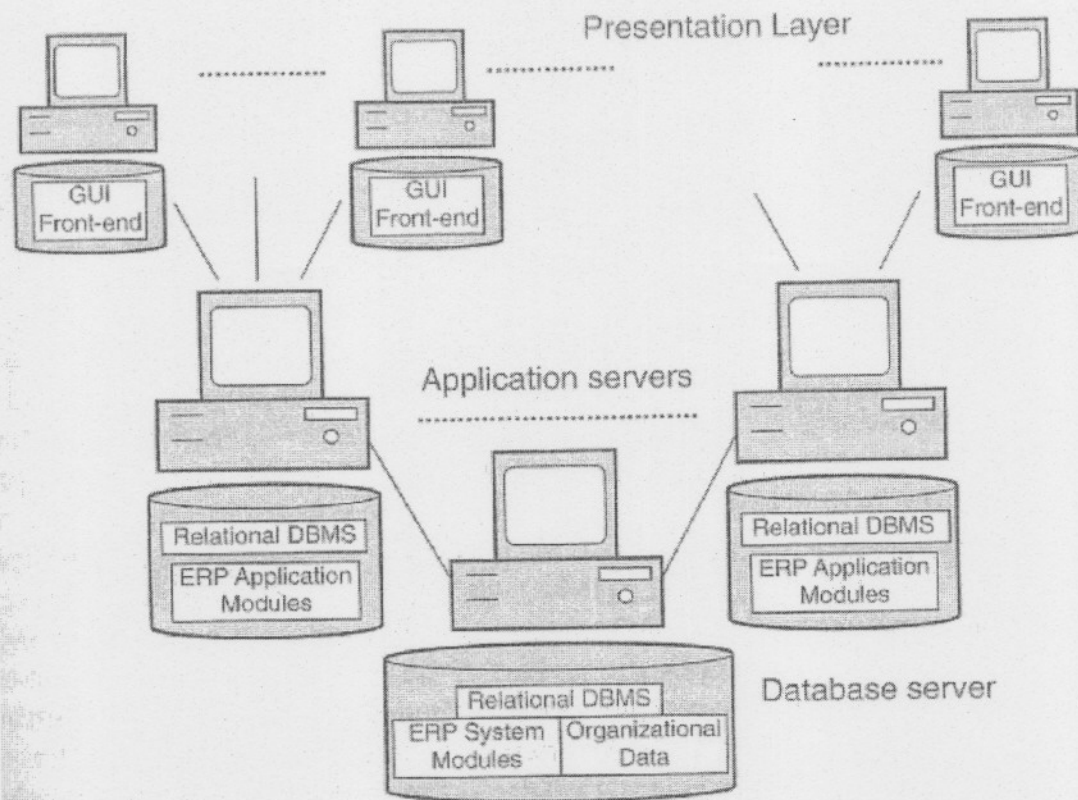
use caching mechanisms. If two users are requiring the same data, the data will be cached after the first user access, and the next user will have quick response for his query.

### 2.7.2 The Application Layer

To improve performance and scalability, data access within ERP systems is usually isolated from program access. This is achieved by physically separating application programs from the database servers. The applications are run on servers called application servers. Multiple application servers can be used in a single ERP implementation. Application servers are responsible for functions like user authentication, communication, maintaining locks on data items and the general tasks involved in the application running.

### 2.7.3 The Presentation Layer

A presentation layer provides screens to capture transaction-specific data from a user and provide appropriate formatted reports using menu driven GUI (Graphical User Interface) screens. Since little processing is done at the processing layer, client machines don't have to be powerful. Caching of GUI screens is also supported to speed up the loading process of the interfaces.



**FIGURE 2.3: LAYERED ARCHITECTURE OF AN ERP SYSTEM**

## 2.8 How ERP Works

The architecture and modules of ERP are now better known, and it is a good idea to look at a typical example of how an ERP system functions. This example is taken from O'Leary (2000, 36-37). The example is based on the SAP R3 ERP system. SAP version R3 is the client server version of SAP.

International Sneaker Company (ISC) is a U.S Company that produces sneakers in Taiwan and they sell their product to the worldwide market. A typical transaction will be as follows.

### 2.8.1 Ordering

A sales representative from ISC takes an order from a retailer in Brazil. Entering the data on her personal computer, the sales representative accesses R/3's sales module. The

system checks the price as well as the discounts that the retailer is eligible for. The system also checks the retailer's credit history to make sure that the firm wants to make the sale.

This type of order placement can also be done by electronic handheld devices and through cell phones, if this function is included in the system installation.

#### 2.8.2 Availability

R/3 software next checks the inventory. If it finds that half the order is available from the warehouse in Brazil, it knows that that portion of the order can be filled immediately. R/3 finds that the other half of the order needs to be delivered from the ISC's factory. Information regarding the preferred delivery times of the customer can also be captured.

#### 2.8.3 Production

R/3 alerts the warehouse to ship the portion of the order that is in stock to the retailer. In addition, R/3's manufacturing software schedules the production of the remainder of the order. An invoice is printed in Portuguese.

#### 2.8.4 Manpower

When scheduling production, R/3 note that there is a shortage of workers to handle the order. It alerts the personnel manager of the requirement to hire temporary workers.

#### 2.8.5 Purchasing

R/3's materials planning module notifies the purchasing manager that it is time to order new raw material and also of the amounts that need to be ordered.

#### 2.8.6 Order Tracking and More Ordering

The Brazilian retailer logs onto ISC's R/3 system through the Internet and sees that a portion of the order has been completed. In addition, the retailer uses this as an opportunity to place yet another order.

It is exactly this enterprise-wide integration that makes the ERP system a necessity for an organisation. There are no more risks of the sales person forgetting to communicate the order to the production department, or that the available stock quantity in the sales system is different from the amount in the inventory system, since it is only using one integrated system across the organisation and not stand-alone systems as was the case previously. It is clear from this example how an ERP system can benefit an organisation with its daily running and organizing.

### **2.9 Return of Investment**

From the previous example it is clear that an ERP system adds value to the organisation that implements it. It is difficult to calculate the benefits provided by an ERP system since: (Brady, Monk, Wager 2001:30).

- ERP sometimes increases revenue and decreases expenses in intangible ways that are difficult to calculate.
- Some changes take place over such a long period of time, that they are difficult to track.

A critical point of any expense in an organisation is if the amount paid for the system will be worth the while. According to Brady, Monk, and Wager (2001:30) the following return of investment can be expected from an ERP system.

- Because ERP eliminates redundant effort and duplicated data, there can be a savings in operations expense. One study (Brady, Monk, Wager 2001:30) indicated that 33% of companies experienced a cost savings in sales order management, and 34% of companies said their ERP systems reduced their personnel needs.
- Because an ERP system can help move goods and services through the pipeline faster, more sales can be generated every month.
- In some instances, a company that doesn't implement an ERP system might be wedged out of business by competitors who have an ERP system.

- A smoothly running ERP system can save a company's personnel, suppliers, distributors, and customers much frustration.

## 2.10 Design

ERP systems are normally associated with highly normalized database structures. This means that data redundancy is kept to a minimum and therefore the storage space needed is also kept to the minimum. This type of design has got its strengths, but it also has its weaknesses. An example of an ERP system is shown in Chapter 5.

### 2.10.1 Entity Relationship Modelling

“Entity-relationship modeling is a logical design technique that seeks to eliminate data redundancy” (Kimball, Reeves, Ross, & Thornthwaite 1998:140). It is important that only one master record is available in the system for a specific customer for example. If the customer record is stored in more than one place, the problem arises when this record must be changed. This record must now be searched and changed in a couple of places. The bigger problem is if one of these customer records is floating in the system without being updated, this can cause deliveries to be sent to the wrong addresses.

“Entity-relationship modeling is a discipline used to illuminate the microscopic relationships among data elements. The highest art form of entity-relationship modeling is to remove all the redundancy in the data. This is immensely beneficial to transaction processing because it makes transactions very simple and deterministic. The transaction of updating a customer's address may devolve to a single record lookup in a customer address master table.” (Kimball, Reeves, Ross, & Thornthwaite 1998:140).

There are problems with this approach since the removing of redundant data goes hand in hand with normalization. Normalization on the other hand goes hand in hand with the creation of more tables.

The problems that arise with ER-Modelling are as follows: (Kimball, Reeves, Ross, & Thornthwaite 1998:142).

- End users cannot understand or remember an entity-relationship model. End users cannot navigate an entity-relationship model.
- Software cannot usefully query a general entity-relationship model. Cost-based optimizers that attempt to do this are notorious for making the wrong choices, with disastrous consequences for performance.
- Use of the entity-relationship modelling technique defeats the purpose of data warehousing, namely intuitive and high performance retrieval of data.

### 2.10.2 Normalization

“Normalization is the process of converting complex data structures into simple, stable data structures.” (McFadden, & Hoffer 1994:209)

Data structures can be classified according to 6 normal forms. The higher the normal form, the more normalized the data become.

“A normal form is the state of a relation that can be determined by applying simple rules regarding dependencies to that relation.” (McFadden & Hoffer 1994:209).

#### 2.10.2.1 The Normal Forms

The six normal forms are as follows: (Agosta 2000:161)

- **First Normal Form** - Data is atomic and elementary. One value is stored in each attribute or column (No repeating groups of data are allowed).
- **Second Normal Form** - Every attribute not in the primary key must depend completely on the primary key.
- **Third Normal Form** - The primary key completely determines every attribute and does so in a non-transitive way. There mustn't be any transitive dependencies. A transitive dependency is a dependency between two or more non-key attributes.

- **Boyce/Codd Normal Form** - The need for transitivity is overcome here. For example, a table represents sales based on a unique key customer, employee, product, promotion, but some of the customers are really employees, so they are not eligible for the promotion. The solution is to break this down into customer, employee, as one table, and customer, product, promotion, sales as the other table.
- **Fourth Normal Form** - Multi-valued relations belong in separate tables. Certain kinds of ambiguities can be eliminated by this rule. Customers can speak multiple languages or drive multiple automobiles. According to this rule, these belong in separate structures to avoid redundant entries.
- **Fifth Normal Form** - Multi-valued relations must satisfy all the defined unique candidate keys. Similar to above with the addition of further business rules or business constraints. The recommendation is the same. This is the navel into the unknown.

Note that to classify a data structure in 3<sup>rd</sup> normal form, for example, is to guarantee that it will satisfy the rules for 3<sup>rd</sup> normal form, however it might also be in 4<sup>th</sup> normal form, since it is possible that for the structure in 3<sup>rd</sup> normal form the conditions of 4<sup>th</sup> or 5<sup>th</sup> normal form will never occur.

## 2.11 Implementation Phases

According to Zeleny (2000:523) an ERP implementation project can be divided into four distinct phases:

### 2.11.1 Organisational and Conceptual Design

This phase consists of defining the objectives of the project, defining the system environment, system setup and test clients, defining existing organisational structures and existing business processes, and mapping these processes to ERP components.

### 2.11.2 Detailed Design and Implementation

This phase includes specifying details of the company structure and details of its processes and functions, defining and entering master data, defining user interfaces, and developing a plan for transferring detailed data.

### 2.11.3 Preparations for Production

This phase consists of setting up the production system, installing the hardware and software for the user interface, developing user documentation, inputting master and detailed data into the production system and testing the system for realistic workloads.

### 2.11.4 Productive Operation

This phase includes developing a user support infrastructure, fine-tuning the system, and closing out the project.

In implementing an ERP system not all applications are deployed at the same time, since ERP implementations often require a business process redesign. To implement it in such a way gives the organisation the needed time to perform these changes before moving to the next application.

## 2.12 Strengths and Weaknesses

ERP systems have proved to be a critical part of any complex organisation and there are a number of advantages that they will add to the organisation, but as with any system there are also some weaknesses in ERP systems.

### 2.12.1 Strengths

- ERP systems allow for easier global integration: Barriers of currency exchange rates, language, and culture can be bridged automatically.
- ERP systems not only integrates people and data, but also eliminate updating and repairing many separate computer systems

- “ERP systems allow management to manage operations, not just monitor them. For example, without ERP, getting an answer to “How are we doing?” requires getting data from each business unit and then putting the data together for a comprehensive integrated picture. The ERP already has all the data, allowing the manager to focus on “What are we going to do better?” (Brady, Monk, Wagner, 2001:28).
- ERP systems force companies to re-engineer their business processes.
- ERP systems are updated on-line and therefore the information being entered into one module is available immediately in the other modules.
- “The quantifiable benefits are increased process efficiency; reduced transaction cost due to the availability and accuracy of data and the ability to turn that data into meaningful information.” (Norris, Hurley, Hartley, Dunleavy, & Balls, 2000:55)
- “Qualitative benefits include a more flexible governance and organisational structure, and a work force ready to change and focus more on high value-added tasks and to more easily capitalize on opportunities that represent themselves.” (Norris, Hurley, Hartley, Dunleavy, & Balls, 2000:56)

#### 2.12.2 Weaknesses

- Costs are one of the major drawbacks of ERP system implementation, and this aspect can have a huge impact on an organisation if the ERP installation is unsuccessful. “Acquiring an R/3 software system is very expensive. In addition, the software is so powerful, many companies find they must buy new hardware to accommodate the software” (Brady, Monk, & Wagner, 2001:26). Other costs involved in the implementation of an ERP system are consultant’s fees, training costs and the time for the implementation that cause disruption in the business - no wonder some companies claim that the implementation of an ERP system caused them to go under.
- Time consumption of ERP implementations. “Installing ERP requires a lot of time, more than 12 months in most cases, and up to 36 months or more in extremely large,

complex companies that are simultaneously engaging in a high degree of process change.” (Norris, Hurley, Hartley, Dunleavy, & Balls, 2000:51)

- Complexity of the system. “The very features that make ERP systems attractive to large organisations also bring about a high degree of complexity. For example, the number of tables in the current version of SAP R/3 is around 15 000” (Zeleny, 2000:520)
- “Some practitioners have argued that ERP developers have not spent sufficient time considering the user interface. Other practitioners have argued that the problem of difficulty of ERP input results because ERP developers generally have designed from the database out, not the user interface in. The screens have been the last part of the whole process” (O’Leary, 2000:57).
- “Insufficient reporting from the ERP systems led to the need for data warehouses (see Chapter 3) to enable the users to access decision-making information in an environment designed to facilitate the generation of reports.” (O’Leary, 2000:57).

### **2.13 Chapter Conclusion**

This chapter explained ERP systems. These systems are becoming more popular since companies can have a consolidated view of their organisation and they can use this to be more competitive. “The reason for companies upgrading to enterprise resource planning systems have concentrated on eliminating the year 2000 problem. This has changed, now that the requirement to consolidate information for effective decision-making has been satisfied. Companies are now focussing on how to use the system for competitive advantage, moving from just saving costs to making money” (Mohammed, 2000:46).

ERP systems are complex and therefore it is difficult to understand the information they contain. This has led to the need for data warehouses where users can easily understand their data. In the next chapter data warehouses, which will satisfy some of the information needs that still remain in organisations, will be discussed.

## **Chapter 3: The Data warehouse**

### **3.1 Chapter Objectives**

Chapter 2 explained operational systems and discussed the ERP system in detail. The drawbacks of these systems led to the development of the data warehouse environment.

The objective of this chapter is to discuss data warehouses. The discussion will include the origin of data warehouses, what data warehouses are, why we need them, who uses them, variations of data warehouses and the design and basic elements of data warehouses.

### **3.2 Background**

Data warehousing has been around for some time although it wasn't classified under the term "data warehousing". Organisations tried to harmonize different sources of data and to clean their databases over the years. Data cleaning is the process of removing unwanted data out of the database and to enforce data quality rules on the data. Analysts in many organisations extracted data from operational systems to their own hard drives to perform data cleaning and roll-ups to enable them to obtain answers to questions asked by management. An example of a roll-up of data is when the data in the database is stored on a transaction level, thus a line of data in the database for every transaction, if one needs the sales amounts per day, all the transactions must be added up (rolled-up) to give totals by day. No standard was in place to perform these rollups and cleaning of data and it caused a lot of problems.

Tools to assist end-users to analyse data have made it possible for organisations to methodically develop a system that was called a data warehouse, and to convert this system from the experimental phase to the stage where it was implemented in the organisation.

The concept of data warehouse originated mainly from the need to convert data into valuable information. “This is important, because many large organisations have long been data rich and information poor, resembling informational wastelands incapable of providing useful business intelligence to management - either because of incompatible databases and disfunctional networks or as a result of ambiguity of definitions leading to irrelevant data.” (Barquin & Edelstein, 1997:9)

It is important to note that a data warehouse does not contain data different than the operational system, but it is rather an alternative data source where data cleansing took place to provide an organisation wide view of the information. Information from different sources can also be combined into a single data warehouse to provide end-users with a single source of information.

Data warehousing is not an off-the-shelf product that can be purchased and installed; it is an architecture and therefore something that must be customized to suit the organisation. Data warehousing is driven by the business environment and not by technology trends.

The data warehouse is not a storage space used to store archived data, but rather the source of company wide information that resembles the organisation and its business processes that is understandable and accessible to users. The interface is very important, since the success of the data warehouse depends on the users having faith and trust in the system. Another requirement for the data warehouse is that information can be retrieved rapidly, and it should enable end-users to drill down into the information to get a better picture of the data. The data warehouse also provides a basis on which multi-dimensional databases (MDD) can be built for Online Analytical Processing (OLAP). (See glossary)

### **3.3 Decision Support Systems (DSS)**

It is important to distinguish between data warehouses and decision support systems. The data warehouse normally provides the source data to the decision support system. The decision support system will give managers added “tools” to assist in decision-making. The decision support system will be an application that is run “on top” of the data warehouse to analyse data in the data warehouse.

“Decision support systems provide sophisticated data analysis capabilities, but generally require more expertise to use than executive support systems. OLAP tools that provide “slicing and dicing” of the data have this capability.” (Bischoff & Alexander, 1997:375)

### **3.4 The Data Warehouse Usage**

Data warehouses will be used by managers and end-users to help them understand the business and make judgements and decisions based on this knowledge. The data warehouse will provide information on a global perspective; it enables users to analyse information with a broad perspective. Users will, for example, be able to compare sales from different periods in history, as well as trends in sales patterns for different customers, items and other variables in the organisation. Data warehouses were designed to provide information on a high level, but it also provides the functionality to drill into the detail. “What was the total sales amount for product type X per month last year?” is an example of a question that the data warehouse must be able to answer. The user can then drill down into the data to find more detail like “what was the top product within type X per month last year?”

Because a data warehouse is responsible for providing information on trends, there is no need for the data to be updated on a second to second basis. The data warehouse will normally be updated on a daily, weekly or monthly basis, and this is usually done when there is no or little traffic on the system.

The data warehouse must be designed to handle queries rather than updates, since this system will be used to answer multiple questions. Normally the data warehouse will not allow users to update any records in the database; this will be done by the administrators and on a timely basis.

### **3.5 Data Warehouse Users**

It is important to distinguish between the different types of users that the system will provide information for (Kelly 1997:24). A brief description of different types of users follows.

#### **3.5.1 The Operational User**

This type of user requires frequent pre-defined reports. These needs are generally satisfied by the provision of standard reports, which can be executed to extract the information in a pre-defined format. This type of user will normally have little knowledge of the business.

#### **3.5.2 The Power User**

This type of user will have a range of computer skills and is capable of creating his/her own reports if such an environment is provided. The needs of such a user will normally be satisfied through the provision of query tools to access the data, and a comprehensive information catalog describing the data.

#### **3.5.3 The Executive User**

This type of user will normally not have the time to create custom reports, and they might not have the skills to do this. This type of user will normally have a great deal of business knowledge and knows exactly what information they want. The executive user is likely to have ad hoc requirements that are not easily defined. This type of user will also have power users to create the reports for them, and a query desk where professionals will generate the reports. This type of user normally has an Executive

Information System (EIS - see Glossary) type interface that allows the executive to drill down into the data.

This type of interface is like a digital “dashboard” where the executive user can see a summary of the information on a high level; the executive can then drill down into the problem areas if required.

### **3.6 Data Coverage**

“The need to access historical data is one of the primary incentives for adopting the warehouse approach. In particular, historical data plays a significant role in warehouses used for trend analysis of buying and usage patterns, which focus on particular areas of business data.” (Devlin 1997:104)

The data warehouse will normally contain historical and current data. The most current data in the data warehouse will normally be a day old, depending on how often the data warehouse is updated.

The requirements for maintaining historical data fall into two broad categories. (Devlin 1997:105)

- A view of the business at a given time. End users need to see views of the business as it exists at different times in the past. The need for some of these views of the business is known in advance, so the Information Technology department can store the required data. Other needs are not predictable, and the data can therefore not be stored in advance, so end users need an approach that enables them the generation of such retrospective data.
- Business trend analysis. The analysis of business trends can be expressed in terms of different time periods. For example sales can be compared on a month-to-month level, but a monthly level might not be sufficient and a daily level would be required. It is thus important to store the historical data on a level that will enable users to do trend analysis on the level that they require.

### **3.7 Data Integration**

One of the key factors of a data warehouse is to provide integrated data to end-users. This is not always easy or possible especially if data are integrated from different sources. Assume that each source system contains sales information from two different companies. If the two companies also sell goods to the same customer, they will probably have different customer numbers and descriptions for the same customer. To be accurate these two customer records must be integrated to provide only one customer record in the data warehouse. If this isn't taken care of it will be difficult analyse a specific customer.

### **3.8 Data Quality**

Aspects that might prevent quality data in the data warehouse are for example missing values in the source data, such as a missing price of a sale or recording incorrect values. Differences in the way data are defined and used or data heterogeneity can influence the quality of the data. An example is when two fields have the same name, but refer to different things (homonyms), such as cost meaning the price of an item or the selling price of a collection of items. A related problem is synonyms where the same item is called by two different names, for example two fields named "color" and "colour", or "grade" and "class".

Another aspects that must be considered for data quality is the correctness of data. Data in the warehouse must be balanced back to their original sources where possible. End users will not accept the data warehouse if they suspect that the data in the data warehouse is not correct.

It is important that the data warehouse design will exclude the possibility of multiple query paths. If there are multiple ways to join different tables in the data warehouse, the chances are good that different joining paths will provide different answers.

For example, suppose we have a query for the sales amount of a specific item for a specific customer and a specific time period. If the parameter values are the same, but the joins between the folders were different one might get different answers to the same query. Thus, it is important that such queries can follow only one join path, and if there are more than one join path, the users must be informed on exactly what each is for, or the data warehouse might be seen as a system that provides inconsistent information.

To solve these problems, developers use a process called “Extract Transform and Load” (ETL). During this process in which the data is loaded into the data warehouse the developers will need to discuss various issues to find solutions to these problems. The ETL process will help take care of data integration and consistency. The ETL process will be discussed later.

### **3.9 The Basic Elements of the Data Warehouse**

#### **3.9.1 Source System**

A data warehouse needs data from a system that is responsible for capturing the data. This will normally be the operational system of the organisation. One of the advantages is that the data warehouse can contain data from multiple source systems.

#### **3.9.2 Data Staging Area**

The data staging area is a storage area where all the cleaning, aggregation and other data transformation will take place. The data staging area will normally be distributed over number of machines. “The key defining restriction on the data staging area is that it doesn’t provide query and presentation services. As soon as a system provides query and presentation services, it must be categorized as a presentation service.” (Kimball, Reeves, Ross, & Thornthwaite, 1998:16)

### 3.9.3 Presentation Server

The physical machine where the final data is stored is called the presentation server. The presentation server will be available for direct querying by end users.

### 3.9.4 Dimensional Model

Dimensional modeling is a logical design technique that seeks to present the data in a standard framework that is intuitive and allows for high-performance access. It is inherently dimensional and adheres to a discipline that uses the relational model with some important restrictions. Every dimensional model is composed of one table with a multi-part key called the fact table and a set of smaller tables called dimension tables. (Kimball, Reeves, Ross, & Thornthwaite 1998:144)

“Dimensional modelling is a specific discipline for modelling data that is an alternative to entity relationship (E/R) modeling. A dimensional model contains the same information as an E/R model but packages the data in a symmetric format whose design goals are user understandability, query performance, and resilience to change”. (Kimball, Reeves, Ross, & Thornthwaite, 1998:17)

The goal of dimensional modelling is to design the database in such a way that it represents the organisation. The main components of a dimensional model are fact and dimension tables. In E/R modelling the main components are entities and relations. (Fact and dimensional tables will be discussed later in this chapter.

### 3.9.5 Data Mart

A data mart is a logical subset of the data warehouse. The data warehouse contains the union of all its data marts. The data mart is probably built by a single part of the organisation, and is usually centred around a single business process. The data warehouse will, for example, consist of a data mart for sales, a data mart for inventory and a data mart for production.

### 3.9.6 Online Analytic Processing (OLAP)

The OLAP (see glossary) vendors' technology is non-relational and is almost always based on an explicit multi-dimensional cube (see glossary) of data. OLAP databases are also known as multi-dimensional databases. OLAP are the processes where managers can "slice and dice" through the available data to find correlation patterns that help them in making decisions.

### 3.9.7 Relational Online Analytic Processing (ROLAP)

ROLAP is a type of OLAP that gives a relational database a multi-dimensional flavor. The data for ROLAP is in a relational database, but will be accessed through a multi-dimensional definition of the data, and thus will look like a "multi-dimensional" data source.

### 3.9.8 Multi Dimensional Online Analytic Processing (MOLAP)

MOLAP is a type of OLAP where the source data will be extracted and transferred into a different data source that resembles a cube structure. Aggregations of data will be pre-calculated to a specific granular level, and this will usually have performance advantages over ROLAP.

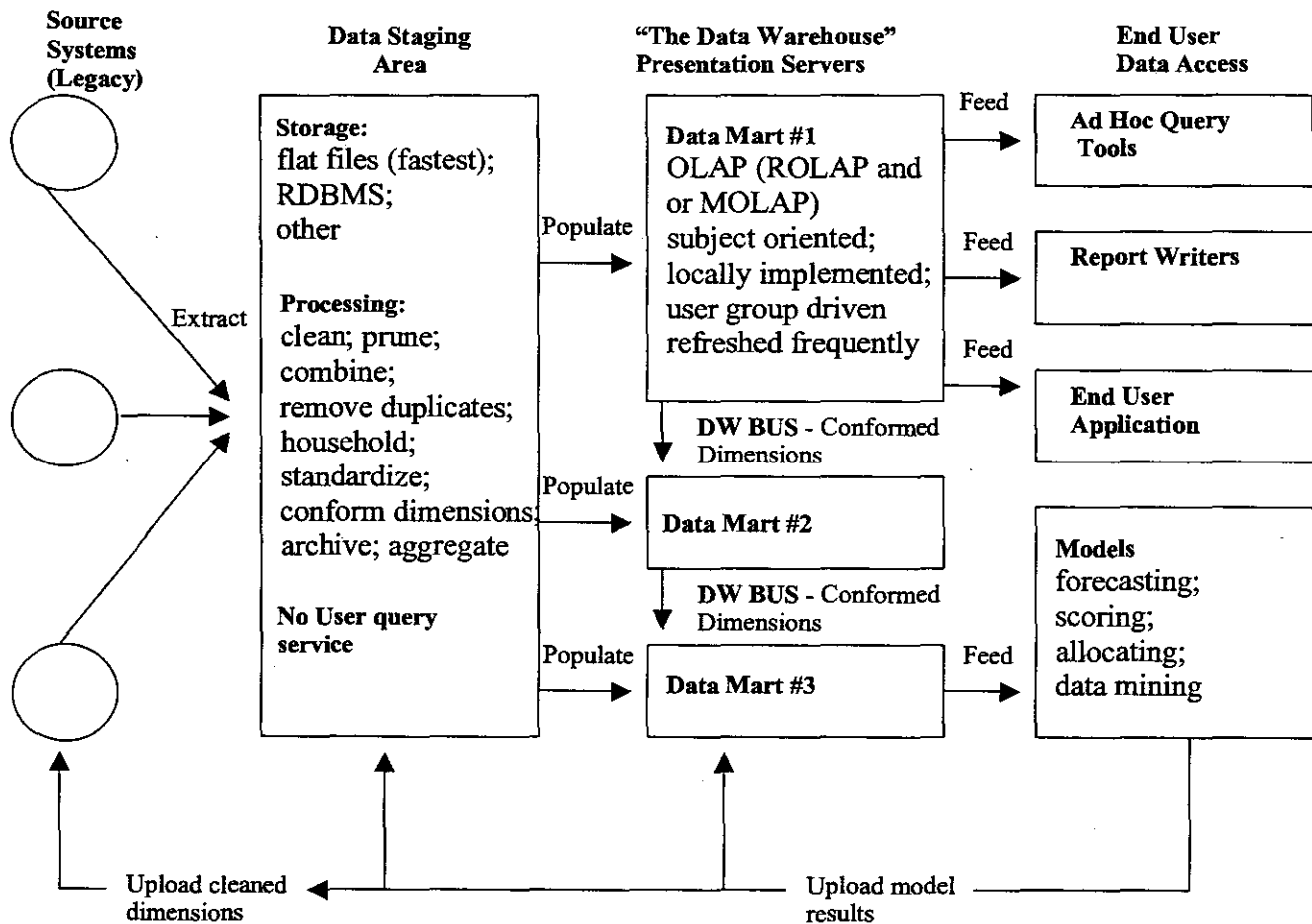
### 3.9.9 End User Application

"A collection of tools that query, analyze and present information targeted to support a business need. A minimal set will consist of an end user data access tool, a spreadsheet, a graphics package and a user interface facility for electing prompts and simplifying the screen presentation to users." (Kimball, Reeves, Ross, & Thornthwaite, 1998:21)

### 3.9.10 End User Data Access Tool

This forms part of the end user application, but is responsible for the connection to the database and communication with the database.

“A client of the data warehouse. In a relational data warehouse, such a client maintains a session with the presentation server, sending a stream of separate SQL requests to the server until the end user data access tool is done with the SQL session and it turns around to present a screen of data or report.” (Kimball, Reeves, Ross, & Thornthwaite, 1998:21)



**FIGURE 3.1: THE BASIC ELEMENTS OF THE DATA WAREHOUSE**

### 3.10 Extract Transform Load (ETL)

The process that is used for populating the data warehouse is called the ETL process. The ETL process will be developed, tested and then be used to populate the data warehouse. The first step will normally load all the history data into the data warehouse.

The data warehouse will then be incrementally updated with new data on a regular basis. The regularity of data loads can be adjusted to suit the business needs. Every data record pass the ETL process before it can be loaded into the data warehouse. Exception handling will also be build into the ETL process, thus if records are encountered with missing values an exception will be raised. Exceptions can be encountered for a number of reasons, like duplicate or invalid records. These exception records and reasons for causing the exceptions will be stored after each load. The data warehouse administrator will need to investigate the exceptions to see if the records can be validated and loaded. The ETL process will normally be performed on a daily basis when the computer network traffic is low, since the execution of the ETL process can be resource intensive and will slow down the normal daily operations.

A brief overview of the three steps of the ETL process follows.

#### 3.10.1 Extract

During this phase data will be extracted from different data sources and will normally be loaded in a staging area. The staging area is a temporary data storage area (the source for the transform stage).

#### 3.10.2 Transform

The transformation phase is the “heart” of the process and where the data are transformed into information. The transformation phase can include the following tasks.

- **Data integration** – This is where data are integrated to form a single version of a record where needed.
- **Data Cleaning** – Garbage data is transformed into more useful data, for example to fill blanks with default values.
- **Summarization/Aggregation** – Data is pre-processed in order to build summarizations and aggregations of the data, which can be associated with calendar periods, regions departments or projects.
- **Derivation** – Data elements can be combined and subject to calculations or other processing to derive new data or information.

### 3.10.3 Load

If records have been transformed into the required format it can be loaded into the data warehouse. Records are loaded from the staging area into the actual data warehouse tables and will be ready use.

### 3.11 Data Warehouse Scope

“Data warehousing is not a technology trend per se, and primarily driven by the business environment. Data warehousing represents a response to the urgent demands of the business users who have, not for the first time, been stridently articulating an inchoate and poorly defined vision for an integrated corporate database.” (Kelly 1997:16)

The data warehouse will incorporate some technology trends, but it isn't driven by the need to be a technology first. The scope of the data warehouse is therefore narrowed down to the data that is used to manage the business.

### 3.12 Development Methodology

There are a number of different methodologies for the development of data warehouses. One such a methodology is called the Metis Methodology.

There are eight stages defined by the Metis methodology to develop a data warehouse. An overview of the different elements that need to be addressed in each stage follows. (Kelly, 1997:78)

- **Organisational readiness assessment** - This stage is concerned with assessing the key business drivers, the organisational culture, the legacy systems, data quality and skills required.
- **Business strategy definition** - This stage is concerned with capturing the business vision for data exploitation and should result in a comprehensive business strategy model.

- **Data warehouse architecture definition** - This stage defines the logical architecture for the data warehouse. It defines how the business strategy model is to be supported by information technology, and it also defines the scope of the data warehouse.
- **Data warehouse infrastructure design** - During this stage the physical infrastructure components are assembled and integrated. Issues such as performance, meta-data management, copy management, data access and system integration are addressed.
- **Design and build** - The physical design of the data warehouse database and the data staging strategy are completed during this stage and the first iteration of data is populated onto the target platform.
- **Data exploitation** - By now it is possible to commence the development of applications that are envisioned in the strategy model.
- **Implementation** - The implementation stage is concerned with the rollout of the first applications and is directed at user training, support, publicity and assessing the business value achieved.
- **Administration and operation** - This stage is concerned with ensuring that the environment is made more robust and rugged with regard to data security, operations and administration procedures, system availability and performance tuning.

### **3.13 Data Warehouse Economic Justification**

“A data warehouse that is tightly linked to an important business objective will, if it meets its promise, maintain its economic justification as long as the objective exists and will be looked at as a model for a solution to other key business objectives that rely on effective use of information.” (Barquin, 1996:80).

Top management will expect the following from the data warehouse: (Barquin, Edelstein, 1996:84)

- **Supports strategic decision making** – The data warehouse will have to support strategic decision making by providing detail and summary data that can be used for trend analysis, performance measurement analysis, correlation, statistical analysis and other similar requirements.

- **Empowers workforce** – Access to data empowers business users to improve analysis capabilities. The data warehouse should streamline the organisation by simplifying data flows, enabled by better access to shared data.
- **Speeds up response time to business queries** – They need faster response to business questions.
- **Data quality** – A consolidated data store that will eliminate the reconciliation of inconsistent data.
- **Document’s organisational knowledge** – A well-documented and centralized data store that will reduce the learning period that new employees will have to go through to understand the business.

### **3.14 Data Warehouse Design**

#### **3.14.1 Star Schema Database Design**

“The goal of a decision support database is often achieved using a database design called a star schema. A star schema design is a simple structure with relatively few tables and well-defined join paths. This database design, in contrast to the normalized structure used for transaction processing databases, provides fast query response time and a simple schema that is readily understood by the analysts and end users, even those who are not familiar with database structures.” (Poe, Klauer, & Brobst, 1998:191) (See glossary). (See appendix for more on star schema variations)

#### **3.14.2 The Benefits of a Star Schema**

The star schema is often used for data warehouse database design because it: (Poe, Klauer, & Brobst, 1998:192)

- Creates a database design providing fast response times.
- Allows database optimizers to work with a more simple database design in order to yield better execution plans.
- Parallels, in the database design, and how the end users think of and use data.
- Simplifies the understanding and navigation of meta-data for both developers and end users.

- Broadens the choice of front-end data access tools, as some products require a star schema design.

### 3.14.3 Limitations of the Star Schema

Although the star schema is a popular design for data warehousing, there are also cases where the star schema will not work well. The star schema was designed primarily for environments where the dimension tables are small compared to the fact tables. When dimensional tables become very large in their numbers of rows and attributes, the star schema will not fare well. This will impact on storage costs, usability and performance of the database design. The number of records in the dimension tables will become too large to effectively use the star joining technique, and the essence of data warehousing will be lost.

### 3.14.4 Fact and Dimension Tables

A star schema contains two types of tables, the fact tables and dimension tables. (Devlin 1997:236)

- **Fact tables**

Fact tables, sometimes called major tables, contain the quantitative or factual data about the business - the information being queried. This table normally contains numerical values and the table can consist of millions of rows.

- **Dimension tables**

Dimension tables are also called the minor tables, are smaller in size, and hold descriptive data that reflects the dimensions of business.

In a sales data warehouse for example the fact tables will contain data like sales revenue, sales quantity, net invoice sales, and all the primary and foreign keys that will link it to the dimensions. The dimension tables in this example will define the product information, customer information, time information and other information that the business needs.

### 3.14.5 De-normalization

De-normalization is the process of combining tables in a careful and thoughtful manner to improve performance. This is really breaking the rules of third normal form.(see 2.11 for more on normalization). The reasons for this are: (Poe, Klauer, & Brobst, 1998:208)

- To reduce the number of joins that must be processed in the average queries, thereby improving database performance.
- To map the physical database structure more closely to the user's dimensional business model. Structuring the tables along the lines of how the users will ask questions will provide the opportunity to tune the common access paths, and it will be much easier for users to understand the data.

An example of this is where the customer data were stored as a table for customers, and another table for customer type joined to the customer table. If this is de-normalized to form one table, the customer type information will now be included in the single customer table instead of having to join to a different table to get the data. This process will normally increase the query speed because of the join paths that have been reduced. The drawback of this is that the single customer table will use up more hard disk space than the combined space required for the two tables for customer and customer type. The reason for this is because some of the information "pieces" are stored repeatedly. (See Figure 3.2)

Customer Head Office		
H/O Code	Customer Name	Description
1	ABC Stores Gauteng	ABC Stores H/O Gauteng
2	AAA Stores Natal	AAA Stores H/O Natal
3	ABC Stores Western Cape	ABC Stores H/O Western Cape

Customer Branches				
H/O Code	Branch No	Branch Name	Description	Address
1	1	ABC Stores Sandton	ABC Stores Sandton Branch	174 Rivonia Ave
1	2	ABC Stores Germiston	ABC Stores Germiston Branch	91 Newton Rd
2	3	AAA Stores Durban	AAA Stores Durban Branch	78 Voortrekker Rd

a) Normalized

Customers						
H/O Code	Customer Name	H/O Description	Branch No.	Branch Name	Branch Description	Branch Address
1	ABC Stores Gauteng	ABC Stores H/O Gauteng	1	ABC Stores Sandton	ABC Stores Sandton Branch	174 Rivonia Ave
1	ABC Stores Gauteng	ABC Stores H/O Gauteng	2	ABC Stores Germiston	ABC Stores Germiston Branch	91 Newton Rd
2	AAA Stores Natal	AAA Stores H/O Natal	3	AAA Stores Durban	AAA Stores Durban Branch	78 Voortrekker Rd
3	ABC Stores Western Cape	ABC Stores H/O Western Cape				

b) De-Normalized (The two tables in Figure 3.2 a have been de-normalized into one table)

FIGURE 3.2: AN EXAMPLE OF A NORMALIZED AND DE-NORMALIZED CUSTOMER RECORD

### 3.15 Characteristics

Kelly (Kelly 1997:29) describes the main characteristics of a data warehouse as follows.

- **Separation of operational computing** - The data warehouse assumes a high-level conceptual architecture that separates query intensive decision support from transaction intensive transaction processing.

- **Extensible architecture** - There is no data warehouse architecture that can be applied to every enterprise. As the system grows the need for it to extend will become impossible to avoid. The data warehouse architecture does allow this extension to take place.
- **Scalable architecture** - The data warehouse will grow, through a number of iterations, from the relatively modest first prototype to potentially enormous volumes of data and number of users. Therefore the ability to scale hardware and software in small increments is important.
- **Two-phase project planning** - The data warehouse project is divided into two separate phases. In the first phase the primary concern is with the capture of data from different sources, and reaching consensus about the architectural principles. The second phase is concerned with the exploitation of data using a suite of data access tools.
- **Business driven** - Successful data warehouses are business driven and not technology driven.
- **Iterative approach** - The standard project planning methodology for the data warehouse will be based on an iterative or evolutionary approach. This will normally incorporate elements of rapid application development and the first iteration will usually result in a working prototype that is retained rather than discarded.
- **Quick delivery of benefits** - The data warehouse should deliver business benefits quickly. A 90-day standard has emerged as an outside limit to deliver the first iteration of a data warehouse implementation.
- **Systems integration** - The data warehouse architecture will incorporate a wide range of components, therefore care must be taken to ensure that all of the components will work together.
- **Skills transfer** - The data warehouse is for most organisations a step into the unknown and the failure rate for data warehouse projects is located in the 30-50% range. Most projects utilize specialist external assistance for architecture design and project methodology at the very least.

### **3.16 Chapter Conclusion**

This chapter discussed the origin of data warehousing, what data warehousing are and why we need them. Variations of data warehouses and the basic elements of data warehouses were also outlined.

The next chapter compares data warehouses and operational systems.

## **Chapter 4: Comparison between Operational Systems and Data Warehouses**

### **4.1 Chapter Objectives**

The objective of this chapter is to compare operational systems, discussed in Chapter 2, and data warehouses discussed in Chapter 3.

### **4.2 Operational vs Analytical Needs**

We need to distinguish between operational and analytical needs. “This fundamental distinction is between information technology that drives the business operations on a daily basis and those that determine the outcome of decisions about strategic moves in the market relating to customers, products and suppliers.” (Agosta, 2000:36).

The operational system is responsible for the operational needs of the organisation, and the data warehouse is responsible for the analysis of data and critical decision-making in the organisation. “When you hear people talking about operational processing, they are referring to the systems that run the day-to-day business of a company. These will often be on-line transaction systems, which are updated continually throughout the day.” (Poe, Klauer, & Brobst, 1998:3).

The data warehouse assists an organization to gain a competitive market advantage. “Analytical databases do not hold up-to-the-minute information but hold information as of a specific point in time. This makes perfect sense for a system that is providing information being used for comparisons and trends.” (Poe, Klauer, & Brobst, 1998:4).

### **4.3 Operational and Analytical Data**

The different goals of operational systems and data warehouses mean different data requirements for each system to achieve these goals. “Data warehouses need years of historical data so users can analyze business trends such as changing demographics or

sales. Operational systems need only enough historical information to support day-to-day delivery of products and services.” (Poe, Klauer, & Brobst, 1998:53).

According to (Devlin 1997:180) data can be classified as follows:

- **Transient Data** – The detail of each transaction replaces the previous one. Therefore, each record is available for capture only until the next transaction that changes that record occurs.
- **Periodic Data** – History of changes is maintained for a significant and known period.
- **Semi-periodic Data** – History of changes is maintained for only a short or undetermined time. For semi-periodic data, records are not overwritten, but superseded records may be discarded at any time. For example the operational system may retain only five statuses for each record. As in the case of transient data, it is impossible to predict how long any record will remain in the operational database.
- **Status Data** – This type of data captures information regarding the state of an entity. An example of this is the depositing of money into a bank account. If the account balance is R1000 and another R200 is deposited, then the status of the bank account at a certain time will change from R1000 to R1200.
- **Event Data** – This type of data is concerned with the event that took place.

We can now distinguish between operational and analytical data based on the above classification of data (Devlin 1997:180).

#### 4.3.1 Data in the Operational System

Data in the operational system can be classified even further into the following.

- **Transient or semi-periodic status data**

This category represents the majority of data in the operational system. Examples are:

- order and sales databases
- bank accounts
- customer files

- **Periodic status or event data**

This category of data is much less common than the previous type. It is found where operational systems must deal with historical data on a daily basis. The most common examples are from the insurance industry.

#### 4.3.2 Data in the Data Warehouse

The data in the data warehouse is by definition periodic and persistent over a time period determined by the business need. Most of the time, the data warehouse will store all the information that is available for history analysis purposes.

The data warehouse serves as a big advantage in this case since data can be removed from the operational system and stored in the data warehouse. If no data warehouse, or other method of archiving were available, the organisation would have no other choice than to store all the information in the operational system.

#### 4.4 Update Regularity

Since the operational system is responsible for day-to-day data needs of the organisation the data need to be updated on a regular basis to enable the system to add value to the organization

With data warehouses there is no need to update data every time something new happens. In fact it will be difficult if the data keeps on changing while analysts are analyzing the data. It is important that the users know when the data warehouse will be populated with new data to enable them to understand the changes in the information that might occur. The data warehouse update interval is dependent on the business needs.

#### 4.5 Availability

Because the operational system is updated online, and the fact that the day-to-day activities of the business depends on it, it is very important that this system is available when needed. Most operational systems must be available 24 hours a day and 7 days a

week, for example ATMs. Any maintenance on these systems must be well planned and communicated to users.

The availability of the data warehouse isn't as critical as is the case of operational systems. "Initial deployment of decision support capability is usually quite successful if the system is available for perhaps 12 hours a day for 6 days a week." (Poe, Klauer, & Brobst, 1998:56). This will give the data warehouse administrators enough time to perform updates and maintenance.

#### **4.6 Data Flexibility**

The main difference in data flexibility is the fact that the data warehouse provides users with the added functionality of "slicing and dicing". (See 3.9.6) Where the reports from the operational system are mostly static reports where the users can adjust the parameters, the data warehouse provides a complete flexible reporting environment. With the data warehouse the users can change the format in the way they want, they can request more or less detail and they can still specify the parameter values.

#### **4.7 Design Objectives**

It is expected from the data warehouse to provide a flexible read-only information source to its users. The design of the data warehouse is in sync with this objective, since it will provide a simple and flexible source of information if the star scheme (as discussed in 3.14.1) is implemented. "There are many examples of the trade-offs to be considered. If a database is de-normalized for a short-term performance benefits, there will be an offsetting loss in future flexibility." (Bischoff 1997:179).

The objective of the operational system is to provide a stable system that will be able to handle updates effectively and quickly.

#### **4.8 Requirements**

The development of a data warehouse is normally a process that must be well thought through to ensure success. The requirements of a data warehouse are normally very

vague and difficult to capture. There is a need for some innovation to make it successful. The requirements for an operation system are normally well known, since it must assist in the well-known process of running the business. The operational system is normally purchased from a vendor (discussed in 2.5) and implemented, where the data warehouse is a system that must be or will be developed for the specific organisation.

#### **4.9 Tuning**

The operational system will be tuned for frequent access to small amounts of data, and the data warehouse for infrequent access to large quantities of data. The data warehouse will also not have the need for frequent updates, since it is updated only at predefined intervals, but it must provide the users with fast response to undefined questions. The data warehouse will normally be indexed on different combinations of data to provide fast read access. It will not have any visible effect on end users during updates, since updates are performed in batches when users are not using the system.

“In an operational environment, information retrieval typically focuses on very well-defined and highly-tuned transactions that use indexed access to find a relatively small number of records (usually less than a few hundred). These transactions usually specify an individual customer, account, or other very selective retrieval criteria that make primary and foreign key indexing very effective.” (Poe, Klauer, & Brobst, 1998:59).

If the number of indexes in the operational system is high, it means that the data retrieval will be fast since the required records are located quickly, but updates will take more time since the indexes must be updated. The trade-off between these two requirements must be found. Since the types of data inserts and retrievals are the same on a daily basis in the operational system, it is easy to get this balance.

The data warehouse queries will normally be performed or include a number of records, where the operational system will only update small sets of data.

#### **4.10 Chapter Conclusion**

This chapter compared operational systems and data warehouses. In the next chapter these comparisons are used to design a conceptual model to combine these systems into one system.

## **Chapter 5: The Conceptual Model**

### **5.1 Chapter Objectives**

The objective of this chapter is to describe a conceptual model to replace the need for two specialised systems; the operational system and the data warehouse. The discussion will make use of an example of an operational system and the comparative data warehouse to build a single system.

### **5.2 Examples of Operational and Data Warehouse Systems**

This section describes simplified but actual examples of an operational system, and a data warehouse.

#### **5.2.1 Operational System**

The operational system in Figure 5.1 was taken from real Oracle applications, and simplified to a great extent. To simplify the discussion only two modules will be considered (OE – Order Entry, and AR – Accounts Receivable). These two modules were also simplified to include only the basic information.

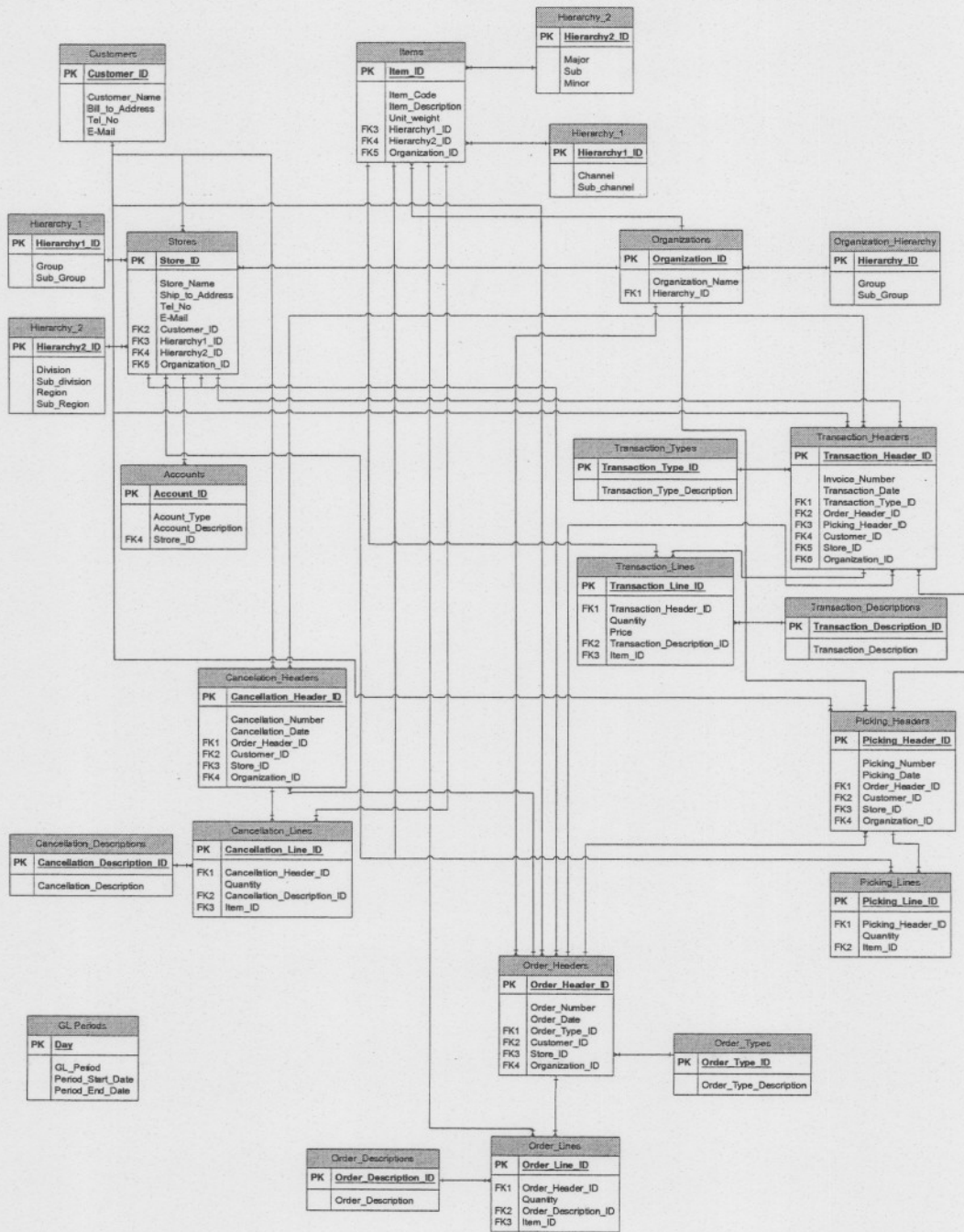


FIGURE 5.1: AN EXAMPLE OF AN OPERATIONAL SYSTEM

The operational system as shown in Figure 5.1 only includes information about the following entities:

- **Customers** – The tables that contain information about customers are Customers, Stores, Accounts, Hierarchy1 and Hierarchy2. The Customers table will contain the customer information at a high level, like head office references to Customers. This will normally be the customers who are responsible for paying the accounts. Pick ‘n Pay Gauteng is an example of such a customer. The next level of customer, the stores table, will have data regarding the actual store where the delivery must be made. This will be like Pick ‘n Pay Fourways Mall for example. It is possible for a customer to have multiple accounts in the operational system and the accounts table will contain this detail. The two hierarchy tables will not be called Hierarchy 1 and 2 in practice, but the actual hierarchy names. They are called Hierarchy 1 and 2 to show that these are tables used for classifying customers in certain views and hierarchies that the organisations wants to view them in. There can be a number of tables with this information but two are shown here.
- **Items** – Item information is stored in the tables Items, Hierarchy1 and Hierarchy2. The items master records are stored in the Items tables where the Hierarchy1 and Hierarchy2 are for the same purpose as discussed in the customer’s section.
- **Organisations** – Information is stored in the Organisation and Organisation\_Hierarchy tables. Again the master records are stored in the Organisation table and the hierarchy table is used for the classification of the organisations. An organisation in this context will be a distribution point from where stock will be shipped to various customers.
- **GL (General Ledger) Periods** – This table in Figure 5.1 isn’t linked to any other table. To show the relationships between this table and the other tables, another module (GL – General Ledger) must be included. This module wasn’t included in the discussion since it will complicate the design, and it will fall into the financial sector rather than sales (used in this discussion). The only table of GL, the GL periods table, cannot be excluded since it defines the financial periods of the organisation, and will be needed in the data warehouse design.

Note that this table is not an isolated table, but will be linked via different modules to the modules discussed here. Normally the sales information will also be carried over to the data warehouse only after the entries has been made into the GL, since the transaction is considered to be final at this stage. The entry into GL of a transaction will serve as a flag to notify the data warehouse that the information can be loaded from the other modules into the data warehouse.

- **Orders** – This data will be stored in the Order\_Header, Order\_Lines, Order\_Types and Order\_Descriptions tables. An order can be linked to only one customer and organisation, and can have only one date on it. This information will be stored only once per order. An order can have a number of items for each item that was ordered. The Order\_Lines table will be linked to the Items table for this information. This is a good example of normalization, since instead of storing the date, customer and organisation information for every item on the order, it is stored once in the order header table and then a different table is used for the more detailed information namely the Order\_Lines table. An order can have only one type, this will tell what type of order was placed. Every order line can also have a description linked to it, to provide information about the specific item and quantity on order.
- **Cancellations** – Cancellations are using almost the same tables as orders except for the Cancellation\_Type table. Information regarding cancellations is stores in the Cancellation\_Headers, Cancellation\_Lines and Cancellation\_Type tables. These tables serve the same purpose as for orders, but only for cancellation information. The information in the cancellation table will contain information about cancellation of orders.
- **Pickings** – After an order was completed successfully a pick slip will be provided for the dispatch. An order can result in multiple pick slips, since a pick slip will only be made out for stocks available. If the required amount of stock is not available, another pick slip will be provided for the remainder when the stock becomes available.

The information is stored in the tables `Picking_Headers` and `Picking_Lines`, and again the relationship between these tables server the same purpose as in the case of orders.

- **Invoices** – A pick slip will result in an invoice. The information regarding invoices is stored in the tables `Transaction_Headers`, `Transaction_Lines`, `Transaction_Types`, `Transaction_Descriptions`. These tables serve the same function as they do for orders. Credit notes are also taken into consideration in the Orders and Transaction tables, since this will be shown by a `Transaction_Type` entry of “credit note”, a `Transaction_Description` entry of the actual credit reason and a negative amount. This is also the case with credits in orders.

The areas discussed here can be linked back to the operational modules as follows. Orders, Cancellations and Picking is part of Order Entry (OE) and Invoices is part of the Accounts receivable (AR) module. The information regarding items, customers and organisations are shared among the different modules.

### 5.2.2 Data Warehouse

A possible data warehouse design that can be with the operational system in Figure 5.1, can be seen in Figure 5.2. The item, customer, periods and organisation tables have been de-normalized to form only one table for items, customers and organisations. The primary keys of each of the tables are stored in the single table. Note that there is a new key in the de-normalized table, and this will be used as the primary key of the single table for organisations, customers, items and periods. This is not necessary since a combination of the columns in the de-normalized table or the key of the normalized table can also guarantee uniqueness. This is however considered good practice to generate an surrogate key (see glossary), since this key will be used for linking to the other tables, and if the data in the data warehouse are extracted from multiple operational systems, it might happen that this “old “ key can appear in a different operational system resulting in a non-unique primary key. It is therefore important to generate a surrogate key in the data warehouse, where this key will be guaranteed to be unique.

The Customers, Items, Periods, and Organisations tables are the dimension tables in the data warehouse design in Figure 5.2. The tables for Transactions, Orders, Pickings and Cancellations are also de-normalized to form four tables. The primary keys of each of these tables are the combination of the foreign keys from the dimension tables. The tables Cancellations, Orders, Pickings and Sales in Figure 5.2 are the fact tables of the data warehouse (See 3.14.4 for more on fact and dimension tables).

The purpose of a data warehouse is to analyse data, and therefore it will normally store data at a higher level than an operational system. It will, for example, not be necessary for a data warehouse to provide answers to questions like “Which transaction resulted from the order x?”, but rather “How many tons did we sell for Jan 2001 of item y?”. The first question will be answered from the operational system, since this is not an analysis type of question. The second question will be answered by the data warehouse. The data warehouse will not have a direct link between the fact tables since these tables will normally be very large compared to the dimension tables. A query where these tables are linked together will take a long time to execute and might even result in a cartesian product (see glossary). Fact tables will be linked to each other via dimension tables. This problem will not result from the operational system, since the operational system will normally store data for only a fraction of the time, for example the operational system will store data for, say, three months whereas a data warehouse will store data for five years.

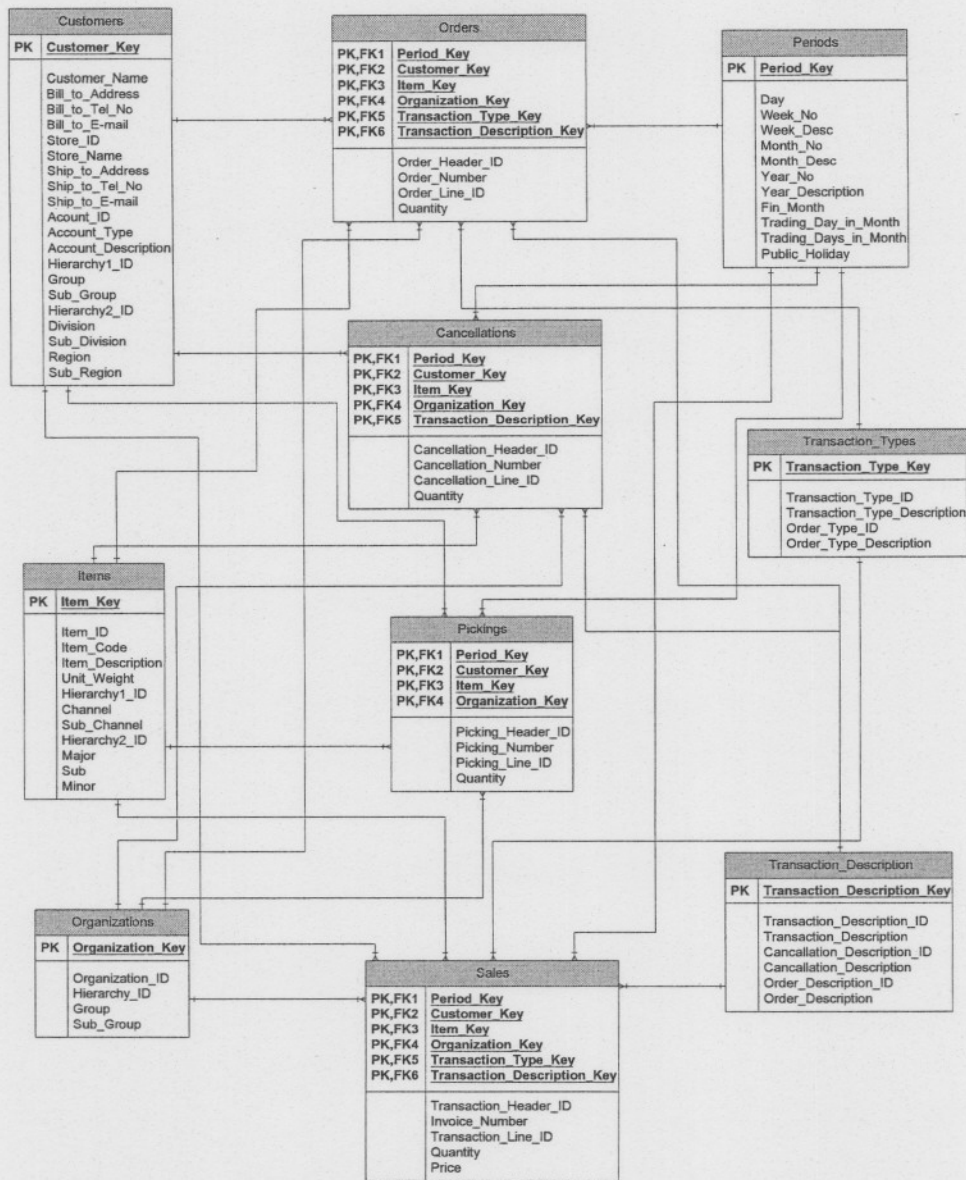


FIGURE 5.2: AN EXAMPLE OF A DATA WAREHOUSE SYSTEM

It might however happen that this question must be answered by the data warehouse to track the invoices to an orders link for a specific time in history. The join problem can be solved by storing the order number with the invoice number in the sales table (see Figure 5.3). This can also be done for pick slips.

The sales table will not grow extensively since it is only one extra field to store, and it is not necessary to join the order and sales tables. It is very important to capture all the requirements for the data warehouse in a document before development starts, since requirements like this can occur and result in constant changes to the data warehouse.

Sales	
PK	<u>Period Key</u>
PK	<u>Customer Key</u>
PK	<u>Item Key</u>
PK	<u>Organization Key</u>
PK	<u>Transaction Type Key</u>
PK	<u>Transaction Description Key</u>
	Transaction_Header_ID
	Invoice_Number
	Transaction_Line_ID
	Quantity
	Price
	Order_Number
	Picking_Number

**FIGURE 5.3: SALES FACT TABLE**

Note that the data warehouse design in Figure 5.2 is not the only design that can be used for the operational system in Figure 5.1, but that the data warehouse design is very flexible and depends on the requirements of the organisation. Figure 5.4 is a different data warehouse design where all four fact tables have been de-normalized to form only one fact table, and an extra dimension table (fact types) was added to classify the records as an order, sales, picking or cancellation fact. In Figure 5.2 there is a data mart (see 3.9.5 and 4.12) for every functional area (see glossary) namely orders, cancellations, pickings and transactions, and they share the shared dimensions. In Figure 5.4 there is only one data mart for transactions and the same dimensions connected to this. The number of data marts and dimensions is not fixed, and it will depend on the requirements of the business and what they see as sub-areas of their business.

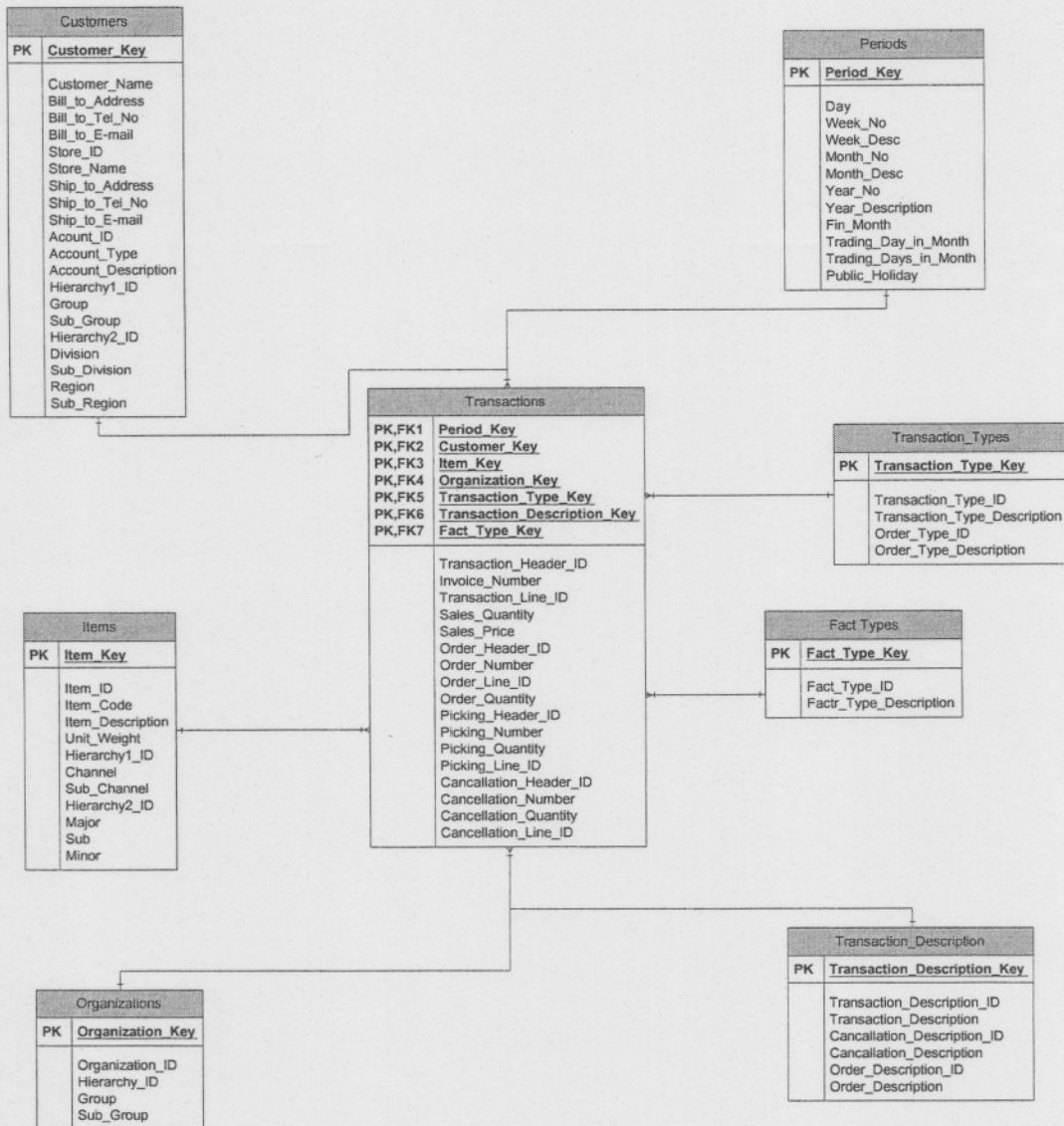


FIGURE 5.4: AN ALTERNATIVE DATA WAREHOUSE DESIGN

### 5.3 An Alternative to Operational and Data Warehouse Systems

The main objective of this study is to build a conceptual model of a system that will satisfy the needs provided by the operational systems and data warehouses. This single system needs to provide as many as possible of the advantages of both operational systems and data warehouses.

The disadvantages of operational and data warehouses must be minimized.

In Chapter 4 operational systems and data warehouses were compared. The following factors were discussed:

- Operational vs Analytical needs
- Update regularity
- Availability
- Data flexibility
- Design objectives
- Tuning
- Normalization
- Data history

We will describe the conceptual model by:

- using the above-mentioned factor in a single system environment, and
- an example of a single system based on the examples of an operational system and a data warehouse in Figure 5.1 and Figure 5.2 respectively.

### 5.3.1 Operational vs Analytical Needs

In an environment where both operational and data warehouse systems are available it is possible to design each system to perform specialised tasks. If the two systems are to be replaced by one system there will be compromises to make, but there will also be advantages from this. It is however possible to have both the operational data requirements and analytical data requirements in one system, since the analytical data is normally at a higher level than the operational data; a summary of the operational data. The drawback here will be speed for doing analysis, since this must be calculated every time, instead of being stored at this level.

### 5.3.2 Update Regularity

The conceptual model must again serve two roles and therefore it will be impossible to be updated once a day or even on an hourly basis. The data in the conceptual model will have to be updated online, since the information in the system must be immediately available for running the daily tasks of the business. There are some drawbacks since the data is always live, and there are no cut-off points as there was with the data warehouse. It might happen that a data analysis can report on data and within an hour the data can change, whereas with a data warehouse it is known that the data can only change next day.

The second drawback is the issue of traffic on the system. The system will now be used for capturing data, querying data for daily running and doing analysis, where with the two specialised systems the workload was divided.

### 5.3.3 Availability

Availability is also a crucial issue in such a system, since there is only one system that must be available at all times. With two systems it might be possible to take down the one system and get some of the information from the other. With the two separate systems the data warehouse doesn't have to be available all the time. Normally it will be available for about 8-12 working hours a day, where after it will go down and the data loads will start from the operational system. The operational system must be available for 24 hours a day as in the case of ATMs, but downtime can also be scheduled when the requirement for the system is low. If the two systems are to be replaced by only one system, it means that this single system will have to cope with more traffic and less downtime.

### 5.3.4 Data Flexibility

The single system will also have to provide the standard reports for the daily running of the business, and it must also provide a flexible interface for doing any ad-hoc queries and analyses of history. It is however possible to provide the flexible interface on the

operational system as well, but it will require more knowledge from the users doing it from the operational system than doing it from a data warehouse.

#### 5.3.5 Design Objectives

A single system will be designed to be simpler than the operational system and a bit more complex than the data warehouse, but it must still be easily understandable by its end-users. The complex design is normally associated with normalization (more tables and joins) where a simpler design is associated with de-normalization (less tables and joins). Both have its benefits and drawbacks, a compromise must be made to provide it in one system.

#### 5.3.6 Tuning

Operational systems are designed with the update purpose in mind, where more updates are being done than queries. This means that the indexes on such a system will normally be only on the primary keys, since the updates will enforce referential integrity on these keys. There is no need to cater for every possible query, since the reports are mostly static, and indexes can also be created on fields in these standard reports to speed them up. In the data warehouse it is possible for a user to do searches on any fields, and there will be no way of telling what type of query to expect next, therefore this system will be indexed on almost every possible field to speed up these queries. The updates in a data warehouse are also much slower because every time that an update is performed all the indexes must be updated. In the single system again more indexes will be added than are normally found in an operational system but less than in the data warehouse to give the best of both worlds.

#### 5.3.7 Normalization

In Section 4.11 normalization and de-normalization was discussed and the benefits it has in the specialised systems. In the single system it must be decided where speed is needed and where data storage must be kept to a minimum. Out of this it should be possible to see which tables to normalize and which tables to de-normalize.

### 5.3.8 Data History

In the operational system the history data are kept to the minimum that is required to run the daily operation, normally about three months to be on the safe side. This will have huge speed advantages on this system for reports and the normal running of the system. Note that the operational system not only performs queries when reports are run from the system, but also for lookups to tables to retrieve possible values for fields in data entry windows. These are static queries however. The data warehouse on the other hand will normally store all the available information for analysis purposes. To replace both these systems with a single system there will be no choice to store all the data, unless some rule can be put into place where only data for three years are kept. This will have a negative impact on the performance of such a system since every look-up must search through a table containing data for five years instead of three months.

### 5.4 The Conceptual Model: An example

In the previous section some requirements of a single system were discussed, which will highlight the critical points that the design of such a system must cater for. Some of the topics in the previous section will not have any influence on this design but rather on the way the system will be used. The design in Figure 5.5 is an example of such a single system that can serve as an alternative to the two specialised systems.

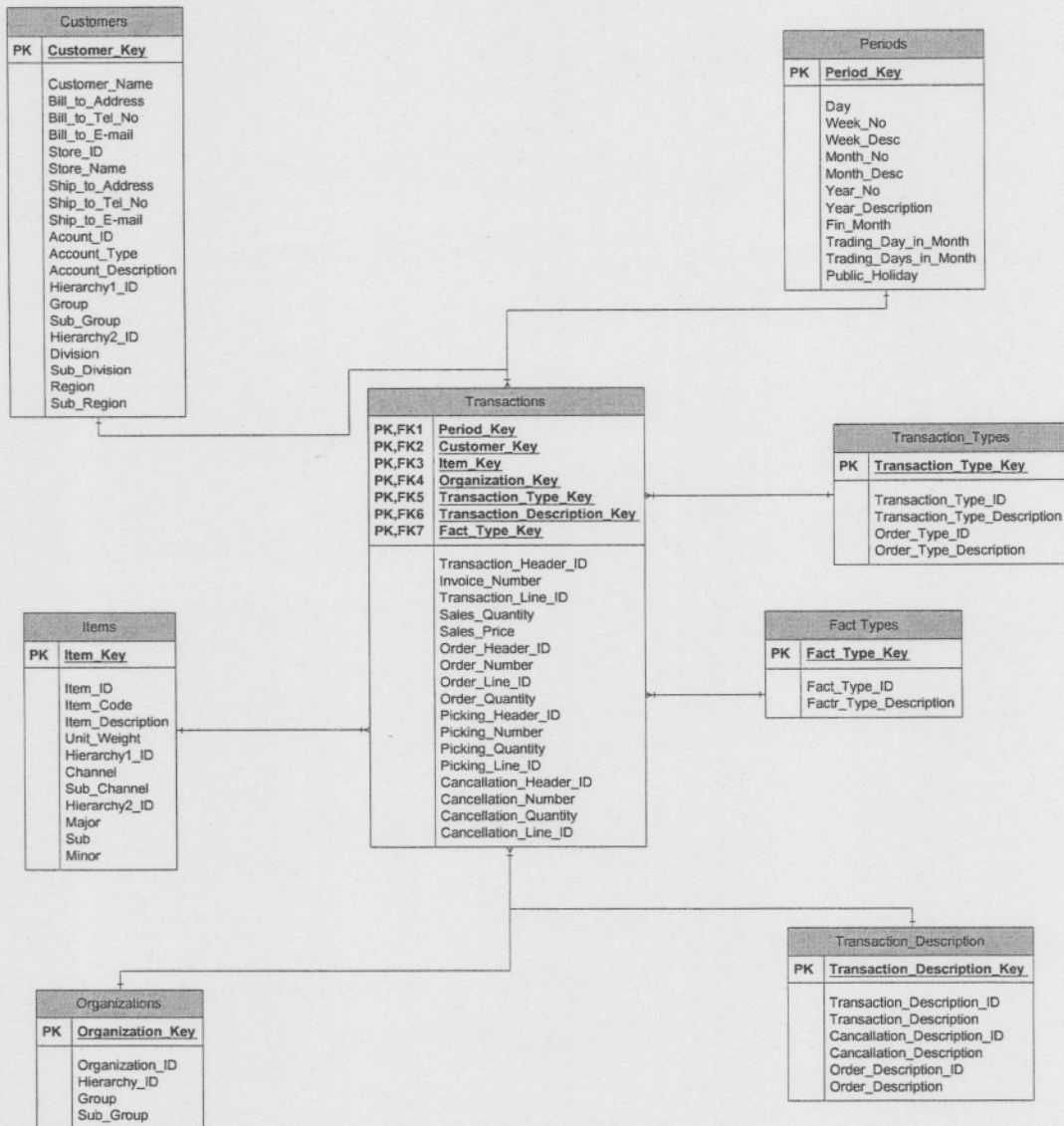


FIGURE 5.4: AN ALTERNATIVE DATA WAREHOUSE DESIGN

### 5.3 An Alternative to Operational and Data Warehouse Systems

The main objective of this study is to build a conceptual model of a system that will satisfy the needs provided by the operational systems and data warehouses. This single system needs to provide as many as possible of the advantages of both operational systems and data warehouses.

Figure 5.5 shows that the dimension tables are the same as they were in the data warehouse design. The reason for this is because the de-normalization will not have a huge impact on the performance or storage space when the number of data records are small as in the case of dimension tables. De-normalization of these tables will mean that more storage will be needed for the same information, and performance of updates will be slower because every repeating data group, like customer hierarchies, must be updated for every record. Performance for queries on these tables will be fast because there are no joins.

The fact tables are normalized and resemble the same design as in the operational system except for the transaction types information that are stored in a single dimension table. The fact table store much more information than the dimension tables. The fact tables will grow for every transaction where the dimension tables will stay exactly the same for most of the time. Dimension tables will only change in cases where say, a new customer starts doing business with the organisation or a new item is introduced into the market. The fact tables are the critical part where the conceptual model can gain many improvements. The reason for the normalization of the fact tables is that the storage space required is at a minimum since the header information is stored only once per transaction and not for every transaction as was the case in the data warehouse. This will have a big impact if the large number of records that fact tables store are taken into account. The keys in the header tables, Order\_Header\_Key, Cancellation\_Header\_Key, Picking\_Header\_Key and Transaction\_Header\_Key are keys generated to link the detail tables, and these keys will guarantee uniqueness of the header table lines, and they will serve as surrogate keys on top of all the foreign keys in the header tables. Updates will be faster since the header information is updated only once. The queries on the fact tables will however be slower than they were for the data warehouse, since this alternative system will store information for all history that there is for the organisation, and join paths must be resolved to retrieve information.

Note that there will still be less joins in this system than there were in the operational system to get the required information, but more data to search, so the time for queries should be about the same.

Another advantage is that the queries for the normal daily operation will be fixed, and the database can be indexed to speed this up. Only analysis type queries will be a disadvantage compared to the data warehouse.

## **Chapter 6: Summary and Conclusion**

### **6.1 Chapter Objectives**

In the previous section the conceptual model was shown and the design was discussed, the question that now comes to mind is if there will be a need somewhere for such a system. Let's start off by looking at the advantages and disadvantages of the single system compared to two specialised systems, and then decide whether there is a market for this type of system.

### **6.2 Advantages of the Conceptual Model**

- The biggest advantages will be costs, since only one system is needed and thus only one server is needed to host the system and also less people are needed to maintain the system.
- No issues regarding the balancing of the one system to the other. - If there is only one system, it will be trusted.

### **6.3 Disadvantages of the Conceptual Model**

- Dependence on the single system: if this system fails, there is nothing to fall back on.
- The system specializes in no single department, thus some of the functionality of the specialised system is lost in the single system. The system must be designed to serve two roles and are bound to do a fair job of both rather than an excellent job of each, as is the case with the two specialised systems.
- Double the traffic on one system.
- All the history data in one system. This will slow down the system and will not be acceptable if the organisation is having to cope with huge numbers of transactions, such as ATMs.
- Downtime on this system will be difficult to schedule since it must serve two roles and the demand for availability of the system will be high.

The advantages and disadvantages give some guidelines as to what type of market this single system can be used in. This single system will typically excel in a small to

medium size environment where the amount of data to be processed are relatively small and thus the speed vs size issue will not have such a big impact as in an environment where that is critical. The system will also have to cope with about double the amount of queries and this will not be good enough for organisations where speed is critical, and again the feeling here is that the two specialised environments are by far better in providing the best solution if the organisation can afford it. The single system does however have a market where the need for performance isn't that critical, but the costs of the system must be kept down. Typically the big organisations that can afford the specialised solutions will not even have a second look at the single system, since fair performance is not good enough, and they can afford the extra costs for servers and disks. The single system will come into play for small to medium organisations, and that will be the target market for this type of system.

The single system should in general consume more space for data storage than the operational system, but less than the data warehouse because of the normalization used on the fact and dimension tables. The less space required for data storage than the data warehouse is because of the normalization of the fact tables. Performance of the single system will be slower for updates than the operational system but faster than the data warehouse, since less indexes must be updated in the single system than in the data warehouse but more than in the operational system.

Data extraction for daily running of the business will be slower than from the operational system since all the data is kept in one system, and data queries for analysis will be slower than it is from the data warehouse since joins must be resolved and data must be rolled up to the required level.

## **6.4 Limitations and Future Research**

### **6.4.1 Limitations**

- This conceptual model was based on a single example. Such a system must be tested on many more examples to make a decision on the benefits of such a conceptual model.

- This is a conceptual model only. It still needs to be implemented and tested.

#### 6.4.2 Future Research

- To implement such a conceptual model.
- Tests must be done over an extended period of time to determine if the advantages of such a system are viable, and if the disadvantages are at a minimum.

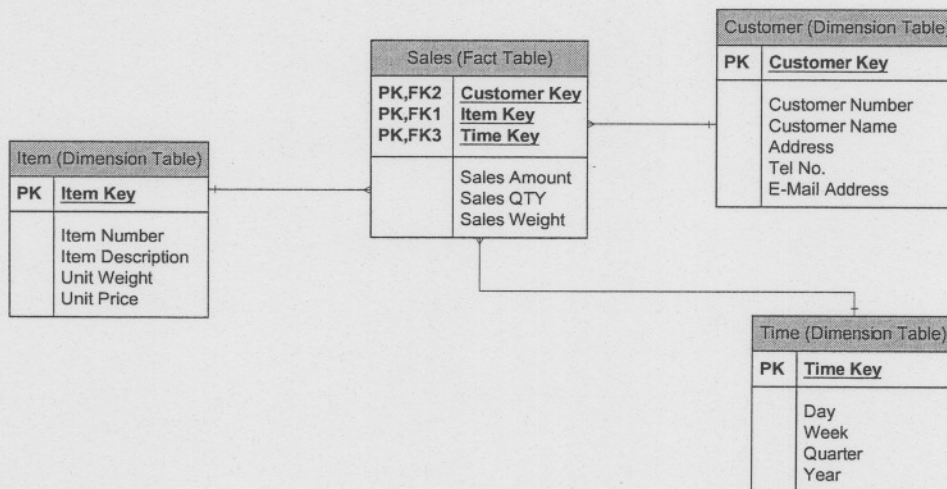
## Annexures

### 7.1 Varieties of Star Schemas

#### 7.1.1 Simple Star Schema

In the simple star schema design, each table (facts and dimensions) will have a primary key, which is a column or a group of columns whose contents uniquely identify each row. In a simple star schema, the primary key for the fact table is composed of one or more foreign keys; a foreign key is a column in one table whose values are defined by the primary key of another table. This type of design will consist of only 1 fact table with a number of dimensions connected to it. Notice the many-to-one relationships between the foreign keys in the fact table and the primary keys they reference in the dimension tables.

In Figure 7.1 the item table defines the items. Each row in the item table represents a distinct item and has a unique item identifier (primary key of the item table). The item identifier can occur multiple times in the sales table, representing sales of that particular item during a period for a specific customer.

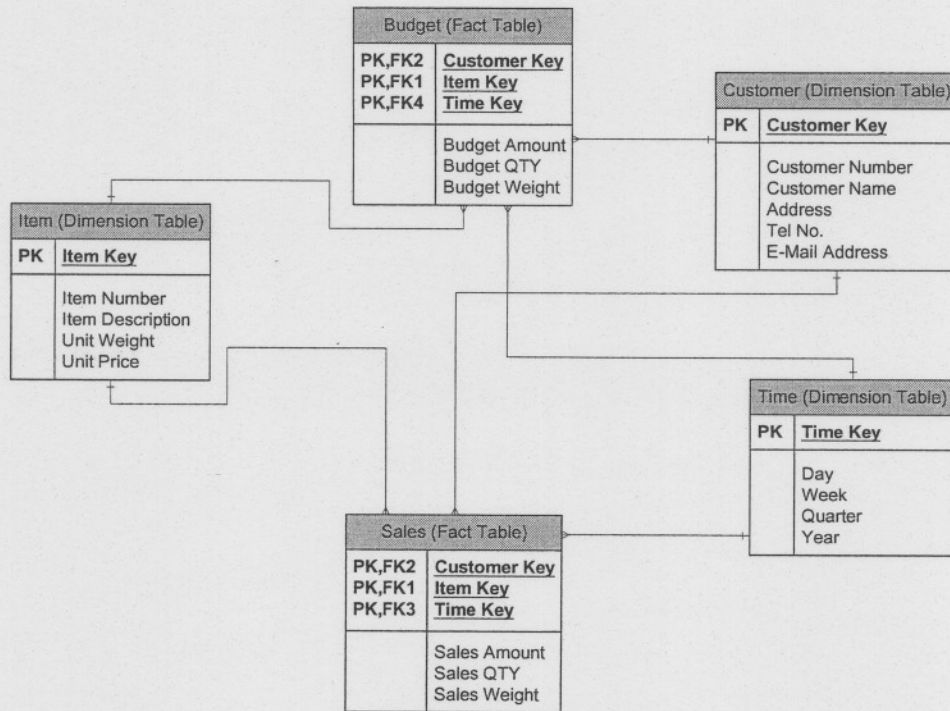


**FIGURE 7.1: A SIMPLE STAR SCHEMA**

7.1.2 Multiple Fact Tables

More sophisticated database implementations may require the use of more than one fact table. It might be necessary to store information regarding sales and budgets for items and customers for certain periods. The granularity of the budget table and sales table doesn't have to be on the same level. It is also possible to include a sales fact table that is on a daily granular level and a sales fact table that has been aggregated to a monthly level in the same star schema.

This will make faster querying possible if the sales are needed on a monthly level, because the aggregation or roll-up has been performed in advance.



**FIGURE 7.2: A STAR SCHEMA WITH TWO FACT TABLES**

Another use of a fact table in a star schema is as an associative table to resolve a many-to-many relationship between items and item groups. See Figure 7.3

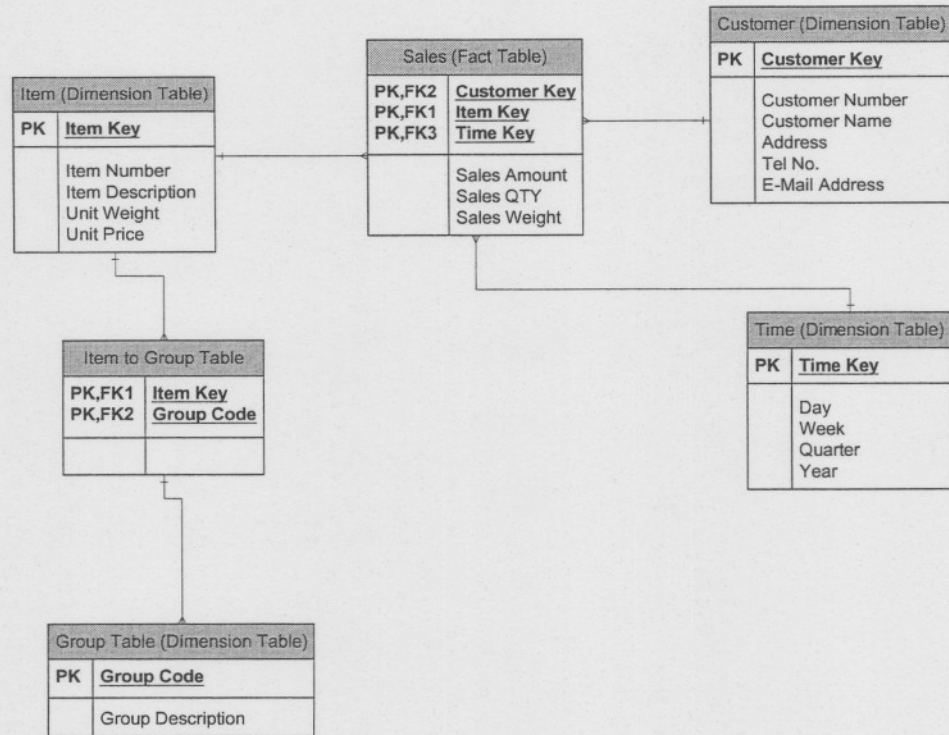


FIGURE 7.3: A FACT TABLE AS AN ASSOCIATIVE TABLE

### 7.1.3 Outboard Tables

A dimension table can also contain a foreign key that references the primary key in another dimension table. The referenced dimension table is sometimes referred to as a secondary dimension table. Outboard tables may also be chained together to provide a hierarchy of dimension tables organized in a more normalized database design.

A normalized design reduces the size of the dimension tables, but also reduces the performance and usability advantages derived from a pure star schema. In Figure 7.4 the “Brand table” and “Group table” are secondary dimension tables.

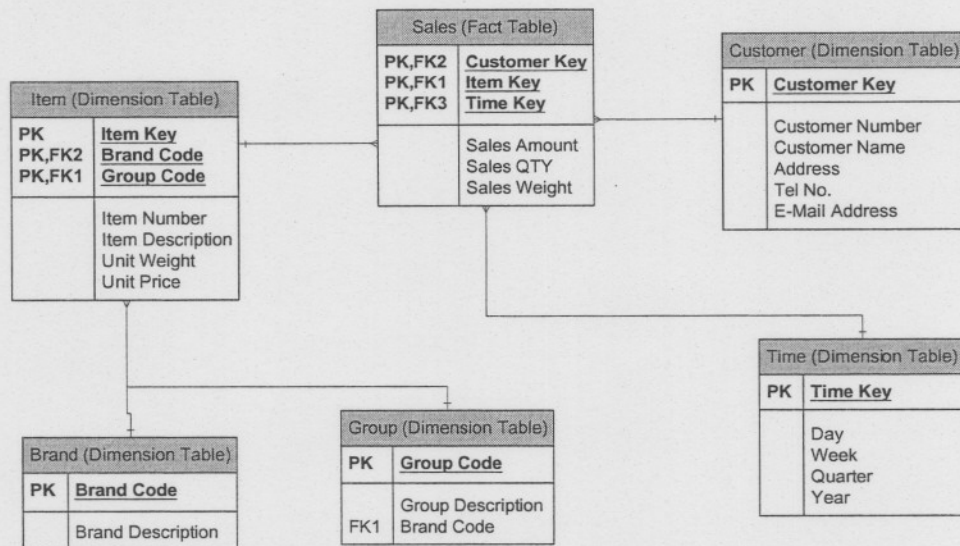


FIGURE 7.4: AN EXAMPLE OF SECONDARY DIMENSION TABLES

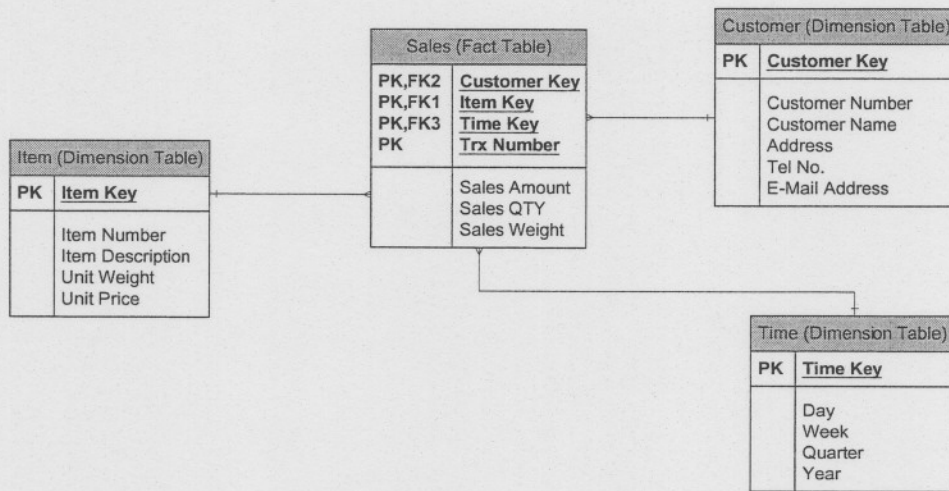
One variation of a star schema is to store all dimensional information in third normal form, while keeping fact tables structures the same. This type of star schema is referred to as a snowflake schema. There are two common reasons for adopting this:

- the emergence of advanced decision support tools that can fully exploit this type of structure and
- many information systems organisations feel more comfortable with a design in third normal form.

It should be kept in mind that the number of tables should be kept to a minimum to avoid user confusion.

### 7.1.4 Multi-Star Schema

In a multi-star schema, the foreign key columns that relate to dimension tables in the database design define the primary key in the fact table. In some applications it might be possible that the dimension table foreign keys don't provide a primary key in the fact table. These applications require a multi-star schema. In a multi-star schema, the fact table has both a set of foreign keys, which reference the dimension tables, and a primary key, which is composed of one or more columns that provide a unique identifier for each row. See Figure 7.5



**FIGURE 7.5: AN EXAMPLE OF A MULTI-STAR SCHEMA**

## **Bibliography**

Agosta, L. 1999. The Essential Guide to Data Warehousing. USA: Prentice Hall International. 499p.

Barquin R. & Edelstein H. 1997. Planning and Designing the Data Warehouse. USA: Prentice Hall, Inc. 311p.

Bishoff, J. Alexander, T. 1997. Data Warehouse – Practical Advice from the Experts. USA: Prentice-Hall Inc. 428p.

Brady, J A. Monk E F. Wagner, B J. 2001. Concepts in Enterprise Resource Planning Systems. Canada: Thomson Learning. 164p.

Buhrman, A. Van der Riet, T. Govender, T. Holmes, B. & Wilder, P. 1999. ERP – What is it? *Business Technology*, 4(6): 14-16 July.

Davenport, T. H. 1998. Putting the Enterprise into the Enterprise System. *Harvard Business Review*. Vol. 76.(9) 4:121-131. Jul – Aug.

Devlin, B. 1997. Data Warehouse from Architecture to Implementation. Addison Wesley Longman, Inc. 432 p.

Kelly, S. 1997. Data Warehousing in Action. England: John Wiley & Sons Ltd. 320 p.

Kimball, R. 1995 (A). The Data Market Splits. *Data Warehouse Architect*. Sept.

Kimball, R., Reeves, L., Ross, M. & Thornthwaite, W. 1998. The Data Warehouse Lifecycle Toolkit – Experts’ Methods for Designing, Developing, and Deploying Data Warehouses. USA: John Wiley & Sons, Inc. 771 p.

Mcfadden, F. R. & Hoffer, J. A. 1994. Modern Database Management. 4th ed. Redwood City: The Benjamin/Cummings Publishing Company, Inc. 597 p

Mohammed, A.H. 2000. Implementation of Enterprise Resource Planning Model Based Solution in Medium-Sized South African Companies. Potchefstroom:University of Potchefstroom. 107 p.

Norris, C., Hurley, J. R. Hartley, K. M., Dunleavy, J. R. & Balls. J. 2000. E-Business and ERP – Transforming the Enterprise. USA: John Wiley & Sons Inc. 194 p.

O’Leary, D. E. 2000. Enterprise Resource Planning Systems – Systems, Life Cycle, Electronic Commerce and Risk. Cambridge University Press: USA. 227 p.

Poe, V., Klauer, P. & Brobst, S. 1998. Building a Data Warehouse for Decision Support. 2<sup>nd</sup> ed. USA: Prentice Hall, Inc. 285 p.

Shankarnarayanan, S. 1999. ERP Systems – Using IT to gain a competitive advantage [Web] <http://www.whatis.com> [Date of access 1 Oct 2002]

Zeleny, M. 2000. International Encyclopedia of Business & Management – The Handbook of Information Technology in Business. UK: Thomson Learning. 870 p.

**Other Material Used:**

Adamson, C., Venerable, M. Kimball, R. 1998. Data Warehouse Design Solutions. USA: John Wiley & Sons. 523p.

Brackett, M H. 1996. *The Data Warehouse Challenge – Taming Data Chaos*. USA: John Wiley & Sons, Inc. 579p.

Browne, J., Harhen, J. & Shivnam, J. 1996. *Production Management Systems – an Integrated Perspective 2<sup>nd</sup>*. Great Britain: Addison-Wesley Publishers Ltd. 425p.

Callegari, N. 2002. ERP and what it means to your business enterprise. *Computing SA*. Vol. 22 (12): 26.8 Apr.

Callegari, N. 2002. Advancing ERP. *Computing SA*. Vol. 22. (12): 28.8 Apr.

Callegari, N. 2002. Mature ERP sites gearing up to leverage investments. *Computing SA*. Vol. 22 (12): 28. 8 Apr.

Callegari, N. 2002. Dirty data hinders attempts to extract value from ERP. *Computing SA*. Vol. 22 (12): 29. 8 Apr.

Chunng, Y. Kim, Park, J-W. & Lee, K. 2001. Performance evaluation for CC\_NUMA multiprocessors using an OLTP workload. *Microprocessors and Microsystems* . 221-229. 25 Apr.

Cimino, D. & Dash, J. 1996. From OLTP to ITP. *Software Magazine*, Vol 16, (12): . 14. Dec.

Claybrook, B. 1992. *OLTP Online Transaction Processing Systems*. John Wiley & Sons Inc, USA 358 p.

Cox, J. 1995. HP boosts distributed OLTP. *Communications Week*, 541: 16. 1 p 30 Jan.

English, L.P. 1999. *Improving Data Warehouse and Business Information Quality - Methods for Reducing Costs and Increasing Profits*. USA: John Wiley & Sons, Inc. 518 p.

Fazlollahi, B. & Vahidov, R. 2001. A method for generation of alternatives by Decision Support systems. *Journal of Management Information Systems*, Volume 18,. (2): 229-250.

Franck, B. J. 1999. *An Analysis of the Failure of Enterprise Resource Planning Systems to Fulfil the Requirements of Good Value Chain Management in Three Case Studies*. Cape Town. MBA Dissertation. 66 p.

Girard, K. 2000. Back-Office Rebirth – Enterprise Resource Planning companies bite the bullet and count the corpses. *Intelligence*: . 28-29. Aug.

Goldsborough, R. 1994. *Straight Talk about the Information Superhighway*. USA. (publisher? Publication?) 340 p

Gower, M. 2001. ERP facilitates planning in changing environment. *Computing SA*. Vol. 21 (29): 18. 6 Aug.

Hammond, M. 1998. Informix eyes warehousing. *PC Week*, Volume 15 (42): 68, 1 p. 19 Aug.

Heske, P. 2001. ERP – Vendors and customers face new challenges. *Computing SA*. Vol. 21 (23): 23-24. 25 Jun.

Heske, P. 2001. Informed decisions key to Data Warehouse success. *Computing SA*. Vol. 21 (28): 25. 30 Jul.

Heske, P. 2001. Business needs-planning underpins successful Data Warehousing. *Computing SA*. Vol. 21 (28): 26. 30 Jul.

Heske, P. 2001. Data Warehousing bridges e-gap. *Computing SA*. Vol. 21 (28): 28. 30 Jul.

Jacobs, R. F. & Whybark, C. D. 2000. Why ERP? – A Primer on SAP Implementation. USA: The McGraw-Hill Companies Inc. 127 p.

Kimball, R. 1995 (B). Is ER modeling hazardous to DSS ? *Data Warehouse Architect*. Oct.

Kimball, R. 1998. The Operational Data Warehouse. *Data Warehouse Architect*. Jan.

Kimball, R. 2000. Many alternate realities. *Intelligent Enterprise*. 9 Feb.

Kimball, R. & Inmon, W.H. 1996. The Data Warehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons Inc. USA: 388 p.

Mabert, V. A., Soni, A. & Venkataramanan, M. A. 2001. Enterprise Resource Planning: common myths versus evolving reality. *Business Horizons*, , Vol. 44 (3): 69 – 77 May/Jun.

Mabert, V. A., Soni, A. & Venkataramanan M.A. 2001. Enterprise Resource Planning: measuring value. I 46 – 51. Third/Fourth Quarter.

Mallach, E. G. 1994. Understanding Decision Support Systems and Expert Systems. USA: R.R. Donnelley & Sons Company. 695 p.

Marshall, M. 1996 ....and integrates OLTP with the web. *Communications Week* (616): 12, 1 p. 24 Jun.

McNurlin, B. 2001. Will users of ERP stay satisfied? I, Vol. 42 (2):13. Winter.

Mitt, L. M., Wu, D. J. & Zhou, X. 2002. Investment in Enterprise Resource Planning: business impact and productivity measures. *Journal of Information Systems* Vol. 19 (1): 71-98. Summer.

Mouton, J. 2002. How to Succeed in Your Master's & Doctoral Studies – A South African Guide and Resource Book. Van Schaik Publishers: South Africa. 280 p

Nikolaou, C. N., Marazakis, M. & Georgiannakis, G. 1997. Transaction routing for distributed OLTP systems: survey and recent results. *Information and Computer Science* 97. 45-82.

O'Mara, T. 1998. A new-generation approach to OLTP. *ENT*, Volume 3 (2): 46 -48. 2 Apr.

Quaddus, M. & Intrapairot, A. 2001. Management Policies and the diffusion of data warehouse: a case study using system dynamics-based decision support system. *Decision Support Systems* Volume 31 (2): 223-230. Jun.

Raden, N. 1996. Maximizing your warehouse. *InformationWeek* (571): 42, 6 p. 18 Mar.

Ragowsky, A. & Somers, T. M. 2002. Enterprise Resource Planning. *Journal of Management Information Systems* Vol. 19 (1): 11-15.

Scala, G. 2000. New ERP solution a major hit for Information Systems. *Computing SA*. Vol. 20 (5): 8. 7 Feb.

Shtub, A. 1999. Enterprise Resource Planning (ERP) – The dynamics of Operations Management. USA: Kluwer Academic Publishers. 146 p

Umble, M., Umble, E. & Von Deylen, L. 2001. Integrating Enterprise Resource Planning and theory of constraints: A case study. *Production and Inventory Management Journal* 43 – 48 . Second quarter.

USX Engineers & Consultants.2000. Picking the right ERP can make e-commerce click. *33 Metalproducing* Vol. 38 (7): 18.

Van Schalkwyk, 2000. The anatomy of a successful ERP project – make or break. *African Connection – Year 15* First Edition. 95.

Willet, S. & Moltzen, E. F. 1998. HP servers aim at Data Warehousing, OLTP arenas. *Computer Reseller News* (822): 32, 1 p. 21 –28 Dec.

