A CASE STUDY IN WEATHER PATTERN SEARCHING USING A SPATIAL DATA WAREHOUSE MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÇAĞLAR KÖYLÜ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

JUNE 2008

Approval of the thesis:

A CASE STUDY IN WEATHER PATTERN SEARCHING USING A SPATIAL DATA WAREHOUSE MODEL

submitted by **ÇAĞLAR KÖYLÜ** in partial fulfillment of the requirements for the degree of **Master of Science in Geodetic and Geographic Information Technologies Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Assoc. Prof. Dr. Şebnem Düzgün _____
Head of Department,
Geodetic and Geographic Information Technologies

Assoc. Prof. Dr. S. Zuhal Akyürek _____
Supervisor, **Civil Engineering, METU**

Exammining Committee Members:

Assoc. Prof. Dr. Nihan Kesim Çiçekli _____
Computer Engineering, METU

Assoc. Prof. Dr. S. Zuhal Akyürek _____
Civil Engineering, METU

Assoc. Prof. Dr. İnci Batmaz _____
Statistics, METU

Tuncay Küçükpehlivan _____
Technical Manager, Başar Computer Systems

Selami Yıldırım _____
Specialist, Devlet Meteoroloji İşleri Gn. Müd.

Date: 09.06.2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Çağlar Köylü

Signature :

ABSTRACT

A CASE STUDY IN WEATHER PATTERN SEARCHING
USING A SPATIAL DATA WAREHOUSE MODEL

Köylü, Çağlar
M.S., Department of Geodetic and Geographic Information Technologies
Supervisor: Assoc. Prof. Dr. S. Zuhal Akyürek

June 2008, 102 pages

Data warehousing and Online Analytical Processing (OLAP) technology has been used to access, visualize and analyze multidimensional, aggregated, and summarized data. Large part of data contains spatial components. Thus, these spatial components convey valuable information and must be included in exploration and analysis phases of a spatial decision support system (SDSS). On the other hand, Geographic Information Systems (GISs) provide a wide range of tools to analyze spatial phenomena and therefore must be included in the analysis phases of a decision support system (DSS). In this regard, this study aims to search for answers to the problem how to design a spatially enabled data warehouse architecture in order to support spatio-temporal data analysis and exploration of multidimensional data. Consequently, in this study, the concepts of OLAP and GISs are synthesized in an integrated fashion to maximize the benefits generated from the strengths of both systems by building a spatial data warehouse model. In this context, a multidimensional spatio-temporal data model is proposed as a result of this synthesis. This model addresses the integration problem of spatial, non-spatial and temporal data and facilitates spatial data exploration and analysis. The model is evaluated by implementing a case study in weather pattern searching.

Keywords: Spatial data warehousing, Spatial Decision Support System (SDSS), spatio-temporal exploration and analysis, Geographic Information Systems (GISs), Online Analytical Processing (OLAP), multidimensional modeling.

ÖZ

MEKANSAL VERİ AMBAR MODELİ KULLANARAK
HAVA DESEN ARAŞTIRMASI

Köylü, Çağlar
Yüksek Lisans, Jeodezi ve Coğrafi Bilgi Teknolojileri Bölümü
Tez Yöneticisi: Doç. Dr. S. Zuhal Akyürek

Haziran 2008, 102 sayfa

Çok boyutlu, kümelenmiş ve özetlenmiş verilere erişim, görselleştirme ve analiz amacıyla veri ambarı ve Çevrimiçi Analitik İşleme (OLAP) teknolojileri kullanılmaktadır. Bu verilerin büyük kısmı mekansal bileşen içermektedir. Bu sebeple, tüm bu mekansal bileşenler taşıdıkları değerli bilgiler gereği mekansal karar destek sistemlerinin inceleme ve analiz safhalarında kullanılmalıdır. Öte yandan, Coğrafi Bilgi Sistemleri (CBS) mekansal olguların analizi için çok çeşitli araçlar sunmaktadır ve bu yüzden karar destek sistemlerinin analiz safhalarında yer almalıdır. Bu bakımdan, bu çalışma çok boyutlu verilerin mekan uzamsal inceleme ve analizini gerçekleştirecek mekansal etkinlik kazandırılmış veri ambarı mimarisini nasıl kurulabileceği ile ilgili soruna cevaplar aramayı amaçlamaktadır. Dolayısıyla, bu çalışmada OLAP ve CBS kavramları tümleşik biçimde, her iki sistemin de etkin yönlerinden azami derecede faydalanma amacıyla mekansal veri ambarı modeli oluşturularak sentezlenir. Bu bağlamda, sentezin bir sonucu olarak çok boyutlu mekan uzamsal bir veri modeli önerilmektedir. Bu model mekansal, mekansal olmayan ve zamansal verilerin bütünleştirme sorununa çözüm önermekte ve mekansal veri inceleme ve analiz safhalarının kolaylaştırılmasını sağlamaktadır. Model, hava desen araştırması üzerine yapılan örnek bir çalışma ile değerlendirilmektedir.

Anahtar kelimeler: Mekansal veri ambarcılığı, Mekansal Karar Destek Sistemi, mekan uzamsal inceleme ve analiz, Coğrafi Bilgi Sistemleri, Çevrimiçi Analitik İşleme, çok boyutlu modelleme.

To Süha, Hürriyet, Bingo and Gamze who are always with me.

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my supervisor, Assoc. Prof. Dr. S. Zuhal Akyürek, for her guidance, advice, criticism, encouragement, and insight throught the research.

I would like to thank to my friends, Gamze Köseoğlu, Hikmet Demirkol, Balkan Uraz, Bayramhan Boyer, Melike Kazanç, Filiz Kuzu, Utku Serkan Zengin, Sanem Yumurtacı, Lale Pekel for all their technical and moral support. My special thanks go to Hüseyin Akyol for motivating me throughout my studies.

I want to thank to GGIT assistants, and particularly to Serkan Kemeç. Thanks also go to my friend, Seda Ünal Çalargün for her technical support.

I am greatly indebted to my family, Süha, Hürriyet Köylü and Bingo, for their unconditional support and encouragement to pursue my interests. I want to dedicate this study particularly to my grandfathers Sebahattin Köylü and İsmail Erdemir for their belief in my abilities.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

API:	Application Programming Interface
DBMS:	Database Management System
DM:	Data Mining
DS:	Decision Support
DSS:	Decision Support System
GIS:	Geographic Information System
HOLAP:	Hybrid Online Analytical Processing
KDD:	Knowledge Discovery in Databases
MDDB:	Multidimensional Database
MDDBS:	Multidimensional Database System
MDX:	Multidimensional Expressions
MOLAP:	Multidimensional Online Analytical Processing
OLAP:	Online Analytical Processing
OLTP:	Online Transactional Processing
ROLAP:	Relational Online Analytical Processing
SDSS:	Spatial Decision Support System
SQL:	Structured Query Language
STEA:	Spatio Temporal Exploration and Analysis
UDM:	Unified Dimensional Model

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Decision Support (DS) is a general term which is used in a variety of context related to decision-making. Simon (1960) suggests that decision-making process can be structured as three major phases. The first phase is called “Intelligence” which is based on finding occasions for making a decision. The second phase is called “Design” and involves inventing, developing, and analyzing possible alternatives. Lastly, the third phase is called “Choice” and involves choosing a particular alternative from those available. The relationship between these three major phases of decision making is illustrated in Figure 1.1. As seen in the figure, implementation of these three phases is a routine, repetitive process and it is possible to redefine them after evaluating each phase.

Problems and opportunities are identified in “Intelligence Phase”, decision process tasks are developed in “Design Phase”, and decisions are made in “Choice Phase”. The concept of DS covers all these phases and cycling relationships between them. In this context, a decision support system (DSS) is defined as “an interactive computer-based system whose purpose is to help decision makers perform analysis oriented operations to identify and solve problems, complete decision process tasks, and make decisions within the cycle of three phases of decision-making” (http://www.dssresources.com/glossary).

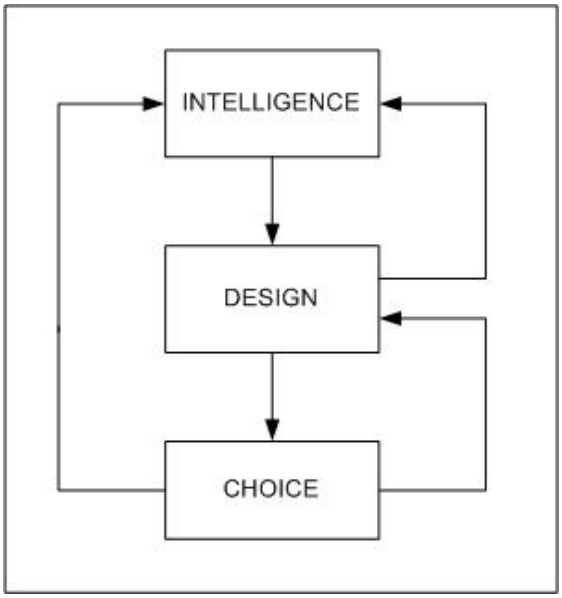


Figure 1.1 Three Phases of Decision-making

Developments in the database technology have improved the understanding of the need for developing Decision Support Systems (DSSs) which encompass seperate analysis-oriented databases to support decision-making processes. This need has emerged from the fact that transactional databases and online transactional processing (OLTP) applications are used to handle daily, process-oriented operations and they are not intented to support decision making tasks. On the other hand, data warehousing and online analytical processing (OLAP), have increasingly become a focus of the database industry as being essential elements of DS (Chaudhuri et al., 1997). Data warehouses have been designed for analysis oriented operations and targetted for decision making. They provide consolidated and historical data with a multi-dimensional perspective, and mainly intended for analysis-oriented DS applications (http://www.dssresources.com/glossary).

According to Bedard, Merret, and Han (2001), a data warehouse is defined as "an enterprise-oriented, integrated, non-volatile and read only collection of data imported from heterogenous sources and stored at several levels of detail". Having all these characteristics, data warehouses are the best source of information to support

decision-making (Bedard et al., 2001). They provide complex analysis and knowledge discovery through storing decision oriented data.

First, OLAP is supported to provide complex analysis of data from the data warehouse. In the data warehouse, data are stored multi dimensionally, in a time-variying structure with different levels of detail in the form of data cubes. OLAP queries are used to extract information at different levels of detail from the data warehouse. Therefore, OLAP analysis provide faster and better decision-making by the help of different levels of granularity.

Second, knowledge discovery in databases (KDD), which is also refered to as data mining (DM) is used to extract hidden information from large databases (Han et al., 2001). Chen et al (1996) described the requirements and challenges of DM: "A knowledge discovery system should be able to perform on different kinds of data using efficient and scalable algorithms; usefulness, certanity and expressiviness of results; expression of various kinds of results; interactive mining knowledge at multiple abstraction levels; mining information from different sources of data; protection of privacy and data security should be accomplished in order to conduct effective DM" (Chen et al., 1996). In this regard, the data warehouse provide effective DM techniques since it has these relevant features that are required for a knowledge discovery system.

Beyond all these advantages of traditional DSSs including data warehousing, OLAP and knowledge discovery tools, these technologies disregard spatial information while supporting decision-making activities. However, DS tasks usually necessitate spatial information which is critical for decision-making purposes.

On the other side of decision making activities regarding spatial data, Geographic Information Systems (GISs) have been used as DSSs in organizations where spatial information is the focal point of analysis. They provide decision makers with useful information by means of spatial data analysis. However, GISs are based on operational spatial databases and which makes them inefficient for DS.

Consequently, both GISs and data warehousing technologies have strentghs and weaknesses which result in a need for combining both technologies within a Spatial Decision Support System (SDSS) to support decision-making operations of various disciplines that use spatial data as an essential part of analysis. In this context, many studies have been done to integrate GISs with data warehousing technologies to get benefit from these technologies. Below, the statement of the problem is described to present a brief overview of the outcomes of these previous studies, the purpose of this thesis is explained, and research settings, objectives are established in order to define where this study fit in the literature.

Data warehousing and OLAP technology has been used to access, visualize and analyze multidimensional, aggregated, and summarized data which are needed for DS purposes. Large part of data contains spatial components. Thus, these spatial components convey valuable information and GISs which provide a wide range of tools to analyze spatial phenomena must be integrated into exploration and analysis phases of a DSS.

In this regard, this study aims to search for answers to the problem how to design a spatially enabled data warehouse architecture in order to support spatio-temporal data exploration and analysis of multidimensional data. The implementation stages of a spatial data warehouse model and the application of spatio-temporal data exploration and analysis techniques are demonstrated in this model to assist spatially related decision-making tasks.

Implementation stages of a spatial data warehouse model are studied through using a case study in weather pattern searching and real weather data. Multidimensional analysis and geo-visualization of spatio-temporal data are performed to extract knowledge about weather data. Furthermore, the proposed model is evaluated and extracted knowledge is used to support hyphothesis generation, forecasting, trend analysis in weather pattern searching.

In order to combine the benefits of OLAP and GISs, some research instutions developed commercial tools and softwares: Kheops Technologies developed JMap Spatial OLAP which enables performing OLAP operations on spatial databases (Kheops Technologies, 2005). Geominer which includes spatial data cube construction module, spatial OLAP module and spatial data mining module, was developed by Geominer Research Group at Simon Fraser University (Han et al., 1997). Spatial OLAP Visualization and Analysis Tool (SOVAT) that combines OLAP and GIS functionalities, was developed by a research group at University of Pittsburgh (Scotch et al., 2007). Geographical On-line Analytical Processing (GOLAP) system which integrates data warehousing and GISs was developed by Gerstner Laboratory for Intelligent Decision Making and Control at Czech Technical University (Kouba et al., 2002). Also, Sentim and Başar Computer Systems developed a DSS for Ministry of Education in Turkey. This environment provides integration of spatial features such as streets, points of interests and geocoding information with central information systems in order to analyze efficiencies of public schools in Turkey
(http://www.basarsoft.com.tr/basar/tr/yenihaber/haber_detay.asp?id=104).

These tools and softwares involve integration modules which help to handle OLAP and spatial components in an integrated environment. In this context, this study differs from these tools and software in that data warehouse and the spatial components are handled separately and integrated using ARC GIS OLAP Add-on. Geographic integration with OLAP is provided by joining spatial components with spatial dimensions in data cubes. Without depending on specific commercial software, the model proposed in this study provides an easy way for geographic display, navigation and spatial analysis of OLAP data.

CHAPTER 2

LITERATURE REVIEW

2.1. Data Needs for Decision Support

Inmon (2002) categorizes data into two kinds – primitive data and derived data. According to his classification, primitive data are used to run day-to-day operations of a company where as derived data have been summarized or calculated to meet the needs of the management of that company. This categorization emphasizes data needs of DS in managing a company. But it is also valid for other areas which have something to do with DS.

There are major differences between primitive data and derived data and they are listed in Table 2.1.

Because of the differences listed in Table 2.1, primitive data and derived data cannot be kept in the same database. Primitive data are stored in transactional databases where as derived data are stored in DS databases which are called data warehouses. As well as the requirement for databases to store primitive data and derived data seperately, the database management systems to maintain and analyze these data have to be designed by seperate approaches. At first, OLTP as a class of systems that facilitate and manage primitive data in transaction-oriented applications is used. From the other point of view, OLAP is used to provide quick answers to analytical queries that are based on derived data on data warehouses.

In the next section, OLTP is compared to OLAP in order to better understand the maintaining and analyzing needs of a data warehouse as a DS database.

Table 2.1 Differences Between Primitive and Derived Data Adapted From Inmon (2002)

	PRIMITIVE DATA/OPERATIONAL DATA	DERIVED DATA/DSS DATA
DATA	can be updated	is not updated
	high availability	relaxed availability
	compatible with the Systems Development Lifecycle (SDLC)	completely different life cycle
DATA ACCESS	accurate, as of the moment of access	represents values over time, snapshots
	requirements for processing understood a priori	requirements for processing not understood a priori
	accessed a unit at a time	accessed a set at a time
DATA STRUCTURE	detailed	summarized, otherwise refined
	no redundancy	redundancy is a fact of life
	static structure; variable contents	flexible structure
FUNCTION	application oriented	subject oriented
	transaction driven	analysis driven
	supports day-to-day operations	supports managerial needs
MANAGEMENT	managed in its entirety	managed by subsets
	high probability of access	low, modest probability of access
	control of update a major concern in terms of ownership	control of update no issue
PERFORMANCE	performance sensitive	performance relaxed
SIZE	small amount of data used in a process	large amount of data used in a process
TASKING	run repetitively	run heuristically
USER	serves the clerical community	serves the managerial community

2.2. Online Transactional Processing versus Online Analytical Processing

OLTP is typically used for performing data entry activities for databases. Decision-making activities require heavy use of aggregations and much more complex than typical OLTP queries (Harinarayan et al., 1996). Especially, in large databases it becomes far inefficient to perform aggregations and complex queries in OLTP systems. In comparison to OLTP, OLAP queries are performed on simplified table structures where detailed data from individual records have been cleaned and abstracted. Furthermore, OLTP systems are process-oriented while OLAP systems are history-oriented with their emphasis on cleaned, abstracted historical data. In Table 2.2, the major differences between OLAP and OLTP are listed and categorized according to the criteria related with database and purpose of these two systems.

Table 2.2 OLAP vs OLTP

Criteria	**OLAP**	**OLTP**
Use	Analysis, Reporting, Modeling, Planning	Transaction processing
Data contents	Historical and aggregated data	Current, up-to-date data to run fundamental operational tasks
Dimensionality	Multi-dimensional, hierarchical	Two dimensional, normalized
Querying	Standardized and simple queries	Often complex queries with aggregations
Database Design	Monitored redundancy (star and snowflake), de-normalized table structure	Minor redundancy, highly normalized table structure
Data Access	Mainly reading	Reading and writing

2.3. Data Warehouse

Decision-makers require fast and derived information from vast amounts of data which cannot be achieved with transactional database systems. Transactional databases have different requirements in terms of database technology. They are used to handle operational data which are usually needed for daily operations. However, decision-makers need to make better and faster decisions by searching for summarized and consolidated data that are not explicitly stored at such databases.

At this point, "data warehouse" concept has been originated in order to respond to the needs for DS. It refers to a typical DS database that is particularly designed for decision-making purposes. In this regard, Kimball (2002), a leading proponent of the dimensional approach to building data warehouses, briefly defined "data warehouse" as a copy of transaction data specifically structured for query and analysis. Inmon (2002), being an early and influential practitioner of data warehousing, characterized a data warehouse as a "subject-oriented, integrated, time varying, non-volatile collection of data in support of management's decisions" (cited in (Chaudhuri et al., 1997), (Elmasri et al., 2004)). Bedard et al (2001) extended this definition by adding new characteristics such as enterprise-oriented, granularity, and read onlyness to the concept of "data warehouse".

Data warehousing has been deployed in many industries and applied to many disciplines to support decision-making tasks. Some examples of these industiries and disciplines are manufacturing, retail, financial, insurance, transportation, telecommunication, utilities, weather pattern searching, healthcare.

2.3.1. Characteristics of Data Warehouse

In the following sections, the characteristics of the data warehouse is given in detail.

Subject-oriented: The data warehouse is subject-oriented since it is oriented to the major subject areas of a decision-making process (Inmon, 2002). It is developed in

numereous organizations in order to meet particular needs. Data in the data warehouse are organized in accordance with the requirements of decision-making in those organizations.

Integrated: Data in the data warehouse are imported from different sources which may contain different semantics, constraints, formats and codings (Bedard et al., 2001). Therefore, this characteristic implies that a data warehouse consists of a homogenous data model which is built by transforming heterogenous data from different sources.

Non-volatile: Operational data are needed to satisfy the short-term needs of organizations. In most cases, they are archived or destroyed after they have been used for certain purposes. Therefore such data stored in operational databases are called volatile meaning that they contain current data only and are replaced by most recent values when they are not needed for organizational purposes. On the other hand, in data warehouses, historical data are stored to enable trends and prediction analysis over time. Therefore, data kept in data warehouses are non-volatile since they are not replaced by new values after they have been out-of-date (Bedard et al., 2001).

Time-variant: The term time-variant expresses the focus of data warehousing on change over time in order to make trend and prediction analysis. In the data warehouse, historical data are stored in different levels of detail to make analysis of time-variant data.

Granularity: Granularity is summarization of data which enables keeping them in different levels of detail in the datawarehouse. When more detail is needed, the level of granularity decreases. On the contrary, the level of granularity increases if the less detail is desired (Inmon, 2002).

Read Only: Bedard et al (2001) state that the data warehouse cannot alter the state of the source databases for technical concerns such as avoiding update loops and

inconsistencies. In this concern, most data warehouses are not allowed to write back into databases from which the data are extracted to build the data warehouse.

2.3.2. Data Warehouse Design

Data warehouse architecture consists of some basic components where each component has different functionalities in building the data warehouse stage. These components are illustrated in Figure 2.2. First of all, data are extracted from multiple data sources into a relational database where data cleaning and transformation steps are handled. After performing the necessary cleaning operations, data are loaded into the warehouse which has the multidimensional structure and refreshed periodically when the new update information become meaningful for the goal of warehousing. The data warehouse may involve several data marts which may serve to different departments of an organization, for which the warehouse is constructed. OLAP servers are used to provide multidimensional views of the warehouse and the data marts. Users access the warehouse via these servers. This access helps end users to explore and analyze the data using different visualization methods and operations such as drill-down, roll-up, slice, etc. These servers provide the front-hand tools: OLAP, data analysis, query and reporting and knowledge discovery tools; to be applied to the warehouse (Chaudhuri et al., 1997).

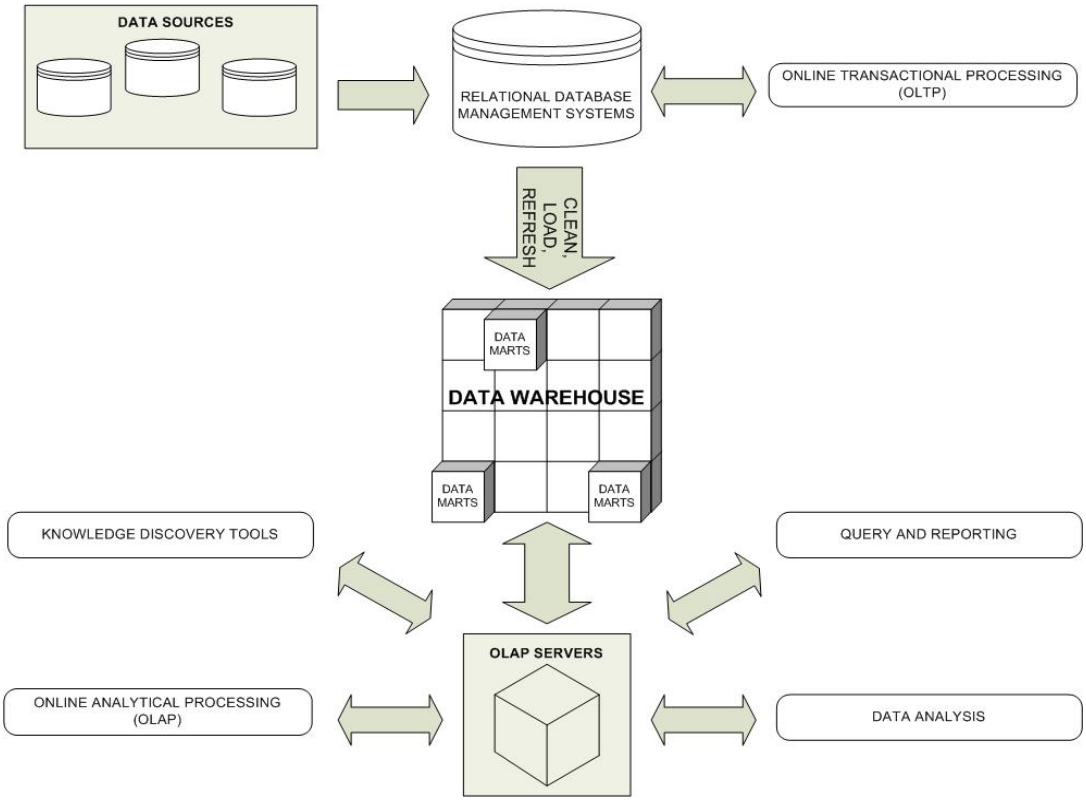


Figure 2.2 Data Warehouse Architecture (Chaudhuri et al., 1997)

Data warehouse is designed to enable the efficient query processing over huge amounts of data. Efficient query processing can be accomplished through using efficient access methods and query processing techniques. In this regard, Chaudhuri et al (1997) pointed out the important issues regarding the design of data warehouse:

- First, since data warehouses use redundant data structures such as indices and materialized views, which indices to build and which views to materialize must be chosen.
- Second, existing indices and materialized views must be effectively used to answer queries.
- Third, complex queries must be optimized.
- Fourth, index scans must be efficiently used to improve the efficiency of scans.
- Fifth, query response times must be reduced by exploiting parallelism.

In addition to the components in the design of warehouse architecture, metadata management is crucial to run and maintain a warehouse and must be involved in the architecture. Metadata consist of data about data in a warehouse. It is used to add context and understanding of data for users of a warehouse. Chaudhuri et al (1997) defined three different kinds of metadata to be managed in a warehouse: Administrative, business and operational.

The first type, administrative metadata involves data required for constructing and using the warehouse. It consists of descriptions of data sources, back-end and front-end tools, definitions of the warehouse schema, derived data, dimension, hierarchies, queries, reports, data mart locations and contents, physical organizations, data cleaning steps and procedures, load and refresh policies, user authorizations and access controls. Business type involves terms and definitions related to the subject of the warehouse, ownership of data and charging policies. The third type, operational metadata includes information about the operation of a warehouse. It consists of lineage reports of migrated and transformed data, currency of data and monitoring information such as usage statistics error reports and audit trails. All these three types of metadata should be managed in order to run and maintain a warehouse.

2.3.3. Building the Data Warehouse

Chaudhuri et al (1997) introduced the steps taken to build the data warehouse as Back End Tools and Utilities. These steps are categorized into three parts: data cleaning, loading and refreshing. In the next three sections, the steps to build the data warehouse are explained in accordance with this categorization.

2.3.3.1. Data Cleaning

In order to create the data warehouse, data must be extracted from operational transaction-oriented databases then entered into the data warehouse. Depending on the subject of the data warehouse, large volumes of data can be extracted from multiple, heterogenous sources. This results in a high probability of errors and

anomalies in the data. In this context, data cleaning must be performed to ensure validity in the data warehouse. Being the very first step of building the data warehouse, data cleaning is an involved and complex process that has been identified as the largest labor-demanding component of data warehouse construction (Elmasri et al., 2004). This step involves a collection of tools that help to detect and correct data anomalies. Chaudri et al (1997) list the main cases where data cleaning is necessary: inconsistent field lengths, inconsistent descriptions, inconsistent value assignments, missing entries and violation of integrity constraints. Inmon (2002) points out the reasons for these inconsistencies and emphasizes the importance of data cleaning and integration by simplifying the relationship between the different sources and integration of a company's data warehouse. The lack of integration between different sources and the reasons for inconsistencies are illustrated by Inmon (2002) in Figure 2.3.

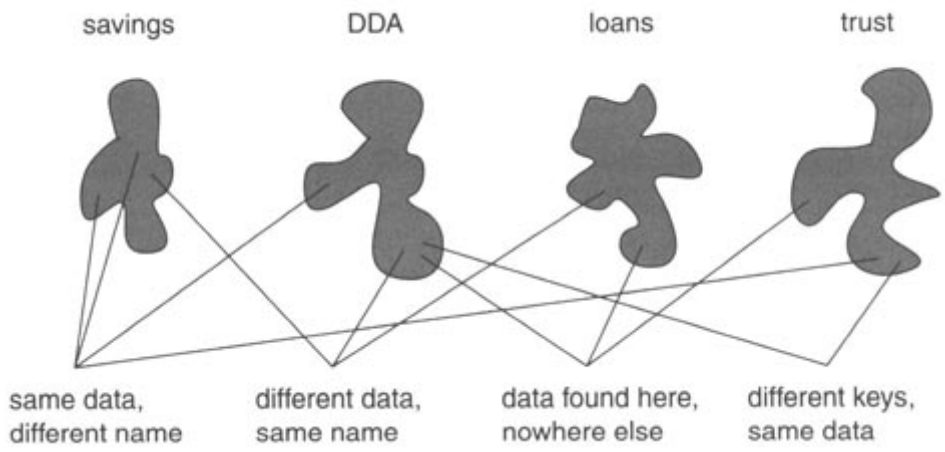


Figure 2.3 Data across the different applications is severely unintegrated (Inmon, 2002)

Chaudri et al (1997) introduce three different classes of data cleaning tools which are data migration, data scrubbing and data auditing tools in order to overcome the inconsistencies of these unintegrated data that Inmon (2002) previously pointed out. In the next sections, these tools are explained in detail.

i) Data Migration Tools

Data migration tools consist of transformation rules in order to form a unified data warehouse. As seen in Figure 2.3, same data might be encoded differently in different databases. When these databases are integrated to form a data warehouse, data should be recoded with proper values. Some situations are illustrated to give an overview of such cases:

For example, constructing a data warehouse to assist customer relationship management, several databases must be integrated. In those databases, address information of customers are stored and these information might be kept with different field names. In one database, it might be stored in a field called “Location” where as in another database it might be stored in a field called “Address”.

Not only the field names but also the values stored in these fields might have different values. For example, a data warehouse can be built in order to assist urban planning. Usually, in such databases the same types of data sources are kept differently. For example, fire escape property of a building might be encoded with different value types in different databases. In one database it might be encoded as a true/false value, in another it might be encoded by a string with an explanation.

Additionally, some fields in different databases might represent different measurement types. For example, building a data warehouse for a parcel based real estate application, several databases from different cities must be integrated in that warehouse. In one database, total area of a parcel might be measured in square meters whereas in some other databases, it might be measured in square kilometers.

Each case given in the examples represent certain inconsistencies while constructing data warehouses. Data migration tools help to overcome these inconsistencies by converting different encodings into consistent ones.

ii) Data Scrubbing Tools

Data scrubbing which is also known as data cleansing is the act of detecting and correcting corrupt or inaccurate records from a database (Han et al., 2001; Kimball et al., 2004). These inaccurate records resides in a data warehouse as a result of different data definitions of similar entities in different databases which are integrated into the warehouse, entry errors, corrupted data in transmission or storage. Although they may include the same techniques or processes, data scrubbing is different from data validation in the context of building the data warehouse. At one side, validation of data is performed at the entry stage of data into the database and inconsistent data are not validated to reside in the database. On the other side, data scrubbing is applied on batches of data in the warehouse.

According to Chaudhuri et al (1997), data scrubbing tools are used to do the scrubbing of data in the context of a domain specific knowledge. It involves validating and correcting values and typological errors against a known list of entities. Parsing and fuzzy matching techniques are usually used to accomplish data cleansing. The act of rejecting any address in a data warehouse if it does not have a valid postcode may be given as an implementation example of these tools (Chaudhuri et al., 1997).

iii) Data Auditing Tools

Data auditing tools help to discover rules and relationships that are irrelevant to DS task, and detect violation of stated rules in warehouses (Chaudhuri et al., 1997). They involve profiling and assessing poor quality data. For example, information regarding a car dealer, which has never received any complaints, presents a suspicious case in the warehouse functioning for a car dealer organization. Data auditing tools, being variants of DM tools, can be used to discover such cases in a warehouse (Chaudhuri et al., 1997).

2.3.3.2. Loading

Loading is the second main stage of building the data warehouse. After data have been cleaned through data migration, data scrubbing, and data auditing tools, they have to be loaded into the warehouse. However, before the loading stage, data have to be preprocessed for checking integrity constraints, sorting, summarization, aggregation, and other computation to build the derived tables stored in the warehouse; building indices and other access paths; and partioning to multiple target storage areas (Chaudhuri et al., 1997).

Chaudhuri et al (1997) emphasize the need for a system, which is called batch load utilities, to control the load activity in the warehouse. Batch load utility helps the system administrator of the warehouse monitor status, cancel, suspend and resume a load and restart after a failure with no loss of data integrity (Chaudhuri et al., 1997).

2.3.3.3. Refreshing

Data warehouses are snapshots of transactional databases in a time interval. Due to the frequency of operational activities, data in a transactional database are updated continuously. Therefore, these propagating changes have to be updated into the warehouse. In addition to that, the derived data in the warehouse have to be updated correspondingly. In this regard, refreshing stage refers to updating both the propagated and derived data into the warehouse.

Chaudhuri et al (1997) introduce two sets of issues to consider refreshing propagating changes into the warehouse: when to refresh and how to refresh the warehouse. These issues for refreshing depend on two main factors:

i) First, refereshing depend on user needs. For example, in a weather data warehouse, which is aimed to support weather pattern searching by analyzing historical data, there is no point in refreshing the warehouse daily. Because, such analysis require aggregated data over weeks, months and even years to search for

general patterns of weather data. However, daily information refreshed into the warehouse cannot have such an effect that would result in a change in weather patterns. Hence, it is more meaningful to update the weather warehouse at least weekly, where updated data may imply more valuable information. On the other hand, an example of a warehouse designed for business process management, have different requirements in terms of user needs. In business process management, process optimization is the basic need of decision makers. Therefore, the users of the warehouse have to realize the changing needs of clients in order to optimize the processes to save costs. Such a requirement necessitate monitoring the current situation as frequent as possible. Thus, the warehouse need to be refreshed daily, even hourly in such a case.

ii) Second, refreshing depend on the characteristics of the source and the capabilities of the database servers. At this point, full refreshing and incremental refreshing are the two choices for refreshing strategy. Full refreshing refers to extracting the entire data source and building the warehouse from scratch whenever an update is scheduled. In other words, it means replacing the warehouse with a new one and consequently this takes much more time than incremental refreshing. On the other side, incremental refresh refers to loading only the updates to the warehouse when the sources change. Replication servers are used to accomplish incremental refresh in the warehouse and data shipping and transaction shipping are used as replication techniques in these servers. In data shipping, row triggers are used to update tables in the warehouse. Whenever the source tables change the warehouse are updated by a refresh schedule. In transaction shipping, transaction log is used to refresh the warehouse. Transaction shipping is advantageous since it does not require triggers which can increase the workload on operational databases. On the other hand, it cannot be used by some DBMSs, since there are no standard APIs for accessing the transaction log (Chaudhuri et al., 1997).

Additionally, the derived data in the warehouse which consist of summary tables, single table indices, join indices have to be updated correspondingly. Similar to propagated data, the timing and scope to replace or append updated derived data are

strategic design choices dependent on the time available and the requirements of warehouse users.

2.3.4. Multidimensional Data Model Approach

Multidimensional approach used by data warehouses, OLAP, and DM in data cubes, provides efficient analysis and exploration in DS activities. Efficient analysis and exploration requires aggregated data accross many dimensions. In multidimensional approach, this is achieved by constructing a multidimensional data model, which is highly denormalized and aggregated along conceptual hierarchies called dimensions (Marchand et al., 2004).

In transactional databases, updating the database is the primary concern in designing the database model. Normalization techniques are used to prevent functional dependencies in the data so that the database can be updated easily and efficiently. In a normalized database model, querying usually require many joins of table to combine information and these joins result in increasing the query running time which is not desired in data warehouses. On the contrary, in data warehouses, querying the database is the primary concern in designing the multidimensional data model. Thus, denormalization is typically used to reduce the number of joins in the database. Consequently, the query running time is decreased significantly, and better performance can be achieved (Microsoft SQL Server 2005, Books Online, 2007).

Mutidimensional data model in a data warehouse is based on facts, dimensions and measures. A measure represents the property of a fact within a dimension of interest. For example, in a data warehouse for analyzing sales data, an example of a measure is total cost, which represents the fact of purchase within the dimensions of interest such as region, date, product.

Bedard et al (2001) explain the relationship between statistics and multidimensional approach used in data warehouses. According to this explanation, the term “measure” used in a multidimensional data model refers to a dependent variable and

"dimension" refers to an independent variable used in statistics. Therefore, analyzing a multidimensional data in a data warehouse is similar to the act of fixing the independent variables first and then finding what the dependent variable is in statistics. The major reason why multidimensional approach is intuitive is because it provides easy ways of analyzing complex data to decision-makers who are not experts in statistics (Thomsen, 1997).

Multidimensional data model can be applied to the data warehouse using three data structures: the Star Schema, the Snowflake Schema, and the Fact Constellation (Stefanovic et al., 2000; Bedard et al., 2001; Chaudhuri et al., 1997; Han et al., 2001).

The Star Schema: A star schema consists of one central fact table and several dimension tables where each of them is based on a single dimension table and directly linked to the fact table by a primary key – foreign key relationship (Microsoft SQL Server 2005, Books Online, 2007).

Figure 2.4 illustrates a subsection of a data warehouse for sales purposes. The fact table in the center is related to six dimension tables which are Order, Customer, Salesperson, Product, Date and City. Underlined columns representing the foreign keys in the dimension tables are linked to the primary key columns in the fact table and this is illustrated by arrows in the figure.

The Snowflake Schema: The snowflake schema consists of one fact table and several dimension tables. It differs from the star schema in that the dimension tables in the snowflake schema are normalized. As shown in Figure 2.5, the dimension tables: Product is normalized and decomposed into the tables Product and Category; Date is normalized and decomposed into the tables Date, Month and Year; City is normalized and decomposed into the tables City and State.

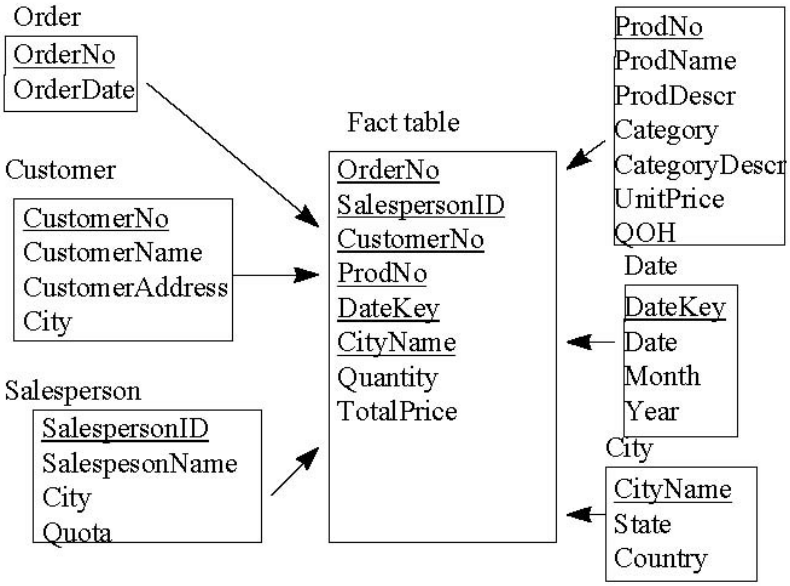


Figure 2.4 A Star Schema(Chaudhuri et al., 1997)

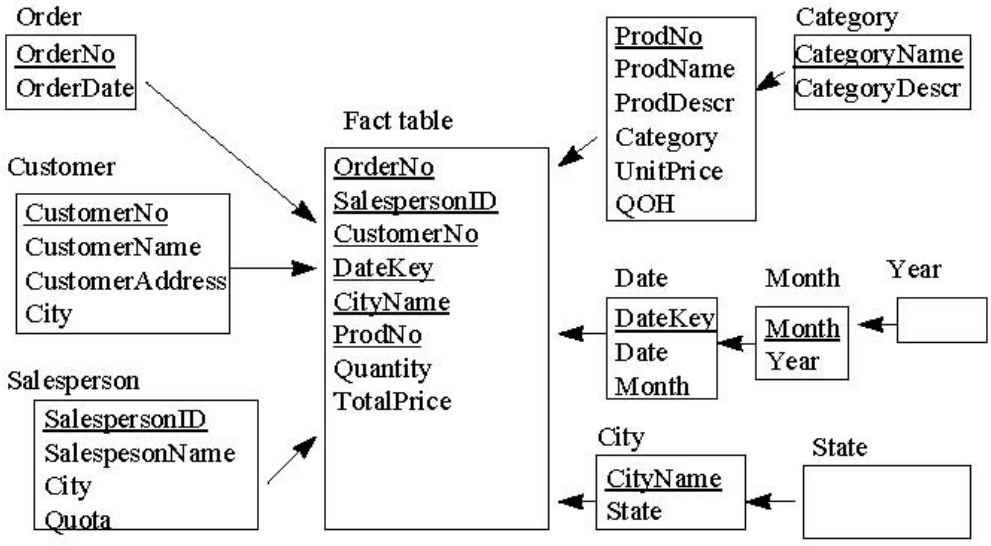


Figure 2.5 A Snowflake Schema (Chaudhuri et al., 1997)

Although the snowflake schema is advantegous in maintaining dimension tables, the star schemas are more efficient for query running time and more appropriate for browsing the dimensions as compared to snowflake schemas since star schemas provide denormalized structure of dimension tables (Chaudhuri et al., 1997).

The Fact Constellation: The fact constellation consists of multiple fact tables that share dimension tables (Chaudhuri et al., 1997). An example of a fact constellation table is given in Figure 2.6. In the figure, there are two fact tables: Sales and Purchases. These two tables share the same dimension tables: Product, Date and City.

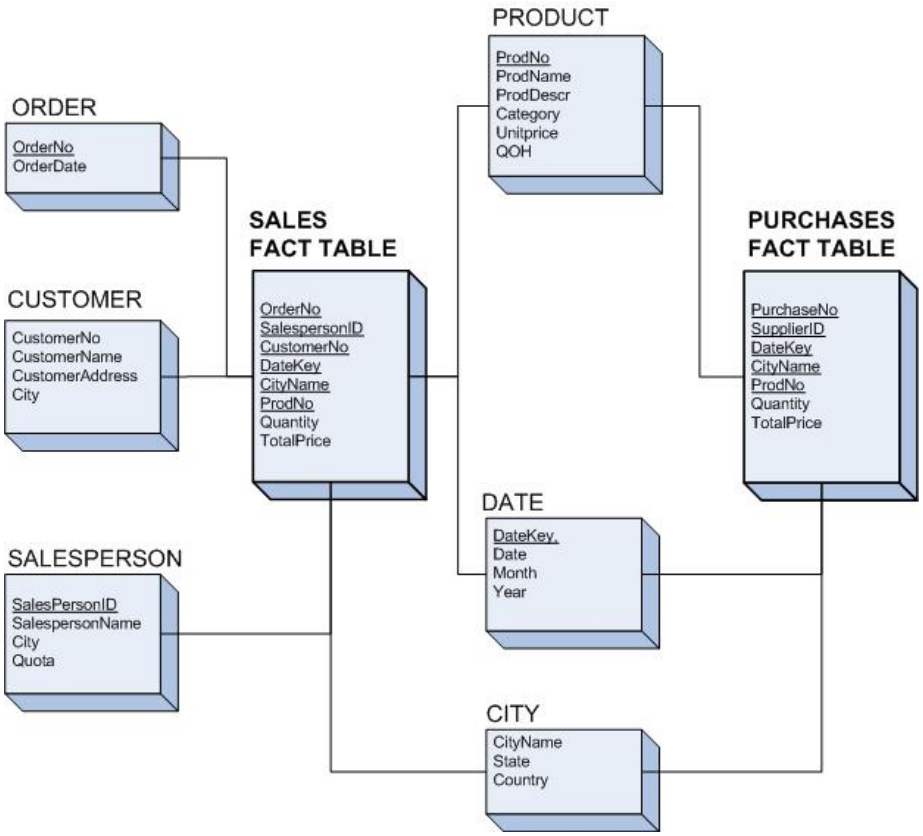


Figure 2.6 A Fact Constellation Schema

If the need for selecting many variants for different levels of aggregation is considered, the fact constellation schema becomes complicated. Moreover, dimension tables become larger.

Data cubes are the data structures which provide multidimensional approach to be implemented into a data warehouse. A data cube consists of a set of measures aggregated according to a set of dimensions. Chaudhuri et al (1997) illustrated multidimensional data model in Figure 2.7 in the form of a data cube in a data warehouse which is constructed to analyze and explore the sales of a company. In this example, sale amount, number of sales, budget, revenue, inventory can be considered as the object of analysis and used as numeric measures. These numeric measures depend on a set of dimensions which are product, city and date.

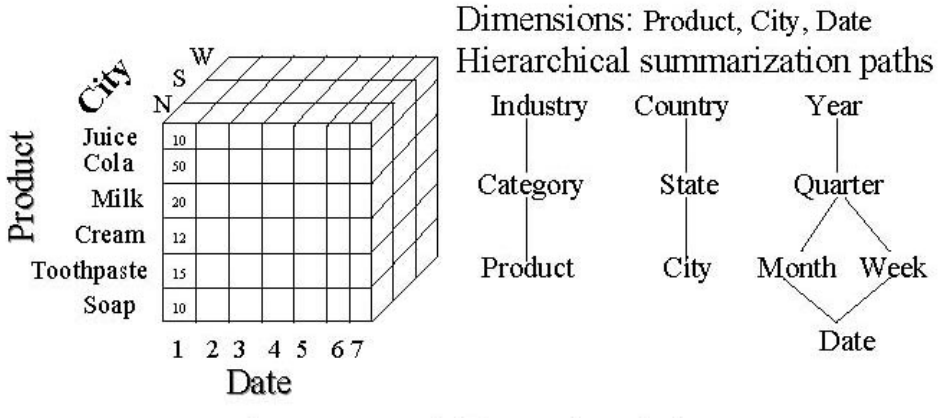


Figure 2.7 Multi Dimensional Data (Chaudhuri et al., 1997)

Hierarchical summarization paths in Figure 2.7 show the different levels of granularity in the cube. Members of dimensions are structured in a hierarchical manner. In the product dimension, products, juice, cola, and milk may fall into the category beverages which belong to the food and bevarage industry. In the time dimension, year is subdivided into quarters, quarters subdivided into months and

weeks, months and weeks subdivided into days. In the city dimension, country is subdivided into states and states subdivided into cities.

i) Implementation Alternatives

Heavy use of aggregations and complexity of queries result in taking longer times to evaluate queries in a data warehouse. In order to achieve better response times, frequently-asked queries have to be precomputed. The action of precomputing is also called as materialization of queries. Harinarayan et al (1996) proposed three implementation alternatives to decide which sets of queries to materialize.

The first alternative is physically materializing the whole data cube which means precomputing and storing every cell in a data cube. This alternative provides better response times in terms of querying. On the contrary, it results in consuming larger space in data cubes and indirectly increases the time for creating data cubes. Moreover, it affects indexing which brings more cost to the alternative.

The second alternative is materializing nothing. This alternative proposes computing every cell on request. While it does not need extra space to store precomputed cells, on the other hand it does not provide quick query reponses.

The third alternative is materializing only part of the data cube. In this approach, queries are evaluated in order to pick the right cells to materialize. Therefore, for large data cubes in which constraints such as space and time for creating cubes are important concerns, only a part of a data cube is materialized and consequently, this provides handling large data cubes efficiently and scalably. Materializing the sets of cells in a data cube is equivalent to materializing corresponding SQL queries to materialize (Harinarayan et al., 1996).

Materialized data can be stored in fact tables or in additional summary tables where sets of cells of the data cube are assigned to different summary tables. In the next section, methods for storing materialized data is given in detail.

ii) Methods for Storing Materialized Data

Data cube structure necessitates possible aggregations of measures on all possible combinations of dimension members to be materialized. Therefore, it helps to increase query performance in comparison to the conventional transaction oriented structures.

Materializing data which corresponds aggregating the fact table on one or more dimensions is essential for meeting query performance requirements. Therefore, as well as fact and dimension tables, summary tables that contain pre-aggregated data should be stored in data warehouses. Chaudhuri et al (1997) proposed two different ways to store pre-aggregated summary data in a data structure.

In the first method, summary table can be represented as a seperate fact table. To demonstrate this, a summary table can be constructed on total sales basis by year and by product from the data cube given in Figure 2.7. Table 2.3 illustrates a separate fact table which shares the dimension "Product" and a seperate consolidated dimension for time which consists of only years.

Table 2.3 Summary Data in a Separate Fact Table

Date	Product	Number of Sales	Sales Amount
..	..	..	..
2006	Beverages	150456	21256400
2006	Food	210500	20243653
..	..	..	..
2007	Beverages	120310	22745000
2007	Food	198760	21198030
..	..	..	..
2008	Beverages	138000	21567045
2008	Food	25480	30175466

In the second method, summary tables can be encoded in the same fact table. Table 2.4 is given below to represent keeping summary data in the fact table which is derived from the data cube example given in Figure 2.7.

Table 2.4 Summary Data in the Same Fact Table

Date	Product	City	Number of Sales	Unit Price	Sales Amount
30/07/2006	Milk	Ankara	150	2	300
31/12/2006	Juice	İstanbul	115	3	345
..	..	..	..	..	..
2006	**Beverages**	**NULL**	**150456**	**NULL**	**21256400**
30/08/2007	Cola	Eskişehir	236	3	708
25/11/2007	Cola	Ankara	184	4	736
16/1/2007	Milk	İzmir	101	4	404
..	..	..	..	..	..
2007	**Beverages**	**NULL**	**120310**	**NULL**	**22745000**
30/05/2008	Juice	Antalya	67	5	335
14/06/2008	Milk	İzmir	35	6	210
..	..	..	..	..	..
2008	**Beverages**	**NULL**	**138000**	**NULL**	**21567045**

In the table, summarized data and general data are stored in the same fact table which results in having null values at some fields. The rows starting with 2006, 2007, and 2008 give the summarized data according to the certain aggregation levels of the dimensions: Date, City and Product. In this example, Date dimension is aggregated into its upper most level of aggregation which is “Year”; Product dimension is aggregated into its middle level of aggregation which is “Category”; City dimension is not aggregated and takes the value “NULL” to include all the items. Gray et al (1996) proposed using the dummy value “ALL” instead of NULL in order to fill in these super-aggregation items.

The first method by keeping a seperate summary fact table is better in maintaining the multidimensional structure. However, the number of tables and joins cause higher response times. On the other hand, keeping fewer tables, the second method is more

efficient and reduce response times. However, such a structure results in operational errors since the level field has high risks of misinterpretation. Therefore, both methods should be evaluated in accordance with the summary requirements of the structure in different cases.

In addition to choosing a method for storing summary data, the system for storing materialized cubes must also be chosen. Relational systems and Multidimensional Database Systems (MDDBS) are the alternatives to store these materialized cubes (Harinarayan et al., 1996). MDDBS are more advantages in terms of performance metrics but they require much larger spaces as compared to relational structures because usually all possible combinations of dimensions and measures are calculated in these systems.

iii) Methods for Summarizing Data

Two possible system alternatives, relational and multidimensional systems, have different methods for summarizing data. In relational structures, data are summarized using SQL aggregate functions and operators. On the other hand, in a sample multidimensional stucture this is achieved through using an interface and a specific query language, Multidimensional Expressions (MDX) in OLAP databases.

First of all, in a relational multidimensional system, data in a data cube structure are typically aggregated across many dimensions as in Table 2.4. In a relational structure, zero or one dimensional aggregates are provided by the SQL aggregate functions and the GROUP BY operator (Gray et al., 1996). In Table 2.4, the measures in the fields: Number of Sales, Unit Price and Sales Amount are computed using the SQL Standard, GROUP BY operator and its functions SUM( ) and COUNT( ). Gray et al (1996) pointed out the problems with GROUP BY operator, and introduced CUBE and ROLL-UP operators to overcome these problems. The CUBE operator corresponds to computing group-bys on all possible combinations of a list of attributes. It is the N dimensional generalization of simple aggregate functions (Gray et al., 1996). Therefore, it is equivalent to the union of a number of standard group-

by operations (Agarwal et al., 1996). In Table 2.4, aggregated rows which start with 2006, 2007, 2008 can be acquired using the cube operator as follows:

```
SELECT    Product, Date, Sum (Number_of_sales),
          Sum (Unit_price), Sum (Sales_amount)
FROM      Sales
CUBE-BY   Product, Date
```

Figure 2.8 SQL Sample Cube By Query

In order to obtain the same result by using group-bys in this example, $2^2 = 4$ group-bys are needed: Product-Date, Product, Date, and all, where all refers to the empty group-by. The CUBE operator can only be applied to distributive functions such as max, min, count and sum; and algebraic functions such as average (Gray et al., 1996).

In the second alternative, just as the SQL commands to obtain aggregated rows in a relational system, MDX are used to query a multidimensional cube in a multidimensional system. MDX provides syntaxes for querying and manipulating the multidimensional databases (MDDBs). MDX expressions and queries enable formatting query results, returning data to clients from servers, perform cube design tasks, perform administrative tasks including dimension and cell security. In this study, only the querying and cube design functionalities of MDX are studied (Microsoft SQL Server 2005, Books Online, 2007). In order to illustrate the querying function of an MDX query, an example scenario would be to analyze "Sales_amount" and "Unit_price" measures by the "month" level of "Date" dimension and by the product categories level of the "Product" dimension in a single analysis. The sample MDX query to perform this scenario is given in Figure 2.9. This query necessitates two-dimensional view of the query result since there are two dimensions to be analyzed in the scenario. Therefore, in the first part of the MDX

query in Figure 2.9, four categories of the dimension of Product is selected and arranged to be displayed “ON ROWS”. In the second part of the query, “Month” level of the dimension Date is selected and the months in a year are involved in the statement and arranged to be displayed “ON COLUMNS”. In the third part of the query, in the 20th line of Figure 2.9, measures Sales_amount and Unit_price are multiplied with the selected dimensions Product and Date. This means that measures will be calculated on the combinations of these two dimensions.

This scenario shows that it is possible to analyze measures not only by a single categorical dimension but by elements of multiple dimensions such as Product and Date in the example query given in Figure 2.9. In Table 2.5, a part of the query result is given to illustrate the two dimensional analysis of the two measurements Sales_amount and Unit_cost.

Table 2.5 Sample MDX Query Result for Sales Analysis

	January		February	
Category	Sales Amount	Total Unit_price	Sales Amount	Total Unit_price
Beverages	78491.238	35136.9294	83854.0568	38394.8251
Food	5602193.746	4636857.523	8741352.105	7677423.651
Clothing	105420.4722	83479.7606	127973.3448	100541.4354
Components	282957.0739	257151.0458	535609.6779	484204.6155
Grand Total	6069062.531	5012625.259	9488789.184	8300564.527

```
1    SELECT
2    {{([Product].[Category].&[4])},
3    {([Product].[Category].&[1])},
4    {([Product].[Category].&[3])},
5    {([Product].[Category].&[2])}}
6    ON ROWS,
7    {
8    {{([Date].[Month of Year].&[1])},
9    {([Date].[Month of Year].&[2])},
10   {([Date].[Month of Year].&[3])},
11   {([Date].[Month of Year].&[4])},
12   {([Date].[Month of Year].&[5])},
13   {([Date].[Month of Year].&[6])},
14   {([Date].[Month of Year].&[7])},
15   {([Date].[Month of Year].&[8])},
16   {([Date].[Month of Year].&[9])},
17   {([Date].[Month of Year].&[10])},
18   {([Date].[Month of Year].&[11])},
19   {([Date].[Month of Year].&[12])}
20   } * {([Measures].[ Sales_amount]),([Measures].[Unit_price])}}
21   ON COLUMNS

22   FROM [Sales Cube]
```

Figure 2.9 Sample MDX Query for Sales Analysis

2.3.4.1. Online Analytical Processing

The goal of OLAP is to support efficient querying for specific purposes which are important for decision-making activities. Architecture of OLAP is based on three parts: the multidimensional cube structure, OLAP server which manages the multidimensional data and performs calculations and OLAP client which access data through OLAP server and perform OLAP operations. The multidimensional data cube structure provides enhanced spreadsheet functionality, efficient query processing, structured queries, ad-hoc queries which refer to queries for specific purposes, knowledge discovery and materialized views (Elmasri et al., 2004).

OLAP server alternatives are based on physical storage options which affect the performance, storage requirements, and storage locations of partitions and their parent measure groups and cubes (Microsoft SQL Server 2005, Books Online, 2007). There are three alternatives: Multidimensional OLAP (MOLAP), Relational OLAP (ROLAP), Hybrid OLAP (HOLAP).

In a MOLAP server, data are stored in a multidimensional, highly optimized special structures to maximize query performance in OLAP operations. Such servers support the multidimensional view of data through a multidimensional storage engine. MOLAP servers provide excellent indexing properties, but storage utilization is poor when the data set is sparse (Chaudhuri et al., 1997).

ROLAP servers are called intermediate servers that is constructed between a relational back end server and client front end tools (Chaudhuri et al., 1997). In such servers, data are stored in relational warehouses. In order to efficiently support multidimensional OLAP queries, they extend relational servers with specialized middleware. ROLAP servers provide more scalable systems. However, due to the lack of sequential processing and column aggregation, these servers may result in performance bottlenecks for OLAP servers (Chaudhuri et al., 1997).

The third alternative server structure is Hybrid OLAP (HOLAP). These servers are a combination of MOLAP and ROLAP servers. These servers enable storing part of the data in MOLAP and some other part of the data in ROLAP servers. Designing the cube structure and partitioning the data may vary depending on the cases. For example, aggregations may be stored to achieve fast query performances and detailed data can be stored in ROLAP servers to optimize time of cube processing.

As the last component of an OLAP architecture, OLAP clients access data through OLAP servers explained above, and perform operations to access and query multidimensional data. OLAP operations provide some functionalities in terms of accesing and exploring data. These functionalities are given in Table 2.6.

Table 2.6 OLAP Operations

OLAP OPERATIONS	**FUNCTIONALITY**
Roll-up	Summarizing (Aggregating) data with increasing generalization
Drill-down	Increasing levels of detail - opposite of Roll-up
Pivot	Rotating the dimensions to perform cross-tabulation
Slice and dice	Performing projection operations on the dimensions to extract sub-cubes
Sorting	Sorting data
Filtering	Filtering data by value or range
Derived attributes	Computing attributes by operations on stored and derived values

Apart from the server architectures and operation , Harinarayan et al (1996) introduced the two basic implementation approaches that facilitate OLAP. The first

approach is to use a relational database system and query the raw data directly from this system. This approach provides handling very large data warehouses and much better scailability in comparison to MDDB systems. However, in this approach, performance is much lower. To overcome this performance disadvantage, Harinarayan et al (1996) proposed materializing parts of the data cube into summary tables by using a simple greedy algorithm. The second approach is based on a MOLAP architecture, special MDDBSs and APIs for OLAP. In this approach, multidimensional data structures are used to handle data instead of SQL and the relational database systems. In this study, this approach is applied by using an API for creating MDX queries instead of using SQL syntax. In this approach, although efficient query responses are obtained, this approach results in storing very large amounts of data since all the cells of the data cube present in the raw data are materialized.

CHAPTER 3

METHODOLOGY

Many disciplines involve decision-making phases which have some spatial context and use locational data as an essential part of analysis for decision-making. Various implementations of GISs have been used to analyze spatially related information. However, traditional spatial databases do not meet the requirements of decision makers because of four main reasons.

First, GISs analysis on traditional spatial databases do not provide quick responses to decision making practises which involve complex analysis of spatial and non-spatial data. Bedard et al (2001) state that today's GIS packages have been designed and used mostly for transaction processing and minimal analysis. Complex analysis can be accomplished by combining summarized, consolidated and time varying data which is not done with traditional GISs vendors. Therefore, similar to non-spatial transactional databases and OLTP, GISs are not efficient to support decision-making applications.

Second, because of its complex user interfaces, GISs make users involve in analysis process. However, decision-making process requires focusing on the results of the analyses. In other words, "what to obtain" is the focal point of decision-making rather than "how to obtain" (Kheops Technologies, 2005).

Third, in transactional spatial databases, complex spatial indexing structures, spatial reasoning, geometric computation and spatial knowledge representation techniques lead to difficulties to extract information from such databases. Consequently, GISs vendors that use traditional spatial databases do not provide efficient knowledge discovery tools.

Fourth, high levels of user-friendliness and interactive navigation between different levels of detail are required to support decision making process (Kheops Technologies, 2005). This requirement points out the need for a user profile of DSSs in which users are ordinary users rather than experts.

Following the general trends of mainstream information technologies, data warehousing and GISs have been integrated to help overcome the limitations of GISs to better fulfill the needs of spatial decision support. Therefore, GISs with their new spatio-temporal, multidimensional notion, are becoming a corner stone for decision-makers to analyze various types of data in a spatial context (Zhang et al., 2001). In this context, spatial data warehouses have been implemented in order to support decision making processes in analysis oriented operations. Using the penetration of data warehouses into management and exploiation of spatial databases has become a major trend in GIS market (Bedard et al., 2001). Therefore, studies have been focused on understanding the complexities of spatial implications in rapidly expanding business world of using spatially related information for business decision making.

3.1. Spatial Data Warehouse

Stefanovic et al (2000) define the term "spatial data warehouse" by adding the spatial context into in the definition of a data warehouse as follows:

> "A spatial data warehouse is a subject oriented, integrated, time-variant, and non-volatile collection of both spatial and non-spatial data in support of management's decision-making process."

Spatial data warehouses enable storing large amounts of spatial data, aggregating data in different dimensions at different granularity, and performing spatial knowledge discovery.

3.1.1. Challenging Issues in Implementing Spatial Data Warehouses

According to Stefanovic et al (2000), there are two challenging issues in implementing spatial data warehouses: integration issues and providing fast and flexible analysis issues.

The very first step of building the warehouse begins with integrating different data from different systems and sources. The integration issue in a spatial data warehouse is much more complex in comparison to a non-spatial data warehouse. Because, spatial data are usually stored in various sources. Additionally, data formats differ in terms of the structure they are kept and vendors they are processed. Therefore, integrating these data from different sources and formats require complex transformations, cleaning, loading and refreshing procedures in building the spatial data warehouse.

Secondly, different from a non-spatial data warehouse, a spatial data warehouse necessitates realization of fast and flexible OLAP in the spatial context. In other words, collective, aggregated or general properties of spatial objects at different dimensions and at different levels of abstraction should be performed with fast and flexible OLAP methods. In this context, spatial indexing and accessing methods alone cannot provide efficient support for OLAP of spatial data. On the other hand, models and techniques used in data warehousing are not adequate for handling spatial data. To overcome these difficulties and to support online analysis of large volumes of spatial data, Stefanovic et al (2000) proposed the construction of spatial multidimensional data warehouse model and introduced efficient implementation techniques used in spatial data warehousing.

3.1.2. Spatial Data Cubes

Spatial data cubes are the data structures which enable storing both spatial and non-spatial data in spatial data warehouses. Spatial data cubes differ from data cubes in that they provide spatial context to be used in the multidimensional databases.

Stefanovic (1997) and Bedard et al (2001) introduced the concepts of spatial dimension and spatial measure to enrich multidimensional approach. The main purpose of this integration is adding a spatial persepective to the multidimensional approach in order to improve analysis, design and usage of spatial data cubes. Star and snowflake schemas, which form the basis for data cube structures in non-spatial data warehouses, can also be used to organize spatial data warehouse structure and facilitate OLAP operations (Stefanovic et al., 2000; Bedard et al., 2001; Han et al., 2001). The star schema of a spatial data warehouse which is constructed for sales analysis is given in Figure 3.10 below. In this figure, Order, SalesPerson, Customer, Date, City, Product are the dimension tables and they are connected to the Sales Fact Table through primary key–foreign key relationships. For dimensions, hierarchies are given in Table 3.7. In Table 3.7, Date, City and Product dimensions have multiple levels of granularity.

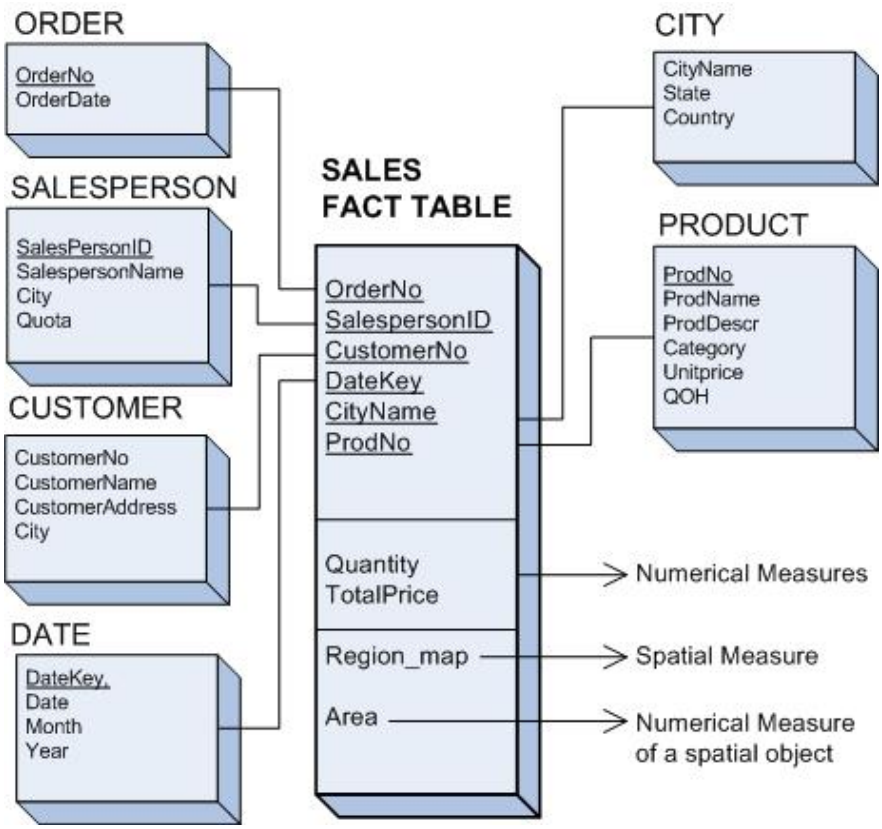


Figure 3.10 A Star Schema Example for a Spatial Data Warehouse

In order to handle uncertain or imprecise data, fuzzy set theory has been integrated into data cubes. In this context, Işık (2005) has done some studies to harmonize spatial data cubes and fuzzy data cubes to enable better analysis, knowledge discovery techniques and understanding of spatial data by using fuzzy set theory. In Işık's study, fuzzy spatial data cube construction and its use in fuzzy association rule mining are introduced. Çalargün (2008) extended Işık's studies by performing fuzzy association rule mining on spatio-temporal data using data cubes and Apriori algorithm techniques.

Table 3.7 Hierarchy Table for the Star Schema

	DIMENSIONS					
	ORDER	**SALESPERSON**	**CUSTOMER**	**DATE**	**CITY**	**PRODUCT**
GENERALIZATION →	ANY	ANY	ANY	ANY	ANY	ANY
	-	-	-	YEAR	COUNTRY	CATEGORY
	-	-	-	MONTH	STATE	PRODUCT DESCRIPTION
	-	-	-	DATE	CITYNAME	-

3.1.2.1.Spatial Dimensions

Stefanovic (1997) first introduced the concept of spatial dimension in order to define dimensions which hold geometric spatial reference in multidimensional data model. Therefore, multidimensional data model is extended by applying spatial dimensions that are based on mesurement scales such as nominal, ordinal, internal and ratio.

Additionally, *measure-folded* dimensions, which are computed measures used as dimensions, can also be included in multidimensional data model. For example,

monthly average temperature in a region is actually a measure but can be used as a dimension in a weather data warehouse (Stefanovic et al., 2000).

Bedard et al (2001) categorize three types of spatial dimensions used in a spatial data warehouse: Non-geometric spatial dimension, geometric to non-geometric spatial dimension, full geometric spatial dimension. In the next three sections, these dimensions are explained in detail.

i) Non-geometric Spatial Dimension

This dimension consists of non-geometric data. They are used to represent spatial dimension in nominal measures such as place names and not represented cartographically. In Figure 3.10, in City dimension, CityNames, and their generalizations such as States, Countries etc. are used to locate a pheonmenon in space. Although cartographic representations are not included in this type of spatial dimension, spatial cognition can be achieved through these nominal references. In other words, even the dimension of City is not connected to sets of polygons to enable geometric representations, it still represents spatial information. This type of spatial dimension is used by OLAP tools (Miller et al., 2003).

ii) Geometric to Non-geometric Spatial Dimension

This type of dimension consists of a root level geometric data with a cartographic representation in the dimensional hierarchy. The root level geometric data become non-geometric when generalized at a certain level of aggregation. In the example given in Figure 3.10 and Table 3.7, for the hierarchies City and State, maps with polygons can be used to represent geometric dimensions where as geometric representations may not be used for the upper level of aggregation which is Country.

This approach demonstrates a transformation of a geometric dimension into a non-geometric one at upper levels of hierarchy and benefits in terms of the simplification of the measurement scales from quantitative to qualitative (Marchand et al., 2004).

iii) Full Geometric Spatial Dimension

In a full geometric dimension, all data in different levels of hierarchy have to be geometric. Polygons defining equitemporate regions and other levels of these data in the hierarchy such regions covering 0-5 degree, 5-10 degree areas also represent geometric information in the dimension (Stefanovic et al., 2000).

3.1.2.2. Measures within a Spatial Data Cube

There are two types of measures within a spatial data cube. The first one is numerical measure which contains only numerical data. This type of measure is the same as a numerical measure in a non-spatial data cube. In Figure 3.10, Quantity, TotalPrice and Area represent numerical measures. Since it represents area property of its corresponding spatial object in the hierarchy, area is a numerical measure based that does not typically reside in a non-spatial data warehouse.

The second type is spatial measure which Stefanovic (1997) introduced it in order to define the measures of multidimensional databases which include spatial representations. In his definition, a "spatial measure" contains a collection of pointers to spatial objects. In Figure 3.10, Region_map represents a spatial measure that contains the collection of spatial pointers for regions in the hierarchy and functions as a connector between the cartographic features and dimensions in the warehouse.

3.1.3. Spatio-temporal Exploration and Analysis on Spatial Data Cubes

The term "Spatio-temporal" contains both spatial and temporal aspects of data. Spatio-temporal data have a multidimensional and multi-scalar nature. As compared to seperate analysis of spatial and temporal data, it necessitates more complex queries. According to Yuan et al (2002), spatio-temporal queries are used to extract information about attributes, in terms of space, duration, and changes. Therefore,

spatio-temporal exploration and analysis are processes of spatio-temporal knowledge discovery, which relate to four main aspects: what, where, when, how.

Exploration and analysis are two complementary processes of spatio temporal knowledge discovery. Spatio-temporal exploration refers to identifying hypotheses that are of interest to the user. Hypotheses comprise potential patterns, associations and unusual occurences which help to detect the future patterns. On the other hand, spatio-temporal analysis determine the validity of the hypotheses developed in the exploration process, and propose new hypotheses in some cases (Marchand et al., 2004).

Before performing spatio-temporal exploration and analysis (STEA), approaches used to establish identity of change have to be understood well. In this regard, Marchand et al (2004) categorize these approaches into two categories: inductive and deductive methods, in order to guide the identification of change in spatio-temporal data.

Inductive methods are used when the focus is on the evolution of large sets of entities. Observation and statistical analysis such as point pattern analysis, spatial autocorrelation, centrographic analysis of geographical entities are the examples of these methods.

On the contrary, deductive methods are used when the focus is on the evolution of small sets of entities. Mathematical and qualitative representations of geographical changes and priori knowledge of possible spatio-temporal relationships are used to construct deductive methods.

Rivest et al (2001) stated that usage of the multidimensional approach is independent of deductive and inductive approaches since spatio-temporal exploration and analysis (STEA) helps users to browse multidimensional spatial data cubes by using graphic representations built according to custom or predefined semiology rules. Mutidimensional structure reflects users' cognitive model of the data. Furthermore,

OLAP supports the iterative nature of the analysis process since it helps users to explore and navigate accross the different dimensions at different levels of detail (Rivest et al., 2001).

CHAPTER 4

APPLICATION

Having the strengths of GISs and data warehousing, spatial data warehouses can be constructed as SDSSs to assist spatial data related decision making processes. As a result, spatial data warehousing possesses the coupling of OLAP and GISs. In this research, a spatial data warehouse model has been implemented for weather pattern searching to provide effective data visualization and analysis.

4.1 The Case Study

Weather pattern searching is chosen as a case study to illustrate the implementation of a spatial data warehouse model. Weather pattern refers to repeating weather conditions such as repeating hot and dry weather for several days in a row. Weather patterns usually repeats in a pattern and changes periodically. For example, dry weather continue for two weeks becomes rainy weather for a week (http://www.theweatherprediction.com/habyhints2/450).

There are several meteorological variables that determine weather patterns. Temperature and precipitation values are the basic building blocks and used to illustrate and analyze weather patterns. In this context, weather pattern searching is the term used for the studies to identify the relationship between the different sources determining weather patterns, temperature and precipitation. It provides valuable information for both understanding the weather dynamics and forecasting.

In this context, weather pattern searching can be defined as a decision making process having the goal of discovering how certain attributes within the meteorological data will behave in the future (Han et al., 2001). There are certain kinds of atmospheric models to predict the weather patterns. In these models, it is

difficult to examine large volumes of spatial data. Therefore, emerging as one of the top technologies for the near future, spatial data warehousing and spatial knowledge discovery methods brings another point of view to the subject. It enables discovering new rules and patterns from the vast amounts of both spatial and non-spatial data by the multidimensional structure it encompasses. Therefore, it helps analyzing the data multi-dimensionally and assists the decision making process by extracting more natural and precise knowledge.

Weather pattern searching necessitates discovering general weather patterns according to different combinations of some spatial and non-spatial attributes within the dimensions of space and time. In this context, this thesis demonstrates a framework to discuss the topics in implementing a spatial data warehouse model for weather data and aims to improve the spatio-temporal data exploration techniques by using weather pattern searching as a real world case study.

4.2 Research Questions

This study focuses on the research questions below:

- How spatial data cubes can be built to better support forecasting, trend analysis and hyphothesis generation regarding spatial and non-spatial data?

- How better integration of diverse data can be handled in spatio-temporal multidimensional databases?

- How data classifications and fuzziness can be used in spatio-temporal multi-dimensional databases to improve data exploration?

- How visualization can be improved to assist decision making in spatial data warehouses?

- In a weather data warehouse, which cuboids are meaningful and convey valuable information for weather pattern searching?

4.3 Lifecycle of a Spatial Data Warehouse

In this section, the steps in implementing a spatial data warehouse model and how this model assists spatio-temporal data exploration and analysis will be explained.

A SDSS necessitates a well integration of activities in its lifecycle. To satisfy this, the steps in implementing the spatial data warehouse model as a SDSS are structured on a development lifecycle in Figure 4.11. In this study, lifecycle of a spatial data warehouse is categorized into five basic steps. As seen in Figure 4.11, these steps are requirements definition, data collection, building the data warehouse, integrating GISs and OLAP, and maintanence and growth. In order to illustrate these steps, spatially enabled weather data warehouse has been constructed. Following five phases of the lifecycle of spatially enabled weather data warehouse are given in detail.

4.3.1 Requirements Definition

At the core step of the lifecycle, requirements for the SDSS must be defined. Data warehouses are subject-oriented databases which means that they are built for a specific purpose. Therefore, definition of requirements is the basic step in which the goal of a data warehouse is set. In this step, requierements refers to the user requirements who would be using the warehouse and the analysis environments. In this study, requirements of a spatially enabled weather data warehouse have been analyzed to the requirements definition phase of a data warehouse and given below:

- Different combinations of non-spatial attributes which include precipitation, temperature, time, and spatial attributes such as station and basin dimensions have to be analyzed to capture weather patterns in terms of space, time and attributes such as precipitation, temperature, and etc.

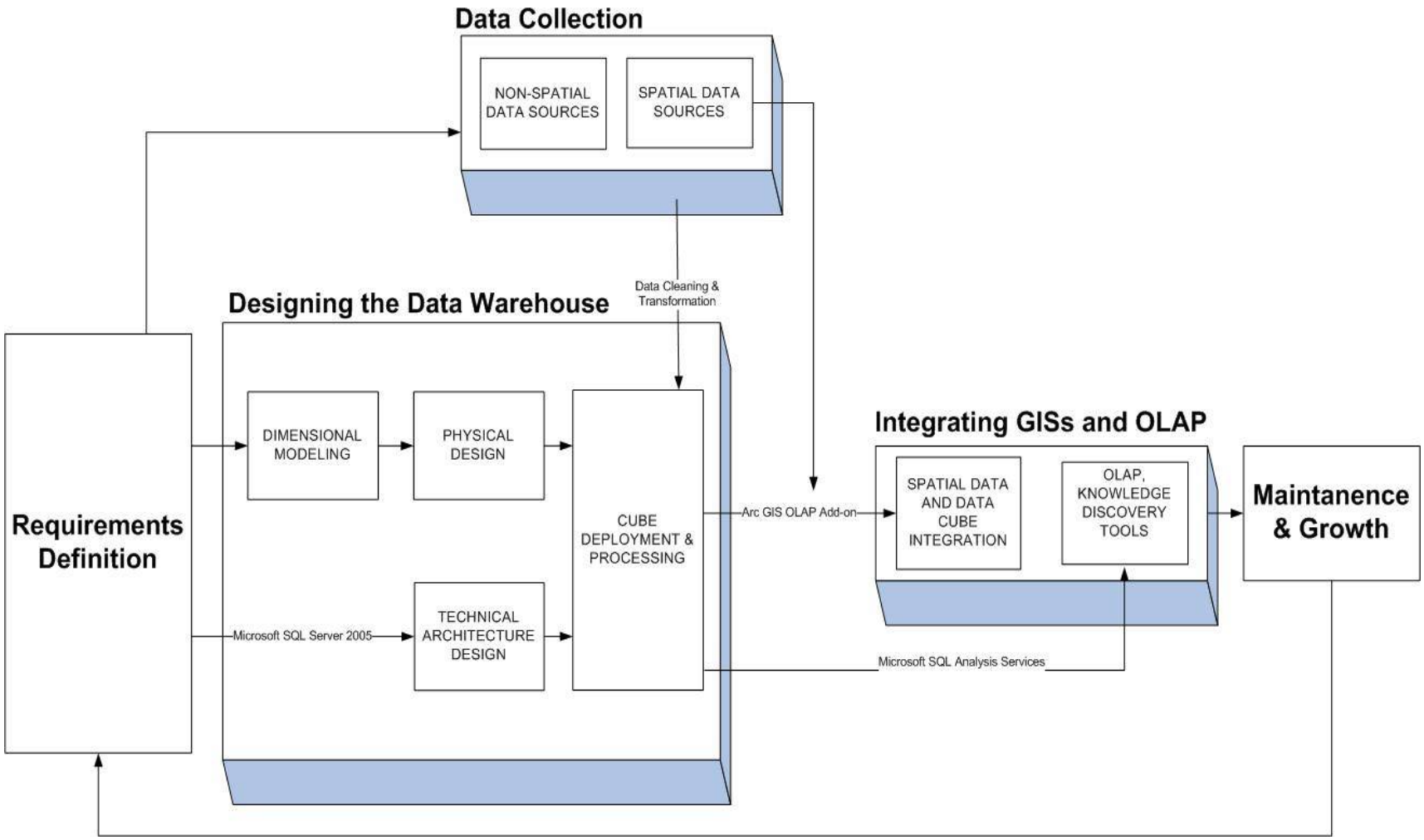


Figure 4.11 Spatial Data Warehouse Implementation Stages

- Different levels of aggregations in dimensions are needed to search for more meaningful rules and patterns. For example, temperature values measured in a region may be needed to query monthly and yearly values. This example represents an aggregation in the time dimension. Additionally, each dimension in the warehouse may have aggregations. For example, the user may need to retrieve temperature values which are classified into range values such as temperature measurements ranging from -10 C to 0 C in an aggregation level. In an upper hierarchical level, the user may only need to retrieve values of cold temperature. In such a case, values ranging from -20 to 0 may be categorized into a fuzzy definition, "cold".

- Different data sources have to be integrated into the warehouse to enable querying data from different combinations and perspectives. The user may like to view precipitation, temperature values geographically and simultaneously.

- In order to find weather patterns monthly, seasonally, and yearly, a wide time horizon with these different levels of aggregations must be provided. This requirement refers to the need for a seperate time dimension to retrieve historical information in different aggregation levels.

- Rapid retrieval of highly summarized weather data have to be provided for performing better analysis. Response time is a crucial criterion in a DSS. Therefore, in a SDSS to perform weather pattern searching, it is also crucial to retrieve valuable information in a timely manner.

The requirements have also effects on the other phases of the lifecyle. They directly affect both building and data collection phases of the warehouse which is illustrated by the arrows in Figure 4.11. First of all, requirements set boundaries for the building stage of the warehouse. Combinations of different attributes that are defined in the requirements definition phase, convey the necessary information to accomplish

dimensional modeling in the building phase of the warehouse. Additionally, requirements regarding the size and content of the warehouse present important notifications such as size and record limits to decide on technical architecture design of the warehouse. Secondly, requirements phase helps to organize the data collection phase. In the data collection stage, type of data, measurements and data collection standards are organized according to the requirements. Therefore, integration of the data into the warehouse can be managed properly. On the other hand, requirements not only affect the other phases but also they may be affected from the other phases. They can be re-defined and changed after maintanence and growth decisions. This relation between the maintenance and growth phase and the requirements phase is illustrated by the arrow between these two phases in Figure 4.11. The weather data warehouse is a non-volatile collection of data, which is updated at standard intervals. When new measurements are collected, they are updated into the warehouse. Therefore, the warehouse grows and as a result, requirements may be subjected to change.

4.3.2 Data Collection

In this study, spatial and non-spatial weather data from different sources are integrated into the spatial data warehouse to perform weather pattern searching in Turkey's national weather data from 1970 to 2006. Precipitation, temperature data, are measured by Turkish State Meteorological Service at the meteorological stations that are located in all regions of Turkey. Monthly total precipitation, monthly average, maximum and minimum temperature data are obtained from Turkish State Meteorological Service in 2006. Temperature and precipitation data used in this study are composed of monthly aggregated data. These data are produced by Turkish State Meteorological Service through aggregating daily measurements. Data sources used in this study are given below:

Weather Stations: Weather stations are spatial data which consist of the point features extracted from the geographic coordinates of the stations (see APPENDIX B).

Basins: Basins are spatial data which consist of basin polygons defining the basin regions of Turkey (see APPENDIX A). Basin data are obtained from Turkish State Hydraulic Works.

Monthly Total Precipitation: Monthly total precipitation are non-spatial data which consist of a derived, monthly total precipitation values which are actually measured daily (see APPENDIX C).

Monthly Average Temperature: Monthly average temperature are non-spatial data which consist of a derived, monthly average temperature values which are actually measured daily (see APPENDIX D).

Monthly Maximum Temperature: Monthly maximum temperature are non-spatial data which consist of a derived, maximum temperature values in months
(see APPENDIX E).

Monthly Minimum Temperature: Monthly minimum temperature are non-spatial data which consist of a derived, minimum temperature values in months
(see APPENDIX F).

4.3.3 Data Warehouse Design

In this study, designing phase of a data warehouse is categorized into four main stages: dimensional modelling, physical design, technical architecture design and cube deployment and processing.

4.3.3.1 Dimensional Modeling and Physical Design

Depending on the requirements of warehousing and the data available, dimensional modeling is the first step in designing a data warehouse. A schema model, which describe the entities, attributes, relationships is defined in this step. Then, defined schema is translated into database structures during the physical design process. In

this step, entities are mapped on to tables, attributes are mapped on to columns, constraints and indexes are designed. Values in dimensions are transformed into fuzzy description in order to improve the interpretation of data. In this section, dimensional modeling and physical design phases are given below.

The dimensional structure proposed in this study is shown in Figure 4.12. This structure consists of a fact table and four basic dimension tables. In this structure, the fact table, StationMeasures contains the detailed weather data. Columns in this table are the foreign key columns which are referenced through primary-foreign key relationships between the fact table and the dimension tables; Precipitation, Temperature, Time, Station and Basin. Primary key is abbreviated as PK and foreign key is abbreviated as FK in Figure 4.12.

Precipitation dimension is designed to support two levels of hierarchies. The first hierarchy level is the description of precipitation values. Values in this hierarchy are presented by the fuzzy words: dry, fair and wet. These values are defined according to the yearly total precipitation values of stations. The second hierarchy level is the corresponding ranges of precipitation values. There are seven ranges defined in this hierarchy to represent the precipitation values. The precipitation dimension and its corresponding levels of hierarchies are given in the columns of Table 4.8. In addition to that, data types of the columns are given in Table 4.9.

Similar to precipitation dimension, temperature dimension is designed to support two levels of hierarchies. The first level is the description of temperature values. Values in this hierarchy are presented by the fuzzy words: extremely cold, cold, warm, hot and extremely hot. These values are defined according to the temperature measurements. The temperature dimension and its corresponding levels of hierarchies are given in the columns of Table 4.10. In addition to that, data types of the columns are given in Table 4.11.

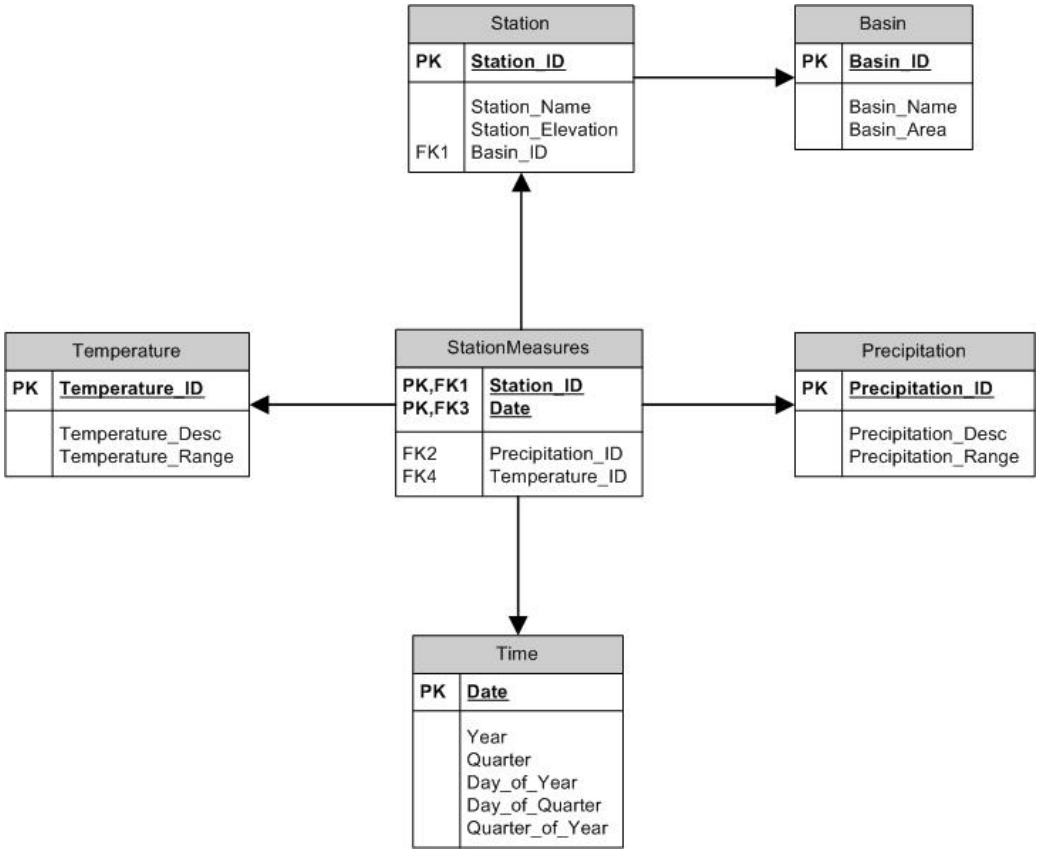


Figure 4.12 Snowflake Schema of Weather Cubes

Table 4.8 Precipitation Dimension Table

PK_Precipitation_ID	Precipitation_Description	Precipitation_Ranges
1	dry	below 200
2	dry	200 to 400
3	dry	400 to 600
4	fair	600 to 800
5	fair	800 to 1000
6	wet	1000 to 1200
7	wet	above 1200

Table 4.9 Data Types of Precipitation Dimension

Table Column	Data Type
PK_Precipitation	int
Precipitation_Desc	nvarchar(50)
Precipitation_Range	nvarchar(50)

Table 4.10 Temperature Dimension Table

PK_Temperature_ID	Temperature_Description	Temperature_Range
1	extremely cold	below -20
2	cold	-20 to -10
3	cold	-10 to 0
4	cold	0 to 10
5	warm	10 to 15
6	warm	15 to 20
7	hot	20 to 25
8	hot	25 to 30
9	hot	30 to 35
10	extremely hot	above 35

Table 4.11 Data Types of Temperature Dimension

Table Column	Data Type
PK_Temperature	int
Temperature_Desc	nvarchar(50)
Temperature_Range	nchar(10)

Station dimension consists of one hierarchy level which is based on the basin attribute. Basin attribute presents the basin that each station is in and extracted through applying point in polygon analysis to the point station features and polygon basin features. Values in this hierarchy are presented by the basin numbers. These values are defined according to the basins in Turkey which is shown in Table 4.12.

Table 4.12 Basin Hierarchy

PK_Basin_ID	Basin Name	Basin Area km^2
19	Asi Havzasi	791239
17	Dogu Akdeniz Havzasi	2180703
8	Bati Akdeniz Havzasi	2122388
10	Burdur Havzasi	630617
9	Antalya Havzasi	2033084
20	Ceyhan Havzasi	2159850
6	Kucuk Menderes Havzasi	705973
26	Dicle Havzasi	5453970
7	Buyuk Menderes Havzasi	2613320
11	Akarcay Havzasi	798258
18	Seyhan Havzasi	2224158
5	Gediz Havzasi	1703401
16	Konya Kapali Havzasi	5003780
25	Van Golu Havzasi	1797698
4	Kuzey Ege Havzasi	997362
21	Firat Havzasi	12211739
2	Marmara Havzasi	633129
3	Susurluk Havzasi	2433197
12	Sakarya Havzasi	6335774

Table 4.12 cont'd

2	Marmara Havzasi	765007
14	Yesilirmak Havzasi	3962795
15	Kizilirmak Havzasi	8219728
22	Dogu Karadeniz Havzasi	2284456
23	Coruh Havzasi	2024870
24	Aras Havzasi	2811457
13	Bati Karadeniz Havzasi	2892975
1	Meric-Erhene Havzasi	1444414
2	Marmara Havzasi	912577

Basin hierarchy provides analysis of stations in accordance with the basins they are completely within. Therefore, this hierarchy implicitly convey spatial information as an attribute of each station and presents a spatial to non-spatial type of generalization. Finally, Station dimension consist of the following attributes: PK_Station_ID, Station_Name, Station_Elevation, Basin_ID. A part of the table of station dimension is given in Table 3.13.

Table 3.13 Station Dimension Table

PK_Station_ID	Station_Name	Station_Elevation	FK_Basin_ID
17022	ZONGULDAK	137	13
17024	İNEBOLU	64	13
17026	SİNOP	32	13
17030	SAMSUN	4	14
…..	…….	……..	…….

In the multidimensional structure, historical aggregations and hierarchies are handled by Time dimension. In this study, Time dimension contains six levels of detail: Year, Quarter, Day_of_Year, Day_of_Quarter and Quarter_of_Year. The data to be loaded into the warehouse have months as the lowest level of detail. Because in weather pattern searching only long time intervals convey recognizable and meaningful patterns. Therefore monthly data at the lowest level of detail are stored and aggregated into quarters and years.

4.3.3.2 Technical Architecture Design

In this study, the spatial data warehouse is implemented by storing non-spatial and spatial data seperately in order to avoid developing complex integration modules. The data warehouse containing the non-spatial data and concept hierarchies are stored in Microsoft SQL Server. Microsoft SQL Server Analysis Services 2005 is used to implement OLAP into the warehouse. On the other hand, spatial data are stored in ESRI shp file format. ARC GIS Add-on for OLAP is used to integrate the data warehouse with spatial data.

4.3.3.3 Data Cleaning and Transformation

After dimensions, measures, aggregations and hierarchies are defined, the next step is the preparation of data to be loaded into the warehouse. Raw data to be used in a warehouse are usually collected by minimizing data entry through a relational structure. However, in a data warehouse, aggregations and abstractions are needed in order to optimize analysis. Therefore, data obtained in the data collection phase are cleaned before uploading into the warehouse.

Data cleaning phase necessitates aggregating and abstracting data in order to better support analysis in a warehouse. This aggregation and abstraction needs result in transforming data into more meaningful information for analysis. Data cleaning and transformation phase depend on the dimensional structure modeled in the design phase of the warehouse. Data cleaning and transformation tasks are involved in many

steps in the lifecycle of a spatial data warehouse model that is proposed in this study. In Figure 4.13, data transformation phases of this model is presented.

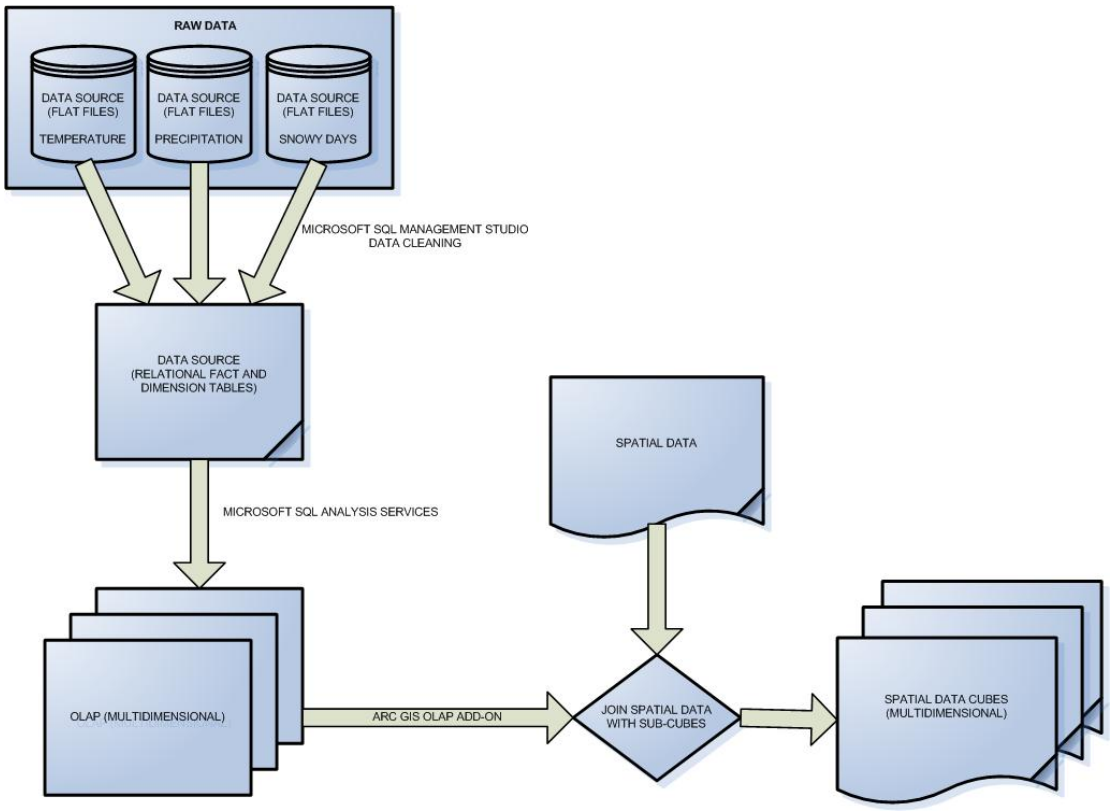


Figure 4.13 Data Transformation Chart of a Spatial Data Warehouse

As seen in Figure 4.13, first of all, raw data from different sources are cleaned and integrated into a relational database. This relational database hold the data used to load values into the multidimensional structure. In this study, before uploading data into the multidimensional structure data are cleaned and abstracted in data cleaning process by using some procedures in Microsoft SQL Server Management Studio.

In the very first step of the data cleaning process, non-spatial data from different sources are integrated into a table. In order to perform this integration, these data are joined into a single table containing multiple attributes of different sources. Raw data

has different field names and values. Thus, data migration commands are used to standardize the field names and values. Several SQL commands have been used to migrate and standardize the data to load into the warehouse. To illustrate this, extraction and standardization of the fields of Basin data is shown in Table 4.14. Through using a SQL statement, required part of the data from original source is filtered and standardized in order to be loaded into the Basin Hierarchy which is shown in Table 4.12.

Table 4.14 Extracting and Standardizing Basin Hierarchy

Original Field Name	**Standardized Field Name**
OBjectID	Not used in basin hierarchy
ShapeLength	Not used in basin hierarchy
ShapeArea	**Basin_Area**
HavzaNo	**Basin_ID**
Havza_Rome	Not used in basin hierarchy
Havza_Ad	**Basin_Name**

The raw data represent monthly measurements taken by different instruments located at the same stations. Therefore, different data sources have the same primary key combination which consist of Station_ID, year and month information of the measurement and this provides combining them into the same table. As a result, these different data sources are joined into a single table as defined in Figure 4.14. In this join operation, month and year attributes with data type of integer are translated into the attribute, Date which holds the date time information of the rows.

Storing data in a relational table before uploading into the warehouse prevents performing seperate uploading procedures for each data source. Furthermore, it helps to handle inconsistencies by cleaning data by previously defined procedures.

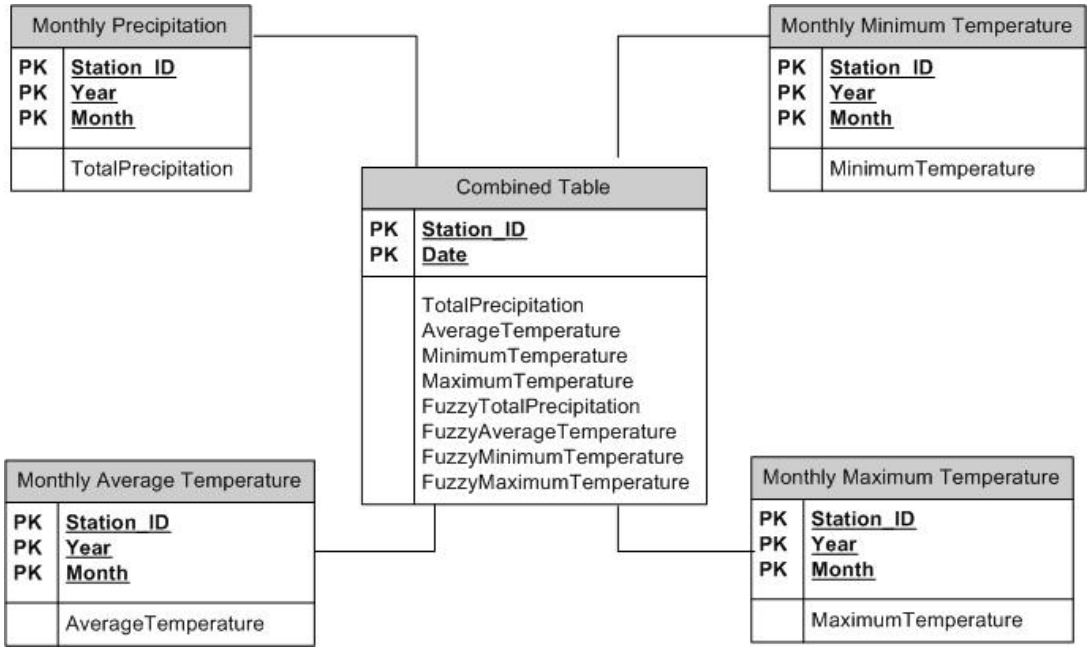


Figure 4.14 Raw Data Source Integration

Not only the field names but also the values stored in these fields are transformed before loading into the warehouse. After cleaning and integrating into a relational table, numeric data, which consist of the measurements such as total precipitation, maximum temperature, average temperature are transformed into fuzzy descriptions to enhance the understandibility of the results of queries. Meaningful queries that include several dimensions are abstracted into fuzzy words which describe the ranges and descriptions of those values. These fuzzy words are defined in accordance with the dimensions which are designed in the dimensional modeling phase. Thus, the values in the combined table are transformed into the corresponding attributes in each dimension. To illustrate this, temperature measurements are converted into the range classes by applying the SQL query written in Figure 4.15. The query given in Figure 4.15, presents a part of the SQL command to abstract the average temperature values into categories which are defined in the dimension of Temperature. Finally, fuzzy average temperature values, which consist of integer unique values of temperature categories are loaded into the warehouse.

```
UPDATE CombinedTable
SET CombinedTable.FuzzyAverageTemperature = 1
WHERE (((CombinedTable.AverageTemperature) > -20 And
(CombinedTable. AverageTemperature) <= -10));

UPDATE CombinedTable
SET CombinedTable.FuzzyAverageTemperature = 2
WHERE (((CombinedTable.AverageTemperature) > -10 And
(CombinedTable. AverageTemperature) <= 0));
….
```

Figure 4.15 Sample SQL Query for Abstracting Measurements

After the relational database holding the relational fact and dimension tables has been constructed as in Figure 4.13, the second transformation is performed by moving the relational data into multidimensional structures by cube processing which is given in detail in the next section “Cube Processing”. After that, the last transformation step is performed by extracting sub-cubes from the multidimensional structure and joining sub-cubes with spatial data files. Also, this step is given in detail in the section, “Adding Spatial Dimension to Weather Cubes”.

4.3.3.4 Cube Processing

In the lifecycle of the warehouse, dimensional modeling phase is completed by creating a relational schema which shows the relationship between the fact and the dimension tables (Figure 4.12). After that, a relational database is created in order to clean and transform the data to be loaded into the warehouse. The cube processing step is the operation that loads data from the relational database into the multidimensional cube.

The schema in Figure 4.12 enables inserting data into the dimension and fact tables in preparation for viewing those data in the cube. Cube structure of the case study, Weather Cubes is illustrated in Figure 4.16. In this figure, data source defines the connection information for the source database; data source view provides representation of the table structures and relationships that are used when making modifications to the cube and when loading data into the cube. Data source view is added by the schema after the relational structures have been built. It defines which tables from the warehouse contain the data to load into the dimension. The cube is processed by right-clicking the related cube, Weather DW.cube in Figure 4.16 and selecting Process option. When a cube is processed, Analysis Services load data from the fact table into propriatery data structures.

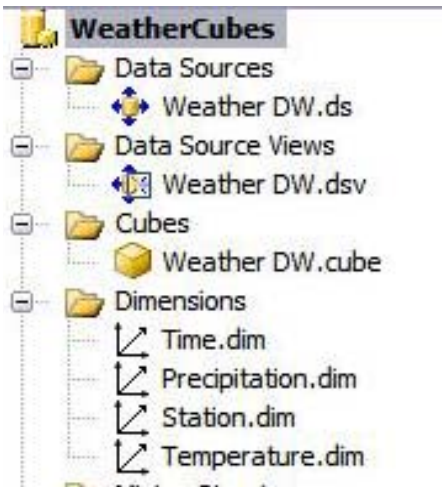


Figure 4.16 Cube Structure

4.3.4 Adding Spatial Dimension to Weather Cubes

The data warehouse containing the non-spatial data and attribute hierarchies are stored in a multidimensional warehouse structure in the form of a data cube which is fully materialized. In order to perform spatial exploration and analysis on the cube structure two steps are performed in this study.

4.3.4.1 Extracting Sub-cubes

First of all, sub-cubes are extracted from the fully materialized cube according to the point of analysis. Sub-cube extraction is performed by slice and dice operations using Arc GIS OLAP Add-on Interface in Arc Catalog. This interface provides a connection to the warehouse and helps to browse the cube with required dimensions and measures. The steps to create an OLAP connection are illustrated in the following figures, Figure 4.17 and Figure 4.18. In the first step, the corresponding data provider is chosen to support SQL Analysis Services as seen in Figure 4.17. In the next step, data connection properties are set: data source server is defined and related cube is selected as seen in Figure 4.18.

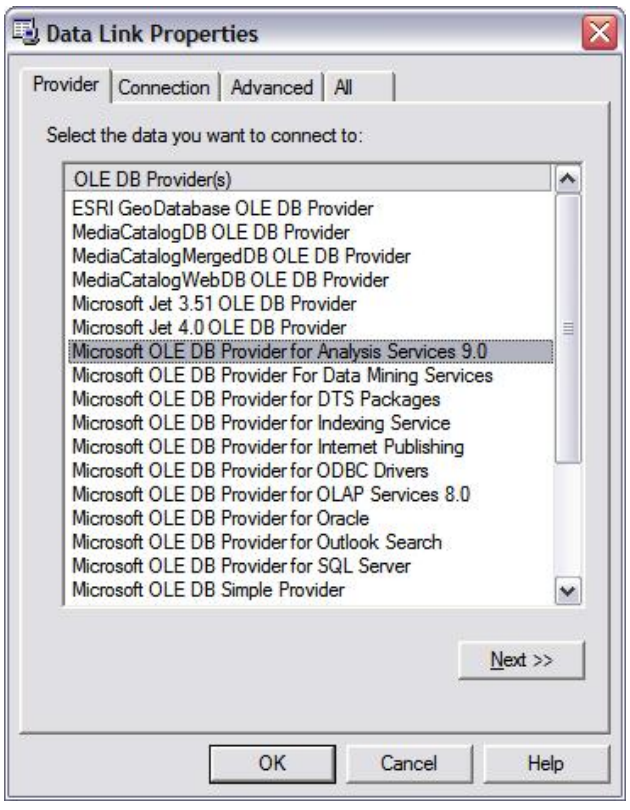


Figure 4.17 Data Provider for OLAP Connection

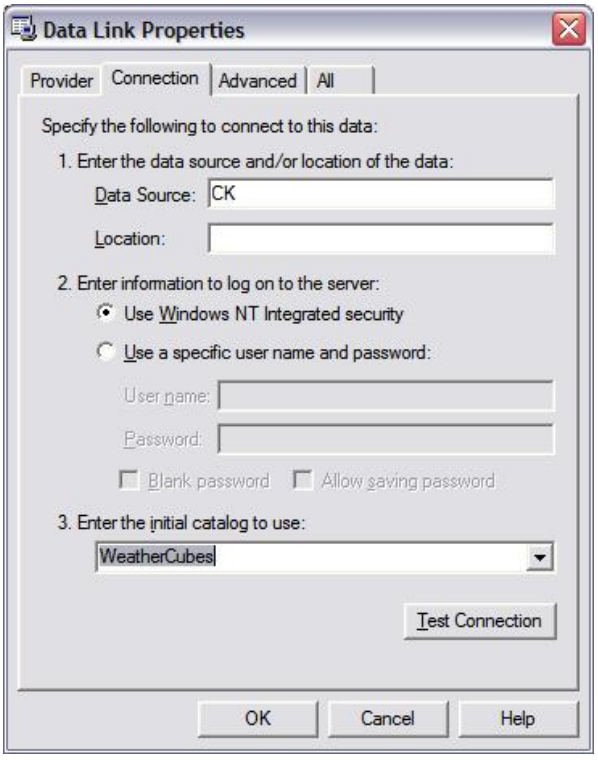


Figure 4.18 OLAP Connection

After the connection has been established, the next step is to create a two dimensional OLAP table from the fully materialized weather cube. First, corresponding cube is selected which is illustrated in Figure 4.19.

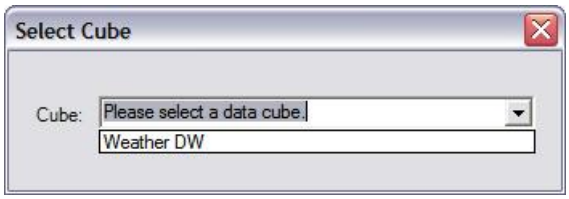


Figure 4.19 Selecting the Fully-materialized Cube

When the cube is selected, an empty data grid is displayed as in Figure 4.20. Sub-cube structure is defined by dragging dimensions and measures into this grid. Dimensions and measures are selected from PivotTable Field List as in Figure 4.21.

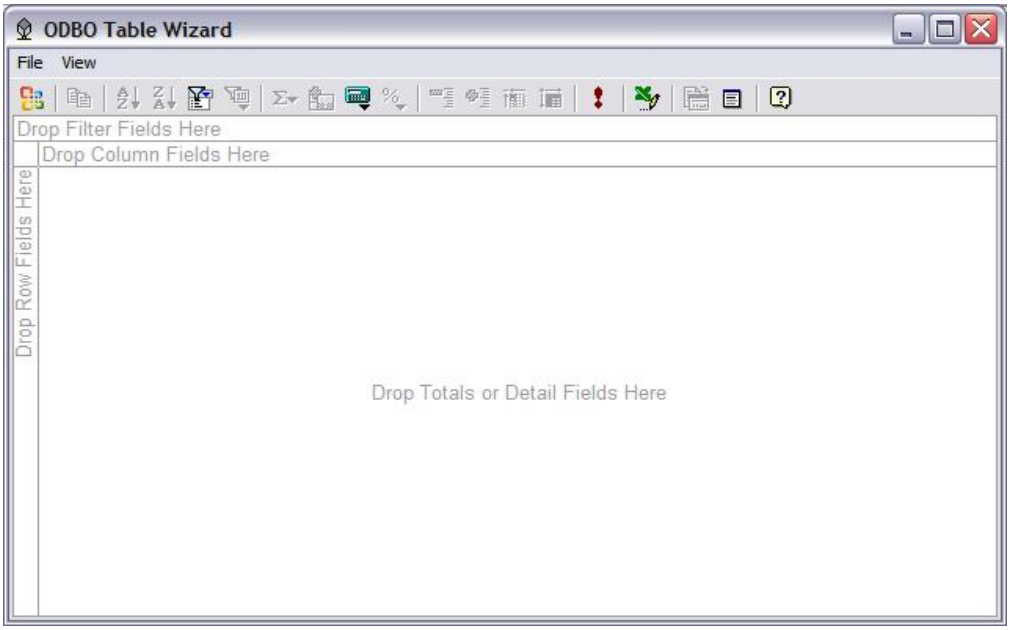


Figure 4.20 Pivot Table

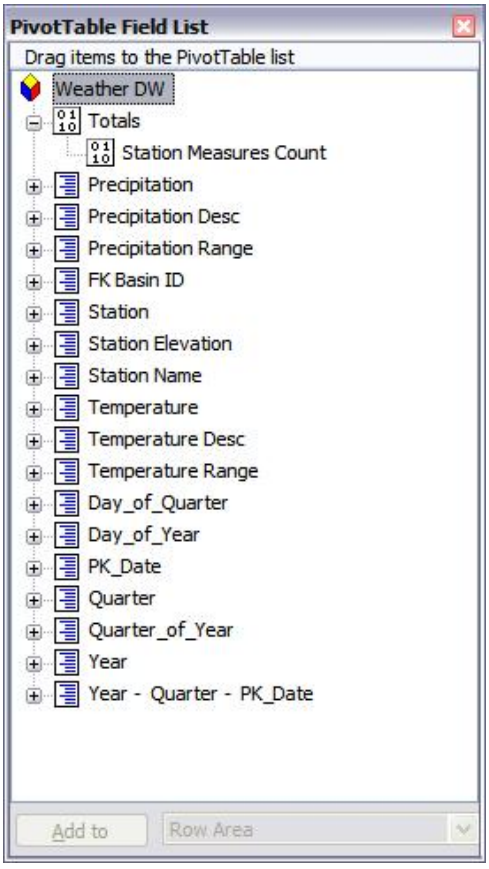


Figure 4.21 PivotTable Field List

This interface helps to construct MDX queries by arranging dimensions and measures visually on a pivot table model. MDX queries are generated automatically after dragging and dropping the measures and dimensions into the pivot table. As a result, pivot table which represents the sub-cube is created. To illustrate this, an example of a sub-cube is created according to the dimensions, measure, hierarchies and filters defined in Figure 4.22. In this example, precipitation values are taken as the central point of analysis. The goal of this sub-cube is to search for patterns based on station measure counts of all stations according to precipitation range values, in years from 2000 to 2006 and having "warm" temperature values. Therefore, slice and dice operation is performed for Precipitation dimension. Temperature and Time dimensions are used as filtering dimensions and taken into analysis to narrow down the search to extract more meaningful patterns. Therefore, while more detailed hierarchy level, Precipitation Range is selected, in other words, drill-down operation is performed for Precipitation Dimension, less detailed hierarchy levels are selected, in other words, roll-up operation is performed and filters are applied for Time and Temperature dimensions. As in Figure 4.22, only "warm" values are taken in Temperature dimension and only the years from 2000 to 2006 are taken to narrow down the cube.

A part of the result table of the sample sub-cube is given in Figure 4.23. In this table, station counts are calculated according to the dimensions and filters defined in the sub-cube.

Dimensions:

Station, Temperature, Precipitation, Time

Measures:

Station Measures Count

Hierarchies used in dimensions:

Station – Station

Temperature – Temperature Desc

Precipitation – Precipitation Range

Time - Year

Filters:

Temperature Desc – warm

Time – Year – years from 2000 to 2006

Figure 4.22 Sample Sub-cube

4.3.4.2 Joining Sub-cubes with Spatial Data

Secondly, extracted sub-cubes are joined with spatial data through a spatial dimension. In the example table given in Figure 4.23, Station dimension is joined with Meteorological Stations' point shp file in ArcMap (Appendix A.).

The attributes of the sub-cube consists of the following attribute fields in Table 4.15. In this table primary key is Station_ID attribute and other fields are automatically generated by Arc GIS OLAP Add-on after the sub-cube is saved as a pivot table as in Figure 4.23. The other fields represent the measure counts of stations by precipitation range values, years and temperature description filter defined in the sample sub-cube. For example, the field just below the primary key field in Table 4.15 represents each stations' precipitation ranging from 1000 to 1200 in the year of 2000.

Joining operation starts with opening the sub-cube table which is generated in the previous step in ArcMap. The shp file containing the spatial data is joined with the

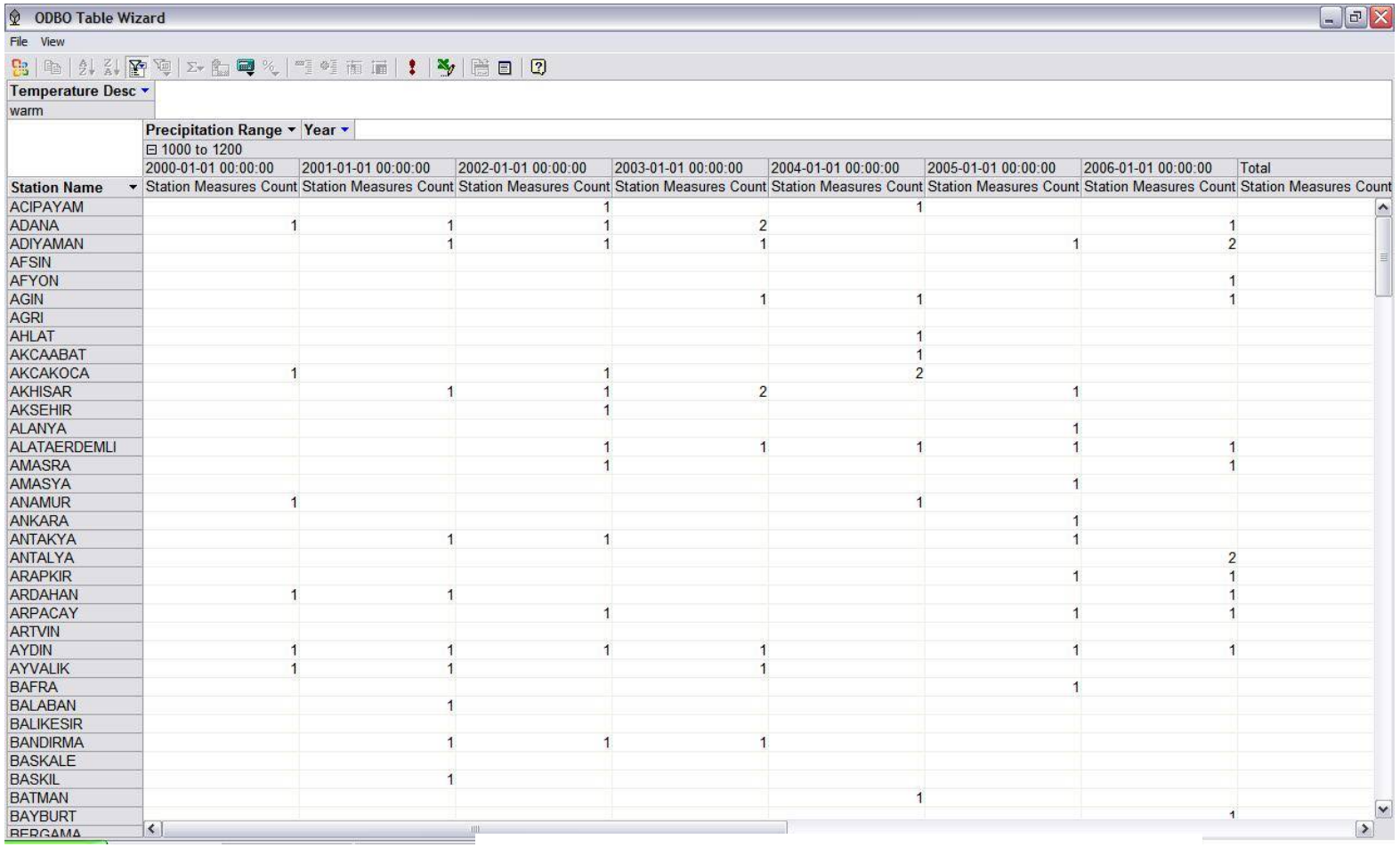
sub-cube through the primary key, Station_ID field. After the join operation, Station shp file includes all the fields of the sub-cube listed in Table 4.15. Therefore, spatial data analysis can be performed on the spatially enabled weather cubes.

4.3.5 Spatial Data Exploration and Analysis of Weather Data

In order to perform spatial data exploration and analysis on a spatial data cube structure, there are two basic steps to perform.

First, queries have to be identified to meet the requirements of the spatial data warehouse. Some of the most meaningful and frequent queries that would represent meaningful patterns in weather data are listed in order to evaluate the implementation of the model to weather pattern searching. These queries are evaluated in the following section.

Second, for each query, a sub-cube is extracted by performing slice and dice, drill-down, roll-up and filtering operations by using Arc GIS OLAP Add-on's pivot table interface as in the steps which are illustrated in Figure 4.17 - Figure 4.22.



Station Name	2000-01-01 00:00:00	2001-01-01 00:00:00	2002-01-01 00:00:00	2003-01-01 00:00:00	2004-01-01 00:00:00	2005-01-01 00:00:00	2006-01-01 00:00:00	Total
	Station Measures Count	Station Measures Count	Station Measures Count	Station Measures Count	Station Measures Count	Station Measures Count	Station Measures Count	Station Measures Count
ACIPAYAM			1		1			
ADANA	1	1	1	2			1	
ADIYAMAN		1	1	1		1	2	
AFSIN								
AFYON							1	
AGIN				1	1		1	
AGRI								
AHLAT					1			
AKCAABAT					1			
AKCAKOCA	1		1		2			
AKHISAR		1	1	2		1		
AKSEHIR			1					
ALANYA						1		
ALATAERDEMLI			1	1	1	1	1	
AMASRA			1				1	
AMASYA						1		
ANAMUR	1				1			
ANKARA						1		
ANTAKYA		1	1			1		
ANTALYA							2	
ARAPKIR						1	1	
ARDAHAN	1	1					1	
ARPACAY			1			1	1	
ARTVIN								
AYDIN	1	1	1	1		1	1	
AYVALIK	1	1		1				
BAFRA						1		
BALABAN		1						
BALIKESIR								
BANDIRMA		1	1	1				
BASKALE								
BASKIL		1						
BATMAN					1			
BAYBURT							1	
BERGAMA								

Figure 4.23 Sample Pivot Table

Table 4.15 Sample Sub-cube Fields

Field Names
Primary Key – Station_ID
below 200.2000-01-01 00:00:00.Station Measures Count
below 200.2001-01-01 00:00:00.Station Measures Count
below 200.2002-01-01 00:00:00.Station Measures Count
below 200.2003-01-01 00:00:00.Station Measures Count
below 200.2004-01-01 00:00:00.Station Measures Count
below 200.2005-01-01 00:00:00.Station Measures Count
below 200.2006-01-01 00:00:00.Station Measures Count
200 to 400.2000-01-01 00:00:00.Station Measures Count
200 to 400.2001-01-01 00:00:00.Station Measures Count
200 to 400.2002-01-01 00:00:00.Station Measures Count
200 to 400.2003-01-01 00:00:00.Station Measures Count
200 to 400.2004-01-01 00:00:00.Station Measures Count
200 to 400.2005-01-01 00:00:00.Station Measures Count
200 to 400.2006-01-01 00:00:00.Station Measures Count
400 to 600.2000-01-01 00:00:00.Station Measures Count
400 to 600.2001-01-01 00:00:00.Station Measures Count
400 to 600.2002-01-01 00:00:00.Station Measures Count
400 to 600.2003-01-01 00:00:00.Station Measures Count
400 to 600.2004-01-01 00:00:00.Station Measures Count
400 to 600.2005-01-01 00:00:00.Station Measures Count
400 to 600.2006-01-01 00:00:00.Station Measures Count
600 to 800.2000-01-01 00:00:00.Station Measures Count
600 to 800.2001-01-01 00:00:00.Station Measures Count
600 to 800.2002-01-01 00:00:00.Station Measures Count
600 to 800.2003-01-01 00:00:00.Station Measures Count
600 to 800.2004-01-01 00:00:00.Station Measures Count
600 to 800.2005-01-01 00:00:00.Station Measures Count
600 to 800.2006-01-01 00:00:00.Station Measures Count
……

CHAPTER 5

RESULTS AND DISCUSSION

The generated sub-cubes from the defined queries which are intended to search for weather patterns are grouped into two categories according to two different sets of data. The results of these queries are shown and evaluated in this section. The first set is based on Station Cubes which are used to extract valuable information that are based on weather stations. Queries beginning from Query 1 to Query 9 are based on station cubes. The second set is based on Basin Cubes which enable searching patterns that are related with basins. In this type of queries, station measure counts are aggregated with respect to basins. Query 10 is an example of this set. In addition to these types of queries, Query 11 and Query 12 are developed to search for the relationship between height of the stations, temperature and precipitation.

5.1 Evaluation of Queries

Query 1

What is the count of station measures for each station according to the categories of precipitation range values, which have cold and extremely cold temperature description values in the years after 2000?

The result of Query 1 shows the number of station measure counts with respect to the distribution of their precipitation range values for each station, having temperature description values of cold and extremely cold, and the years after 2000. Geovisualization of this query helps to extract a significant weather pattern in Fırat and Dicle Basins of Turkey. The bar charts which define the distribution of precipitation range values for each station in Turkey, show that for most of the stations in Fırat and Dicle basins, when temperature is cold and extremely cold,

yearly precipitation range values have been measured mostly above 1200 mm per square meter among the measurements taken after 2000.

Query 2

What is the count of station measures for each station which has precipitation range values above 1200, temperature description values of hot and extremely hot in the years after 2000?

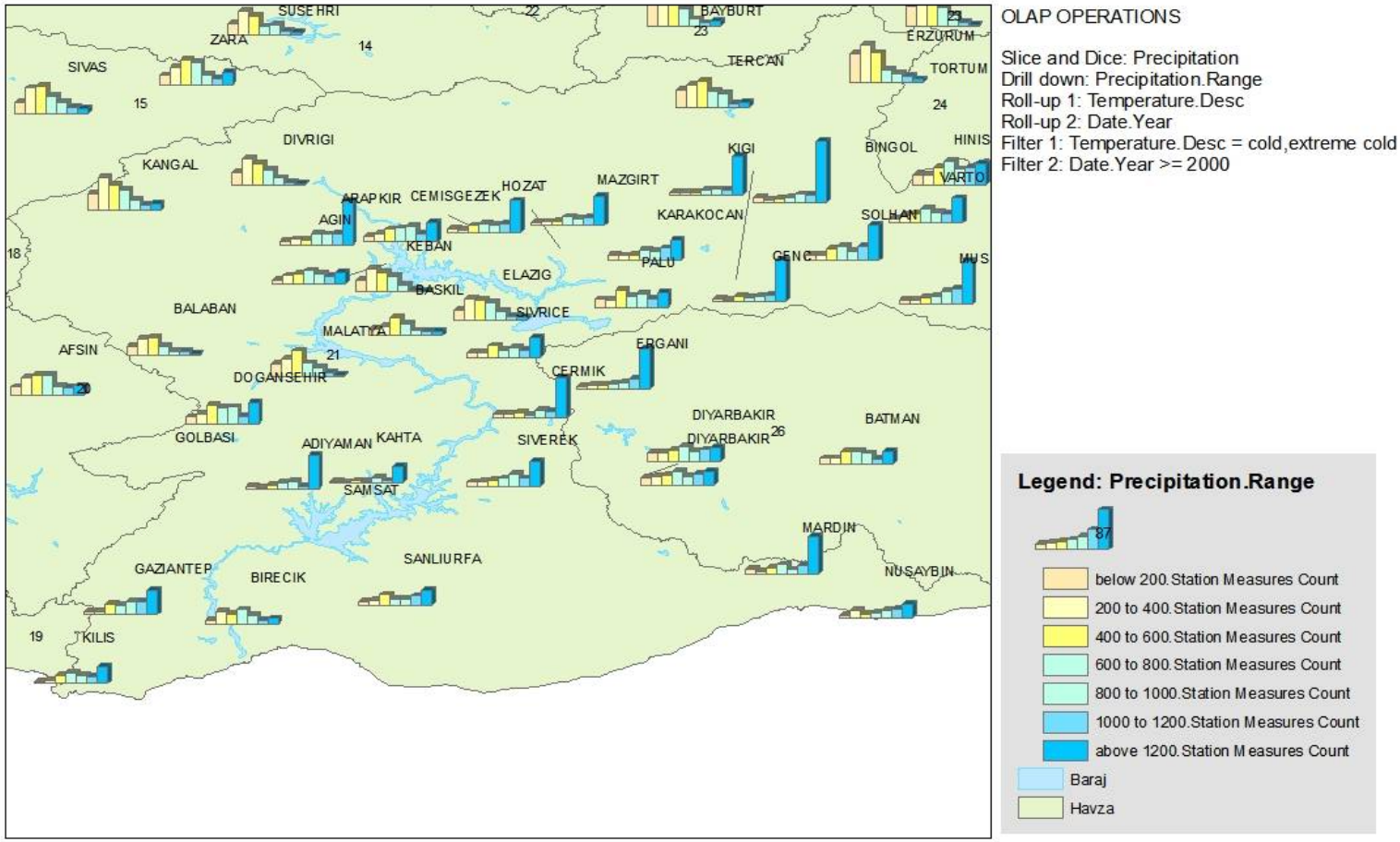
The result of Query 2 shows the number of station measure counts with respect to the distribution of their precipitation range values above 1200 with respect to the years after 2000 for each station, having temperature description values of hot and extremely hot. Geovisualization of this query helps to extract a significant weather pattern in Giresun, Rize and Hopa stations of Doğu Karadeniz Basin. The bar charts show that only Giresun, Rize and Hopa stations in Doğu Karadeniz basin, have precipitation values above 1200 mm per square meter when temperature is hot and extremely hot, among the measurements taken after 2000.

Query 3

What is the count of station measures for each station which has precipitation range values below 200, temperature description values of hot and extremely hot in the years after 2000?

The result of Query 3 shows the number of station measure counts with respect to the distribution of their precipitation range values below 200 with respect to the years after 2000 for each station, having temperature description values of hot and extremely hot. Geovisualization of this query helps to extract significant weather patterns in Büyük Menderes, Küçük Menderes, Kuzey Ege, Gediz and Batı Akdeniz Basins. The bar charts show that in these basins precipitation values have been measured continuously below 200 mm per square meter among the measurements taken after 2000 when temperature is hot and extremely hot.

Geovisualization of Query 1: What is the count of station measures for each station
according to the categories of precipitation range values,
which have cold and extreme cold temperature description values in the years after 2000?



OLAP OPERATIONS

Slice and Dice: Precipitation
Drill down: Precipitation.Range
Roll-up 1: Temperature.Desc
Roll-up 2: Date.Year
Filter 1: Temperature.Desc = cold,extreme cold
Filter 2: Date.Year >= 2000

Legend: Precipitation.Range

- below 200.Station Measures Count
- 200 to 400.Station Measures Count
- 400 to 600.Station Measures Count
- 600 to 800.Station Measures Count
- 800 to 1000.Station Measures Count
- 1000 to 1200.Station Measures Count
- above 1200.Station Measures Count
- Baraj
- Havza

Figure 5.24 Geovisualization of Query 1

Geovisualization of Query 2: What is the count of station measures for each station which has precipitation range values above 1200,
temperature description values of hot and extreme hot in the years after 2000?

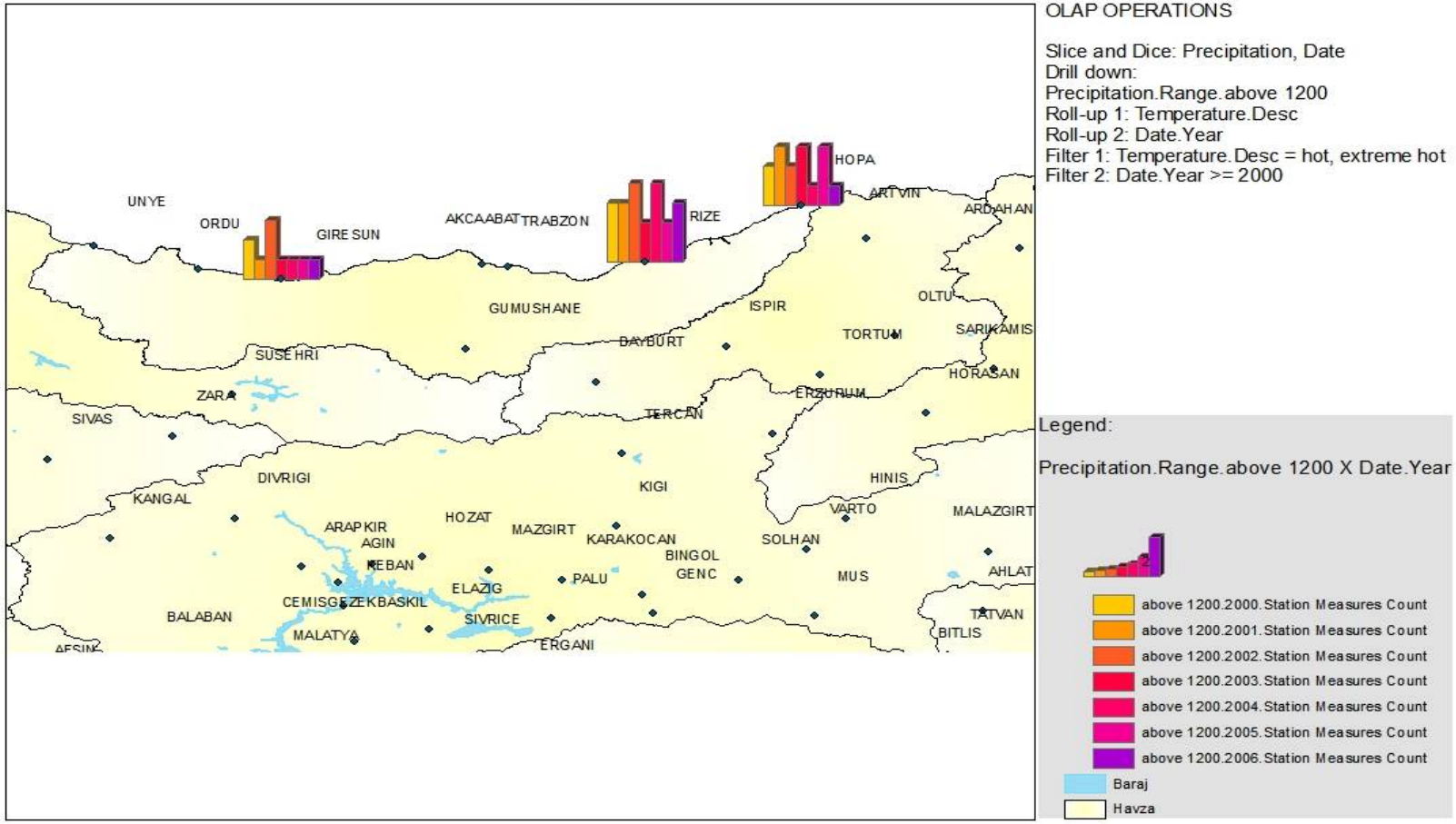
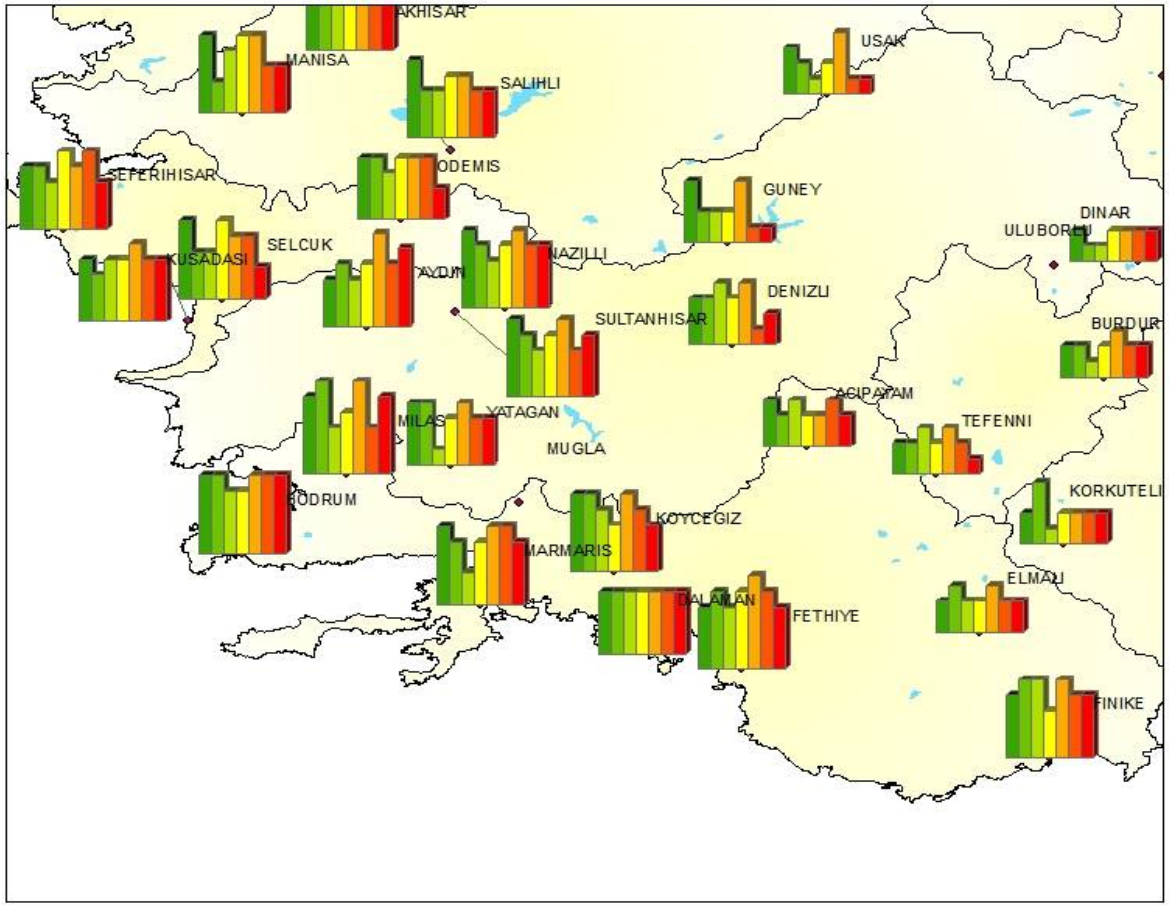


Figure 5.25 Geovisualization of Query 2

Geovisualization of Query 3: What is the count of station measures for each station which has precipitation range values below 200,
temperature description values of hot and extreme hot in the years after 2000?



OLAP OPERATIONS

Slice and Dice: Precipitation, Date
Drill down:
Precipitation.Range.below 200
Roll-up 1: Temperature.Desc
Roll-up 2: Date.Year
Filter 1: Temperature.Desc = hot, extreme hot
Filter 2: Date.Year >= 2000

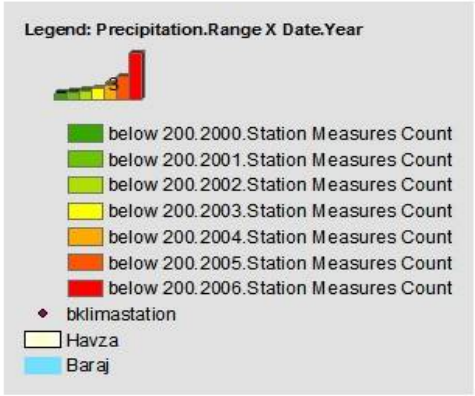


Figure 5.26 Geovisualization of Query 3

Query 4

What is the count of station measures for each station which has precipitation range values below 200, temperature description values of hot and extremely hot in the years before 2000?

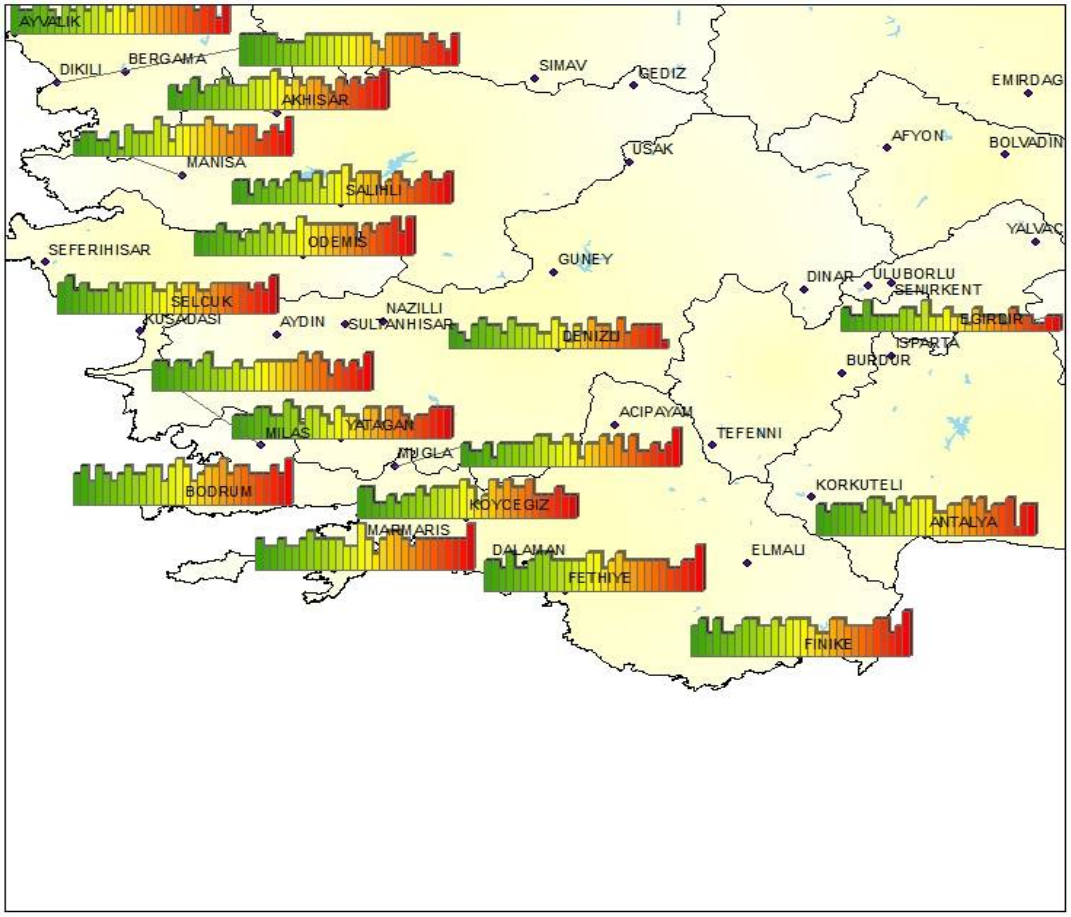
Query 4 is generated to search for a change in weather patterns in the years before and after 2000. Thus, the only difference between Query 3 and Query 4 is that in Query 4, the years before 2000 has been taken into consideration instead of the years after 2000. According to the comparison of these two queries, Büyük Menderes, Küçük Menderes, Kuzey Ege, Gediz and Batı Akdeniz Basins have been continously measured below 200 precipitation range values both before and after 2000. Therefore, similar type of weather patterns are captured in these two queries.

Query 5

What is the count of station measures for each station according to the categories of precipitation range values, which have warm temperature description values in the years after 2000?

The result of Query 5 shows the number of station measure counts with respect to the distribution of their precipitation range values for each station, having warm temperature description values and the years after 2000. Geovisualization of this query helps to extract a significant weather pattern along the southern coasts of Turkey. The bar charts which define the distribution of precipitation range values for each station in Turkey, show that for most of the stations in the southern coast of Turkey when temperature is warm, yearly precipitation range values have been measured mostly above 1200 mm per square meter among the measurements taken after 2000. On the other hand, in the same combination of attributes, yearly precipitation range values have been measured mostly below 200 mm per square meter in inner regions of Turkey.

Geovisualization of Query 4: What is the count of station measures for each station which has precipitation range values below 200,
temperature description values of hot and extreme hot in the years before 2000?



OLAP OPERATIONS

Slice and Dice: Precipitation, Date
Drill down:
Precipitation.Range.below 200
Roll-up 1: Temperature.Desc
Roll-up 2: Date.Year
Filter 1: Temperature.Desc = hot, extreme hot
Filter 2: Date.Year < 2000

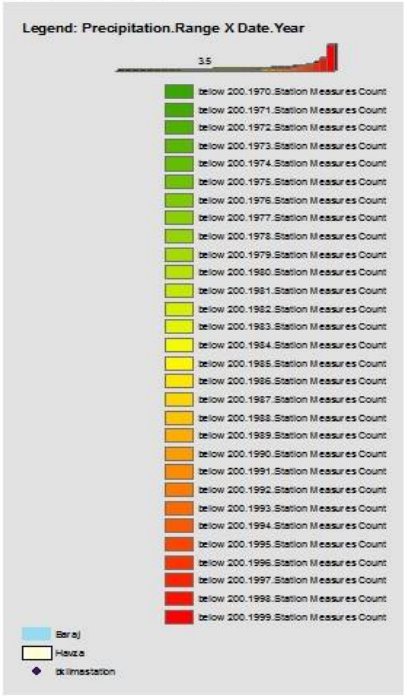


Figure 5.27 Geovisualization of Query 4

Geovisualization of Query 5: What is the count of station measures for each station according to the categories of precipitation range values, which have warm temperature description values in the years after 2000?

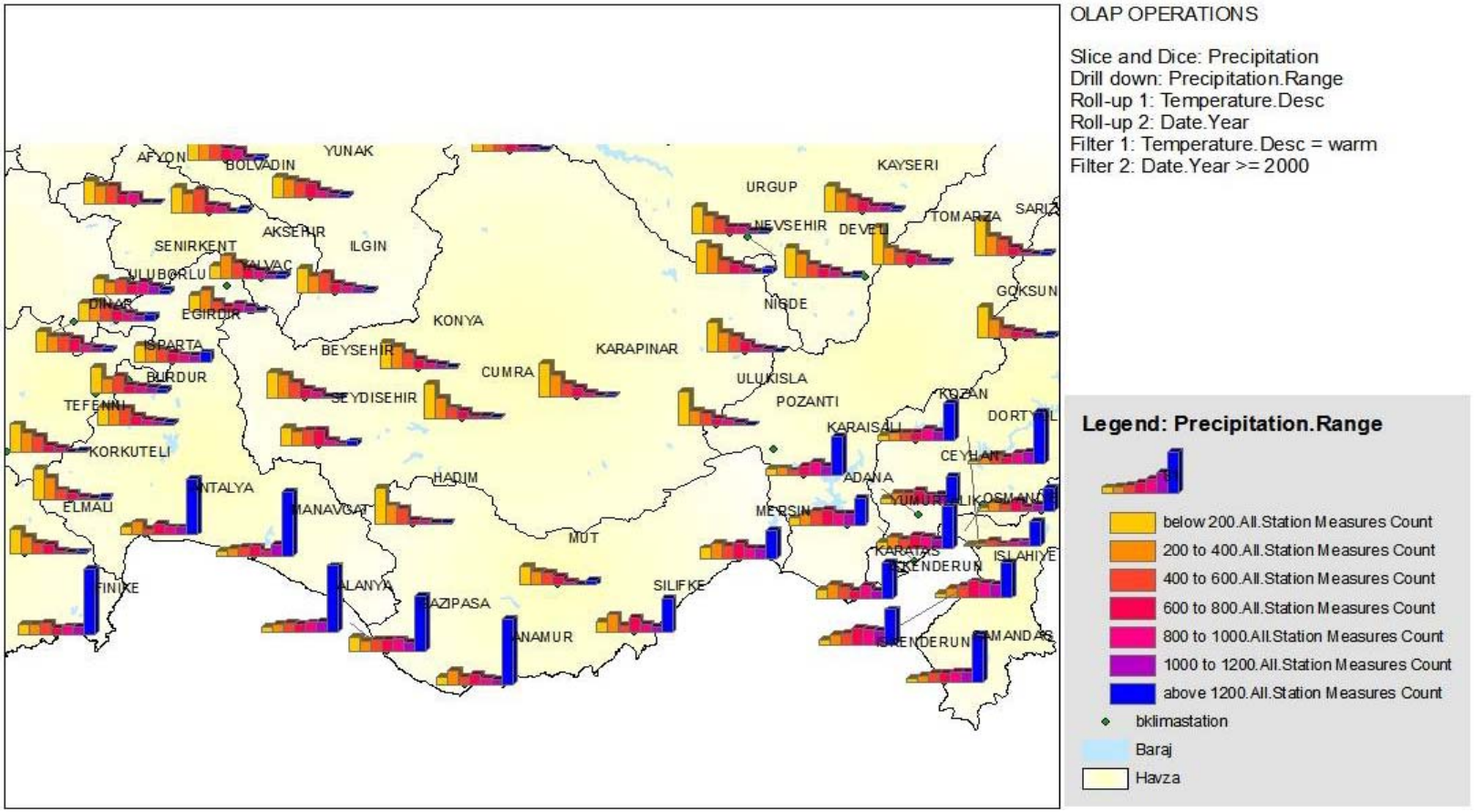


Figure 5.28 Geovisualization of Query 5

Query 6

What is the count of station measures for each station according to the categories of precipitation range values, which have hot and extremely hot temperature description values in the 4th quarter of years since 1970?

The result of Query 6 shows the number of station measure counts with respect to the distribution of their precipitation range values in the fourth quarters of the years since 1970, having temperature description values of hot and extremely hot. Geovisualization of this query helps to extract a significant weather pattern in most of the stations in Doğu Akdeniz, Seyhan, Ceyhan and Asi basins. In this combination, while Anamur, Mersin Silifke stations mostly have precipitation values below 200, Kabatas, İskenderun and Dörtyol stations mostly have precipitation values above 1200.

Query 7

What is the count of station measures for each station according to the categories of precipitation range values, which have hot and extremely hot temperature description values in the 2nd quarter of years since 1970?

The result of Query 7 shows the number of station measure counts with respect to the distribution of their precipitation range values in the second quarters of the years since 1970, having temperature description values of hot and extremely hot. In this combination, three important weather patterns are extracted. First type of pattern is captured in Meriç Erhene Basin. In this basin, stations mostly have below 200 and 200 – 400 mm of precipitation range values. The second type is captured in some stations located in Batı Akdeniz and Antalya Basins. In these stations, below 200 mm of precipitation range values are seen dominantly. The third type is captured in some stations located in the coastal parts of Doğu Akdeniz, Ceyhan, Seyhan Basins. In these stations, below 200 mm of precipitation range values are seen dominantly and

other range values are mostly seen in decreasing number of measurements from below 200 to above 1200.

Query 8

What is the count of station measures for each station according to the categories of precipitation range values, which have hot and extremely hot temperature description values in the 3rd quarter of years since 1970?

The result of Query 8 shows the number of station measure counts with respect to the distribution of their precipitation range values in the third quarters of the years since 1970, having temperature description values of hot and extremely hot. In this combination, three important weather patterns are extracted. First type of pattern is captured in Meriç Erhene and Marmara Basins. In these basins, below 200 mm of precipitation range values are seen dominantly and other range values are mostly seen in decreasing number of measurements from below 200 to above 1200. The second type is captured in some stations located in Doğu Karadeniz Basin and some parts of Black Sea Coast. In these stations, above 1200 mm of precipitation range values are seen dominantly.

The third type is captured in some stations located in Ceyhan, Seyhan and Asi Basins. In these stations, below 200 mm of precipitation range values are seen dominantly.

Geovisualization of Query 6: What is the count of station measures for each station
according to the categories of precipitation range values,
which have hot and extreme hot temperature description values in the 4th quarter of years since 1970?

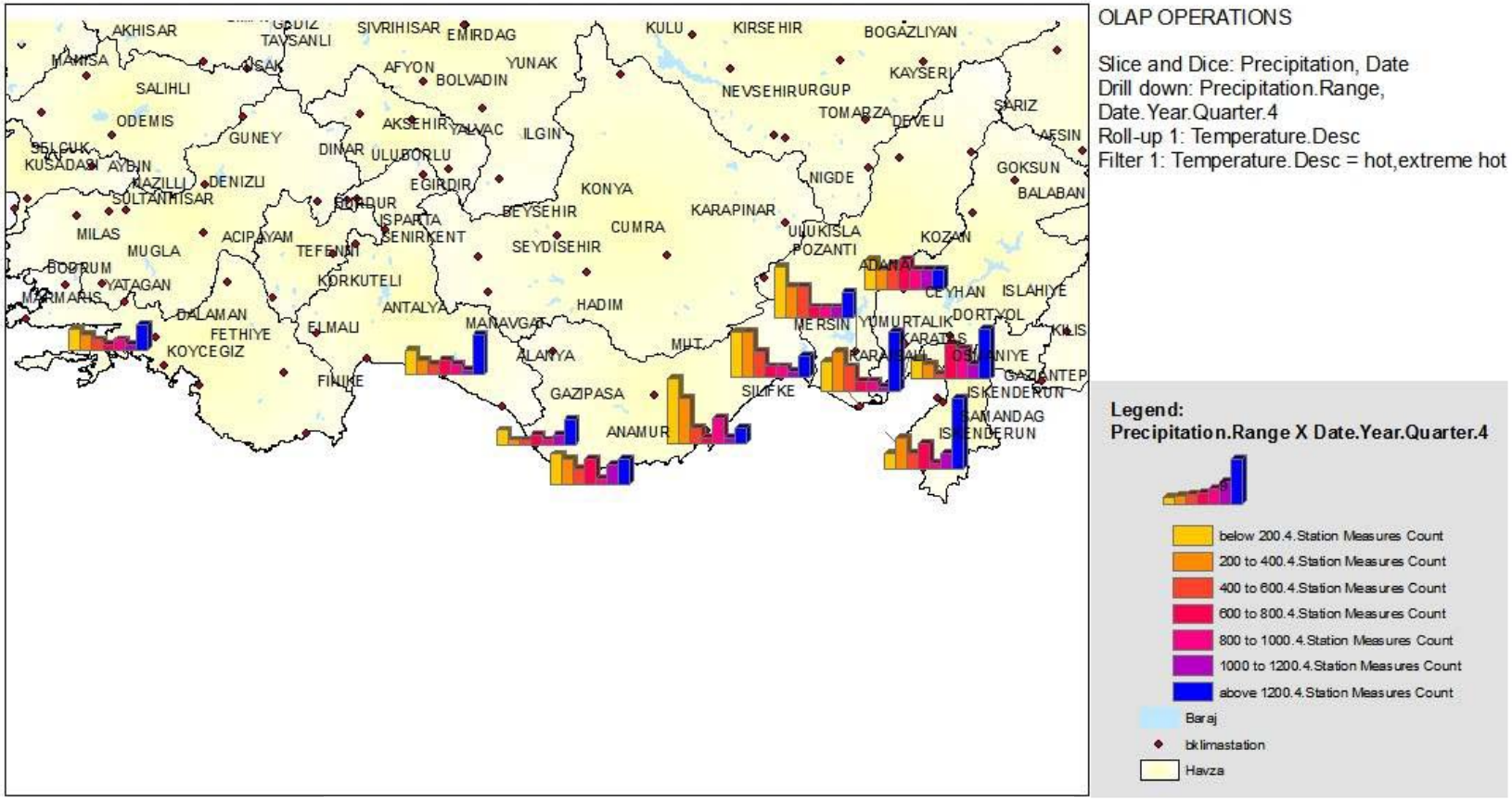
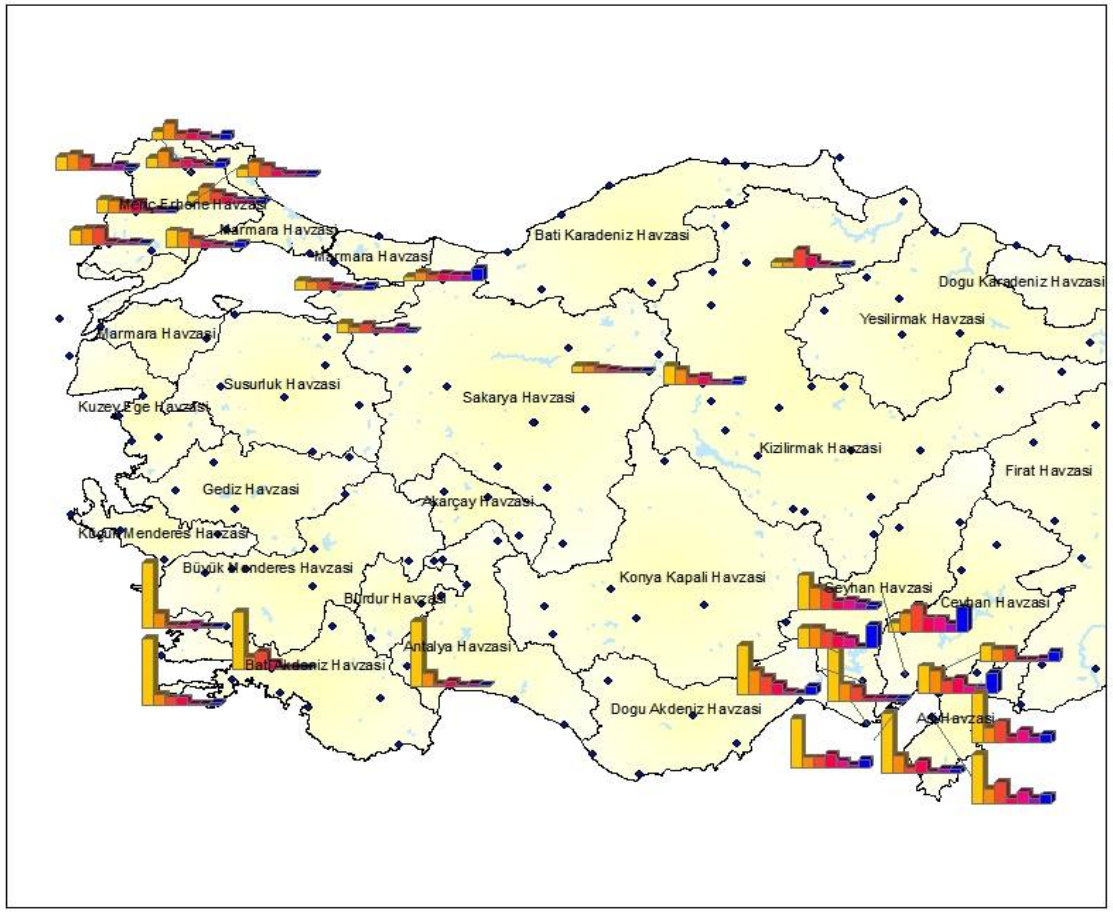


Figure 5.29 Geovisualization of Query 6

Geovisualization of Query 7: What is the count of station measures for each station
according to the categories of precipitation range values,
which have hot and extreme hot temperature description values in the 2nd quarter of years since 1970?



OLAP OPERATIONS

Slice and Dice: Precipitation, Date
Drill down: Precipitation.Range,
Date.Year.Quarter.2
Roll-up 1: Temperature.Desc
Filter 1: Temperature.Desc = hot,extreme hot

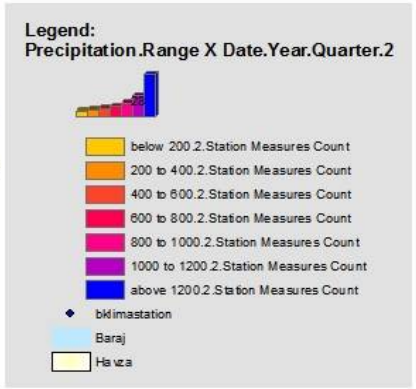


Figure 5.30 Geovisualization of Query 7

Geovisualization of Query 8: What is the count of station measures for each station
according to the categories of precipitation range values,
which have hot and extreme hot temperature description values in the 3rd quarter of years since 1970?

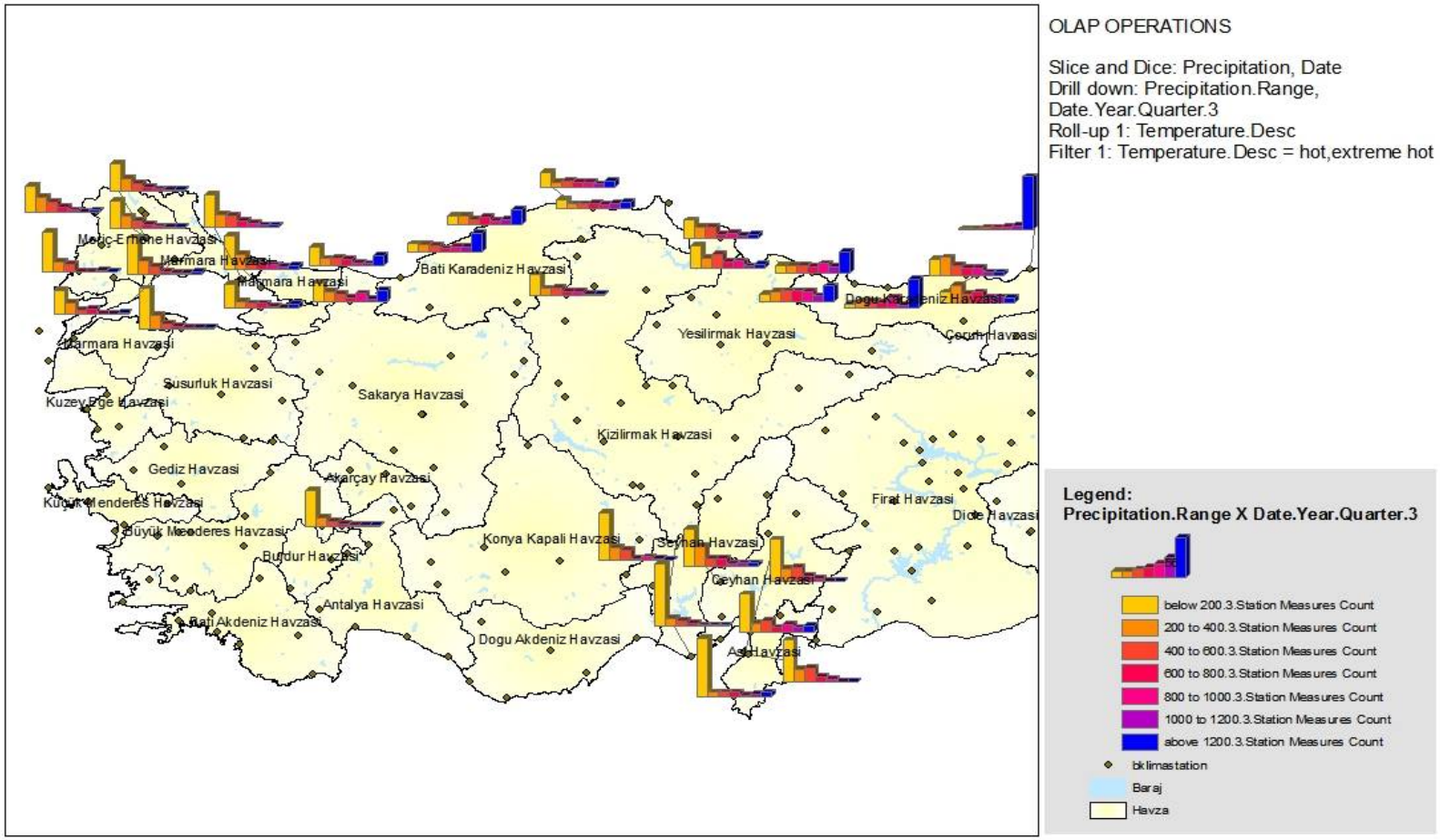


Figure 5.31 Geovisualization of Query 8

Query 9

What is the count of station measures for each station according to the categories of precipitation range values, which have cold and extremely cold temperature description values in the 1st quarter of years since 1970?

The result of Query 9 shows the number of station measure counts with respect to the distribution of their precipitation range values in the first quarters of the years since 1970, having temperature description values of hot and extremely hot. Geovisualization of this query helps to extract a significant weather pattern in the eastern part of Turkey. In this pattern, above 1200 mm of precipitation range values have been mostly seen in most of the stations in Fırat and Dicle Basins.

Query 10

How are each basin's normalized total station measure counts distributed according to the categories of precipitation range values which have hot and extremely hot temperature description values since 1970?

The result of Query 10 shows the normalized number of station measure counts of each basins with respect to the distribution of their precipitation range values since 1970, having temperature description values of hot and extremely hot. In this type of geovisualization, distribution of precipitation range values are illustrated using pie-charts. Station measure counts are normalized by dividing total aggregated counts in each basin to the number of stations in that basin. Thus, varying pie-chart sizes show the differences between basins and this results in allowing comparison of basins. Geovisualization of this query helps to extract three different types of weather patterns. The first type is seen along Black Sea Coast.

In this pattern, above 600 mm of precipitation range values have been mostly seen in most of the stations in Batı Karadeniz, Doğu Karadeniz Basins.The second type is seen in Marmara, Meriç-Erhene, Asi, Ceyhan and Seyhan Basins. In this pattern,

about % 50 of normalized station measure counts in each basin have below 200 mm of yearly precipitation range values and other range values are distributed approximately equally. The third type is seen all other basins in Turkey. In this pattern, below 200 mm of precipitation range values have been dominantly seen in most of the stations in Turkey.

Query 11

What is the relationship between hotness as a temperature description value and height of stations where these measures have been collected?

In order to visualize this query, temperature description values of stations are interpolated by using inverse distance weighted method and height values of stations are categorized by unique values of the height field. First, as a result of the interpolation, a raster map is created as in Figure 5.34. In this raster map, colors ranging from yellow to red show the station measure counts and they are classified into nine classes ranging from 0 to 225. Reddish colors illustrate the zones that have hotness cases higher in number. Second, height values of stations are categorized into 5 classes: 0 to 100 meters, 100 to 500 meters, 500 to 1000 meters, 1000 to 2000 meters and 2000 to 4000 meters. In this categorization, colors of points ranging from white to brown illustrate the height of stations beginning from the lowest to the highest range. The result of this query shows that the zones in which hot measurements are seen more than the other zones, have heights mostly below 100 meters.

Query 12

What is the relationship between wetness as a precipitation description value and height of stations where these measures have been collected?

Similar to Query 11, precipitation description values of stations are interpolated by using inverse distance weighted method and categorized height values of stations are

used as in Query 11. First, as a result of the interpolation, a raster map is created as in Figure 5.35. In this raster map, colors ranging from yellow to blue show the station measure counts and they are classified into six classes ranging from 0 to 379. Blue and green colors illustrate the zones that have wetness cases higher in number. The result of this query shows that the zones in which wet measurements are seen more than the other zones, have heights mostly below 500 meters.

After evaluating the queries presented in this section, it can be easily seen that the model proposed in this study provides an easy way to evaluate complex and multidimensional queries. No query language is needed to perform these queries. The proposed model provides Multidimensional Expressions (MDX) interface for the user to perform complex queries visually. On the other hand, to obtain the result of these queries in a traditional spatial database requires performing complex SQL queries and too much processing times. Moreover, in a traditional spatial database it's not possible to navigate accross the different dimensions at different levels of detail.

This study differs from other studies in spatial data warehousing in two aspects: First, Stefanovic et al (2000) studied weather pattern analysis in British Columbia of Canada as a motivating example in order to demonstrate implementation stages of spatial data warehousing. Stefanovic et al (2000) focus on the efficient implementation of spatial data warehousing which is based on handling the spatial measures. Different from their study, this study focuses on spatial data exploration, analysis and geovisualization of spatial data by adding spatial dimensionality to data warehousing rather than dealing with spatial measures. Second, Işık (2005) harmonized spatial data cubes and fuzzy data cubes to enable better analysis and understanding of spatial data by using fuzzy set theory. In her research, weather pattern searching is used as a case study in order to demonstrate mining association rules from fuzzy spatial data cubes constructed by using hypothetic data. Different from her study, this study deals with the implementation stages of spatial data warehousing by using real weather data and applies spatial data exploration, analysis and geovisualization of these data.

Geovisualization of Query 9: What is the count of station measures for each station
according to the categories of precipitation range values,
which have cold and extreme cold temperature description values in the 1st quarter of years since 1970?

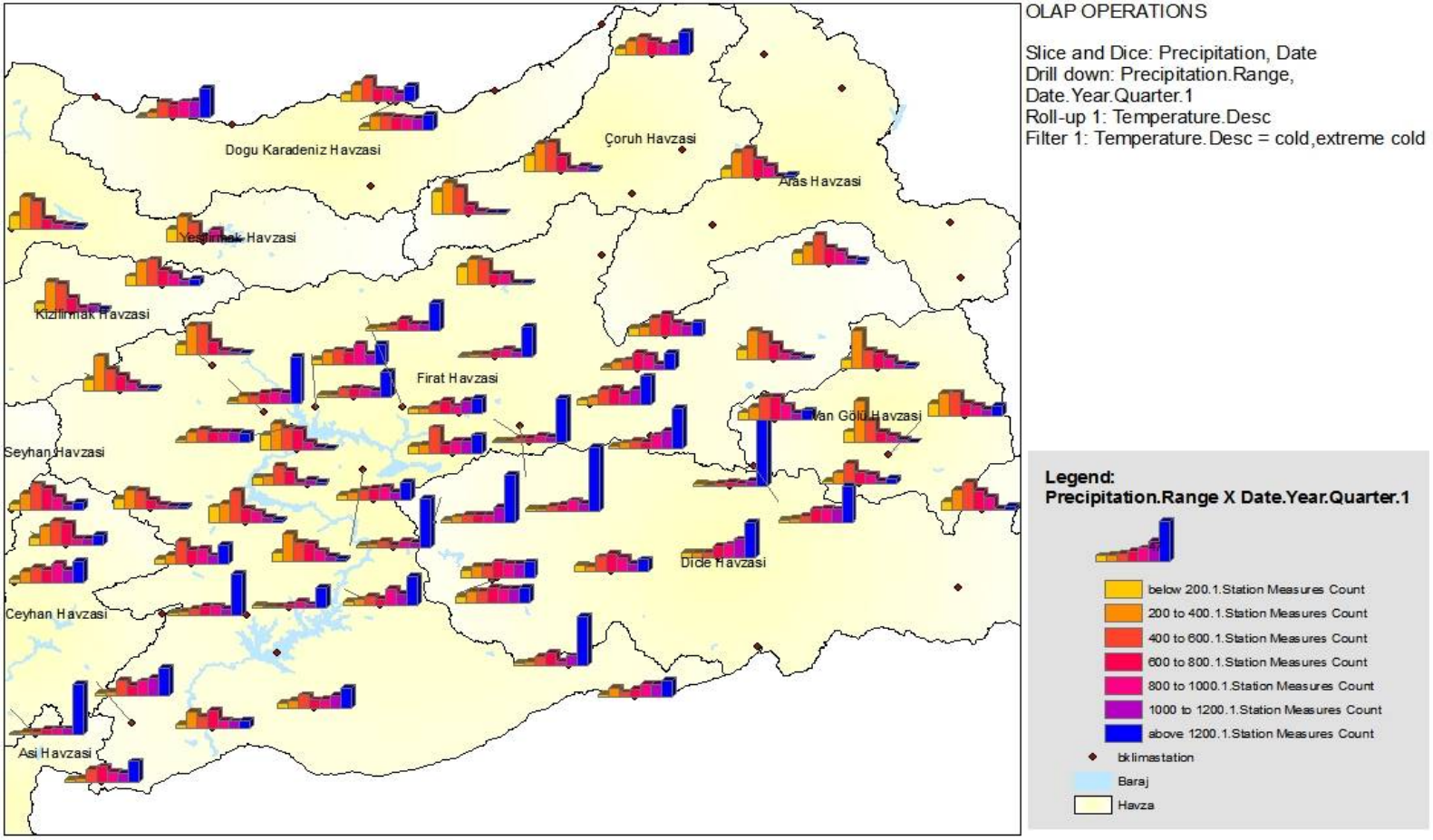


Figure 5.32 Geovisualization of Query 9

Geovisualization of Query 10: How are each basin's normalized total station measure counts distributed according to the categories of precipitation range values
which have hot and extreme hot temperature description values since 1970?

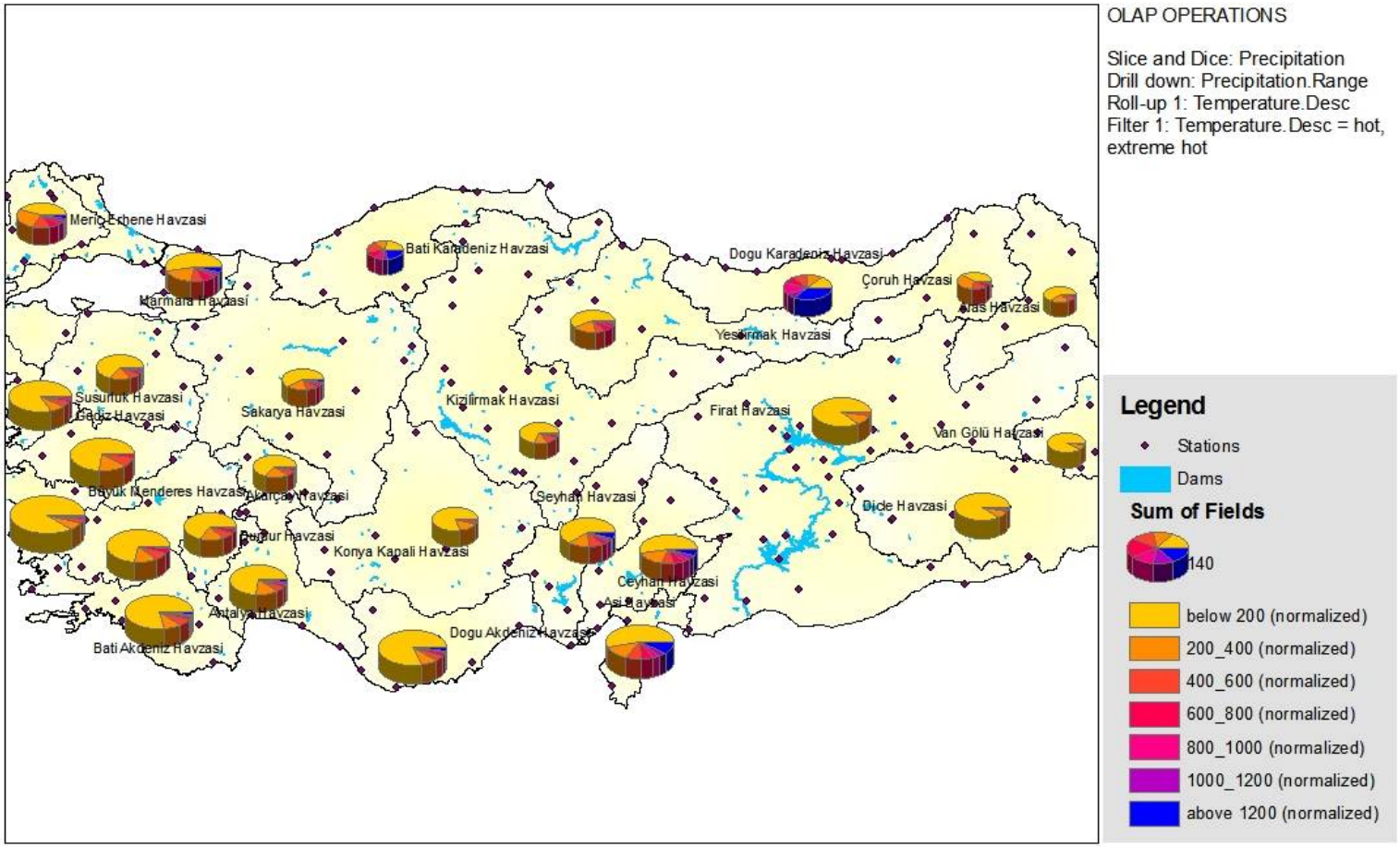


Figure 5.33 Geovisualization of Query 10

Geovisualization of Query 11: What is the relationship between hotness as a temperature description value and height of stations where these measures have been collected?

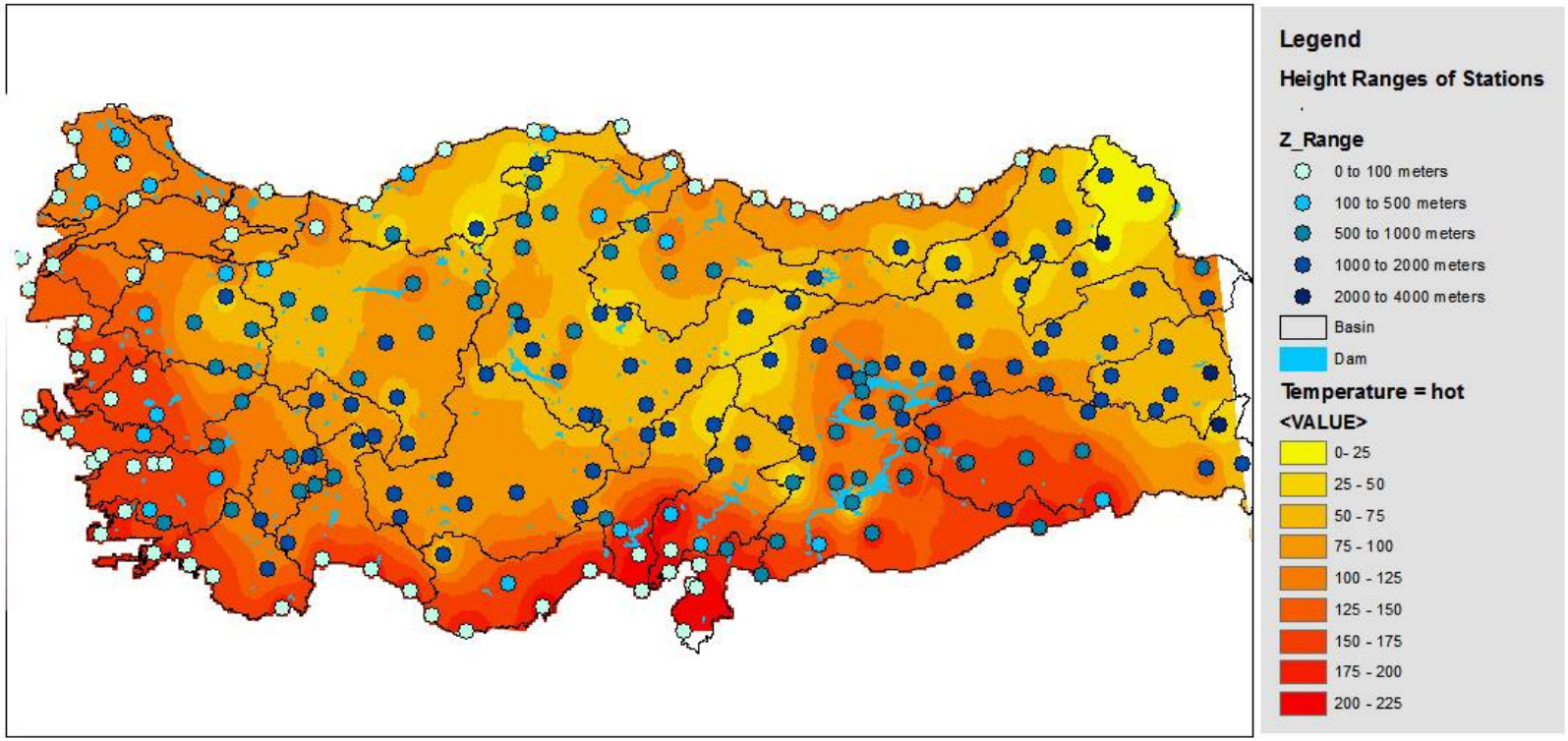


Figure 5.34 Geovisualization of Query 11

Geovisualization of Query 12: What is the relationship between wetness as a precipitation description value and height of stations where these measures have been collected?

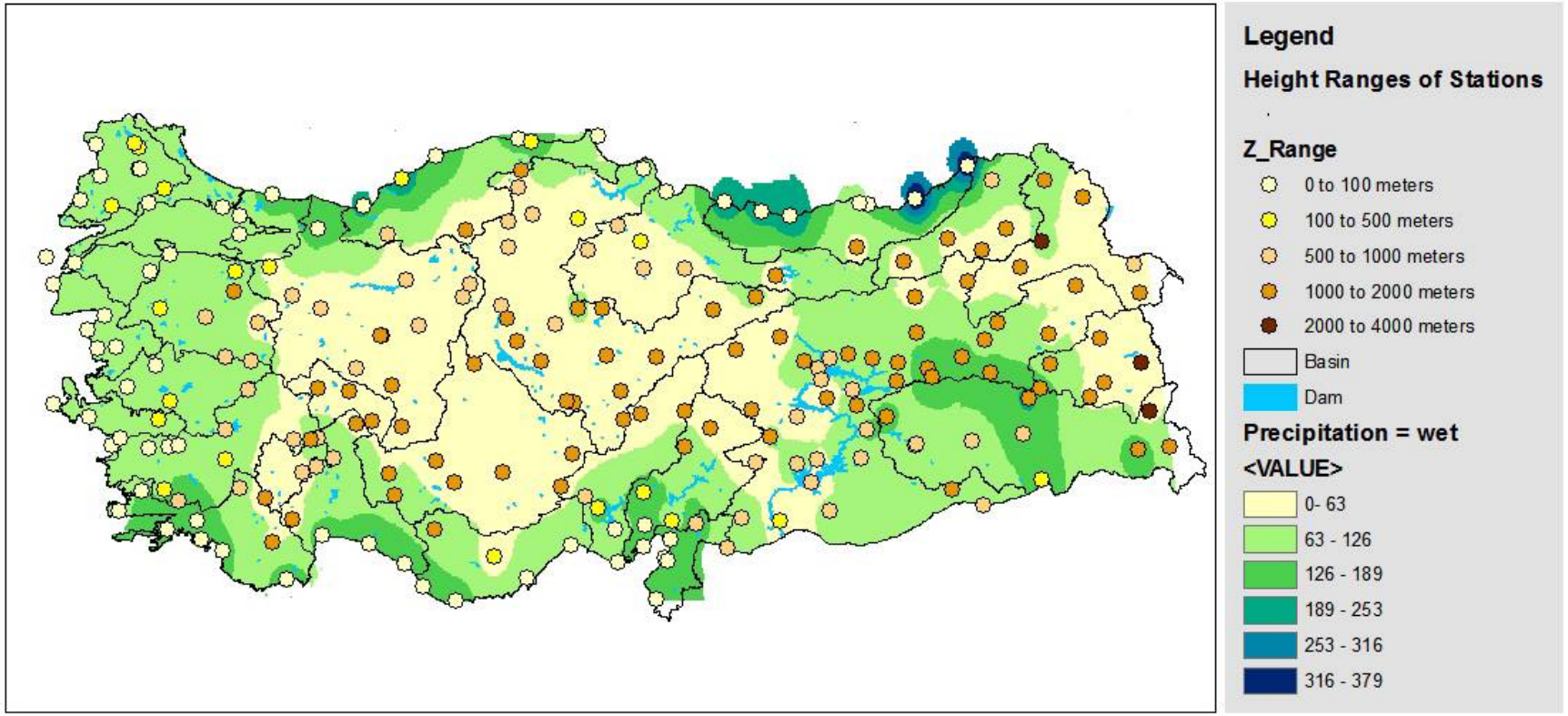


Figure 5.34 Geovisualization of Query 12

CHAPTER 6

CONCLUSION

Integration of data warehouses with GISs has become a major trend in constructing SDSSs. Such an integration provides combination of the strengths of OLAP and GISs. In this context, this research proposes a spatially enabled data warehouse model to provide solutions to two basic group of challenges in the lifecycle of as a SDSS. The first group of challenges is related with how to design and construct an efficient warehouse model and this type of challenges usually concerns the designer of a SDSS. The second group of challenges is related with providing efficient and easy information retrieval from that model and this type of challenges usually concerns the user of a SDSS.

Designing and building phases of a warehouse model, poses some challenges to the designer of a SDSS. First, adapting needs and requirements of the subject of the warehouse is the most critical challenge for the warehouse designer. All requirements and needs must be clearly defined to avoid rework and reconstruction. Second, the integration of diverse data to the warehouse poses a great challenge to the designer of the warehouse. Therefore, data collection and integration steps must be well designed. Third, the integration of spatial and non-spatial data poses a challenge in meeting the required standards of the goal of a SDSS.

The challenges that a user of a SDSS faces in retrieving information from a decision support database can be categorized into five. First, most systems require expert users who involve both in design and development process of the system to understand and analyze the contents of the data. However, the user must be capable of making decisions without knowing the structure of the warehouse. Second, data in a DS database consist of multiple heterogenous data sources. Even they are combined into a common database structure, the user should know the each

particular data source in order to extract the information of interest. Third, when the need for aggregating data in multiple levels is considered, quick responses cannot be provided for decision-making stage. Fourth, interpretation of data is not included in the data sources. Thus, the user should also capable of interpreting any kind of information extracted from the DS database. Fifth, spatio-temporal multidimensional analysis cannot be efficiently provided by transactional systems since they require complex query building and data navigation operations.

In this context, the proposed model in this study presents solutions to two group of challenges presented above and contributes to the literature with the solutions it proposes.

For the first group of challenges, the spatially enabled data warehouse model which is proposed in this study provides basis for:

- the definition of the requirements of DS tasks,
- management of data collection, transformation and integration phases,
- a unified view with its cleaned, transformed, integrated and analysis oriented data
- the design of the warehouse with defining dimensional modeling, physical, technical architecture design and cube processing phases in detail,
- the integration of spatial data with the sub-cubes extracted from the warehouse,
- the application of knowledge discovery tools and techniques,
- spatial data exploration and analysis of spatio-temporal data,

For the second group of challenges, this study provides solutions which are summarized below:

The model provides fast and easy access to the decision makers because of two reasons. First, no query language is needed to conduct analysis or understand the

underlying structure. It provides a bridge between the user and the data source and the user interacts dimensional structure through an MDX interface. Therefore, the users directly interacts with data and results. Second, data are aggregated and fast response times are possible for complex queries.

Secondly, the model allows users to explore and navigate across the different dimensions of the weather cube at different levels of detail. Therefore, users have access to possible combinations and views of the weather data and this facilitates the emergence of new hyphotheses and spatial knowledge discovery.

Thirdly, the model provides visual display of the results quickly, and convey information about the case that might require hours of study to extract the same information from mathematical and statistical analysis results.

Fourthly, the model adds spatial context into data warehouses which need integration of data from different sources facilitates spatial data exploration and assists decision-making tasks about spatial data. It allows rapid and easy navigation within the spatial data warehouse which offers many levels of information granularity, and many display modes of maps, tables and diagrams. As an important functionality of spatial data exploration, geovisualization of spatio-temporal data is provided through integrating cartographic displays with non-cartographic displays such as tabular displays, statistical charts, diagrams and histograms. These integrated displays help end-users to make better decisions. Moreover, the model provides visualization of only significant aggregates for certain types of queries. Therefore, data that may be irrelevant in a particular case are excluded by choosing less detailed level of a hierarchy.

Fifthly, spatially related weather patterns are identified to support hyphothesis generation regarding the spatial relationship between the spatial and non-spatial dynamics of weather data. As a result, the main functionalities and advantages of a spatial data warehouse structure are evaluated.

The model proposed in this study can be applied to many disciplines which involve decision-making in any part of their operations. Therefore, there is a wide range of end-users in several sectors, state and educational institutions. Some examples are hospital organizations, universities, corporations, Ministry of Agricultural and Rural Affairs, Ministry of Health, Municipalities, State Hydraulic Works, Ministry of Culture and Tourism.

This study proposed solutions to some of the challenges in implementing a spatial data warehouse model. However, there are also other challenges which must be handled by performing future research. First of all, this study provides an integrated visualization environment for geovisualization of cartographic and tabular displays. However, in the model proposed in this study, users must extract a sub-cube, and then join that sub-cube with spatial data in order to display a combination of dimensions and measures. Therefore, a flexible display management can be developed to allow simultaneous displays of spatiotemporal data as in Rivest et al (2001)'s study. Secondly, only spatial dimensions are used in the model provided in this study. In future studies, efficient implementation techniques based on handling of spatial measures can also be integrated in these studies. Thirdly, this study provides extracting spatiotemporal patterns by visualization of cartographic and tabular data in an integrated environment. Further studies can be done for extracting these patterns by applying efficient spatial data knowledge discovery methods.

REFERENCES

Agarwal S., Agrawal R., Deshpande P. M., On the Computation of Multidimensional Aggregates, Proc. Of the 22nd VLDB Conference, 1996.

Başarsoft Haberler, http://www.basarsoft.com.tr/basar/tr/yenihaber/haber_detay.asp?id=104, last access date: 6.06.2008

Bedard Y., Merret T., Han J., "Fundamentals of spatial data warehousing for geographic knowledge discovery", Geographic Data Mining and Knowledge Discovery, CRC Press, 2001.

Bédard Y., Larrivée S., Proulx M.J., Létourneau F., Caron P.Y., Étude de l'état actuel et des besoins de R&D relativement aux architectures et technologies des data warehouses appliquées aux données spatialesResearch Report for National Defence Canada Centre for Research in Geomatics, Laval University, 1997.

Chaudhuri S., Dayal U., "An Overview of Data Warehousing and OLAP Technology", ACM SIGMOD Record, voi 26, pp. 65-74, 1997

Chen M-S., Han J., Yu P.S., "Data Mining An Overview from Database Perspective", Ieee Trans. On Knowledge And Data Engineering, 1996

Çalargün, S. Ü., Fuzzy Association Rule Mining From Spatio-temporal Data: An Analysis of Meteorological Data in Turkey, MSc thesis, Computer Engineering Dept., Middle East Technical University, Ankara, 2008.

Elmasri R., Navathe S.B., Fundamentals of Database Systems, Pearson Education, 2004.

Gray J., Bosworth A., Layman A., Pirahesh H., Data Cube: A Relational Operator Generalizing Group-by, Cross-tab, and Roll-up, Proc. of the12th Int. Conf. on Data Engineering, pp. 152-159, 1996.

Haby J., What is a Weather Pattern?, http://www.theweatherprediction.com/habyhints2/450/, last access date: 27.02.2008.

Han J., Kamber M., Data Mining: Concepts and Techniques, Academic Press, 2001.

Han J., Koperski K., Stefanovic N., Geominer: A System Prototype for Spatial Data Mining, Proceedings of 1997 ACM-SIGMOD, International Conference on Management of Data, pp.553-556, 1997.

Harinarayan V., Rajaraman A., Ullman J. D., Implementing Data Cubes Efficiently Proc. 1996, ACM-SIGMOD, Int'l Conf. Management of Data, pp. 205-216, 1996.

Işık N., Fuzzy Spatial Data Cube Construction and Its Use in Association Rule Mining. MSc thesis, Computer Engineering Dept., Middle East Technical University, Ankara, 2005.

Inmon W.H., Building the Data Warehouse, John Wiley and Sons, 2002.

Kheops Technologies, Innovative technology to support intuitive and interactive exploration and analysis of spatio-temporal multidimensional data, Jmap Spatial OLAP, 2005.

Kimball R., The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, John Wiley and Sons, 2002

Kimball R, Caserta J., The Data Warehouse ETL Toolkit, Wiley and Sons, 2004.

Kouba Z., Matousek K., Miksovsky P., “Novel Knowledge Discovery Tools in Industrial Applications”, Gerstner Labarotory for Intelligent Decision Making and Control, 2002.

Marchand P., Brisebois A., Bedard Y., Edwards G., Implementation and evaluation of a hypercube-based method for spatio-temporal exploration and analysis, Science Direct, ISPRS Journal of Photogrammetry and Remote Sensing 59, 6 – 20, 2004.

Matuschak B. J., "GIS as a Complement to OLTP, Data Warehousing, Data Mining, and OLAP", The Electronic Atlas Newsletter, Vol. 10, No. 10, October 1999.

Mennis J., J. W. Liu, "Mining Association Rules in Spatio-Temporal Data: An Analysis of Urban Socioeconomic and Land Cover Change", Transactions in GIS, 2005, 9(1): pp. 5 – 17.

Microsoft SQL Server 2005, Books Online, 2007.

Microsoft SQL Server 2005 Analysis Services Step by Step, R. Jacobson, S. Misner, Hitachi Consulting, 2005.

Microsoft SQL Server 2005 Analysis Services Performance Guide, SQL Server Technical Article, E. Vitt, 2007.

Miller H. J., Han J., Geographic Data Mining and Knowledge Discovery, CRC Press, 2003.

Pestana G., Mira da Silva M., "Multidimensional Modeling based on Spatial, Temporal and Spatio-Temporal Stereotypes", July 2005.

Power D. J., Decision Support Systems Glossary, http://www.dssresources.com/glossary, last access date: 13.03.2008

Rivest S., Bedard Y., Marchand P., Towards better support for spatial decision-making: defining the characteristics of Spatial On-Line Analytical Processing (SOLAP). Geomatica 55 (4), 539– 555, 2001.

Scotch M., Parmanto B., Monaco V., Usability Evaluation of the Spatial OLAP Visualization and Analysis Tool (SOVAT), Vol. 2, Issue 2, pp. 76-95, February 2007.

Simon H. A, The new science of management decision, NewYork: Harper & Row, 1960.

Stefanovic N., Design and Implementation of On-Line Analytical Processing (OLAP) of Spatial Data. MSc thesis, Computing Sci. Dept., Simon Fraser University, Vancouver. 118 p, 1997.

Stefanovic N., Han J., Koperski K., "Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes", IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 6, 2000.

Thomsen E., OLAP Solutions: Building Multidimensional Information Systems, Wiley Computer Pub, 1997.

Yuan M., and McIntosh J., A Typology of Spatio-temporal Information Queries. *Mining Spatio-temporal Information Systems*. K. Shaw, R. Ladner, and M. Abdelguerfi (eds.). Kuwer Academic Publishers. PP. 63-82, 2002.

Zhang D., Tsotras V., “Improving Min/Max aggregation over Spatial Objects”, In Proc. GIS, 2001.

APPENDIX A

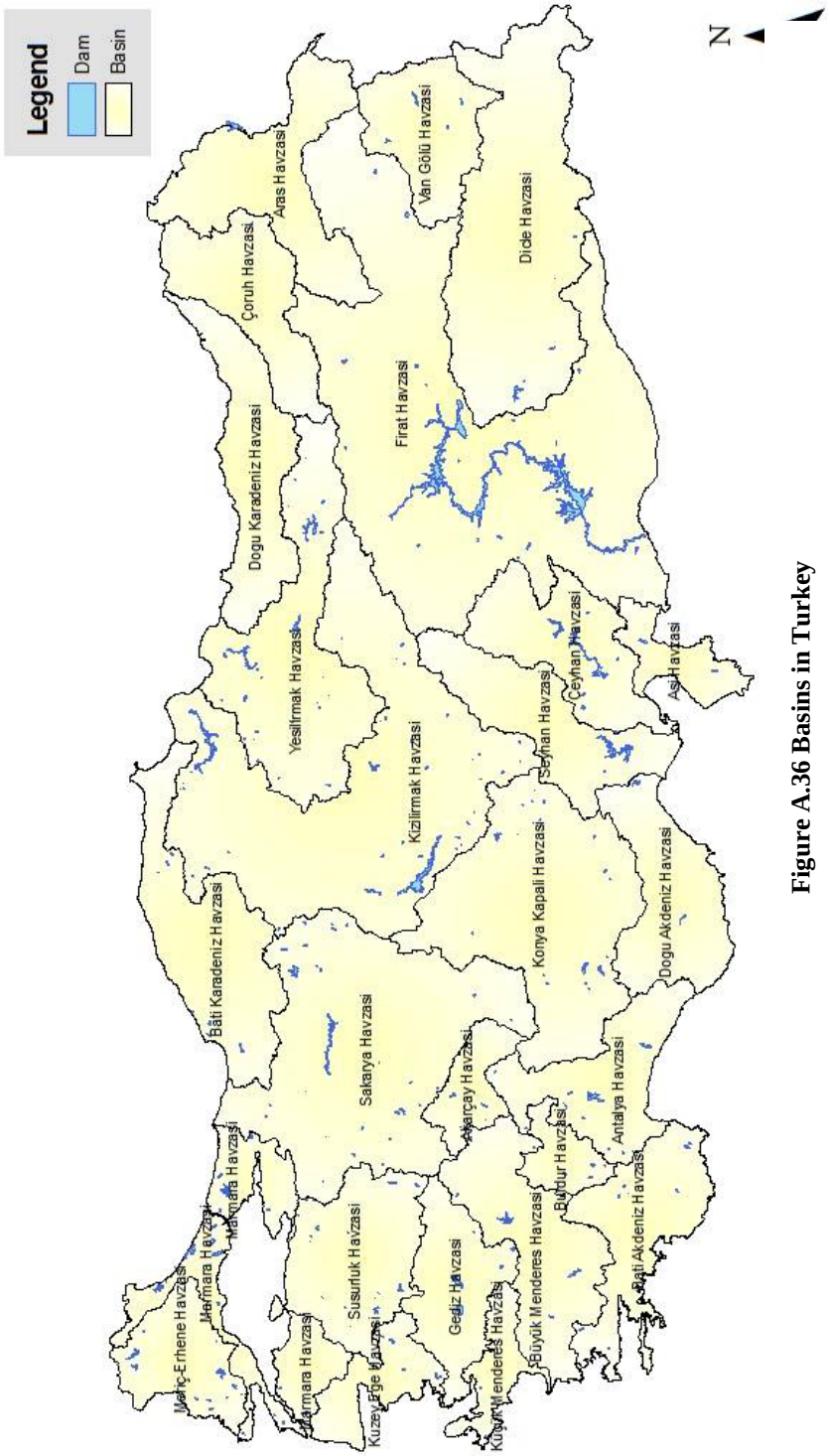


Figure A.36 Basins in Turkey

APPENDIX B

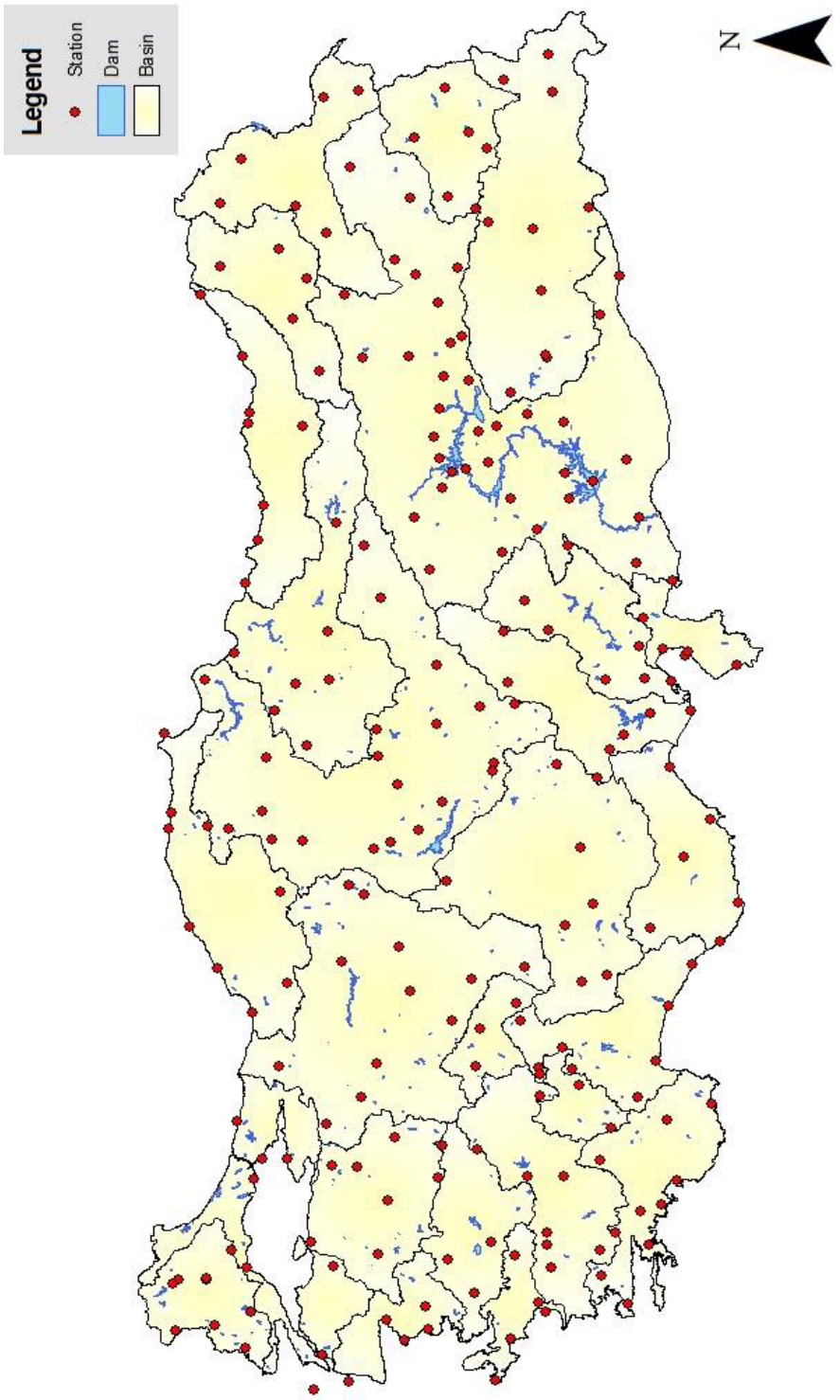


Figure A.37 Meteorological Stations In Turkey

APPENDIX C

Table C.16 Monthly Total Precipitation Sample Raw Data

Station ID	Year	Month	Total Precipitation
……..	…….	…..	…….
9030	1979	1	150
9030	1979	2	43.3
9030	1979	3	29.2
9030	1979	4	17.5
9030	1979	5	36.7
9030	1979	6	1.4
9030	1979	10	21.9
9030	1979	11	143.7
9030	1979	12	83.6
17020	1970	1	109.7
17020	1970	2	138.2
17020	1970	3	79.5
17020	1970	4	64.7
17020	1970	5	113.8
17020	1970	6	42.7
17020	1970	7	32.8
17020	1970	8	226.1
17020	1970	9	55.9
17020	1970	10	140.5
17020	1970	11	78.2
17020	1970	12	192.3
17020	1971	1	39.5
17020	1971	2	147.2
17020	1971	3	90.2
17020	1971	4	51.2
17020	1971	5	70.5
17020	1971	6	37.7
17020	1971	7	23.8
17020	1971	8	23
17020	1971	9	69.4
17020	1971	10	174.4
17020	1971	11	93.9
17020	1971	12	313.5
…..	…..	..	....

APPENDIX D

Table D.17 Monthly Average Temperature Sample Raw Data

Station ID	Year	Month	Average Temperature
……	….	….	……
17056	2003	1	7.1
17056	2003	2	1.1
17056	2003	3	4.5
17056	2003	4	8.8
17056	2003	5	17.9
17056	2003	6	23
17056	2003	7	24.8
17056	2003	8	25.2
17056	2003	9	19.3
17056	2003	10	16
17056	2003	11	10.6
17056	2003	12	6.4
17056	2004	1	4
17056	2004	2	5.7
17056	2004	3	8.3
17056	2004	4	12
17056	2004	5	16.3
17056	2004	6	21
17059	1970	1	6.9
17059	1970	2	8
17059	1970	3	8.7
17059	1970	4	13.1
17059	1970	5	15
17059	1970	6	19.4
17059	1970	7	23.6
17059	1970	8	23.2
17059	1970	9	19.5
17059	1970	10	14.4
17059	1970	11	12
17059	1970	12	7.9
17059	1971	1	8.3
17059	1971	2	5.8
17059	1971	3	7.2
17059	1971	4	10.4
17059	1971	5	16.3
17059	1971	6	20.7
17059	1971	7	21.7
17059	1971	8	23
17059	1971	9	19.1
17059	1971	10	13.4
17059	1971	11	11.5
17059	1971	12	7.2
….	…….	..	…..

APPENDIX E

Table E.18 Monthly Maximum Temperature Sample Raw Data

Station ID	Year	Month	Maximum Temperature
….	….	..	…..
17244	2003	1	17.6
17244	2003	2	13.6
17244	2003	3	14.2
17244	2003	4	26.3
17244	2003	5	29.6
17244	2003	6	32.3
17244	2003	7	37
17244	2003	8	36.3
17244	2003	9	36.1
17244	2003	10	31
17244	2003	11	22.6
17244	2003	12	13.4
17244	2004	1	12.6
17244	2004	2	20.5
17244	2004	3	25
17244	2004	4	29.4
17244	2004	5	28
17244	2004	6	32.2
17246	1970	1	15.1
17246	1970	2	18
17246	1970	3	22.6
17246	1970	4	30.5
17246	1970	5	30.1
17246	1970	6	33.4
17246	1970	7	36
17246	1970	8	37.4
17246	1970	9	30.4
17246	1970	10	25.2
17246	1970	11	21.9
17246	1970	12	13
17246	1971	1	21.2
17246	1971	2	17
17246	1971	3	19.5
17246	1971	4	24.8
17246	1971	5	28.4
17246	1971	6	33.7
17246	1971	7	36.5
17246	1971	8	37
17246	1971	9	34.2
17246	1971	10	27
17246	1971	11	22
17246	1971	12	16.4
…….	….	..	….

APPENDIX F

Table F.19 Monthly Minimum Temperature Sample Raw Data

Station ID	Year	Month	Minimum Temperature
……	….	..	….
17842	1996	1	-15
17842	1996	2	-15.2
17842	1996	3	-5.8
17842	1996	4	-4.9
17842	1996	5	3.8
17842	1996	6	4.6
17842	1996	7	10.3
17842	1996	8	9.4
17842	1996	9	5
17842	1996	10	-2
17842	1996	11	-6.5
17842	1996	12	-4
17842	1997	1	-14.1
17842	1997	2	-24.3
17842	1997	3	-13.9
17842	1997	4	-10
17842	1997	5	0
17842	1997	6	3.8
17842	1997	7	8.5
17842	1997	8	7.9
17842	1997	9	0
17842	1997	10	-2.5
17842	1997	11	-5.1
17842	1997	12	-12.2
17842	1998	1	-12
17842	1998	2	-14.6
17842	1998	3	-10.4
17842	1998	4	-2.5
17842	1998	5	4
17842	1998	6	6.8
17842	1998	7	8.9
17842	1998	8	9.2
17842	1998	9	3.6
17842	1998	10	-0.5
17842	1998	11	-4.5
17842	1998	12	-8.1
……	….	..	…..