CASE STUDY

1 Data Warehousing in the Tamil Nadu Government

C1.1 GISTNIC DATA WAREHOUSE

General Information Service Terminal of National Informatics Centre (GISTNIC) Data Warehouse* is an initiative taken by National Informatics Centre (NIC) to provide a *comprehensive information database* by the government on national issues ranging across diverse subjects like food and agriculture to trends in the economy and latest updates on science and technology. This information base was collated to fulfil the information needs of bureaucrats, politicians, economists and, most important of all, the citizens.

The GISTNIC data warehouse is a *Web-enabled* SAS software solution. The data warehouse aims at providing online information to key decision-makers in the government sector enabling them to make better strategic decisions with regard to administrative policies and investments. The Government of Tamil Nadu is the first one to perceive the need and importance of converting *data into valuable information for better decision-making*. The GISTNIC Web site currently has an online data warehouse which includes data marts on village amenities, rainfall, agricultural census data, essential commodity prices, malaria statistics, Indian economy statics and school health (see Fig. C1.1). This data warehouse was developed during 1998–99.

Information technology pioneers from several public and private sectors appreciate the GISTNIC data warehouse as under:

> The GISTNIC data warehouse is a web-enabled solution, which intends to provide easy access with point and click interfaces to key decision makers across various levels in the government. The effectiveness is evident in the form of simultaneous availability of information and the speed at which it is delivered.
>
> *D. Prakash, Secretary, Information Technology, Government of Tamil Nadu*

> The GISTNIC data warehouse endorses the beginning of a strategic business alliance between SAS Institute and the government sectors. SAS Institute is committed the world over to government organizations and intends to

*This GISTNIC data warehouse for Tamil Nadu was implemented during 1998.

pursue this in India as well. We shall endeavor to extend full support at all times.

Rohini Midha, Managing Director, SAS Institute India Pvt. Ltd.

We look at this data warehouse as a means of preparing the government to face the challenges of the next millennium. The Tamil Nadu Government will be India's first State Government which will ride on the technology wave to streamline processes and drive better decision making on the basis of communication of information in whatever form it may take, unconstrained by distance, time and volume.

C.S.R. Prabhu, Senior Technical Director and State Informatics Officer, National Informatics Centre, Tamil Nadu

The GISTNIC data warehouse for Tamil Nadu indicates the commitment of SAS Institute in working together with the government to enable them to move towards their being IT-savvy.

Gourish Hosangady, National Sales and Technical Manager, SAS Institute India Pvt. Ltd.

C1.2 OBJECTIVES OF THE WEB-ENABLED DATA WAREHOUSE

- To provide powerful decision-making tools in the hands of the end-users in order to facilitate prompt decision-making
- To reduce the amount of resources—time and manpower—spent on managing the volumes and variety of database handled by NIC.

C1.3 DATA ANALYSIS

Data marts of various sectors and their applications are given in Table C1.1 as follows:

Table C1.1 Data Marts of various sectors and their applications

Data mart	Applications
Village amenities. This data mart contains the 1991 census data of village amenities in all the villages in Tamil Nadu. It contains information on availability for amenities like education, health, drinking water, transportation, communication and irrigation (see Figs. C1.2–C1.5).	• Village amenities analysis • Irrigation analysis • Top/bottom analysis • Range analysis—amenities • More amenities—analysis
Rainfall statistics. This data mart has information on daily levels of rainfall across various weather stations in Tamil Nadu. This will help them to plan the water supply to various districts in Tamil Nadu and using various models to forecast rainfall levels (see Fig. C1.6).	• Time-based rainfall analysis • Geography/time-based rainfall analysis

(contd.)

Table C1.1 Data Marts of various sectors and their applications *(contd.)*

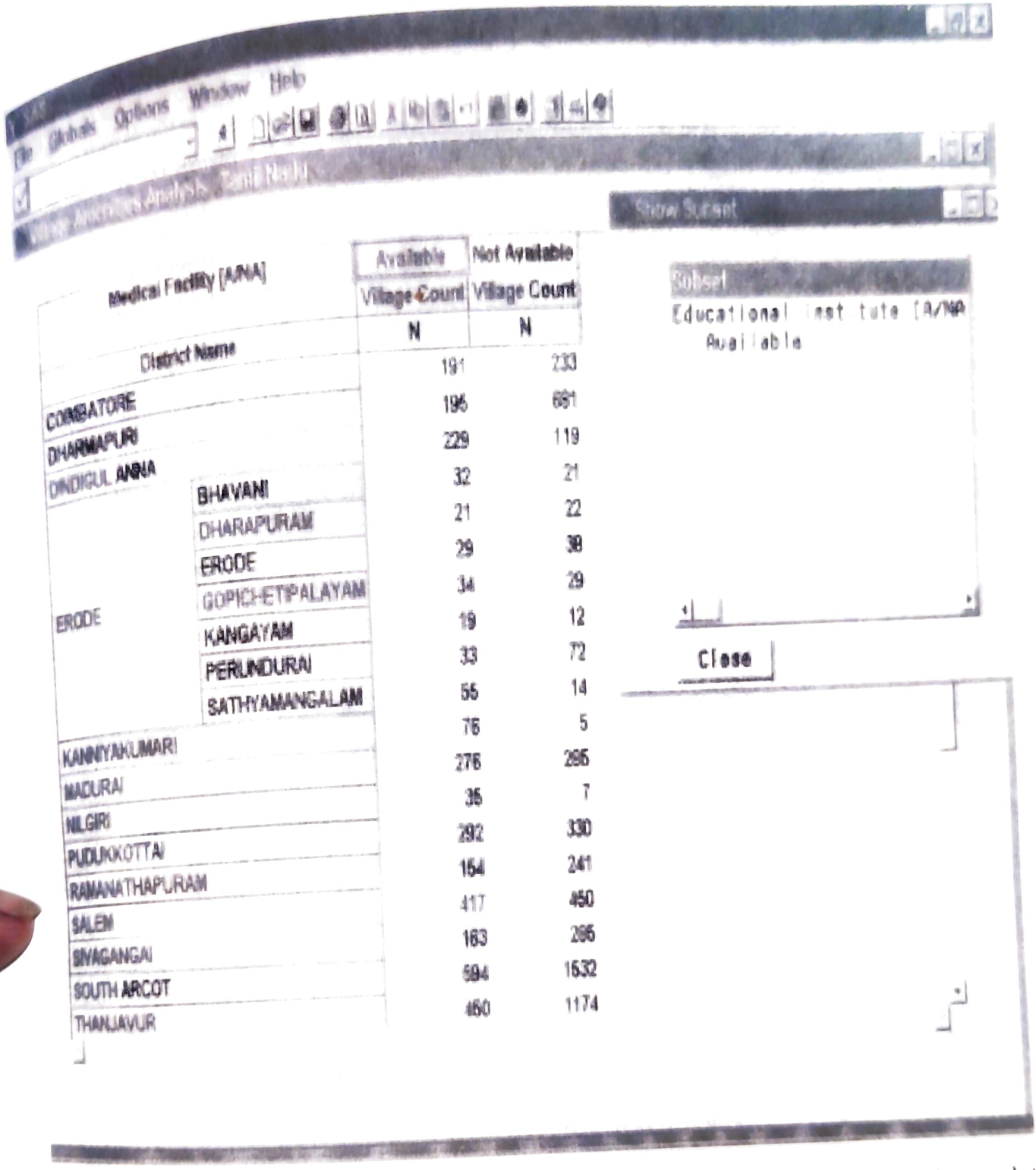
Data mart	Applications
Agricultural census. This data mart has information on land-holding patterns across the villages in Tamil Nadu. It can be used to analyse information about land-holding amongst individuals, institutions, males, females, scheduled castes and scheduled tribes, etc. (see Figs. C1.7–C1.8).	• Land-holding analysis • Land-holding analysis–multidimensional • Top/bottom analysis • Medium-holding analysis
Essential commodities. To provide updated information on various essential commodity prices, NIC collects the retail/wholesale prices of various essential commodities like vegetables, sugar, rice, oil, cereals, etc. Using the GISTNIC data warehouse, end-users now have the updated information about trends in price change and will thus be able to closely monitor the prices more effectively (see Fig. C1.9).	• Commodity price analysis • Time-wise commodity • Qtr3–4 analysis • Rice analysis • Forecast—rice prices
Malaria statistics. This data mart has information on various health camps conducted across Tamil Nadu to detect and cure malaria patients. This has vital information like number of people suffering from malaria, deaths caused due to malaria, source of malaria infection, demographic information of malaria patients, etc. Using the data warehouse, the end-users will be able to plan various precautionary measures to reduce the number of people suffering from malaria in Tamil Nadu (see Fig. C1.10).	• MDR census on samples collected and tested • MDR source of malarial parasites • MDR age- and sex-wise malarial census • Graph- and sex-wise malarial census
Indian economy. This data mart has information about statistics on the telecom sector, stock exchange (NSE and BSE) and India's foreign trade. This data is collected on monthly basis from CMIE, Mumbai.	• Capital market analysis • Capital market analysis—MDDB report • Basic telecom analysis—overview • Basic telecom analysis—state-wise • External trade analysis (1997–1998) • Combine report

(contd.)

Table C1.1 Data Marts of various sectors and their applications *(contd.)*

Data mart	Applications
School health. This data mart has information about various health check-up camps conducted in various schools across Tamil Nadu. It has information about students suffering from various diseases, defects, immunization programmes, etc. (see Fig. C1.11)	• Disease analysis—MDDB report • Disease analysis—graph • Immunization analysis—MDDB • Immunization analysis—graph

***Fig.* C1.1** The first Web-enabled data warehouse for Tamil Nadu.



Medical Facility [ANA]		Available	Not Available
		Village Count	Village Count
District Name		N	N
COIMBATORE		191	233
DHARMAPURI		195	691
DINDIGUL ANNA		229	119
ERODE	BHAVANI	32	21
	DHARAPURAM	21	22
	ERODE	29	38
	GOPICHETIPALAYAM	34	29
	KANGAYAM	19	12
	PERUNDURAI	33	72
	SATHYAMANGALAM	55	14
KANNIYAKUMARI		76	5
MADURAI		276	295
NILGIRI		35	7
PUDUKKOTTAI		292	330
RAMANATHAPURAM		164	241
SALEM		417	450
SIVAGANGAI		163	295
SOUTH ARCOT		594	1532
THANJAVUR		460	1174

Fig. C1.2 SAS/multidimensional report on the village amenities data. This report is used to analyse villages in Tamil Nadu based on availability of various amenities.

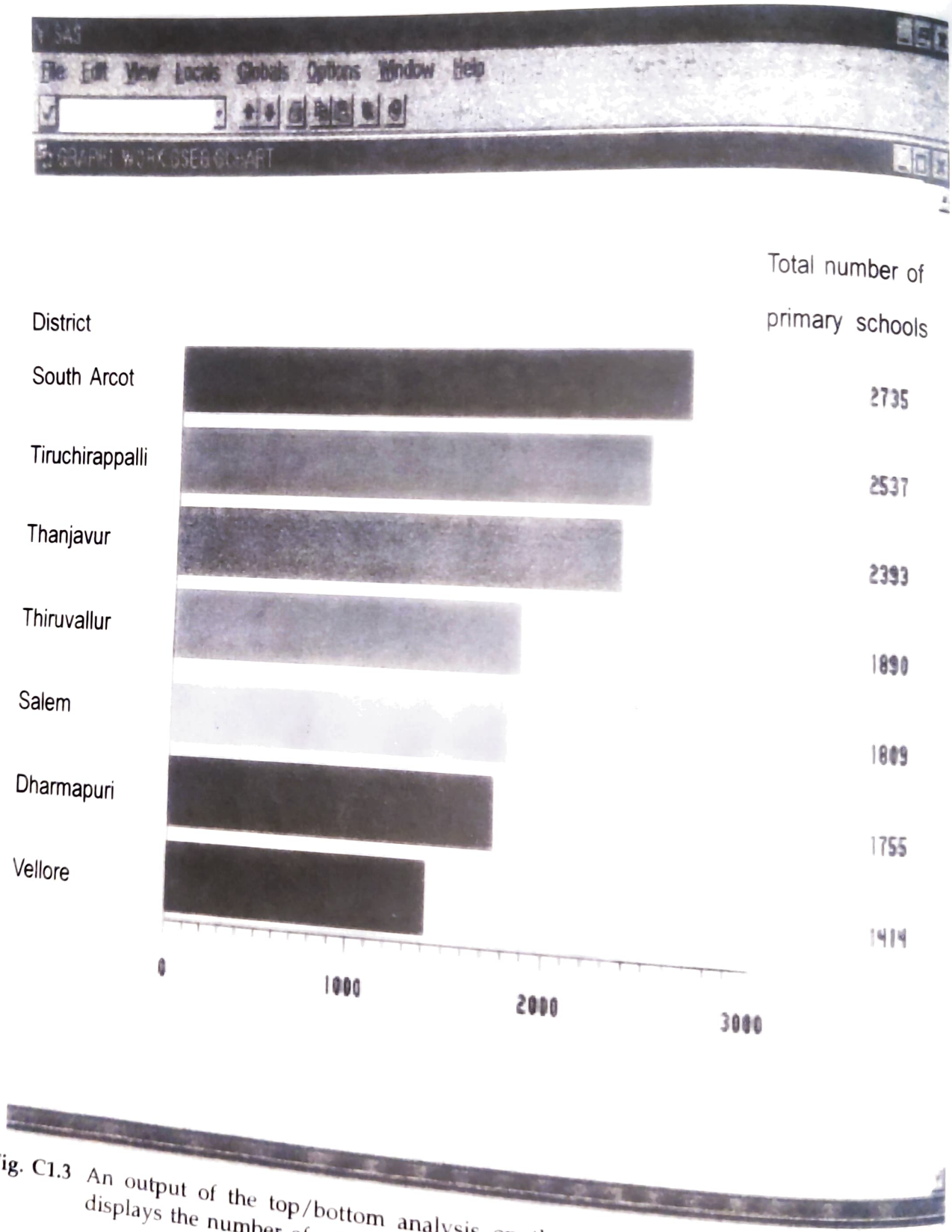


Fig. C1.3 An output of the top/bottom analysis on the village amenities data. This screen displays the number of primary schools in top seven districts in Tamil Nadu.

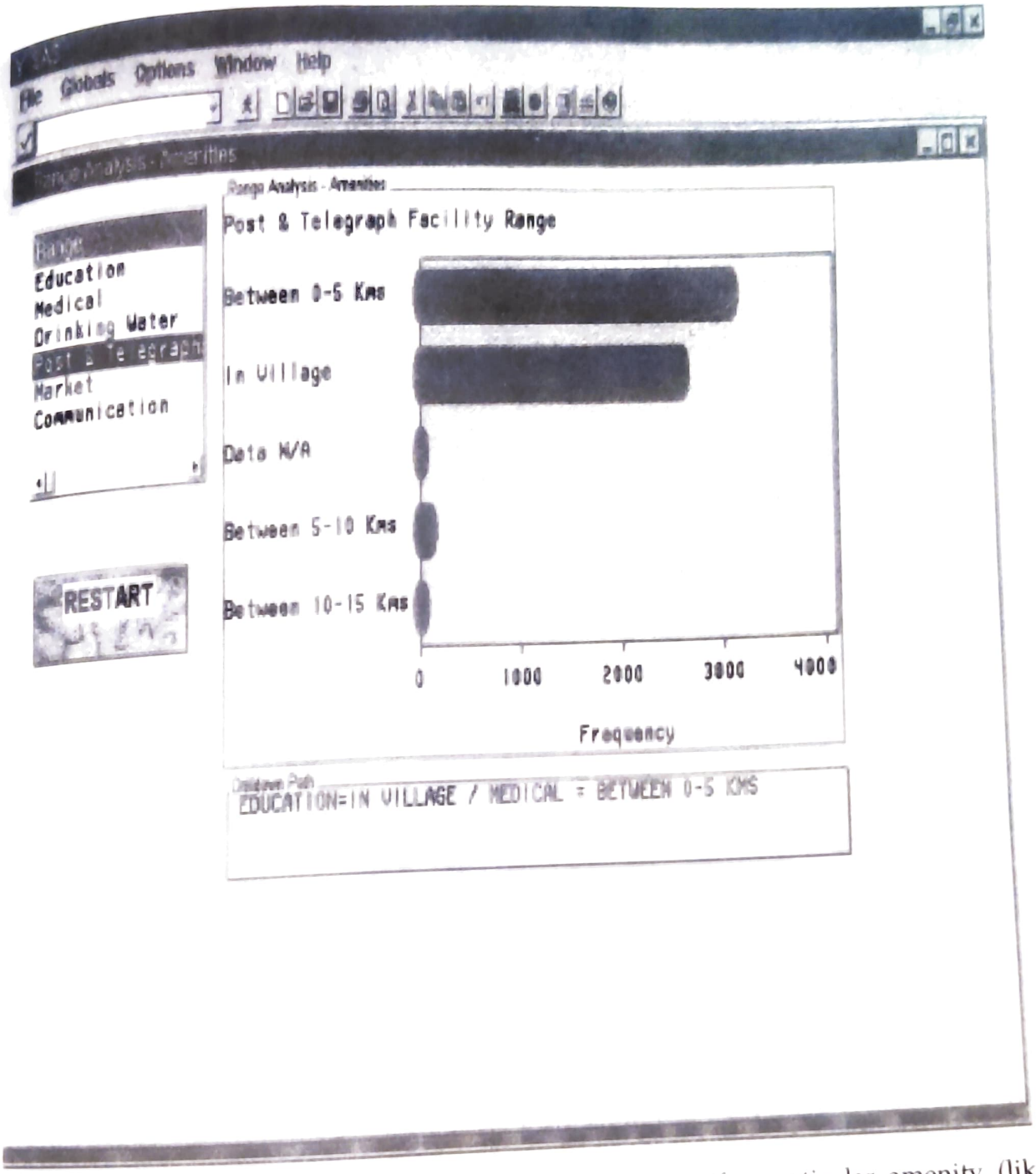


Fig. C1.4 An amenities range analysis screen. It helps analyse if a particular amenity (like education) is not available in the village, then how far (0–5 km, 5–10 km, 10–15 km) is it available.

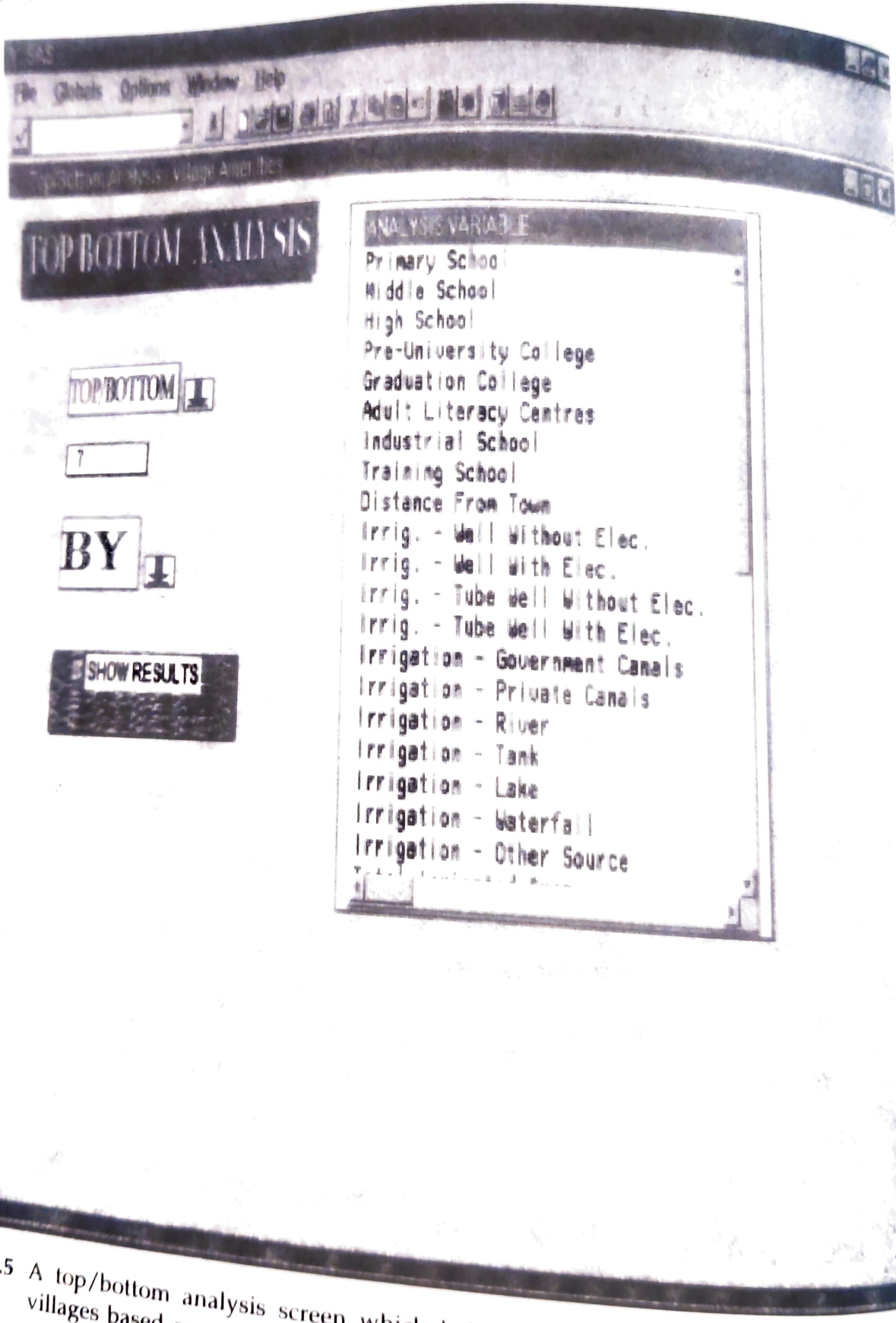


Fig. C1.5 A top/bottom analysis screen which helps in listing names of districts/talukas/villages based on various parameters mentioned in the list box.

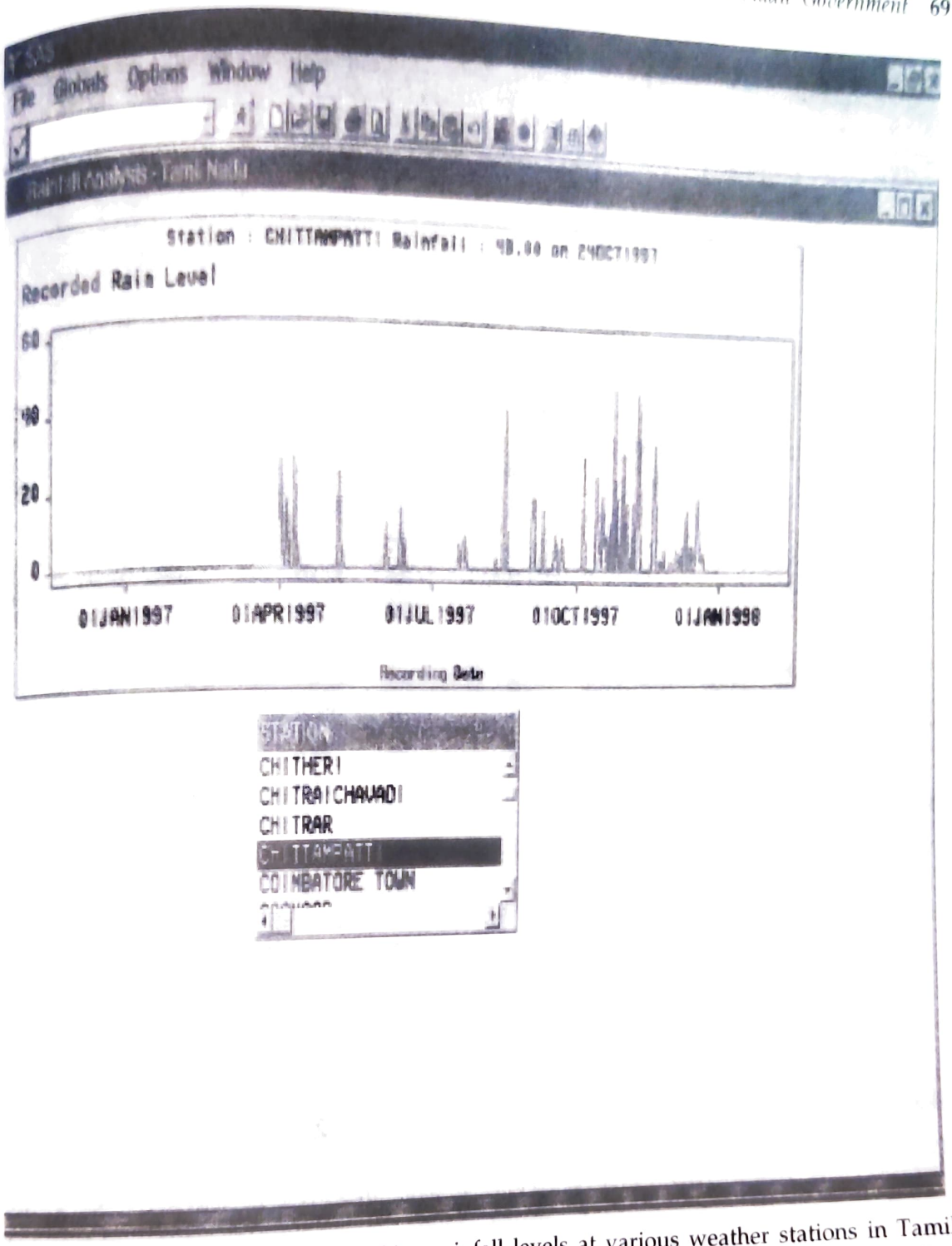
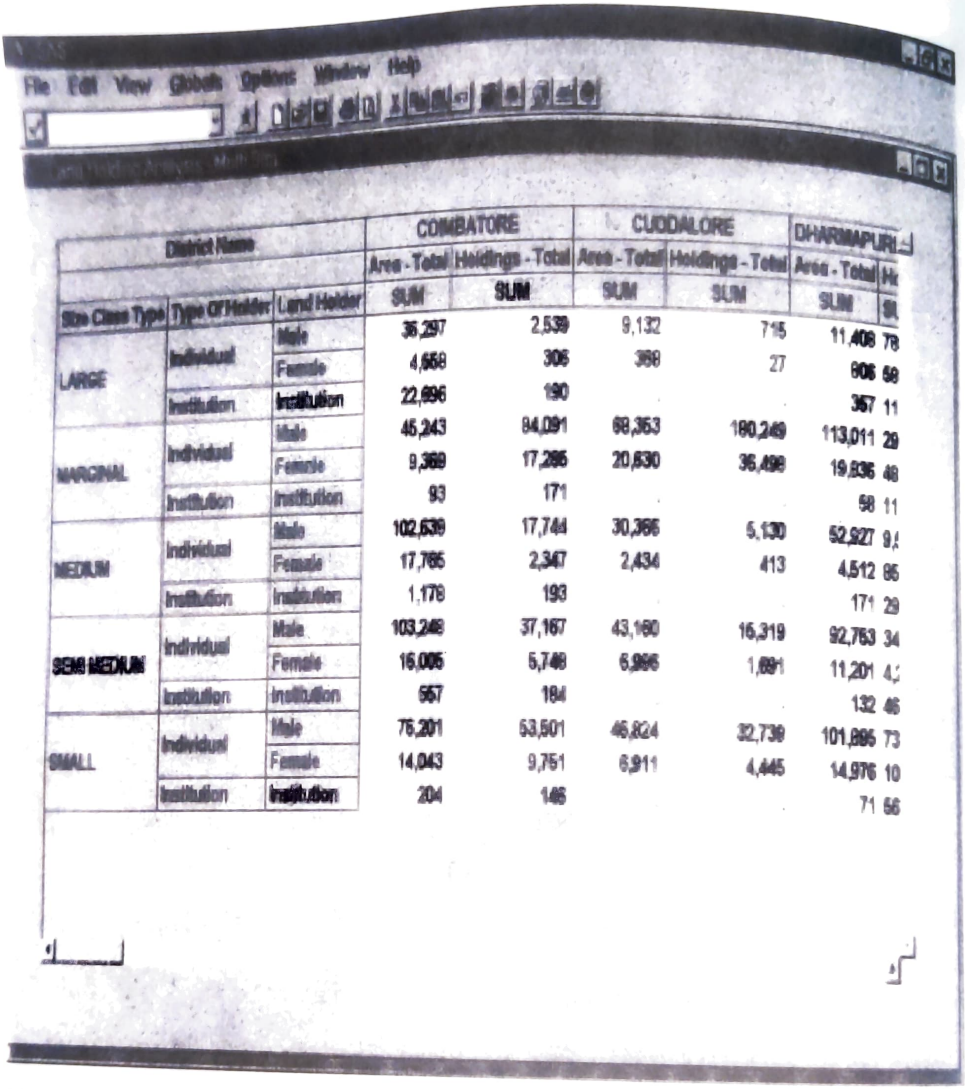


Fig. C1.6 A line plot screen for tracking rainfall levels at various weather stations in Tamil Nadu. On selecting one of the weather stations in the list box on the screen, the line plot changes to reflect the rainfall level for the selected weather station. On clicking any point on the line plot the graph displays the data and rainfall level for the data point.



District Name			COIMBATORE		CUDDALORE		DHARMAPURI	
			Area - Total	Holdings - Total	Area - Total	Holdings - Total	Area - Total	Ho
Size Class Type	Type of Holder	Land Holder	SUM	SUM	SUM	SUM	SUM	SU
LARGE	Individual	Male	36,297	2,539	9,132	715	11,408	78
		Female	4,558	306	368	27	806	59
	Institution	Institution	22,696	190			357	11
MARGINAL	Individual	Male	45,243	84,091	69,353	180,249	113,011	29
		Female	9,369	17,286	20,830	36,498	19,836	48
	Institution	Institution	93	171			58	11
MEDIUM	Individual	Male	102,539	17,744	30,366	5,130	52,927	9,5
		Female	17,765	2,347	2,434	413	4,512	86
	Institution	Institution	1,178	193			171	29
SEMI MEDIUM	Individual	Male	103,248	37,167	43,160	16,319	92,753	34
		Female	16,005	5,749	6,996	1,891	11,201	4,2
	Institution	Institution	557	184			132	46
SMALL	Individual	Male	75,201	53,501	46,824	32,739	101,895	73
		Female	14,043	9,751	6,911	4,445	14,976	10
	Institution	Institution	204	146			71	56

Fig. C1.7 A SAS/EIS multidimensional report on the agricultural census data. The report is displaying area under holding/number of holdings based on size, type of holder, sex of the holder, name of the district, etc.

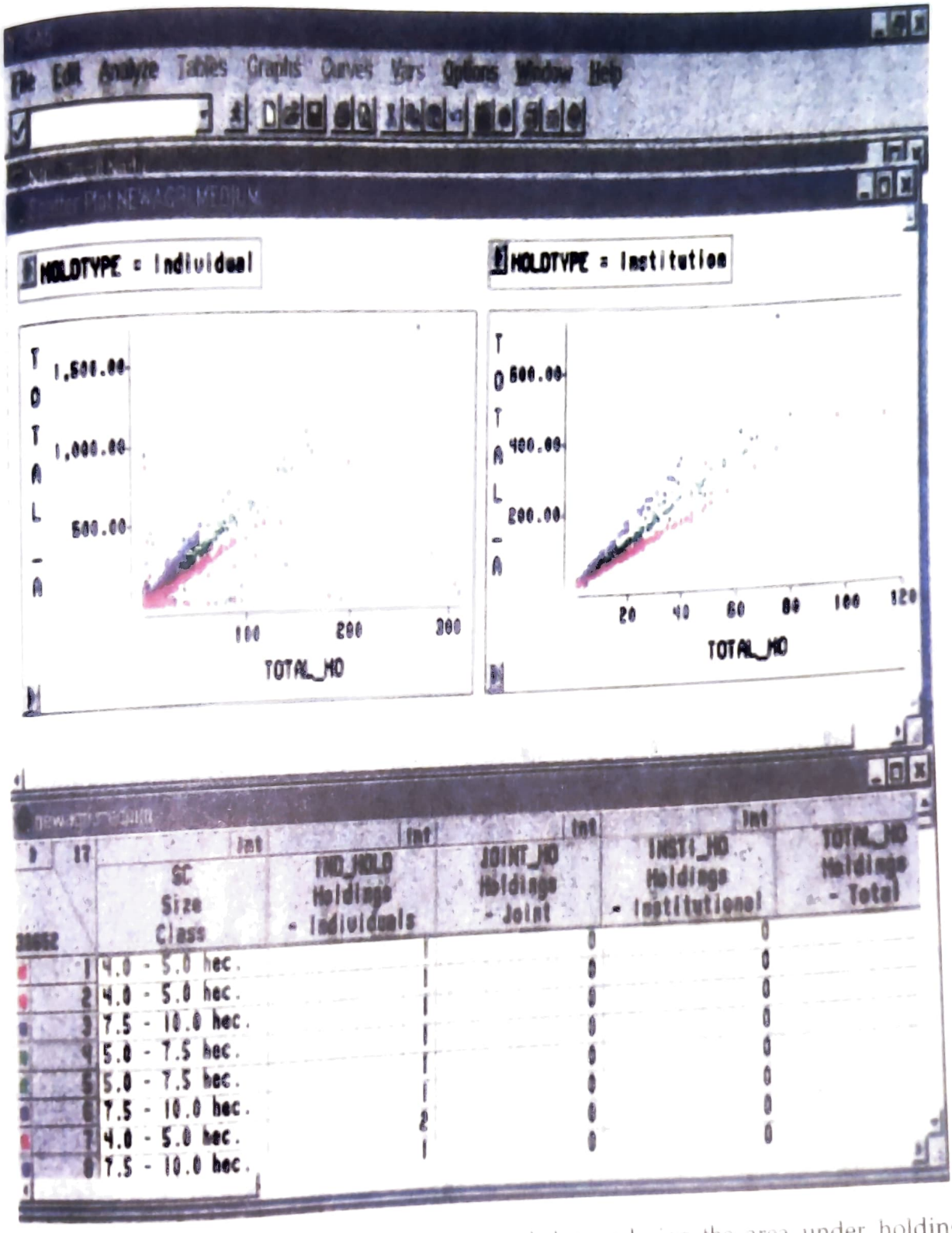


Fig. C1.8 A data visualization screen. The screen helps analysing the area under holdings for individuals and institutions. The various colour patterns in the data reflect the various sizes of the holdings.

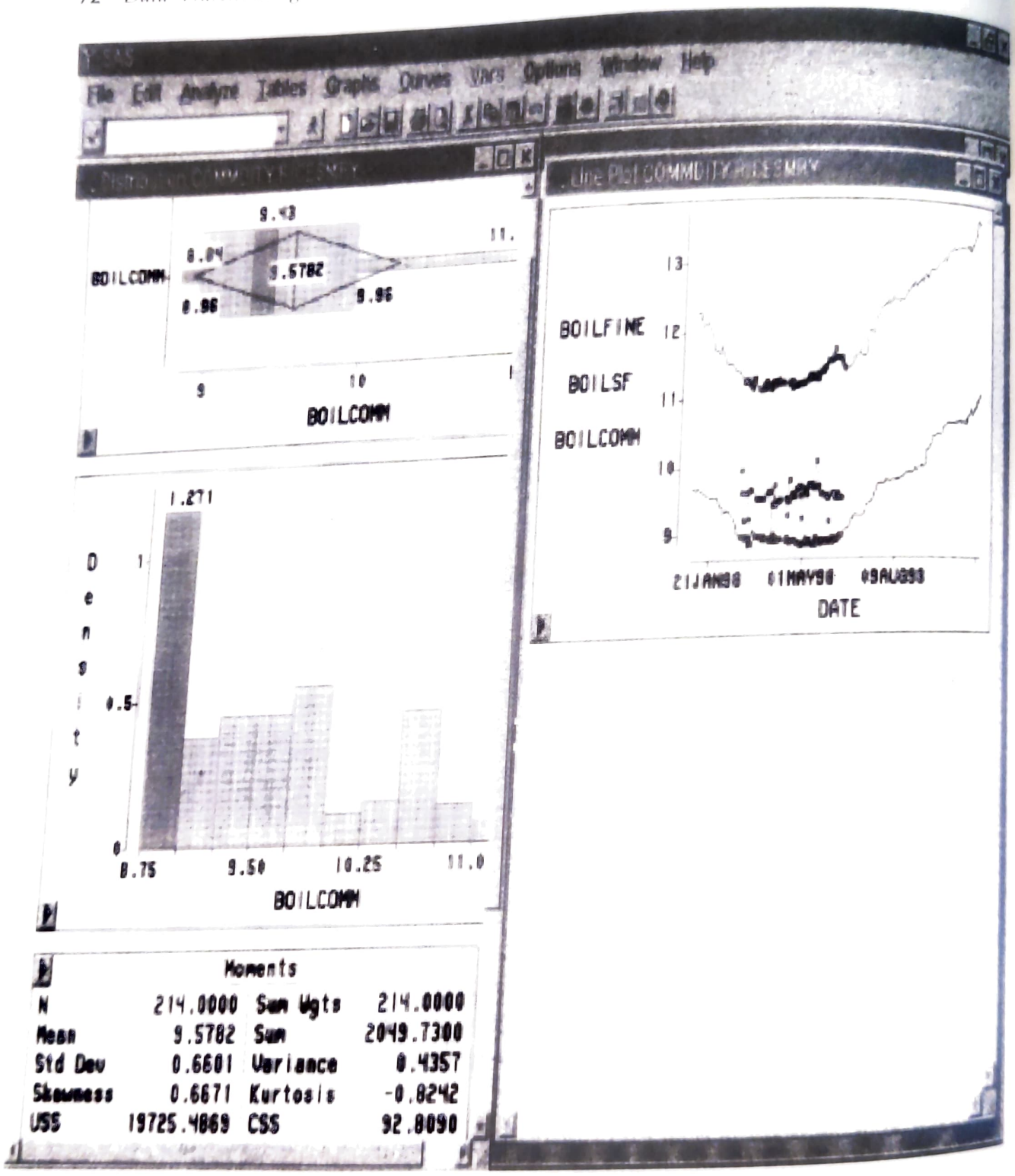


Fig. C1.9 An output of analysis on the retail price of rice in 1998. It displays a line plot, box plot, histogram and certain descriptive statistics on rice.

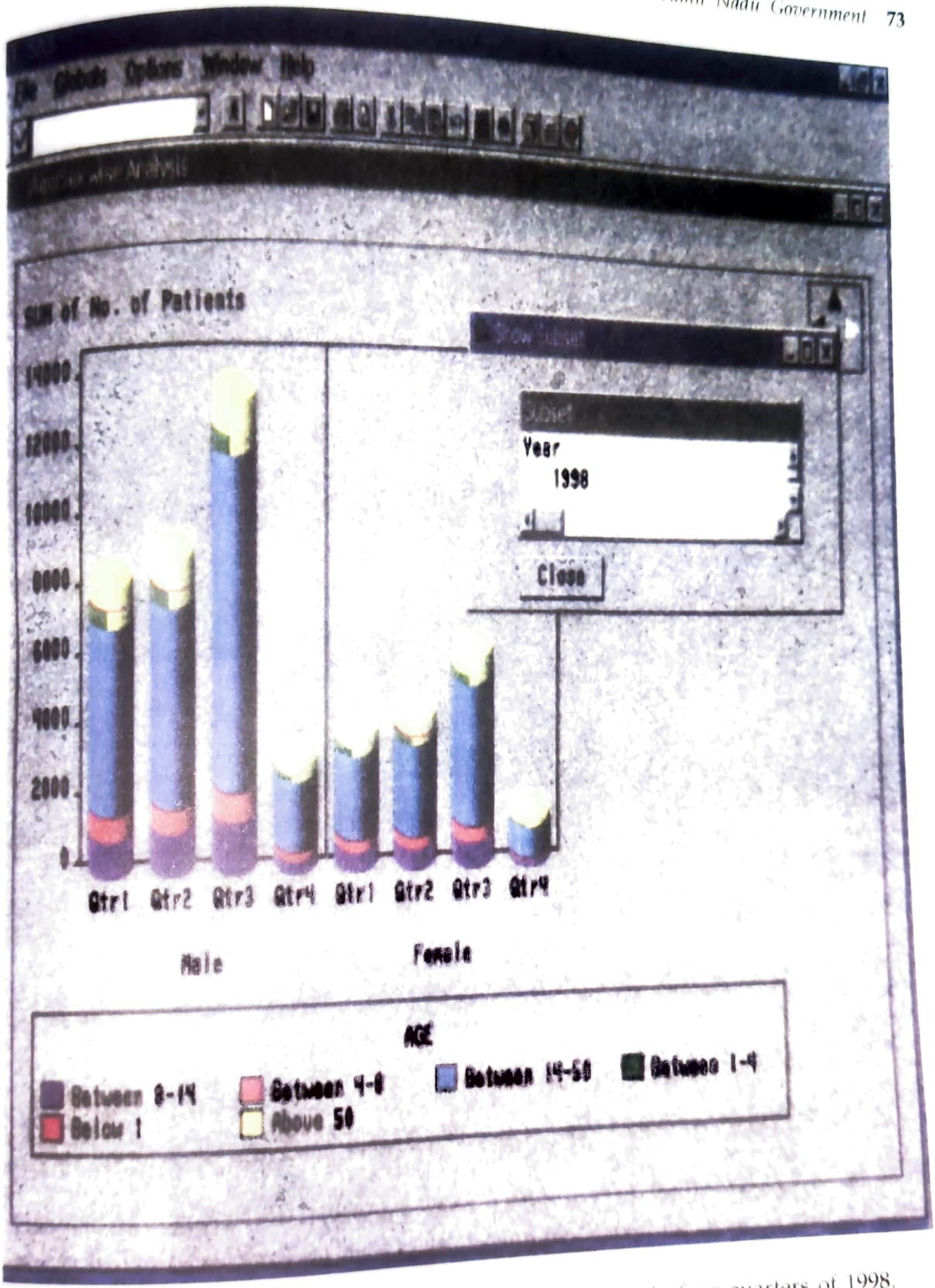


Fig. C1.10 Number of the male/female malaria patients in four quarters of 1998.

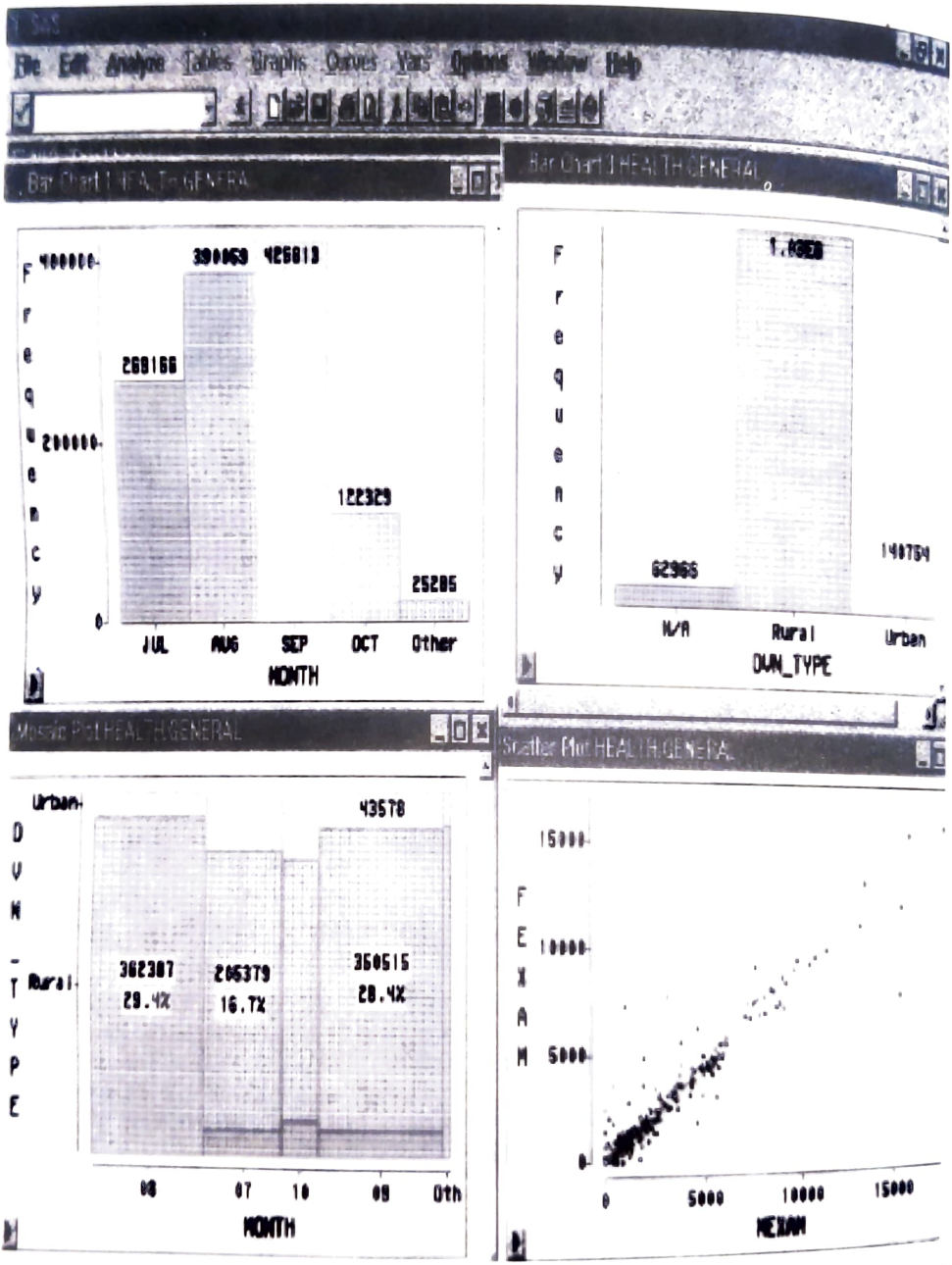


Fig. C1.11 This screen displays the number of male students examined in the school health camps.

CASE STUDY

3 Data Warehouse for the Government of Andhra Pradesh

C3.1 A DATA WAREHOUSE FOR FINANCE DEPARTMENT

C3.1.1 Responsibilities of the Finance Department

The finance department of the Government of Andhra Pradesh has the following responsibilities:

1. Preparing a department-wise budget up to the sub-detail head and submission to the legislature for its approval.
2. Watching out the government expenditure and revenue department-wise.
3. Looking after development activities under various plan schemes.
4. Monitoring other administrative matters related to all heads of departments.

C3.1.2 Treasuries in Andhra Pradesh

Money standing in the government account are kept either in treasuries or in banks. Money deposited in the banks shall be considered as general fund held in the books of the banks on behalf of the State. Treasuries have the following officers:

Director of Treasuries. Treasuries are under the General Control of the Director of Treasuries and Accounts.

District Treasury Officers (DTO). The immediate executive control of treasuries in a district vests in the District Treasury Officer who is subordinate to the Director of Treasuries and Accounts. The District Treasury Officer is responsible for the proper observance of the procedures prescribed by the rules. He also monitors the punctual submission of all returns required by the treasury by the State Government, the Accountant General and the Reserve Bank of India.

Sub-treasuries

If the requirements of the public business make necessary the establishment of one or more sub-treasuries under a district treasury, the arrangement for the same is made by the Finance Minister after consultation with the Accountant General.

The accounts of receipts and payments at a sub-treasury must be included monthly in the accounts of the district treasury.

Treasuries handle all the government receipts and payments, receipts are paid through 'challans'. (A challan is a form which is used for all types of receipts.) Payments against challans are based on the submission of different types of bills by the drawing officers. These bills may be: paybills, abstract contingent bills, advance GPF, travelling allowances, fully vouchered contingent bills, deposit repayments, pensions, grants-in-aid, refunds, scholarships and loans. Every transaction in the government is made through related departments.

Description of the account head	Length of the code (digit)
Major	4
Sub-major	2
Minor	3
Group sub-head	1
Sub-head	2
Detail head	3
Sub-detail head	3

Altogether it is an 18-digit code.

Each department (or scheme or deposit account) is identified by major head. Other levels are related to respective sub-levels of the department. Major heads are broadly classified as five categories:

1. Less than 2000 : Receipts
2. 2000 < 4000 : Service major heads
3. 4000 < 6000 : Capital outlay
4. 6000 < 8000 : Loans
5. More than 8000 : Deposits

Release of budget to all heads of departments of different government schemes are passed by the legislature, this is known as *voted transaction under plan*.

A project for building a data warehouses (DW) for online analytical processing (OLAP) is implemented by NIC for the department of treasuries. The concept of building DW in the department of treasuries has been established for providing easy access to integrated up-to-date data related to various aspects of the departments functions (revenue, expenditure, economic, financial, etc.). In the Treasury Department, DW technology is used to develop analytical tools designed to provide support for decision-making at all levels of the department.

The information accumulated in most of the treasuries cannot be efficiently used in the framework of traditional Information Systems. Traditional Information Systems implemented in the Department of Treasuries are based on transactional databases (OLTP) which are not designed for providing fast and efficient access to information critical for decision-making. One of the main limitations inherent in such Information Systems is that the managers responsible for decision-making are not provided with direct access to historical business data accumulated over a significant period of time.

Data required for analysis are typically distributed among a number of isolated Information Systems meeting the needs of different sub-treasuries. In the case of complicated queries on a database, the system response time is excessively high. Data models used are not suitable for decision support systems since they are specifically designed to handle only short transactions.

Thus the information stored in traditional systems cannot be efficiently retrieved by managers and analysts involved in decision-making.

Data warehouse technology provided to the Department of Treasuries by National Informatics Centre eliminates these problems by storing current and historical data from disparate Information Systems. These information are required by business decision-makers in a single, consolidated system. The DW technology is based on utilizing multidimensional data modelling which represents a conceptual model of treasury business processes as a collection of facts, each characterizing just one of the features of such processes. This approach makes data readily accessible to the people who need it without interrupting online operational workloads.

Data warehouses provide efficient analysis and monitoring of financial data of treasuries, its structural subdivisions and subsidiaries. It also evaluates the internal and external business factors related to operational, economic and financial conditions of treasuries budget utilization.

The most important problems which can be addressed using DW and OLAP technologies within the Department of Treasuries include the following:

- Analysis of expenditure and revenue of heads of accounts by departments, drawing officers by regions, districts and sub-treasuries
- Operational and financial analyses of the Department of Treasuries and
- Evaluation and monitoring of the financial positions.

National Informatics Centre (NIC) assisted the Department of Treasuries with services in building data warehouses and systems for online analytical processing. NIC officers work closely with the clients, thereby transferring state-of-the-art technologies for developing, updating and maintaining the DW. Some software tools for building DW and OLAP applications are SQL Server 7.0 with OLAP services and ASP 3.0.

C3.1.3 Dimensions Covered Under Finance Data Warehouse

Following are the different *dimensions* taken for drill-down approach against two measures—payments and receipts:

- Department
- District Treasury Office
- Sub-Treasury Office
- Drawing and Disbursing Officer
- Time (year/month/week/day)
- Bank-wise
- Based on different forms (bills)

Refer to Figs. C3.1–C3.14.

C3.1.4 COGNOS Graphic User Interface For Treasuries Data Warehouse

The features of COGNOS PowerPlay version 6.0 used in this category are discussed as follows:

Impromptu. *Impromptu* is used for generating various kind of reports like simple, crosstab, etc. Once an Impromptu report has been published on HTML, one can view the report using the Web browser. Here a Web browser is required but not Impromptu to view HTML reports.
One can view an HTML report:

- on Internet or Intranet Web page;
- on a network; and
- that has been sent via e-mail.

Transformer. *Transformer* model objects may contain definitions of queries, dimensions, measures, dimension views, user classes and related authentication information, as well as objects for one or more cubes that Transformer creates for viewing and reporting in PowerPlay. Transformer stores models as files with the extensions.

Once the model is ready, creating one or more cubes or cube groups based on the model contents is possible. By creating several cube objects and basing different cubes on those objects, one can create subset cubes that serve the needs of distinct user groups. One can also create cubes that provide drill-through access to other cubes.

PowerPlay. COGNOS PowerPlay is used to populate reports with drill-down facility. Popular reports are as follows (also refer to Figs. C3.8–C3.14):

- Dynamic rank position
- Financial report
- Business trend
- Comparative performance
- Foreign currency report

Scheduler. *Scheduler* coordinates the execution of automated processes, called tasks, on a set date and time, or at recurring intervals. Scheduler supports tasks that run once and tasks that run repeatedly. Through Scheduler, Impromptu users can submit Impromptu report requests to be executed either locally, or by an Impromptu Request Server.

Authenticator. *Authenticator* is a user class management system. It provides COGNOS client applications with the ability to create and show data based on user-authenticated access. It also serves as a repository for log-in information, thus providing client applications with auto-access to data sources and servers.

Following are a few sample reports generated by using the above methods (Figs. C3.1–C3.14):

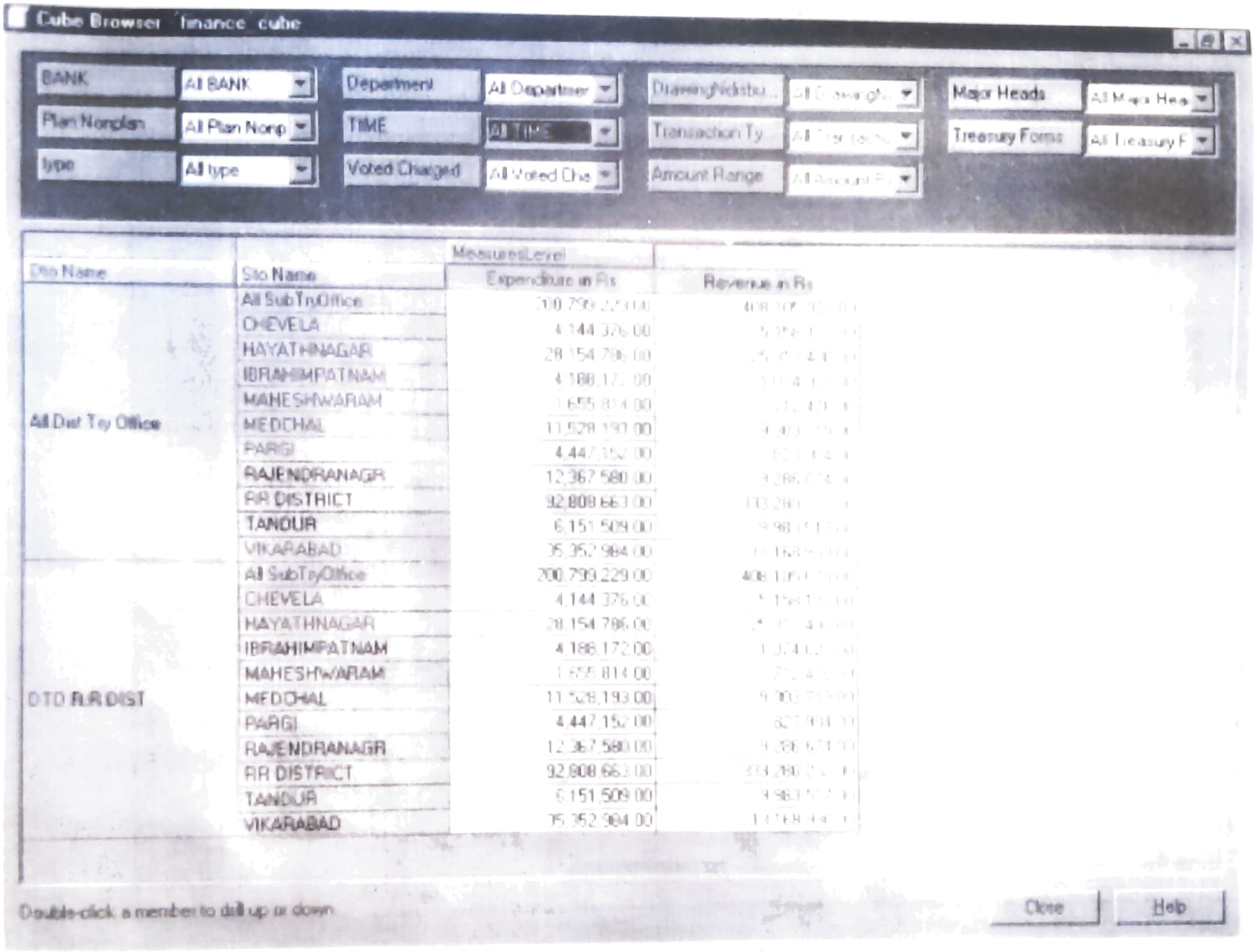


Fig. C3.1 DTO and STO wise breakup of Income and Expenditure. (DTO—District Treasury Officer; STO—Sub Treasury Officer.)

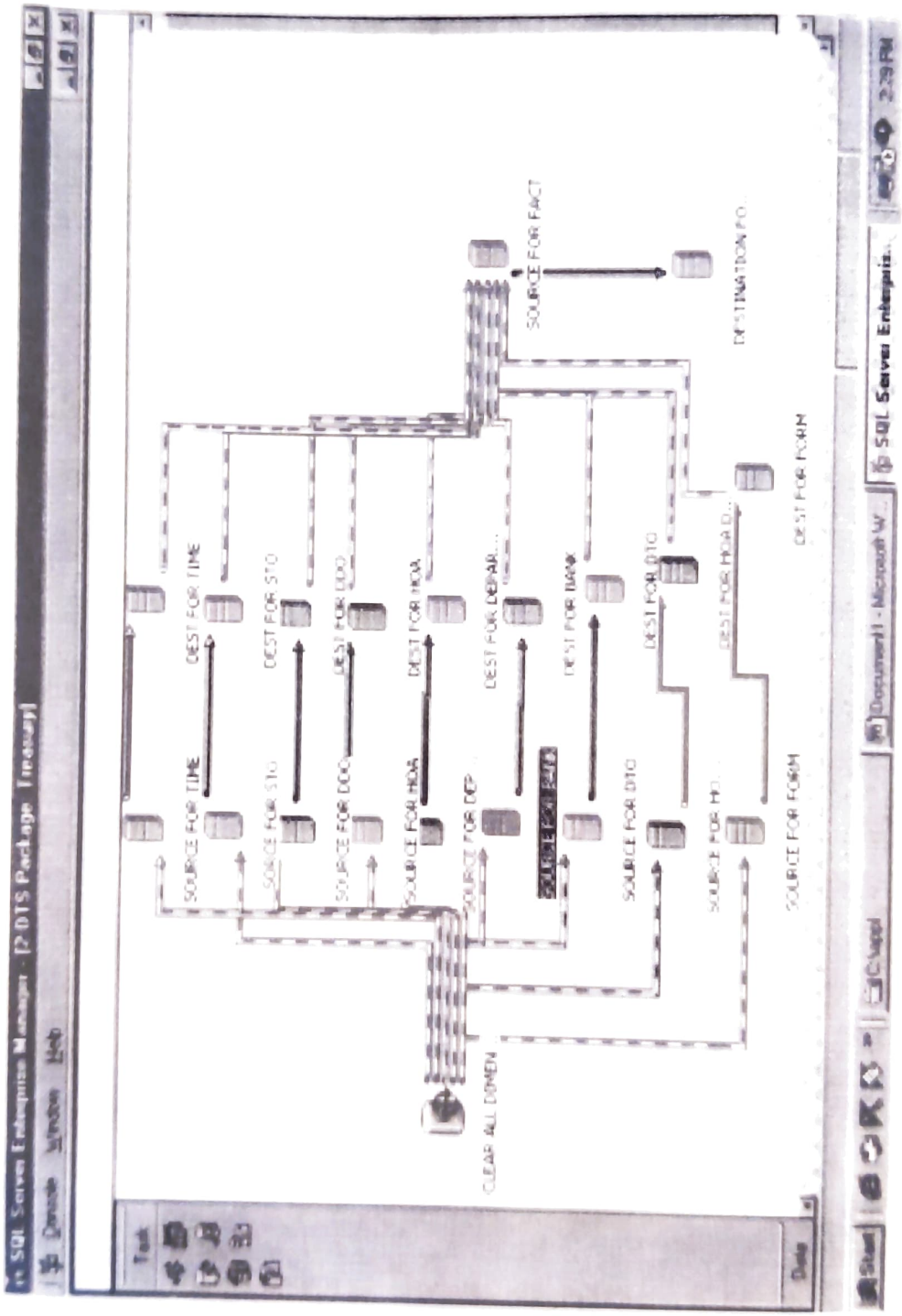


Fig. C3.2 Data transformations services of Finance data cube.

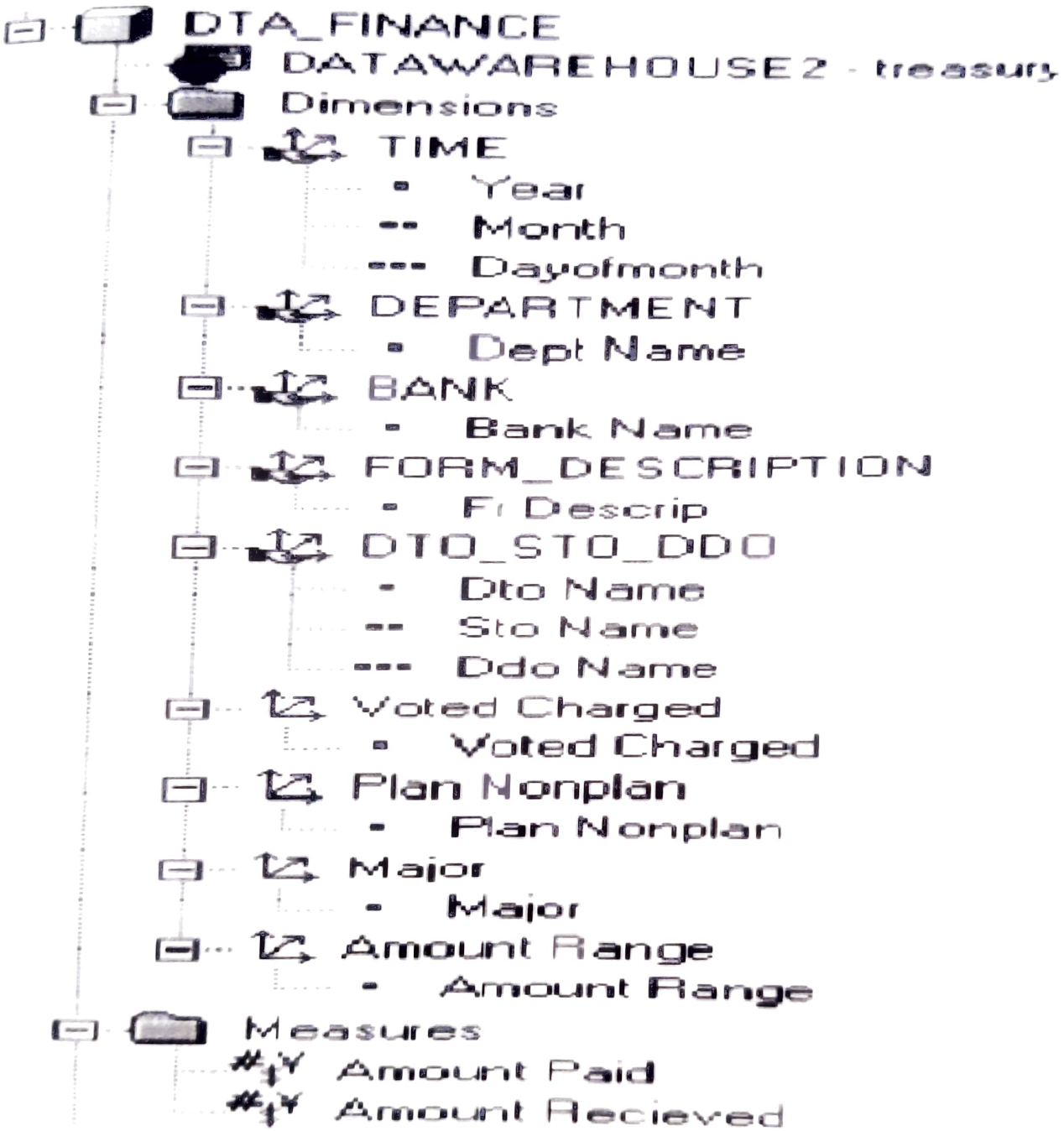


Fig. C3.3 Public, Private dimensions and measures of Finance cube.

dta_finance

Amount Range	BANK	DEPARTMENT	FORM_DESCRIPTION	Major	Plan Nonplan	Voted Charged
All Amount Range	All BANK	All DEPARTMENT	All FORM_DESCRIPTION	All Major	Plan	Voted

			Year / Month Dayofmonth			
			⊞ 2000		Grand Total	
Dto Name	Sto Name	Ddo Name	Amount Recieved	Amount Paid	Amount Recieved	Amoun
⊟ DTO R.R	⊞ CHEVELA		0	292193	0	2
	⊞ HAYATHNAGAR		0	325729	0	3
	⊟ IBRAHIMPATNAM	PROJECT OFICER NFE IBRPATAN	0	200053	0	2
		CDPO ICDS IPATAN	11335	174176	11335	1
		A D A SC RR DIST	0	151000	0	1
		M O PHC MANCHAL	0	78862	0	
		M O PHC DANDU MAILARAM	0	70540	0	
		ASST SM RFWPC YACHARAM	0	23384	0	
		EE PR(RWS)DIV RR DIST	0	9121	0	
		GAZETTED ADMN OFFICER DEO RR	0	6664	0	
		C A S PP UNIT IPATAN	0	4000	0	
		Total	11335	717800	11335	7
	⊞ MAHESHWARAM		0	201060	0	2
	⊞ MEDCHAL		0	479306	0	4
	⊞ PARGI		0	378698	0	3
	⊞ RAJENDRANAGR		0	143634	0	1

Fig. C3.4 Drill-down information of DTO/STO/DDO Revenue and Expenditure details. (DTO—District Treasury Officer; STO—Sub Treasury Officer; DDO—Drawing and Disbursement Officer.)

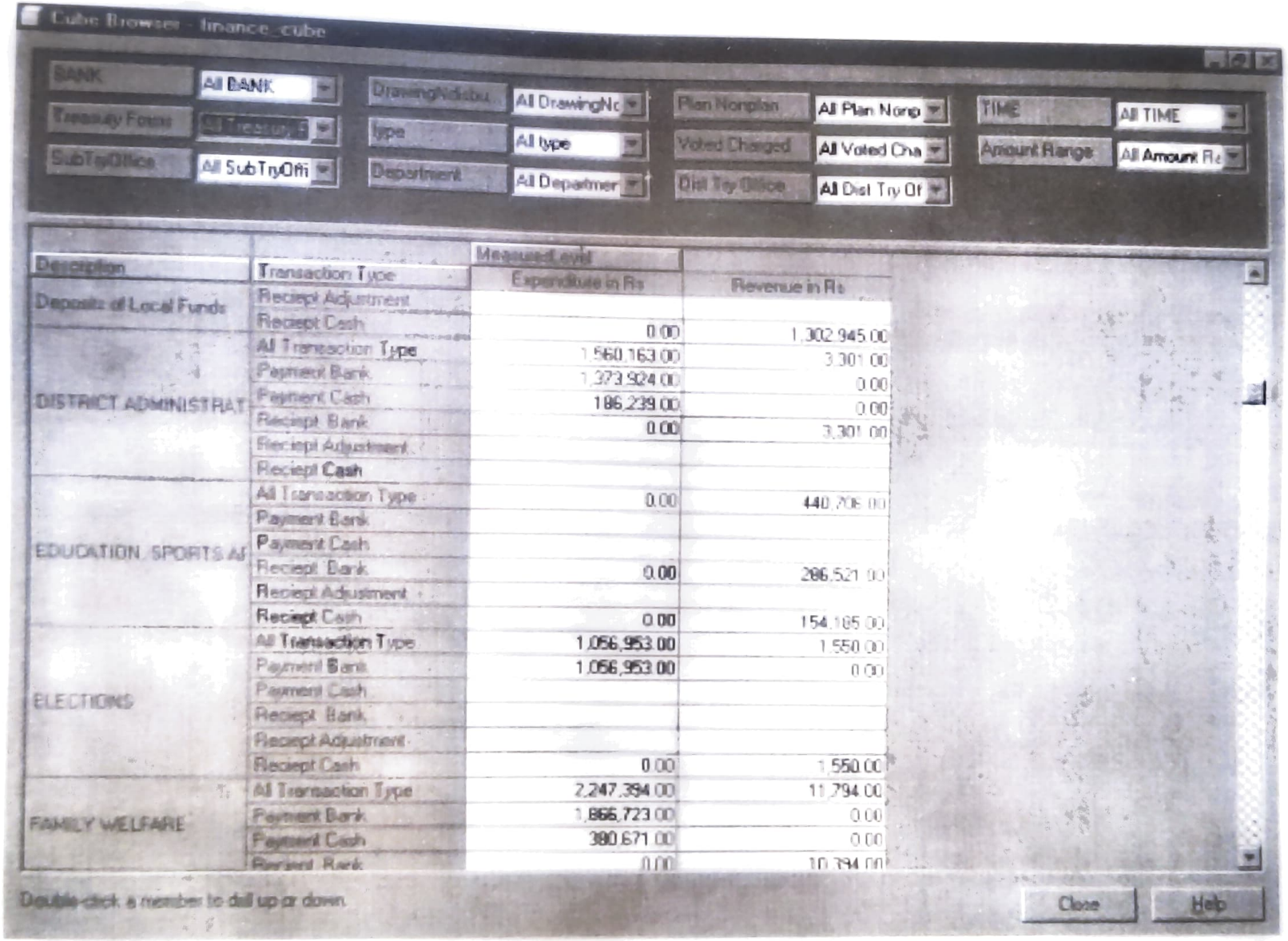


Fig. C3.5(a) Department-wise breakup of different types of transactions.

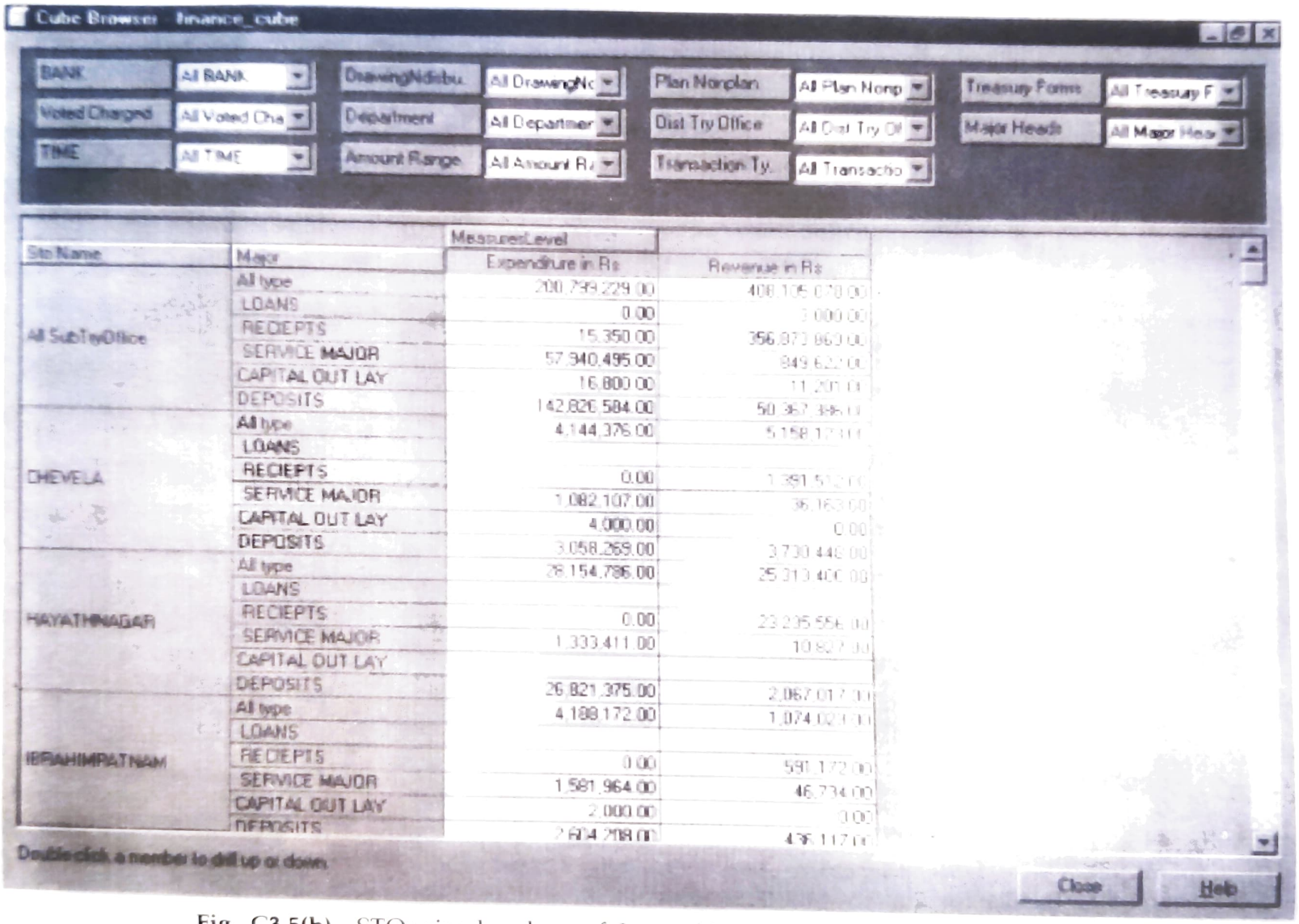


Fig. C3.5(b) STO-wise breakup of Loans/Receipts/Deposits/Capital outlay.

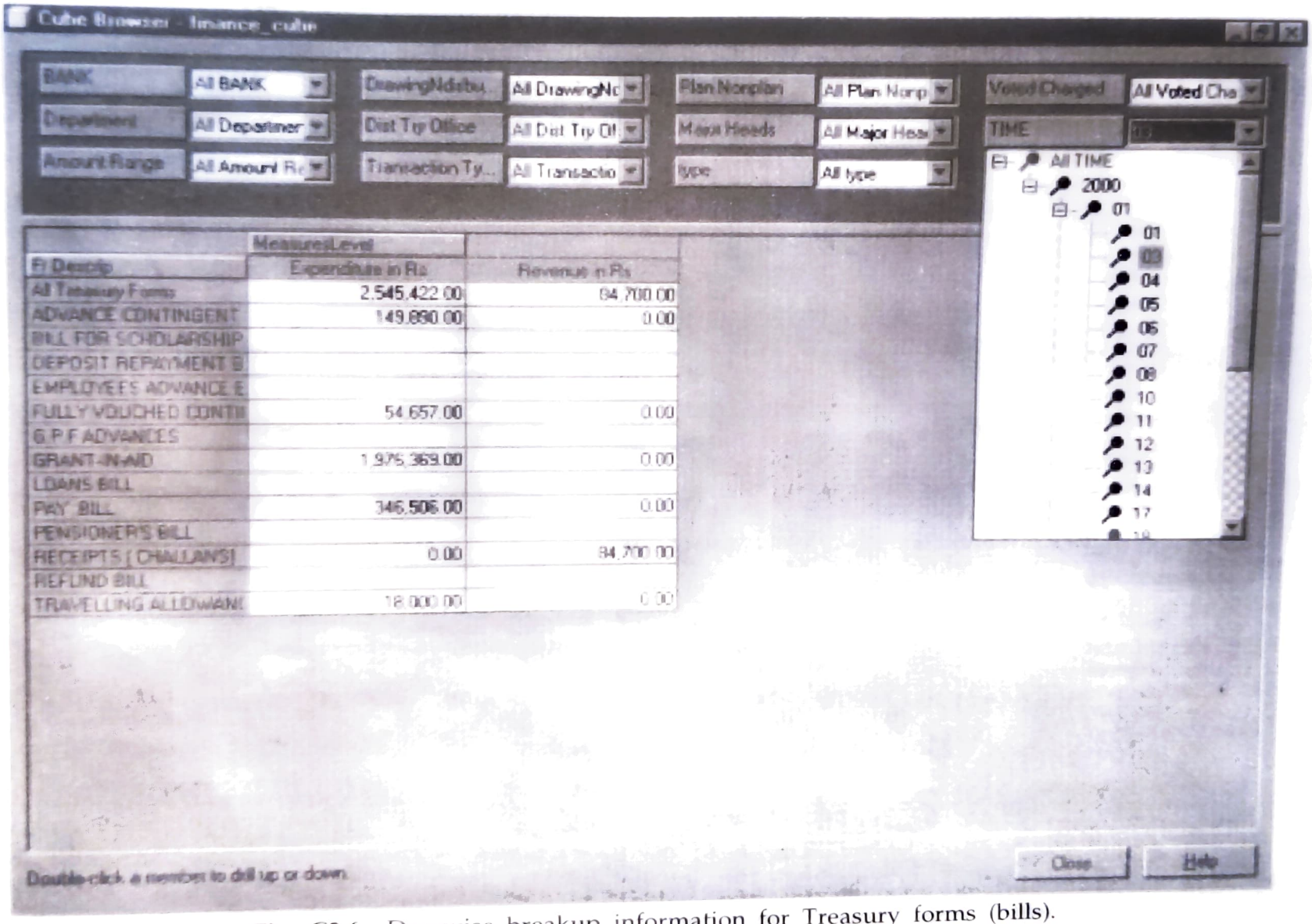


Fig. C3.6 Day-wise breakup information for Treasury forms (bills).

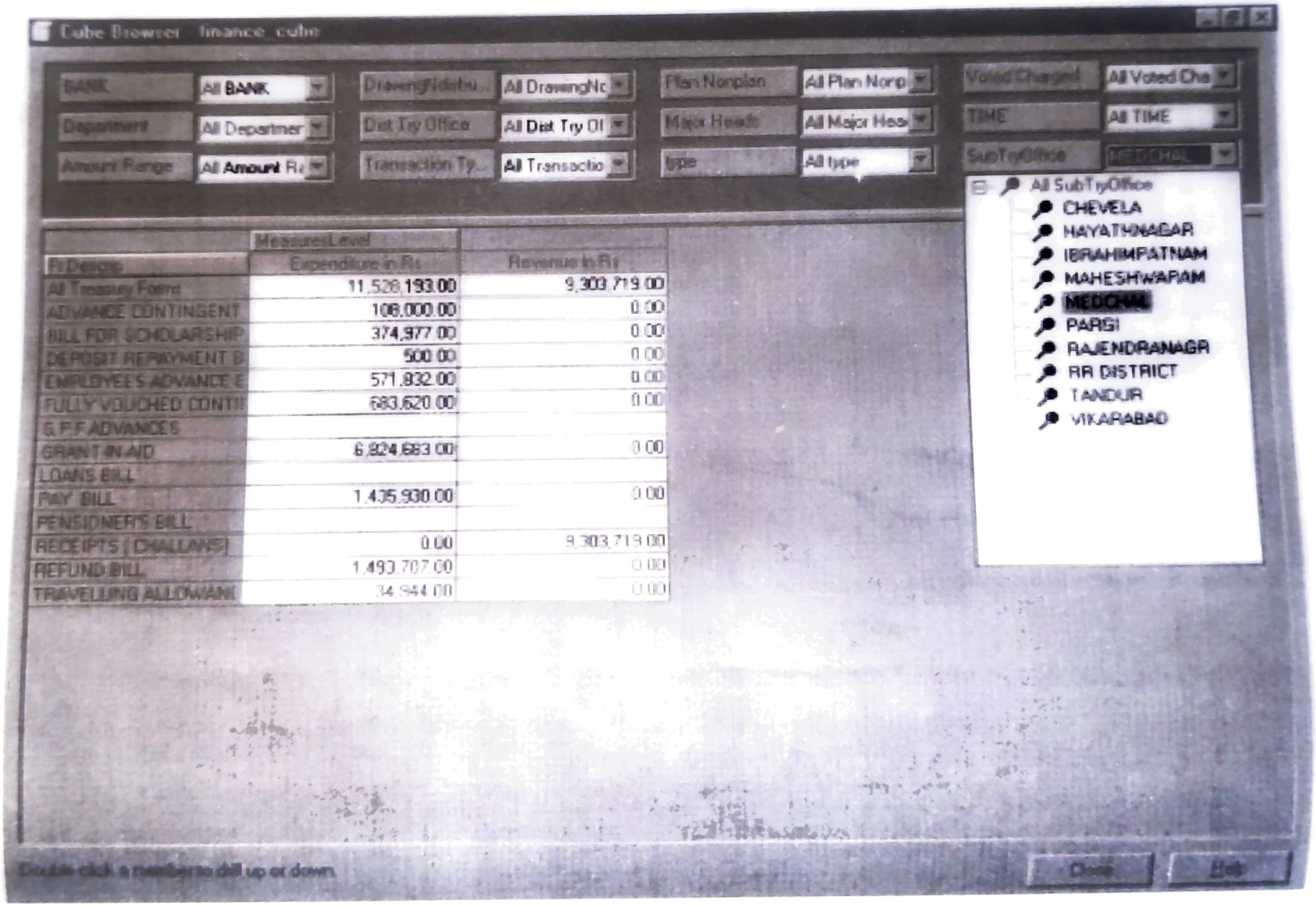


Fig. C3.7 Treasury form-wise breakup information for particular STO.

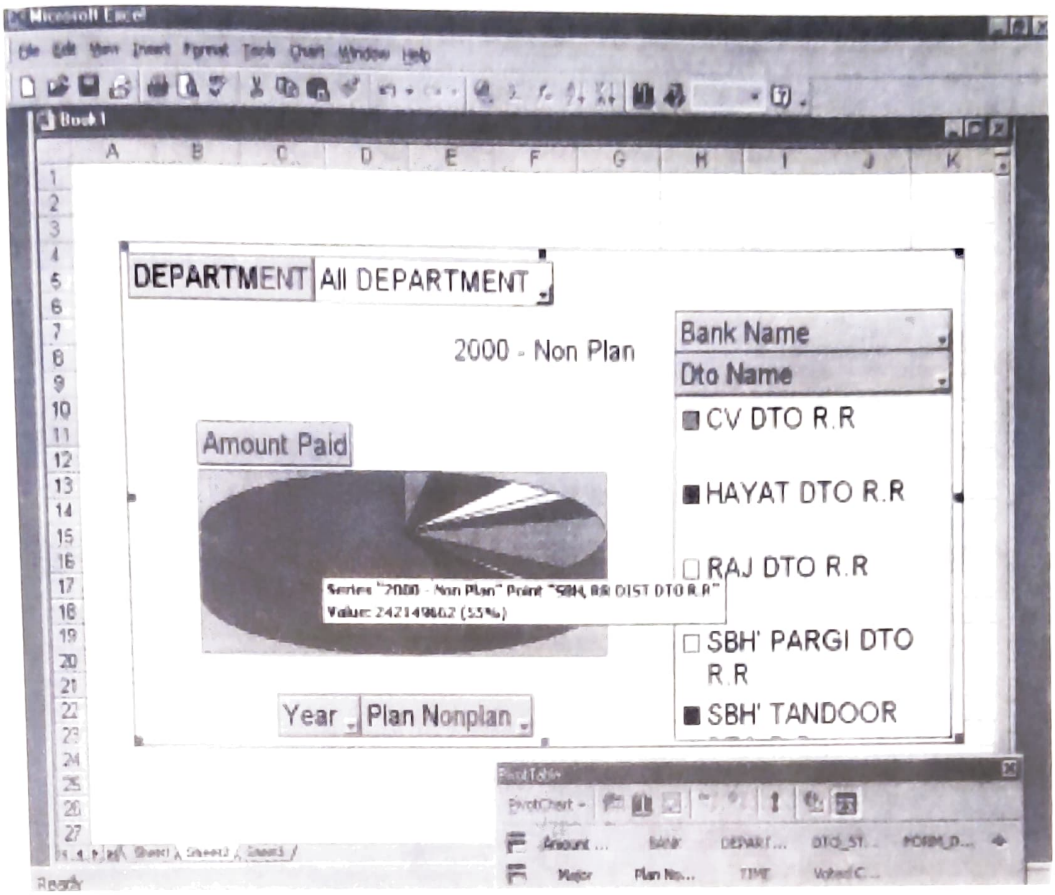


Fig. C3.8 Pivot table (graphical representation) for all departments, plan and non-plan-wise.

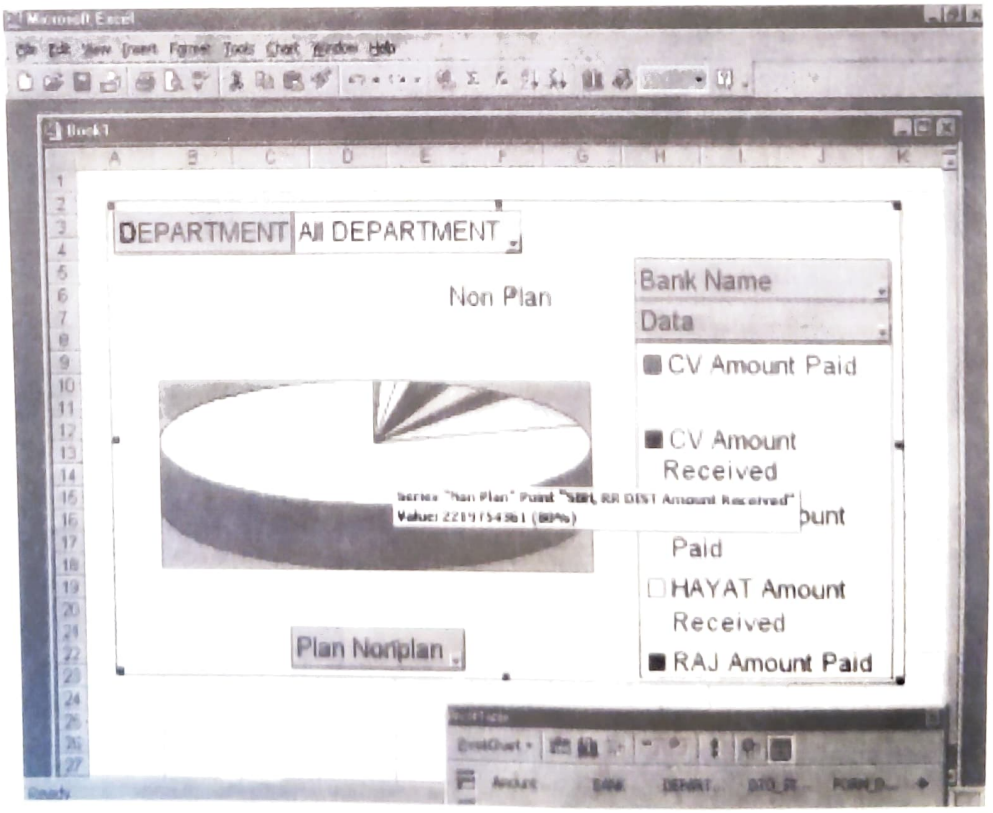


Fig. C3.9 Graphical representation of all departments, bank-wise.

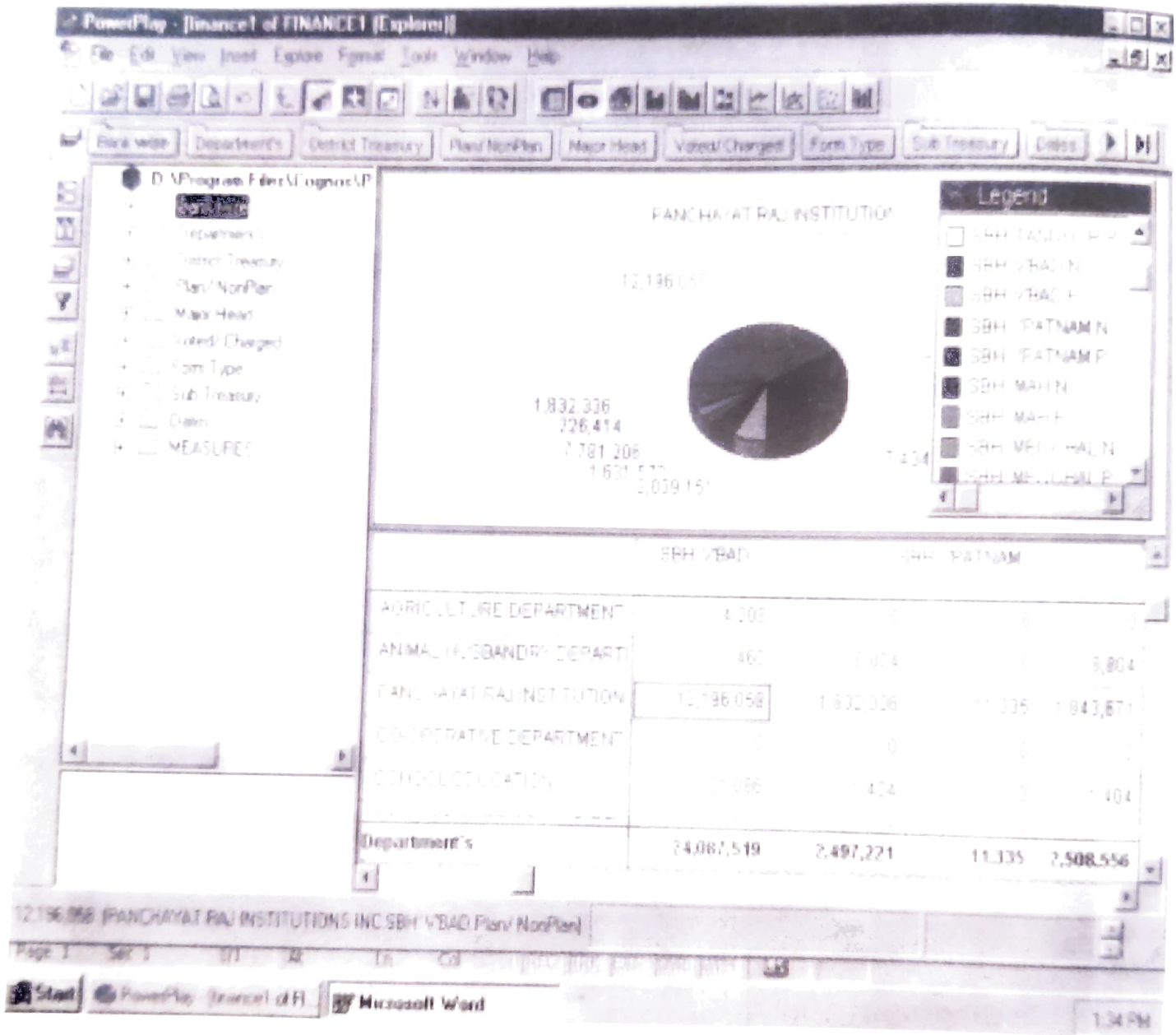


***Fig.* C3.10** COGNOS PowerPlay report on department/bank-wise expenditure.

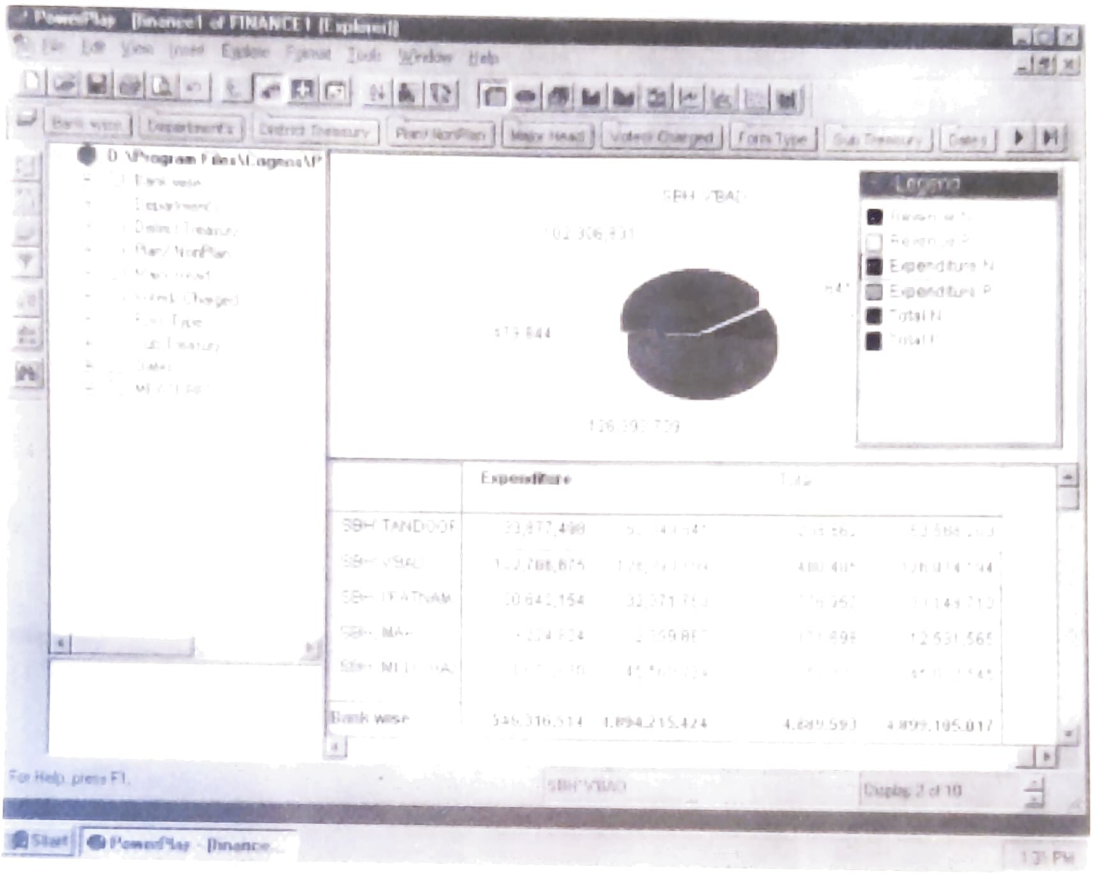


Fig. C3.11 Bank-wise expenditure statement.

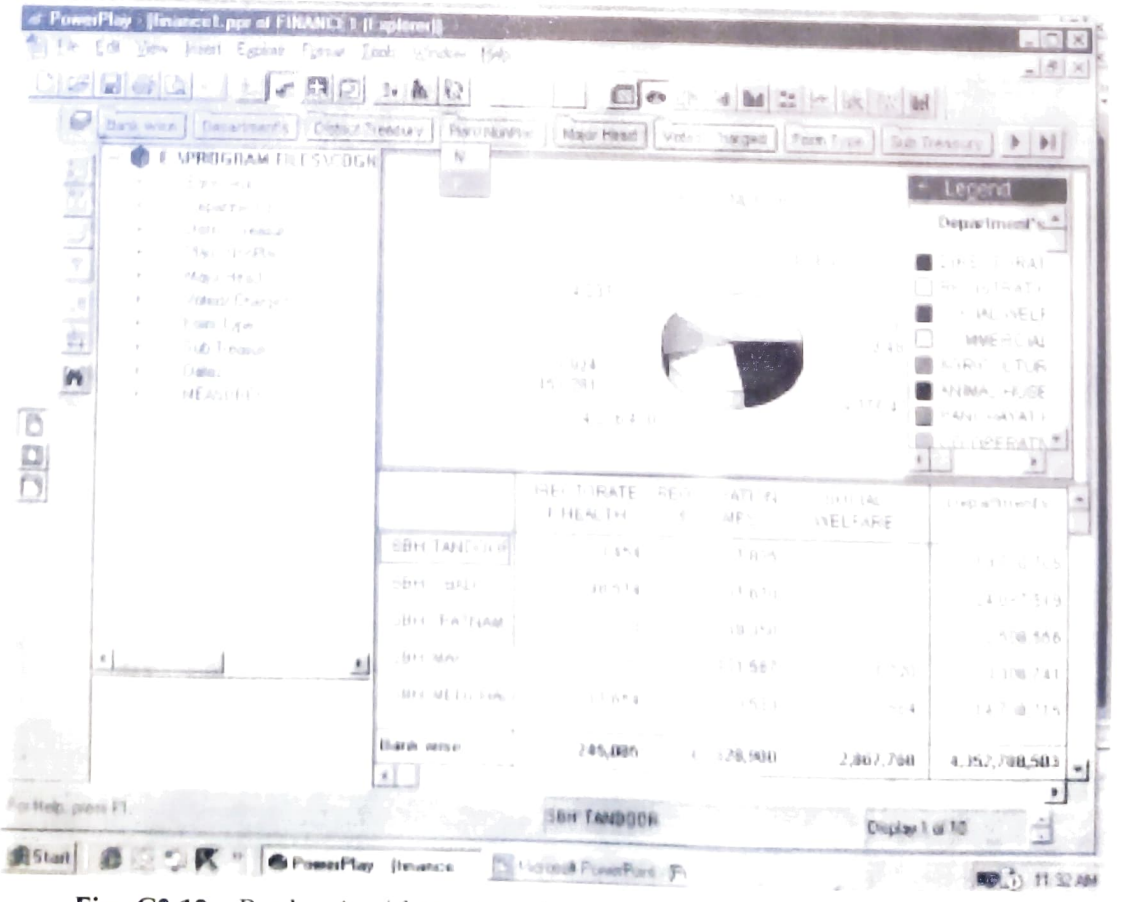


Fig. C3.12 Bank-wise/department-wise revenue under plan schedule.

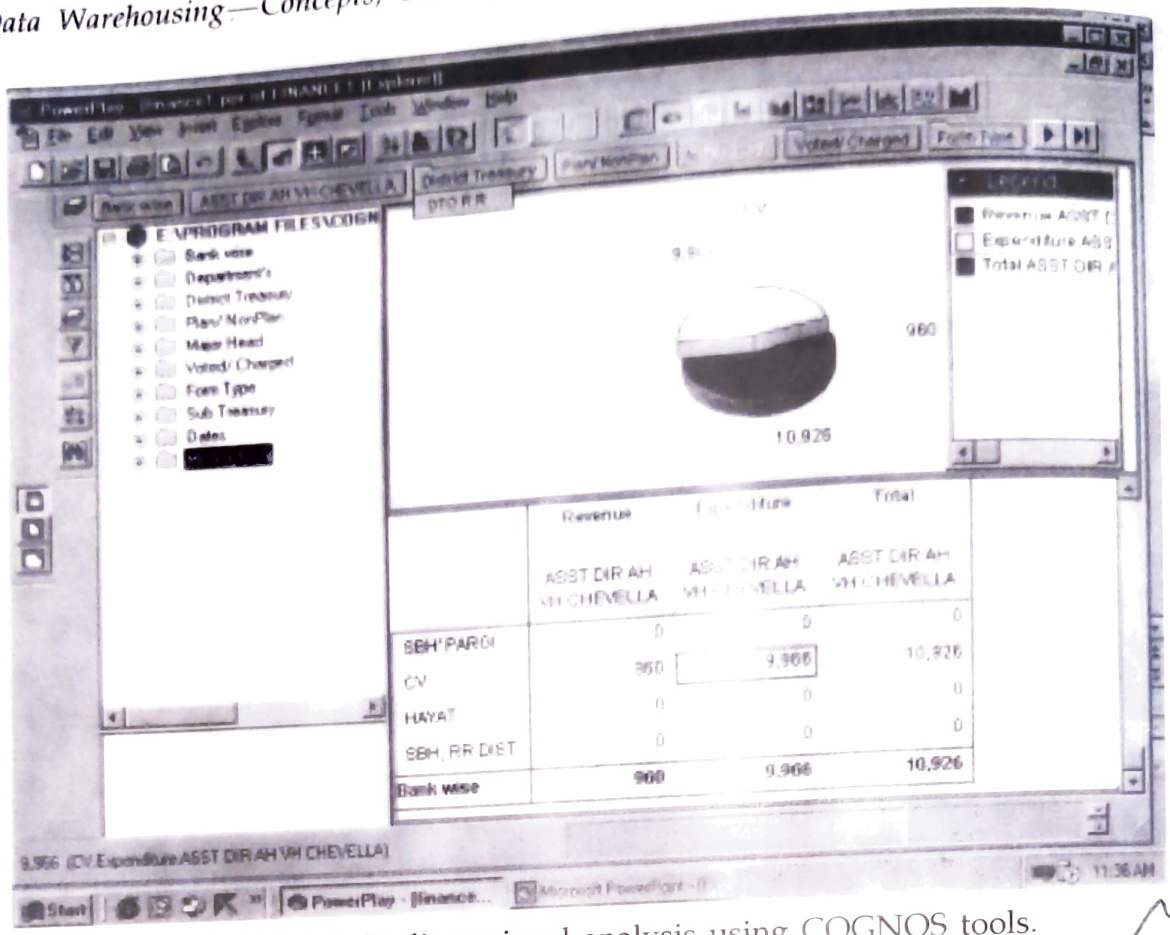


Fig. C3.13 Multi-dimensional analysis using COGNOS tools.

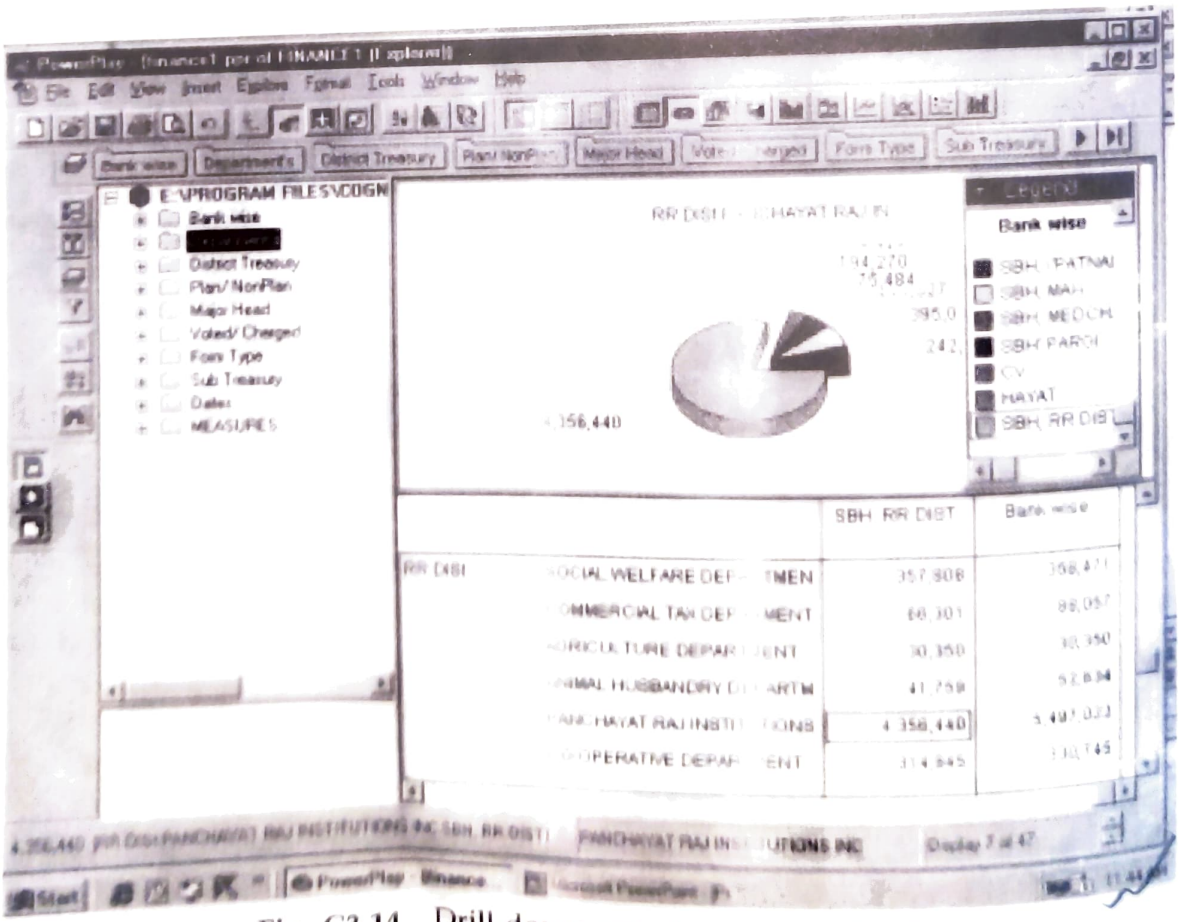
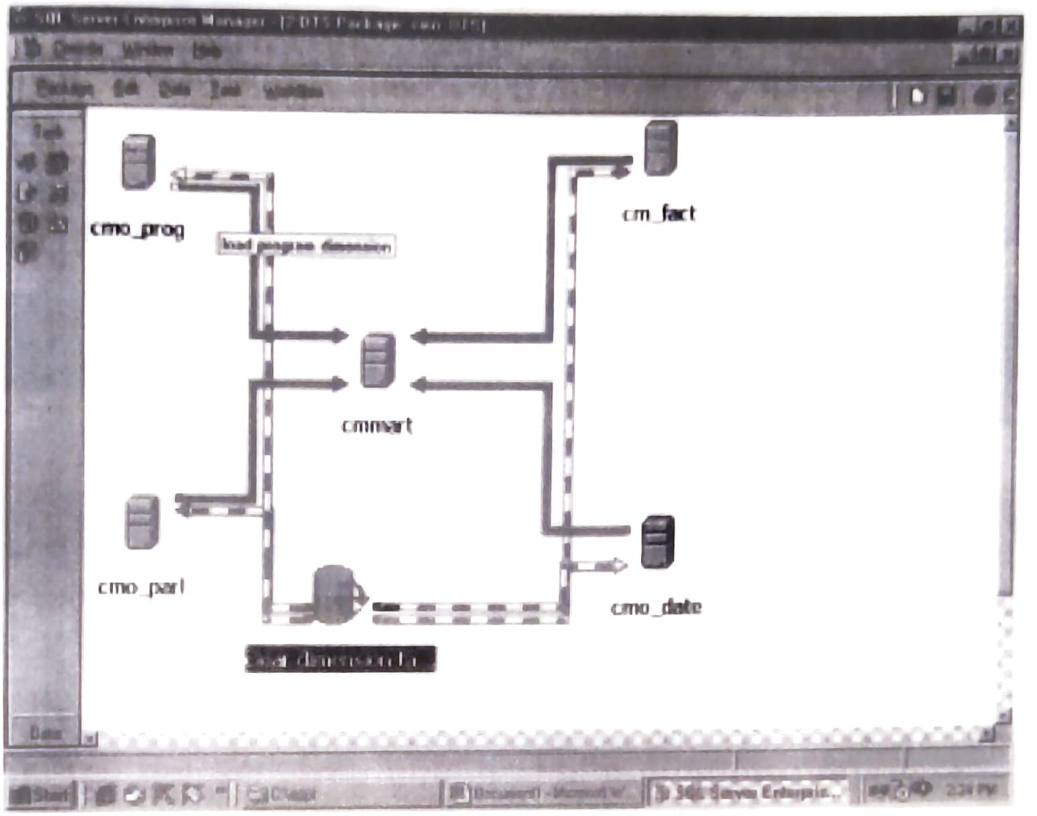


Fig. C3.14 Drill-down report, using COGNOS.

C3.2 A DATA WAREHOUSE FOR JANMABHOOMI WORKS MONITORING



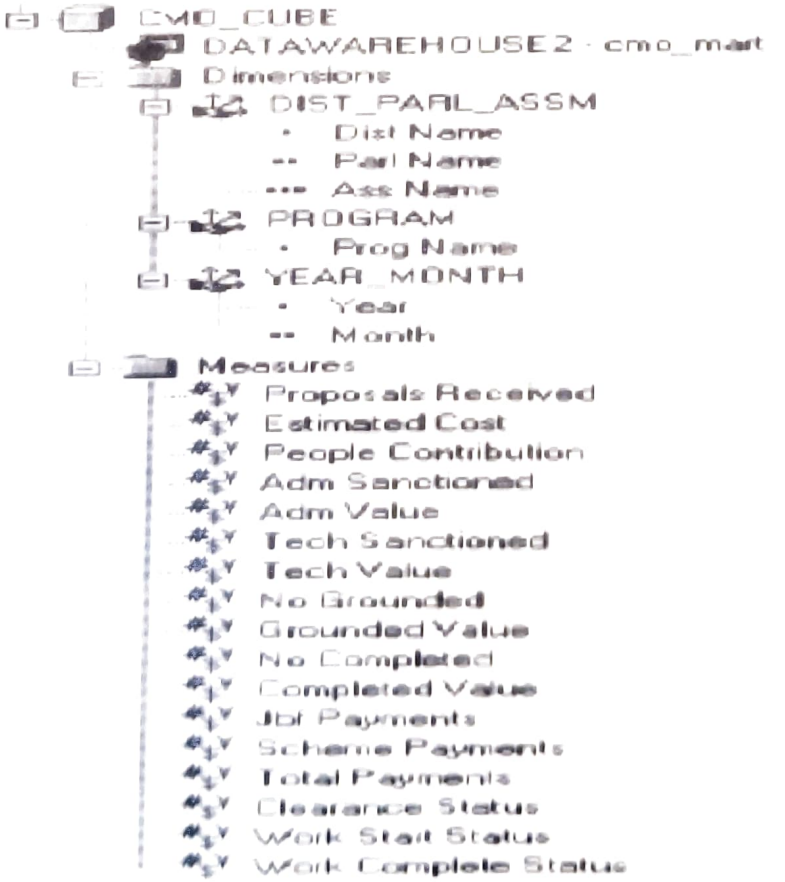


Fig. C3.15 Data transformation map for Janmabhoomi data.

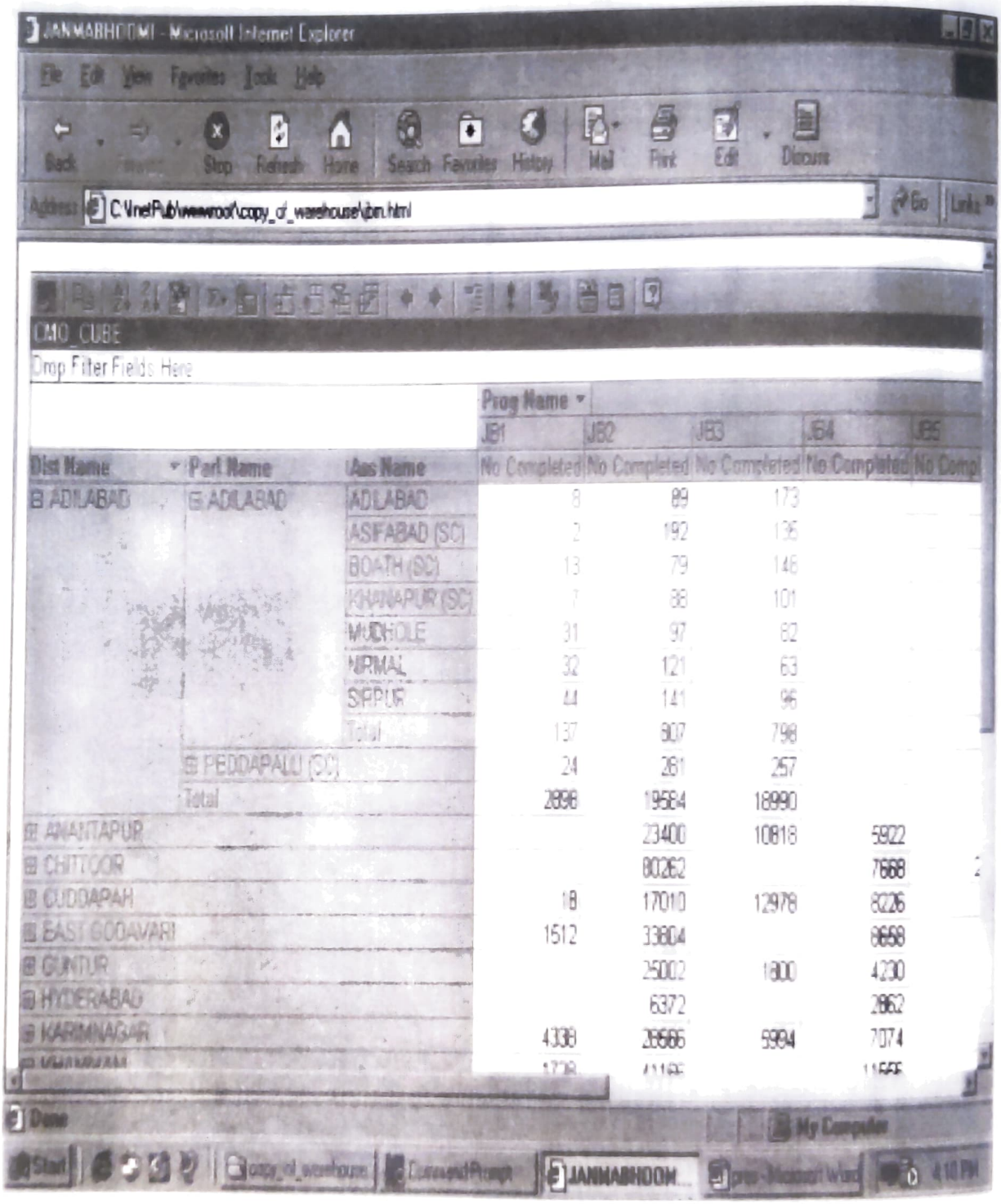
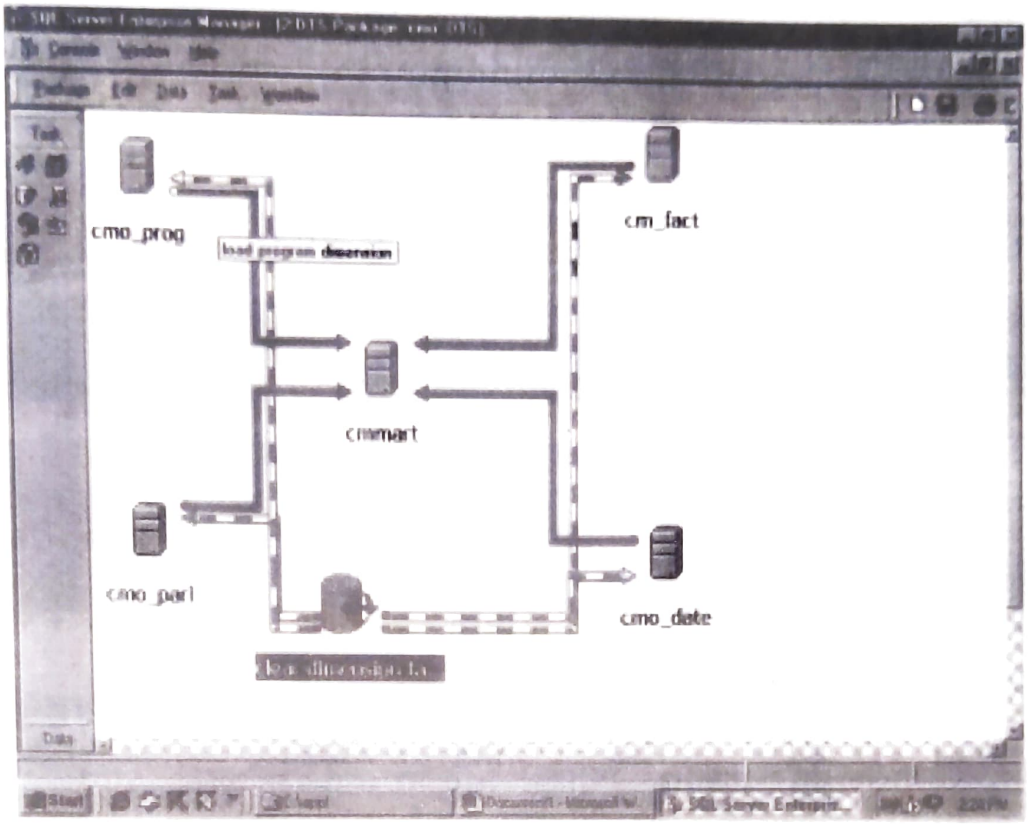


Fig. C3.16 Sample data drill-down picture for Janmabhoomi data.

C3.3 A DATA WAREHOUSE FOR CROPS MONITORING



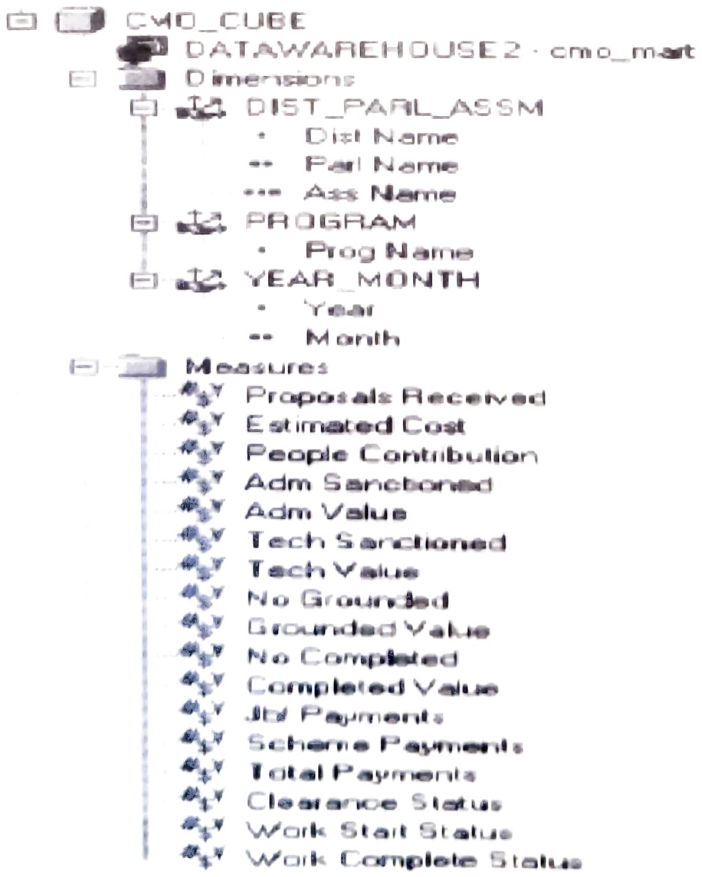


Fig. C3.17 Data transformation map for crop data.

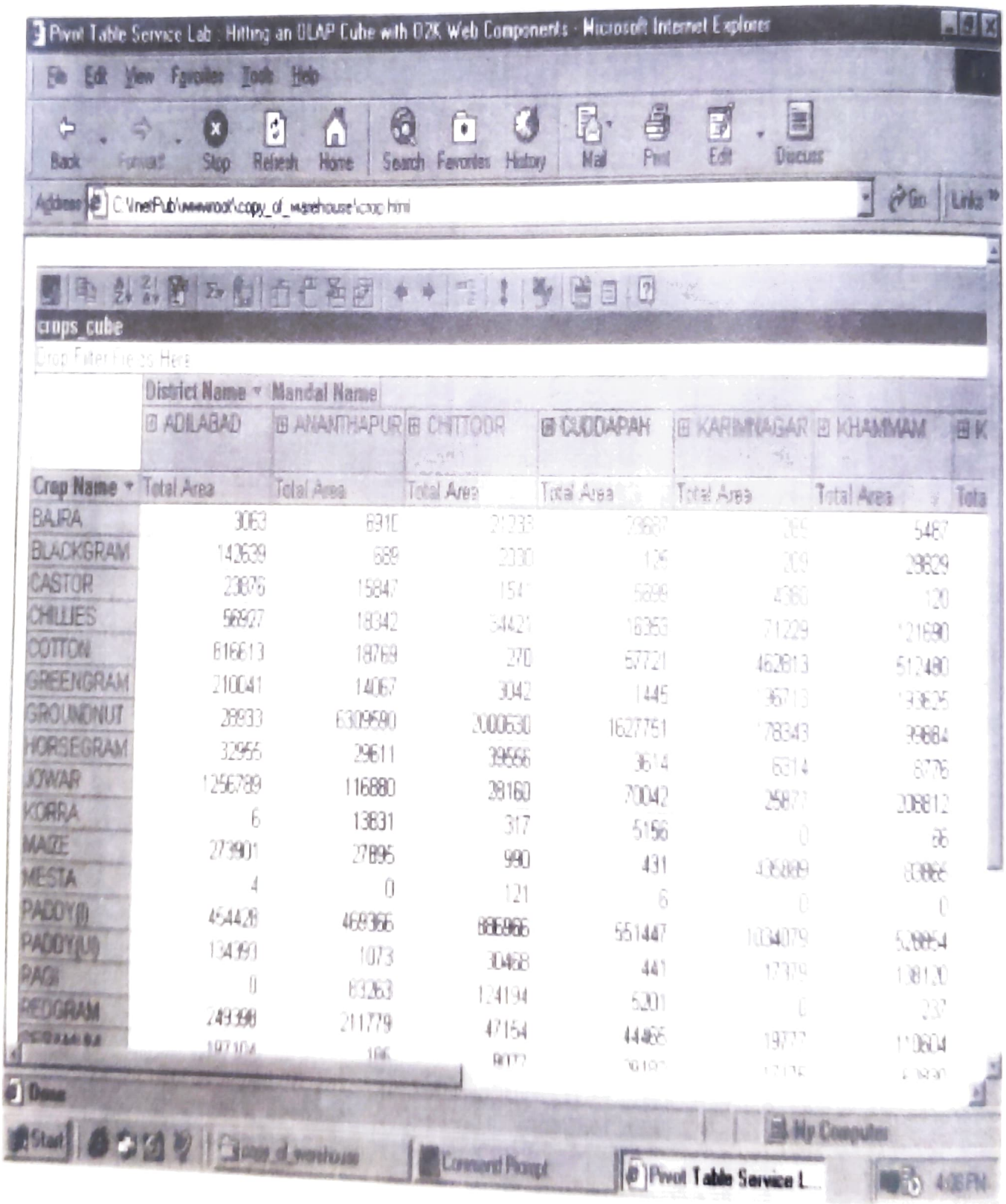


Fig. C3.18 Sample data drill down picture for crop data.

CASE STUDY

4 Data Warehousing in Hewlett-Packard*

With an OLAP system based on Microsoft® SQL Server™ 7.0 and Knosys ProClarity™, HP can easily access and quickly analyse enormous volumes of sell-through data to help its reseller customers improve the efficiency and profitability of their businesses.

C4.1 INTRODUCTION

With an installed base of more than 55 million printers, Hewlett-Packard is a worldwide market leader in the $18 billion inkjet industry. The company's Consumer Products Group develops and manufactures a full range of imaging products using HP-pioneered thermal-inkjet technology. Responsible for creating more new product categories for HP over the last 10 years than any other part of the company, the Consumer Products Group introduced the world's first desktop inkjet printer in 1984 and pioneered large-format inkjet printers in 1993, all-in-one (printer/fax/scanner/copier) devices in 1994, PC photography in 1997 and colour copiers in 1997.

In addition to its Home Business, Office Business, and Inkjet Supplies Business units, the Consumer Products Group includes HP's Consumer Products Business Organization (CPBO), responsible for worldwide retail sales and distribution of all of HP's consumer-targetted products. While the market for these products continues to grow, it is not experiencing the explosive, double-digit growth rates that characterized the past decade. With technical barriers to entry falling, more and more competitors are bringing products to market, giving each competitor a smaller piece of a shrinking pie.

With approximately 80 per cent of HP's consumer products volume sold directly to top national accounts such as CompUSA, Best Buy, and Office Depot, these resellers are as important a 'customer' as the end-user who plugs in the

*This case study is provided by Microsoft Corporation and therefore the views expressed in the case study belong to Microsoft Corporation.

printer. In the past, HP's brand recognition and reputation for reliability were enough to ensure that a reseller would carry HP products. Today, while those factors are still important, in an increasingly competitive landscape, they are simply not enough.

"To be successful in our business, we have to come up with solutions for the reseller", explains Greg Stanley, Manager of HP CPBO's Business Analysis Group, which is responsible for aggregating and analysing the market data—including information on advertisement spending, market share, pricing, demographics, econometrics, and channel spending—that CPBO uses to run its business. "Resellers have specific needs and wants just as individual users do. To maximize their profitability, they're focussing on the expense and inventory investment side of the profit equation. We have to share information that will help them maximize their efficiency and profitability. If we don't, the competition will walk in and do it for us".

C4.2 ACCESS TO INFORMATION NEEDED USING DATA WAREHOUSING TECHNOLOGY

As a technology company, HP has done a superb job of capturing and storing the information its resellers need, both from primary research and from third parties. But the repositories of information that exist in huge pipelines or reservoirs do not communicate with one another. "When it comes to connecting all the data in the reservoirs to the people who need to use it, they can not get to it", Stanley comments. "It gets so difficult to extract information from the system and put it in a meaningful context that a lot of our users just are not willing to do it".

The Business Analysis Group decided they needed a system that would provide market metric data to help field sales force managers or account teams make brand and channel management decisions. And, at a higher level, the group wanted a unified analytical environment that would allow HP decision-makers to clearly see the complex trends, patterns, and relationships that impact their business—including such data as how advertising affects sales and who is capturing market share and why.

Due to its smaller size and limited IT expertise, the group needed help evaluating potential solutions. "We wanted a system that required low cost, low maintenance, and as simple to administer as possible", Stanley explains. So the group turned to Knosys Inc., a Boise, Idaho-based software company that has developed a business analysis/online analytical processing (OLAP) package called ProClarity™. Having built its software from the ground up for the Microsoft SQL Server 7.0 data mart/data warehouse environment, Knosys recommended that HP adopt the SQL Server 7.0 relational database and ProClarity solution.

At first, Stanley's group was opposed to creating yet another data store. "But Knosys showed us that this solution would enable us to move the data so quickly and at such a low cost of maintenance and ownership that it would solve our problems", he says. "We would be duplicating data, but the new system would bridge the gap between users and data. This was very compelling". But the group also wanted to evaluate other potential solutions, so they brought in a leading OLAP consulting company called Symmetry Inc. With over a decade of OLAP consulting experience, Symmetry offered the product and industry knowledge to effectively evaluate the Knosys and Microsoft approach.

With the endorsement of symmetry, the Business Analysis Group decided to move ahead with the SQL Server 7.0 and ProClarity solution, which they are currently rolling out. Knosys has helped the group build the data flow algorithms with the Microsoft Visual Basic® development system and SQL Server 7.0's Data Transformation Services. Some of the original data was in Microsoft Excel spreadsheets, other flat files, and OLAP exports from other vendors. HP needed some common ground to establish the data flow procedures. Visual Basic for Applications (VBA) and Microsoft Access 97 provided that common ground, plus made the data easier to maintain. From Access, the data goes into SQL Server 7.0.

The OLAP Services include a number of additional features that are attractive to HP. According to Clay Young, Vice President of Marketing for Knosys Inc., the Business Analysis Group was particularly impressed with SQL Server 7.0's hybrid OLAP (HOLAP) capabilities.

"Because of HP's enormous sell-through data volumes, it would take too long to build analytical models with a pure, multidimensional OLAP solution", explains Young. "And pure relational OLAP solutions don't meet the query performance requirements of HP decision makers. The hybrid architecture of SQL Server 7.0's OLAP Services will enable HP to deal with high data volumes and still deliver fast query response".

HP used SQL Server 7.0's virtual cubes and cube partitioning capabilities. Cubes are databases with multiple dimensions: HP has at least eight OLAP cubes, each of which will support a particular group of decision-makers. Virtual cube capabilities also allow decision-makers to cross-analyse data from all these OLAP sources simultaneously. Cube partitioning will allow HP to more effectively manage a large number of OLAP cubes and to easily manage differing levels of aggregation and views of time. Additionally, the virtual cube capabilities allow HP to easily derive new business views from the existing cubes making it considerably easier for the Business Analysis Group to manage its business views. "Basically", Young remarks, "it will give them information about market share, econometrics, demographics, pricing, sell through, and channel activity—all in one view if they desire".

"Knosys ProClarity provides HP decision-makers with the key to analyzing masses of data", explains Young. "It gives them previously unavailable data

visualization techniques as well as easy to use, full Web-enabled analysis capabilities". ProClarity is fully integrated with Microsoft products, and its PC-based client is modelled after Internet Explorer 4.0. "This is because analysing business data is a lot like trying to find information on the Web", he adds.

ProClarity's powerful analytical features, which take full advantage of the robust capabilities found in SQL Server 7.0's OLAP Services, help knowledge workers understand complex data. The software's state-of-the-art data visualization tools enable workers to quickly see patterns, trends, and exception—even in complex environments such as HP's, where they must analyse hundreds of products across thousands of reseller partners.

Given HP's past experience with proprietary, monolithic solutions, Stanley's group wanted an open, highly flexible, analytical application that could be deployed in a variety of ways, including as a PC client, a Web-based client for HP's intranet, and a custom Executive Information System. ProClarity's ActiveX/COM architecture and OLE DB for OLAP connectivity fulfil this need.

CONCLUSION

When the new SQL Server 7.0 and ProClarity system is fully operational, Stanley (the manager of Business Analysis Group of Hewlett-Packard) expects it to provide significant benefits to account representatives in the field, business analysts in his and other groups, and HP's reseller customers: "I expect that account reps who are calling on a major account will log onto a Web page on Monday morning before going to call on that account and pull up last week's sales and inventory levels all the way down to the store level. They can produce a report that shows inventory problems in particular stores—for example, the new printer we just introduced three weeks ago is selling really well but it looks like we are having stock outs in particular stores. Then they can take our in-store audit data, which might show a problem with product placement in these stores. They can suggest that the problem may be that the product is displayed in the computer aisle, and people may not be shopping the computer aisle for a printer. Then, as analysts, we can study these trends over time and say, 'These stores are continually out of stock. These other stores are continually over stock and don't have as high a sell-through rate. Let's move inventory from these stores where it just sits on the shelves for two weeks to these stores where it sells out in half a week'".

"The bottom line", concludes Stanley, "is that this new system, through more accurate, detailed, and timely data, will make our business more efficient so we, in turn, can help our resellers make their businesses more efficient".

CASE STUDY

6 Data Warehousing in the World Bank

C6.1 INTRODUCTION

The World Bank collects and maintains huge data of economic and developmental parameters for all the third world countries across the globe. Originally the Bank performed analysis on this huge data manually and later with limited tools for analysis. Through the 'Live Database' and Country Economic Time Series, World Bank aimed to provide the economists world over with greater power of analysis of the data.

For the purpose of monitoring the effectiveness of the various World Bank assisted projects in the third world countries, the Bank started collecting and analysing macroeconomic financial statistics and also information on parameters such as poverty, health, education, environment and public sector. For more than one hundred developing countries, several thousands of economic indicators were being captured. It was a great challenge to refresh this data continuously (periodically) and make it accessible to various levels of Governments all over the world and also to economists, educators, academicians and any member of the general public. This broad accessibility was expected to contribute in many ways positively towards the growth of these developing economies.

To quote Ronnic Hammad, an economist at the World Bank, "Bringing the Live Databases to the clients will allow policy makers, civil society, researchers and others to have access to the latest economic information so as to influence decisions that will help in planning for better future. Such Live Databases in every Government, Ministry of Finance, every Central Bank and every Statistics Office, all made accessible will mean a lot for the economic planning".

C6.2 THE 'LIVE DATABASE' (LDB) DATA WAREHOUSE

In 1995 the 'Live Database' of the World Bank was developed by World Bank East Africa team. This was developed on SQL Server 2000 platform for the database. The OLAP cubes were defined for this database using OLAP server module of SQL Server 2000. Universal access was provided for this data

warehouse which was called 'Live Database'. This was so popular and so successful that the American Productivity Council called it the 'number one example of the transfer of best practice within an organization'. This product was given *meritorious award* and also resulted in much cost savings. The African region reported a saving of $12 million after the introduction to LDB data warehouse.

C6.3 BENEFITS OF THE SECOND-GENERATION LDB DATA WAREHOUSE

The first-generation LDB data warehouse had certain limitations—only a programmer could modify or add the data to it. Therefore, the second-generation LDB data warehouse was built using SQL Server 2000 Analysis Server along with 'Proclarity' (GUI tool) of Knosys Corporation. This package offered direct user functionality which was otherwise requiring technical intervention by a programmer. Proclarity also provided Web enablement, thereby ensuring universal accessibility. The highlight of this second-generation LDB data warehouse was that economists in the World Bank and all over the world were able to perform analysis using complex queries using Multi Dimensional extension (MDX) on OLAP cubes in SQL Server 2000 platform over the Web.

This exercise has resulted in significant cost savings by reducing the time and effort required to prepare a large variety of reports to suit varying needs of the economists and other governmental decision-makers to aid effective and better economic planning.

CASE STUDY

7 HARBOR, A Highly Available Data Warehouse

C7.1 INTRODUCTION

Today's data warehouses depend critically on various computational elements and networking elements. The computational elements are: processors, disk units, etc. The networking elements include switches, hubs, modems and lines. If any of these elements fail then it will result in the failure of the data warehousing and OLAP services offered. In other words a highly available data warehouse is needed to ensure a high-level user satisfaction of the services.

Any highly available database system or data warehousing system will use data replication to ensure that the data access continues with no or very few interruptions, the latter may arise if some computational system or component fails. Each database object, e.g. a tuple, a partition or a table, will have to be replicated at least once and distributed among many sites. Normally, the approaches for high availability include identical sites, identical replicas and identical ways of storing the replicas and identical mechanisms for distribution. The system under consideration for a highly available data warehouse, called HARBOR (Highly Available Replication-Based Online Recovery) is more flexible and does not insist on all these requirements of identical copies, as long as they represent logically the same data. With this flexibility, it is possible to store data redundantly in different sort orders in various data compression formats. The different updatable materialized views are also possible so that a wider variety of queries can be answered by the query optimizer and query processor (than is possible with the uniform formats of copies stored). A system called **C-Store** achieves an order of magnitude higher performance than is usual by storing the data in different sort orders and different compression formats. Data redundancy can, therefore, provide higher performance and also higher availability in HARBOR.

Normally, data warehouses cater specifically to large ad hoc query workloads over large read data sets intermingled with a small number of OLTP transactions which may touch a few records only, but affect the most recent data. Under such conditions the conventional locking techniques cause poor performance due to

considerable lock contention. A better solution is to use 'snapshot isolation' in which the read-only transactions read data without locks and updated transactions modify a copy of the data. In this the conflicts are resolved by an optimistic concurrency protocol with locks. In the case of HARBOR, a time travel mechanism is used, similar to snapshot isolation that involves explicit versioned and time stamped representation of data to isolate read-only transactions. Historical queries are as of some earlier T read time slices of the data that are guaranteed to remain unaffected by the subsequent transactions and, hence, proceed without acquiring read locks. Updated transactions and read-only transactions that wish to read the most up-to-date data can use the conventional read and write locks for isolation (as in two-phase locking).

C7.2 FAULT/FAILURE TOLERANCE

Any highly available database system will deploy some form of replication for fault or failure tolerance. A fault/failure tolerant system is defined to provide K-safety if upto K-sites fail and the system will still continue to provide services for any query or transaction. The minimum number of sites required for K-safety is K+1, where the K+1 workers store the same replicated data. In the approach taken in HARBOR it is assumed that the database designer has replicated the data and structured the database in such a way as to provide K-safety. The high availability in HARBOR guarantees that K simultaneous failures can be tolerated and still bring the failed sites online. If more than K-sites fail simultaneously, then this approach will not be applicable and the recovery can be achieved by other time tested solutions such as restoring from archival copy or rolling back changes to restore some globally consistent state. However, the database design can choose an appropriate value for K to reduce the probability of K simultaneous failures down to some acceptable value (for a specific application), as this recovery approach can be applied to bring sites online as they fail.

The approach assumes reliable network transfers via TCP/IP. It also assumes 'fail stop failures' (this means that when a failure occurs it is assumed to be a complete stop and not partial or incompletely written disk pages and network particulars) and does not deal with corrupted data, incompletely written disk pages and network partitions.

C7.3 HISTORICAL QUERY PROCESSING

In HARBOR, the historical queries are processed in a unique manner. By definition, a historical query, as of some past time, is a read-only query that returns a result, as if the query had been executed on the database at time T, i.e. a historical query at time T sees not committed or uncommitted updates after time T. This time travel feature enables the inspection of past states of the database.

Such historical queries are supported in HARBOR using 'versioned' representation of data in which time stamps are associated with each tuple. 'Insertion time' and 'deletion time' are added on to the tuple as insertion time, deletion time, a_1, a_2, a_3 ... a_N instead of the usual tuple a_1, a_2, ..., a_N.

The time stamp values are assigned at commit time as part of the commit protocol. Upon insertion, the insertion time stamp is inserted with a '0' value for deletion time stamp and vice versa upon deletion. If there is an update transaction which updates the tuple rather than updating the tuple in place, such an update is represented as a deletion of the old tuple and an insertion of the new tuple.

On doing this, the information necessary to answer historical queries is preserved. To answer a historical query on a tuple at time 'T', it has to be confirmed whether the tuple was inserted at or before time T and deleted T, or after T.

As historical queries view an old time slice of the database and all subsequent update transactions use later time stamps hence no (current) update transactions can ever affect the output of a historical query for some past time. Moreover, no locks are required for historical queries. The amount of history maintained in a system can be confirmed by the user by deleting the tuples before a specific time.

C7.4 RECOVERY ALGORITHM

The algorithm for recovery consists of three phases and uses time stamps associated with tuples to answer time-based range queries for the tuples inserted or deleted during a specified time range. In the first phase, a checkpointing mechanism is used to determine the most recent time T such that it is guaranteed that all the updates upto and including time T have been flushed to the disk. The crashed site then uses the time stamps available for historical queries to run local update transactions to restore itself to the time of its last checkpoint. In order to record checkpoints, the buffer pool will have to maintain a *duty pages table* with the identity of all *in memory* pages and a flag for each page indicating whether it contains any changes not yet flushed to the disk. During the normal processing, the database periodically writes a checkpoint for some past time T by first taking a snapshot of the dirty pages table at time T+1. For each page that is dirty in the snapshot, the system obtains a write latch for the page, flushes the page to the disk, and then releases the latch. After all the dirty pages are flushed, T is recorded onto the same well-known location on the disk and the checkpoint is completed. The checkpoint will be 0 on initial condition or when it got corrupted.

In the second phase, the site executes historical queries on other live sites that contain replicated copies of its data in order to catch up with the changes made between the last checkpoint and sometime closer to the present. It is to be noted

as a significant feature of the algorithm that read locks are not required to run historical queries—this ensures that the system is not quitened or slowed down while large amounts of data are copied over the network—otherwise this approach will not be viable (if read locks are required to be obtained for recovery query).

In the third (or final) phase, the site executes the standard non-historical queries with read locks to catch up with any committed changes between the start of the second phase and the current time. The coordinator then forwards any relevant update requests of ongoing transaction to the site to enable the crashed site to join any pending transaction and come online—this is possible only if the coordinator maintains a queue of update reports for each ongoing transaction.

C7.5 PERFORMANCE

Performance evaluation had shown that the recovery approach described in the foregoing section works well for data warehouses and similar environments, where the updated work loads consist primarily of insertions with relatively few updates to historical data.

CONCLUSION

In this case study we have presented an advanced and effective methodology for ensuring high availability of a data warehouse, HARBOR. Similar techniques can be used for ensuring high availability for any data warehouse.

CASE STUDY

8 A Typical Business Data Warehouse for a Trading Company

Let us consider a typical trading company. This hypothetical company sells products around the world and records data into its sample database that is created on a MS SQL Server 2000. The organization's businessowners would like to have analytical views, including graphs and charts that display the company's performance in terms of customer, employee, supplier, and product. Having such a tool would help the stakeholders promote the products that fall short of being sold in hot trading areas.

The example considered in this case study is very simple, as it is hypothetical and therefore, quite easy to build. The reasons are as follows:

First, we have a set of reports in mind to be supported by the warehouse. Usually, business owners know they need a data warehouse but are unable to give a concrete idea of the warehouse. It might take at least a few interviews to figure out and identify exactly what type of reports and analytical views the users would like to see. But in our present case study we know very clearly what type of reports are required.

As the data is already in a MS SQL Server database, which has a fairly simple structure, the first few steps of a typical warehouse project are already ready for us. In reality, however we may not always be so lucky: The data warehouse architect usually has to identify the multiple data sources that will be used to populate the warehouse. The organization's data could be stored in various relational database management systems (Oracle, MS SQL Server, DB2, and MS Access being the most common), spreadsheets, e-mail systems, and even in paper format. Once we identify all the data sources, we need to create data extraction routines using ETL (Extract Transform and Load) or DTS (Data Transformation Services) as the case may be, to transfer the data from its source to a SQL Server database. Furthermore, depending on the sources, we may not be in a position to manipulate the data until it is in MS SQL Server format. The data undergoes cleansing and validation as defined by the user in the ETL/DTS software (whichever may be used).

The main database has very clear object names; for example, the Orders table contains information on customer orders, Employees table has data about the

employees, and Order Details table has details of each order. Again, in the real world this might not always be the case—so simple and straightforward. We may have to figure out what some of the cryptic object names mean and exactly which data elements we require. The data warehouse architect often needs to create a list of data mappings and clean the data as it is loaded into the warehouse. For example, the customer names might not always be stored in the same format in the various data sources.

Once we have decided which data we need, we can create and populate a staging database and then the dimensional data model. Depending on the project, we may or may not have to have a staging database. If we have multiple data sources and need to correlate the data from these sources prior to populating a dimensional data structure, then having a staging database is convenient. Furthermore, a staging database will be a handy tool for testing. We can compare a number of records in the original data source with the number of records in the staging tables to ensure that the ETL routines work correctly. In this case the database already has all the data we need in easily accessible format; therefore, we need not create a staging database.

Now let us perform the dimensional modelling for this data warehouse.

Dimensional modelling is different from its relational counterpart, the relational database modelling. The dimensional models consist of fact tables and dimension tables. The typical fact tables contain numerous foreign keys referencing the dimension tables. The dimension tables, on the other hand, usually contain very few columns—dimension key, value, create, and update date, and perhaps an obsolete date. The fact tables record occurrences of a measurable fact, such as customer orders. The dimension tables provide a way to slice business data across various diagonals of company's operations; for example, we can examine orders by customer or by product.

We can use the `obsolete_date` column within the dimension tables to track the history of the values that change over time. This concept is known as *slowly changing dimension*. For example, consumers of the products may change their last names for any personal reason. Similarly, multiple departments within the organization can be combined into one, or one department can be divided into many. In some cases, we may keep only the current value. If so, we can simply override the existing value with the new value in the dimension table. In other cases, we must keep track of the old value, and the new value. This is when we use the record obsolete date to track the timeframe during which the record was valid.

In the case of our present data warehouse, the dimensional model will be very simple, consisting of the following dimension tables and one fact table. Since this is just a static database and there is no new data to populate it regularly, we need not add the `obsolete_date` column to the dimensions. We can create fact and dimension tables using the following script (in MS SQL Server):

```
CREATE TABLE dbo.dim_supplier(
    supplier_ident INT IDENTITY(1, 1),
    supplier_id INT NOT NULL,
    supplier_name VARCHAR(255) NOT NULL,
    supplier_city VARCHAR (255) NULL,
    country VARCHAR(255) NULL
)
)

CREATE TABLE dbo.dim_product (
    product_ident INT IDENTITY(1, 1),
    product_id INT NOT NULL,
    product_name VARCHAR(255) NOT NULL,
    discontinued BIT NOT NULL
)

CREATE TABLE dbo.dim_customer (
    customer_ident INT IDENTITY(1, 1),
    customer_id VARCHAR(20) NOT NULL,
    customer_name VARCHAR(255) NOT NULL,
    customer_city VARCHAR(255) NULL,
    customer_country VARCHAR(255) NULL
)

CREATE TABLE dbo.dim_employee (
    employee_ident INT IDENTITY(1, 1),
    employee_id INT NOT NULL,
    employee_name VARCHAR(85) NOT NULL,
    employee_city VARCHAR(255) NULL,
    employee_country VARCHAR(255) NULL
)

CREATE TABLE dbo.dim_time (
    time_member_key INT NOT NULL ,
    calendar_date_dt DATETIME NOT NULL ,
    calendar_day_of_week_num INT NOT NULL ,
    calendar_day_of_week_name VARCHAR(15) NOT NULL ,
    calendar_day_of_month_num INT NOT NULL ,
    calendar_day_of_year_num INT NOT NULL ,
    calendar_week_num INT NOT NULL ,
    calendar_month_num INT NOT NULL ,
    calendar_month_name VARCHAR (15) NOT NULL ,
    calendar_quarter_num INT NOT NULL ,
    calendar_year_num INT NOT NULL
)

CREATE TABLE fact_sales (
    customer_ident INT NOT NULL,
    product_ident INT NOT NULL,
    employee_ident INT NOT NULL,
    supplier_ident INT NOT NULL,
```

```
    total_sale SMALLMONEY NOT NULL,
    time_member_key INT NOT NULL
)
```

Next let us populate these tables using the following queries:

```
-- supplier dimension:
INSERT dim_supplier (
    supplier_id ,
    supplier_name ,
    supplier_city ,
    country )
SELECT
    supplierid,
    companyname,
    city,
    country
FROM suppliers

-- product dimension:
INSERT dim_product (
    product_id,
    product_name,
    discontinued)
SELECT
    productid,
    productname,
    discontinued
FROM products

-- customer dimension:
INSERT dim_customer (
    customer_id,
    customer_name,
    customer_city,
    customer_country)
SELECT
    customerid,
    companyname,
    city,
    country
FROM customers

-- employee dimension:
INSERT dim_employee (
    employee_id,
    employee_name,
    employee_city,
    employee_country)
SELECT
    employeeid,
```

```
    TItleOfCourtesy + ' ' + FirstName + ' ' + LastName AS employee_name,
    city,
    country
FROM employees
```

Note that dim_time is a special dimension. It is not populated by data that is already in the warehouse. Instead we populate it with calendar dates and date parts (day, month, quarter, year, and so forth) so that we can aggregate the warehouse data as needed. You can come up with a routine that populates your own time dimension; here is a sample store procedure that one can use to populate the time dimension:

```
CREATE PROCEDURE load_dim_time (
    @dim_table_name VARCHAR(255),
    @start_date_dt SMALLDATETIME,
    @end_date_dt SMALLDATETIME
)
AS
SET NOCOUNT ON
DECLARE
    @sql_string NVARCHAR(1024)
    , @time_member_key INT
    , @calendar_date_dt SMALLDATETIME
    , @calendar_day_of_week_num INT
    , @calendar_day_of_week_name VARCHAR(10)
    , @calendar_day_of_month_num INT
    , @calendar_day_of_year_num INT
    , @calendar_week_num INT
    , @calendar_month_num INT
    , @calendar_month_name VARCHAR(10)
    , @calendar_quarter_num INT
    , @calendar_year_num INT

SET @calendar_date_dt = @start_date_dt
WHILE (@calendar_date_dt <= @end_date_dt)
    BEGIN
    IF NOT EXISTS
    (
        SELECT time_member_key
        FROM dim_time
        WHERE calendar_date_dt = @calendar_date_dt
    )
BEGIN
    SELECT
    @calendar_day_of_week_num = DATEPART(dw, @calendar_date_dt)
    , @calendar_day_of_week_name = DATENAME(WEEKDAY, @calendar_date_dt)
    , @calendar_day_of_month_num = DATEPART(DD, @calendar_date_dt)
    , @calendar_day_of_year_num = DATEPART(DY, @calendar_date_dt)
    , @calendar_week_num = DATEPART(WK, @calendar_date_dt)
```

```
    , @calendar_month_num = DATEPART(M, @calendar_date_dt)
    , @calendar_month_name = DATENAME(MONTH, @calendar_date_dt)
    , @calendar_quarter_num = DATEPART(QQ, @calendar_date_dt)
    , @calendar_year_num = DATEPART(YYYY, @calendar_date_dt)
    , @time_member_key =
    CAST(
    CAST(@calendar_year_num AS VARCHAR) +
    RIGHT('00' + CAST(@calendar_day_of_year_num AS VARCHAR), 3)
    AS INT)
    SELECT @sql_string =
    'INSERT INTO ' + @dim_table_name +
    ' (' +
    'time_member_key, ' +
    'calendar_date_dt, ' +
    'calendar_day_of_week_num,' +
    'calendar_day_of_week_name,' +
    'calendar_day_of_month_num,' +
    'calendar_day_of_year_num,' +
    'calendar_week_num,' +
    'calendar_month_num,' +
    'calendar_month_name,' +
    'calendar_quarter_num, ' +
    'calendar_year_num' +
    ') ' +
    'VALUES ' +
    '(' +
    CHAR(39) + CAST(@time_member_key AS VARCHAR) + CHAR(39) + ',' +
    CHAR(39) + CAST(@calendar_date_dt AS VARCHAR) + CHAR(39) + ',' +
    CAST(@calendar_day_of_week_num AS VARCHAR) + ',' +
    CHAR(39) + @calendar_day_of_week_name + CHAR(39) + ',' +
    CAST(@calendar_day_of_month_num AS VARCHAR) + ',' +
    CAST(@calendar_day_of_year_num AS VARCHAR) + ',' +
    CAST(@calendar_week_num AS VARCHAR) + ',' +
    CAST(@calendar_month_num AS VARCHAR) + ',' +
    CHAR(39) + @calendar_month_name + CHAR(39) + ',' +
    CAST(@calendar_quarter_num AS VARCHAR) + ',' +
    CAST(@calendar_year_num AS VARCHAR) + ')'

    EXEC sp_executesql @sql_string
END

    SET @calendar_date_dt = @calendar_date_dt + 1
END
    SET @calendar_date_dt = @calendar_date_dt + 1
END

/* now use load_dim_time procedure to populate dim_time table with needed dates
*/

EXEC load_dim_time dim_time, '1/1/96', '1/1/99'
```

MS SQL Server has an ETL tool—DTS—which can be used to execute and schedule the data warehouse population routines. A typical DTS package determines which data rows need to be extracted from their source and inserts such rows into the appropriate dimension and fact tables. As this is a sample application, there is no need to create any DTS packages, but in the real world ETL routines may become considerably complicated and might take several weeks or more to develop.

CONCLUSION

In this case study, we have introduced the steps involved in building a typical data warehouse. We showed how to model, create and populate a dimensional data model which will be a cornerstone of the warehouse and analytical reports, using MS SQL Server 2000 for a typical trading company.

Note: The above case study is derived from the open websites—we thankfully acknowledge the original author of the case on the web.

CASE STUDY

9 Customer Data Warehouse of the World's First and Largest Online Bank in the United Kingdom

Egg PLC provides banking, insurance, investments and mortgages to more than 3 million customers through its Internet site and other distribution channels. Established in 1998, Egg PLC pioneered online banking not only in the United Kingdom but also throughout the world. Its technical infrastructure and support was provided by Sun Microsystems.

C9.1 BACKGROUND AND MOTIVATION FOR A DATA WAREHOUSE

Egg PLC, with more than 2.5 million transactions per day, required highly scalable but reliable IT and Internet infrastructure, as it supported all its customer services through the Internet. This high traffic Internet banking application was supported by high-end Sun V880 servers running the Solaris operating system. Sun Java System Application Server software was the web server utilized. Egg PLC supported 85% of its total transactions on the Internet through its site www.egg.com. Until 2001 only online transactions were provided. Originally, Egg was able to obtain customer information (for serving it online) by outsourcing the data warehousing activity to a company by the name Experian. However, the delay or latency between outsourcing and the provision of data became a serious issue for Egg, and it was forced to build and maintain its own data warehouse using Oracle and SAS, in addition to the existing Sun software and hardware infrastructure. The goal of the data warehouse was to provide a single source of truth to Egg regarding the details of the customers. Thus, a new data warehouse was designed, built and tested.

C9.2 THE CUSTOMER DATA WAREHOUSE: ENVIRONMENT, DATA FLOW

The first version of the Customer Data Warehouse (CDW) of Egg was built on Sun Fire 6800 Server and later, as it was expanded and called for greater computing resources, on Sun Fire 15K.

Finally, Egg's data warehouse resided on Sun Fire 15K with 16 CPUs and 10 GB for the core system. All the incoming data feeds for the data warehouse were fed onto this system. The storage application is an EMC's SAN (Storage Area Network). The system works on Solaris 9 operating system, with Oracle 9i as the DBMS pending migration to Oracle 10g. Virtual domaining, a special feature supported by Solaris 10 is utilized by Egg's data warehouse. Oracle 10g supports RAC (Real Application Cluster) for better performance by load sharing along with failover facility within the cluster of 2 or more servers.

C9.3 USER INTERFACE

While the data comes out from Oracle, Egg's internal users (staff of the Bank) can use any technology of their choice for analysis, as per their requirements. They use SAS (of the SAS Institute) to extract, join and mine the data for applications such as credit decisions or campaign management. Comms Builder, a tool written in SAS, and other modules of SAS such as SAS BASE, SAS Connect, SAS Share, SAS SPDS and SAS Start are also used. Oracle's Discoverer is also utilized for data mining.

Egg's CDW is about 2 TB in size, with about 10 GB added each month. Customers also use this data warehouse. They can get not only the regular transactions (OLTP) on the banking applications but also can use data mining services along with OLAP and data warehousing services.

C9.4 SOURCES OF DATA

For the CDW, the data is sourced from a variety of input customer data channels, both internal and external. These include credit cards, loans and insurance services, etc. Therefore, if a customer has all the three, viz. credit card, loan and insurance with Egg, that customer is seen as one entity having three Egg products. Hence, by CDW it is possible to see all the possible ways in which the customer is actually engaged with Egg. This, in turn, will enable Egg to determine the propensity and potentiality of the customer to buy additional products and also make the right choice of such products for the customer.

C9.5 BENEFITS

Egg has availed great benefits out of CDW. Sales and Marketing campaigns, up to 120 per month, could be developed using CDW. It also made possible daily credit decisions with due daily risk assessment and risk management. For example, if a customer has a credit card, the Egg management is actually able to

make a decision, using the data in the data warehouses, on a transactional basis to determine whether that customer's account requires collection, is just slightly overdue, or conversely, that the customer is a good credit risk and should be offered a pre-approved loan.

Campaigns developed on the basis of CDW have resulted in greater sales.

C9.6 SECURITY AND VERSION MANAGEMENT

Egg has implemented many security controls to ensure that access to CDW conforms to the internal policies and also external regulatory requirements. The data warehousing team comprising DBAs, meta data analysts and Oracle developers are responsible for the data security. Only they can make changes to the code. The data warehouse team decides and implements the changes that are required from time to time. External Regulatory Acts such as the UK's Data Protection Act of 1998, Financial Services Authority within the UK, Banking Act in the UK provide the regulatory framework for which the CDW has to be made fully compliant.

C9.7 REFRESH POLICY AND DATA MARTS

The data in CDW is refreshed in real time. For example, when a customer applies for a credit card, by the time he gets his credit card, his details has already been entered in the CDW.

The data is cleansed, matched and used to populate the core data warehouse. Then, data marts are published from CDW for individual departments of Egg. The back-end credit decisioning data mart is refreshed everyday. The materialized views for financing are refreshed everyday for financial reporting and counting of customer numbers. The marketing data mart is refreshed three times every week.

However, not all data inflows are in real time. Some data refreshes may come in batch mode from the internal sources themselves. Service Level Agreements (SLAs) may be required, but often they may be delayed or late in sending the refreshes. This may result in delays, or latency problems which may affect the accuracy and updateness of data in the data warehouse.

C9.8 CUSTOMER'S BENEFITS

Without the CDW Egg customers may be severely handicapped. The CDW plays a strong and crucial role in providing information, regarding finances necessary for the customers. All the services of Egg critically depend on CDW and thus, the customers are critically dependent on it.

C9.9 RELIABILITY

The architecture and design of IT infrastructure is required to be reliable. Hot plugins are provided to cover any failure in hardware components. Multiple fail-safes are made available with SAN. A disaster recovery system is also provided with the help of the original Sun Fire 6800 server. 90% uptime is maintained with a scheduled 10% downtime. SUN systems are found to be reliable and never has had any failure that was beyond the scope of maintenance engineers on a 24/7 basis.

CONCLUSION

In this case study we have seen how a large online bank, Egg of the UK deploys data warehousing for their core functions using reliable IT infrastructure. We have also seen how such deployment has resulted in enhanced success in banking service delivery to the customers and was helped the internal staff in better planning of marketing decisions.

Note: This case study is derived from Egg's websites on its data warehouse. We thankfully acknowledge Egg for this.

CASE STUDY

10 A German Supermarket EDEKA's Data Warehouse

EDEKA Gmbh is a German wholesale and retail food chain. The top management of EDEKA felt a need to be able to forecast specific product demands by identifying emerging market trends—very crucial for its business growth.

EDEKA engaged Melsungen, a company based in Germany, to implement the data warehouse applications based on DB2 of IBM.

C10.1 OBJECTIVES OF THE DATA WAREHOUSE

In an environment of fierce competition with the local retail industry in Germany, EDEKA felt a data warehouse could give it an advantage by providing the ability to conduct analyses on information such as sales turnover and inventory levels. The super market wanted to acquire the ability to adapt more quickly to the changing business conditions and requirements, with 50% faster responses to the queries. It aimed at obtaining higher availability and richer marketing data, improved end-user productivity with more reliable application, reduced administration costs and be able to leverage the existing skill base.

C10.2 SOFTWARE ENVIRONMENT

EDEKA used IBM's DB2 Universal Database DBMS for iSeries and DB2 Connect softwares on IBM hardware environment.

C10.3 HARDWARE ENVIRONMENT

EDEKA used the following hardware environment for its data warehouse

1. IBM @ Server iSeries 850 (processor)
2. IBM TotalStorage® Enterprise Storage Server™ (disk storage)
3. IBM TotalStorage Enterprise Tape System 3590 (tape storage)

C10.4 BUSINESS GROWTH AND THE DATA WAREHOUSE

EDEKA has expanded its business at a rapid pace. In the last two decades, it set up 60 wholly owned super markets and 1000 retailers in Hessen and Thuringian regions of Germany. Today, the company has about 7000 employees. The initial data warehouse was valuable but it could not keep track with EDEKA's business growth. As the business grew, the number of users of the data warehouse also increased proportionately. With increased usage, the demand on the system to provide greater number of analyses went up. This made the system slow. A highly scalable hardware and software architecture alone could withstand this situation successfully.

By upgrading the data warehouse to DB2 for iSeries, EDEKA was able to leverage on scalability and performance. EDEKA gained a business intelligence infrastructure that delivered on a wide variety of valuable marketing information, faster and more efficiently. Now, analyses could be done for various business divisions of EDEKA, such as the central purchasing department, financial control department and logistics department—all these depended on the data from the sales at the wholesale level that were supplied from the data warehouse.

Moreover, analysts could assess the volume of returned goods, find the reasons for changing shopping patterns and subsequently propose corrective measures. In the end, the system resulted in improving the employee productivity, reduced operating costs and optimized the utilization of IT and business resources.

C10.5 PERFORMANCE

With the help of iSeries servers the transaction processing was made 50% faster. As the data warehouse was run on the IBM DB2 Connect, it contributed to a better system performance. 'Business Objects' product from IBM's business partner also helped in to better access the sales information and track, understand and manage the wealth of information stored in the data warehouse.

C10.6 UPDATES AND REFRESHING

The information in the data warehouse is updated within an hour the concerned sales activity is completed, ensuring that the data is fresh always. By using the timely business analysis, EDEKA is able to anticipate market patterns more accurately.

C10.7 FORECASTING

Based on latest data EDEKA performed the analysis were able and to anticipate market patterns and purchase patterns in the immediate and near future. This allowed them for example, to forecast demand growth and suggest corrective measures in the event of a sudden surge in returned goods. Therefore, the data warehouse helped the company in responding quickly to changing business conditions and competitive pressures and enabled it to differentiate and distinguish itself in the open market place from other super markets in the region.

C10.8 EXPANSION

EDEKA plans to include additional data storage capacity. In future, the data warehouse will play a key role in addressing EDEKA's ongoing efforts to understand better consumer shopping patterns. This will help its executives keep the right products in the stores at the right time, based on accurate and up-to-date sales data from the data warehouse.

C10.9 PERFORMANCE

On the whole, EDEKA feels immensely satisfied with the present data warehouse as it "empowers Edeka to make much more logical business decisions which in turn result in improving profits and business objectives significantly".

CONCLUSION

In this case study, we have examined how a German super market chain, EDEKA leverages the data warehousing technology with reliable hardware to enhance its business objectives and profits, while faring better against its competition.

Note: This case study is derived from EDEKA's web sites. We thankfully acknowledge the same.