Answers to Review Questions

1. What are decision support systems, and what role do they play in the business environment?

   [Homework Problem: Solution to be posted next week.]

2. Explain how the main components of a DSS interact to form a system.

   [Homework Problem: Solution to be posted next week.]

3. What are the most relevant differences between operational and decision support data?

   [Homework Problem: Solution to be posted next week.]

4. What is a data warehouse, and what are its main characteristics?

   [Homework Problem: Solution to be posted next week.]

5. Give three examples of problems likely to be found when operational data are integrated into the data warehouse.

   [Homework Problem: Solution to be posted next week.]

Use the following scenario to answer questions 6 through 13.

While working as a database analyst for a national sales organization, you are asked to be part of its data warehouse project team.

6. Prepare a high-level summary of the main requirements to evaluate DBMS products for data warehousing.

   There are four primary ways to evaluate a DBMS that is tailored to provide fast answers to complex queries:
   
   - the database schema supported by the DBMS
   - the availability and sophistication of data extraction and loading tools
   - the end user analytical interface
   - the database size requirements

   Establish the requirements based on the size of the database, the data sources, the necessary data transformations, and the end user query requirements. Determine what type of database is needed, i.e., a multidimensional or a relational database using the star schema. Other valid evaluation criteria include the cost of acquisition and available upgrades (if any), training, technical and development support, performance, ease of use, and maintenance.

7. Your data warehousing project group is arguing about prototyping a data warehouse before its implementation. The project group members are especially concerned about the need to acquire some data warehousing skills before implementing the enterprise-wide data warehouse. What would you recommend? Explain your recommendations.

   Knowing that data warehousing requires time, money, and considerable managerial effort, many companies create data marts, instead. Data marts use smaller, more manageable data sets that are targeted to fit the special needs of small groups within the organization. In other words, data marts are small, single-subject data warehouse subsets. Data mart development and use costs are lower and the implementation time is shorter. Once the data marts have demonstrated their ability to serve the DSS, they can be expanded to become data warehouses or they can be migrated into larger existing data warehouses.

8. Suppose you are selling the data warehouse idea to your users. How would you explain to them what multidimensional data analysis is and explain its advantages?

   Multidimensional data analysis refers to the processing of data in which data are viewed as part of a multidimensional structure, one in which data are related in many different ways. Business decision makers usually view data from a business perspective. That is, they tend to view business
data as they relate to other business data. For example, a business data analyst might investigate the relationship between sales and other business variables such as customers, time, product line, and location. The multidimensional view is much more representative of a business perspective. A good way to visualize the development and use of relationships is to examine data pivot tables in MS Excel.

9. The Data Warehousing project group has invited you to provide an OLAP overview before making a commitment. The group's members are particularly concerned about the OLAP client/server architecture requirements and how OLAP will fit the existing environment. Your job is to explain to them the main OLAP client/server components and architectures.

OLAP systems are based on client/server technology and they consist of these main modules:

- OLAP Graphical User Interface (GUI)
- OLAP Analytical Processing Logic
- OLAP Data Processing Logic

The location of each of these modules is a function of different client/server architectures. How and where the modules are placed depends on hardware, software, and professional judgment. Any placement decision has its own advantages or disadvantages. However, the following constraints must be met:

- The OLAP GUI is always placed in the end user's computer. The reason it is placed at the client side is simple: this is the main point of contact between the end user and the system. Specifically, it provides the interface through which the end user queries the data warehouse's contents.
- The OLAP Analytical Processing Logic (APL) module can be placed in the client (for speed) or in the server (for better administration and better throughput). The APL performs the complex transformations required for business data analysis, such as multiple dimensions, aggregation, and period comparison, and so on.
- The OLAP Data Processing Logic (DPL) maps the data analysis requests to the proper data objects in the Data Warehouse and is, therefore, generally placed at the server level.

10. One of your vendors recommends using an MDBMS. How would you explain this recommendation to your project leader?

Multidimensional On-Line Analytical Processing (MOLAP) provides OLAP functionality using multidimensional databases (MDBMS) to store and analyze multidimensional data. Multidimensional database systems (MDBMS) use special proprietary techniques to store data in matrix-like arrays of n-dimensions.

11. The project group is ready to make a final decision between ROLAP and MOLAP. What should be the basis for this decision? Why?

The basis for the decision should be the system and end user requirements. Both ROLAP and MOLAP will provide advanced data analysis tools to enable organizations to generate required information. The selection of one or the other depends on which set of tools will fit best within the company's existing expertise base, its technology and end user requirements, and its ability to perform the job at a given cost.

The proper OLAP/MOLAP selection criteria must include:

- purchase and installation price
- supported hardware and software
- compatibility with existing hardware, software, and DBMS
- available programming interfaces
- performance
- availability, extent, and type of administrative tools
- support for the database schema(s)
- ability to handle current and projected database size
- database architecture
- available resources
- flexibility
- scalability
- total cost of ownership.

12. The data warehouse project is in the design phase. Explain to your fellow designers how you would use a star schema in the design.

The star schema is a data modeling technique that is used to map multidimensional decision support data into a relational database. The reason for the star schema's development is that existing relational modeling techniques, E-R and normalization, did not yield a database structure that served the advanced data analysis requirements well. Star schemas yield an easily implemented model for multidimensional data analysis while still preserving the relational structures on which the operational database is built.

The basic star schema has two four components: facts, dimensions, attributes, and attribute hierarchies. The star schemas represent aggregated data for specific business activities. Using the schemas, we will create multiple aggregated data sources that will represent different aspects of business operations. For example, the aggregation may involve total sales by selected time periods, by products, by stores, and so on. Aggregated totals can be total product units, total sales values by products, etc.

13. Trace the evolution of DSS from its origins to today's advanced analytical tools. What major technologies influenced this evolution?

DSS development -- use the text's Table 13.6 -- can be traced along these lines:

Stage 1. The DSS are based, at least in general terms, on the reporting systems of the 1980's. These reporting systems required direct access to the operational data through a menu interface to yield predefined report structures.

Stage 2. DSS improved decision support by supplying lightly summarized data extracted from the operational database. These summarized data were usually stored in the RDBMS and were accessed through SQL statements via a query tool. At this stage, the DSS began to grow some ad hoc query capabilities.
14. What is OLAP, and what are its main characteristics?

OLAP stands for On-Line Analytical Processing and uses multidimensional data analysis techniques. OLAP yields an advanced data analysis environment that provides the framework for decision making, business modeling, and operations research activities. Its four main characteristics are:

1. Multidimensional data analysis techniques
2. Advanced database support
3. Easy to use end user interfaces
4. Support for client/server architecture.

15. Explain ROLAP, and give the reasons you would recommend its use in the relational database environment.

Relational On-Line Analytical Processing (ROLAP) provides OLAP functionality for relational databases. ROLAP's popularity is based on the fact that it uses familiar relational query tools to store and analyze multidimensional data. Because ROLAP is based on familiar relational technologies, it represents a natural extension to organizations that already use relational database management systems within their organizations.

16. Explain the use of facts, dimensions, and attributes in the star schema.

Facts are numeric measurements (values) that represent a specific business aspect or activity. For example, sales figures are numeric measurements that represent product and/or service sales. Facts commonly used in business data analysis are units, costs, prices, and revenues. Facts are normally stored in a fact table, which is the center of the star schema.

The fact table contains facts that are linked through their dimensions. Dimensions are qualifying characteristics that provide additional perspectives to a given fact. Dimensions are of interest to us, because business data are almost always viewed in relation to other data. For instance, sales may be compared by product from region to region, and from one time period to the next. The kind of problem typically addressed by DSS might be "make a comparison of the sales of product units of X by region for the first quarter from 1995 through 2005."

In this example, sales have product, location, and time dimensions.

Dimensions are normally stored in dimension tables. Each dimension table contains attributes. The attributes are often used to search, filter, or classify facts. Therefore, the data warehouse designer must define common business attributes that will be used by the data analyst to narrow down a search, group information, or describe dimensions. For example, we can identify some possible attributes for the product, location and time dimensions:

- Product dimension: product id, description, product type, manufacturer, etc.
- Location dimension: region, state, city, and store number.
- Time dimension: year, quarter, month, week, and date.

These product, location, and time dimensions add a business perspective to the sales facts. The data analyst can now associate the sales figures for a given product, in a given region, and at a given time. The star schema, through its facts and dimensions, can provide the data when they are needed and in the required format, without imposing the burden of additional and unnecessary data (such as order #, po #, status, etc.) that commonly exist in operational databases. In essence, dimensions are the magnifying glass through which we study the facts.

17. Explain multidimensional cubes and describe how the slice and dice technique fits into this model.

To explain the multidimensional cube concept, let's assume a sales fact table with three dimensions: product, location, and time. In this case, the multidimensional data model for the sales example is (conceptually) best represented by a three-dimensional cube. This cube represents the view of sales dimensioned by product, location, and time. The intersections of the slices yields smaller cubes, thereby producing the "dicing" of the multidimensional cube. By examining these smaller cubes within the multidimensional cube, we can produce very precise analyses of the variable components and interactions. In short, slice and dice refers to the process that allows us to subdivide a multidimensional cube. Such subdivisions permit a far more detailed analysis than would be possible with the conventional two-dimensional data view. The text's Figures 13.13 through 13.16 illustrate the slice and dice concept.

To gain the benefits of slice and dice, we must be able to identify each slice of the cube. Slice identification requires the use of the values of each attribute within a given dimension. For example, to slice the location dimension, we can use a STORE_ID attribute in order to focus on a given store.

18. In the star schema context, what are attribute hierarchies and aggregation levels and what is their purpose?

Attributes within dimensions can be ordered in an attribute hierarchy. The attribute hierarchy yields a top-down data organization that permits both aggregation and drill-down/roll-up data analysis. Use Figure Q13.18 to show how he attributes of the location dimension can be organized into a hierarchy that orders that location dimension by region, state, city, and store.

Figure Q13.18 A Location Attribute Hierarchy
The attribute hierarchy gives the data warehouse the ability to perform drill-down and roll-up data searches. For example, suppose a data analyst wants an answer to the query “How does the 2005 total monthly sales performance compare to the 2000 monthly sales performance?” Having performed the query, suppose that the data analyst spots a sharp total sales decline in March, 2005. Given this discovery, the data analyst may then decide to perform a drill-down procedure for the month of March to see how this year’s March sales by region stack up against last year’s. The drill-down results are then used to find out whether the low-over-all March sales were reflected in all regions or only in a particular region. This type of drill-down operation may even be extended until the data analyst is able to identify the individual store(s) that is (are) performing below the norm.

The attribute hierarchy allows the data warehouse and OLAP systems to use a carefully defined path that will govern how data are to be decomposed and aggregated for drill-down and roll-up operations. Of course, keep in mind that it is not necessary for all attributes to be part of an attribute hierarchy; some attributes exist just to provide narrative descriptions of the dimensions.

19. Discuss the most common performance improvement techniques used in star schemas.

The following four techniques are commonly used to optimize data warehouse design:

- **Normalization of dimensional tables**
  is done to achieve semantic simplicity and to facilitate end user navigation through the dimensions. For example, if the location dimension table contains transitive dependencies between region, state, and city, we can revise these relationships to the third normal form (3NF). By normalizing the dimension tables, we simplify the data filtering operations related to the dimensions.

- **Creating and maintaining multiple fact tables**
  We can also speed up query operations by creating and maintaining multiple fact tables related to each level of aggregation. For example, we may use region, state, and city in the location dimension. These aggregate tables are pre-computed at the data loading phase, rather than at run-time. The purpose of this technique is to save processor cycles at run-time, thereby speeding up data analysis. An end user query tool optimized for decision analysis will then properly access the summarized fact tables, instead of computing the values by accessing a “lower level of detail” fact table.

- **Denormalizing fact tables**
  Denormalization is done to improve data access performance and to save data storage space. The latter objective, storage space savings, is becoming less of a factor: Data storage costs are on a steeply declining path, decreasing almost daily. DBMS limitations that restrict database and table size limits, record size limits, and the maximum number of records in a single table, are far more critical than raw storage space costs.

Denormalization improves performance by storing in one single record what normally would take many records in different tables. For example, to compute the total sales for all products in all regions, we may have to access the region sales aggregates and summarize all the records in this table. If we have 300,000 product sales records, we wind up summarizing at least 300,000 rows. Although such summaries may not be a very taxing operation for a DBMS initially, a comparison of ten or twenty years’ worth of sales is likely to start bogging the system down. In such cases, it will be useful to have special aggregate tables, which are denormalized. For example a YEAR_TOTAL table may contain the following fields:

```
YEAR_ID, MONTH_1, MONTH_2,...,MONTH12, YEAR_TOTAL
```

Such a denormalized YEAR_TOTAL table structure works well to become the basis for year-to-year comparisons at the month level, the quarter level, or the year level. But keep in mind that design criteria such as frequency of use and performance requirements are evaluated against the possible overload placed on the DBMS to manage these denormalized relations.

- **Table partitioning and replication**
  Table partitioning will split a table into subsets of rows or columns. These subsets can then be placed in or near the client computer to improve data access times. Replication makes a copy of a table and places it in a different location for the same reasons.

20. Explain some of the most important issues in data warehouse implementation.

It is important to stress that, although the data warehouse data represent a snapshot of operational data, the data warehouse is a dynamic decision support framework that is always a work in progress. Because it is the foundation of a modern DSS, the design and implementation of the data warehouse requires the design and implementation of an infrastructure for company-wide decision support.

Quite clearly, the organization as a whole should benefit from the data warehouse portion of the decision support infrastructure. Designing a data warehouse means being given an opportunity to
• help develop an integrated data model ....
• capture the organization's data ....
• develop the information that is considered to be essential from both end user and business perspectives.

21. What is data mining, and how does it differ from traditional DSS tools?

Data mining describes a new breed of specialized decision support tools that automate data analysis. Data mining tools are based on algorithms that form the building blocks for artificial intelligence, neural networks, inductive rules, and predicate logic. Data mining differs from traditional DSS tools because it is proactive. That is, instead of having the end user define the problem, select the data, and select the tools to analyze such data, the data mining tools will automatically search the data for anomalies and possible relationships, thereby identifying problems that have not yet been identified by the end-user. In other words, data mining tools analyze the data, uncover problems or opportunities hidden in the data relationships, form computer models based on their findings, and then use the model to predict business behavior... without requiring end user intervention. Therefore, the end user is able to use the system's findings to gain knowledge that may yield competitive advantages. (See Section 13.7.)

22. How does data mining work? Discuss the different phases in the data mining process.

Data mining is subject to four phases:

- In the **data preparation phase**, the main data sets to be used by the data mining operation are identified and cleansed from any data impurities. Because the data in the data warehouse are already integrated and filtered, the Data Warehouse usually is the target set for data mining operations.
- The **data analysis and classification phase** objective is to study the data to identify common data characteristics or patterns. During this phase the data mining tool applies specific algorithms to find:
  - data groupings, classifications, clusters, or sequences.
  - data dependencies, links, or relationships.
  - data patterns, trends, and deviations.
- The **knowledge acquisition phase** uses the results of the data analysis and classification phase. During this phase, the data mining tool (with possible intervention by the end user) selects the appropriate modeling or knowledge acquisition algorithms. The most typical algorithms used in data mining are based on neural networks, decision trees, rules induction, genetic algorithms, classification and regression trees, memory-based reasoning, or nearest neighbor and data visualization. A data mining tool may use many of these algorithms in any combination to generate a computer model that reflects the behavior of the target data set.
- Although some data mining tools stop at the knowledge acquisition phase, others continue to the **prognosis phase**. In this phase, the data mining findings are used to predict future behavior and forecast business outcomes. Examples of data mining findings can be:
  - 65% of customers who did not use the credit card in six months are 88% likely to cancel their account
  - 82% of customers who bought a new TV 27" or bigger are 90% likely to buy a entertainment center within the next 4 weeks.
  - If age < 30 and income <= 25,000 and credit rating < 3 and credit amount > 25,000, the minimum term is 10 years.

The complete set of findings can be represented in a decision tree, a neural net, a forecasting model or a visual presentation interface which is then used to project future events or results. For example the prognosis phase may project the likely outcome of a new product roll-out or a new marketing promotion.

**Problem Solutions**

**ONLINE CONTENT**

The databases used for this problem set are found in the Student Online Companion for this book. These databases are stored in Microsoft Access 2000 format. The databases, named Ch13_P1.mdb, Ch13_P3.mdb, and Ch13_P4.mdb, contain the data for Problems 1, 3, and 4, respectively. The data for Problem 2 are stored in Microsoft Excel format on the Course Technology Web site for this book. The spreadsheet filename is Ch13_P2.xls.

1. The University Computer Lab's director keeps track of the lab usage, measured by the number of students using the lab. This particular function is very important for budgeting purposes. The computer lab director assigns you the task of developing a data warehouse in which to keep track of the lab usage statistics. The main requirements for this database are to:

- Show the total number of users by different time periods.
- Show usage numbers by time period, by major, and by student classification.
- Compare usage for different major and different semesters.

Use the Ch13_P1.mdb database, which includes the following tables:

- USELOG contains the student lab access data
- STUDENT is a dimension table containing student data

Given the three bulleted requirements and using the DW-P1.MDB data, complete Problems 1a–1g.

a. Define the main facts to be analyzed. (*Hint*: These facts become the source for the design of the fact table.)

b. Define and describe the possible dimensions. (*Hint*: These dimensions become the source for the design of the dimension tables.)

c. Draw the lab usage star schema, using the fact and dimension structures you defined in Problems 1a and 1b.

d. Define the attributes for each of the dimensions in Problem 1b.
e. Recommend the appropriate attribute hierarchies.

f. Implement your data warehouse design, using the star schema you created in problem 1c and the attributes you defined in Problem 1d.

g. Create the reports that will meet the requirements listed in this problem’s introduction.

Before problems 1 a-g can be answered, the students must create the time and semester dimensions. Looking at the data in the USELOG table, the students should be able to figure out that the data belong to the Fall 2005 and Spring 2006 semesters, so the semester dimension must contain entries for at least these two semesters. The time dimension can be defined in several different ways. It will be very useful to provide class time during which students can explore the different benefits derived from various ways to represent the time dimension. Regardless of what time dimension representation is selected, it is clear that the date and time entries in the USELOG must be transformed to meet the TIME and SEMESTER codes. For data analysis purposes, we suggest using the TIME and SEMESTER dimension table configurations shown in Tables P13.1A and P13.1B. (We have used these configurations in the DW-P1sol.MDB database that is located on the CD.)

**Table P13.1A The TIME Dimension Table Structure**

<table>
<thead>
<tr>
<th>TIME_ID</th>
<th>TIME_DESCRIPTION</th>
<th>BEGIN_TIME</th>
<th>END_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Morning</td>
<td>6:01AM</td>
<td>12:00PM</td>
</tr>
<tr>
<td>2</td>
<td>Afternoon</td>
<td>12:01PM</td>
<td>6:00PM</td>
</tr>
<tr>
<td>3</td>
<td>Night</td>
<td>6:01PM</td>
<td>6:00AM</td>
</tr>
</tbody>
</table>

**Table P13.1B The SEMESTER Dimension Table Structure**

<table>
<thead>
<tr>
<th>SEMESTER_ID</th>
<th>SEMESTER_DESCRIPTION</th>
<th>BEGIN_DATE</th>
<th>END_DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FA00</td>
<td>Fall 2005</td>
<td>15-Aug-2005</td>
<td>18-Dec-2005</td>
</tr>
<tr>
<td>SP01</td>
<td>Spring 2006</td>
<td>08-Jan-2006</td>
<td>15-May-2006</td>
</tr>
</tbody>
</table>

The USELOG table contains only the date and time of the access, rather than the semester or time IDs. The student must create the TIME and SEMESTER dimension tables and assign the proper TIME_ID and SEMESTER_ID keys to match the USELOG’s time and date. The students should also create the MAJOR dimension table, using the data already stored in the STUDENT table. Using Microsoft Access, we used the Make New Table query type to produce the MAJOR table. The Make New Table query lets you create a new table, MAJOR, using query output. In this case, the query must select all unique major codes and descriptions. The same technique can be used to create the student classification dimension table (In our solution, we have named the student classification dimension table CLASS.) Naturally, you can use some front-end tool other than Access, but we have found Access to be particularly effective in this environment.

To produce the solution we have stored in the PW-P1sol.MDB database, we have used the queries listed in Table P13.1C.

**Table P13.1C The Queries in the DW-P1sol.MDB Database**

<table>
<thead>
<tr>
<th>Query Name</th>
<th>Query Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update DATE format in USELOG</td>
<td>The DATE field in USELOG was originally given to us as a character field. This query converted the date text to a date field we can use for date comparisons.</td>
</tr>
<tr>
<td>Update STUDENT_ID format in STUDENT</td>
<td>This query changes the STUDENT_ID format to make it compatible with the format used in USELOG.</td>
</tr>
<tr>
<td>Update STUDENT_ID format in USELOG</td>
<td>This query changes the STUDENT_ID format to make it compatible with the format used in STUDENT.</td>
</tr>
<tr>
<td>Append TEST records from USELOG &amp; STUDENT</td>
<td>Creates a temporary storage table (TEST) used to make some data transformations previous the creation of the fact table. The TEST table contains the fields that will be used in the USEFACT table, plus other fields used for data transformation purposes.</td>
</tr>
<tr>
<td>Update TIME_ID and SEMESTER_ID in TEST</td>
<td>Before we create the USEFACT table, we must transform the dates and time to match the SEMESTER_ID and TIME_ID keys used in our SEMESTER and TIME dimension tables. This query does that.</td>
</tr>
<tr>
<td>Count STUDENTS sort by Fact Keys: SEM, MAJOR, CLASS, TIME.</td>
<td>This query does data aggregation over the data in TEST table. This query table will be used to create the new USEFACT table.</td>
</tr>
<tr>
<td>Populate USEFACT</td>
<td>This query uses the results of the previous query to populate our USEFACT table.</td>
</tr>
<tr>
<td>Compares usage by Semesters by Times</td>
<td>Used to generate Report1</td>
</tr>
<tr>
<td>Usage by Time, Major and Classification</td>
<td>Used to generate Report2</td>
</tr>
<tr>
<td>Usage by Major and Semester</td>
<td>Used to generate Report3</td>
</tr>
</tbody>
</table>

Having completed the preliminary work, we can now present the solutions to the seven problems:

**a. Define the main facts to be analyzed. (Hint: These facts become the source for the design of the fact table.)**

The main facts are the total number of students by time, the major, the semester, and the student classification.
b. Define and describe the possible dimensions. (*Hint: These dimensions become the source for the design of the dimension tables.*)

The possible dimensions are semester, major, classification, and time. Each of these dimensions provides an additional perspective to the total number of students fact table. The dimension table names and attributes are shown in the screen shot that illustrates the answer to problem 3.

c. Draw the lab usage star schema, using the fact and dimension structures you defined in Problems 1a and 1b.

Figure P13.1c shows the MS Access relational diagram — see the Ch13-P1sol.mdb database in the teacher's database folder for this chapter located on your instructor's CD -- to illustrate the star schema, the relationships, the table names, and the field names used in our solution. The students are given only the USELOG and STUDENT tables and they must produce the fact table and dimension tables.

Figure P13.1c The Microsoft Access Relational Diagram


d. Define the attributes for each of the dimensions in Problem (b).

Given problem 1c's star schema snapshot, the dimension attributes are easily defined:

**Semester dimension:** semester_id, semester_description, begin_date, and end_date.

**Major dimension:** major_code and major_name.

**Class dimension:** class_id, and class_description.

**Time dimension:** time_id, time_description, begin_time and end_time.

e. Recommend the appropriate attribute hierarchies.

See the answer to question 18 and the dimensions shown in Problems 1c and 1d to develop the appropriate attribute hierarchies.

**NOTE**

To create the dimension tables in MS Access, we had to modify the data. These modifications can be examined in the update queries stored in the Ch13-P1sol.mdb database. We used the switch function in MS Access to assign the proper SEMESTER_ID and the TIME_ID values to the USEFACT table.

f. Implement your data warehouse design, using the star schema you created in problem (c) and the attributes you defined in Problem (d).

The solution is included in the Ch13-P1sol.mdb database on the Instructor's CD.

g. Create the reports that will meet the requirements listed in this problem's introduction.

Use the Ch13-P1sol.mdb database on the Instructor's CD as the basis for the reports. Keep in mind that the Microsoft Access export function can be used to put the Access tables into a different database such as Oracle or DB2. Take a look at Appendixes 1 and 2 to see how you can move the Access database tables to a different database.
2. Ms. Victoria Ephanor manages a small product distribution company. Because the business is growing fast, Ms. Ephanor recognizes that it is time to manage the vast information pool to help guide the accelerating growth. Ms. Ephanor, who is familiar with spreadsheet software, currently employs a small sales force of four people. She asks you to develop a data warehouse application prototype that will enable her to study sales figures by year, region, salesperson, and product. (This prototype is to be used as the basis for a future data warehouse database.)

Using the data supplied in the Ch13-P2.xls file, complete the following seven problems:

- **a. Identify the appropriate fact table components.**

  The dimensions for this star schema are: Year, Region, Agent, and Product. (These are shown in Figure P13.2c.)

- **b. Identify the appropriate dimension tables.**

  (These are shown in Figure P13.2c.)

- **c. Draw a star schema diagram for this data warehouse.**

  See Figure P13.2c.

  **Figure P13.2C The Star Schema for the Ephanor Distribution Company**

- **d. Identify the attributes for the dimension tables that will be required to solve this problem.**

  The solution to this problem is presented in the Ch13-P2sol.xls file on the Instructor’s CD.

- **e. Using a Microsoft Excel spreadsheet (or any other spreadsheet capable of producing pivot tables), generate a pivot table to show the sales by product and by region. The end user must be able to specify the display of sales for any given year. (The sample output is shown in the first pivot table in Figure P13.2E.)**

  **FIGURE P13.2E Using a pivot table**
The solution to this problem is presented in the Ch13-P2sol.xls file on the Instructor's CD.

f. Using Problem 2e as your base, add a second pivot table (see Figure P13.2E) to show the sales by salesperson and by region. The end user must be able to specify sales for a given year or for all years and for a given product or for all products.

The solution to this problem is presented in the Ch13-P2sol.xls file on the Instructor's CD.

g. Create a 3-D bar graph to show sales by salesperson, by product, and by region. (See the sample output in Figure P13.2G.)

FIGURE P13.2G 3-D bar graph showing the relationships among agent, product, and region.
The solution to this problem is presented in the Ch13-P2sol.xls file on the Instructor's CD.

3. Mr. David Suker, the inventory manager for a marketing research company, is interested in studying the use of supplies within the different company departments. Mr. Suker has heard that his friend, Ms. Ephanor, has developed a small spreadsheet-based Data Warehouse model (see problem 2) that she uses in her analysis of sales data. Mr. Suker is interested in developing a small Data Warehouse model like Ms. Ephanor’s so he can analyze orders by department and by product. He will use Microsoft Access as the Data Warehouse DBMS and Microsoft Excel as the analysis tool.

NOTE

The solution to these problems is in the file named Ch13-P3sol.mdb. The solution file also contains all the queries necessary to derive the dimension tables and the main fact table from the orders data. You will also find an ORDTEMP table that is used to clean up the data and to perform necessary data validation and transformation routines before uploading the data to the ORDFACT table. The fact table contains monthly aggregates for total cost of orders by department, vendor and product. This is an arbitrary decision based on the end user needs; students might decide to use daily aggregates. In that case, proper TIME dimension codes must be generated and included in the TIME dimension table and in the ORDFACT tables.

a. Develop the order star schema.

Figure P13-3A's MS-Access relational diagram reflects the star schema and its relationships. Note that the students are given only the ORDERS table. The student must study the data set and make the queries necessary to create the dimension tables (TIME, DEPT, VENDOR and PRODUCT) and the ORDFACT fact table.

b. Identify the appropriate dimension attributes.

The dimensions are: TIME, DEPT, VENDOR, and PRODUCT. (See Figure P13.3A.)

c. Identify the attribute hierarchies required to support the model.

The main hierarchy used for data drilling purposes is represented by TIME-DEPT-VENDOR-PRODUCT sequence. (See Figure P13.3A.) Within this hierarchy, the user can analyze data at different aggregation levels.

Additional hierarchies can be constructed in the TIME dimension to account for quarters or, if necessary, by daily aggregates. The VENDOR dimension could also be expanded to include geographic information that could be used for drill-down purposes.

d. Develop a crosstab report (in Microsoft Access), using a 3-D bar graph to show sales by product and by department. (The sample output is shown in Figure P13.3.)

FIGURE P13.3 A Crosstab Report: Sales by Product and Department
Given those requirements, complete the following:

a. Create a star schema for the charter data.

**NOTE**

The students must first create the queries required to filter, integrate, and consolidate the data prior to their inclusion in the Data Warehouse. The Ch13-P4.mdb database contains the data to be used by the students. The Ch13-P4sol.mdb database contains the data and solution to the problems.

The problem requires the creation of the time dimension. Looking at the data in the CHARTER table, the students should figure out that the two attributes in the time dimension should be year and month. Another possible attribute could be day, but since no one pilot or airplane was used more than once a day, including it as an attribute would only reduce the database's efficiency. The analysis to be done on the time dimension can be done on a monthly or yearly basis.

The CHARTER table contains the date of the charter. No time IDs exist and the date is contained within a single field. The student must create the TIME dimension table and assign the proper TIME_ID keys and its attributes. A temporary table is created to aid in the creation of the CHARTER_FACT table. The queries in Table P13.4-1 are used in the transformation process:

**Table P13.4-1 The ROBCOR Data Warehouse Queries**

<table>
<thead>
<tr>
<th>Query Name</th>
<th>Query Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make a TEMP table from CHARTER, PILOT, and MODEL</td>
<td>Creates a temporary storage table used to make the necessary data transformations before the creation of the fact table.</td>
</tr>
<tr>
<td>Update TIME_ID in TEMP</td>
<td>Used to create the TIME_ID key used in the TIME dimension table.</td>
</tr>
<tr>
<td>Update YEAR and MONTH in TEMP</td>
<td>In order to get the year and month attributes in the TIME dimension it is necessary to separate that data in the temporary table first. The date is in the TEMP table but will not be in the fact table.</td>
</tr>
<tr>
<td>Make TIME table from TEMP</td>
<td>This query is used to create the time table using the appropriate data from the TEMP table.</td>
</tr>
</tbody>
</table>
Aggregate TEMP table by fact keys

This query does data aggregation over the data in the TEMP table. This query table will be used to create the new CHARTER_FACT table.

Populate CHARTER_FACT table

This query uses the results of the previous query to populate our CHARTER_FACT table.

The MS Access relational diagram in Figure P12-4a reflects the star schema, the relationships, the table names, and field names used in our solution. The student is given only the CHARTER, AIRCRAFT, MODEL, EMPLOYEE, PILOT, and CUSTOMER tables, and they must produce the fact table and the dimension table.

Figure P13.4A The RobCor Relational Diagram

b. Define the dimensions and attributes for the charter operation’s star schema.

The dimensions are TIME, MODEL, and PILOT. Each of these dimensions is depicted in Figure P13.4a’s star schema figure. The attributes are:

- **Time dimension**: time id, year, and month.
- **Model dimension**: model code, manufacturer, name, number of seats, etc.
- **Pilot dimension**: employee number, pilot license, pilot ratings, etc.

c. Define the necessary attribute hierarchies.

The main attribute hierarchy is based on the sequence year-month-model-pilot. The aggregate analysis is based on this hierarchy. We can produce a query to generate revenue, hours flown, and fuel used on a yearly basis. We can then drill down to a monthly time period to generate the aggregate information for each model of airplane. We can also drill down to get that information about each pilot.

d. Implement the data warehouse design, using the design components you developed in Problems 4a-4c.

The Ch13-P4sol.mdb database contains the data and solutions for problems 4a-4c.

e. Generate the reports that will illustrate that your data warehouse is able to meet the specified information requirements.

The Ch13-P4sol.mdb database contains the solution for problem 4e.

Using the data provided in the SaleCo Snowflake schema in Figure 13.24, solve the following problems.

**ONLINE CONTENT**

The script files used to populate the database are available in the Student Online Companion. The script files assume an Oracle RDBMS. If you use a different DBMS, consult the documentation to verify whether the vendor supports similar functionality and what the proper syntax is for your DBMS.

5. What is the SQL command to list the total sales by customer and by product, with subtotals by customer and a grand total for all product sales? (Hint: Use the ROLLUP command.)

```sql
SELECT CUS_CODE, P_CODE, SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAVSFACT NATURAL JOIN DWCUSTOMER
GROUP BY ROLLUP (CUS_CODE, P_CODE)
ORDER BY CUS_CODE, P_CODE;
```

6. What is the SQL command to list the total sales by customer, month and product, with subtotals by customer and by month and a grand...
total for all product sales? (Hint: Use the ROLLUP command.)

```sql
SELECT CUS_CODE, TM_MONTH, P_CODE, SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAYSALESFACT NATURAL JOIN DWCUSTOMER NATURAL JOIN DWTIME
GROUP BY ROLLUP (CUS_CODE, TM_MONTH, P_CODE)
ORDER BY CUS_CODE, TM_MONTH, P_CODE;
```

7. What is the SQL command to list the total sales by region and customer, with subtotals by region and a grand total for all sales? (Hint: Use the ROLLUP command.)

```sql
SELECT REG_ID, CUS_CODE, SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAYSALESFACT NATURAL JOIN DWCUSTOMER
NATURAL JOIN DWREGION
GROUP BY ROLLUP (REG_ID, CUS_CODE)
ORDER BY REG_ID, CUS_CODE;
```

8. What is the SQL command to list the total sales by month and product category, with subtotals by month and a grand total for all sales? (Hint: use the ROLLUP command.)

```sql
SELECT TM_MONTH, P_CATEGORY, SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAYSALESFACT NATURAL JOIN DWPRODUCT
NATURAL JOIN DWTIME
GROUP BY ROLLUP (TM_MONTH, P_CATEGORY)
ORDER BY TM_MONTH, P_CATEGORY;
```

9. What is the SQL command to list the number of product sales (number of rows) and total sales by month, with subtotals by month and a grand total for all sales? (Hint: use the ROLLUP command.)

```sql
SELECT TM_MONTH, COUNT(*) AS NUMPROD, SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAYSALESFACT NATURAL JOIN DWTIME
GROUP BY ROLLUP (TM_MONTH)
ORDER BY TM_MONTH;
```

10. What is the SQL command to list the number of product sales (number of rows) and total sales by month and product category with subtotals by month and product category and a grand total for all sales? (Hint: use the ROLLUP command.)

```sql
SELECT TM_MONTH, P_CATEGORY, COUNT(*) AS NUMPROD,
SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAYSALESFACT NATURAL JOIN DWPRODUCT
NATURAL JOIN DWTIME
GROUP BY ROLLUP (TM_MONTH, P_CATEGORY)
ORDER BY TM_MONTH, P_CATEGORY;
```

11. What is the SQL command to list the number of product sales (number of rows) and total sales by month, product category and product with subtotals by month and product category and product and a grand total for all sales? (Hint: use the ROLLUP command.)

```sql
SELECT TM_MONTH, P_CATEGORY, P_CODE, COUNT(*) AS NUMPROD,
SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAYSALESFACT NATURAL JOIN DWPRODUCT
NATURAL JOIN DWTIME
GROUP BY ROLLUP (TM_MONTH, P_CATEGORY, P_CODE)
ORDER BY TM_MONTH, P_CATEGORY, P_CODE;
```

12. Using the answer to Problem 10 as your base, what command would you need to generate the same output but with subtotals in all columns? (Hint: Use the CUBE command.)

```sql
SELECT TM_MONTH, P_CATEGORY, COUNT(*) AS NUMPROD,
SUM(SALE_UNITS*SALE_PRICE) AS TOTSALES
FROM DWDAYSALESFACT NATURAL JOIN DWPRODUCT NATURAL JOIN DWTIME
GROUP BY CUBE (TM_MONTH, P_CATEGORY)
ORDER BY TM_MONTH, P_CATEGORY;
```