MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2005 Certified)
WINTER – 15 EXAMINATION
Model Answer

Subject Code: 17520 Subject Name: DATAWARE HOUSING AND DATA MINING

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in themodel answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given moreImportance (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Marks

1. a) Attempt any three of the following: **12**

a) Describe the need of datawarehousing.

(For each point – 1 Mark (any four points))

Ans:

1) **Advanced query processing:** in most businesses, even the best database systems are bound to either a single server or a handful of servers in a cluster. A data warehouse is a purpose-built hardware solution far more advanced than standard database servers. What this means is a data warehouse will process queries much faster and more effectively, leading to efficiency and increased productivity.
2) **Better consistency of data:** developers work with data warehousing systems after data has been received so that all the information contained in the data warehouse is standardized. Only uniform data can be used efficiently for successful comparisons. Other solutions simply cannot match a data warehouse's level of consistency.
3) **Improved user access:** a standard database can be read and manipulated by programs like SQL Query Studio or the Oracle client, but there is considerable ramp up time for end users to effectively use these apps to get what they need. Business intelligence and data warehouse end-user access tools are built specifically for the purposes data warehouses are used: analysis, benchmarking, prediction and more.

4) **All-in-one:** a data warehouse has the ability to receive data from many different sources, meaning any system in a business can contribute its data. Let's face it: different business segments use different applications. Only a proper data warehouse solution can receive data from all of them and give a business the "big picture" view that is needed to analyze the business, make plans, track competitors and more.
5) **Future-proof:** a data warehouse doesn't care where it gets its data from. It can work with any raw information and developers can "massage" any data it may have trouble with. Considering this, you can see that a data warehouse will outlast other changes in the business' technology. For example, a business can overhaul its accounting system, choose a whole new CRM solution or change the applications it uses to gather statistics on the market and it won't matter at all to the data warehouse. Upgrading or overhauling apps anywhere in the enterprise will not require subsequent expenditures to change the data warehouse side.
6) **Retention of data history:** end-user applications typically don't have the ability, not to mention the space, to maintain much transaction history and keep track of multiple changes to data. Data warehousing solutions have the ability to track all alterations to data, providing a reliable history of all changes, additions and deletions. With a data warehouse, the integrity of data is ensured.
7) **Disaster recovery implications:** a data warehouse system offers a great deal of security when it comes to disaster recovery. Since data from disparate systems is all sent to a data warehouse, that data warehouse essentially acts as another information backup source. Considering the data warehouse will also be backed up, that's now four places where the same information will be stored: the original source, its backup, the data warehouse and its subsequent backup. This is unparalleled information security.

b) Describe the role of metadata in datawarehouse.

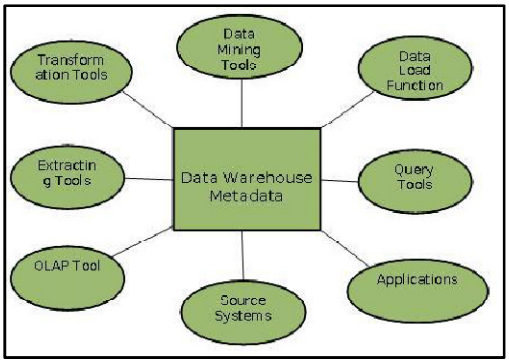
(For each point – 1 Mark (any four points))

Ans:

Metadata has very important role in data warehouse. The role of metadata in warehouse is different from the warehouse data yet it has very important role. The various roles of metadata are explained below.

- ✓ The metadata act as a directory. This directory helps the decision support system to locate the contents of data warehouse.
- ✓ Metadata helps in decision support system for mapping of data when data are transformed from operational environment to data warehouse environment.
- ✓ Metadata helps in summarization between current detailed data and highly summarized data.

- ✓ Metadata also helps in summarization between lightly detailed data and highly summarized data.
- ✓ Metadata are also used for query tools.
- ✓ Metadata are used in reporting tools.
- ✓ Metadata are used in extraction and cleansing tools.
- ✓ Metadata are used in transformation tools.
- ✓ Metadata also plays important role in loading functions.



c) Explain schema for star multidimensional database.

(Explanation of Schema – 2 Marks, Example – 2 Marks)

Ans:

Star Schema is the special design technique for multidimensional data representations. It is called a star schema because the diagram resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables. Usually t he fact tables in a star schema are in third normal form (3NF) whereas dimensional tables are de-normalized.

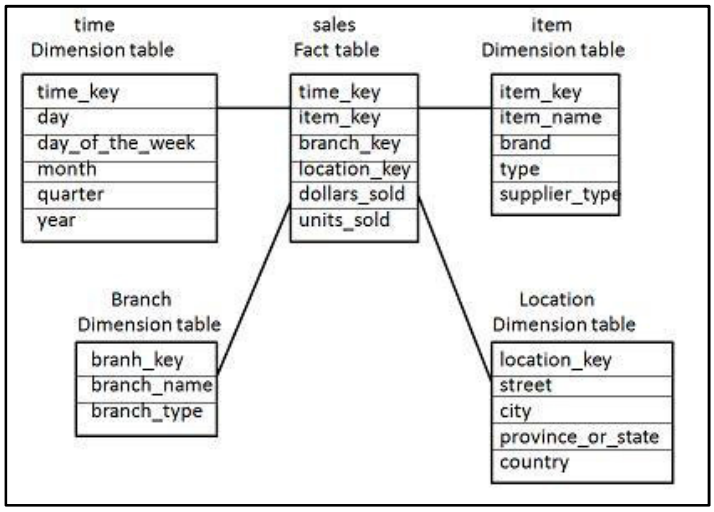
Fact Tables: A fact table typically has two types of columns: foreign keys to dimension tables and measures those that contain numeric facts. A fact table can contain fact's data on detail or aggregated level.

Dimension Tables: A dimension is a structure usually composed of one or more hierarchies that categorizes data. If a dimension hasn't got a hierarchies and levels it is called flat dimension or list. The primary keys of each of the dimension tables are part of the composite primary key of the fact table. Dimensional attributes help to describe the dimensional value. They are normally descriptive, textual values. Dimension tables are generally small in size then fact table. Typical fact tables store data about sales while dimension tables data about geographic region (markets, cities), clients, products, times, channels.

Steps in designing star schema

- Identify a business process for analysis.
- Identify measures or facts.
- Identify the dimensions for facts.
- List the columns that describe the each dimension.
- Determine the lowest level of summary in a fact table.

In the following diagram we have shown the sales data of a company with respect to the four dimensions namely, time, item, branch and location.



There is a fact table at the center. This fact table contains the keys to each of four dimensions. The fact table also contain the attributes namely, dollars sold and units sold.

d) Explain concept description.

(Explanation – 4 Marks)

Ans:

Concept Description: Concept description refers to the data to be associated with the classes or concepts. For example, in a company, the classes of items for sales include computer and printers, and concepts of customers include big spenders and budget spenders. Such descriptions of a class or a concept are called class/concept descriptions. These descriptions can be derived by the following two ways

1. **Data Characterization –** This refers to summarizing data of class under study. This class under study is called as Target Class.

2. **Data Discrimination –** It refers to the mapping or classification of a class with some predefined group or class.

The simplest kind of descriptive data mining is concept description. A concept usually refers to a collection of data such as frequent buyers, graduate students, and so on. As a data mining task, concept description is not a simple enumeration of the data. Instead, concept description generates descriptions for characterization and comparison of the data. It is sometimes called class description, when the concept to be described refers to a class of objects. Characterization provides a concise and succinct summarization of the given collection of the data, while concept or class comparison (also known as discrimination) provides discriminations comparing two or more collections of data. Since concept description involves both characterization and comparison, techniques for accomplishing each of these tasks will study. Concept description has close ties with the data generalization. Given the large amount of data stored in database, it is useful to be describe concepts in concise and succinct terms at generalized at multiple levels of abstraction facilities users in examining the general behavior of the data. Given the ABCompany database, for example, instead of examining individual customer transactions, sales managers may prefer to view the data generalized to higher levels, such as summarized to higher levels, such as summarized by customer groups according to geographic regions, frequency of purchases per group, and customer income. Such multiple dimensional, multilevel data generalization is similar to multidimensional data analysis in data warehouses.

b) Attempt any one of the following: 6

a) Describe various characteristics of datawarehouse.

(For each point – 1 Mark (any six characteristic))

Ans:

1) **Subject Oriented:** Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?" This ability to define a data warehouse by subject matter, sales in this case makes the data warehouse subject oriented.
2) **Integrated:** Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as

3) naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated.
4) **Nonvolatile:** Nonvolatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable you to analyze what has occurred.
5) **Time Variant:** In order to discover trends in business, analysts need large amounts of data. This is very much in contrast to online transaction processing (OLTP) systems, where performance requirements demand that historical data be moved to an archive. A data warehouse's focus on change over time is what is meant by the term time variant.
   Typically, data flows from one or more online transaction processing (OLTP) databases into a data warehouse on a monthly, weekly, or daily basis. The data is normally processed in a staging file before being added to the data warehouse. Data warehouses commonly range in size from tens of gigabytes to a few terabytes. Usually, the vast majority of the data is stored in a few very large fact tables.
6) **Separate:** The DW is separate from the operational systems in the company. It gets its data out of these legacy systems.
7) **Available:** The task of a DW is to make data accessible for the user.
8) **Aggregation performance:** The data which is requested by the user has to perform well on all scales of aggregation.
9) **Consistency:** Structural and contents of the data is very important and can only be guaranteed by the use of metadata: this is independent from the source and collection date of the data

b) Why there is need of preprocessing of data?

(Explanation - 6 Marks)

Ans:

Incomplete, inconsistent and noisy data are commonplace properties of large real-world databases. Attributes of interest may not always be available and other data was included just because it was considered to be important at the time of entity. Relevant data may not sometimes be recorded. Furthermore, the recording of the modifications to the data may not have been done. There are many possible reasons for noisy data (incorrect attribute values). They could have been human as well as computer errors that occurred during data entry. There could be inconsistent in the naming conventions adopted. Sometimes duplicate tuples may occur. Data cleaning routines work to ―clean the data by filling in the missing values, smoothing noisy data, identifying and remo0ving outliers, and resolving inconsistencies in the data. Although mining routines have some form of handling noisy data, they are always not robust. If you would like to include files from many sources in your analysis then requires data integration. Naming inconsistencies may occur in this context. A large amount of redundant data may confuse or slow down the knowledge discovery process. In addition to data cleaning steps must take to remove redundancies in the data. Sometimes data would have to be normalized so that it scaled to a specific range e.g. [0.0,

1.0] in order to work data mining algorithms such as neural-networks, or clustering. Furthermore, you would require aggregating data e.g. as sales per region-something the t is not part of the data transformation methods need to be applied to the data. Data reduction obtains a reduced representation of the data set that is much smaller in volume, yet produces the same or almost the same analytical results. There are a number of strategies for data reduction-data compression, numerosity reduction, generalization; data reduction

2. Attempt any two of the following: 16

a) Describe the method of handling missing value for data cleaning.

(Explanation of Missing values - 2 Marks, Methods – 6 Marks (1 Mark for each method))

Ans:

Data cleaning is performed as data preprocessing step while preparing the data for a data warehouse. Real-world data tend to be incomplete, noisy, and inconsistent. Data cleaning (or data cleansing) routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

Missing Values: Imagine that you need to analyze All Electronics sales and customer data. You note that many tuples have no recorded value for several attributes, such as customer income. Filling in the missing values of this attribute can be done using the following methods:

1). **Ignore the tuple:** This is usually done when the class label is missing (assuming the mining task involves classification). This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.

2). **Fill in the missing value manually:** In general, this approach is time-consuming andmay not be feasible given a large data set with many missing values

3). **Use a global constant to fill in the missing value:** Replace all missing attribute values by the same constant, such as a label like ―Unknown or ¥. If missing values are replaced by, say, ―Unknown, then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common—that of ―Unknown. Hence, although this method is simple, it is not foolproof.

4). **Use the attribute mean to fill in the missing value:** For example, suppose that the average income of All Electronics customers is $56,000. Use this value to replace the missing value for income.

5). **Use the attribute mean for all samples belonging to the same class as the given tuple:** For example, if classifying customers according to credit risk, replace the missing value with the average income value for customers in the same credit risk category as that of the given tuple.

6). **Use the most probable value to fill in the missing value:** This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction. For

example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for income.

b) Describe schema for following multidimensional database:

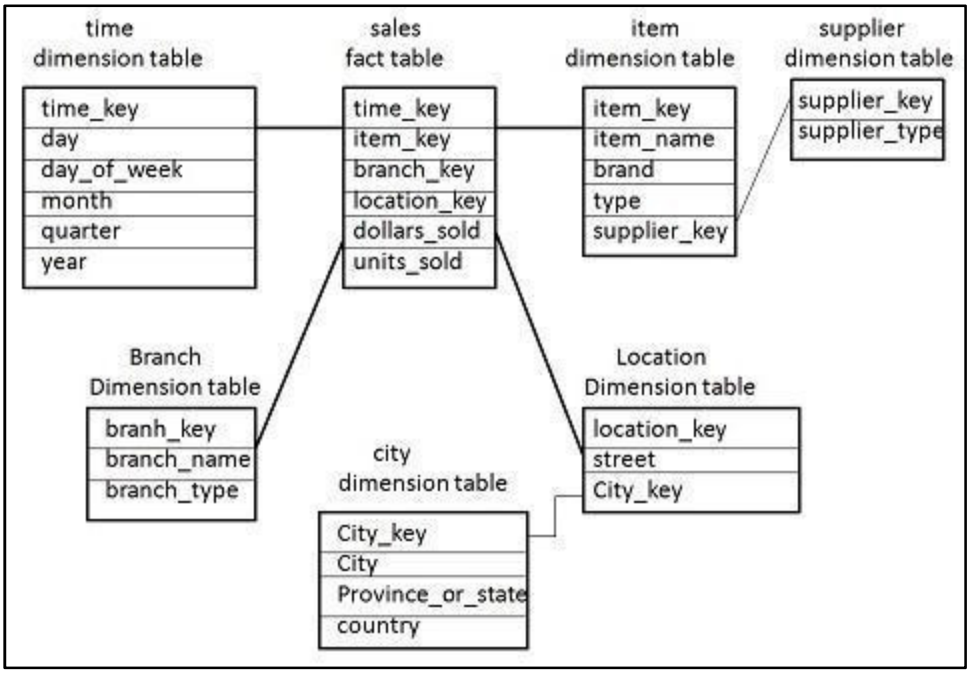
1) **Snowflake**
2) **Star join**

(Snowflake - 4 Marks, Star Join – 4 Marks)

Ans:

1) Snowflake

A snowflake schema applies normalization over a star schema, in which very large dimension tables are normalized into multiple tables. Dimensions with hierarchies can be decomposed into a snowflake structure when you want to avoid joins to big dimension tables when you are using an aggregate of the fact table. In Snowflake schema some dimension tables are normalized. The normalization split up the data into additional tables. Snowflake schema helps in saving space by normalizing dimension tables.
Unlike Star schema the dimensions table in snowflake schema is normalized for example the item dimension table in star schema is normalized and split into two dimension tables namely, item and supplier table.



Therefore now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key. The supplier key is linked to supplier dimension table. The supplier dimension table contains the attributes supplier_ key, and supplier_type.

2) Star Join

Star Schema is the special design technique for multidimensional data representations. It is called a star schema because the diagram resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables. Usually the fact tables in a star schema are in third normal form(3NF) whereas dimensional tables are de-normalized.

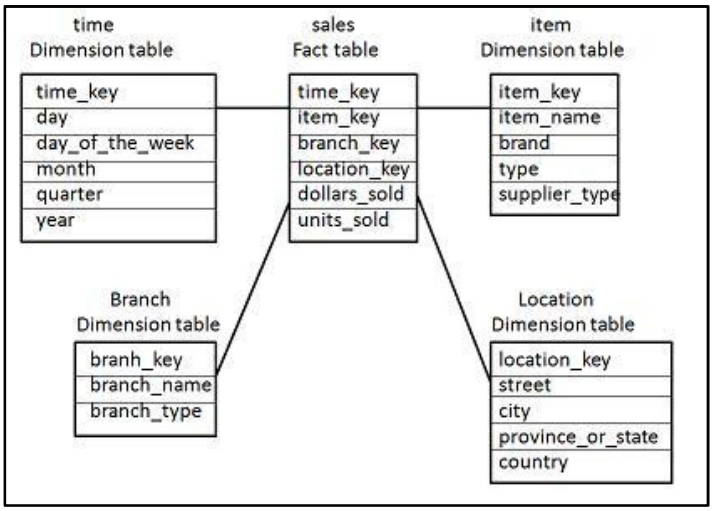
Fact Tables: A fact table typically has two types of columns: foreign keys to dimension tables and measures those that contain numeric facts. A fact table can contain fact's data on detail or aggregated level.

Dimension Tables: A dimension is a structure usually composed of one or more hierarchies that categorizes data. If a dimension hasn't got a hierarchies and levels it is called flat dimension or list. The primary keys of each of the dimension tables are part of the composite primary key of the fact table. Dimensional attributes help to describe the dimensional value. They are normally descriptive, textual values. Dimension tables are generally small in size then fact table. Typical fact tables store data about sales while dimension tables data about geographic region (markets, cities), clients, products, times, channels.

Steps in designing star schema

- Identify a business process for analysis.
- Identify measures or facts.
- Identify the dimensions for facts.
- List the columns that describe the each dimension.
- Determine the lowest level of summary in a fact table.

In the following diagram we have shown the sales data of a company with respect to the four dimensions namely, time, item, branch and location.



There is a fact table at the center. This fact table contains the keys to each of four dimensions. The fact table also contain the attributes namely, dollars sold and units sold.

c) Explain market basket analysis.

(Definition - 2 Marks, Example - 2 Marks, Explanation - 4 Marks)

Ans:

Market basket analysis is a technique that discovers relationships between pairs of products purchased together. The technique can be used to uncover interesting cross-sells and related products. The idea behind market basket analysis is simple. Simply examine your orders for products have been purchased together. For example using market basket analysis you might un cover the fact that customers tend to buy hot dogs and buns together. Using this information you might organize the store so that hot dogs and buns are next to each other. In an e-commerce environment you might create a cross-sell rule to offer the shopper buns whenever they place hot dogs in their shopping cart. There are a couple of measures we use when doing market basket analysis and they are described here. The first measure is the frequency. The frequency is defined as the number of times two products were purchased together. If hot dogs and buns were found together in 820 baskets this would be its frequency.

Frequency by itself doesn't tell the whole story. For instance if I told you hot dogs and buns were purchased 820 times together you wouldn't know if that was relevant or not. Therefore we introduce two other measures called support and confidence to help with the analysis.

If you divide the frequency by the total number of orders you get the percentage of order containing the pair. This is called the support. Another way to thinking about support is as the probability of the pair being purchased. Now if 820 hot dogs and buns were purchased together and your store took 1000 orders the support for this would be calculated as (820 / 1000) = 82.0%

We can extend this even further by defining a calculation called confidence. Confidence compares the number of times the pair was purchased to the number of times one of the items in the pair was purchased. In probability terms this is referred to as the conditional probability of the pair. So going back to our hot dogs example if hot dogs were purchases 900 times and out of those 900 purchases 820 contained buns we would have a confidence of (820 / 900) = 91.1%.Now that we've defined frequency, support and confidence we can talk a little about what a market basket analysis report might look like. The report would have the user select the product they are interested in performing the analysis on (i.e. hot dogs). Then it would list all the products that were purchased together with the selected products ranked by it frequency. It might look something like the following

Market Basket Analysis: Hot Dogs:

Product	Frequency	Support	Confidence
Buns	820	82.0%	91.1%
Ketchups	800	80.0%	23.2%
Mustard	750	75.0%	34.5%
Jello	321	32.1%	45.2%

Example2: Suppose, as manager of an All Electronics branch, you would like to learn more about the buying habits of your customers. Specifically, you wonder, ―Which groups or sets of items are customers likely to purchase on a given trip to the store? To answer your question, market basket analysis may be performed on the retail data of customer transactions at your store. You can then use the results to plan marketing or advertising strategies, or in the design of a new catalog. For instance, market basket analysis may help you design different store layouts. In one strategy, items that are frequently purchased together can be placed in proximity in order to further encourage the sale of such items together.

Association rule mining is a two-step process:

1. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count.
2. Generate strong association rules from the frequent itemsets: By definition, these rules must satisfy minimum support and minimum support and minimum confidence. Additional interestingness measures can be applied, if desired. The second step is the easiest, of the two. The overall performance of mining association rules is determined by the first step.

3. Attempt any four of the following: 16

a) Give brief introduction of decision support system.

(Explanation - 3 Marks, Types - 1 Mark)

Ans:

Decision support systems are interactive software-based systems intended to help managers in decision making by accessing large volume of information generated from various related information systems involved in organizational business processes, like, office automation system, transaction processing system etc. DSS uses the summary information, exceptions, patterns and trends using the analytical models. Decision Support System helps in decision making but does not always give a decision itself. The decision makers compile useful

information from raw data, documents, personal knowledge, and/or business models to identify and solve problems and make decisions.

Programmed and Non-programmed Decisions

There are two types of decisions - programmed and non-programmed decisions.

Programmed decisions are basically automated processes, general routine work, where:

- These decisions have been taken several times
- These decisions follow some guidelines or rules

Non-programmed decisions occur in unusual and non-addressed situations, so:

- It would be a new decision
- There will not be any rules to follow
- These decisions are made based on available information
- These decisions are based on the manger's discretion, instinct, perception and judgment

Decision support systems generally involve non-programmed decisions. Therefore, there will be no exact report, content or format for these systems.

b) Explain data integration in warehouse.

(Explanation – 4 Marks)

Ans:

Data analysis task will involve data integration, which combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files.

There are a number of issues to consider during data integration. Schema integration and object matching can be tricky. How can equivalent real-world entities from multiple data sources be matched up? This is referred to as the entity identification problem.

For example, how can the data analyst or the computer be sure that customer id in one database and cust number in another refers to the same attribute? Examples of metadata for each attribute include the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values.

Such metadata can be used to help avoid errors in schema integration. The metadata may also be used to help transform the data (e.g., where data codes for pay type in one database may be "H" and "S", and 1 and 2 in another). Hence, this step also relates to data cleaning, Redundancy is another important issue. An attribute (such as annual revenue, for instance) may be redundant if it can be "derived" from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

c) State the term mining which applied on world wide web.

(Term – 1 Mark, Explanation – 3 Marks)

Ans:

Data mining refers to extracting or "mining" knowledge from large amounts of data. Mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web.

The World Wide Web contains the huge information such as hyperlink information, web page access info, education etc that provide rich source for data mining. The basic structure of the web page is based on Document Object Model (DOM). The DOM structure refers to a tree like structure. In this structure the HTML tag in the page corresponds to a node in the DOM tree. We can segment the web page by using predefined tags in HTML. The HTML syntax is flexible therefore; the web pages do not follow the W3C specifications. Not following the specifications of W3C may cause error in DOM tree structure.

The DOM structure was initially introduced for presentation in the browser not for description of semantic structure of the web page. The DOM structure cannot correctly identify the semantic relationship between different parts of a web page.

d) Describe method of generalization based on characterization.

(Explanation – 4 Marks)

Ans:

Data and objects in databases often contain detailed information at primitive concept levels. .For example, the item relation in sales database may contain attributes describing low-level item information such s item _ID, name, brand, category, supplier, place_made, and price. It is useful to be able to summarize a large set or data and present it at a high conceptual level. For example, summarizing a large set of items relating to Christmas season sales provides a general description of such data, which can be very helpful for sales and marketing managers. This requires an important functionality in data mining: data generalization.

Data generalization is a process that abstracts a large set of task-relevant data in a database from a relatively low conceptual level to higher conceptual levels. Methods for the efficient and flexible generalization of large data sets can be categorized according to two approaches:

(1) The data cube (or OLAP) approach and

(2) The attribute –oriented induction approach .In this section, we describe the attribute-oriented induction approach.

Attribute-Oriented Induction - The attribute-oriented induction (AOI)) approach to data generalization and summarization-based characterization was first proposed in 1989, a few years prior to the introduction of the data cube approach. The data cube approach can be considered as a data warehouse-based, pre-computation oriented materialized-view approach. It performs off-line

aggregation before an OLAP or data mining query is submitted for processing. On the other hand, the attribute-oriented induction approach, at least in its initial proposal, is a relational database query –oriented, generalization –based, on-line data analysis technique. However, there is no inherent barrier distinguishing the two approaches based on on-line aggregation versus off-line pre computation. Some aggregations in the data cube can be computed on-line, while off-line while off-line pre -computation of multidimensional space can speed up attribute –oriented induction as well.

e) Describe the concept of hierarchy generation for numeric data.

(Explanation – 4 Marks)

Ans:

Concept hierarchies for numeric attributes can be constructed automatically based on data distribution analysis. Five methods for concept hierarchy generation are defined below- binning histogram analysis entropy-based discretization and data segmentation by ―natural partitioning‖. Binning: Attribute values can be discretized by distributing the values into bin and replacing each bin by the mean bin value or bin median value. These technique can be applied recursively to the resulting partitions in order to generate concept hierarchies.

Histogram Analysis: Histograms can also be used for discretization. Partitioning rules can be applied to define range of values. The histogram analyses algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy, with the procedure terminating once a pre specified number of concept levels have been reached. A minimum interval size can be used per level to control the recursive procedure. This specifies the minimum width of the partition, or the minimum member of partitions at each level.

Cluster Analysis: A clustering algorithm can be applied to partition data into clusters or groups. Each cluster forms a **node** of a concept hierarchy, where all noses are at the same conceptual level. Each cluster may be further decomposed into sub-clusters, forming a lower level in the hierarchy. Clusters may also be grouped together to form a higher-level concept hierarchy.

Segmentation by natural partitioning: Breaking up annual salaries in the range of into ranges like ($50,000-$100,000) are often more desirable than ranges like ($51, 263, 89-$60,765.3) arrived at by cluster analysis. The 3-4-5 rule can be used to segment numeric data into relatively uniform ―natural intervals. In general the rule partitions a give range of data into 3,4,or 5 equinity intervals, recursively level by level based on value range at the most significant digit. The rule can be recursively applied to each interval creating a concept hierarchy for the given numeric attribute.

4. A) Attempt any three of the following: 12

a) How does data reduction technique help to reduce size of data?

(List of Strategies - 1 Mark, Explanation - 3 Marks)

Ans:

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results. Strategies for data reduction include the following:

1. Data cube aggregation: where aggregation operations are applied to the data in the construction of a data cube.

2. Attribute subset selection: where irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed.

3. Dimensionality reduction: where encoding mechanisms are used to reduce the data set size.

4. Numerosity reduction: where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model parameters instead of the actual data) or nonparametric methods such as clustering, sampling, and the use of histograms.

5. Discretization and concept hierarchy generation: where raw data values for attributes are replaced by ranges or higher conceptual levels. Data discretization is a form of numerosity reduction that is very useful for the automatic generation of concept hierarchies. Discretization and concept hierarchy generation are powerful tools for data mining, in that they allow the mining of data at multiple levels of abstraction

b) Describe need for OLAP in warehouse.

(Explanation - 4 Marks)

Ans:

OLAP (online analytical processing) is a function of business intelligence software that enables a user to easily and selectively extract and view data from different points of view. OLAP technology is a vast improvement over traditional relational database management systems (RDBMS). Relational databases, which have a two-dimensional structure, do not allow the multidimensional data views that OLAP provides. Traditionally used as an analytical tool for marketing and financial reporting, OLAP is now viewed as a valuable tool for any management system that needs to create a flexible decision support system.

Today's work environment is characterized by flatter organizations that need to be able to adapt quickly to changing conditions. Managers need the tools that will allow them to make quick, intelligent decisions on the fly. Making the wrong decision or taking too long to make it can affect the competitive position of an organization. OLAP provides the multidimensional capabilities that most organizations need today.

By using a multidimensional data store, also known in the industry as a hypercube, OLAP allows the end user to analyze data along the axes of their business. The two most common forms of analysis that most businesses use are called "slice and dice" and "drill down".

c) Draw block diagram of datawarehouse architecture and state the function of each component.

(Block Diagram - 2 Marks, Function - 2 Marks)

Ans:

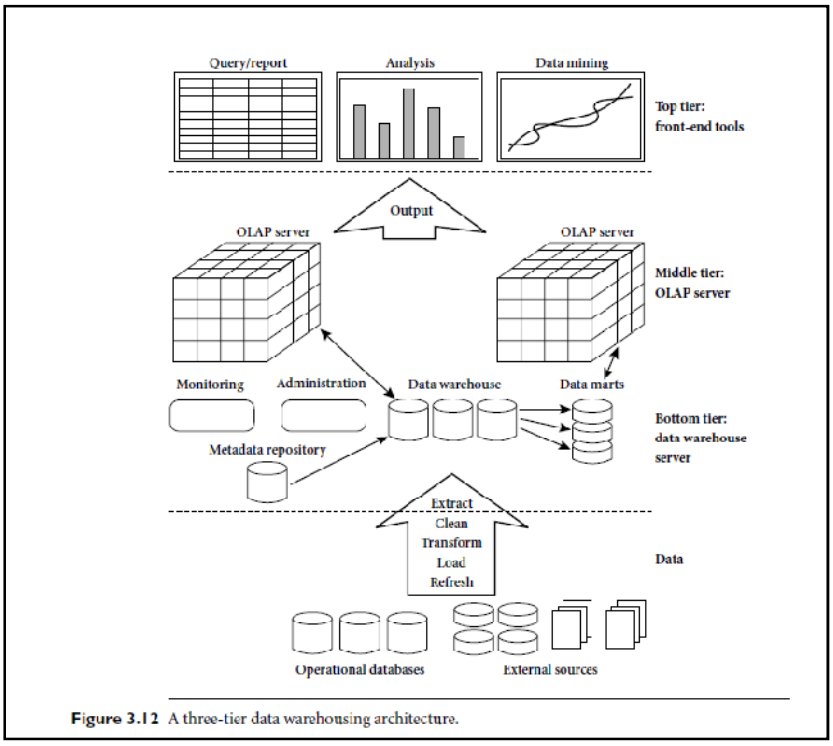


Figure 3.12 A three-tier data warehousing architecture.

1). The bottom tier is a warehouse database server that is almost always a relational database system. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources (such as customer profile information provided by external consultants). These tools and utilities perform data extraction, cleaning, and transformation (e.g., to merge similar data from different Sources into a unified format), as well as load and refresh functions to update the data warehouse. The data are extracted using application program interfaces known as gateways. A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server. Examples of gateways include ODBC (Open Database Connection) and OLEDB (Open Linking And Embedding for Databases) by

Microsoft and JDBC (Java Database Connection).This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

2). The middle tier is an OLAP server that is typically implemented using either (1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or (2) A multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

3). The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on).

d) Describe data cube aggregation strategy for data reduction technique.

(Explanation - 4 Marks)

Ans:

Imagine that you have collected the data for your analysis. These data consist of the All Electronics sales per quarter, for the years 2002 to 2004. You are, however, interested in the annual sales (total per year), rather than the total per quarter. This aggregation is illustrated in Figure 2.13 Thus the data can be aggregated so that the resulting data summarize the total sales per year instead of per quarter. The resulting data set is smaller in volume, without loss of information necessary for the analysis task.

Data cubes store multidimensional aggregated information. For example, Figure 2.14 shows a data cube for multidimensional analysis of sales data with respect to annual sales per item type for each All Electronics branch. Each cell holds an aggregate data value, corresponding to the data point in multidimensional space. (For readability, only some cell values are shown.) Concept hierarchies may exist for each attribute, allowing the analysis of data at multiple levels of abstraction. For example, a hierarchy for branch could allow branches to be grouped into regions, based on their address. Data cubes provide fast access to pre computed summarized data, thereby benefiting on-line analytical processing as well as data mining. The cube created at the lowest level of abstraction is referred to as the base cuboid. The base cuboid should correspond to an individual entity of interest, such as sales or customer. In other words, the lowest level should be usable, or useful for the analysis.

A cube at the highest level of abstraction is the apex cuboid. For the sales data of Figure 2.14, the apex cuboid would give one total—the total sales for all three years, for all item types, and for all branches. Data cubes created for varying levels of abstraction are often referred to as cuboids, so that a data cube may instead refer to a lattice of cuboids. Each higher level of abstraction further reduces the resulting data size. When replying to data mining requests, the smallest available cuboid relevant to the given task should be used.

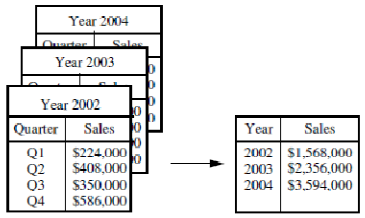


Figure 2.13 Sales data for a given branch of *AllElectronics* for the years 2002 to 2004. On the left, the sales are shown per quarter. On the right, the data are aggregated to provide the annual sales.

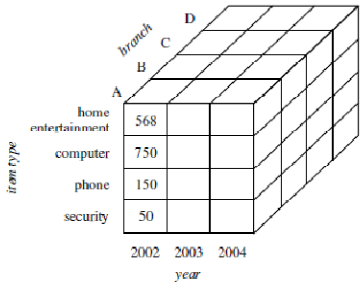


Figure 2.14 A data cube for sales at *AllElectronics*.

B) Attempt any one of the following: 6

a) Describe the Apriori algorithm.

(Algorithm - 4 Marks, Description of algorithm - 2 Marks)

Ans:

Apriori is a seminal algorithm for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties Apriori employs an iterative approach known as a level-wise search, where k itemsets are used to explore (k + 1)-itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L1. Next, L1 is used to find L2 the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database.

Once the frequent itemsets from transactions in a database D have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence). This can be done using Equation for confidence, which we show again here for completeness:

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

Input:
D, a database of transactions;
Min_sup, the minimum support count threshold

Output: *L*, frequent itemsets in *D*

Method:

```
(1)  L1 = find_frequent_1-itemsets(D);
(2)  for (k = 2; Lk−1 ≠ ϕ; k++) {
(3)      Ck = apriori_gen(Lk−1);
(4)      for each transaction t ∈ D { // scan D for counts
(5)          Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)          for each candidate c ∈ Ct
(7)              c.count++;
(8)      }
(9)      Lk = {c ∈ Ck | c.count ≥ min_sup}
(10) }
(11) return L = ∪k Lk;

procedure apriori_gen(Lk−1:frequent (k − 1)-itemsets)
(1)  for each itemset l1 ∈ Lk−1
(2)      for each itemset l2 ∈ Lk−1
(3)          if (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧ ... ∧ (l1[k − 2] = l2[k − 2]) ∧ (l1[k − 1] < l2[k − 1]) then {
(4)              c = l1 ⋈ l2; // join step: generate candidates
(5)              if has_infrequent_subset(c, Lk−1) then
(6)                  delete c; // prune step: remove unfruitful candidate
(7)              else add c to Ck;
(8)          }
(9)  return Ck;

procedure has_infrequent_subset(c: candidate k-itemset;
         Lk−1: frequent (k − 1)-itemsets); // use prior knowledge
(1)  for each (k − 1)-subset s of c
(2)      if s ∉ Lk−1 then
(3)          return TRUE;
(4)  return FALSE;
```

b) Describe various categories of DSS.

(List - 2 Marks, Explanation – 4 Marks)

Ans:

DSS have been classified in different ways as the concept matured with time. As. and when the full potential and possibilities for the field emerged, different classification systems also emerged. Some of the well known classification models are given below:

According to Donovan and Madnick (1977) DSS can be classified as:

1). Institutional-when the DSS supports ongoing and recurring decisions

2). Ad hoc-when the DSS supports a one off-kind of decision.

Hackathorn and Keen (1981) classified DSS as,

1). Personal DSS

2). Group DSS

3). Organizational DSS

Alter (1980) opined that decision support systems could be classified into seven types based on their generic nature of operations. He described the seven types as,

1). File drawer systems. This type of DSS primarily provides access to data stores/data related items. Examples--ATM Machine, Use the balance to make transfer of funds decisions

2). Data analysis systems. This type of DSS supports the manipulation of data through the use of specific or generic computerized settings or tools. Examples: Airline Reservation system, use the info to make flight plans

3). Analysis information systems. This type of DSS provides access to sets of decision oriented databases and simple small models.

4). Accounting and financial models. This type of DSS can perform 'what if analysis' and calculate the outcomes of different decision paths. Examples: calculate production cost, make pricing decisions

5). Representational models. This type of DSS can also perform 'what if analysis' and calculate the outcomes of different decision paths, based on simulated models.

6). Optimization models. This kind of DSS provides solutions through the use of optimization models which have mathematical solutions.

7). Suggestion models. This kind of DSS works when the decision to be taken is based on well structured tasks. Examples: Expert System• Applicant applies for personal loan

5. Attempt any two of the following: 16

a) Describe OLAP operation in multidimensional data model.

(Diagram – 2 Marks, Points - 2 Marks, Explanation - 4 Marks)

Ans:

In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives. A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand. Hence, OLAP provides a user-friendly environment for interactive data analysis.

Example: OLAP operations:

Each of the operations described below is illustrated in below Figure. At the center of the figure is a data cube for *All Electronics* sales. The cube contains the dimensions *location, time*, and *item*, where *location* is aggregated with respect to city values, *time* is aggregated with respect to quarters, and *item* is aggregated with respect to item types.

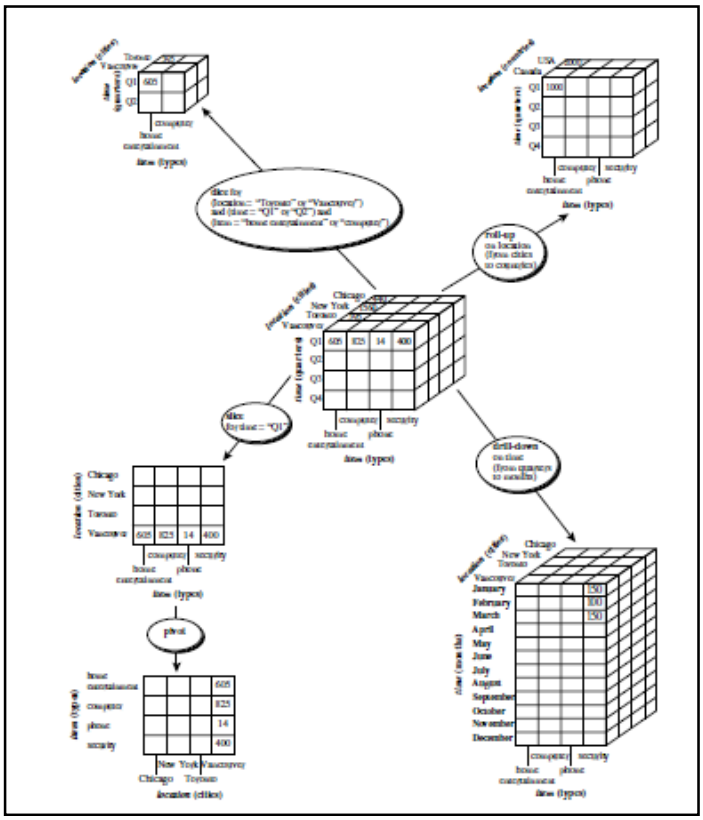


Figure: Example of typical OLAP operation on Multidimensional data

This cube is referred to as the central cube. The measure displayed is *dollars sold* (in thousands). The data examined are for the cities Chicago, New York, Toronto, and Vancouver.

Roll-up: The roll-up operation (also called the *drill-up* operation by some vendors) performs aggregation on a data cube, either by *climbing up a concept hierarchy* for a dimension or by *dimension reduction*. Figure shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for *location* given in Figure. This hierarchy was defined as the total order "*street< city < province or state < country*." The roll-up operation shown aggregates the data by ascending the *location* hierarchy from the level of *city* to the level of *country*. In other words, rather than grouping the data by city, the resulting cube
groups the data by country. When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube. For example, consider a sales data cube containing only the two dimensions *location* and *time*. Roll-up may be performed by removing, say, the *time* dimension, resulting in an aggregation of the total sales by location, rather than by location and by time.

Drill-down: Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either *stepping down a concept hierarchy* for a dimension or *introducing additional dimensions*. Figure shows the result of a drill-down operation performed on the central cube by stepping down a concept hierarchy for *time* defined as "*day < month < quarter < year*." Drill-down occurs by descending the *time* hierarchy from the level of *quarter* to the more detailed level of *month*. The resulting data cube details the total sales per month rather than summarizing them by quarter. Because a drill-down adds more detail to the given data, it can also be performed by adding new dimensions to a cube. For example, a drill-down on the central cube of Figure can occur by introducing an additional dimension, such as *customer group*.

Slice and dice: The *slice* operation performs a selection on one dimension of the given cube, resulting in a subcube. Figure shows a slice operation where the sales data are selected from the central cube for the dimension *time* using the criterion *time = "Q1"*. The *dice* operation defines a subcube by performing a selection on two or more dimensions. Figure shows a dice operation on the central cube based on the following selection criteria that involve three dimensions: (*location = "Toronto"* or *"Vancouver"*) and (*time = "Q1"* or *"Q2"*) and (*item ="home entertainment"* or *"computer"*).

Pivot (rotate): *Pivot* (also called *rotate*) is a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data. Figure shows a pivot operation where the *item* and *location* axes in a 2-D slice are rotated. Other examples include rotating the axes in a 3-D cube, or transforming a 3-D cube into a series of 2-D planes.

Other OLAP operations: Some OLAP systems offer additional drilling operations. For example, drill-across executes queries involving (i.e., across) more than one fact table. The drill-through operation uses relational SQL facilities to drill through the bottom level of a data cube down to its back-end relational tables.

Other OLAP operations may include ranking the top *N* or bottom*N* items in lists, as well as computing moving averages, growth rates, and interests, internal rates of return, depreciation, currency conversions, and statistical functions. OLAP offers analytical modelling capabilities, including a calculation engine for deriving ratios, variance, and so on, and for computing measures across multiple dimensions. It can generate summarizations, aggregations, and hierarchies at each granularity level and at every dimension intersection. OLAP also supports functional models for forecasting, trend analysis, and statistical analysis. In this context, an OLAP engine is a powerful data analysis tool.

b) Explain the concept of constrain based association mining.

(Description - 4 Marks, Types – 4 Marks)

Ans:

A data mining process may uncover thousands of rules from a given set of data, most of which end up being unrelated or uninteresting to the users. Often, users have a good sense of which "direction" of mining may lead to interesting patterns and the "form" of the patterns or rules they would like to find. Thus, a good heuristic is to have the users specify such intuition or expectations as *constraints* to confine the search space. This strategy is known as constraint-based mining. The constraints can include the following:

Knowledge type constraints: These specify the type of knowledge to be mined, such as association or correlation.

Data constraints: These specify the set of task-relevant data.

Dimension/level constraints: These specify the desired dimensions (or attributes) of the data, or levels of the concept hierarchies, to be used in mining.

Interestingness constraints: These specify thresholds on statistical measures of rule interestingness, such as support, confidence, and correlation.

Rule constraints: These specify the form of rules to be mined. Such constraints may be expressed as Meta rules (rule templates), as the maximum or minimum number of predicates that can occur in the rule antecedent or consequent, or as relationships among attributes, attribute values, and/or aggregates.

The above constraints can be specified using a high-level declarative data mining query language and user interface. The first four of the above types of constraints have already been addressed in earlier parts of this book and chapter. In this section, we discuss the use of *rule constraints* to focus the mining task. This form of constraint-based mining allows users to describe the rules that they would like to uncover, thereby making the data mining process more *effective*. In addition, a sophisticated mining query optimizer can be used to exploit the constraints specified by the user, thereby making the mining process more *efficient*. Constraint-based mining encourages interactive exploratory mining and analysis.

c) Explain application of knowledge discovery in fraud detection.

(Explanation – 8 Marks)

Ans:

Traditional methods of data analysis have long been used to detect fraud. They require complex and time-consuming investigations that deal with different domains of knowledge like financial, economics, business practices and law. Fraud often consists of many instances or incidents involving repeated transgressions using the same method. Fraud instances can be similar in content and appearance but usually are not identical.

The first industries to use data analysis techniques to prevent fraud were the telephony companies, the insurance companies and the banks (Decker 1998). One early example of successful implementation of data analysis techniques in the banking industry is the FICO Falcon fraud assessment system, which is based on a neural network shell.

Retail industries also suffer from fraud at POS. Some supermarkets have started to make use of digitized closed-circuit television (CCTV) together with POS data of most susceptible transactions to fraud.

Internet transactions have recently raised big concerns, with some research showing that internet transaction fraud is 12 times higher than in-store fraud.

Fraud that involves cell phones, insurance claims, tax return claims, credit card transactions etc. represent significant problems for governments and businesses, but yet detecting and preventing fraud is not a simple task. Fraud is an adaptive crime, so it needs special methods of intelligent data analysis to detect and prevent it. These methods exist in the areas of Knowledge Discovery in Databases (KDD), Data Mining, Machine Learning and Statistics. They offer applicable and successful solutions in different areas of fraud crimes.

Techniques used for fraud detection fall into two primary classes: statistical techniques and artificial intelligence. Examples of statistical data analysis techniques are:

- Data preprocessing techniques for detection, validation, error correction, and filling up of missing or incorrect data.
- Calculation of various statistical parameters such as averages, quantiles, performance metrics, probability distributions, and so on. For example, the averages may include average length of call, average number of calls per month and average delays in bill payment.
- Models and probability distributions of various business activities either in terms of various parameters or probability distributions.
- Computing user profiles.
- Time-series analysis of time-dependent data.
- Clustering and classification to find patterns and associations among groups of data.

- Matching algorithms to detect anomalies in the behavior of transactions or users as compared to previously known models and profiles. Techniques are also needed to eliminate false alarms, estimate risks, and predict future of current transactions or users.

Some forensic accountants specialize in forensic analytics which is the procurement and analysis of electronic data to reconstruct, detect, or otherwise support a claim of financial fraud. The main steps in forensic analytics are (a) data collection, (b) data preparation, (c) data analysis, and (d) reporting. For example, forensic analytics may be used to review an employee's purchasing card activity to assess whether any of the purchases were diverted or divertible for personal use. Forensic analytics might be used to review the invoicing activity for a vendor to identify fictitious vendors, and these techniques might also be used by a franchisor to detect fraudulent or erroneous sales reports by the franchisee in a franchising environment.

Fraud management is a knowledge-intensive activity. The main AI techniques used for fraud management include:

- Data mining to classify, cluster, and segment the data and automatically find associations and rules in the data that may signify interesting patterns, including those related to fraud.
- Expert systems to encode expertise for detecting fraud in the form of rules.
- Pattern recognition to detect approximate classes, clusters, or patterns of suspicious behavior either automatically (unsupervised) or to match given inputs.
- Machine learning techniques to automatically identify characteristics of fraud.
- Neural networks that can learn suspicious patterns from samples and used later to detect them.
- Other techniques such as link analysis, Bayesian networks, decision theory, land sequence matching are also used for fraud detection.

Machine learning and data mining

Early data analysis techniques were oriented toward extracting quantitative and statistical data characteristics. These techniques facilitate useful data interpretations and can help to get better insights into the processes behind the data. Although the traditional data analysis techniques can indirectly lead us to knowledge, it is still created by human analysts.

To go beyond, a data analysis system has to be equipped with a substantial amount of background knowledge, and be able to perform reasoning tasks involving that knowledge and the data provided. In effort to meet this goal, researchers have turned to ideas from the machine learning field. This is a natural source of ideas, since the machine learning task can be described as turning background knowledge and examples (input) into knowledge (output).

If data mining results in discovering meaningful patterns, data turns into information. Information or patterns that are novel, valid and potentially useful are not merely information, but knowledge. One speaks of discovering knowledge, before hidden in the huge amount of data, but now revealed.

Supervised and unsupervised learning

The machine learning and artificial intelligence solutions may be classified into two categories: 'supervised' and 'unsupervised' learning. These methods seek for accounts, customers, suppliers,

etc. that behave 'unusually' in order to output suspicion scores, rules or visual anomalies, depending on the method.

Whether supervised or unsupervised methods are used, note that the output gives us only an indication of fraud likelihood. No stand alone statistical analysis can assure that a particular object is a fraudulent one. It can only indicate that this object is more likely to be fraudulent than other objects.

Supervised methods

In supervised learning, a random sub-sample of all records is taken and manually classified as either 'fraudulent' or 'non-fraudulent'. Relatively rare events such as fraud may need to be over sampled to get a big enough sample size. These manually classified records are then used to train a supervised machine learning algorithm. After building a model using this training data, the algorithm should be able to classify new records as either fraudulent or non-fraudulent.

Supervised neural networks, fuzzy neural nets, and combinations of neural nets and rules, have been extensively explored and used for detecting fraud in mobile phone networks and financial statement fraud.

Bayesian learning neural network is implemented for credit card fraud detection, telecommunications fraud, auto claim fraud detection, and medical insurance fraud.

Hybrid knowledge/statistical-based systems, where expert knowledge is integrated with statistical power, use a series of data mining techniques for the purpose of detecting cellular clone fraud. Specifically, a rule-learning program to uncover indicators of fraudulent behaviour from a large database of customer transactions is implemented

Cahill et al. (2000) design a fraud signature, based on data of fraudulent calls, to detect telecommunications fraud. For scoring a call for fraud its probability under the account signature is compared to its probability under a fraud signature. The fraud signature is updated sequentially, enabling event-driven fraud detection.

Link analysis comprehends a different approach. It relates known fraudsters to other individuals, using record linkage and social network methods.

This type of detection is only able to detect frauds similar to those which have occurred previously and been classified by a human. To detect a novel type of fraud may require the use of an unsupervised machine learning algorithm.

Unsupervised methods

In contrast, unsupervised methods don't make use of labelled records.

Some important studies with unsupervised learning with respect to fraud detection should be mentioned. For example, Bolton and Hand use Peer Group Analysis and Break Point Analysis applied on spending behaviour in credit card accounts. Peer Group Analysis detects individual objects that begin to behave in a way different from objects to which they had previously been similar. Another tool Bolton and Hand develop for behavioural fraud detection is Break Point Analysis. Unlike Peer Group Analysis, Break Point Analysis operates on the account level. A

break point is an observation where anomalous behaviour for a particular account is detected. Both the tools are applied on spending behaviour in credit card accounts.

Also Murad and Pinkas focus on behavioural changes for the purpose of fraud detection and present three-level-profiling. Three-level-profiling method operates at the account level and points to any significant deviation from an account's normal behaviour as a potential fraud. In order to do this, 'normal' profiles are created based on data without fraudulent records (semi supervised). In the same field, also Burge and Shawe-Taylor use behaviour profiling for the purpose of fraud detection. However, using a recurrent neural network for prototyping calling behavior, unsupervised learning is applied.

Cox et al. combines human pattern recognition skills with automated data algorithms. In their work, information is presented visually by domain-specific interfaces, combining human pattern recognition skills with automated data algorithms (Jans et al.).

6. Attempt any four of the following: 16

a) State association rule in data mining. Write application of each rule.

(Statement of Rule – 2 Marks, Application - 2 Marks)

Ans:

Similar to the mining of association rules in transactional and relational databases, *spatial association rules* can be mined in spatial databases.
A spatial association rule is of the form
***A)B* [*s%;c%*]** where *A* and *B* are sets of spatial or nonspatial predicates, *s%* is the support of the rule, and *c%* is the confidence of the rule.
For example, the following is a spatial association rule:
is a(*X*; "*school*")^*close to*(*X*; "*sports center*"))*close to*(*X*; "*park*") [0:5%;80%].
This rule states that 80% of schools that are close to sports centers are also close to parks, and 0.5% of the data belongs to such a case. Various kinds of spatial predicates can constitute a spatial association rule. Examples include distance information (such as *close to* and *far away*), topological relations (like *intersect, overlap*, and *disjoint*), and spatial orientations (like *left of* and *west of*).
Since spatial association mining needs to evaluate multiple spatial relationships among a large number of spatial objects, the process could be quite costly. An interesting mining optimization method called progressive refinement can be adopted in spatial association analysis. The method first mines large data sets *roughly* using a fast algorithm and then improves the quality of mining in a pruned data set using a more expensive algorithm. To ensure that the pruned data set covers the complete set of answers when applying the high-quality data mining algorithms at a later stage, an important requirement for the rough mining algorithm applied in the early stage is the superset coverage property: that is, it preserves all of the potential answers. In other words, it

should allow a *false-positive test*, which might include some data sets that do not belong to the answer sets, but it should not allow a *false-negative test*, which might exclude some potential answers. For mining spatial associations related to the spatial predicate *close to*, we can first collect the candidates that pass the minimum support threshold by Applying certain rough spatial evaluation algorithms, for example, using an MBR structure (which registers only two spatial points rather than a set of complex polygons), and Evaluating the relaxed spatial predicate, *g close to*, which is a generalized *close to* covering a broader context that includes *close to, touch*, and *intersect*. If two spatial objects are closely located, their enclosing MBRs must be closely located, matching *g close to*. However, the reverse is not always true: if the enclosing MBRs are closely located, the two spatial objects may or may not be located so closely. Thus, the MBR pruning is a false-positive testing tool for closeness: only those that pass the *rough* test need to be further examined using more expensive spatial computation algorithms. With this preprocessing, only the patterns that are frequent at the approximation level will need to be examined by more detailed and finer, yet more expensive, spatial computation. Besides mining spatial association rules, one may like to identify groups of particular features that appear frequently close to each other in a geospatial map. Such a problem is essentially the problem of mining spatial co-locations. Finding spatial co-locations can be considered as a special case of mining spatial associations. However, based on the property of spatial autocorrelation, interesting features likely coexist in closely located regions. Thus spatial co-location can be just what one really wants to explore. Efficient methods can be developed for mining spatial co-locations by exploring the methodologies like Aprori and progressive refinement, similar to what has been done for mining spatial association rules.

Mining Associations in Multimedia Data

Association rules involving multimedia objects can be mined in image and video databases. At least three categories can be observed:

Associations between image content and non-image content features: A rule like "*If at least 50% of the upper part of the picture is blue, then it is likely to represent sky*" belongs to this category since it links the image content to the keyword *sky*. Associations among image contents that are not related to spatial relationships: A rule like "*If a picture contains two blue squares, then it is likely to contain one red circle As well* belongs to this category since the associations are all regarding image contents. Associations among image contents related to spatial relationships: A rule like "*If a red triangle is between two yellow squares, then it is likely a big oval-shaped object is underneath*" belongs to this category since it associates objects in the image with spatial relationships.

b) Define metadata and classify metadata into technical and business metadata.

(Definition – 1 Mark, Classification - 3 Marks)

Ans:

Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects. Figure 3.12 showed a metadata repository within the bottom tier of the data warehousing architecture. Metadata are created for the data names and definitions of the given

warehouse. Additional metadata are created and captured for time stamping any extracted data, the source of the extracted data, and missing fields that have been added by data cleaning or integration processes.

A metadata repository should contain the following:

- A description of *the structure of the data warehouse*, which includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents
- *Operational metadata*, which include data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived, or purged), and monitoring information (warehouse usage statistics, error reports, and audit trails)
- *The algorithms used for summarization*, which include measure and dimension definition
- algorithms, data on granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports
- *The mapping from the operational environment to the data warehouse*, which includes source databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control)
- *Data related to system performance*, which include indices and profiles that improve data access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles
- *Business metadata*, which include business terms and definitions, data ownership information, and charging policies

A data warehouse contains different levels of summarization, of which metadata is one type. Other types include current detailed data (which are almost always on disk), older detailed data (which are usually on tertiary storage), lightly summarized data and highly summarized data (which may or may not be physically housed). Metadata play a very different role than other data warehouse data and are important for many reasons. For example, metadata are used as a directory to help the decision support system analyst locate the contents of the data warehouse, as a guide to the mapping of data when the data are transformed from the operational environment to the data warehouse environment, and as a guide to the algorithms used for summarization between the current detailed data and the lightly summarized data, and between the lightly summarized data and the highly summarized data. Metadata should be stored and managed persistently (i.e., on disk).

c) State the meaning of mining text database.

(Explanation – 4 Marks)

Ans:

A substantial portion of the available information is stored in text databases (or document databases), which consist of large collections of documents from various sources, such as news articles, research papers, books, digital libraries, e-mail messages, and Web pages. Text databases are rapidly growing due to the increasing amount of information available in electronic form, such as electronic publications, various kinds of electronic documents, e-mail, and the World Wide Web (which can also be viewed as a huge, interconnected, dynamic text database). Nowadays most of the information in government, industry, business, and other institutions are stored electronically, in the form of text databases. Data stored in most text databases are *semistructured data* in that they are neither completely unstructured nor completely structured. For example, a document may contain structured fields, such as *title*, *authors*, *publication date*, *and category*, and so on, but also contain some largely unstructured text components, such as *abstract* and *contents*. There have been a great deal of studies on the modelling and implementation of semistructured data in recent database research. Moreover, information retrieval techniques, such as text indexing methods, have been developed to handle unstructured documents. Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data. Typically, only a small fraction of the many available documents will be relevant to a given individual user. Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users need tools to compare different documents, rank the importance and relevance of the documents, or find patterns and trends across multiple documents. Thus, text mining has become an increasingly popular and essential theme in data mining.

Text Mining Approaches:

There are many approaches to text mining, which can be classified from different perspectives, based on the inputs taken in the text mining system and the data mining tasks to be performed. In general, the major approaches, based on the kinds of data they take as input, are:

1. the keyword-based approach, where the input is a set of keywords or terms in the documents,
2. the tagging approach, where the input is a set of tags, and
3. The information-extraction approach, which inputs semantic information, such as events, facts, or entities uncovered by information extraction.

d) Describe the concept of sequential mining.

(Concept – 2 Marks, Explanation – 2 Marks)

Ans:

A sequence database consists of sequences of ordered elements or events, recorded with or without a concrete notion of time. There are many applications involving sequence data. Typical examples include customer shopping sequences, Web click streams, biological sequences, sequences of events in science and engineering, and in natural and social developments. In this section, we study *sequential pattern mining* in transactional databases.

Concepts and Primitives:

Sequential pattern mining is the mining of frequently occurring ordered events or subsequences as patterns. An example of a sequential pattern is "*Customers who buy a Canon digital camera are likely to buy an HP color printer within a month*." For retail data, sequential patterns are useful for shelf placement and promotions. This industry, as well as telecommunications and other businesses, may also use sequential patterns for targeted marketing, customer retention, and many other tasks. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, production processes, and network intrusion detection. Most of the studies of sequential pattern mining concentrate on *categorical* (or *symbolic*) *patterns*, whereas numerical curve analysis usually belongs to the scope of trend analysis and forecasting in statistical time-series analysis.

The sequential pattern mining problem was first introduced by Agrawal and Srikant in 1995 [AS95] based on their study of customer purchase sequences, as follows:

"*Given a set of sequences, where each sequence consists of a list of events (or elements) and each event consists of a set of items, and given a user-specified minimum support threshold of min sup, sequential pattern mining finds all frequent subsequences, that is, the subsequences whose occurrence frequency in the set of sequences is no less than min sup.*"

Let I = f*I*1, *I*2, : : : , *Ip*g be the set of all *items*.

An itemset is a nonempty set of items. A sequence is an ordered list of events. A sequence *s* is denoted h*e*1*e*2*e*3 _ _ _*el*i, where event *e*1 occurs before *e*2, which occurs before *e*3, and so on. Event *e j* is also called an element of *s*. In the case of customer purchase data, an event refers to a shopping trip in which a customer bought items at a certain store. The event is thus an itemset, that is, an unordered list of items that the customer purchased during the trip. The itemset (or event) is denoted (*x*1*x*2 _ _ _*xq*), where *xk* is an item. For brevity, the brackets are omitted if an element has only one item, that is, element (*x*) is written as *x*. Suppose that a customer made several shopping trips to the store. These ordered events form a sequence for the customer. That is, the customer first bought the items in *s*1, then later bought the items in *s*2, and so on. An item can occur at most once in an event of a sequence, but can occur multiple times in different events of a sequence. The number of instances of items in a sequence is called the length of the sequence. A sequence with length *l* is called an *l*-sequence. A sequence a = h*a*1*a*2 _ _ _*an*i is called a subsequence of another sequence b = h*b*1*b*2 _ _ _*bm*i, and b is a supersequence of a, denoted as a v b, if there exist integers 1 _ *j*1 < *j*2 < _ _ _ < *jn* _ *m* such that *a*1 _ *bj*1 , *a*2 _ *bj*2 , . .

. , *an _ bjn* . For example, if a = h(*ab*), *d*i and b = h(*abc*), (*de*)i, where *a*, *b*, *c*, *d*, and *e* are items, then a is a subsequence of b and b is a supersequence of a. A sequence database, *S*, is a set of tuples, h*SID*, *s*i, where *SID* is a *sequence ID* and *s* is a sequence. For our example, *S* contains sequences for all customers of the store. A tuple h*SID*, *s*i is said to contain a sequence a, if a is a subsequence of *s*. The support of a sequence a in a sequence database *S* is the number of tuples in the database containing a, that is, *supportS*(a) = j fh*SID*, *s*ij(h*SID*, *s*i 2 *S*)^(a v *s*)g j. It can be denoted as *support* (a) if the sequence database is clear from the context. Given a positive integer *min sup* as the minimum support threshold, a sequence a is frequent in sequence database *S* if *supportS*(a)_*min sup*. That is, for sequence a to be frequent, it must occur at least *min sup* times in *S*. *A frequent sequence is called a* sequential pattern. A sequential pattern with length *l* is called an *l*-pattern.

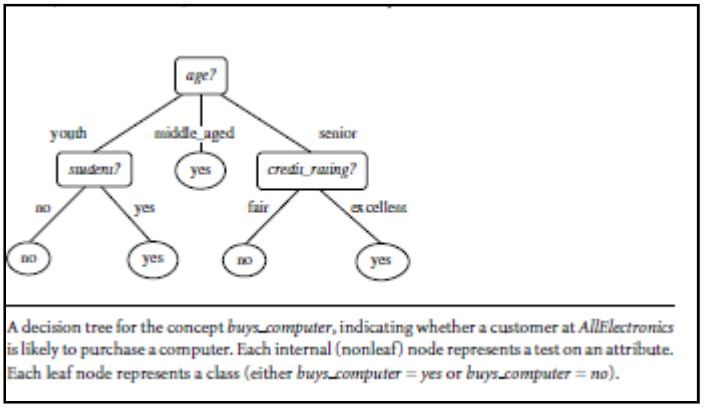
e) Describe data classification by decision tree induction

(Diagram – 1 Mark, Example – 1 Mark, Explanation – 2 Marks)

Ans:

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or *terminal node*) holds a class label. The topmost node in a tree is the root node.

A typical decision tree is shown in below Figure:



A decision tree for the concept *buys_computer*, indicating whether a customer at *AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys_computer = yes* or *buys_computer = no*).

It represents the concept *buys computer*, that is, it predicts whether a customer at *All Electronics* is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals. Some decision tree algorithms produce only *binary* trees (where each internal node branches to exactly two other nodes), whereas others can produce nonbinary trees. Given a tuple, ***X***, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class

prediction for that tuple. Decision trees can easily be converted to classification rules. The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans. The learning and classification steps of decision tree induction are simple and fast. In general, decision tree classifiers have good accuracy. However, successful use may depend on the data at hand. Decision tree induction algorithms have been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology. Decision trees are the basis of several commercial rule induction systems.

During tree construction, *attribute selection measures* are used to select the attribute that best partitions the tuples into distinct classes. When decision trees are built, many of the branches may reflect noise or outliers in the training data. *Tree pruning* attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data.