CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Temporal Data Mining (TDM), a vital extension of data mining is a non-trivial extraction of inherent and unrecorded information from large databases. TDM is anxious with the temporal data analysis and for temporal patterns finding. The TDM is used in finding the regularities in the sets of temporal data. TDM is highly inclined to the areas of temporal databases and temporal reasoning. Many applications of the database technology are temporal in nature. Some of the samples for temporal databases are portfolio management, accounting, weather monitoring, etc. Most of the TDM techniques transform the temporal data into static representation and exploit existing static machine learning techniques.

TDM permits the computer-driven and the automatic exploration of the data. The TDM provides a new way of interacting with the temporal databases by mentioning the questions at the abstract level called Temporal Structured Query Language (TSQL). The analysis process starts with a set of temporal data and uses a methodology to develop an optimal representation of the data structure during which time knowledge is reached. The final goal of the temporal data mining is to discover the hidden relations between the sequences and subsequences of events. The recognition of relations between sequences of events involves three steps:

1. Modeling and Representation- In this sequence, the temporal data is converted into an appropriate form
2. Similarity Measure- Measures between sequences
3. Mining operation- Application of models and demonstrations to definite mining problems.

1.1.1 Temporal Database

A temporal database is a database with built-in support for handling data involving time, being related to a slowly changing dimension concept, for example a temporal data model and a temporal version of the Structured Query Language (SQL). A database, which is temporal in nature, is modeled by compiling and storing temporal data. The difference between the non-temporal data and temporal data is the time period added to data expression when it was valid or stored in the database. The data stored in the traditional databases considers data to be valid at present as in the time instance “now”. When the data is altered, removed or inserted in the database, the state of the database is overwritten to form a new state.

The state of the database is subject to any changes and no longer available. Thus, databases are likely to store the states, which is associated with the time. Temporal databases encompass all DB applications that need some aspect of time when arranging their information. This demonstrate the need for developing a set of unifying concepts for application developers to use. There are many examples of applications where some aspect of time is needed to maintain the information in a DB, like:

- Health care- patient history is maintained
- Insurance-claims and accident history which are required
- Finance-stock price history is maintained
- Personnel management- salary and position history are maintained
- Banking- credit history is stored.
Different form of Temporal Databases

Time is construed as a valid time (when data occurred or is true in reality) or transaction time (when data were entered into the database).

- Historical database stores the valid time data
- Rollback databases store the transaction time data
- Bi-temporal database stores both the valid and the transaction time data and the history of the data with respect to valid time and transaction time.

Temporal Queries

Temporal databases are of first order structures and extends the first order logic of the temporal extensions. The temporal databases use the first order query language.

Temporal Logic

Temporal logic is the variant of the modal logic used to express the statements whose truth is relative to an underlying time domain. The time domain is commonly a linearly ordered set of time instants. The natural language statements are the expressed modalities in such forms as:

- Sometime in the future
- Always in the future, etc.

Time connectives are words or phrases which are used to tell a reader WHEN something is happening. They are sometimes called temporal connectives. The temporal connectives are used for capturing the syntax of the temporal logic. The temporal logic is rooted in the propositional logic for the purpose of querying. The following are used for querying, i.e.
- Linear-Time First Order Temporal Logic (FOTL)
- Point-Stamped temporal databases
- Extension of first order logic with temporal connectives

The FOTL queries are the point-stamped temporal relations which is for valid time.

**Temporal connectives**

The temporal connectives are listed below:

- $\Diamond$: is used to define sometime in the future
- $\Box$: is used to define always in the future
- $\blacklozenge$: is used to define sometime in the past
- $\blacksquare$: is used to define always in the past

For discrete linear order, the ‘$\bigcirc$’ operator is used to represent the next data and the ‘$\bullet$’ operator is used to represent the previous data. The connectives ‘$\blacklozenge$’, ‘$\blacksquare$’ and ‘$\bullet$’ are called the past temporal connectives and symbols ‘$\bigcirc$’, ‘$\Diamond$’ and ‘$\Box$’ are used for representing the future temporal connectives. These symbols are used to represent until and the since condition.

### 1.1.2 Temporal Data

Temporal data is simply a data that signifies a state in time and are formed by time-stamping ordinary data. In the relational data model, the time-stamped tuples are used and in an object-oriented data model, the object/attributes are time-stamped. Each regular data has two time values attached to it, a start time and an end time, which is used to establish the time interval of the data. Temporal data is collected to analyze weather patterns and
other environmental variables, monitor traffic conditions, demographic trends and so on. The various types of data are termed as follows:

**Static Data**- This type of temporal data has zero temporality i.e. static data are free from any temporal reference. Inferences that can be derived from this data are free from any temporality. A state or an instance of a database is its current contents, which does not certainly reflect the current status of the real world. Updating the state of the database is carried out using data manipulation operations such as insertion, deletion, or replacement. In this process, the past states of the database and the database of the real world are rejected and forgotten completely.

**Sequences**- Sequences are well-ordered events or transactions. Though there may not be any explicit reference to time, there occurs a sort of qualitative temporal relationship between data items. In any transactional data if a transaction appears before another, it suggests that the former transaction has occurred before the latter. This type of temporal data has a temporal relationship like before, or after, or during the overlap, etc. Such relationships are called as the qualitative relationship between the time events.

**Time Stamped**- This group of the temporal data has an explicit time related information. Relation van is quantitative, i.e. one can find the exact temporal distance between data element. The inference made through this type of data may be temporal or non-temporal.

**Time Series**- This time series data is an extraordinary case of the time stamped data. The time series data events are uniform on the time scale.

**Fully Temporal**- The data in this type of category is dependent on the time fully. The inferences in the temporal database are strictly temporal.
1.1.3 **Temporal dimensions**

Temporal dimension can be defined as valid time describing the fact or information is true in the real world. It also includes transaction time which affirms that the fact or information is current and occurs in the database management system (DBMS). Temporal dimensions include three aspects:

*Snapshot*- It can be defined as the state of the system at a particular point in time. These databases support the functionality to access a single state of the real world. The only time-variant attributes in the snapshot are defined by the user which means it has no specific language to support. It can be used for attributes such as “hiring date”, etc.

*Valid time*- This relates to the time of an event takes place in the world. It is independent of its storage in a database and also concentrates on events that occurred in the past, present and future.

*Transaction time*- The transaction time of an event describes the times during which the event is stored in a database and illustrates the correct image of the modelled world. Time stamps of the transaction time events are defined in the schedule which is accepted by the operating system.

1.2 **CONTENT BASED IMAGE RETRIEVAL (CBIR)**

In CBIR, the retrieval of an image is based on the contents and queries of the image. Using the color and size features of the images in the database, the extraction process can be automated. Though the semantic extraction is automatic, the exact images are not correctly identified. The functionality of CBIR are extraction facility, feature storage and similarity measure. The CBIR is also known as the Query by Image Content (QBIC) which is used for retrieving the relevant image based on their contents. The images are retrieved from the
large database using the unique descriptors from the trained image. Figure 1.1 shows the general diagram of CBIR using the HSV colour model. The extraction facility extracts the appropriate information from the input data.

![Figure 1.1 CBIR using HSV colour Model](image)

The feature storage offers an optimal storage for the extracted information. The similarity measure evaluates the difference between the images. Generally, the following visual features are used for the image retrieval (2014) and is depicted in Figure 1.2.

![Figure 1.2 Features Used for Visualizing](image)

- Colour Descriptor
- Texture Descriptor
- Shape Descriptor
- Edge Descriptor
- Neural Network
- Feed Forward Back Propagation Neural Network (FFBP NN)

**Colour Descriptor**

The colour of the image is an important feature that depicts most of the information about it. The Red, Green, Blue (RGB) colour model is not mostly preferred by humans because it does not provide a comfortable colour visualization. Further, it does not segregate the luminance component from the chrominance component. In case of the Hue Saturation Value (HSV) colour model, the visual perception is easy, hence most of the image retrieval system uses the HSV colour model. The various colours are distinguished by the hue component. The saturation denotes the amount of white light that should be added to the pure colour and the value represents the light intensity.

The information about the image is obtained from the pixels in the images. The colour histograms are also used as the colour descriptors in the digital images. It represents the number of coloured pixels in the entire image colour space. Generally the histogram of the image is calculated in two steps. During step 1, the images from the database is read, then only the RGB pixels from the image is extracted. During step 2, the normalized histograms are created for the each RGB component of the images. Hence, each image have three histograms.

There are several colours used for the retrieval of image and is listed as follows and is based on some colour models:

- RGB colour model
- HSV colour model
• Colour Co-occurrence matrix (CCM)

**RGB Colour model**

The RGB colour model has the combination of three Colours such as Red, Green and Blue. The monitor use the CRT and the raster graphics use the RGB colour model. The Cartesian co-ordinate system is used by the RGB colour model and the colours are called the additive primitives which is formed by the adding the primary colours together.

**HSV Colour Model**

The H, S and V stands for the Hue, Saturation and Value model, which describes colours in terms of the shade and the brightness. It shows the relationship between the colours and basically a colour model is the specification in the single point.

**Hue**

The Hue is the dominant wavelength in the light. The Hue colour is in terms of pure spectrum colours and is expressed in the range of $0^\circ$ to $360^\circ$.

**Saturation**

The dominance in the hue colour is the Saturation, and it is also thought as the intensity in the colour spectrum and is used to define the purity of the colour. The highly saturated colour is the vivid and the low saturated colour is muted and if there is no saturation then the image is said to be grey image.
**Value**

The brightness or the value of the intensity is described by the value parameter. The relative lightness or the darkness of the colour is defined by the value.

**Colour Co-occurrence Matrix (CCM)**

A Colour co-occurrence matrix (CCM) is a matrix which is defined over an image for distributing the co-occurring values at a given offset. The values are taken as the values of the Hue, Saturation and the values of a specified level. The texture of the image is mainly analyzed by the CCM. The features generated using this CCM is called the Haralick Features. The query image is selected for the retrieval process and the hue, saturation and the values of the pixels are taken into account and the CCM is formed using the formula below:

\[
\text{Colour Co-occurrence matrix} = 9H + 3S + V \ldots \tag{1.1}
\]

Where, H, S and V is the Hue, Saturation and Value respectively. The values of the hue range from 20 to 316, saturation values range between 0 and 1 and the value ranges from 0 and 1.

**Texture Descriptor**

The texture descriptor describes the surface information of an image and also illustrates the image relationship with the environment. Generally, the Gray Level Co-occurrence Matrix (GLCM) is used to represent the texture information of the images because it is similar to the features of the human visualization system. The features that are extracted using the GLCM are entropy, correlation, energy, etc. The other commonly used texture descriptors are wavelets, Fourier transforms, entropy, correlation methods, etc. Among these
descriptors, the wavelet is very complex. In wavelets, the wavelet is sampled into multiple sub bands.

**Shape Descriptor**

The shapes of the digital images are extracted using multiple methods such as contour based shape extraction, region based shape extraction, boundary based methods and Generalized Hough Transform (GHT). Among these techniques, GHT is mostly preferred as the shape extraction technique. It provides the complete information about the object shape. The performance of GHT is not affected by the noise and shape deformalities.

**Edge Descriptor**

The edge of the images are more sensitive to the human eye. The histogram of the image depicts the edge features including the frequency and the directionality of brightness change in the image.

**Neural Network (NN)**

The neural network is composed of multiple nodes. It is used in multiple applications such as classification, optimization, control theory, etc. Since the classification problems demand complex detection and recognition, the neural network is mostly preferred. When compared to the other techniques, the NN is dynamic. The optimal output is obtained by adjusting the weights of the final output and input data. The learning is the other name of the weight adjustment process.

**Feed Forward Back Propagation Neural Network (FFBP NN)**

The commonly used technique for the pattern recognition and classification is the FFBP NN technique. It is a multi-layer neural network used
for deploying the nonlinear differentiable functions. It contains three layers i.e., input layer, hidden layer and output layer. The main merit of the FFBP NN is the movement in both the onward and backward direction. The output is computed in the forward procession and the errors are computed in the backward procession. In this research the neural network based image retrieval is used for the feature recognition and learning the similarity measure.

1.3 TEMPORAL DATA MINING TECHNIQUES

TDM is a single step process of knowledge discovery in temporal databases which computes the structures such as temporal patterns or models over the temporal data. The algorithm which extracts the temporal patterns or fits models to temporal data is called the TDM algorithm. The temporal data mining tasks defines the relevant or important information to apply the data mining techniques on a temporal database. TDM involves several tasks which are described below and depicted in Figure 1.3

![Figure 1.3 Temporal Data Mining Techniques](image)

Figure 1.3 Temporal Data Mining Techniques
Temporal data characterization and comparison

Temporal cluster analysis

Temporal Classification

Temporal association rules

Temporal pattern analysis

Temporal prediction and analysis of trend

Temporal search and Retrieval

The temporal data model which supports the time granularity and the time hierarchies is developed based on the two criteria:

- Temporal data structures
- Temporal Semantics

**Temporal data characterization and comparison**

The characterization is the summarization of common characteristics or feature of a target class of data. A characterization is extended to temporal data. The output of characterization is represented in several forms, e.g. pie charts, bar graph, curves and multidimensional data cubes. The decision tree is constructed based on the temporal attributes (for example; a rule). The characterization outputs are represented in various forms listed below:

- Pie charts
- Bar charts
- Curves
- Multidimensional data cubes
Temporal clustering analysis

Clustering methods are used to split the data in the group on the basis of similarity measure. The clustering series or time series is concerned with grouping a collection of time series or sequences based on their resemblance. Temporal gathering includes two approaches, temporal similarity and temporal optimal partition. Clustering in TDM offers a good mechanism to automatically find some structure in large data sets. There are many applications where a time series clustering activity is same e.g. web activity logs where clusters indicate the navigation patterns of several user groups. Temporal clustering targets splitting the temporal data into subsections that are comparable to one another.

Temporal classification

The temporal classification is used for forecasting the temporally related fields in a temporal database. In the classification stage one classifies the unknown set of attributes are classified in any one of the predefined classes. In the temporal classification stage, each temporal sequence implemented in the database is assumed to belong to one of the predefined classes or categories and the goal is to automatically detect the relevant category/class for the given input temporal set of attributes.

Temporal association rules

The detection of relevant association rule is one of the most important methods used to perform the data mining task on the transactional databases. An effective algorithm used to determine the association rule is apriori. In association rule finding, the relation between the attribute on the basis of support and confidence is extracted. The manipulation of temporal sequence needs some variation to be made to the apriori algorithm. The unknown set of attributes of
the predefined class are defined using the classification in temporal data mining. The temporal sequence present in the database belongs to one of the predefined classes in the classification of the temporal data mining. The examples of the classification applications are listed below and is shown in Figure 1.4

- Handwriting recognition
- Speech Recognition
- Gesture recognition
- Demarcating gene and non-gene regions
- Online signature verification, etc.

**Figure 1.4 Applications of Classification Algorithms**

**Temporal pattern analysis**

Unlike in search and retrieval applications, in pattern discovery there is no specific query at hand to locate for the database. A temporal pattern is basically a combination of events where the events occur in the same order with
the consecutive time distance between the successive pattern components which is relatively invariant with respect to an expectation.

**Temporal prediction and analysis of trend**

The prediction in temporal data mining has a versatile significance which forecasting the future on the basis of the past. The future values of the time series is based on the past samples. In order to predict the future values in data mining, the predictive model is developed. In several cases, the prediction is formulated as the classification, association and the clustering problems. Trend analysis in temporal database is the change in attribute due to the change in time. The analysis of the time series of continuous data may show the same trends, i.e. the same shape across the time axis. The analysis of time series is very useful for the decision support system and decision maker. Temporal prediction has a versatile significance and forecasts the future which is based on its past samples. For this purpose a predictive model is built for the data. In many cases, prediction can be expressed as classification, association rule finding or clustering problems.

**Temporal search and Retrieval**

The searching was concerned with efficiently locating the subsequences often referred as the queries in large archives in a single long sequence. The query is presented to the system in the form of a sequence in the content based image retrieval system. If two sequences are given as input of equal length, then a measure of similarity is considered between the corresponding elements.
1.4 OBJECT ORIENTED DATA MODELS IN SQL

Object oriented analysis and design (OOAD) is a common method for examining and constructing a software system. SQL is the standard relational database definition and manipulation language. Extensions to SQL include the object oriented features. These features support an extension of the relational data model called the object-relational data model where includes all the outdated ideas found in the relational data model and some features of the object oriented data model. The object-relational extensions to SQL contains types of rows, types of collection, user defined types (UDT), tables, which are typed and type of references. This OOAD model does not support all the features of the object oriented data model. Instead, it indicates an evolutionary step from the relational data model to the object oriented data model.

1.5 SQL3 MINING IN TEMPORAL DATABASE

The SQL temporal certifies a smooth transition of legacy application code. Upward compatibility and temporal upward compatibility guarantees the legacy application code which does not require any alterations during migration. The requirements are persuaded for four levels of temporal functionality and is defined in SQL/Temporal,

Level 1- This is the lowest level and seizes the minimum functionality essential for the query language to satisfy the upward compatibility with SQL3. Thus, there is a provision for the legacy SQL3 statements, but there are no tables with valid time support and no temporal queries. The functionality at this level is same as that of SQL3.

Level 2- The level 2 adds to the previous level just by permitting for the presence of a table with valid-time support. The temporal upward
compatibility requirement is applicable to this subset of SQL/Temporal. The level 2 adds no new syntax for queries or modifications-only queries and alterations with SQL3 syntax are possible.

Level 3- The functionality of level 2 is enhanced with the opportunity of giving sequenced temporal functionality to queries, views, constraints, assertions and modifications on tables with valid-time support. The third level of functionality is likely to deliver suitable support for many applications. Starting at this level, temporal queries exist, so SQL/Temporal must be a sequenced consistent extension of SQL-3.

Level 4- Finally, the full temporal functionality is usually associated and is added to a language that is temporal which has non-sequenced temporal queries, assertions, constraints, views and alterations. These additional queries include temporal queries and modifications that have no syntactic counterpart in SQL3.

1.6 TIME SERIES IN TEMPORAL DATABASES

Time in temporal databases can be stated in many ways. It may be of interest to record the exact period when an event took place or the period in which an event takes place or the duration of an event or even the periodicity of an incident. The recording of the exact period permits the identification when the proofs are correct in the design and in realism or when the truth is currently in the database. Time can stamp either tuples or attributes in relations. This diversity has formed many different temporal database models over the last twenty years. The Time series is a record of the standards of any fluctuating quantity measured at the different points of time. Application of time series analysis techniques in temporal data mining is often called as time series data
mining. Analysis of time series method is applied in the following major categories of temporal data mining:

1. *Representation of temporal sequence*- This refers to the illustration of data before real temporal data mining takes place. There are two main classifications in temporal data mining:
   
   - *General representation of data*- Representation of data into time series data in either continuous or discontinuous, linear/nonlinear models, stationary/non-stationary models or distribution models.
   - *General transformation of data*- The demonstration of time series data is split into two either continuous or discontinuous transformations.

2. *Measure of Temporal sequence*- Measuring the temporal characteristic element in giving the definitions of resemblance and/or periodicity in a temporal sequence or between temporal sequences. There are two methods:
   
   - *Characteristic distance measurement in the time domain*- measuring the distance between temporal characteristics in either continuous or discontinuous time domain.
   - *Characteristic distance measuring in domain other than time*- computing the distance among the temporal characteristics in either continuous or discontinuous domain other than time.

3. *Prediction of Temporal Sequence*- The aim of the temporal sequence is to predict some fields in a database based on TIME domain. These methods are categorized into two models.
• **Temporal classification models** - the goal of the temporal classification model is to determine the state of a categorical variable (the class) in the TIME domain.

• **Temporal regression models** - the basic goal of regression model is to detect a numeric variable in a set by using different transformations on the databases to detect the temporal information of different categorical data sets.

In tuple time stamping, each row (tuples) is improved by two attributes used to record the start and end time point of the corresponding time interval corresponding to the data. It is regularly applied in temporal relational data models which means that the first normal form has to be preserved. Attribute time stamping can be defined as the time related to every attribute which varies in time. A history of the attribute is maintained for each time-varying attribute within each tuple. This results that the reduction of degree relation by one or two when compared with tuple time stamping.

1.7 **TEMPORAL VERSIONS OF RELATIONAL OPERATORS**

Seven terms are used to integrate the temporal specifications to the relational operators (restriction, natural join, negation, division, cross product and set operators). These terms are stated by using the following keywords, namely, HISTORY, PAST, FUTURE, @ ‘Dat’, BETWEEN ‘Dat1’ AND ‘Dat2’, WHEN ‘Condition’, and {BEFORE | AFTER | SINCE} {‘Dat’ | ‘Condition’}. Each of these keywords set the measured operator by a date or time period. All these seven terms are used to the valid-time dimension for all the operators. FUTURE cannot apply to the transaction-time dimension.
**Temporal restriction (selection)**

The temporal operator is applied on the temporal columns or temporal tables and allows for a restriction on the values or rows. All these terms are applied to the valid-time dimension for all operators.

**Table 1.1 Meaning of the restriction of temporal columns or temporal tables**

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTORY {Column</td>
<td>Table}</td>
</tr>
<tr>
<td>PAST {Column</td>
<td>Table}</td>
</tr>
<tr>
<td>FUTURE {Column</td>
<td>Table}</td>
</tr>
<tr>
<td>{Column</td>
<td>Table} @ ‘Dat’</td>
</tr>
<tr>
<td>{Column</td>
<td>Table} BETWEEN ‘Dat1’ AND ‘Dat2’</td>
</tr>
<tr>
<td>{Column</td>
<td>Table} WHEN ‘Condition’</td>
</tr>
<tr>
<td>{Column</td>
<td>Table} {SINCE</td>
</tr>
</tbody>
</table>
Temporal natural join

This temporal operator is applied in the temporal relationship between the two tables that can be temporal or non-temporal. These relationships are recognized by a temporal column and a referential constraint. The verification of the timestamps of these columns, with respect to the possible timestamps associated with the concerned objects, is made at the introduction links of the measured relationship using the constraints stated at the creation of the DB. The temporal operator relies on the timestamps of the links.

Table 1.2 Temporal join between tables

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTORY Natural Join</td>
<td>Considers all the valid links as well as current, past and future links.</td>
</tr>
<tr>
<td>PAST Natural Join</td>
<td>Considers only the valid links in the past.</td>
</tr>
<tr>
<td>FUTURE Natural Join</td>
<td>Considers only the valid links in the future.</td>
</tr>
<tr>
<td>Natural Join @ ‘Dat’</td>
<td>Considers the valid links at the specified date.</td>
</tr>
<tr>
<td>Natural Join BETWEEN ‘Dat1’ AND ‘Dat2’</td>
<td>Considers the valid links during the specified period</td>
</tr>
<tr>
<td>Natural Join WHEN ‘Condition’</td>
<td>Considers the valid links during the period when the condition is verified.</td>
</tr>
<tr>
<td>Natural Join {SINCE</td>
<td>BEFORE</td>
</tr>
</tbody>
</table>
**Temporal negation and temporal division**

The Negations and the divisions are articulated in SQL with correlated blocks using the NOT EXISTS operator and joins between the used table. These temporal operations are enhanced by checking one of the conditions that are specified for each version of temporal join, considering only the links timestamps.

**Temporal Cartesian product (cross product)**

This operator is applied between the two temporal tables. The temporal Cartesian operation makes the conventional Cartesian product between the two tables while associating with each concatenation timestamp that relates to the intersection of the two timestamps of the concerned rows.

**Table 1.3 Temporal versions of the Cartesian product between tables**

<table>
<thead>
<tr>
<th>HISTORY Cross Join</th>
<th>Considers all the concatenation between all valid rows, as well as current, past or future.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAST Cross Join</td>
<td>Considers only the concatenation between the past valid rows</td>
</tr>
<tr>
<td>FUTURE Cross Join</td>
<td>Considers the concatenation among the future valid rows</td>
</tr>
<tr>
<td>Cross Join @ ‘Dat’</td>
<td>Considers the concatenation between rows valid at the specified date.</td>
</tr>
<tr>
<td>Cross Join BETWEEN ‘Dat1’ AND ‘Dat2’</td>
<td>Considers the concatenation between the rows for validating during the specified period.</td>
</tr>
<tr>
<td>Cross Join WHEN ‘Condition’</td>
<td>Considers the concatenation between the rows valid during the period when the condition is verified.</td>
</tr>
<tr>
<td>Cross Join {SINCE</td>
<td>BEFORE</td>
</tr>
</tbody>
</table>
**Temporal set of operators**

It operates on the results of two sub-queries to perform the joint operation, the union or the difference between the values/rows of these sub-queries. It contains the temporal set interaction, temporal set union, and temporal set difference.

**Bi-temporal versions of SQL operators**

When two dimensions of the SQL operator are present in a column or a table, it is likely to find a particular information, such as data introduced with retroactive effects with post active effects or erroneous data. This version reduces the search of the data with the exception of the cross product operator. The option ‘WITH Value SCALE’ specifies a phase period from the moment of the execution of the transaction inserting the data.

### 1.8 SIMILARITY MEASURE BASED ON KEYWORD

In temporal databases, the similarity measure groups the different queries with the same or similar keywords. However, a single query term represents a various information. The similarity is calculated based on the overlap of the identical terms between the queries. The term similarity is important to the definition of a cluster. It measures the similarity between two queries which is important to the query clustering procedure. However, two queries are similar when,

- The queries have one or more terms in common and
- The queries have common results that have more than one item in common.
Similarity measure system based on the Keyword receives a query and
the system needs to detect this query means in terms of the data repository. If the
query consists of a single keyword then the single keyword can take place in
many locations in the database. If there are more keywords, the system has to
identify the relationship between the keywords in the underlying databases. If the
data in the underlying database are connected by many relationships then which
give meaning to the data. In the temporal database, not all the keywords play a
role on the matching function after the statistics of large amount data.

1.9 SUMMARY

Research in temporal database has been pursued for almost twenty
years. However, the migration of this research into commercial database is
limited. The retrieval of an image from the temporal database demands an
effective framework and human assistance. The image retrieval is enhanced
using a novel and a powerful framework named, Event Matching Agent
(EMAGT). When compared to the individual descriptors, the suggested
framework provided an optimal similarity function computation. The SQL is
often the language for developing the applications which utilizes the information
in the SQL databases. Plain SQL does not seem to provide an adequate support
for the temporal applications. The standard SQL3 does not allow manipulating
and querying of the temporal database. In order to avoid these problems a
temporal SQL3 mining technique is implemented by enhancing the query
manipulation in the temporal databases.
1.10 ORGANIZATION OF THESIS

The thesis is organized as follows:

**Chapter 2** provides a review of the relevant literature pertaining to temporal SQL3 mining and the techniques related to the neural network based image retrieval.

**Chapter 3** defines the research problem and the proposed research methodology.

**Chapter 4** describes the use of the EMAGT algorithm-based retrieving the images from the temporal databases.

**Chapter 5** presents an enhanced query manipulation in temporal database using TSQL3MINE approach.

**Chapter 6** describes a detailed description and performance of various existing and the proposed methods that were analyzed.

**Chapter 7** describes the conclusion of the proposed work with key points for future progressive work. The illustration of the thesis organization is shown in figure 1.5.
Figure 1.5 Organization of Thesis