CHAPTER-1

Introduction
CHAPTER-1

INTRODUCTION

1.1 HISTORY OF GRAPH THEORY

Graph theory formally started in the 18th century, when the famous mathematician Leonhard Euler solved the puzzle of the Königsberg bridge problem. Königsberg had seven bridges spanning branches of the Pregel River. The objective of the Königsberg bridge problem was to check whether a walk through the city could cross each bridge exactly once. Euler formulated the problem as a multi-graph by representing the land areas as vertices and the bridges as edges. Euler showed that the problem can be solved tabulating the degree. If a graph has no more than two odd vertices, then there exists some path traversing each edge once. All four vertices in the Königsberg bridge problem (graph) are odd [1]. Königsberg Bridge problem (in1735) lead to the concept of Eulerian Graph. Königsberg Bridge problem was studied by Euler and constructed a graph structure called Eulerian graph to solve the problem [5][6]. In 1840, A.F Mobius introduced the idea of complete graph and bipartite graphs. Later Kuratowski proved that these are planar graphs. In 1845, Gustav Kirchhoff introduced the concept of trees and he used the graph theoretical ideas to calculate the currents in electrical networks or circuits. In 1852, Thomas Gutherie proposed the famous four color problem. The four color problem was solved after a century by Kenneth Appel and Wolfgang Haken. Then in 1856, a new type of graph called the Hamiltonian graph has been proposed by Thomas. P. Kirkman and William R.Hamilton by studying trips that visited certain sites exactly once. This time is considered as the birth of Graph Theory [6].
People found it very useful to represent and describe facts and incidents as graphs. Graph has a good number of application domains like communications and transportation networks, the neural networks of the brain, sports, economics, market analysis, etc [1]. In the 20th century graph theory became more statistical and algorithmic. Study of random graphs, has begun. Random graphs are graphs which are formed by starting with isolated vertices and adding edges one at a time subsequently as needed. The operations performed on the World Wide Web and other very large graphs are also statistical and algorithmic in nature. It is noticed that most of the large graphs are not deliberately constructed. They have naturally grown through some evolutionary process. The Web is an example of such kind [1]. Another example of a very big graph is the graph for telephone billing records. This graph is termed as a “call graph”. The vertices of this graph are telephone numbers, and the edges are calls made from one number to another. The call graph has become a test-cases for algorithms designed to run quickly on data held in external storage [1].

Graph theory is become popular mainly because of its applications in diverse fields such as biochemistry, medicine, structural design, electrical engineering, computer science and operations research. The wide scope of these and other applications has been well-documented. The powerful combinatorial methods found in graph theory have also been used to prove significant and well-known results in a variety of areas in mathematics itself. The best known of these methods are related to a part of graph theory called matching [2].

Graph theory plays a very important role in the field of computer science and VLSI design [140]. In the field of computer science there are lot of graph applications in the graph based algorithms in software engineering [148],[149], data mining[11][57]
including mobile computing and pervasive computing. There are many graph based algorithms in computer networks like routing management, network security [144]. Distributed system itself can be represented as a graph or hyper graph. In distributed computing there are lot of graph based algorithms in distributed deadlock handling, load distribution, commit protocols.

Graph theory has many applications in the field of Chemistry [73][76]. Graphs were widely used in chemical science to represent the structural diagrams of molecules – atoms as vertices, and bonding between atoms as edges. Graph theory is used to explore the complete set of all possible monocyclic aromatic and heteroaromatic compounds.[73] There are many special kind of graphs used in chemistry. Some of them are-DUALIST GRAPHS for benzenoids and diamond hydrocarbons to enumerate isomers [76], Reaction graphs (1966) and SYNTHON GRAPHS for chemical synthesis [73][76].

Graphs are also used in Linguistics. Knowledge Graph was first introduced in 1982 [3]. Initially this kind of graphs is used to represent knowledge in the form of expert systems. In late 80's, expert systems designed based on knowledge graph theory are used in medical and social science. Later, knowledge graphs are used to represent natural language. Now knowledge graph representation has become language independent and can be applied to represent almost any characteristic feature in various languages [3]. Linguistic studies extensively employ graph theory [3][54][105][114][115][116]. Graph based approaches are used in monolingual and cross-lingual word sense disambiguation [105], linguistic annotation [114], natural language processing
Vincent Archer developed a model called “MuLLinG” for knowledge extraction based on multilevel graphs [116].

In VLSI design, graphs are important tools to analyze electronic circuit design, test printed circuit board for short circuit,[122][133][136][137], and floor planning [133][134].

Data mining is another area where graph theory and graph based approaches are extensively used [11][33]. Some of the specific fields of data mining are - data cleaning [33], frequent pattern mining [57], clustering [102][113], semi-supervised learning [103].

Graphs and graph theoretic methods are also used to simulate and study biological systems[7], biological networks[56][58][66][82], protein analysis [63], human brains[65][79], genomics[33], DNA sequencing[129], DNA fragment assembly[94], graph data management for molecular biology[100]. In biological networks nodes represents bio-molecules such as genes, proteins or metabolites, and edges connecting these nodes interactions (functional, physical or chemical) between the bio-molecules [7][58][66][82]. Graph theory is used to study the models of neural networks, anatomical connectivity, and functional connectivity [64]. Another application domain of graph application is bioinformatics. In bioinformatics, graphs are used for modeling gene-gene relationships [58].

In the past decade, graph theory has got a tremendous changes and transformation. This change is due to the large amount of humongous information that is available in recent application domains. Networks formed from the interrelationships among the data are the target systems for analyzing the massive data sets. For example,
any algorithms on WWW are based on the WWW graph. In the WWW graph, Web
texts are vertices and hyperlinks are edges. There are various information networks,
such as biological networks built from biological databases and social networks formed
by email, phone calls, instant messaging, etc., as well as various types of physical
networks. Yet another form of graph is the collaboration graph. In the collaboration
graph mathematicians are vertices and if a mathematician wrote a joint paper with
another mathematician then there will be an edge between these two mathematicians
(vertices) [4].

1.2 APPLICATIONS OF GRAPH THEORY IN DATABASE TECHNOLOGY

Graph has extensive use in the field of database technology. Graphs and hyper
texts are important tools in the theory of databases. Beginning from data modeling
(schema diagram, schema diagram, database graph, reference graph, etc.) to deadlock
control, concurrency control (Wait for Graph, Resource Allocation Graph, etc.);
transaction processing, query processing and optimization [101], data distribution, there
are many applications of graph theory in database technology. Quite a few of them
include functional dependency analysis (Functional dependence graphs (FDGs) [17],
conceptual modeling [24][30], database schema design [36], automatic database schema
matching [109], distributed deadlock [95], implementation of relations and relational
operator using hypergraph model [60]. Functional dependency graphs (FDG) is used to
represent the dependencies among the entities of a system. From the FDG it is possible
to find out all the candidate keys of a relational scheme [138], abstract mathematical
model from which, in turn, a Database Requirement Specification (DRS) is obtained
[24]. Database schemes can be represented as hyper graphs [36][60]. Wait For Graphs
(WFGs) and Resource Allocation Graphs (RAGs) are used in deadlock detection in
distributed databases. There are many graph based approaches that are applied in database recovery and concurrency control [95]. In object oriented database, class diagrams are nothing but graphs.

1.3 GRAPH DATABASES

Graph databases are storage system and provide index-free adjacency. Graph databases make use of graph theory. They include nodes, edges, and properties. This model is faster for associative data sets and uses schema less, bottom-up model for rapidly changing data. In recent years, graph databases received a great attention from the researchers. A lot of research papers have been published. A good number of graph databases have been developed in the recent years and they have their own merits and demerits depending on the graph models used, functionality, usability in heterogeneous application domains [42]. Graph databases have got popularity because of their capability to represent complex relationships among data. Relational databases have limitations in representing complex relationships as the relational databases follow fixed schema and new relationships between the data cannot be added easily.[42][44][47]. Graph databases are capable of handling highly heterogeneous and dynamic data [98].

Darshana Shimpi, et. al, discussed about the basic structure of graph database. A comparative study of different graph databases is presented and it is shown that Neo4j graph database is the most popular graph database [42]. Shalini Batra, et.al. , provided a comparative study between the relational databases and the graph databases. They also concluded that Neo4j graph database is the most popular graph database [44]. Justin J. Miller also presented a comparison between relational database systems (Oracle, MySQL) and graph databases (Neo4j) based on data structures, data model features and
Chad Vicknair, et. al., have done a comparison between the graph database Neo4j and the relational database system MySQL[112]. Mike Buerli [47], discussed about the application of different graph database models and categorized them into different types. Ravinsingh Jain, et. al. [98], discussed about few NoSQL-like databases including Google’s “Big Table”, Facebook’s Cassandra, Amazon’s DynamoDB and LinkedIn’s Project Voldemort. There are several other graph databases like Neo4j, InfiniteGraph, InfoGrid, DEX, AllegroGraph etc. [42]. Leonid Libkin, et. al. [90], concluded that XPath-like languages can also be applied to graph databases in addition to dedicated graph database query languages like Cypher Query etc. Yuanyuan Tian presented in his Ph.D. thesis about different graph matching algorithms like SAGA, TALE and two aggregation operations – SNAP and k-SNAP[135]. Abdurashid Mamadolimov, in his paper [107], has presented few search algorithms for conceptual graph databases.

A list of few graph databases is presented [47]. They are categorized based on their types and functionalities.

![Figure 1.1 Graph Databases Structure]
**Main-Stream Graph Databases:**

The main-stream graph databases provide an object model for nodes and relationships:

- **Allegro Graph (2005)** high-performance, software oriented database model.
- **DEX (2007)** efficient, bitmaps-based graph database model written in C++.
- **Neo4j (2007)** disk-based transactional graph database advertised as *the world’s leading graph database*. It works on a network oriented model with relations as first class objects.
- **HyperGraphDB (2010)** an open-source database focused on supporting generalized hyper-graphs.
- **Sones (2010)** An object-oriented database written in C#.

**Distributed Graph Databases:**

Distributed Graph databases focus on distributing large graphs across a framework.

- **Horton(2010)** A transactional graph processing framework created by Microsoft.
- **InfiniteGraph(2010)** A distributed-oriented system that supports large-scale graphs and efficient graph analysis.

**Key-Value Graph Databases:**

Key-value graph databases simplify the object-related model of graph databases to allow for greater horizontal scalability.

- **VertexDB(2009)** A key-value disk store.
- **CloudGraph(2010)** An in-development, fully transactional graph database written in C#.
Redis Graph (2010) An implementation of a graph database in python using Redis. Redis is a modern key-value store;

Trinity (2011) A RAM-based key value store under development by Microsoft Research.

**Document Graph Databases:**

Like key-value stores, document based graph databases introduce a higher level of data complexity for a given node.


CouchDB(2012) This implementation makes use of the document store, in order to serve low latency queries for large graph databases.

Document-stores, much like key-value stores provide quick data retrieval for structured data.

**SQL Graph Databases:**

Filament (2010) A graph persistence library built on top of PostgreSQL. It allows SQL querying through JDBC with navigational queries for querying the graph data.

G-Store (2010) A prototype query language and storage manager for large graphs. It is also build on top of PostgreSQL.

**Map/Reduce Graph Databases:**

To handle very large graphs, one can implement Map/Reduce functionality, in order to achieve the maximum amount of parallelism. Partitioning nodes of a graph across many machines will result in only a sizable amount of computation to be one on each machine.
Pregel (2009)  
A vertex-based infrastructure for graphs built on top of Hadoop. Hadoop, a Map/Reduce framework provides batch jobs for processing the distributed vertices with message passing. Offline queries of the graph data.

Phoebus (2010)  
Another implementation of Pregel, again building on top of Hadoop in order to benefit from the Map/Reduce framework.

Giraph (2011)  
builds on top of Pregel with the addition of fault tolerance. If the application coordinator has a fault, one of the available nodes will automatically become the new coordinator.

Figure 1.2: A Graph database instance (SOURCE: WIKIPEDIA)

1.4 OUTLINE OF THE THESIS

In this thesis, the following topics are discussed.

a) In chapter-2, a graph based approach to find all the candidate keys of a relational scheme is presented. First, from the set of FDs, a FD G graph is created.
Secondly, the additional FDs implied by the set of FDs are taken into consideration to add new nodes in FDG. This process is called the Augmentation process and produces an augmented FDG $G^+$. Lastly, the augmented graph is reduced by removing certain nodes and cycles that satisfy the condition $\forall K \in G$, in-degree ($K$) $> 0$ and out-degree ($K$) $= 0$, $K$ may be either a node or a cycle. During this reduction process, all useless nodes, other than the candidate nodes, are removed and the Candidate Graph $G^C$ is obtained. From $G^C$ the candidate keys of the relational scheme are obtained [138].

b) In chapter-3, a method to design the database schema using reference graph is discussed. This approach is actually giving us an abstract mathematical model from which the actual database schema can be obtained. The abstract model gives all possible alternatives for designing the database schema and the appropriate one can be chosen at the time of implementation as per the requirements [24].

c) In chapter-4, a procedure for converting existing relational databases to graph database models is presented [25]. An implementation of this procedure is presented in APPENDIX-I.

d) In chapter-5, two application domains for graph database are discussed.

i. Multi-Lingual Word Bank(Dictionary)

ii. World Population Database