COMPARATIVE STUDY OF DATABASE MODELING APPROACHES

Department of Computer Science

Algoma University

APRIL 2014

Advisor

Dr. Simon Xu

By

Mohammed Aljarallah
ABSTRACT

An overview and comparative study of different database modeling approaches have been conducted in the thesis. The foundation of every structure is important and if the foundation is weak the whole system can collapse and database is the foundation of every software system. The complexity and simplicity of this core area of software development is also a very important issue as different modeling techniques are used according to the requirements of software and evaluation of these techniques is necessary. The approach of the thesis is a literature survey and background of data modeling techniques. All the requirements of data modeling and database systems are encapsulated here along with the structure of database. The foundation of the thesis is developed by discussing some of the cases studies regarding the database models from the practical field to develop an understanding of database systems, models and techniques from the practical perspective. The study of database system and most of the models are investigated in the thesis to establish a scenario in which these modeling approaches could be compared in a more innovative and better way. Relational database modeling approach is one of the important techniques used to develop the database system and the technique that could be used as replacement of relational model as single or in hybrid way is also an interesting aspect of survey. The comparisons of traditional and modern database modeling methodologies are also discussed here to highlight the features of current database management systems. The languages used to develop the database are also important aspects in this concern and development in the languages is also surveyed. Some of the results are deduces based upon the comparison that how these modeling approaches could be used in the DBMS in different ways for the development of the software systems.
# Table of Contents

**I. INTRODUCTION** .................................................................................................................. 1  
1. Motivation ............................................................................................................................... 2  
2. Research Questions ................................................................................................................. 3  
3. Related Work ........................................................................................................................... 4  
4. Structure of Thesis .................................................................................................................... 4  
5. Research Method ..................................................................................................................... 5  

**II. BACKGROUND** ................................................................................................................... 7  
1. Database Management & Data Models .................................................................................... 7  
2. Overview of DB Design ........................................................................................................... 10  
   2.1 Modeling ............................................................................................................................... 11  
   2.2 Keys & Entities ...................................................................................................................... 11  
   2.3 Relationship & Mapping ....................................................................................................... 12  

**III. STUDY OF DATABASE SYSTEMS** ....................................................................................... 14  
1. Database Models ..................................................................................................................... 14  
   1.1 Hierarchical Database Management System ...................................................................... 14  
   1.2 Network Database Management System ........................................................................... 15  
   1.3 Relational Database Management System ......................................................................... 16  
   1.4 Object Oriented Database Management System ................................................................. 19  
   1.5 Deductive Database Management System ......................................................................... 22  

**IV. MODERN DATA MODELS** ................................................................................................ 24  
1. Key Value Stores ..................................................................................................................... 24  
   1.1 Database Design Overview ................................................................................................. 25  
   1.2 Management of Hashing Functions .................................................................................... 27  
   1.3 Keys and Value Design ....................................................................................................... 28  
2. Big Table Style Database ........................................................................................................ 30  
   2.1 Rows .................................................................................................................................. 33  
   2.2 Columns .............................................................................................................................. 33  
   2.3 Time-Stamps ...................................................................................................................... 34  
   2.4 Application Programming Interface .................................................................................... 34  
2.5 Implementation of Bigtable as Database .............................................................................. 35  
3. Document Oriented Database ................................................................................................. 36  
4. Full Text Search Engines ......................................................................................................... 41  
5. Graph Database ....................................................................................................................... 45  

**V. CASES STUDIES** ................................................................................................................. 52  
1. Case Study 1 (Document Oriented Database) ......................................................................... 52  
2. Case Study 2 (Graph Database) ............................................................................................. 52  
3. Case Study 3 (Key Value Store Database) .............................................................................. 53  
4. Case Study 4 (Big Table Style Database) ............................................................................... 53
5. Case Study 5 (Full Text Search Engines) ................................................................. 54

VI. COMPARISON OF APPROACHES ........................................................................ 55
1. Comparison of Approaches .................................................................................. 55
2. Applications/Fields ............................................................................................... 64

VII. RECOMMENDATIONS .......................................................................................... 66

VIII. CONCLUSION ........................................................................................................ 68

REFERENCES ............................................................................................................ 69

APPENDIX .................................................................................................................. 73
FIGURES

Figure 1. Flow Chart of Thesis ........................................................................................................................................ 6
Figure 2. Accessing Database through DBMS ............................................................................................................. 8
Figure 3. Historical Methods of Data Storage ............................................................................................................. 9
Figure 4. Unary Mapping ................................................................................................................................................ 13
Figure 5. Binary Mapping ............................................................................................................................................ 13
Figure 6. Ternary Mapping ............................................................................................................................................ 13
Figure 7. Hierarchical Model ........................................................................................................................................ 15
Figure 8. Network Model .............................................................................................................................................. 16
Figure 9. Components of Object in Database ........................................................................................................... 20
Figure 10. Specifications ................................................................................................................................................ 23
Figure 11. Overview of Database Design .................................................................................................................... 26
Figure 12. Collision Detection & Data Storage .......................................................................................................... 27
Figure 13. Architecture of key-Value Store ................................................................................................................. 29
Figure 14. Architecture of BigTable Data Model ......................................................................................................... 31
Figure 15. Bigtable and Tablets Structure .................................................................................................................. 32
Figure 16. Structure of SSTable .................................................................................................................................. 32
Figure 17. Communication between Client & Servers .................................................................................................. 35
Figure 18. Chubby File Structure ................................................................................................................................ 36
Figure 19. Communication of Application with Primary/Secondary nodes ................................................................. 39
Figure 20. Architecture of Shards ................................................................................................................................ 40
Figure 21. Communication in Text Search Engine ..................................................................................................... 41
Figure 22. Architecture of Full Text Search Engine .................................................................................................. 44
Figure 23. Data Structure of Graph ................................................................................................................................ 46
Figure 24. Specific Types of Graph .............................................................................................................................. 48
Figure 25. Relational & Modern Data Models ............................................................................................................. 63
I. INTRODUCTION

The storage of data, its usage and importance can never be ignored either from a business perspective or current technological trends. This is the age of information technology, and almost everything is directly or indirectly related with the database systems. Most of the computing world is dedicating its power and speed to maintain or improve the database. It is a common fact that now every business requires some kind of database system to keep a record and these records could be financial or any other records such as contact details or email storage. The database system is now become the key integral part of almost every system.

The development and usage of database system are done according to the requirement of the system or kind of usage in the practical field. The pace of development and expansion of software technology has also highlighted the field of database system and database modeling. Different software languages are developed to design such database models and the performance of these models is directly related with provision of data modeling facilities in such languages. The data model could be considered as the architecture of the building in terms of the engineering world and is normally called conceptual representation of the data structure. Data models are not for the purpose that how operations are done on the data, but they are to develop a structure of data requirement and their organization. The representation of the data models is quite close to the user as he imagines the data in the world, and it should not be dependent on the software and structure or working of hardware.

It is very important to discuss here the most critical and differentiating aspect of data model from the functional model as in the data model the only aspect is the type of data to be stored, no processing of data is concerned in this regard. If taking the example of relational database then
modeling of database is developing the tables and the function is done by the queries on these tables. The modeling of database is restricted to certain strategy and information, and this information is gathered by the requirement of database, developer and final user. The result of data modeling consists of two major parts in which one is the E-R diagram, and other is the document that is elaborating the rules for that data model. The importance of the data model is like the blueprint of the house before developing the structure of that house.

The comparison of database modeling is the best approach before constructing any database as it gives the path for the developer to analyze the requirement of data in a particular project. There are various data modeling approaches to design the database systems and these used in different software development projects. The comparative study is also important to draw a line between the traditional and modern data modeling techniques.

### 1. Motivation

Research of Data modeling and database system has always remained an attractive field, and the motivating factor in this regard is the pace of development in the software systems and strong database requirements. As data modeling is the base of database development so it is very important for any developer to have the deep understanding of these modeling techniques to have strong developmental skills. This section of the software development not only impacts the skills but also enhances the value of any developer to lead the market with according to the future perspective of the particular industry.

The innovation in the design of software system can never be achieved by having the little or no knowledge of data modeling methods so it is very critical to get the in-depth knowledge of all the
techniques and especially the comparison of all the techniques can open new gateway of knowledge and understanding.

The study of data modeling is not only done to have the greater knowledge of database systems but the innovation required in the field is very important that how the data models could be used for highly advanced level of software development and there is a great chance that research would end up with a new technique that could be used conveniently in the future programs. It is an important aspect to develop a chain process of research in this field as the new technologies and the understanding of their implementation is also necessary for the future generation.

2. Research Questions

The section mainly introduces some of the questions that were raised while doing the thesis and it was tried to answer all questions during the research and thesis writing.

What is actually the Data and Data Modeling?

The research done on the Data Modeling is the startup of study and should be answered before going into deep technical knowledge.

What are different Data Modeling Techniques used in the Practical field?

All the Techniques used in the software development during database design needed to be discussed here in the thesis.

How Data Modeling Techniques could be compared with each other?

A clear comparison of every technique with every other technique is necessary to have in-depth knowledge of all the methodologies.
What are the Limitations of every DB Modeling Technique?

The critical analysis of these techniques is very important aspect of thesis to define the pros and cons and usage of each technique.

What were the traditional DB Modeling Techniques and how they were implemented in the development of DBMS?

Traditional techniques need to be discussed in the great detail with all practical applications.

What are the Modern DB Modeling techniques and what are the future trends?

The discussion of future trends in DB models is the most important aspect of the thesis and requires huge discussion.

3. Related Work

The continuous research in the field of DB modeling and DBMS is an obvious fact as this one of the growing field. The students related with software development have done research in DB modeling and techniques but I could find the article which is closely related with comparative study of DB modeling techniques. TB Pedersen has identified different data modeling techniques with query processing but the comparative study of these models is not done even in that thesis. Comparative analysis of predictive data mining techniques is done by GC Nsofor and in that document some of the data modeling techniques are also discussed.

4. Structure of Thesis

The structure of thesis could be divided into 5 different and major sections and in which the first section will include the details of background of database modeling. Then thesis gradually moves towards the technical details of the database and these details are not highlighted in the
theoretical way but the support of some case studies are taken to develop the basic sketch of trend of the thesis. The reason of selecting case studies in the second section is to get the understanding of data modeling and its applications even before going into the technical details of the modeling techniques.

The third section is the longest section of the paper, as it will include all the literature survey with all the required details for database modeling and database management system. In this section, the requirement of further analysis will also be focused so that no important method or detail should remain un-discussed.

The next important section, which is the continuity of literature survey is the comparative study of data modeling approaches. Although this section will be brief and shorter than previous literature survey but this section will be the core of the thesis as this will be the most technical part of the thesis. All the analysis and results would be discussed here from the major perspectives. One idea is to compare the techniques form the theoretical perspective and other is the application of all of the important modeling techniques and their comparative analysis. The future prospect of any field is very important part of the paper as a whole of the gained knowledge and understanding in this research paper will be a recommendation or the future perspective of the technology.

The last two sections are references and Appendix I in which the list of data sources and list of abbreviations are included respectively.

5. Research Method

The research will involve the understanding of the all data modeling techniques by studying and deeply analyzing the literature of the different articles available on the internet and online
available books. The traditional methods of data modeling would be studied at the first instance and then the current data modeling techniques used in the development of database system and software development would be studied. The language that is used in the development of data models are also the integral part of this research as without having the knowledge of the language and their difference the full understanding of database models is impossible.

The understanding of reasoning behind the usage of current data models and shifting to the trend from previous data models is a better method of research to get an idea of the whole scenario of database modeling. This method will lead the research towards the phase of comparison between different techniques. The study of different cases regarding data modeling will develop a sense to create a hypothesis. The support of hypothesis and rejection could be done on the base of these case studies. The selection of scholarly articles is the most difficult phase of the thesis but once this phase is done the study of articles will lead the research towards perfection.

Figure 1. Flow Chart of Thesis
II. BACKGROUND

The abstract model for the representation of data and the relationship of data in a structured way is called data model. In other words, it is descriptive representation of data, which consists of a structure that describes the attributes, relationship and dependencies of the data. The database model represents the behavior of the system along with organization of the database to reduce redundancy in the data entries. The quality of good data model is easy to understand and simple or represents the data in a simple form. The opportunity of expansion is also one of the critical qualities of good data model.

1. Database Management & Data Models

The handling of data on the papers is the story of the previous century in which huge man power was used, and the chance of data loss and errors in the data was immense. The system of data storage and efficiency of searching the data was at lowest grade. The time passed, and computing technology came into the market. In the early era of computer technology data was stored in the files and any software that requires the data could get through the files, but the problem of data redundancy occurred due to this method. It was required to remove the drawbacks of file system and to overcome the problem of application specific software for the data. The general framework of software system is developed, and this is called database management system. This time data remained neither specific to particular application software nor to a particular storage, but DBMS is responsible to manage the data at any machine and for any application. This general purpose software allowed the users to manage the data and user could access the data any time. The base of this DBMS is data model, and the model establishes that how data could defined and accessed or even modified in the database. The user is provided the facility of data structure and this data structure hides the details about the physical storage of data. Data
models use logical concepts, which are more close to the user such as objects and their relationships or their properties. The data is defined in the uniform way by the data models for the storage and in this way different applications can access the data through a single interface known as DBMS.

![Diagram of Database Access through DBMS]

*Figure 2. Accessing Database through DBMS*

The deep understanding to database management requires the strong knowledge about the database. The database could be considered as the data that is interrelated and collected in the form of single unit. There are various examples of database systems developed by the different vendors and used all over the world. These vendors develop their database system in different ways. Some of the vendors store the data a single file like MS Access and some places the same data in a single file but the whole database consists of multiple files, which is controlled by the software instance. The data models in these database systems define the pattern or the way data is stored and related with each other.
Figure 3. Historical Methods of Data Storage
2. Overview of DB Design

The designing of database is an important field in relation to the subject under discussion. The design consists of three main sections and these sections are listed as follows:

- External Layer or Conceptual Design
- Logical Layer or Logical Design
- Physical Layer or Physical Design

By developing all these designs or layers of abstraction a database is developed, and database is also handled or modified by performing actions through these layers. All the data files are present at physical layer that need for the database. The number of files in which the entire database is stored and located in the different drives, and this mechanism is adopted by all DBMSs. The DBMS which stores whole of the database in single file is not capable for the software associated with multiple users or for large organizations due to its single file limitation.

The understanding about the structure of the database is not required by the database user or administrator and could be managed with user interface, but a person who is responsible for the software development should be aware of types and structure of the database. The designer of software should also have knowledge about the database models required for a particular purpose and software. The user of specific DBMS associated with personal software might need to handle the files in the database but the DBMS, which are server based like MS SQL and Oracle never requires the user to modify the files of database as these software contains the capability to manage their files. The layer above concrete physical layers of files is the logical layer, and it consists of the data structure, which is linked with the physical layer for data management. The tables shown in the database management system or other structures are actually the logical
layer. The second layer of abstraction is an external layer, and it is above the logical layer and normally used by the application program to send the queries against a database.

2.1 Modeling

It is an important aspect of the thesis to differentiate between Database Models and Data Models before going deep into the research. Database models are actually the architecture of DBMS, on the other hand, Data Models are not the architectures but design of database in a specific way. There is a long history of database modeling approaches and these approaches are changed over a period of time according to the requirement of the systems. As far as data models is concerned it reduces the chances of data duplication and contains the tendency for expansion.

2.2 Keys & Entities

Keys are the most important parts of the database, and these are used to sort and store the records of the database. The identification of different relationship among the tables is also done with the help of keys. There are two types of keys which are as follows

- Primary keys
- Foreign Keys

In designing the database primary keys and their identification, is the most important task and these primary keys are actually the set of characteristics to find any record in the unique way. The example of primary key is the identification of student roll number in the table. Foreign keys are used to reference the primary keys and example of foreign key is if a table with course identification number is used as primary key then it could be considered as foreign key in the table with students data that which student is taking which course or in short it could be said that it is the basic key of another table but also present in student record.
Entities are things or persons in the database about which data is stored and the types of entities are as follows

- Independent Entities
- Dependent Entities
- Associative Entities
- Subordinate Entities

2.3 Relationship & Mapping

The observation of relationship among the data is one of the important aspects of data modeling and developing the structure of database on the base of relationship among different objects is called mapping. The number of entities involved in the relationship defines the degree of relationship. Multiple relations are possible between different entities but commonly unary, binary and ternary relations are seen. These are the following sub-categories of relationships in unary and binary relations

- One to One Relationship
- One to Many Relationship
- Many to Many Relationship

In one to one relationship two entities are in exact one relationship with each other just like real life example of husband and wife. The following figure shows the unary type with one to one and one to many relations.
Figure 4. Unary Mapping

The following figure shows the mapping of binary relations in all three categories.

Figure 5. Binary Mapping

The following figure shows one of the practical scenarios of ternary relations between different objects.

Figure 6. Ternary Mapping
III. STUDY OF DATABASE SYSTEMS

1. Database Models

The initial development of data storage started with the file system, and these files have actually no relation or information of the application using these files. These files are accessed through the application containing the information or name of the files so these files cannot be considered as a database but used to store information about the database. The application software is not aware of the structure of the file but as compared to it database management system consists of metadata and it is the responsibility of metadata to translate the files in the physical layer as well as creation of logical layer. Conceptually metadata is that data which is stored in the catalog of database that how data is stored in database and how relationship would be developed among data. The example of flat file system is just like an application dealing with customer records through cutomer_data.txt.

1.1 Hierarchical Database Management System

In hierarchical modeling technique, the database is a bit revolutionized and organized and the scheme is just like the record file indicating the address of certain data or pointing towards the place of the actual record. The following example shows that two pointers are clearly pointing towards the web page and street address but for the pointer address or URL could be taken from the record, which is one step above the actual information or data.
The relationship in this case is parent child with one parent and many children.

1.2 Network Database Management System

Network database model is somewhat similar to the hierarchical data model, but this model is first time developed by CODASYL network although initial network database was Honeywell IDS but it was not standardize at that time as network model. The network model developed by CODASY used same method of parent child relationship like hierarchical model. In this model, a member record could have only one owner record for each type of set but record type can be a member of more than one set.
1.3 Relational Database Management System

Relational data model is not a new data modeling technology in the current century as its roots are linked with the previous century as it was developed in 1969 by E.F Codd. Relational database model is widely used data modeling technique for the small and medium data storage systems all around the world.

The models are based on the logic and mathematics and management of data could be done with the help of table sets not any type of individual data levels. The table consists of a particular structure that is known as schema. In every table of there is one primary key for the identification of tuple. Values should be present in all of the columns of the primary key, and it is not allowed that columns having Null values. In relation database it important to understand the concept of
referential integrity, which is exact matching of foreign key in any other table with the primary key. The data structure involved in relational data modeling is relation as shown in the following table in which each section is related with employee

<table>
<thead>
<tr>
<th>Employee_ID</th>
<th>Name_of_Employee</th>
<th>Grade</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>Brad</td>
<td>17</td>
<td>10000$</td>
</tr>
<tr>
<td>2002</td>
<td>John</td>
<td>16</td>
<td>90000$</td>
</tr>
</tbody>
</table>

The standard language that is used for relational database management system is SQL. There are three parts of SQL language which are listed here as

- Language for Query
- Language for Defining Data
- Language for Data Modification

There are various functions of the query language in relational database such as data retrieval which done according to the algebraic properties, ordering in the rows and grouping of data. Data is modified by inserting a new data in the tables or by data deletion. This modification is done by the SQL application itself or through any other software application interface. Data definition language is used for the creation of base tables and indexes. Base table of database is actually relationship table that is physically present in the database and indexes are used as paths to locate the data. View is virtually stored table in the data as compared to base table and specification of data is done through query. SQL query only tells the type of data to be accessed from the database, not method of data retrieval. Relational database supports many integrity constraints such as foreign key, primary key and reference integrity. The disadvantage of
relational database modeling system is that the relation should be in first normal form which means a domain of one attribute must have single value and value of that attribute must contain one value in that domain. Example of first normal form violation is shown in the table

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Name</th>
<th>Contact Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>Brad</td>
<td>44444444444</td>
</tr>
<tr>
<td></td>
<td></td>
<td>55555555555</td>
</tr>
<tr>
<td>2002</td>
<td>John</td>
<td>33333333333</td>
</tr>
<tr>
<td>2003</td>
<td>Julia</td>
<td>22222222222</td>
</tr>
</tbody>
</table>

Now to achieve 1NF limitation table should be segregated on the base of customer ID

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Contact Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>44444444444</td>
</tr>
<tr>
<td>2001</td>
<td>55555555555</td>
</tr>
<tr>
<td>2002</td>
<td>33333333333</td>
</tr>
</tbody>
</table>
This disadvantage is catered with the help of non first normal form and incursion of complex attributes.

1.4 Object Oriented Database Management System

There are several restrictions in the relational database modeling and management system. These restrictions include the limitation of first normal form and extension of functionality. So it was the need of the time to develop such models in which these issues are resolved. In object-oriented data modeling, complex values could be stored, and additionally application can share the codes and data. The base of data model is developed on object oriented and identity based programming paradigm as compared to the relational model in which values are considered an important parameter. Objects in the database are located by using number rather than using a primary key value in the table. The system generates number is called OID (object identifies) and does not change as the key in the relational model. In object database, the information related with an entity is considered as one object, while on the other side relational database stores, complex objects spread in many relationships. The following figure consists of components of object in the database with behavior and state
Figure 9. Components of Object in Database

Data structure specifies the state of the object and state values for the objects are based on attributes. There are three different types of attributes

- Simple Attribute
- Reference Attribute
- Complex Attribute

Integer, real and string data type values are present in simple attributes. OID of any other object is present in the reference attribute. List of values are stored in the complex attributes. There are various similarities and differences between object oriented and relational DBMS such as presence of tables in the database is like class in OODBMS. Tuple is equivalent of class instance in RDBMS and column is like attribute but the difference here is that in RDBMS only primitive data type could be stored in column but class attribute can store any type of data.

<table>
<thead>
<tr>
<th>Database Model</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>OODBMS</td>
<td>Objects can stores many numbers of other objects and atomic types.</td>
<td>In RDBMS this is done by huge table or by various smaller normalized</td>
</tr>
<tr>
<td>RDBMS</td>
<td>OODBMS</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Complex data could be handled here and can perform better than RDBMS.</td>
<td>Complex data could be handled here and can perform better than RDBMS.</td>
<td></td>
</tr>
<tr>
<td>Hierarchical data characteristics are present in the database.</td>
<td>Hierarchical data characteristics are present in the database.</td>
<td></td>
</tr>
<tr>
<td>Presence of no primary key and unique recognition is done by OID.</td>
<td>Presence of no primary key and unique recognition is done by OID.</td>
<td></td>
</tr>
<tr>
<td>Schema change does not take huge time as it is done by deletion or</td>
<td>Schema change does not take huge time as it is done by deletion or</td>
<td></td>
</tr>
<tr>
<td>creation of tables.</td>
<td>creation of tables.</td>
<td></td>
</tr>
<tr>
<td>RDBMS is not language specific so no dependencies of language.</td>
<td>RDBMS is not language specific so no dependencies of language.</td>
<td></td>
</tr>
<tr>
<td>New tables could be developed by joining earlier developed tables.</td>
<td>New tables could be developed by joining earlier developed tables.</td>
<td></td>
</tr>
<tr>
<td>Difficult and time consuming depending upon the size and complexity</td>
<td>Difficult and time consuming depending upon the size and complexity of</td>
<td></td>
</tr>
<tr>
<td>of database.</td>
<td>of database.</td>
<td></td>
</tr>
<tr>
<td>This database is language specific and tied to specific application</td>
<td>This database is language specific and tied to specific application</td>
<td></td>
</tr>
<tr>
<td>program interface.</td>
<td>program interface.</td>
<td></td>
</tr>
<tr>
<td>A query on the data is dependent on the system design.</td>
<td>A query on the data is dependent on the system design.</td>
<td></td>
</tr>
</tbody>
</table>
1.5 Deductive Database Management System

This database is developed on the grounds of deductive logic or rules and facts related to the data and these rules and facts are stored in the database. Queries are answered in combination to the rules and facts as query language is designed with a logical model. The particular database type is used in various fields of applications such as cloud computing, networking and security systems. Deductive database could be considered as a combination of relational database along with logic programming so this database becomes more supportive for the logic programming system but less desirable for the relational database management system. A closed world assumption is taken while developing this database as everything deals in a logical way and any fact that is part of the real world but not presents in the database, never exists for the database.

Deductive database claims to adopt several properties of relational database and declarative property is one these properties. Declarative property is about querying or updating the information according to the requirement without knowing anything about operation. There are two important variations in the deductive database system and these are as follows

- Expert database system
- Knowledge based database system

In the pure deductive system data might be present in the both primary and secondary sections of the memory but in both of the above types data should be present in the primary section. Programming languages, which are used to implement deductive system, are known as Prolog and Data log.
Facts in the deductive data modeling are just similar to the tables in the relational database and also like describing relationships. There are no concepts of column and attributes in this type of modeling. Views are taken as rules in the relational models and these rules could also be considered as virtual tables. In relational database information could be extracted by using view and similarly in deductive modeling information could from the facts through rules.

**Figure 10. Specifications**
IV. MODERN DATA MODELS

Whenever there is discussion of databases one thing that comes to mind is SQL so it means SQL and database are almost the same or synonym. There are so many traditional methodologies for storing and retrieving the data, but the expansion of internet and web based applications have created a need of scaling. The Traditional databases are poor in terms of scaling and demand of modern technologies is to have a database that could be scaled in an easy way. The modern database models are also called NoSQL databases and these are first started to overcome real time problems of the services in the companies like Amazon, Google, Facebook, Yahoo etc. These modern databases are not directly started by replacing the SQL but first SQL is used for such application but rejected on the basis of an important issue which is known as scaling. The issues that were faced by all the companies were huge data transaction and requirement of perfection in the service. These companies adopted relational and traditional databases tried to run the system by enhancing the hardware but did not get any success due to huge data storage. They also scaled traditional databases but the extension did not resolve the core issue of data limitations rather proved to be temporary solutions.

1. Key Value Stores

Web service in the current era contains key value store as database to obtain high performance. This approach of database provides the opportunity of scalability as compared to traditional databases. The shape of databases is changing with the emerging technologies of distributed applications. Key value store databases are the backbone of web based modern applications. Applications that are required to perform complex operations on data and ad hoc queries could relay on the relational database and all this is possible due to flexible nature of relational model. The only disadvantage here is quick information retrieval as some applications required efficient
methods which could be done without combining data from different tables as it becomes overhead for any application. In addition to this disadvantage relational models have one more problem of handling non-uniform data which could be flexibly adjusted by entity attribute value but in this case queries take more time so key value store is suitable option. This database does not focus on the data structure but focuses on capability to store and extract data. This makes the system more scalable, flexible and efficient for large data application.

The items in the database are stored with a key or with the attribute name along with the value. The tables are schema free unlike relational databases and normally called domain. Domain contains different items with different schemas and these items are identified by their keys. One item can have different attribute and different implementations. For some implementations attributes are string and for other implementations codes of ‘int’ and ‘string’ etc could be used. There is no explicit relationship between domains for the key value store although key value store is much similar to the SQL database but only two columns are present with one as a key and other as value.

1.1 Database Design Overview

Key value store database is portioned over distributed servers and for this assurance that data is distributed; key values are hashed for determination of location at which data is stored. In certain types of key value store models, queries are supported to fetch consecutive data and data is stored according to the key order. When application requires any value from the database a key is provided to key/value store database and data is retrieved from the database after hashing. Process of data retrieval is as follows
The key value store database could be divided into the following categories:

- Key Value Eventually Consistent
- Key Value Hierarchical
- Key Value Cache in RAM
- Key Value Solid State
- Key Value Ordered

The initial understanding of key value store data model requires a comparative discussion of key value store with traditional database. The following example shows the database tables for the truck driver’s information in the company:

<table>
<thead>
<tr>
<th>Driver-ID</th>
<th>Lic-No.</th>
<th>DOB</th>
<th>Lic.Type</th>
<th>Driver-ID key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020</td>
<td>70</td>
<td>22-11-70</td>
<td>HTV</td>
<td>2020</td>
<td>70</td>
</tr>
<tr>
<td>2021</td>
<td>50</td>
<td>30-09-75</td>
<td>LTV</td>
<td>2021</td>
<td>50</td>
</tr>
<tr>
<td>2022</td>
<td>60</td>
<td>28-01-80</td>
<td>PSV</td>
<td>2022</td>
<td>60</td>
</tr>
</tbody>
</table>
1.2 Management of Hashing Functions

A hash function locates or calculates the values with the help of key stored in the database. This function plays a fundamental role in overall performance to the system. The reason for this critical nature of function could be explained by an example of collision that is if two keys locate to the same value after hashing. Key/value store database must have the capability of collision detection and should also be able to resolve the issue of collision. Different strategies are as follow:

- One of the strategy for this problem is secondary hashing for finding new location
- Second solution is linear search in the database for next slot available from the previous calculated value

These issues could degrade the performance of the system but are also necessary for proper functioning. The process of collision detection and data storage is as follows:

**Figure 12. Collision Detection & Data Storage**
Same process is done while getting the data from the table as if hash value matches the value stored, value is retrieved from the specific key.

### 1.3 Keys and Value Design

DBMS cannot directly access the data from the database as all data seems opaque to the DBMS. The only way to get the data is to use the key and information stored in the value is useless for data access, so keys are designed in a way that queries could be performed efficiently. Two keys cannot have same values as this could cause a clash of data retrieval. Performance of a system is also dependent upon the structure of key, if the key is lengthy; system requires more effort for the hash function. Lengthy keys can also adversely affect the performance of the system as it increases the size of the database, although the length of key is tradeoff between expected volume of database and performance of system. It is discussed earlier that data value are not known by the management system so any value could be stored in the database especially metadata which could be used by applications for parsing. In some systems values are directed towards fixed partitions and keys are used to hash particular location in a specific partition. The following figure is a representation of this strategy in which customer and orders are stored in specific partitions. Values are stored as XML data so a peculiar application can access the data and show it to the user.
Examples

There are various examples of database structures, which are using key value store, or its different classes and these are as follows

<table>
<thead>
<tr>
<th>Class of Key Value Store</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Key Value Eventually Consistent</td>
<td>Apache Cassandra, Dynamo</td>
</tr>
<tr>
<td>• Key Value Hierarchical</td>
<td>Intersystem Cache</td>
</tr>
</tbody>
</table>
2. Big Table Style Database

BigTable style database falls in the category of columns oriented database and all industries that are using column database are following the trend of Google BigTable so it is better to discuss the architecture of Google’s BigTable rather than discussing column oriented database. This is the particular database that is owned by Google with compressed structure for data storage along with high performance. It is developed on the file system of Google and its related technologies although any outside party could utilize the database through Google App Engine. This database is used to manage the data, which is in a structured form, and database can be scaled at very large size, even petabytes of data could be stored on different servers. Different projects of Google like Google Earth and web indexing are using Bigtable as these applications require huge data size and real-time data processing. This particular data model is used for more than 60 applications developed by Google.

Bigtable is not completely parallel with relational database as it consists of a simple model that provides an opportunity for the user to control the data format and layout in a dynamic way. Data is stored in the form of rows and columns along with indexes and provides the flexibility for the client to serve the data from the disk or outside the memory. The database is like sorted map with
rows its keys and columns are also indexed through a key along with value of time. Before going into the architectural details of BigTable database, it is important to discuss some of the problems that are involved in traditional database as compared to modern data models.

BigTable database provides an opportunity to store huge amount of unstructured data as compared to common relational model that is why such modern models are being used in social media application where possibility of data is around petabytes. It was not possible to add millions of rows for data storage in the traditional models, as these models do not provide enough power of data analysis as well as data storage.

![Figure 14. Architecture of BigTable Data Model](image)

The important and main architectural components of BigTable database consists of Tables, Tablets, Tablet servers, chubby server and SS Tables as indicated in above figure. Tables used in the database are further divided into tablets and these tablets are distributed over several machines. These tablets are composed of number of rows but limited with size of around 100 to 200 mega bits. The size of tablet is maintained by the tablet server as it grows, tablet server splits the tablet and these tablets are stored on Google file system.
SS Table is sorted string table and plays vital role in Big Table architecture as it is used to store huge number of keys and values and gives high throughout in read and write operation. Bigtable is designed to process huge data even in Gigabytes and random access is not good option for such type of workload so data should be in the form of stream for efficient processing. Sorted string table contains sorted keys and values and table can also contain duplicate keys. In case if large file is required to be read a special key/offset could be used for fast data access.

<table>
<thead>
<tr>
<th>Key</th>
<th>Offset</th>
<th>SS Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
<td>Key</td>
</tr>
</tbody>
</table>

**Figure 15. Bigtable and Tablets Structure**

**Figure 16. Structure of SSTable**
2.1 Rows

The keys in the rows consist of strings with a size from 10 bytes to 100 bytes typically and 64 KB with a maximum size. Read and write operation utilizes the basic row key irrespective of the read and write operation done in various numbers of columns. This design procedure makes the system more efficient regarding different data up gradations in the same row. The data is ordered in alphabetic order through the row key. The range of row in a certain table is called a tablet and it is portioned in a dynamic way. This attribute makes system more efficient by having short row range. This situation could be analyzed with an example of web pages of a single domain, which are grouped in a contiguous manner. In this way, domain and host analysis becomes more efficient.

2.2 Columns

The basic unit in Bigtable database for column keys grouping is called column families. Same data is stored or compressed in the column family or in the same family. The process of data storage starts with column family creation and then data could be stored under particular column key. Number of column families should be distinct and small and should not change in the operation. The example of column family could language and it must store the language of the web page. Now one column key would be used for the family of language and this family would store the ID of a language for every web page. There are many other functions, which are also performed at level of column family. These functions are accounting of memory and disk along with access control.
2.3 Time-Stamps

There are different cells in the Bigtable database and same data with a different version are required to be saved in these cells. If timestamps are not used in the table then there would be confusion of data versions so time is the indication of data version. A time stamp consists of an integer number with 64 bits and could be generated by database itself or its application. The order of data storage is done according to the time value and most recent data is stored at the start. The data is mostly organized in a way that only new versions are kept in the database and old version of data could be removed. For this purpose two column families are used to store new data and delete the old data.

2.4 Application Programming Interface

Controls of managing Bigtable database are associated with API as all creation and deletion of tables is done through this interface. Rights of access control are also handled by the application interface along with cluster changing and management of metadata in column families. The other features that are supported by Bigtable database are as follows:

- Atomic read, modification and write operation could be performed by using single key of any row
- The database does not allow general transactions by using row keys
- Interface for batching writes is available by using row keys
- Cells are used like counters
- Client scripts are also supported in the servers address spaces

Performance of Google database is enhanced by the API as it has different controls for accessing the data. Read, modification and writing of data could be managed in a particular row
irrespective of the number of columns being updated. Rows that are adjacent to each other could be accessed in single network call. Map reduce function could also be implemented on the Bigtable by using API. Performance of the system could be even increased by using client applications either than using API. Access and controlling rights and data serialization could be done according to the requirement. Deletion of data and updating of column and families could be done by client application. There are features could also be used in parallel with features given by API.

### 2.5 Implementation of Bigtable as Database

In basic implementation of Bigtable architecture one of the important section is client library. Master server is used for coordination and removable tablet servers are used for data storage. It is the responsibility of master server to assign different tablets in the tablet server. In this way master server distributes the load on all the servers and also manages the garbage files from the Google file system. It is shown in the figure below that master server is after client library but data from client does not move through master and client is capable to develop direct communication with tablets.

![Figure 17. Communication between Client & Servers](image-url)
Chubby is an important fault tolerant system used by Bigtable for storing files, metadata and coordination and there is one chubby in one data center. Fault tolerance is done by five replicas running in chubby cell, in which one is used as master for serving the requests. It also provides the directories and these directories are used as locks.

![Chubby File Structure](image)

**Figure 18. Chubby File Structure**

Chubby ensures that only one master is active at one time and it also stores the location of bootstrap for Bigtable. It also discovers the tablet servers, stores the information of schema as well as access control list. Rows in the Bigtable could be located in three discrete levels known as root tablet, metadata tablet and user tablet. It means root tablet stores the location of metadata tablet and metadata tablet stores the location of user tablet.

### 3. Document Oriented Database

It is the advanced model of BigTable database with number of improvements. The trend of Internet expansion accelerated the development of different databases to fulfill the ever increasing requirements of data and document oriented database is also one of part of that chain.

The major requirement of the system was to have a database that should be non-relational as well as scalable along with the quality of distributed system. As the name suggests the data is stored...
in the form of documents and these documents are in certain form or encoded. The languages used for encoding are as follows

- XML (Extensible Markup Language)
- YAML (Yet Another Markup Language)
- JSON (Java Script Object Notation)
- BSON (Binary JSON)

It is the database that is developed by self-contained documents means whole of the data for any document is stored in document shape unlike the tables in the relational database. The databases, which contain tables are according to certain schema so well structured tables are developed but document oriented database is schema free. Any addition of the new field could be easily added in the document without making any change in the other documents. In the database objects could be stored as document and documents could be complex, as it is not required to use insert statements for data storage but entire object could read at one time. Following is the list of databases, which are using document-oriented structure

- MongoDB, CouchDB, Terrastore, RavenDB
- OrientDB, SisoDB, RaptorDB, Jackrabbit

<table>
<thead>
<tr>
<th>Document Oriented Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoDB</td>
</tr>
</tbody>
</table>
### Document Databases

<table>
<thead>
<tr>
<th><strong>Database</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>CouchDB</td>
<td>In this database a document is an object with the field name which could be string or any integer. The database could be accessed with JSON API and query-able feature requires JavaScript to be used as query language.</td>
</tr>
<tr>
<td>Terrastore</td>
<td>It is also document oriented database with high scalability without inconsistency.</td>
</tr>
<tr>
<td>RavenDB</td>
<td>This is the document database for .NET and Windows platform and can store JSON documents in its database.</td>
</tr>
<tr>
<td>OrientDB</td>
<td>It is schema less and also schema mixed database as data are stored in the form of documents but a relationship also remains intact through graph database.</td>
</tr>
<tr>
<td>SisoDB</td>
<td>It is database for the SQL server although it is document oriented.</td>
</tr>
</tbody>
</table>

Some document databases are purely based on the file system and directories are used to manage the files and in this way documents are developed. In this type of file system tools for sorting and searching are limited. In some cases document oriented database is developed as top layer of other database in this way database is used to store the data contents and basic storage unit is a document.

The relational database model is different from the object-oriented model, as it requires lot of effort from the programmer for back and forth translation. This issue is resolved in object relational mapping but still remains various issues, which could be resolved by document-
oriented database. In document oriented database document could be mapped directly in the
language used by programmer. This is beneficial from the programmer’s end that it makes easy
programming but issues of data integrity could be raised. Document oriented database and its
architecture could be analyzed if one of the example or database model could be presented and
for this purpose Mongo DB is selected for detailed analysis.

Mongo DB is schema free database written in C++ for scalable data requirements. Database
does not store data as documents but format is BSON. It contains the support of indexing like
relational database, as all of the features of relational model are not eliminated in document-
oriented database due their performance in traditional models. It contains distributed storage
systems, which are named as shards. Mongo DB is developed to handle fast queries like
relational model. Documents are stored as arrays, integers, objects and hash tables and values of
collected documents are indicated by indexing. Mongo query optimizer sorts the documents with
help of index or sparse index which only includes documents. Indexes are always unique so that
no two documents could be stored with same reference.

![Figure 19. Communication of Application with Primary/Secondary nodes](image-url)
Mongo DB uses replication means sharing of information between various resources. Database especially focuses on master and slave replication. In Mongo DB architecture primary (master) nodes are those nodes that can allow write operation, these nodes are also called master nodes and one master can allow write operation at one time. If master node fails due to any problem then other nodes can selection one primary node. Client can send reads to the primary as well as to the secondary (slave) node. Replica set is the group of instances that share information to control situation of failure in the system. Horizontal portioning in the database is called sharding and is supported by Mongo DB for scaling, data distribution and load balancing in the machines.

![Figure 20. Architecture of Shards](image)

The cluster of shards consist of configuration servers, holding the metadata of clusters, mogos for routing the data from application servers. Shard keys are present in all documents and these documents are distributed in the form of chunks with same size. In huge data processing schemes shard is kept balanced by moving chunks to different shards and this process is done by mongo.
4. Full Text Search Engines

The emergence of unstructured text data motivated the organizations to use non-traditional databases and as result text search engines emerged after traditional database to store unstructured data. The text documents are not capable to fit into the traditional database or the databases, which are of table styles. The technology of search engine is relying heavily on such databases although traditional databases are also used in these applications up to some extent. The development also forced many companies to shift the entire system into text search engine.

![Diagram of Communication in Text Search Engine]

The main section of the database is main document index in which every row contains only single document and indexing process only starts by putting the data in the row through any application. The key section of main index contains the name of the file and some of the

---

Figure 21. Communication in Text Search Engine
additional fields like title could also be used in the database or could be changed according to the requirement. The process of document opening consists of following steps

1. External Document is opened after inserting the data
2. Ordered word list is created
3. Document is loaded in main word index
4. Above process is repeated by the engine through document index

In full text engine searching is done through the process of indexing and only those documents could be searched which are complete in their records. A search engine optimizes the performance during query through sorted document search by matching the contents of search with document title. Creation of document list as pointers to the main document index also optimizes the search. After whole of the process result list is displayed on the engine, which is actually list of the field contained in the section of main index. The following table shows the main advantages and technological perspective of search engine databases.

<table>
<thead>
<tr>
<th>Search Engine Database</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Technology</strong></td>
</tr>
</tbody>
</table>
The database is different from the most important perspectives like data structures, syntax of query, types of data and document formats.

It is the fact the full text search engine is similar to the relational database in many aspects but there are lots of differences between these two databases. One of fundamental difference is data type used for indexing and options for retrieving data. These features make database an innovative addition in scalable database technologies. The features that differentiate the database from others are explained in the following section.

**Analysis of Features**

In relational model program processes the query and as result data field in a row is returned back but in case of full text search engine meta-data is returned which represents the documents. The collection of data in the traditional model is assumed to be database but in this particular technology it is assumed as document. Outer joins and joins are not used in the database like relational model and most of emphasis is on reading the documents from database not writing. The database is optimized to store and process both unformatted and formatted documents and commands are also optimized for retrieving documents in flexible way. Query syntax is quite broader in full text engines as they can even support NOT, AND, OR and nested operations. Modern search engines support multiple type of syntax and could also be configured according to the requirement. Query operators supported by modern text engines may include

- Proximity Qualifiers
- Language Stemming
- Thesaurus
In relational model it seen that tables of database perform very important role but in full text model only few tables of documents and these are also called catalogs. They keys of documents are name of files or URLs. The format of document includes HTML, binary, text, pdf and XML etc. Following figure shows the architecture of full text SQL server 2008.

Figure 22. Architecture of Full Text Search Engine
Filter daemon host is considered as process started by engine for filtering and breaking the data from the tables. Word breaker locates the boundaries of a word and stemmer that is linked with word breaker finds verbs of that particular language. Filter is to process the document from formatted structure to textual stream. Stop list contains some set of words are called stop words. These are the words (a, are, is etc), which are not useful in processing queries and are ignored. Thesaurus files are empty and could be edited to store required information about the language and business according to the requirement. Gatherer or crawl thread is used to schedule indexes of search engine and monitoring the catalogs.

5. **Graph Database**

The pace of expansion in the Internet and digital system has not only increased the data but complexity of data is also increasing at the rapid pace. It is estimated by the IBM that 90% of the data that exists in the world is only created in last two years. Corporations that are linked with the financial activities and social networking have data even around petabytes. The most of the data is created though the web traffic and emails. Now the need of the world is to develop such systems that could be utilized to enhance the intelligence in data management as well as efficiency in the data processing so that this huge amount of data should not remain any headache for the organization working at such scale. It is expected that data would increase at rapid pace in the coming years and because of this different modern database models are discussed in the thesis to manage the data. Queries could be done in a fast way with the help of graph database and if the model is like data store structure queries are even faster.

Historically graph database is linked with graph theory, which is branch of mathematics. The foundation of graph theory is developed by Leonhard Euler. As the name suggests graph
database is just like graph, which consists of nodes and there is no indexing in the data storage like other databases.

![Data Structure of Graph](image)

**Figure 23. Data Structure of Graph**

One element is linked with other element through a pointer without any index. There are three main parameters of graph database and these are as follows

- Nodes
- Properties
- Edges

Nodes are the entities just like persons or any item that could be tracked in the database. Property is the particular information that links the nodes and edges are the lines between the nodes and these edges represent relationship between the nodes. A graph database is organized on the base of relationship and if the relationship is complex the structure of the graph would also be complex. Query in the graph database is actually traversal or navigation in the graph from
starting point to the destination node. Indexing in the database is used to avoid traversing whole of the graph to find any particular node. In the operation of traversal path track is kept between two vertices. The speed of traversal depends upon the presence of objects in the type of memory and type of graph database model. There are five different types of graph database and these are listed as follows

<table>
<thead>
<tr>
<th>Types of Graph Database</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Undirected Graph</td>
</tr>
<tr>
<td>Directed Graph</td>
</tr>
<tr>
<td>Multi Graph</td>
</tr>
</tbody>
</table>
Above representation shows the types of graph in a general way but types of graphs could be categorized in more specific way as follows:

- Labeled Graph
- Weighted Graph
- Property Graph

Figure 24. Specific Types of Graph
Architectures of graph databases are designed in such a way that they can operate on a single machine, and databases can be accessed from remote sites through any web-based application. Writing processes in the database are managed at the database level to ensure that no two processes can write to the database simultaneously. The database can work efficiently in both single-server and distributed environments. There is no comparison of graph databases with relational databases as relational databases are also an important and useful data model in the world of computer science, but as far as the interconnection of components is concerned, graph databases are better than relational databases.

**Examples**

There are several database projects that are based on graph models, and some of these are as follows:

- Allegro Graph
- Bigdata
- Neo4j
- R2DF
- OrientDB
- Infinite Graph

The importance of metadata model and query language for graph databases can be ignored at this stage of discussion as RDF and SPARQL provide a standard for data updating in graphs and queries. RDF actually describes the graph in the form of source vertex, type of link, and target vertex. SPARQL is the query language for graph databases that is used as a high-level language. The
language is somewhat similar to the SQL but standardized for the graph database and is
developed according to the pattern of graph.

5.1 Resource Description Framework Centric Solutions

Graph database includes two major types of database solution in which one is generic solution
and other is RDF based solution. RDF is decentralized graph with labels and directions and
starting of arc occurs with a subject URIs along with predicate (property) and ending occurs with
object URIs. In other way RDF could be seen from logical way and from database perspectives.
RDF started as language for the presentation of metadata on the website but later turned into data
model for the database systems. According to the broader view of data model RDF fully fits into
the definition of data model with tools of real world entities. RDF is not only the standardized
solution but also developed on simple data model with powerful query language. There are
several advantages associated with RDF based database and these are as follows

- Tool-chain Interoperability
- Data Portability
- No Product Lock
- Long Life of Scheme

RDF-based solutions provide the facility to export the data without any code conversion or
changing in data structure. There are several abstraction layers in relational database
management system but in RDF system tool chain and libraries are available and could be used
with other DBMS. So the whole system could work in many layers with RDF performing its
function as one layer and at other layers common DBMS is performing its functions. There are
various solutions of RDF database system so if one solution is adopted for the DBM and at other
time a variation is required then other solution could be easily replaced with first adopted solution of RDF unlike non-RDF solutions. RDF is considered as solution of future as it is becoming the standard for the future web development and linked data. In contrary No-SQL solutions are now with short life according to the future perspective of data management.
V. CASES STUDIES

1. Case Study 1 (Document Oriented Database)

A company X was using traditional database for the marketing application and within few years the rows in the database tables increased up to millions. This change caused queries to take longer time and some of the features required for the customer were removed to make the system more efficient as the queries were taking more time to run. Offloading some queries from the system helped but very little and the backup database did not have enough margins and the continuous extension in the database tables would ultimately move towards system collapse. The idea of using CouchDB came into the mind of “Dave” who was working in the company X which is the document oriented database and it is similar to the database used by Amazon.

The feature of CouchDB is data storage just like JSON documents, and it also provides the view engine that could be used to write Javascript code for document selection in the database. The other features like bi-directional conflict detection and replications are also improved in this database. This database was then used as an archive database to move all the data in a single night. This was although a difficult task to move the data from a relational database into the document-oriented database, but the problem of scalability was removed by adopting this method.

2. Case Study 2 (Graph Database)

Viadeo is the social networking site in the United State and also used for the purpose of business with over 35 million users. The database system was previously based on MySql and the process of recommendation took around an hour. It was required to shift the entire system into real time system to handle such number of clients. The company is now using Neo4j graph database,
which is not only robust, but also highly scalable and dominating the relational database. The
database could manage the indexes in the graph along with 32 billion nodes, the same number of
relationships and 64 billion properties. The graph operations in Neo4j are highly efficient and the
most important benefit is open source.

3. Case Study 3 (Key Value Store Database)
Garrett Eardley is a software engineer working at Riot Games presented a talk about the game
known as the league of legends in which million of player are active in a day. Terabaytes of data
is created during the tracking and game play statistics and this should be presented in the real
time for the players. In mySQL the scaling of the system and developing additional feature is not
possible for this huge data storage and processing. So any other database solution that contains
the feature of scalability should be used and for this purpose different options of Hbase,
CouchDB, Cassandra and Riak were under discussion. The previous problems in the system like
conflict resolution control motivated the team to choose Riak, which is key value storage
database. Objects in the database consist of unique key value and stored in the bucket. The
capability of Raik is to store anything like images, text and XML, HTML documents. The
database can also store session; user data along with log files.

4. Case Study 4 (Big Table Style Database)
Google, Yahoo and other large companies like Amazon which are using websites cannot use a
relational database as this type of database is not developed to manage a huge amount of data.
The first company who created Big Table database is Google as this database is developed on a
new architecture with the capability to handle a large amount of data. In this database whole of
the data is stored in a single table, which have different dimensions of row, column and time.
There is great importance of time in the table as selection of data and deletion is done through
the versions. The database used by Yahoo is Apache Hadoop which is an open source database written in Java. This database is also based on Big Table architecture just like Google and distributed on more than 10,000 servers.

5. Case Study 5 (Full Text Search Engines)

Daffodil is an Indian IT based company with more than 100 employee and three major offices in U.S, Singapore and UAE. The company has launched two products that are entirely web-based and for the purpose of customer relationship management and ERP. Initially, company tried to use MySQL and other relational database for handling its applications, but scalability became an important issue with the passage of time. The old system of Daffodil was capable of the scalability, but it requires manual scalability with increasing customers, which was a tedious job. Company decided to shift the entire system to Google App Engine that would be used as full text search engine for the database management. Shifting and integration of system with Google allowed the company to accommodate the variable customers and they did not need to manage the customers in a manual way. The integration of database with Google App Engine, Daffodil is saving almost 80 hours of development in one month along with the benefit of scaling and simple database management. The application of Daffodil went live in half of the time by using Google SQL Cloud, which is required for the development by company itself.
VI. COMPARISON OF APPROACHES

The development in the field of computer science has also shifted the trend of data storage due to the limitation in the historical systems. There was a time when people used to work with limited data; so direct filing was used for data storage. This was a simple method as most of the databases worked with particular user interface, but this was not general method to store data, as one program was linked with only one type of system. Further development occurred, requirements of people changed as they need more data storage and more general models to manage the working at organizational levels. Different companies came into the field of database development to fulfill the data requirement of large organizations. New models were developed for data storage, but these traditional models were not able to withstand the working environment of twenty first century. The reason that this thing happened is quite obvious everyone. Computer technology and information exchange remained a rapidly developing field in 20th century, but the explosion of information technology happened in the last decade. Social networking was a new horizon for whole of the mankind and people dealing in such technology also required new ways and methods for data storage. These new technologies required an enormous amount of data, which was not possibly managed by traditional models. So modern data modeling techniques which were based on scaling came into the practical field and organization started using these data models.

1. Comparison of Approaches

In general comparison different data models are categorized on the base of generation and utilization of these models in the market. General comparison is as follows

<table>
<thead>
<tr>
<th>Generations of Data Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>First</td>
</tr>
<tr>
<td>Second</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Third</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Fourth</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
In the last four decades four generations of database models came into the market and now fifth generation is used in different systems of information technology and also passing through the phase of rapid development. The need of different database systems from generation to generation is the result of over increasing complex computer technology and size of data. A general comparison of different data models is shown in the above table but it also needs detailed elaboration to reflect exact difference or similarities between these data models. First generation of database system includes data storage in files such as ISAM. In this system developer uses the programming interface to locate the file by searching in different indexes unlike query optimizer for index selection in an automatic way. This system was upgraded in next generation of data modeling and in file system it is discussed that one folder may contains one or many files so there is one-one and one-many relationship between different entities. It means this is the hierarchy in which one has to navigate to find the particular entity. IBM’s information management system that is based on hierarchical model is so widely used in the practical filed that hierarchical system is called IMS or IMS data model. In comparison to the file system and hierarchical model, relational database set a new trend towards data modeling as base of data model was sets and relation, which is mathematics. Associated algebra includes the operation of union, intersection, Cartesian product etc. Data is now organized unlike files system and presented in the form of tables that contains rows and columns. All the information defined in the rows with attributes in the columns. The reason for the 3rd generation of relational database
technology is the result of analysis done by the experts that 1\textsuperscript{st} and 2\textsuperscript{nd} generation technology is not suitable for the data sharing through environment of any standard application and navigation is not even suitable for such shared environment. Main aspect of 3\textsuperscript{rd} generation relational data model is declarative query formula for data retrieval and is one the widely used database in all generations. Relational approach divided the application and database into two different segments so that program remains less sensitive to physical changes in the data and metadata. This means physical and logical layers are separate and tables more understandable than pointers so performance of system also increased during record searching. Relational database model was performing well in the market but problems of integrating object oriented system with relational database forced developers to develop a system that would have attributed of relational database with additional object oriented features. This feature allowed new database system to store objects in the tables of relational database and that is why this DBMS is called hybrid model. This model is quite different from the object oriented model which also developed in fourth generation but these 4\textsuperscript{th} generation model did not find a way to replace 3\textsuperscript{rd} generation models. These models were used in different database systems along with relational model and opened variety of options for the users during development.

All data models until fourth generation could be considered as traditional data models and a need of fifth generation rich database is developed which should be able to meet the requirements of data storage and processing for modern applications. Before highlighting the benefits of fifth generation technology in comparison to modern modeling techniques, it is very important to elaborate some of the drawback of traditional models. Relational data model is quite simple and cannot tackle complex and nested entities. These complex entities include complex designs, documents and objects related with the field of engineering. Traditional databases could be used
for limited atomic values and support is not present for the data types used in programming. Traditional databases, particularly relational model could not be used for intensive computing and complex simulations. These databases are also unfit for temporal dimensions of modern applications like time, schema and objects’ versions.

The drawbacks of traditional databases induced a thrust of fifth generation models, which could fulfill the requirements of modern world. Fourth generation technology was developed on the base of third generation technology but scenario was entirely different for the 5th generation. In reality fifth generation database models could be divided into two major segments one is related with initial web application, multimedia, data warehousing in 1990’s. The second phase started in twenty first century when web technology shifted and started XML and social networks.

<table>
<thead>
<tr>
<th>Comparison of Database</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relational Model</strong></td>
</tr>
<tr>
<td>In relational model one instance is represented in each row and one column represents one attribute. Data types defined in each column is same as shown in the above figure the column of driver identification contains an integer data.</td>
</tr>
</tbody>
</table>
Modification in the relational database is according to scheme and if any new entity attribute is required to be entered into the table the definition of table required some changing.

In case of key value data model new entity attribute could be added by just adding a new row in database.

The value column contains different data types which makes search difficult in the key value database.

<table>
<thead>
<tr>
<th>Relational Model</th>
<th>Graph Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational database is suitable for low data storage applications.</td>
<td>Graph is highly scalable data models and suitable for high data modern applications</td>
</tr>
<tr>
<td>No defined navigation is available in relational model as similar data is stored in different tables.</td>
<td>This is schema free architecture and nodes can store dynamic properties and one node is linked with other node. This attribute provides better navigation in retrieving the data.</td>
</tr>
<tr>
<td>Consistency and Integrity of data is quality of this model.</td>
<td>This feature could be implemented in database through Application interface.</td>
</tr>
<tr>
<td>This database is not developed on object oriented methodology but tables are used in</td>
<td>Model follows object oriented approach and</td>
</tr>
</tbody>
</table>
place of objects. queries are used to traverse in the graph

<table>
<thead>
<tr>
<th><strong>Graph Database</strong></th>
<th><strong>Document Model</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Base of graph model could be considered as document modeling approach. Nodes are directly linked with each other and there is no relational joins.</td>
<td>In document modeling approach no edges are used like graphical modeling technique and connections are available in single direction and for bi-directional structure two way connections are developed.</td>
</tr>
<tr>
<td>Edges in the graph also increase the size of database as these are also records for the database.</td>
<td>There are no edges in the database, so size of database remains according to the size of documents.</td>
</tr>
<tr>
<td>Traversing also requires loading of edges so extra burden on the database which makes the process slower.</td>
<td>Data loading becomes faster as compared to graph database due to absence of edges</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Big-Table Database</strong></th>
<th><strong>Google Full Text Search Engine</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the database used by Google for data storage and developed on the top of GFS (Google File System).</td>
<td>This data store is developed on the top of Big-Table and simplifies the process of accessing Big-Table.</td>
</tr>
<tr>
<td>It could be considered as internal database which is not directly accessed by external users or could be consider as base of Google</td>
<td>It allows the third party application to be run on infrastructure developed by Google.</td>
</tr>
</tbody>
</table>
application Engine.

| Its structure is similar to key value store but mapping of key to value is not done in this case rather mapping of tuple is done in which row, column keys and timestamps are mapped to value. | Objects are stored in the serial manner and row key are used as identifiers. |

The relational database is primary based upon tables as compared to other modern NoSQL database management system. These databases are required to use SQL for managing the data and preferable to be used in the system where huge data is either processed or stored. The shift in technology for using modern data models does not mean relational database is eliminated but still in use for the systems which are managing small or medium size databases. The reason of using modern data models was the only matter that relational database was not able to handle huge amount of data with real time processing. The relational model consists of tables in which data is given and data present in one table is collected on the base of relationship with each other. There could be number of tables in the database and these tables also have some type of relationship with each other as shown in the figure below.
As compared to relational model key value store also based upon tables but in somewhat different manner. This table contains keys and values and keys are numerical or alphabetical identifiers. Data is searched by finding a particular key or a particular value. Design of key value store is simpler than relation database so data could be retrieved in very small time period. This is the reason that most of organizations that deal with selling of different items use key value store database for searching items. Now as compared to key value store document database is designed to store the documents. XML or JSON is used to encode these documents and value column of database stores semi-structured data. In this type of database not only the key are searchable but values could also be searched. Database is quite able to store documents in different formats and documents with huge size could be even stored in the database. Graph data model replaced the relational model with relational and structured graph and this model become
similar to the object-oriented with network of nodes. This model is helpful in the area where relations of objects or data is required instead if someone is interested in the analysis of data.

2. Applications/Fields

Key value store database architecture is used in Amazon Dynamo database, Berkeley database for new type of the applications required by the Internet and in many other applications. Voldemort is the key value store database used by LinkedIn. Massive web based applications are managed by the Aerospike with more than 39.5 billion data transactions in one day. Google developed Level database with sorted keys and capability to do number of changes in the database at one time. Document oriented databases are used by many online applications in which documents could assessed and different tasks of data management could be performed. Couch and Mongo databases are major examples of document-oriented database and Couch DB is used by BBC, Ubuntu, Meebo etc. Mongo DB is very popular database management system and used by many other organizations for document based web applications. Graph database is used in social networking and solving routing issues. Graph database is also used by the telecommunication networks and managing data for information security. Neo4j is famous graph database used all around the world in leading organizations. This database is used by Huwaei, Telenor, and Global telecommunication for managing telecom networks. Adobe, ebay, carrier builder and Cisco are also using graph database for their applications. Some of the software which are using full text search engine mechanism are an Elastic search, Xapian, BaseX and Searchdaimon etc. Most famous search engines used by organizations themselves is Google, yahoo and Bing etc. Bigtable database is developed by Google itself and is used with Google application interface, although many other organizations have also developed databases, which are same as Google Bigtable. These databases are Apache Casandra, Hypertable, Apache
Accumulo and KDI etc. Apache HBase is currently used by famous social networking website Facebook for managing its message application.
VII. RECOMMENDATIONS

Scalability and real time processing with millions of users is one of the basic reasons for using modern data modeling methodologies but it does mean that traditional models could be eliminated from the field. In this section of the paper features of modern data models would be analyzed in the light of their suitability to be used in the practical field.

Systems like shopping cart, profiles of different users and list of sessions require such a database that can store single value and key in the database. Users can select any item by using a particular key. In the same way profiles are accessed with single string or key. Applications that are linked with key-value store can easily write into the database by entering the key. Now the point of discussion is scalability of the database because there are a large number of users for such systems and database should be able to manage a huge amount of data and real time processing. Key-value store is developed on the base of simple structure of key and values, more flexible than relational model for search into the database. It is quite suitable for such applications, even Amazon is using key-value store for its shopping cart. Document oriented database stores data in the form of encoded documents and it should be used in the applications, which often require changing in the data or there is frequent change of data requirement in the documents but there are several other points, which should be given importance while using document database. Relational database has been traditionally used in various applications and no one can say that document database is replacement of relational model. Important aspect is to give proper value to the modern system according to the requirement and functional capabilities. Suppose if in some system consistency is the most critical factor than relational model could not be replaced by document-oriented database. Examples are banking system and security program of any country. In case if important factor is availability of data, then document database is better
as it is faster and scalable. Document database is not suitable for the connected data with high complexity as graph data structure. Graph database is mostly used in the applications of social networking and it is also suitable for applications in which complex routing problems are involved because different algorithms are developed for finding complex routes. Structure of computer networks is quite similar to graphs and if these networks are using a graph database then problem at any network node could be easily identified. There are several applications in which direct and indirect relationship of customers or any other factor is an important parameter for the working database; here graph database should be used as whole of the database is developed on structured relations but much faster and flexible than traditional model like relational database management system. It means for quick data retrieval from the database system having relationships, one should always use a graph database but for generalized operations and queries relational model can perform the required function. Graph database cannot be used in the place where huge structured entities are used but if it is necessary to be used, it should be used as hybrid model, which means it could be used along with other model. In this specific case of large entities, graph database should be used with document database. BigTable is distributed data storage architecture and used in such application, which are dealing with data at large scale. This type of database is also feasible for real-time applications and is currently used by Google earth, Google finance and web indexing. This is also one of the scalable database models but in a different and simple way as database could be extended by having more machines for storing the data. Full text search engine database provided new features of data storage and database indexing. These databases are not only useful for text searching but also feasible for indexing of SQL database as well as indexing of any other NoSQL data storage technology.
VIII. CONCLUSION

Advancement in computer technology created a need of data storage. Traditional database models fulfilled the requirement of data storage for few decades but global and social networking placed a new challenge in front of computer experts. This time challenge was quite huge as petabytes of data management put a limit on the traditional SQL based DBMS. This change of trend in technology directed the experts to develop such a system that is not only feasible to handle current data load, but it should be scalable for future emerging data requirements. New technologies emerged into the market and these technologies replaced traditional methods for handling big data storage. Relational database management system remained active in small organizations but was not able to be used in large applications. Various organizations started developing scalable database management system. Some of these organizations developed the systems to cater their own data, while other organizations started working on the platforms that could help other organizations not only to manage their data but also provide data integrity, robustness and flexibility in the system. These modern database technologies proved several other benefits in the practical filed such as fast computation, which is the basic requirement of modern networks with rapidly increasing number of users. Load balancing of modern database management systems is also a great achievement as compared to traditional models. Different organizations are working on scalable database systems and there is enough margin of expansion in this field.
REFERENCES


# APPENDIX

<table>
<thead>
<tr>
<th>Sr #</th>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>2</td>
<td>DB</td>
<td>Data Base</td>
</tr>
<tr>
<td>3</td>
<td>MS</td>
<td>Microsoft</td>
</tr>
<tr>
<td>4</td>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>5</td>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>6</td>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>7</td>
<td>ID</td>
<td>Identification</td>
</tr>
<tr>
<td>8</td>
<td>OID</td>
<td>Object Identifier</td>
</tr>
<tr>
<td></td>
<td>OODBMS</td>
<td>Object Oriented Database Management System</td>
</tr>
<tr>
<td>---</td>
<td>--------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>11</td>
<td>JSON</td>
<td>Java Script Object Notation</td>
</tr>
<tr>
<td>12</td>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>13</td>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>14</td>
<td>U.S</td>
<td>United State</td>
</tr>
<tr>
<td>15</td>
<td>U.A.E</td>
<td>United Arab Emirates</td>
</tr>
<tr>
<td>16</td>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>17</td>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>Page</td>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>18</td>
<td>YAML</td>
<td>Yet Another Markup Language</td>
</tr>
<tr>
<td>17</td>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>18</td>
<td>SPARQL</td>
<td>Simple Protocol and RDF Query Language</td>
</tr>
<tr>
<td>19</td>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>20</td>
<td>VSM</td>
<td>Value Stream Mapping</td>
</tr>
<tr>
<td>21</td>
<td>UDB</td>
<td>Universal Database</td>
</tr>
</tbody>
</table>