HBRP PUBLICATION

Journal of Advances in Computational Intelligence Theory
Volume 5 Issue 2

Database Management System: Data Model

Deepak Prajapat*[1*]*, Akanksha Kulkarni
[1] *Student, Dept. School of Engineering, Ajeenkya D.Y. Patil University, Pune, India*
[2]*Professor, Dept. School of Engineering, Ajeenkya D.Y. Patil University, Pune, India*

***Corresponding Author**
***E-mail Id:**-deepak27prajapat@gmail.com*

ABSTRACT

In this paper, we are talking about the modern application require database to catch and execute more in closure semantics than traditional application. In order to collect and analyse data, a database management system communicates with applications, end users, and the database itself. A database is a structured collection of data that is electronically accessible and stored. Large databases are housed on clusters of computers or cloud storage, whilst smaller ones can be maintained on a file system. Data modelling, effective data representations and storage spaces. Languages, security and privacy of private information, and distributed computing concerns that permit concurrent access are all part of the design of databases. The primary tools offered to manage the database are also included in the DBMS software. A database system is the collective name for the database, the database management system, and any related applications. In this paper, we revisit the data models, in which such models play a role in the process. Analysis of the data objects and how they relate to other things is done through the process of data modelling. It is employed for the analysis of the data requirements needed for business processes. On the other hand we also talk on other essential part of data model and its types like relational, hierarchical, network and entity relational (ER) model.

Keywords:- *Data Model, Hierarchical Model, Network Model, Entity-Relationship Model, Relational model, Object-Oriented Data Model*

INTRODUCTION

Data models are frequently used in database management systems to display how data is linked, stored, accessed, and modified. In order for members of an organisation to understand and comprehend the information and then communicate, we represent it using a set of symbols and terminology.

It is underlying structure of a database; conceptual tools in the design prospective are physical, logical and view level. Our understanding of the ultimate system's appearance after full implementation comes from the data model. It describes the data elements as well as their connections. The first thing to understand is the data model is a shell. A data model is an abstract model that organise element of data in a certain format, most data models also include a set of basic operation for specifying data retrieval and data updating.

Building a data model is a crucial stage in the architecture of the data warehouse because the data will actually be stored within this structure on a database.

The bulk of the money spent on building a data warehouse goes towards feeding the model with information coming from your source system, not into designing the models themselves, which is where the

major costs and labor-intensive tasks are involved.

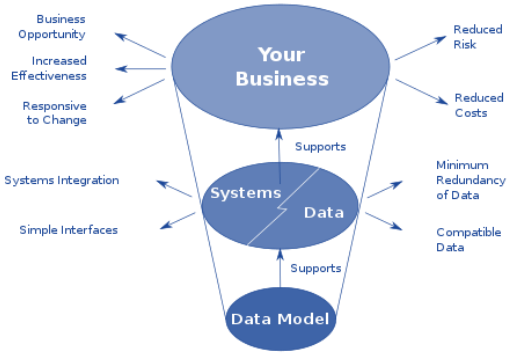

Fig.1

Data models is a high level design which defines the kind of tables, the field in those tables and the relation between different tables. Data model gives the overall logical view of database model specially in term of their entity relationship whereas database model is more concentric on entity and their attribute. To make some potential extensions, we examine the E-R model's structural layout. To give the extensions motivation, we use examples to exemplify them. We also go over the broader topic of semantics capture and the function of conceptual models.

TYPES OF DATA MODELS IN DBMS

Although alternative data models are still in use today, the relational one is the most common. In addition to the relational approach, there are other other data structures that we will go into great detail about in this post. The following are examples of data models in DBMS:

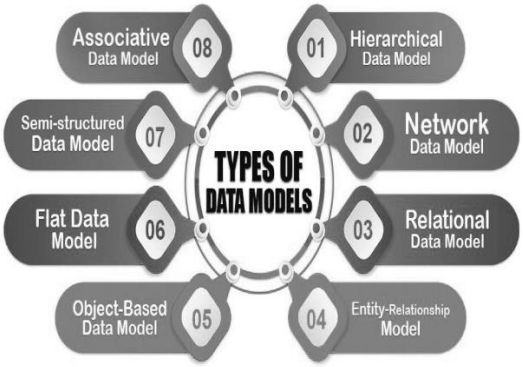

Fig. 2

Hierarchical Model

It was the first DBMS model. The data in this idea is organised using a hierarchical tree structure. The root of the hierarchy, which includes the root data, is where it all starts. As child nodes get added to the primary node, the hierarchy develops into a tree. It is one of the most established data set models created by IBM for data The executives Framework. In a various levels data set model, the information is coordinated into a tree-like design. In basic language we can say that it is a bunch of coordinated information in tree structure.

This sort of Information base model is seldom utilized these days. Its design resembles a tree with hubs addressing

records and branches addressing fields. The windows library utilized in Windows XP is an illustration of a various levels data set. Arrangement settings are put away as tree structures with hubs. The main drawback of this model is that, it can have only one to many relation-ship between nodes. Hierarchical models are rarely used now.

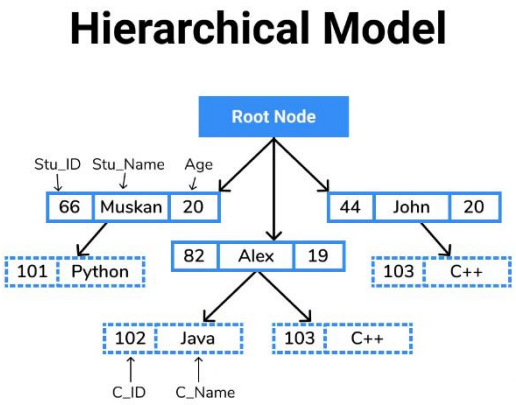

Fig.3:-*Hierarchical model [1]*

Network Model

This model is an extension of the hierarchical model. The main difference between this model and the hierarchical model is that any record can have several parents in the network model. Similar to a hierarchical model, a network model has a structure that is more like a graph than a tree. It was the most popular model before the relational model. It is a straightforward and simple to-build information base model.

The Organization Model in DBMS depends on the set hypothesis (numerical set hypothesis), so the model of the data set is built with the arrangement of related records (information).

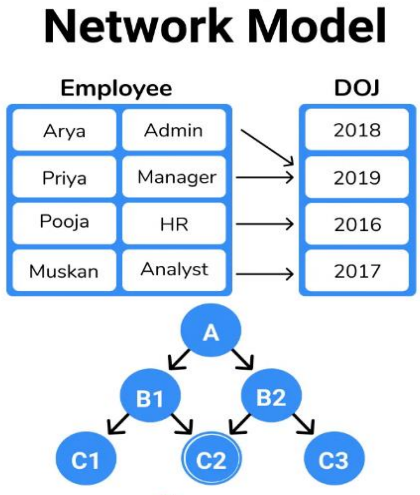

Fig.4:- *Network model [1]*

Entity-Relationship Model

In this model, the real-world issue is represented visually to make it simpler for stakeholders to understand. Additionally, the ER diagram makes it very easy for engineers to understand the system. E-R

model is a high -level data model diagram. E-R model describe the structure of a diagram with the help of a diagram, which is known as entity relationship diagram (ER diagram). An ER model is a diagram or blueprint of a database that can later be implemented as a database, it is based on the notion of real-world entities and relationship among them.
ER diagram has the following three components :

Entities

-

Entity is a real-world thing or object. It can be a person, place, or even a concept.
Example: Teachers, Student, Courses, Buildings, Department etc.

Attributes

An entity contains a real-world property called attributes. This is the characteristics of that attributes.
Example:

- The entity teacher has the property like teacher id, name, salary, age etc.
- Student has the property of doing or reading any course.

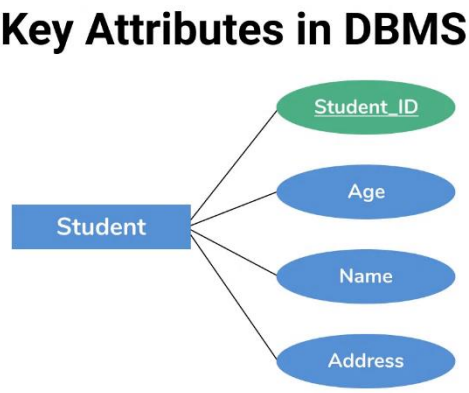

Fig.5:-*Key Attributes* [1]

Relationship

The association among entities is called a relationship.
For example:

- An employee works at a department, a student enrols in a course. Here, works _at and enrols are called relationships.
- Teacher works for a department.

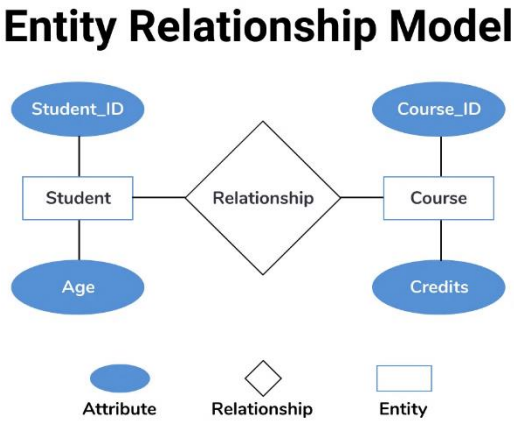

Fig.6:-*ER model [1]*

Relational model

The data in this model is kept in the form of a table that is two-dimensional. A technique to conceptually representing and managing the data kept in a database is the relational model for managing a database.

The data in this model is arranged into a set of two-dimensional, mutually exclusive tables, or relations. Each relation is made up of a set of rows and columns, where the columns correspond to an entity's properties and the rows, or tuples, to its records.

A simple, effective, and adaptable method of storing and accessing structured data was made possible by the use of databases to store the data. This data model's simplicity makes it simple to access and sort the data. As a result, it is widely utilised for data processing and storage globally. All of the data is kept in the form of rows and columns. Tables are the foundation of a relational paradigm.

- A relational database is built on the relational paradigm, as was previously explained. Various parts of this database are built using the relational approach. These consist of:
- **Relation**: A table with two dimensions used to store a group of data components.
- **Tuple:** Row of the relation, depicting a real-world entity.
- **Attribute/Field:** Column of the relation, depicting properties that define the relation.
- **Attribute Domain:** It describes the lawful numbers that an attribute can take and is a set of established atomic values.
- **Degree:** It is the total number of attributes present in the relation.
- **Cardinality**: It identifies the total number of rows included in the connection and thus the number of entities included in the relation.
- **Relational Schema**: It describes the relationship's structure and design; it is the relationship's

Relational Model Concepts

- logical blueprint

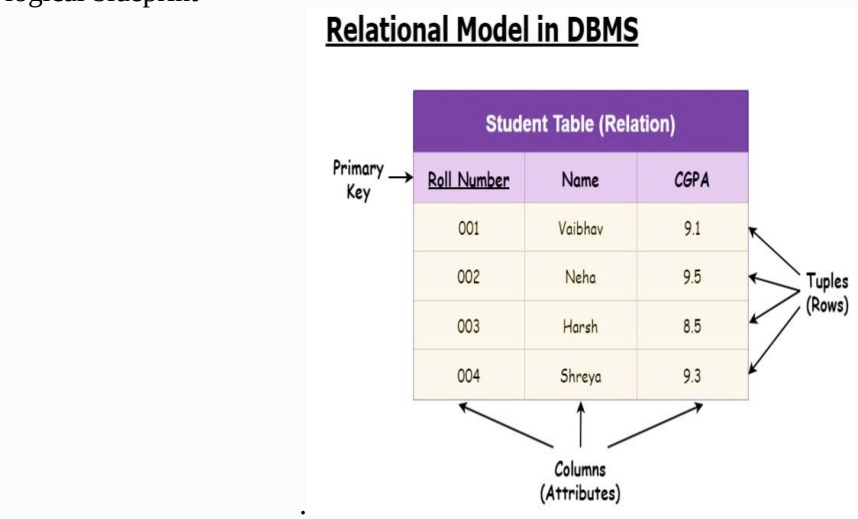

***Fig.7:-** Relational model [1]*

Object-Oriented Data Model

In this approach, an object is a single structure that contains both the data and the relationship. Both the data and the relationship are contained in a single structure that is known as an object in this model. Two or more objects are connected through links. We use this link to relate

one object to other objects. We can now store audio, video, pictures, and other types of data in databases, which was previously impossible with the relational approach (Although you can store video and audio in relational DB, it is advised not to store them in the relational database.

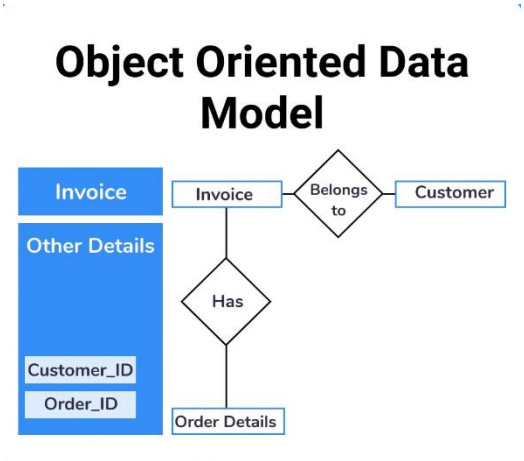

Fig.8:- *OODM [1]*

Object-Relational Data Model

It is a hybrid of relational and object-oriented models. In this both data and their relationship are organised or contained in a single structure known as an object. To fill the gap between object-oriented and relational models, this model was created. To represent complex real-world problem there was a need for a data model that is more closely represent the real world. it is also important to note that object-relational database is based on both the relational model and the object-oriented database model. Object include information about relationship between the facts within the object, as well as information about its relationship with other objects.

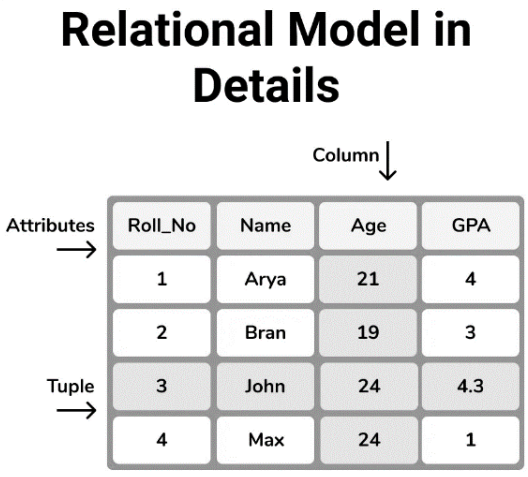

Fig.9:-*Relational model [1]*

Flat Data Model

It's a simple model where the database is shown as a set of tables with rows and columns. A data model is a conceptual representation of the data structure that are required by database. The data model is comparable to an architect's architectural designs, to use a typical comparison. A data model is not restricted by hardware or software.

Flat File Model

	Route No.	Miles	Activity
Record 1	I-95	12	Overlay
Record 2	I-495	05	Patching
Record 3	SR-301	33	Crack seal

***Fig.10*:-**

Semi-Structured Data Model

Semi-structured model is an evolved form of the relational model. The semi-structured data model provides for data specifications in situations where distinct sets of attributes may be present in separate data items of the same type. The semi-structured model has developed from the relational model. We are unable to distinguish among data and schema in this model. In this model, some entities may have missing attributes while others may have an extra attribute.

Data innovation in the medical services area is moving from structure-based information to semi-organized, difficulties of large information, for example, how to manage expanding information volume and the requirement for a semi-structured information type to store and deal with enormous sums of information with an adaptable outline. This model gives flexibility in storing the data. It also gives flexibility to the attributes. It provides a flexible format for data exchange between different types of databases.

It can be helpful to view structured data as semi-structured. The schema can easily be changed. The data transfer format may be portable.

As an illustration, if we store any information in any attributes, that information may be either a single value or a group of related information.

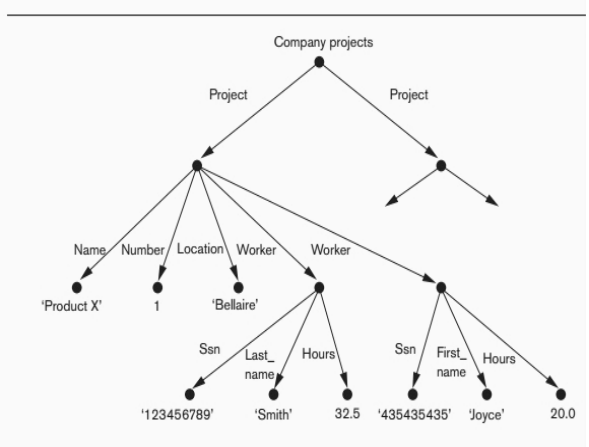

Fig.11*:-*Semi-structured [1]

Associative Data Model

It is a model in which the data is separated into two sections. Everything that has its own existence is referred to as an entity, and the relationships between these entities are referred to as associations. In this, a cooperative cluster is utilized as an essential data set in which a singular key is connected with only one worth in an assortment. For the qualities, keys are extraordinary identifiers. Any sort of element can be esteemed. The assortment of key-esteem matches put away on independent records is called key-esteem data sets and they don't have an all-around characterized structure. Items and links are two types of data that are separated into two components.

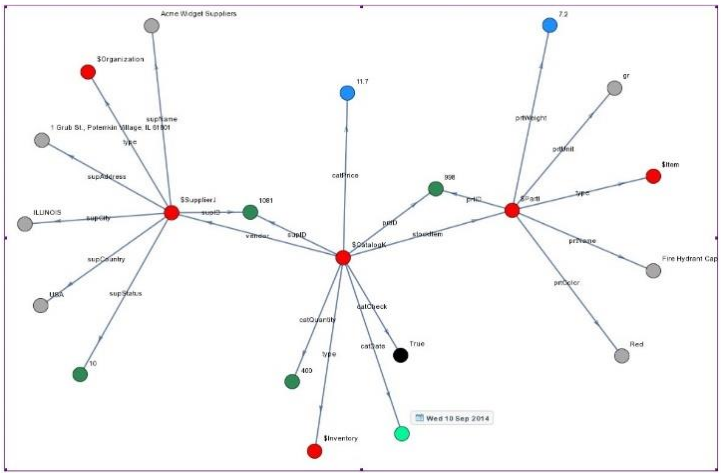

***Fig.12**:- Data model [1]*

Context Data Model

A variety of models make up the context data model. This contains several models, including relational and network models

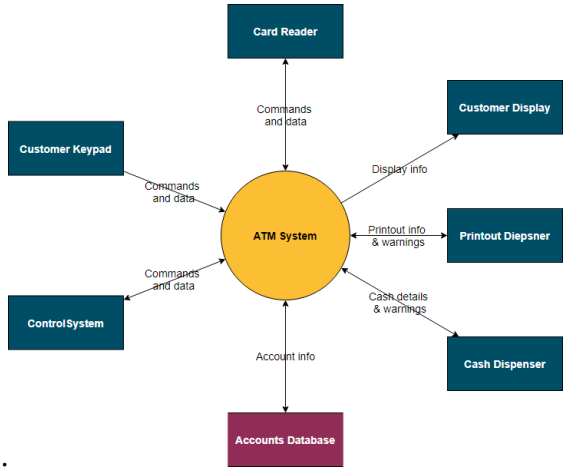

***Fig.13**:-Context Data model [1]*

This information model empowers you to do different things that you won't have the option to do assuming you utilize just a single information model.

DISCUSSION

As said within the starting, any extension of a conceptual model should be well stimulated. Since the E-R model guy-ages to stability expressivity and complexity

quite well, it should also be discussed whether or not any extensions are necessary. We argue that new programs, from E-trade to Decision Support to Document and Workflow Management, demand more semantics from database structures. Traditional applications, primarily based on a version of small, nearby transactions, worried easy, localized database get admission to. Today's applications require complicated, worldwide information manipulation.

CONCLUSION

E-R model are widely used because of their simplicity, intuitive appeal and ability to capture beneficial semantics across many diff erent domain names. In the past, it could have been argued that, at the same time as having more semantics is continually suitable, the E-R model off ered a good exchange-off point. However, with databases called to aid greater complex and sophisticated applications, the want to capture extra area semantics is growing. The structural facts that the E-R model contains supply us enough to determine ordinary paperwork, which in turn is the premise to check the correctness of the database schema.In the end, it is clean that no conceptual model will ever capture all the semantics of an utility, and that there isa need to balance expressiveness and complexity. However, we have to periodically revisit the boundaries of our models in a try and increase them. Precisely because of the use-fulness and significance of the E-R version, we ought to revisit it often.

REFERENCES

1. *All the figures are reference from prepinsta.com. A device for conceptual layout of information and transactions in the entity-courting version, in Proceedings of the second International Conference on Entity-Relationship Approach to Information Modelling and Analysis, 2011.*
2. *Camps Pare, R. From Ternary Relationships to Relational Tables: A Case Against Common Beliefs, SIGMOD Record, v. 31, n. 2, June 2002.*
3. *Castellanos, M. Semantic Enrichment of interoperable databases, in Proceedings of the RIDE-IMS, 2019.*
4. *Chen, P. The Entity-Relationship Model -Towards a Unified View of Data, ACM Transactions on Database Systems, v. 1, n. 1, 2019.*
5. *Connolly, T., C. And Strachan, A. Database Systems, Addison-Wesley, 2015.*
6. *Davis, F. Requirements Specification: Objects, Functions and States, Prentice-Hall, 1993.*
   Dos Santos, C. S., E. J. And Furtado, A.L. A Data Type Approach to the Entity-Relationship Model, in Proceedings of the first International Conference on Entity-Relationship Approach to Software Engineering, 2013.[8] Eder, J., Kappel.
7. *Gogolla, M. And Hohestein, U. Towards a Semantic View of an Extended Entity-Relationship Diagram, ACM Transactions on Database Systems, v. Sixteen, n.3, 2019.*
8. *Streams D.M. (2018).* Power-Mindful Microar-chitectures: Plan and Difficulties for Cutting edge Microchips.. *IEEE Micro. 20(6):26-44.*
9. *Skadron K et al(2015).* Temperature-Mindful Com-puter Frameworks: Open doors and Challenges. *IEEE Miniature.23(6).*