ASSIGNMENT

SUBMITTED TO:Sr.ELSIN CHAKKALACKAL S.H

DEPARTMENT: MCA

AMAL JYOTHI COLLEGE OF ENGINEERING

SUBMITTED BY: ATHIRA BIJU

DEPARTMENT:MCA

CLASS: REGULAR MCA2021-23B

ROLL NO:4

A STUDY ON DIFFERENT DATABASE MANAGEMENT SYSTEM

Data

Data, in the context of databases, refers to all the single items that are stored in a database, either individually or as a set. Data in a database is primarily stored in database tables, which are organized into columns that dictate the data types stored therein. So, if the “Customers” table has a column titled “Telephone Number,” whose data type is defined as “Number,” then only numerals can be stored in that column

What is a Database?

A database is a collection of related data which represents some aspect of the real world. A database system is designed to be built and populated with data for a certain task.

Database Management System

Itis a software for storing and retrieving users’ data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

DBMS allows users to create their own databases as per their requirement. The term “DBMS” includes the user of the database and other application programs. It provides an interface between the data and the software application.

1. Hierarchical Databases

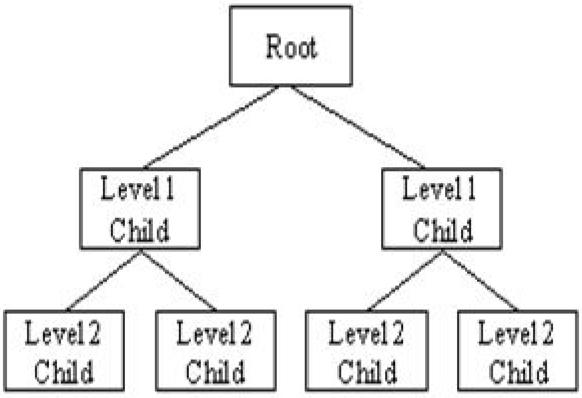
In a hierarchical database management system (hierarchical DBMSs) model, data is stored in a parent-children relationship node. In a hierarchical database, besides actual data, records also contain information about their groups of parent/child relationships.

In a hierarchical database model, data is organized into a tree-like structure. The data is stored in the form of a collection of fields where each field contains only one value. The records are linked to each other via links into a parent-children relationship. In a hierarchical database model, each child record has only one parent. A parent can have multiple children.

To retrieve a field's data, we need to traverse through each tree until the record is found.

The hierarchical database system structure was developed by IBM in the early 1960s. While the hierarchical structure is simple, it is inflexible due to the parent-child one-to-many relationship. Hierarchical databases are widely used to build high-performance and availability applications usually in the banking and telecommunications industries.

The IBM Information Management System (IMS) and Windows Registry are two popular examples of hierarchical databases.



Advantage

A hierarchical database can be accessed and updated rapidly. As shown in the figure above, its model structure is like a tree and the relationships between records are defined in advance. This feature is a double-edged sword.

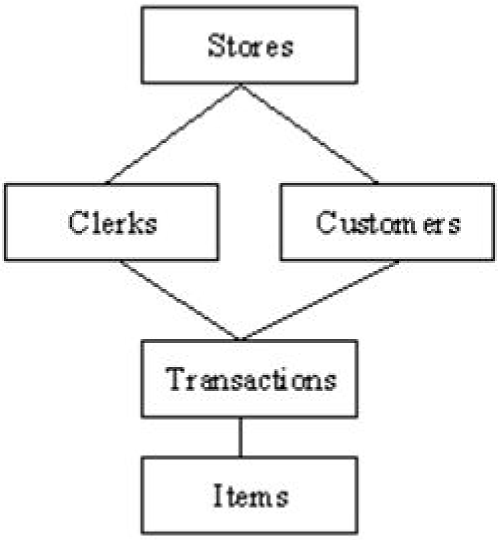
Disadvantage

This type of database structure is that each child in the tree may have only one parent. Relationships or linkages between children are not permitted, even if they make sense from a logical standpoint. Hierarchical databases are like this in their design. Adding a new field or record requires that the entire database be redefined.

2. Network Databases

Network database management systems (Network DBMSs) use a network structure to create a relationship between entities. Network databases are mainly used on large digital computers. Network databases are hierarchical databases, but unlike hierarchical databases where one node can have a single parent only, a network node can have a relationship with multiple entities. A network database looks more like a cobweb or interconnected network of records.

In network databases, children are called members and parents are called occupiers. The difference between each child or member is that it can have more than one parent.



The approval of the network data model is similar to a hierarchical data model. Data in a network database is organized in many-to-many relationships.

The network database structure was invented by Charles Bachman. Some of the popular network databases are the Integrated Data Store (IDS), IDMS (Integrated Database Management System), Raima Database Manager, TurboIMAGE, and Univac DMS-1100

3. Relational Database

This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation). A relational database uses SQL for storing, manipulating, as well as maintaining the data. E.F. Codd invented the database in 1970. Each table in the database carries a key that makes the data unique from others. **Examples** of Relational databases are MySQL, Microsoft SQL Server, Oracle, etc.

Properties of Relational Database

There are following four commonly known properties of a relational model known as ACID properties, where:

A means Atomicity: This ensures the data operation will complete either with success or with failure. It follows the 'all or nothing' strategy. For example, a transaction will either be committed or will abort.

C means Consistency: If we perform any operation over the data, its value before and after the operation should be preserved. For example, the account balance before and after the transaction should be correct, i.e., it should remain conserved.

I means Isolation: There can be concurrent users for accessing data at the same time from the database. Thus, isolation between the data should remain isolated. For example, when multiple transactions occur at the same time, one transaction effects should not be visible to the other transactions in the database.

D means Durability: It ensures that once it completes the operation and commits the data, data changes should remain permanent.

4. Object-Oriented Model

In this Model, we have to discuss the functionality of object-oriented Programming. It takes more than the storage of programming language objects. Object DBMS's increase in the semantics of C++ and Java. It provides full-featured database programming capabilities while containing native language compatibility. It adds the database functionality to object programming languages.

This approach is analogical of the application and database development into a constant data model and language environment. Applications require less code, use more natural data modeling, and code bases are easier to maintain.

Object developers can write complete database applications with a decent amount of additional effort.

The object-oriented database derivation is the integrity of object-oriented programming language systems and consistent systems.

The power of object-oriented databases comes from the cyclical treatment of both consistent data, as found in databases, and transient data, as found in executing programs.

Object-oriented databases use small, recyclable separated from software called objects. The objects themselves are stored in the object-oriented database.

Each object contains two elements:

1. A piece of data (e.g., sound, video, text, or graphics).
2. Instructions, or software programs called methods, for what to do with the data.

Object-oriented database management systems (OODBMs) were created in the early 1980s. Some OODBMs were designed to work with OOP languages such as Delphi, Ruby, C++, Java, and Python. Some popular OODBMs are TORNADO, Gemstone, ObjectStore, GBase, VBase, InterSystems Cache, Versant Object Database, ODABA, ZODB, Poet. JADE, and Informix.

5. NoSQL Databases

NoSQL databases are databases that do not use SQL as their primary data access language. Graph database, network database, object database, and document databases are common NoSQL databases. This article answers the question, what is a NoSQL database.

NoSQL database does not have predefined schemas, which makes NoSQL databases a perfect candidate for rapidly changing development environments.

NoSQL allows developers to make changes on the fly without affecting applications.

NoSQL databases can be categorized into the following five major categories, Column, Document, Graph, Key-value, and Object databases.

Here is a list of 10 popular NoSQL databases:

1. Cosmos DB
2. ArangoDB
3. Couchbase Server
4. CouchDB
5. Amazon DocumentDB
6. MongoDB, CouchBase
7. Elasticsearch
8. Informix
9. SAP HANA
10. Neo4j

6. Graph Databases

Graph Databases are NoSQL databases and use a graph structure for semantic queries. The data is stored in the form of nodes, edges, and properties. In a graph database, a Node represents an entity or instance such as a customer, person, or car.

A node is equivalent to a record in a relational database system. An Edge in a graph database represents a relationship that connects nodes. Properties are additional information added to the nodes.

The Neo4j, Azure Cosmos DB, SAP HANA, Sparksee, Oracle Spatial and Graph, OrientDB, ArrangoDB, and MarkLogic are some of the popular graph databases.

Graph database structure is also supported by some RDBMS including Oracle and SQL Server 2017 and later versions.

We live in a connected world, and understanding most domains requires processing rich sets of connections to understand what's really happening.

Often, we find that the connections between items are as important as the items themselves.

Although unhelpfully named, the NoSQL ("Not only SQL") space brings together many interesting solutions offering different data models .

database systems, each more suitable than traditional SQL solutions for certain use cases and shapes of data.

With the advent of the NoSQL movement, the "one-size-fits-all" proposition of large relational systems was replaced by conscious decisions about finding the right tool for the job.

7. ER Model Databases

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type.

In a relational database, a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

The entity-relationship model was developed by Peter Chen in 1976 with variants of the idea existing previously.

Some ER models show super and subtype entities connected by generalization-specialization relationships,[3] and an ER model can be used also in the specification of domain-specific ontologies.

An entity may be defined as a thing capable of an independent existence that can be uniquely identified. An entity is an abstraction from the complexities of a domain.

When we speak of an entity, we normally speak of some aspect of the real world that can be distinguished from other aspects of the real world.

An **entity–relationship model** (or **ER model**) describes interrelated things of interest in a specific domain of knowledge.

A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

Entities may be characterized not only by relationships, but also by additional properties (*attributes*), which include identifiers called "primary keys".

Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity–relationship models.

8. Document Databases

Document databases (Document DB) are also NoSQL databases that store data in the form of documents. Each document represents the data, its relationship between other data elements, and attributes of data. Document database store data in a key-value form.

Document DB has become popular recently due to their document storage and NoSQL properties. NoSQL data storage provides a faster mechanism to store and search documents.

Popular NoSQL databases are Hadoop/Hbase, Cassandra, Hypertable, MapR, Hortonworks, Cloudera, Amazon SimpleDB, Apache Flink, IBM Informix, Elastic, MongoDB, and Azure DocumentDB.

Document database are also referred as document database management systems, document-oriented databases, or document store database.

Here are the key characteristics of document databases:

1. Document DBMSs are NoSQL databases.
2. Document DBMSs use key/value to store and access documents data.
3. Document DBMSs have a flexible schema that can be different for each document. For example, one document can be an Author profile, while other document can be a blog.
4. Common examples of document DBMS include JSON, XML docs, Catalogs, serialized PDFs and Excel docs, Profile data, and serialized objects.

Traditional relational DBMSs are not designed to provide efficient access to large documents or unstructured data. In case of catalogs, or profiles, or document storages, we don't need structured design. For example, storing a document in a CMS does not require a structured format . Document databases are designed to store large documents in a key/value store that are easy to search and access. The entire document is read into a memory object that is easy to read and present.

9. Enterprise Database

Large organizations or enterprises use this database for managing a massive amount of data. It helps organizations to increase and improve their efficiency. Such a database allows simultaneous access to users.

An enterprise database is robust enough to successfully handle the queries of multiple users simultaneously, and is capable of handling a range of 100 to 10,000 users at a time.

Enterprise databases are widely used by enterprises in order to strategize, plan and standardize practices.

They are primarily employed to boost efficiency in the company. By reducing costs, they help in promoting the effectiveness within an organization.

An enterprise database must allow simultaneous access of a large number of users. Some of the features such databases include:

- Parallel query
- Multiprocess support
- Clustering features

An ideal enterprise database is loaded with an array of features, all of which are focused to improve productivity and efficiency of the organization.

Advantages of Enterprise Database:

Multi processes are supportable over the Enterprise database.

It allows executing parallel queries on the system.

10. Operational Database

The type of database which creates and updates the database in real-time. It is basically designed for executing and handling the daily data operations in several businesses. For example, An organization uses operational databases for managing per day transactions.

An operational database management system is software that is designed to allow users to easily define, modify, retrieve, and manage data in real-time. While conventional databases rely on batch processing, operational database systems are oriented toward real-time, transactional operations. Operational databases, which can be based on SQL or NoSQL, are the source for data warehouses and are critical to business analytics operations. Popular operational database examples include Apache Cassandra and AWS Dynamo.

Database requirements for operational data include:

- **Indexing and Cataloging**: indexing divides data into primary, secondary and clustered indexes so that it can be easily retrieved; cataloging assigns key attributes to files so that they can be recalled effectively
- **Replication**: copies and stores data at multiple locations to lower latency and provide resilience
- **File Storage and Structure**: files are sorted and stored at relevant locations to help simplify complex use cases
- **Query Processing**: in an instant, queries input by the user are translated into simple instructions, then analyzed, evaluated, and optimized; then relevant data is extracted from the database and presented to the user
- **Transactions Support**: a transactional system ensures the operational database is working consistently in a steady-state, in a concurrent state, and maintains recovery services, also ensures the system is ACID-compliant, delivering Atomicity, Consistency, Isolation, and Durability.