International Journal of Circuit, Computing and Networking

E-ISSN: 2707-5931
P-ISSN: 2707-5923
IJCCN 2022; 3(2): 06-16
Received: 06-05-2022
Accepted: 05-06-2022

Ogirima SAO
Department of Information Systems, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

Adedeji OT
Department of Information Systems, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

Amusan EA
Department of Cyber Security Science, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

Corresponding Author:
Ogirima SAO
Department of Information Systems, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

Database development, design and trending tools

Ogirima SAO, Adedeji OT and Amusan EA

DOI: https://doi.org/10.33545/27075923.2022.v3.i2a.44

Abstract
It is very important for database management system (DBMS) users to make sure there is illustration of the logical view of data using a precise data model; it includes hierarchical, network and relational models. The commercially available DBMS are based on the models but the relational product seems to have overshadowed other models. There is an effort been made to break the monopoly of the relational model which resulted in new data models like Object-oriented data model, Entity-Relationship data model, Functional data model, and Semantic data model. Also, an advance in the area of networking has given rise to database servers and its descendant, distributed databases. With this development, geographically distributed organization has its communication bills reduced significantly. For future research, database development tools may depend on the involvement of programming language researchers in databases, the impact of the object technology on the dominating relational model and whether the attempts to integrate knowledge and data bases ultimately yield commercially affordable products.

Keywords: Database, Database development, design, trends, tools

1. Introduction
Database is a detailed collection of mutually related data, which includes the computer hardware being used to store it and the software to manipulate it. A database comprises of Schema, Table, Column, Row and Primary Key [1]. Database design tool is portrayed to as an integrated software package developed to support all of the following database-related computations:

1. Creation of the data collection in storage
2. Changes to elements in the data collection
3. Retrieval of data from the collection
4. High quantities of data transformation into a usable form (Big data linked to existing data).

In order to present a comprehensive logical view of data, three major data models have been proposed namely the Network model [2, 4] the Relational model [5, 7] and the Hierarchical model [8]. The network model has not helped users to achieve data independence although it naturally separates entities and relationships to a certain extent. The relational model uses relational theory to achieve a high degree of data independence although some important information about the real world is traded off. The IT uses set theory to achieve a high degree of data independence but its view of values may be unnatural to people.
Although the relation model continues to persist as the most popular, there are a few emerging "post-relational" models. The example of the E-R model that adopts the most natural view the real world consists of entities and relationship. It is the only data model which incorporates all the advantages of the three major data models; even then it has not succeeded in achieving a utility as wide as the relational model. Other newer data models are functional model typified by the Data Language DAPLEX [9]; Data Model typified by SDM [10] and Object Oriented Data Model [11]. The trend is towards extending the functionality of relational database systems or a different orientation of data definitions and rules in which the data can be stored and retrieve when needed. Object-oriented database systems and object-relational systems were designed to address the inadequacies of relational systems. The advent of computer network, improving on the existing multi-user computer configurations, gave rise to the idea of database servers and distributed databases. The PC database server has been described as the biggest development in LAN database arena [12]. This is more exciting than the advent of distributed databases.

This makes it possible for organizations to run applications that access data in multiple databases on multiple computer systems, thus cutting their communication spending.

The rest of the study described briefly with three major data models. It is followed by various attempts at replacing the dominating relational technology. Lastly, the current concept of database servers and the emerging Distributed, Centralized, NoSQL, Cloud, Relational, Network, Object-oriented and Hierarchical database are discussed. The study is concluded with a personal opinion on the future of database design tools. To have an efficient database design and development to fit the core needs of any web based or enterprise application lies of the database (where the data, content, and configuration of an application are stored), it must be properly designed. Database is largely desired to create structures that can meet the necessary requirements of storing information accurately and permanently for a long period while enabling the database developers or users to easily search for information within the stored data for decision making when needed [13].

As the world is becoming digitalize growing fast and more complex in the volume (terabyte to petabyte), variety (structured and un-structured) and velocity in nature. The Big Data global phenomenon comes up in which data collection has grown so large that cannot be effectively managed or exploited using Relational Database Management Systems (RDBMS). There are needs or database developers alternative databases such as NoSQL, Cloud databases etc [14].

Generally, there are two components of any information system; it comprises of the database and database management system (DBMS). SQL is the query language for retrieving data from the database, as well as other data management tasks [15]. The current trends in database management systems are extremely powerful and reliable good and quickly parsing a query and executing it in the fastest method possible. Most of the databases are capable of storing objects and XML structures and querying them using a powerful extension to the SQL query language directly.

In the current trends in database, there are two major types of database designs, these includes Decision Support System (DSS) and On-line Transactional Processing (OLTP). DSS based database designs were tailored towards historical data and business intelligence. The data is infrequently updated, however the reports and queries are constantly being run against these tables hence indexing and minimization of query overhead is an important factor. In the case of OLTP databases, it is usually found in line of (LOB) applications and is constantly updating and inserting new records. The reports and queries are still being performed against the tables found, though the data is not the same magnitude as that of DSS databases and is usually much more normalized to provide the best efficiency and data integrity [16].

Parallel database systems were being used for a wider variety of systems, right from database applications to decision support systems. The implementation involves the database processing and querying over parallel systems. For effective and efficient parallel databases, various optimizing solutions need to be implemented so that the solutions can address various issues associated with such database systems (such as query optimization and data allocation within the database) [17].

Databases are usually viewed as passive repositories of data that can store, retrieve, and alter data in a database when there is need with the database not responding to actions performed to it. The passive paradigm is now changing rapidly as databases incorporate rules which permit them to detect and respond to important events. The resulting active databases have expanded capabilities by enforcing constraints, maintain derived data, propagate the effect of updates, and implement change management protocols. The incorporation and active functionality must consider the existing database functionalities (such as safe concurrent access to data, secure access to data, and efficient access to large amounts of data) and each of the functionality must be integrated with the rule model [18].

Some years back, various organizations decided to be using centralized databases systems (CDBS) for daily transactions in different domains such as banking, industries, commerce, booking and so on. Nowadays, there is still some working under this approach. Though, it can be observed that some issues related to the complexity, maintenance, performance, and cost of data communication in a centralized database system during query processing, depending on end-user demand from different sites. Therefore, there are certain times; some of them are motivated to implement efficient distributed database systems (DDBS). Distributed databases systems (DDBS) are networked computer databases, managed by different sites or locations and appearing to the user as a single database [19].

2. Fundamentals of Database Development

Database systems that are capable of dealing with large amounts of data generally are categorized under Centralized, Distributed, NoSQL, Relational, Cloud, Network, Object-oriented and Hierarchical database.

2.2.1 Centralized Databases

A centralized database is a database sited or located, stored, and maintained in a single location. Usually maintenance and modification can be done only from that location and accessed with the aid of an internet connection such as a LAN or WAN. Organizations such as colleges, universities, and companies make uses of centralized database. Information such organizations are stored in a single database.

2.2.2 Distributed Database

The set of physical components in a distributed database is partitioned among its sites. That is, each site of a distributed database system accommodates a subset of the set of physical components in the system, and each physical component belongs to exactly one site. .Distributed databases require the ability to work transparently with data, regardless of where the data resides, since the system consists of many servers and many users. Such a database has two key features [20, 21]. It is able to do a relational join with two tables from two database servers. Secondly, it is able to write updates to two or more sites as part of one transaction. A popular distributed database system in the PC arena is VIA/DRE (from VIA Information systems). VIA/DRB takes care of relational join feature through data Dictionaries. It makes uses of two-phase commit routine to handle multiple site update. Distributed database application can be developed for this database using C or the C# object-oriented Language. Front end products acceptable to VIA/DRE are NPL/R and EXCEL. The upcoming years are expected to witness a significant impact of The Distributed Database technology. With the emergence of different commercial products, it is expected that Distributed Database management systems (DDMS) will replace the current centralized ones within the next decade. Initially, the

implementation was hindered due to different challenges perused in past researches. Many of these difficulties include high system costs, unreliable technologies, difficult network topologies, and insecurity among users. With the current advancement in technologies, the system cost has gradually decreased. Therefore, users are well informed with the benefits that are involved in distributed database systems [22].

2.2.3 NoSQL Database

NoSQL (Not SQL or Not Only SQL) is a generic term used for databases that does not depend on a relational model. NoSQL is of four categories key-value, column, document and graph. Each category has data model for managing its data. NoSQL does not support SQL and there exists a huge base of users familiar with SQL. The data neither have a strict schema nor the usual SQL table structure [23]. By using NoSQL databases, a database programmer can store large volumes of unstructured data as it comes in and structure it at a later. NoSQL databases can maintain data consistency and establish relationships between data sets. NoSQL databases help organizations analyze enormous amounts of unstructured data. NoSQL is gaining ground today. It is a non-relational database being considered as the future of databases that provide important features for data storage, dynamic schema, scale-out architecture, flexible data model and access requirements [24]. NoSQL is an alternative technology that gives solution to the ever-growing data requirements [13, 25]. It is purposely meant for overcoming the performance issues of a relational database. It processes information available for multiple virtual servers around the globe. NoSQL databases are alternative databases for storage and processing the so-called Big Data today [26].

2.2.4 Cloud Database

Some year back, storing database in the cloud was not something that was thought off. Nowadays cloud database has improved drastically to an extent that there are more advantages than disadvantages Table 1. Cloud database offers many services to business with great benefits. Nevertheless, there are still some challenges in using cloud databases. Cloud computing is the buzzword in this decade, and businesses are making the switch to the cloud each day. Harnessing the immense potential that the cloud database can offer to a business is amazing and makes life simple. Cloud database is a virtual database that is hosted on a public or private cloud platform. Compare to other type of databases it doesn't require all the huge storage and server hardware as the service is hosted over a secure network. Microsoft Azure is an example of a cloud database.

2.2.5 Relational Databases

Relational database allow multiuser access to a database system with selective permit to users to share data, while retaining the ability to restrict data access. There is mode of mechanism provided to protect and secure, permitting information to be accessed only by properly authorized users. The tables or restricted views of tables are created and destroyed dynamically, the granting, authentication, and revocation of authorization to use them must also be dynamic. When a database programmer or user creates a table, it is fully and solely authorized to perform upon operations (such as read, insert, update, and delete). Other users may explicitly grant or privileges on the table of the created database and users further grant these privileges to still other users. The privileges can only be granted through the originator or table creator [27]. MySQL is an example for Relational database. A Relational Database Management System (RDBMS) is the software that helps the database developers or users, in designing, managing, and implementing a relational database [28]. It is primarily model for data storage, retrieval and management [1].

2.2.6 Network Database

The Network model of data happens to be the second major category of database design tool. The model uses mathematical set theory but requires the user to have comprehensive knowledge of data to be organized before the database is built. A network structure is much more complex than a hierarchical structure because reference pointers are inherent in most implementations. Typically, data is organized by record types which are defined and maintained through sets. These sets are used to describe one-to-many relationships among the various record types, resulting in a network of interconnected records. Examples of minicomputer DBMS having network model structure are MDBS and TOTAL. Currently, the microcomputer market also offers a few DBMS, based on the network model. An example is db-Vista III a powerful DBMS from Raima Corp. For C programmers willing to develop database applications in C. Q-Pro 4 from Q-N-E International programmer's tool.

2.2.7 Object-Oriented Database

An object-oriented database (OODB) is a database system that can work with complex data objects. OODB is in the form of objects, as used in object-focused programs. OODB uses concepts focused on object such as object categories, polymorphism, encapsulation, and inheritance [28]. It is meant for database programmers willing to develop database using object-oriented programming languages such as C++, C#, Python and Java. Examples of OODs are ObjectStore (www.ignitetech.com) and Versant Object Database (www.versant.com). Versant provides transparent object persistence from C++, Java, and .NET. Versant supports different APIs depending on the OOP language used. Standard SQL queries are also available, making Versant a NoSQL database. Relational databases have also added object-oriented features. Example of OOD is UniSQL.

2.2.8 Hierarchical Database

The hierarchical database is a database that data are organized into a tree-like structure. These organize data into highly structured "master-slave" or "grandfather-son" arrangement. A hierarchical-based DBMS stores data as fields and records within segments, at least one of which must be a parent. Other segments descend from the parent and are related to it as children. You can locate the segments within the same logical file or outside of it. Users must generally know all about the hierarchical structure, because looking up data or "querying the database" that requires a command statements that will trace the tree to the appropriate level for the data sought.

In the mainframe world, Information Builders Inc. launched FOCUS, a hierarchical DBMS/fourth generation language program. PC/FOCUS is the microcomputer version of FOCUS. Other products in the Hierarchical DBMS market place are Unlimited Processing Inc's Team-Up and PC Manager's Data Edge. Older DBMS known to be popular are system 2000 and AD ABAS. Other databases include Parallel and Hybrid database.

Parallel Database

A parallel is a DBMS implemented on a multiprocessor computer. It implements the concept of horizontal

partitioning by distributing parts of a large relational table across multiple locations or sites that are to be processed in parallel. It requires a partitioned execution of the SQL operators. Some basic operations, like a simple SELECT, can be executed in-dependently on all the locations or sites. Complex operations are executed through a multiple-operator pipeline. Different multiprocessor parallel system architectures, like share-memory, share-disks or share nothing, define possible strategies to implement a parallel DBMS, each with its own advantages and drawbacks. The share-nothing approach distributes data across independent nodes and has been implemented by many commercial systems as it provides extensibility and avail-ability. Based on the above definitions, it can be conclude that parallel database systems improve performance of data processing by parallelizing loading, indexing and querying data.

Hybrid Database

*H*ybrid database is an embedded database management system that offers high performance data processing in main memory with the vast storage capabilities of physical disk. A hybrid database poses both in-memory database and on-disk database features in a single unified engine. As a result, data can be stored and manipulated in main memory alone, on disk alone or a combination of both. The *integrated* combination of both database types allows for unrivaled flexibility and robust. Hybrid Database supports both the types of databases. For programmers willing to develop hybrid database, maintenance balance can be achieved between performance, cost, and persistence**.** An example of a Hybrid Database is Altibase**.**

2.2.9 Database Creation and Viewing

Database programmer can be create a database with the help of SQL programming languages (such as C++, C#, Java, Python, PHP and son). In any SQL programming language a metadata of the database needs to be created as show in

Appendix A

The database can be created in this manner: SQL Create < database name>, then the CREATE DATABASE statement as shown below:

```
SQL> CREATE TABLE Herbal Plant TB (Herb ID Int.,
 Description VARCHAR (50) NOT NULL,
 Common Name Char (50), NOT NULL,
 Local Name Char (50),
 Scientific Name Char (50),
 Image Char (50));
```

In SQL, the user can view a table based on the result-set of an SQL statement. A view contains rows and columns.
A view is created with the CREATE VIEW statement.

```
CREATE VIEW [Description] AS
SELECT CommonName
 FROM HerbalPlantTB
 WHERE CommonName = 'Sensitive plant';
```

There are many other operations that can be perform when using SQL language such as UPDATE, DROP, EPLACE, INDEX, INSERT, JOIN, TUNCATE, ALTER, and so on. You can see more Source Programming Code in Appendix B (C# programming Codes)

Table 1: Showing Ailment medication and prescription

SN	Ailment	Treatment	Prescription
1	Abdominal pain	Juice of aloe vera mix with honey.	Take 3times for 7days
2	Acne	Boil catnip,lemon grass,garlic,neem,vervain,tumeric , orange.	Take 1 table cup 2times daily
3	Acute dysentery	Chew leaves of Siam weed with urine	every morning for 3days
4	Allergies	Boil chickweed,pawpaw leaves,peppermint,nettle,parsley with orange.	Take 1cup 3times daily
5	Arthritis	Boil mistletoe,garlic,thorney pigweed,avocardo leaves,chickweed with honey.	Take 2times daily
6	Asthma	Mix powdered thorny pigweed,garlic,tumeric & ginger with honey.	Take 3times daily
7	Bad Breath (Halitosis)	Mix powdered garlic,tumeric, ginger & thyme with honey.	Take 3times daily
8	Baldness	Mix Aloe vera juice,onion,horsetail,nettle with honey.	Rubb 3time daily
9	Bedwetting	Mix powdered plantain,nettle,mistletoe,pepper,corn silk with honey.	Take 3times daily
10	Bladder Cancer	Mix powdered garlic,tumeric,corn silk,nettle & thyme with honey.	Take 3times daily
11	Breast Cancer	same treatment for bladder cancer but rubb some to the surface part of breast	3 times daily
12	Bronchitis(Cough)	Mix powdered garlic,ginger,thorney pigweed,thyme,bitter kola with honey.	Take 3spoon 3times daily
13	Burns	Apply honey on affected part then chew small quantity of garlic.	3times daily
14	Canker Sores (Cold Sores)	Mix powdered garlic,ginger,tumeric,peppermint,and vervain with honey,aloe vera juice and orange.	Take 3times daily
15	Cataract	Apply Aloe vera juice directly until condition improve	Twice daily
16	Cervical Cancer	Same treatment to breast cancer but Aloe vera juice drpped at virginal	At night
17	Cholera	Squeeze scent leaves with little quantity of salt.	Take 3times daily
18	Chronic Obstructive Pulmonary Disease (COPD)	Mix aloe vera juice with orange.	Take 3times daily
19	Colon Cancer	Boilgarlic,tumeric,aloe vera leaves,nettle ,pawpaw leaves & thyme mix with honey.	Take 3times daily
20	Congestive Heart Failure (CHF)	Mix powdered garlic,ginger,thorney pigweed,thyme,bitter kola with honey.	Take 3spoon 3times daily
21	Convulsion	Mix powdered garlic,ginger,tumeric,pepper,nettle,mistletoe and vervain with honey or urine.	Take 2times daily
22	Conjunctivitis(Apollo)	Wash eyes with self urine then applied aloe vera	morning and night
23	Dandruff	Wash hair with urine then apply aloe vera gel to the hair	Daily after bath
24	Dehydration	Mix aloe vera juice with orange & neem leaves.	Take with salt 3times daily
25	Depression	Boil neem leaves,lime,lemon grass,mistletoe and garlic.	Take 3times daily
26	Diabetes	Boil onion,garlic,ginger,plantain,mistletoe,tumeric,unripe pinapple,orange with bitter leaves.	Take at night
27	Diaper Rash	Mix aloe vera gel,honey and olive oil.	Rub to affected 3times daily
28	Diarrhea	Squeeze scent leaves with little quantity of salt or aloe vera juice with orange.	Take 3times daily
29	Drug Abuse	Boil onion,garlic,ginger,mistletoe,tumeric,cigrette(tobaco) leaves with honey.	Take at night
30	Ear Problems	Mix quantity of aloe vera juice & honey.	Drop into ear 3times daily

Table 2: Showing Symptoms with its corresponding ailment

Symptom List

SN	Symptom	Ailment
1	severe pain in stomach, vomitting,unable to stand well	Abdominal pain
2	Swolen gland on skin,red pimples on face & neck	Acne
3	Blood stain on stool,Pains in anus,hard stool	Acute dysentery
4	Body itching,swollen gland,red pimples on the body	Allergies
5	Pains on joints,neck,back,knees,wrist,finger etc	Arthritis
6	Persistent coughing,unable to breath well,sweating	Asthma
7	Bad odour from mouth when talking or breathing	Bad Breath (Halitosis)
8	Hair cutting, dry scalp	Baldness
9	Passing urine on bed during sleeping at night	Bedwetting
10	Severe pain in affected part, weight loss,tiredness,painful urination & bad odour	Bladder Cancer
11	severe pain in breast, weight loss,tiredness,rise in temperature	Breast Cancer
12	Coughing, chest pain,hard breath, mucus with spit	Bronchitis(Cough)
13	Swollen glands as result of fire or hot water	Burns
14	Difficult in breathing,coughing,headache,sneezing,watery eyes,fever,pains and aches	Canker Sores (Cold Sores)
15	Growth covering the eye,redish eye	Cataract
16	Bleeding from virginal,pains,pieces of wart droping on pants	Cervical Cancer
17	Vomitting, stooling, weight loss, unable to eat	Cholera
18	Difficult stools,slight stomach pain	Chronic Obstructive Pulmonary Disease (COPD)
19	Severe pain in colon, internal bleeding,stools stain wih blood,pale,rise in temperature	Colon Cancer
20	Unable to breath,headache,slight chest pain,pale,weight loss	Congestive Heart Failure (CHF)
21	Fallen down,Dizzness,Abnormal cerebral stimulation,uncontrollable contraction of muscles	Convulsion
22	Redish eyes,unable to open eyes,pains as if stones on eyes	Conjunctivitis(Apollo)

Table 3: Showing Herbal information list available in the database

Herbal Information List

SN	Scientific Name	Common Name
1	Acacia catechu	cutch tree, black catechu
2	Acacia nilotica	Indian gum arabic tree, babul
3	Achillea millefolium	yarrow, milfoil
4	Achyranthes spp.	chaff flower, devil's horsewhip
5	Aconitum spp.	aconite, monkshood, wolfsbane
6	Acorus calamus	calamus, sweet flag
7	Actaea racemosa	black cohosh
8	Aesculus hippocastanum	horse chestnut
9	Alangium salvifolium	alangium
10	Allium cepa	onion
11	Allium sativum	garlic
12	Aloe vera	aloe vera
13	Andrographis paniculata	andrographis, king of bitters, chiretta
14	Angelica archangelica	angelica
15	Angelica sinensis	dong quai, dang gui
16	Angelica spp.	angelica (various)
17	Apium graveolens	celery
18	Aralia spp.	spikenard (various)
19	Arctium lappa	burdock, gobo
20	Arctostaphylos	uva-ursi, bearberry
21	Areca catechu	betelnut palm, areca nut
22	Aristolochia spp.	birthwort (various)
23	Arnica montana	arnica
24	Artemisia spp.	southernwood, wormwood, absinthe, mugwort, tarragon
25	Aspalathus linearis	rooibos
26	Asparagus spp.	asparagus

Trends in Data Modeling

The relational model has retained its popularity since the early 1970s and has almost replaced the older hierarchical and network data models since that time. The newer attempts are more or less 'post relational" models having not been able to successfully replace the relational model. A few alternative successors to the relational model are Entity-relationship model, Functional model, the Semantic data model and the Object-Oriented data model. Historically, they were proposed in that order. Research and development in the area of database technology during the past decade is characterized by the striving for better support for applications beyond the traditional world, where most high volumes of simply structured data had to be processed efficiently. With this result, future DBMS need to include more functionality, and explicitly cover more real world semantics (in various forms) that otherwise would have to be included in applications themselves.

Advanced database technology, however, is in a sense ambivalent. While it provides new and much-needed solutions in many important areas, these same solutions often require thorough consideration in order to avoid the introduction of new problems. One such area is database security. This study only considers three prominent areas of nonstandard database technology: object-oriented, active and federated database management systems. In particular, we show which typical security problems (with the focus on access control) have to be solved for these systems. The kinds of DBMS and its underlying mechanisms, respectively, can beneficially be used to solve security problems. The issue of secure design and construction of database management systems are detail in this context [29]. In year 2022, database management system trends includes Cloud-based DBMS, Automation DBMS, Augmented DBMS, Increased security, In-memory databases, Graph databases, Open source DBMSs, and Databases-as-a-service.

3.1 Relational Model

Relational database model works with the concept of mathematical set theory. Set theory make uses of two dimensional collection of information called relation. RDBMS provide a simple way to construct, access, and update database. The model supports only text and numeric value but it does not support abstract data type (such as audio, video and geographical information). Under the relational model, a database is viewed simply as a collection of tables, each composed of one or more columns and zero or more rows. The way or manner by which tables are stored and retrieved is hidden from the users. Though links or pointers may exist at the physical level, the relational model makes no restrictions on how the physical storage system operates. Relational operators are designed to generate new tables from older ones, therefore when subsets of rows or columns from a table are extracted, they are also considered to be tables. A high degree of data independence is guaranteed as changes are made to the storage structure and access strategy fail to affect the database.

In a true relational system, applications are not even aware of what access strategy is being used. Although the strict adherence to the original concept or relationality is in doubt, supposedly relational products now abound on the PC. The popular ones are Dbase IV from Ashton Tate, Paradox from Borland, Dataflex from Data Access Corp, RBase from Microrim and Advanced Revelation from Revelation Technologies to name a few. Both mini and mainframe versions of ORACLE, INGRESS, System R, SQL/DS are more faithful to the relational model than their microcomputer counterparts. Relational DBMSs are widely used compare to types of database systems.

3.2 Entity-relationship Model

This happens to be the earliest alternate to the relational data model but which has not been able to gain enough attention of commercial database product developers as one would have expected. E-R model seeks to extend the "semantically impoverished" relational model with die! Concepts of: Entities, Attributes and Relationships. The model is presumed to be more natural view based on set and relation theory. This can achieve a high degree of data independence. The model is so good that it provides a framework from which the three major models can be derived [30]. Four levels of logical view of data can be identified in connection with data model studies. These are classified as: level 1: information concerning entities and relationships; level 2: information structure; level 3: access-path-independent data structure; level 4: access-path-dependent data structure. Hierarchical and network data models are mainly concerned with levels 3 and 2. The E-R-model is concerned with levels 1 and 2. In E-R model, there are four steps involved in designing a database using, namely:

1. Easy identification of entity sets and its relationship of interest
2. Identification of semantic information in the relationship sets.
3. Definition of value sets and attributes; and
4. Organize the data into entity, relations and determination of primary keys.

Rather than using the E-R model as the underlying data-model for commercial products, the tendency has been to use it as a powerful tool for performing logical database design for relational systems. An excellent discussion on why the E-R model seemed to have been so relegated to the background can be found in [31]. For example, E-R Diagram of Herbal Prescription in Telediagnosis Environment as shown in Figure 6.

3.3 Functional Model

The model presents a database as a collection of functions. For example a typical university database is made up of four different entities: STUDENT, COURSE, INSTRUCTOR and DEPARTMENT. Depending on the entity being considered, functions can be defined and applied to the entities to yield another entity type. For example, DECLARE Name of Student as STRING states that "Name" which is a FUCNTION that maps entities of type "Student" to entities of type STRING. Suppose each entity has a unique identifier or key. All attributes of that entity can be thought of as functions of the key of that entity. Any 1:1 and l: n relationships can also be considered as a function from one entity identifier to another entity identifier. A proposal along this line is DAPLEX [9]. Though the model has not been accepted for commercial purposes but for some research prototype, notably IRIS [32] incorporates the ideas.

3.4 Semantic Data Model

An attempt has been made to compensate for the "semantic poverty" of the relational model by including other constructs that add more semantic content. Examples of such constructs are Classes, Roles, Objects with no fixed type or composition, set valued-attributes and un normalized

relations. A complete list exists in [31]. A typical example of the semantic data model also included in [31] is Semantic data model (SDM). SDM views database as a collection of entities that correspond to the actual .objects in the application environment. These entities are organized into classes that are logically related by means of interclass connections. SDM entities and classes has attributes that describe its characteristics and other related entities. SDM has added too much complexity to the data model that it may be difficult to use it in a production database system. Worse still it has no data manipulation language defined for implementers.

3.5 Object-Oriented Data Model

This applies the object technology to database systems incorporation of inheritance, union types, and unique identifiers that is needed in a database system. Typing in traditional DBMS is quite limited and does not cover nonbusiness data-processing objects such as complex numbers, texts, spatial objects, vectors etc. This is the novel idea of object-oriented databases. The data abstraction, object identity, and inheritance. Most data manipulation and data modeling in object-oriented database systems rests on these fundamental notions. Object-oriented databases combine database capabilities with object orientation. Object-oriented data model defines OODB in a precise manner. Figure 12 illustrates the type of integration present in OODBs. Object-oriented databases combine the features of object orientation such as abstract data typing and distinctive object identities with database capabilities, such as persistence, transaction processing, concurrency control, recovery, security, and performance.

Research ideas in this area are either trying to introduce persistent objects into object-oriented languages or are extending an existing data model using the object-oriented concepts highlighted above.

4. Impact of Networking on Databases

Computer networking is the means by which one computer can "talk" to another even when each one may be running on different platforms. If computers 'are located on the same sites then a Local Area Networking solution is the best for its connectivity otherwise geographically remote computers can only communicate over a Wide Area Network. What is ' the impact of this arrangement on database design tools? A user will have to choose between two alternatives: the traditional file-server-based database system or a database server arrangement. In fact because companies are fundamentally distributed, the distributed database solution will be preferred to save on communication costs. Distributed database system is a descendant of the database server. The database designs are of two types. They are decision support system (DSS) and On-line Transactional Processing (OLTP). DSS designed was based to gear towards historical data and business intelligence. The data are updated infrequently, but reports and queries are constantly being run against these tables so as to index and minimize the query overhead. The OLTP was found in line of business (LOB) applications and it is constantly updated and insert new records. Reports and queries may be performed against the tables found. The data is not the same magnitude compares to that of DSS databases and it is more normalized to provide the best efficiency and data integrity.

4.1 File-Server-Based Systems

Traditional file server database has only one program. This program is stored on the file server, it runs in the workstation and does all the work of both the database server and the front-end software. When it searches for a record, for example, it must go to the file' server to get the database, look through the database, find the requested records and then display them on the screen. The whole database travels on the network from the file server to the workstation so the program may perform the search.

4.2 Database Server

A database server is a piece of software which is responsible for updating, deleting, adding, changing and protecting data which is usually stored on a hard disk to which many users on a network have access. The structured form of stored data depends on the data model underlying the database server. In order for the database server to regulate multiple simultaneous accesses, it-exercises rules called concurrency controls. It also enforces security control rules to regulate access including password and data encryption. The database server takes advantage of the LAN's distributed architecture rather than coexisting with the LAN operating system as it is the case with traditional databases. The client-server model on which a database server is based divides the database into a front-end and a back-end. The back-end (or database server) resides on a powerful machine and is responsible for number crunching, data integrity and security cases. The front-end resides on the user's PC and is largely responsible for displaying data and sending the user's queries to the back-end. Examples of database server products in this category are SQL Server (from Ashton-Tate/Microsoft/Sybase); Oracle Server (from Oracle; Belmout, California), Netware- SQL (from Novell Development Product Division), SQLBase (from Gupta technologies) and VIA DRE (from VIA Information Systems) to name the major ones. Front- ends often are, but do not have to be a database.

Front end products can be classified into four types namely DBMSs based on fourth generation languages, spreadsheets, high level languages and tools. Most Database vendors have launched a DBMS supporting one or more SQL database server. A version of FOCUS (from Information Builders) was the first product that will work with SQL server. On the spreadsheet side, Microsoft's Excel and Lotus 1-2-3 are ready to work with database servers. The majority of the vendors are including this capability in their spreadsheet. The common and popular language for developing database server applications then is C. SQL server have included other languages such Java, Python and C# apart from C. SQL Base allows either COBOL or C. Oracle users have more options. The most used Programming languages among database developers worldwide, as of 2021 can found online [33]. Examples of tools which can be used with database server are SQLFILE from Vansant and SQL Windows from Gupta Technologies. The former allows users to manipulate data stored in an SQL server without knowing SQL or programming application. The last is true database-server application development tool. It helps programmers generate Windows-based applications that can access SQLBase, DB2, DB4, Oracle or others.

Generally, there are three pieces in the arrangement the database_server or back end manipulates, manages and protects data. The front-end software enters and displays

data according to users' desires. The SQL or other query languages are the mechanism of communication between the others.

5. Database Design Tools

Database design is a prerequisite for an efficient, high-performance database, but producing a design is a complex, time-consuming job and Microsoft SQL Server&#39s built-in SQL Diagrammer n&#39 makes it easier. Good database design tool will better than paying for itself by increasing the productivity and efficiency of the resulting database. To design a flexible, fast, and efficient database is not that easy. It involves learning about and analyzing intending organization's information needs, conceptualize what the database will look like and what data to be store to meet those needs, transforming that conceptual representation into a logical design that includes rules and information about the structure and type of data, and adapting the logical design to a database management system (DBMS). SQL Server normally comes along with SQL Diagrammer. It is a basic tool that database programmers use to design databases. Though, SQL Diagrammer is a bare-bones tool has its own significant limitations. There are other database management tools that provide sufficient access; frameworks queries and analyze data sources, appropriate systems for dataset size; and database collaboration networks. These tools include:

- **MySQL:** It used companies like Google, LinkedIn, YouTube, PayPal and Twitter, etc. It is an open-source relational database for web applications like WordPress. .
- **SQL Server Management Studio:** graphical tools that can be used for variety of script editors to provide access and management for SQL Server.
- **Oracle RDBMS:** Is an object-relational database management software tools that incorporates cloud computing. It supports multiple Windows, Linux and UNIX versions.
- **Salesforce:** Database programmers make use of this tool to built a business implementing database solution on this platform.
- **DevOps:** Tool that automate processes to increase the software delivery speed and stability. This enhances productivity and reduces employee burnout.
- **Visual Studio Code:** SQL Server Management Studio linked Server connections to many other databases. For example, Visual Studio Code has connections to all database types.
- **ESM Tools:** Manage data into account of the database, workflow, reporting and process automation around that data.
- **PhpMyAdmin:** Allows non-tech operators to do its job well without constantly pinging database administrators (DBAs). It can work well for small organizations.

6. Conclusion

This study stated all the models and the types of database in existence with advantages and disadvantages to an intending organization depend on its structure. Database development tools are based on data models and the market place for the tools is almost stabilized on the relational model of data despite its well-recognized semantic poverty. A number of candidate successors to the relational model have not made it because of the ongoing standardization of standard query language, SQL (a query language based on the relational model) as the major communication link between the back-end and front end in database server architecture. Other query languages exist but SQL is dominating because it originated from IBM. Nearly all other database developers want to provide database products that can support the language and other SQL-based IBM products. In the current Big Data applications, it is good to transform SQL database schema into NoSQL so that there can be easy data migration between NoSQL and other databases.

Though, the monopoly of relational DBMSs may be short-lived unless the object technology is introduced to enhance them. It is clear from the newly proposed object-oriented data model that there may be a right-about turn towards the abandoned CODASYL systems. This is not to say that some database programmers are not willing to consider it worthwhile to add object capability to any of the existing data models or in fact build persistent versions of existing object-oriented languages like Smalltalk.

Some researchers agreed that the data and knowledge management should be strongly integrated in future systems. Currently, a knowledge base is either managed separately by a knowledge manager separate from the DBMS or it is integrated into the DBMS. The latter approach (meant for non-partitionable problems) requires the tight coupling technique of knowledge management.

Finally, it is expected that the programming language researchers should become involved with the trends in database in future. When this happens future programming environment will tend to be database oriented. The implication is that computer programs, functions, procedures, specification etc. will be database objects and editors, compiler, linker and so on will be database application programs. What this means is that we will have a completely better user interface to work with whether you are a professional or a novice.

7. References

1. Belkadi F, Esbai R. A Model-Driven Engineering: From Relational Database to Document-oriented Database in Big Data Context. In Proceedings of the 16th International Conference on Software Technologies (ICSOFT 2021), 2021, 653-659.
2. Bachaman CW. Software for Random access Processing Datamation, 1965, 36-41.
3. Bachaman CW. Data structure diagrams Database. 1967;1(2):4-10.
4. Codasyl Database task group report ACM, New York, c1971.
5. Codd EF. A relational model of datafor large shared databanks comm.ACM. 1970;13(6):377-387.
6. Adigun MO. Trends in Database Design Tools. Computer Association of Nigeria Proceedings Conference of the 4th International Conference, c1992.
7. StudyCorgi. Relational Database Management Systems in Business, 2021. Retrieved from https://studycorgi.com/relational-database-management-systems-in-business/
8. Senko M. Information Systems: Records, Relations, Sets, Entities and Things. Information Systems Pergamon press. 1975;1(1):3-13.

9. Shipman DW. The Functional Data Language DAPLEX, ACM Transaction on Database Systems. 1981;6(1):140-173.
10. Hammer H, Mcleod D. Database Description with SMD. A semantic Database Model, ACM Transactions on Database Systems. 1981;6(3):351-386.
11. Copeland GP, David M. "Making smalltalk a database system." SIGMOD '84 (1984). 1984.
12. Brunner KP, Korfhage RR. An Automatic Improvement Processor for Information Retrieval System. Proceeding from the Second International Conference on Expert Database System, George Mason University; c1988.
13. Basit A, Bushra S, AL-Khafaji N. NoSQL: Will it be an alternative to a relational database? MySQL vs MongoDB comparison. IMDC-IST 2021, September 07-09, Sakarya, Turkey; c2021. DOI 10.4108/eai.7-9-2021.2314925
14. Salah AJ, Ramadan FS, Antisar YA, Mahmood SS. Traditional RDBMS to NoSQL Database: New Era of Databases for Big Data Journal of Humanities and Applied Science (JHAS) Issue No. 2016, 29.
15. Taipalus T, Grahn H, Ghanbari H. Error messages in relational database management systems: A comparison of effectiveness, usefulness, and user confidence. The Journal of Systems & Software. 2021;181:111034.
16. Ogirima S.AO, Afolabi AO, Adigun AA. An integrated framework for telediagnosis and prescriptions in herbal medicine. International Journal of Advance Networking Application. 2017;8(4):3124–43.
17. Talwadker AS. Survey of performance issues in parallel database systems. Journal of Computing Sciences in Colleges. 2003;18:5-9.
18. Smith KP. Managing rules in active databases; c1992.
19. Ezéchiel K, Kant S, Agarwal R. A Systematic Review on Distributed Databases Systems and their Techniques; 2019.
20. Garcia-Molina H, Lindsay BG. Research directions for distributed databases. SIGMOD Rec. 1990;19:98-103.
21. Rajkumar MN, Radhika R, Karthika J. Efficient separation and allocation of dataset in structured and unstructured databases. Materials Today: Proceedings, 2021;37:2547-2552.
22. Jain G. Distributed Data Management: Challenges and Solution on Distributed Storage; c2016.
23. Mohamed W, Abdel-fattah M, Abdelgaber S. A Comprehensive Approach for Converting Relational to Graph Database Using Spark. Journal of Theoretical and Applied Information Technology. 2021;99:12.
24. Musa G, Hassan A. A Comparison of NoSQL and Relational Database Management Systems (RDBMS). KASU Journal of Mathematical Sciences (KJMS). 2020, 1(2). http:/www.journal.kasu.edu.ng/index.php/kjms
25. MAH. Relational and NoSQL Databases: The Appropriate Database Model Choice, 2021 22nd International Arab Conference on Information Technology (ACIT), 2021, p. 1-6.
26. Pokorný J. Integration of Relational and NoSQL Databases .Vietnam Journal of Computer Science. 2019;6(4):389-405.
27. Griffiths PP, Wade BW. An authorization mechanism for a relational database system. ACM Trans. Database Syst. 1976;1:242-255.
28. Verma O, Chaudhary I, Khan M, Chaudhary A Isha. A Comparative Study of Relational Database Management System and Object Oriented Database Management System; c2021.
29. Dittrich KR, Jonscher D. Current Trends in Database Technology and Their Impact on Security Concepts. DBSec; c1994.
30. Chen PP. The Entity-Relationship Model Towards a Unified View of data in Reading database systems, M. Stonebraker Ed., Morgan Kaunfmann Pub. Inc. 1988, 374-387.
31. Stonebraker M. Readings in Database Systems Morgan Kaufmann Publishers, 1994 Science 970.
32. Fishman. An Object Oriented Database Management System," ACM TOIS. 1987, 5(1).
33. https://www.statista.com/statistics/869092/worldwide-software-developer-survey-languages-used/

Appendix A: metadata of Herbal Telediagnosis database

Metadata element	Data type	Literal meaning
Herbal Plant TB	Data type	Literal Meaning
Herb ID	Int.	Integer
Description	Nvarchar (50)	String
Common Name	Nvarchar (50)	String
Local Name	Nvarchar (50)	String
Scientific Name	Nvarchar (50)	String
Image	Nvarchar (50)	String
Part Use TB	Data type	Literal Meaning
Plant Part ID	Int.	Integer
Herb ID	Int.	Integer
Part ID	Int.	Integer
Prescription TB	Data type	Literal Meaning
Ailment	Varchar (50)	String
Request ID	Varchar (20)	String
Treatment	Varchar (130)	String
Prescription	Varchar (50)	String
Dosage TB	Data type	Literal Meaning
Dosage ID	Int.	Integer
Description	Nvarchar (50)	String
Treatment ID	Int.	Integer
Admin ID	Unique identifier	Globally Unique

Ailment TB	Data type	Literal Meaning
Ailment ID	Int.	Integer
Description	nvarchar (50)	String
Ailment Symptom TB	Datatype	Literal Meaning
Symp ID	Int.	Integer
Symptom ID	Int.	Integer
Ailment	Int.	Integer
Symptom TB	Data type	Literal Meaning
Symptom ID	Int.	Integer
Description	Nvarchar (50)	String
Telediagnosis Conference TB	Data type	Literal Meaning
Id	Int.	Integer
Centre Id	Int.	Integer
Consultant ID	Int.	Integer
Patient ID	Int.	Integer
Interpreter ID	Int.	Integer
Conversation Date	Date time	Date Time
Herbal Consultant TB	Data type	Literal Meaning
Id	Int.	Integer
Surname	Nvarchar (MAX)	String
Firstname	Nvarchar (MAX)	String
Gender	Nvarchar (MAX)	String
Phone Number	Nvarchar (50)	String
Email Address	Nvarchar (50)	String
Centre Id	Int.	Integer
Interpreter TB	Data type	Literal Meaning
Id	Int.	Integer
Surname	Nvarchar (MAX)	String
First name	Nvarchar (MAX)	String
Gender	Nvarchar (MAX)	String
Phone Number	Nvarchar (50)	String
Email Address	Nvarchar (50)	String
Centre Id	Int.	Integer
Patient Rec TB	Data type	Literal Meaning
Id	Int.	Integer
Surname	Nvarchar (200)	String
Othernames	Nvarchar (200)	String
Phone No	Nvarchar (200)	String

Appendix B: Souce Program

```
using System;
using System. Collections. Generic;
using System. Component Model;
using System. Data;
using System. Drawing;
using System. Linq;
using System. Text;
using System. Windows. Forms;
using System. Data. SqlClient;
namespace Herbal
{public partial class Form1 : Form
{public Form1()
{InitializeComponent();}
{private void button1_Click(object sender, EventArgs e)
try{string query = "SELECT Name FROM Table1 WHERE (Name='" + textBox1.Text + "' AND Password='" + textBox2.Text + "')";
Sql Command cmd = new SqlCommand(query, conn);
Sql Data Reader dr = cmd.ExecuteReader();
string see = "";
while (dr.Read())
{see = dr[0].ToString();}
dr.Close(); cmd.Dispose(); dr.Dispose();
if (see == "") MessageBox.Show("Invalid Login", "Herbal Management System");
else
{groupBox2.Visible= true;
menuStrip1.Visible = true;}}
private void button6_Click(object sender, EventArgs e)
{try{string query="INSERT INTO HerbInfo (Scientific Name, Common Name) VALUES('" + textBox10.Text + "','" + textBox7.Text + "')";
Sql Command cmd = new SqlCommand(query, conn);
cmd. Execute NonQuery();
textBox7.Clear(); textBox10.Clear();}
catch { MessageBox.Show("Error! Check your input."); } }
private void deleteToolStripMenuItem_Click(object sender, EventArgs e)
{if (dataGridView1.SelectedRows.Count > 0)
{dataGridView1.Rows.RemoveAt(dataGridView1.Selected Rows[0].Index);
string q = "delete from table2";
Sql Command cmd = new SqlCommand(q, Form1.conn);
cmd. Execute Non Query();
for (int k = 0; k < dataGridView1.Rows.Count; k++)
{dataGridView1.Rows[k].Cells[0].Value = (k + 1).ToString();
```

```
q= "INSERT INTO Table2 (Disesase, Request ID, Treatment, Prescription) VALUES('" + dataGridView1.Rows[k].Cells[1].Value + "','" + dataGridView1.Rows[k].Cells[2].Value + "','" + dataGridView1.Rows[k].Cells[3].Value + "','" + dataGridView1.Rows[k].Cells[4].Value + "')"; cmd. Command Text = q;
cmd. Execute Non Query();}}}
namespace herbal
{public partial class WebForm1 : System.Web.UI.Page
{SqlConnection conn; protected void Page_Load(object sender, EventArgs e)
{conn = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\OGI RIMA\Desktop\latest code\mr ogirima\Herbal.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"); conn.Open();}
 Protected void Drop Down List1_SelectedIndexChanged(object sender, EventArgs e)
{string q = "select treatment,prescription from table2 where ailment='" + Drop Down List1. Items[DropDownList1.SelectedIndex].ToString().Trim() + "'"; Sql Command cmd = new SqlCommand(q, conn);
SqlDataReader dr = cmd.ExecuteReader();
while (dr.Read())
{Label2.Text = "Treatment: " + dr[0].ToString().Trim() +
 "Prescription : " + dr[1].ToString().Trim();} dr.Close(); cmd.Dispose(); dr.Dispose()}}}
```