Chapter 1 - Introduction to Database - Part 1

1 Introduction to Database - Part 1
- 1.1 Types of Databases
- 1.2 Terminology

References

1 Introduction to Database - Part 1

Databases are a persistent collection of data that is organized in a way that facilitates the ease of data management. If a database is a collection, then what is data? Data is information collected from qualitative or quantitative means from one or more people, object or events and it can be textual, numerical, in the form of graphics, reports, etc. Data can be either messy or tidy (structured in some way), it is always easily available, and plentiful. Therefore we need a container to store these data and those containers are called databases.

Before computers were invented, a physical means to store large amounts of information would be a library. But in todays terms database are not only used to store information, they are also used to organize, protect and deliver data. In order to store and manage these databases, a system call the **database management system** (DBMS) was created.

In this module, we will be learning about the processes (figure 1 below) of how data is treated then stored in a database and the different ways upon which we can access and manipulate this stored data.

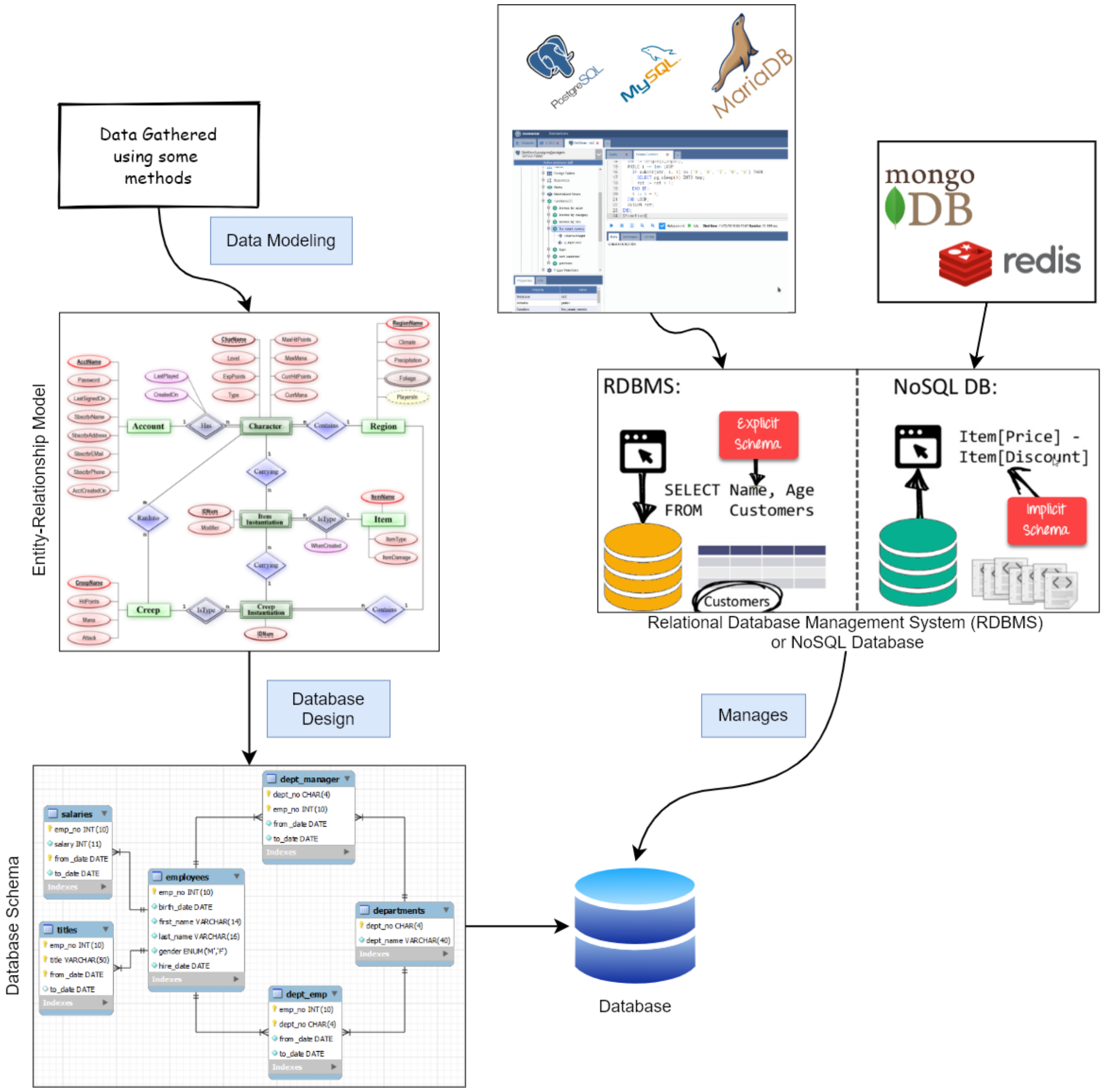

Figure 1: Databases module roadmap

1.1 Types of Databases

There are several types of databases used for various applications

- **Centralized Databases** are used to store data in a centralized location where users from different locations can access the data (figure 2 below). Users can use various applications with different kinds of authentication procedures to access the data securely. An example of a centralized database is a public library.

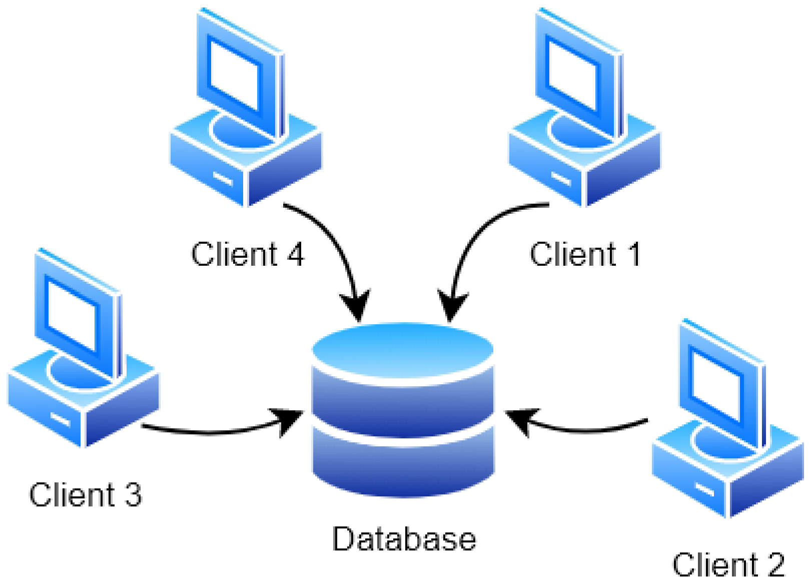

Figure 2: Pictorial representation of a centralized database.

- **Distributed Databases** are the opposite of centralized databases. Databases in distributed databases are stored across various locations and sites of an organization. These databases are connected to each other through the use of Local Area Networks and/or Wide Area Networks. Refer to figure 3 below.

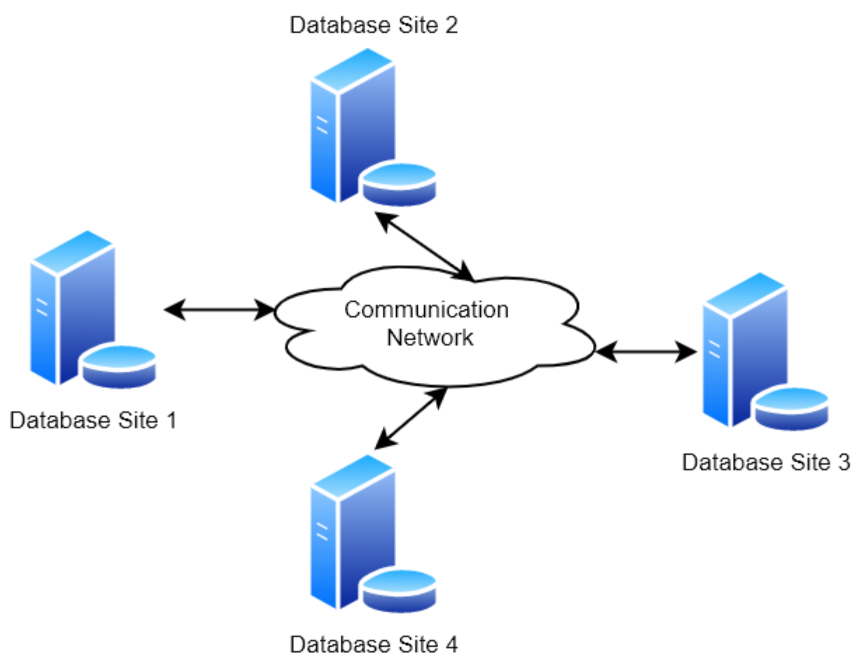

Figure 3: Pictorial representation of a distributed database.

Distributed databases are divided into 2 classifications: **homogenous** and **heterogeneous**. These classifications refer to the type of hardware, operating systems and application procedures that the databases operate on. Homogenous databases have **all the same** type of hardware, operating systems and application procedures, and heterogeneous uses **different** types of hardware, operating systems and application procedures.

- **Relational Databases** are databases that uses the relational data model to store data. This model categorizes data into a set of tables. Those tables consists of rows and columns where each column defines the specific category of data and each row contains a record of the data according to the categories stated by the columns (figure 4 below). Structured Query Language (SQL) statements are used for accessing, manipulating and maintaining the data in relational database. Examples of relational databases are MySQL, Oracle, Postgres, MariaDB, etc.

Branch Table

branch_name	assets	branch_city
Howe	6 000 000	Vancouver
Downtown Kleve	12 000 000	Kleve
Signal Hill	8 000 000	Calgary
Clear Water Bay	9 000 000	Hong Kong

Loan Table

branch_name	loan_number	amount
Howe	1	150 000
Downtown Kleve	2	50 000
Clear Water Bay	3	80 000
Clear Water Bay	4	10 000

Figure 4: Pictorial representation of some tables in a relational database.

- **NoSQL Databases** are databases that stores a wide range of data sets. The format of the stored data is not only in tabular form but also in several other different ways. The ways can be broken down as follows
    - **Key-value storage** - the data is stored in key-value pairs where the keys denote the attribute of the value it is holding (figure 5 below). Examples of some key-value store databases are Clusterpoint Database Server and Apache Ignite.

Key	Value
Name	John Doe
Age	45
Gender	Male
Address	55 Bukit Timah Drive

Figure 5: Pictorial representation of a key-value database.

- **Document-oriented Database** - stores data using a JSON-like document-model that mirrors the application code (figure 6 below). Examples of some document-oriented databases are Clusterpoint Database Server and MongoDB.

Figure 6: Pictorial representation of a document-oriented database.

- **Graph Database** - data is stored in a graph-like structure consisting of nodes, edges and properties. Nodes contain entities or instances of data such as a person's information. A node is similar to a row in a relational database. Edges represent the relationships between the nodes. Properties are information associated to the nodes. Refer to figure 7 below. Examples of some graph databases are Neo4j and Amazon Neptune.

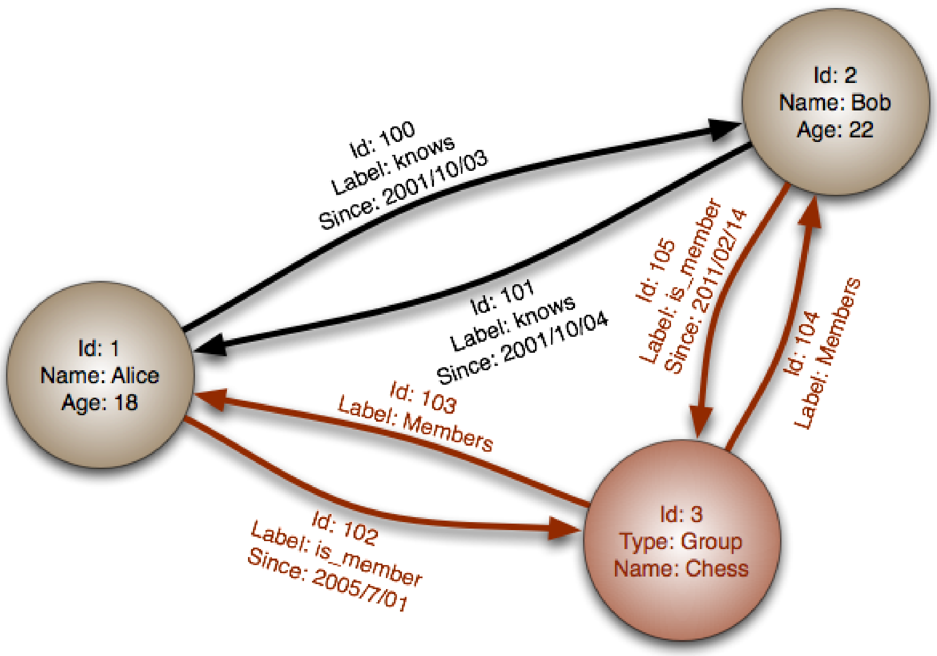

Figure 7: Example of a Graph Database nodes, edges and properties. From Wikipedia Commons.

    - **Wide-column stores** - are databases that stores data in columns instead of rows like in relational databases therefore the names and format of the columns can vary from row to row in the same table (figure 8 below). An Example is Google's Bigtable.

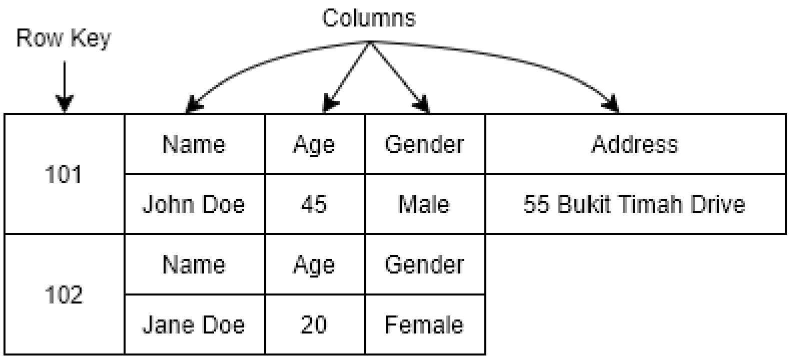

Figure 8: Pictorial representation of wide-column database.

- **Cloud Databases** are databases where the data is stored in a virtual environment on a cloud computing platform. These databases are optimized to execute on such virtual environment using the various cloud computing services (like SaaS, PaaS, IaaS, etc) for accessing and manipulation of the database. Some well known cloud platforms are Amazon Web Services (AWS), Microsoft Azure and Google Cloud SQL.
- **Operational Databases** are a type of database that handles day to day transactions. It creates and updates the database allowing operations such as add, change or delete on the data in real-time. A typical use for this database is to record daily bank transactions like transfers, interest payments and withdrawals.
- **Object-Oriented Databases** are databases that stores data using the object-based data model. This object-based data model is used by object-oriented programming languages to represent objects. An example is ZODB for Python where it stores Python objects using an extended version of Python's built-in object persistence (pickle).
- **Commercial Databases** are huge databases that encompasses many different types of databases (as mentioned above). The usage rights to these databases are sold to organizations for a fee and access to these databases are provided through commercial links. Some examples are Oracle, SQL Server and DB2.
- **Personal Databases** are database that are typically residing on personal computers or computers within a small organization. They are small, easy to manage and generally used by a small group of people or a single user.

1.2 Terminology

As mentioned in the introduction, a database is an organized collection of a set of related data. A DBMS is used by users to interact with the databases and to access and manipulate the data contained in those databases within limits. The following list describes some of the more common database and DBMS related terms that are common between RDBMS and NoSQL databases.

- **ACID** - an acronym that stands for Atomicity, Consistency, Isolation and Durability. These are the properties maintained by any standard DBMS. We will expand on these properties in the section "Transactional Processing".
- **Client/Server** - is an architecture that has 2 parts, namely a client and a server. A server is a program that generally runs on a computer that has direct access to the database and a client is a separate program that communicates with the database server through some specific protocols like Remote Procedure Call (RPC) or Representational State Transfer (REST) API.
- **Connection** - a means of communication between a client and a server on an DBMS. Processes can have multiple connections to one or more databases at any time.
- **Cursor** - a cursor is like an iterator in Python, it allows the traversal of records or documents in a database. Depending on the DBMS used, cursors can be returned after a connection is made or after a query has been executed.
- **Database Definition Language (DDL)** - this language is used to define the database structure. The statements associated with it deals with the creation, modification and removal of databases and objects (such as tables) within it.
- **Data Manipulation Language (DML)** - this language is used for accessing and manipulating data in a database. The statements associated with it generally deals with user requests like retrieving data (SELECT/find), inserting data (INSERT), updating data (UPDATE), deleting records (DELETE), etc from a database.
- **Data Control Language (DCL)** - this language is used to control the access to data stored in a database. Typical commands are GRANT which allows specified users to perform specified tasks and REVOKE which removes a user's accessibility rights to the database.
- **Primary Key** or **_id** - are columns that store unique values that helps identify each record in a database or collection.
- **Transaction Control Language (TCL)** - this language are used to make sure that the changes made by DML are permanent to the database and made visible to other users. Typical commands are COMMIT which saves the transaction on the database and ROLLBACK which restores the database to the last commit.

References

1. Database, https://en.wikipedia.org/wiki/Database
2. Types of databases, https://www.tutorialspoint.com/Types-of-databases
3. Types of Database, https://www.javatpoint.com/types-of-databases