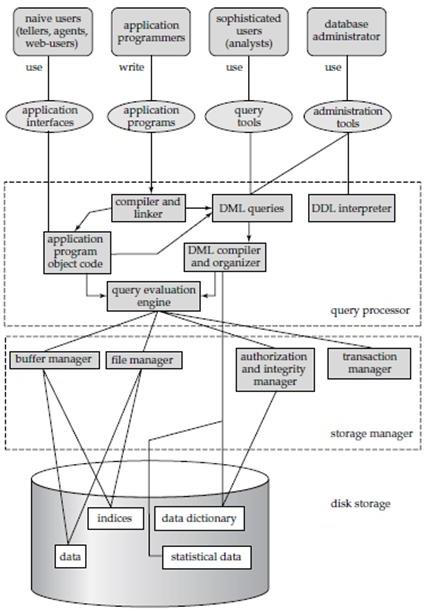
**Unit-2**

**2.1 Database Architecture - Centralized database, Client-server, Parallel database, Distributed database**

****

**We are now in a position to provide a single picture of the various components of a database system and the connections among them.**

**The architecture of a database system is greatly influenced by the underlying computer system on which the database system runs.**

**Database systems can be centralized, or client-server, where one server machine executes work on behalf of multiple client machines.**

**Database systems can also be designed to exploit parallel computer architectures.**

**Distributed databases span multiple geographically separated machines.**

**A database system is partitioned into modules that deal with each**

**of the responsibilities of the overall system.**

**The functional components of a database system can be broadly divided**

**into the storage manager and the query processor components.**

**The storage manager is important because databases**

**typically require a large amount of storage space.**

**The query processor is important because it helps the database system simplify and facilitate access to data.**

**It is the job of the database system to translate updates and queries written in a nonprocedural language(SQL Language),**

**Procedural language(C,Cobol Lan.) at the logical level(what data),**

**into an efficient sequence of operations at the physical level(how data).**

**Database applications are usually partitioned into two or three parts, as in Figure :**

**In a two-tier architecture, the application resides at the client machine, where it invokes database system functionality at the server machine through query language statements.**

**Application program interface standards like ODBC (open database connectivity) and (JAVA)JDBC are used for interaction between the client and the server.**

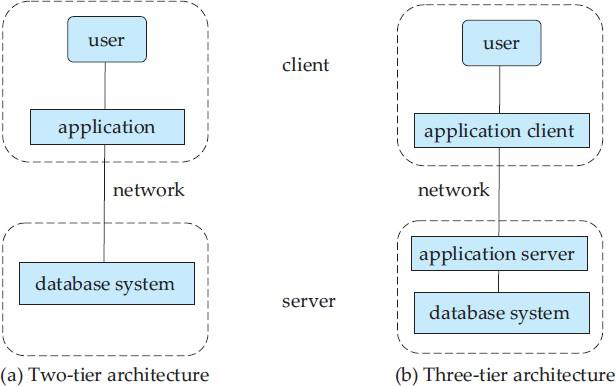
**In contrast, in a three-tier architecture, the client machine acts as merely a front end and does not contain any direct database calls.**

**Instead,the client end communicates with an application server,**

**usually through a forms interface.**

**The application server in turn communicates with a database system to access data. The business logic of the application, which says what actions to carry out under what conditions, is embedded in the application server, instead of being distributed across multiple clients.**

**Three-tier applications are more appropriate for large applications, and for applications that run on the WorldWideWeb.**

****

**Figure :Two-tier and three-tier architectures.**

**Two-tier example:-Excel and word document.**

**Three-tier example:-Web based application**

**Query Processor:**

**The query processor components include**

**· DDL(create,alter,drop) interpreter, which interprets DDL statements and records the definitions in the data dictionary(meta data).**

**· DML compiler, which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.**

**A query can usually be translated into any of a number of alternative evaluation plans that all give the same result.**

**The DML compiler also performs query optimization, that is, it picks the lowest cost evaluation plan from among the alternatives.**

**Query evaluation engine, which executes low-level instructions generated by the DML compiler.**

**Storage Manager:**

**A storage manager is a program module that provides the interface between**

**the low level data stored in the database and the application programs and**

**queries submitted to the system.**

**The storage manager is responsible for the interaction with the file manager.**

**The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system.**

**The storage manager translates the various DML statements into low-level file-system commands. Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.**

**The storage manager components include:**

**· Authorization and integrity manager, which tests for the satisfaction of integrity Constraint and checks the authority of users to access data.**

**· Transaction manager, which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.**

**Transaction - manager ensures that the database remains in a consistent (correct)state despite system failures (e.g., power failures and operating system crashes) and transaction failures.**

**· File manager, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.**

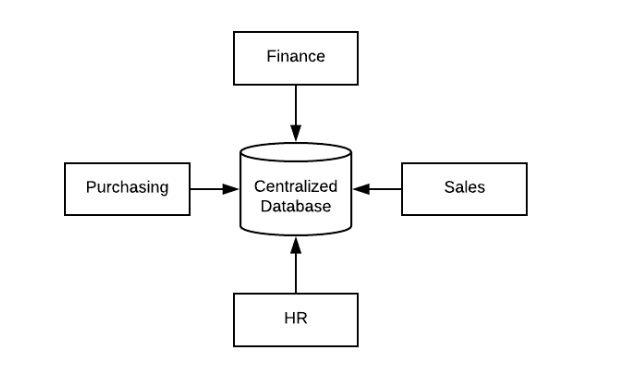
**· Buffer manager, which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory.**

**· The buffer manager is a critical part of the database system, since**

**it enables the database to handle data sizes that are much larger than the size of main memory.**

**Centralized Database Management System**

**A centralized database is stored at a single location such as a mainframe computer. It is maintained and modified from that location only and usually accessed using an internet connection such as a LAN or WAN. The centralized database is used by organizations such as colleges, companies, banks etc.**

****

**As can be seen from the above diagram, all the information for the organization is stored in a single database. This database is known as the centralized database.**

**Advantages of Centralized Database Management System are −**

**• The data integrity is maximized as the whole database is stored at a single physical location. This means that it is easier to coordinate the data and it is as accurate and consistent as possible.**

**• The data redundancy is minimal in the centralized database.So, it is easier to make sure there is no redundant data available.**

**• Since all the data is in one place, there can be stronger security measures around it. So, the centralized database is much more secure.**

**• Data is easily portable because it is stored at the same place.**

**• The centralized database is cheaper than other types of databases as it requires less power and maintenance.**

**• All the information in the centralized database can be easily accessed from the same location and at the same time.**

**Disadvantages**

**• Since all the data is at one location, it takes more time to search and access it. If the network is slow, this process takes even more time.**

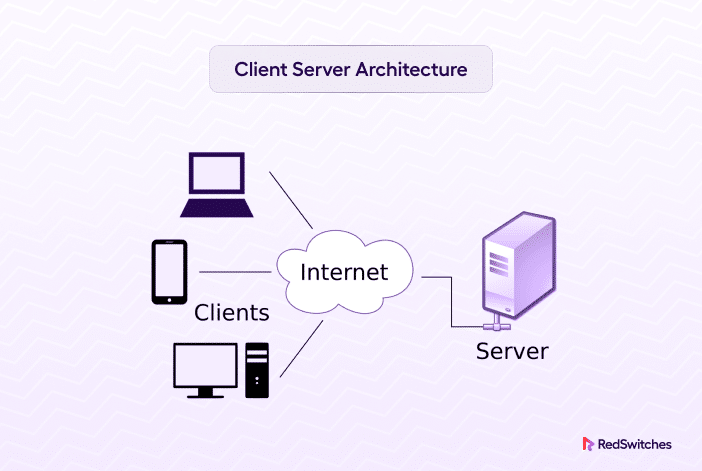
**• There is a lot of data access traffic for the centralized database.**

**• Since all the data is at the same location, if multiple users try to access it simultaneously it creates a problem. This may reduce the efficiency of the system.**

**• If there are no database recovery measures in place and a system failure occurs, then all the data in the database will be destroyed.**

**Client-server Database Architecture in DBMS**

**In client-server architecture many clients connected with one server.**

****

**The server is centerlines.it provides services to all clients.**

**All clients request a different Service.**

**The server displays the results according to the client’s request.**

**Client/server architecture is a computing model in which the server hosts (computer), sends and manages most of the resources and works to be required by the client.**

**This type of architecture has one or more client computers attached to a central server over a network. This system shares different resources.**

**Client/server architecture is also called a networking computing model and client-server network because all the requests and demands are sent over a network.**

**Working of Client-server Database Architecture in DBMS Basically client-server model defines how the server provides services to clients Server is a centralized computer that provides services to all attached clients.**

**For example file server, web server, etc.**

**each the basic work of the server to provide services to each client.**

**The client can be a laptop computer, tablets, and smartphones, etc.**

**The server has many types of relationships with clients.**

**Many servers have one too many relationships with clients.**

**in one too many relationships many clients connected with one server.**

**When one client wants to communicate with the server. The server may accept or reject the request of clients.**

**When the server computer accepts the request of clients then the server maintains a connection according to a defined protocol.**

**The protocol rules over the network.**

**That must be followed for any network connection.**

**If one client wants to send an email over the network.**

**It requests the server, the SMTP (the protocol that is SMTP stands for simple mail transfer protocol that is used to transfer mail over the network. SMTP is a set of commands or commands that check authentication and the transfer of email.**

**When configuring the settings for your email program, you usually need to set the SMTP server to your local Internet Service Provider’s SMTP settings.**

**Advantages of Client-server Database Architecture in DBMS**

**1. All the data and resources are controlled by the server this way all data and resources are very consistent.**

**2. You can easily increase the number of clients in this architecture at any time. This all increases the scalability of the network.**

**3. This is very easy to maintain and you can easily repair, replace or add clients in this network. the independence of the changes also known as encapsulation.**

**4. This network is very easy to use and it is not complicated.**

**Disadvantages of Client-server Database Architecture in DBMS**

**1. Traffic is a big problem in this network.**

**2. When you add large numbers of the client with server this network will be more complicated.**

**3. When the server goes down all the clients are not able to send their request. The whole work will be stopped**

**4. The hardware and software are very expensive.**

**Parallel database system improves performance of data processing using multiple resources in parallel, like multiple CPU and disks are used parallelly. It also performs many parallelization operations like data loading and query processing.**

**Goals of Parallel Databases The concept of Parallel Database was built with a goal to:**

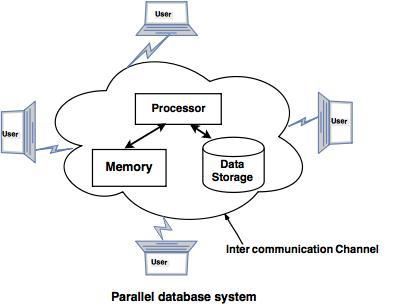
**Improve performance: The performance of the system can be improved by connecting multiple CPU and disks in parallel.**

**Many small processors can also be connected in parallel.**

**Improve availability of data: Data can be copied to multiple locations to improve the availability of data.**

**For example: if a module contains a relation (table in database) which is unavailable then it is important to make it available from another module. Improve reliability: Reliability of the system is improved with completeness, accuracy and availability of data.**

**Provide distributed access of data: Companies having many branches in multiple cities can access data with the help of a parallel database system.**

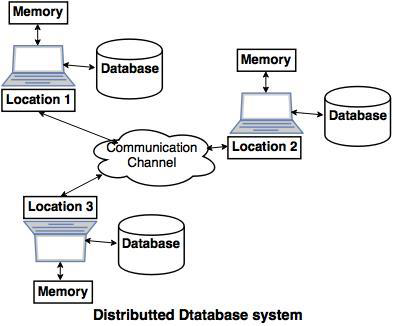
****

**Distributed databases:-**

**• Distributed database is a system in which storage devices are not connected to a common processing unit.**

**• Database is controlled by Distributed Database Management System and data may be stored at the same location or spread over the interconnected network. It is a loosely coupled system.**

**• Shared nothing architecture is used in distributed databases.**

****

**• The above diagram is a typical example of a distributed database system, in which a communication channel is used to communicate with the different locations and every system has its own memory and database.**

**The concept of distributed database was built with a goal to improve:**

**Reliability: In distributed database system, if one system fails down or stops working for some time another system can complete the task.**

**Performance: Performance can be achieved by distributing databases over different locations. So the databases are available to every location which is easy to maintain.**

**2.2 Introduction to data models -**

**1)Relational,**

**2) E-R,**

**3) Object-Oriented**

**Relational Model in DBMS**

**E.F. Codd proposed the relational Model to model data in the form of relations or tables. After designing the conceptual model of the Database using** [**ER diagram**](https://www.geeksforgeeks.org/introduction-of-er-model/)**, we need to convert the conceptual model into a relational model which can be implemented using any** [**RDBMS**](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/) **language like Oracle SQL, MySQL, etc. So we will see what the Relational Model is.**

**The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. Tables are also known as relations. The relational model is an example of a record-based model. Record-based models are so named because the database is structured in fixed-format records of several types. Each table contains records of a particular type. Each record type defines a fixed number of fields, or attributes. The columns of the table correspond to the attributes of the record type. The relational data model is the most widely used data model, and a vast majority of current database systems are based on the relational model.**

**What is the Relational Model?**

**The relational model represents how data is stored in Relational Databases. A relational database consists of a collection of tables, each of which is assigned a unique name.**

**Consider a relation STUDENT with attributes ROLL\_NO, NAME, ADDRESS, PHONE, and AGE shown in the table.**

**Table Student**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **AGE** |
| **1** | **RAM** | **DELHI** | **9455123451** | **18** |
| **2** | **RAMESH** | **GURGAON** | **9652431543** | **18** |
| **3** | **SUJIT** | **ROHTAK** | **9156253131** | **20** |

**Important Terminologies**

* **Attribute: Attributes are the properties that define an entity. e.g.; ROLL\_NO, NAME, ADDRESS**
* **Relation Schema: A relation schema defines the structure of the relation and represents the name of the relation with its attributes. e.g.; STUDENT (ROLL\_NO, NAME, ADDRESS, PHONE, and AGE) is the relation schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.**
* **Tuple: Each row in the relation is known as a tuple. The above relation contains 4 tuples, one of which is shown as:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **RAM** | **DELHI** | **9455123451** | **18** |

* **Relation Instance: The set of tuples of a relation at a particular instance of time is called a relation instance. Table 1 shows the relation instance of STUDENT at a particular time. It can change whenever there is an insertion, deletion, or update in the database.**
* **Degree: The number of attributes in the relation is known as the degree of the relation. The STUDENT relation defined above has degree 5.**
* **Cardinality: The number of tuples in a relation is known as** [**cardinality**](https://www.geeksforgeeks.org/cardinality-in-dbms/)**. The STUDENT relation defined above has cardinality 4.**
* **Column: The column represents the set of values for a particular attribute. The column ROLL\_NO is extracted from the relation STUDENT.**

|  |
| --- |
| **ROLL\_NO** |
| **1** |
| **2** |
| **3** |

* **NULL Values: The value which is not known or unavailable is called a NULL value. It is represented by blank space. e.g.; PHONE of STUDENT having ROLL\_NO 4 is NULL.**
* **Relation Key: These are basically the keys that are used to identify the rows uniquely or also help in identifying tables. These are of the following types.**
  + [**Primary Key**](https://www.geeksforgeeks.org/primary-key-constraint-in-sql/)
  + [**Candidate Key**](https://www.geeksforgeeks.org/difference-between-primary-and-candidate-key/)
  + [**Super Key**](https://www.geeksforgeeks.org/difference-between-super-key-and-candidate-key/)
  + [**Foreign Key**](https://www.geeksforgeeks.org/postgresql-foreign-key/)
  + [**Alternate Key**](https://www.geeksforgeeks.org/sql-alternate-key/)
  + [**Composite Key**](https://www.geeksforgeeks.org/composite-key-in-sql/)

**Constraints in Relational Model**

**While designing the Relational Model, we define some conditions which must hold for data present in the database are called Constraints. These constraints are checked before performing any operation (insertion, deletion, and updation ) in the database. If there is a violation of any of the constraints, the operation will fail.**

**Domain Constraints**

**These are attribute-level constraints. An attribute can only take values that lie inside the domain range. e.g.; If a constraint AGE>0 is applied to STUDENT relation, inserting a negative value of AGE will result in failure.**

**Key Integrity**

**Every relation in the database should have at least one set of attributes that defines a tuple uniquely. Those set of attributes is called keys. e.g.; ROLL\_NO in STUDENT is key. No two students can have the same roll number. So a key has two properties:**

* **It should be unique for all tuples.**
* **It can’t have NULL values.**

**Referential Integrity**

**When one attribute of a relation can only take values from another attribute of the same relation or any other relation, it is called** [**referential integrity**](https://www.geeksforgeeks.org/cascading-referential-integrity-constraints-in-sql-server-management-studio/)**. Let us suppose we have 2 relations**

**Table Student**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **AGE** | **BRANCH\_CODE** |
| **1** | **RAM** | **DELHI** | **9455123451** | **18** | **CS** |
| **2** | **RAMESH** | **GURGAON** | **9652431543** | **18** | **CS** |
| **3** | **SUJIT** | **ROHTAK** | **9156253131** | **20** | **ECE** |
| **4** | **SURESH** | **DELHI** |  | **18** | **IT** |

**Table Branch**

|  |  |
| --- | --- |
| **BRANCH\_CODE** | **BRANCH\_NAME** |
| **CS** | **COMPUTER SCIENCE** |
| **IT** | **INFORMATION TECHNOLOGY** |
| **ECE** | **ELECTRONICS AND COMMUNICATION ENGINEERING** |
| **CV** | **CIVIL ENGINEERING** |

**BRANCH\_CODE of STUDENT can only take the values which are present in BRANCH\_CODE of BRANCH which is called referential integrity constraint. The relation which is referencing another relation is called REFERENCING RELATION (STUDENT in this case) and the relation to which other relations refer is called REFERENCED RELATION (BRANCH in this case).**

**Anomalies in the Relational Model**

**An** [**anomaly**](https://www.geeksforgeeks.org/anomalies-in-relational-model/) **is an irregularity or something which deviates from the expected or normal state. When designing databases, we identify three types of anomalies: Insert, Update, and Delete.**

**Insertion Anomaly in Referencing Relation**

**We can’t insert a row in REFERENCING RELATION if referencing attribute’s value is not present in the referenced attribute value. e.g.; Insertion of a student with BRANCH\_CODE ‘ME’ in STUDENT relation will result in an error because ‘ME’ is not present in BRANCH\_CODE of BRANCH.**

**Deletion/ Updation Anomaly in Referenced Relation:**

**We can’t delete or update a row from REFERENCED RELATION if the value of REFERENCED ATTRIBUTE is used in the value of REFERENCING ATTRIBUTE. e.g; if we try to delete a tuple from BRANCH having BRANCH\_CODE ‘CS’, it will result in an error because ‘CS’ is referenced by BRANCH\_CODE of STUDENT, but if we try to delete the row from BRANCH with BRANCH\_CODE CV, it will be deleted as the value is not been used by referencing relation. It can be handled by the following method:**

**On Delete Cascade**

**It will delete the tuples from REFERENCING RELATION if the value used by REFERENCING ATTRIBUTE is deleted from REFERENCED RELATION. e.g.; For, if we delete a row from BRANCH with BRANCH\_CODE ‘CS’, the rows in STUDENT relation with BRANCH\_CODE CS (ROLL\_NO 1 and 2 in this case) will be deleted.**

**On Update Cascade**

**It will update the REFERENCING ATTRIBUTE in REFERENCING RELATION if the attribute value used by REFERENCING ATTRIBUTE is updated in REFERENCED RELATION. e.g;, if we update a row from BRANCH with BRANCH\_CODE ‘CS’ to ‘CSE’, the rows in STUDENT relation with BRANCH\_CODE CS (ROLL\_NO 1 and 2 in this case) will be updated with BRANCH\_CODE ‘CSE’.**

**Codd Rules in Relational Model**

**Edgar F Codd proposed the relational database model where he stated rules. . For any database to be the perfect one, it has to follow the rules.**

**Advantages of the Relational Model**

* **Simple model: Relational Model is simple and easy to use in comparison to other languages.**
* **Flexible: Relational Model is more flexible than any other relational model present.**
* **Secure: Relational Model is more secure than any other relational model.**
* **Data Accuracy: Data is more accurate in the relational data model.**
* **Data Integrity: The integrity of the data is maintained in the relational model.**
* **Operations can be Applied Easily: It is better to perform operations in the relational model.**

**Disadvantages of the Relational Model**

* **Relational Database Model is not very good for large databases.**
* **Sometimes, it becomes difficult to find the relation between tables.**
* **Because of the complex structure, the response time for queries is high.**

**Object Oriented Data Model**

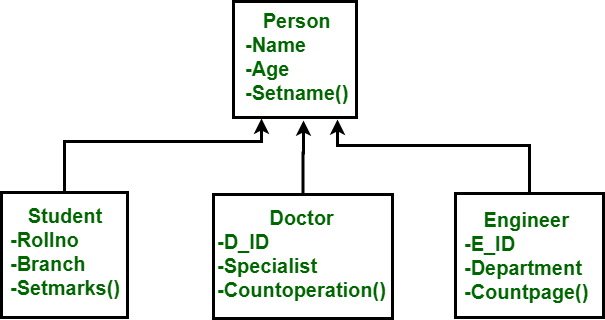
**To represent the complex real world problems there was a need for a data model that is closely related to real world. Object Oriented Data Model represents the real world problems easily.**

**Object Oriented Data Model :   
In Object Oriented Data Model, data and their relationships are contained in a single structure which is referred as object in this data model. In this, real world problems are represented as objects with different attributes. All objects have multiple relationships between them. Basically, it is combination of Object Oriented programming and Relational Database Model as it is clear from the following figure :**

**Object Oriented Data Model**

**= Combination of Object Oriented Programming + Relational database model**

**Components of Object Oriented Data Model :**

****

***Basic Object Oriented Data Model***

* **Objects –   
  An object is an abstraction of a real world entity or we can say it is an instance of class. Objects encapsulates data and code into a single unit which provide data abstraction by hiding the implementation details from the user. For example: Instances of student, doctor, engineer in above figure.**
* **Attribute –   
  An attribute describes the properties of object. For example: Object is STUDENT and its attribute are Roll no, Branch, Setmarks() in the Student class.**
* **Methods –   
  Method represents the behavior of an object. Basically, it represents the real-world action. For example: Finding a STUDENT marks in above figure as Setmarks().**
* **Class –   
  A class is a collection of similar objects with shared structure i.e. attributes and behavior i.e. methods. An object is an instance of class. For example: Person, Student, Doctor, Engineer in above figure.**

**class student**

**{**

**char Name[20];**

**int roll\_no;**

**--**

**--**

**public:**

**void search();**

**void update();**

**}**

**In this example, students refers to class and S1, S2 are the objects of class which can be created in main function.**

* **Inheritance –   
  By using inheritance, new class can inherit the attributes and methods of the old class i.e. base class. For example: as classes Student, Doctor and Engineer are inherited from the base class Person.**

**Advantages of Object Oriented Data Model :**

* **Codes can be reused due to inheritance.**
* **Easily understandable.**
* **Cost of maintenance can reduced due to reusability of attributes and functions because of inheritance.**

**Disadvantages of Object Oriented Data Model :**

* **It is not properly developed so not accepted by users easily.**