UNIT- 1

1. WHAT IS SCHEMA, INSTANCE, METADATA, GENEARLIAZATION, , foreign key,SPECIALIZATION, data dictionary.

Ans – SCHEMA - A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them

INSTANCE - The situation where a data or information is stored in the database at a particular moment of time is called an instance. An instance is also called a current state or database state. The database schema that defines variables in tables which belong to a specific database, the records of these variables at a particular moment are called the instance of the database.

METADATA - **Metadata** is simply defined as data about data. It means it is a description and context of the data. It helps to organize, find and understand data. Let me explain to you by giving a real-world example of metadata:

Every time you take a photo with today’s cameras a bunch of metadata is gathered and saved with it. Such as

* File name,
* Size of the file,
* Date and time,
* Camera settings etc.

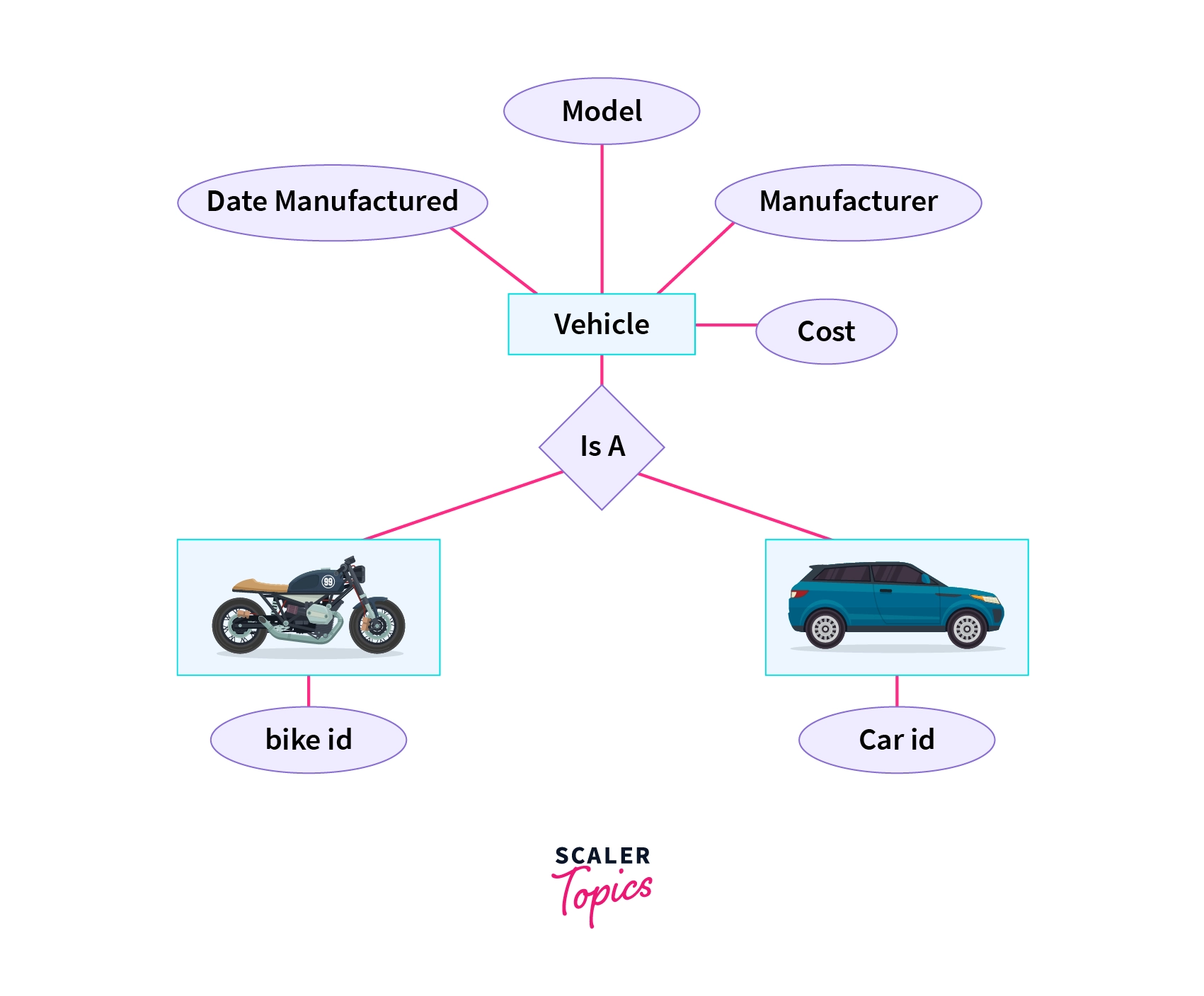
DATA DESCRIPTION - A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them.

A data dictionary contains a list of all files in the database, the number of records in each file, and the names and types of each field. Most database management systems keep the data dictionary hidden from users to prevent them from accidentally destroying its contents.

FOREIGN KEY - A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it references the primary key of another table, thereby establishing a link between them.

GENERALIZATION - Generalization is a process in which a new entity is formed using the common attributes of two or more entities. Generalization simplifies the ER diagram by clubbing the common attributes together. Below are some characteristics of generalization:

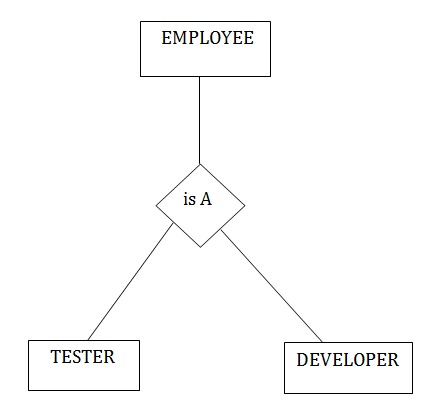
* Generalization follows the bottom-up approach.
* It generalizes or simplifies the entities.
* Higher-level entities can also be combined with lower-level entities.



SPECIALAIZATION - Specialization is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.

* Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
* Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added.

**For example:** In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.



1. WHAT IS DATA MODELS AND TYPES?

ANS - A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. While the **Relational Model** is the most widely used database model, there are other models too:

* Hierarchical Model
* Network Model
* [Entity-relationship Model](https://www.studytonight.com/dbms/er-model-concepts.php)
* [Relational Model](https://www.studytonight.com/dbms/rdbms-concept.php)

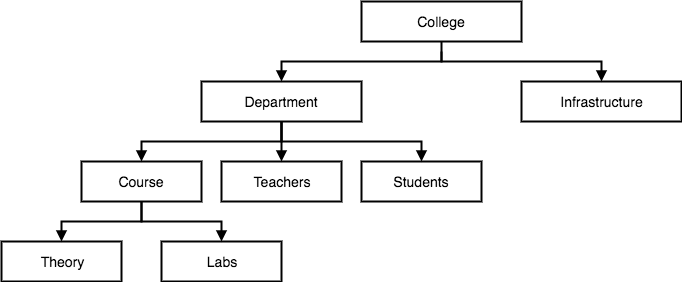
## Hierarchical Model

This database model organises data into a tree-like-structure, with a single root, to which all the other data is linked. The heirarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

This model efficiently describes many real-world relationships like index of a book, recipes etc.

In hierarchical model, data is organised into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of-course many students.

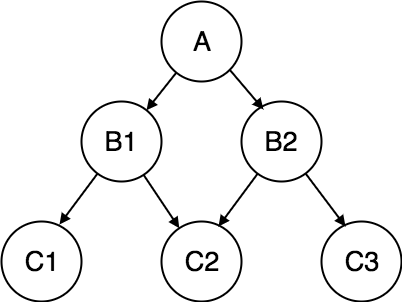


## Network Model

This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships.

This was the most widely used database model, before Relational Model was introduced.



## Entity-relationship Model

In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes.

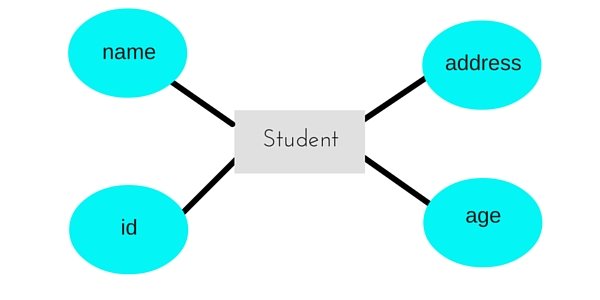
Different entities are related using relationships.

E-R Models are defined to represent the relationships into pictorial form to make it easier for different stakeholders to understand.

This model is good to design a database, which can then be turned into tables in relational model(explained below).

Let's take an example, If we have to design a School Database, then **Student** will be an **entity** with **attributes** name, age, address etc. As **Address** is generally complex, it can be another **entity** with **attributes** street name, pincode, city etc, and there will be a relationship between them.

Relationships can also be of different types. To learn about [E-R Diagrams](https://www.studytonight.com/dbms/er-diagram.php) in details, click on the link.



## Relational Model

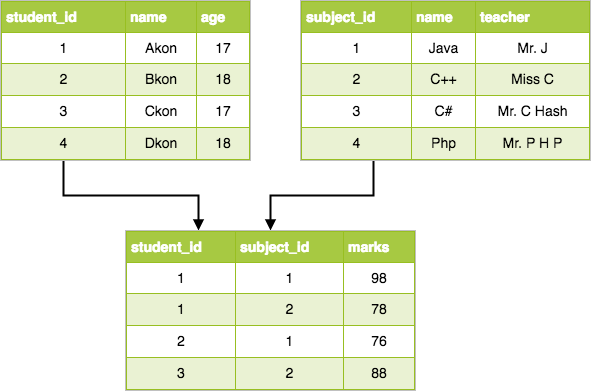
In this model, data is organised in two-dimensional **tables** and the relationship is maintained by storing a common field.

This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, infact, we can say the only database model used around the world.

The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.

Hence, tables are also known as **relations** in relational model.

In the coming tutorials we will learn how to design tables, [normalize them](https://www.studytonight.com/dbms/database-normalization.php) to reduce data redundancy and how to use [Structured Query language](https://www.studytonight.com/dbms/introduction-to-sql.php) to access data from tables.



1. DBMS ARCHITECTURE?

ANS - Database Architecture is logically of two types:

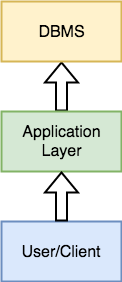
2-tier DBMS architecture

3-tier DBMS architecture

## 2-tier DBMS Architecture

2-tier DBMS architecture includes an **Application layer** between the user and the DBMS, which is responsible to communicate the user's request to the database management system and then send the response from the DBMS to the user.

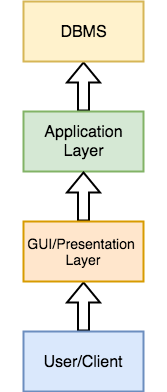
An application interface known as **ODBC**(Open Database Connectivity) provides an API that allow client side program to call the DBMS. Most DBMS vendors provide ODBC drivers for their DBMS.



Such an architecture provides the DBMS extra security as it is not exposed to the End User directly. Also, security can be improved by adding security and authentication checks in the Application layer too.

## 3-tier DBMS Architecture

3-tier DBMS architecture is the most commonly used architecture for web applications.



It is an extension of the 2-tier architecture. In the 2-tier architecture, we have an application layer which can be accessed programatically to perform various operations on the DBMS. The application generally understands the Database Access Language and processes end users requests to the DBMS.

In 3-tier architecture, an additional Presentation or GUI Layer is added, which provides a graphical user interface for the End user to interact with the DBMS.

For the end user, the GUI layer is the Database System, and the end user has no idea about the application layer and the DBMS system.

If you have used **MySQL**, then you must have seen **PHPMyAdmin**, it is the best example of a 3-tier DBMS architecture.

3. what is functions of dba? And define data independence?

# Ans - Functions and responsibilities of DBAs

**DBA:** person in the organization who controls the design and the use of the database refers as DBA.

### *1. Schema Definition:*

* The DBA definition the logical Schema of the database.A Schema refers to the overall logical structure of the database.
* According to this schema, database will be developed to store required data for an organization.

### *2. Storage Structure and Access Method Definition:*

* The DBA decides how the data is to be represented in the stored database.

### *3. Assisting Application Programmers:*

* The DBA provides assistance to application programmers to develop application programs.

### *4. Physical Organization Modification:*

* The DBA modifies the physical organization of the database to reflext the changing needs of the organization or to improve performance.

### *5. Approving Data Access:*

* The DBA determines which user needs access to which part of the database.
* According to this,various types of authorizations are granted to different users.

### *6. Monitoring Performance:*

* The DBA monitors performance of the system.The DBA ensures that better performance is maintained by making changes in physical or logical schema if required.

### *7. Backup and Recovery:*

* Database should not be lost or damaged.
* The DBA ensures this periodically backing up the database on magnetic tapes or remote servers.
* In case of failure, such as virus attack database is recovered from this backup.

Data independence is **the ability to modify the scheme without affecting the programs and the application to be rewritten**. Data is separated from the programs, so that the changes made to the data will not affect the program execution and the application.

1. Difference between file system and database management system?

Ans -

|  |  |  |
| --- | --- | --- |
| Basis | File System | DBMS |
| Structure | The file system is software that manages and organizes the files in a storage medium within a computer. | DBMS is software for managing the database. |
| Data Redundancy | Redundant data can be present in a file system. | In DBMS there is no redundant data. |
| Backup and Recovery | It doesn’t provide backup and recovery of data if it is lost. | It provides backup and recovery of data even if it is lost. |
| Query processing | There is no efficient query processing in the file system. | Efficient query processing is there in DBMS. |
| Consistency | There is less data consistency in the file system. | There is more data consistency because of the process of normalization. |
| Complexity | It is less complex as compared to DBMS. | It has more complexity in handling as compared to the file system. |
| Security Constraints | File systems provide less security in comparison to DBMS. | DBMS has more security mechanisms as compared to file systems. |
| Cost | It is less expensive than DBMS. | It has a comparatively higher cost than a file system. |
| Data Independence | There is no data independence. | In DBMS data independence exists. |
| User Access | Only one user can access data at a time. | Multiple users can access data at a time. |
| Meaning | The user has to write procedures for managing databases | The user not required to write procedures. |
| Sharing | Data is distributed in many files. So, not easy to share data | Due to centralized nature sharing is easy |
| Data Abstraction | It give details of storage and representation of data | It hides the internal details of Database |
| Integrity Constraints | Integrity Constraints are difficult to implement | Integrity constraints are easy to implement |
| Example | Cobol, C++ | Oracle, SQL Server |