

Introduction of ER Model

An attribute is a property or characteristic of an entity.

An entity may contain any number of attributes.

Simple attribute :

An attribute that cannot be further subdivided into components is a simple attribute.

Example: The roll number of a student, the id number of an employee.

Composite attribute :

An attribute that can be split into components is a composite attribute.

Example: The address can be further split into house number, street number, city, state, country, and pin code, the name can also be split into first name middle name, and last name.

Single-valued attribute :

The attribute which takes up only a single value for each entity instance is a single-valued attribute.

Example: The age of a student.

Multi-valued attribute :

The attribute which takes up more than a single value for each entity instance is a multi-valued attribute.

Example: Phone number of a student: Landline and mobile.

Derived attribute :

An attribute that can be derived from other attributes is derived attributes.

Example: Total and average marks of a student.

Complex attribute :

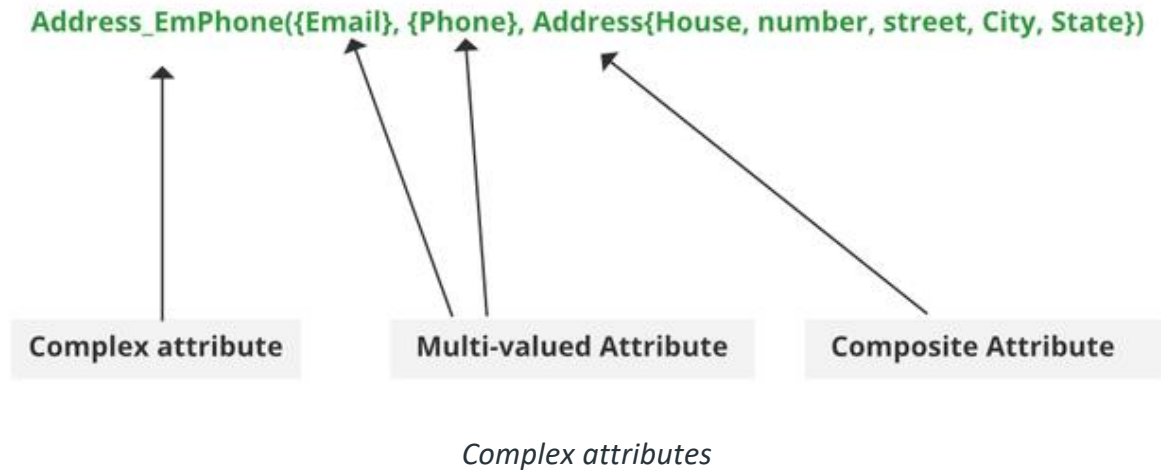
Those attributes, which can be formed by the nesting of composite and multi-valued attributes, are called "*Complex Attributes*".

Representation:

Complex attributes are the nesting of two or more composite and multi-valued attributes.

For example: let us consider a person having multiple phone numbers, emails, and an address.

Here, phone number and email are examples of multi-valued attributes and address is an example of the composite attribute, because it can be divided into house number, street, city, and state.



Components:

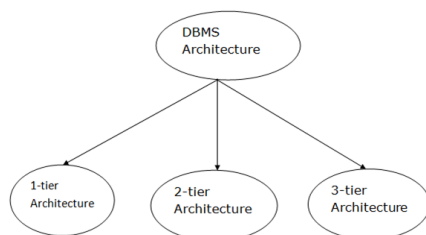
Email, Phone number, Address(All are separated by commas and multi-valued components are represented between curly braces).

Complex Attribute: Address_EmPhone(You can choose any name).

DBMS Architecture

- The DBMS design depends upon its architecture.
- The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers that are connected with networks.
- DBMS architecture depends upon how users are connected to the database to get their request one.

Types of DBMS Architecture



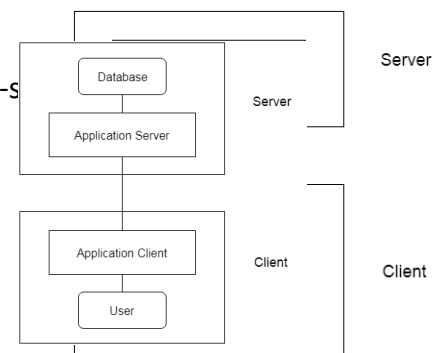
Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: **2-tier architecture** and **3-tier architecture**.

1-Tier Architecture

- the database is directly available to the user.
- Any changes done here will directly be done on the database itself
- is used for development of the local application

2-Tier Architecture

- The 2-Tier architecture is same as basic client-server.
- applications on the client end can directly communicate with the database at the server side.
- The user interfaces and application programs are run on the client-side
- The server side is responsible to provide the functionalities



3-Tier Architecture

- contains another layer between the client and server.
- End user has no idea about the existence of the database beyond the application server.
- is used in case of large web application.

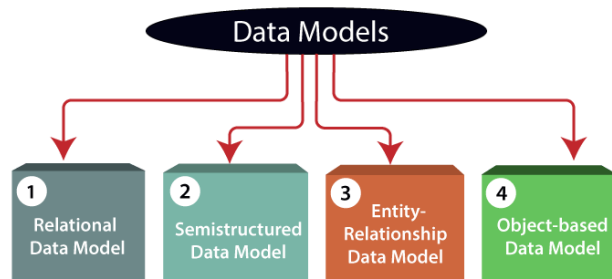
?what is disadvantages of file oriented approach?

Limited Data Sharing:

Data are scattered in various files. also different files may have different formats and these files may be stored in different folders may be of different departments. So, due to this data isolation, it is difficult to share data among different applications.

Data Models

Data Model is the modeling of the data description, data semantics, and consistency constraints of the data. It provides the conceptual tools for describing the design of a database at each level of data abstraction.



1) Relational Data Model: This type of model designs the data in the form of rows and columns within a table.

2) Entity-Relationship Data Model: An ER model is the logical representation of data as objects and relationships among them.

3) Object-based Data Model: An extension of the ER model with notions of functions, encapsulation, and object identity, as well.

4) Semistructured Data Model: This type of data model is different from the other three data models (explained above)

Basis	File System	DBMS
Structure	manages and organizes the files in a storage medium within a computer.	managing the database.
Data Redundancy	Redundant data can be present in a file system.	no redundant data.
Backup	It doesn't provide backup.	It provides backup
Query processing	There is no efficient query processing	Efficient query processing
Consistency	There is less data consistency	There is more data consistency
Complexity	It is less complex	It has more complexity

?5 main functions of dbms administration?

- Software installation and Maintenance.
- Data Extraction, Transformation, and Loading.
- Specialised Data Handling.
- Database Backup and Recovery.
- Security.

Advantage	Disadvantage
A database management system (DBMS) may store and retrieve data in a series of manners.	The expense of DBMS hardware and software is enormously expensive, which impacts your organization's overall budget.
A database management system (DBMS) is a reliable manager for handling numerous applications requiring identical data.	A database management system (DBMS) is inadequate for doing complex mathematical calculations.
Data Integrity and Security are guaranteed by DBMS.	Since most database management systems are difficult and complicated, employees must always be instructed on how to operate them.
The DBMS implements integrity constraints to provide a high degree of protection against unauthorized access to information.	Many individuals using the same application simultaneously time potentially lead to data loss.

.what are the commands in dbms?

- SELECT - extracts data from a database.
- UPDATE - updates data in a database.
- DELETE - deletes data from a database.
- INSERT INTO - inserts new data into a database.
- CREATE DATABASE - creates a new database

SELECT (σ)

The SELECT operation is used for selecting a subset of the tuples according to a given selection condition.

$\sigma_p(r)$

Example 1

$\sigma_{\text{topic} = \text{"Database"}}(\text{Tutorials})$

Output – Selects tuples from Tutorials where topic = 'Database'.

Projection(π)

The projection eliminates all attributes of the input relation but those mentioned in the projection list.

Example of Projection:

Consider the following table

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

Here, the projection of CustomerName and status will give

$\pi_{\text{CustomerName, Status}}(\text{Customers})$

CustomerName	Status
Google	Active
Amazon	Active
Apple	Inactive
Alibaba	Active

Union operation (\cup)

UNION is symbolized by \cup symbol. It includes all tuples that are in tables A or in B. It also eliminates duplicate tuples.

Example

Table A		Table B	
column 1	column 2	column 1	column 2
1	1	1	1
1	2	1	3

A U B gives

Table A U B	
column 1	column 2
1	1
1	2
1	3

Static (Embedded) SQL

In Static SQL, how database will be accessed is predetermined in the embedded SQL statement.

It is more swift and efficient.

SQL statements are compiled at compile time.

Parsing, Validation, Optimization and Generation of application plan are done at compile time.

It is generally used for situations where data is distributed uniformly.

Dynamic (Interactive) SQL

In Dynamic SQL, how database will be accessed is determined at run time.

It is less swift and efficient.

SQL statements are compiled at run time.

Parsing, Validation, Optimization and Generation of application plan are done at run time.

It is generally used for situations where data is distributed non uniformly.

EXECUTE IMMEDIATE, EXECUTE and PREPARE statements are not used.

EXECUTE IMMEDIATE, EXECUTE and PREPARE statements are used.

It is less flexible.

It is more flexible.

data types in SQL and constraints

- [NOT NULL](#) - Ensures that a column cannot have a NULL value
- [UNIQUE](#) - Ensures that all values in a column are different
- [PRIMARY KEY](#) - A combination of a **NOT NULL** and **UNIQUE**. Uniquely identifies each row in a table
- [FOREIGN KEY](#) - Prevents actions that would destroy links between tables
- [CHECK](#) - Ensures that the values in a column satisfies a specific condition
- [DEFAULT](#) - Sets a default value for a column if no value is specified
- [CREATE INDEX](#) - Used to create and retrieve data from the database very quickly

Relational Algebra is procedural query language, which takes Relation as input and generate relation as output. Relational algebra mainly provides theoretical foundation for relational databases and SQL.