**Chapter 3 : Database Design**

A database is a collection of information  that is organized so that it can be easily accessed, managed and updated. Computer databases typically contain aggregations of data records or files, containing information about sales transactions or interactions with specific customers.

In a [relational database](https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system), digital information about a specific customer is organized into rows, columns and tables which are indexed to make it easier to find relevant information through [SQL](https://searchsqlserver.techtarget.com/definition/SQL) or [NoSQL](https://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL) queries. In contrast, a [graph database](https://whatis.techtarget.com/definition/graph-database) uses nodes and edges to define relationships between data entries and queries require a special [semantic search](https://whatis.techtarget.com/definition/semantic-search) syntax.  As of this writing, [SPARQL](https://whatis.techtarget.com/definition/SPARQL) is the only semantic query language that is approved by the World Wide Web Consortium ([W3C](https://whatis.techtarget.com/definition/W3C-World-Wide-Web-Consortium)).

Typically, the database manager provides users with the ability to control read/write access, specify report generation and analyze usage. Some databases offer [ACID](https://searchsqlserver.techtarget.com/definition/ACID) (atomicity, consistency, isolation and durability) compliance to guarantee that data is consistent and that transactions are complete.

**Types of databases :**

Databases have evolved since their inception in the 1960s, beginning with hierarchical and network databases, through the 1980s with [object-oriented databases](https://searchoracle.techtarget.com/definition/object-oriented-database-management-system), and today with SQL and NoSQL databases and [cloud databases](https://searchcloudapplications.techtarget.com/definition/cloud-database).

In one view, databases can be classified according to content type: bibliographic, full text, numeric and images. In computing, databases are sometimes classified according to their organizational approach. There are many different kinds of databases, ranging from the most prevalent approach, the relational database, to a [distributed database](https://searchoracle.techtarget.com/definition/distributed-database), [cloud database](https://searchcloudapplications.techtarget.com/definition/cloud-database), graph database or NoSQL database.

* **Relational database**

A [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database), invented by [E.F. Codd](https://searchoracle.techtarget.com/definition/E-F-Codd) at IBM in 1970, is a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways.

Relational databases are made up of a set of tables with data that fits into a predefined category. Each table has at least one data category in a column, and each row has a certain data instance for the categories which are defined in the columns.

The Structured Query Language (SQL) is the standard user and application program interface for a relational database. Relational databases are easy to extend, and a new data category can be added after the original database creation without requiring that you modify all the existing applications.

* **Distributed database**

A distributed database is a database in which portions of the database are stored in multiple physical locations, and in which processing is dispersed or replicated among different points in a network.

Distributed databases can be homogeneous or heterogeneous. All the physical locations in a homogeneous [distributed database system](https://searchsqlserver.techtarget.com/definition/DDBMS) have the same underlying hardware and run the same operating systems and database applications. The hardware, operating systems or database applications in a heterogeneous distributed database may be different at each of the locations.

* **Cloud database**

A cloud database is a database that has been optimized or built for a virtualized environment, either in a hybrid cloud, public cloud or private cloud. Cloud databases provide benefits such as the ability to pay for storage capacity and bandwidth on a per-use basis, and they provide scalability on demand, along with [high availability](https://searchdatacenter.techtarget.com/definition/high-availability).

A cloud database also gives enterprises the opportunity to support business applications in a [software-as-a-service](https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service) deployment.

* **NoSQL database**

[NoSQL databases](https://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL) are useful for large sets of distributed data.

NoSQL databases are effective for big data performance issues that relational databases aren't built to solve. They are most effective when an organization must analyze large chunks of [unstructured](https://searchbusinessanalytics.techtarget.com/definition/unstructured-data) data or data that's stored across multiple [virtual servers](https://searchnetworking.techtarget.com/definition/virtual-server) in the cloud.

* **Object-oriented database**

Items created using [object-oriented programming languages](https://searchmicroservices.techtarget.com/definition/object-oriented-programming-OOP) are often stored in relational databases, but object-oriented databases are well-suited for those items.

An object-oriented database is organized around objects rather than actions, and data rather than logic. For example, a multimedia record in a relational database can be a definable data object, as opposed to an alphanumeric value.

* **Graph database**

A graph-oriented database, or [graph database](https://whatis.techtarget.com/definition/graph-database), is a type of NoSQL database that uses [graph theory](https://whatis.techtarget.com/definition/graph-theory) to store, map and query relationships. Graph databases are basically collections of nodes and edges, where each node represents an entity, and each edge represents a connection between nodes.

Graph databases are growing in popularity for analyzing interconnections. For example, companies might use a graph database to [mine data](https://searchsqlserver.techtarget.com/definition/data-mining) about customers from [social media](https://whatis.techtarget.com/definition/social-media).

Graph databases often employ [SPARQL](https://whatis.techtarget.com/definition/SPARQL), a [declarative programming](https://searchitoperations.techtarget.com/definition/declarative-programming) language and protocol for [graph database](https://whatis.techtarget.com/definition/graph-database) analytics. SPARQL has the capability to perform all the analytics that [SQL](https://searchsqlserver.techtarget.com/definition/SQL) can perform, plus it can be used for semantic analysis, the examination of relationships. This makes it useful for performing analytics on data sets that have both [structured](https://whatis.techtarget.com/definition/structured-data) and [unstructured](https://searchbusinessanalytics.techtarget.com/definition/unstructured-data)data. SPARQL allows users to perform analytics on information stored in a relational database, as well as friend-of-a-friend (FOAF) relationships, [PageRank](https://whatis.techtarget.com/definition/PageRank) and [shortest path.](https://searchnetworking.techtarget.com/definition/Shortest-path-bridging)

**Advantages of Database :**

#### Reducing Data Redundancy

The file based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which lead to data redundancy.

This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

#### Sharing of Data

In a database, the users of the database can share the data among themselves. There are various levels of authorisation to access the data, and consequently the data can only be shared based on the correct authorisation protocols being followed.

Many remote users can also access the database simultaneously and share the data between themselves.

#### Data Integrity

Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data that is visible to multiple users. So it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

#### Data Security

Data Security is vital concept in a database. Only authorised users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorised users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

#### Privacy

The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

#### Backup and Recovery

Database Management System automatically takes care of backup and recovery. The users don't need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

#### Data Consistency

Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

**Definition of Entity in DBMS :**

## Entity

An entity is any object in the system that we want to model and store information about. Entities are usually recognizable concepts, either concrete or abstract, such as person, places, things, or events which have relevance to the database. Some specific examples of entities are Employee, Student, Lecturer. An entity is analogous to a table in the relational model.

**Types of entities in DBMS :**

**Strong Entity Type –** are the entities which has a key attribute in its attribute list or a set that has a primary key. The strong entity type is also called regular entity type

**Recursive Entity Type –**It is also called Self Referential Relationship Entity Type. It is an entity type with foreign key referencing to same table or itself. Recursive Entity Type occurs in a unary relationship.

**Weak Entity Type** – Entity Type with no key or Primary Key are called weak entity Type.  
The Tuples of weak entity type may not be possible to differentiate using one attribute of weak entity.For every weak entity, there should be unique OWNER entity type.

**Composite Entities –**  
If a Many to Many relationship exist then we must eliminate it by using composite entities. Composite Entities are the entities which exists as a relationship as well as an entity. The many to many relationship will be converted to 1 to many relationship.  
Composite Entities are also called Bridge Entities, because they acts like a bridge between the two entities which have many to many relationship.  
Bridge or Composite entity composed of the primary keys of each of the entities to be connected.  
A composite entity is represented by a diamond shape with in a rectangle in an ER Diagram.

**Supertypes and Subtypes Entities –**

A subtype is a subprouping of the entities in an entity type that is meaningful to the organisation. For example, In a University, a STUDENT is an entity type. Two subtypes of STUDENT entity are

* GRADUATE STUDENT
* UNDERGRADUATE STUDENT

**Definition of Attribute in DBMS :**

In a database management system (DBMS), an attribute may describe a component of the database, such as a table or a [field](https://searchoracle.techtarget.com/definition/field), or may be used itself as another term for a field.

**Types of attributes in DBMS :**

**Single valued Attributes :** An attribute, that has a single value for a particular entity. For example, age of a employee entity.

**Multi valued Attributes :**An attributes that may have multiple values for the same entity. For example colors of a car entity.

**Compound /Composite Attribute :** Attribute can be subdivided into two or more other Attribute. For Example, Name can be divided into First name, Middle name and Last name.

**Simple/Atomic Attributes :**The attributes which cannot be divided into smaller subparts are called simple or atomic attributes. For example, age of employee entity

**Stored Attribute :**An attribute, which cannot be derived from other attribute, is known as stored attribute. For example, BirthDate of employee.

**Derived Attribute :** Attributes derived from other stored attribute. For example age from Date of Birth and Today’s date.

**Complex Attributes :** If an attribute of an entity, is built using composite and multivalued attributes, then these attributes are called complex attributes. For example, a person can have more than one residence and each residence can have multiple phones, an addressphone for a person entity can be specified as – {Addressphone (phone {(Area Code, Phone Number)}, Address(Sector Address (Sector Number,House Number), City, State, Pin))}

Here {} are used to enclose multivalued attributes and () are used to enclose composite attributes with comma separating individual attributes.

**Key Attribute :**represents primary key. (main characteristics of an entity). It is an attribute, that has distinct value for each entity/element in an entity set. For example, Roll number in a Student Entity Type.

**Non Key Attributes :**These are attributes other than candidate key attributes in a table. For example Firstname is a non key attribute as it does not represent the main characteristics of the entity.

**Required Attribute :**A required attribute is an attribute that must have a data value. These attributes are required because they describe what is important in the entity. For example, In a STUDENT entity, firstname and lastname is a required attribute.

**Optional Attribute/Null Value Attribute –** An optional attribute may not have a value in it and can be left blank. For example, In a STUDENT entity, Middlename or email address is an optional attribute. as some students may not have middlename or email address.

**Definition of Relationship in DBMS :**

A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

**Types of Relationships in DBMS :**

* **One-to-one:** Both tables can have only one record on either side of the relationship. Each primary key value relates to only one (or no) record in the related table. They're like spouses—you may or may not be married, but if you are, both you and your spouse have only one spouse. Most one-to-one relationships are forced by business rules and don't flow naturally from the data. In the absence of such a rule, you can usually combine both tables into one table without breaking any normalization rules.
* **One-to-many:** The primary key table contains only one record that relates to none, one, or many records in the related table. This relationship is similar to the one between you and a parent. You have only one mother, but your mother may have several children.
* **Many-to-many:** Each record in both tables can relate to any number of records (or no records) in the other table. For instance, if you have several siblings, so do your siblings (have many siblings). Many-to-many relationships require a third table, known as an associate or linking table, because relational systems can't directly accommodate the relationship.

**Definition of ERD :**

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an ERD are:

* Entities
* Relationships
* Attributes

Steps involved in creating an ERD include:

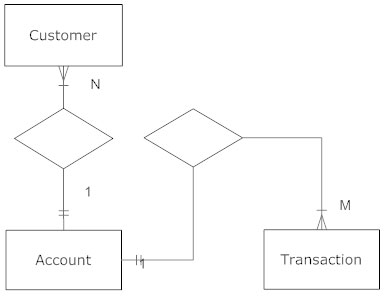
1. Identifying and defining the entities
2. Determining all interactions between the entities
3. Analyzing the nature of interactions/determining the cardinality of the relationships
4. Creating the ERD

**Definition of Schema in DBMS :**

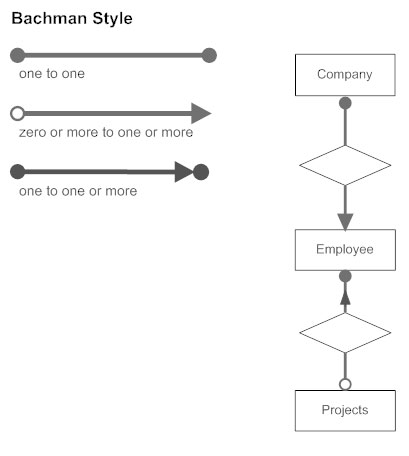
A database schema is a visual and logical architecture of a database created on a database management system.

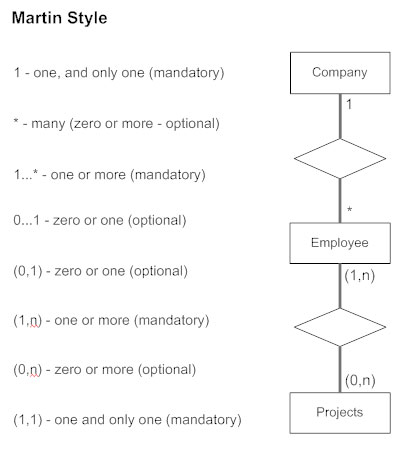
It provides a graphical view of the entire database architecture and structure. It provides a means for logically grouping and displaying database objects such as tables, fields, functions and relations.

**Symbols of ERD :**

* **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.  A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone. 
* **Actions**, which are represented by diamond shapes, show how two entities share information in the database.  In some cases, entities can be self-linked. For example, employees can supervise other employees.  
  
* **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.  
   A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.  A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary. 
* **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.
* **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.  
  

There are many notation styles that express cardinality.   
  
 





**Project’s SCHEMA and ERD :**

