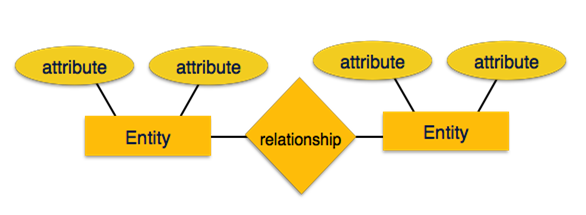
DBMS - DATA MODELS

Data models define how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system.

The very first data model could be flat data-models, where all the data used are to be kept in the same plane. Earlier data models were not so scientific, hence they were prone to introduce lots of duplication and update anomalies.

Entity Relationship Model

Entity-Relationship *ER* Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

ER Model is best used for the conceptual design of a database. ER Model is based on

**Entities** and their *attributes.*

**Relationships** among entities.

These concepts are explained below.

**Entity** − An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.

**Relationship** − The logical association among entities is called ***relationship***. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

Mapping cardinalities −

* one to one
* one to many
* many to one
* many to many

**Relational Model**

The most popular data model in DBMS is the Relational Model. It is more scientific a model than others. This model is based on first-order predicate logic and defines a table as an **n-ary relation**.

The main highlights of this model are −

Data is stored in tables called **relations**.

Relations can be normalized.

In normalized relations, values saved are atomic values.

Each row in a relation contains a unique value.

Each column in a relation contains values from a same domain.

**What are CODD rules?**  
In 1969 Dr. E. F. Codd laid down some 12 rules which a DBMS should adhere in order to get the logo of a true RDBMS.  
**Rule 1:**Information Rule- “All information in a relational data base is represented explicitly at the logical level and in exactly one way – by values in tables.”  
**Rule 2:**Guaranteed access Rule- “Each and every datum (atomic value) in a relational data base is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.”In flat files we have to parse and know exact location of field values. But if a DBMS is truly RDBMS you can access the value by specifying the table name, field name, for instance Customers.Fields [‘Customer Name’].  
**Rule 3:** Systematic treatment of null values- “Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type.”.  
**Rule 4:** **Dynamic on-line catalog based on the relational model**- “The data base description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.”The Data Dictionary is held within the RDBMS, thus there is no-need for off-line volumes to tell you the structure of the database.  
**Rule 5:** Comprehensive data sub-language Rule- “A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all the following items  
Data Definition  
View Definition  
Data Manipulation (Interactive and by program).  
Integrity Constraints  
Authorization.  
Transaction boundaries ( Begin , commit and rollback)  
**Rule 6:** View updating Rule- “All views that are theoretically updatable are also updatable by the system.”  
**Rule 7:** High-level insert, update and delete- “The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data.”  
**Rule 8:** Physical data independence- “Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.”  
**Rule 9:** Logical data independence- “Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit un-impairment are made to the base tables.”  
**Rule 10:** Integrity independence- “Integrity constraints specific to a particular relational data base must be definable in the relational data sub-language and storable in the catalog, not in the application programs.”  
**Rule 11:** Distribution independence- “A relational DBMS has distribution independence.”  
**Rule 12:** Non-subversion Rule- “If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity Rules and constraints expressed in the higher level relational language (multiple-records-at-a-time).”