

**Definition. Data Model** *Notion of describing data or information*

1. **Structure of data**
2. **Operation on data**
3. **Constraint on data**

*important data models*

1. **relational model (SQL)** *large, efficient*
2. **semistructured-data model (XML)** *more flexible*

**Definition. Relational model in brief**

1. **Structure of data** *structure of relational model may resemble an array of structs in C. the table column headers are field names, and each of the row represent the values of one struct in the array*
2. **Operation on data** *relational algebra*
3. **Constraint on data**

**Definition. Basics of the Relational Model**

1. **Relation** *A two-dimensional table is called a relation*
  - (a) **arity** *of a relation is the number of attributes*
  - (b) **cardinality** *of a relation is the number of tuples*
2. **Attributes** *columns of a relation are named by attributes*
3. **Schemas** *The name of a relation and the set of attributes for a relation is the schema for that relation*

*Movies(title, year, length, genre)*

*By virtue of set, order of columns of relation are immaterial*

4. **Database** *one or more relations*
5. **Database Schema** *the set of schemas for relations of a database*
6. **Tuples** *The row of a relation (other than the header row containing the attribute names) are called tuples.*

*(Inception, 2008, 231, thriller)*

7. **Component** *A tuple has one component for each attribute of the relation (ordered by order of attributes). Components are atomic (elementary types only)*

8. **Domains** (data type) the elementary-type associated with each attribute of a relation is a domain

*Movies(title : string, year : integer, length : integer, genre : string)*

9. **Relation instance** A set of tuples for a given relation is an instance of that relation

10. **Current instance** the set of tuples for a given relation now

*Note.* In relational database theory, a **relation**, is a set of tuples  $(d_1, d_2, \dots, d_n)$ , where each element  $d_j$  is a member of  $D_j$ , a data domain. There is no ordering to the elements of the tuples of a relation. Instead, each element is termed an **attribute value**. An **attribute** is a name paired with a domain (data type). An **attribute value** is an attribute name paired with an element of that attribute's domain, and a **tuple** is a set of attribute values in which no two distinct elements have the same name. Thus, in some accounts, a tuple is described as a function, mapping names to values.

**Definition. Keys of Relations**

1. **Key constraint** A set of attributes forms a key for a relation if no two tuples in a relation instance have the same values in all the attributes of the key

*Moves(title, year, length, genre)*

**Definition. Keys**

1. **Superkey** a set of one or more attributes whose combined values are unique (implies no two tuples have same values on all of these attributes)

(a) Does every relation have a superkey? Yes, the entire set of attributes form the superkey

(b) superkey is not minimal.

2. **Key** is a minimal superkey.

(a) usually underline the key in schemas

(b) superkey is the superset (not necessarily proper) of keys

(c) often invent attributes to ensure all tuples will be unique (SIN, ISBN)

(d) defines a kind of integrity constraint

3. **Foreign Key** an attributes that refers to an attribute that is a key in another table

(a) a foreign key may need to have several attributes

(b) **declare foreign key**

$$R[A]$$

where  $R$  is a relation and  $A$  is a list of attributes in  $R$ .  $R[A]$  is a set of all tuples from  $R$  but with only attributes in list  $A$

(c) **declare foreign key constraint**

$$R_1[X] \subseteq R_2[Y]$$

where  $X$  and  $Y$  are lists of attributes of same arity and  $Y$  must be a key in  $R_2$

(d) **Referential integrity constraints** constraint of the form  $R_1[X] \subseteq R_2[Y]$ .  
Note foreign key constraint are a kind of referential integrity constraint or **inclusion dependency** Not all integrity constraints are foreign key constraints ( $A_2$  not a key in  $R_2$ )

4. **Primary key(SQL)** SQL specific