

Data Model

- Notation to describe data
- Three parts:
 - Structure of the data – Higher level than data structures
 - Operations on the data – queries, modifications
 - Constraints on the data – limitations on the data

Popular data models

- Relational model
- Semistructured data model (Last two weeks of the quarter)

Relational Model

Based on tables or relations

Movies Relation

Title	Year	Length	Genre
Gone With The Wind	1939	231	Drama
Star Wars	1977	124	sciFi
Wayne's World	1992	95	comedy

Semi-structured data model - XML

```
<Movies>
  <Movie title = "Gone With the Wind">
    <Year>1939 </Year>
    <Length>231 </Length>
    <Genre>drama</Genre>
  </Movie>
  <Movie title = "Star Wars">
    <Year>197 7 </Year>
    <Length>124 </Length>
    <Genre>sciFi</Genre>
  </Movie>
  <Movie title = "Wayne's World">
    <Year>1992 </Year>
    <Length>95</Length>
    <Genre> comedy </Genre>
  </Movie>
</Movies>
```

Comparing the two models

Model	Flexible	Ease of use	Operations
Semistructured	X		
Relational		X	X

Relational Model Terminology

- Data represented in the form of a table, we hereby call **Relation**
 - Movies
- Columns of a relation are **Attributes**
 - title, year, length, genre
- Name of a relation and the set of attributes for the relation is called **Schema**
 - **Movies (title, year, length, genre)**

Title	Year	Length	Genre
Gone With The Wind	1939	231	Drama
Star Wars	1977	124	sciFi
Wayne's World	1992	95	comedy

Relational Model Terminology

Cont..

- Rows of a relation are called **Tuples**
 - (Gone With the Wind, 1939, 231, drama)
- Each component of a tuple has to be atomic. The type of the element is called the **Domain**
 - Movies(title:string, year : integer , length : integer , genre : string)
- Relations are sets of tuples. Hence **order is irrelevant**.
- A set of tuples for a given relation is an **Instance** of that relation.
- A set of attributes forms a **key** for a relation. No two tuples can have the same values in all the attributes of the key.
 - Movies (**title**, **year**, length, genre)

```
Movies(  
    title:string,  
    year:integer,  
    length:integer,  
    genre:string,  
    studioName:string,  
    producerC#:integer  
)  
MovieStar(  
    name:string,  
    address:string,  
    gender:char,  
    birthdate:date  
)  
StarsIn(  
    movieTitle:string,  
    movieYear:integer,  
    starName:string  
)  
MovieExec(  
    name:string,  
    address:string,  
    cert#:integer,  
    netWorth:integer  
)  
Studio(  
    name:string,  
    address:string,  
    presC#:integer  
)
```