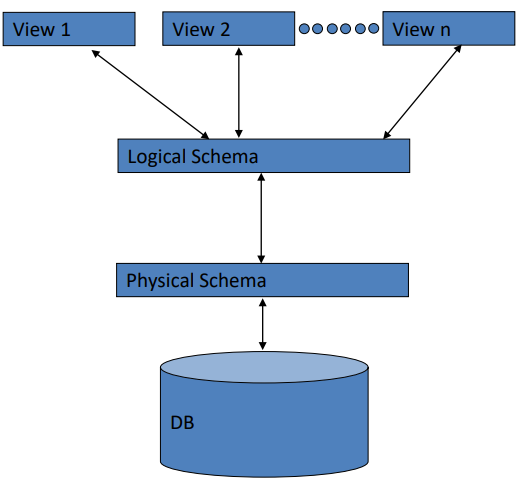
## \*Data Design Processes\*

1. Produce a conceptual or semantic model
2. Translate them into Relational schema
3. Evaluate the schema for quality
4. Constantly improve as needed

## \*Level of Abstraction in BDMS\*



## \*Roles with DBMS\*

### \*End User\*

* People who do something that advances the organization’s purpose
  + Off-line users who receive **reports**
  + Parametric users who execute **pre-written applications**
  + Ad hoc user who **explore data**

### \*Application Programmer\*

* IT professionals who produce the application that end users run
  + Need understanding on how to create an application with access to data on DBMS

### \*DB Admin\* \*DBA\*

* Responsible for management of effective and efficient use of resources in providing access to data
  + Design logical/physical schema
  + Handle security and authorization
  + Data availability, crash recovery
  + Database tuning

### \*DBMS Vendor Staff\*

* DBMS software staff who help DBMS interactions from customer’s perspective
  + Operational support staff
  + Sales support staff
  + Training staff
  + DBMS implementors
  + Tools Development

## \*Database can exist without a DBMS\*

* Data will be stored in a file and directly accessed by programs

## \*Files vs DBMS\*

**DBMS adv**

* Data definition occurs **once**
* Declarative Queries
* Data Integrity
* Single point of data
* Security management
* Concurrent access
* System crash recovery
* Quick application development
* Absence of Data redundancy

\*Database Design Sequence\*

1. Requirement analysis
   * What to store
   * What application to build
   * What operations to optimize
2. Conceptual design
   * Develop high level description of data about how user think of the data
3. Logical Design
   * Conceptual design > database schema
4. Physical Design
   * Logical Design > physical schema

\*Null values\*

* Allows better Aggregate methods
  + Instead of confusion due to -1
* Causes complications in the definition of many operations

\*Data Base Security\*

* Security
  + Restricting accessing **without** permission
* Integrity
  + Restricting modification **without** permission
* Availability
  + Allowing modification and access to those **with** access

\*Database Access Control\*

* Mandatory Access Control (**Authentication**)
  + **User Profile/log in**
* Discretionary Access Control (**Authorization**)
  + **Access Rights or privileges**

\*Relational Algebra\*

1. Set Operations

|  |  |
| --- | --- |
| Selection | Select a subset of rows from relation  **WHERE** |
| Projection | Deletes unwanted columns from relation  **SELECT** |

1. SQL declarative statements

|  |  |
| --- | --- |
| Union | Tuples in relation 1 OR relation 2 |
| Intersection | Tuples in relation 1 AND relation 2 |
| Difference | Tuples in relation 1 NOT relation 2 |

1. Join Operations

|  |  |
| --- | --- |
| Cross-Product  X | Allows us to fully combine two relations |
| Join  |⪥|(attribute 1 = attribute 2) | Combine matching tuples |
|  | |

1. Rename

|  |  |
| --- | --- |
| Rename ρnew name | Allows renaming |
|  | |

1. Relational Division

|  |  |
| --- | --- |
| R/S | R is the big set  S is the subset we want to satisfy |

**\*Data Modeling\***

* Conceptual modelling
  + Focuses on abstract representation of the key entity types in the business domain of interest, their attributes and their iner-relationships
  + ERD
* Logical Modeling
  + More detailed description of each attribute, relationships and key values
  + Functional dependencies, Schema
* Physical Domain
  + Detail of physical storage of the data including indexes