**\*Data\***

* Fact/information that can be stored and recorded

**\*Data Base\***

* Collection of data that contains all the information to operate a function

**\*Data Base Management System\***

* Software Package designed to store and manage Data Base(s)

**\*System Catalogue\* \*Data Dictionary\***

* Stores metadata about the Database
  + Format
  + Data type

**\*Relational DBMS\***

* Linked DBMS via special key to normalize database

**\*Instance\***

* Content of Database
* Restricted by **schema structuring.**

**\*Schema\***

* Description of the **Structure** of the data in Database
  + What tables exist
  + Attributes, types, format

**\*Integrity Constraint\* \*Data Integrity\***

* Restriction of possible instance/values that goes against logical structure/schema
* Constantly checked when data is manipulated

**\*Data Design\***

* **Decision of schema** that will be used for the data
* **Decisions on what information** is necessary and in what **format**

**\*Data Definition Language\* \*DDL\***

* Used to define Schema.
* Allows one to tell DBMS what tables exist and what structure they have.

**\*Data Control Language\* \*DCL\***

* Commands that control a database
  + Administration
  + privileges
  + users

**\*Data Manipulation Language\* \*DML\***

* Used to Access Data
* Commands to:
  + Query
  + Update
  + Insert
  + Delete

\***SQL\***

* Combination of DDL and DML that is used to run commands on BDMS

\***Declarative Command\***

* Demands “what is needed” rather than “how to retrieve”

**\*View\***

* Describes how a user sees the data

**\*Logical Schema\***

* Defines the **structure of data** as it is shared among all users

**\*Physical Schema\***

* Describes the **files and indexes** used for **storage on disk**

**\*Data Independence\***

* **Applications** not affected by changes in **data structure**

**\*Logical data independence\***

* Protection from changes in **Logical Schema**
  + Adding extra column

**\*Physical Data Independence\***

* Protection from changes in **physical structure and location**

**\*Entity\***

* Person, place, object about which you want to gather and store data

**\*Entity Type\* \*Entity Set\***

* Collection of entities that share common properties or characteristics
* Described by a set of Attributes
  + Students, courses, accounts

**\*Attribute\***

* Description of one aspect of an entity type

**\*Domain\***

* Acceptable values within Data Integrity constraints

**\*Key\***

* **Minimal set of attributes** that uniquely identify an **entity** in a set

**\*Entity Schema\***

* Schema for entities

**\*Data Model\***

* Collection of concepts for describing data
  + Structure of data
  + Operations on the data
  + Constraints on the data

**\*Relation\***

* Named, two-dimensional table of data
* Must be at least 1NF
* Stems from mathematical concept of “sets”

**\*Relation Schema\***

* Specifies name of relation as well as it’s attributes and data types

**\*Relation instance\***

* Set of tuples for a schema (table)

**\*Relational Algebra\***

* Defines some basic operators that can express declarative commands

**\*Functional Dependency\***

* When the value of one attribute determines the value of another attribute

**\*Data Minimalism\***

* The theory of storing as little as possible to make sure hackers wouldn’t have access to information even if they break in

**\*Presentation Logic\***

* how data should be presented
  + GUI

**\*Processing Logic\***

* How data should be processed
  + Procedures, functions and programs

**\*Data Management\***

* How data should be managed
  + DBMS

**\*Interactive SQL\***

* SQL Statement input from terminal

**\*Non-Interactive SQL\***

* SQL statements are included in an application program written in a host language