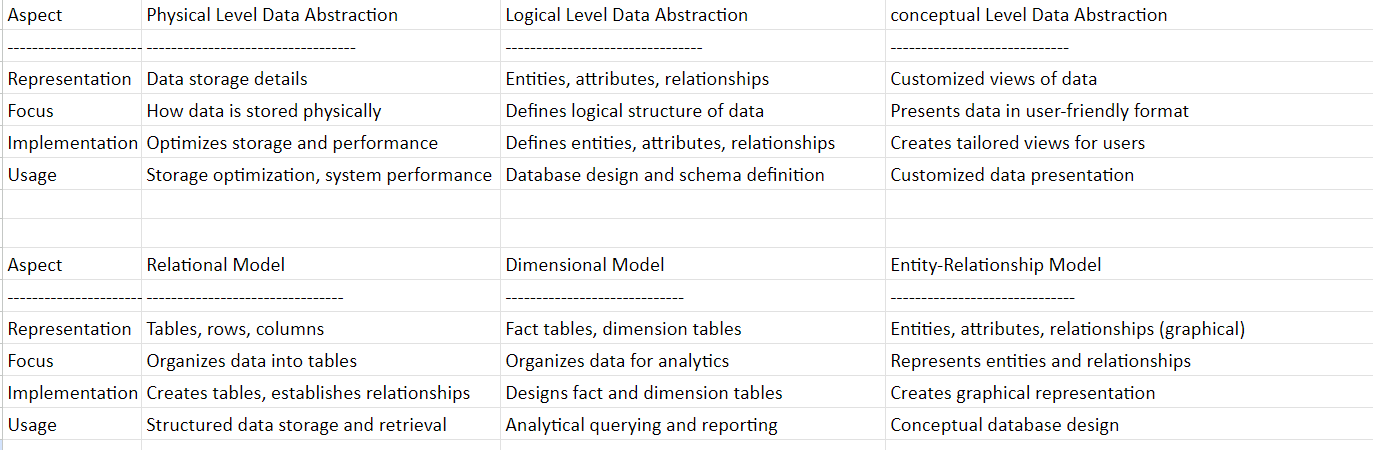
# **WEEK 2: CONCEPTS OF DATA MODELING:**

* What is data modelling ?
  + Organizing data into some form or structure, say in the table format to store it in the datawarehouse using different schemas.
  + Basically we ( abstract the data) have to decide what info is useful for the further designing of the model and how can we efficiently organize it in such a manner that it abides all the rules of the business as well as government’s.
  + Here we scale and normalize the data.
* Why we use data modelling?
  + So that the working of the queries and is efficient and less time consuming , resulting in the reducing cost and easy maintenance for the company’s or business data on the system.
* So in DATA ABSTRACTION what we do is design the basic structure for the implementation in that the first we conceptually decide the design and not the details ..
* Say for example we have different legos we know that but here we only decide the outer structure ….to decide the colors of wall of building or say roof and other details we use data models ….depending on the requirement of the business.
* So basically conceptual data model is the starting point for designing the structure for everything.
* There are 3 levels of data abstraction:
  + 1) Conceptual:
    - Where the information is collected and then the table names and there relationship is decided.
    - That is, it just forms the blueprint for the finalized structure that is to be stored.
    - Say for example the work of an architecture …as it makes the blueprints i.e a drawing of the building (rough drawing ) without any details .
    - Similarly the conceptual modelling designs the outer structure of the schemas.
    - So here we have to decide which all entities are going to be included in the structure.
  + 2)Logical:
    - Here in this step we decide the attributes of the each table that are to be included and are essential.
  + 3) Physical:
    - Which type of storage to be used ,etc
    - Eg: we can use cloud storage for our ease , no maintenance and no paying for the extra storage ,etc
* There are 3 types of data models:
  + Relational:
    - this is the older version where there are measures and dimensions.
    - Measures stores only numerical value on which the mathematical functions can be applied whereas dimensions store the descriptions as well as numeric values.
    - It also states the relation of the tables, tuples, and attributes.
    - Here the main focus is on normalizing the data for the business and the structure of the data.
    - Relational model has data integrity and referential integrity i.e the tables are explicitly related (through foreign keys.)
    - For instance , SQL is relational model.
  + Dimensional:
    - here the data is divided into fact and dimension tables, where fact is the main table which stores only numerical values and dimension can have both numerical and descriptions.
    - All the dimension tables are connected to the one or more main **fact** table.
    - And further dimension table can have there dimension tables which from the **star** and **snowflake** structure.
    - Here the primary key and foreign key’s are decided, on the basis of which the tables will be connected to each other.
    - Here the focus is on the data analytics that how the two tables i.e. facts and dimensions can be used for the data analyzing.
    - For instance tableau is used for dimensional model.
  + Entity- relationship:
    - Here the data is graphically represented and visualized.
    - To analyze the data graphically using the relations between the tables before the implementation.



* Types of keys:
  + Keys are used for the structured data(through which you can get the insights abt data in mathematical terms or say numerically whereas the unstructured data is in the text or paragraph format for example survey with categorical values ) where it has has to be stored, managed, retrieved and maintained.
  + Primary key:
    - Here the table has a column or combination of columns which uniquely identity the each row of the table .
    - A table can have composite key but only one primary key.
  + Composite key:
    - Here we take the 2 columns which can uniquely identify the tuple(row).
    - How it is different from super key then?
    - It can take both the columns as non unique identifier columns .
    - Just there combination must be unique .
    - Say for example , student first name and last name.
  + Super key:
    - It is the possible combinations of each column through which we can identify the row uniquely.
    - Redundant attributes.
    - By adding atleast one or no attribute to the candidate key , it becomes the super key. Eg :
      * + student\_id
        + name
        + age
        + aadhar
        + email
        + voterid
    - so here voterid, emai, adhar, stu\_id are 4 candidate keys, and also a super key.
    - But name and age are not super key because they cannot uniquely identify any tuple.
    - by adding the name to any of it the candidate key or say or say both the whole set those keys becomes the super key.

* + - For example, we have
      * student\_id as primary key
      * name
      * aadhar no
        + so here what will happen is by the combination of **aadhar and id** we can search for a row
        + similary other combinations like

**Aadhar and name**

**Id and aadhar**

**Aadhar , id and name**

**Only aadhar**

**Only id**

* + Candidate key:
    - aka alternate key.
    - Taking the above example ,
      * Where there is no redundancy in the col name we call it as candidate key (uniquely identified)
      * So here student\_id and aadhar are unique for each student so they are said to be candidate key.
      * **Minimal super key** with no redundant attributes.
      * So from many candidate keys we select the most appropriate key and make it as a primary key.
  + Surrogate key:
    - They are the keys that don’t have any meaning unlike primary keys or say natural keys.
      * + aka technical key / fact-less key / synthetic key
    - They are auto increasing numbers .
    - They are there just to make the integration of the data easier to maintain , retrieve and to insert the data .
    - For example say there is a chain restaurant and they have different outlets around the world.
    - So there must be a central system where data has to be accumulated to analyze the performance of the stores .
    - So , suppose india stores the data in the or say primary key as r309 and Japan stores it as 509, so what we do is will use the surrogate key so the integration of these data of both the countries becomes easier.
    - And what if we have the 2 outlets in the same city ?
      * So the name of the store , and city name will be redundant, and will make the retrieval of the data complex
      * So we use surrogate key which will number them , and through which we can easily identify if we have to analyze or visualize the performance of each outlet that which store is performing well .
* **chunk dimension**
* **snowflake schema**
* **star schema**
* and few other links for the online courses

surrogate key for joining the tables having string as dtype.

Discussed the surrogate key and where and why and why we don’t use pk and fk

SCD Type2 – slowly changing the admission type 2

Data warehouse toolkit by ralph kimbal

Next week:

etl