Introduction to Data Modeling

# Normalization

* Normalization is the process of efficiently organizing data in a database
* Two Goals of Normalization
  + Eliminate Redundant data
  + Ensure data dependencies make sense
* Advantages of Normalization
  + Achieve a highly flexible design
  + Reduces data redundancy & data anomalies
* Normal Forms are guidelines for normalizing databases
* Types of Normal Forms
  + First Normal Form
  + Second Normal Form
  + Third Normal Form
  + Fourth Normal Form
  + Fifth Normal Form
  + BC Normal Form
* A de-normalized table looks as below…

|  |  |  |
| --- | --- | --- |
| Student Details | Course Details | Result details |
| 1001   Ram               11/09/1986 | M4       Basic Maths                       7 | 11/11/2004       89           A |
| 1002   Shyam           12/08/1987 | M4       Basic Maths                       7 | 11/11/2004       78           B |
| 1001   Ram               23/06/1987 | H6                                                    4 | 11/11/2004       87           A |
| 1003   Sita                16/07/1985 | C3        Basic Chemistry                 11 | 11/11/2004       90           A |
| 1004   Gita               24/09/1988 | B3                                                     8 | 11/11/2004       78           B |
| 1002   Shyam           23/06/1988 | P3        Basic Physics                     13 | 11/11/2004       67           C |
| 1005   Sunita           14/09/1987 | P3        Basic Physics                      13 | 11/11/2004       78           B |
| 1003   Sita                23/10/1987 | B4                                                      5 | 11/11/2004       67           C |
| 1005   Sunita           13/03/1990 | H6                                                     4 | 11/11/2004       56           D |
| 1004   Gita               21/08/1987 | M4      Basic Maths                         7 | 11/11/2004       78           B |

## Key Definitions Insert Anomaly

* In the above table, a student who is not registered to any course or a course that that does not have any students cannot be inserted

### Determinant

* Attribute x can be a determinant if it defines Attribute Y in a given relationship or entity.
* Determinants are represented as X -> Y i.e. attribute X decides attribute Y
* To be a determinant, attributes need not be a key attribute
* For e.g. Marks decide Grade. So Marks is a determinant although it is not a key attribute

### Functional Dependency

## First Normal Form

* A Relation is in First Normal Form if every attribute is single valued for each row/tuple
* Steps to convert to 1NF
  + Eliminate repeating groups in a table
  + Create a separate table for each set of related data
  + Have a Primary key for each table

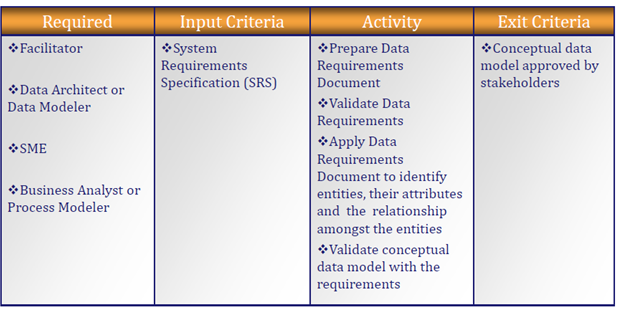
# Requirements Analysis

Goal of Requirement Analysis is to collect information from Users of data

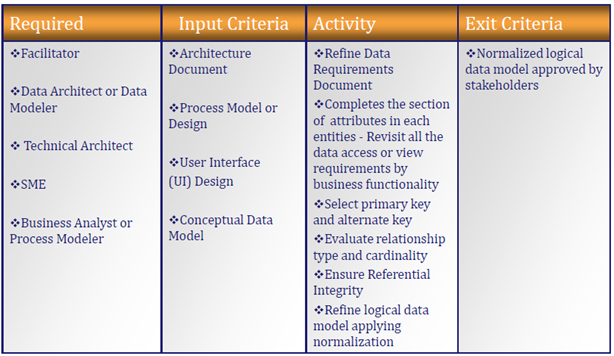
## Requirement Analysis phase Steps

* Determine the data requirements of the database in terms of primitive objects.
  + Steps in Data Requirement Collection…
    - Reviewing existing documents
    - Interview End Users
    - Reviewing existing automated systems
  + Below are the points to be kept in mind while collecting data requirements…
    - Talk to the end-users about the **data in** **real world terms**. Users do not think the data in terms of entities, attributes etc., but about the actual people, things, and activities they deal with daily
    - Take the time to **list down the basics** of the organization and its activities
    - Talk to **larger number of Users** to get inputs. Users view data in different ways according to their function/department in the organization
* Classify and describe the information about these objects
* Identify and Classify the relationships between those objects
* Determine the types of transactions to be executed on the database and the interactions between the data and the transactions
* Identify the rules that govern the data integrity
* ER Diagram is built based on the above information and reviewed by end users. The entire cycle is repeated till ER diagram is signed off by end user
* Following things are required to be documented…
  + Information Collected
  + Data Objects that are identified and Classified (as entities, relationships, attributes etc.)
  + Names that are familiar to end users

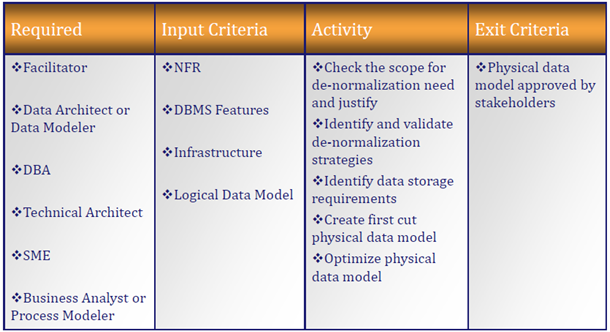
## Conceptual Data Modeling Process



## Logical Data Modeling process



## Physical Data Modeling Process



## Abbreviations in Requirement Analysis Phase

### System Requirement Specifications (SRS)

* Complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. It can also include NFR
* A Typical SRS may contain…
  + Introduction
    - Purpose
    - Definitions
    - System overview
    - References
  + Overall description
    - Product perspective
      * System Interfaces
      * User Interfaces
      * Hardware interfaces
      * Software interfaces
      * Communication Interfaces
      * Memory Constraints
      * Operations
      * Site Adaptation Requirements
  + Product functions
  + User characteristics
  + Constraints, assumptions and dependencies
  + Specific requirements
    - External interface requirements
    - Functional requirements
    - Performance requirements
  + Design constraints
    - Standards Compliance
  + Logical database requirement
  + Software System attributes
    - Reliability
    - Availability
    - Security
    - Maintainability
    - Portability
    - Other requirements

### Non-Functional Requirements (NFR)

* Non-functional requirements define the overall qualities of the resulting system by placing restrictions on the product being developed & the development process, and by specifying external constraints that the product must meet
* Examples of NFR include safety, security, usability, reliability, performance requirements, Project management issues (costs, time, schedule)

# Key Definitions…

### Entity

* ***Entities are principal data object for storage***
* In Database terms, An entity can be equated to Tables
* Types of Entities
* Independent/Strong Entity – Does not rely on another entity for identification
* Dependent/Weak Entity
  + Entities that cannot be uniquely identified by its attributes alone are Weak entities
  + They need a Foreign Key along with its attributes to create a Primary key
* Associative/Resolving Entity
  + Associates two or more entities to reconcile many-to-many relationships
  + For e.g. Student and Course are two entities. Student can subscribe to many courses and A Course can have many students. To create a many-to-many relationship between Student and Course, any one of the entity must be a parent. In this case neither is a Parent, an associative entity is created to resolve this conflict.
  + **Must Haves** for Associate Entity
    - All Relationships must be many-to-many
    - Should not have additional surrogate Keys for association
  + An Associative entity may participate in relationships other than the associated relationship
* Subtype Entity – Represent subset of instances of its Parent entity

### Relationship

* ***An Association of entities***
* Relationship Classification
  + Degree
    - Total number of entities participating in the relationship
    - **Types** - Binary, Ternary and N-ary
  + Cardinality
    - Number of related occurrences for each of the two entities
    - **Types** – One, Many
  + Direction
* Relationship Types
  + Identifying Relationships - are those where the child entity is dependent on its parent entity and cannot exist without it

In other words, if primary key of child entity contains all attributes in a foreign key, then child entity is said to be “identifier dependent” on the parent entity and the relationship is called an "identifying relationship”.

* + Non-Identifying Relationships - If any attributes in a foreign key do not belong to the primary key of the child, then child is not identifier dependent on the parent and the relationship is called “non-identifying”
  + Recursive Relationships – In Recursive relationships, an entity is associated with itself. The Parent and Child tables are the same

### Attribute

* ***A Property of an entity that either identify or describe an entity***
* In Database terms, An attribute can be equated to Column Names
* Attributes that identify entities are called Key Attributes
* Attributes that describe entities are called Non-Key Attributes (e.g. First Name, Last Name etc)
* Attribute Types
  + Required Attribute *(Employee Number, First Name)* vs Optional Attribute *(Landline Number)*
  + Simple Attribute *(State, ZipCode etc)* vs Composite Attribute *(Home Address)*
  + Single Value Attribute vs Multi-Valued Attribute *(Employee has more than one skill represented by comma)*
  + Stored Attribute vs Derived Attribute *(Age)*

### Instance

* An entity occurrence is an Instance
* In Database terms, Instances can be equated to Rows

### Domain

* Valid Set of values for an attribute

### Primary Key

* ***An attribute or a set of attributes that uniquely identify a specific instance of an entity***
* **Must Have** attributes of Primary Key
* have a non-null value for each instance of the entity
* be unique for each instance of an entity
* not change or become null during the life of each entity instance
* A Primary Key with more than one attribute is called as **Composite Key**

### Candidate Key

* ***Any key or minimum set of keys that could be a primary key***
* All **Must Have** attributes of Primary key applies to Candidate key
* Candidate Keys not chosen as Primary Key is called as **Alternate Key**

### Artificial/Surrogate Key

* ***Keys that has no meaning to the business or organization***
* Surrogate Keys are allowed only when…
* No attribute in the entity has all the primary key properties
* The primary key is large and complex

### Foreign Key

* ***An attribute that completes a relationship by identifying the parent entity***
* FKs are useful for…
* Maintain integrity in the data (called referential integrity)
* Navigates between different instances of an entity
* Every relationship in the model **must be** supported by a foreign key

# Best Practices

### Rules for Modeling Relationships…

* Entities must participate in relationships
* Many-to-Many Relationships must be resolved - Replace the relationship with an Associate entity & relate the two entities to the associate entity
* Transform Complex Relationships (Ternary, N-ary) to Binary Relationships
* Eliminate Redundant relationships

### Creating an Attribute

* Should be Atomic – Represents a single value for each instance of a primary key
* Should have a Primary Key whose values uniquely identify occurrences of an entity
* Do not split single domains into multiple attributes. For e.g. do not split Gender column into two columns Male and Female and fill True/False for each
* Artificial/Surrogate Keys are allowed only when
  + No attribute possesses all the primary key characteristics
  + Candidate keys are complex and large
* No two entities should have a Primary key with same attributes or set of attributes

### Naming an Attribute

* Should be clear, concise, and self-explanatory.
* Name attributes according to logical considerations, not physical characteristics i.e. Provide the attribute names which has meaning to the business
* Apply minimum number of words necessary to uniquely identify attributes and to convey its meaning
* Avoid using special characters
* Should not replace or contradict the definition of the attribute.
* Should not contain abbreviations or acronyms with exception of established or approved abbreviations.
* Should not contain any of the following…
  + Names of organization, computer or information systems, reports
  + Plural words
  + Articles (a, an, the) like “the annual membership fee”
  + Conjunctions (and, but, or) like “batch or lot number”
  + Verbs like “person owns property”
  + Prepositions (from, at, by, for, of …) like “cost of item”