Conceptual model:

1. A model that is nontechnical and is free of ambiguities: that represents the data objects and relationships of the systems. It is a top-down approach where entities and relationships among them are identified.
2. Entity Type: Group of objects with same properties, identified by enterprise as having an independent existence.
3. Entity occurrence: uniquely identifiable object instance of an entity type.
4. A line represents a binary relationship and connect entity instances.
5. Relationship type: a set of meaningful associations among entity types.
6. Relationship occurrence: uniquely identifiable association, which includes one occurrence from each participating entity type.
7. Semantic net is a detailed object-level model. A dot reps an entity occurrence and a diamond reps a relationship occurrence.
8. Attributes can be classified as:
   1. Simple: composed of single component with an independent existence.
   2. Composite: composed of multiple components, each with an independent existence. Ex: address
   3. Single-valued: holds a single value for each occurrence of an entity type.
   4. Multi-valued: holds multiple values for each occurrence of an entity type.
   5. Derived: it is derivable from value of a related attribute, or set of attributes.
9. Structural constraints:
   1. Multiplicity: no. of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a relationship.
   2. Cardinality: max no. of relationship occurrence for an entity participating in each relationship type.
   3. Participation: determines whether all or only some entity occurrences participate in a relationship, describes minimum no. of possible relationships occurrences.
10. Problems with ER models:
    1. Fan trap: where a model represents a relationship b/w entities, but the pathway b/w certain entity is ambiguous. To resolve, restructure the entities.
    2. Chasm trap: where a model suggests the existence of a relationship b/w entity types, but the pathway does not exist b/w certain entity. Relationships with a min. multiplicity of zero. To resolve, find the missing relationship.

Enhanced ERD:

1. Superclass is an entity type that includes one or more distinct subgroupings of its occurrences.
2. Subclass is a distinct subgrouping of occurrences of an entity type.
3. Specialization is the process of maximizing differences b/w members of an entity by identifying their distinguishing characteristics. Ex: manager, sales personnel are subclasses of a specialized superclass.
4. Generalization is the process of minimizing differences b/w entities by identifying their common characteristics.
5. Constraints for the hierarchy:
   1. Participation: determines whether every member in superclass must participate as a member of subclass. {Mandatory, Optional}
   2. Disjoint: describes the relationship b/w members of the subclasses and indicates whether a member of a superclass can be member of one or more than one subclass.
6. Aggregation represents a “has-a” or “is-part-of” relationship b/w entity types, where one entity represents the whole and the other the part. Represented by diamond sign.
7. Composition is a specific form of aggregation that represents an association b/w entities, where there is a strong ownership aka Parts cannot be shared.

Relational data model:

1. Cartesian product, ́DD12,is the set of all ordered pairs, where first element is member of D1and second element is member of . Properties’:
   1. Order does not matter.
   2. Duplicates aren’t allowed.
2. Candidate key is one or more attributes that must be unique for each tuple. In each tuple of R, the values of K uniquely identify that tuple.
3. Constraints: rules to ensure an accurate representation of the concept.
   1. Domain constraint limits the values that can be stored in a field. Ex : type defined for a field.
   2. NULL constraint allows the creator of a database to specify which attributes may be missing. The value may be missing because it is unknown or is not applicable to the tuple.
   3. Entity integrity each tuple must be unique from the other tuples. A primary key provides the uniqueness. In a base relation, no attribute of a primary key can be NULL
   4. Referential integrity states that if a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its home relation or the foreign key value must be wholly null.
4. Views: Dynamic result of one or more relational operations operating on base relations to produce another relation. Purpose:
   1. Powerful and flexible security mechanism by hiding parts of the db from users.
   2. Customized data to user needs
   3. Simplifies complex operations on the base relations.

Database:

1. Shared collection of logically related data, designed to meet the information needs of an organization.
2. System catalog : repository of information describing the data in the database. Updated by DDL commands.
3. DDL : used to specify the structure of the db schema. The data generated from DDL commands change the system catalog.
4. DML : used to read and update the data.
5. Behavior for FKs:
   1. RESTRICT: if a record being referenced with a FK, the DELETE or UPDATE would fail.
   2. SET NULL: if a constraint allows, it will set the record to null.
   3. CASCADE: if parent record is deleted, child ones are deleted as well.

Commands:

1. CREATE TABLE `clinic` ( `clinicNo` INT PRIMARY KEY NOT NULL AUTO\_INCREMENT, `petNo` INT NOT NULL );
2. ALTER TABLE `clinic` ADD CONSTRAINT `FK\_clinicPetNo` FOREIGN KEY (`petNo`)

REFERENCES `pet` (`petNo`) ON DELETE NO ACTION ON UPDATE NO ACTION;

Conceptual: the process of constructing a model of the data used in an enterprise, independent of all physical conditions.

Logical database design: The process of constructing a model of the data used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations.

Physical database design describes the base relations, file organizations, and indexes and any associated integrity constraints and security measures.

Data Redundancy and Update Anomalies:

1. Updates to the data stored in the database are achieved with a minimal number of operations thus reducing the opportunities for data inconsistencies.
2. Reduction in the file storage space required by the base relations thus minimizing costs.
3. Redundant information can suffer from update anomalies:
   1. Insertion anomaly: inserted tuple contain data fields that are inconsistent.
   2. Deletion: possible loss of information
   3. Modification: modification of one tuple is dependent on the other.

Functional dependency:

1. Functional dependency describes a relationship between attributes. For example, if A and B are attributes of relation R, B is functionally dependent on A (denoted A → B), if each value of A in R is associated with exactly one value of B in R.
2. Transitive dependency describes a condition where A, B, and C are attributes of a relation such that if A → B and B → C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

Normalization:

1. The normalization process is a series of tests that help identify the optimal grouping of attributes to relations. GOAL: reduce data redundancy.
   1. Reduce the need for restructuring the collection of relations.
   2. Make the relational model more informative to users
   3. Make the collection of relations neutral to the query stats.
2. UNF -> no PK or NULL values. Attributes with one or more repeating groups.
3. INF -> Identify the repeating group(s) in the unnormalized table which repeats for the key attribute(s). • Remove the set by creating a separate table for the set or if there is an upper limit to the set you can flatten it into fields
4. 2NF -> A relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key.
5. 3NF -> A relation that is in first and second normal form and in which no non-primary-key attribute is transitively dependent on the primary key.