

# Theory

---

## Database fundamentals

Data Model (Conceptual representation of data structures required by a database):

- Data structures (include data objects and association between them)
- Data objects and association between them
- Independent of hardware and software
- Focuses on what data is required and how it should be organised

Data Model Elements:

- Entity – a unique real world thing
- Entity type
  - Logical category
  - Objects described with same attributes
  - Eg.: Milk, Bread, Chocolate: → Product
- Entity Instance - One specific member of an entity type
- Attribute:
  - Piece of information that describes an entity
  - We describe things with multiple attributes Product{Name, price, instock}
- Relationship:
  - How entities depend on or connect to each other
- Entity Key:
  - A property or a set of properties of an entity that are used to determine identity
  - The properties that make up an entity key are chosen at design time.
  - The values of entity key properties must uniquely identify an entity instance
  - The key value should not be modified

Types Of Relationship:

- On-to-one
- One-to-many
- Many-to-many

Levels Of Data Modeling

- Conceptual data model
  - User's view
  - Independence of physical implementation
  - Identifies entities and the highest level relation between entities
  - No attributes specified

- Logical data model
  - Meets the requirements of the database system
  - HOW the system will be implemented (Specified):
    - All attributes for each entity are specified.
    - The primary key(s) for each entity is specified.
    - Foreign keys (keys identifying the relationship between different entities) are specified.
    - Normalization occurs at this level.
- Physical data mode
  - Implementation within the database system
  - HOW the system will be implemented using a specific DBMS (Specifies):
    - Specifies the tables and the columns.
    - Foreign keys are used to identify relationships between tables.
    - Denormalization may occur based on user requirements.
    - Physical considerations may cause the physical data model to be quite different from the logical data model.

Transforming ER model into relational model:

- Many-to-many relationship transforms into a table
- One-to-many doesn't transform into a table

Normalization:

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

- Includes creating tables and establishing relationships between the tables
- Eliminates redundant data

Database Normalization:

- The inventor of the relational model Edgar Codd proposed the theory of normalization of data with the introduction of the First, Second and Third Normal Forms. Later he joined Raymond F. Boyce to develop the theory of Boyce-Codd Normal Form.
- Normalization forms:
  - First Normal Form (1NF)
    - Eliminate Repeating Groups
  - Second Normal Form (2NF)
    - Eliminate Repeating Data
  - Third Normal Form (3NF)
    - Eliminate Columns Not Dependent on Key
  - Boyce-Codd Normal Form (BCNF)
    - Only key dependency exists

- Fourth Normal Form (4NF) - Eliminate Multi-Valued Dependencies - Fifth Normal Form (5NF) - Eliminate Join Dependencies

## Functional Dependencies

If one set of attributes in a table determines another set of attributes in the table, then the second set of attributes is said to be functionally dependent on the first set of attributes.

## First Normal Form (1NF)

- The relation must have atomic domains.
- Eliminate repeating groups
- Horizontal or vertical extension of the table
- Identify with primary key

## Second Normal Form (2NF)

For a table to be in 2NF, there are two requirements

- The table is in first normal form
- All non-key attributes in the table must be functionally dependent on the entire primary key

Note: Remember that we are dealing with non-key attributes

## 2NF - Decomposition

1. If a data item is fully functionally dependent on only a part of the primary key, move that data item and that part of the primary key to a new table.
2. If other data items are functionally dependent on the same part of the key, place them in the new table also
3. Make the partial primary key copied from the original table the primary key for the new table. Place all items that appear in the repeating group in a new table

## Third Normal Form (3NF)

For a table to be in 3NF, there are two requirements

- The table should be in 2NF
- There can be no interdependencies among non-key attributes

## 3NF - Decomposition

1. Move all interdependent non-key items to a new entity.
2. Identify a primary key for the new entity.
3. Place the primary key for the new entity as a foreign key on the original entity.

## SQL Fundamentals

- Data Definition Language
- Data Manipulation Language
- Data Control Language
- Transaction Control

## Practice

---