**Technical blog on Database modelling and normalization**

**Data modeling** is the process of creating a simplified diagram of a software system and the data elements it contains, using text and symbols to represent the data and how it flows. Data models provide a blueprint for designing a new database or reengineering a legacy application.

It is a visual representation of database concepts and the relationships between them identifying the high-level user view of data. Rather than the details of the database itself, it focuses on establishing entities, characteristics of an entity, and relationships between them

**Types of database models:**

**Relational database model** the most common model, the relational model sorts data into tables, also known as relations, each of which consists of columns and rows. Each column lists an attribute of the entity in question, such as price, zip code, or birth date. Together, the attributes in a relation are called a domain. A particular attribute or combination of attributes is chosen as a primary key that can be referred to in other tables, when it’s called a foreign key.

Each row, also called a tuple, includes data about a specific instance of the entity in question, such as a particular employee.

The model also accounts for the types of relationships between those tables, including one-to-one, one-to-many, and many-to-many relationships

Relational databases are typically written in Structured Query Language (SQL)

**Hierarchical model** the hierarchical model organizes data into a tree-like structure, where each record has a single parent or root. Sibling records are sorted in a particular order. That order is used as the physical order for storing the database. This model is good for describing many real-world relationships

**Network model** the network model builds on the hierarchical model by allowing many-to-many relationships between linked records, implying multiple parent records. Based on mathematical set theory, the model is constructed with sets of related records. Each set consists of one owner or parent record and one or more member or child records. A record can be a member or child in multiple sets, allowing this model to convey complex relationships.

**Object-oriented database model** This model defines a database as a collection of objects, or reusable software **elements**, with associated features and methods. There are several kinds of object-oriented databases:

A multimedia database incorporates media, such as images, that could not be stored in a relational database.

A hypertext database allows any object to link to any other object. It’s useful for organizing lots of disparate data, but it’s not ideal for numerical analysis.

The object-oriented database model is the best known post-relational database model, since it incorporates tables, but isn’t limited to tables. Such models are also known as hybrid database models.

**Object-relational model** This hybrid database model combines the simplicity of the relational model with some of the advanced functionality of the object-oriented database model. In essence, it allows designers to incorporate objects into the familiar table structure.

**Entity-relationship model** This model captures the relationships between real-world entities much like the network model, but it isn’t as directly tied to the physical structure of the database. Instead, it’s often used for designing a database conceptually.

Here, the people, places, and things about which data points are stored are referred to as entities, each of which has certain attributes that together make up their domain. The cardinality, or relationships between entities, are mapped as well.

**Database normalization**

**What is 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.

Database Normal Forms

Here is a list of Normal Forms in SQL:

⦁ 1NF (First Normal Form)

⦁ 2NF (Second Normal Form)

⦁ 3NF (Third Normal Form)

⦁ BCNF (Boyce-Codd Normal Form)

⦁ 4NF (Fourth Normal Form)

⦁ 5NF (Fifth Normal Form)

⦁ 6NF (Sixth Normal Form)

**1NF (First Normal Form) Rules**

⦁ Each table cell should contain a single value.

⦁ Each record needs to be unique.

1NF Example



2**NF (Second Normal Form) Rules**

For a table to be in the Second Normal Form, it must satisfy two conditions:

⦁ The table should be in the First Normal Form.

⦁ There should be no Partial Dependency.

OR

*A relation that is in First Normal Form and every non-primary-key attribute is fully functionally dependent on the primary key, then the relation is in Second Normal Form (2NF)*



**3NF (Third Normal Form) Rules**

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.



