DATA MODELING: CONCEPTUAL, LOGICAL, PHYSICAL DATA MODEL TYPES

Data Modeling

Data modeling is the process of creating a data model for the data to be stored in a database. This data model is a conceptual representation of data objects, the associations between different data objects, and the rules.

Data modeling helps in the visual representation of data and enforces business rules, regulatory compliances, and government policies on the data. Data Models ensure consistency in naming conventions, default values, semantics, and security while ensuring quality of the data.

Data Model

The data model is defined as an abstract model that organizes data description, data semantics, and consistency constraints of data.

The data model emphasizes on what data is needed and how it should be organized instead of what operations will be performed on data.

Data Model is like an architect's building plan, which helps to build conceptual models and set a relationship between data items.

The two types of Data Modeling Techniques are

1. Entity Relationship (E-R) Model
2. UML (Unified Modeling Language)

Why use Data Model?

The primary goals of using data model are:

- Ensures that all data objects required by the database are accurately represented. Omission of data will lead to creation of faulty reports and produce incorrect results.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data Model structure helps to define the relational tables, primary and foreign keys and stored procedures.
- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.

- Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

Types of Data Models

There are mainly three different types of data models: conceptual data models, logical data models, and physical data models, and each one has a specific purpose. The data models are used to represent the data and how it is stored in the database and to set the relationship between data items.

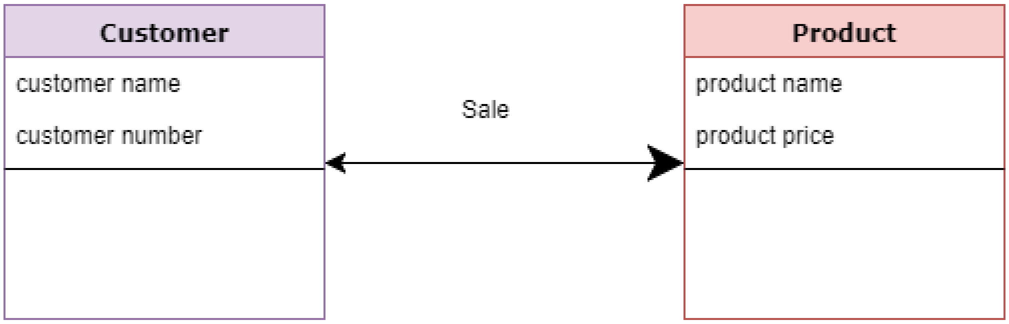
1. **Conceptual Data Model:** This Data Model defines **WHAT** the system contains. This model is typically created by business stakeholders and data architects.
   A conceptual data model usually just includes the main concepts (entities) required to store information and the relationships that exist between these entities. We don’t usually include any details about each piece of information. We can consider the conceptual stage as an initial model, without all the details required to create a database.
   Normally, entities are specified at a high level, using business rather than technical names; this allows the models to be understood by management and users and not just the technical staff.
   The purpose is to organize scope and define business concepts and rules.

The 3 basic tenants of Conceptual Data Model are

- **Entity**: A real-world thing
- **Attribute**: Characteristics or properties of an entity
- **Relationship**: Dependency or association between two entities

Data model example:

- Customer and Product are two entities. Customer number and name are attributes of the Customer entity
- Product name and price are attributes of product entity
- Sale is the relationship between the customer and product



Conceptual Data Model

Characteristics of a conceptual data model

- Offers organization-wide coverage of the business concepts.
- This type of data models are designed and developed for a business audience.
- The conceptual model is developed independently of hardware specifications like data storage capacity, location or software specifications like DBMS vendor and technology. The focus is to represent data as a user will see it in the “real world.”

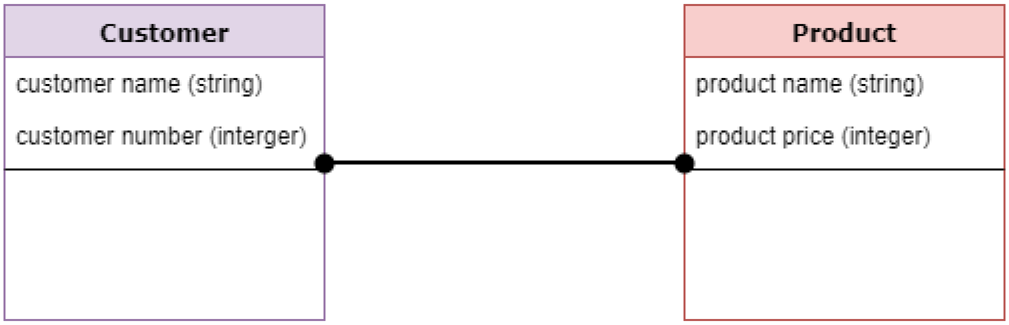
Conceptual data models known as Domain models create a common vocabulary for all stakeholders by establishing basic concepts and scope.

2. **Logical Data Model:** Defines **HOW** the system should be implemented regardless of the DBMS. This model is typically created by data architects and business analysts.
   A logical data model is probably the most-used data model. It goes beyond the conceptual model; it includes entities, relationships, details on entities’ different attributes, and unique ways to identify entities (primary keys) and establish the relationships between them (foreign keys).
   A logical data model should contain all the details to define an information domain. However, it does not consider the technologies (i.e. the database and platform) that will be used to implement the model. It should use business names for entities and attributes, since its purpose is to describe the data structures required, not to create the actual database.

The **Logical Data Model** is used to define the structure of data elements and to set relationships between them. The logical data model adds further information to the conceptual data model elements.

The advantage of using a Logical data model is to provide a foundation to form the base for the Physical model.

However, the modeling structure remains generic.



Logical Data Model

At this Data Modeling level, no primary or secondary key is defined. At this Data modeling level, you need to verify and adjust the connector details that were set earlier for relationships.

Characteristics of a Logical data model

- Describes data needs for a single project but could integrate with other logical data models based on the scope of the project.
- Designed and developed independently from the DBMS.
- Data attributes will have data types with exact precisions and length.
  Normalization processes to the model is applied typically till 3NF.

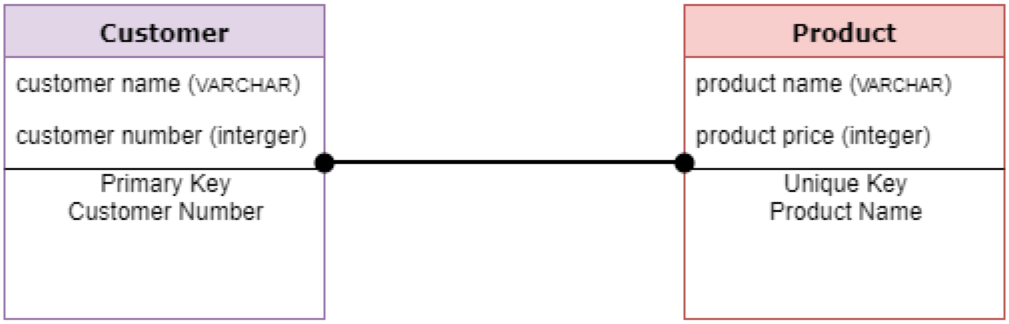
3. **Physical Data Model**: This Data Model describes **HOW** the system will be implemented using a specific DBMS system. This model is typically created by DBA and developers.

   A physical data model is usually derived from a logical data model for a particular relational database management system (RDBMS), thus taking into account all technology-specific details.

One big difference between logical and physical data models is that we now need to use table and column names rather than specifying entity and attribute names. This allows us to adapt to the limits and conventions of the desired database engine. We also provide the actual data types and constraints that allow us to store the desired information.
Physical data models should be easy to transform into a SQL script that allows us to create the database structure.

The purpose is actual implementation of the database.

A **Physical Data Model** describes a database-specific implementation of the data model. It offers database abstraction and helps generate the schema. This is because of the richness of meta-data offered by a physical data model.
The physical data model also helps in visualizing database structure by replicating database column keys, constraints, indexes, triggers, and other RDBMS features.



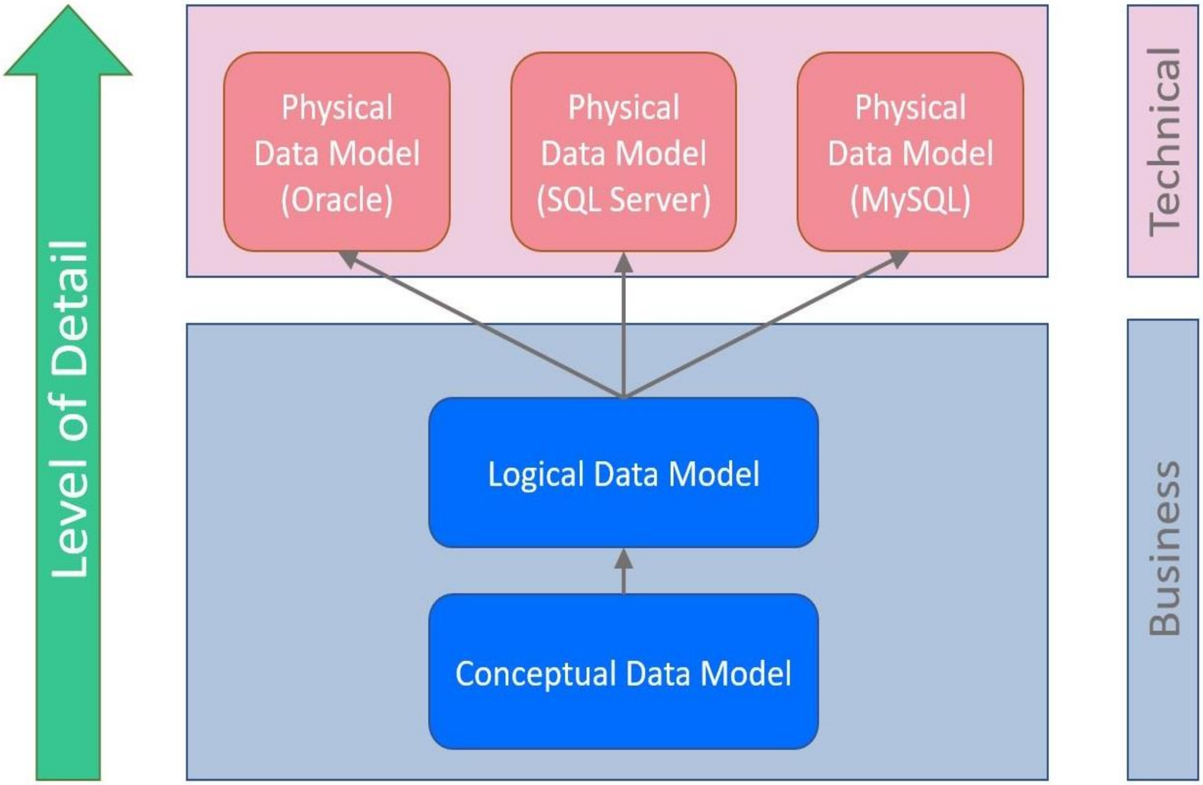
Physical Data Model

Characteristics of a physical data model:

- The physical data model describes data need for a single project or application though it may be integrated with other physical data models based on project scope.
- Data Model contains relationships between tables that which addresses cardinality and nullability of the relationships.
- Developed for a specific version of a DBMS, location, data storage or technology to be used in the project.
- Columns should have exact data types, lengths assigned and default values.

- Primary and Foreign keys, views, indexes, access profiles, and authorizations, etc. are defined.

N/B: From a single conceptual data model, we would expect to get a more detailed logical data model, both of them designed to represent the required data structures from a business perspective. From that logical data model, we can then create as many physical data models as needed, each of them designed for a particular database engine, as shown below:


Level of Detail
Physical Data Model (Oracle)
Physical Data Model (SQL Server)
Physical Data Model (MySQL)
Technical
Logical Data Model
Conceptual Data Model
Business

Advantages of Data model:

- The main goal of designing data model is to make certain that data objects offered by the functional team are represented accurately.
- The data model should be detailed enough to be used for building the physical database.
- The information in the data model can be used for defining the relationship between tables, primary and foreign keys, and stored procedures.
- Data Model helps business to communicate within and across organizations.
- Data model helps to documents data mappings in ETL process
  (ETL, which stands for extract, transform and load, is a data integration process that combines data from multiple data sources into a single, consistent data store that is loaded into a data warehouse or other target system.)
- Help to recognize correct sources of data to populate the model

Disadvantages of Data model:

- To develop Data model one should know physical data stored characteristics.
- This is a navigational system that produces complex application development, management. Thus, it requires knowledge of the biographical truth.
- Even a small change made in structure requires modification in the entire application.
- There is no set data manipulation language in DBMS.