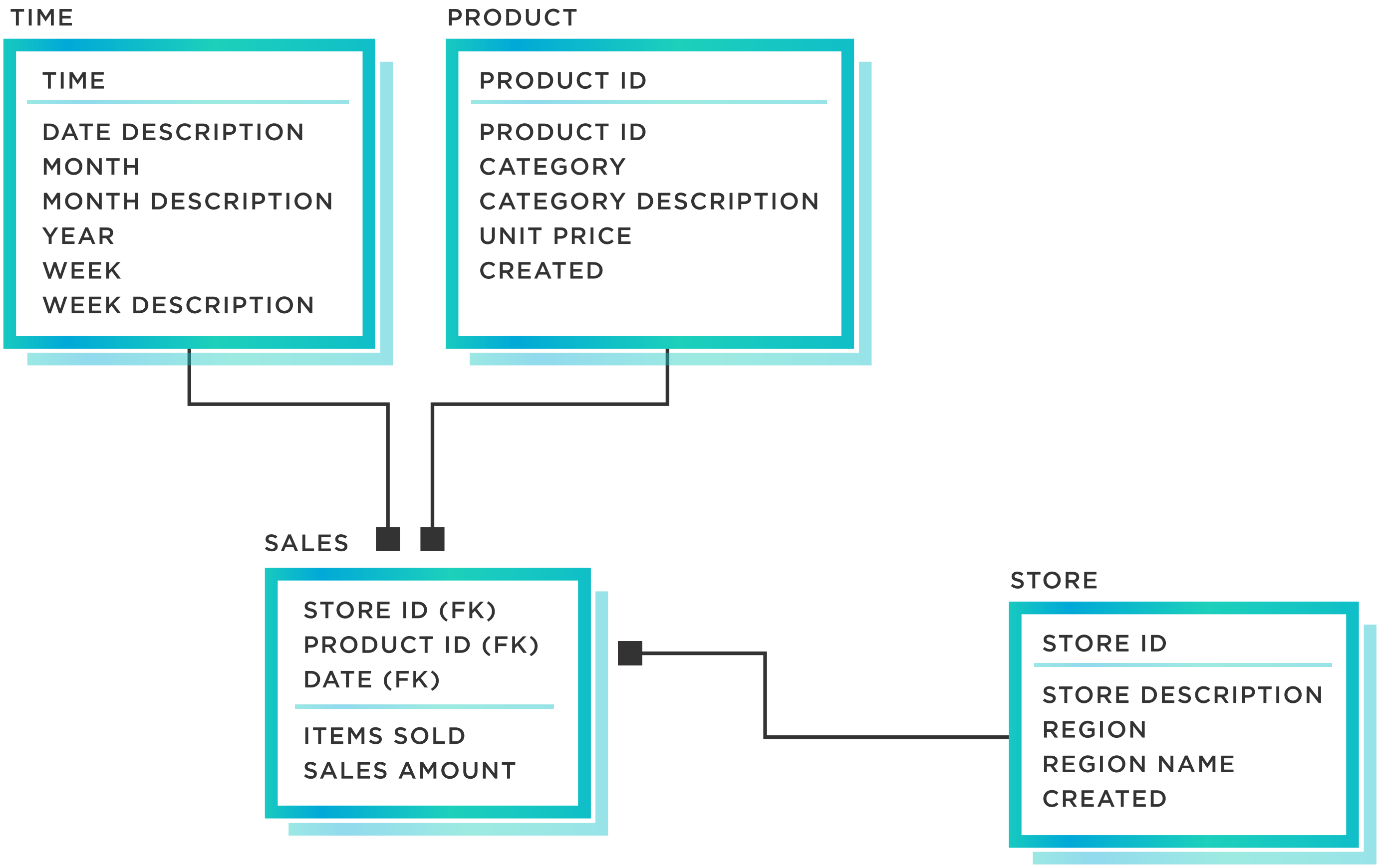
What is a Logical Data Model?

A **logical data model** establishes the structure of data elements and the relationships among them. It is independent of the physical database that details how the data will be implemented. The logical data model serves as a blueprint for used data. The logical data model takes the elements of conceptual data modeling a step further by adding more information to them.



The logical data model incorporates all of the elements of information that are vital in the running of the day to day business.

Components of a Logical Data Model

A logical data model has three main components:

- Entities: Each entity represents a set of things, persons, or concepts relevant to a business
- Relationships: Every relationship represents an association between two of the above entities
- Attributes: Each attribute is a descriptive piece, characteristic or any other information that is useful to further describe an entity

Each of these components of a logical data model are given a name and a textual definition. These serve to continually document business rules and outline the information requirements. The above components, however, are limited to descriptions of business requirements only.

They are not concerned with how said business requirements are processed, implemented, or stored.

TIBCO

Applying Data Virtualization: 13 Use Cases that Matter

The Need for a Logical Data Model

Given that data embodies the most crucial aspect of any application, program, or system, quality data processing and storage systems must be built upon a strong and accurate underlying data structure. A sound data structure gives application developers the freedom to design the best possible user interface, processing system, or statistical analysis and reporting set-up.

No matter how elegant or technical your system, it has to meet requirements, follow rules, and serve the purposes of the business or enterprise it is built for—or else it is of no practical use. Therefore, logical data modeling brings together the two most vital basics of application development:

1. Business requirements
2. Quality data structure

Characteristics of a Logical Data Model

These are the most important characteristics of a logical data model:

- A logical data model can describe the data needs for each individual project. Yet, it is designed to seamlessly integrate with other logical data models should the project demand it to do so.
- A logical data model can be developed and designed independently from the database management system. The type of database management system does not affect it that much.
- Data attributes contain data types with exact length and precisions.
- In logical data modeling, no primary or secondary key is defined. At this level of data modeling, it is required to verify and tweak connector details that were set prior to defining

relationships.

- A logical data model is like a graphical representation of the information requirements of a business area. It is not a database or database management system itself.
- A logical data model is independent of any physical data storage device, such as a file system.
- A logical data model must be designed to be independent of technology, so as not to be affected by the rapid changes in technology.

The Details of Logical Data Modeling

A data model, in a nutshell, is a set of data specifications and diagrams related to explain data requirements and related designs. Generally speaking, there are three types of data modeling types and activities:

Conceptual Data Model

This data model basically defines what the system inherently contains. Business stakeholders and data architects are typically the ones who create conceptual data models with the intent to organize and define various business concepts and rules and to set the parameters or scope thereof.

Logical Data Model

A logical data model serves to define how a system has to be implemented regardless of the database management system being used. Data architects and business analysts are usually the creators of a logical data model. The goal of creating a logical data model is to develop a highly technical map of underlying rules and data structures.

Physical Data Model

The physical data model pertains to how the system will be implemented, and factors in the specific databases management system. This model is typically created by developers. The idea is more to define how the actual database will be used or implemented for business purposes.

Generally speaking, both conceptual data modeling and logical data modeling are “requirements analysis” types of activities, while physical data modeling is considered to be a design activity.

A logical data model serves as the foundation for a physical data model, incorporating business requirements and gathering metadata. Logical data modeling can be done using standard techniques and data modeling notations.

Data modeling is an activity geared towards organizing data semantics, describing data, and addressing the consistency limits of data. It can be likened to an architect's drawing or building schematic, which forms the basis for conceptual modeling and establishes the relationships between various data components.

Data modeling techniques fall into one of two categories:

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

Logical data modeling belongs to the entity relationship model, built using an Entity Relationship Diagram (known as ERD), a standard modeling technique used as a communication tool by data modelers worldwide. Within it are the complete set of business requirements but not technical components.

Advantages of a Logical Data Model

- As data remains stable over time, a logical data model is also a stable one and highly conducive to data re-use and physical data sharing, which ultimately leads to reduced storage of redundant data.
- Components of a logical data model can be recycled, re-used, and adapted as more teams weigh in with their (often changing) needs.
- Costs associated with building and maintaining a logical data model are offset in the long run by the advantages it confers, not least by identifying and integrating all business needs and rules at the outset.
- Components of the building process, namely, design, coding, testing, and deployment go faster, as a direct result of the integration and clarification of business rules.
- Having a logical data model in place makes it easier, and therefore cost effective, to make changes, correct mistakes, or enter missing data during the development life cycle itself prior to implementation.

- User requests for making changes can be minimized by being proactive.
- Logical data models can be used for impact analysis, as each and every business process plus rule is connected within it.
- As objects in the logical data model bear textual definitions in business language, it makes it easier to maintain and access system documentation.

What Happens When a Logical Data Model is Not Developed?

Simply put, there can be issues. Users can get carried away with processes and activities if they are not reminded to highlight data, not technology, as the key ingredient when designing a new system. Designing a data model purely based on physical workflow misses out on representing crucial business requirements.

Tables and files that designers create without having the data elements outlined according to business requirements tend to be poorly organized and lack a sound underlying structure. Discovering and trying to factor in additional data elements from screen or report layouts during the process of coding, testing, or even during deployment forces developers to be reactive as opposed to proactive. The output is a mix-and-match entity that is difficult to operate or maintain, full of errors or excess text, minus system documentation, time-sucking and possibly unusable.

As the logical data model defines the structure of the data elements based on fundamental business requirements as well as the relationships between them, not having a logical data model in place means a lot of missed opportunities for improving business processes. Developers simply end up automating existing procedures, or recreating legacy systems on a newer technological platform which may eventually become obsolete.

Applying logical data modeling allows data analysts to think independent of the latest technology and focus on bettering business processes.

Therefore, a logical data model must be made a vital and indelible component of every application development project. It is an important step that should ideally precede database design.