**What Is a Data Model?**

A [data model](https://www.gooddata.com/blog/what-a-data-model/) is a visual representation of data elements and the relationships between them based on real-world objects. Data models reveal and define how data is connected within business processes and support the creation of efficient information systems or applications. For example, in business intelligence, a data model defines what kind of data users can utilize within their analytics.

To get a better image of what a data model looks like, check out the examples in our previous blog post, which covers the most used [data model types](https://www.gooddata.com/blog/relational-dimensional-data-models/). This will help you to create your own data model, but to achieve a fully functional and change-adaptable version, be sure to keep reading.

**What Is a Logical Data Model?**

A [logical data model](https://www.gooddata.com/blog/how-build-logical-data-models-scale-analytical-applications/) is a data model that provides a detailed, structured description of data elements and the connections between them. It includes all entities — a specific object transferred from the real world (relevant to business) — and the relationships among them. These entities have defined their attributes as their characteristics.

Logical data models bring together the two most vital basics of application development — business requirements and quality data structure — into a visual representation. Business analysts and data architects are responsible for creating these models. They map relevant business processes and reveal the business requirements in order to create a model which meets company goals. Moreover, they prepare a technical map of rules and structures depending on the scope of the project.

**Logical Data Model Examples**

The following schemas represent logical data model examples for different purposes. A logical data model diagram/schema contains all of the entities, attributes, and relationships in a visual form. Names are given to entities and attributes based on how they are actually used in a business environment.

The first logical data model example is related to a small e-commerce business. This logical data model captures orders of various products via an online store. All of the data detailing each customer's order is stored within the *Order Line*. One order includes information about the product and customer from the individual entities; *Product* and *Customer*. Within these tables, there is no duplicate information — if a specific customer has ordered something in the past, their data is already stored in the *Customer* table and with a new order this personal data is reused without the need to save it again. The same applies to the *Product* table — information about a specific product is stored only once, but this product can be a part of different orders.

Then, to get detailed information, the order date is added to each order.

Logical data model diagram for e-commerce

Another, more complex, logical data model example refers to a model for purchasing bank services. This logical data model diagram captures information about customer accounts. The information will be stored in *Account*, where data about a customer's personal details as well as additional information about the account will be gained from the individual entities; *Customer* and *Account Type*. Moreover, this model is also ready to store information about different bank purchases made via the customer's account, within the Purchase entity. Bank service details are stored in *Bank Service* and *Service Type*.

Logical data model: bank service purchases

Once the logical data model is finished and approved, the physical data model can be tackled.

**What Is a Physical Data Model?**

A physical data model specifies how the data model will be built in the database. It outlines all table structures, including column name, data types, column constraints, primary key and foreign key with indexes to the relevant table column, relationships between tables, stored procedures, and views.

The responsibility regarding physical data model creation usually lies with database administrators and developers. Information systems and software applications heavily rely on interactions with physical databases. Physical data models need to be designed and implemented correctly. It is challenging to modify physical data models once data from the existing application has been inserted into databases.

**Physical Data Model Examples**

The following physical data model examples are derived from the logical data model examples. It is worth noting that while logical data models serve as blueprints for physical data models, their attributes and names do not have to be the same as physical data model tables and columns. In short, objects in physical data models can be named differently, but they still have the same purpose.

Each logical data model example is translated into a physical data model schema. The first, which refers to an e-commerce use case, can be transformed into the following:

Physical data model diagram for e-commerce

Entities have been transformed into tables and attributes into table columns. Their names are also translated into technical terms — how they could be implemented and stored in the database. In addition, each column's data type has been specified.

For bank service purchases the schema might look like the following:

Physical data model diagram for bank service purchases

Each of the physical data models noted above is simplified for illustrative purposes; in a real-world environment, these physical models would be significantly more detailed and offer a broader perspective of the specific areas in question. You can try creating a logical and physical data model within our product.