**Logical database model**

In the logical database model we illustrate our database in a simple form. We can view the logical connection within our database as the product owner demanded it. We can divide this way the database to three parts: users, products, orders. We can also see the relationship between these parts. As in the physical model, we have a “one-to-many” relationship between the “users” and the “orders”. Every user can have more orders, but one order has only one owner.

The relationship between the orders and products is a bit more complex. These parts have a “many-to-many” connection. Every order can contain one or more products, and every product can be stored in zero or more orders.

**Physical database model**

With the database we tried to create as simple tables that we could. We divided the stored data, so we can work with them easily. We built simple relationships between the tables. We didn´t use a many-to-many relationship between the records in diferrent tables.

In our “users” we store all our info about the users of our product. As a primary key, we use an auto-number ID column. This way we can simply search for specific user in our database. Many of the data about the user is stored in text type. We also save the date when the given user joined our site.

The “users” table connects to the “orders” table with a “one-to-many” relationship. One user can have many orders, but one order can have only one owner. In the “orders” table we have the ID of the user that took an order. We have also the price of the order, and the time when the order was created. We also store the shipping state, so we can control the shipping department. We use an ID number as a primary key in this table. The more specific details of the order are saved in our “orderdetails” table. Between these two tables is also a “one-to-many” relationship.

In the “orderdetails” we have the concrete data regarding the products that were ordered. The number of products is also stored in this table. This table is linked directly to the table that has the data about our products in itself, the “products” table. Here we have the price, name, description and other info about the products in our store. To distinguish the products we use an ID number as the primary key.

The “products” table does not have the brand and the category of the product hardcoded in itself. We separated this information to two separate tables. The “brands” and the “categories”. That way we can find a specific brand or category easier in our database. The “products” table stores only the IDs. These two tables our connected to the “products” table with “one-to-many” relationships.

The “categories” table contains the ID, name of the category. Besides that, we save a link to the location in which is the default product photo of the category in case we do not have a product photo. We also use a parent-child system for the categories. Every category has a parent. The ID of this parent category is also stored within this table.