

Task 1: ER diagram

Figure 1 shows our ER model of the database for AltOnline. It structures the attributes and relations of departments, products, users, and orders in a way to satisfies all of the customer specifications for their website. The first assumption we make is that the store logo is the same for every page and only changes very rarely. Therefore, the logo should be handled fully in the front-end of the application and no URIs need to be stored in the database. Links to other departments and products can be accessed over their respective breadcrumbs. Each entity has a unique identifier in the form of <entity>_id except for ser which contains an identifying personnummer.

Every department further has the attributes breadcrumbs, title, and description, of which we require the breadcrumbs to be unique as the URL to the page has to be unique and derived from the relations with other departments. The departments are arranged in a hierarchy and can be top-level (no departments above), mid-level (departments above and below) or bottom-level/leaves (no departments below but products). We establish this hierarchy with a recursive relation in which an optional parent can have many optional children or products. Thus, top-/mid-/bottom-level departments can be identified. A special root department stores the desired welcome text as its description.

Each product is required to belong to exactly one department. It has the same basic attributes required for a page entry as a department but has to be vastly extended to offer more features. Stock is required to show whether a product is available, price_without_VAT and VAT_percentage are used to calculate the retail price as well as display the value-added tax percentage and price without value-added tax. Furthermore, a product can be sold by specifying the current_sale_percentage (default 0) and can be specified as is_featured to show it on the homepage. A many-valued attribute is used to store keywords for keyword-related recommendations.

Users are attributed with their data including personnummer (inherently unique and used to identify users), name, telephone, address, email_address, password (as a secure hash to login to the store), and a preference for newsletters can_receive_newsletters. Users can interact with products in two ways. The first one is that a user can optionally review many products. Each review depends on both a user and a product and consists of a rating (1-5) and optionally a comment. This is defined by the review's relation.

Second, a user may place many orders. Each order has a unique payment preference for the third-party payment service and tracking number for the package whereas the address is always tied to the user who placed the order. Additionally, an order includes information on its status (new, open, dispatched), order_date, and when it was last_updated_at. Of course, it is mandatory that an order includes many products and on the flip side, a product may be included in many orders. Their relation includes attributes for price_at_purchase and quantity. All other information like the title or description can be taken from the product. Since the retail price of a product can change over time, it is necessary to include the price at purchase e.g. refund a returned item. More extremely, a product may be taken out

of the store altogether. To differentiate between products for sale on the store page and products only relevant for past orders, the `is_being_sold` is added to the product.

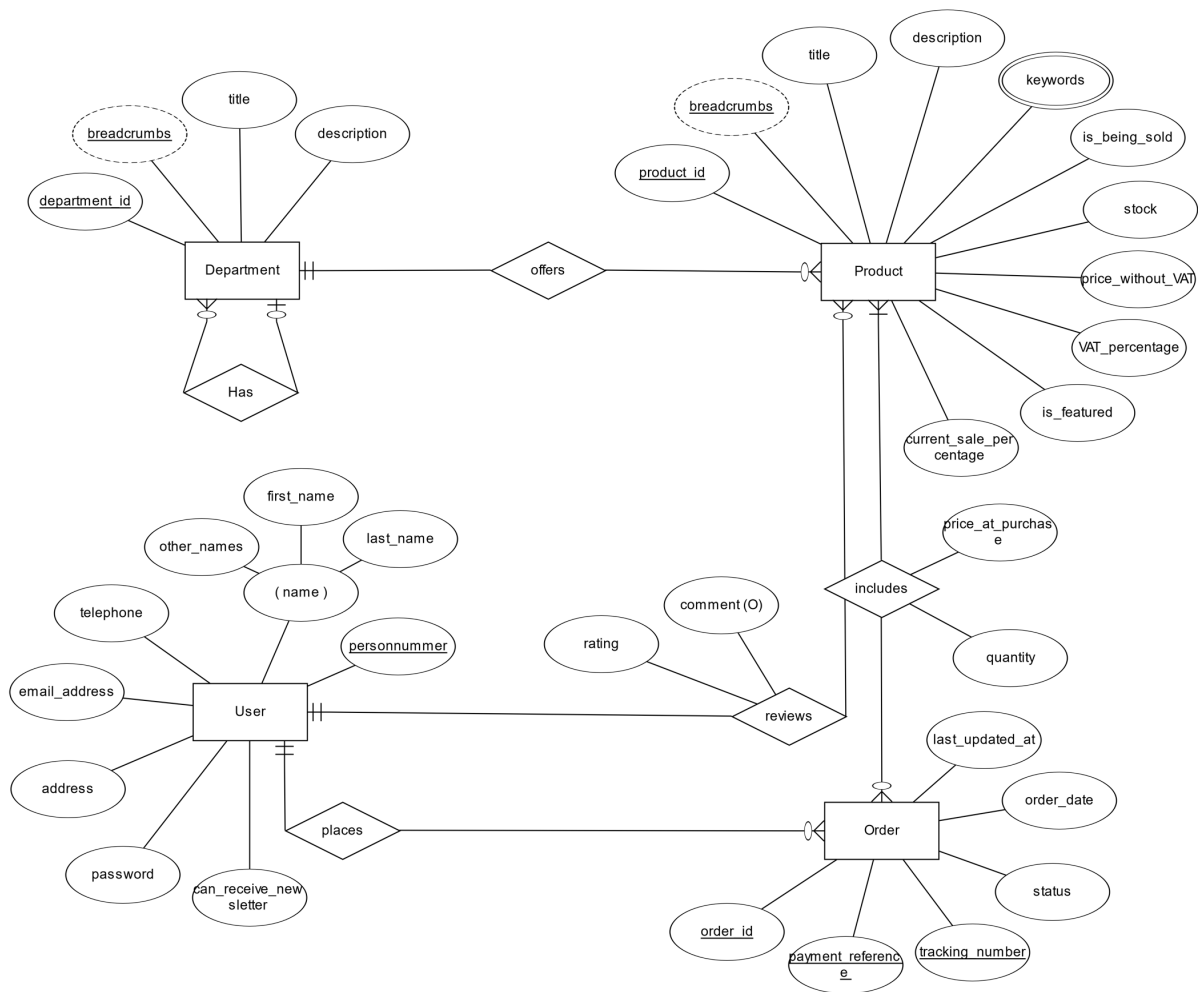


Figure 1: ER model of the database for AltOnline.