

*** ATG ER diagram Explanation ***

The ER diagram for ATG, the name of the dummy company for whom the shopping and inventory interface has been built, is based around four entities total; two strong and two weak.

The strong entities consist of ORDER and PRODUCT. Beginning with the ORDER entity, it is comprised of an ORDER_NUMBER which serves as the primary key. Every other attribute of the ORDER entity is related to either billing information such as the card number and the amount to charge, or the shipping address. The only attribute that does not specifically relate to either of these categories is NOTES, which serves as a place to store miscellaneous information about an order. Only the primary key ORDER_NUMBER is not allowed to be null, so as to allow partial orders to be served in the database in case an employee only has partial information at one time. ORDER has twelve attributes in total.

By contrast, the other strong entity, PRODUCT, consists of only three attributes. As PRODUCT_NAME is unique to each product, it serves as the primary key for the entity. The PRODUCT entity is meant to store the information related to the products currently held in inventory, so the attribute QUANTITY refers to the number of the product currently held in the master inventory of ATG, not the quantity ordered by a certain customer. Lastly, PRICE stores the dollar amount that each product sells for at ATG. Again, only the primary key must not be null so as to allow for partial completion of an entry, that can be completely filled in at a later time by an employee.

Moving on to the weak entities, we begin with ITEM_ORDER which tracks how many of each product are included in an order. Thus PRODUCT_NAME and ORDER_NUMBER are both primary and foreign keys. The only other attribute is QUANTITY, which tracks how many of that specific product were desired in the specific order. As ITEM_ORDER is completely dependent upon both a specific order and a specific product, it is weak.

Lastly, there is CART. The CART entity is used to add PRODUCTS to an order and to keep track of the master inventory. The primary key is PRODUCT_NAME and the other attributes are QUANTITY and COST. This time, the QUANTITY attribute refers to how many of the specific product the user wants to purchase, while COST reflects the specific cost of the CART based upon the quantity of the PRODUCT in question. As a CART entity cannot exist without a PRODUCT entity, it is weak.

Among these relationships, an ORDER entity may have many ITEM_ORDERS comprising it and needs at least one to exist as an ORDER, so the relationship is one-to-many. The ITEM_ORDER entity, on the other hand, is unique to an ORDER and cannot exist without one, so the relationship is one-to-one. Similarly, each ITEM_ORDER entity is unique to both a product, and an order, so the ITEM_ORDER to PRODUCT relationship is one-to-one. However,

a PRODUCT may not be associated with any ITEM_ORDER at all if none of the product has yet been sold. Alternatively, a PRODUCT can be included in multiple different orders at a time, so the relationship is zero-to-many. This is extremely similar to the relationships between the PRODUCT entity and the CART entity, and thus they have the same ones. A PRODUCT can be apart of many or no CARTs whatsoever, thus the zero-to-many relationship. Likewise a CART entity is entirely dependent on a PRODUCT entity for its existence, and each CART entity is unique to a sole PRODUCT, which is where we see the one-to-one relationship.