

Subject: E-Commerce Inspired Entity Relation Database

In order to begin the process of drawing the ERD, our team took some time to go around the Amazon website. One of, if not, THE largest e-commerce site to exist today. We wanted to make sure that we understood who the players were (buyers, sellers and everything in between) but more importantly how each entity reacts with one another. Needless to say that a lot goes on.

Our first draft of the ERD counts about 23 tables (shown in page 2 of the document) & 5 bridge tables! The ERD didn't seem to flow organically. We decided to break the relationships into 3 "main groups". (shown in page 3 of the document)

First group are the users and their respective administrative data (i.e. type of user (buyer/seller) and address). The second group contains everything surrounding a placed Order: invoice, payments, shipment, etc. Last but not least, is the Product "group", where we can find shopping cart, wish list, offers, reviews, etc. Dividing the ERD in terms of similar "actions/events" made more sense to us and helped us comprehend further the matter at hand. As seen in the diagram, we broke down some of the tables into hierarchies.

Finally, our final ERD become much more "lean" for academic purposes because we could have added 50+ more tables to really mimic Amazon's database design. The ERD on page 4 is more concise with 2 bridge tables that break down a many-to-many relationship found between orders and products, as well as, between products and discounts. You can also find 2 hierarchies in order to help us differentiate payment methods and user types. The hierarchies have strong relationships (shown with filled lines) because the 'child' tables are dependent on their respective parent tables. The rest of the ERD relationships are 'weak' relationships because the PK of parent is not found as PK of child table; the entities are entity-independent, so to speak.

An important note to add: all attributes/tables that have "disappeared" from our first draft have actually been more efficiently stored within the remaining tables. For example, the invoice hierarchy has been changed into an attribute found in the Orders table.

Although, we drastically reduced the number of tables & relationships, we believe that we have maintained the data integrity while keeping a clear and concise format to help the querying flow more efficiently.





