

Database Systems Project

Christopher Heath, Eric Pereira

December 7th, 2018

Our database contains a total of 7 tables. The main tables are Product, Individual_Product, Customer, and Order. The remaining tables are Products_Ordered, Membership, and Addresses. We chose this design to best support the required queries and to simplify the design.

The Individual_Product table is a member of Product. When a product is first created, say for instance an XBOX, an instance of Product is called upon to create that type of product with an associated name, price, type, and most importantly ID. When we need to make multiple instances of an XBOX product we create an Individual_Product, with the XBOX id in this case, and give it a unique serial number. We chose this design so that specific items could be found, and so that creating an individual item would be as simple as creating something with the correct id and a new serial number.

The customer table is where emails, passwords, usernames, phone numbers, and names are stored. If the customer does decide to become a member their data will be stored into the Members table with their start_date and end_date. The customer table has another associated table, Addresses, that stores contact information including main address, email, and main city.

The final tables are Order and Products_Ordered. Products_Ordered contains multiple instances of Individual_Products. This table contains the tracking_number foreign key from the Order table, and the serial_number foreign key from individual_product. The tracking_number and the serial_number together are the primary key for this table. We chose this design to support finding tracking information for specific items quick and efficiently. The Order table contains the attributes date_ordered, the primary key tracking_number, and the foreign key email of the Customer table. This design

allows us to look at all the Orders from a specific Customer.

To answer question number 6 in task_4 there is the question of giving two decompositions and proving they are lossless join and functional preserving. For example, with the customer and order tables it is lossless because if customer and order were to intersect they would only match at email, thus preserving data. There is an instance of this also with the product table and individual_product table where the only things that intersect are the product_id's. This is lossless as it preserves data and it is functionally preserving only necessary data.