

Name: Cushing, Bradley
Date: 11/02/24
NYU ID: N10695516
Course Section Number: CSCI-GA-2433-001
Project #1 Documentation

Documentation

This document contains information about a fictitious company called Circuit Blocks, Inc. which makes a toy called Circuit Blocks which adults and kids can use to build things with. Please see the attached 6W proposal for more information about the company and product.

There are two platforms for execution, a website maintained by the company and a single storefront owned and operated by the company. In short, the company will have a method for manufacturing, keeping inventory, marketing, and selling the toy using online means. They should also have a means for interacting with their customers after a purchase is made.

The requirements for modeling such a business are included below including any entities and relationships between those entities which are used to create the E-R model. Any relevant assumptions required to develop the E-R model are listed as well. Below requirements I have separated distinct and relationships mentioned to make them explicit.

Requirements

The company Circuit Blocks, Inc sells a single product, a toy with the same name called Circuit Blocks. For context, the toy consists of 27 small 1 inch by 1 inch blocks which can be connected together with another block on any side. 26 of the blocks have lights on all sides which light up when a current is passing through them. There is one special block which acts as a battery, where if any other block is connected to it directly or indirectly, that block will light up. In other words, if there is a current passing through the block from the battery, through the block, and back to the battery, then the block will light up.

Circuit Blocks Inc. sells this product to customers in the form of a single package. They also sell additional or replacement blocks to customers willing to purchase any number of them. Each PRODUCT has an Id, Type, and Cost associated with it specific to the unit cost of that type. Customers purchase either of the above products directly through the website circuitblocks.com or in person at the brick and mortar storefront. When a customer makes a purchase, an ORDER is created which lists the Order_id, Status, Date, Quantity, and Total_amount. Each order can be related to any number of different products.

The company only accepts credit cards and stores a single credit card on file for each customer. The information stored for a CREDIT_CARD is Fname, Minit, Lname, Number, Expiration_date, Security_code, and Zip_code. For every CUSTOMER a Customer_id is created and stored alongside a Name, Email, and Phone. For every customer that becomes a MEMBER, we

additionally store a Join_date, Expiration_date, and Status_tier. You can only become a member if you are already a customer. Status_tier is based on the number of years since a member's Join_date where 1 year is bronze, 2 years is silver, and anything above 3 years is gold. Memberships cost \$5 dollars yearly and must be renewed every year for the same amount. One month before the Expiration_date an email is sent to the member reminding them to renew their membership.

Customers can register their products to have them under warranty. All that's required from a customer to do so is their Order_id. When a customer registers a product a WARRANTY is created with a Warranty_id, Expiration_date, and Is_under which if true means currently valid. Expiration dates are automatically set to one year from the time the warranty is created.

For each product an INVENTORY is kept to make sure the product is stocked before an order is created and the product is sold. The inventory is rather simple, it stores the Location and Quantity in stock for a product at any given moment, and an Is_backordered indicator. When inventory is low, specifically when inventory is below 1000 packaged units of the toy or 5000 units of replacement blocks at any location, if it hasn't been back ordered already, a BACK_ORDER is placed containing an Id, Manufacturer, Quantity, and Cost which must relate to exactly one product and location. This is done automatically by monitoring the database and the manufacturer bill their card on file which happens outside of this system.

For marketing purposes the website allows a PROSPECT to sign up for different mailing lists with their Name and Email to receive email promotions. The MAILING_LIST also includes information for Type, Last_sent_date, Total_emails, and Is_unsubscribed indicating subscribed status. An outgoing marketing email is sent to prospects that are part of the mailing list. Each EMAIL contains information about its Title, Description, Markup (HTML). An email can be sent to any number of prospects in any number of mailing lists.

There are product designers who design the products to be manufactured. Each DESIGNER has an Id, Name, Salary and is assigned to a department and product to work on and can only work in one department and on one product at a time. The company is small so the only people that work there are the designers and programmers who develop the website. Each PROGRAMMER is also identified by an Id, Name, and Salary but related to a DEPARTMENT (Engineering, Product Design, Accounting, Marketing, or Sales) where they are assigned. Both designers and programmers are paid by PAYROLL every 2 weeks.

Programmers also want to be able to upload files to a file server so that built, static website files like HTML, JavaScript, CSS, and images among other things can be stored and served to prospects or customers visiting the website. These FILES have an Id, Name, and Contents to keep things simple for now.

Departments

Engineering
Design

Accounting
Marketing
Sales

Entities

- CUSTOMER
- MEMBER
- ORDER
- CREDIT CARD
- PRODUCT
- WARRANTY
- INVENTORY
- BACK_ORDER
- PROSPECT
- MAILING_LIST
- EMAIL
- DESIGNER
- PROGRAMMER
- PAYROLL
- FILES

Relationships

- Member IS_A Customer
- Customer BECOMES Member
- Member RENEWS Member
- Customer PLACES Order
- Order SELLS Product
- Customer USES Credit card
- Customer REGISTERS Warranty
- Product STORED_BY Inventory
- Inventory PLACES Back order
- Prospect SIGNS_UP_FOR Mailing list
- Mailing list SENDS Email
- Product DESIGNED_BY Designer
- Designer PAYED_BY Payroll
- Programmer PAYED_BY Payroll
- Programmer ASSIGNED Department
- Payroll PAYS Designer
- Payroll PAYS Programmer
- Programmer UPLOADS Files

Design choices

I'm using pgModeler which comes with some limitations that I've encountered while creating my conceptual schema, so I'm listing them here so my design choices make more sense.

pgModeler allows me to develop a conceptual schema and export this schema by creating an actual database locally on my machine. I am also able to export the DDL SQL code to more easily instantiate the database schema and tables elsewhere. Keep in mind I've chosen pgModeler so that I can use PostgreSQL which is an open-source relational database.

Inheritance

pgModeler doesn't work well with inheritance so I've either represented inheritance using classical relationships or have omitted it in some cases like with employees. For employees I just have entities for the subtypes and no supertypes.

Self-Relationships

Having a relationship where an entity is related to itself is also buggy in pgModeler and causes the conceptual model not to save so I've avoided using them. Since they aren't possible I have omitted these types of relationships so things are a bit less interesting.

Constraints

Custom constraints and domains are an advanced feature that get tricky quickly, so I've kept to using the basics that are necessary like basic data types and key constraints.

Schema

I'm only using a single schema to keep things simple as I'm not sure what the limitations are around having multiple schemas and how the tool and PostgreSQL handle them. Since we are dealing with a limited number of entities one schema should be good enough even though we're modeling multiple departments and functions within our business.