Electronics Vendor Database

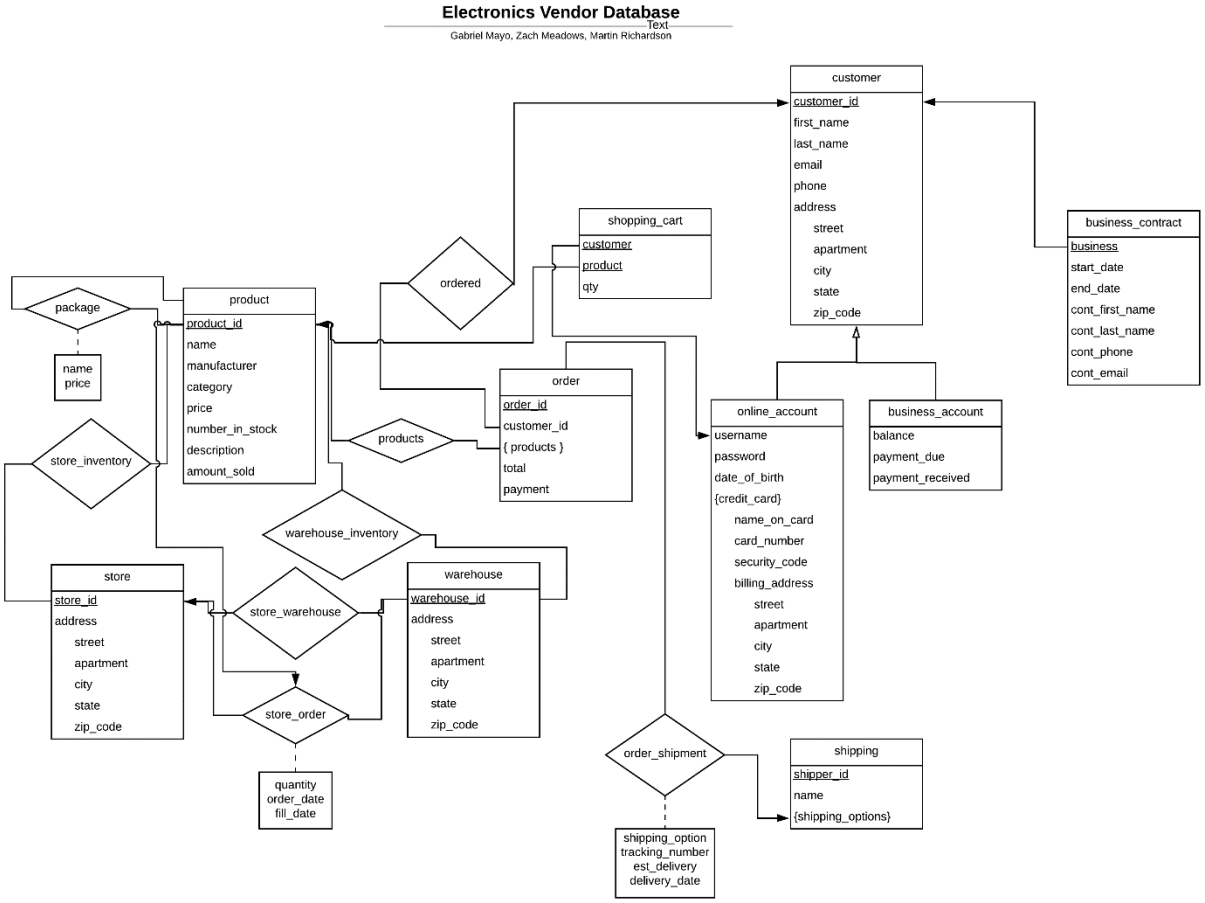
Gabriel Mayo, Zachary Meadows,
Martin Richardson

CSC 411

Ins. Bo Li, Ph.D.

May 5, 2019

Abstract: Retailers rely on a large volume of data to practice their business. Use of a relational database system is therefore extremely useful for tracking and managing this data. This project implements a relational database system in MySQL to model the operations of a fictional electronics vendor named DB Electronics.



For the database design we first identified several major entities: products, customers, physical stores, warehouses, and orders. These formed the core of our model, as shown in the above diagram (a full-size version is included in the project files).

From here we designed a relational schema that would reflect the above design while also taking MySQL's features into account. The schema is as follows (also included in a separate file):

Electronics Vendor Relational Schema

customer(customer_id, first_name, last_name, email, phone, street, street2, city, state, zip)
online_customer(customer_id, username, password, date_of_birth) foreign key customer_id references customer
credit_card(customer, option_number, name_on_card, cc_no, security_code, bill_street, bill_street2, bill_city, bill_state, bill_zip) fk customer references customer
business(business_id, name, balance, payment_due, payment_due_date, last_payment_received, last_payment_date, street, street2, city, state, zip)
business_contract(business, start_date, end_date, cont_first_name, cont_last_name, cont_phone, cont_email) foreign key business references business
product(product_id, name, manufacturer, category, price, description, no_in_stock, amt_sold)
package(package_id, name, price)
package_products(package, product) foreign key package references package, foreign key product references product
shopping_cart(customer, product, qty) foreign key customer references customer, foreign key product references product
order(order_id, customer, date, total, payment, shipping_option) foreign key customer references online_customer, foreign key shipping_option references shipping_option, foreign key payment references credit_card
order_products(order, product, qty) foreign key order references order, foreign key product references product
store(store_id, phone, street, street2, city, state, zip, warehouse) foreign key warehouse references warehouse
store_inv(store, product, qty) foreign key store references store, foreign key product references product
warehouse(warehouse_id, phone, street, street2, city, state, zip)
warehouse_inv(warehouse, product, qty) foreign key warehouse references warehouse, foreign key product references product
store_order(store_order_id, store, order_date, fill_date, warehouse) foreign key store references store, foreign key warehouse references warehouse
store_order_product(store_order, product, qty) foreign key store_order references store_order, foreign key product references product
business_order(order_id, business, date, total, shipping_option) foreign key business references business, foreign key shipping_option references shipping_option
business_order_products(order, product, qty) foreign key order references business_order, foreign key product references product
shipper(shipper_id, name, phone)
shipping_option(option_id, shipper, name, price) foreign key shipper references shipper

shipment(shipment_id, order, shipping_option, tracking_number, est_delivery, delivery_date) foreign key order references order, foreign key shipping_option references shipping_option
shipment_contents(shipment, order, product) foreign key shipment references shipment, foreign key (order, product) references order_product
business_shipment(shipment_id, order, shipping_option, tracking_number, est_delivery, delivery_date) foreign key order references business_order, foreign key shipping_option references shipping_option
business_shipment_contents(shipment, order, product) foreign key shipment references shipment, foreign key (order, product) references business_order_products

We populated the database with sample data and set about working on the interfaces. Due to a lack of time, we only completed the call center interface, the website, and the customer service interface. We also did not complete the sample queries.

The call center and customer service interfaces were coded in Java. The website interface was coded in PHP running on an Apache sever to connect to a MySQL database (an AMP stack) and was developed using XAMPP.

We learned a lot about the design and operation of databases in a very short time, but got started too late to complete much of the coding.