Ben Nordstrom
Max Grier
Project Group 49 - Castaways
CS 340 - Section 400 - Summer 2020
August 13, 2020
Link to project website

Project Step 6 - Portfolio Assignment

Submission Requirements (in order) - links go to portion of doc

1. Summary

a. Step 1

i. **Changes made:** Early on, we made some changes to our attributes to reflect best practices - they were originally written in plain English.

b. Step 2

i. **Changes made:** We chose not to change the table names where it was suggested. We added some clarifications for certain instances where it wasn't clear (e.g. invoices-orders relationship). We also added who would be in charge of implementing each respective table/relationship since that wasn't clear initially.

c. Step 3

i. **Changes made:** We reduced the amount of information displayed to the user. The UI was too cluttered which was a consistent feedback point. We left the data available to be used on the back-end for purposes of scalability later on (some could automatically be populated based on entry, e.g. time of invoice creation). We also cleaned up the HTML to be consistent across each page for a better user experience. We chose to keep our date fields separate by month/day/year for downstream use (from personal experience dealing with date timestamps within databases for analysis in SQL or on Dashboards).

d. Step 4

i. **Changes made:** We added the DDQ insert information for orders to our sql file as it was pointed out that wasn't originally included and also cleaned up our M:M select functionality within our DMQ and is implemented on our website. Outside of user feedback we had some buttons that did not do anything that we cleaned up on our UI.

e. Step 5

i. **Changes made:** We did some cleaning up of the HTML and SQL codes to fix some broken functionality pointed out by TA/peers (e.g. menu re-routes from the update pages weren't working). We also re-worked some of the relationships so that FK values could be null and we could actually nullify our relationship. We changed our warehouse_inventory table to include a primary key so we can utilize delete functionality and satisfy the delete M:M requirement in the project

guide. We also originally had an M:M relationship between orders and product_inventory, but the way it is implemented is as a 1:M relationship (1 product can be related to many orders) so it is now reflected that way. Since we have functionality to modify customers and it allows an employee to be mapped to many customers, it is technically a 1:M relationship. We have left it as 1:1 in our outline/ERD/Schema since that was the original intent. Being 1:M could also be appropriate if it just routes to an employee for a "friends and family" type discount instead of to employees only.

2. Project Outline

We are running a small-scale bike sales shop. Annual revenue is $20-30 million and we have a few locations in our greater vicinity with a strategically located warehouse for most of our inventory. The goal of our database is to easily track customer information and order history as well as our inventory and its location. This will ensure a quality data source which keeps track of our business operations. It also allows us to easily track and display a customer's order information as well as information about our products and where they are located. We also like to reward our employees so we use our database to show if a customer is an employee (for discount). The design of the database also allows us to scale our operations more easily as well as increase our product offerings if we choose to do so.

3. Database Outline

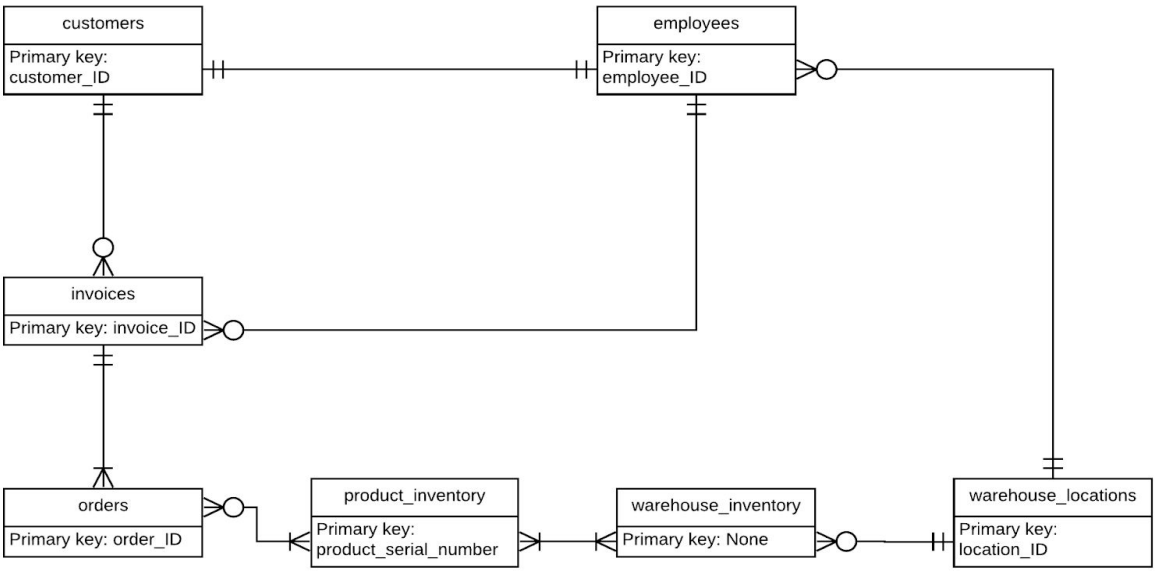
Our entities, attributes, and relationships:

- Bike shop - keep track of orders, inventory, etc.
    - customers: Details of the customers we do business with
        - Attributes
            - customer_ID: int, unique, auto_increment, NOT NULL, Primary Key
            - first_name: varchar, NOT NULL
            - last_name: varchar, NOT NULL
            - email: varchar, NOT NULL
            - phone_number: varchar, NOT NULL
            - employee_ID: integer, FK
            - gender: varchar
            - address_street: varchar
            - address_city: varchar
            - address_state: varchar
            - address_ZIP: integer
            - Payment info
                - credit_card_type: varchar
                - credit_card_number: integer
                - cc_expiration_month: integer
                - cc_expiration_year: integer
                - cc_cvv_code: integer
                - billing_address_street: varchar
                - billing_address_city: varchar
                - billing_address_state: varchar
                - billing_address_ZIP: integer
        - Relationships:

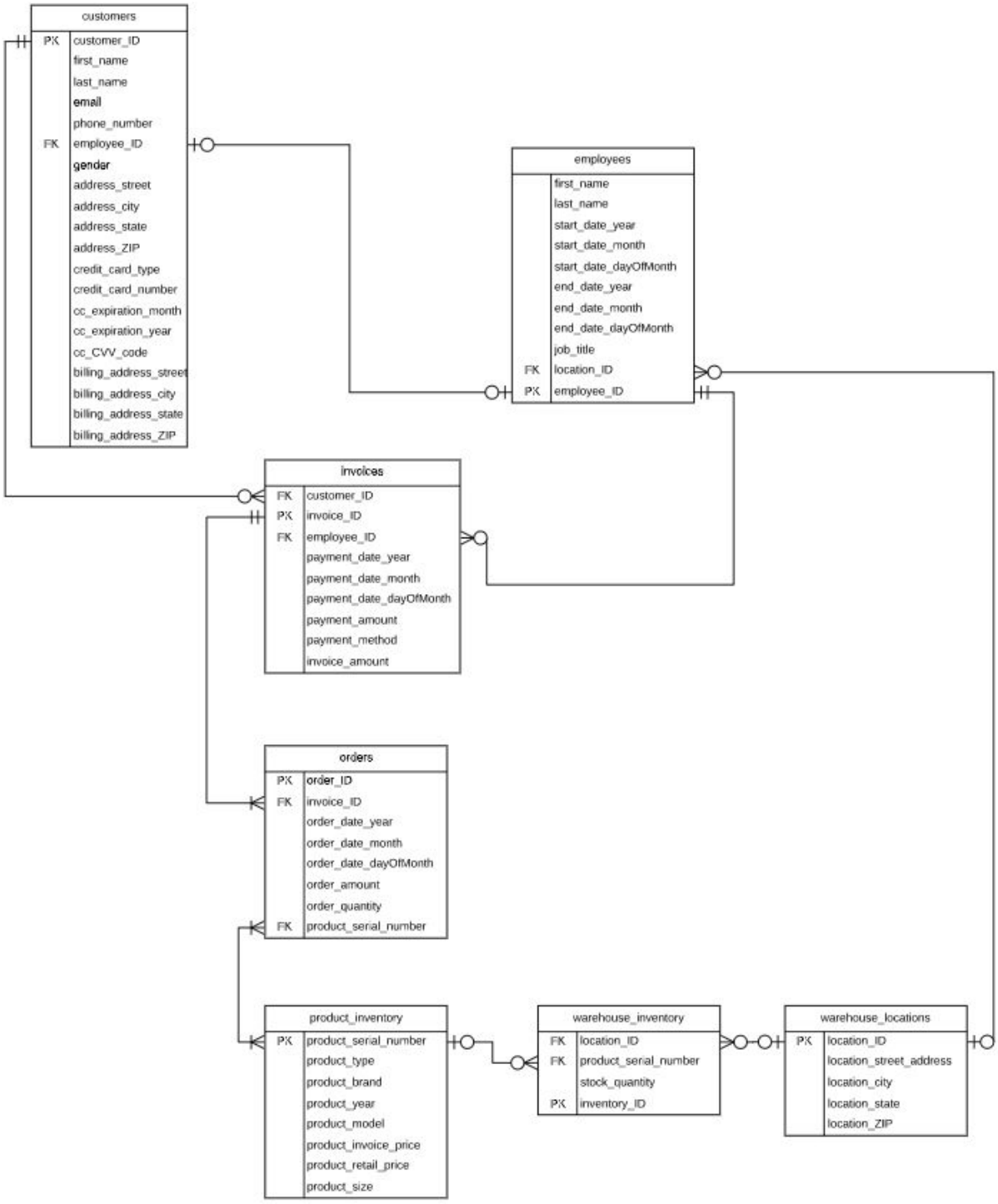
            - 1:M relationship with invoice (invoice_ID FK)
            - 1:1 relationship with employee ID (employee_ID FK)
    - invoices: What's billed to the customer and credited to an employee sale
        - Attributes:
            - invoice_ID: integer, unique, auto_increment, NOT NULL, PK
            - customer_ID: integer, FK
            - employee_ID: integer, FK, NOT NULL
            - payment_date_year: integer, NOT NULL
            - payment_date_month: integer, NOT NULL
            - payment_date_dayOfMonth: integer, NOT NULL
            - payment_amount: integer, NOT NULL
            - payment_method: varchar, NOT NULL
            - invoice_amount: integer, NOT NULL
        - Relationships:
            - M:1 relationship with customer (customer_ID FK)
            - M:1 relationship with employees (employee_ID FK)
            - 1:M relationship to orders (order_ID FK)
    - employees: Information on the employees
        - Attributes
            - employee_ID: integer, unique, auto_increment, NOT NULL, PK
            - first_name: varchar, not NULL
            - last_name: varchar, not NULL
            - start_date_year: integer
            - start_date_month: integer
            - start_date_dayOfMonth: integer
            - end_date_year: integer
            - end_date_month: integer
            - end_date_dayOfMonth: integer
            - job_title: varchar, NOT NULL
            - location_ID: integer, FK
        - Relationships:
            - 1:M relationship with invoices (1 employee is related to many invoices), invoice_ID is the FK
            - 1:1 relationship with customers (1 employee is related to 1 customer), customer_ID is the FK
            - M:1 relationship with warehouse locations, location_ID is the FK
    - orders: orders customers make for bike inventory (carries to invoices)
        - Attributes:
            - invoice_ID: integer, FK (from invoices)
            - order_ID: int, unique, auto_increment, NOT NULL, PK
            - order_date_year: integer
            - order_date_month: integer
            - order_date_dayOfMonth: integer
            - order_amount: integer, NOT NULL
            - order_quantity: integer
            - product_serial_number: integer, FK
        - Relationships:
            - M:1 relationship with product_inventory (product_serial_number as PK)

            - M:1 relationship with invoice (invoice_ID FK)
    - product_inventory: Information on the bikes the shop sells
        - Attributes:
            - product_serial_number: int, unique, NOT NULL, PK
            - product_type: varchar, NOT NULL
            - product_brand: varchar, NOT NULL
            - product_year: integer, NOT NULL
            - product_model: varchar
            - product_invoice_price: integer, NOT NULL
            - product_retail_price: integer, NOT NULL (we added to differentiate between cost to shop and cost to consumer)
            - product_size: varchar, NOT NULL
        - Relationships:
            - M:1 relationship with orders (an order_ID is the FK)
    - warehouse_inventory: Information on stock and location of inventory. This is represented as an intersection table between product_inventory and warehouse_locations.
        - Attributes:
            - inventory_ID: integer, NOT NULL, AUTO INCREMENT, PK
            - location_ID: integer, FK
            - product_serial_number: integer, FK
            - stock_quantity: integer
        - Relationships:
            - M:1 relationship with product_inventory (product_serial_number is the FK)
            - M:1 relationship with warehouse_locations (location_ID is the FK)
    - warehouse_locations: where the warehouses are located
        - Attributes:
            - location_ID: integer, not NULL, PK, AUTO INCREMENT
            - location_street_address: varchar, not NULL
            - location_city: varchar, not NULL
            - location_state: varchar, not NULL
            - location_ZIP: varchar, not NULL
        - Relationships:
            - 1:M relationship with warehouse_inventory (location_ID is the FK)
            - 1:M relationship with employees (employee_ID is the FK)

4. ERD



5. Schema



6. **Screen Captures**
    a. **Customers - CREATE (add new), READ, and UPDATE data**

Customers

First Name	Last Name	Email	Phone Number	Employee ID	Edit Customer Record
Ben	Nordstroms	fake-email-123@fakeemail.com	800-867-5309	4	Modify
Max	Grier	fakeR-email-321@fakeremail.com	867-530-9999	3	Modify
Lance	Armstrong	i-got-DQed@DQme.race	312-321-3121	1	Modify
Peter	Sagan	iRidebikes@bike.bk	111111111	None	Modify
				1	Modify
Sir	Lancelot	no-email@medievel.eu	000-000-0000	3	Modify
Tod	Cruzs	lol@lol.lol	123-456-789	2	Modify

New Customer Information:
First Name:
Last Name:
Email:
Phone Number:
Employee Tag: 1 - Max Grier
Submit Customer

Customer Update

Update customer information

Home | Customers | Invoices | Orders | Product Inventory | Warehouse Inventory | Warehouse Locations | Employees

First Name: Ben
Last Name: Nordstroms
Email: fake-email-123@fakeem
Phone Number: 800-867-5309
Employee Tag: 4 - Count Dracula
Submit

b. Invoices - CREATE (add new) and READ data

Customer Invoices

Invoice ID	Employee ID	Payment year	Payment month	Payment day	Payment Amount	Payment Method	Invoice Amount
1	4	2020	4	15	500	Credit Card	500
2	5	2020	5	15	1000	Credit Card	1700
3	5	2020	5	15	700	Credit Card	1700
4	2	2019	1	1	100	Check	9000

New Invoice Information:
Employee Responsible: Max Grier
Payment Date Year:
Payment Date Month:
Payment Date Day:
Payment Amount:
Payment Method:
Invoice Amount:
Submit Invoice

c. Orders - CREATE (add new), READ, and UPDATE data (nullify 1:M relationship)

Orders

Order ID	Invoice ID	Order Amount	Product Serial Number	Modify Order
1	2	10	1010	Modify
2	2	9000	1010	Modify
3	3	340	2345	Modify
4	4	1	2346	Modify

New Order Information:
Invoice ID: 4
Order Amount:
Product Serial Number: 1010 - Santa Cruz Chameleon
Submit Order

d. Product Inventory - CREATE (add new) and READ data (with filter)

Available Inventory

Filter Bikes By Type:
All
Submit

Bike Type	Brand	Model Year	Model Name	Invoice Price	Retail Price	Size
Mountain	Santa Cruz	2020	Chameleon	3500	3299	28
Road	Schwinn	2020	Cruiser	1000	1100	21
Electric	RadRunner	2020	Voltage	1200	1199	25
Hybrid	Cannondale	2019	SystemSix	1000	2000	44
Cruiser	Huffee	1981	OG Cruiser	1	1	1

New Product Information:
Product Type:
Product Brand:
Product Year:
Product Model:
Product Invoice Price:
Product Retail Price:
Product Size:
Submit Product

Available Inventory

Filter Bikes By Type:
Mountain
Submit

Bike Type	Brand	Model Year	Model Name	Invoice Price	Retail Price	Size
Mountain	Santa Cruz	2020	Chameleon	3500	3299	28

New Product Information:
Product Type:
Product Brand:
Product Year:
Product Model:
Product Invoice Price:
Product Retail Price:
Product Size:
Submit Product

e. Warehouse Inventory - CREATE (add new), READ, and DELETE data (M:M delete requirement)

Warehouse Inventory

Location ID	City	Serial Number	Brand	Model	Location Quantity	Delete Inventory
1	Portland	1010	Santa Cruz	Chameleon	None	Delete
1	Portland	2345	RadRunner	Voltage	None	Delete
3	Seattle	2345	RadRunner	Voltage	55	Delete
4	Transyvalia	2347	Huffee	OG Cruiser	100	Delete
2	San Francisco	2346	Cannondale	SystemSix	0	Delete

New Warehouse Inventory Information:
Location ID: 1 - Portland
Product Serial Number: 1010 - Santa Cruz Chameleon
Location Quantity:
Submit Warehouse Inventory

f. Warehouse Locations - CREATE (add new) and READ data

Warehouse Locations

Street Address	City	State	ZIP
123 N Mississippi Ave	Portland	OR	97205
4567 Howard St	San Francisco	CA	94103
1221 24th Ave E	Seattle	WA	98112
1 Transylvania Ave	Transyvalia	TV	0

New Location Information:
Street Address:
City:
State:
ZIP:
Submit Location

g. Employees - CREATE (add new) and READ data

Employees

First Name	Last Name	Job Title	Location
Max	Grier	Co-Founder	Portland
Peter	Sagan	Brand Ambassador	Seattle
Ted	King	Team Racer	San Francisco
Count	Dracula	Sucker of Blood	None

New Employee Information:
First Name:
Last Name:
Job Title:
Location: Portland
Submit Employee

7. Team Evaluation Form

CS 340 TEAM EVALUATION FORM

AUGUST 12, 2020

RATE YOUR TEAMS PERFORMANCE USING THE SCALE BELOW.

1 = Strongly Disagree 2 = Disagree 3 = Agree 4 = Strongly Agree

GROUP NUMBER	Group 49 - Castaways	
NAME OF GROUP TEAM MEMBERS:	Ben Nordstrom & Max Grier	
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS

HOW PREPARED WAS YOUR TEAM? **Research, reading, and assignment complete**	4	We could have included additional functionality beyond the project requirements which we could technically lower our score here, but we are not because the functionality is not part of the project requirements (e.g. input validation, etc.).
HOW RESPONSIVE & COMMUNICATIVE WERE YOU BOTH AS A TEAM? **Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.**	4	Both team members responded promptly and were available to meet when needed. We also communicated well to initiate if there needed to be some discussion around portions of the project.
DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY **Contributed best academic ability**	4	Yes both members participated equally. There were natural ebbs and flows because of personal obligations throughout the quarter, but we feel that the contributions were equal when evaluating across the entire term.
DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT? **Were both team members both positive and productive?**	4	The contract was followed. There was never a time where there was tension/disagreement between us and we were both understanding of the balance between school obligations and personal/work/other obligations and how those can change over time / throughout the term. Communication as noted above was prompt - we never went more than a couple of days without communicating on progress for the project or what needed to be done etc.

Are there any suggestions for improvement for your team and what are your goals moving forward?

(Better communication, follow the contract better, modify the initial team contract, more contribution, etc?)?

We could have potentially set our goals higher than the basic requirements for the project to make our submission something that would be expected in real-world implementation (user validation, better error messaging, etc.). But with the condensed Summer term the turnaround was difficult. I think we could have implemented more functionality and made it a little sleeker/cleaner with more time and added those extras that separate good projects from great projects. Overall it was a good experience and definitely one to learn a lot from because I imagine that a lot of software development at a company is done as a team, so you will need to learn to work with different people and deliver on a set of project requirements like we did, while also balancing other personal obligations (or pandemics).