CS340_400
11/23/20
Group 34 members:
Matthew Wong
Lucas Ball

URLS:
Homepage: http://flip3.engr.oregonstate.edu:5558/
Dealership: http://flip3.engr.oregonstate.edu:5558/dealerships
Employees: http://flip3.engr.oregonstate.edu:5558/employees
Cars: http://flip3.engr.oregonstate.edu:5558/cars
Transaction: http://flip3.engr.oregonstate.edu:5558/transactions
Customers: http://flip3.engr.oregonstate.edu:5558/customers

Project Step 6 Draft Version: Implement CREATE + READ operations

--

- **What works**
    - The Update and Delete functions on Employees and Cars
- **What doesn't work**
    - The Update and Delete functions on everything else. As well has foreign key synchronization for the relationships
- **Where/why you are blocked.**
    - Not sure how to make the foreign keys sync up with the queries. (Maybe make separate queries?)

Project Overview:

Group 34 Used Cars is a chain of used car dealerships that sells used automobiles. The average sales per day are around 2-5 cars and each dealership however there are multiple dealerships with over 50+ different makes and models of cars in their inventory, multiple employees that deal with different sales per day. Our app would make it simple for Group 34 to track their sales and inventory for every existing dealership in their chain.

Entities to Implement:

- Dealerships - Lucas
- Employees - Matthew
- Cars - Lucas
- Dealership_cars - Matthew
- Transaction - Lucas
- Customers - Matthew

**update the entities below to reflect the nullable customer > dealership*

Dealerships: ***holds unique information per dealership in the chain***

- Dealership_ID: int, unique, not NULL
- Dealership_Name: varchar, not NULL
- Dealership_PhoneNum: int, not NULL
- Dealership_EmployeeNum: int, auto_increment, not NULL

Employees: ***holds unique information on employee as well as which dealership they work at***

- Dealership_ID: int, unique, not NULL
- Employee_ID: int, unique, not NULL

- Employee_type: varchar, not NULL
- fName: varchar, not NULL
- lName: varchar, not NULL

Cars: ***detailed information on make, model, etc of an individual automobile***

- Car_ID: int, unique, auto_increment, not NULL
- Car_Make: varchar, not NULL
- Car_Model: varchar, not NULL
- Car_Year: int, not NULL
- Car_Price: int, not NULL

Dealership_cars: ***tracks the inventory of cars based on the cars entity per dealership***

- Dealership_ID: int, unique, not NULL
- Dealership_CarStock: int, auto_increment, not NULL
- Car_ID: int, not NULL

Transaction: ***information on a sale between a dealership/ the employee making the sale and customer***

- Transaction_ID: int, unique, auto_increment, not NULL
- Dealership_ID: int, not NULL
- Employee_ID: int, not NULL
- Date: int, not NULL
- Car_ID: int, not NULL
- Customer_ID: int, not NULL

Customers: ***holds information on customer visiting a dealership***

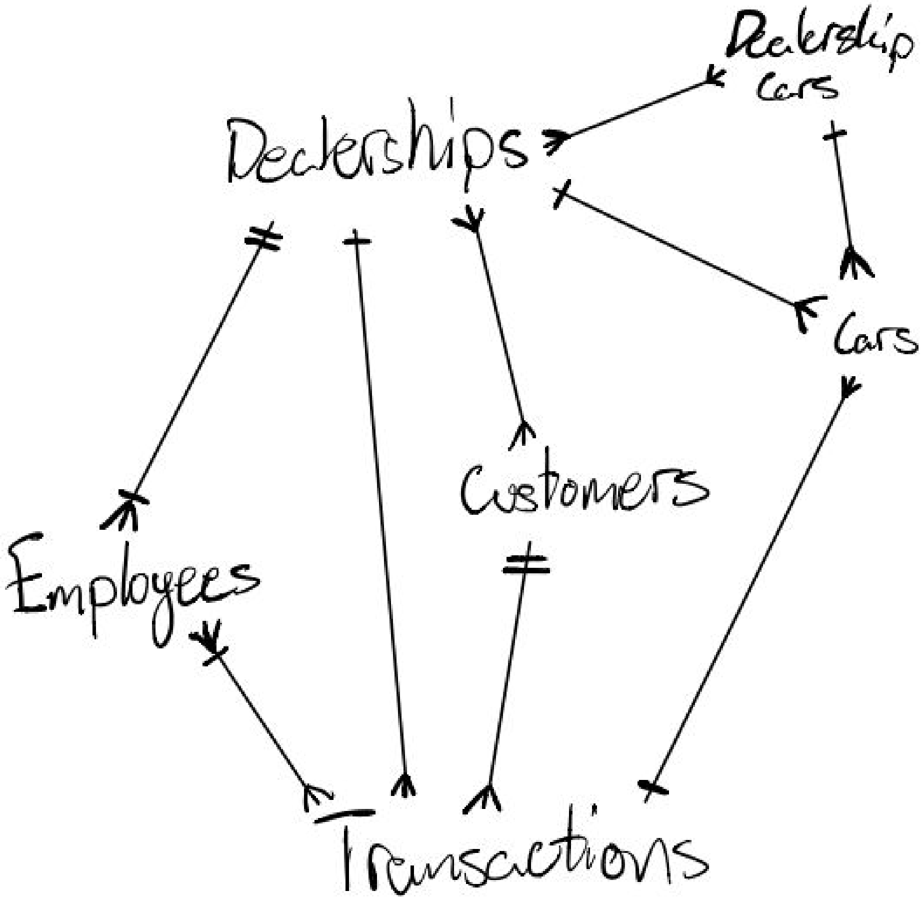
- Customer_ID: int, unique, auto_increment, not NULL
- Transaction_ID: int, not NULL
- Dealership_ID: int
- fName: varchar, not NULL
- lName: varchar, not NULL

Relationships:

- Dealership has a one and only one relationship with Dealership_Cars
- Dealership has a zero or many relationship with customers
- Dealership has a one or many relationship with Employees
- Employees has a one and only one relationship with dealership
- Customers has a zero or many relationship with dealerships
- Dealership_Cars has a one or many relationship with Dealerships
- Dealership_Cars has a one and only one relationship with cars
- Cars has a one and only one relationship with cars
- Cars has a one and only one relationship with transactions
- Transactions has a one or many relationship with cars

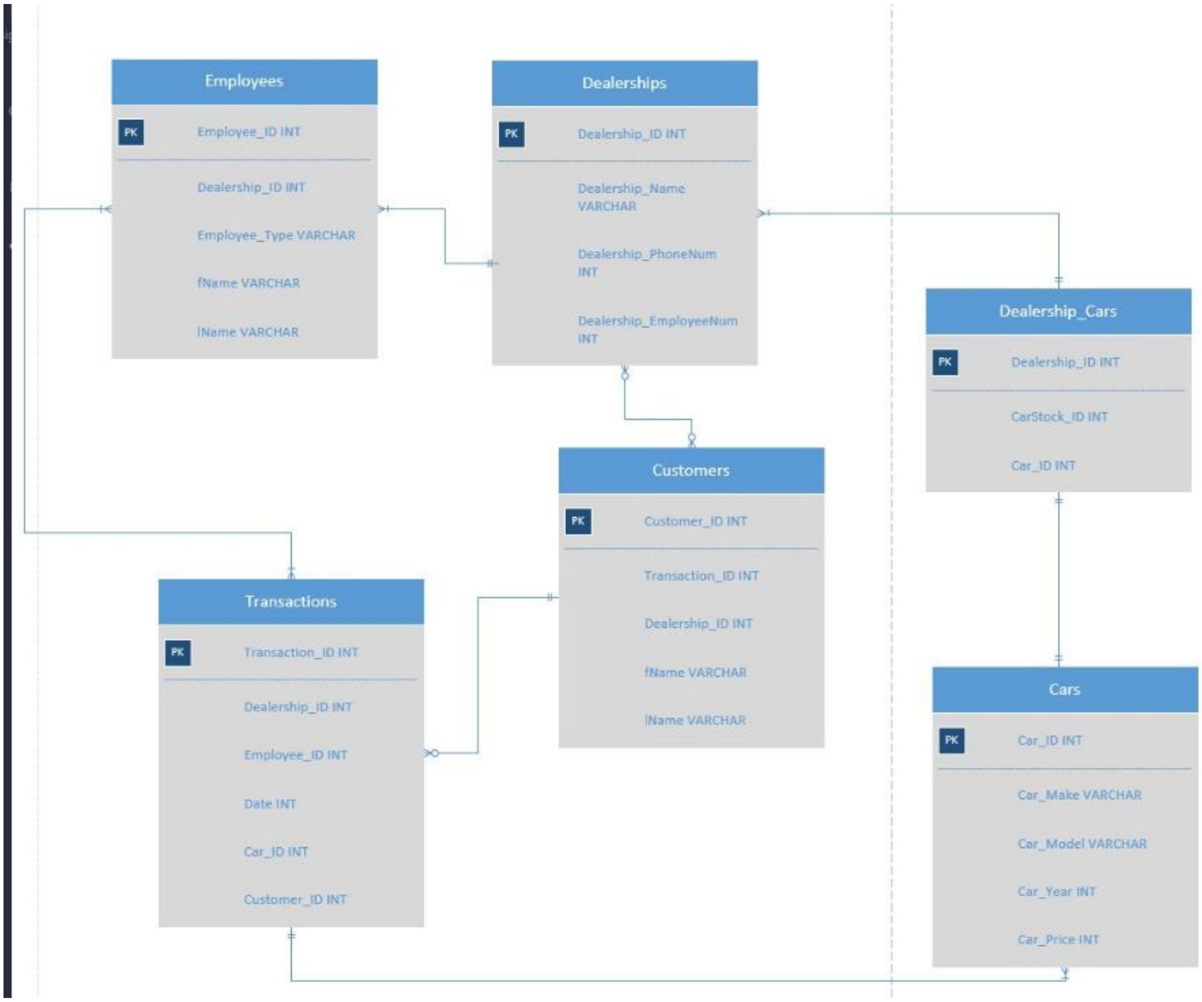
- Transactions has a one and only one relationship with customers
- Customers has a zero or many relationship with transactions
- Employees has a one or many relationship with transactions
- Transactions has a one or many relationship with Employees

Entity-Relationship Diagram:



Schema:

- This schema should follow the database outline and the ER diagram exactly. It will be graded in the Final Version for this Step on the extent with which it matches the database outline, with an emphasis on whether the relationships, tables, and keys are set up correctly. You may use the tools listed on Tools for this course, or draw by hand and upload a scanned legible copy. Please stick to the notation from Week 3.



RATE YOUR TEAMS PERFORMANCE USING THE SCALE BELOW.

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

GROUP NUMBER:	34	
NAME OF GROUP TEAM MEMBERS:	Lucas Ball, Matthew Wong	
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
HOW PREPARED WAS YOUR TEAM? Research, reading, and assignment complete	4	N/A
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	N/A
DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY Contributed best academic ability	4	N/A
DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT? Were both team members both positive and productive?	4	N/A

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?)?

CS 340 TEAM EVALUATION FORM

DATE \@ "MMMM D, YYYY" APRIL 1, 2020