

Group 49 – CS 340 Intro to Databases

Mohamed Ibrahim

Jesus Ponce

URL:

<http://flip1.engr>

[.oregonstate.e](http://flip1.engr.oregonstate.edu)

[du:9123](http://flip1.engr.oregonstate.edu/du:9123)

## Project Step 1 Draft

Reviews:

Ahmed Mustafa

Great job with the write-up! The overview gave a lot of insightful information into your goals with that database and the information you would be tracking.

Problem Description: Managing, recording transactions, inventory, and customer interactions for an Alfa Romeo car dealership is the goal. It establishes the brand's market presence and targets the specific needs of the car dealership. The project details the number of models offered and the annual sales figure. Mohamed and Jesus provided an understanding of the needs of the database.

Entities: The proposal outlines five main entities that are Models, Trims, Customers, Employees, and Invoices. They

cover aspects of the dealership's operation. Each entity is well-defined with a single purpose. All are easy to understand as well.

Attributes and Relationships: The attributes for each entity are specified. These include the data type and any constraints. The relationships, include 1:M and M:M. The logic flows well. The proposal could benefit from specifying the data types more precisely like `varchar(100)`. Second, perhaps you could simplify your Trim table a bit. I like the current way you're doing it, however it might become more confusing in the long run since you have 5 models and then multiple trim levels for each specific car. You could simplify this by having the model table with an MSRP with something like 3 trims only. Something along the lines of 'none', 'sport', and 'luxury.' This way the MSRPs you manage in the trim table would be something along the lines of \$0, \$10,000, and \$20,000. Then the Model's would be the base for each of the 5. Your total price field on the invoice would just be the sum of these two.

Names and General Formatting: Both look good. I found the names fairly easy to understand. All variables look to be capitalized correctly. Data types also make sense.

ERD: ERD looked good! It has all the data types and item names lists. The types look to match the outline from my look. Good job! Wasn't quite sure if the green highlight meant anything? If it didn't, I would remove it for the future.

Recommendations: I wanted to make sure the purpose of the database. It seems like it manages people and invoices, but not cars. I was wondering if a “current inventory” table would be useful for you? Perhaps the cars are made to order, so you wouldn't need an inventory, but I thought it could be helpful to you!

Overall: I like your plan for an Alfa Romeo dealership database. It has structure and looks to solve a real world problem. It provides a fairly comprehensive design for tracking data and supports the organization. Make sure to update the data if you implement additional tables or change the current model. With some small fixes and gradual improvements, this project should be successful!

Hopefully this is helpful! Best wishes to both of you on the project.

### **Annette Tongsak**

Hi Group 49,

This is a very useful idea for a small-scale, luxury car brand that depends on a select market.

Does the overview describe what problem is to be solved by a website with DB back end?

Yes, the overview clearly describes how a DB backend will help an Alfa Romeo car dealership pinpoint best-selling models, highest-performing employees, and customers to follow up with.

Does the overview list specific facts?

Yes, the overview mentions specific facts like the number of Alfa Romeo models offered in the U.S. (5) and how many units are sold a year within the U.S. (roughly 20,000).

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, there are five entities described (Models, Trims, Customers, Employees, Invoices), each of which can be stored in a list. Each entity has a name, like Models's modelName to specify a car's model.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes, the purpose of each entity is described.

Attribute datatypes and constraints are listed. One suggestion I have is to store employees' and customers' full names instead of keeping their first and last names separate. I believe you can use the VARCHAR datatype to store this information.

Relationships between entities are also described. Here is my understanding of them: Invoices & Models - M:N

- An Invoice can have many Models

- A Model can have many Invoices

□

- I think an invoice should capture data for the purchase of a single vehicle and that an invoice needs a model type. In that case, it would make more sense for this to be 1:M so that instead, it's:
  - A Model can have many Invoices
  - An Invoice must have one Model

#### Models & Trims - 1:M

- A Model must have at least one Trim
- A Trim must have one Model

#### Invoices & Trims - M:N

- An Invoice can have many Trims
- A Trim can have many Invoices

#### Invoices & Customers - 1:M

- An Invoice must have one Customer
- A Customer must have at least one Invoice

#### Invoices & Employees - 1:M

- An Invoice must have one Employee
- An Employee must have at least one Invoice

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

As mentioned above, I think it would make more sense for the relationship between Invoices and Models to be 1:M instead of M:N.

Yes, there is at least one M:N relationship.

The ERD presents a logical overview of the database but some FKs are incorrectly labeled, making it hard to see such relationships.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

There are some issues with the ERD, including the labeling of FKs, inconsistent naming and capitalization of attributes, and lack of proper participation lines for optional participation relationships. The ERD has some inconsistencies with the rest of the written outline, like how modelsID from Models is a previously

unmentioned FK in Trims. It would help to be more specific about what specific attribute is being used as FKs in the outline.

## **Benjamin Mammet**

1. offers the dealership more information to make decisions with in regards to their performance and sales.
2. The only specific fact mentioned in the overview is that there are roughly 20,000 units sold in a year with this company. There is no information such as revenue for a dealership, or foot traffic and interest of this brand.
3. There are five entities, and each is a single idea. However, it seems that the Models and Trims entities are a bit excessive. There only a small number of both trims and models, so it seems that having an entire entity dedicated to each could further complicate things.
4. The entity outlines clearly outline the purpose and type of each data point, however, sizes are not given for any of the data points. The naming convention of each makes the purpose clear, but there is no further description given.
5. The 1:M relationship for the *Employee* entity is not formulated correctly. The relationship is defined as 1:M, however, it is written that an invoice can have at most one employee, making it a 1:1 relationship. There are two M:N relationships, and the diagram is correct, minus the note above.
6. A) There is consistency in naming between the overview and the entity properties. The naming is clear enough to understand what their purpose is. B) All entity names are plural, and attribute names are singular. The capitalization scheme is consistent throughout, in that, if there are multiple words, only the first word start with a lowercase.

## **Jacob Barber**

- Does the overview describe what problem is to be solved by a website with DB back end?

- Generally, yes. While it does not describe a specific problem that the database is solving, it provides a detailed overview of the specific functionality the DB back end is expected to provide, as well as general benefits for the Alfa Romeo dealership.
- Does the overview list specific facts?
  - Yes, it describes the volume of Alfa Romeros sold in the U.S. and the number of brands. It might be useful to see more specific figures for the dealership itself, to have a better idea of the scope of the DB back end.
- Are at least four entities described and does each one represent a single idea to be stored as a list?
  - It has five entities, but I believe Trim could be broken down into more specific entities. It seems a bit vague. What makes a trim level? What facts about trim are relevant at each level? Additionally, I'm not sure if the entities provided are a sufficient model for the features described in the overview.



- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
  - Yes, but more description of some attributes could be useful. As someone without a mental model of what the database should handle, not all attributes are self-evident and I think the relationships might be a bit over-simplified.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
  - There are two M:M relationships and the diagram is correctly laid out.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
  - Yes, all names are consistently formatted and capitalized.

### Actions Based on Feedback:

We think the overview properly states the problem the database is intended to solve, as it is meant to simplify infrastructure and is intended to streamline the financial and administrative load on the company.

We think our outline is great as is, but, with the contributions of our peers, adding more facts such as number of dealerships, revenue and MSRPs of models could help our overview.

Although combining the trim and model entities could simplify the ERD, we think that having them separate will allow for future changes to not affect the entities and DB systems. Such as if the manufacturer decided to come out with a new trim, this way we are updating only the trim entity and adding a new model, just because a new trim was

created.

We do think adding a current inventory entity could better the DB, keep track of the inventory at this dealership. With attributes giving information regarding the available units for purchase and where they may be coming from – for example, *On Site* vs *For Order*. We have added these entities into the overview.

We do need to edit the naming on the attributes since some are inconsistent, some ending with s and others not, we will change to make the Entity titles end in S, and the attributes as singular.

### **Upgrades:**

Current research is being done to bring forth more quantitative data when it comes to sales numbers, expenses, and profit margins for these kinds of businesses that would justify using this backend DB to help run the business. No further upgrades were made at this point.

### **Overview:**

This is a database structure serving as a repository for an Alfa Romeo car dealership.

Accordingly, it will record data detailing the models offered, the employees working within the dealership, the customers who come in registering interest, as well as the invoices created upon sale of the vehicles. A brand with only 5 models offered in the United States, Alfa Romeo instead depends on its popularity amongst the car enthusiast community, selling roughly 20,000 units a year through their 126 dealers in the United States. This database will allow the users to identify which *Models* are selling at a greater rate, which *Employees* are performing the best, and will also allow them to follow up with *Customers*, marketing additional offers and services even after sale. Each *Model* is further specified by the *Trim* level, which is where the price distinction can be recorded. As a result, *Employees* can be offered commissions based on a percentage of the numbers cited in the *Invoices* table. Total sales can be calculated by adding all of the *Invoices*, and profit, minus overhead expenses, can then be calculated by taking the total sales and subtracting the *Employees'* salaries and commissions.

This database simplifies infrastructure and is intended to streamline the financial and administrative load on the company. It allows for added functionality, in terms of customer support, and even allows for customizability, without impacting the entire system, down to the trim level. This allows for information regarding the pricing of each unit on the lot to be updated, deleted, and modified without impacting broader information stored in *Models*.

The five tracked entities are: *Models*, *Trims*, *Customers*, *Employees*, and *Invoices*

### **Outline:**

- *Models*: Records information regarding the cars being sold including make, model, year
  - *model\_ID*: int, auto\_increment, unique, not NULL, PK
  - *model\_Name*: varchar, not NULL
  - *model\_Year*: int, not NULL
  - *relationships*:
    - M:N relationship between *Invoices* (M) and *Models* (M) with *Models* as a FK in *Invoices* – many models can be on a single invoice and many invoices may have the same model
    - 1:M relationship between *Models* and *Trim* – a single model can have many trim levels, a trim level can only be associated with its model
- *Trims*: Records information regarding the vehicle unique that can be determined by the trim level
  - *trim\_ID*: int, auto\_increment, unique, not NULL, PK
  - *trim\_Name*: varchar, not NULL
  - *trim\_Availability*: BOOLEAN, not NULL
  - *trim\_Units*: int, not NULL

- *trim\_MSRP*: int, not NULL
- *relationships*:
  - M:N relationship between *Invoices* (M) and *Trims* (M) with *Trims* as a FK in *Invoices* – many trim levels can be on a single invoice and many invoices may have the same trim level
  - 1:M relationship between *Models* and *Trim* – a single model can have many trim levels, a trim level can only be associated with its model
- *Customers*: Records information regarding the customers who register interest or have purchased a vehicle

- *customer\_ID*: int, auto\_increment, unique, not NULL, PK
  - *customer\_FirstName*: varchar, not NULL
  - *customer\_LastName*: varchar, not NULL
  - *customer\_StreetAddress*: varchar, not NULL
  - *customer\_PhoneNumber*: varchar, not NULL
  - *customer\_EmailAddress*: varchar, not NULL
  - *relationships*:
    - 1:M relationship between *Invoices* and *Customers* – each of the *Invoices* (1) must have at most 1 of the *Customers* (M), but each of the *Customers* can potentially have multiple invoices, FK in *Invoices*
- *Employees*: Records information regarding the employees within the database for personnel records
    - *employee\_ID*: int, auto\_increment, unique, not NULL, PK
    - *employee\_FirstName*: varchar, not NULL
    - *employee\_LastName*: varchar, not NULL
    - *employee\_StreetAddress*: varchar, not NULL
    - *employee\_EmailAddress*: varchar, not NULL
    - *employeePhoneNumber*: varchar, not NULL
    - *relationships*:
      - 1:M relationship between *Invoices* and *Employees* – each of the *Invoices*

(1) must have at most 1 of the *Employees* (M), but each of the *Employees* can potentially have multiple invoices, FK in *Invoices*

- *Invoices*: Transaction documentation using data compiled and imported from the other data tables
  - *invoice\_ID*: int, auto\_increment, unique, not NULL, PK
  - *customer\_ID*: int, unique, not NULL, FK
  - *employee\_ID*: int, unique, not NULL, FK
  - *quantity*: int, not NULL
  - *model\_ID*: int, unique, not NULL, FK

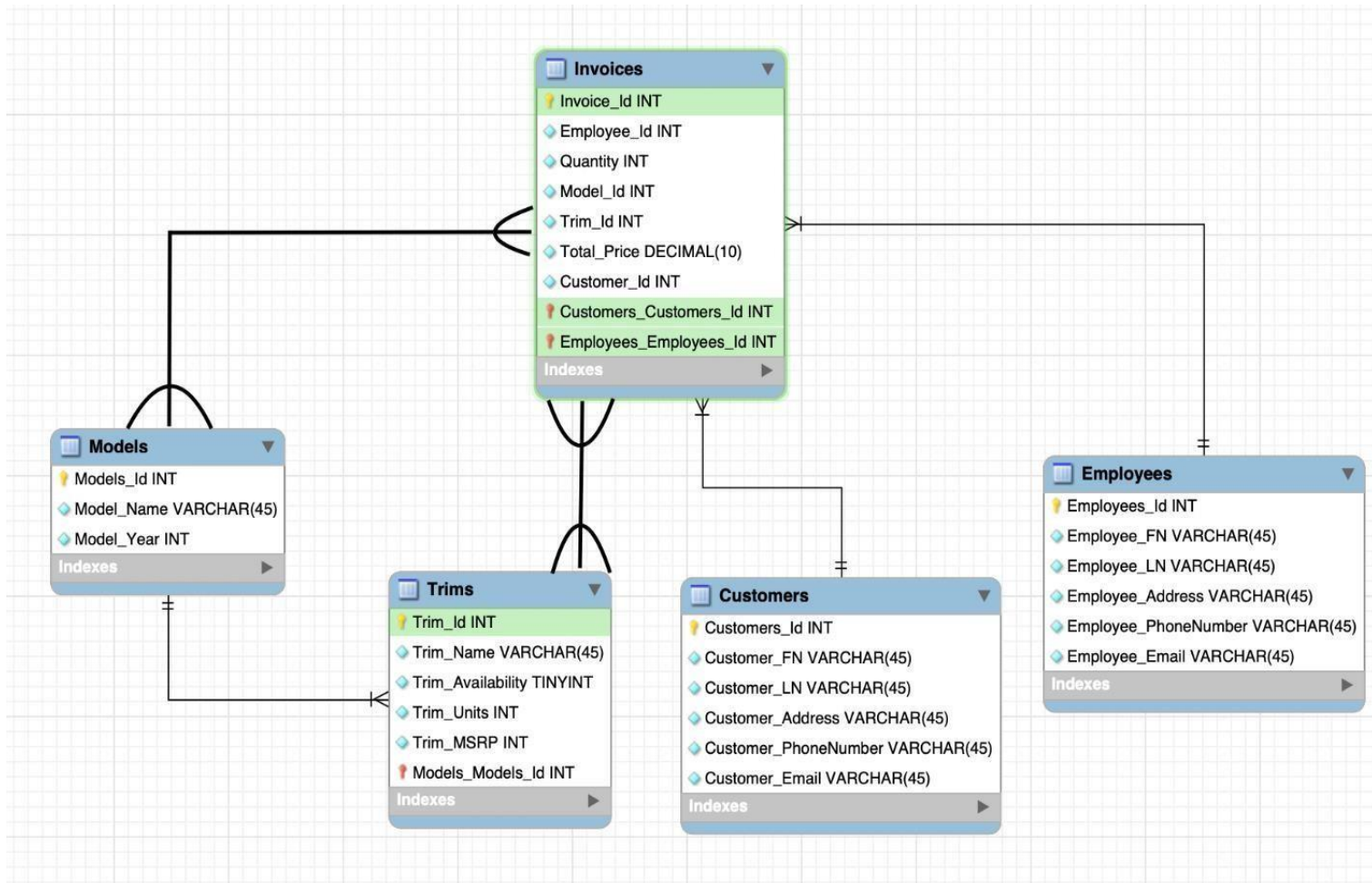


- *trim\_ID*: int, unique, not NULL, FK
- *total\_Price*: decimal, not NULL
- *relationships*:
  - 1:M relationship between *Invoices* and *Customers* – each of the *Invoices* (1) must have at most 1 of the *Customers* (M), but each of the *Customers* can potentially have multiple invoices, FK in *Invoices*
  - 1:M relationship between *Invoices* and *Employees* – each of the *Invoices* (1) must have at most 1 of the *Employees* (M), but each of the *Employees* can potentially have multiple invoices
  - M:N relationship between *Invoices* (M) and *Models* (M) with *Models* as a FK in *Invoices* – many models can be on a single invoice and many invoices may have the same model
  - M:N relationship between *Invoices* (M) and *Trims* (M) with *Trims* as a FK in *Invoices* – many trim levels can be on a single invoice and many invoices may have the same trim level
- *Inventory*: records item availability information and provides location of available units \
  - *available\_UnitsOnSite*: tinyurl
  - *available\_UnitsForOrder*: tinyurl
  - *relationships*:
    - 1:1 relationship between *Inventories* and *Trims* – each trim has a certain availability, and each availability is based on a single trim

**Added edits for Step 2:**

Made all cases consistent as per instructor feedback

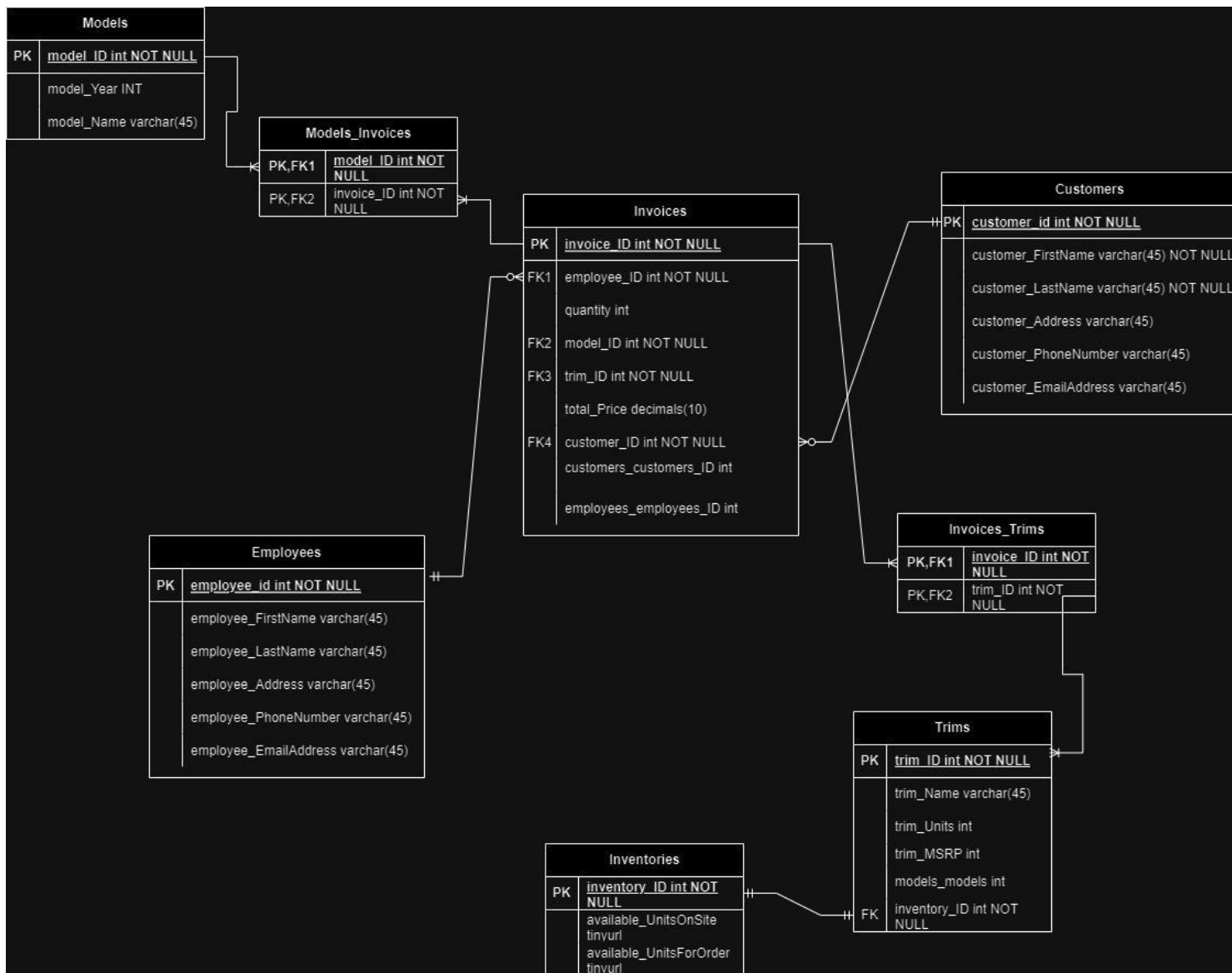
## ERD Diagram (1):



**ERD Diagram (revised with feedback):**



### ERD Schema:



**Data input:**

invoice_ID	employee_ID	customer_ID	model_ID	trim_ID	quantity	total_Price
1	207	100	5002	2004	1	64,000
2	206	101	5001	1004	1	65,000
3	205	100	5003	3002	1	52,000
4	204	103	5002	2001	1	46,000
5	203	104	5001	1004	1	67,000
6	202	102	5001	1003	1	52,000

7	201	106	5001	1004	1	64,500
employee_ID	employee_FirstName	employee_LastName	employee_Address	employee_PhoneNumber	employee_EmailAddress	
103	Juan	Netflix	2740 Kelley Road	405-622-5872	JuanNetflix@email.com	
105	Jose	Hulu	4964 Carter Street	318-265-0275	JoseHulu@email.com	
107	James	Prime	1939 Ritter Avenue	405-622-5873	JamesPrime@email.com	
109	Catherine	Yoda	1066 Davis Lane	318-265-0276	CatherineYoda@email.com	
111	Kenneth	Lights	4830 Straford Park	405-622-5874	KennethLights@email.com	
210	Tom	Sony	619 Tator Patch Road	318-265-0277	TomSony@email.com	
520	Jennifer	Microsoft	3233 Ridenour Street	405-622-5875	JenniferMicrosoft@email.com	
900	Alice	Apple	478 Lakeland Terrace	318-265-0278	AliceApple@email.com	
730	M	Valentine		405-622-5876	MValentine@email.com	
777	Q	Gnome	3573 Hott Street	318-265-0279	QGnome@email.com	
customer_ID	customer_FirstName	customer_LastName	customer_Address	customer_PhoneNumber	customer_EmailAddress	
520	Jennifer	Microsoft	2740 Kelley Road	318-265-0275	JenniferMicrosoft@email.com	
103	Juan	Netflix	4964 Carter Street	405-622-5875	JuanNetflix@email.com	



777	Q	Gnome	1939 Ritter Avenue	318-265-0278	QGnome@email.com	
900	Alice	Apple	1066 Davis Lane	405-622-5876	AliceApple@email.com	
109	Catherine	Yoda	4830 Straford Park	318-265-0276	CatherineYoda@email.com	
107	James	Prime	619 Tator Patch Road	405-622-5872	JamesPrime@email.com	
105	Jose	Hulu	3233 Ridenour Street	405-622-5874	JoseHulu@email.com	
730	M	Valentine	478 Lakeland Terrace	318-265-0279	MValentine@email.com	
111	Kenneth	Lights	4846 Polk Stre	405-622-5873	KennethLights@email.com	
210	Tom	Sony	3573 Hott Street	318-265-0277	TomSony@email.com	
model_ID	model_Year	model_Name				
1007003	2024	Giulia				
1007002	2024	Stelvio				
1007001	2024	Tonale				
1007000	2023	Giulia				
1006999	2023	Stelvio				
1006998	2023	Tonale				
1006997	2022	Giulia				

1006996	2022	Stelvio				
1006995	2022	Tonale				
1006994	2021	Giulia				
1006993	2021	Stelvio				
1006992	2021	Tonale				
trim_ID	trim_Name	trim_Units	trim_MSRP	model_ID	inventory_ID	
70	Sprint	22	43,845	1007001	999998	
69	Ti	20	46,500	1007001	999997	
68	Veloce	20	51,040	1007001	999996	
inventory_ID	available_UnitsOnSite	available_UnitsForOrder				
999998	22	2				
999997	20	2				
999996	20	2				
999995	22	4				
999994	25	4				
999993	23	3				

999992	21	3				
999991	27	3				
999990	24	3				
999989	25	2				
999988	27	3				
999987	21	4				
999986	24	4				
999985	26	4				
999984	23	1				
999983	26	2				

## **Peer Reviews:**

### **Andrew Boland**

- First some comments -
  - I found that CREATE OR REPLACE TABLE ... would help me avoid that issue of needing to wipe out the database every time. I like your solution, though.

- Using the "`" to escape each of your SQL definitions, I think, was a good idea. I read some stack overflow chats and there were folks discussing the appropriateness of such a strategy and how they looked down on it, but our database had 3 Table names using reserved words. Lucky us, I hadn't used the ` key to escape them, and it caused a lot of debugging.
- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
  - Nice Schema. Follows logic well, look at your naming conventions. Some attributes are capitalized and some aren't.
- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
  - EmployeePhoneNumber --> Employee\_Phone\_Number? I think that is the only inconsistent thing I spotted.
  - Intersection tables naming different in schema.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
  - Good layout! Latest image has lower resolution for some reason. Not a spot to dock points in any sense as I can squint but idk if you can fix that easily.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
  - I think if you use MySQL and have it create those intersection tables it will insert that 'has' into the table name. I see it in your SQL and not in your schema
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
  - In terms of normalcy, looks great! I don't see any issues.

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
  - Invoices\_has\_Trims` (errno: 150 "Foreign key constraint is incorrectly formed"), result of my attempt.
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
  - I looked, datatypes seem good. Accurate to their purpose. Nice.
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
  - I can honestly say, I don't know how to implement these atm. I also don't see any cascade, so no?
- In the SQL, are relationship tables present when compared to the ERD/Schema?
  - I see the corresponding tables, but as I said, an inconsistency between Schema and SQL here.
- In the SQL, is all example data shown in the PDF INSERTED?
  - Example data appears all to be present in the INSERT statements
- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
  - I'd go over your SQL some more, the inserted 'has' in the data tables has me thinking some of this may be from SQL and may not be exactly what you intend in your design.

**Ethan Shaw**

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

- - Yes, follows logic from above
- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
- No intersection tables in outline if those are required
- Consistent
- Capitalization for invoice\_ID inconsistent in ERD and schema
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
  - - Two line pairs crossing in revised ERD and one pair crossing in schema. One line in the schema (bottom right) is inside the table itself.  
Beyond these lines the positioning makes sense.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
  - - Yes!
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
  - - Looks normalized!

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
  - - Yes! Worked perfectly after taking out the tables drops in the beginning, instead using the “CREATE OR REPLACE” for tables would eliminate this
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
  - - Yes, all seem fitting and appropriate
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
  - - Yes
- In the SQL, are relationship tables present when compared to the ERD/Schema?
  - - Yes, relationship tables are present
- In the SQL, is all example data shown in the PDF INSERTED?
  - - Yes all example data from PDF is in the SQL file

- Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
  - - Appears to be hand authored although I've not seen “ ON DELETE NO ACTION” and “ON UPDATE NO ACTION” used before.

**Eric Kwak:**

- **Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**
  - The schema presents a good physical model that is consistent with the database outline and the ER. Some of the variable names however have capitalization while the same objects representing the Er diagram do not.
- **Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**
  - In the newly updated ER table as well as the schema, there is consistency between the variable names and tables. I didn't see any errors in naming.
- **Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**
  - The schema is easy to read and follow.
- **Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**



- Yes, it looks like the invoices\_Trims and Models\_invoices tables give the M:N relationship between their respective tables.
- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**
  - The sample data does not give any suggestion of non-normalized issues.
- **Is the SQL file syntactically correct? This can be easily verified by using PhpMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)**
  - When running the SQL file, I did not see any immediate syntax errors.
- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**
  - Datatypes are consistent with their outlined description and function
- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**
  - Both primary and foreign keys are correctly defined when comparing the schema and SQL file, and operations are declared.
- **In the SQL, are relationship tables present when compared to the ERD/Schema?**
  - Yes, relationship tables are present when compared
- **In the SQL, is all example data shown in the PDF INSERTED?**
  - Yes, all data in Inserted
- **Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?**

- There seems to be a lack of side comments in some of the commands, Looks hand-authored overall but I am unfamiliar with the "ON DELETE NO ACTION  
  
ON UPDATE NO ACTION" commands. Good job overall though, this looks good

**William Demsar:**

**·Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**

The ERD should contain just the entities and their PKs and no intersection tables. The schema should contain more detail. Now there are just two versions of a schema.

The entities in the schema often do not reflect all of the attribute constraints that are listed in the outline (e.g. Models and Employees).

I don't understand how the schema shows a relation between Trims and Models.

Attributes and constraints of Trims differ between the schema and the outline. trim\_Availability is on the outline, but not in the schema. Inventory\_ID is in the schema but not on the outline. Models\_models is in the schema, but not on the outline.

I think the line connecting Invoices and Invoices\_Trims is backwards. The lines don't always show cardinality and optionality correctly.

I don't understand what customers\_customers\_ID is in Invoices. There is already a customer\_ID. The modeling of the relationship between Invoices and models is incorrect. It should go M:1 from each main table to the intersection table.

I don't understand the relation between Trims and Inventory. It seems like the availability attributes in Inventory should just be attributes in Trims. More generally, I think you should rethink the Trims, Inventory and Models entities and their relations. It seems like it would be better to have Invoices (M:1)>-Inventory (1:M)-<Models (1:M)-<Trims. An inventory item may relate to many models and each model may relate to many trims. Then you might work it out so that there's a cost associated at each level that flows through to the invoice.

**·Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

Trim is plural in the schema, but sometimes singular in the outline (see Models).

The naming conventions for the attributes in Customers and Employees are different between the outline and the schema.

In Employees, the attribute is employee\_id. In Invoices, it is employees\_employees\_ID

**·Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**

It will probably be easier to read if you copy the schema as an image and paste it, rather than clipping it.

**·Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**

With M:N relationships, the intersection table receives the 1 and the tables being connected have the many, so that each table being connected stands in a M:1 relation with the intersection table. They stand in M:M relation to each other, but this relation is mediated (on both sides) by a M:1 relation with the intersection table.

· **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**

No, but I think the main thing to watch out for here is the way that prices are associated.

· **Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)**

I didn't have any trouble running the file. Everything seems to function. The use of DROP instead of DROP IF EXISTS is unusual, but I can't think of why it should be a problem. Just make sure that there is nothing untoward about this.

· **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**

Yes, but there are some naming differences., e.g. Models\_has\_Invoices and Invoices\_has\_Trims vs Models\_Invoices and Invoices\_Trims in the schema.

· **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**

There are no CASCADE operations defined. It would probably be good to have some. If you delete a model, you probably want to delete all of the trims associated with it, for example.

· **In the SQL, are relationship tables present when compared to the ERD/Schema?**

They are present, but some of them are named differently.

· **In the SQL, is all example data shown in the PDF INSERTED?**

The data is mostly the same, but not always, the model numbers are different, and some of the data appears in a different order. Trim\_Units is different for Veloce. There's no data for the intersection tables, but it's also not clear that data is offered for them in the PDF.

· **Is the SQL well structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?**

I think there are some things that can be improved:

INDEX `fk\_Trims\_Models1\_idx` (`Models\_models` ASC) VISIBLE,

INDEX is visible by default, and to change VISIBILITY it's used with 'ALTER INDEX'

[MySQL :: MySQL 8.0 Reference Manual :: 10.3.12 Invisible Indexes](#)

There's a misspelling in line 49:

UNIQUE INDEX `**Invoice**\_Id\_UNIQUE` (`Invoice\_Id` ASC) VISIBLE,

I didn't understand these comments in the CREATE Invoices section:

-- UNIQUE INDEX `Trim\_Id\_UNIQUE` (`Trim\_Id` ASC) VISIBLE,

-- UNIQUE INDEX `Employee\_Id\_UNIQUE` (`Employee\_Id` ASC) VISIBLE,

-- UNIQUE INDEX `Customer\_Id\_UNIQUE` (`Customer\_Id` ASC) VISIBLE,

There are no corresponding INDEX statements.

There are some places where one line can be written instead of two. For example, under CREATE Employees:

```
`Employee_Id` INT NOT NULL PRIMARY KEY,
```

can be written instead of

```
`Employee_Id` INT NOT NULL,
```

```
PRIMARY KEY (`Employee_Id`));
```

I suppose this is a matter of taste.

Are you sure you want all of these “ON DELETE NO ACTION” statements? Wouldn’t you want DELETE CASCADE in some of the areas where you have ON DELETE NO ACTION? If an Inventory item or model were deleted, then wouldn’t you want to delete the corresponding trims?

There are a few things to clean up here, but this is an interesting project, and you are off to a good start

### **Added Edits for Step 3:**

- Changed customer and employee email labels in the outline to reflect their descriptions in the schema and SQL
- Changed customer and employee first and last name labels to be descriptive and not abbreviations
- Went through the SQL to make things more consistent in terms of naming and capitalization
- Resized schema and corrected some typos

### **Step 3 Feedback:**

*1*

- *Does the UI utilize a SELECT for every table in the schema?* In other words, data from each table in the schema should be displayed on the UI.

Note: it is generally not acceptable for just a single query to join all tables and displays them.

Most tables utilize a select in the UI, however the intersection tables Models\_Invoices and Invoices\_Trims do not provide UI pages or selects. This for example would be selecting all the data in either, or selecting only a certain Trim, Model, etc. as a search function of sorts.

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

If this means using the WHERE along with user variables to be filled in later, then yes it has more than one.

- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.

Aside from the intersection tables mentioned before, there seems to be a button for each table to that would use the INSERT, yet html forms/ UI is not present for any New/Edit/Delete function for entities.

- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line\_total).

No, the SQL code does not INSERT into intersection tables for any of the connected tables.

- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

There are three simple deletes for employees, customers and trims, but they do not delete corresponding rows in intersections.



- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, the SQL satisfies this requirement.

- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

I see in the diagrams that there are optional signs, but all FKs are set to NOT NULL. I may be mistaken, but I don't think there is an actually nullable relationship.

- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*

As is, I think its pretty good. I might suggest to split it up into multiple pages, the assignment page gives an example where for a single page/table the “Show all” shows the Insert, Delete, Update forms for the selected table, and the toggling seems to work just as your page is showing all tables right now. That's all I think I could offer, So great job!

- *Does the UI utilize a SELECT for every table in the schema?*
  - There are SELECTS for every table **except the intersection table**. I do not see those tables on any of the HTML forms.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
  - **Yes looking at the DML there are multiple.**
- *Does the UI implement an INSERT for every table in the schema?*
  - There is an insert for every table in the schema **except the intersection tables. Those should be added.**
- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?*
  - Since there are no intersection tables in here or in the DML queries, **they need to add intersection and the associated queries.**
- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?*
  - **There are no queries related to the intersection table with the DML queries and there is no DELETE cascade in those cases for the DDL queries so those will need to be added. Other DELETE's are good.**
- *Is there at least one UPDATE for any one entity?*
  - **There are updates for every displayed entity. That looks good.**
- *Is at least one relationship NULLable?*
  - **As far as I can see on their DDL sheet, every entity attribute has NOT NULL as part of the query so something will need to be changed as to have the option to be NULL.**
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*

- Everything looks good! Overall the design and ideas are great, I think the main thing is that intersection tables and the associated queries need to be added.

### 3

- *Does the UI utilize a SELECT for every table in the schema?* In other words, data from each table in the schema should be displayed on the UI.

Note: it is generally not acceptable for just a single query to join all tables and displays them.

- I do not believe there is a select for Models\_Invoices. Every other one appears to be present. I'm a little unclear what is being displayed in the Inventories table. It may be necessary to do a JOIN here to display the trim name alongside the inventory info. This is based on the UI, the selects are not in the DML file at this point.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
  - A few of the tables will have a search function.
- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.
  - Every UI has insert available.

- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line\_total).
  - The insert statements do not seem to currently account for the appropriate foreign IDs, for instance when inserting an Invoice, there is no mention of the employee ID or customer ID.
- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.
  - There are three DELETE queries, but not one for any of the intersection tables.
- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
  - There are updates for a few of the entities, allowing updates of a single record.
- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
  - It appears that every relationship is NOT NULL currently, according to the schema.
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*
  - It may make sense to use JOINS to display the names of employees, customers, models and trims on the invoice, rather than the IDs.

# 4

- ***Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.***
  - There is a SELECT \* from Customers, Models, Invoices, and Employees. There does not seem to be a SELECT for Trims and Inventories. There aren't any INNER JOINS, which means that the Intersection Tables don't utilize a SELECT.
- ***Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?***
  - The SELECT filter utilizes a search/filter using one attribute to dynamically populate a list of properties.
- ***Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.***
  - Aside from the Intersection Tables missing the SELECT function, each table has an INSERT, but NEW/EDIT/DELETE functions don't seem to show up in html.
- ***Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line\_total).***

- There does not seem to be any INSERT that applies to either intersection table, therefore the M:M relationship criteria hasn't been met and corresponding FK attributes for these tables.
- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.*
  - There appears to be DELETE for Customers, Employees, and Trims.
- *Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?*
  - It seems that there are several UPDATES present for many entities.
- *Is at least one relationship Nullable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.*
  - There does not seem to be at least one relationship that is Nullable, all FKs are set to NOT NULL
- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*
  - Great job with the concept! Having two ERD schemas is a bit confusing; perhaps consolidate the schema with the intersection tables with the schema without the intersection tables into one table. On a side note, for easier readability, adding a few newlines might make easier so that there's less scrolling from side-to-side.

### **Edits Made Based on Feedback:**

- Based on the feedback, we changed the FK's so that they are no longer NOT NULL and so, accordingly, can be changed to null
- We changed the CASCADE functionality so that deletion would remove the corresponding rows as well
- We edited the names of the intersection tables to remain consistent with the schema
- We enabled insertion into and selection from the intersection tables
- We separated the pages to allow for easier navigation
- Added inner joins for ease of FK usage

## STEP 4 Feedback:

Eric Tran[5d](#)

Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?

Creating a customer works perfectly! I was able to assign a customer ID, first name, last name, address, phone number, and email into the textboxes, and after pressing Submit, the details of the new customer appeared in the list above. Since customer ID is an auto incremented attribute of the Customers entity, I would assume that manually assigning this attribute won't be necessary, so when a user needs to enter a customer, this option will not be visible. I realize you may have had it for testing, and if that's the case, then no worries!

Updating and Deleting both have limited functionality, as you have mentioned. Drop down menus in the Update form look great. The customer names were populated but the customer IDs were not. Do you plan to have customer IDs populated with respect to the chosen customer name? For instance, if there were multiple "John Doe", the customer ID drop down would have a list of respective "John Doe" IDs to select from?

Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.

The UI is simplistic in organization and detail, which is nice. I would add some CSS styling to make it stand out from a "template" style, but the layout is organized and readable. A user should easily be able to use the provided user interface. I draw attention back to the Update form, as it lacks the textbox fields that could be updated by the user.

What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit. If the team is not blocked or having difficulty encouraging and supportive comments would be a better response than NO feedback.

The Read and Create steps from CRUD are working super well! I'd pay attention to the unnecessary customer ID field in the Create form, and the two dropdown menus in the Update form (I feel like only one is needed). Once you get the Customers page implemented with CRUD, implementing every other page should be a piece of cake :) Nice work!

Reply

E

Eric Horn[6d](#)



Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?

Per the author's comments, adding a new customer (CREATE) is the only functionality they've gotten to work at this point. When I attempt to add a customer, I receive a "Bad Request" error 400. Looking at Dev Tools > Network, it appears the issue spawns from the add-person-form. After troubleshooting, this is because the Customer ID field is present and should not be; adding data to this field causes a bad request error.

I attempted to DELETE a customer, and it does not work, as stated by the author. The UPDATE field functionality is missing the ability to select a customer\_ID, so I didn't attempt that either.

Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.

- The UI mirrors that of the node-starter-app, which is fine.
- I like the DELETE button to the right of each customer.
- I would update the HTML to reflect human-readable field names, rather than the database attribute names (e.g. customer\_FirstName => First Name, customer\_EmailAddress => Email, etc.).
- When adding data using an HTML form, the Customer ID field needs to be removed. That attribute will be auto generated by the database, and adding data into the field causes the website to crash.
- When updating a customers information, only one of the fields should be a select box. It doesn't make sense to use two of them. I would consider selecting the customer name, and then have a text entry box for another field (like phone number). That way you select which customer you'd like to update, then enter an updated phone number.

What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit. If the team is not blocked or having difficulty encouraging and supportive comments would be a better response than NO feedback.

Good work getting the READ and CREATE steps to work on the back end! I would work on getting the DELETE step to work next.

When troubleshooting DELETE, it appears the issue stems from delete\_customer.js (per Dev Tools > Network tab). This is where the issue is breaking down. If you look in the delete\_customer.js file, you'll see that it's trying to execute "/delete-customer-ajax", but in your app.js file the route is written as "/delete-person-ajax". Both routes should be written as: "/delete-customer-ajax". That should get your DELETE functionality closer to working.

Reply

M

Matthew Villa6d

- Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?
  - It mostly works as expected except for the delete functionality as indicated by the authors here.
  - Also as indicated, there is nothing showing up under customer id for the update functionality.
- Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.
  - The UI gets the job done, the functionality is explained above the options.
  - There is no menu yet which might be helpful to get to the other pages, if other pages are going to be made.
- What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit. If the team is not blocked or having difficulty encouraging and supportive comments would be a better response than NO feedback.
  - Right now you can create a customer with just an ID which might not really make sense, maybe there should be a way to catch this and not allow it. In fact it appears you can create a new entry with non or just one of the fields.
  - Overall it looks simple but effective. once the functionality can be completed, the layout is good and should perform well. Good job!

## Reply

Joseph Moore [5d](#)

- Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?
  - An INSERT of the same customerID is met with a bad request response. Perhaps some check for same customer ID would be necessary to prevent a bad request. Maybe even a pop-up that says the customer ID is already present.
  - The UPDATE seems to load the appropriate Customer Names but the Customer ID dropdown is not populated.
- Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.
  - The UI follows the guide we have posted and can be updated in later steps. It seems getting the functionality to work is the most pressing step. Much of this can be conducted after the functionality is input with CSS
- What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit. If the team is not blocked or having difficulty encouraging and supportive comments would be a better response than NO feedback.

- I don't have suggestions apart from what I'm sure you all already know. I just want to encourage as you are reaching the end. I'm facing similar troubles with my own project and I would like you to know we're mostly all in the same boat pushing to the finish line. Good luck and I hope everything turns out great!

### **Changes made based on feedback:**

- Fixed and implemented delete functionality
- Fixed and implemented update functionality
- Other feedback is still being implemented in conjunction with adding the other necessary features