Aaron Bertell
Owen Williamson

Daimler Systems Trucking Database

URL: http://flip1.engr.oregonstate.edu:65438/

Feedback by the Peer Reviewers:

Step Three:

Peer Review 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.

Data from each table in the schema is definitely displayed on the UI. The UI does indeed seem to use a SELECT for every table in the schema, as even information from the intersection tables is selected in the DQM.sql file.

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

Yes, there's more than one doing so.

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.

It does, insertions are implemented such that there are enough for every table.

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).

Yes, this seems to be the case for both of the intersection tables DeliveryOrders and TruckDrivers.

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 DELETES, of which two remove things from a M:M relationship. I noticed you're not using any CASCADE statements in your DQM.sql file when I was looking through your material with this question in mind. It got me thinking of the following hypothetical: Say you want to delete just one truck that's currently the only truck of yours at some particular facility. Would that end up then deleting the facility ID as well? I'm not sure how valid that observation is given that

I'm just learning SQL myself this summer, but it seems as though it might be worth looking into how deletions might affect child data.

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, there are multiple UPDATE queries.

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 seem to be a couple relationships that are NULLable, for example the current_facility_id of a Truck entity can be NULLable.

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.

Nothing out of the ordinary, I guess just give some CSS a go when you're able to sort out the flip issues. Vary some of the text sizes as well so it's more legible.

Peer Review 2:

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.

Each of the 6 main entities in the schema is represented in the UI by a table with SELECT. However, as of now it does not appear that there is any way to SELECT intersection table data from the UI. For example, it would be useful to be able SELECT data from the TruckDrivers intersection table to check which trucks correspond to a driver and which drivers share a truck.

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

At the moment, I don't see any use of dynamic population for a search/filter. Based on my understanding of the database schema, dynamically populating a search for Drivers or Customers could be a good implementation in the UI.

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.

There is a UI implementation of an INSERT for every main entity in the schema. At the moment the UI does not seem to support INSERT for intersection tables. For example, I don't see any way to associate a driver with a new truck in the UI. However, the team already does have the corresponding sql code in the DMQ.sql file for INSERT into the intersection tables so I'm sure they are planning on adding it.

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 DMQ.sql file does show that FKs are being added during INSERT. However, as stated above there does not appear to be a way to INSERT into M:M relationships in the current UI implementation.

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.

The text for the DELETEs are visible on each page, however the specific UI implementation is not finished yet. Each table currently allows for deletion, so if this implementation is maintained than both TrucksDrivers and DeliveryOrders will represent an M:M deletion.

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 DMQ.sql file contains UPDATE statements for all tables.

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.

Yes, for example in Facilities the customer_id can be set to 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.

It would be a good idea to use "AS aliases" to format the table column names to allow for capitalization and spaces instead of underscores.

Peer Review 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.

Every table in the schema appears to have a representation. Specifically a SELECT for every table. Even the intersection tables have data represented correctly

Aaron Bertell
Owen Williamson

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

Yes, multiple properties have this effect.

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.

Yes every table excellently has nice inputs to insert data into the database

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).

Yes, DeliveryOrders and TruckDrivers tables both represent the inserts between these properly

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.

It appears that some deletes might not be properly cascaded through related tables. I would recommend looking into this.

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 there are updates represented for entities on this project.

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 seem to be a couple of relationships that are NULLable, for example, the current_facility_id of a Truck entity can be NULLable.

Yes, I noticed the facility id that relates trucks and facilities is nullable (not sure if this actually makes sense for the real-world use case this scenario maps to?)

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.

Aaron Bertell
Owen Williamson

All of the SQL stuff looks solid! I would just recommend that you guys spend some time enhancing your styles!

Step Two:

Peer Review 1:

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

Yes it does but I noticed that there is an attribute missing that is seen in the diagram but missing in the outline. This is driver_id under Deliveries. Also, there could be more clarity surrounding how current_facility_id works if the Truck is on the road. In orders, customer_id is missing from the outline. Also, I believe that the late attribute should be Not Null because 3-state booleans are a source of subtle bugs (what is the difference between Null and False?). I'm curious to know what happens when a delivery has a different driver-truck id combination seen in Deliveries than the TruckDrivers combination. Maybe TruckDrivers is the current point in time while Deliveries have a driver_id and truck_id combination for the future/past deliveries? This part of the schema needs more clarity.

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?

There is consistency between all of these. All entities and plural and attributes are singular. There is use of capitalization.

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

The diagram is very easy to read. No relationship lines are crossed and the name choices are clear.

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

Yes, both intersection tables have 2 FKs that facilitate a M:N relationship.

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

I did not detect any 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!)

Yes, it is syntactically correct.

Aaron Bertell
Owen Williamson

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

Yes they are appropriate but I believe that late should be Not Null.

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

Yes, they are correctly defined when compared to the Schema. The CASCADE operations do not appear to be declared.

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

Relationship tables are present when compared to the ERD/Schema.

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

Example data is missing from the PDF.

Peer Review 2:

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

I appreciate that you kept the older ER diagram attached to compare, and show the changed made between drafts. I may be misunderstanding what the schema is, but I don't think I see it in your PDF. I may totally be wrong, but I thought the ER diagram is meant to show relationships, while the schema shows the attributes, and a rough idea of relationships. However, I can see that you've got the attributes in your ER, so I would say all the correct information is there.

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?

The naming conventions are consistent

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

The schema is clear, and readable. There is a lot going on, but you've got a great grasp on it.

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

There are two intersection tables to facilitate the M:M relationships by using delivery orders and truck drivers tables.

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

I believe the normalization is fine. There is a lot of detailed data that needs to be tracked in this database and I can see realistic reasons for each table to have the number of attributes and foreign keys that they have. However, I recommend some heavy testing to ensure there are no possible anomalies given the complexity of the relationships between tables.

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!)

The SQL file loads up perfectly when importing with PhPMyAdmin.

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

The data types are properly chosen for the given attributes.

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

Primary and foreign keys are correctly defined.

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

The relationship tables match up with the ERD/ schema.

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

I do not see insert data in the PDF.

This database can be a tricky, and intricate one. You have right idea with it all. I recommend going to office hours too, just to see if there is anything that you might miss when designing this. Maybe there's a way to simplify a table or relationship? But overall, great work and good luck!

Peer Review 3:

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

The model follows the ERD accurately. I have some comments on the relationships and how they are modeled, but keep in mind that this is just how I am interpreting the data. So use it as you will.

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?

Aaron Bertell
Owen Williamson

All entities are plural, and almost all attributes are singular, with the exception of early_deliveries/late_deliveries. Perhaps these values could be simplified into a single calculated metric called performance? This would possibly allow the object table to be updated less often than it currently is? And maybe spin off a separate table tracking driver performance (outside of the current scope).

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

Tables are well-positioned and lines are not crossed. Some relationships are confusing.

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

I have a few questions about relationships. I understand the difference between orders and deliveries and the two definitely deserve separate tables. However, I'm not sure that facilities need to be tracked in both orders (start, current, and end facilities) and deliveries (start and end facilities). Orders leads me to believe that an order can be routed through multiple facilities – if an order is routed through 4 facilities and the current facilities is the third stop, does the record of passing through the second facility get lost? (For tracking purposes.) Perhaps a DeliveryDetail intersection table would track this better.

And since only a single driver (or team) can drive a given truck on a given leg, perhaps a LegDetail intersection table could be useful in reducing the number of foreign keys (and make it easier to track driver performance).

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

Is the customer a facility owner (part of the business) or a purchaser – in other words, is this a retail operation or wholesale? I guess I'm seeing something like a UPS or FedEx where the Company is shipping to a Customer that made an Order which has to be shipped through Facilities owned by the company or a third-party vendor – unless the facility is creating orders for its own facilities (making multiple facilities a bit more confusing but possible I guess). Or if the facility is tied to the customer then I'm not sure why the Trucks table (an object table) is tracking the facility_id (more transactional). Again, I'd probably suggest a DeliveryLeg table to track movement.

Is the shipping company responsible for Orders? Is this like an Amazon-type service tracking order and delivery, or more UPS/USPS where it's just deliveries? Or is Orders just Delivery invoices? I'm still a little unclear.

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!)

Appears to be.

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

Looks like it.

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

Yes, it looks like it.

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

All relationship tables are present.

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

The sample data is not shown in the pdf, but the INSERTed data is present in the tables.

Peer Review 4:

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

Yes, the schema presents a physical model that follows the database outline and the ER logical diagram.

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?

Yes, there is consistency in the naming between the overview, outline, ER, and schema, with entities being plural, attributes being singular, and it uses consistent capitalization.

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

Yes, the schema is clear and easy to read.

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

Yes, the intersection tables of TruckDrivers and and DeliveryOrders are properly formed.

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

There is one concern I have with weight and volume of Orders and total_volume and total_weight in Deliveries. As I currently understand it, if you were to add an Order, then the total_weight and total_volume of Deliveries wouldn't be updated, and you would have to manually update these attributes, creating an update anomaly.

Aaron Bertell
Owen Williamson

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, the SQL file is syntactically correct.

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

Yes, the attributes have appropriate data types.

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

Primary and foreign keys are correctly defined when compared to the schema. I don't see any CASCADE operations declared though. These are probably necessary at some points. For example, if you delete an order, you would probably want to remove it from DeliveryOrders so it doesn't get shipped out.

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

Yes, all relationship tables are present.

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

There is no sample data in the PDF, but data is INSERTED into the tables.

Step One:

Peer Review 1:

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

Yes, the overview describes the main problem being solved by the database, real time truck location tracking, but does not mention that the database also tracks driver performance with the late_deliveries and early_deliveries attributes in the Drivers table.

Does the overview list specific facts?

Yes, the overview lists specific facts such as defining late as 60 minutes past delivery and that a truck is either late, early, or on time. Additional facts such as number of trucks, drivers, deliveries, facilities, shipments, and customers would provide more context.

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

Yes, six entities are described: trucks, drivers, deliveries, facilities, shipments, and customers. Each entity represents a single idea to be stored as a list.

Aaron Bertell
Owen Williamson

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

Yes, the database outline describes the purpose of each entity and the relationships between the entities. I am a little confused by the word “order” in the Shipments entity description. What is the difference between a shipment and an order? Can a customer order only have one shipment? Also, I think the words “Deliveries” and “Shipments” are too similar to each other for two entities in the same database. Maybe consider renaming the Shipments entity to Orders.

Also, describe what the late_deliveries and early_deliveries attributes in the Drivers table represent and how they are calculated.

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

1:M relationships are correctly formulated. Yes, there are multiple M:M relationships: Shipments and Deliveries, and Trucks and Drivers. The ERD presents a logical view of the database.

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

All entities are plural and all attributes are singular with the exception of late_deliveries and early_deliveries in the drivers table. All entities and attributes are capitalized. My one suggestion is to rename two tables: Delivieries_has_shipments and Trucks_has_drivers. The use of the word “has” does not imply a M:M relationship. I would concatenate both table names and not use a word that could be interpreted as a 1:M relationship: DeliveriesShipments and TrucksDrivers would work.

Peer Review 2:

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

Yes, the problem solved is one of organization of shipments, trucks, and truck drivers. The DB described will make the status of shipments in progress and made in the past available to view by interested parties.

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

Yes, the four entities are drivers, trucks, facilities, shipments, customers, and deliveries

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

Yes, these are all listed for each entity.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

For the most part... although -I found it odd that trucks could have multiple drivers. Technically possible I suppose. -Wasn't sure what facility_owner_id is referring to? Looks like the ERD might have been from an earlier version. The start/end facility makes a little bit more sense in the doc description. -I think it might make more sense to have early and late be status flags on the deliveries, not on the drivers. Or maybe even let the drivers have their own unrelated late/early flag. If you want a list of early/late packages being driven by a driver, you can make a report out of that later with a query. I believe this is a more logical way of doing things? Not sure.

a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming

Everything looks good, save for early/late package status under drivers. See above for my two cents.

Peer Review 3:

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

Yes, this is a very straightforward shipment/delivery tracker DB and I think it will serve Daimler well.

Does the overview list specific facts?

The overview mentions 20,000 customers as a potential challenge to overcome and examines a case of tracking customers. It might be easier to understand the problem set with just the raw data: 20,000 customers, 5 deliveries per customer per year, each delivery has up to 5 shipments with an average of 2, etc. I’d recommend looking at the entities and make sure each of them is paired with a number in the overview. For example: The company owns 50 trucks, 75 drivers, and has 20000 customers who order on average 5 shipments a year which takes an average of 2.3 deliveries per shipment.

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

Yes.

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

Yes, the outline is adequate. The types associated with each of the attributes makes sense and the relationships make sense as well. My dad is a truck driver and I think its cool you put multiple drivers for the trucks because at FEDEX it’s a reward to get your own truck after x miles without any safety issues. I think additional attributes for the Drivers like hourly wage, miles driven, accidents, etc. would be useful.

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

Yes, there are two M:M relationships that go through exchange tables. The 1:M relationships are also done correctly as far as I can tell.

I think the Deliveries entity needs a start time to aid in your math for calculating the late arrivals. And I would recommend adding some sort of buffer for late/early deliveries. For example, make the buffer 5% of the total expected delivery time (expected arrival – departure time) or something like that and then if the delivery arrives within that buffer its considered an on-time delivery. This is something you could expand later as the business grows its shipping area because you'd expect the buffer to be a smaller percentage for longer travel times and a larger percentage for shorter travel times (a 5 min red light would be a large portion of a 25 min delivery, but a short portion of a 25 hour delivery).

I guess in short: I think you need to define "on-time" deliveries under drivers.

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

I'm not sure why, but on the Trucks_has_Drivers table and Deliveries_has_Shipments table, the attributes don't match the same format. I'd address that with changing the format of the truck/driver table to make it match the automatic naming convention that workbench throws in.

Peer Review 4:

Hey team 70. Great job on this assignment. My feedback below will outline anything I found in your draft that I thought could use a second glance.

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

The overview does a good job of explaining the problem that a database would solve in the trucking industry such as real time deliveries and location of truck but there are no attributes in your database outline that track location. I also noticed that you have an expected arrival time but you don't have a departure time in your deliveries entity. By showing the departure time, it will help the company to figure out if the truck is late due to traffic, a delayed departure, or something else.

Does the overview list specific facts?

Kind of. They list the specific example of late being stored as an int for minutes and they state that a company dealing with 20000 customers will have more difficulty updating customer info than a company with 10 customers. I feel like they could have included more specific facts such as how many miles the average trucker covers in a day and how many factories some of the well known trucking companies have.

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

Yes. They have 6 entities in total. I am a little confused on why certain entities have some attributes such as facilities having a customer_id but deliveries does not, but I trust that the team knows why they have the entities and attributes built that way.

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

The outline gives a brief overview of what the entities do without going into specifics about the attributes until the relationships with other attributes. Adding a little more to the entity description so that it also includes a short blurb about the attributes may be very helpful in identifying where duplicate data is created and where attributes need to be moved.

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

Yes, they implement the 1:M relationships well and there are intermediate tables in place to make the M:M relationships. It may have been helpful to see the intermediate tables described in the outline but I am unsure if that is required for this assignment.

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

Yes, they keep a consistent naming scheme in that all entities are plural and they always use “_” to represent spaces in entities and attributes. They also capitalize all entities but never capitalize the attributes. They do not capitalize non-important words in entities, such as "has".

Actions Based On The Feedback

Step 3:

Completed Suggestions:

Added Intersection Tables - Originally we did not have the intersection tables displayed, but as recommended we added the intersection tables and a way to insert update them.

Add Search Filter - We did not have a search filter in the UI, so one was added to customers, although it is not yet functional with javascript so it doesn’t do anything just yet, but the outline for it is there.

Add Cascades - We had not yet added the cascade statements, so the feedback was a good reminder to add that to our DDL

Fix Select Queries - The reviewers pointed out that our select queries found foreign keys instead of useful information like names, so we will need to go back and fix that.

Updating Styles - We did not have very detailed CSS at the time of the draft and many of the reviewers offered great CSS-related suggestions, and we added some and will continue to add more as the exact functionality of our website develops.

Step 2:

Completed Suggestions:

Inserting Sample Data - It looks like we forgot to insert sample data into this document, so after that was pointed out we added it.

Inserting Schema - It looks like the schema is actually different from the ERD, so we constructed and inserted a schema as well.

Consistency Issues - Fixed a few foreign keys and data types that were mismatching across our many diagrams, pointed out by the reviewers.

Ignored Suggestions:

LegDetails - One reviewer suggested that we add a LegDetails entity to attract every location that a given order has been in the order process, however that can indirectly be done through the deliveries and deliveriesOrders themselves, so we don't need to add an extra entity.

Delivery Weight/Volume - Multiple reviews pointed out we may have update errors when trying to log total weight and volume, so we may not keep track of it as a value itself but rather a select query that does a summation.

Step 1:

Completed Suggestions:

Rename Shipments to Orders - Delivery and shipment are very similar in nature and can be kind of confusing, and we agree with the reviewers that naming them orders instead of shipments would make the concept less confusing.

Rename Deleveries_has_Shipments and Trucks_has_drivers - these names were auto generated but we were advised to change them to more descriptive names. We also fixed some formatting issues with these tables that were pointed out.

Make Overview more descriptive - There were a few different pieces of advice we were given on this, firstly including all relevant information the the software will track in the overview, and secondly including details on the size and scale that this database will be operating on.

Properly defining Early/Late Deliveries - Each delivery now also includes a start_time, and we will define a truckload as being late if the actual delivery time is 5% longer than the predicted delivery time.

More context and specific facts to Overview - There were several mentions of people wanting more context and specific facts to the overview, so the previous example with numbers was replaced to include a more holistic look at how the database would work when a customer creates an order that provides that context. Sort of in the way a user story works.

Adding more entity attributes - We were recommended to add attributes that manage order and delivery location and delivery departure time. We also renamed a mislabeled attribute that was in the ER diagram.

Ignored Suggestions:

Confusion in delivery and facility customer_id - Deliveries do not have a customer_id because deliveries can contain multiple orders (previously shipments), and each order can be for a different customer. This is why the customer_id is left on the shipment and not the delivery. Facilities have a customer_id because they can be owned by a customer, but it is optional. Two example facilities might include a warehouse owned by the trucking company, with no cusomter_id, and a store owned by a customer, which would have a customer_id.

Attribute suggestions to drivers - Suggestion to include hourly wage and accidents to drivers was not the issue we were trying to solve with this particular database and seemed outside the scope of our main focus.

Upgrades To The Draft Version

Several additions and subtractions were made to the overview to include more specific detailed facts and a clearer description of the process the table would be used for, so users can better understand the use of the database and the solution it provides. The second paragraph of the original was deleted since it was speaking of databases more broadly and not our specific database which helped add clarity. There were also changes made to attribute names, for example, start_time under deliveries was changed to departure_time to make what it represents more clear. Additional attributes were also added as a result of suggestion provided that would make the database perform closer to expectations. Changes were also made to the table name shipments, where shipments was a little confusing for our readers, we changed it to orders instead. These changes are reflected in the ERD diagram as well.

All of our changes can be seen below:

Overview:

The problem that a backend database will solve at the simplest level is being able to get information about real time deliveries and locations of the trucks whereabouts. It will be able to tell when a truck leaves a facility and when that truck is late, early or on time. It will also provide information about customers and to the customers on their orders. For example, a customer orders an item, that would create one order. There could be many other customers that have orders as well, say there are 100 orders in a single delivery. A truck will then carry this delivery to some facility. These trucks will have several drivers, in that many drivers will sometimes be driving different trucks and there will be many different trucks carrying deliveries to different facilities.

Daimler is a North American truck manufacturing company hoping to provide services that make their trucks both user and driver friendly. Our website provides services for drivers, truck fleet organizers, and customers to make and view deliveries. Our website will allow drivers to record their truckload, called a delivery, and manage their location at various drop off points, called facilities. Customers can order and view their orders, and truck fleet managers can see these orders as well as the locations of all the trucks in the fleet in order to optimize which orders should be placed in which deliveries. The database also calculates late and early delerviers by marking a delivery as late if its actual arrival time is 5% greater than it's predicted arrival time. Early and late deliveries are summed into totals for each driver, which allows fleet managers to monitor the quality of the drivers. Weight and volume of truckloads are calculated making it easy for fleet managers to ensure that all of their trucks are safe to drive. Daimler is not a trucking company but rather a truck manufacturer, and their trucks are bought by companies of all sizes. Therefore, our database needs to be functional both on small scale companies with only a few trucks, and on larger scale companies with thousands of them.

Database Outline:

Trucks - This entity holds information about the trucks that are used to make deliveries.

● truck_id:	int, NN, AI, PK
● max_weight:	int, NN
● max_volume:	int, NN
● current_facility_id:	int, NN, FK

Relationships:

- Drivers (M:M) - Each driver can drive multiple trucks, and multiple drivers can be in a truck at a time
- Deliveries (M:1) - Each truck can make many deliveries, however a single delivery is only made by one truck
- Facilities (M:1) - each truck can be parked at one facility, however a given facility may have multiple trucks.

Drivers - This entity holds information about the drivers used to make deliveries.

● driver_id:	int, NN, AI, PK
● name:	varchar(45), NN
● email:	varchar(45), NN
● late_deliveries:	int, NN, ZF
● early_deliveries:	int, NN, ZF

Relationships:

- Trucks (M:M) - Each driver can drive multiple trucks, and multiple drivers can be in a truck at a time
- Deliveries (M:1) - Each delivery can be made by one driver, but a driver can complete multiple deliveries.

Aaron Bertell
Owen Williamson

Deliveries - This entity holds information about a singular trip a truck and driver make between two facilities.

- delivery_id: int, NN, AI, PK
- total_weight: int, NN
- total_volume: int, NN
- start_facility_id: int, NN, FK
- end_facility_id: int, NN, FK
- departure_time: datetime, NN
- expected_arrival_time: datetime
- actual_arrival_time: datetime
- late: boolean

Relationships:

- Orders (M:M) - Each delivery can hold multiple different orders. Each order may require multiple deliveries.
- Facilities (M:1) - Every delivery has a start and end facility location, but no more. A facility can be a location in many deliveries.
- Trucks (M:1) - Each truck can make many deliveries, however a single delivery is only made by one truck
- Drivers (M:1) - Each delivery can be made by one driver, but a driver can complete multiple deliveries.

Facilities - This entity holds information about a specific facility, which are the locations that trucks drive between.

- facility_id: int, NN, AI, PK
- customer_id: int, FK
- location: varchar(45), NN
- name: varchar(45), NN

Relationships:

- Orders (M:1) - Every order has a start and end facility location but no more. A facility can be a location in many orders.
- Deliveries (M:1) - Every delivery has a start and end facility location, but no more. A facility can be a location in many deliveries.
- Customers (M:1) - Each facility can be owned by a customer, and a customer can own more than one facility.
- Trucks (M:1) - each truck can be parked at one facility, however a given facility may have multiple trucks.

Orders - This entity holds information about a specific order that a customer has made. If current_facility_id is NULL it means the order is en-route.

- order_id: int, NN, AI, PK
- customer_id: int, NN, FK
- weight: int, NN
- volume: int, NN

- start_facility_id: int, NN, FK
- current_faciility_id int, FK
- end_facility_id: int, NN, FK

Relationships:

- Customers (M:1) - Each order is an order by a customer to move something from one location to another.
- Deliveries (M:M) - Each delivery can hold multiple different orders. Each order may require multiple deliveries.
- Facilities (M:1) - Every order has a start and end facility location but no more. A facility can be a location in many orders.

Customers - This entity holds information about the customers that make specific orders.

- customer_id: int, NN, AI, PK
- name: varchar(45), NN
- email: varchar(45), NN

Relationships:

- Orders (M:1) - Each order is an order by a customer to move something from one location to another.
- Facilities (M:1) - Each facility can be owned by a customer, and a customer can own more than one facility.

TruckDrivers - This entity represents a many to many relationship between trucks and drivers.

- truck_id: int, NN, FK
- driver_id: int, NN, FK

Relationships:

- Trucks and Drivers - This Entity maintains one relationship between trucks and drivers, creating the possibility of a many to many relationship.

DeliveryOrders- This entity represents a many to many relationship between deliveries and orders.

- delivery_id: int, NN, FK
- order_id: int, NN, FK

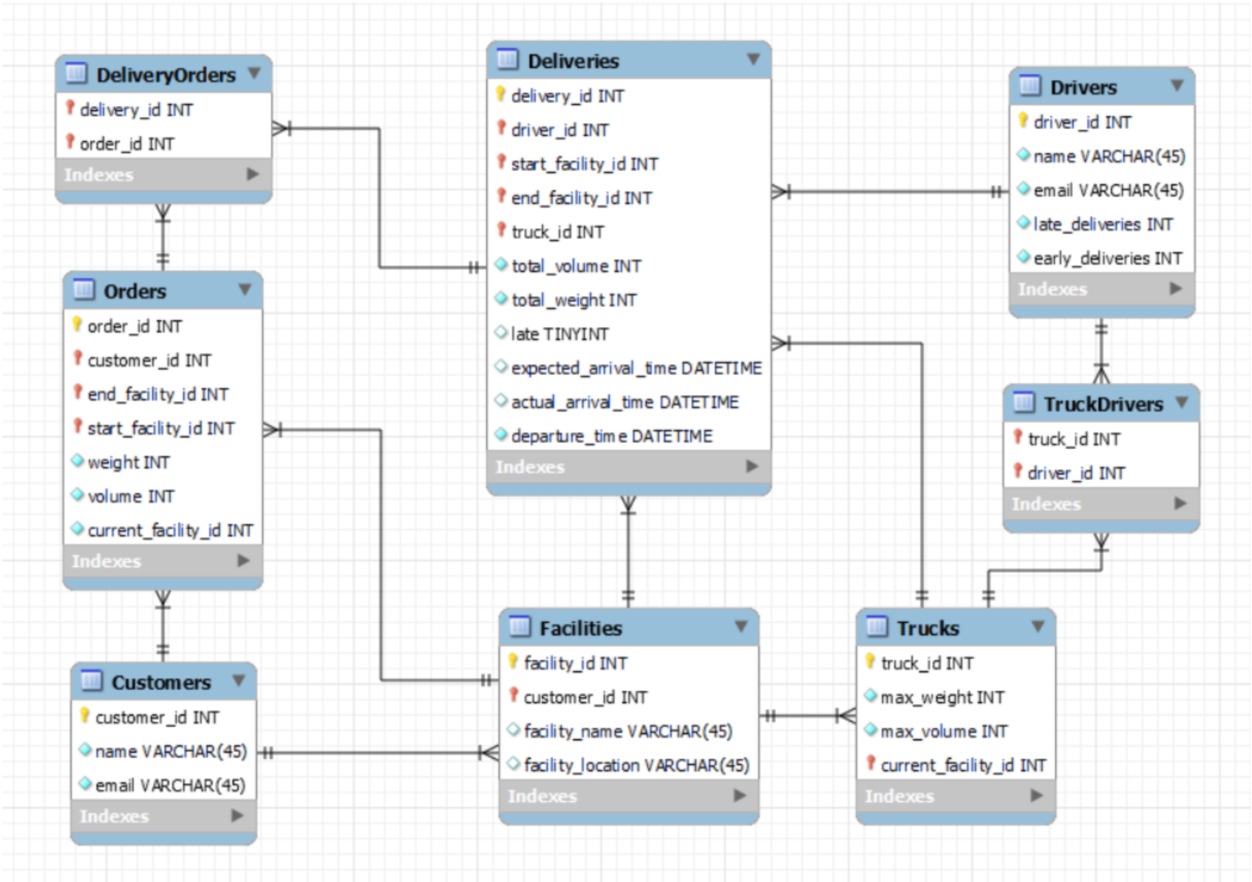
Relationships:

- Deliveries and Orders - This entity maintains one relationship between orders and deliveries, creating the possibility of a many to many relationship.

Entity-Relationship Diagram:

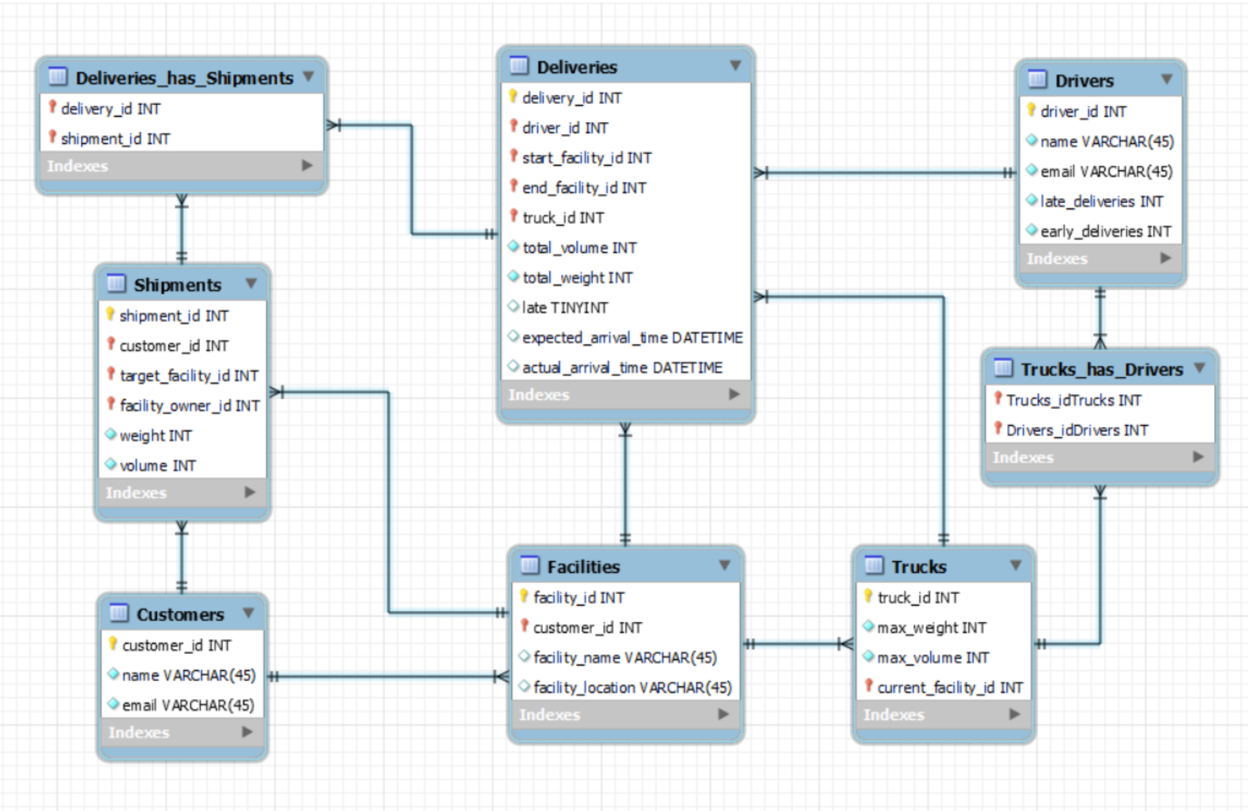
Updated Diagram:

Aaron Bertell
Owen Williamson



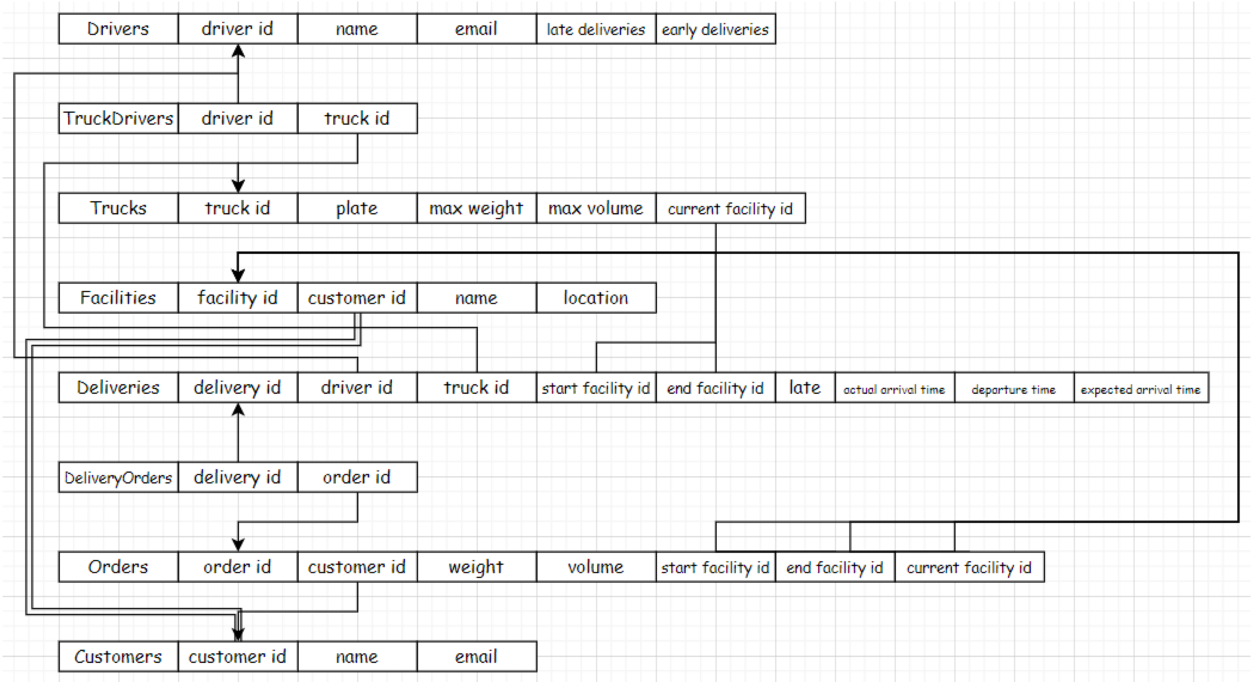
Aaron Bertell
Owen Williamson

Old Diagram:



Aaron Bertell
Owen Williamson

Schema:



Drivers | driver id | name | email | late deliveries | early deliveries

TruckDrivers | driver id | truck id

Trucks | truck id | plate | max weight | max volume | current facility id

Facilities | facility id | customer id | name | location

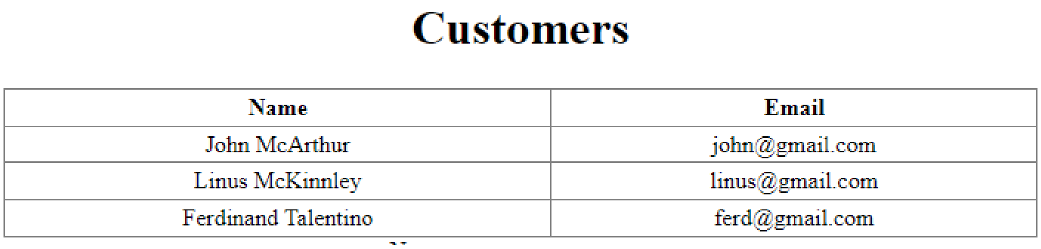
Deliveries | delivery id | driver id | truck id | start facility id | end facility id | late | actual arrival time | departure time | expected arrival time

DeliveryOrders | delivery id | order id

Orders | order id | customer id | weight | volume | start facility id | end facility id | current facility id

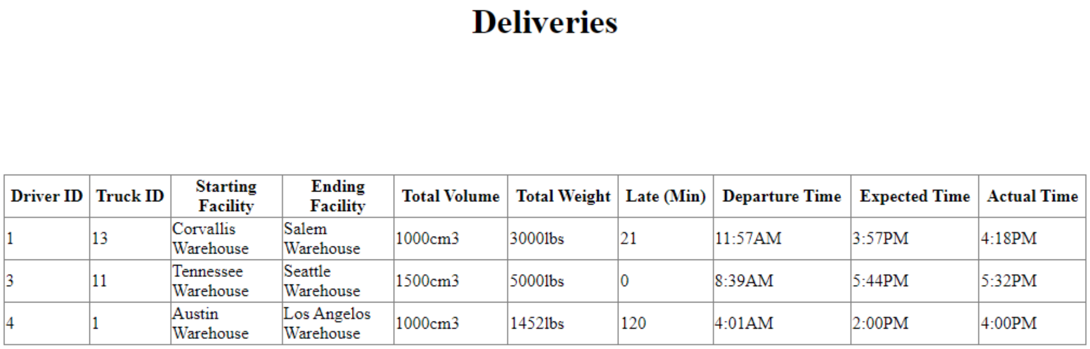
Customers | customer id | name | email

Sample Data:



Customers

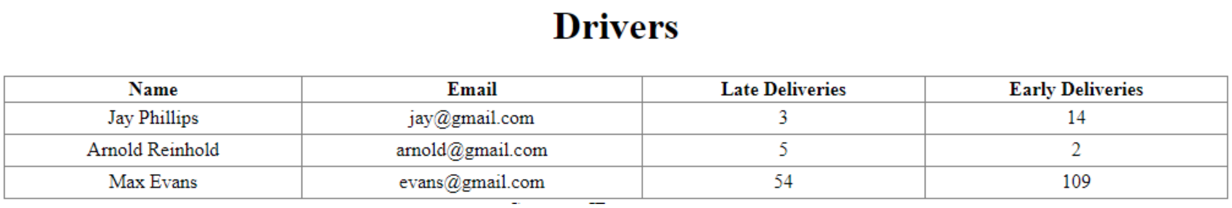
Name	Email
John McArthur	john@gmail.com
Linus McKinnley	linus@gmail.com
Ferdinand Talentino	ferd@gmail.com



Deliveries

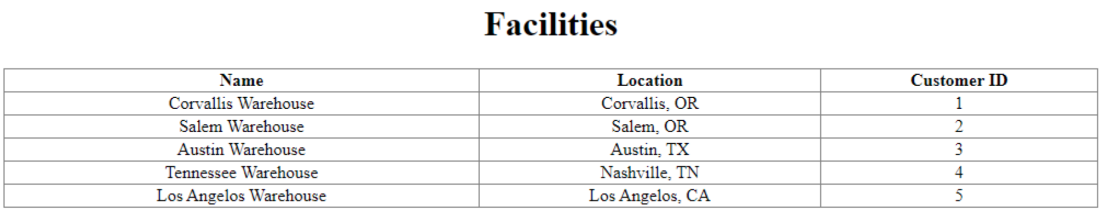
Driver ID	Truck ID	Starting Facility	Ending Facility	Total Volume	Total Weight	Late (Min)	Departure Time	Expected Time	Actual Time
1	13	Corvallis Warehouse	Salem Warehouse	1000cm3	3000lbs	21	11:57AM	3:57PM	4:18PM
3	11	Tennessee Warehouse	Seattle Warehouse	1500cm3	5000lbs	0	8:39AM	5:44PM	5:32PM
4	1	Austin Warehouse	Los Angelos Warehouse	1000cm3	1452lbs	120	4:01AM	2:00PM	4:00PM

Aaron Bertell
Owen Williamson



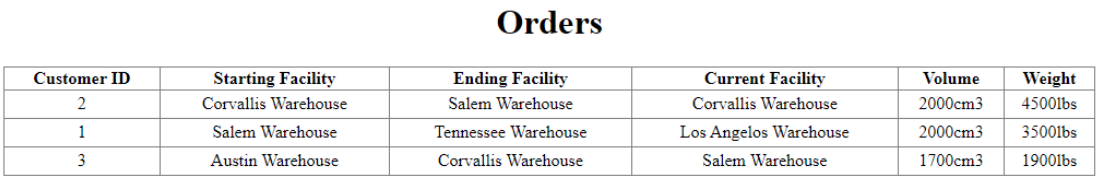
Drivers

Name	Email	Late Deliveries	Early Deliveries
Jay Phillips	jay@gmail.com	3	14
Arnold Reinhold	arnold@gmail.com	5	2
Max Evans	evans@gmail.com	54	109



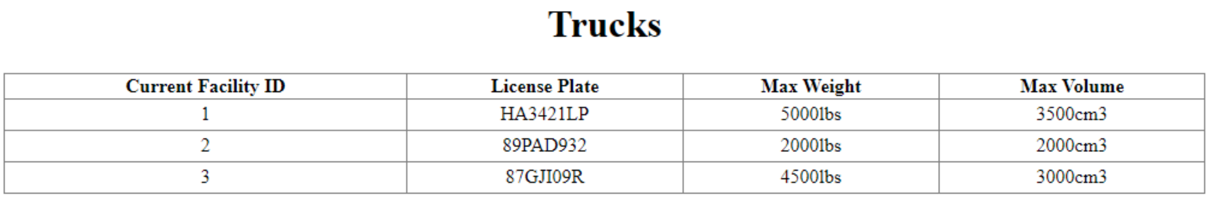
Facilities

Name	Location	Customer ID
Corvallis Warehouse	Corvallis, OR	1
Salem Warehouse	Salem, OR	2
Austin Warehouse	Austin, TX	3
Tennessee Warehouse	Nashville, TN	4
Los Angelos Warehouse	Los Angelos, CA	5



Orders

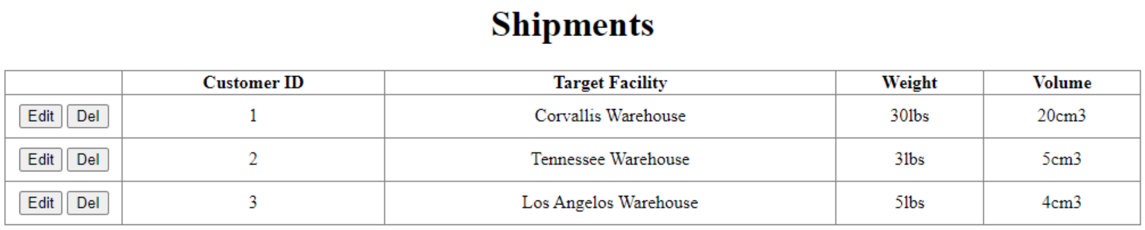
Customer ID	Starting Facility	Ending Facility	Current Facility	Volume	Weight
2	Corvallis Warehouse	Salem Warehouse	Corvallis Warehouse	2000cm3	4500lbs
1	Salem Warehouse	Tennessee Warehouse	Los Angelos Warehouse	2000cm3	3500lbs
3	Austin Warehouse	Corvallis Warehouse	Salem Warehouse	1700cm3	1900lbs



Trucks

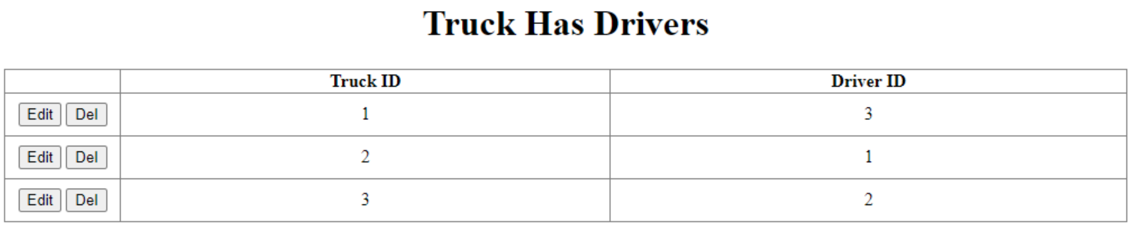
Current Facility ID	License Plate	Max Weight	Max Volume
1	HA3421LP	5000lbs	3500cm3
2	89PAD932	2000lbs	2000cm3
3	87GJI09R	4500lbs	3000cm3

Aaron Bertell
Owen Williamson



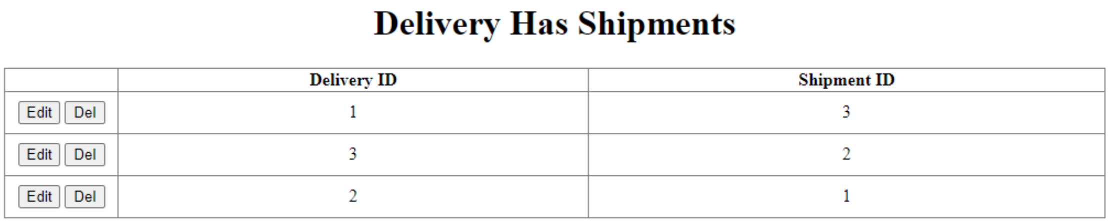
Shipments

	Customer ID	Target Facility	Weight	Volume
Edit Del	1	Corvallis Warehouse	30lbs	20cm3
Edit Del	2	Tennessee Warehouse	3lbs	5cm3
Edit Del	3	Los Angelos Warehouse	5lbs	4cm3



Truck Has Drivers

	Truck ID	Driver ID
Edit Del	1	3
Edit Del	2	1
Edit Del	3	2



Delivery Has Shipments

	Delivery ID	Shipment ID
Edit Del	1	3
Edit Del	3	2
Edit Del	2	1

Document TO DOS:

- ☑ ~~Sample data in document~~
- □ Schema
- ☑ ~~Cascade commands~~
- □ Redo Select queries
- ☑ ~~Step 2 feedback~~
- ☑ ~~Step 3 feedback~~

Website TO DOS:

- ☑ ~~CSS~~