Group 19 — Michael Andrews and Ian Collier Project Step 3

URL for draft HTML

~~www.iancollier.host~~
Temporary location: `http://flip3.engr.oregonstate.edu:8343`

Outline

The project will be a simplified and abstracted view of an automotive suppply chain. The selection of automobiles is arbitrary, and intended simply to make the relationships intuitive for the user. The basic intent is to have a series of relationships that will allow the user to track parts suppliers, parts requirements, and orders in such a way that they can easily determine at any given time what supplies need to be procured, and at what price, to meet orders.

Changes for Step 2

- Which keys are foreign (and which domestic) has been clarified, both in the Outline and in the Schema. (Foreign keys in the schema are now indicated by an asterisk.)
- The Model-Colors relationship has been replaced with a Trimline-Colors relationship, and removed as an entity-in-itself. This suggestion was made in both TA and peer feedback. Additionally, because this is a many-to-many relationship table, that table no longer has its own auto-incrementing key.
- A suggestion was made in group review to add default values for several attributes which cannot be null. That suggestion has not been adopted, since the database is (within its own fictional universe) intended to reflect physical realities; a default value for the "Quantity on Hand" attribute of the "Parts" entity, which was one of the suggestions, could result in the database being automatically populated with inaccurate data. The correct behavior if an attempt is made to add rows without knowing the values for those attributes should be to fail with an error. Similarly, if a customer attempts to order a car but does not specify either a trimline (which implies a model) or a model (which defaults to a trimline,) they should be told they need to pick something.
- The ERD has been embiggened.
- The Base Model attribute of Model has had its name changed to Base Trimline rather than making it recursive; the previous name was a typo.

Changes for Step 3 Draft

- Key and table names have been changed to lower case, underscore separated. Keys named 'key' have been changed to 'id' because MariaDB considers key a reserved word.
- Both attributes in model_colors have had on delete cascade added to them.
- Trimline's default model now cascades on deletion.
- Trimline's default color now cascades to null on deletion.

Review Feedback on Step 3 Draft

Only one peer review had specific questions and changes; it's included below:

- "Your DDL file looks good. I was able to successfully import it into phpmyadmin with no errors or issues. I imported again after tables had been created and your drop table if exists statements also look good! Also the samples data was successfully inserted into the tables.

  "Looking at your HTML I see on your "Body" page that you have a search field. I know it is still early so you will probably make changes, but maybe specify what can be searched there. Only models? You could also throw and update table header and delete table header in your html table.

  "What will the "save changes" buttons do? Will that be a type of 'update' function? If I decide to change trimline and color on the drop downs on the Order page, will hitting save changes update it? Will it update only the ones that I have changed they drop down values or specific ones?

  "One question I have is about colors. The color name is the primary key but how are you handling potential duplicate colors being entered into the table?

  "I don't have any comments on your query file, everything looks good to me. You have all the insert, delete, and update statements needed as well as some really good query statements that I think will work really well with your database. Good job!

  With regard to colors, duplicates are not an issue. Because a primary key needs to be unique, duplicates cannot be entered, and that's intended behavior. A duplicate color would serve no purpose.

  With regard to what the update and save functions do, the intended behavior will be for update to populate a form with the fields of the item, and for save to commit the item back to the database, incorporating any (legal) changes that might have been made. (The auto-incrementing keys will be reported, but will not be alterable.)

  With regard to the search behavior, a new page has been added with checkboxes to allow for searching any or all of Models, Trimlines, Orders, and Parts by name.

1. Changes for Step 3 Final

Dedicated search page added; queries added to support its behavior.

Entities

We will have the following entities:

models. The model represents a basic car model. The intrinsic attributes it needs are:

- id (Primary Key)

  This should be an autoincrementing integer.

- name
  Each Model type needs name; this should be a 255 character max string. It must not be the same as the name of any other Model, or of any Trimline. It cannot be null.
- base_trimline (Foreign Key)
  Each Model must have a Trimline designated as its base version; this field cannot be null and must contain the key of a Trimline which exists in the database. Consequently, this field will be an integer.

The other properties of cars will be simulated through relationships with other entities.

trimlines.

- id (Primary Key)
  Each Trimline needs a key; this can be an autoincrementing integer. Cannot be null.
- name
  Each Trimeline needs a name; this should be 255 character max string, and must not be the same as the name of any Model or any other Trimline. Cannot be null.
- model (Foreign Key)
  Each Trimline must have exactly one Model associated with it; this field will store the key of that Model. It cannot be null. This should cascade on deletion.
- default_color (Foreign Key)
  Each Trimline needs a default color. 30 character-max string, must be the key of a Color. Should cascade to null.

colors.

- name (Primary Key) Each color must have a unique name. This should be a string, thirty characters maximum, must be unique, and cannot be null. This can also serve as the key.

parts. The part entity represents components, by SKU.

- id (Primary Key)
  This should be an auto-incrementing integer.
- name
  This should be a 255 character string field, which does not need to be unique.
- quantity_on_hand
  This should be an integer, and should never be null.
- cost
  This should be an integer, and can be null. (If a part has become unavailable.)

part_requirements. This entity will hold tuples indicating how many of each type of part a given model or trimline requires. (A specific trimline may require additional parts that the underlying model does not; for example, a luxury model might require a fancy GPS while the base model gets an Etch-a-Sketch and a road atlas from 1962.)

- id (Primary Key) This should be an autoincrementing integer.

- associated_model (Foreign Key) This should be the key of the Model to which the tuple applies. Integer, can be null if and only if Associated Trimline is not null, must exist in DB if not null.
- associated_trimline (Foreign Key) This should be the key of the Trimline to which the tuple applies. Integer, can be null if and only if Associated Model is not null, must exist in DB if not null.
- associated_part (Foreign Key)
  This should be the key of the part with which the tuple is associated. Integer, cannot be null, must exist in DB.
- quantity
  This should be the quantity of the part needed for the pairing; must be an integer, must be at least one, cannot be null.

orders. This represents an actual order for a vehicle. We'll abstract the customer details to just a name; the jerks can come pick it up at the factory.

- id This should be an autoincrementing integer; it will represent the order number, so maybe we'll start it off at like 5 million so it looks like we're more successful than we are.
- customer
  The name of the customer; 255 character string, cannot be null, does not need to be unique.
- trimline (Foreign Key)
  The key of a Trimline; cannot be null and must exist in the database.
- color (Foreign Key)
  The color desired by the customer. Thirty character string, cannot be null, must be the key of a Color which has a row in ModelColors corresponding to the Model corresponding to the desired Trimline. (e.g., the customer wants a Trimline called a Thuggee in Blue. The Thuggee is a trimline of Dacoity; ModelColors does not have a row for the Dacoity in Blue. The order is invalid.)

This is already complicated enough, so we won't worry about when the customer wants it by. The jerks can come by the factory every day until we have something for them. Our motto is, "The Customer is Always a Sap."

Relationships

Models have Trimlines. Each model must have at least one Trimline, and can have as many as needed. Each Trimline is associated with one and only one Model. This is a one-to-many relationship.

Models have Colors. Each model must be available in at least one color, but can be available in as many as needed. This is a many-to-many relationship.

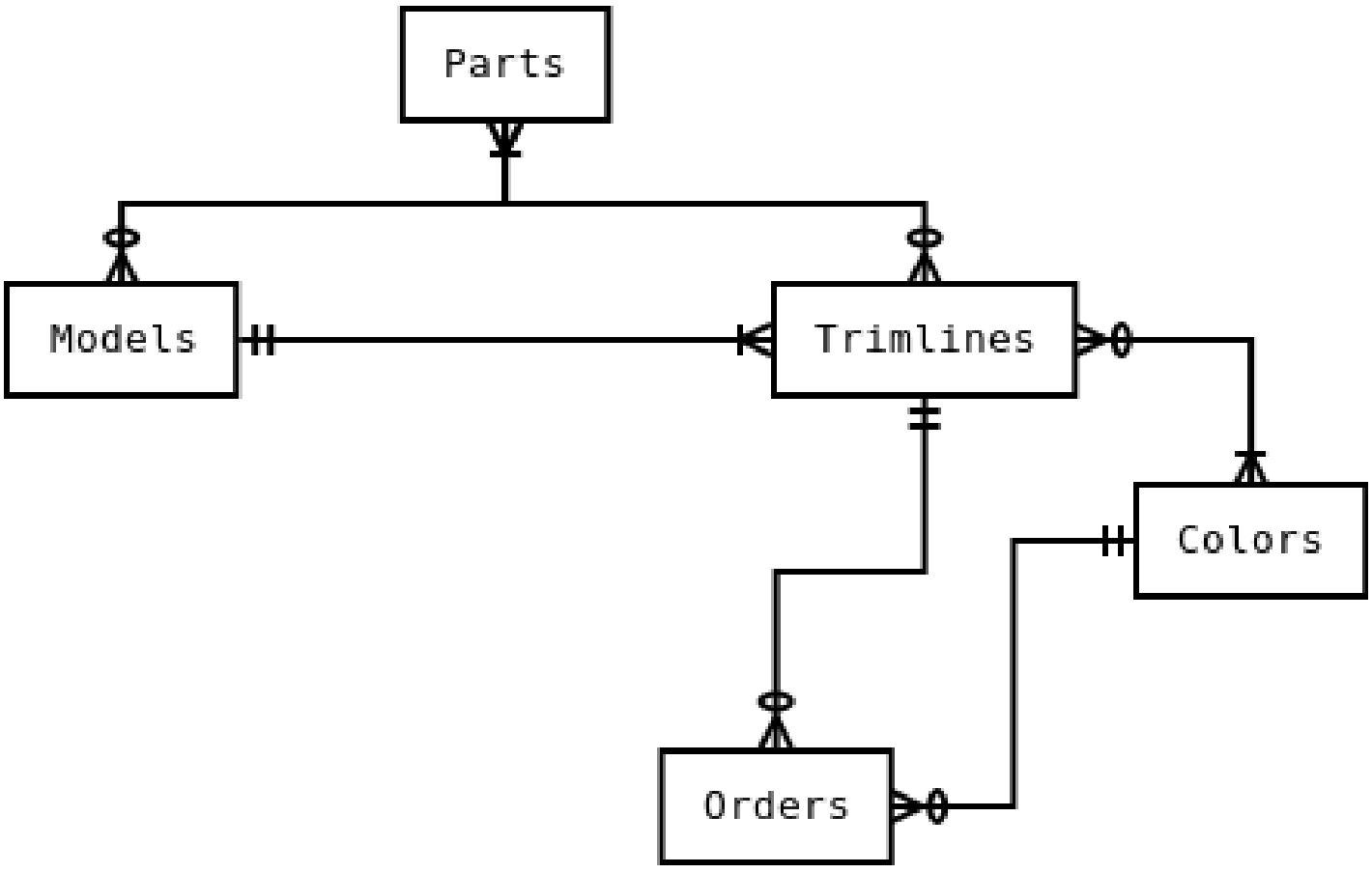
Models need Parts. Each Model needs at least one instance of one part, but may need an arbitrary quantity of an arbitrary number of parts. Any given part can be needed by any number of models, but can be needed by none. This is a many-to-many relationship.

Trimlines need Parts. Each Trimline may need any quantity of any number of parts, but can require none. The parts needed can overlap with those required with the underlying Model of the Trimline. Parts need not be needed by Trimlines, but can be needed by any number of them. This is a many-to-many relationship.

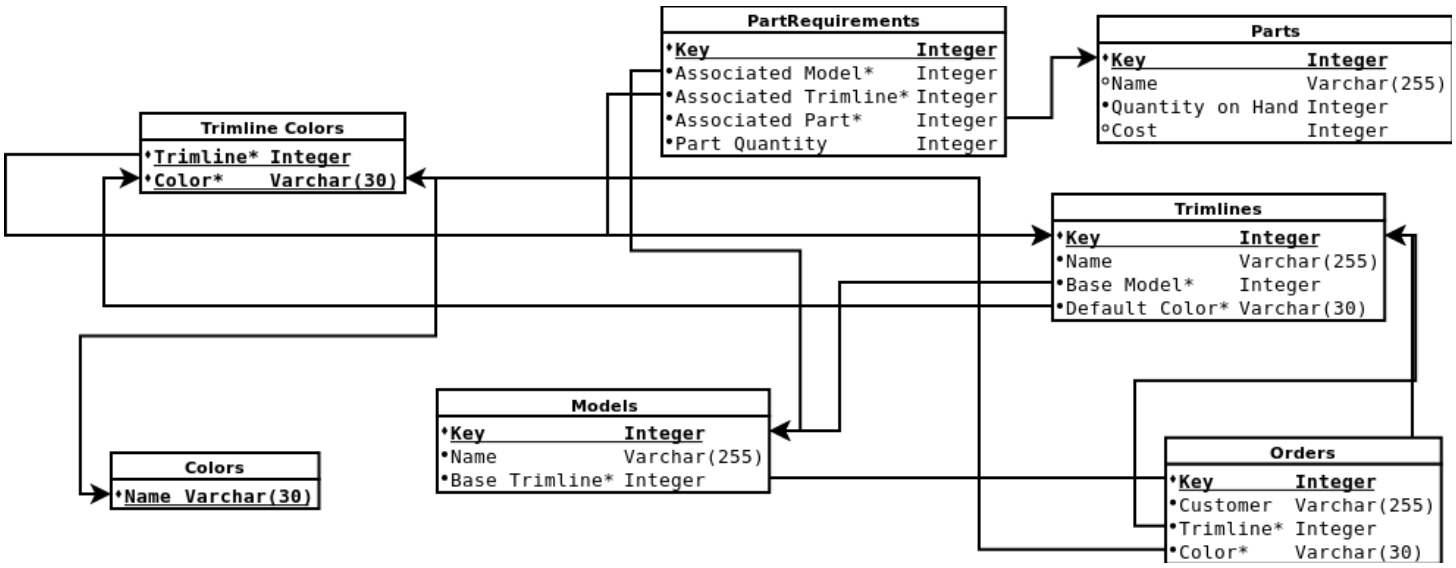
Orders need Trimlines. Each order must have exactly one Trimline specified. Any number of orders can specify the same Trimeline. This is a many-to-one relationship.

Orders need Colors. Each order must specify a Color which must be available for the Model associated with the Trimline specified in the order. An order which specifies an invalid combination should be rejected. A Color, of course, may be associated with any number of orders. This is a many-to-one relationship.

Entity Relationship Diagram



Schema


PartRequirements
Key Integer
Associated Model* Integer
Associated Trimline* Integer
Associated Part* Integer
Part Quantity Integer
Parts
Key Integer
Name Varchar(255)
Quantity on Hand Integer
Cost Integer
Trimline Colors
Trimline* Integer
Color* Varchar(30)
Trimlines
Key Integer
Name Varchar(255)
Base Model* Integer
Default Color* Varchar(30)
Models
Key Integer
Name Varchar(255)
Base Trimline* Integer
Colors
Name Varchar(30)
Orders
Key Integer
Customer Varchar(255)
Trimline* Integer
Color* Varchar(30)