# CS 121 Final Project Reflection

**Due: Wednesday March 19th, 11:59PM PST**

*Note: This specification is ~~currently for 24wi, subject to minor changes for 25wi~~ current for 25wi.*

This reflection document will include written portions of the Final Project. Submit this as **`reflection.pdf`** with the rest of the required files for your submission on CodePost.

For ease, we have highlighted any parts which require answers in **blue**.

**Student name(s): Edward Zhang**

**Student email(s): ezhang3@caltech.edu**

---

## Part L0. Introduction

Answer the following questions to introduce us to your database and application; you may pull anything relevant from your Project Proposal and/or README.

### Database/Application Overview

What application did you design and implement? What was the motivation for your application? What was the dataset and rationale behind finding your dataset?

***Database and Application Overview Answer (3-4 sentences) :***

The goal of this project is to make a Store Simulation for cars that are on sale. It should be able to help users efficiently browse through thousands to hundreds of thousands cars on sale, and help them filter based on their specific desires such as make, model, year, and price. It will also provide a lot of helpful information when looking at purchasing a car, such as the fuel type, transmission type (automatic or manual), number of previous owners, mileage, and more. Users will thus be able to either buy or sell the cars in this store.

***Data set (general or specific) Answer:***

Data used was AI generated by asking ChatGPT to generate 100 rows of each of the attributes for each of the 14 tables in my schema. Adjustments were then made accordingly, such as changing the format of phone numbers, making dealership locations more realistic, having only 10 dealers and randomly distributing cars into one of the 10 dealers (instead of having 100 dealers with 1 car per 1 dealer), and more.

***Client interface user(s) Answer*:**

Clients will come in two forms: buyers and sellers. Buyers are individuals who are looking to buy a car from a dealership, and they will be able to search for their desired car using filters such as the price of the car, the location of the dealership offering the car, as well as the make, model, and other specifications of the car.

Sellers are individuals who wish to list a car for sale for other buyers to purchase, and should be people working at certain dealerships.

***Admin interface user(s) Answer (or approved exception to client/admin):***

The intended admin would be able to add, edit, or delete the listings of a car in the store, as well as the users (either buyers or sellers). For example, an admin may edit an entry when they think that a seller user had poorly entered in the information, or they may delete an entry if they think the car should not be sold (perhaps it is too broken down).

---

# Part A. ER Diagrams and Updated UI Diagram

**ER Diagrams**

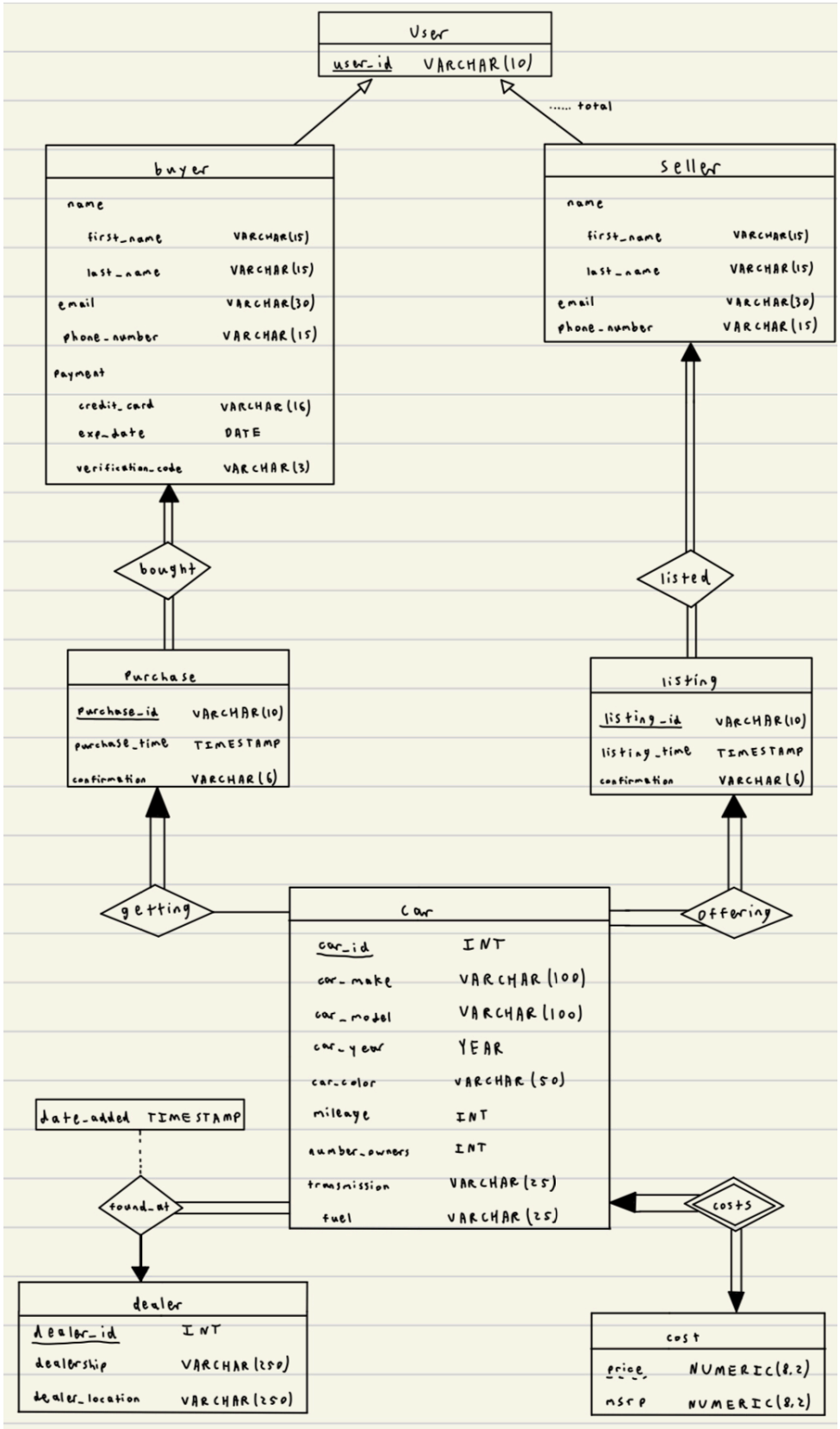
As we've practiced these past few weeks, the ER model is essential for designing your database application, and we expect you to iterate upon your design as you work through the ER and implementation steps. In this answer, you should provide a full ER diagram of your system. Your grade will be based on correct representation of the ER model as well as readability, consistency, and organization.

**Notes:** For this section **only**, we will allow (and encourage) students to share their diagrams on Discord (**#er-diagram-feedback**) to get feedback from other students on their ER diagrams given a brief summary of your dataset and domain requirements. This is offered as an opportunity to test your ER diagrams for accuracy and robustness, as another pair of eyes can sometimes catch constraints that are not satisfied or which are inconsistent with your specified domain requirements.

**Requirements:**

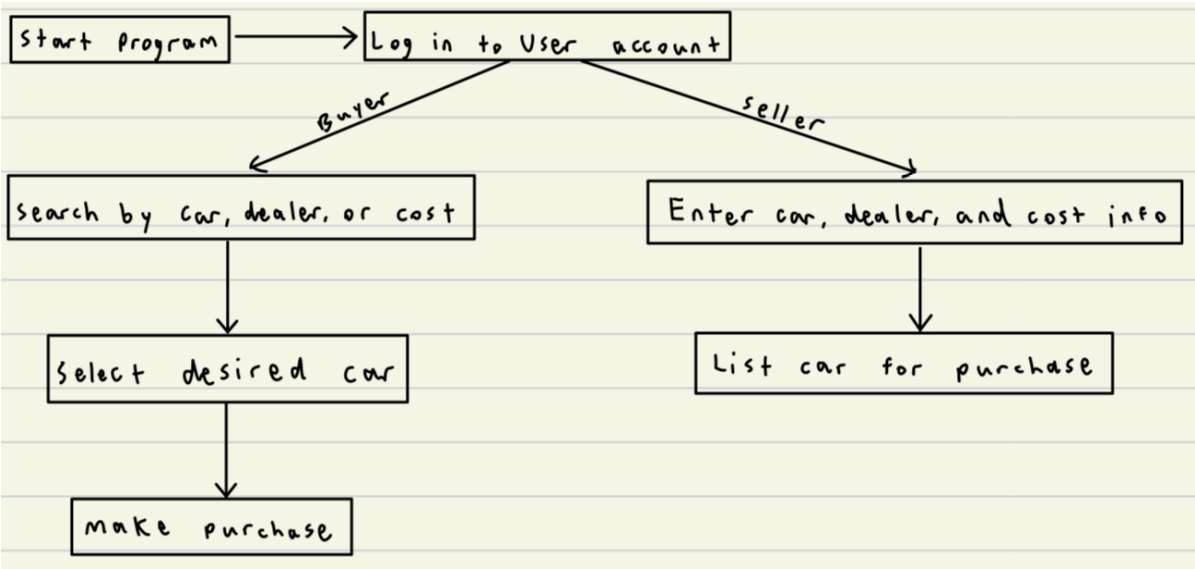
- Entity sets, relationship sets, and weak entity sets should be properly represented (also, do not use ER symbols not taught in class)
- Mapping cardinalities should be appropriate for your database schema, and in sync with your DDL
- Participation constraints should be appropriate and in sync with your DDL (total, partial, numeric)
- Use specialization where appropriate (e.g. *purchasers* and *travelers* inheriting from a *customers* specialization in A6)
- Do not use degrees greater than 3 in your relationships, do not use more than one arrow in ternary relationships.
- Use descriptive attributes appropriately
- Underline primary keys and dotted-underline discriminators
- Expectations from A6 still apply here
- Note: You do not need ER diagrams for views

**ER Diagrams:**


User
user_id VARCHAR(10)
..... total
buyer
name
first_name VARCHAR(15)
last_name VARCHAR(15)
email VARCHAR(30)
phone_number VARCHAR(15)
Payment
credit_card VARCHAR(16)
exp_date DATE
verification_code VARCHAR(3)
seller
name
first_name VARCHAR(15)
last_name VARCHAR(15)
email VARCHAR(30)
phone_number VARCHAR(15)
bought
listed
Purchase
purchase_id VARCHAR(10)
purchase_time TIMESTAMP
confirmation VARCHAR(6)
listing
listing_id VARCHAR(10)
listing_time TIMESTAMP
confirmation VARCHAR(6)
getting
Car
car_id INT
car_make VARCHAR(100)
car_model VARCHAR(100)
car_year YEAR
car_color VARCHAR(50)
mileage INT
number_owners INT
transmission VARCHAR(25)
fuel VARCHAR(25)
offering
date_added TIMESTAMP
found_at
costs
dealer
dealer_id INT
dealership VARCHAR(250)
dealer_location VARCHAR(250)
cost
price NUMERIC(8,2)
msrp NUMERIC(8,2)

Also include any updated UI flowcharts from your proposal/milestone; these should be more refined (many students got feedback in project meetings on extending UI flowcharts). If you only included one flow for a client or admin, your updated flowcharts should include the other(s) implemented since your proposal. Students are encouraged to continue sharing their iterations on Discord to share their updated flowchart(s) ask for feedback similar to the proposal requirements.

***Updated UI Flowcharts and summary of changes:***




Compared to my previous flowchart from the proposal, I've updated this new flowchart to be a lot more simple and straightforward. I also added user compatibility, and allowed for two types of users with different roles: buyers and sellers.

***Other possible visuals/diagrams created during the project (optional, potential Above and Beyond)***

---

## Part B. DDL (Indexes)

As mentioned in Part B, you will need to add at least one index at the bottom of your **`setup.sql`** and show that it makes a performance benefit for some query(s).

Here, describe your process for choosing your index(es) and show that it is used by at least one query, which speeds up the performance of the same query on a version of the same table without that index. You may find lec14-analysis.sql and Lecture 14 slides on indexes useful for strategies to choose and test your indexes. **Remember that indexes are already created in MySQL for PKs and FKs, so you should not be recreating these.**

***Index(es):***

CREATE INDEX idx_car_make_model ON car(car_make, car_model);

***Justification and Performance Testing Results:***

EXPLAIN SELECT car_id, car_make, car_model, car_year
FROM car
WHERE car_make = 'Toyota' AND car_model = 'Corolla';

This index allows for users to quickly search by a car's make and model, which are often the most common search criteria that buyers will initially search for when looking for a car that they want. Thus having an index for this common query will help make the user's experience more efficient.

---

# Part C. Functional Dependencies and Normal Forms

**Note:** For 24wi, this part will be optional, but students can earn up to 8 additional points for answers. A comprehensive slide deck is provided on Canvas covering Functional Dependencies and Normal Forms (the most important Normal Form to know about is BCNF; 3NF and 4NF are included for additional material).

**Requirements (from Final Project specification):**

- What is the purpose of a functional dependency? Why is it relevant to database design?
- Why is BCNF ideal for a database schema? Your answer should include a specific reference to the slides and/or supplementary recording (e.g. we don't want to see an answer copied from a Google search; we want you to be able to answer this and ask questions if you are unsure!)
- Identify *at least 2 non-trivial functional dependencies* in your database
- Choose and justify your decision for the normal form(s) used in your database for at least 2 tables (if you have more, we will not require extra work, but will be more lenient with small errors). BCNF and 3NF will be the more common NF's expected, 4NF is also fine (but not 1NF).
    - Your justification will be strengthened with a discussion of your dataset breakdown, which we expect you to run into trade-offs of redundancy and performance.
- For up to two of your relations having at least 3 attributes (each) and at least one functional dependency, prove that they are in your chosen NF, using similar methods from the BCNF assignment.
    - If you have identified functional dependencies which are not preserved under a BCNF decomposition, this is fine

**Purpose/Relevance of Functional Dependencies:**

**Purpose/Relevance of BCNF:**

**Identified Functional Dependencies:**

**Normal Forms Used (with Brief Justifications):**

**NF Proof 1:**

**NF Proof 2:**

---

# Part G. Relational Algebra

**Requirements (from Final Project specification, Part G):**

- Minimum of 3 non-trivial queries (e.g. no queries simply in the form **`SELECT <x> FROM <y>`**)
- At least 1 group by with aggregation
- At least 3 joins (across a minimum of 2 queries)
- At least 1 update, insert, and/or delete
    - This may be equivalent to said SQL statements elsewhere (e.g. queries or procedural code), but are not required to be; in other words, you can write these independent of other sections
- At least 2 of your SQL queries (Part H) should be equivalent to 2 of your RA expressions, and these should be clearly indicated.
- Appropriate projection/extended projection use
- Computed attributes should be renamed appropriately
- Part of your grade will come from overall demonstration of relational algebra in the context of your schemas; obviously minimal effort will be ineligible for full credit; it is difficult to formally define "obviously minimal", but refer to A1 and the midterm for examples of what we're looking for
- Above each query, briefly describe what it is computing; we will use this to grade for correctness based on what the query is supposed to compute; lack of descriptions will result in deductions, since we have no idea otherwise of what the query is intended to do.
- If you have a SQL query that uses ORDER BY, you can omit this in your relational algebra equivalent (implementing via RA is possible, but not required for this exercise)

Below, provide each of your RA queries following their respective description.

**Search for all cars sold at a certain dealership ordered by price:**

$\Pi_{car\_id,\ car\_make,\ car\_model,\ car\_year,\ car\_color,\ price}\ (\sigma_{dealer\_id\ =\ 1}\ (car \bowtie found\_at \bowtie dealer \bowtie cost))$

**Computes the average price of cars sold by each dealer:**

$_{dealer\_id,\ dealer\_location}G_{\mathbf{avg}(price)}\ (dealer \bowtie found\_at \bowtie car \bowtie cost)$

**Finds all Toyota cars below a certain price along with the dealer they are at:**

$\Pi_{car\_id,\ car\_make,\ car\_model,\ car\_year,\ car\_color,\ price,}$ dealership, dealer_location $(\sigma_{\text{car\_make = 'Toyota'} \wedge \text{ price < 25000}}$ (car ⋈ cost ⋈ found_at ⋈ dealer))

**Updates the price of a specific car:**

cost ← (cost - $\sigma_{\text{car\_id = 5}}$ (cost)) ∪ $\Pi_{car\_id,\ (\text{price - 1000})}$ ($\sigma_{\text{car\_id = 5}}$ (cost))

# Part L1. Written Reflection Responses

### Challenges and Limitations

List any problems (at least one) that came up in the design and implementation of your database/application (minimum 2-3 sentences). Include anything that may help us distinguish between conceptual issues or untested code vs. scoping/time constraints.

***Answer:*** *Overall, I think the hardest part for me was getting started. I tend to struggle with tasks that require creativity, so having to figure out how I best wanted to design my project was a bit tough. The ER Diagram was extremely helpful in having a plan of action, but perfecting the ER Diagram to my desired needs took quite a bit of time and effort.*

### Future work

If you are particularly eager for a certain application (have your own start-up in mind?), it is easy to over-scope a final project, especially one that isn't a term-long project. You can list any stretch goals you might have "if you had the time" which staff can help give feedback on prioritizing (2-3 sentences).

***Answer:*** *This was a really cool project! Originally I wanted to implement a sort of "garage" for buyer clients to keep track of what cars they had bought, or which cars they wanted to save as their wishlist. I also wanted to do the same thing with an "inventory" for seller clients to keep track of what cars they had sold or what cars they were still in the process of selling. However I couldn't figure out how to implement a way for the garage/inventory entity to include multiple car listings but only be seen by their respective clients. In other words, I wasn't sure how to give clients specific permissions to view/edit multiple rows within a single garage/inventory entity.*

### Self-Evaluation

What is your expected grade for the Final Project (out of 100)? Justify what you think are the strongest points, especially pointing to demonstrated improvement in areas that may have had lower scores in assignments throughout the term. Also provide any notes here on areas that could be improved (and what you would do differently next time).

***Answer:*** *I think overall, I probably deserve somewhere between a 90 and 92 on the Final Project. I worked really hard on both the design and implementation, and I am quite proud of the design of my ER Diagram, as well as the process I went through to implement it. Although I did struggle with getting the user to be properly authenticated, I do believe that everything else should work. It really is disappointing that I wasn't able to figure out how to login to the Python command line though, and so I couldn't try out my final product for myself :(*

What is your expected grade for the course overall?

***Answer:*** *Overall, I'd probably expect an A- in the course overall. I think I did quite well on the homework assignments, and I really enjoyed learning a lot about databases and SQL (especially ER Diagrams), however I struggled with the user authentication portion of the final project and so although everything else works and should be correct, I don't have a working running Python command line app :(*

## Collaboration (Required for Partners)

This section is required for projects which involved partner work. Each partner should include 2-3 sentences identifying the amount of time they spent working on the project, as well as their specific responsibilities and overall experience working with a partner in this project.

***Partner 1 Name:***
***Partner 1 Responsibilities and Reflection:***

***Partner 2 Name:***
***Partner 2 Responsibilities and Reflection:***

## Requested Feedback

To what extent would you like feedback on your project vs. a grade? We will prioritize feedback to students who read through it, particularly as a potential portfolio project to use in future job/grad school applications (if there are areas you would like to improve or learn more about, let us know!), but will otherwise keep written feedback brief.

## Sharing Work

We like to share student examples to highlight diversity of projects and showcase the work done by students each term. Would you be open to any of the following being shared?

Anything: ***Y***
Diagrams/visuals: ***Y***
Final Project (demo, code snippets): ***Y***
Reflection: ***Y***

***Would you prefer to be credited by name or anonymous for any of the above?***
*Anonymous please!*

## Other comments

The Final Project has generally been received positively for students since replacing the Final Exam, but we continue to collect feedback to improve the experience and learning outcomes. Recent feedback incorporated has included moving more requirements to the proposal, an added milestone, incorporating projects into lectures/assignments, and adjusting some requirements to account for project diversity. We would appreciate your feedback on what you found most helpful, and what you might find helpful to change.

***Answer:*** *I think the Final Project was a really cool idea! I really liked how it brought everything from the class together, and it all culminated in one big project for something that we all thought was personally cool or useful. The midterm checks and Final Project Proposals throughout the term were also extremely helpful in making sure we got a head start on planning and working on the Final Project, and I think that having students work on the Final Project throughout the term was a really good idea.*