CS 121 Final Project Reflection

Due: Saturday March 16th, 11:59PM PST

Note: This specification is currently for 23wi, subject to minor changes for 24wi.

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

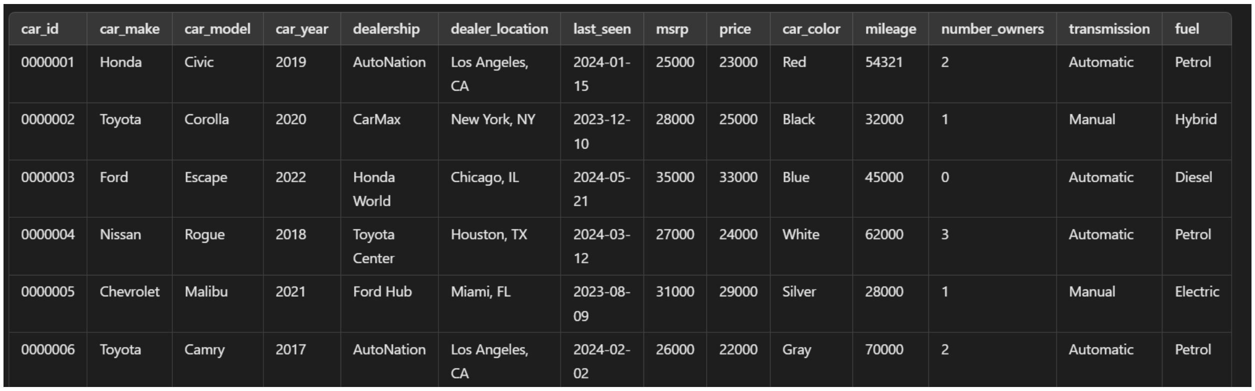
Answer the following questions to introduce us to your database and application; you may pull anything relevant from your Project Proposal and/or README.

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.

Data set (general or specific) Answer:
Ideally I'd have a huge pre-made dataset with all of the columns that I want (or I could combine multiple datasets to create this), however for now I will use ChatGPT to auto-generate 10,000 rows of unique car_id (unique to our given dataset, will likely just be an auto-incrementing 7 digit number), car_make (brand, e.g. 'Honda'), car_model (e.g. 'Civic'), car_year, dealership (who is selling the car), dealer_location (location of dealer), last_seen (YYYY-MM-DD of the last date that the vehicle was seen at the dealership), msrp (Manufacturer's Suggested Retail Price, aka default price of vehicle), price (actual current price of vehicle), car_color, mileage, number_owners (number of previous owners), transmission (automatic or manual), and fuel (petrol, diesel, etc.).
An example of the first few rows is shown below.



car_id	car_make	car_model	car_year	dealership	dealer_location	last_seen	msrp	price	car_color	mileage	number_owners	transmission	fuel
0000001	Honda	Civic	2019	AutoNation	Los Angeles, CA	2024-01-15	25000	23000	Red	54321	2	Automatic	Petrol
0000002	Toyota	Corolla	2020	CarMax	New York, NY	2023-12-10	28000	25000	Black	32000	1	Manual	Hybrid
0000003	Ford	Escape	2022	Honda World	Chicago, IL	2024-05-21	35000	33000	Blue	45000	0	Automatic	Diesel
0000004	Nissan	Rogue	2018	Toyota Center	Houston, TX	2024-03-12	27000	24000	White	62000	3	Automatic	Petrol
0000005	Chevrolet	Malibu	2021	Ford Hub	Miami, FL	2023-08-09	31000	29000	Silver	28000	1	Manual	Electric
0000006	Toyota	Camry	2017	AutoNation	Los Angeles, CA	2024-02-02	26000	22000	Gray	70000	2	Automatic	Petrol

***Client user(s) Answer*:**
The intended user base would be individuals looking to buy a car from a dealership, and ideally have a pretty good idea of what kind of car they are looking for. They should be able to search and navigate through the database to find a car that fits their needs.

Admin user(s) Answer:
The intended admin would be a car dealership salesperson who is either adding or removing one of their listings onto the database.

Part A. ER Diagrams

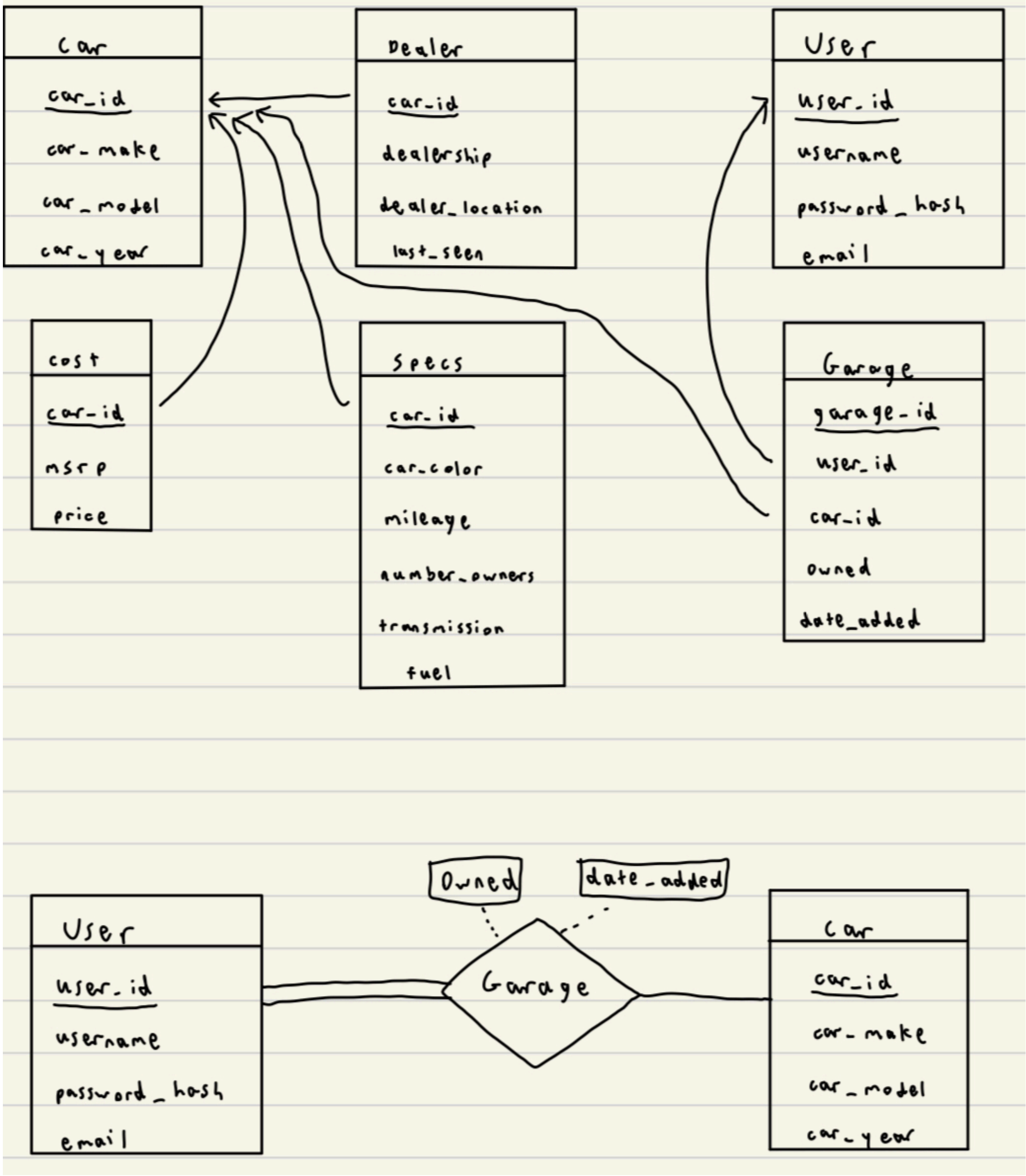
As we've practiced these past few weeks, the ER model is essential for designing your database application, and we expect you to iterate upon your design as you work through the ER and implementation steps. In this answer, you should provide a full ER diagram of your system. Your grade will be based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


Car
car_id
car_make
car_model
car_year
Dealer
car_id
dealership
dealer_location
last_seen
User
user_id
username
password_hash
email
cost
car_id
msrp
price
specs
car_id
car_color
mileage
number_owners
transmission
fuel
Garage
garage_id
user_id
car_id
owned
date_added
Owned
date_added
User
user_id
username
password_hash
email
Garage
Car
car_id
car_make
car_model
car_year

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_dealer_location ON dealer(dealer_location);
CREATE INDEX idx_dealership ON dealer(dealership);
CREATE INDEX idx_cost_price ON cost(price);
CREATE INDEX idx_specs_transmission ON specs(transmission);
```

Justification and Performance Testing Results:

```
SELECT * FROM dealer WHERE dealer_location = 'Chicago';
SELECT * FROM dealer WHERE dealership = 'Best Cars';
SELECT * FROM cost WHERE price BETWEEN 10000 AND 20000;
SELECT * FROM specs WHERE transmission = 'automatic';
```

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
- 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.

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

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)

Answer:

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:

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:

What is your expected grade for the course overall?

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:

OTHER COMMENTS

This is the first time CS 121 has had a Final Project, and we would appreciate your feedback on whether you would recommend this in future terms, as well as what you found most helpful, and what you might find helpful to change.

Answer: