Tufts University
CS 115: Database Systems
Problem Set 1
Spring 2022

Complete the following exercises to the best of your ability and submit your answers on Gradescope by 11:59 PM on Wednesday, February 16, 2022. There will be a 10% deduction for submissions made by Thursday, February 17 at 11:59 PM, and a 20% deduction for submissions made by Friday, February 18 at 11:59 PM.

You can access Gradescope here: https://www.gradescope.com/. Click Log In and then School Credentials and scroll down to Tufts to log in using your Tufts credentials.

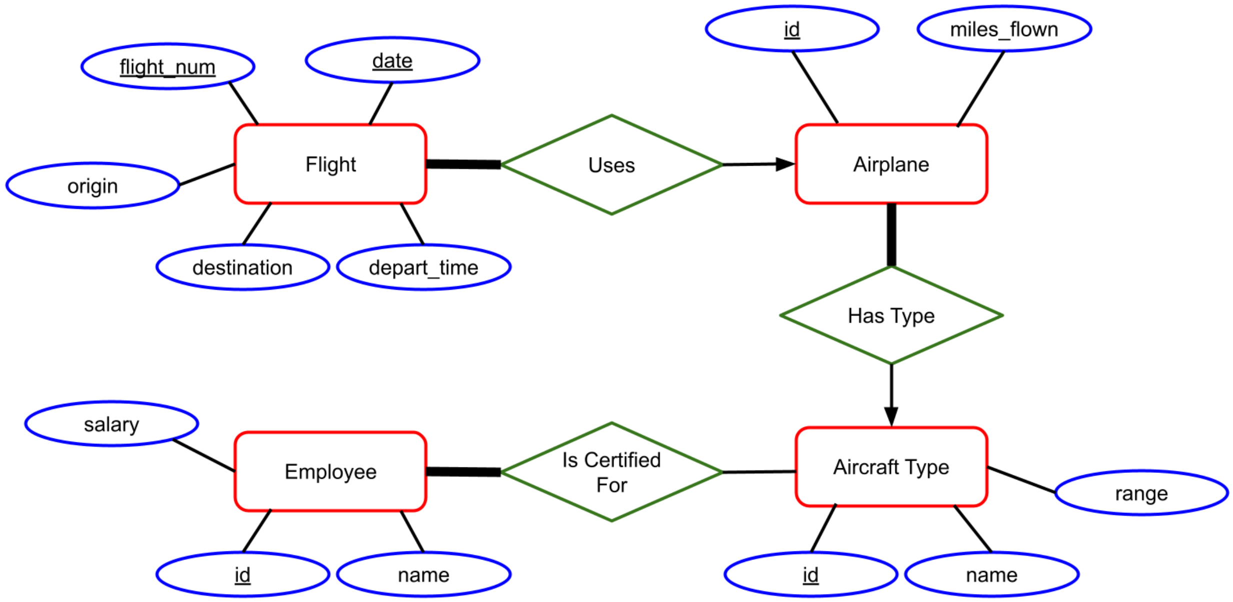
Format your answers as a PDF file. I recommend making a copy of this file as a Google Doc (File > Make a copy), filling in your answers, and then saving it as a PDF (File > Download > PDF document). You can also download this Google Doc as a Word document and save that as a PDF, or use other software entirely.

If you choose to, you can work with 1-2 partners on this assignment. If you worked with partners, mark in Gradescope the partners that you worked with. Make only one submission per group.

Direct any questions about the assignment to Piazza.

Problem 1 (10 points)

Consider the following ER diagram that describes an airline domain:



a. (2 points) At least one of the relationship sets captures a many-to-one relationship. Which one(s)? In your answer(s), you should specify the direction of the relationship (e.g., _________ *is a many-to-one relationship from* __________ *to* ____________).

Uses is a many-to-one relationship from Flight to Airplane
Has Type is a many-to-one relationship from Airplane to Aircraft Type

b. (2 points) Describe all relationships and constraints that are specified by the diagram. Use words that describe the problem domain (e.g., *Each course meets in at most one room...*) rather than technical terminology.

Each Flight uses exactly one Airplane.
Each Airplane has exactly one Aircraft Type.
Each Employee is certified for at least one Aircraft Type.

c. (3 points) Explain why the Date field must be a part of the primary key of the Flight entity set.

The Date field must be part of the primary key of the Flight entity set to make the primary key unique. Not having the Date field (and just the flight_num field) will not make a unique primary key because two different flights can have the same flight_num but just be on different days.

d. (3 points) Because of the specified constraints, the diagram only permits pilots to exist in the Employee entity set. Why? Also, which constraint could be changed to allow other types of employees (flight attendants, gate agents, etc.) to exist in the Employee entity set? Explain how to change it.

The diagram only permits pilots to exist in the Employee entity set because the participation constraint (for Employee) on Is Certified For makes it so every employee must be certified for at least one aircraft type and pilots are the only employees that satisfy this. Therefore, this constraint should be removed in order to allow other types of employees to exist in the Employee entity set.

Problem 2 (10 points)

Consider the schema and an example row of the following relation, which describes a pet store:

purchase_id	store_name	cust_id	cust_name	cust_phone	pets_types	pet_prices	purchase_date
20394	Salem	0345039	J. Brown	(555) 555-1212	fish, dog	5.00, 300.00	‘2022-01-11’

Notice that this database is not well-structured. To start normalizing it, we could first convert it to first normal form by marking the primary keys and forcing every cell to contain only a single value. For the primary key, we assume that pet types are unique within a purchase, so the primary key is (purchase_id, pet_types).

purchase_id	store_name	cust_id	cust_name	cust_phone	pet_types	pet_price	purchase_date
20394	Salem	0345039	J. Brown	(555) 555-1212	fish	5.00	‘2022-01-11’
20394	Salem	0345039	J. Brown	(555) 555-1212	dog	300.00	‘2022-01-11’

This table also reflects the following functional dependencies:

purchase_id → store_name, cust_id, cust_name, cust_phone, purchase_date
pet_type → pet_price
cust_id → cust_name, cust_phone

a. (5 points) Continue to normalize this database by converting it to second normal form. Your answer should be the schemas of the resulting tables.

purchase_id	pet_types

purchase_id	store_name	cust_id	cust_name	cust_phone	purchase_date

pet_types	pet_price

b. (5 points) Continue to normalize this database by converting your answer in (a) to third normal form. Your answer should be the schemas of the resulting tables.

purchase_id	pet_types

purchase_id	store_name	cust_id	purchase_date

cust_id	cust_name	cust_phone

pet_types	pet_price

Problem 3 (15 points)

Consider the following relations R and S:

R

a	b
2	4
6	8
10	12

S

c	b	a
1	4	2
2	8	6
3	7	11

For each of the following questions, use the *Insert* → *Table* menu option (if using Google Docs) to insert a table for the answer, and then fill the table with the correct values.

a. What is the Cartesian product of R and S?

R.a	R.b	c	S.b	S.a
2	4	1	4	2
2	4	2	8	6
2	4	3	7	11
6	8	1	4	2
6	8	2	8	6
6	8	3	7	11
10	12	1	4	2
10	12	2	8	6
10	12	3	7	11

b. What is the natural join of R and S?

a	b	c
2	4	1
6	8	2

c. What is the left outer join of R and S?

a	b	c
2	4	1
6	8	2
10	12	null

d. What is the right outer join of R and S?

a	b	c
2	4	1
6	8	2
11	7	3

e. What is the full outer join of R and S?

a	b	c
2	4	1
6	8	2
10	12	null
11	7	3

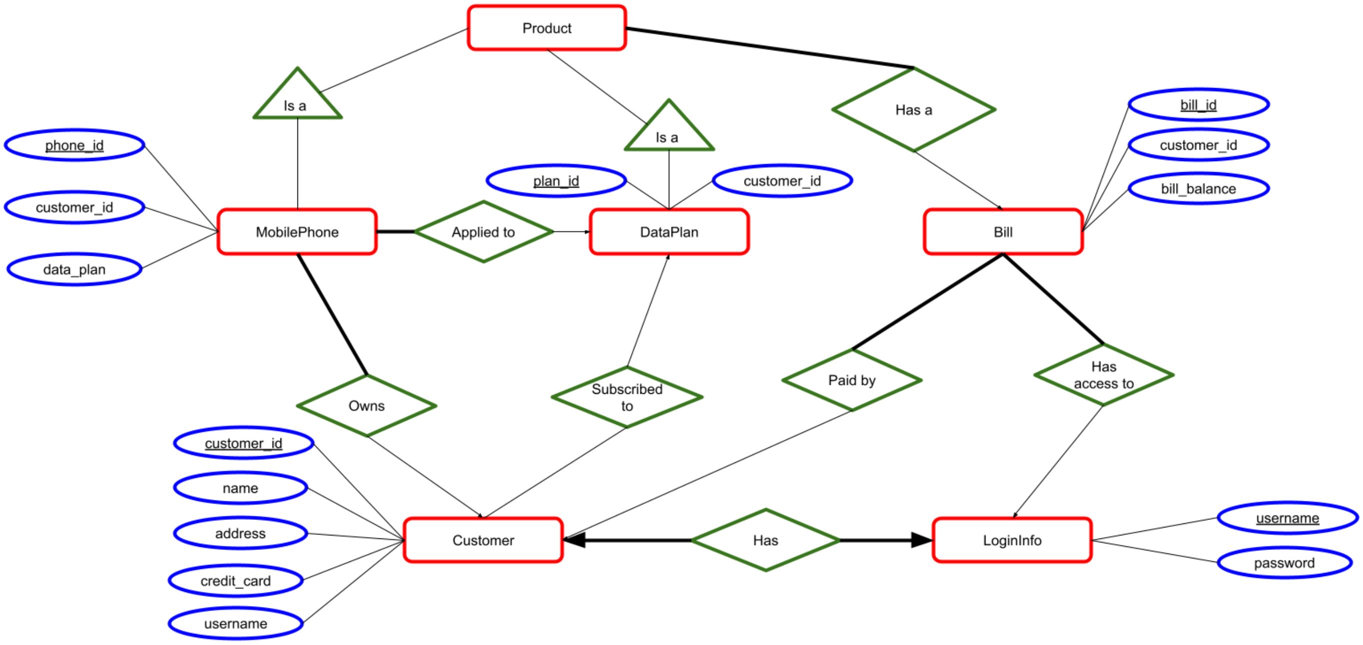
Problem 4 (20 points)

Create an ER diagram that corresponds to the following scenario. Be sure to capture all relevant entities, relationships, attributes, primary keys, and constraints. If necessary, state any additional assumptions you make, as long as they are reasonable and do not contradict the problem statement.

> A mobile phone carrier wants to create a new database to capture information about its customers and products. Customers in the system are tracked by their customer ID, but personal data (name, address, etc.) and payment information (credit card number) is also maintained.
>
> The company offers two kinds of products: mobile phones and data plans to use with those phones. They are sold and tracked separately -- customers can buy a phone without a data plan, or could buy a data plan to apply to a phone that they already have. There is no limit to the number of phones that a customer could buy, but a customer can have at most one data plan that is applied across all of his phones.
>
> Customers pay bills that correspond to either phones or data plans they have purchased. The first bill is generated immediately upon a customer's purchase, and continue to be generated according to the customer's payment plan. Each bill for a phone or data plan is given to exactly one customer.
>
> To pay the bills, customers maintain a username/password to log in to an online system and make payments. This login information is mandatory for all customers. Bills also display this username/password information to remind users of how to make payments.

Assumptions:

- A Product has (points to) exactly one Bil, but a single Bill can encompass many 1 or more Products (i.e. a Customer can receive one Bill per Product they have, or a Customer can receive one Bill that represents the 'total' cost across all Products)



Problem 5 (20 points)

Convert the ER diagram you designed in Problem 4 to a relational schema using the algorithm presented in class. Be sure to specify all primary keys and foreign keys.

MobilePhone(<u>phone_id</u>, customer_id, data_plan)
DataPlan(<u>plan_id</u>, customer_id)
Bill(<u>bill_id</u>, customer_id, bill_balance)
LoginInfo(<u>username</u>, password)
Customer(<u>customer_id</u>, name, address, credit_card, username)

Foreign Keys:

- 'customer_id' in MobilePhone, DataPlan, and Bill are foreign keys to Customer(customer_id)
- 'username' in Customer is a foreign key to LoginInfo(username)
- 'data_plan' in MobilePhone is a foreign key to DataPlan(plan_id)

Problem 6 (25 points)

Instructions

Write relational algebra queries to answer the prompts below. In your answers, you can either use the relational algebra symbols that we covered in class (π, σ, ρ, ←, ×, ⋈, ⟕, ⟖, ⟗, ∩, ∪, -) or replace the relational algebra symbols with appropriate words (e.g., JOIN for ⋈, SELECT for σ, etc.).

Database Description

The following relational algebra queries will use the following Oscars database:

Actor(actor_id, movie_id)
Director(director_id, movie_id)
Oscar(type, year, movie_id, person_id)
Person(id, name, dob, pob)
Movie(id, name, year, rating, runtime, genre, earnings_rank)

Actor(actor_id, movie_id) are foreign keys Person(id), Movie(id)
Director(director_id, movie_id) are foreign keys to Person(id), Movie(id)
Oscar(movie_id, person_id) are foreign keys to Movie(id), Person(id)

Some notes about this database:

1. Oscars are awarded for a given movie and a given person. The possible types of Oscars given are 'BEST-PICTURE', 'BEST-DIRECTOR', 'BEST-ACTRESS', 'BEST-ACTOR', 'BEST-SUPPORTING-ACTRESS', 'BEST-SUPPORTING-ACTOR'. For ‘BEST-PICTURE’ Oscars, the *person_id* field is null.

2. The *genre* field of *Movie* is a string consisting of individual characters that represent the movie’s genre, such as ‘A’ for action, ‘C’ for comedy, ‘D’ for drama, etc. For example, the string ‘AD’ means it is an action and drama movie.

Query Problems

a. (2 points) Find the places of birth and dates of birth for Katie Holmes and Eddie Murphy. The tuples in your result should be of the form (name, pob, dob). There should only be two tuples in your result.

$$\Pi_{\text{name, pob, dob}} \left(\sigma_{\text{name = “Katie Holmes” OR name = “Eddie Murphy”}} (\text{Person})\right)$$

b. (2 points) Find all movies that are categorized as animated (‘N’) films. The tuples in your result should be of the form (movie ID, movie name, genre). Hint: read the description of this column above.

$$\Pi_{\text{id, name, genre}} (\sigma_{\text{genre = “\%N\%”}} (\text{Movie}))$$

c. (3 points) Find the names of all directors in the database who were born in Ohio. Assume that all such directors have the string “Ohio” somewhere in their *pob* field. The result of the query should be of the form (director name, place of birth).

$$\Pi_{\text{name, pob}} (\sigma_{\text{pob = “\%Ohio\%”}} (\text{Director} \bowtie_{\text{director\_id = id}} \text{Person}))$$

d. (3 points) Find information about all of the Oscars won in the year 1990. Your result should have six rows -- one for each Oscar won in that year (Best Actor/Actress, Best Supporting Actor/Actress, Best Picture, and Best Director). The results of your query should be of the form (Oscar type, movie name, person name). For the Best Picture Oscar, the person name field should be null.

$$\Pi_{\text{type, movie\_name, person\_name}} (((\sigma_{\text{year = 1990}} (\text{Oscar})) \bowtie_{\text{movie\_id = id}} \text{Movie}) \text{⟕}_{\text{person\_id = id}} \text{Person})$$

e. (7 points) Find all of the people who have won **at least two** Oscar awards. Hint: join the Oscar table with itself to find actors who have won different awards.

$$\Pi_{\text{name}} (\text{Person} \bowtie_{\text{id = person\_id}} (\text{Oscar} \bowtie_{\text{person\_id = person\_id AND (type != type OR year != year)}} \text{Oscar}))$$

**Note: The attribute name on the left-hand side of the predicate is for the ‘left’ Oscar table and that on the right-hand side of the predicate is for the ‘right’ Oscar table

f. (8 points) Find all of the people who have won **exactly two** Oscar awards. Hint: use the answer from (e) in a set difference.

$$\Pi_{\text{name}} (\text{Person} \bowtie_{\text{id = person\_id}} (\text{Oscar} \bowtie_{\text{person\_id = person\_id AND (type != type OR year != year)}} \text{Oscar}) - \sigma_{\text{(Oscar1.person\_id = Oscar2.person\_id AND Oscar1.person\_id = Oscar3.person\_id) AND (Oscar1.type != Oscar2.type OR Oscar1.year != Oscar2.year) AND (Oscar1.type != Oscar3.type OR Oscar1.year != Oscar3.year)}} ((\rho_{\text{Oscar1}}(\text{Oscar}) \bowtie \rho_{\text{Oscar2}}(\text{Oscar})) \bowtie \rho_{\text{Oscar3}}(\text{Oscar}))))$$