CS-GY 6083 Section INET Midterm Exam Spring 2024

READ THESE INSRUCTIONS:

- This is a take-home exam. It is due at 11:59 p.m. on Sunday March 17, 2024. Until then, you may use as much time as you'd like to do the exam. Typeset your answers or hand-write them *very neatly.* Be sure to mark which problem is which on the outline on GradeScope.
- Do not wait until the last minute to hand in your work. You may submit some and then resubmit later and I will see your latest version when I grade it. Late work will not be accepted.
- You may consult the textbook, course lectures, your notes, homework solutions, or other previously published material. You may execute your queries against sample test data of your choosing. *You may not consult any other "intelligent" entities, be they human or artificial.*
- Before handing the exam in, you will be expected to sign the following academic integrity statement:

  > I certify that my solutions to this exam are my own work. I did not consult with other people (with the possible exception of seeking clarification from Prof Frankl) and did not use any generative AI systems to help me with the exam. I understand that if the Professor discovers that I have violated this agreement, I will get a zero on the exam and will be referred to the CS Department and/or Tandon's disciplinary officers for possible additional sanctions.

- If you think there's an error on the exam, please let Prof Frankl know. Keep your eye on BrightSpace and Ed Discussion in case any corrections or clarifications are posted. If something is unclear or ambiguous, try to use common sense to figure out what is intended and note any assumptions you are making. You may contact Prof Frankl through private messages on Ed Discussion or e-mail to seek clarification.
- Good Luck!

Use this HotelBooking schema for Part 1.

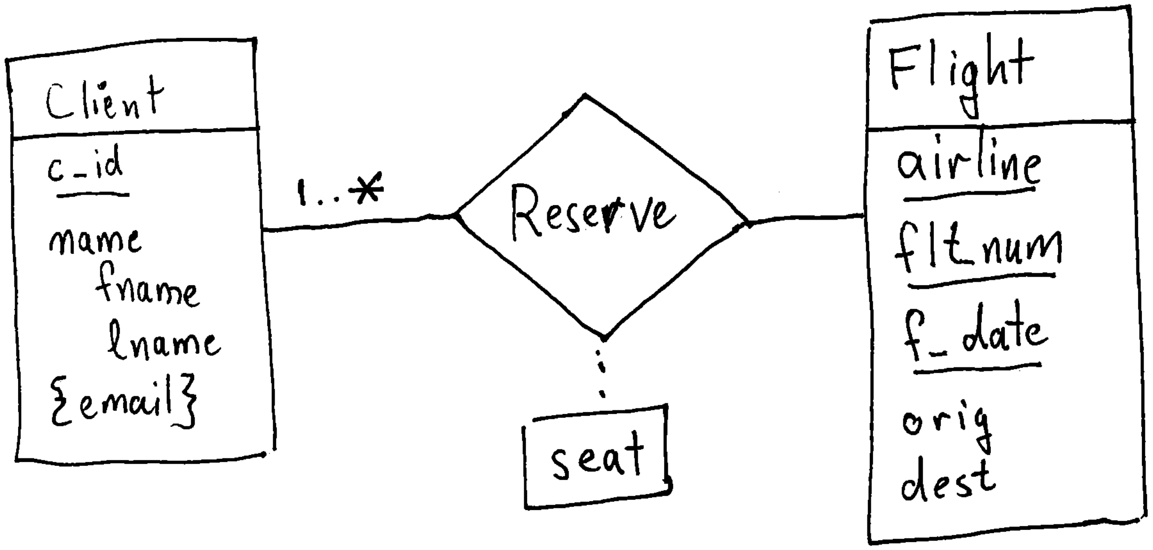
Room(floor, line, configCode, basePrice)
Guest(gID, fname, lname, phone, memStatus)
Night(month, day weekend, holiday)
Reserve(month, day, floor, line, gID, pricePaid, notes)
Adjoin(floor,line1,line2)

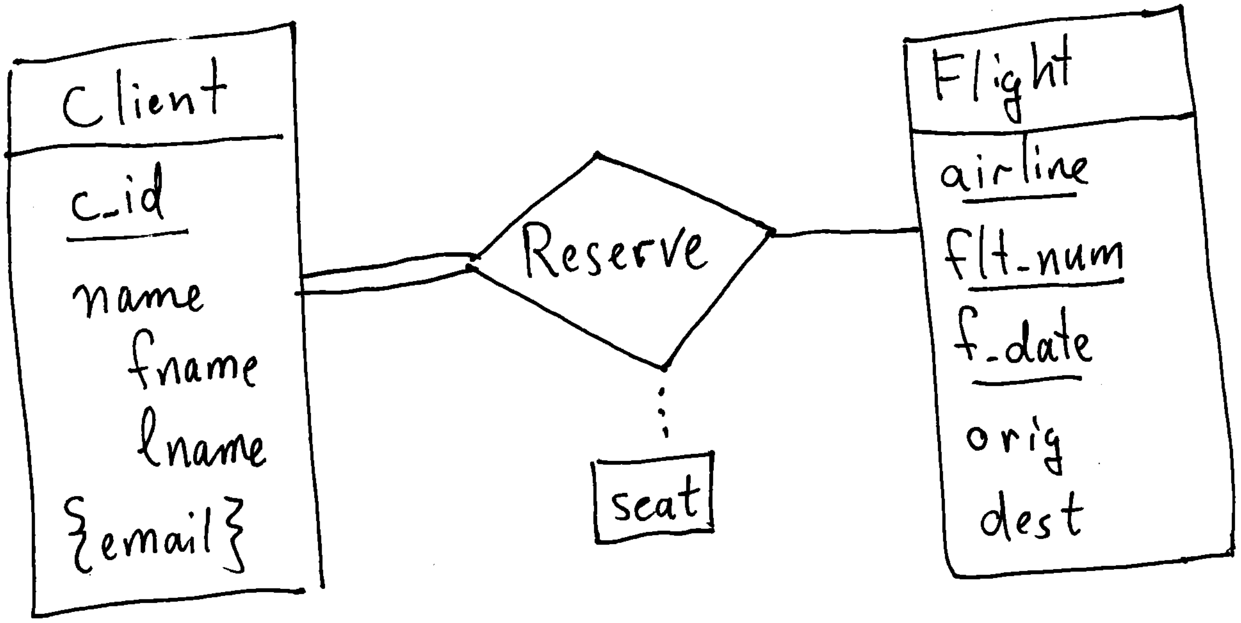
Explanation of the Schema:

A hotel maintains a database with information about their rooms, guests, and reservations. The `Room` table identifies rooms by their `floor` and their `line`, e.g. room “5C” is in line C on the fifth floor. The `Room` table also has some additional information about each room. The `Guest` table has data about people who have reserved rooms, including a unique `gID`, the guest’s name (`fname, lname`), contact info for the guest, and their status in the hotel’s membership club. The `Night` table indicates dates as well as special info about the date that might affect prices or availability. To avoid dealing with details about datatypes used for dates and times, dates here are represented with attributes `month` and `day`, and we’ll assume the database only has date for the current calendar year; so (`month, day`) = (’March’, 20) represents March 20, 2024. The `Reserve` table indicates which guests have reserved which rooms for which nights. Since a guest may pay more or less than the `basePrice` this table also records the `price` the guest actually pays. The hotel has some adjoining rooms (rooms that are next to one another with a door between them that can optionally be unlocked to create a two-room suite). These are indicated in the `Adjoin` table. For example if rooms in lines ’A’ and ’B’ on the fourth floor are adjoining the table would include tuples (4, ’A’, ’B’) and/or (4, ’B’, ’A’). In any queries you write that involve this table, you should specify any your assumptions about how the two rooms are represented (e.g. two rows, one in each order; *or* one row and which one.)

Part 2

Use this ER diagram for ~~problem 3:~~ It represents data for a travel agency or a business's travel department, keeping track of which clients are booked on which flights. (Two versions of the ER diagram are shown, using different notations for cardinality constraints.)





Part 1

Use the Hotel reservation schema on page 2 for Part 1. For SQL queries, use NATURAL JOIN, JOIN USING(...) or JOIN ON(...), when possible, rather than putting the join condidtion in the WHERE clause. Please write the queries neatly, starting new clauses (SELECT, FROM, WHERE, ...) on new lines and indenting for readability. Feel free to create VIEWS or use WITH statements to make your queries more readable. You may use any keywords from standard SQL, even if they're not supported by MySQL. Points will be deducted if your queries are much more complicated than necessary or involve unneded tables.

1. Write table definitions from the schema, including any foreign key constraints that you think should be included. You may choose any reasonable data types for attributes, but then be consistent when you write queries in the remaining parts of this problem.

2. Write an SQL query to find the name (`fname, lname`) and phone number of each guest who reserved a room in March with a price higher than the basePrice.

3. Write a Relational Algebra query to find the name and phone number of each guest who reserved a room in March with a price higher than the basePrice.

4. Write a Domain Relational Calculus query to find the name and phone number of each guest who reserved a room in March with a price higher than the basePrice.

5. Write an SQL query to find the `gID` and phone number of each guest who booked a pair of adjoining rooms for March 22 and March 23.

6. Write a Relational Algebra query to find the `gID` and phone number of each guest who booked a pair of adjoining rooms for March 22 and March 23.

7. Write a Domain Relational Calculus query to find the `gID` and phone number of each guest who booked a pair of adjoining rooms for the nights of March 22 and March 23.

8. Write an SQL query to find the name, gID, and phone number of the guest who paid the highest total price for all of their bookings in February. (If there are ties, list all of the guests who are tied for the highest.)

9. Write an SQL query to find the gIDs, names, and phone numbers of guests who've booked rooms with every configCode during April.

10. Write an SQL query to find rooms with configuration code 'DeluxeKing' that are available (not reserved by anyone) on March 22 and March 23 with base price less than $150 each night.

11. Using your netID or your NYU N-number as your `gID`, Write any SQL statements that would be needed for you to reserve room 5C for the nights of March 22 and March 23 at 90% of the `basePrice`. (You may make up a phone number and other personally identifying info, other than the `gID` and your name.)

12. Briefly discuss the pros and cons of adding a table ”Available1(month, day, floor, line, status)” that indicates whether or not each room is available on each night. Which operations that you expect to be done frequently would be easier and which would be harder. Explain.

Part 2

Consider the ER diagram on page 3.

1. indicate whether each of these statement is True or False, according to the ER diagram. Please write out the whole word “True” or “False” to avoid handwriting ambiguity issues.
   (a) A client can have reservations for two flights on the same f_date
   (b) A client can have two reservations with different seats for the same flt_num, airline and f_date.
   (c) Every client has a reservation on at least one flight.
   (d) Every client has a reservation on at most one flight.
   (e) Every flight has at least one client with a reservation on it
   (f) Every flight has at most one client with a reservation on it

2. Derive a relation schema from the E-R diagram, using the technique we studied. You may show your answer as a schema diagram or by writing SQL table definitions *Remember to underline primary keys and to indicate foreign key constraints with arrows from the referencing tables to the referenced tables.* Alternatively you can use the Name(attribute list) notation and clearly note primary and foreign keys constraints.

3. Now suppose we want to also keep track of which clients belong to which airlines' frequent fliers programs and the number of points they have in those programs. You may assume that every airlines has such a program. Redesign the ER diagram to include:
   - An entity set representing airlines, with attributes airline_name and ff_program indicating the airline name and the name of its frequent flier program.
   - A relationship set indicating which clients belongs to the programs of which airlines
   - Any changes to existing entity sets to avoid redundancy. **Hint**: Flight numbers are not unique across all airlines, but they are unique within a given airline. Use a weak entity set.
   - An additional attribute in the appropriate place to show the number of bonus points each client has in each of the frequent flier programs to which they belong.
   - Any cardinality constraints needed.

   Please draw the entire diagram, including those entity sets, relationship sets, attributes that are the same as in the original diagram on pg 2. You may use either notation (or a mix of notations).

4. Derive a relation schema from the modified E-R diagram. You may show the entire schema or just very clearly describe all of the changes, compared to the schema from the original E-R diagram.

Part 3

In this problem, you'll demonstrate your knowledge of the material we studied, using some scenario of your choice to illustrate database design and query writing. Think of some data you know about or use. It can be about anything – your hobbies, your courses, sports, your workplace or school, travel, entertainment, you name it. It should be fairly simple, but complex enough that your ER model will have at least three entity sets and at least two relationship sets; at least one of the entity sets should be a weak entity set or should have a primary key with multiple attributes. There should be at least one totality or cardinality constraint. It shouldn't be copied or almost the same as an example we used in class, homeworks, or practice problems, or be copied from an ER diagram example in a textbook or online resource. If you really can't think of anything more original, you may add some features to the Hotel Reservation system from Part 1 of this exam.

1. Write a textual description of the data, similar in style to the data modeling problems you've had in homeworks and on this exam.
2. Draw an ER diagram and briefly explain design decisions you've made. You must use the notation we studied. (Either of the notations we studied for cardinality and participation constraints are OK.)
3. Using the technique we studied, derive a relational database schema from the ER diagram.
4. Write CREATE TABLE statements from the schema. Remember to include primary key and foreign key constraints.
5. Write three questions someone might have about the data and write an SQL query for each. The questions should be complex enough that between your three queries you demonstrate your understanding of joins, grouping, aggregation. A little extra credit for particularly interesting queries, if they're explained well.

Part 4: Academic Integrity Statement

I certify that my solutions to this exam are my own work. I did not consult with other people (with the possible exception of seeking clarification from Prof Frankl) and did not use any generative AI systems to help me with the exam. I understand that if the Professor discovers that I have violated this agreement, I will get a zero on the exam and will be referred to the CS Department and/or Tandon's disciplinary officers for possible additional sanctions.

Signature:

Date: