1. This is a closed book exam. But you are allowed to bring hand-written notes on one double-sided 8.5x11 sheet of paper with you.
2. Write your login ID (uniqname) on each sheet in the exam.
3. Write your answers in the answer booklet.
4. You have 120 minutes to complete this exam.
5. If you see typos that are confusing, ask us to clarify. If a problem is ambiguous and you don't have time to clarify, state the assumptions and answer the problem.
6. No electronic devices are allowed, including calculators, smartphones, computers, etc. Please power them down and place in your backpack.
7. Please sign the honor pledge, turn in this exam, and show a picture ID when turning in the exam to a member of the teaching staff. Thank you.

Your exam room number:

Person to your left (if none, left-most person behind you):

Person to your right (if none, right-most person ahead of you):

Write a brief explanation if you are not filling in a name (e.g., "Left-most person in the back row"). Not filling out the above will lead to a deduction of 2 points.

Honor Code Pledge: I have not received or given aid on this exam and have not concealed any violation of the Honor Code

Your signature:

Your name:

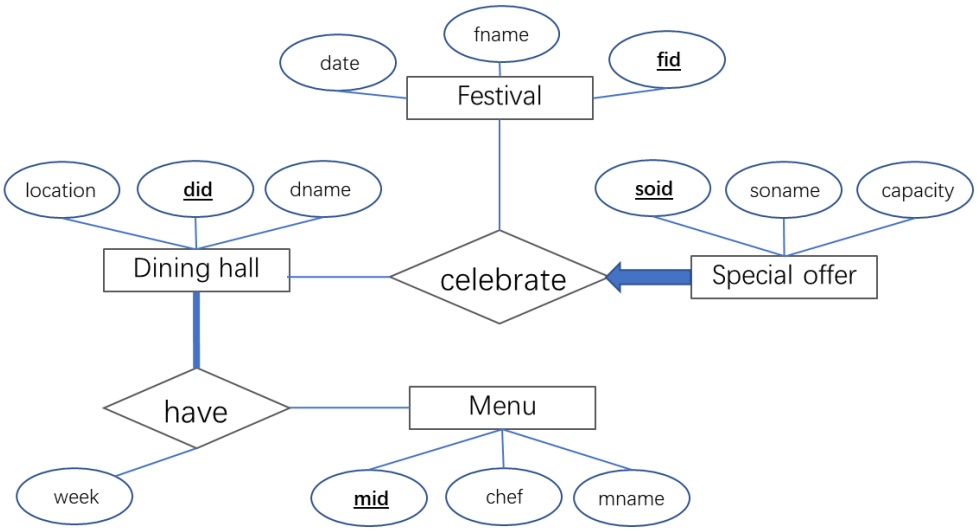
Your uniqname:

Question	Points	Score
Database Design using ER diagrams	23	
Relational Algebra & Relational Calculus	18	
SQL: Winter 2020 EECS 484 Autograder	27	
Functional Dependencies	22	
Total:	90	

(page intentionally left blank. You can use it for rough work. Additional sheets at the back.)

Question 1: (23 points) Database Design using ER diagrams

(a) (6 points) The following ER diagram shows the relationships around dinning hall. Determine whether each of the following is true or false, given the constraints reflected in the ER diagram. Circle your choice on the Answer Book.



i) It is possible to have the same special offer in multiple dining halls for the Thanksgiving event this year.

ii) A dining hall can give special offers even if there's no festival involved.

iii) A dining hall might not celebrate any festivals, and a festival might not be celebrated in any dining hall.

(b) (8 points) Write the CREATE TABLE commands for the two relationships (**celebrate** and **have**). You may find the following rules useful:

i) Use type **INTEGER** for id's and type **VARCHAR(255)** for other attributes.

ii) Each dining hall is required to have a unique dhname.

iii) Each special offer is required to have a distinct soname and a capacity.

iv) If a dining hall is closed, any dependencies on it must be automatically deleted (i.e. have menu, celebrate festival).

For your reference, here is an example for table MENU:

```
CREATE TABLE MENU(
mid INTEGER PRIMARY KEY,
chef VARCHAR(255),
mname VARCHAR(255) NOT NULL);
```

(c) (9 points) Given the ER-diagram on the Answer Book, add entities and relationships to capture constraints from the following description. Notice there may be information that cannot be represented.

i) Dining halls are each monitored by at least one supervisor. Each supervisor is identified by an sid that is unique **only in the scope of that dining hall**. If a dining hall is closed, we no longer keep track of the corresponding supervisor.

ii) There are students, who can be customer or waiter in a dining hall. Students are identified by unique UMID; we also store their sname, age and department in our database.

iii) A customer is a student. Besides their regular information, we also record customers' meal-plan. Each customer may dine in any number of dining halls, hence their meal-plan is not limited to a specific dining hall. And a dining hall may have any number of customers.

iv) A waiter is also a student, that works at exactly one dining hall. A dining hall must have more than one waiters, so that it may serve customers.

Question 2: (18 points) Relational Algebra & Relational Calculus

Village StardeyValley is going to design a database for their village to store the data for villagers' appointments with certain professors. However, some of their villagers can only understand RA and RC but we, developers, should make sure that all of the villagers can understand the statement under our database. Please help us to translate among English words and RA/RC.

Schema:

- Villagers (VID, vname, address)

  (1) VID: villager ID; (2) vname: villager name; (3) address: villager address

- Professors (PID, degree, age, pname)

  (1) PID: professor ID; (2) Degree: highest degree earned; (3) Age: professor age; (4) pname: professor name

- Rooms (RID, rname, size)

  (1) RID: room ID; (2) rname: room name; (3) size: room size (Range - [1,10])

- Appointments (VID, PID, RID, date)

  (1) GID: villager ID; (2) PID: professor ID; (3) RID: room ID; (4) date: appointment date (string type).

  (**Clarification:** In Appointment table, GID, PID and RID are foreign keys refering to table Village, Professor and Room.)

Problems:

(a) (4 points) Please translate the following statement into RA - Find the VIDs of the villagers who don't have any appointment.

(b) (5 points) Please translate the following statement into RA - Find the names of the villagers who have appointments with all professors except professors with name = "Jamie".

(c)

Question 3: (27 points) SQL: [redacted]

Consider a database consisting of the following four tables. Underlined attributes are the primary key of their corresponding tables. Fields with identical names in different tables can be safely assumed to be foreign keys.

- Students(SID, SName, Standing)
- Projects(PID, PName, TotalPoints)
- Teams(TID, SID1, SID2)
- Autograder(PID, TID, Subm ID, Score)

Notes:

- Students table: The attribute Standing can only be one of the five values: "Freshman", "Sophomore", "Junior", "Senior", and "Graduate Student".
- Projects table: There are four projects in total with PID 1, 2, 3, 4 respectively.
- Teams table: Each team consists of exactly two students and each student has to be in a team. We can assume students won't change their team across projects.
- Autograder table: A team may submit multiple times for a project. Each submission has a score. The highest score among all submissions is taken as the final score.

(a) (8 points) Write a query that finds the TID of teams that have the maximum score among all teams for both project 1 (with PID 1) and project 2 (with PID 2). The result should be in descending order.

(b) (6 points) Write a query that creates a view called PassedTeams. In the view, store the TID of teams that have a score over 60 (including 60) for either project 1 (with PID 1) or project 2 (with PID 2). The result should not contain duplicates.

(c) (5 points) Write a query that finds the SID of students who are sophomores and are members of the passing teams in the previous problem. The result should not contain duplicates. You can assume that you are given a correct PassedTeams view (this problem will be graded separately from your answer to the previous problem).

(d) (8 points) Write a query that finds (TID, SID1, SID2) of teams that have fewer than 10 submissions for project 4 (with PID 4) while obtaining full points (TotalPoints in Projects table) for this project. The result should have no duplicates.

Question 4: (22 points) Functional Dependencies

Suppose you have the following table R, with fields ABCDEFG:

```
R(A, B, C, D, E, F, G)
```

The following are the dependencies:

1. $A \rightarrow E$
2. $E \rightarrow F$
3. $AD \rightarrow B$
4. $B \rightarrow C$
5. $D \rightarrow G$

(a) (2 points) What are (minimal) key(s) for the table R? List all of them, one key per line.

(b) (2 points) Is decomposition of R into ABCD and ADEFG a lossless join decomposition?

(c) (2 points) Is decomposition of R into ABCDG and AEF a lossless join decomposition?

(d) (4 points) For the five functional dependencies given for R, identify all that violate BCNF. If none violate, state so. Justification is not required. You may provide it for a partial score in case your answer is not fully correct.

(e) (4 points) For the five functional dependencies given for R, identify all that violate 3NF. If none violate, state so. Justification is not required. You may provide it for a partial score in case your answer is not fully correct.

(f) (8 points) If needed, perform BCNF decomposition on the original relation using the violation dependencies by the order they occur in the question. What are the resulting relations in each iteration and what dependency is lost during decomposition?

Page intentionally left blank

Page intentionally left blank