NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
MIDTERM ASSESSMENT FOR
Semester 2 AY2024/2025

IT5008 Database Design and Programming

March 2025   Time Allowed 60 minutes

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains **9** questions and comprises **6** printed pages, including this page. The last **2** pages is left blank for your working.

2. The total marks for this assessment is **50**.

3. This is a **CLOSED-BOOK** assessment. You are only allowed to refer to one double-sided A4-size paper.

4. Shade and write your student number in the answer sheet. Do **NOT** write your name.

5. Submit only the answer sheet at the end of the assessment.

6. Write all your answers in the answer sheet provided.

7. Your answer must fit into the given space (*i.e.,* number of lines, box, *etc*).

8. Please read the additional instruction for multiple choice (MCQ) and multiple response (MRQ) questions. More specific instructions *may* be given at the beginning of each section, please read them carefully.

   - Any answer not in the space provided will not be graded.
   - For MCQ and MRQ, **shade your answer in the corresponding bubble on the answer sheet**.
   - For MCQ, select the most appropriate answer.
   - For MCQ, if multiple answers are equally appropriate, pick one and shade **ONLY** the chosen answer on the answer sheet. Do **NOT** shade more than one answer.
   - For MRQ, shade **ALL** correct answers. Partial marks *may* be given, but not guaranteed.
   - For MRQ, choose "*None of the above*" only if there is no other appropriate choice. If chosen, you should not choose any other choice.

9. All SQL query in this assessment are run on PostgreSQL 16.

10. We will use the monospace NULL to represent NULL-values. We will represent string with single-quote `'string'`.

Question	Points
1 - 4	14
5 - 8	24
9	12
TOTAL	50

GOOD LUCK

In this paper, we will use the following schema.

```sql
CREATE TABLE employee (
  eid   INT PRIMARY KEY,
  name  VARCHAR(256) NOT NULL
);

CREATE TABLE room (
  name  VARCHAR(32) PRIMARY KEY,
  num   INT NOT NULL,
  floor INT NOT NULL,
  UNIQUE (num, floor),
  CHECK (num > 0),
  CHECK (floor > 0)
);

CREATE TABLE booked_room (
  name    VARCHAR(32) REFERENCES room(name),
  date    DATE,
  purpose VARCHAR(256) NOT NULL,
  PRIMARY KEY (name, date)
);

CREATE TABLE booking (
  eid   INT NOT NULL REFERENCES employee(eid),
  name  VARCHAR(32),
  date  DATE,
  FOREIGN KEY (name, date) REFERENCES booked_room(name, date),
  PRIMARY KEY (name, date)
);
```

We further consider the following **preliminary data**.

employee Table

eid	name
101	'Amy'
102	'Dieter'
103	'Isidor'

booked_room Table

name	date	purpose
'Meeting Room 1'	2025-03-11	'Meeting
'Meeting Room 2'	2025-03-10	'Meeting
'Meeting Room 2'	2025-03-09	'Meeting
'Meeting Room 3'	2025-03-09	'Meeting

room Table

name	num	floor
'Meeting Room 1'	1	1
'Meeting Room 2'	1	2
'Meeting Room 3'	2	1
'Meeting Room 4'	2	3

booking Table

eid	name	date
101	'Meeting Room 2'	2025-03-10
103	'Meeting Room 1'	2025-03-11

You may tear this page as we will only collect the answer sheet.

Data Definition Language   (14 points)

1. (3 points) **(MCQ)** Which of the following INSERT statement will cause an error. Treat each insertion separately starting from the given preliminary data.

   A. **INSERT INTO** room **VALUES** ('*Meeting Room 5A*', 2, 2);

   B. **INSERT INTO** room **VALUES** ('*Meeting Room 5B*', 1, 3);

   C. **INSERT INTO** room **VALUES** ('*Meeting Room 5C*', 3, -1);

   D. **INSERT INTO** room **VALUES** ('*Meeting Room 5D*', 3, 1);

   E. *None of the above.*

2. (3 points) **(MCQ)** We focus on the table room. We want to reimplement the table room differently. Assume that by modifying room, we also modify the other tables referencing room.

   Which of the following alternative capture exactly the same constraint as the schema above?

   A. **CREATE TABLE** room (
      name VARCHAR(32) **UNIQUE NOT NULL**,
      num   INT,
      floor INT,
      PRIMARY KEY (num, floor),
      CHECK (num > 0 **AND** floor > 0)
   );

   B. **CREATE TABLE** room (
      name VARCHAR(32) **UNIQUE**,
      num   INT,
      floor INT,
      PRIMARY KEY (num, floor),
      CHECK (num > 0),
      CHECK (floor > 0)
   );

   C. **CREATE TABLE** room (
      name VARCHAR(32) **UNIQUE NOT NULL**,
      num   INT,
      floor INT,
      PRIMARY KEY (num, floor),
      CHECK (num > 0 **OR** floor > 0)
   );

   D. **CREATE TABLE** room (
      name VARCHAR(32),
      num   INT,
      floor INT,
      PRIMARY KEY (name, num, floor),
      CHECK (num > 0),
      CHECK (floor > 0)
   );

   E. *None of the above.*

3. (4 points) **(MRQ)** Which of the following constraints are not enforced by the given schema? Select all the answers that apply. Choose "*None of the above*" only if no other answers apply. If you choose "*None of the above*", you should not choose other answers.

   A. Rooms can be identified by the floor number (floor) and room number (num).

   B. Rooms can be identified by the room name (name).

   C. No two different employees can book the same room on the same day (date).

   D. No two different rooms can be booked by the same employee on the same day (date).

   E. *None of the above.*

4. (4 points) **(MRQ)** Which of the following constraints are not enforced by the given schema? Select all the answers that apply. Choose "*None of the above*" only if no other answers apply. If you choose "*None of the above*", you should not choose other answers.

   A. Employees are identified by their employee id (eid).

   B. There can be a booked room (*i.e.*, according to booked_room) but not by any employee.

   C. It is possible that an employee does not book any room at all.

   D. It is possible that a room is not booked at all.

   E. *None of the above.*

SQL Query   (24 points)

5. (4 points) **(MRQ)** Which of the following query **cannot be answered** given the schema above? Note that there are no other constraints except those specified by the schema. Select all answers that apply.

   A. Find the different employees (`employee.eid`) that have never booked any room (`room.name`).

   B. Find the different floors (`room.floor`) that have no room (`room.name`).

   C. Find the different room (`room.name`) that has never become a booked room (*i.e.,* according to `booked_room`).

   D. Find the different room (`room.name`) that has never been booked (*i.e.,* according to `booking`).

   E. *None of the above.*

6. (4 points) **(MRQ)** Which of the following query is guaranteed to have no duplicate? Note that you should consider any possible valid instance satisfying the schema and not just the current data. Select all answers that apply.

   A. **SELECT** b.date, r.num, r.floor
      FROM room r, booked_room b
      WHERE b.name = r.name;

   B. **SELECT** b.date, r.name
      FROM room r, booked_room b
      WHERE b.name = r.name;

   C. **SELECT** e.eid, b.name, b.date
      FROM employee e, booking b
      WHERE b.eid = e.eid;

   D. **SELECT** e.name, b.name, b.date
      FROM employee e, booking b
      WHERE b.eid = e.eid;

   E. *None of the above.*

7. (4 points) **(Fill in the Blank)** What is the result of the following query? Fill in the table in the answer sheet. You do not have to use all rows. The order of the rows does not matter.

```
SELECT br.name, br.date, e.name AS employee
FROM booked_room br, booking b, employee e
WHERE br.name = b.name AND b.eid = e.eid
  AND br.name IN (
        SELECT r.name
        FROM room r
        WHERE r.floor = 1
);
```

8. (12 points) **(SQL)** Find the different room that is not booked (*i.e.,* according to `booked_room`) on 2025-03-09. Order the result in **descending order** of room name. Recap that you can write a date by enclosing it in single quote like a string (*e.g.,* `'2025-03-09'`).

   Using the data above, we will have the following result.

name	num	floor
'Meeting Room 1'	1	1
'Meeting Room 4'	2	3

   (a) (6 points) Use algebraic query in your solution.

   (b) (6 points) Use nested query in your solution. Note that you cannot use algebraic query for this part.

Entity-Relationship Diagram   (12 points)

9. (12 points) **(ERD)**   Given the schema, draw one possible entity-relationship diagram such that by using schema translation rule in the lecture (*i.e.,* 3 rules + 3 exceptions), we will get the schema.

END OF PAPER

This space is intentionally left blank.

This space is intentionally left blank.