NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
MIDTERM ASSESSMENT FOR
Semester 1 AY2024/2025

IT5008 Database Design and Programming

October 2024 Time Allowed 60 minutes

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains **10** questions and comprises **6** printed pages, including this page.
2. More specific instructions *may* be given at the beginning of each section. Please read them carefully.
3. Write all your answers in the answer sheet provided. Please read the additional instruction for multiple choice (MCQ) and multiple response (MRQ) questions.
   - 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, 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.
   - If there are no correct answer or no appropriate choices, shade X in the answer sheet. No partial mark given if X is chosen.
4. Shade and write your student number in the answer sheet. Do **NOT** write your name.
5. The total marks for this assessment is 40. Answer **ALL** questions.
6. This is a **CLOSED-BOOK** assessment. You are only allowed to refer to one double-sided A4-size paper.
7. All SQL query in this assessment paper are run on PostgreSQL 16.
8. We will use the monospace `NULL` to represent NULL-values. We will represent string with single-quote `'string'`.
9. The last two pages are left blank. You may use them for your own work.

Question	Points
1 - 4	10
5 - 9	20
10	10
TOTAL	40

Data Definition Language (10 points)

1. (2 points) **(MCQ)** Consider the following table creation.

```
CREATE TABLE q1 (
  attr1 INT PRIMARY KEY,
  attr2 TEXT UNIQUE
);
```

Which of the following instance is a valid instance? In other words, all the rows satisfies the constraint specified.

A.

attr1	attr2
2	'a'
2	'b'

B.

attr1	attr2
2	'a'
1	'a'

C.

attr1	attr2
2	'a'
1	NULL

D.

attr1	attr2
2	'a'
NULL	'b'

Select X if there is no answer.

Notes: C. `attr2` can be `null` and `attr1` has to be unique and not `null`.

2. (2 points) **(MCQ)** Consider the following table creation.

```
CREATE TABLE q2 (
  attr1 INT CHECK (attr1 > 1),
  attr2 TEXT
);
```

Which of the following instance is a valid instance? In other words, all the rows satisfies the constraint specified.

A.

attr1	attr2
2	'a'

B.

attr1	attr2
1	'b'

C.

attr1	attr2
0	''

D.

attr1	attr2
-1	'c'

Select X if there is no answer.

Notes: A. 2 is the only value strictly greater than 1.

3. (2 points) **(MCQ)** Consider the following table creation.

```
CREATE TABLE q3 (
  attr1 INT UNIQUE,
  attr2 TEXT NOT NULL
);
```

Which of the following `iINSERT` statement will **NOT** cause an error. Treat each insertion separately starting from an empty table.

A. `INSERT INTO q3 VALUES (10, 'a'), (10, 'b');`

B. `INSERT INTO q3 VALUES (10), (11);`

C. `INSERT INTO q3 (attr2, attr1) VALUES (10, 'a'), (11, 'b');`

D. `INSERT INTO q3 (attr2) VALUES ('a');`

Select X if there is no answer.

Notes: D. It is better to see other choices one by one.

A. `attr1` is not unique.

B. `attr2` is not given so it will be `NULL` but it cannot be `NULL`.

C. The second element is `attr1`. In this case, `'a'` and `'b'` are not `INT`.

D. We do not have `attr1` and it will be `NULL`, but it is allowed to be `NULL`.

4. (4 points) **(MRQ)** Consider the following relation instance.

student	**course**	**grade**
3118	101	3.0
3118	112	4.0
5609	112	1.0
1423	311	4.5

The table was created with the following `CREATE TABLE` with the `PRIMARY KEY` missing.

```
CREATE TABLE q4 (
  student   INT,
  course    INT,
  grade     NUMERIC,
  -- PRIMARY KEY( ... )
);
```

Select **ALL** possible `PRIMARY KEY` constraints such that the table above does not violate the primary key constraint.

A. `PRIMARY KEY (student)`

B. `PRIMARY KEY (grade)`

C. `PRIMARY KEY (student, course)`

D. `PRIMARY KEY (student, grade)`

E. `PRIMARY KEY (course, grade)`

F. `PRIMARY KEY (student, course, grade)`

Select X if there is no answer.

Notes: We have to find the *unique* combinations without worrying about the meaning. In this case, the following are unique: B, C, D, E, F.

SQL Query (20 points)

For the next 5 question, we will use the following table `student`, `course`, and `enrolment`.

student

student	**name**
3118	'Alice'
5609	'Bob'
1423	'Charlie'

course

course	**name**
101	'Intro'
112	'Algo'
311	'Advanced'

enrolment

student	**course**	**grade**
3118	101	3.0
3118	112	4.0
5609	112	1.0
1423	311	4.5
1423	112	4.0

Notes: If you wish to try it out on your own, you can use the following simplified schema.

```
CREATE TABLE student (
  student INT,
  name TEXT
);

CREATE TABLE course (
  course INT,
  name TEXT
);

 CREATE TABLE enrolment (
   student INT,
   course INT,
   grade NUMERIC
 );

 INSERT INTO student VALUES (3118, 'Alice');
 INSERT INTO student VALUES (5609, 'Bob');
 INSERT INTO student VALUES (1423, 'Charlie');

 INSERT INTO course VALUES (101, 'Intro');
 INSERT INTO course VALUES (112, 'Algo');
 INSERT INTO course VALUES (311, 'Advanced');

 INSERT INTO enrolment VALUES (3118, 101, 3.0);
 INSERT INTO enrolment VALUES (3118, 112, 4.0);
 INSERT INTO enrolment VALUES (5609, 112, 1.0);
 INSERT INTO enrolment VALUES (1423, 311, 4.5);
 INSERT INTO enrolment VALUES (1423, 112, 4.0);
```

5. (4 points) 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 DISTINCT e.course, e.grade
FROM enrolment e
WHERE e.grade >= 2.0;
```

Notes: The keyword DISTINCT is to be applied to the entire (course, grade) pair. We have a duplicate (112, 4.0). Only one will be printed. Note that you may put 3 or 3.0 for the grade.

course	grade
101	3.0
112	4.0
311	4.5

6. (4 points) 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 s.name AS student, c.name AS course, e.grade
FROM enrolment e, student s, course c
WHERE s.student = e.student
  AND c.course = e.course
  AND s.name = 'Alice';
```

Notes:

student	course	grade
'Alice'	'Intro'	3.0
'Alice'	'Algo'	4.0

7. (4 points) 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 c.name, MAX(e.grade) AS max
FROM enrolment e, course c
WHERE grade < 4.5
  AND e.course = c.course
GROUP BY e.course, c.name;
```

Notes:

name	max
'Intro'	3.0
'Algo'	4.0

8. (4 points) Find the different student who has taken the `'Algo'` course with a grade higher than or equal to `2.0`. Return the student name and the grade. Order the output in **ascending** order of grade followed by **descending** order of name.

   You must use only **simple query**. Using the data above, we will have the following result.

name	grade
'Charlie'	4.0
'Alice'	4.0

Notes:

```
SELECT s.name, e.grade
FROM student s, enrolment e, course c
WHERE e.grade >= 2.0
  AND c.name = 'Algo'
  AND s.student = e.student
  AND c.course = e.course
ORDER BY e.grade ASC, s.name DESC;
```

9. (4 points) For each student, find their average grade. Return the student name and the average grade. Order the output in **descending** order of average grade followed by **descending** order of name.

   Do not forget to rename the attributes but you do not have to truncate/round the average. Using the data above, we will have the following result.

name	average
'Charlie'	4.25
'Alice'	3.50
'Bob'	1.00

Notes:

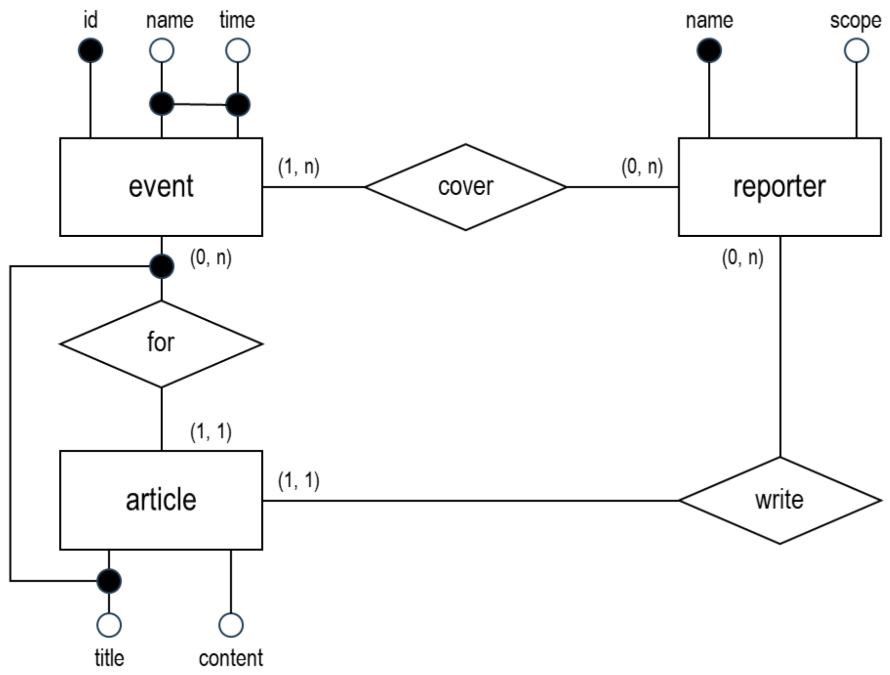
```
SELECT s.name, AVG(e.grade)
FROM student s, enrolment e
WHERE s.student = e.student
GROUP BY s.student, s.name
ORDER BY AVG(e.grade) DESC, s.name DESC;
```

ER Diagram (10 points)

10. (10 points) You are starting your work at a newspaper helping to build their database system. In this newspaper, reporters write articles about events. Events are covered by reporters. You are given the following constraints.

    1. Each event can be uniquely identified by either the event id or the combination of event name and event time.
    2. Each reporter can be uniquely identified by their name. The job scope of the reporter is also recorded.
    3. The title of an article can uniquely identify the article only among the article written for a given event. The content of the article is also recorded.
    4. A reporter may write 0 or more articles.
    5. An article must be written by exactly one reporter.
    6. An event may be covered by different reporters but it must be covered by at least one reporter.
    7. A reporter may cover 0 or more event.
    8. An event may be written in 0 or more articles.
    9. An article must be written for exactly one event.
    10. An event may have 0 or more article written about them.

    Draw the ER diagram in **our convention** that captures as much constraints as possible.



Notes:

END OF PAPER

This page is intentionally left blank.

This page is intentionally left blank.