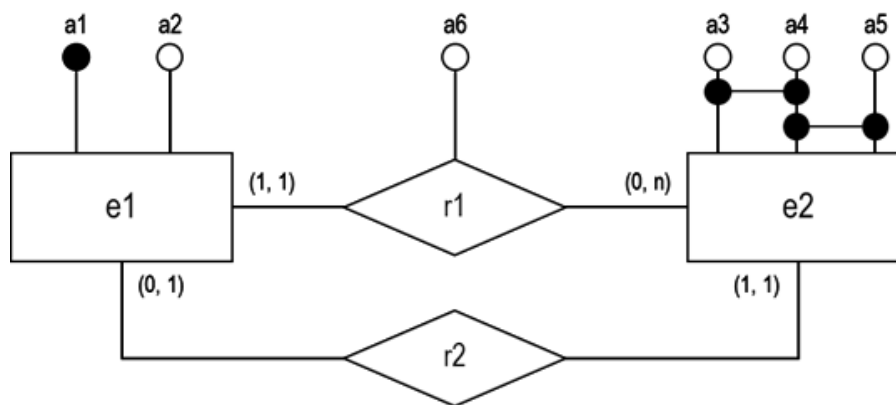


Midterm: 2024/2025 Semester 2

For the first 7 questions, we will be using the following entity-relationship diagram.



One possible schema translation is shown below. Any candidate keys can be primary key. Changing the primary key will also change the foreign key.

```
1 CREATE TABLE e1r1 (
2   a1 INT PRIMARY KEY,
3   a2 INT NOT NULL,
4   a6 INT NOT NULL,
5   a3 INT NOT NULL,
6   a4 INT NOT NULL,
7   FOREIGN KEY (a3, a4) REFERENCES e2r2(a3, a4) -- need ALTER TABLE
8 );
9
10 CREATE TABLE e2r2 (
11   a3 INT,
12   a4 INT,
13   a5 INT NOT NULL,
14   a1 INT NOT NULL UNIQUE REFERENCES e1r1(a1), -- candidate key
15   PRIMARY KEY (a3, a4), -- primary key
16   UNIQUE (a4, a5) -- candidate key
17 );
```

1. How many tables are created?

Notes: B

- e1 and r1 are merged.
- e2 and r2 are merged.

79.64% got this question correct.

2. Foreign key in the table for **e2**.

Notes: AC

- (a1) referencing **e1.a1**
- (a1) referencing **r1.a1**

61.09% got this question correct.

3. Foreign key in the table for **e1**.

Notes: ABCD

- (a3, a4) referencing (**e2.a3**, **e2.a4**)
- (a4, a5) referencing (**e2.a4**, **e2.a5**)
- (a3, a4) referencing (**r2.a3**, **r2.a4**)
- (a4, a5) referencing (**r2.a4**, **r2.a5**)

60.49% got this question correct.

4. Primary key in the table for **r1**.

Notes: A

(a1) only.

69.6% got this question correct.

5. Primary key in the table for **r2**.

Notes: ABC

(a1), (a3, a4), or (a4, a5).

12.16% got this question correct.

6. Minimum number of **e2**.

Notes: B: 1

Cannot be 0 because of relationship **r1**, we must have all **e1** participating. They can participate with a single **e2**.

45.59% got this question correct.

7. Maximum number of **e2**.

Notes: D: 100

All 100 must participate in **r2**. Since for each participation, must have the corresponding **e1**, the max is the same as **e1**.

65.35% got this question correct.

Info: For MRQ, the grading was done as follows.

- If “None of the above” is selected, 0 mark is given.
- For each incorrect options chosen, subtract 0.5 mark. Capped at 0 mark.

For SQL, we run the SQL against a different data set. The insertion is randomized to ensure the **ORDER BY** is needed. In the case of least/most, we ensure that there are multiple results which may result in **LIMIT 1** giving wrong result.

The next 4 queries uses the ICPC dataset from assignment 1.

8. Last contest (*i.e.*, find all largest).

Notes: There can be other answers. Use of “LIMIT 1” often leads to incorrect result.

```
1 SELECT c.name, c.year FROM contest c, site s
2 WHERE c.site = s.name
3     AND c.date >= ALL (
4         SELECT c1.date FROM contest c1, site s1
5         WHERE c1.site = s1.name
6             AND s1.region = s.region AND c1.year = c.year
7     )
8 ORDER BY c.name ASC;
```

46.81% got this question correct.

9. Average (*i.e.*, aggregate function).

Notes: There can be other answers.

```
1 SELECT c.site, c.year, ROUND(AVG(solve), 2) AS avg
2 FROM contest c, participate p, site s
3 WHERE c.site = p.site AND c.year = p.year
4     AND c.site = s.name AND p.solve > 0 AND s.region = 'Europe'
5 GROUP BY c.site, c.year
6 ORDER BY AVG(solve) DESC; -- cannot use 'avg'
```

62.92% got this question correct.

10. Least popular (*i.e.*, find all smallest).

Notes: There can be other answers. Note that we assume that there is always a team participating. Alternatively, we use OUTER JOIN with the respective COALESCE to produce the COUNT of 0.

The test data assume there is always a team participating. Use of “LIMIT 1” often leads to incorrect result

```
1 SELECT c.site, c.year
2 FROM contest c, participate p
3 WHERE c.site = p.site AND c.year = p.year
4 GROUP BY c.site, c.year
5 HAVING COUNT(team) <= ALL (
6     SELECT COUNT(p1.team)
7     FROM contest c1, participate p1
8     WHERE c1.site = p1.site AND c1.year = p1.year
9     GROUP BY c1.site, c1.year
10 )
11 ORDER BY c.site, c.year;
```

33.43% got this question correct.

11. Different Region (*i.e.*, simple queries).

Notes: There can be other answers.

```
1 SELECT t.name, c.site, c.year
2 FROM university u, team t, participate p, contest c, site s
3 WHERE u.name = t.univ AND t.name = p.team AND p.site = c.site
4     AND p.year = c.year AND c.site = s.name AND u.region = 'Europe'
5     AND u.region <> s.region
6 ORDER BY team ASC;
```

55.62% got this question correct.