Questions

Question 1

(1 mark) Consider the following database schema consisting of two relations where **X** denote some attribute in relation R.

```
create table R (
    A integer,
    B integer,
    C integer not null,
    D integer unique,
    E integer unique not null,
    primary key (A),
    unique (B,C)
);

create table S (
    I integer,
    J integer,
    primary key (I),
    foreign key (J) references R(X)
);
```

Select all the statements that are true about the database schema.

1. The database schema is valid if X is replaced by A.
2. The database schema is valid if X is replaced by B.
3. The database schema is valid if X is replaced by C.
4. The database schema is valid if X is replaced by D.
5. The database schema is valid if X is replaced by E.
6. All of the above statements are false.

Question 2

(1 mark) Questions 2 and 3 are based on the same database schema shown below.

```
create table R (
    A integer primary key,
    B integer
);

create table S (
    C integer primary key,
    D integer,
    A integer references R(A)
);

create table T (
    E integer primary key,
    F integer,
    A integer references R(A)
);
```

Let r, s, and t, denote, respectively, the number of tuples in relations R, S, and T.

Let **m** denote the maximum number of tuples in the result of the natural join of R and S; i.e., among all the legal database instances with r records in R, s records in S, and t records in T, the largest cardinality of $R \bowtie S$ is m.

Select the most appropriate statement about m.

1. m = r.
2. m = s.
3. m = min(r,s).
4. m = max(r,s).
5. m = r + s
6. m = r $\times$ s
7. All of the above statements are false.

Question 3

(1 mark) Questions 2 and 3 are based on the same database schema shown below.

```
create table R (
    A integer primary key,
    B integer
);

create table S (
    C integer primary key,
    D integer,
    A integer references R(A)
);

create table T (
    E integer primary key,
    F integer,
    A integer references R(A)
);
```

Let r, s, and t, denote, respectively, the number of tuples in relations R, S, and T.

Let **n** denote the maximum number of tuples in the result of the natural join of R, S, and T; i.e., among all the legal database instances with r records in R, s records in S, and t records in T, the largest cardinality of $(R \bowtie S) \bowtie T$ is n.

Select the most appropriate statement about n.

1. $n = r$.
2. $n = s$.
3. $n = t$.
4. $n = \min(r, s, t)$.
5. $n = \max(r, s, t)$.
6. $n = \min(r \times s, r \times t)$
7. $n = \max(r \times s, r \times t)$
8. $n = r + s + t$.
9. $n = r \times s$.
10. $n = r \times t$.
11. $n = s \times t$.

12. $n = r \times s \times t$.

13. All of the above statements are false.

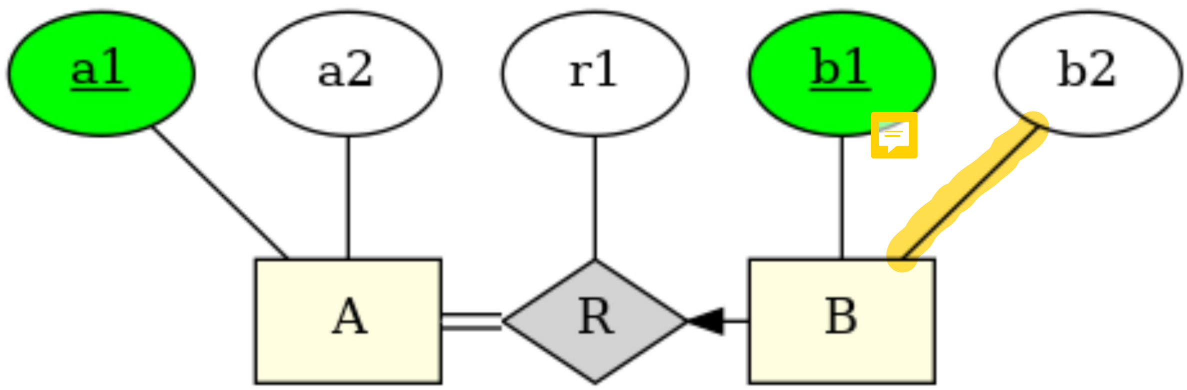
Question 4

(2 marks) Consider the following database schema which is translated from the ER data model shown in the question's attachment file. Assume that all the attributes have non-null integer values.

```
CREATE TABLE B (
    b1  integer primary key,
    b2  integer not null
);

CREATE TABLE AR (
    a1  integer primary key,
    a2  integer not null,
    b1  integer unique not null,
    r1  integer not null,
    foreign key (b1) references B(b1)
);
```

Write down all the **errors** in the translated relational schema. The errors could be **missing constraints** (i.e., constraints that are specified in the ER data model but are not captured by the relational schema) or **extraneous constraints** (i.e., constraints that are enforced by the relational schema but are not specified by the ER data model).



Question 5

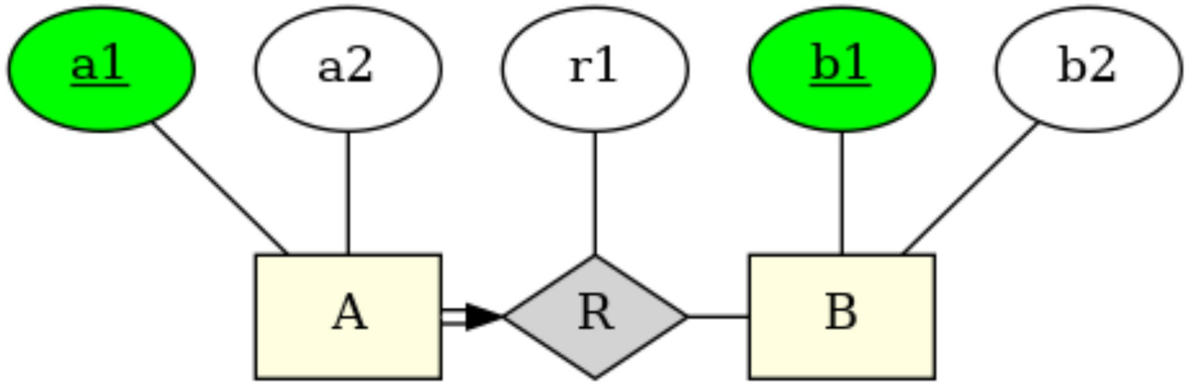
(2 marks) Consider the following database schema which is translated from the ER data model shown in the question's attachment file. Assume that all the attributes have non-null integer values.

```
CREATE TABLE B (
    b1  integer primary key,
    b2  integer not null
);

CREATE TABLE R (
    a1  integer primary key,
    b1  integer not null,
    r1  integer not null,
    foreign key (b1) references B(b1)
);

CREATE TABLE A (
    a1  integer primary key,
    a2  integer not null,
    foreign key (a1) references R(a1)
);
```

Write down all the **errors** in the translated relational schema. The errors could be **missing constraints** (i.e., constraints that are specified in the ER data model but are not captured by the relational schema) or **extraneous constraints** (i.e., constraints that are enforced by the relational schema but are not specified by the ER data model).



Key and Total participation on A wrt R

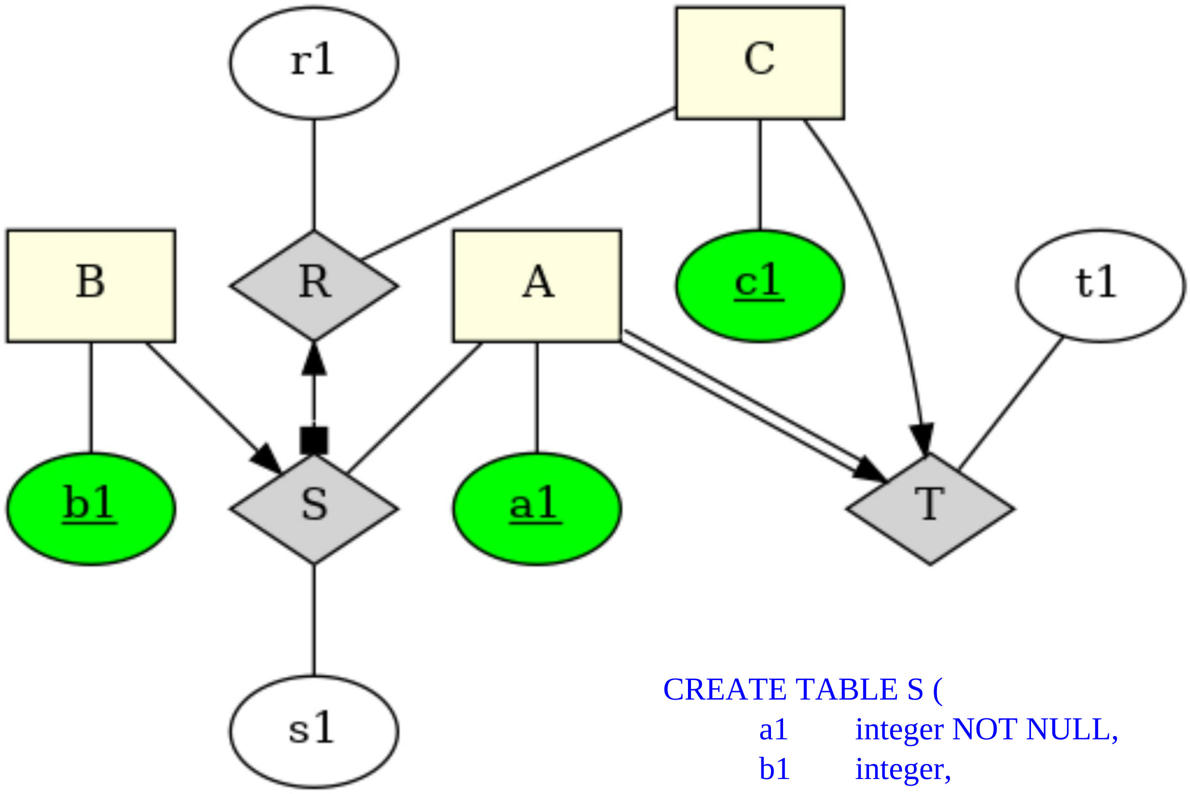
Both constraints are enforced by foreign key constrain from A to R in relational schema

Given a binary relationship R between entity sets A and B, each relationship instance (a,b) in R must satisfy the entity membership constraint that (1) a is some instance in A and (2) b is some instance in B.
While the translated relational schema enforces that each b1 in R references some record in B, this is not enforced for a1 in R. Thus, there's a missing constraint in the relational schema.
Similarly here, you should answer the question directly (i.e., state the errors in the translated relational schema). If you just translate the ER data model into a relational schema, you are not answering the question.

Question 6

(6 marks) Consider the ER data model shown in the question's attachment file. Assume that all the attributes have non-null integer values.

Translate the ER data model into a relational schema using SQL's CREATE TABLE statements. Your relational schema should capture as many of the constraints in the ER data model as possible and should not impose any extraneous constraints (i.e., constraints that are not specified by the ER data model).




```
CREATE TABLE B (
        b1      integer,
        PRIMARY KEY (b1)
);
CREATE TABLE C (
        c1      integer,
        PRIMARY KEY (c1)
);
CREATE TABLE AT (
        a1      integer,
        c1      integer NOT NULL,
        t1      integer NOT NULL,
        PRIMARY KEY (a1),
        UNIQUE (c1),
        FOREIGN KEY (c1) REFERENCES C
);
```

```
CREATE TABLE S (
        a1      integer NOT NULL,
        b1      integer,
        s1      integer NOT NULL,
        PRIMARY KEY (b1),
        FOREIGN KEY (a1) REFERENCES AT,
        FOREIGN KEY (b1) REFERENCES B
);
CREATE TABLE R (
        b1      integer,
        c1      integer NOT NULL,
        r1      integer NOT NULL,
        PRIMARY KEY (b1),
        FOREIGN KEY (c1) REFERENCES C,
        FOREIGN KEY (b1) REFERENCES S
);
```

Question 7

Questions 7 to 10 require reading/writing relational algebra queries. These questions are based on assignment 1's database schema and the syntax for valid relational algebra expressions (which are shown in the question's attachment file).

(1 mark) Consider the following three relational algebra queries (Q1, Q2, and Q3):

Query Q1:

```
R1 = project{eid}(Works ~ select{pbudget > 10}(rename{eid:mid}(Projects)));
R2 = project{eid}(Works ~ select{pbudget <= 10}(rename{eid:mid}(Projects)));
Q1 = R1 - R2
```

Query Q2:

```
R2 = project{eid}(Works ~ select{pbudget <= 10}(rename{eid:mid}(Projects)));
Q2 = Engineers - R2
```

Query Q3:

```
R1 = project{eid}(Works ~ select{pbudget > 10}(rename{eid:mid}(Projects)));
Q3 = project{eid}(Works) - R1
```

Given a legal database instance D and a query Q, let **Q(D)** denote the output table computed by Q on D.

Select all the statements that are true.

1. Q1 and Q2 are equivalent queries.
2. Q1 and Q3 are equivalent queries.
3. Q2 and Q3 are equivalent queries.
4. For every legal database instance D, if Q1(D) contains the tuple t, then Q2(D) also contains t.
5. For every legal database instance D, if Q2(D) contains the tuple t, then Q1(D) also contains t.
6. For every legal database instance D, if Q1(D) contains the tuple t, then Q3(D) also contains t.
7. For every legal database instance D, if Q3(D) contains the tuple t, then Q1(D) also contains t.

8. For every legal database instance D, if Q2(D) contains the tuple t, then Q3(D) also contains t.
9. For every legal database instance D, if Q3(D) contains the tuple t, then Q2(D) also contains t.
10. All of the above statements are false.

Question 8

Questions 7 to 10 require reading/writing relational algebra queries. These questions are based on assignment 1's database schema and the syntax for valid relational algebra expressions (which are shown in the question's attachment file). For convenience, we show the information on the relations and their attributes here:

- Offices (oid, address)
- Departments (did, dbudget, oid, eid)
- Employees (eid, did)
- Engineers (eid)
- Managers (eid)
- Projects (pid, pbudget, eid)
- Works (pid, eid, hours)
- Areas (aid)
- Specializes (eid, aid)

(2 marks) Write a relational algebra query to find all projects that (1) have at least one employee who specialized in the area with aid = 1 working on that project, but (2) do not have any employee who specialized in the area with aid = 2 working on that project. The schema of the output table is (pid), where pid denote the project identifier.

```
R1(pid) = set of projects with at least one engineer who specializes in area aid="1"
R2(pid) = set of projects with at least one engineer who specializes in area aid="2"

R1 = project{pid}(select{aid="1"}(Works ~ Specializes));
R2 = project{pid}(select{aid="2"}(Works ~ Specializes));
R1-R2
```

Question 9

Questions 7 to 10 require reading/writing relational algebra queries. These questions are based on assignment 1's database schema and the syntax for valid relational algebra expressions (which are shown in the question's attachment file). For convenience, we show the information on the relations and their attributes here:

- Offices (oid, address)
- Departments (did, dbudget, oid, eid)
- Employees (eid, did)
- Engineers (eid)
- Managers (eid)
- Projects (pid, pbudget, eid)
- Works (pid, eid, hours)
- Areas (aid)
- Specializes (eid, aid)

(2 marks) Write a relational algebra query to find the project with the **second highest** project budget among all the projects. Assume that the pbudget values are unique (i.e., no two projects have the same value for pbudget). The schema of the output table is (pid), where pid denote the project identifier.

```
R1 = Set of (p1,b1,p2,b2), where bi is the project budget of project pi, and b1 < b2.
project{p1}(R1) = Set of pid where there is at least one project whose budget is higher than that of pid
R2 = Set of pid where there are at least two projects whose budgets are higher than that of pid

R1 = project{p1,b1,p2,b2}(
        select{b1 < b2}(
                rename{pid:p1,pbudget:b1}(Projects) * rename{pid:p2,pbudget:b2}(Projects)
        )
);

R2 = project{p1}(
        select{b2 < b3}(
                R1 * rename{pbudget:b3}(Projects)
        )
);

project{p1}(R1) - R2
```

Question 10

Questions 7 to 10 require reading/writing relational algebra queries. These questions are based on assignment 1's database schema and the syntax for valid relational algebra expressions (which are shown in the question's attachment file). For convenience, we show the information on the relations and their attributes here:

- Offices (oid, address)
- Departments (did, dbudget, oid, eid)
- Employees (eid, did)
- Engineers (eid)
- Managers (eid)
- Projects (pid, pbudget, eid)
- Works (pid, eid, hours)
- Areas (aid)
- Specializes (eid, aid)

(2 marks) Write a relational algebra query to find all engineers who work on exactly **n - 1** projects, where n is the number of projects in the database (i.e., n is the cardinality of the Projects table). Note that your answer must be correct for any value of n; i.e., your answer should not assume any specific value of n. The schema of the output table is (eid), where eid denote the employee identifier.

```
R1(eid,pid) = Engineer eid who does not work on pid
R2(eid) = Engineer eid who does not work on at least 2 projects

R1 = Engineers * project{pid}(Projects) - project{eid,pid}(Works);
R2 = project{eid}(R1 ~{(eid = eid2) and (pid < pid2)} rename{eid:eid2,pid:pid2}(R1));
project{eid}(R1) - R2
```