Midterm exam 10 March 2019, answers

Database Systems (City University of Hong Kong)

CS3402 Database Systems

2018/19 Semester B

Mid-Term Test (A)

March 13, 2019 (Wednesday)

Student Name:	
Student ID:	
Seat No:	

Instructions:

1. This paper consists of 1 question in 5 pages, including this page.
2. Answer ALL questions in the space provided within each question.
3. Notes and books are allowed.
4. Calculators, smart devices, electronic devices and communication devices are NOT allowed.
5. Time allowed: 60 minutes

Question 1. Given below the database schema with three tables, answer the following questions.

```
CREATE TABLE PERSON
( PID   INT             NOT NULL,
  PNAME VARCHAR(20)     NOT NULL,
  PRIMARY KEY(PID));

CREATE TABLE EVENT
( EID        INT              NOT NULL,
  ENAME      VARCHAR(20)      NOT NULL,
  START_TIME TIMESTAMP        NOT NULL,
  END_TIME   TIMESTAMP        NOT NULL,
  PRIMARY KEY(EID),
  CONSTRAINT EVENT_TIME_CK CHECK(START_TIME < END_TIME));

CREATE TABLE INVITED
( PID        INT     NOT NULL,
  EID        INT     NOT NULL,
  ATTENDED   INT     NOT NULL,
  PRIMARY KEY(PID, EID),
  FOREIGN KEY(PID) REFERENCES PERSON(PID),
  FOREIGN KEY(EID) REFERENCES EVENT(EID));
```

Note that the attribute ATTENDED has the value of 1 if the person attended the event; otherwise, the attribute ATTENDED has the value of 0.

(a) Write a SQL query that returns the number of persons who have attended to events with a name that starts with ‘Food’? (10 marks)

```
SELECT COUNT(*)
FROM EVENT E, INVITED I
WHERE E.EID = I.EID AND E.ENAME LIKE ‘Food%’ AND I.ATTENDED = 1;
```

(b) Write a SQL query that returns the event ids and names of the events that the most persons have attended. (10 marks)

```
SELECT E.EID, E.ENAME
FROM EVENT E, INVITED I
WHERE E.EID = I.EID AND I.ATTENDED = 1
```

```
GROUP BY E.EID, E.ENAME
HAVING COUNT(*) >=
                              ALL (SELECT COUNT(*)
                              FROM INVITED I2
                              WHERE I2.ATTENDED = 1
                              GROUP BY I2.EID);
```

(c) Write a SQL query that returns the ids and names of all persons who have attended to two distinct events that have the same start time and end time. (10 marks)

```
SELECT DISTINCT P.PID, P.PNAME
FROM PERSON P, EVENT E1, EVENT E2, INVITED I1, INVITED I2
WHERE P.PID = I1.PID AND P.PID = I2.PID AND I1.ATTENDED = 1 AND I2.ATTENDED = 1
AND I1.EID = E1.EID AND I2.EID = E2.EID AND E1.EID <> E2.EID AND E1.START_TIME =
E2.START_TIME AND E1.END_TIME = E2.END_TIME;
```

(d) Write a SQL query that returns the id and name of each event and how many persons who have attended it. (10 marks)

```
SELECT E.EID, E.ENAME, COUNT(*)
FROM EVENT E, INVITED I
WHERE E.EID = I.EID AND I.ATTENDED = 1
GROUP BY E.EID, E.ENAME
```

(e) Write a SQL query that returns ids and names of persons who have NOT attended the most events. (10 marks)

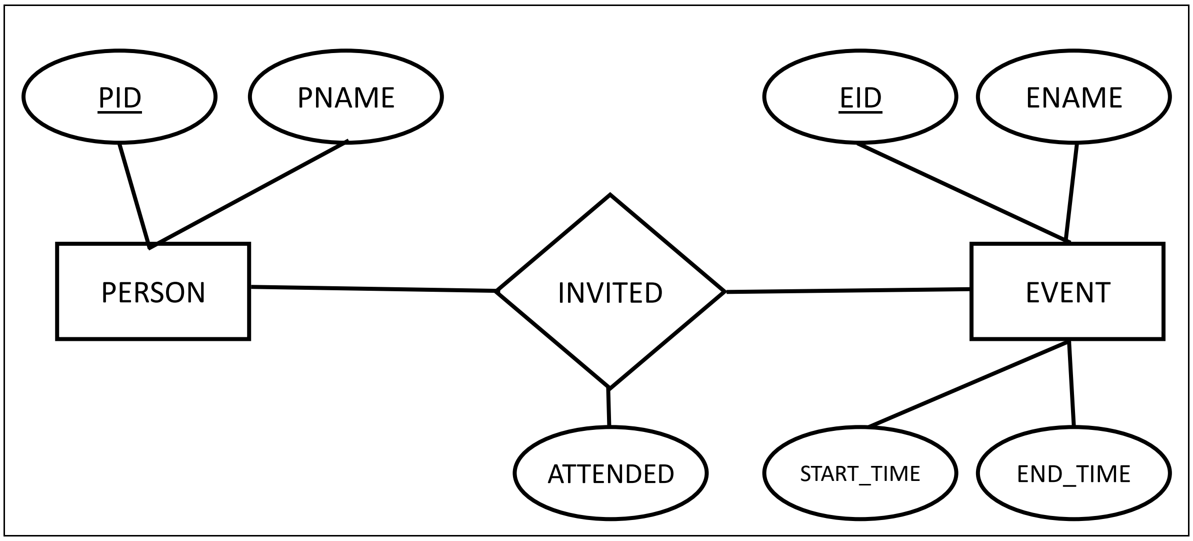
```
SELECT P.PID, P.PNAME
FROM PERSON P, INVITED I
WHERE P.PID = I.EID AND I.ATTENDED = 0
GROUP BY P.PID, P.PNAME
HAVING COUNT(*) >=
                              ALL (SELECT COUNT(*)
                              FROM INVITED I2
                              WHERE I2.ATTENDED = 0
                              GROUP BY I2.PID);
```

(f) Write SQL queries to update table INVITED to set attribute ATTENDED to 1 if the person did not attend the event and ATTENDED to 2 if the person attended the event. You must label the execution order of your queries with (1), (2), (3), etc.

```
    (1)  UPDATE INVITED SET ATTENDED = 2 WHERE ATTENDED = 1
```

(2) UPDATE INVITED SET ATTENDED = 1 WHERE ATTENDED = 0

(g) Draw an ER diagram for the database schema. A person can attend more than one event. An event can be attended by more than one person. (10 marks)



(h) Write a relational algebra that returns ids and names of events that have been attended by persons with name ‘Peter’. (10 marks)

$$tmp1 \leftarrow \sigma_{\text{PNAME="Peter" AND ATTENDED=1}}(\text{PERSON} * \text{INVITED})$$
$$RESULT \leftarrow \pi_{\text{EID,ENAME}}(tmp1 * \text{EVENT})$$

(i) Write a relational algebra that returns the ids and names of persons who have NOT attended any event. (10 marks)

$$RESULT \leftarrow \pi_{\text{PID,PNAME}}(\text{PERSON}) - \pi_{\text{PID,PNAME}}(\sigma_{\text{ATTENDED=1}}(\text{PERSON} * \text{INVITED}))$$

(j) Write a relational algebra that returns the ids and names of persons who have attended events with name “Food Festival” or “Food Fair”. (10 marks)

$$RESULT \leftarrow \pi_{\text{PID,PNAME}}\left(\sigma_{\text{(ENAME="Food Festival" OR ENAME="Food Fair") AND ATTENDED=1}}(\text{EVENT} * \text{PERSON} * \text{INVITED})\right)$$