CSIE30600/CSIEB0290 Database Systems, Fall 2020
Midterm Exam

Name: ________________ Student ID:_______________

1. **(30 points)** Based on the problem type in parentheses (True/False, fill in the Blank (FIB), or briefly answer (ANS)), answer the following questions.

1-1 What is **data independence**? What is the *main mechanism* to facilitate data independence? *Why* is it *important* to database systems?

Capacity to change the schema at one level w/o having to change the schema at the next higher level._Three-level architecture with schema mapping is the main mechanism. This allow us to change the structure of data at one level without affecting existing operations of other levels. Each level can be improved and optimized independently.___ (ANS)

1-2 A database system contains not only data itself but also the description of data (meta-data). What is the **term** we use to describe such a property? _self-describing______ (FIB)

1-3 Since database systems for different applications have different domains, they must use *different* DBMSs. _____False______ (True/False)

1-4 What are the advantages of **three-level architecture** in database systems? What is its relationship with **schema mapping**?

______Data independence, separation of concerns, modularity, flexibility____
______The structure and schema of each level is independent from other levels.____
______Schema mapping is the process of converting data in one level to the ____
______schema structure in another level._When one level is changed, only the _____
______mapping need to be changed without affecting other_______ _______ (ANS)

1-5 The **result** of a relational expression is always a *relation*. ____ True _______ (True/False)

1-6 Give an example of a **multi-value composite attribute.** ___________{ (Colledge, Major, Year, Degree) }_____________ (FIB)

1-7 Briefly describe what does **referential integrity constraint** mean.

______When a referencing tuple refers to a referenced tuple through a foreign key,
______the referenced tuple must exists._________________________________

____

____

____ (ANS)

1-8 In the relational model, what is the **main property** for a set of attributes to form a key? __can uniquely identify a tuple________________ (FIB)

1-9 A **candidate key** is a minimal superkey and therefore must be a superkey with fewest number of attributes. _____False_____ (True/False)

1-10 A **DELETE** operation may violate all types of integrity constraints. ____False_____ (True/False)

1-11 What does it mean by specifying the **CASCADE** option for database updates? What is(are) its implication?

______CASCADE option means that if the referenced rows are updated, the referencing row is also updated accordingly.

____

____

____ (ANS)

1-12 As the name suggest, the Structured Query Language (**SQL)** is a language just for querying existing databases. _____False____ (True/False)

1-13 What are the correspondences between the SELECT, FROM, and WHERE clauses of SQL to the relational algebra operations?

______SELECT → projection attributes___________________________________

______FROM → the target relations, ×, join________________________________

______WHERE → selection condition, join condition_________________________

____

____ (ANS)

1-14 How do we describe the property of **relational algebra** that the result of any expression is also a relation? ____The algebra is closed._________ (FIB)

1-15 What does it mean by saying that the set of six relational algebra operations {σ, π, $\cup$, ρ, $-$, $\times$} is **complete**? What about other operations?

__The set of operations is complete since any other relational algebra__________

___operation can be expressed as a sequence of operations from the set._________

___Other operations are provided for convenient of expression._________________

____

____(ANS)

2. **(15 points)** You are called upon to design a database system for an international conference on the COVID-19 pandemic. In the system, we need to keep track of the following types of data and the associated information (at least):

- **Countries**: Information of countries around the world. A country record contains a name, geographical region, population, size(area), classification (developed, developing, or underdeveloped).
- **DiseaseStatus**: A daily update about the disease status of all countries around the world. Each record contains the country name, total confirmed cases, total deaths, total recoveries, daily new cases, daily reported deaths, and date.
- **ConferenceRooms**: A room has a room number, name, and capacity of the room.
- **Representatives**: The representatives from all countries. A representative has an ID, name, country, and title.
- **Attenders**: Officially invited experts to attend the meeting. An attender has an ID, name, country, title, seniority level, and subjects of expertise. Note that an attender may have expertise in more than one subjects
- **Subjects**: The subjects of discussion in the conference. A subject record contains a subject name, subject area (medical, economical, …), and severity level,
- **Meetings**: A meeting has an ID, title, subject of discussion, room of the meeting, and date. A meeting should invite all attenders with expertise in the subject of discussion.

2a) Design a set of relational schemas for the target database above. Draw the schema diagram for your design. Indicate the primary keys, foreign keys and any integrity constraints that need to be satisfied.

2b) Are the schemas you just designed good enough for keeping various information about the conference? For example, where do you keep the information about the all subjects of expertise of a particular attender? What about the attenders of each Meeting?

2c) If the answer to question 2(b) above is negative, then how do you refine your schemas to cover that information? If the answer is positive, can your design be further improved in any other way?

(This page is left empty for you to answer the exam question.)

2a)

Countries

<u>name</u>	region	population	size	class

DiseaseStatus

<u>cname</u>	**<u>date</u>**	total_cases	total_deaths	total_revoveries	new_cases	new_deaths
F.K.						

ConferenceRooms

<u>rnumber</u>	rname	capacity

Representatives

<u>pID</u>	pName	pCountry	pTitle
		F.K.	

Attenders

<u>aID</u>	aname	acountry	title	seniority
		F.K.		

Subjects

<u>sName</u>	area	severity

Meetings

<u>mID</u>	mTitle	subject	room	date
		F.K.	F.K.	

2b)

Expertises

<u>aID</u>	**sName**
F.K.	F.K.

3. **(5 points)** Let **r** and **s** be two relations:

r

A	B	C	D
a1	b1	10	d1
a1	b2	30	d2
a2	b1	10	d3
a2	b3	20	d4
a2	b1	20	d4
a1	b1	30	d1
a1	b2	30	d3
a1	b3	30	d4
a3	b1	20	d1
a4	b1	20	d2
a4	b2	40	d3
a2	b2	20	d2
a3	b1	10	d4
a2	b2	40	d3
a4	b1	40	d2
a4	b2	30	d2
a4	b1	40	d1

s

B	C	D
b1	10	d1
b2	35	d2
b2	25	d3
b1	20	d2
b3	30	d4
b2	40	d2
b2	15	d3

Evaluate the result of the relational algebra expression $r \div \pi_{B,D}(\sigma_{C \geqq 30}(s))$

b2 d2

b3 d4

A	C
a1	30
a2	20

4. **(25 points)** Consider the following simple relational schema:

Students(s<u>ID</u>, sname, dID, year, advisorID**)**

Professors(pID, pname, dID, rank**)**

Department(<u>dID</u>, dname, office, chairID**)**

Courses(cID, cname, dID, year, pID)

Grades(cID, year, sID, grade)

Specify relational algebra expressions to answer the following queries.

4a) Find the department names and years of all courses offered by professor Wu.

$\pi_{dname,year}(Courses*(\sigma_{pname='Wu'}Professors)*Department)$

4b) Find the names, departments(names) and years of all students whose advisor is professor Wu.

$\pi_{sname,\ dname,\ year}((Students \bowtie_{advisorID=pID} (\sigma_{pname='Wu'}Professors))$

$*Department)$

4c) Find the names of departments with more than 300 students but less than 10 professors.

$\pi_{dname}((\pi_{dID}({}_{dID}\mathcal{F}_{count(sID)>300}Students) \cap$

$\pi_{dID}({}_{dID}\mathcal{F}_{count(pID)<10}Professors)) * Department)$

4d) Find the names of students who obtain grade "A" on ALL courses of the CSIE department.

$\pi_{sname,cID}(\sigma_{grade='A'}(Students*Courses*Grades)) \div (\pi_{cID}(Courses$

$* \sigma_{dname='CSIE'}Department))$

5. **(25 points)** This problem is based on the relations:

Students(<u>sid: string</u>, sname: string, age: integer, gender: string, country: string)

Courses(<u>cid: string</u>, cname:string, credit: integer, year: integer, instructor:string)

Enrolled(<u>sid: string, cid: string, year:integer</u>, grade:real)

The key fields are underlined, and the domain of each field is listed after the field name. Thus, ***sid*** is the key for **Students, *cid*** is the key for **Courses**, and ***sid***, ***cid*** and ***year*** together form the key for **Enrolled.** The meaning of these relations is straightforward; for example, **Enrolled** has one record per student-course-year for each student who enrolled in that class. Express the following queries in SQL statements.

5a) Find the names of all courses that the student with sid "610971" enrolled in year 2020.

5b) Find the names and years of courses enrolled by both student "610971" and "70483".

5c) Find the names of students who have not enrolled in any class at year 2020.

5d) Find the names of the students who have enrolled all courses offered by Professor Wu.

5e) Find the names of the courses which have more than 50 students enrolled.

5a)

```
SELECT      C.cname
FROM        Courses C, Enrolled E
WHERE       C.cid=E.cid AND E.sid='610971' AND E.year=2020;
```

5b)

```
SELECT      C.cname, C.year
FROM        Courses C,   Enrolled E
WHERE       C.cid=E.cid AND E.sid='610971';
INTERSECT
SELECT      C.cname, C.year
FROM        Courses C,   Enrolled E
WHERE       C.cid=E.cid AND E.sid='70483';
```

5c) There are many ways to express this query.

```
SELECT   sname
FROM   Students
WHERE   sid   NOT IN (
      SELECT sid
```

```
    FROM    Enrolled
    WHERE   year=2020);

(SELECT  sname
FROM     Students)
EXCEPT
(SELECT S.sname
FROM Students S, Enrolled E
WHERE E.year=2020 AND E.sid=S.sid);

SELECT  S.sname
FROM    Students S
WHERE   NOT EXISTS (
    SELECT      *
    FROM        Enrolled
    WHERE       Enrolled.sid=S.sid AND Enrolled.year=2020);
```

5d)

```
SELECT  S.sname
FROM    Students as S
WHERE   NOT EXISTS (
    (SELECT   cid
    FROM    Courses
    WHERE   instructor='Wu')
    EXCEPT
    (SELECT   C.cid
    FROM    Courses as C, Enrolled as E
    WHERE   C.cid=E.cid AND E.sid=S.sid)
);
```

5e)

```
SELECT  C.cname
FROM    Courses as C
WHERE   C.cid   IN   (
    SELECT    E.cid
    FROM    (
        SELECT   E.cid, COUNT(*)
        FROM     Enrolled as E
```

```
            GROUP BY E.cid
            HAVING COUNT(*) > 50)
        )
);
```