Sample midterm exam CMPT 321 Fall 2017

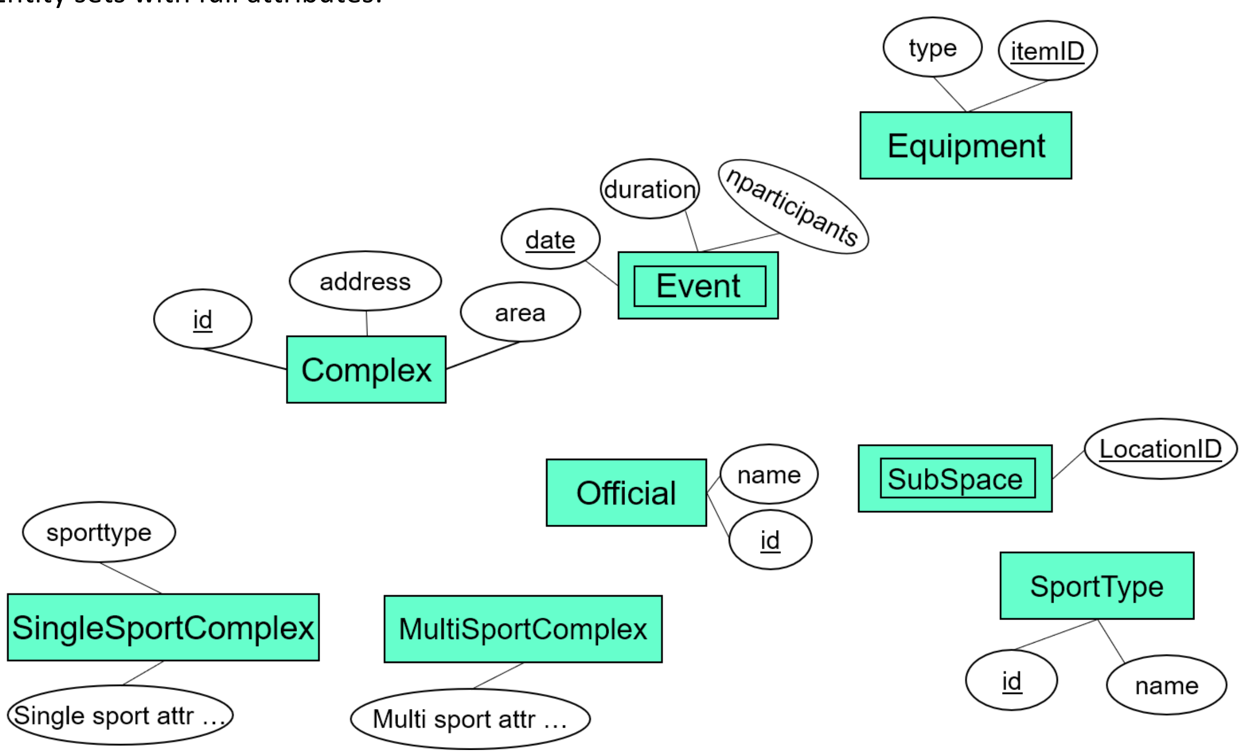
The following narrative describes a simplified version of the organization of Olympic facilities.

The Olympic facilities are divided into sports complexes. Sports complexes are divided into one-sport and multisport types. Multisport complexes have areas of the complex designated for each sport with a location indicator (e.g., center, NE corner, and so on). A complex has a location, chief organizing individual, total occupied area, and so on. Each complex holds a series of events (e.g., many different races). For each event there is a planned date, duration, number of participants, number of officials, and so on. A roster of all officials will be maintained together with the list of events each official will be involved in. Different equipment is needed for the events (e.g., goal posts, poles, parallel bars). The two types of facilities (one-sport and multisport) will have different types of information.

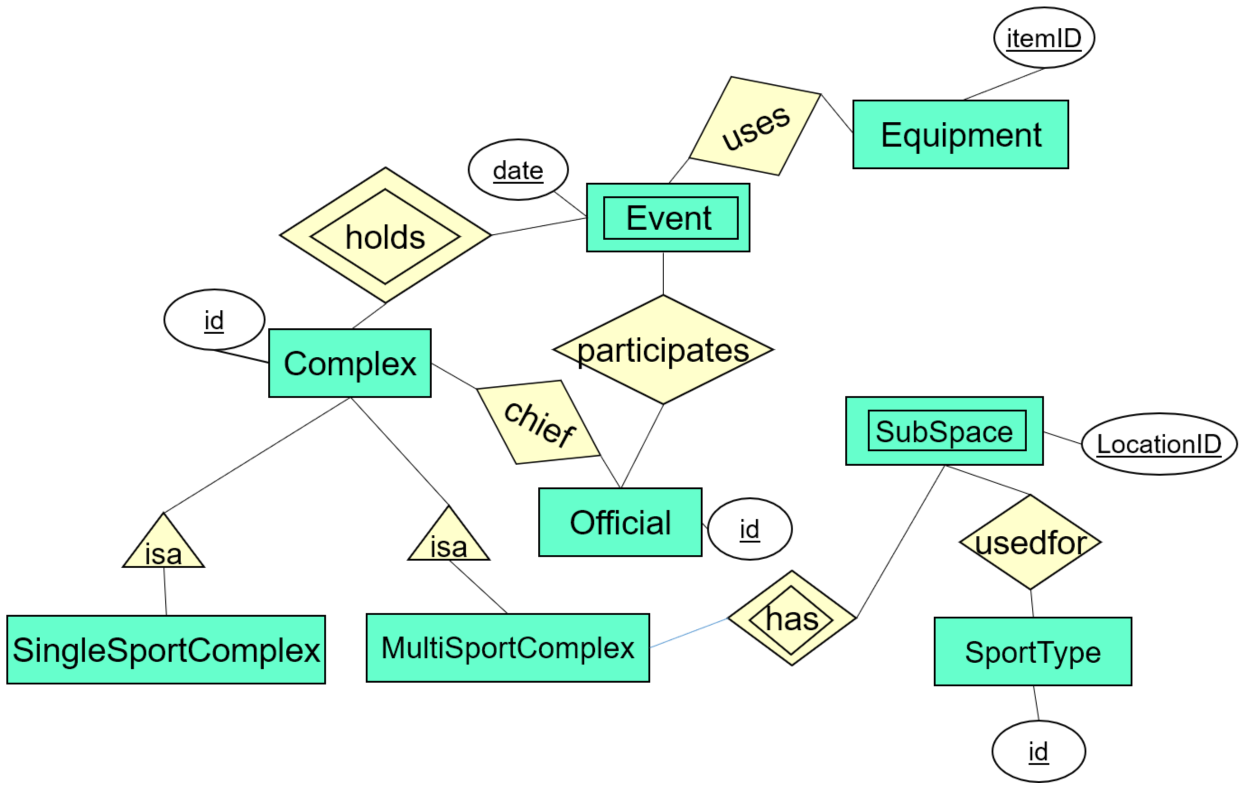
Question 1 [7 points]

Draw an E/R diagram that shows the entity sets, attributes (including keys), and relationships for this application.

Entity sets with full attributes:



Relationships, attributes reduced to keys only:



Question 2 [5 points]

Translate the E/R design for your previous exercise into a relational schema <u>using the Object-oriented approach.</u>

Entities:

Complex (<u>id</u>,address,area)

Event (<u>complexID</u>, <u>date</u>,duration,nparticipants)

Equipment (<u>id</u>,type)

Official (<u>id</u>,name)

SportType (<u>id</u>, name)

SingleSportComplex (<u>id</u>, address, area, sporttype, …)

MultySportComplex (<u>id</u>, address, area, …)

SubSpace (<u>complexID</u>, <u>locationID</u>)

Relationships:

complexChief (<u>complexID</u>, <u>officialID</u>)

participates (<u>complexID</u>, <u>date</u>, <u>officialID</u>)

subspace (<u>complexID</u>, <u>locationID</u>)

usedFor (<u>complexID</u>, <u>locationID</u>, <u>sportID</u>)

eventEquipment (<u>complexID</u>, <u>date</u>, <u>itemID</u>)

Question 3 [5 points]

Consider this schema:

One (<u>this</u>, something)
Two (<u>before</u>, <u>after</u>)
Two[before] ⊆ One [this][1)]

[1)]This means that the values in *Two.before* are a subset of values in *One.this*, that is each value of *Two.before* must exist in column *One.this*.

One has 10 tuples. *Two* has 200 tuples.
What do we know about the minimum and maximum number of tuples in the results of the following queries?

a. One X Two

min: 2000
max: 2000

b. Two $\ltimes_{before=this}$ One

min: 200
max: 200

c. $\sigma_{this='b'}$(One)
min: 0
max:1

d. $\sigma_{this='a'}$ (Two $\bowtie_{before = this}$ One)

min: 0
max: 200

Question 4 [15 points]

Consider the Supplier-Parts-Catalog schema presented below.

Suppliers (<u>sid</u>, sname, address)

Parts (<u>pid</u>, pname, color)

Catalog (<u>sid</u>, <u>pid</u>, cost)

The Catalog relation lists the prices charged for parts by Suppliers.

State what the following queries compute:

1. $\pi_{sname}(\ \pi_{sid}\ ((\sigma_{color='red'}Parts) \bowtie \sigma_{cost<100}Catalog)) \bowtie Suppliers)$

Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars.

2. $\pi_{sname}(\ \pi_{sid}\ ((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers))$

This Relational Algebra statement does not return anything because of the sequence of projection operators. Once the sid is projected, it is the only field in the set. Therefore, projecting on sname will not return anything.

3. $(\pi_{sname}((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers)) \cap$
   $(\pi_{sname}((\sigma_{color='green'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers))$

Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

4. $(\pi_{sid}((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers)) \cap$
   $(\pi_{sid}((\sigma_{color='green'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers))$

Find the Supplier ids of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

5. $\pi_{sname}((\pi_{sid,sname}\ ((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog)) \bowtie Suppliers) \cap$
   $(\pi_{sid,sname}\ ((\sigma_{color='green'}Parts) \bowtie (\sigma_{cost<100}Catalog)) \bowtie Suppliers)))$

Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

6. $\pi_{sname}((\sigma_{color='green'}Parts) \bowtie Suppliers) - \pi_{sname}((\sigma_{color!='green'}Parts) \bowtie Suppliers)$

Find the names of suppliers that supply exclusively green parts.

7. $P1 = \rho_{name1,cost1} (\pi_{pname,cost} (\sigma_{color='green'}Parts) \bowtie Catalog)$
   $P2 = \rho_{name2,cost2} (\pi_{pname,cost} (\sigma_{color='green'}Parts) \bowtie Catalog)$
   $R = \pi_{name1} (\sigma_{cost1<cost2} (P1 \times P2))$
   $S = \pi_{pname}(Parts) – R$

Find the most expensive green part(s).

8. $A = \pi_{pname} (Parts)$
   $B = \pi_{color} (Parts)$
   $Y = A \times B$
   $Z = Y - \pi_{pname,color} (Parts)$
   $S = \pi_{pname} (Parts) - \pi_{pname} (Z)$

Find parts (names) that are offered in **every** possible color listed in the database.

Question 5

Write the following SQL queries for the above database.

1. For every supplier, print the name of the supplier and the total number of parts that she supplies.

SELECT sname, sid COUNT (pid)
FROM suppliers s, parts p
WHERE s.pid = p.pid
GROUP BY sname, sid;

2. For every supplier that supplies at least five parts, print the name and price of the most expensive part that she supplies.

CREATE VIEW supplier5 AS
SELECT sid, sname
FROM supplier s, parts p

```
WHERE s.pid = p.pid
GROUP BY sid, sname
HAVING COUNT(p.pid) >=5;

CREATE VIEW maxpricepart AS
SELECT sid, MAX (price) AS maxprice
FROM catalog
GROUP BY sid;

SELECT sid X, sname, pname, price
FROM supplier5 s, catalog c, part p
WHERE c.sid = s.sid AND
c.pid = p.pid AND
c.price =
        (SELECT maxprice
        FROM maxpricepart m
        WHERE m.sid = X.sid);
```