Name: _____ Student#: _____ Date: _____

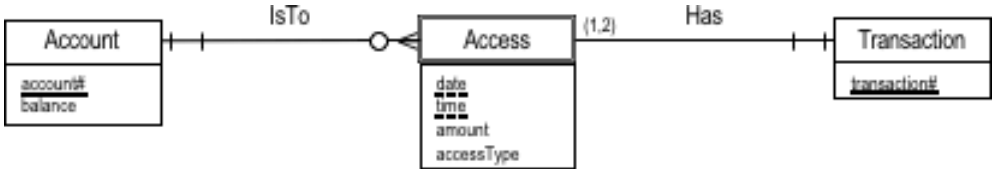
COMP 3311: Database Management Systems

Midterm Review

Q1 The three entities Competitor, Event and Trial are used to schedule and score athletic competitions such as gymnastics, diving and figure skating. A competitor is described by a unique competitor number and name. An event is described by a unique name. Each trial has a number that is unique for a given competitor and event. An athletic competition can have several events and competitors. Each competitor may enter several events and each event can have many competitors. The focal points of the competitions are the trials. Each trial is an attempt by one competitor to turn in the best performance possible in one event. A competitor receives an overall score for each trial entered.

In the space below, construct an E-R diagram using the lecture notes E-R notation showing how the three entity types are related. Show, as necessary, relationships, relationship attributes, generalizations, and any constraints on relationships that can be clearly inferred either from the problem description, from real-world knowledge or from common sense. Cardinality and participation constraints that cannot be inferred should be left unspecified. All relationships should be named. Only relationships that are necessary to show how the entities are related should be shown. Any weak entities and their dependent relationship(s) should be clearly identified

Q2 An outline of the reduction of a banking E-R schema to relation schemas is given below. Complete the reduction for each relation schema by adding any required additional attributes, underlining the key and writing the referential integrity constraints including actions that apply to the relation schema, if any, below it.



Account(account#, balance

Access(date, time, amount, accessType

Transaction(transaction#

Q3 **Given:** R(A, B, C, D, E) F = {A→BC, B→AC, AD→E, E→D}

3.1 Which of the following sets is a subset of $\{A, D\}^+$?

a) {A, B}

b) {B, C, D}

c) {E}

d) All the above

e) None of the above

3.2 Which of the following is a candidate key for R?

a) AD

b) AE

c) BD

d) BE

e) All the above

3.3 For the following decomposition, which statement is true?

R_1(A, B, C) R_2(A, D, E)

a) The decomposition is 3NF, lossless join and dependency preserving.

b) The decomposition is 3NF, lossless join but not dependency preserving.

c) The decomposition is 3NF, dependency preserving, but not lossless join.

d) The decomposition is lossless join, dependency preserving but not 3NF.

e) The decomposition is 3NF, but neither lossless join nor dependency preserving.

3.4 For the following decomposition, which statement is true?

R_1(A, B, C) R_2(A, E) R_3(D, E)

a) The decomposition is BCNF, lossless join and dependency preserving.

b) The decomposition is BCNF, lossless join, but not dependency preserving.

c) The decomposition is BCNF, dependency preserving, but not lossless join.

d) The decomposition is lossless join, dependency preserving, but not BCNF.

e) The decomposition is BCNF, but neither lossless join nor dependency preserving.

3.5 Consider relation R(A, B, C, D, E). Given the functional dependencies in the first column of the table, complete the table accordingly. The first row is given as an example.

Functional dependencies	**List all candidate keys for R**	**Give a maximal decomposition of R into 3NF**	**Decompose R into BCNF (*where possible choose a dependency preserving decomposition*)**
{A→BCDE}	A	R(A, B, C, D, E)	R(A, B, C, D, E)
{C→D}			
{A→BC, D→AE}			

Q4 In the following tables, keys are underlined and foreign keys are in italics. Primary keys are **not null**.

Proposal(pid, *sid*, title, area)	// The foreign key sid is **not null** and corresponds to the sid of the submitter who **submitted** the proposal.
Submitter(sid, name, email)	// A submitter may submit several proposals.
Reviewer(rid, name, email, expertise)	
Review(*pid*, *rid*, score)	// pid and rid are foreign keys corresponding to the pid of the proposal that was **reviewed** by reviewer rid. The values for score are in the range [1..5]. A reviewer may review several proposals.

4.1 Given the foreign keys and assuming the referential integrity constraints are included in the SQL create statements, what should be the create order?

4.2 Construct an E-R diagram that reduces to the above tables.

Q5 Use the following tables for this question.

Proposal(pid, *sid*, title, area)	Reviewer(rid, name, email, expertise)
Submitter(sid, name, email)	Review(*pid*, *rid*, score)

5.1 Write a relational algebra query to return the names of all reviewers who reviewed a proposal in the Database area, submitted by Prof. Dimitris (i.e., the submitter name is Dimitris).

5.2 Write a relational algebra query to return the IDs of reviewers who have only reviewed proposals in the area of their expertise (i.e., these reviewers have reviewed at least one proposal and have not reviewed any proposal in an area different from their expertise).

5.3 Write a relational algebra query that gives the same result as the following SQL query.

```
select sid
from Proposal
group by sid
having count(*)>=2;
```

Proposal(pid, *sid*, title, area) Reviewer(rid, name, email, expertise)

Submitter(sid, name, email) Review(*pid*, *rid*, score)

5.4 Write an equivalent SQL query ***without sub-queries*** for the following SQL query.

```
select name
from Reviewer
where rid in (select rid
              from Review
              where score=5
                    and pid in (select pid
                                from Proposal
                                where area='Database'));
```

5.5 Write an SQL query to return the title and average score of each proposal in the Database area.

5.6 Write an SQL query to return the name, maximum and minimum score of each reviewer who reviewed exactly five proposals.

5.7 Express in English the result of the following SQL query.

```
select temp.title
from (select P.title as title, avg(score) as avgScore
      from Proposal P, Review R
      where P.pid=R.pid
      group by P.pid, P.title) as temp
where temp.avgScore>(select avg(avgScore)
                     from temp);
```


You must upload this completed exercise as a single PDF document to Canvas by **11 p.m. today.**