COMP 3311
DATABASE MANAGEMENT SYSTEMS

MIDTERM REVIEW

MAIN TOPICS

- Entity-Relationship (E-R) Model and Database Design
- Relational Model and Relational Database Design
- Relational Algebra
- Structured Query Language (SQL)

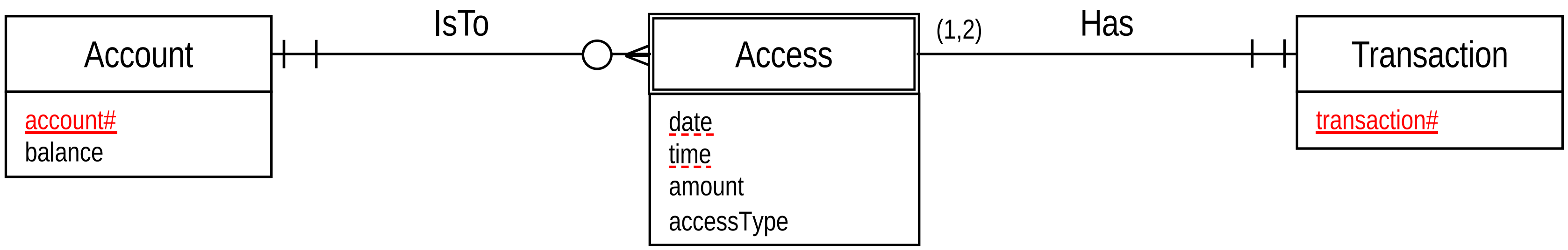
QUESTION 1

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.

Construct an E-R diagram showing how the three entity types are related. Show relationship constraints and attributes. Weak entities and their dependent relationship(s) should be clearly identified.

QUESTION 2

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 that apply to the relation schema, if any, below it.



Account(account#, balance)

Access(*transaction#, account#, date, time*, amount, access-type)
    foreign key account# references Account(account#) on delete cascade
    foreign key transaction# references Transaction(transaction#) on delete cascade

Transaction(transaction#)

QUESTION 3

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 of the above

e) None of the above

{A, D}$^+$ = {A, D}

= {A, D, B, C} using A⟶BC

= {A, D, B, C, E} using AD⟶E

QUESTION 3 (CONT'D)

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

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

a)	AD	$AD^+ =$	{A, D, B, C, E}	using A⟶BC and D⟶E
b)	AE	$AE^+ =$	{A, E, B, C, D}	using A⟶BC and E⟶D
c)	BD	$BD^+ =$	{B, D, A, C, E}	using B⟶AC and D⟶E
d)	BE	$BE^+ =$	{B, E, A, C, D}	using B⟶AC and E⟶D

e) All of the above

QUESTION 3 (CONT'D)

Given: R(A, B, C, D, E) $F = \{A \rightarrow BC, B \rightarrow AC, AD \rightarrow E, E \rightarrow D\}$

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.

A and B are superkeys in R_1; A, D and E are prime attributes in R_2.
$R_1 \cap R_2 = A$ is a key of R_1. $\{A \rightarrow BC, B \rightarrow AC\}$ in R_1; $\{AD \rightarrow E, E \rightarrow D\}$ in R_2.

QUESTION 3 (CONT'D)

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

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.

A and B are superkeys in R_1; AE is a superkey in R_2; E is a superkey in R_3.
$R_1 \cap R_2$=A is a key of R_1. $R_2 \cap R_3$=E is a key of R_3. AD→E is not preserved.

QUESTION 3 (CONT"D)

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.

- In the second column, list all candidate keys for R.
- In the third column, provide a maximal decomposition of R into 3NF (we only decompose when there is a violation of 3NF) – if R is already in 3NF just write R(A, B, C, D, E) instead of a decomposition.
- In the fourth column, do the same for BCNF decomposition. If there are multiple options, choose any dependency preserving decomposition.

Functional dependencies	Candidate keys for R	Decompose R into 3NF	Decompose R into BCNF
$\{A \rightarrow BCDE\}$	A	R(A, B, C, D, E)	R(A, B, C, D, E)
$\{C \rightarrow D\}$	ABCE	R_1(A, B, C, E) R_2(C, D)	R_1(A, B, C, E) R_2(C, D)
$\{A \rightarrow BC, D \rightarrow AE\}$	D	R_1(A, B, C) R_2(D, A, E)	R_1(A, B, C) R_2(D, A, E)

QUESTION 4

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

Proposal(<u>pid</u>, *sid*, title, area) // The foreign key sid is **not null** and corresponds to the sid of the submitter who **submitted** the proposal.

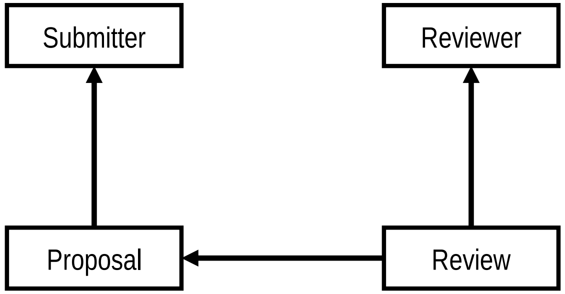
Submitter(<u>sid</u>, name, email) // A submitter may submit several proposals.

Reviewer(<u>rid</u>, name, email, expertise)

Review(<u>*pid, rid*</u>, 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.

QUESTION 4 (CONT'D)

4.1 According to the foreign keys, in what order do the tables need to be created?



Create Order

1. Submitter and Reviewer *in any order*
2. Proposal
3. Review

Proposal(pid, *sid*, title, area)

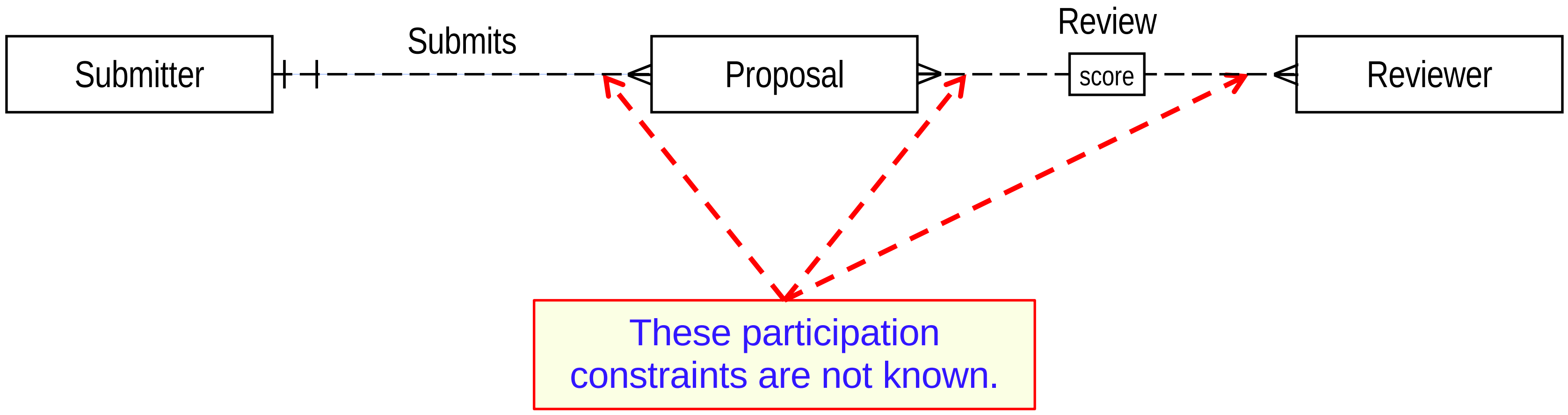
Submitter(sid, name, email)

Reviewer(rid, name, email, expertise)

Review(*pid, rid*, score)

QUESTION 4 (CONT'D)

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



Proposal(pid, *sid*, title, area)

Reviewer(rid, name, email, expertise)

Submitter(sid, name, email)

Review(*pid, rid*, score)

QUESTION 5

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).

$\pi_{Reviewer.name}(\sigma_{Proposal.area='Database' \wedge Submitter.name='Dimitris'}(\text{Reviewer JOIN}_{rid} \text{ Review JOIN}_{pid}$

$\text{Proposal JOIN}_{sid} \text{ Submitter}))$

Proposal(pid, *sid*, title, area)

Reviewer(rid, name, email, expertise)

Submitter(sid, name, email)

Review(*pid, rid*, score)

QUESTION 5 (CONT'D)

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).

$(\pi_{rid}\text{Review}) - (\pi_{\text{Reviewer.rid}}(\sigma_{\text{Reviewer.expertise<>Proposal.area}}(\text{Reviewer JOIN}_{rid}\text{ Review JOIN}_{pid}$

$\text{Proposal})))$

Proposal(pid, *sid*, title, area)

Reviewer(rid, name, email, expertise)

Submitter(sid, name, email)

Review(*pid, rid*, score)

QUESTION 5 (cont'd)

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;
```

$\pi_{P1.sid}(\sigma_{P1.sid=P2.sid \wedge P1.pid<>P2.pid}(\rho_{P1}(Proposal) \times \rho_{P2}(Proposal)))$

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

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

QUESTION 5 (cont'd)

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'));
```

```
select name
from Reviewer R, Review RV, Proposal P
where R.rid=RV.rid
      and RV.pid=P.pid
      and score=5
      and area='Database';
```

QUESTION 5 (cont'd)

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

```
select title, avg(score)
from Proposal, Review
where Proposal.pid=Review.pid
      and area='Database'
group by Proposal.pid, title;
```

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

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

QUESTION 5 (cont'd)

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

```
select name, max(score), min(score)
from Reviewer, Review
where Reviewer.rid=Review.rid
group by Reviewer.rid, name
having count(*)=5;
```

Proposal (pid, *sid*, title, area)

Submitter (sid, name, email)

Reviewer (rid, name, email, expertise)

Review (*pid, rid*, score)

QUESTION 5 (cont'd)

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);
```

Find the titles of the proposals whose average score is greater than the average score of all proposals.

Proposal (pid, *sid*, title, area)

Submitter (sid, name, email)

Reviewer (rid, name, email, expertise)

Review (*pid, rid*, score)