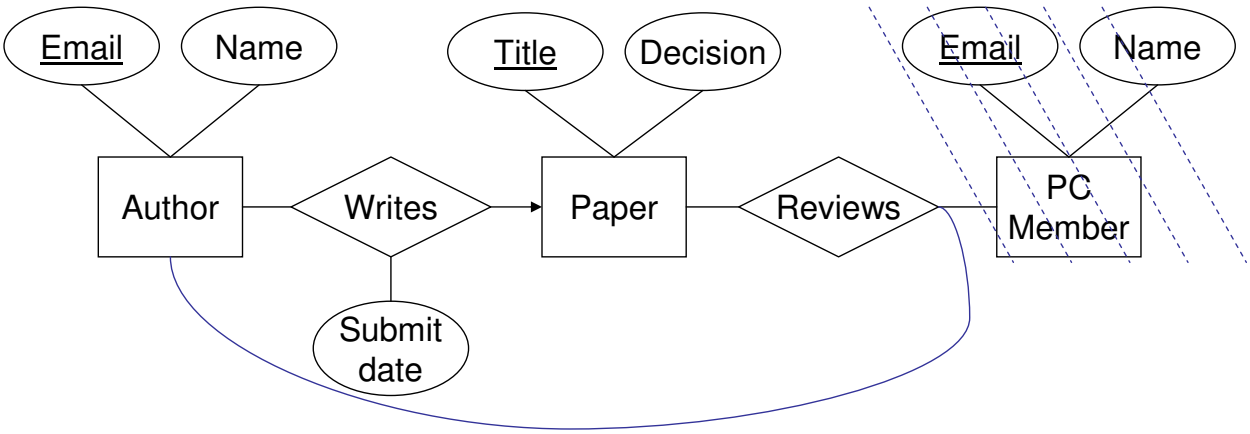
CS541 Fall 2002 Midterm Solutions, October 14, 2002
Prof. Chris Clifton

I have included some notes on how I expect to grade these – these are subject to change. These are not the only correct answers, so if you think you got it right but didn't get this answer, don't worry.

My expectation is that someone who gets at least 31 of the possible 38 points is on track to getting an A in the course. With 22 points, a B is a reasonable expectation. If you get fewer than 15, we need to talk.

1 Data Modeling (15 minutes, 14 points)

Given the following ER diagram, for a database to manage papers submitted to a conference:



1.1 Convert to Relational (3 minutes, 3 points)

Give relations corresponding to the ER diagram. Don't worry about normalization - just give the conversion from the diagram.

Author(Email, Name)
Paper(Title, Decision)
Writes(Email, Title, Submit Date)
PCMember(Email, Name)
Reviews(Title, PEmail)

One to two points for getting the basics, one point for keys.

1.2 Give Functional Dependencies (3 minutes, 6 points)

What functional dependencies hold on this data? For each dependency, note briefly (one sentence) how you derived it (is it inherent in the ER diagram? Something you think should be true?)

AE → AN AE	Email is a key for Author	(1 point)
T → T D	Key	(1 point)
PE → PE PN	Key	(1 point)
T AE → S	Key for writes	(1 point)
AE → T D S	Many to one relationship	(1 point)

One additional point for quality / correctness of reasoning.

1.3 Modifications (3 minutes, 2 points)

Currently "people" are referenced in two different ways: As PC Members, and as Authors. It is quite possible that someone could be both a PC member and an Author, giving us redundant information. Modify the above diagram to eliminate this redundancy.

(See above.)

1.4 Constraints (4 minutes, 3 points)

Come up with at least one other constraint you feel should hold on the data. Can you represent this in the ER model? If so, sketch how, if not, suggest why not.

An author shouldn't review his own paper. This can't be represented in the Entity Relationship model. A multiway relationship (as opposed to the way I drew it), but ER doesn't support a comparison between entity sets.

2 Relational Design (20 minutes, 14 points)

You are to develop a relational schema for a database to be used for conference registration and planning. The goal is to represent when and where talks are being held, who is speaking and attending each, and contact information of the people involved.

You are given the following data:

UniversalRelation(Speaker, SAddress, SEmail, Participant, Address, Email, Room, Hour, Talk)

the following functional dependencies (use capitalized letters above for abbreviations):

E → P	SE → S
P → E	S → SE
P → A	S → SA
T → RH	T → S
RH → T	

and the additional constraint that every Speaker must be registered as a Participant, with the Speaker email (SE) and address (SA) the same as the address used when they registered as a participant (E,A).

2.1 Boyce-Codd Normal Form (7 minutes, 5 points)

Give a BCNF decomposition that maintains lossless join.

UniversalRelation(Speaker, SAddress, SEmail, Participant, Address, Email, Room, Hour, Talk)

Participant(Email, Participant, Address) *Rest(Email, S,SA,SE,R,H,T)* $E \rightarrow PA$

Speaker(SEmail, Speaker, SAddress) *Rest(SEmail, E,R,H,T)* $SE \rightarrow SSA$

Event(Talk, SEmail, Room, Hour) Attends(Talk, Email) $T \rightarrow S\ SE\ SA\ R\ H$

Since every speaker must be a participant, and the Speaker and Participant relations have the same schema, the speaker relation can be dropped, and Speaker, SEmail, and SAddress replaced by Participant, Email, and Address.

Three points for BCNF, one for lossless join, one for speaker must be a participant.

2.2 4NF

Some of the talks may be day-long or multi-day tutorials. Thus their may be several rooms and hours for a single talk, however each has the same speaker and participants. The functional dependencies above prevent this.

2.2.1 Dependencies (3 minutes, 3 points)

Show what functional dependencies need to be removed, and what functional or multivalued dependencies need to be added, to support such multi-room/hour talks.

Drop T → RH (1 point)

Add T →→ S, T →→ P (1 point each)

2.2.2 Schema (4 minutes, 3 points)

Give a good schema for your new collection of dependencies.

Divide Event into TalkBy(Talk, SEmail) and Talk(Talk, Room, Hour).

Two points for 4NF, 1 point for lossless join.

2.3 Lossless join (4 minutes, 3 points)

Give a query in relational algebra that reconstructs the original relation *UniversalRelation* from your decomposition of either question 2.1 or 2.2.2 (note which you use).

Based on 2.2.2:

$\rho_{P \to S, E \to SE, A \to SA} joinTalkBy joinTalk joinAttends joinParticipant$

2 points for getting the join, one for renaming.

3 Queries (15 minutes, 10 points)

You will need to answer queries on the following schema, used to capture event information, registrations, and payments at the conference (for this conference, each event is registered for and charged separately.)

```
create table Receipts (
  Participant varchar(40),
  Amount      numeric
);
```

```
create table Registrations (
  Participant varchar(40),
  Event       char(5) references EventInfo(Event),
  primary key (Participant, Event)
);
```

```
create table EventInfo (
  Event    char(5) primary key,
  Speaker  varchar(40),
  Room     char(5),
  Time     timestamp,
  Price    numeric
);
```

Answer each of the following queries. If you do one in SQL, and one in relational algebra, you get an extra point.

3.1 Simple query (5 minutes, 3 points)

Produce a schedule for participant *Chris Clifton* showing what event, where, and when. For full credit, the events should be listed in order of the time that they occur.

```
select E.event, E.room, E.time
from Registrations R, EventInfo E
where R.participant='Chris Clifton' and R.event = E.event
order by E.time;
```

(1 point for select, 1 for join, 1 for project)
Note: You can't get full credit if you did this one in relational algebra, as you couldn't order it by time.

3.2 Reports (5 minutes, 3 points)

Produce a report showing the total earned by each event (i.e., number of participants * price).

$\gamma_{Event,sum(Price)} RegistrationsjoinEventInfo$
(1 points for group by, 1 for join, 1 for correct)

3.3 Understanding a Query (3 minutes, 3 points)

Describe briefly what the following queries means (i.e., what the values in the result correspond to in real-world terms).

```
select O.Participant, O.Amount-R.Amount Total
from (select R.Participant, sum(E.Price) Amount
      from   Registrations R, Eventinfo E
      where  R.Event = E.Event
      group by R.Participant) O,
     (select Participant, sum(Amount) Amount
      from   Receipts
      group by Participant ) R
where R.Participant = O.Participant;
```

For each participant (1 point) *who has paid something* (1 point), compute the amount still owed (1 point).