Anne Surkey (Provided by Paea LePendu)
Database Processing CIS 451/551
October 31, 2006
Assignment 3

1.

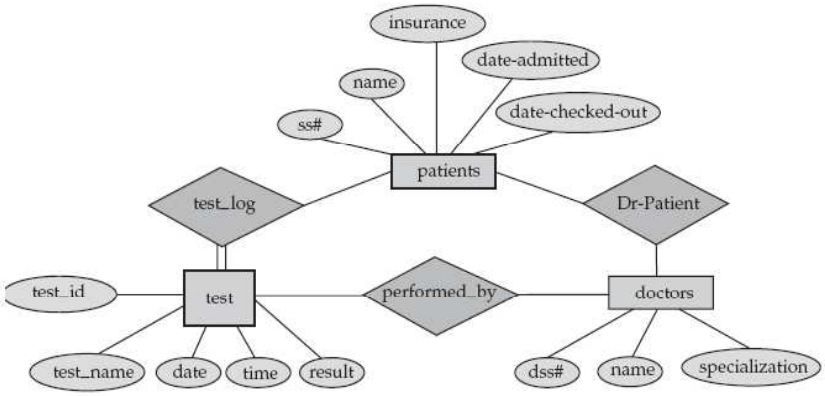

Figure 1: Hospital E-R diagram.

2.

```
patients (patient-id, name, insurance, date-admitted, date-checked-out)
doctors (doctor-id, name, specialization)
test (testid, testname, date, time, result)
doctor-patient (patient-id, doctor-id)
test-log (testid,patient-id)
performed-by (testid, doctor-id)
```

3. loan (loan_number, amount)
   payment (loan_number, payment_number, payment-date, payment-ammount)

The main points we were looking for in regards to the purpose of weak-entity sets are: (1) that the discriminator of the weak entity set forms a "sort-of" key by separating each entity distinctly within some kind of context, (2) that the identifying entity's key finally helps to make each weak entity distinct – it provides exactly the context needed to form the key.

Furthermore, an astute student made a very good point in class: "what is the purpose of a weak entity set since you have to include the identifying entity's key which gives you a primary key anyway – so you end up with a regular entity anyway?" The answer is: that is an absolutely correct IMPLEMENTATION of a weak entity set! However, when we MODEL a weak entity set, the subtle knowledge about the properties of a weak entity set are clear. We lose that explicit knowledge in the implementation. In other words, an ER-Model is more expressive than the schema we use to implement it; and a weak entity set is something a schema can implement but not clearly describe like an ER-Model can.

4. a) The superkey we expected was: {A,B,C,D}, since C1 and C2 are both subsets of that.
   b) Either C1 or C2 can be used as a primary key (the choice is subjective). The important thing to realize in this question is that after you pick a primary key, don't forget the important knowledge about the other candidate keys; in a database you can usually specify an alternate key – which is a good

thing to do .

The main point in the second part of this question is to show data that conforms to each candidate key separately, but not both keys taken together. There are many variations that work. Students gave a good variety of answers – I hope you can see from each variation a slightly different problem scenario involved in data integration:

- DB1={ <a,b,c,d>, <a,b,x,d>}, DB2={} :: DB1+2={<a,b,c,d>, <a,b,x,d>}
  - But abd should be a key. This shows there is inconsistency between c and x in DB1 that DB2 has helped to detect; a LACK OF INTEGRITY ENFORCEMENT in DB1.

- DB1= {<a,b,c,d>},DB2={<a,b,c,x>} :: DB1+2={<a,b,c,d>, <a,b,c,x>}
  - But abc is a key. This shows an INCONSISTENCY introduced by DB2 (or DB1); a content DISAGREEMENT. *Which one is correct for entity abc (d or x)?*

- DB1= {<a,b,c,d>},DB2={<a,b,c,d>} :: DB1+2={<a,b,c,d>, <a,b,c,d>}
  - But abc and abd are keys. This shows REDUNDANCY between DB1 and DB2 in which case it can obviously be solved by removing the DUPLICATE INFORMATION.