UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

M.Sc./DIPLOMA IN ARTIFICIAL INTELLIGENCE
M.Sc./DIPLOMA IN COGNITIVE SCIENCE
M.Sc./DIPLOMA IN COMPUTER SCIENCE
M.Sc./DIPLOMA IN INFORMATICS

DATABASE SYSTEMS

Friday 19 April 2002

09:30 to 11:00

Examiners:
N. Chater (External)
N. Jennings (External)
E. Hancock (External)
K. Hawick (External)
R.N. Ibbett (Convener)
J. Oberlander (Co-Convener)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

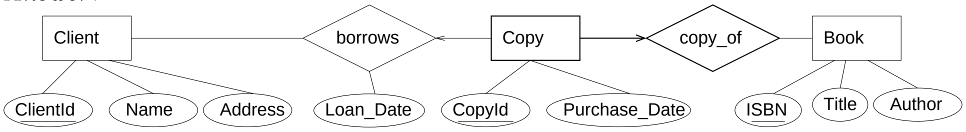
All questions carry equal weight.

1. (a) Draw an E-R diagram for a library. Libraries contain copies of books and have clients (people who borrow books.)

- A *Book* has attributes *ISBN* (key), *Author* and *Title*.
- A book *Copy* is identified by an additional *CopyId* and has a *Purchase_Date*.
- A *Client* has attributes *ClientId* (key),*Name*, and *Address*.
- Clients *borrow* book copies. The *Loan_Date* is recorded.
- A given book copy can only be borrowed by one client; that is, we only record the client (if any) who is *currently* borrowing the book.

Draw the E-R diagram for the library. [*6 marks*]

Answer:



Note that `copy` is a weak enitity.

(b) Write down the SQL DDL for the library database. (Choose any reasonable types for attributes.) [*6 marks*]

Answer:

```
create table book (
    ISBN string
    Author string
    Title string
    primary key (ISBN) )

create table copy (
    ISBN string
    CopyId integer
    Purchase-Date date
    ClientId string
    Loan-Date date
    primary key (ISBN, CopyId)
    foreign key (ISBN) references Book (ISBN)
    foreign key (ClientId) references Client (ClientId) )

create table client (
    ClientId string
    Name string
    Address string
    primary key ClientId )
```

(c) Suppose that, instead of recording only the current borrower of a book, we now record the complete history of how a book is borrowed. How would you change the SQL DDL? [*6 marks*]

Answer: The `copy` table is broken into a `copy` and `borrows` relation:

```
create table copy (
    ISBN string
    CopyId integer
    Purchase-Date date
    primary key (ISBN, CopyId)
    foreign key ISBN references Book )
```

```
create table borrows (
    ISBN string
    CopyId integer
    ClientId string
    Loan-Date date
    primary key (ISBN, CopyId, ClientId, Loan-Date)
    foreign key (ISBN, CopyId) references Copy (ISBN, CopyId)
    foreign key (ClientId) references Client (ClientId) )
```

(d) Using the your answer to 1c, write an SQL query that lists for each book in the library for which a copy has been borrowed, the `ISBN` of that book and the *ClientId* of the first person(s) to borrow a copy of that book. [*7 marks*]

Answer: Create a table of `T` of `ISBN, Id, Date` by doing the appropriate join

```
select T1.ISBN T1.Id
from T T1
where date <= ALL select date
              from T T2
              where T1.Id = T2.Id
```

2. Given the functional dependency set:

$$\mathcal{F} = \{A \rightarrow BE, B \rightarrow C, B \rightarrow E, DE \rightarrow C, C \rightarrow A\}$$

(a) Give a minimal cover of $\mathcal{F}$. [*4 marks*]

Answer: $\{A \rightarrow B, B \rightarrow C, B \rightarrow E, DE \rightarrow C, C \rightarrow A\}$

(b) Consider the decomposition $\{AB, BCDE\}$ with respect to $\mathcal{F}$. Justify your answers to the following three questions.

i. Is it lossless? [*4 marks*]

Answer: Yes. From $\mathcal{F}$ we can infer $B \rightarrow A$ ($B \rightarrow C \rightarrow A$). Therefore $AB \cap BCDE \rightarrow AB$, which is the lossless join condition.

ii. Is it dependency preserving? [*4 marks*]

Answer: No. The only FD that is "broken" by this decomposition is $C \rightarrow A$, but this cannot be recovered from the other FD's.

iii. Is it in Boyce-Codd normal form? [*4 marks*]

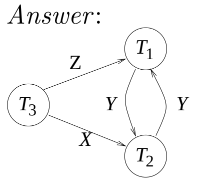
Answer: No. DE is a candidate key for $BCDE$, and $B \rightarrow E$ violates the BCNF condition.

(c) Consider the transactions:

$$\begin{array}{ll} T_1: & r_1(X), w_1(Y) \\ T_2: & r_2(Y), w_2(Z) \\ T_3: & r_3(Z), w_3(X) \end{array}$$

i. Draw the conflict graph for the schedule $r_3, r_2, w_3, r_1, w_1, w_2$ [*4 marks*]

Answer:



ii. Write down *three* distinct schedules that are serial equivalent to $T_1; T_3; T_2$. [*5 marks*]

Answer:

$$r_1, w_1, r_3, w_3, r_2, w_2$$
$$r_1, r_3, w_1, w_3, r_2, w_2$$
$$r_1, w_1, r_3, r_2, w_3, w_2$$

3. (a) Consider the following tables.

R =

A	B
a1	b1
a2	b2
a3	b3

S =

B	C	D
b1	c1	d1
b4	c4	d4

Write down

i. a suitable translation into XML, and [*4 marks*]

ii. a DTD for your translation. [*4 marks*]

Note that this question has several correct answers. You may introduce additional tags (tags that are not names in the relational schema) if needed.

(b) In each of the following DTDs there is something strange, fishy, weird, or downright wrong. For each, say *in at most two sentences* what the problem is.

i.
```
<!ELEMENT A (B,B)>
<!ELEMENT B (#PCDATA)>
<!ELEMENT C (#PCDATA)>
```
[*3 marks*]

Answer: The DTD has two roots: `A` and `C`

ii.
```
<!ELEMENT A (B,C)>
<!ELEMENT B (C,C)>
<!ELEMENT B (#PCDATA)>
<!ELEMENT C (#PCDATA)>
```
[*3 marks*]

Answer: `B` is defined twice

iii.
```
<!ELEMENT A (B*,B?)>
<!ELEMENT B (#PCDATA)>
```
Answer: `(B*,B?)` is equivalent to `(B*)`!

iv.
```
<!ELEMENT A (B)>
<!ELEMENT B (C)>
<!ELEMENT C (D,B)>
<!ELEMENT D (#PCDATA)>
```
[*3 marks*]

Answer: In this recursive DTD every `B` element must contain a `C` element and *vice versa*, which means that there is no (finite) document satisfying this DTD.

(c) An XML file `"xml.xml"` has the following DTD.

```
<!ELEMENT root (A|B)*>
<!ELEMENT A (C,D)>
<!ELEMENT B (C,D*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (#PCDATA)>
```

For each of the following XQuery queries, write down the most restrictive DTD to which the output will always conform.

i.
```
<ANS>{
  for $x in document("xml.xml")/A
    for $y in $x/C
      for $z in $x/D
      return <A> <C> {$y/text()} </C> <D> {$z/text()} </D> </A>
}</ANS>
```
[*4 marks*]

Answer:

```
<!ELEMENT ANS (A*)>
<!ELEMENT A (C,D)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (#PCDATA)>
```

ii.
```
<ANS>{
   for $x in document("xml.xml")/B
      return <B>{
       for $y in $x/C
         for $z in $x/D
         return <C> {$y/text()} </C>
       }</B>
}</ANS>
```
[*4 marks*]

Answer:

```
<!ELEMENT ANS (B*)>
<!ELEMENT B ((C)*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (#PCDATA)>
```