The University Of New South Wales

Final Exam

June 2005

COMP9311

Database Systems

Time allowed: **2 hours**
Total number of questions: **7**
Total number of marks: **90**
Textbooks, study notes, calculators, mobile phones, etc.
are **not** permitted in this exam.
Questions are **not** worth equal marks.
Answer **all** questions.
You can answer the questions in any order.
Start each answer on a new page in a script book.
If you use more than one script book,
fill in your details on the front of *each* book.
You may **not** take this question paper out of the exam.

Name:

Student#:

Question 1

(20 marks) Consider the following set of requirements to model some aspects of the book publishing industry in Australia:

- there are two kinds of people to consider: authors and editors
- for each person, we need to record their tax file number (TFN), their real name, and their address
- everyone who earns money in Australia has a distinct tax file number
- *authors* write books, and may publish books using a "pen-name" (a name which appears as the author of the book and is different to their real name)
- *editors* ensure that books are written in a manner that is suitable for publication
- every editor works for just one publisher
- editors and authors have quite different skills; someone who is an editor cannot be an author, and vice versa
- a book may have several authors, just one author, or no authors (published anonymously)
- every book has one editor assigned to it, who liaises with the author(s) in getting the book ready for publication
- each book has a title, and an edition number (e.g. 1st, 2nd, 3rd)
- each published book is assigned a unique 13-digit number (its ISBN); different editions of the same book will have different ISBNs
- *publishers* are companies that publish (market/distribute) books
- each publisher is required to have a unique Australian business number (ABN)
- a publisher also has a name and address that need to be recorded
- a particular edition of a book is published by exactly one publisher

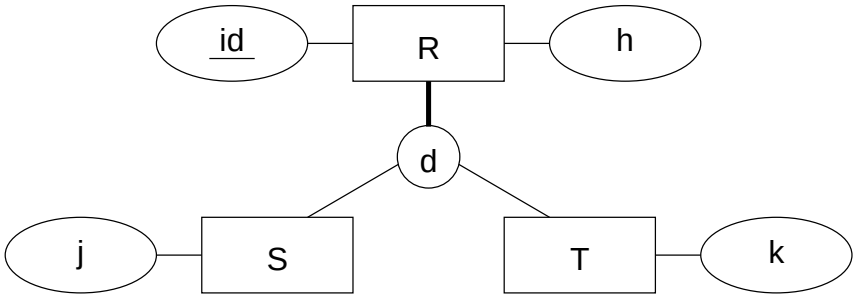
Using the information above, draw an ER diagram to model this scenario.

Note that you must:

a) underline all primary key attributes
b) clearly indicate relationship cardinalities
c) clearly indicate participation constraints
d) choose sensible names for attributes, entities and relationships
e) use a single, well-recognised ER notation for the entire diagram

Question 2

(10 marks) Convert the ER class hierarchy below into two different relational schemas, where each schema is expressed as one or more Oracle CREATE TABLE statements.



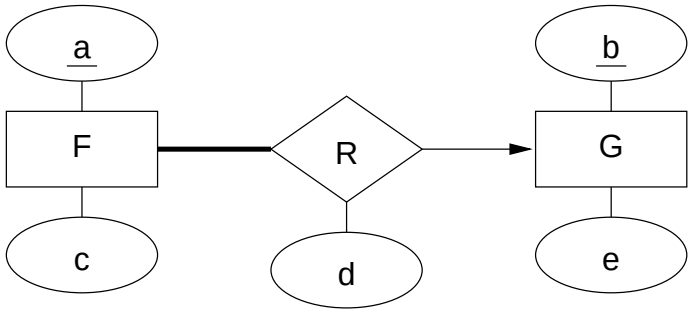
Assume the following data types for each of the attributes: id is an integer, h is a string of up to twenty characters, j is a string of exactly four upper-case letters (e.g. `'ABCD'`), and k is a floating point number which must be in the range 1.0 to 5.0 inclusive. Your schema must show all primary key constraints, foreign key constraints and "not null" constraints suggested by the diagram. You should also include any domain constraints necessary to ensure that attributes contain only valid values. You may add any new attributes that you need to represent all aspects of the ER diagram. Also, document any semantic aspects suggested by the diagram which *cannot* be represented in the relational schema.

a) Convert the diagram using the ER mapping (one table per entity).

b) Convert the diagram using the single-table mapping (one table for the entire hierarchy).

Question 3

(10 marks) Convert the ER design below into a relational schema, expressed as a collection of Oracle CREATE TABLE statements.



Assume that the primary keys are integers, and all other attributes are long strings (use type `varchar(1000)`). Your schema must show all primary key constraints, foreign key constraints and "not null" constraints suggested by the diagram. You may add any new attributes that you need to represent all aspects of the ER diagram. Also, document any semantic aspects suggested by the diagram which *cannot* be represented in the relational schema.

Background for Questions 4-6. Airlines nowadays typically have an online booking service which allows passengers to book and pay for flights via the Web. The following schema describes some of the data that would be required to implement such an online booking system.

```
create table Airports (
   id      integer primary key,
   code    char(3),      -- e.g. 'SYD', 'MEL', ...
   name    varchar(100), -- e.g. 'Kingsford Smith', 'Tullamarine', ...
   city    varchar(100)  -- e.g. 'Sydney', 'Melbourne', ...
);

create table Planes (
   id      integer primary key,
   craft   varchar(100), -- e.g. 'Boeing 767-300'
   nseats  integer check (nseats > 0)
);

create table Flights (
   id      integer primary key,
   fltNum  char(6),      -- e.g. 'QF0512', 'SG0012', ...
   plane   integer not null references Planes(id),
   source  integer not null references Airports(id),
   dest    integer not null references Airports(id),
   departs date,         -- date and time of departure
   arrives date,         -- date and time of arrival
   price   real          -- base price for seats on this flight
);

create table Passengers (
   id      integer primary key,
   name    varchar(100), -- e.g. 'John Smith', 'Peter Wang', ...
   address varchar(100), -- e.g. '16/64 Barker St., Kingsford, ...'
   phone   varchar(100)  -- e.g. '0410222333'
);

create table Bookings (
   pax     integer references Passengers(id),
   flight  integer references Flights(id),
   paid    real,        -- how much passenger paid for ticket
   primary key(pax,flight)
);
```

(continued on next page)

Most of the details in the above schema are self-explanatory. The following may need some additional comments:

- each passenger has a separate booking; even if a booking is made for a group of people (e.g. a family), each person in the group will have a separate entry in the `Bookings` table
- `Flights.price` is the "standard" fare for the flight; the actual price paid by passengers (`Bookings.paid`) may be different to this, depending on the availability of special deals, or when the booking is made
- `Planes.nseats` is the number of passenger seats available on a particular aircraft; this is also the maximum number of bookings that can be made for a flight using this plane

Question 4

(10 marks) One important operation in an online booking system is to check whether there are still seats available on a particular flight. Write a PL/SQL function `seatsAvail()` that takes a value for `Flights.id` and returns a single integer indicating how many seats are still available for booking on that flight. Use the following function heading:

```
create function seatsAvail(flid integer) return integer is ...
```

If the argument is not a valid flight `id`, then the function should return a value of `null`.

Question 5

(20 marks) After the online booking system has been operational for a while, the database administrators discover that the most frequent and expensive operation in the system is checking how many seats are available on a flight. In order to improve system performance, it is decided to replace the `seatsAvail()` function by maintaining an extra field in the `Flights` table to maintain a count of available seats. As a first step, a new field is added to the `Flights` table:

```
alter table Flights add avSeats integer check (avSeats > 0);
```

Write Oracle CREATE TRIGGER statements (plus any auxiliary CREATE PROCEDURE statements) to set up triggers to ensure that the value of `Flights.avSeats` always contains the number of available seats.

For the purposes of this exercise, assume that SQL UPDATE operations are never performed on the `Bookings` table (i.e. the only way to change a flight is to delete an existing booking and then make a new one). Thus, only the following changes need to be considered:

- adding a new flight into the database ("before" trigger)
- adding a new booking for a particular flight ("after" trigger)
- deleting a booking for a particular flight ("after" trigger)

For the "after" triggers, assume that the regular DBMS constraint checking will handle cases with invalid foreign keys, so you do not need to worry about checking for them. For the "before" trigger, you will explicitly need to check foreign keys. If you need to terminate an operation, raise an exception with a suitable message to indicate what the problem is.

Question 6

(10 marks) Consider an electrical retailer who sells both to the public and to other electrical stores. Their record keeping is based on an old spreadsheet and they now wish to convert to more reliable database technology. They store each sale as a row in the spreadsheet. The row contains the time of the sale, the item being sold, including its unique product code and its price, the number (Qty) of units sold, the customer's name/address/phone, and their unique customer code. A fragment of the spreadsheet is given below:

TimeOfSale	**Product**	**Pcode**	**Price**	**Qty**	**Cust#**	**Customer**	**Address**	**Phone**
...	...	...	...	...	...	...	...	...
15/6/2005 10:30	Toaster	555	25.00	1	1234	John Smith	Smith St.	99234567
15/6/2005 10:30	Kettle	557	15.00	1	1234	John Smith	Smith St.	99234567
15/6/2005 14:00	Toaster	555	25.00	20	3251	Betta Electrical	Big Pde.	99327890
15/6/2005 15:30	Kettle	557	15.00	1	3423	Adam Harper	Beach St.	99229876
15/6/2005 15:40	Toaster	555	25.00	1	3422	Jill Brown	Sandy St.	99346565
15/6/2005 15:45	Toaster	555	25.00	3	3251	Cafe Coogee	Rainbow St.	99228765
15/6/2005 15:45	Steamer	553	75.00	1	3251	Cafe Coogee	Rainbow St.	99228765
...	...	...	...	...	...	...	...	...

Treat this spreadsheet as an unnormalised relational table

$$R(T, Pr, Pc, Pe, Q, C\#, Cu, Ad, Ph)$$

where the attributes are given in the same order as the spreadsheet columns.

To accomplish the conversion to a relational database:

a) identify all of the functional dependencies in this table

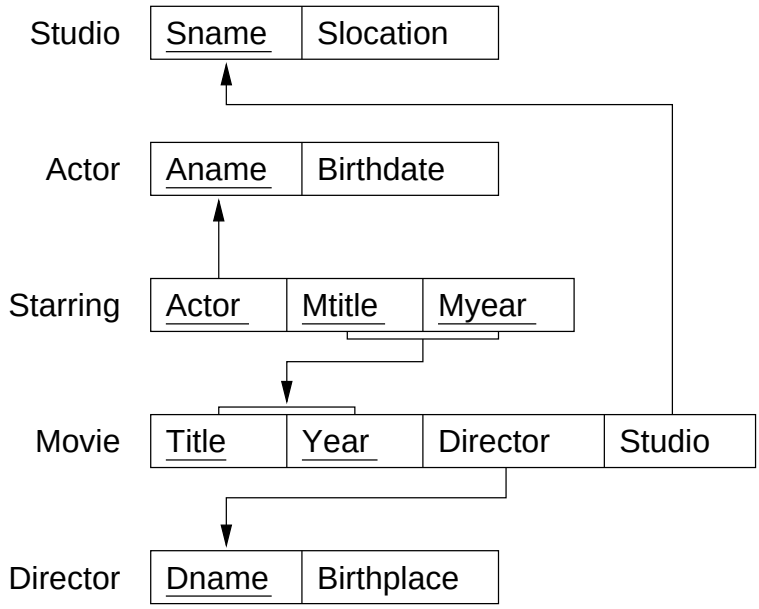
b) use these functional dependencies to transform the table into a BCNF schema

In giving the transformation, show your working for each step, i.e.

- state why the schema is not in BCNF
- describe a transformation that makes it closer to BCNF

Question 7

(10 marks) Consider the following relational schema describing aspects of the film industry:



Give efficient relational algebra expressions to the answer the queries below. Correct, but inefficient, queries score only half marks.

a) Which studios has Peter Weir directed films for?

b) Which actors have starred in films from Paramount Studios?

c) Which films starred both Tom Cruise and Nicole Kidman?

d) Which actors have starred in all films directed by Stanley Kubrick?

You must include all `Rename` operations that are strictly required by the relational algebra operations. For writing relational algebra, use the following notation: `Sel` for selection, `Proj` for projection, `Join` for natural join, `Div` for division, `Cross` for cross product, `ThetaJoin`, `LeftOuterJoin`, `Union`, `Intersect`, `Minus`, `GroupBy`, `Count`. Make use of named intermediate results to make the expression clearer.