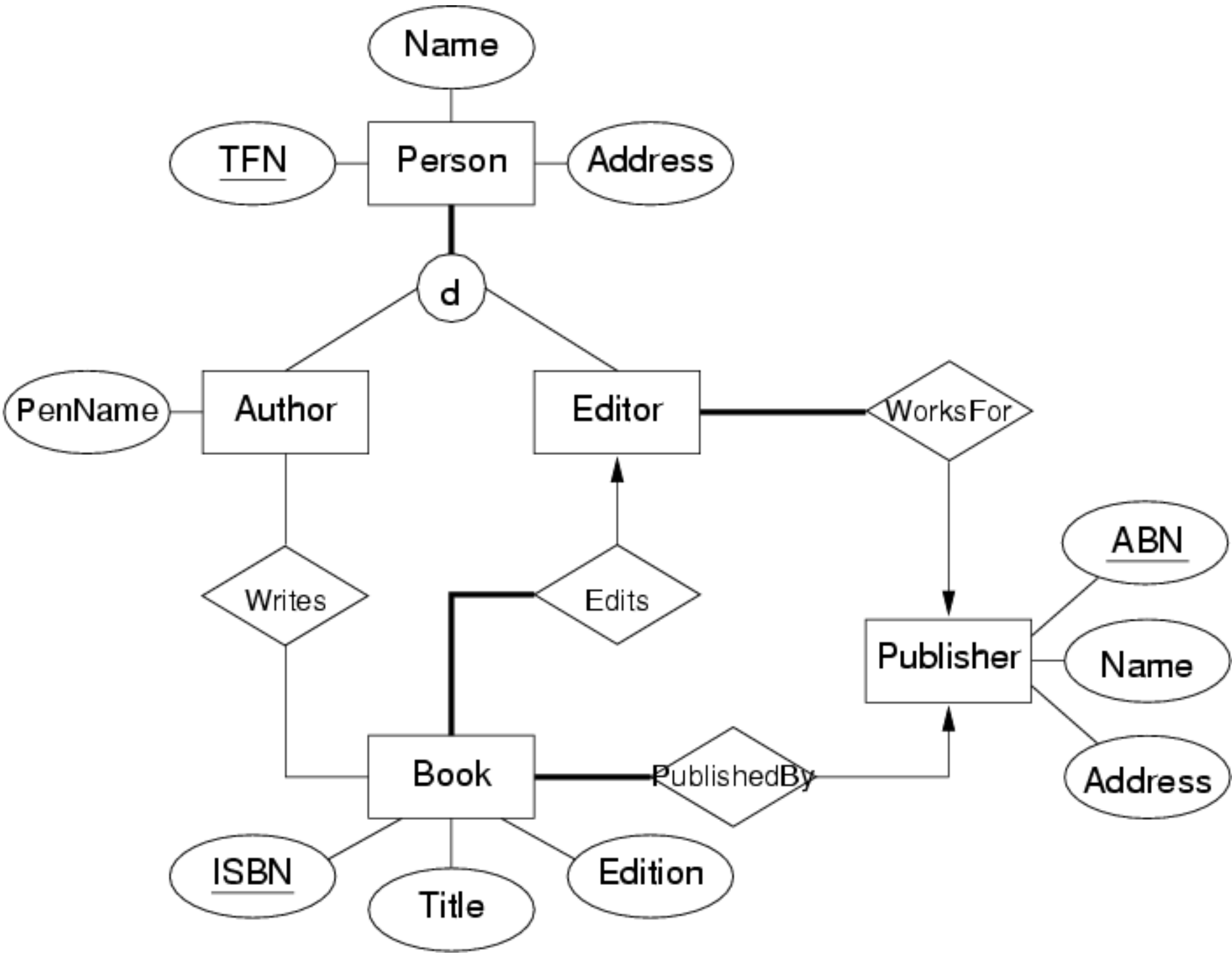
COMP9311 05s1 Theory Exam
Sample Solutions

Question 1

An ER diagram to represent information about the Australian book-publishing industry:



Some allowable variations:

- the relationships don't have to have precisely these names, as long as the meaning is clearly the same
- the lines marked in blue could have been made "thick" to indicate total participation
- there could have been an additional relationship between Authors and Editors called "liasesWith" (or some term like this)

Question 2

Relational schemas based on ER diagram with class hierarchy:

a. Version using the "ER mapping" (one table per entity)

```
create table R (
        id      integer,
        h       varchar(20),
```

```
        primary key (id)
);
create table S (
        id      integer,
        j       char(4) check (j >= 'AAAA' and j <= 'ZZZZ')
        primary key (id),
        foreign key (id) references R(id)
);
create table T (
        id      integer,
        k       float check (k >= 1.0 and k <= 5.0),
        primary key (id),
        foreign key (id) references R(id)
);
```

This schema cannot represent:

- that an R must appear in one of S or T
- that an R appears only in S or T and not both

b. Version using the "single table mapping" (one table for hierarchy)

```
create table R (
        id      integer,
        h       varchar(20),
        kind    char(1) not null check (kind in ('S','T')),
        j       char(4) check (j >= 'AAAA' and j <= 'ZZZZ')
        k       float check (k >= 1.0 and k <= 5.0),
        primary key (id)
);
```

This schema cannot represent:

- that attribute `j` should have value if `kind='T'`
- that attribute `k` should have value if `kind='S'`

Allowable variations in the schemas:

- `integer` could be replaced by `int`
- `float` could be replaced by `real`
- any name can be used for the `kind` attribute; it's type must have two values that clearly distinguish `S` and `T` tuples

Question 3

Relational schemas based on ER diagram with class hierarchy:

```
create table F (
    a       integer,
    c       text,
    R       integer not null,
    d       text,
    primary key (a),
    foreign key (R) references G(b)
);
create table G (
    b       integer,
    e       text,
    primary key (b)
);
```

Question 4

Function to return the number of seats available on a plane flight:

```
create or replace function
    seatsAvail(flid integer) return integer
as
    totSeats    integer;
    bookedSeats integer;
begin
    select p.nseats into totSeats
    from   Flights f, Planes p
    where  f.id = flid and f.plane = p.id;
    select count(b.pax) into bookedSeats
    from   Bookings b
    where  b.flight = flid;
    return totSeats - bookedSeats;
end;
/
```

Question 5

Triggers for maintaining the `Flights.avSeats` attribute:

```
create trigger insertFlightTrigger
before insert on Flights
for each row
declare
    ns integer;
begin
    -- if no such plane, then exception terminates insert
    select nseats into ns from Planes where id = :new.plane;
    :new.seatsAvail := ns;
end;
```

```
create trigger insertBookingTrigger
after insert on Bookings
for each row
begin
    update Flights set seatsAvail = seatsAvail - 1 where id = :new.flight;
end;

create trigger deleteBookingTrigger
after delete on Bookings
for each row
begin
    update Flights set seatsAvail = seatsAvail + 1 where id = old.flight;
end;
```

Assumes that the interface (and domain constraints) prevents INSERT operations being invoked on a full flight.

Question 6

a. Functional dependencies:

```
Pcode -> Product
Pcode -> Price
Cust# -> Customer
Cust# -> Address
Cust# -> Phone
```

or, to simplify and use question notation

```
(a) Pc -> Pr Pe
```

```
(b) C# -> Cu Ad Ph
```

b. Conversion to BCNF schema: Start from:

```
R = (T, Pr, Pc, Pe, Q, C#, Cu, Ad, Ph)  with key = (Pc,C#,T,Q)
```

Existence of FD (a) means `R` is not in BCNF (partial-key dependence), so decompose to

```
R' = (T, Pc, Q, C#, Cu, Ad, Ph)  with key = (Pc,C#,T,Q)

P  = (Pc, Pr, Pe)  with key = (Pc)
```

Table `P` is already in BCNF (all non-key attributes depend only on whole key)
Existence of FD (b) means `R'` is not in BCNF (partial-key dependence), so decompose to

```
R'' = (T, Pc, Q, C#)  with key = (Pc,C#,T,Q)

C   = (C#, Cu, Ad, Ph)  with key = (C#)
```

Both of the above tables are in BCNF (no FDs violate BCNF rules), so the final schema is:

```
Prod = (Pc, Pr, Pe)
Cust = (C#, Cu, Ad, Ph)
Sale = (T, Pc, Q, C#)
```

Allowable variations:

- No need to use exactly the same naming scheme
- Can apply the FDs in either order
- If other FDs were defined in part (a), they must be used correctly here

Question 7

a. Which studios has Peter Weir directed films for?

```
Res = Proj[studio](Sel[director='Peter Weir'](Movie)
```

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

```
PFilms  = Proj[title,year](Sel[studio='Paramount'](Movie))
PFilms' = Rename[mtitle,myear](PFilms)
Res     = Proj[actor](Starring Join Pfilms')
```

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

```
TFilms = Proj[mtitle,myear](Sel[actor='Tom Cruise'](Starring))
NFilms = Proj[mtitle,myear](Sel[actor='Nicole Kidman'](Starring))
Res    = TFilms Intersect NFilms
```

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

```
KFilms  = Proj[title,year](Sel[director='Stanley Kubrick'](Movie))
KFilms' = Rename[mtitle,myear](KFilms)
Res     = Starring Divides KFilms'
```