COMPS320F *Specimen Exam Solution*

Q1

(a)

i	T
ii	T
iii	F
iv	F
v	T

(b)

i. data definition and data manipulation

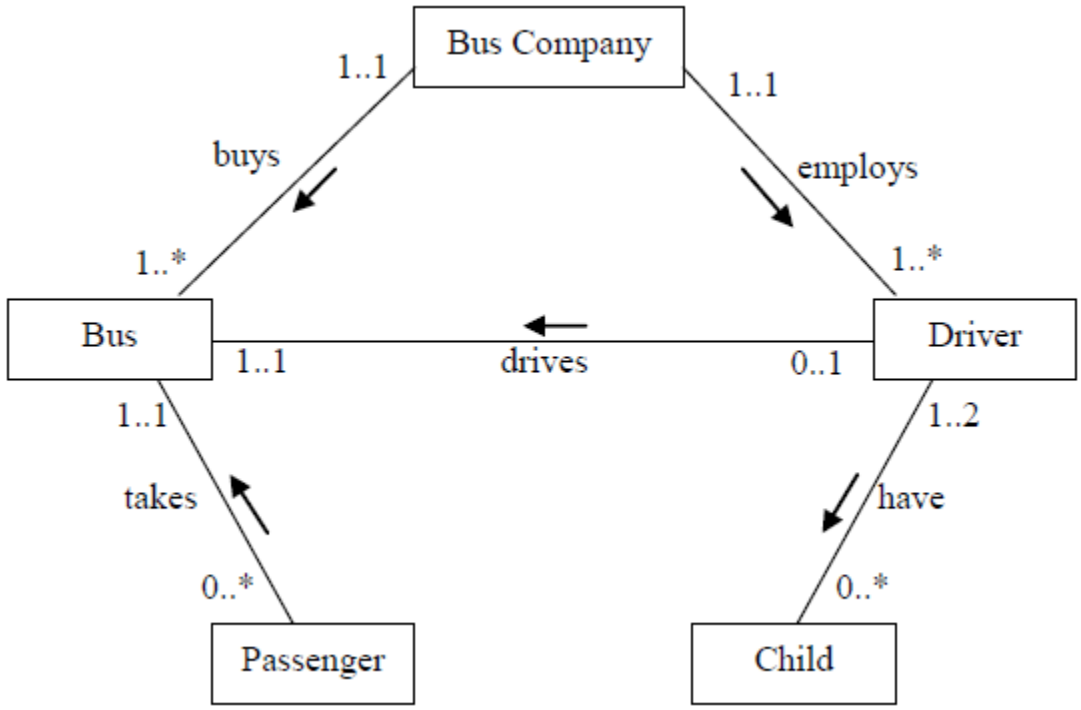
ii. COUNT

iii. WHERE

iv. NOT IN

[Note]: Answers are case-insensitive

Q2



Q3

(a) A candidate key is determining and minimal. It has a potential of being selected as the primary key. A primary key is a candidate key which is selected to be the primary key.

(b) The relationship of an entity with itself is a recursive relationship.
An example is a staff (manager) manages other staff.

(c) Any four below:

- Separation and isolation of data
- Duplication of data
- Data dependence
- Incompatible file formats
- Fixed Queries/Many application programs

Q4

(a)

i. Invalid, since non-aggregate columns needs to be put in a group-by clause when an aggregate function is used in the select list

ii. Invalid, since "order by" cannot be put before "where"

iii. Valid,

PRODUCTNAME
Personal computer
Notebook computer

[Note]: Column names are case-insensitive

(b)

i.

Candidate Keys:

- {CustID}

Primary Key:

- {CustID}

ii.

Alternate Key:

- Nil

Foreign Key:

- {CustID} in Order table is the foreign key references Customer table.
- {BranchID} in Order table is the foreign key references Branch table.

Q5

a.

```
CREATE TABLE room(
   roomNo char(3) NOT NULL,
   class char(1)DEFAULT 'D',
   price numeric(7,2) check (price between 500 and 10000),
   PRIMARY KEY (roomNo)
);
```

b.

```
ALTER TABLE room
   add constraint classChk check(class IN ('S', 'F', 'D'));
```

c.

```
ALTER TABLE guest MODIFY contactNo char(8);
```

d.

```
SELECT SUM(price*stayDays) as income
FROM Booking b, Room r, Hotel h
WHERE b.roomNo = r.roomNo
AND b.hotelNo = h.hotelNo
AND hotelName = 'First World';
```

e.

```
SELECT roomNo, class,
        price - (SELECT AVG(price) FROM room) As priceDiff
FROM room
WHERE price >
   (SELECT AVG(price)
    FROM room);
```

f.

```
CREATE VIEW hotelBookingCount (hotelNo, bookingCount)
AS SELECT hotelNo, COUNT(*)
   FROM Booking
   GROUP BY hotelNo;
```

g.

```
UPDATE Room SET price = price*0.95;
```

h.

```
DELETE from guest
where guestName = 'Ashley'
and guestAddress like '%Germany%';
```

Q6

(a)

```
DECLARE
   g_no booking.guestNo%type := 'G02';
   g_name guest.guestName%type;
   totDay int := 0;
BEGIN
   SELECT guestName, sum(stayDays)
   into g_name, totDay
   FROM guest g, booking b
   WHERE g.guestNo = b.guestNo
   AND b.guestNo = 'G02'
   GROUP by guestName;

   dbms_output.put_line('Guest No.: ' || g_no);
   dbms_output.put_line('Guest Name: ' || g_name );
   dbms_output.put_line('Days stayed: ' || totDay );
END;
/
```

(b)

```
CREATE OR REPLACE PROCEDURE p_listGuestStay IS
   CURSOR cur_guestStay is
        SELECT g.guestNo, guestName, sum(stayDays)
        FROM guest g, booking b
        WHERE g.guestNo = b.guestNo
        GROUP by g.guestNo, guestName
        ORDER by sum(stayDays) DESC;
   g_no guest.guestNo%type;
   g_name guest.guestName%type;
   totDay int := 0;
   VIP varchar(10) := '';
BEGIN
    OPEN cur_guestStay;
    LOOP
        FETCH cur_guestStay into g_no, g_name, totDay;
        EXIT WHEN cur_guestStay%notfound;

        IF totDay > 7 THEN
            VIP := ' [VIP]';
        END IF;

        DBMS_OUTPUT.put_line(g_no|| ' (' || g_name || ') Stayed ' ||
totDay || ' Days' || VIP);
        VIP := '';
    END LOOP;
    CLOSE cur_guestStay;
END;
/

EXEC p_listGuestStay;
```

****** End ******