ASSIGNMENT 2 SOLUTIONS AND MARKING GUIDE

Part 1. Relational Database Design (30 Marks)

1. List all of the non-trivial FDs in the `InsurancePolicy` relation, including those implied by a key. Is the `InsurancePolicy` relation in BCNF? Explain briefly your answer.

`InsurancePolicy (PolicyNumber, AgentNumber, PremiumAmount, CoverageAmount)`

FDs

`PolicyNumber -> AgentNumber`

`PolicyNumber -> PremiumAmount`

`PolicyNumber -> CoverageAmount`

`Yes, is in BCNF. All FDs pass BCNF test because the LHS of all FDs is a key.`

It is possible for several, different insurance policies to have the same agent assigned (that is, to have the same agent number listed in the `AgentNumber` attribute). Does this kind of redundancy cause any update anomalies?

Even though AgentNumber will appear many times in the `InsurancePolicy table,` there is no concern about update anomalies.

These anomalies only occur when a given value for Policy Number maps to the same Agent Number more than once. In these circumstances, it is possible to miss updating Agent Number for every occurrence of Policy Number, leading to an anomaly.

Since Policy Number is the primary key, each value only occurs once, thereby removing opportunity for anomalies.

Can we normalise this relation in order to remove this redundancy?

No. You cannot remove this repeated data by decomposition.

Marking

- 4 Marks : FDs and BCNF
- 4 Marks : Discussion on repeated data in AgentNumber. (noting Policy number is not repeated)
- 2 Marks : No decomposition (0 marks for any attempt to decompose.)

2. List all of the non-trivial FDs in the `Person` relation, including those implied by a key. Is the `Person` relation in BCNF? Explain briefly your answer.

```
Person (Id, TFN, Name, Phone)
```

FDs

ID -> TFN

~~ID -> Name~~ redundant (because ID -> TFN & TFN -> Name)

~~ID -> Phone~~ redundant (because ID -> TFN & TFN -> Phone)

TFN -> ID

TFN -> Name

TFN -> Phone

Yes, is in BCNF. All FDs pass BCNF test because the LHS of all FDs is a key.

Note : it can be argued that TFN -> Name and TFN -> Phone are redundant instead. This does not change the outcome.

If the `Person` relation is not in BCNF, decompose the `Person` relation into relations that are in BCNF.

Decomposition is not required.

Marking

- 7 Marks : FDs, BCNF
- 3 Marks : No decomposition (0 marks for any attempt to decompose.)

3. List all of the non-trivial FDs in the `PolicyOwner` relation, including those implied by a key. Which normal form is this relation in?

PolicyOwner (PersonId, PolicyNumber, TFN, Start-Date)

FDs

a) PersonId, PolicyNumber -> Start-Date

b) PersonId -> TFN

c) ~~TFN, PolicyNumber -> Start-Date~~ redundant, see (1)

d) TFN -> PersonId

(1) TFN, PolicyNumber -> Start-Date is redundant because;

From d) TFN, PolicyNumber -> PersonId, PolicyNumber ___(d1)
(by augmentation)

(d1) and (a) gives : TFN, PolicyNumber -> Start-Date
(by transitivity)

Alternatively, it can be argued in the same way that;

PersonId, PolicyNumber -> Start-Date

is redundant.

The final set of FDs is;

FDs

a) PersonId, PolicyNumber -> Start-Date
b) PersonId -> TFN
c) TFN -> PersonId

All attributes can be found with the closure of the candidate key {PersonId, PolicyNumber}. This is the only key for the relation.

Testing the FDs for BCNF and 3NF, we find that FD b) fails both so **decomposition is required**.

If it's not in BCNF, decompose the PolicyOwner relation into BCNF.

Decomposition;

R1(PersonId*, PolicyNumber, Start-Date)

R2(PersonId, TFN)

Marking

- 6 Marks : FDs, Minimal Basis, BCNF testing
- 4 Marks : R1 - 3 Marks

R1 - 1 Mark

Part 2. SQL (30 Marks)

SQL Marking Comments.

- Several questions have more than one possible answer. These answers are sample answers and are a guide only.
- Marking notes highlight key points in the question that answer should handle.
- Award part marks based on how close the query is to being logically and syntactically correct.
- Part marks down to the 0.25 mark can be used.
- In many cases, you may choose to comment on an minor error rather than deduct marks for it.

1. Give the `bookdescID` of books by authors whose surname starts with "S" or by publishers whose name starts with "S".

```
select WB.bookdescID
from written_by WB, publisher P, author Au, published_by PB
where WB.authorID =Au.authorID
and WB.bookdescID = PB.bookdescID
and PB.publisherID = P.publisherID
and ((P.publisherfullname like 'S%')
        or
        (Au.lastname like 'S%'));
```

marking notes

- *5 marks*
- *May use sub-queries for selection process.*

2. With which publisher(s) the author Alfred Aho published his book(s)? Display publishers' full names. You must not use any subquery.

```
select distinct(publisherfullname)
from written_by WB, publisher P, author Au, published_by PB
where WB.authorID =Au.authorID
and WB.bookdescID = PB.bookdescID
and PB.publisherID = P.publisherID
and Au.LastName like 'Aho'
        and Au.FirstName like 'Alfred';
```

marking notes

- *5 marks*
- *-2 mark if a sub-query is used.*

3. Who are the authors that published books with the MC GRAWHILL publisher? Display the firstname and lastname of these authors.

```
select distinct Au.firstname, Au.lastname
```

```
from publisher p, author Au, written_by wb, published_by PB
where WB.authorID =Au.authorID
and WB.bookdescID = PB.bookdescID
and PB.publisherID = P.publisherID
and upper(P.publisherfullname) like 'MC GRAW-HILL';
```

marking notes

- *6 marks*
- *Note that queries with* **like** 'MC GRAWHILL' are also considered correct.

4. Display the first name and lastname of authors who wrote more than 3 books. Along with each name, display the number of books as well.

```
select  firstname, lastname, count(bookdescid)
from written_by wb, author Au
where Au.AUTHORID=wb.AUTHORID
group by Au.AUTHORID, firstname, lastname
having count(bookdescid) >3;
```

marking notes

- *6 marks*

5. Display the title of the book which has the largest number of physical copies. If there are more than one book with the largest number of copies, show them all. Your query should show the number of copies along with the title.

```
select  title, count(b.bookdescid)
from book b, book_copy bc
where b.bookdescid=bc.bookdescid
group by b.bookdescid, title
having  count(b.bookdescid) = (select max(count(bookdescid))
                               from book_copy
                               group by bookdescid);
```

marking notes

- *8 marks*

Part 3. ER Modelling (30 marks)

These are the key points for the marking. They show the maximum marks for different mappings. From these, further deductions apply; -0.5 marks attribute errors and -1 mark for PK errors, etc.

Entities

ACCOUNT(AcctNo, Balance)	2
CUSTOMER(custID, c_lname, c-fname)	2
PRODUCT(pro_code, name)	2
EMPLOYEE(empID, e-fname, e-lname)	2
BRANCH(branchCode, address)	2

Relationships

YouthSaver :

FK : PK of Account into Dependent	5
FK: Pk of Dependent into Account (must be full key)	5
New relation: Either FK as PK	5
Full merge	3

Own :

New relation: both FKs as PK	2
FK : PK of Customer into Account	1
FK: Pk of Account into Customer	1

Promotion:

New relation:	
all three FKs. custID and Pro-code as PK. outcome as attribute	4
missed outcome	-1 (deduction)
Wrong PK	2

Work :

FK : PK of Branch into Employee	4
FK: Pk of Employee into Branch	2
New relation: both FKs. PK is empID	2
New relation: both FKs. PK is branchCode or both	1

Weak Enitity

DEPENDENT(seqNo, custID*, d-fname, d-lname, school)	5
FK: Pk of Dependent into Customer	3
New relation: Either FK as PK	1

Part 4. Research Questions (10 Marks)

Q1 Returns (6 Marks)

The problem is that the transaction as a whole has the returndate attribute rather than individual books. The Borrow and Book_copy relation schemas need to be changed, as follows:

borrow(transactionID, personID, borrow date, due date)

borrow_copy(transitionID*, bookID, returndate)

Moving the returndate to the BORROW_COPY achieves this.

Notes:

- Adding bookID to BORROW table introduces bad normal form.
- Moving other attributes (like Duedate and BorrowDate) from BORROW to BORROW_COPY introduces un-necessarily repeated data.

Marking

6 Marks for fully correct answer.
Max 4 marks for answers similar those in NOTES.
Max 2 marks if no schema change is discussed.

Q2 Extensions (4 Marks)

It is likely that extensions are handled by updating the due date. A limitation of such approach is that all books loaned at the same time have to be extended and the number of extensions are not recorded in the database.

Marking

2 Marks for explanation of how extension might be done.
2 marks for identifying any schema-related limitation identified.