UNSW AUSTRALIA

SCHOOL OF ENGINEERING

ASSIGNMENT COVER SHEET

[INDIVIDUAL SUBMISSION]

For submission to: Xueming Lin

Course code : 9311

Course name : Database system

Assignment : Assignment2

ACADEMIC REQUIREMENTS

Before submitting this assignment, students are strongly advised to:

- Review the assessment requirements contained in the briefing document for the assignment.
- Review the various matters related to assessment in the relevant Course Outline.
- Review the Plagiarism and Academic Integrity website at https://student.unsw.edu.au/plagiarism to ensure they are familiar with the requirements to provide appropriate acknowledgement of source materials.
- Retain a copy of this assessment for their records and in case it is misplaced and has to be re-submitted.

If after reviewing this material there is any doubt about assessment requirements then in the first instance the student should consult with the Course Coordinator and then if necessary with the Director – Undergraduate Teaching and Learning Committee.

While students are generally encouraged to work with other students to enhance learning, all assignments submitted for assessment by a student must be their entire own work and they may be required to explain any or all parts of the assignment to the Course Coordinator or other authorised persons. Collusion is where another person (s) assist in the preparation of an assignment without the consent of knowledge of the Course Coordinator.
Plagiarism and Collusion are considered as Academic Misconduct and will be dealt with according to University Policy.

STUDENT DECLARATION OF ACADEMIC INTEGRITY

I declare that:

- This assessment item is entirely my own original work, except where I have acknowledged use of source material [such as books, journal articles, other published material, the Internet, and the work of other student/s or any other person/s].
- This assessment item has not been submitted for assessment for academic credit in this, or any other course, at UNSW or elsewhere.

I understand that:

- The assessor of this assessment item may, for the purpose of assessing this item, reproduce this assessment item and provide a copy to another member of the University.
- The assessor may communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the assessment item on its database for the purpose of future plagiarism checking).

Student first name: TANG **Surname:** NANYANG

Student number: z5103095 **Date:** 03/05/2018

Unless stated otherwise, assignments must be lodged in the nominated Assignment Box, no later than 5.00pm on the assignment due date.

Assignment 2

Question1:

$R(A,B,C,D,E,G,H,I,J,K)$

$F = \{A \to BC, E \to AD, BD \to E, CE \to DH, H \to G, EI \to J\}$

1):

NO, $C \nrightarrow J \in F^+$.

$C^+ = \{C\}$, $J \notin C^+$, therefore $C \nrightarrow J \in F^+$.

2):

Step1:

$F' = \{A \to B, A \to C, E \to A, E \to D, BD \to E, CE \to D, CE \to H, H \to G, EI \to J\}$

Step2:

$EI \to J$

$I^+ = \{I\}$ $E^+ = \{A,B,C,D,E,G,H\}$; thus $I \to J$ and $E \to J$ are not inferred by F'. Hence, $EI \to J$ could not be replaced by $E \to J$ or $I \to J$.

$BD \to E$

$B^+ = \{B\}$ and $D^+ = \{D\}$; thus $B \to E$ and $D \to E$ are not inferred by F'.

Hence, $BD \to E$ could not be replaced by $B \to E$ or $D \to E$.

$CE \to D$

$C^+ = \{C\}$ and $E^+ = \{A,B,C,D,E,G,H\}$; thus $C \to D$ is not inferred by F' but $E \to D$ is inferred by F'. Hence, $CE \to D$ could be replaced by $E \to D$.

$CE \to H$

$C^+ = \{C\}$ and $E^+ = \{A,B,C,D,E,G,H\}$; thus $C \to H$ is not inferred by F' but $E \to H$ is inferred by F'. Hence, $CE \to H$ could be replaced by $E \to H$.

$F'' = \{A \to B, A \to C, E \to A, E \to D, BD \to E, E \to H, H \to G, EI \to J\}$

Step3:

$E^+|_{F''-\{A\to B\}} = \{A,C,E,H,G\}$; thus $A \to B$ is not inferred by $F'' - \{A \to B\}$. That is $A \to B$ is not redundant.

$E^+|_{F''-\{A\to C\}} = \{E,A,D,B\}$; thus $A \to C$ is not inferred by $F'' - \{A \to C\}$. That is $A \to C$ is not redundant.

$E^+|_{F''-\{E\to A\}} = \{E,D,H,G\}$; thus $E \to A$ is not inferred by $F'' - \{E \to A\}$. That is $E \to A$ is not redundant.

$E^+|_{F''-\{E\to D\}} = \{E,A,B,C,H,G\}$; thus $E \to D$ is not inferred by $F'' - \{E \to D\}$. That is $E \to D$ is not redundant.

$E^+|_{F''-\{E\to H\}} = \{E, A, B, C, D\}$; thus $E \to H$ is not inferred by $F'' - \{E \to H\}$. That is $E \to H$ is not redundant.

$E^+|_{F''-\{H\to G\}} = \{E, A, B, C, D, H\}$; thus H$\to G$ is not inferred by $F'' - \{H \to G\}$. That is $H \to G$ is not redundant.

$BD^+|_{F''-\{BD\to E\}} = \{B, D\}$; thus $BD \to E$ is not inferred by $F'' - \{BD \to E\}$. That is $BD \to E$ is not redundant.

$EI^+|_{F''-\{EI\to J\}} = \{E, I\}$; thus $EI \to J$ is not inferred by $F'' - \{EI \to J\}$. That is $EI \to J$ is not redundant.

Iteratively, we could find that F'' does not change.

Thus,

$F'' = \{A \to B, A \to C, E \to A, E \to D, BD \to E, E \to H, H \to G, EI \to J\}$.

3):

$R_1 = \{ABCDE\}, R_2 = \{EGH\}, R_3 = \{EIJK\}$

	A	B	C	D	E	G	H	I	J	K
R_1	a	a	a	a	a	b	b	b	b	b
R_2	b	b	b	b	a	a	a	b	b	b
R_3	b	b	b	b	a	b	b	a	a	a

Note: row1 and row2 and row2 and row3 of S agree on $\{E\}$, which is the left hand side of $E \to G$. Therefore, change the G value on row1 and row3 to a. After that, row1 and row2 and row2 and row3 of S agree on $\{G\}$ which is the left hand side of $G \to H$, Therefore, change the H value on row1 and row3 to a.

	A	B	C	D	E	G	H	I	J	K
R_1	a	a	a	a	a	a	a	b	b	b
R_2	b	b	b	b	a	a	a	b	b	b
R_3	b	b	b	b	a	a	a	a	a	a

Note: row1 and row3 and row2 and row3 of S agree on $\{H\}$, which is the left hand side of $H \to I$. Therefore, change the I value on row1 and row2 to a. After that, row1 and row2 and row2 and row3 of S agree on $\{I\}$ which is the left hand side of $I \to J$, Therefore, change the J value on row1 and row2 to a. Similarly, we should change the K value on row1 and row2 to a.

	A	B	C	D	E	G	H	I	J	K
R_1	a	a	a	a	a	a	a	a	a	a
R_2	b	b	b	b	a	a	a	a	a	a
R_3	b	b	b	b	a	a	a	a	a	a

Now, row 1 is entirely a's, so the decomposition is lossless.

4): NO, the reason is below.

$\{A,B,C,D,E,G,H,I,J,K\}$ is super-key

$\{A,D,E,G,H,I,J,K\}$ is super-key, cause $A \to BC$

$\{E,B,C,G,I,J,K\}$ is super-key, cause $E \to ADH$

$\{E,B,C,G,I,K\}$ is super-key, cause $E \to ADH$ and $EI \to J$

$\{E,G,I,K\}$ is super-key, cause $E \to ADH$, $A \to BC$ and $EI \to J$

$\{E,I,K\}$ is super-key, cause $E \to ADH$, $A \to BC, H \to G$ and $EI \to J$

5):

$R(A,B,C,D,E,G,H,I,J,K)$

$F = \{A \to BC, E \to AD, BD \to E, CE \to DH, H \to G, EI \to J\}$

Candidate key is {E,I,K}, $BD \to E$ but BD is not super-key so

$$R\begin{cases} R_1 = \{B,D,E\} \\ R_2 = \{A,B,C,D,G,H,I,J,K\} \end{cases}$$

For R_1, because $BD^+ = \{B,D,E\}$ and $E^+ = \{B,D,E\}$, therefore, BD and E could be super-key. So, R_1 is in BCNF

For R_2, Candidate key is {A,D,H,I,J,K}, $H \to G$ but H is not super-key so

$$R_2\begin{cases} R_{21} = \{H,G\} \\ R_{22} = \{A,B,C,D,H,I,J,K\} \end{cases}$$

For R_{21}, H is the super-key therefore, R_{21} is in BCNF

For R_{22}, Candidate key is {A,D,H,I,J,K}, $A \to BC$ but A is not super-key so

$$R_{22}\begin{cases} R_{221} = \{A,B,C\} \\ R_{222} = \{A,D,H,I,J,K\} \end{cases}$$

For R_{221}, A is the super-key therefore, R_{221} is in BCNF

For R_{222}, Candidate key is {A,D,H,I,J,K}, $A \to H$ which is derived from $A \to C, E \to D, CE \to DH = CE \to EH = C \to H$ so $A \to$H but A is not super-key so

$$R_{222}\begin{cases} R_{2221} = \{A,H\} \\ R_{2222} = \{A,D,I,J,K\} \end{cases}$$

For R_{2221}, A is the super-key therefore, R_{2221} is in BCNF

For R_{2222}, $\{A,D,I,J,K\}$ is super-key of R_{2222} thus, R_{2222} is in BCNF

The decomposition of R is $R_1, R_{21}, R_{221}, R_{2221}, R_{2221}$ and R_{2222}. In this process, it is lossless-join, however, $EI \to J$ has been lost so it is not dependency-preserving.

Therefore, it is not possible to decompose R into a collection of BCNF relations and ensure the decomposition is dependency-preserving and lossless-join.

Question2:

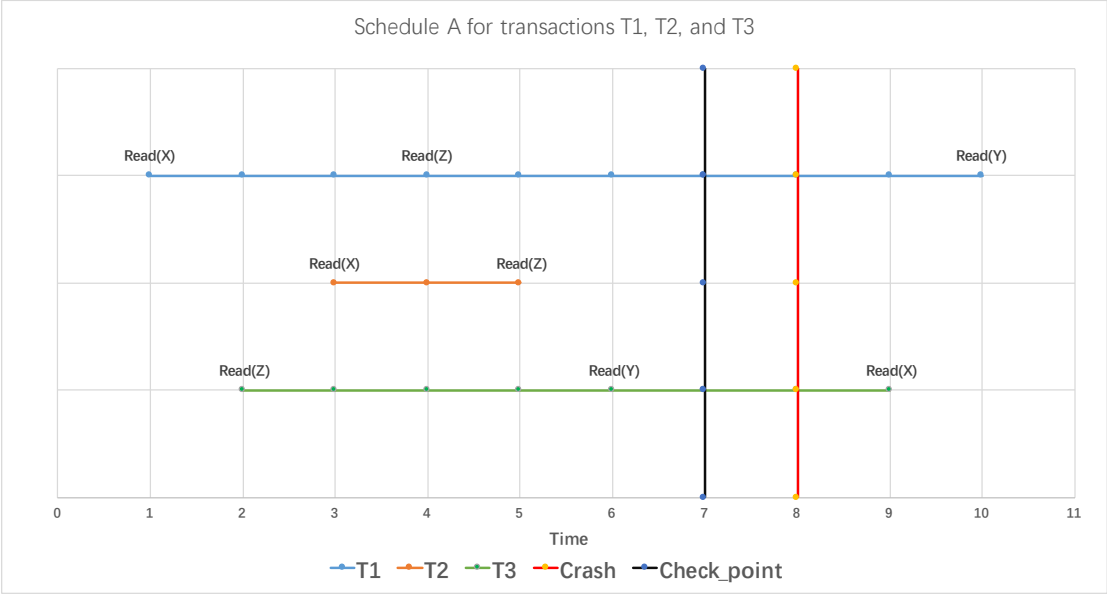

Figure 1

a):

As can be shown from figure1, T1 and T3 have not been completed before time 8, therefore, T1 and T3 are undo. T2 has been committed before time so T2 is redo.

b):

T1 and T3 should be undo since the times they consumed exceed time8.

We need not redo T2 as the database updates including T2 in buffers are force-written before Time7.

Question3:

1):

Date page: p1, p2 and p3

Queries: $Q_1{:}p1,\ Q_2{:}p2,\ Q_3{:}p3, (Q_4{:}p2,\ Q_5{:}p3)_{\boldsymbol{loop}}$

Buffer:

p1	p2

For First in First Out (FIFO) buffer replacement policy:

In Q3, p3 would be introduced and p1 would be replaced as p1 is first in.

Buffer would become

p3	p2

Then in Q4 and Q5, system would directly read p2 and p3. In each loop, system would also directly read p3 and p2.

For Most Recently Used (MRU) buffer replacement policy:

In Q3, p3 would be introduced and p2 would be replaced since it is mostly recently used.

Buffer:

p1	p3

Then in Q4, p2 would be introduced and p3 would be replaced since p3 become the new one which is mostly recently used.

Buffer would become

p1	p2

In Q5, p3 would be introduced and p2 would be replaced since p2 become the new one which is mostly recently used.

Buffer would become

p1	p3

In this loop, for Most Recently Used (MRU) buffer replacement policy, the cache miss would generate and the system would need to repeatedly read p2 and p3 in hard disk, thus, the buffer would introduce p2 and p3 repeatedly. This would take much time.

Therefore, First in First Out (FIFO) buffer replacement policy is better than Most Recently Used (MRU) buffer replacement policy in this scenario.

2):

Date page: p1, p2 and p3

Queries: $Q_1: p1,\ Q_2: p2,\ Q_3: p1, Q_4: p3, (Q_5: p2, Q_6: p1, Q_7: p3)_{\boldsymbol{loop}}$

Buffer:

p1	p2

For First in First Out (FIFO) buffer replacement policy:

In Q3, p1 would be directly read and the buffer is consistent. In Q4, p3 would be introduced and the p1 would be replaced as it is first in.

Buffer would become

p3	p2

Then in Q5, p2 would be directly read and in Q6, p1 would be introduced and p2 would

be replaced. The buffer would become

p3	p1

In Q7, p1 would be directly read.

Therefore, in each loop, the hard disk is only accessed once.

Least Recently Used (LRU) buffer replacement policy:

In Q3, p1 would be directly read and the buffer is consistent but p2 would become the one which is least recently used. In Q4, p3 would be introduced and the p2 would be replaced as it is least recently used.

Buffer would become

p1	p3

Then in Q5, p2 would be introduced and p1 would be replaced, the buffer would become

p2	p3

and in Q6, p1 would be introduced and p3 would be replaced. The buffer would become

p2	p1

In Q7, p3 would be introduced and p2 would be replaced. The buffer would become

p3	p1

Therefore, in each loop, the hard disk is accessed thrice. This method speeds more much than FIFO in this scenario.

As a result, First in First Out (FIFO) buffer replacement policy is better than Least Recently Used (LRU) buffer replacement policy in this case.