FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

ACADEMIC YEAR 2021/2022

OCTOBER FINAL ONLINE ASSESSMENT

BACS3183 ADVANCED DATABASE MANAGEMENT

ANSWER SHEET

INSTRUCTIONS

1. Type / attach your answer in the spaces provided.
2. Do not add or remove any session from this answer sheet.
3. Upon finishing, convert your answer sheet into **PDF format** and rename the file with the format [Programme]_[Group]_[YourName] (e.g: REI2_G5_Adam Lim)
4. Submit your answer sheet to Google classroom. Note that only one submission is allowed and it is your responsibility to ensure that you're submitting the correct answer and version.
5. For any late submission after the stipulated time frame or no submission, it is deemed to fail this final online assessment.

STUDENT'S DECLARATION OF ORIGINALITY

By submitting this online assessment. I declare that this submitted work is free from all forms of plagiarism and for all intents and purposes is my own properly derived work. I understand that I have to bear the consequences if I fail to do so.

Course:	BACS3183
Course Title:	ADVANCED DATABASE MANAGEMENT
Date:	Tuesday, 5 October 2021 (9:00 AM – 12:00 NOON)
Programme & Group:	RSD3G6
Student Full Name:	Lim Chan How
Student ID:	20WMR09309
Signature:	

BACS3183 ADVANCED DATABASE MANAGEMENT (Answer Sheet)

QUESTION 1 a) (i)

Insertion anomaly
In insertion anomaly, there are circumstances in which certain facts cannot be recorded at all. For example, a new book with the new catalog no C126 cannot be inserted into the table until a member borrows it.

Modification anomaly
For modification anomaly, the same information can be expressed on multiple rows and cause the modification of the table may result in logical inconsistencies. For example, if the first row of the C116 book title name is updated, the second row of the C116 book title will not automatically update and remain the same old data.

Deletion anomaly
For deletion anomaly, the deletion of the data representing certain facts necessitates deletion of data representing completely different facts. For example, member name Cindy deleted and all the information of Cindy will also be lost.

QUESTION 1 a) (ii)

CatalogNo -> BookID,Title,CategoryCode,CategoryDesc,MemberNo,MemberName,GuardianHP,DateOut,DateReturn

CatalogNo,BookID,MemberNo -> DateOut,DateReturn

BookID -> Title,CategoryCode

CategoryCode -> CategoryDesc

MemberNo -> MemberName,GuardianHP

BACS3183 ADVANCED DATABASE MANAGEMENT (Answer Sheet)

QUESTION 1 a) (iii)

Catalog(CatalogNo,BookID*,MemberNo*,DateOut,DateReturn)

Book(BookID,Title,CategoryCode*)

Category(CategoryCode,CategoryDesc)

Member(MemberNo,MemberName,GuardianHP)

QUESTION 1 b) (i)

```
CREATE VIEW NewDesign_report AS
SELECT COUNT(a.empId) AS totalEmp, SUM(a.salary) AS totalSalary, b.branchId, c.deptID,
c.deptName, c.budget
FROM Employee a, Branch b, Department c, Job d
WHERE a.deptId = c.deptId
AND b.branchId = c.branchId
AND a.jobCode = d.jobCode
AND c.zone = 1 AND c.zone = 3
AND d.jobTitle = 'Administrative Assistant'
AND YEAR(a.joinDate) - YEAR(sysdate) >=5
AND YEAR(a.joinDate) - YEAR(sysdate) <=9
GROUP BY b.branchId, c.deptId, c.deptName, c.budget
```

Advantages of view

- Security
  Can set the permission to different users to access the view.

- Modification restrictions
  View table can restrict any user to do modifications including create, update and delete.

- Performance
  Because of view only contain select query, it can be execute and show the data quickly.

QUESTION 1 b) (ii)

- List the details of all department in ascending order of budget
  B + tree indexing. First, the table will get all department data in ascending budget. By B+ binary tree indexing, the indexing will fetch for the intermediary node one by one in the department table. In the intermediary node, then will direct to the leaf node of each department ID.

- List the ID and name of each employee with salary above RM9000 in a specific branch
  Hash indexing. Because of a specific branch, hashing is good for searching for it. Hash index will organize the search keys with their associated record pointers and into the hash file structure. The hash index will use overflow buckets to store the primary key.

QUESTION 1 b) (iii)

Clustering index

- The ordering of index matches the ordering of value

Non-clustering index

- Stored separately, may have multiple non-clustered indexes for each table.

QUESTION 2 a) (i)

	T1	**T2**		
t1	read(X)	read(Z)	100	150
t2	X = X- 10	read(X)	90	100
t3	write(X)	X = X+Z	90	250
t4	read(Y)	write(X)	90	250
t5	Y = Y+10			
t6	write(Y)			

Lost update problem is found when the T1 successfully completed update is overridden by the user in T2.

In t1 X is still a value of 100. After that, X is minus 10 and becomes 90 in T1 while X is still 100 in T2. In t3, T1 writes the value X 90 into the database, but T2 plus the X with value Z 150 become 250. Lastly in t4, T2 write the new value X which is 250 into database and cause the update value X in T1 is lost and overridden with the new value.

QUESTION 2 a) (ii)

	T1	**T2**		
t1	write_lock(X)			
t2	read(X)	read(Z)	100	150
t3	X = X- 10	wait	90	150
t4	write(X)	wait	90	150
t5	commit/unlock(X)	wait	90	150
t6	read(Y)	write_lock(X)	120	90
t7	Y = Y +10	read(X)	130	90
t8	write(Y)	X = X+Z	130	240
t9	commit	write(X)		240
t10		commit/unlock(X)		

First, T1 will write a lock with X value 100. In t3, X minus with 10 and become 90 then write into the database and unlock. At this moment, T2 will wait until T1 is unlocked X. After that, T2 will write lock X which 90 is updated from T1 and add with 150 becomes 240. The X will be written into the database again in T2 and unlocked after it is done. Until t10, the value of X is updated to 240, Y is updated to 130 and the Z remains the same since there is no transaction of it.

QUESTION 2 a) (iii)

	T1	**T2**
t1	read(X)	
t2	X = X- 10	
t3	write(X)	read(Z)
t4		read(X)
t5	rollback	X = X+Z
t6	read(Y)	write(X)
t7	Y = Y+10	
t8	write(Y)	

Occurs when one transaction can see intermediate results of another transaction before it has committed.

QUESTION 2 b)

Checkpoint is the point of synchronization of database and log file. All buffers are force-written to secondary storage.

1. Suspend execution of transactions temporarily
2. Force write modified buffer data to disk.
3. Write modified buffer data to disk.
4. Resume normal transaction execution.

T1 has committed before the most recent checkpoint. T2 needs to be redo execution because the most recent checkpoint was active. T3 needs to be undo the execution because it executes at the time of system failure. T4 is redo execution to most recent checkpoint but has completed its execution before system failure.

QUESTION 2 c)

Fragmentation

- horizontal
  Different rows of table distributed to different sites
- vertical
  Different columns of table distributed to different sites
- mixed
  Combination of horizontal and vertical

Replication

- snapshot
  generally used when data changes are infrequent
- near real-time
  needed in environment with near-real-time requirements

Centralized

Centralized locking is that system maintains a single lock manager that resides in a single chosen site.