Assignment 6 Solution

Name: Yanjun Chen, PSU ID: yfc5289

Part I: Problem 1

A: Atomicity
- either the txn performs all the effects successfully, or it performs none of the effects. In another words, either all the functionality works perfect, or none of them works.
For instance, we perform a one dollar transaction from account A to account B. We need to deduct 1 dollar from A and then add that 1 dollar to account B. For this atomicity, either A loses 1 dollar and B gets 1 dollar (all operations perform successfully), or A keeps that 1 dollar and B remains the same (none of the operations performed).
C: Consistency
- Integrity holds for all the place that txn moved.
Continue use the example as above: if account only has 1 dollar and B has no money before transaction, then after transaction performed successfully, A has no money while B has 1 dollar. The overall amount of money between these 2 accounts remains 1 dollar overall, which indicates the integrity holds well for all the place that txn moved.
I: Isolation
- The txns are going to be executed one after another and perform each effects.
If we add 1 dollar to account A first, and then add 2 dollars to account B, account A will not be effected by the step of adding money to account B.
D: Durability
- Once the txn has been performed and committed, its effects will be always recorded inside the database.
For instance, account A transfers 1 dollar to account B, successfully and committed. Even if user of account B cancelled his account right after this transaction, the transaction will be recorded inside the database.

Part I: Problem 2

1. Read commit.
T1 first reads all x = 100, then T2 updates all x = 100 into x = 102 and committed. Then T1 read again to find all x which less than 200. This will effect the result as x = 100 is now x = 102. There is no dirty read, phantom read and non-repeatable read. So we could use read commit level.

2. Serializable.
For the given actions, we first read all x = 100, then insert x = 102 and commit. When we are finding all x ¡ 200, we will read both x = 100 and x = 102. Reading x = 100 will not effect the insertion of x = 102 (The insertion of x = 102 could be moved around).
This schedule satisfies serializable isolation level because the result is same when executing Transaction 1 followed by Transaction 2.

3. Repeatable Read.
For the given actions, T1 reads x = 100 and T2 insert another x = 100. It caused overlapped data. They are different data even though they are both x = 100. This will cause or need a phantom read as we committed T2 first. Therefore, we need to use repeatable read as the isolation level.

Part I: Problem 3

1.
This is not a serializable schedule. When T_1 performs $R_1(A)$ and then T_3 performs $W_2(A)$. A has been changed. While T_1 trys to read A again at the end of this schedule, they will not read the same thing as the first time T_1 read A. So there is an unrepeatable read and this is not a serializable schedule.

2.
Here is the schedule:
$R_1(A), W_1(B), R_2(B), W_2(D), R_3(B), W_1(D), R_1(A), W_3(A), R_3(D), W_2(D), W_2(C), W_3(C)$**.**
Compare with the original schedule, I swap the sequence of T_2 and T_3 read B. Because this is the first time for both T_2 and T_3 read B and there is no write operations between the 2 read operations. No matter they read in which sequence, the result they read will be the same.
Therefore, this is a conflict equivalent schedule to the original one.

3.
$R_1(A), R_2(B), R_3(B), W_2(B), W_1(B), R_3(D), W_1(D), R_1(A), W_2(A), W_2(C), W_3(C)$**;**
$R_1(A), R_3(B), R_2(B), W_2(B), W_1(B), R_3(D), W_1(D), R_1(A), W_2(A), W_2(C), W_3(C)$
Compare with the 2 conflict serializable schedules I provided, I only swap the sequence when T_2 and T_3 first read B. Because they will read the same B when there is no write operations between, the sequence will not effect their results.

Part II: Problem 1

1.

```
CREATE VIEW StudentName AS
 SELECT DISTINCT sname From Enrollment;
CREATE VIEW StudentGradeReport AS
 SELECT AVG(grade), sname AS std_avg_grade FROM Enrollment
 GROUP BY sname;
```

2.
As we only grant the privileges to know only the CSE and MATH department, we need to create the view for these 2 departments first.
After we create this view,

```
CREATE VIEW CSE_MATH_AvgGrades AS
 SELECT dname, AVG(grade) AS dpt_avg_grade
 FROM Enrollment WHERE dname IN ('CSE', 'MATH')
 GROUP BY dname;
GRANT SELECT ON CSE_MATH_AvgGrades TO director;
```

3.
Notes: **We will use the same view create in the first subquestion.**

```
GRANT SELECT ON StudentNames TO secretary WITH GRANT OPTION ;
```

Part II: Problem 2

1.
- Information over-collect and use the personal data in wrong way:
Data brokers like NPD collect large amounts of personal information from various sources, including public records, online activity, and so on.
Sensitive information like Social Security Numbers may affect the user's personal privacy issues, or even personal credit issues.
There will be more potential ethical issues when these companies collect more data than necessary such as ask your SSNs when you just need to sign up for some video website, and use this kind of information while the users may not explicitly consented.
If this sensitive data is exposed in a breach, it can lead to identity theft, financial fraud, or unauthorized use of personal profiles. Individuals who affected by this situation may face long-term harm, including but not limited to reputational damage and financial loss.

- Selling personal data to third parties.
Data brokers like NPD often sell collected data to third parties for profit. Some buyers may use this data for legitimate purposes, such as text messaging to promote products and deliver targeted surveys. Others may have unethical or illegal intentions, such as targeting vulnerable people (the elderly or low-income individuals) to defraud. Data breaches make users' sensitive information public, further increasing the risk that this information can be used for harmful purposes such as illegal surveillance. This may also cause damage to the user's reputation, property and so on.

2.
Scenario: NPD collects and sells personal data to third parties, including sensitive information such as Social Security numbers (SSN) and financial details. One of its clients was a fraud organization that used the data to target individuals. This can result in significant financial losses and potential identity theft issues for affected individuals.

To solve this problem, we can use methods as following:
1. Use encryption and tokenization of sensitive data to ensure that even vetted buyers only have access to data they are authorized to see.
2. Require buyers to sign contracts that explicitly prohibit misuse of data.

3.
Here is the link to the similar event / scenario I described: Equifax data breach in 2017.

`https://en.wikipedia.org/wiki/2017_Equifax_data_breach?utm_source=chatgpt.com`

Conditions:
The information obtained in the breach includes the first and last names, Social Security numbers, dates of birth, addresses and sometimes driver's license numbers of about 143 million Americans, according to an analysis by Equifax.
Discussion:
1. Both NPD and Equifax did not implement strong cybersecurity protocols, which cause large amounts of personal data vulnerable to unauthorized access.
2. In both cases, as the result of data breach, affected individuals have risks like identity theft and financial fraud.
Areas that needs extra cautions:
1. Carefully check the background of any company before sharing data and make them follow strict rules.
2. Quickly let people know about breaches so they can take steps to protect themselves.

Part III

Use my project for this course as an example: this is an e-commerce system with large scale amount of data and corresponding attributes such as catagories, prices, item reviews, ratings and so on.
It will compare the data horizontally, analyze them and perform results in the form of tables and charts.
NoSQL document DBMS has advantages: more flexible schema and support horizontal scaling. But also has disadvantages: less ACID properties.