Database Administration and Tuning

2nd semester

Mini-Project 2

Question 1 – Transaction Isolation Levels

Consider a relational database for information related to cheese products, with the following tables:

CHEESE(cheeseID, type, producer, calories, proteins)

REGION(regionID, name, country)

PRODUCTION (productionID, cheeseID, season, amount)

PROVENANCE (productionID, regionID)

Assume that the following three stored procedures can run concurrently in a given application that is supported by the relational database:

1) insert_cheese : creates new records in the CHEESE table, for new cheeses that are produced. This procedure uses only the CHEESE table.

2) update_production: inserts new records, or modifies existing PRODUCTION and PROVENANCE records. This procedure writes to the PRODUCTION and PROVENANCE tables, updates the tuples in the PROVENANCE table, and may be reading from the REGION and CHEESE tables.

3) delete_region: deletes a given region from the REGION table.

1.1. Give a scenario that leads to a possible dirty read in the concurrent execution of operations from this group of stored procedures, or explain why a dirty read cannot happen in this group of stored procedures.

1.2. Give a scenario that leads to a possible non-repeatable read in the concurrent execution of operations from this group of stored procedures, or explain why a non-repeatable read cannot happen in this group of stored procedures.

1.3. Give an example of a possible overwriting of uncommitted data in the concurrent execution of operations from this group of stored procedures, or explain why a phantom read cannot happen in this group of stored procedures.

1.4. Indicate what transaction isolation level would you use for executing each of the three procedures above, and why? For each procedure you should use the least restricted transaction isolation level that ensures correctness.

Question 2 – Concurrency Control

2.1. Consider the following schedule for three concurrent transactions:

	T1	**T**2	**T**3
1	write(**A**)		
2			write(**B**)
3		write(**A**)	
4	write(**B**)		
5		write(**B**)	

(a) Is the schedule allowed in Strict 2-Phase Locking? Justify.
(b) Is the schedule allowed by the timestamp-based protocol? Justify.

2.2. Consider the following schedule for two concurrent transactions:

	T1	**T**2
1	lock-S(**A**)	
2	read(**A**)	
3	unlock(**A**)	
4		lock-X(**B**)
5		write(**B**)
6		unlock(**B**)
7	lock-S(**B**)	
8	read(**B**)	
9	unlock(**B**)	

(a) Is the schedule allowed in Strict 2-Phase Locking? Justify.
(b) Is the schedule allowed by the timestamp-based protocol? Justify.

Question 3 – Recovery System

Consider the following simplified representation for the log records that correspond to a given execution, and suppose the ARIES algorithm is followed by the recovery system:

LSN	**Type**	**Transaction**	**Page**
10	Update	T1	P1
20	Update	T2	P2
30	Begin_checkpoint	-	-
40	End_checkpoint	-	-
50	Commit	T1	
60	Update	T3	P3
70	Commit	T2	-
80	Update	T3	P2
90	Update	T3	P5
	CRASH!!!		

Show the contents of the log, the dirty page table, and the active transactions table after (a) the analysis phase, (b) the redo phase, and (c) the undo phase.

Question 4 – Schema and Index Tuning

Consider the relation CHEESE from Question 1. For each cheese, the relation lists an id, a name, a number of calories, and a number of proteins:

CHEESE(cheeseID, Type, Producer, Calories, Proteins)

The following two queries are extremely important and frequent:

Q1) Finding the average number of calories by cheese type, for all cheeses.

Q2) Listing the cheese ids of the cheeses with most proteins.

4.1 Write the two queries in SQL.

4.2. Suppose you have a heap file to store the records. What indexes would you create to speed up the two queries above? Justify.

4.3. Suppose your customers complain that performance is not satisfactory. If you consider a schema redesign as a tuning option, explain how you would try to obtain better performance by describing the schema for the relation(s) that you would use and your choice of file organizations and indexes on these relations.

Question 5 – Query Tuning

For each of the following queries, identify one possible reason why an optimizer might not find a good execution plan. Rewrite the query so that a good plan is likely to be found. Any available indexes, or known constraints, are listed before each query.

5.1. An index is available on the *calories* attribute:

```
SELECT type
FROM CHEESE
WHERE calories * 100 < 800
```

5.2. A B+ Tree index is available on the *calories* attribute:

```
SELECT type
FROM CHEESE
WHERE calories<90 AND calories>40
```

5.3. An index is available on the *calories* attribute:

```
SELECT type
FROM CHEESE
WHERE calories=90 OR calories=40
```

5.4. No index is available:

```
SELECT DISTINCT CheeseId, type
FROM CHEESE
```

5.5. No index is available:

```
SELECT AVG (proteins)
FROM CHEESE
GROUP BY producer
HAVING type = “Alverca”
```

Question 6 – Database Tuning

Consider the following two queries:

```
Q1: SELECT type, producer
      FROM CHEESE NATURAL JOIN PRODUCTION NATURAL JOIN PROVENANCE NATURAL JOIN REGION
      WHERE country = ‘Portugal’ AND amount > 10,000 AND season = ‘Winter’

Q2: SELECT type, COUNT(*)
       FROM CHEESE
       GROUP BY type;
```

6.1. Which of these queries corresponds to a typical access pattern of OLTP applications? And which one corresponds to a typical access pattern of OLAP queries? Justify your answer.

6.2. In which of the queries would it make sense to use a low isolation level (e.g. READ UNCOMMITED)? And a higher isolation level (e.g. SERIALIZABLE)? Justify your answer.

6.3. Consider the different tuning levels considered in the lectures (schema, index, queries, etc). For the first query, indicate two possible optimizations at two different levels.