# COMP 451/651 – Database Design

Department of Computer Science
Concordia University

Khaled A. Jababo
Answer all questions
**Closed Book**

**Final Exam**
180 min
Fall 1995

**Question 1** (20 points) Briefly explain in your own words, the following

1. System error.
2. Log.
3. Wound-wait scheme.
4. Advantages and disadvantages of replication in distributed databases.
5. Advantages and disadvantages of distributed databases.
6. Advantages of object oriented databases over relational databases.
7. Difference between equality of object identifiers and value equality in Object Oriented Databases.

**Question 2** (6 points) Consider transactions $T_1$, $T_2$ , and $T_3$ such that $T_1$ reads x and then writes y, $T_2$ writes x and then writes y, and finally $T_3$ reads x, then reads y and then writes z.

Give example executions of $T_1$, $T_2$ and $T_3$ such that they are serializable and

1. not recoverable [1.5 points].
2. recoverable but not cascadeless [1.5 points].
3. recoverable and cascadeless but not strict [1.5 points].
4. strict [1.5 points].

**Question 3** (20 points) Answer the following questions, (be brief).

1. Explain multi-version protocol in your own words [5 points].
2. Do you think that the three phase commit protocol is better than the two phase commit protocol ? (Why?) [5 points].
3. Consider a multidatabase system. Do you think ensuring local serializability in each site ensures global serializability ? Why ? Use example(s) to clarify your points. [5 points].
4. Assume distributed database system, all local databases use strict two phase locking. Can global serializability be ensured ? Justify your answer [5 points].

**Question 4** (20 points) Consider a distributed database system with several sites.

(i) Site1 contains the relation suppliers, with 15,000 tuples. Each tuple is 300 bytes. The relation has the following scheme: SUPPLIERS = {S#, Name, Address}. S# is 12 bytes, and is the primary key. There is a $B^+$-tree index on S#.

(ii) Site 2 contains the relation parts, with 150,000 tuples. Each tuple is 150 bytes. The relation has the following scheme: PARTS = {P#, Part-Name, Color, Wheight}. P# is 12 bytes, and is the primary key. There is a $B^+$-tree index on P#.

(iii) Site 3 contains the relation CAN-SUPPLY with 15,000,000 tuples. Each tuple is 60 bytes. The relation has the following scheme: CAN-SUPPLY = {S#, P#, Price}. {S#, P#} is the primary key. There is no index available on this relation.

**Assumptions:** Assume uniform distribution of data. For example in CAN-SUPPLY, each supplier can supply the same number of parts, and each part can be supplied by the same number of suppliers, on the average. Write down any other assumption that you need to make. Your assumptions should be reasonable.

The following query is submitted at site 4, and the answer is needed at site 4.

**select** suppliers.Name, suppliers.Address
**from** Suppliers , CAN-SUPPLY
**Where** suppliers.S# = can-supply.S# *and*
CAN-SUPPLY.P# = 343536

(Query list supplier names & addresses for suppliers who can supply part 343536.)

1. Describe one of the best way of processing this query. The primary objective is to minimize network transmission. The secondary objective is to reduce disk I/O. Be precise, and give a step by step description. [10 points].
2. For each step of your answer to part (a) estimate the number of blocks/pages of data transmitted over the network, and the number of blocks/pages of disk I/O. Be precise and brief. Justify your answers. Each block/page is 3000 bytes. [10 points].

**Question 5** (34 points)

Consider the following schedule S. Assume that transaction $T_1$ is submitted at time 1, transaction $T_2$ is submitted at time 2, and similarly transaction $T_3$ is submitted at time 3. Initial values for data items $a$, $b$, and $c$ are respectively 100, 200 and 300.

| | $T_1$ | $T_2$ | $T_3$ |
|---|---|---|---|
| 1 | read(a) | | |
| 2 | a := a+10 | | |
| 3 | | | read(b) |
| 4 | | | b := b+20 |
| 5 | | read(c) | |
| 6 | | c := c+10 | |
| 7 | | | write(b) |
| 8 | | | commit |
| 9 | read(c) | | |
| 10 | read(b) | | |
| 11 | b := b+c | | |
| 12 | write(a) | | |
| 13 | write(b) | | |
| 14 | commit | | |
| 15 | | read(a) | |
| 16 | | read(b) | |
| 17 | | a := a+c | |
| 18 | | b := b+a | |
| 19 | | write(a) | |
| 20 | | write(b) | |
| 21 | | write(c) | |
| 22 | | commit | |

1. Assume that the system is running under validation protocol and is utilizing incremental log with immediate updates for crash recovery. Show the log entries assuming each transaction starts with its first instruction and commits immediately following its last instruction [6 points].

2. Now assume that the system crashes after step 21. What recovery action do you suggest when the system is brought back up? [6 points].

3. Is the schedule S conflict serializable? (explain) If it is identify a serial schedule S' that is conflict equivalent with S [6 points].

4. Is the schedule possible under time stamp ordering protocol? (explain why/why not) [8 points].

5. Name one protocol under which the schedule is possible, if at all (explain) [8 points].

# COMP 451/651 – Database Design

Department of Computer Science
Concordia University

**Final Exam**
180 min
Winter 1995

Khaled A. Jababo
Answer all questions
**Closed Book**

1. **Question 1** (20 points) Briefly explain in your own words, the following
   (a) Query optimization.
   (b) Difference between a clustering index and a non-clustering index.
   (c) Consistency error.
   (d) Checkpointing.
   (e) Rollback.
   (f) Conflict serializable schedule.
   (g) Precedence graph.
   (h) Advantages and disadvantages of distributed databases.
   (i) Class hierarchy and inheritance.
   (j) Difference between a composite object and a versioned object in Object Oriented Databases.

2. **Question 2** (6 points) Suppose transaction $T_1$ writes x, then writes y, then writes z and then commits. Suppose transaction $T_2$ reads y, then writes x and then commits.

   Give example executions of $T_1$ and $T_2$ that are serializable and
   (a) not recoverable [1.5 points].
   (b) recoverable but not cascadeless [1.5 points].
   (c) recoverable and cascadeless but not strict [1.5 points].
   (d) strict [1.5 points].

3. **Question 3** (20 points) Answer the following question. (be brief)

   (a) Explain multiversion protocol in your own words [5 points].

   (b) Explain three phase commit protocol in your own words [5 points].

   (c) Explain how the failure of a participating site is handled in two phase commit protocol [10 points].

4. **Question 4** (14 points) Consider the following query on the relations scheme below.

   *Supplier* = {*SupId, Name, City, Province*}

   *Supplies* = {*SupId, Item, Price*}

   **select** Name
   **from** Supplier U, Supplies S
   **Where** S.Name = 'Bay' *and*
   U.SupId = S.SupId *and*
   S.Item = 'Microwave' *and*
   S.Price ≤ 250 *and*
   'Quebec' *in*
   (**select** Province
   **from** Supplier V
   **where** V.SupId = S.SupId)

   Assume that Supplier relation has 2,000 tuples and Supplies relation has 150,000 tuples. A disk block size is 2048 bytes and each record in Supplier is 200 bytes long, and each record in supplies is 20 bytes long. $B^+$-tree clustering indexes are available on SupId in relation Supplies and on Province in relation Supplier. Non-clustering $B^+$-tree indexes are also available on Item and Price in Supplies.

   Following additional statistics are also available. V(SupId, Supplies) = 100, V(Item, Supplies) = 3,000, V(SupId, Supplier) = 100, V(Province, Supplier) = 13.

   (a) Give an efficient RA expression for the above query [4 points].

   (b) Estimate the cost (block accesses, assume index fit in main memory) of the query using block-oriented join strategy [5 points].

   (c) How many block accesses we can save if we use Merge-join method in (b) instead of using block-oriented method [5 points].

5. **Question 5** (40 points) Consider the following schedule S. Assume that transaction $T_1$ is submitted at time 1, transaction $T_2$ is submitted at time 2, and similarly transaction $T_3$ is submitted at time 3.

Initial values for data items $a$, $b$, and $c$ are respectively 100, 200 and 300.

| | $T_1$ | $T_2$ | $T_3$ |
|---|---|---|---|
| 1 | read(a) | | |
| 2 | | | read(b) |
| 3 | a := a+10 | | |
| 4 | | | b := b+20 |
| 5 | write(a) | | |
| 6 | commit | | |
| 7 | | | write(b) |
| 8 | | read(a) | |
| 9 | | | read(a) |
| 10 | | | read(c) |
| 11 | | z := 100 | |
| 12 | | a := a+10 | |
| 13 | | b := z+a | |
| 14 | | write(b) | |
| 15 | | | c := c+a+b |
| 16 | | | write(c) |
| 17 | | | commit |
| 18 | | write(a) | |
| 19 | | commit | |

(a) Assume that the system is running under validation protocol and is utilizing incremental log with immediate updates for crash recovery. Show the log entries assuming each transaction starts with its first instruction and commits immediately following its last instruction [8 points].

(b) Now assume that the system crashes after step 18. What recovery action do you suggest when the system is brought back up? [8 points].

(c) Is the schedule S conflict serializable? (explain) If it is identify a serial schedule S' that is conflict equivalent with S [8 points].

(d) Is the schedule possible under time stamp ordering protocol? (explain why/why not) [8 points].

(e) Name one protocol under which the schedule is possible, if at all (explain) [8 points].