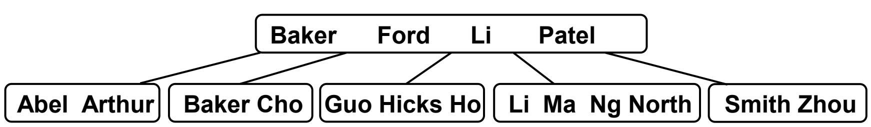
Review Problems

Computer Science 460
Boston University

David G. Sullivan, Ph.D.

Tree-Based Index Structure Problem

- Consider the following tree-based index structure, in which the keys are a person's last name:



- Is this a B-tree or a B+tree?

- If the tree has an order of 2, what would it look like after inserting a key/value pair for a person whose last name is Partridge?

Two-Phase Locking Problem

- Consider the following schedule:

T_1	T_2
	w(A)
	r(C)
r(A)	
	w(C)
w(A)	

- Is it possible under regular two-phase locking? If so, insert lock and unlock operations as needed.
- Is it possible under *strict* two-phase locking?

	regular	**strict**
A.	yes	yes
B.	yes	no
C.	no	yes
D.	no	no

Logging and Recovery Problem, Part I

- At the start of recovery, what are what are the possible on-disk values under *undo-redo* logging?

LSN	*record contents*
100	txn: 1; BEGIN
250	txn: 1; item: D1; old: 15; new: 25; olsn: 0
300	txn: 2; BEGIN
400	txn: 2; item: D2; old: 50; new: 35; olsn: 0
450	CHECKPOINT (active txns = 1, 2)
500	txn: 1; item: D3; old: 40; new: 20; olsn: 0
550	txn: 1: item: D1: old: 25; new: 75; olsn: 250
600	txn: 1; COMMIT
650	txn: 2; item: D3; old: 20; new: 10; olsn: 500
700	txn: 2; item: D2; old: 35; new: 40; olsn: 400

Logging and Recovery Problem, Part II

- What steps are taken during recovery if we use the LSNs to ensure that actions aren't taken unnecessarily?

LSN	*record contents*	*backward pass*	*forward pass*
100	txn: 1; BEGIN		
250	txn: 1; item: D1; old: 15; new: 25; olsn: 0		
300	txn: 2; BEGIN		
400	txn: 2; item: D2; old: 50; new: 35; olsn: 0		
450	CHECKPOINT (active txns = 1, 2)		
500	txn: 1; item: D3; old: 40; new: 20; olsn: 0		
550	txn: 1: item: D1: old: 25; new: 75; olsn: 250		
600	txn: 1; COMMIT		
650	txn: 2; item: D3; old: 20; new: 10; olsn: 500		
700	txn: 2; item: D2; old: 35; new: 40; olsn: 400		

datum LSNs
D1: 550
D2: 400
D3: 0

ACID Properties Problem

- Recall that a transaction has the following "ACID" properties:
  - Atomicity: either all of its changes take effect or none do
  - Consistency preservation: its operations take the database from one consistent state to another
  - Isolation: it is not affected by other concurrent transactions
  - Durability: once it completes, its changes survive failures
- If you disable locking in your DBMS, which of these properties do you lose?
- What if you disable logging?
- What other properties do we need to ensure full isolation?

SQL Problem

- Recall our movie database from PS 2:

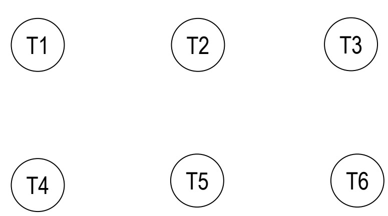
  Movie(id, name, year, rating, runtime, genre, earnings_rank)
  Person(id, name, dob, pob)
  Actor(actor_id, movie_id)
  Director(director_id, movie_id)
  Oscar(movie_id, person_id, type, year)

- Write a query to retrieve the winners (name and type) of the acting and directing Oscars in the year 2000.

Deadlock Detection Problem

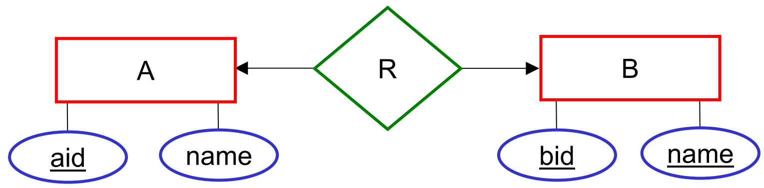
- Would the following schedule produce a deadlock under rigorous 2PL? Assume that no commits occur during the sequence of operations shown.

  $r_4(B)$; $r_1(B)$; $r_3(A)$; $r_5(C)$; $r_2(C)$; $r_6(D)$; $w_2(B)$; $r_1(A)$; $w_5(A)$; $w_6(C)$; $w_3(D)$; …



ER Diagram Problem

- Consider the following ER diagram:



- What constraints does this diagram specify?

  uniqueness constraints:

  cardinality constraints:

- What attributes would be needed for the primary key of the relation used to capture R?

XML Problem

- Consider an XML document that looks like this:

```
<students>
  <student>
      <name>Alan Turing</name>
      <course>CSCI E-119</course>
      <course>CSCI E-268</course>
      <course>CSCI E-50a</course>
      <course>CSCI E-50b</course>
      <course>CSCI E-160</course>
      <course>CSCI E-215</course>
      <course>CSCI E-113</course>
  </student>
  …
</students>
```

- Give two different ways of finding the names of all students taking CSCI E-268.

MongoDB Problem

```
{ _id: "0499549",
  name: "Avatar",
  year: 2009,
  rating: "PG-13",
  runtime: 162,
  genre: "AVYS",
  earnings_rank: 1,
  actors: [ { id: "0000244", name: "Sigourney Weaver" },
            ... ],
  directors: [ { id: "0000116", name: "James Cameron" } ]
}
```

- Write a query to find, for each director of a top-200 movie, the number of top-grossing movies by that director.

Timestamp-Based Concurrency Control Problem

- How will this schedule be executed (without commit bits)?

$w_1(B); r_2(A); w_2(A); r_1(A); r_3(C); w_3(B); w_2(B); w_2(C); r_3(A); w_3(C)$

T1	T2	T3	A	B	C
			RTS = WTS = 0	RTS = WTS = 0	RTS = WTS = 0

Replication Problem

A database is being replicated across 20 sites,
and a client wants to update one of the data items.

- If the database is being managed by a system that uses synchronous replication and *fully distributed locking*, what is the minimum number of copies that a client would need to update?

- What if the system were using synchronous replication and *primary-copy locking*?

Two-Phase Commit (2PC) Problem

A transaction T requires subtxns at sites A, B, and C.
Site A is the coordinator.

Consider the following sequence of events from the 2PC of T:

1. After the last operation in T, site A:
   - sends prepare messages to B and C
   - puts its own subtxn in the ready state
2. Sites B and C:
   - put their subtxns in the ready state
   - force-write ready records and send ready messages to A
3. Site A force-writes its commit record and sends commit messages to B and C.
4. Site B receives the commit message from A and force-writes its own commit record.
5. Site C receives the commit message from A and force-writes its own commit record.

(continued)

Two-Phase Commit (2PC) Problem (cont.)

Now consider the following variations on the above scenario:

1. If C crashes *before* force-writing its commit record, what should happen during its recovery?

2. If B crashes *after* force-writing its commit record, what should happen during its recovery?

3. If B crashes *before* force-writing its **ready** record:
   - what should happen during its recovery?
   - what should A and C do?

4. If A crashes *before* force-writing its commit record, what should B and C do?