Concurrency Control 2

R&G Chapter 17

(slides adapted from content by J.Gehrke, J.Shanmugasundaram, and/or C.Koch)

Reminders

- Midterm on Monday.
  - Questions similar to the homeworks.
  - Closed book.
  - Homework 1-4 solutions posted tonight.
- No homework this week.

Recap-Equivalence

- Schedule Equivalence (when are 2 scheds the same)
  - Conflict Equivalent
    - All conflicting ops are in the same order
  - View Equivalent
    - All reads read the same value (initial or from the same xact)
    - All final writes come from the same xact.
- (Conflict | View) Serializable
  - = (Conflict | View) Equivalent to a Serial schedule.

Recap-Deadlock Prevention

- 1) Track which xacts are waiting for which xacts (waits-for graph)
  - If a cycle exists, kill one of the xacts.

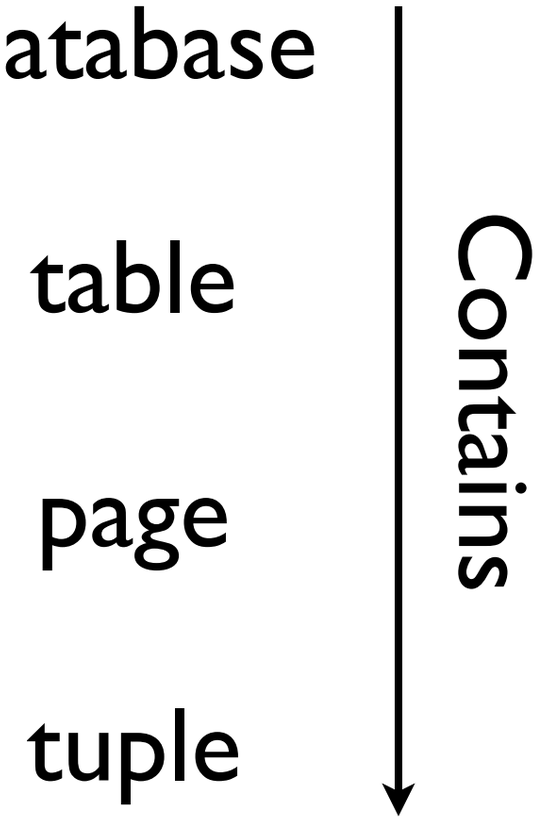
- 2) Prioritize transactions
  - a) Lower priority xacts die when they try to take a lock held by a higher priority xact.
  - b) Higher priority xacts kill any xact holding a lock that they need.

What is an Object (that we can lock)?

What do we lock?

- The entire **database**?
- A table in the database?
- Individual **pages** a table is stored on?
- Individual **tuples** in a table?

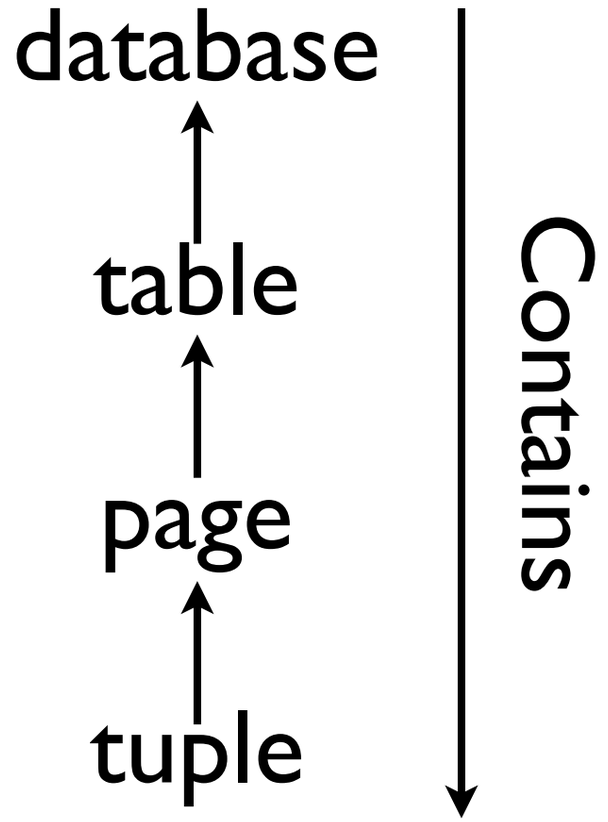
What do we lock?



What do we lock?

- We don't need to decide what to store!

- Data containers are nested!



What do we lock?

Is it safe to allow some transactions to lock tables while other transactions to lock tuples?

New Lock Modes

- ‘Intent’ to lock (a child)
  - Intent-to-Lock Shared (IS)
  - Intent-to-Lock Exclusive (IX)
- Actual lock (on the object)
  - Lock-Shared (S)
  - Lock-Exclusive (x)
- Lock Shared + Intent-to-Lock Exclusive (SIX)

New Lock Modes

Lock Mode(s) Currently Held By Other Xacts

Lock Mode Desired

	None	IS	IX	S	X
None	valid	valid	valid	valid	valid
IS	valid	valid	valid	valid	fail
IX	valid	valid	valid	fail	fail
S	valid	valid	fail	valid	fail
X	valid	fail	fail	fail	fail

Hierarchical Locks

- Lock Objects Top-Down
  - Before acquiring a lock on an object, an xact must have at least an intention lock on its parent!

- For example:
  - To acquire a S on an object, an xact must have an IS, IX on the object's parent (why not S, SIX, or X?)
  - To acquire an X (or SIX) on an object, an xact must have a SIX, or IX on the object's parent.

Hierarchical Locks

	None	IS	IX	S	X
None	valid	valid	valid	valid	valid
IS	valid	valid	valid	valid	fail
IX	valid	valid	valid	fail	fail
S	valid	valid	fail	valid	fail
X	valid	fail	fail	fail	fail

T1: Holds a S on a table

T2: Wants an X on a row of the table

Hierarchical Locks

	None	IS	IX	S	X
None	valid	valid	valid	valid	valid
IS	valid	valid	valid	valid	fail
IX	valid	valid	valid	fail	fail
S	valid	valid	fail	valid	fail
X	valid	fail	fail	fail	fail

T1: Holds a X on a table

T2: Wants an S on a row of the table

Hierarchical Locks

	None	IS	IX	S	X
None	valid	valid	valid	valid	valid
IS	valid	valid	valid	valid	fail
IX	valid	valid	valid	fail	fail
S	valid	valid	fail	valid	fail
X	valid	fail	fail	fail	fail

T1: Holds a S on a row of the table
T2: Wants a S on the table

Hierarchical Locks

	None	IS	IX	S	X
None	valid	valid	valid	valid	valid
IS	valid	valid	valid	valid	fail
IX	valid	valid	valid	fail	fail
S	valid	valid	fail	valid	fail
X	valid	fail	fail	fail	fail

T1: Holds a S on a row of the table
T2: Wants an X on the table

Hierarchical Locks

	None	IS	IX	S	X
None	valid	valid	valid	valid	valid
IS	valid	valid	valid	valid	fail
IX	valid	valid	valid	fail	fail
S	valid	valid	fail	valid	fail
X	valid	fail	fail	fail	fail

T1: Holds an X on a row of the table
T2: Wants an X on the table

Hierarchical Locks

- Release locks bottom up (child before parent)
- This protocol is equivalent to acquiring all child locks when the parent is acquired.
  - S requires IS/IX on parent (conflicts with X on parent)
  - X requires IX/SIX on parent (conflicts with S, X on parent)

Example

- T1 scans R and updates several tuples
  - T1 acquires SIX on R
  - **Scan**:T1 already has S on R.
  - **Update**:T1 acquires X on individual tuples as needed (needs IX on pages).

Example

- T2 uses an index to read part of R
- T2 gets an IS lock on R
- **Scan**: T2 repeatedly acquires S locks on pages of R as needed.

Example

- T3 scans all of R
  - T3 gets an S lock on R

    OR

  - T3 could behave like T2

    (IS+S on each page or tuple)
- Use lock escalation to decide which

Problem:
Locking assumes that we can lock all objects!

(What happens if we insert objects?)

T1

T2

```
DELETE FROM Officers
WHERE rank = 1
   AND age =
    (SELECT MAX(age)
     FROM Officers WHERE rank=1)
LIMIT 1;
```

Time

T1

T2

```
DELETE FROM Officers
WHERE rank = 1
   AND age = (71)
    (SELECT MAX(age)
     FROM Officers WHERE rank=1)
LIMIT 1;
```

Time

T1

```
DELETE FROM Officers
WHERE rank = 1
   AND age = (71)
    (SELECT MAX(age)
     FROM Officers WHERE rank=1)
LIMIT 1;
```

T2

```
INSERT INTO Officers(rank,age)
                VALUES (1, 96);
```

Time

Time

T1

```
DELETE FROM Officers
WHERE rank = 1
   AND age = (71)
    (SELECT MAX(age)
     FROM Officers WHERE rank=1)
LIMIT 1;
```

T2

```
INSERT INTO Officers(rank,age)
                VALUES (1, 96);
```

```
DELETE FROM Officers
WHERE rank = 2
   AND age = (80)
    (SELECT MAX(age)
     FROM Officers WHERE rank=2)
LIMIT 1;
```

Time

T1

```
DELETE FROM Officers
WHERE rank = 1
   AND age = (71)
    (SELECT MAX(age)
     FROM Officers WHERE rank=1)
LIMIT 1;
```

```
SELECT MAX(age)
FROM Officers(rank,age) (63)
WHERE rank = 2
```

T2

```
INSERT INTO Officers(rank,age)
                VALUES (1, 96);
```

```
DELETE FROM Officers
WHERE rank = 2
   AND age = (80)
    (SELECT MAX(age)
     FROM Officers WHERE rank=2)
LIMIT 1;
```

Time

T1

```
DELETE FROM Officers
WHERE rank = 1
   AND age = (71)
    (SELECT MAX(age)
     FROM Officers WHERE rank=1)
LIMIT 1;
```

T2

```
INSERT INTO Officers(rank,age)
                    VALUES (1, 96);
DELETE FROM Officers
 WHERE rank = 2
    AND age = (80)
     (SELECT MAX(age)
      FROM Officers WHERE rank=2)
LIMIT 1;
```

[INCONSISTENT]

```
SELECT MAX(age)
FROM Officers(rank,age)
WHERE rank = 2
```

(63)

The Problem

- T1 assumes that is has locked all sailor records with rating = 1
  - Solution 1: Lock entire table (expensive!)
  - Solution 2: Lock a **predicate**.

    (e.g., rating = 1)

    (To be continued next week)