Principles of Information and Database Management
198:336
Week 11 – Apr 18
Matthew Stone

Outline

Transactions
– Concepts
– Implementation
– Shortcuts

Web data
– Hubs and authorities
– Google PageRank
**Transaction**

Definition: an execution of a user program, seen by the DBMS as a series of read and write operations.

**ACID properties of transactions**

Atomic
Consistent
Isolated
Durable
Atomicity

Either all actions in a transaction execute or none of them do.

– Needs to be guaranteed by DBMS

Consistency

When run by itself – any transaction will leave the DB in a good state
Isolation

Each transaction is protected from the effects of other transactions that might be running at the same time
– No transaction can “tell” that other transactions are running

Durability

Once the DBMS informs the user that a transaction completed, its effects persist
Design choice

Transaction can be **aborted** by DBMS
- Terminated unsuccessfully
- May be bounced back
  - in this case none of it ever happened
- May be retried
  - DBMS starts over and makes it work

Transaction details

Oracle details
- In SQLPLUS, everything you do is one xact
- To end a transaction, use SQL commands
  - COMMIT
  - ROLLBACK
Transaction details

In MySQL command interface
– Need InnoDB tables, and transaction mode
  • set autocommit=0;
– Transactions have to be explicitly started
  • Start transaction;
– Then finish transactions as usual
  • COMMIT
  • ROLLBACK

Transaction details

In JDBC, part of the connection interface
– Need to start up transaction mode
  • conn.setAutoCommit(false);
– Like oracle, everything is in current xact
– Just need to end xact
  • conn.commit();
  • conn.rollback();
How transactions help

Actions by one process can put database in temporary, inconsistent state.
– need to make sure other processes don’t use this inconsistent state

Example – “midnight bank transfer”

Transfer $100 from account A to account B
– read A
– write A-$100
– read B
– write B+$100

Halfway through is an inconsistent state
– $100 has “gone missing”
**“Midnight bank transfer”**

Suppose it’s time to pay interest

Algorithm

- read A
- write A * 1.05
- read B
- write B * 1.05

**Bad soup!**

Suppose you pay interest in the moment when $100 is missing!
Either A or B gets ripped off.
Transactions

Let DB program say what should happen
– First
  • start transaction
  • r A, w A-$100, r B, w B+$100
  • commit
– Second
  • start transaction
  • r A, w A*1.05, r B, w B*1.05
  • commit

Transactions

Underlying DBMS makes sure xacts are only interleaved correctly (if at all).
Kinds of things to worry about

Reading uncommitted data
- “dirty read”
- write-read conflicts

Unrepeatable reads
- T2 changes the value of A while
- T1, in progress, has already read A

Kinds of things to worry about

Overwriting uncommitted data
- write-write conflicts
- complementary writes leave DB in bad state
Aside

select … for update
– required to say that you’re using information to compute a change to the database.
– otherwise xact may retry with stale values

Shortcuts

Creating IDs in Oracle
create sequence my_id_sequence start with 1;
insert into my_table values (my_id_sequence.nextval, 0);
select my_id_sequence.currval from dual;
**Shortcuts**

Creating IDs in MySQL
- autoincrement feature
- use null as primary key
- select last_insert_id() from any_table;

**Page Rank**

$$PR(A) = (1-d) + \ d \times (PR(t1)/C(t1) + \ldots + PR(tn)/C(tn))$$

`t1..tn` are the pages that link to A  
`C(ti)` is the number of links out of page ti  
`d` is a “fudge factor” (google’s is 0.85)
Metaphor

Pigeons randomly surfing the internet
– random start point
– click randomly on links
– restart after $1/(1-d)$ clicks
– what percentage of the time do they end up on each page?

Metaphor

Pages vote for their neighbors
– Like stockholders meeting
– You get votes according to your importance
– You can split your votes among any number of candidates
Tricky

Requires an iterative calculation

\[ PR(A) = 0.15 + 0.85 \times \frac{PR(B)}{C(B)} \]
\[ PR(B) = 0.15 + 0.85 \times \frac{PR(A)}{C(A)} \]

In the end

\[ PR(A) = PR(B) = 1. \]

Check by
  – pigeon metaphor
  – solution to equations
Other examples

![Diagram showing relationships between A, B, and C]

Rank

PR(A) ~ .77
PR(B) ~ 1.46
PR(C) ~ .77
Other examples

![Diagram showing nodes A, B, and C with cycles]

Rank

PR(A) ~ 1
PR(B) ~ 1.3
PR(C) ~ 0.7
Issues with real web sites

Reachability
Aliases
Spam

Google police

Require pages to be different
  – identify spam
Penalize links to spam