

Lecture 12: Transactions

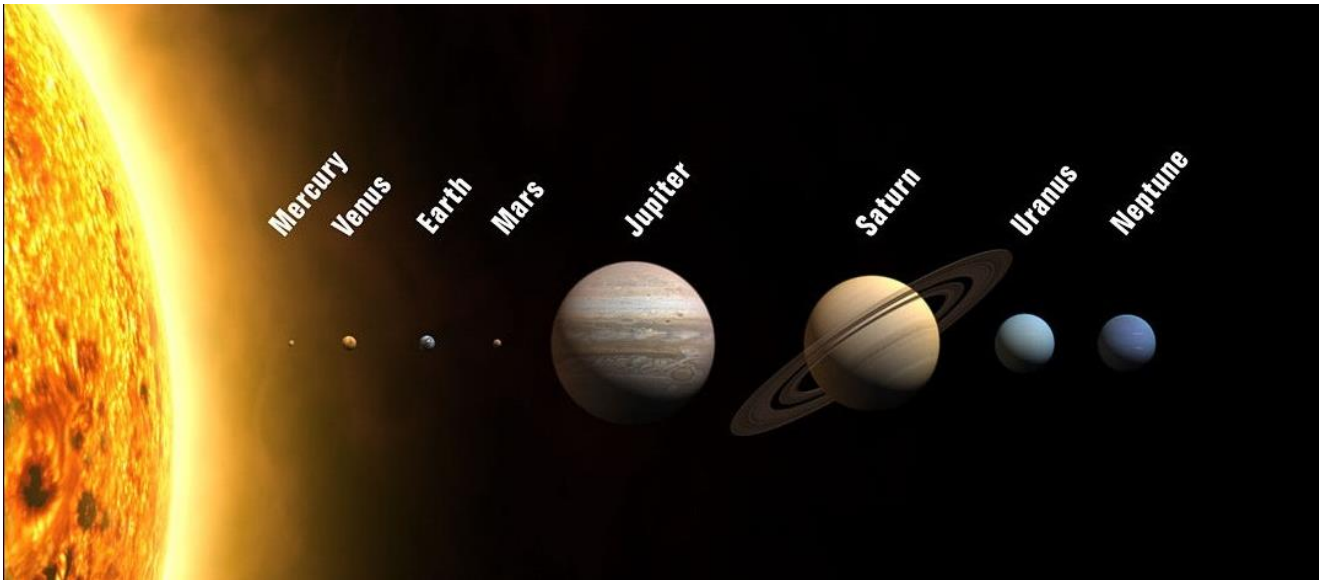
Monday, March 2, 2015

Agenda

- Review HW #3
- Introduce transactions
- Work on class project (Checkpoints #1 and #2)

Review Questions: SQL Design

- How would you represent in SQL:
 - 1-the solar system and its planets?
 - 2-the solar system and its star?
 - 3-space probes and their observed objects?



Source: http://en.wikipedia.org/wiki/Solar_System#mediaviewer/File:Planets2013.jpg

Transactions

- **Problem:** Applications need to run multiple concurrent updates on a database. These updates can interfere with one another. Moreover, the database machine can crash at any time.
- **Solution:** Applications instructions are bundled together into one logical unit called a *Transaction*

A World without Transactions

- Writes to files to ensure durability
- Rely on OS for scheduling and concurrency control
- So what can go wrong?
 - System crashes
 - Data anomalies (3 are famous)

What can go wrong #1: System crashes

Transfer \$500 Alice → Bob:

```
UPDATE Accounts  
SET balance= balance - 500  
WHERE name= 'Alice'
```



Crash!!!

```
UPDATE Accounts  
SET balance = balance + 500  
WHERE name= 'Bob'
```

What can go wrong #2: Lost updates

App instance 1:

```
UPDATE Usage  
SET song_count = song_count + 1  
WHERE cust_name= 'Alice'
```

App instance 2:

```
UPDATE Usage  
SET song_count = song_count + 1  
WHERE cust_name= 'Alice'
```

Two app instances play songs as Alice at around the same time. What happens?

What can go wrong #3: Inconsistent reads

Client 1: renames iPhone5 → iPhone6

```
UPDATE Inventory  
SET quantity = quantity + 10  
WHERE item = 'iPhone6'
```

```
UPDATE Inventory  
SET quantity = quantity - 10  
WHERE item = 'iPhone5'
```

Client 2: generates report

```
SELECT sum(quantity)  
FROM Inventory
```


What can go wrong #4: Dirty reads

Client 1: transfer \$100
Account 1 → Account 2

Account2.balance += 100

```
if Confirm():
    Account1.balance -=100
else:
    # rollback
    Account2.balance -= 100
    print "Transfer Cancelled!"
```

Client 2: transfer \$100
Account 2 → Account 3

Account3.balance += 100

```
if Confirm():
    Account2.balance -=100
else:
    # rollback
    Account3.balance -= 100
    print "Transfer Cancelled!"
```

Transactions: Definition

- **A transaction** = one or more read and/or write operations, which reflects a consistent transformation of state
 - It either happens or does not
- **Examples:**
 - Transfer money between accounts
 - Purchase a pair of movie tickets
 - Register for a class with multiple sections
- By using transactions, all previous anomalies go away

Transactions in Applications

[START TRANSACTION]

SQL statement₁

[SQL statement₂]

...

...

[SQL statement_n]

COMMIT; or ROLLBACK;

Revised Code using Transactions

Client 1: transfer \$100 Account 1 → Account 2

```
UPDATE Accounts  
SET balance = balance+100  
WHERE account_id = 2;
```

if Confirm():

```
    UPDATE Accounts  
    SET balance = balance-100  
    WHERE account_id = 1;  
    COMMIT;
```

else:

```
    # rollback  
    ROLLBACK;  
    print "Transfer Cancelled!"
```

Revised Code using Transactions

Client 2: transfer \$100 Account 2 → Account 3

```
UPDATE Accounts
SET balance = balance+100
WHERE account_id = 3;

if Confirm():
    UPDATE Accounts
    SET balance = balance-100
    WHERE account_id = 2;
    COMMIT;
else:
    # rollback
    ROLLBACK;
    print "Transfer Cancelled!"
```

ACID Properties

- Atomicity
 - State shows either all the effects of tx or none of them
- Consistency
 - Tx moves from a state where integrity holds to another state where integrity holds
- Isolation
 - Effect of interleaving txs is the same as txs running one after another
- Durable
 - Once a tx has committed, its effects remain in the database

ACID: Isolation

- A transaction executes concurrently with other transactions
- Isolation: the effect is as if each transaction executes in isolation from the others
- More on this next time

Checkpoint #1: Project Groups

Grp	Members
1	Matthew Egbom, Jewel Langevine, and Lerone Williams
2	Nathan Waters and Nur Syairah Mohd Ridzuan
3	Steve Franklin, Sadie Sublousky, and Tien-Yu Huang
4	Mills Hill
5	Alexander Crompton and Jacob Rachiele
6	Mitali Sathaye
7	Nikolaj Plagborg-Moller and Fabiana Latorre
8	Hannah Jane DeCiutiis, Kathryn McDermott, and Esther Schenau
9	Khang Pham and Don Pham
10	Alexia Mercado and Cyndia Munoz
11	Thomas Arthur Johnson and John Loftin
12	Ross Yudkin, Kurt Probe, and Andrew Chang-Gu
13	Tianxiang Zhang, Xiaolin Lu, and Happy Yilan Situ
14	Kaitlin Vanderlaan, Julia Haschke, and Sarah Luna
15	Brian Huang and Sergio Ignacio Mier
16	Jose Cortez, David Hernandez and Tara Woolheater
17	Kerri Grier and Chris Oballe

Checkpoint #2: Project Proposal

- **Due next class** (Wednesday, 03/04)
- Should be about **1 page** in length.
- Suggested content:
 - title and group members
 - short description of the project
 - list any interesting issues or unanswered questions
 - expected responsibilities/deliverables for each group member
 - important:** tools and datasets you are planning to use
- Submit proposal in class or by email

Next Class

- Continue transactions
- Work on project proposals