CSE 414 Database Systems
section 10: Final Review

Joseph Xu

6/6/2013

Final Exam

- The final exam is Monday, June 10 from 2:30-4:20
- Materials: You may bring your textbook plus one sheet of 8.5x11" paper with any hand-written notes you wish, plus your hand-written note sheet from the midterm. Otherwise closed book, no computers, etc

Final Topics

- Comprehensive - anything we've done this quarter, although the exam will be biased towards new material since the midterm

Review topics

1. XML
2. E/R Diagrams, Constraints, Conceptual Design
3. Views
4. Transactions
5. Parallel Data Processing

1. XML/Xpath/XQuery

- XML
  - Basic definitions: tags/elements/attributes/text, well-formed/valid XML document
  - DTDs
- Xpath = simple navigation
- Xquery = the SQL of XML

Example: Give the names of all the countries with population at least 10 million.

```
<result>
{ doc("mondial.xml")//country[population/text() >=
10000000]/name }
</result>
```

2. 1 E/R Diagrams, Constraints

- Entities, attributes
- Relationships:
  – Many-many, many-one, one-one.
  – Multi-way relationships
  – Modeling subclasses
- Constraints in E/R diagrams

2. 1 E/R Diagrams, Constraintsm

- Example (HW6, Q1):

Design an E/R diagram for geography that contains the following kinds of objects together with the listed attributes:

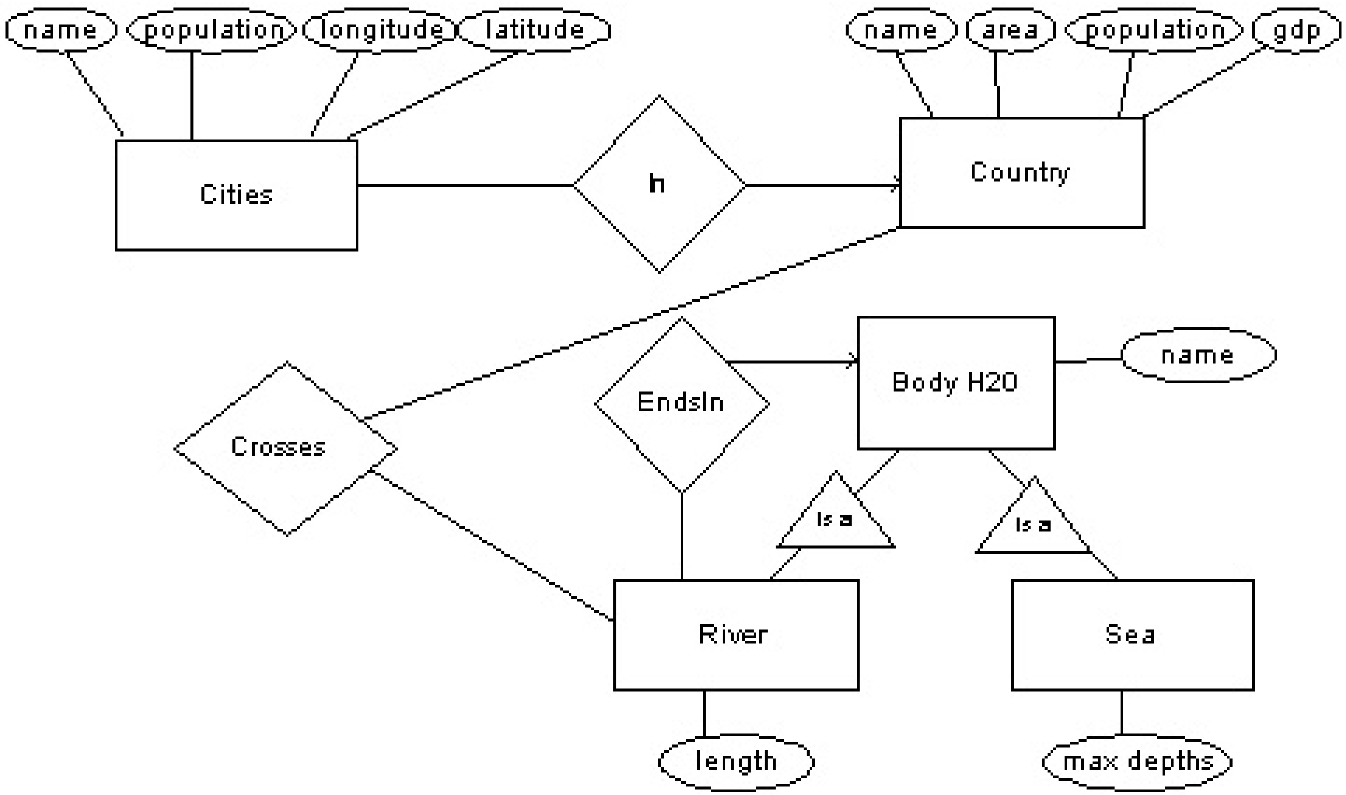
- countries: name, area, population, gdp ("gross domestic product")
- cities: name, population, longitude, latitude
- rivers: name, length
- seas: name, max depths

Model the following relationships between the geographical objects:

- each city belongs to exactly one country
- each river crosses one or several countries
- each river ends in a river or in a sea

2. 1 E/R Diagrams, Constraintsm

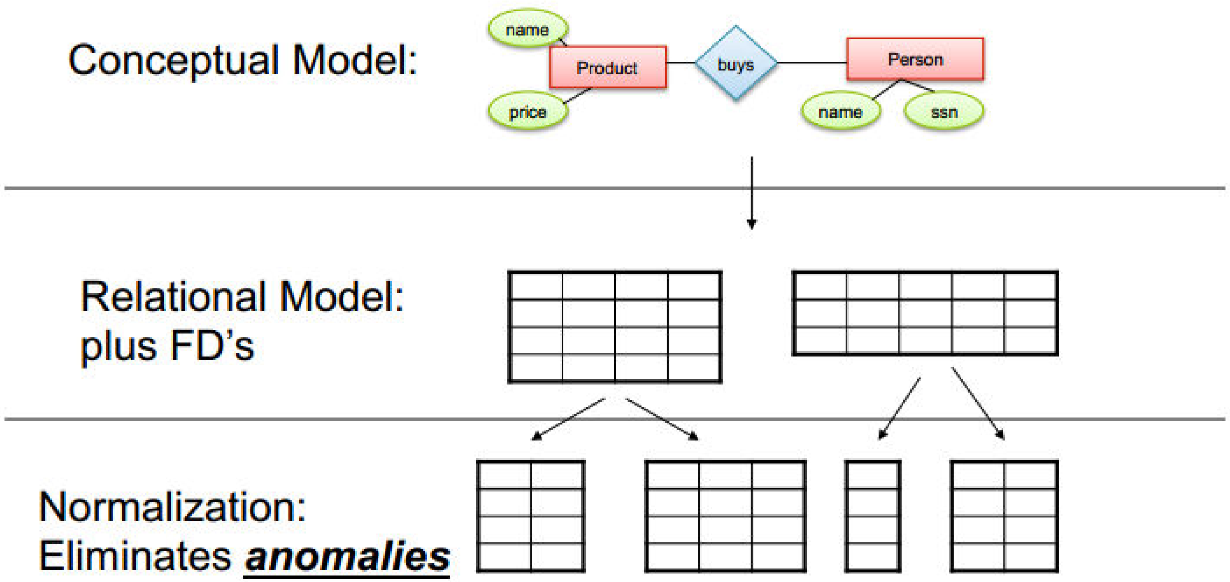
- Example (HW6, Q1) solution:



2. 2 Conceptual Design

Normal forms and functional dependencies:

- Anomalies(redundancy, update/deletion anomalies), functional dependencies, attribute closures, BCNF decomposition



2. 2 Conceptual Design

- Example:

Consider the following relational schema and set of functional dependencies. R(A,B,C,D,E,F,G) with functional dependencies:

A -->D

D --> C

F --> EG

DC --> BF

Decompose R into BCNF. Show your work for partial credit. Your answer should consist of a list of table names and attributes and an indication of the keys in each table (underlined attributes).

2. 2 Conceptual Design

R(<u>A</u>,B,C,D,E,F,G)

A -->D
D --> C
F --> EG
DC --> BF

Solution: Watch-out! The first FD does NOT violate BCNF so we need to pick another one to decompose. We try the second one:
Try $\{D\}^+ = \{B, C, D, E, F, G\}$. Decompose into R1(B, C, <u>D</u>, E, F, G) and R2(<u>A</u>,D).

R2 has two attributes, so it is necessarily in BCNF.

For R1, again not all FDs violate BCNF so we need to be careful.
Try $\{F\}^+ = \{E, F, G\}$. Decompose into R11(E, <u>F</u>, G) and R12(B, C, <u>D</u>, F).

Both R11 and R12 are in BCNF.

3. Views

- Definition: A view is derived data that keeps track of changes in the original data
- Virtual (Computed only on-demand – slow at runtime) **v.s.** materialized views (Pre-computed offline – fast at runtime)
- Application of virtual views:
  - Increased physical data independence (vertical/horizontal data partitioning)
  - Logical data indepence
  - Security

4. Transactions

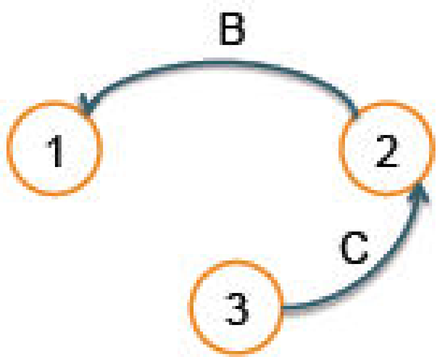
Transactions concepts

- Review ACID properties
- Definition of *serializability*
- Schedules, conflict-serializable and recoverable
- The four isolation levels in SQL
- Concurrency control using locks
  - SQLite and SQLServer examples
- Phantoms, dirty reads, and other problems
- Deadlocks
- Transactions in SQL

4. Transactions

Example 1: Consider the following transaction schedules. For each schedule, indicate if it is conflict-serializable or not.

r1(A), w1(A), r2(B), w2(B), r3(C), w3(C), r2(C), w2(C), r1(B), w1(B), c1, c2, c3

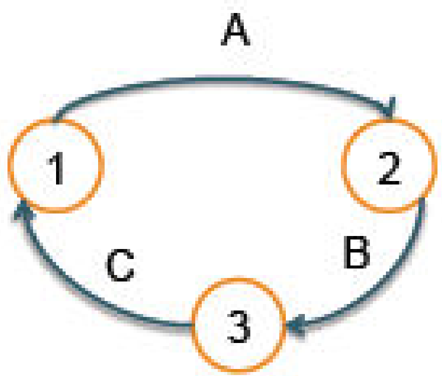


Serializable. The serialization order is: T3, T2, T1.

4. Transactions

Example 2: Consider the following transaction schedules. For each schedule, indicate if it is conflict-serializable or not.

r1(A),r2(B),r3(C),w1(C),w2(A),w3(B),c1,c2,c3



Not serializable, because of the cycle $T1 \rightarrow T2 \rightarrow T3 \rightarrow T1$.

4. Transactions

R:

A	B
1	10
2	20

- Example 3: Three transactions run concurrently on this database, issuing commands at the following time stamps:

Time	T1	T2	T3
1	begin transaction;		
2	select * from R;		
3		begin transaction;	
4		select * from R where $A = 2$;	
5	update R set $B = 30$ where $A = 2$;		
6		select * from R where $A = 2$;	
7	commit;		
8			begin transaction;
9			select * from R where $A = 2$;
10		commit;	
11			
12			
13			commit;

4. Transactions

R:

A	B
1	10
2	20

- Example 3: Three transactions run concurrently on this database, issuing commands at the following time stamps:

Find out what happened. You will consider two scenarios: when the database is managed with SQL Lite, and when the database is managed with SQL Server; in both cases the isolation level is set to SERIALIZABLE. For each command issued by a transaction, indicate one of the following outcome:

SUCCESS The request returns immediately, with success. In this case you should write SUCCESS, and also write the value that the transaction has read, if applicable.

ERROR The request returns immediately, with error. In this case you should write ERROR and indicate at which later time stamp the transaction needs to retry that command; if the command was a read, write down the value read by the transaction, when the command is reissued successfully.

WAIT The request causes the transaction to be placed on wait. In that case you should write WAIT, and also write at which later time stamp the transaction will be allowed to resume; if the command was a read, write down the value read by the transaction, when the command is resumed.

For example, you may answer as follows (not a real answer):

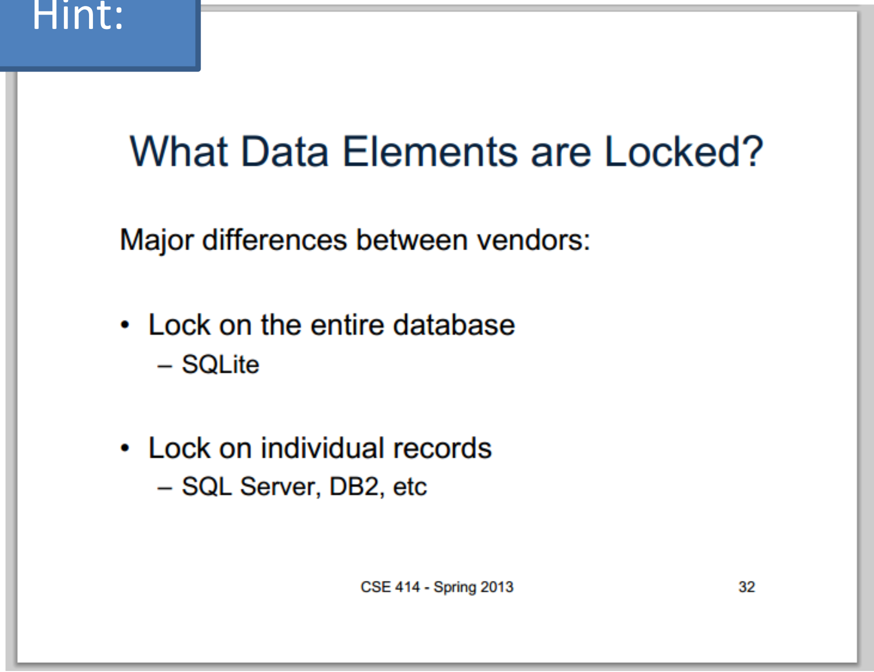
4. Transactions

R:

A	B
1	10
2	20

- Example 3: Three transactions run concurrently on this database, issuing commands at the following time stamps:

Hint:



4. Transactions

- Solution for SQL Lite

R:

A	B
1	10
2	20

Solution:

Time	T1	T2	T3
1	begin transaction;		
2	select * from R; – SUCCESS: values 10,20		
3		begin transaction;	
4		select * from R where $A = 2$; – SUCCESS: value 20	
5	update R set $B = 30$ where $A = 2$; – SUCCESS		
6		select * from R where $A = 2$; – SUCCESS: value 20	
7	commit; ERROR: retry on 11		
8			begin transaction; SUCCESS
9			select * from R where $A = 2$; ERROR: retry on 12
10		commit; – SUCCESS:	
11	– REISSUE: commit; – SUCCESS;		
12			– REISSUE: select * from R where $A = 2$; – SUCCESS: value 30
13			commit; – SUCCESS

4. Transactions

- Solution
  for SQL server

R:

A	B
1	10
2	20

Solution:

Time	T1	T2	T3
1	begin transaction;		
2	select * from R; – SUCCESS: values 10,20		
3		begin transaction;	
4		select * from R where $A = 2$; – SUCCESS: value 20	
5	update R set $B = 30$ where $A = 2$; – SUCCESS		
6		select * from R where $A = 2$; – WAIT: until 7	
7	commit; – SUCCESS	– SUCCESS: value 30	
8			begin transaction; – SUCCESS
9			select * from R where $A = 2$; – SUCCESS: value 30
10		commit; – SUCCESS:	
11			
12			
13			commit; – SUCCESS

5. Parallel Data Processing

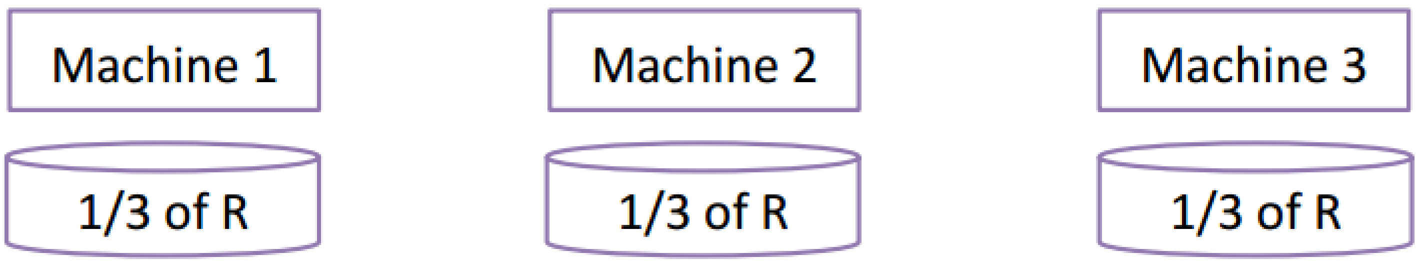
- Parallel databases
  - Speedup/scaleup
  - Shared memory, shared disk, shared nothing
  - How to implement simple algorithms: group-by, join
- MapReduce
- Pig system and Pig Latin language

5. Parallel Data Processing

- Example

Consider a relation R(a,b) that is horizontally partitioned across N = 3 machines as shown in the diagram below. Each machine locally stores approximately 1/N of the tuples in R. The tuples are randomly organized across machines (i.e., R is block partitioned across machines). Show a relational algebra plan for this query and how it will be executed across the N = 3 machines. Pick an efficient plan that leverages the parallelism as much as possible. Include operators that need to re-shuffle data and add a note explaining how these operators will re-shuffle that data.

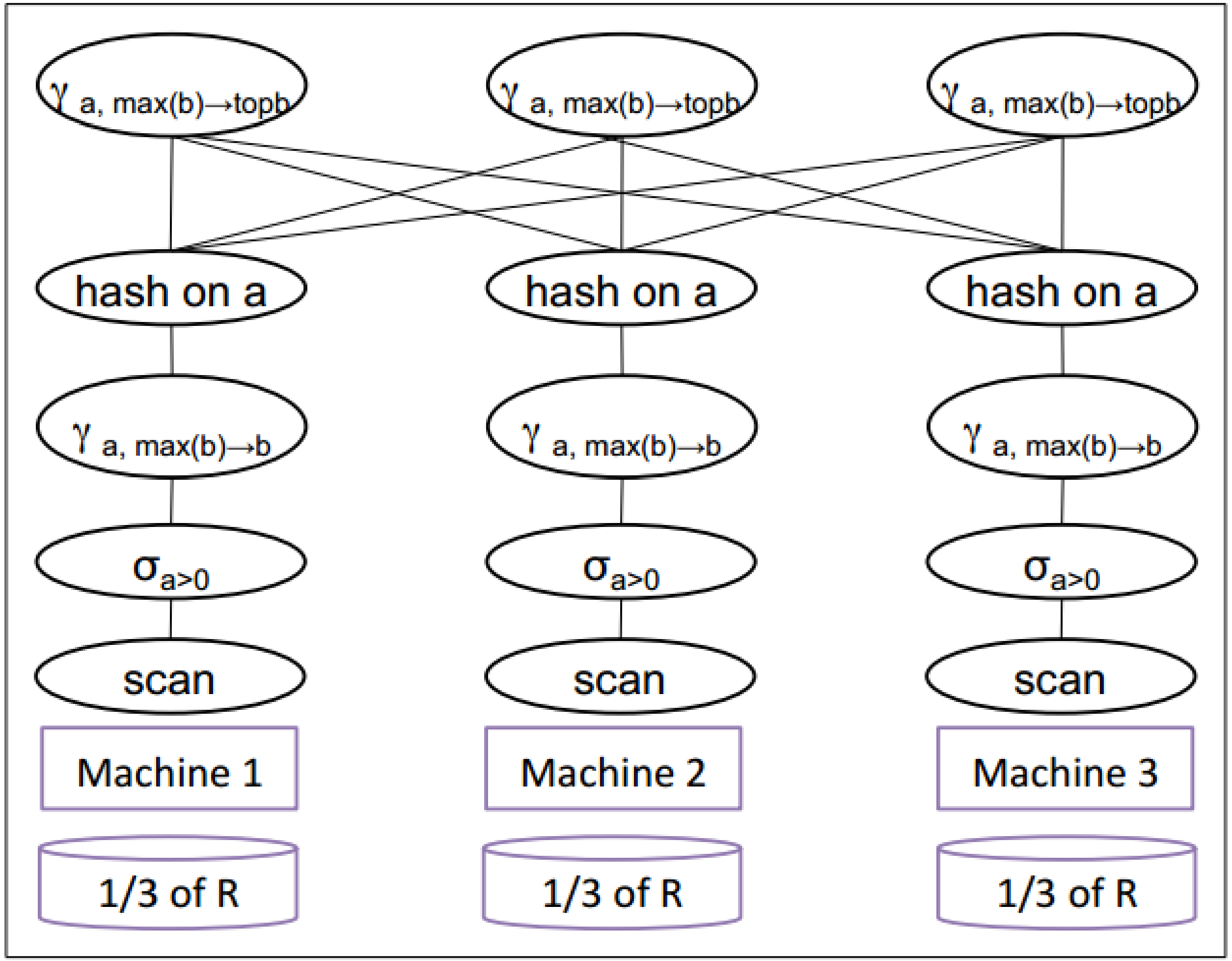
```
SELECT a, max(b) as topb
FROM R
WHERE a > 0
GROUP BY a
```



5. Parallel Data Processing

Solution:

SELECT a, max(b) as topb
FROM R
WHERE a > 0
GROUP BY a



Suggestions

- Go over lecture notes
- Check the solutions of the past homework
- Try example problems from the past final exams