Final Review #1

Wednesday, April 29, 2015

Agenda

- Next two classes: Final Review
- Today: Makeup Quiz

- Reminder: Course evaluations

The Final: Logistics

- Date: Wednesday, May 6, 2015
- Time: 5:00 - 6:30
- Where: In class
- Open book exam :)
- No computers :(

The Final: Content

5 Problems with Sub-Problems:

1. Queries
2. Data modeling
3. Transactions
4. Query processing
5. NoSQL systems

General Advice

- Some problems will require thinking
  - Use judgment
- Problem difficulty may be uneven:
  - do the easy ones first

Problem 1: Queries

- SQL
- MongoDB query language

SQL

- select-from-where
- order by and renamings
- joins
- group-by and having
- aggregations
- views
- insert, update, delete

Interesting question: *does one query return a subset of another?*

SQL Practice Problem #1

Find all students in the database:

```
SELECT *
FROM students
WHERE gpa >= 3.5 OR gpa < 3.5
```

Question: *what's wrong?*

SQL Practice Problem #2

Products(product_id, product_name)
Sales(sale_id, product_id, quarter, year, quantity, price)

Find total sales by product for Q1 of this year:

```
SELECT p.product_name, SUM(s.quantity*s.price) AS total_sales
FROM products p, sales s
WHERE p.product_id = s.product_id
AND s.quarter = 1
AND s.year = 2015
GROUP BY product_name
```

Question: *what's wrong?*

SQL Practice Problem #3

Old Schema:

ProductRequests(customer_id, customer_name, product_id, request_date)

New Schema:

Products(id, name, color, weight, number_available)

Customers(id, name, age, address, city)

Requests(customer_id, product_id, date)

Question: *create ProductRequests view over new schema*

MongoDB

- find and findOne
- comparison operators: $lt, $lte, $gt, and $gte
- logical operators: $in, $or, $and
- arrays: $all
- embedded documents: dot notation
- insert, update (with upsert), remove

An interesting question: *write query to transform a JSON document*

Problem 2: Data Modeling

- E/R diagrams
- Normal forms
- JSON and semi-structured data

E/R Diagrams

- Entities, attributes
- Relationships
- Inheritance
- Translation to relations
- SQL DDL:
  - Creating tables
  - Constraints

An interesting question: *translate an inheritance graph to relations*

Normal Forms

- Data anomalies
- Functional dependencies
- 1NF, 2NF and 3NF definitions
- Checking if a relation is in 3NF
- Decomposing into 3NF

An interesting question: *does a FD hold on a table?*

Revisiting Normal Forms

Recall: Functional Dependencies

Definition:

If two tuples agree on the attributes

$A_1, A_2, \ldots, A_n$

then they must also agree on the attributes

$B_1, B_2, \ldots, B_n$

Formally:

$A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_n$

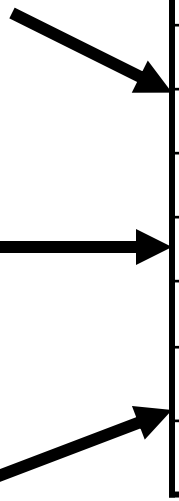
Unnormalized to 1NF

Rule: A database schema is in 1NF *iff* all attributes have scalar values

Students

Student	Semester	GPA	Courses
Alice	Spring15	3.9	Math DB Alg
Bob	Spring15	3.7	DB Alg
Carol	Spring15	3.5	Math Alg

unnormalized



Students'

Student	Semester	GPA	Course
Alice	Spring15	3.9	Math
Alice	Spring15	3.9	DB
Alice	Spring15	3.9	Alg
Bob	Spring15	3.7	DB
Bob	Spring15	3.7	Alg
Carol	Spring15	3.5	Math
Carol	Spring15	3.5	Alg

1NF

1NF to 2NF

Rule: A database schema is in 2NF *iff* it is in 1NF and there are no partial FDs on the primary key (i.e. all non-key attributes must be dependent on the entire PK)

Students

Student	Semester	Course	GPA
Alice	Spring15	Math	3.9
Alice	Spring15	DB	3.9
Alice	Spring15	Alg	3.9
Bob	Spring15	DB	3.7
Bob	Spring15	Alg	3.7
Carol	Spring15	Math	3.5
Carol	Spring15	Alg	3.5

1NF

Enrolls

Student	Course	Semester
Alice	Math	Spring15
Alice	DB	Spring15
Alice	Alg	Spring15
Bob	DB	Spring15
Bob	Alg	Spring15
Carol	Math	Spring15
Carol	Alg	Spring15

2NF

GPA

Student	Semester	GPA
Alice	Spring15	3.9
Bob	Spring15	3.7
Carol	Spring15	3.5

2NF

Assumptions:

1. Student, Semester → GPA
2. GPA is not functionally determined by course

2NF to 3NF

Rule: A database schema is in 3NF *iff* it is in 2NF and there are no transitive dependencies

Students

EID	Name	Major	College
100	Alice	Math	Natural Sciences
200	Bob	CS	Natural Sciences
300	Carol	Finance	Business

2NF

Assumptions:

EID → Name, Major

Major → College

By transitivity, EID → College

Students'

EID	Name	Major
100	Alice	Math
200	Bob	CS
300	Carol	Finance

3NF

Majors

Major	College
Math	Natural Sciences
CS	Natural Sciences
Finance	Business

3NF

JSON

- JSON syntax
- From relations to JSON
- From JSON to relations

An interesting question: *N/A*

Problem 3: Transactions

- Data inconsistencies
- Concurrency control
- Distributed transaction processing

Data inconsistencies

- Dirty reads
- Non-repeatable reads
- Phantom reads

An interesting question: *find inconsistencies in a schedule*

Concurrency Control

- Serializability
- Repeatable Read
- Read Committed
- Read Uncommitted

An interesting question: *explain what happens in a schedule*

Distributed Transactions

- 2 phase commit
- “Eventual” consistency

An interesting question: *identify conflicting vector clocks*

Problem 4: Query Processing

- Query execution without indexes
- Query execution with indexes
- Types of indexes:
  - clustered index
  - unclustered index
- B^+ trees

An interesting question: *select index based on SQL queries*

Problem 5: NoSQL Systems

- Data systems landscape
- MapReduce
- MongoDB
- Replication and “sharding”

An interesting question: *convert a SQL query to MapReduce*

COMMIT

(The End)

Make-up Quiz

Q1: What is the difference between horizontal and vertical partitioning?

Q2: When would you use a virtual view as opposed to a materialized view and why?

Q3: List out what ACID stands for and explain two of them

Q4: List one data model that is used by NoSQL systems