CS 133: Databases

Fall 2019
Lec 26 – 12/12
Data Analytics

Prof. Beth Trushkowsky

Warm-up Exercise

(See exercise sheet. You can start before class.)

Number of reservations for each sid,bid pair

Goals for Today

- Understand how Analytics processing (OLAP) is different than Transactional processing (OLTP)
- Reason about how data is organized and queried in a data warehouse
- Discuss current trends in Big Data processing

Data Analytics and Decision Support

- **Idea**: current and historical data to identify useful patterns and support business strategies
- Complex, interactive, exploratory analysis of data
  - Large datasets
  - Data integrated from across all parts of an enterprise
  - Data is fairly static
- **OLAP**: on-line analytical processing
  - In contrast to **OLTP** (on-line transactional processing)

OLAP vs. OLTP

- **OLTP**
  - Update-heavy
  - Short, simple transactions
  - Goal: transaction throughput
- **OLAP**
  - Mostly reads
  - Longer, complex queries for analysis and decision-making
  - Goal: fast queries

Data Integration

- Data may reside in many distributed, heterogeneous OLTP sources
  - Sales, inventory, customer, …
  - NC branch, NY branch, CA branch, …
- Need to support OLAP over integrated view of the data

Tradeoffs?

- Possible approaches to integration
  - *Eager*: integrate in advance and store the integrated data in a data warehouse

    Need ETL
  - *Lazy*: integrate on demand; process queries over distributed sources—the approach of mediated or federated systems

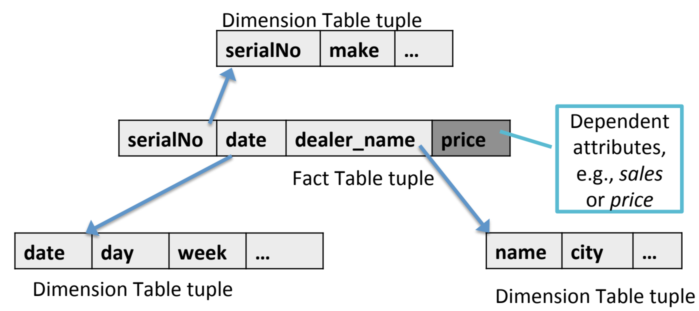
Example: Car Sales Schema

```
Cars(serialNo, make, model, color)
Dealers(name, city, state, phone)
Date(date, day, week, quarter, month, year)

Sales(serialNo, date, dealer, price)
```

Star Schema in Relational OLAP (ROLAP) System

- Fact table **BCNF**; dimension tables possibly **denormalized**
  - Dimension tables are small; updates/inserts/deletes are rare…. anomalies less important than performance
- Star Schema



A Multidimensional View

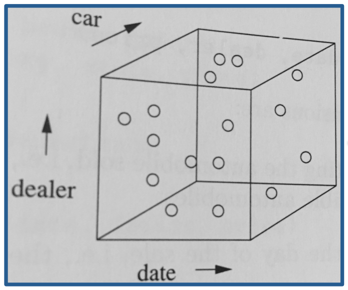
- Example car sales schema:

```
Cars(serialNo, model, color)
Dealers(name, city, state, phone)
Date(date, day, week, month, year)

Sales(serialNo, date, dealer, price)
```

Conceptual dimension table

Fact table



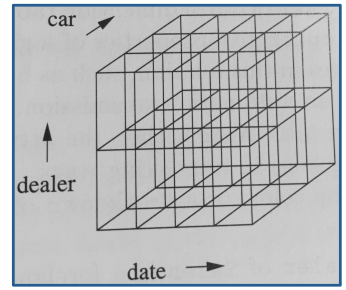
“Raw” data cube

Dicing the Cube

Think SQL “group by”

- Can think of partitioning the raw data cube along each dimension at some level of granularity

- A choice of partition for each dimension “dices” the cube

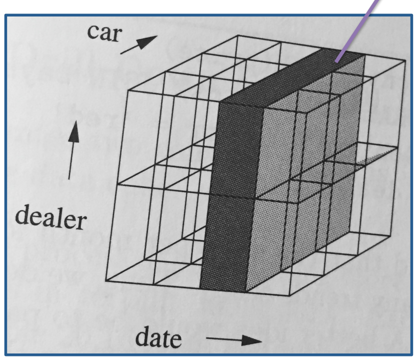


Slicing the Cube

- Idea: want info about a fixed *slice* of the data

- In general, in SQL:
  - **Dice**: `GROUP BY`
  - **Slice**: `WHERE`

Sales data for one date slice (e.g., one year)



Example: Data Analysis

```
Cars(serialNo, make, model, color)
Dealers (name, city, state, phone)
Days(date, day, week, quarter, month, year)

Sales(serialNo, date, dealer_name, price)
```

- Suppose *Mazda3* model is not selling as well as anticipated

- ***Query***: which colors not doing well?

```
SELECT color, SUM(price)
FROM Sales NATURAL JOIN Cars
WHERE model = "Mazda3"
GROUP BY color;
```

Exercise 2 (a-c)

(a)
```
SELECT color, month, SUM(price)
FROM Sales, Cars, Days
WHERE Sales.serialNo = Cars.serialNo
      AND Sales.date=Days.date
      AND model = “Mazda3”
GROUP BY color, month;
```

(b)
```
SELECT dealer_name, month, SUM(price)
FROM Sales, Cars, Days
WHERE Sales.serialNo = Cars.serialNo
      AND Sales.date=Days.date
      AND model = “Mazda3”
      AND color = “red”
GROUP BY month, dealer_name;
```

(c)
```
SELECT dealer_name, year, SUM(price)
FROM Sales, Cars, Days
WHERE Sales.serialNo = Cars.serialNo
      AND Sales.date=Days.date
      AND model = “Mazda3”
      AND color = “red”
      AND (year = 2016 OR year = 2017)
GROUP BY year, dealer_name;
```

Analysis: Cross-tabulation

Also called “pivoting”

- Sales from each dealer by car color
  - View popularized by spreadsheet applications

Car color

Dealer		Red	White	Blue	*total*
	Alice	90K	30K	120K	240K
	Bob	100K	10K	40K	150K
	total	190K	40K	160K	390K

How many SQL queries to generate the data in this table?

OLAP Queries

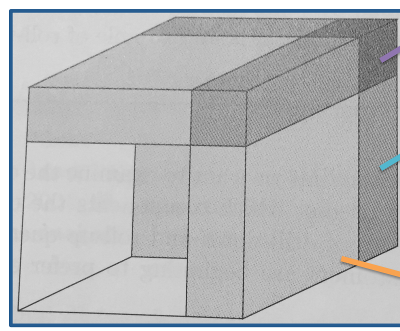
- A common operation is to **aggregate a measure** over one or more dimensions.
- **Roll-up**: Aggregating at coarser granularity, e.g., higher level in dimension hierarchy.
- **Drill-down**: The inverse of roll-up

Specialized *MOLAP* and *ROLAP* systems may store pre-aggregated data (materialized views)

The Data CUBE

Multidimensional OLAP (MOLAP)

- A CUBE relation: generalization of the cross-tabulation



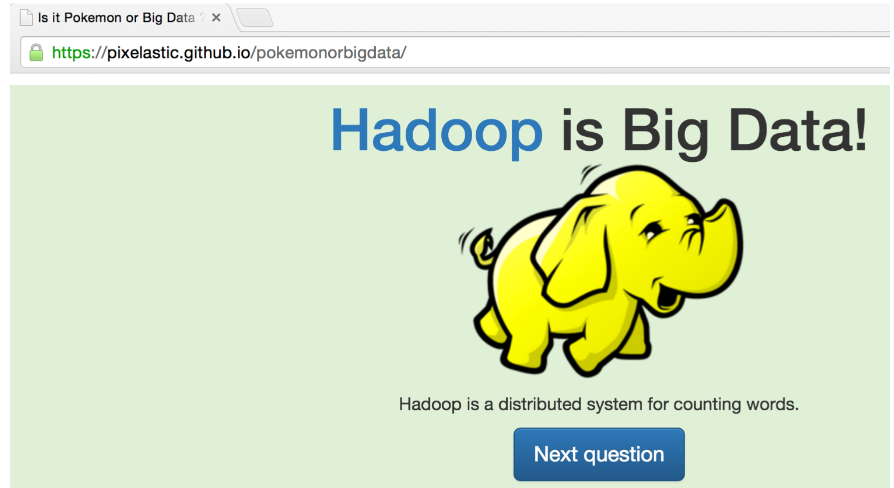
Analyzing Big Data: Current Trends

- Motivation
  - Expensive ROLAP and MOLAP systems not for everyone
  - Desire to analyze semi-structured or unstructured data
- Big Data rampant!
  - E.g., data sets generated by some of the applications backed by NoSQL systems
  - Sensor data, tweets, etc.
- Trend: many people using **MapReduce**/**Hadoop** for Big Data Analysis
  - Scalability and commodity hardware

Open-source version of Google's MapReduce

Pokémon or Big Data?

https://pixelastic.github.io/pokemonorbigdata/



Final Exam: Logistics

- Take-home exam
- Due to my office (Olin 1267) at or before Wednesday, December 18th, 5:15pm
- Two 8.5x11, double-sided note sheets
  - You can use your note sheet from the midterm as one of the two
  - No other resources
- 3-hour timed exam

Possible Topics on Final

- Cumulative-ish
  - Topics we covered earlier still relevant (e.g., hash & tree indexes, estimating cost in I/Os)
  - Won't focus on nitty gritty from before midterm (e.g., linear vs extendible hashing)
- Query Optimization
- Transactions and ACID
- Database design
- ORDBMS, Distributed DBMS and NoSQL, OLAP (high-level)
- General themes
  - Reasoning about cost and tradeoffs
  - Consistency and correctness with concurrent access and failures

Query Optimization

- Query
  → relational algebra tree
  → logical plan
  → physical plan
- Unit of optimization: query block
- Logical plan
  - Relational algebra equivalences
  - Outer vs. inner relation in joins
  - Query plan tree shape: bushy, linear, deep

Query Optimization

- Choosing physical plan
  - Enumerate plan space
    - Join permutations and orders
    - System R choices
  - Estimate cost of plan
  - Picking cheapest
    - Dynamic programming algorithm (idea)
    - Interesting orders
- Cost estimation
  - Operator algorithm cost
    - Estimating cost of different join algorithms
  - Operator result size estimation
    - Selectivity/Reduction Factor, statistics, histograms
    - Using indexes

ACID Transactions

- Transactions, how to achieve ACID
- Isolation (I)
  - Schedules: serializable, conflict-serializable, etc.
  - Anomalies from interleaved actions, conflicting actions
  - Locking, lock granularity and compatibility, deadlock detection and prevention
  - 2PL vs Strict 2PL, cascading aborts
  - Optimistic concurrency control, backwards validation algorithm
- Recovery (A and D)
  - Steal vs. force and implications on UNDO/REDO
  - Write-Ahead-Logging
  - ARIES recovery algorithm

Database design

- E/R modeling (general idea)
  - Entities, relationships, weak entities
  - Capturing key and participation constraints
- Functional dependencies
  - Attribute closure, Armstrong's axioms
  - Determining candidate keys
  - Role in detecting data redundancy
- Schema refinement
  - Normalization
  - BCNF normalization process
- Capturing integrity constraints in relational schema
- General motivation and ideas from ORDBMS

Special Topics

- Distributed DBMS
  - Goals of data partitioning and data replication
    - Types of partitioning: range vs hash
  - Replication
    - Synchronous vs asynchronous
    - Strong vs. eventual/weak consistency
  - Challenges with distributed xacts (generally)
- NoSQL
  - CAP theorem
  - Query restrictions for performance (generally)
- Analytics
  - Generally what OLAP is, vs. OLTP, and what kinds of queries run