Review: relational basics
- Relational model/algebra → physical data independence
- Entity-relationship design
- Design theory (FD’s, MVD’s, 3NF, BCNF, 4NF) → help eliminate redundancy
- SQL
  - NULL and three-value logic: nifty feature, big mess
  - Bug versus set semantics: careful about equivalences
  - SFW (or SPJ): queries, subqueries, grouping and aggregation
  - Modifications
  - Constraints → the more you know the better you can do
  - Triggers (ECA) → “active” data
  - Views → logical data independence
- SQL: to reintroduce redundancy to improve query performance
- Transactions and isolation levels

Review: physical data organization
- Storage hierarchy (DC vs. Pluto) → count I/O’s
- Disk geometry: three components of access cost; random vs. sequential I/O
- Data layout
  - Record layout (handling variable-length fields, NULL’s)
  - Block layout (NSM, PAX) → inter-/intra-record locality
- Access paths
  - Primary versus secondary indexes
  - Tree-based indexes: ISAM, B+-tree
  - Text indexes: inverted lists, signature files, tries
  → Again, reintroduce redundancy to improve performance
  → Fundamental trade-off: query versus update cost

Review: XML
- Data model: well-formed vs. valid (DTD ≈ schema)
- Query languages
  - XPath: (branching) path expressions (with conditions)
  - XQuery: FLWOR, subqueries in return (restructuring), quantified expressions, aggregation, ordering
  - XSLT: structural recursion with templates
- Programming: SAX (one pass) vs. DOM (in memory)
- Relational vs. XML
  - Tables vs. hierarchies (or graphs in general)
  - Storing XML as relations
    - Schema-oblivious: node/edge based, interval based, path based, etc.
    - Schema-aware
      → Joins vs. path traversals

Review: query processing, optimization
- Processing
  - Scan-based algorithms
  - Sort- and hash-based algorithms (and their duality)
  - Index-based algorithms
  - Pipelined execution with iterators
- Optimization (or “goodification”?)
  - Heuristics: push selections down; smaller joins first
    → Reduce the size of intermediate results
  - Cost-based
    - Query rewrite: merge blocks to get a bigger search space
    - Cost estimation: result size estimation; use statistics
    - Search algorithm: dynamic programming (+ interesting orders)

Review: transaction processing
- ACID properties
- Concurrency control
  - Serial and conflict-serializable schedules
  - Locking-based: 2PL, strict 2PL
- Recovery with logging
  - Steal: requires undo logging
  - No force: requires redo logging
  - WAL (log holds the truth)
  - Fuzzy checkpointing
Review: other topics

- Web searches
  - Indexing
    - Term-based: term/document matrix $\rightarrow$ inverted lists vs. signature files
    - Subsequence-based: various tries
  - Ranking
    - Content-based: TF (term frequency); IDF (inverse document frequency)
    - Link-structure-based: backlink count; PageRank

- Data warehousing
  - OLAP vs. OLTP: different workload $\rightarrow$ different degree of redundancy
  - Data warehouse: eagerly integrate data from operational sources and store a redundant copy to support OLAP

- Data mining: frequent itemset mining using apriori property for pruning