Final Exam Review Sheet
Database Systems - Spring '20
CS386D
Professor Daniel Miranker

Exam Date and Time: Thursday, May 14, 10:00am – 12:30pm (2 ½ hours), Exeuction will be identical to Midterm 2, closed book, calculators ok, phones as calculators, not ok, honor system in place. We will skip the upload a page before the exam part. The goal of that was not achieved. (papers were still submitted by email).

This review sheet is intended only as a study guide concerning the breadth of the exam. You are expected to know all the terminology presented as covered in class, the texts and the required reading. To be clear, individual terms and topics in this document are indicative of the breadth, *not* a comprehensive syllabus for the exam.

Reading: Per the lecture schedule posted on Box. (to be brought up to date shortly)

Topics:

1. Basic Relational Database Concepts
    a. Schema(s)
    b. Associative Access
    c. Keys
        i. Candidate key
        ii. Primary key
        iii. Foreign key
        iv. Search/index key
    d. Basic Organization
        i) Data/index files
        ii) transaction log
        iii) SQL Engine
        iv) Role of RDBMS in a three-tier architecture

2. Constraints & Triggers
    a. Referential Integrity
    b. Attribute vs tuple level constraints
    c. Check constraints
    d. Triggers
        i. applications
        ii. integration with and implications of the transaction manager

3. Views
    a. Syntax/Semantics
    b. Use as a subroutine mechanism (nested queries)
    c. Use per logical restructing of a database, (re: external schema).
    d. Maintaining a materialized view

4. Disks and Data
    a. Physical properties of disk drives
    b. Two phase multiway merge sort
    c. ~~RAID~~

5. Indexing
    a. Methods (access paths)
        i. B+-trees

ii. Bit-vector index methods
iii. Bloom filters
b. Secondary Indexes
i. Applicability
ii. Clustering
iii. Effectiveness
- Measures of effectiveness
- Parameters, B(R), T(R), V(R, attr)

6. Query Systems
a. Gross Structure
i. Parsing
ii. Logical Plan
iii. Physical Plan
b. Physical Operators
i. Access Paths
1. table_scan
2. index_scan
c. Join Operators
i. Nested loops
ii. Merge join
iii. Hash-join
iv. Hybrid-hash join
d. Estimated Query Cost
i. Estimating the cost of each operator
ii. Adorning a plan tree.
iii. Estimating the cost of a plan, most notably, impact of relational select and join operators
e. Measures wrt [text] I/O model and the number of rows
f. Architecture and organization (scope of the System Catalog)
g. Optimization
i. Role of axioms/identifies of the relational algebra
ii. Greedy rules e.g. pushing selects
iii. Dynamic programming method of optimizing join orders
iv. Use of a query graph

7. Parallelism in Databases
a. Machine organizations. e.g. shared-nothing
b. Data Partitioning methods
i. round-robin
ii. hashing
iii. range based
c. Application and use of hash partitioning in Cloud databases
i. Data skew
ii. Grace join
d. Use of semi-join reduction and its appoximation using Bloom Filters

8. On Graphs and Database Management
a. Representation of graphs in a relational database
i. Directed graphs are easily represented
ii. Logical models
iii. Use of keys to represent graph edges
b. Recursion in SQL
// there was no homework, so just that
i. it is in the standard
ii. keyword WITH, though common table expression, CTE, is part of that and

dominates “WITH” in common usage.

- iii. Semantics come from Stratified Datalog

- 9. Datalog
    - a. Syntax (all the names of the parts of a rule)
        - i. Safety
        - ii. Negation, implemented as set difference and (semantics) assumes the closed world assumption
    - b. Semantics
        - i. IDB vs. EDB
        - ii. Fixed-point
        - iii. Minimum Model
    - c. Non-Recursive Evaluation
        - i. Dependency graph
        - ii. Evaluate rules per a topological sort of the dependency graph
        - iii. Equivalence with core relational algebra (thus relational algebra may appear on the exam, here as well as as needed for logical query plans).
    - d. Recursion and the Evaluation of Recursive Datalog
        - i. Use for the logical representation of graph queries.
        - ii. Naïve and Semi-naïve evaluation
        - iii. Stratification
    - ~~e. Optimization~~
        - ~~i. Magic-sets~~

- 10. Transactions
    - a. ACID properties (strong transactions)
        - i. A…
        - ii. C…
        - iii. I…
        - iv. D….
    - b. Log-based recovery
        - i. Hardware organization
            - 1. log
            - 2. stable store
        - ~~ii. Redo logging and recovery~~
        - ~~iii. Undo logging and recovery~~
        - iv. Undo/redo logging and recovery
        - v. Role of commit
        - vi. Non-quiescent checkpoint wrt undo/redo logging
        - ~~vii. Check points, backups and the loss of data storage~~
    - c. Conconcurrency Control
        - i. Schedule
            - 1. Serial Schedule
            - 2. Conflict/Bernstein Conditions
        - ii. Serializability
            - 1. Precedence graph
            - 2. Serializability Theorem
        - iii. Locking
            - 1. Two-phase locking
            - ~~2. Escalading/upgrading locks~~ // nothing from here to the end will be on the exam, but you should have the big picture
            - ~~3. Shared locks~~
            - ~~4. Granularity (basic ideas, not full implementation and correctness)~~
            - ~~5. Waits-for-graph deadlock~~