Database System Architectures (INFO-H-417)

Mahmoud SAKR

Written Exam Topics

Translating SQL into Relational Algebra

- Express SQL queries including selection, projection, join, aggregation, CTE, and subqueries.
- Describe the concepts of declarative and procedural query languages
- Describe the different relational algebra (and extended RA) operators
- Translate a given SQL statement into an equivalent RA expression.
- Differentiate the set and bag semantics in RA, and describe how the query results may differ applying one semantic or the other.
- Illustrate the transformation of a sub-query into a join then into an RA expression.

System R & Query Optimizations

- Describe the architecture components of system R.
- In light of the System R paper, describe the concepts of: catalog, tuple identifier, image, clustering image, view, cost-based query optimization, and access path cursor
- Discuss how far the concepts in the previous point are implemented in PostgreSQL
- Given an RA expression, apply equivalence rules to transform it into other equivalent RA expressions
- Assess whether or not a given equivalence rules is valid, under set or bag semantics, and whether there are constraints for its validity
- Illustrate the computational challenge of cost based query optimization, and ways to reduce the cost (e.g., the join ordering problem, heuristic optimization)
- Describe the join order optimization problem, and the solution approach of left deep join trees.

Statistics for Cost Estimation

- Describe the role of statistics in cost-based query optimization Index present
  Number of records in table
  Number of distinct values
- Discuss the use of histogram for attribute statistics selectivity
- Given relation statistics estimate the size of a selection and join queries

Indexing

- Explain the use of indexes in query processing
- Explain the concepts of sequential file, dense index, sparse index, 1st level index, and 2nd level index, secondary index
- Illustrate the insertion/deletion strategies in conventional indexes, also for the case of duplicate keys
- Illustrate the benefits of buckets in secondary indexes
- Illustrate the Btree (also called B+tree and B-tree) index, its parameters, and how insertions and deletions are performed
- Illustrate the properties of the Btree that allows us to answer inequality (<, <=, >, >=) or range searches (between) efficiently
- Illustrate the insertion, and search in Btree. (*Btree Deletion is not included in this written exam*)

Physical Query Plans

- Describe the different physical algorithms for joins: nested loop, merge, with index, and hash joins.
- Given the necessary statistics, and available memory estimate the cost of each of the four join algorithms
- Illustrate the memory requirement of the merge and hash join algorithms

Extending database systems

- Explain the architectural components that make PostgreSQL extensible:
    - What is the role of the catalog ?
    - How is PostgreSQL able to process user types (storage, input, output, statistics, etc)
    - How is PostgreSQL able to compute functions over user types
    - How is PostgreSQL able to use its generalized index structures over user types
    - What is the role of extensions in PostgreSQL
- Describe (in English not in coding) the steps/tasks that one would need in order to create a PostgreSQL extension similar to the complex numbers extension that you created in the exercise session

Failure Recovery

- Describe the concept of a database transaction
- Illustrate undo logging, and the associated crash recovery
- Illustrate redo logging, and the associated crash recovery
- Describe the concept and benefit of checkpoints
- Illustrate undo/redo logging, and the associated crash recovery
- *The part starting Non-quiesce checkpoint till end of lecture is not included in the written exam*

Concurrency Control

- Explain how concurrent transactions can lead to violation of consistency
- Describe the concepts of serializable schedule and conflict-serializable schedules
- Illustrate the use of precedence graph for checking conflict-serializability
- Verify whether a schedule is well-formed, legal, and implements 2PL
- Illustrate the concurrency issues that can happen when the three rules are not implemented
- Motivate the need for increment locks, update locks, shared locks, and multi-granular locks
  No need to memorize the compatibility matrices. They will be given if needed
- Run a given schedule, and trace the execution steps

Distributed databases

- Discuss what are the benefits of distributing a database
- Describe the concepts of: distributed table, replicated table, distribution key, range distribution, hash distribution, spatial distribution, re-balancing, reference tables.
- Illustrate the importance of co-location
- Given a query, illustrate a strategy to distribute data and the corresponding distributed query plan
- Illustrate methods for computing non co-located joins in distributed databases, and reflect on their cost
- *Distributed transactions and replication are not included of this written exam*

With my best wishes.