Final review

INF 551

Wensheng Wu

SQL

- Select…from…where…group by…having…order by…limit…offset…
- MySQL does not support intersect and except
- Subqueries:
  - =, in, >= all/any
  - (not) exists
- Aggregation
  - sum, min/max, count, avg
  - group by
  - having

SQL

- Join
  - Theta-join
  - Natural-join
  - Outer join (left and right, no full outer in MySQL)

- DML vs DDL
  - Alter table …

Constraints & views

- PK, FKs
  - When & how to enforce FKs (set null, cascade)
- Virtual vs materialized views
  - View unfolding
  - Reusing materialized views

Data representation & external sorting

- Fixed & variable-length record
- Packing records into a block
- External sorting
  - Two-way vs multi-way
  - Sorting phase & multiple merging phases: input, output, sizes
  - Costs

Indexing

- Clustered vs non-clustered index
- MySQL creates index automatically for …
- Composite index

- Search in B+tree & costs
- Insertion & deletion:
  - strategy to maintain the balance (splitting, rotating, and merging)
  - Effect on tree structure

Query execution

- One pass:
  - Selection, project
  - Group by, distinct (constraint on memory)
  - Join (one of relations fit in memory)

- NLJ
  - Block-based algorithm
  - Costs: putting smaller relation in outer

Two pass join algorithms

- Sort-merge join
  - Pass 1: sort
  - Pass 2: merge
  - Constraints: B(R) + B(S) <= M * (M-1)

- Partitioned hash join
  - Pass 1: hashing R/S into (M-1) buckets
  - Pass 2: merge Ri and Si
  - Constraints: min(B(R), B(S)) <= (M-1)*(M-2)

Variations

- Simple-sort based join
  - Completely sort R & S
  - Merge

Index-based algorithm

- Selection
  - Costs: clustered index vs non-clustered index
- R(A,B) Join S(A,C): with index S.A
  - Costs: index clustered vs non-clustered
- Compare it to NLJ
- Zig-zag join: clustered indexes on R.A and S.A

MongoDB

- Find()
  - Pattern matching
  - Query operators: $gt, $lt, …, $in, $all
  - projection
  - sort
  - distinct
  - count
  - skip
  - limit
  - update (upsert)

- Aggregate()
  - Pipeline: $match, $group, $match, …

Hadoop MapReduce

- Map function & reduce function
  - Input, output, logic
- Shuffling task
- Combiner
- SQL implementation
  - No need for join

Spark

- Creation: textFile, parallelize
- Transformations
  - map, flatMap, mapValues
  - filter
  - groupByKey, reduceByKey, aggregateByKey, sortByKey
  - distinct
  - join (outer…)
  - Set operations & semantics: union, intersection, subtract
- Action
  - reduce, max, min, sum, count, mean, aggregate, countByKey

Final exam

- MW section: May 6, Wednesday, 4:30-6:30pm
  - DEN
- Tuesday section: May 12, Tuesday, 2-4pm
  - Blackboard
- Online, similar to your quiz
  - closed-book and notes