# Query Evaluation & Optimization

November 19, 2019

Chapter 12-12.3.4, 14.4, 9.4

# Goal

- “Why don’t you just put your slides in the correct order?”
  - anonymous student

- “Begin at the beginning," the King said, very gravely, "and go on till you come to the end: then stop.” - Carroll

# Hashing

- Hash function for hash table
  - goal:

- Hash function for hashing passwords
  - cryptographic hash
  - goal:

# Catalog

- Meta data about the tables
    - names
    - column name, domain
    - indexes
    - size
- Cardinality or NTuples(R)
- Size or NPages(R)
- Index Cardinality or NKeys(I)
- Index Size or INPages(I)
- Index Height or IHeight(I)
- Index Range or ILow(I)/IHigh(I)

# Setup

- Sailors(sid, sname, rating, age)
- Reserves(sid, bid, day, rname)

- Reserves: 40 bytes per tuple
  - NPages(Reserves) = 1000
  - NTuples(Reserves) = 100000
  - NKeys( <rname, bid, sid>) = 100

- Sailors: 50 bytes per tuple
  - NPages(Sailors) = 500
  - NTuples(Sailors) = 40000

# Simple Heuristics

- Indexing
- Iteration
- Partitioning

# Access Path

- How to retrieve a tuple from a table

- File Scan

  OR

- index plus matching selection condition

# Matching

- Conjunctive Normal Form
  - may only match subset
  - primary conjuncts

- Hash index

- Tree index

# Cost

- Selectivity of access path
    - most selective
    - reduction factor
- Index File
- Data File

# Operations

- Selection

- Projection
  - remove duplicates
  - SELECT DISTINCT(FName) FROM Students
  - SELECT COUNT(DISTINCT(FName)) FROM Students
  - partitioning: scan then sort
    - with index
    - with clustered index

# Operations, cont p 403

- Join
  - index nested loops join
  - Reserves.sid=Sailors.sid
  - how many I/O operations are needed?
  - What do we know about Reserves, Sailors, sid?
  - Reserves.rname = Sailors.sname
    - how many I/O operations are needed?
    - sort-merge join

# Evaluation Plans p406

```
SELECT S.sname
FROM Reserves as R, Sailors as S
WHERE R.sid=S.sid AND R.bid=100 AND S.rating > 5
```

- with hash indexes on bid another hash index on Sailors.sid
- what if we had a tree index on rating?

# Joins (14.4)

- Nested Loops Join
- Block Nested Loops Join
- Index Nested Loops Join
- Sort-merge Join (Oracle) - we won’t discuss this
- Hash Join (MariaDB)

```
SELECT *
FROM Reserves as R, Sailors as S
WHERE R.sid=S.sid
```

- p452
- p440 table stats

**https://dev.mysql.com/doc/refman/5.7/en/nested-loop-joins.html**

# Nested Loops Join 14.4.1

- No indexes involved.
- Unrealistic

# Block Nested Loops Join

- Utilize buffer pages
- Load all of or most of the smaller table into memory
- Realistic

```
explain select * from Prof_Ex, JobStatus

where Prof_Ex.StatusID=JobStatus.StatusID;

-- ALL, ALL, Using where; Using join buffer (flat, BNL join)

-- https://mariadb.com/kb/en/library/block-based-join-algorithms/
```

# Index Nested Loops Join

- Take advantage of an index on the inner relation

# Index Nested Loops Join, Example

# Hash Join

- Build in memory hash table
  - 14.3.2