

# COSC 580 DBMS Project

**“Gentle” Guidelines / Suggestions**

1

1

## Architecture

- Query Input Manager
- Data Definition Language
- Data Manipulation Language
- Main Memory Execution
- Storage Structures – Disk
- [Optimization]

2

2

## Query Input Manager

- Query Input Manager
  - Input:
    - “Line” editor
    - “Visual” editor
  - Parser
    - Obtain SQL grammar
    - Recursive descent parser
      - *Should code – if taken, state source.*

3

3

## Data Definition Language

- No actual data are processed
- Implementation
  - Maintain two “tables”: TABLES & COLUMNS
- Must implement
  - CREATE TABLE
    - Enter relation name to TABLE
    - Enter attributes to COLUMNS
  - DROP TABLE
  - CREATE INDEX
    - **Required: single attribute index**
    - **As such – a simple hash to posting block list works!**
  - DROP INDEX

4

4

## Data Manipulation Language

- Actual Tuple Manipulation
  - Operators
    - INSERT
    - DELETE
    - UPDATE
  - Key considerations
    - Duplicates
    - Referential integrity
  - Indexing issues

5

5

## Data Manipulation Language

- SELECT
  - Parse components
  - Validate table selection
  - Project attributes for all identified tables
  - Determine “operators” & “operands”
    - Identify appropriate available indexes
    - [Optimize]
  - Execute query tree

6

6

## Main Memory Execution

- Attribute value distributions
  - Conjunctive and disjunctive selections
  - Determination of inner vs. outer join
- Efficient sorting
  - Expectation of “sorted” elements – sort selection or mostly sorted
- Hash joins vs. sorting
  - If “nearly sorted” – linear matching more efficient
  - Unsorted then hash

7

7

## Storage Structures – Disk

- Trees of packed blocks
  - Storage inefficiency vs. insertion inefficiency
- Delayed writes
  - Efficiency vs. reliability
- Suggested guidance:
  - Implement via simplicity
  - Replace for efficiency (once working)

8

8

## Optimization - Potential

- Rule based
  - Selections and Projections at bottom
- Cost based
  - Operation ordering based on selectivity
  - Dynamic join algorithm selection
    - Inner vs. outer
    - Sort vs. linear
    - Hash

9

9

## Remember:

Slow but functional

Exceeds

Fast but chaotic

10

10