What we learned this semester

- Data modeling and query languages
- DB application design and implementation
- DB system design

Before Midterm

- Data Modeling and Query language
  - ER modeling design
  - Relational model
  - Querying RDB
    - Relational algebra
    - Relational calculus
    - SQL
  - Semi-structured Data
    - XML
    - XPath, XQuery
- DB application design
  - Web-DB application
  - Software engineering

DB system design

- Storage management
- Indexing
- Sorting
- Query evaluation and optimization
- Concurrency control

Storage management

- Basic concepts
  - Memory hierarchy
  - Disk access: operations, units, features
  - Access methods
    - File scan
    - Index scan
- Storage management
  - Organizing tuples on pages
  - File organization
- Buffer management
  - What's buffer
  - Flags
  - Buffer replacement policy

Skill set:

- Given the metadata of a DB, compute the I/O cost of certain access method
- Given the metadata of a DB, compute the I/O cost of certain file updates.

Indexing

- Important concepts
  - Search key
  - Index entry
  - clustered index, non-clustered index, …
- Three types of index
  - ISAM
  - B+ tree
  - Hash

Skill set:

- Given an index, compute the cost of accessing/updating index entries
- Present examples for best/worst case scenarios for index access and update.
- Given the metadata of a DB and a query, choose which index can benefit the query evaluation the most, and explain why.

Sorting

- Terminology
  - Run
  - Sub-file
- Sorting algorithms
  - Simple two way merge sort
  - General external merge sort
  - Double buffering
- Using B^+-tree for sorting

Skill set:

- Given a file and system setting, compute the I/O cost of sorting, using a given sorting algorithm
- Given a file and system setting, construct the intermediate results after certain number of passes, using a given sorting algorithms.

Query Evaluation and Optimization

- Concepts
  - Evaluation plan
  - Operators
- Choosing access methods
  - File scan vs. index scan
- Join algorithms:
  - SNLJ, PNLJ, BNLJ, INLJ, SMJ, HJ
  - Understand
    - When they can be used
    - When they improve query performance the most
- Optimization
  - Strategies
  - Cardinality estimation

Skill set:

- Given metadata and system setting and a query
  - Estimate cardinality of intermediate results
  - Choose access method
  - Choose join algorithm
  - Choose join order
  - Choose operators and evaluation order
  - Construct evaluation plan
  - Explain why one plan is better than another.

Concurrency Control

- Concepts
  - Schedule, equivalent schedule, serializability
  - Lock, deadlock
- Lock management
  - Protocol
  - Types of locks: S, X, U
  - Lock management algorithm

Skill set:

- Given a schedule, determine whether it is serializable
- Given a sequence of transactions, simulate the evaluation using lock-based protocol.
- Detect and resolve deadlock

Question from Previous Exam

Consider a relational schema:

Employee (eID, name, age, salary)

and a query that looks for the name of each employee who is younger than 40 and earns less than 20K a year.

Assume that you know the following:

- The Employee table is stored in a heap file.
- There is a B+tree index on salary and a B+tree index on age of the Employee table; and these are the only indices.
- Statistical information indicates that 1% of all employees earn less than 20K and about 20% of employees are older than 40. There is no functional dependency or correlation between age and salary.

Please design an evaluation plan of the query that is most efficient under the circumstance and explain your rational.

Hint: use both indices, and set operation on the resultant RIDs before file access opeartion.

Question from Previous Exam

Consider the following actions from the transactions T_1 and T_2, as ordered in which the requests are to be issued. Assume that object A and B are two pages in two different tables in the database and 2PL is the concurrency control policy in effect. Please explain how the lock manager handles the requests and how the transactions proceed.

T_1:		R(A)	W(B)		Commit
T_2:	R(A)	R(B)	R(B)	W(A)	Commit

Hint: use U lock to avoid deadlock

Question from Previous Exam

Given the following DB schema:
Student (sid, name, gender, GPA, dept,age)
dept is a foreign key, it references to department(name).
Department (name, location)
Please discuss whether the following two queries are equivalent and why.
Note: two queries q1 and q2 are equivalent if for any given DB instance D, q1(D) = q2(D)

Q1:
```
select distinct S.dept as deptName
from student as S
where (S.age < 20 or S.gender = "female")
and GPA >3.8
```

Q2:
```
select name as deptName
from department
except
select S.dept as deptName
from student as S
where (S.age >= 20 and S.gender = "male")
or GPA <=3.8
```

Hint: use van diagram to interpret the queries.