Final Review
Homework #5: Due Thursday Dec 2\textsuperscript{nd} @ 11:59pm.

Project #4: Due Sunday Dec 5\textsuperscript{th} @ 11:59pm. Additional office hours on Saturday Dec 4\textsuperscript{th} @ 3:00pm.

Final Exam: Friday Dec 10\textsuperscript{th} @ 8:30am at Doherty Hall 2210. Bring pencil and rubber.
Exam focuses on topics after mid-term. But questions may need understanding of earlier lecture material.

Open book/notes/calculator.

We will post announcements on Piazza with practice exam.
COURSE EVALS

Your feedback is strongly needed:
→ [link](https://cmu.smartevals.com)
→ [link](https://www.ugrad.cs.cmu.edu/ta/F21/feedback/)

Things that we want feedback on:
→ Homework Assignments
→ Projects
→ Reading Materials
→ Lectures
STUFF BEFORE MID-TERM

SQL
Buffer Pool Management
Hash Tables
B+Trees
Storage Models
Inter-Query Parallelism
QUERY OPTIMIZATION

Heuristics
→ Predicate Pushdown
→ Projection Pushdown
→ Nested Sub-Queries: Rewrite and Decompose

Statistics
→ Cardinality Estimation
→ Histograms

Cost-based search
TRANSACTIONS

ACID

Conflict Serializability:
→ How to check?
→ How to ensure?

View Serializability

Recoverable Schedules

Isolation Levels / Anomalies
TRANSACTIONS

Two-Phase Locking
→ Rigorous vs. Non-Rigorous
→ Deadlock Detection & Prevention

Multiple Granularity Locking
→ Intention Locks
TRANSACTIONS

Timestamp Ordering Concurrency Control
→ Thomas Write Rule

Optimistic Concurrency Control
→ Read Phase
→ Validation Phase
→ Write Phase

Multi-Version Concurrency Control
→ Version Storage / Ordering
→ Garbage Collection
CRASH RECOVERY

Buffer Pool Policies:
→ STEAL vs. NO-STEAL
→ FORCE vs. NO-FORCE

Write-Ahead Logging

Logging Schemes

Checkpoints

ARIES Recovery
→ Log Sequence Numbers
→ CLRs
DISTRIBUTED DATABASES

System Architectures
Replication
Partitioning Schemes
Two-Phase Commit
FINAL COMMENTS

Know your goal, constraints, and resources
→ Focus on “high pole in the tent”
→ Keep remind yourself and re-evaluate

Avoid pre-mature optimization/engineering for non-exist requirements
→ Prefer simple solutions

Avoid cutting corners
→ Balance engineering effort and extensibility