

Final Exam Study Guide

Quick Comments

- The Final Exam will be comprehensive
- You will have three hours; though some will finish starting about 90 minutes
- Exam 1, Exam 2, assignments, and quizzes are a good start on format and content for final exam questions
- Grades, other than the final exam, should be posted on Brightspace before the exam. I typically have two different weighting schemes for the exam, and give each student the max based on their scores

Relational Algebra

- Know basic operators of
 - projection,
 - selection,
 - cross product,
 - set operations,
 - joining (natural and theta)
 - renaming
- Construct RA expressions that combine the basic operations from English specifications
- Know query evaluation trees
 - Know left deep trees
 - I will NOT ask you to annotate evaluation trees (e.g., with join and indexing annotations) to turn trees into plans, or evaluate them for cost, though we discussed it briefly in lecture (slides 17-23 in 3/22 lecture), but do understand the meaning of the evaluation plan on slide 16

SQL Queries

- Know syntax of all clauses: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY
- Know order of evaluation for these clauses (FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY)
- Assume the SQL standard, in which all attributes listed in the SELECT clause, and not an argument to a aggregate operator, must be listed in the GROUP BY clause when that GROUP BY clause is present
- Know set operations of UNION, INTERSECT, EXCEPT
- Know use of JOIN keywords, but emphasis on representing joins in WHERE clause
- Know the aggregate operators, and know their use with and without GROUP BY
- Know nested queries in FROM, WHERE, HAVING clauses
- Know correlated queries
- Be able to interpret SQLite syntax diagrams (e.g., https://www.sqlite.org/lang_select.html)
- Know SQL and Relational Algebra equivalencies
- Write queries from English specifications

SQL CREATE TABLE statements

- Declaring PRIMARY KEYs, FOREIGN KEYs, UNIQUEs, NOT NULL
- Know that PK attributes cannot be NULL, and that FK attributes can be NULL
- In-table CHECK statements
- Be able to trace DELETE and UPDATE actions on sample tables based on FK declarations, as in January 30 lecture
- Understand the Sample DB table declarations (i.e., dorm energy, book retailer)

SQL CREATE ASSERTION statements

- Know the assertions for complete coverage, no overlap, participation (1..*) constraints

CREATE TRIGGER statements

- BEFORE, AFTER INSERT (new), UPDATE (new, old), DELETE (old)
- INSTEAD OF (in conjunction with Views)
- Using triggers be used to enforce constraints (e.g., assertions)
- Just need to know SQLite triggers

CREATE VIEW statements

- Single table (automatically updatable)
- Multiple tables (updatable through INSTEAD OF triggers)

Unified Modeling Language (UML) models

- Know syntax of all major constructs
 - Classes, association classes, sub-classes, cardinalities,
- Know rules for translating UML into SQL CREATE table and assertion statements
- Be able to identify UML models that are consistent with table and assertion definitions
- Construct from English specifications

Relational Design

- Functional dependencies
- Identify all keys given FDs
- Identify minimal FD sets
- Decompose relations based on FDs
 - into lossless, dependency preserving decomposition
 - of BCNF relations

Indexes

- Know insertion procedures for B+ tree and extendible hashing indexing structures
- Know legal and illegal B+ tree and extendible hashing structures
- Know how joins are evaluated, as nested loops, and how indexes can lead to more efficient joining (and selections)

Know the high level justification strategy for selecting indexes on slides 24 and 25 of in 3/22 lecture

Transactions

Recognize and construct serializable schedules

Know what dirty reads and the thrashing are

Know 2PL with shared and exclusive locks, with examples on indexing structures and tables

Other Material

Know the IEEE Code of Conduct

Know the difference between approximate functional dependencies and association rules

Compute approximate functional dependencies from simple data

Know the difference between supervised and unsupervised machine learning

Be able to interpret a decision tree

Be able to identify which attribute (aka variable) would be used to divide sample data in the first step of the greedy decision tree learning algorithm

Be conversant on equivalencies between RA, SQL, FDs, UML

Identify design and implementation flaws in relational models