

Exam 2 Study Guide

The Exam 2 Concepts include

- Relational Algebra (DB4) • SQL
- Querying and updating (DB5)
- Constraints and Triggers (DB11)
- Unified Modeling Language (DB9)

About 1/3 of it will include material that you were responsible for on Exam 1, and 2/3 on material we studied since then.

Thus far, we have studied design and implementation of relational databases, with UML as primarily a tool for (conceptual) DB “design in the large”, with SQL as the common target language for DB implementation. Relational algebra is a query language of theoretical interest, and it supplies some of the underlying semantics for some of SQL’s query functionality. An important goal is to understand the relationships between the three basic languages/frameworks that you have studied: Relational algebra, SQL, and UML (in exam 3 we will add another framework, functional dependencies, for “design in the small”)

Study sources include the Widom videos, class handouts/exercises, including sample DBs (i.e., Books and Dorm Energy), and Readings (though this latter not highlighted below).

- Relational Algebra Basic Operations
 - Projection, Selection, Cross Product, Join
 - Equivalence to selection on a cross product
 - Equality join
 - Natural join
 - Special case with no attributes with same name/type
 - Theta join
 - Properties of joins, like Symmetry
 - Set Operators
 - Intersection
 - Union
 - Difference
 - Renaming Operator
 - Rho notation
 - Assignment notation
 - Role in self joins
 - Expression Trees
 - Left Deep Expression Tress

Relational algebra (RA) is a relational DB query language, of interest because it is often used to clarify and relate the semantics of actual query languages, most notably SQL, and it is the symbolic language of choice in representing (as expression trees) various query evaluation strategies, in search of the most efficient strategy.

Be prepared to write RA queries that satisfy specification in English, as well as to recognize equivalences and differences between pairs of RA queries, and between RA queries and SQL queries. You may be asked to write queries from scratch, debug and complete partial queries, and identify relationships, most notably equivalences, through multiple choice questions.

- SQL queries
 - Basic clauses
 - SELECT clause
 - Relation to RA projection operator
 - FROM clause
 - Relation to RA cross product operator
 - WHERE clause
 - Relation to RA selection and joining operators
 - Aggregation and Grouping
 - GROUP BY clause
 - HAVING Clause
 - Aggregation operators (SUM, COUNT, MAX, MIN, AVG)
 - In SELECT Cause Without GROUP BY
 - With GROUP BY
 - In HAVING Clause
 - Subqueries (or nested queries)
 - In FROM clause
 - JOIN keyword constructs
 - In Where clause Correlated queries
 - Logical and set operators IN and NOT IN
 - INTERSECT, UNION, EXCEPT
 - EXISTS and NOT EXISTS
 - ANY and ALL (A+ question possibility – there are others that I haven't highlighted)
 - Data modification (UPDATE, both forms of INSERT, DELETE)
 - Other
 - ORDER BY
 - Order of clause evaluation NULL values

Be prepared to write SQL queries that satisfy specification in English, to evaluate queries on sample DBs, and recognize equivalences and differences between pairs of SQL queries, and between SQL queries and RA queries. You may be asked to write queries from scratch, debug and complete partial queries, and identify relationships, most notably equivalences, through multiple choice questions. The complexity of the queries will vary significantly across questions.

- Constraints and Triggers (in SQL)
 - Attribute constraints
 - NOT NULL,

- PRIMARY KEY and UNIQUE
- In table CHECK Statements
 - Evaluation on inserts/updates to relevant table
- FOREIGN KEY Constraints
 - ON DELETE and ON UPDATE actions
 - CASCADE, NO ACTION (OR RESTRICT), SET NULL, SET DEFAULT
 - Emphasis on CASCADE and NO ACTION (RESTRICT)
 - Be able to reason about chain of actions (as with in-class exercise)
- General Assertions
 - Evaluation when inserts or updates made General Assertions
- Triggers
 - ON INSERT
 - ON UPDATE
 - ON DELETE
 - BEFORE and AFTER (focus on AFTER)
 - TABLE and ROW LEVEL (focus on ROW LEVEL)
 - FOCUS on SQLite syntax

Be prepared to write SQL CREATE TABLE, CREATE ASSERTION, and CREATE TRIGGER statements in SQL from English specifications, and UML specifications. For Triggers, assume SQLite syntax, and for table and assertion creation be able to use (at your discretion) SQLite syntax and syntax of various sample DBs and keys/rubrics (Dorm energy monitoring and Books databases) that you have been given. Be able to recognize (in)correct and (in)consistent definitions, between frameworks (SQL, UML) through multiple choice questions.

- Unified Modeling Language (UML) Classes
 - Primary Keys (PK)
 - Associations
 - Multiplicity (or Cardinality) constraints
 - Association Classes
 - Keys for Association Classes
 - Composition and Aggregation
 - PK Box (adding attributes from the primary key of one table to another) – see sample Books and Dorm Energy DB exercises
 - Subclasses (ISA)
 - Coverage (Complete or Partial Coverage)
 - Overlap (allowed or not)

Be prepared to draw UML diagrams that are consistent with specifications in English. Be able to compare UML DB specifications with table/assertion definitions in SQL to identify equivalences and inconsistencies (through multiple choice), and to translate UML to table/assertion definitions. Study the sample DB designs and in-class exercises that have been provided.