CS252:HACD Fundamentals of Relational Databases

Expected Learning Outcomes

These notes on Hugh Darwen's contribution to CS252 are intended to assist students' preparation for the CS252 exam. To be read in conjunction with the slides and notes for each lecture, they give you the important points of learning that you might be tested on in the exam.

1. Introduction

- What is expected of a database, and of a DBMS
- How a relational database and a relational DBMS meet these expectations
- The structure ("anatomy") of a relation and the terms used to describe it

2. Values, Types, Variables, Operators

- Ability to use the terminology accurately
- **Tutorial D** Notation for tuple types and relation types
- **Tutorial D** Notation for tuple literals and relation literals
- The important distinctions listed on Slide 3 and again on Slide 17

3. Predicates and Propositions

- What is a predicate?
- Parameters (free variables) of a predicate
- What is a proposition, and in what ways is it a special case of a predicate?
- Ways of deriving predicates from predicates (substitution, logical connectives, quantification)
- Basic set theory and its terminology

4. Relational Algebra, Principles and Part I

- Intension and extension of a predicate, and how a relation represents an extension
- The Closed World Assumption
- Correspondence between logical operators and relational operators
- JOIN as relational counterpart of AND
- RENAME
- Properties of JOIN and degenerate cases
- Projection as relational counterpart of existential quantification
- Degenerate cases of projection, TABLE_DEE and TABLE_DUM
- **Tutorial D** syntax for invoking JOIN, RENAME, projection

5. Relational Algebra, Part II

- Restriction (WHERE) as another relational counterpart of AND
- Extension (EXTEND) as yet another relational counterpart of AND
- Relation-valued attributes ("relations within a relation")
- Aggregate operators like SUM
- SUMMARIZE ... PER ... as a useful shorthand for statistical analysis (and SUMARIZE ... BY ... as a further shorthand for certain special cases)
- UNION as relational counterpart of OR, and why it is restricted the way it is
- NOT MATCHING as relational counterpart of *p* AND NOT *q*, and why support for negation is thus restricted
- **Tutorial D** syntax for invoking restriction, extension, SUMMARIZE, UNION and NOT MATCHING

6. Relational Algebra, Part III

- Semijoin (MATCHING)
- Composition (COMPOSE)
- GROUP/UNGROUP
- Relational comparison operators
- **Tutorial D** syntax for invoking semijoin, composition, GROUP, UNGROUP, and relational comparison operators
- TUPLE FROM and attribute FROM

7. Constraints and Updating

- The terms *key* and *superkey*
- KEY constraints
- Degenerate cases of keys
- Foreign keys and inclusion dependencies
- Use of relational comparisons in constraints
- IS_EMPTY
- Disjointness constraints ("exclusion dependencies")
- **Tutorial D** syntax for constraint declaration
- INSERT, UPDATE, DELETE
- "Multiple assignment" and its motivation

8. Database Design Issues, Part I

- Reducibility (of relations)
- Join dependency (JD) and notation for expressing it

- Decomposition by projection
- Problems addressed by decomposition
- Functional dependency (FD) and notation for expressing it
- FD terminology: *determinant*, *dependant*
- FD theorems (left augmentation, right reduction, transitivity, and the "general" theorem)
- Left-irreducible FDs
- Using the theorems to determine keys implied by given sets of FDs
- The concepts of normalisation and normal forms

9. Database Design Issues, Part II

- BCNF, and examples of relvars where BCNF does not hold
- Distinction between trivial and nontrivial FDs
- JDs that are consequences of FDs
- Effect of decomposition on constraints
- Anomalous cases where even a BCNF relvar exhibits redundancy
- 5NF
- Distinction between trivial and nontrivial JDs
- Nontrivial JDs that are *not* consequences of FDs
- When a JD is "implied by keys"
- 6NF (and when it is advised or not advised)

End of Notes