

IT420 Spring 2007
Review Sheet

1. Introduction to databases

Covered in:

- Lecture set 1
- Chapter 1

Sub-topics:

- a. Database Management Systems benefits

2. Data Modeling with the Entity-Relationship Model

Covered in:

- Set 2, 3
- Chapter 5

Sub-topics:

a. Entities

- Identifiers /Composite identifiers
- Attributes
- Strong entities
- Weak entities
- Id-dependent entities

b. Relationships

- Has-A relationships
 - Maximum and minimum cardinality
 - Identifying/non-identifying relationships
- Is-A relationships (supertype/subtype)
 - Inclusive/Exclusive

3. The Relational Model

Covered in:

- Set 4
- Chapter 3, pages 69-74, 79-81

Sub-topics:

- a. Relation /Table
 - Attributes
- b. Integrity Constraints
- c. Keys
- d. Primary key
- e. Candidate key
- f. Surrogate key
- g. Foreign key
 - Referential integrity constraint

4. Transforming ER diagrams to Relational Model

Covered in:

- Set 5
- Chapter 6

Sub-topics:

- a. Transform entities
 - Specify primary key
 - Specify candidate (alternate keys)
 - Specify properties for each column
 1. data type
 2. null /not null
 3. default values
 4. other constraints
- b. Transform relationships (foreign keys used here)
 - 1:1 relationships, 1:N relationships
 - identifying relationships
 - non-identifying relationships
 - N:M relationships
 - Supertype/subtype relationships
- c. Specify logic to enforce minimum cardinalities

5. SQL

Covered in:

- Set 6, 7
- Chapter 7, pg 220-234 and Chapter 2

Sub-topics:

- CREATE
- DROP
- ALTER
- INSERT
- DELETE
- UPDATE
- SELECT...FROM... WHERE... framework
- Conceptual evaluation of queries
- DISTINCT keyword
- ORDER BY
- Aggregate operators: COUNT, MIN, MAX, AVG, SUM
- GROUP BY... HAVING
- Subqueries
- Join

6. Normalization

Covered in:

- Set 8
- Chapter 3, pages 74-99
- Chapter 4

Sub-topics:

- Purpose
- Insert /delete/update anomalies
- Functional dependencies
 - Definition of key based on functional dependencies
- Normal forms
 - First normal form
 - Boyce-Codd Normal Form
 - Decomposition into relations that are in Boyce-Codd Normal Form
- Multivalued dependencies
 - Fourth Normal Form

7. PHP and MySQL

- a. variables, constants, arrays (numerically indexed arrays and associative arrays)
- b. control statements (if, for, foreach, while, ...)
- c. files
- d. functions
- e. objects/classes, inheritance
- f. Work with MySQL using PHP
 - Connect to a database
 - Query
 - process results
 - close connection
- g. authentication and sessions
 - Passwords
 - Session variables

8. SQL VIEWS

- a. SQL View is a virtual table that is constructed from other tables or views.
- b. Syntax: `CREATE VIEW viewname AS viewquery`
- c. Order By clause cannot be used in the Create View query.
- d. A view can be queried as if it is a table.
- e. Uses for views
- f. Updating views

9. Triggers

- a. A trigger is a stored program that is attached to a table or view.
- b. Type of triggers
- c. Uses for triggers
- d. Writing a trigger

10. Stored Procedures

- a. A stored procedure is a program that performs some common action on database data and is stored in the database.
- b. Advantages of stored procedures
- c. How to write a stored procedure
- d. Differences between triggers and stored procedures

11. Database security

- a. Create users
- b. Grant/revoke permissions
- c. Changing passwords
- d. General security guidelines

12. Concurrent processing

- a. Transactions
- b. Concurrent transaction processing
 - Lost Updates
 - Inconsistent read
 1. Dirty read
 2. Unrepeatable read
 3. Phantom read
- c. Resource locking
 - Implicit/explicit locks
 - Lock granularity
 - Exclusive/ shared locks
 - Serializable transactions
 - Two-phase locking, String two-phase locking
 - Deadlocks
 1. preventing deadlocks
 2. breaking deadlocks
 - Optimistic / pessimistic locking
 - Using implicit locks/ declaring transaction boundaries
 - ACID transactions
 1. Atomic
 2. Consistent
 - a. statement level
 - b. transaction level
 3. Isolation
 - a. read uncommitted
 - b. read committed
 - c. repeatable read
 - d. serializable
 4. Durable

13. Storage and indexing

- a. Disk components
- b. Time to access a disk block
- c. 'Next' concept
- d. Hash index
- e. B+Tree index
- f. Clustered / unclustered index
- g. **How to choose indexes to improve performance**