1. **Number and title of course:** CS 186, Introduction to Database Systems

2. **Course Objectives:**
   - Fundamental understanding of the theory and practice of data representation, query languages, and their interrelationships. Exposure to leading data and query models including the Relational model, document and web search, and XML.
   - Practical experience with implementation of disk-oriented system internals, with a focus on the efficient management of disk and memory. The ability to benchmark these implementations and determine their efficacy.
   - Theoretical understanding of transactional consistency semantics, and detailed concurrency and recover protocols to implement these semantics.
   - Practical experience in the design and implementation of multi-tier data-intensive applications, integrating Internet-based interfaces, application services, and backend databases.

3. **Topics covered:**
   - System Fundamentals
     i. Disks
     ii. Files
     iii. Buffer management
   - Core query processing techniques
     i. Indexing
     ii. Sorting
     iii. Join and matching algorithms
   - Data models and semantics
     i. The Entity-Relationship Model
     ii. Relational Model
     iii. XML
     iv. Documents and vector spaces
   - Query languages and semantics
     i. Relational Algebra and Calculus
     ii. SQL
     iii. XQuery
     iv. Keyword and ranked search
   - Document and Web Search
     i. Boolean Text Search
     ii. Ranked Search
   - Transactions and data consistency
     i. Theory of Transactions
     ii. Concurrency Control
     iii. Recovery: The ARIES protocol
   - Theory and practice of Database design
     i. Functional Dependencies
     ii. Normalization
     iii. Physical Database Design and Tuning
   - Parallelism
     i. Forms of parallelism
     ii. Parallelizing query operators
     iii. Encapsulating parallelism: the Exchange operator
     iv. Implications for indexing, database design and concurrency control
4. **Relationship of course to program objectives:** The course integrates and exercises a combination of mathematical foundations and engineering discipline from the lower-division curriculum via an integrated approach to information management. Problem sets in query specification and schema normalization exercise theoretical issues in first-order logic and axiomatic reasoning. Hands-on, group-based software projects direct students to solve challenging engineering design problems, including the real-world difficulties of integrating with legacy code inside a production database engine. In the course of these projects they use a host of modern tools and techniques, including debuggers, memory access checkers, web scripting and services, and database programming interfaces. Empirical benchmarking exercises require the students to go beyond design and implementation, to understand the implications of design tradeoffs by examining their impact on performance for different workloads.

5. **Prepared by:** Joseph Hellerstein (3/20/06)