

**72.37 – Databases I**

Software Engineering Department

6 credits

**Content**

**1 Characteristics of a DBMS**

Databases and Database Management System (DBMS). Differences between a DBMS and a File System. Database Models: conceptual, representation/implementation, and physical models. Difference between Metadata and Data. Three-tier architecture proposed by ANSI / SPARC Study Group in Database Management Systems. Profiles of people who interact with a DBMS. Languages for Databases: Data Definition Language (DDL) and Data Manipulation Language (DML). Embedded DML: host language, data language, and precompiler role. DML compiler and run-time database processor. Ways to access databases from an application program: embedded, ODBC and 4GL.

Typical architectures for DBMS: Centralized and Distributed Client / Server. Role played by ~~d~~atabasea in Information Systems.

**2 Conceptual Model: Entity-Relationship Model**

Entity Set: weak and strong. Relationship Set: degree, role, recursive. Cardinality restrictions. Participation (total or partial). Attributes: Domain, identifiers (key) / descriptive, atomic/compound, single-valued/multivalued, stored / derived. Extensions: representation of hierarchies (generalization and is-a). Extended entity / relationship diagram.

**3 Representation / Implementation Model: Relational Model**

Relationship concept. Attributes and Domains. Restrictions: domain, key, entity integrity and referential integrity. Mapping the entity / relational diagram to the relational model schema.

Programming language operations for DDL and DML.

**4 Query Languages ​​Algebra and Relational Calculus**

Procedural query language: Relational Algebra. Fundamental operations: selection, projection, union, difference, Cartesian product. Derivative operations: intersection, natural joint, semi-natural joint, theta-join, division. Non-procedural query language: Relational calculation of tuples and domains. Safe expressions. Comparison of the expressive power of languages.

**5 SQL Query Language**

SQL as a hybrid query language. Statements for the definition of persistent tables. Sentences for data manipulation. Nested queries and use of aggregation functions. Handling of the null value. Views.

Temporary tables. Integrity constraints in SQL2. Concept of completeness in languages.

Recursive queries: fixed point and Recursive SQL.

**6 Introduction to database administration**

Representation of data storage for a database. Index structure for primary keys: B+ Tree. B+ index for non-key attributes. Use of utility programs: extraction of the schema, export and bulk import of data. Database migration between different DBMS.

**7 Normalization Theory**

Analysis of redundancy and anomalies. Functional dependencies. Normalization process: decomposition without loss and preservation of dependencies. Normal forms: First, Second, Third, and Boyce-Codd.

Multivalued dependencies and fourth normal form. Joint dependency and fifth normal form. Armstrong axioms and inference rules. Algorithms. Using Tableau

**8 Advanced SQL (views, triggers, PSM)**

Fictional keys and self number attributes. Integrity constraints in SQL3: programming assertions and triggers. Triggers at the tuple level and at the sentence level. Triggers, authorization and security schema. View management with triggers. Stored Procedures programming. Concept and programming of explicit and implicit cursors. Conditions management and exceptions. Transaction management and choosing the appropriate level of isolation to facilitate concurrence. Permissions management and roles.

**9 Embedded Programming**

Programming of client applications with database access. Processing data obtained in the client: errors communication between the DBMS and the host language, detection of null as value, detection of data truncation. Embedded programming in C / C ++ language. Programming with Java Database Connectivity (JDBC). Different types of Drivers. Introduction to programming with fourth generation languages (4GL). Application workshop with two-tier client / server architecture.