Contents

University de Piura (UDEP)
Sillabus 2022-I

1. COURSE

CS271. Data Management (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS112. Computer Science I. (2^{nd} Sem) • CS1D2. Discrete Structures II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information management (IM) plays a major role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of accessing and updating stored information, data modeling and abstraction, and physical file storage techniques.It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which (IM) methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable restrictions, including Scalability and usability.

5. GOALS

- That the student learn to represent information in a database prioritizing the efficiency in the recovery of the same.
- That the student learn the fundamental concepts of the management of databases. This includes the design of databases, database languages and the realization of databases.
- Discuss the database model with the base in relational algebra, relational calculus and the study of SQL statements.

6. COMPETENCES

Nooutcomes

Nospecificoutcomes

7. TOPICS

Unit 1: Database Systems (14)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Approaches to and evolution of database systems • Components of database systems • Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods) • Database architecture and data independence • Use of a declarative query language • Systems supporting structured and/or stream content • Approaches for managing large volumes of data (e.g., noSQL database systems, use of MapReduce).	• Explain the characteristics that distinguish the database approach from the approach of programming with data files [Usage] • Describe the most common designs for core database system components including the query optimizer, query executor, storage manager, access methods, and transaction processor [Usage] • Cite the basic goals, functions, and models of database systems [Usage] • Describe the components of a database system and give examples of their use [Usage] • Identify major DBMS functions and describe their role in a database system [Usage] • Explain the concept of data independence and its importance in a database system [Usage] • Use a declarative query language to elicit information from a database [Usage] • Describe facilities that datatbases provide supporting structures and/or stream (sequence) data, eg, text [Usage] • Describe major approaches to storing and processing large volumes of data [Usage]
Readings : [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]	

Unit 2: Data Modeling (14)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Data modeling • Conceptual models (e.g., entity-relationship, UML diagrams) • Spreadsheet models • Relational data models • Object-oriented models • Semi-structured data model (expressed using DTD or XML Schema, for example)	• Compare and contrast appropriate data models, including internal structures, for different types of data [Usage] • Describe concepts in modeling notation (eg, Entity-Relation Diagrams or UML) and how they would be used [Usage] • Define the fundamental terminology used in the relational data model [Usage] • Describe the basic principles of the relational data model [Usage] • Apply the modeling concepts and notation of the relational data model [Usage] • Describe the main concepts of the OO model such as object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning [Usage] • Describe the differences between relational and semi-structured data models [Usage] • Give a semi-structured equivalent (eg, in DTD or XML Schema) for a given relational schema [Usage]
Readings : [SW04], [EN04], [KS02]	

Unit 3: Indexing (4)	
Competences Expected: b,d,i	
Topics	**Learning Outcomes**
• The impact of indices on query performance • The basic structure of an index • Keeping a buffer of data in memory • Creating indexes with SQL • Indexing text • Indexing the web (e.g., web crawling)	• Generate an index file for a collection of resources [Usage] • Explain the role of an inverted index in locating a document in a collection [Usage] • Explain how stemming and stop words affect indexing [Usage] • Identify appropriate indices for given relational schema and query set [Usage] • Estimate time to retrieve information, when indices are used compared to when they are not used [Usage] • Describe key challenges in web crawling, eg, detecting duplicate documents, determining the crawling frontier [Usage]
Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unit 4: Relational Databases (14)	
Competences Expected: b,d,i	
Topics	**Learning Outcomes**
• Mapping conceptual schema to a relational schema • Entity and referential integrity • Relational algebra and relational calculus • Relational Database design • Functional dependency • Decomposition of a schema; lossless-join and dependency-preservation properties of a decomposition • Candidate keys, superkeys, and closure of a set of attributes • Normal forms (BCNF) • Multi-valued dependency (4NF) • Join dependency (PJNF, 5NF) • Representation theory	• Prepare a relational schema from a conceptual model developed using the entity- relationship model [Usage] • Explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key) [Usage] • Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division) [Usage] • Write queries in the relational algebra [Usage] • Write queries in the tuple relational calculus [Usage] • Determine the functional dependency between two or more attributes that are a subset of a relation [Usage] • Connect constraints expressed as primary key and foreign key, with functional dependencies [Usage] • Compute the closure of a set of attributes under given functional dependencies [Usage] • Determine whether a set of attributes form a superkey and/or candidate key for a relation with given functional dependencies [Usage] • Evaluate a proposed decomposition, to say whether it has lossless-join and dependency-preservation [Usage] • Describe the properties of BCNF, PJNF, 5NF [Usage] • Explain the impact of normalization on the efficiency of database operations especially query optimization [Usage] • Describe what is a multi-valued dependency and what type of constraints it specifies [Usage]
Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unit 5: Query Languages (12)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Overview of database languages • SQL (data definition, query formulation, update sub-language, constraints, integrity) • Selections • Projections • Select-project-join • Aggregates and group-by • Subqueries • QBE and 4th-generation environments • Different ways to invoke non-procedural queries in conventional languages • Introduction to other major query languages (e.g., XPATH, SPARQL) • Stored procedures	• Create a relational database schema in SQL that incorporates key, entity integrity, and referential integrity constraints [Usage] • Use SQL to create tables and retrieve (SELECT) information from a database [Usage] • Evaluate a set of query processing strategies and select the optimal strategy [Usage] • Create a non-procedural query by filling in templates of relations to construct an example of the desired query result [Usage] • Embed object-oriented queries into a stand-alone language such as C++ or Java (eg, SELECT ColMethod() FROM Object) [Usage] • Write a stored procedure that deals with parameters and has some control flow, to provide a given functionality [Usage]
Readings : [Die01], [EN04], [Cel05], [KS02]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. PLANNING

DATE	TIME	SESSION TYPE	PROFESSOR
See at EDU	See at EDU	See at EDU	See at EDU

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

[CJ11] Date C.J. *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media, 2011.

[Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.

[EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition.* Addison Wesley, 2004.

[ER15] Jim Webber Emil Eifrem and Ian Robinson. *Graph Databases.* 2nd. O'Reilly Media, 2015.

[KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos.* McGraw-Hill, 2002.

[RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition.* Morgan Kaufmann, 2004.

[RG03] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems.* 3rd. McGraw-Hill, 2003.

[SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition.* Morgan Kaufmann, 2004.

[WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition.* Springer, 2001.