Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti
Facultatea de Electronică, Telecomunicaţii şi
Tehnologia Informaţiei

COURSE DESCRIPTION

1. Program identification information

1.1 Higher education institution	**National University of Science and Technology Politehnica Bucharest**
1.2 Faculty	**Electronics, Telecommunications and Information Technology**
1.3 Department	**Electronic Devices, Circuits and Architectures**
1.4 Domain of studies	Electronic Engineering, Telecommunications and Information Technology
1.5 Cycle of studies	Bachelor/Undergraduate
1.6 Programme of studies	Microelectronics, Optoelectronics and Nanotechnologies

2. Date despre disciplină

2.1 Course name (ro) (en)				Baze de date Databases			
2.2 Course Lecturer				S.l./Lect. Dr. Vlad-Alexandru Grosu			
2.3 Instructor for practical activities				S.l./Lect. Dr. Vlad-Alexandru Grosu / Drd. Enache Ștefan-Iulian			
2.4 Year of studies	2	2.5 Semester	II	2.6. Evaluation type	V	2.7 Course regime	Ob
2.8 Course type		D	2.9 Course code	04.D.04.O.020		2.10 Tipul de notare	Nota

3. Total estimated time (hours per semester for academic activities)

3.1 Number of hours per week	3	Out of which: 3.2 course	2	3.3 seminary/laboratory	1
3.4 Total hours in the curricula	42	Out of which: 3.5 course	28	3.6 seminary/laboratory	14
Distribution of time:					hours
Study according to the manual, course support, bibliography and hand notes Supplemental documentation (library, electronic access resources, in the field, etc) Preparation for practical activities, homework, essays, portfolios, etc.					28
Tutoring					2
Examinations					3
Other activities (if any):					0

3.7 Total hours of individual study	33.00
3.8 Total hours per semester	75
3.9 Number of ECTS credit points	3

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	• Fundamental concepts from Algebra (sets, functions) • Computer Programming and Computer Architectures • Internet Communications

4.2 Results of learning	• Provides details about the Oracle environment (e.g., installation, configuration, management), so that the basic concepts regarding SQL-specific syntax (namely PL-SQL) are acquired • Presentations and workshops focus on outlining correct and complete database specifications, especially for security purposes but also for optimizing database performance (so-called *SQL tuning*) • The main goal is to learn ways of designing and planning a database, in order to manage successfully and comprehensively any database-related project.

5. Necessary conditions for the optimal development of teaching activities (where applicable)

5.1 Course	Provision of an optical projector (video projector) together with all related accessories (power cables, data and video signal cables, remote control).
5.2 Seminary/ Laboratory/Project	**Mandatory attendance** at laboratory activities (requirement in accordance with the *Internal Regulations* for conducting activities at UPB).

6. General objective *(Reffering to the teachers' intentions for students and to what the students will be thought during the course. It offers an idea on the position of course in the scientific domain, as well as the role it has for the study programme. The course topics, the justification of including the course in the currcula of the study programme, etc. will be described in a general manner)*

- Course:

- Acquiring domain-specific terminology;
- Developing the skills to learn and identify the characteristics of real-world objects in order to model (translate) them into concepts specific to databases.
- Knowing the traditional programming language, SQL.
- Developing algorithmic thinking and using it in analyzing proposed problems.

- Laboratory:

- Developing and writing short programs (scripts) in SQL/PL-SQL necessary for solving typical database applications.
- Identifying the most suitable techniques in query design, in implementing views, and in choosing security techniques to protect data when interacting with real database users.

7. Competences *(Proven capacity to use knowledge, aptitudes and personal, social and/or methodological abilities in work or study situations and for personal and proffesional growth. They refflect the empolyers requirements.)*

Specific Competences	C3: Applying basic knowledge, concepts, and methods regarding computer system architecture, microprocessors, microcontrollers, programming languages and techniques.

Transversal (General) Competences	CT1: Methodical analysis of problems encountered in activity, identifying elements for which established solutions exist, thus ensuring the fulfillment of professional tasks; CT3: Adapting to new technologies, professional and personal development, through continuous training using printed documentation sources, specialized software and electronic resources in Romanian and at least one internationally circulated language.

8. Learning outcomes *(Synthetic descriptions for what a student will be capable of doing or showing at the completion of a course. The learning outcomes reflect the student's acomplishments and to a lesser extent the teachers' intentions. The learning outcomes inform the students of what is expected from them with respect to performance and to obtain the desired grades and ECTS points. They are defined in concise terms, using verbs similar to the examples below and indicate what will be required for evaluation. The learning outcomes will be formulated so that the correlation with the competences defined in section 7 is highlighted.)*

Knowledge	*The result of knowledge aquisition through learning. The knowledge represents the totality of facts, priciples, theories and practices for a given work or study field. They can be theoretical and/or factual.* • **Acquires** knowledge about data security, scalability, and performance optimization. • **Understands** the fundamental elements of database design, including: entity–relationship modeling, normalization, and data indexing. • **Learns** about different types of databases, such as relational databases (e.g., SQL), NoSQL databases (e.g., MongoDB), and NewSQL databases. • **Understands** query languages, data manipulation languages, and database administration languages (from the perspective of the studied database management system). • **Analyzes** the design requirements and the steps necessary for designing an application itself (as a consequence of the aforementioned statements).
Skills	*The capacity to apply the knowledge and use the know-how for completing tasks and solving problems. The skills are described as being cognitive (requiring the use of logical, intuitive and creative thinking) or practical (implying manual dexterity and the use of methods, materials, tools and intrumentation).* • **Identifies and describes** the concepts of *Structured Query Language* (SQL), the most common database manipulation language. • **Learns** to write programs with packages, debugging procedures, triggers, and database structures using SQL. • **Uses** data models and objects, relational algebra, relational data models, and programs applications. • **Designs and implements** normalized databases using database reports and by creating forms and tables. • **Creates** systems of tables with (and based on) screens, updates, and reports.

Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* • **Designs and implements** database systems that meet the organization's needs. • **Manages** the data stored in the database. • **Monitors and optimizes** the performance of the database it maintains. • **Creates and maintains** backup and recovery procedures to ensure that database data is protected against loss or alteration (Eng. *corruption*). • **Upgrades** (Eng. *upgrade*) the database software and **performs** regular maintenance tasks to ensure the database runs smoothly. • **Ensures** that the database is secured against unauthorized access and data breaches. • **Ensures** that the organization's databases comply with any relevant regulations and industry standards.

9. Teaching techniques *(Student centric techniques will be considered. The means for students to participate in defining their own study path, the identification of eventual fallbacks and the remedial measures that will be adopted in those cases will be described.)*

- Course

- Interactivity with students through the applied part associated with the taught methods.
- Time slots are reserved for presenting useful examples that students encounter in other subjects: economic transactions, Internet-based communications. The comments on these examples presented during the course provide students with the necessary support for laboratory hours and the project and also ensure the interdisciplinarity specific to databases.
- Being an IT (programming) discipline, the distinction between course and laboratory activities is very fine. The concepts presented in the course are revisited in the laboratory (with an obvious practical emphasis) as well as during consultation sessions (the latter are necessary to prepare students for in-term tests).

- Laboratory

- A brief review of the theoretical notions specific to the work.
- The purpose of any laboratory work is to manage to design and write *complete scripts*, starting from the examples provided in the laboratory guide.
- Complete practical problems are presented — this approach offers the possibility to perform comparative evaluations and make the most appropriate decisions in practical situations encountered when designing real problems.
- Accepting databases as a real programming option by using a standard language such as SQL or PL-SQL, using HTML or PHP to develop graphical user interfaces.
- Control of distributed databases (based on the client–server model) and the use of database management systems (MySQL, Oracle).
- Designing a complete database. In the laboratory, the *project workflow style* is adopted. Each student can carry out activities starting from a work plan with a clear perspective, thus being able to consider the *proper management* of working time.

10. Contents

COURSE		
Chapter	**Content**	**No. hours**

1	= Introduction = The concept of ‘database’. Modeling. Abstraction. Definition of the database. Definition of the database management system. Database architecture. Evolution of databases.	2
2	= Types of data models = Hierarchical data model. Network data model. Relational data model. Entity–relationship model. Object data model. Object–relational data model. Comparison of data models.	2
3	= The relational data model = The notion of relation. Basic notions of the relational model. Intension of the relational data model. Extension of the relational data model. Logical schema of the relational database. Presentation of the relational model in DDL (data description language). Fully relational systems.	2
4	= Relational data manipulation languages (DML) = Relational algebra. Operators of relational algebra. Formulating queries in relational algebra. Limitations of relational algebra. Relational calculus. Tuple relational calculus (TRC). Domain relational calculus (DRC). QBE language. Operators in the QBE language. Formulating queries in the QBE language. Defining data in QBE.	2
5	= SQL+ language (part I) Syntax of SELECT constructs. Formulating queries in SQL. Operators in SQL. Functions in SQL.	3
6	= SQL+ language (part II) Commands in SQL. Insert, delete, and update operations. Data definition in SQL. Creating relations. Creating views. Creating indexes.	3
7	= PL/SQL language = Advantages of the PL/SQL language. Structure of the PL/SQL language. Declaring PL/SQL variables. Control structures. Cursors, PL-SQL constructs.	3
8	= Normalization of relations = Functional dependencies. Decomposition of relation schemas. First normal form. Second normal form. Third normal form. Multivalued dependencies. Fourth normal form. Fifth normal form.	3
9	= Transaction management = The notion of a transaction. Conditions for transaction termination. Properties of transactions. Formalization of transactions. Concurrency control.	2
10	= Database security = Protecting the database. Creating users. Creating roles. User privileges. Object privileges. Using views. Using synonyms instead of views.	2
11	= Creating a WEB site and querying databases on the Internet (part I) = Elements of the HTML language. Elements of the PHP language. Elements of the Java language.	2

12	= Creating a WEB site and querying databases on the Internet (part II) = Creating scripts in PHP. Stages of designing a WEB site.	2
	Total:	28

Bibliography:

1. Şl. dr. ing. Vlad-Alexandru Grosu, ing. Ştefan-Iulian Enache - Baze de Date (ETTI, FAIMA) - suport electronic: https://archive.curs.upb.ro/2022/course/view.php?id=8673
2. Edward Honour, Paul Dalberth, Avi Kapalan, Abul Mehto, *ORACLE 8. Secrete*, Editura Teora, 1999.

LABORATORY

Crt. no.	Content	No. hours
1	Creating a relational Database in the Oracle environment in different domains (the database is unique for each student).	2
2	Establishing the database relations.	2
3	Logical schema of the database.	2
4	Queries on the database.	2
5	Creating scripts in PHP.	2
6	Creating a site for the database.	2
7	Final laboratory verification (individually held colloquium).	2
	Total:	14

Bibliography:

1. Şl. dr. ing. Vlad-Alexandru Grosu, ing. Ştefan-Iulian Enache - Baze de Date (ETTI, FAIMA) - suport electronic: https://archive.curs.upb.ro/2022/course/view.php?id=8673
2. Ioan Rusu, Vlad-Al. Grosu, Bogdan Nedelcu, Georgiana Muscalu - *Baze de date. Îndrumar de laborator. Mediul Oracle*, orice ediţie, Ed. MatrixRom, Bucureşti, 2016.
3. I. Rusu, M. Căpătoiu – *Îndrumar de laborator pentru mediul Oracle*, Editura Bren, 2004.
4. Edward Honour, Paul Dalberth, Avi Kapalan, Abul Mehto, *ORACLE 8. Secrete*, Editura Teora, 1999.

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade

11.4 Course	Correct identification of theoretical and practical contexts for applying the taught methods and techniques. Deepening the notions of mathematical analysis and algebra required for the course, with application in electronics (area of fundamental training). C4.1: Defining the concepts, principles, and methods used in the fields of: computer programming, high-level and specific languages, computer system architecture, programmable electronic systems, graphics, reconfigurable software architectures.	Two tests during the semester: partial and final. Once passed without error, each of the two tests brings 20 points (total: 40 points). Both the theoretical part corresponding to each test and the implementation part (conceptualization, abstraction) are covered.	40%
11.5 Seminary/laboratory/project	C1.4 Appropriate use of fundamental evaluation criteria and methods to identify, model, analyze, and qualitatively and quantitatively assess characteristic phenomena, processes, and theories, as well as to process and interpret the results of processes specific to the field of engineering and management. C4.5: Supporting and promoting an exam related to the architecture and functional principles of a functional software structure.	Laboratory activity is assessed throughout the semester. - Students can accumulate the following points: - 10-minute quizzes: 15 points - homework: 15 points - final examination: 30 points. The laboratory ends with a final examination, each student having individual access to the workstation. The student must: - write a short program (script) based on SQL to solve the proposed problem. - provide a short answer to a theoretical question that synthesizes the concepts presented during the semester.	60%
11.6 Passing conditions			

- The possibility to evaluate comparatively different solutions for the same problem and to be able to choose the best one in the given context, in order to design appropriate scripts for each real-life situation.
- The skills of accumulating and then identifying practical situations specific to the presented concepts as well as correctly applying them in the field of electronics are verified.
- Passing the subject requires the accumulation (with no imposed intermediate thresholds) of **at least 50 points** out of a total of 100 points.

12. Corroborate the content of the course with the expectations of representatives of employers and representative professional associations in the field of the program, as well as with the current state of knowledge in the scientific field approached and practices in higher education institutions in the European Higher Education Area (EHEA)

CT.3 Adapting to new technologies, personal and professional development throughout training by using printed documentation resources, dedicated software, and electronic resources, mainly in Romanian but also in another internationally circulated language (in this case — English).

The programming language used is established, standardized, and therefore effective and up to date. The most recent published version was in 1992 by ANSI, as the American national standard X3.135-1992, and by ISO/IEC as international standard 9075:1992. It was also adopted as a Federal Information Processing Standard (*Federal Information Processing Standard* = FIPS) for the U.S. federal government.

During the laboratory works, students use the SQL programming language. The database development and management environments are MySQL and Oracle (in the event of scaling to more demanding work environments), configured to use their own implementations of the SQL language, called SQL and PL/SQL, respectively.

The bibliographic material is available online and covers both fundamental works and more recent titles, including techniques specific to database management systems.

Graduates must be aware of the *responsibilities* of a database administrator, starting from the *job requirements* they will perform in private life. Their own decisions must be correlated with the company's policy and client respect, and can thus impact its image if not accompanied by the principles of professional ethics, which they will acquire through their work.
Without the supervision of a database administrator, application and system interruptions, downtime, and slowdowns will inevitably occur. These types of problems affect any enterprise and can negatively impact revenue, customer experiences, and business reputation.

For implementing a database in the *cloud*, although the database administrator is not responsible for installation, they must actively establish the appropriate configuration, access, and deployment options for use by the organization to which that cloud database belongs.

Last but not least, when problems arise, the database administrator is the central point for resolution, including troubleshooting, root-cause analysis, fine-tuning, and performance optimization of the tasks and programs that access that database.

Date	Course lecturer	Instructor(s) for practical activities

Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației

18.09.2025	S.l./Lect. Dr. Vlad-Alexandru Grosu	S.l./Lect. Dr. Vlad-Alexandru Grosu
		Drd. Enache Ștefan-Iulian
Date of department approval	Head of department	
	Prof. Dr. Claudius Dan	
Date of approval in the Faculty Council	Dean	
	Prof. Mihnea UDREA	