# Syllabus

## *Course Description*

| | |
|---|---|
| **Course Title** | Introduction to Databases |
| **Course Code** | 76209 |
| **Course Title Additional** | |
| **Scientific-Disciplinary Sector** | IINF-05/A |
| **Language** | German |
| **Degree Course** | Bachelor in Computer Science |
| **Other Degree Courses (Loaned)** | |
| **Lecturers** | Prof. Diego Calvanese,<br>Diego.Calvanese@unibz.it<br>https://www.unibz.it/en/faculties/engineering/academic-staff/person/3562 |
| **Teaching Assistant** | |
| **Semester** | First semester |
| **Course Year/s** | 2 |
| **CP** | 6 |
| **Teaching Hours** | 40 |
| **Lab Hours** | 20 |
| **Individual Study Hours** | 90 |
| **Planned Office Hours** | |
| **Contents Summary** | Students attending this course will have acquired the techniques and methods to address problems of database design, and to make use of the basic functionalities (definition, update, and querying of the database) of database management systems in the context of development and deployment of information systems. In addition, students will be able to develop applications that programmatically interact with a database management system. The course explicitly refers to relational databases and to the corresponding database management systems based on the SQL language. However, the taught methods and principles are of a more general nature, and can be applied also in those contexts where data models and |

| | |
|---|---|
| | database systems different from relational ones are adopted. |
| **Course Topics** | - Relational data model and relational algebra<br>- The SQL language<br>- Using SQL in database applications: API, embedded SQL<br>- The Entity Relationship model<br>- Conceptual database design<br>- Logical database design |
| **Keywords** | relational algebra, SQL, ER model, conceptual modeling, database design |
| **Recommended Prerequisites** | Students should have a solid mathematical foundation and be familiar with the basic programming concepts, and with basic data structures (arrays, lists, trees) and algorithms that operate on them (visits, search, updates). These prerequisites are covered in the following courses: Analysis, Introduction to Programming, and Object Oriented and Functional Programming . |
| **Propaedeutic Courses** | |
| **Teaching Format** | The course includes frontal lectures, exercises, and project work. |
| **Mandatory Attendance** | Attendance is not compulsory, but non-attending students have to contact the lecturer at the start of the course or before starting their studies and the project work to agree on the modalities of the independent study. |
| **Specific Educational Objectives and Learning Outcomes** | Knowledge and Understanding<br>- D1.4 Understand the key principles, the structures and the organization of relational databases and methods for designing and developing databases.<br><br>Applying knowledge and understanding<br>- D2.5 Be able to apply the own knowledge to the , design, development and testing of information systems which satisfy given requirements<br>-D2.12 Be able to design, develop and query relational databases<br><br>Ability to make judgments<br>- D3.1 Be able to collect and interpret useful data and to judge information systems and their applicability.<br>- D3.2 Be able to work autonomously according to the own level of knowledge and understanding. |

| | |
|---|---|
| | Communication skills<br>- D4.1 Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately.<br>- D4.5 Be able to work in teams for the realization of IT systems.<br><br>Learning skills<br>- D5.1 Have developed learning capabilities to pursue further studies with a high degree of autonomy.<br>- D5.2 Have acquired learning capabilities that enable to carry out project activities in companies, public institutions or in distributed development communities.<br>- D5.3 Be able to follow the fast technological evolution and to learn cutting edge IT technologies and innovative aspects of last generation information systems. |
| **Specific Educational Objectives and Learning Outcomes (additional info.)** | |
| **Assessment** | For both attending and non-attending students, assessment consists of a project aimed at testing the ability to apply knowledge and communicate effectively, carried out individually or as a group of two and presented orally. In addition, a written exam includes verification questions as well as questions designed to assess both knowledge and understanding, and the ability to apply that knowledge. |
| **Evaluation Criteria** | For both attending and non-attending students, the assessment consists of two components: a project (30% of the final mark) and a written exam (70% of the final mark). The project evaluates the ability to work in teams, apply knowledge in practical contexts, and summarize concepts in one's own words. The written exam assesses clarity of responses, the ability to recall principles and methods used in database systems, and the skill to apply knowledge through tasks such as developing and querying databases.<br><br>To carry out the project, students must agree with the lecturer on the topic and scope, preferably during the early stages of development, based on an initial draft of the conceptual schema, to minimize the risk of having to redo work. Attending students can |

| | |
|---|---|
| | discuss their project during dedicated lab sessions, while non-attending students must make arrangements during office hours or by scheduling an appointment with the lecturer.<br><br>The project is usually discussed two to three working days before the scheduled written exam, and must be submitted by the deadline announced on MS Teams, typically two days before the discussion date. In order to be admitted to the written exam, the project must be discussed and positively evaluated. Without passing the project, students are not allowed to take the written exam.<br><br>The written exam, which lasts at least two hours, requires students to design a database based on a given specification, formulate SQL queries, and possibly respond in writing to questions on topics covered in the course. To pass the course, both the project and the written exam must be passed. The final grade is calculated as a weighted average of the project (30%) and written exam (70%) marks. A positive project mark remains valid for all three regular exam sessions within the same academic year, meaning that if a student fails or does not sit the written exam, the project evaluation is retained and only the written exam needs to be retaken. |
| **Required Readings** | - course lecture notes, available on the Microsoft Teams course page;<br><br>- instructions for carrying out the project, available on the course Web page. |
| **Supplementary Readings** | Raghu Ramakrishnan and Johannes Gehrke. Database Management Systems. McGraw-Hill, Boston, 3rd edition, August 2002. ISBN 978-0-07-246563-1. |
| **Further Information** | - PostgreSQL (https://www.postgresql.org)<br>- RADB (Java relational algebra interpreter) (https://users.cs.duke.edu/~junyang/radb/)<br>- Java (https://openjdk.org) |
| **Sustainable Development Goals (SDGs)** | Quality education |