GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Databases, PG_00068274		
Field of study	Automatic Control, Cybernetics and Robotics		
Date of commencement of studies	October 2025	Academic year of realisation of subject	2027/2028
Education level	first-cycle studies	Subject group	Optional subject group Subject group related to scientific research in the field of study
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	3	Language of instruction	Polish
Semester of study	5	ECTS credits	3.0
Learning profile	general academic profile	Assessment form	exam
Conducting unit	Department of Biomedical Engineering -> Faculty of Electronics Telecommunications and Informatics -> Wydziały Politechniki Gdańskiej		
Name and surname of lecturer (lecturers)	Subject supervisor	mgr inż. Piotr Dalka	
	Teachers	mgr inż. Piotr Dalka	

Lesson types and methods of instruction

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	15.0	0.0	30.0	0.0	0.0	45

E-learning hours included: 0.0

Learning activity and number of study hours

Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
Number of study hours	45	2.0	28.0	75

Subject objectives

The aim of the course is to familiarize students with both fundamental and advanced topics related to databases, particularly relational database systems. During the classes, students will learn key concepts such as the relational model, data types, and methods of organizing and storing information in a database. They will gain knowledge and skills in creating and optimizing queries, joining data from multiple tables, modeling complex relationships, as well as modifying and designing database schemas. Integrity rules, transaction management principles, and performance optimization mechanismssuch as indexes and query planswill be discussed. Students will also learn how to use SQL language extensions, including views, functions, and triggers. The course includes an overview of the most popular relational database management systems and a comparison of their functionalities. It will also cover the integration of databases with programming applications, including the use of ORM libraries. The course also introduces NoSQL databases, their characteristics, applications, and the key differences compared to the relational approach. Upon completing the course, students will be able to design, implement, and efficiently manage relational databases, as well as make informed decisions regarding the choice of database technology for various engineering projects.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_U04] can apply knowledge of programming methods and techniques as well as select and apply appropriate programming methods and tools in computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study	Student uses SQL language to manage relational database Student uses SQL language inside other language Student analyses data using SQL language	[SU1] Assessment of task fulfilment [SU4] Assessment of ability to use methods and tools
[K6_W04] knows and understands, to an advanced extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, and organisation of systems using computers or such devices	Student knows basic database terms Student knows and identifies data models Student designs realtional databases	[SW1] Assessment of factual knowledge [SW3] Assessment of knowledge contained in written work and projects

Subject contents

The course will include 15 lectures related to the following topics (the list of topics is not exhaustive):

- characteristics of relational databases, basic concepts
- data types
- generating SQL queries: filtering, aggregation, grouping, sorting
- types of relationships between tables, queries aggregating data from multiple tables
- modeling many-to-many relationships and inheritance
- data normalization
- editing data in the database
- defining and modifying the database schema
- foreign key
- database transactions, transaction isolation levels, ACID
- database indexes types, advantages, and disadvantages of use
- query and database schema optimization, query execution plan
- views, SQL functions, triggers
- the most popular SQL database engines, their features, and comparison
- SQL databases in Python programs, popular ORM libraries
- NoSQL databases: differences compared to SQL, types of NoSQL databases, applications, and example engines

Lab Exercises

1. SQL Fundamentals and Schema Navigation -- Learning about SELECT syntax, column projection, and system metadata exploration in PostgreSQL and MySQL
2. Data Types and Conversions -- Analyzing the impact of data type selection on query performance and automatic conversion behavior.
3. Filtering and Ordering Results -- Building complex WHERE predicates with logical operators and using ORDER BY with LIMIT.
4. Aggregation and Grouping -- Creating queries with GROUP BY and HAVING for analyzing sales statistics in a data warehouse.
5. Table Joins: INNER JOIN, OUTER JOIN -- Implementing various types of table joins for order analysis and customer relationships.
6. Subqueries and Temporary Views -- Using correlating subqueries, CTE definitions, and creating views to streamline reporting.
7. Many-to-many relationships with link tables -- Designing an intermediary table for linking users to system roles and writing appropriate queries.
8. Inheritance and multi-model mapping -- Implementing tabular inheritance using an electronics product hierarchy as an example and analyzing the consequences.
9. Normalization from 1NF to 3NF -- Identifying anomalies in the denormalized schema and performing the normalization process step by step.
10. Data manipulation: INSERT, UPDATE, DELETE -- Ensuring data consistency through restrictive CHECK clauses and the use of CASCADE operations.
11. Schema definitions and modifications: DDL -- Creating CREATE TABLE, ALTER TABLE, and DROP TABLE scripts with complete migration documentation.
12. Transactions and isolation levels -- Experiencing phantom read and non-repeatable read anomalies during parallel database sessions.
13. Indexes and Query Execution Plans -- Creating different types of indexes and comparing query plans, assessing selectivity and cost.
14. Materialized Views, Functions, and Triggers -- Implementing an integrity trigger and a real-time data aggregation function.
15. Integration with Python, ORM, and NoSQL -- Comparing declarative and imperative mapping styles using SQLAlchemy and comparing the resulting solution with an alternative approach using MongoDB.

Prerequisites and co-requisites

- has basic knowledge of mathematics, including algebra and logic
- knows the basic concepts of programming language syntax (C/C++, Python)

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
laboratory achievements	60.0%	50.0%
exam	60.0%	50.0%

Recommended reading	Basic literature	• Alan Beaulieu. *Wprowadzenie do SQL. Jak generować, pobierać i obsługiwać dane*. Wydanie III. Helion, 2021. • Anthony DeBarros. *SQL w praktyce. Jak dzięki danym uzyskiwać cenne informacje*. Wydanie II. Helion, 2024.
	Supplementary literature	• Anthony Molinaro, Robert de Graaf. *SQL. Zapytania i techniki dla bazodanowców. Receptury.* Wydanie II. Helion, 2021. • Alex Petrov. *Baza danych od środka. Analiza działania rozproszonych systemów danych.* Helion, 2024.
	eResources addresses	
Example issues/ example questions/ tasks being completed	• What data types can be used to store text data in relational databases? • What are the different ways of modeling inheritance in relational databases? • What is the difference between a LEFT JOIN and an INNER JOIN? • What are the advantages and disadvantages of using indexes? • Write an SQL query that, based on a given database schema, generates a report containing the specified columns. • Optimize the given SQL query.	
Work placement	Not applicable	

Document generated electronically. Does not require a seal or signature.