COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

COURSE SYLLABUS 2026-2027

GENERAL INFORMATION

Course information	
Name	Database Design / *Diseño de Bases de Datos*
Code	DCIA-MUIINT-111
Main program	Máster Universitario en Industria Inteligente
Offered in	Máster Universitario en Industria Inteligente [1st year] Máster Universitario en Ingeniería Industrial + Máster Universitario en Industria Inteligente [1st year] Máster Universitario en Ingeniería de Telecomunicación + Máster Universitario en Industria Inteligente [1st year]
Level	Official Master's Degree
Semester	1st (Fall)
Credits	4.5 ECTS
Type	Compulsory
Language	English
Department	Computer Science and Artificial Intelligence
Coordinator	Juan Luis Paz Rojas

Instructor	
Name	Juan Luis Paz Rojas
Department	Computer Science and Artificial Intelligence
e-mail	jlpaz@comillas.edu
Office hours	Arrange an appointment through email.
Lab instructor	
Name	Manuel Alvar Miró
Department	Electronics, Control and Communications
e-mail	malvar@comillas.edu
Office hours	Arrange an appointment through email.

COURSE SPECIFIC INFORMATION

Contextualization of the course

Contribution to the professional profile of the degree

Software has two dimensions: logic, which computes and is then gone, and data, which remembers and must stay right for years. This course owns the data dimension. It teaches students to design a relational database that says what the business means, to build it so that the database itself guarantees what has to stay true, and to question it: to ask a precise question of data, and to tell a right answer from a merely plausible one.

Data outlives the programs that produce it, and its quality decides what an organization is able to know about itself. The age of AI raises that stake rather than lowering it: analytics, automation and language models are worth what the data underneath them is worth, and a schema that does not say what it means yields confident answers that are quietly wrong. The engine used is PostgreSQL, but the subject is databases: standard SQL and relational reasoning transfer to any engine, and what is specific to one product is flagged as such.

Graduates of this master build for organizations that need agility rather than hyperscale infrastructure, and most “the database is the problem” stories turn out to be “we did not know the tool”. The syllabus is therefore built to close the usual wrong reasons for abandoning a relational database — slow queries, connected data, semi-structured data, text search, history, dates and time zones, growth past one machine — each paired with the capability that removes it. It closes on the honest opposite question: when leaving genuinely is the right decision, and how to tell that case from a campaign.

Prerequisites
Students should be familiar with undergraduate-level programming, preferably in Python. No previous knowledge of databases, SQL or the relational model is assumed: every idea is built from zero.

Learning outcomes

Knowledge

CO3. Design and query relational databases for deployment in local and cloud environments, defining schemas, normalization, and indexing that ensure integrity, scalability, and efficiency, as well as being familiar with non-relational database technologies.

Diseñar y consultar bases de datos relacionales para su despliegue en entornos locales y en la nube, definiendo esquemas, normalización e indexado que garanticen integridad, escalabilidad y eficiencia, así como estar familiarizado con tecnologías de bases de datos no relacionales.

Competencies

CP2. Make informed decisions in contexts of uncertainty, using data, models, and technical criteria to solve real industrial problems.

Tomar decisiones informadas en contextos de incertidumbre, utilizando datos, modelos y criterios técnicos para resolver problemas industriales reales.

CP4. Collaborate effectively in multidisciplinary teams and communicate results, conclusions, limitations, risks, and benefits of proposed solutions rigorously and clearly to both expert and non-expert audiences.

Colaborar eficazmente en equipos multidisciplinares y comunicar con rigor y claridad los resultados, conclusiones, limitaciones, riesgos y beneficios de las soluciones propuestas a audiencias expertas y no expertas.

CONTENTS

Contents

Theory

Block A — Foundations and core operations (Units 1–4)

Unit 1. Foundations and the meaning of NULL — software's two dimensions, logic and data · what a database solves over a spreadsheet, and what does not belong in one · SQL as a declarative ISO standard · the relational model, primary and alternate keys, and reasoning about identity · the types the course uses, exact numerics against floating point, and timestamptz as an instant · NULL, three-valued logic and the empty string · reading a crow's-foot ER model · running SQL from Python with parameterized queries, and the transaction a connection opens

Unit 2. Asking questions with SELECT — projection and computed expressions · text functions · WHERE predicates, and the traps a nullable column sets · ordering, and why text does not sort the human way · paging with LIMIT and OFFSET · the logical order of execution

Unit 3. Combining data: joins, set operations, CTEs, views and recursion — why join, and the fan-out effect · inner and outer joins, and the anti-join · how the shape of a relationship decides the join · set operations · CTEs and views, and what a view does not buy · recursive queries over a hierarchy of unknown depth

Unit 4. Creating tables and changing data — CREATE TABLE, types, defaults and the four kinds of constraint · ALTER TABLE on a table that already holds data · set-based INSERT, UPDATE and DELETE, and the N+1 problem · RETURNING and upsert · the missing-WHERE danger, and the transaction that catches it · referential integrity felt from the DELETE a foreign key refuses

Block B — Database design (Units 5–6)

Unit 5. Database design I: modeling by meaning — the conceptual, logical and physical levels of a model · identify the concept each table is about, and that time is part of it · the three data natures — current state, events and analytical — and the rule that one table holds one nature · the Draft, Request and Historical patterns, and why a soft-delete flag is a trap · the three degrees of a relationship — optional, mandatory and existential — and weak entities · cardinality and the junction table · building a crow's-foot ER model and classifying every table of it
Unit 6. Database design II: normalization — redundancy, and the update, insert and delete anomalies · functional dependencies and lossless decomposition · the normal forms from 1NF to 5NF, used as a check rather than as a recipe · what the ladder cannot see: time, and a weak entity's key · the pairing with the design method of Unit 5, and the order in which the check is run

Block C — Constraints, integrity and aggregation (Units 7–8)

Unit 7. Constraints, audit columns and aggregation — CHECK in its three shapes · multi-column UNIQUE · weak entities in DDL, and the two-column foreign key that follows · which UNIQUE a foreign key may aim at · naming constraints, and adding one to a table that already holds rows · structural integrity against a business rule, and validation as defence in depth · control and audit columns, which of them a table earns, and why the decision has a deadline · aggregates, GROUP BY, NULL in aggregates, HAVING, and the two join traps
Unit 8. Transactions and concurrency — what a transaction is, and atomicity · snapshots, and what an isolation level is · the three anomalies, met as events before they are named · delimiting a transaction from Python · the lost update and its three cures · row locks and deadlocks · ACID as a name for guarantees already understood, and how big one job should be

Block D — Scale, and the system around it (Units 9–10)

Unit 9. Indexing, subqueries and searching text — what an index is, and the ones a schema already has · why a foreign key has none · composite indexes, where the column order is the design · when to create one, and reading EXPLAIN and EXPLAIN ANALYZE · subqueries in their four shapes, and NOT IN against NOT EXISTS · searching text by word — full-text search, stemming and ranking — and by meaning, with embeddings and vector distance · what the pair is for: answering a question with AI over your own rows
Unit 10. Beyond one database — returning a parent with its children as JSON · a JSON document in a column, and what the database stops promising · rows against columns, and the second copy a system that needs both answers keeps · transactions across a boundary: the dual-write problem and the transactional outbox · where a database may be cut in two · a system that owns rows your schema names · growing up before out, read replicas, CQRS and replication lag · source of truth against system of record, and the maturity of an organization

Block E — Beyond the relational database (Unit 11)

Unit 11. Beyond the relational database — what this database is genuinely bad at, and the crafts that belong to somebody else · what document stores, key-value stores, graph databases and search engines actually are · what lies outside, and what the database already gives you · where you are standing: a size arrived at by arithmetic, and which kind of system · five tests for what you are offered · failures worth being able to name · why the pressure exists · and where leaving genuinely is the right answer

Laboratory

Lab 1. Setting up the environment — the tools; PostgreSQL in a single Docker container; a database of the student's own with its encoding, its sorting rule and its time zone stated rather than inherited; the Python environment; and a first parameterized query. Worked individually before the first laboratory session, and not marked.
Lab 2. A plant's questions, answered from Python — filtering, sorting and paging, every value bound as a parameter, and the column the user sorts by taken from an allow-list.
Lab 3. Multi-table questions — deciding what each answer is about; the gaps only an outer join shows; set operations and a CTE; changing a view the whole shop floor reads; and walking a parts explosion of unknown depth.
Lab 4. A distribution center's day — building a small register; loading delivery notes and resolving the supplier's codes against our own tables; a load that is safe to run twice; and changing and removing data set-based, timed against one call per row.

Lab 5. Design: a clinic that has run out of paper — turning what somebody says into a schema. What is handed in is the SQL that creates it, with the comments that justify the design.
Lab 6. Design: a festival that ran on somebody's memory — the same exercise in a world of its own, where the weight is in the keys and the requirements arrive being corrected.
Lab 7. A hotel group whose database is not keeping its design — putting the rules in, including onto tables that already hold rows; deciding what each row records about itself; and answering the group's questions about groups of rows.
Lab 8. A theatre box office, two connections at once — an atomic sale; the anomalies a snapshot is subject to; and the lost update, produced and then cured three ways.
Lab 9. A parcel courier with a stopwatch — measure, index, measure again; the subquery whose cost was never the subquery; and searching free text by word instead of by letters.

Final project

Working in small teams, students will design a database for a case of their own — a longer case than those of Units 5 and 6 — implement it in PostgreSQL, load data into it from files, and produce a set of reports, each one an SQL query and its result. The design and the reasoning behind it are what is assessed. The brief is issued during the course.

TEACHING METHODOLOGY

General methodological aspects

The course runs as two parallel tracks, theory and laboratory, in sessions of two hours: twenty-two sessions in all, meeting twice a week during the first part of the term and once a week thereafter. Each theory session is applied by its matching laboratory session on the same topic, so that every concept is met, reasoned about, and then used.

Every idea is built bottom-up: the concrete problem first, then the need for the concept, then the concept and its name. No previous knowledge of databases is assumed, and terminology arrives plain word first, precise term second. No idea arrives whole: each is introduced when its need is felt and then grown across the course, which is why the course interleaves — read, change, design, query — rather than marching straight through. And wherever the subject is data, the data itself is put on the page, and the rule is named afterwards, rather than being handed over as a formalism to be trusted.

In-class activities

- **Lectures and problem-solving sessions:** The lecturer introduces the concepts of each unit and works through examples on data shown on the page, raising open questions to foster discussion and short exercises to be solved in class. Each unit ships two documents, slides and a handout, and the handout is the reference of record: it covers the same ground in full sentences and is written to be read without the lecturer.
- **Laboratory sessions:** Worked individually on the students' own machines, on a case that somebody describes rather than on a specification. Both instructors are present throughout, so that problems are resolved as they appear. Each session is self-contained: it ships its own database, its own data, and the file the student edits.
- **Tutoring** for groups or individual students will be organized upon request.

Out-of-class activities

- Personal study of the course material and resolution of the proposed exercises.
- Finishing and handing in each laboratory assignment.
- Development of a final project in small teams.

STUDENT WORK-TIME SUMMARY

IN-CLASS HOURS			
AF1. Expository and participatory lectures	**AF2. Continuous assessment tests**	**AF3. Laboratory practicals**	**AF4. Projects and assignments**
23	2	12	8

OUT-OF-CLASS HOURS		
AF3. Laboratory practicals	**AF4. Projects and assignments**	**AF7. Personal study**
24	**24**	**42**
	ECTS credits:	**4.5 (135 hours)**

EVALUATION AND GRADING CRITERIA

The use of AI to produce entire assignments or significant parts of them, without citing the source or tool used, or without explicit permission in the assignment description, will be considered plagiarism and will be subject to the University's General Regulations.

Evaluation activities	Grading criteria	Weight
Midterm exam	▪ Understanding of the theoretical concepts. ▪ Application of these concepts to problem-solving.	20%
Final exam	▪ Understanding of the theoretical concepts. ▪ Application of these concepts to problem-solving.	30%
Lab assignments — Labs 2, 3, 4, 7, 8 and 9	▪ Application of theoretical concepts to real problem-solving. ▪ Correctness, and reasoning set-based rather than row by row. ▪ Attitude and effort: initiative and proactive work will be encouraged.	10%
Lab assignments — design (Labs 5 and 6)	▪ Ability to turn what somebody describes into a data model. ▪ Written justification of the design decisions taken.	10%
Final project	▪ Problem analysis and design. ▪ Justification of the design. ▪ Quality of the project implementation. ▪ Teamwork ▪ There will be an intra-group evaluation method to differentiate among team members.	30%

Grading

Regular assessment

- **Theory** will account for 50%, of which:
  - Midterm exam: 20%
  - Final exam: 30%
- **Laboratory** will account for the remaining 50%, of which:
  - Lab assignments: 20%
  - Final project: 30%

In order to pass the course, both theory and laboratory marks must be greater or equal to 5 out of 10 points. Otherwise, the final grade will be the lower of the two marks.

Retake

There will be a retake exam that will replace the final exam of the regular assessment period. As long as the laboratory has a passing grade, all the remaining marks will be preserved; otherwise, a new individual project will have to be developed, and all failed lab assignments repeated. The final grade will be computed as in the regular assessment period and under the same restrictions.

Course rules

- Class attendance is mandatory according to Article 93 of the General Regulations (Reglamento General) of Comillas Pontifical University and Article 6 of the Academic Rules (Normas Académicas) of the ICAI School of Engineering. Not complying with this requirement may have the following consequences:
  - Students who fail to attend more than 15% of the lectures may be denied the right to take the final exam during the regular assessment period.
  - Regarding laboratory, absence to more than 15% of the sessions can result in losing the right to take the final exam of the regular assessment period and the retake. Missed sessions must be made up for credit.
- Students who commit an irregularity in any graded activity will receive a mark of zero in the activity and disciplinary procedure will follow (cf. Article 168 of the General Regulations (Reglamento General) of Comillas Pontifical University).

Guidelines for the use of generative artificial intelligence (AI)

- **Exams.** The use of generative artificial intelligence models or programming assistants is strictly prohibited in the exams. These activities must reflect exclusively the student's own knowledge and individual work.
- **Laboratory and final project.** The use of AI-based programming assistants and generative language models is permitted under the following conditions:
  - These tools may be used as support for understanding technical concepts, obtaining suggestions on how to approach the proposed exercises, and generating code snippets or initial drafts.
  - Their use must always be complementary and must not replace the student's individual work. Submitting automatically generated content as one's own, without proper understanding, review, and adaptation, is not allowed.
  - Any relevant content generated wholly or partially using these tools must be explicitly cited, clearly indicating which parts were generated with AI and which tools were used. The sequence of prompts must be included with the submission, either inside the file that is handed in or as a separate file.
  - Instructors reserve the right to ask oral questions regarding content generated with AI assistance to assess the student's understanding. Failure to explain or justify such content may negatively impact the grade for the activity.
- The responsible use of these tools is encouraged as a means of supporting individual study (e.g., to clarify concepts, generate additional exercises, or receive feedback). However, students should be aware that responses generated by AI models may contain errors, and it is their responsibility to critically assess and verify the information provided.

WORK PLAN AND SCHEDULE

Activities	Date/Periodicity	Deadline
Lectures	Weekly	–
Laboratory sessions	Weeks 2 through 11	–
Review and self-study of the concepts covered in the lectures	Weekly	–
Midterm exam	Week 9	–
Final project development	From Week 10	The last week of the semester
Final exam	Official ordinary evaluation period	–

BIBLIOGRAPHY AND RESOURCES

Basic references

- Slides and handout prepared by the lecturer, one pair per unit (available in Moodle). The handout is the reference of record: it covers the same ground as the slides in full sentences and is written to be read without the lecturer.
- The runnable SQL that ships with each unit: the script that creates its own database, the one that builds its tables and sample data, and the one holding every example query the unit prints, with the result it should return.

Complementary references

- E. Rivero, L. Martínez, and I. Alonso, *Bases de datos relacionales: Fundamentos y diseño lógico*, Colección Ingeniería 20, 1st Ed. Universidad Pontificia Comillas, 2005 [In Spanish]. ISBN-13: 978-8-484-68172-4
- R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th Ed., Pearson Education, 2016. ISBN-13: 978-1-292-09761-9
- E. Matthes, *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*, 3rd Ed., No Starch Press, 2022. ISBN-13: 978-1-718-50270-3
- PostgreSQL 18 documentation, [Online]. Available: https://www.postgresql.org/docs/18/
- Psycopg 3 documentation, [Online]. Available: https://www.psycopg.org/psycopg3/docs/

In compliance with current legislation on the **protection of personal data**, we inform and remind you that you can check the privacy and data protection terms you accepted at registration by entering this website and clicking "download".
https://servicios.upcomillas.es/sedeelectronica/inicio.aspx?csv=02E4557CAA66F4A81663AD10CED66792