DHBW Loerrach

Baden-Wuerttemberg
Cooperative State University

DB202: NoSQL

Prof Dr Veit U.B. Schenk, schenkv@dhbw-loerrach.de

NoSQL - NoLecture

- **eduScrum** is an adaptation of the Scrum framework for use in **education**. It applies agile principles to classroom settings, aiming to increase **student autonomy, collaboration, and motivation**.
- **Originator**: Willem Vermaak, a Dutch chemistry teacher, adapted Scrum to the classroom around 2011.
- **Core idea**: Teachers act more like **Product Owners** (guiding goals), and students form **self-organized teams** that work through **Sprints** to complete learning goals.
- **Artifacts & roles**: eduScrum uses a **flip** (eduScrum board), **Sprint planning**, **reviews**, **retrospectives**, and student **team roles** (e.g., facilitator, timekeeper, quality guard).
- The “Product” is **learning.**
- The focus is less on software and more on **personal growth**, **teamwork**, and **reflective learning**.
- **No“done for you”**-notes in Moodle!

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

Capstone Project (Portfolioprüfung)

- LLM Advisor for **Database System Design**
- Your team (3-4 students) will design and implement a **Retrieval-Augmented Generation (RAG) system** where the LLM acts as a **database expert advisor**.
  The system's purpose is to support **executives and IT professionals** in exploring and deciding on **database solutions tailored to their specific needs**.
- The range of needs is deliberately broad:
  - Some companies may start with nothing more than Excel spreadsheets.
  - Others may already operate with data warehouses or full data lakes.
  - Some want to integrate siloed systems into a unified data layer.
  - Others may be ready to build advanced RAG systems for customer support or decision automation.

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

Database System Design

DB Topic	System Design Angle
Indexing	“Given this scale and access pattern, which index type is optimal? When might we skip indexing entirely?”
Transactions	“If we relax isolation here, what’s the impact on throughput and correctness in a large-scale ordering system?”
Normalization vs Denormalization	“If reads are 100× writes, what trade-offs do we make in schema design?”
SQL vs NoSQL	“Given this set of requirements, is relational even the right tool? If not, what’s our hybrid architecture?”
Query Optimization	“What’s the impact of caching this query result at the application layer vs in-DB materialized view?”

Objectives

Your system should be able to:

1. **Ask clarifying questions** (requirements elicitation).
2. **Retrieve and present background knowledge** about database models, architectures, and trade-offs.
3. **Provide options and trade-offs** (e.g. “normalized relational schema vs. denormalized document store”) with clear explanations of consequences for performance, cost, complexity, and scalability.
4. **Simulate a “thinking aloud” process**: show reasoning, alternatives, and open questions—like a real system design discussion with a senior database architect.

Although each team grounds its work in the **industry of their SME**, the final tool should work for **any business context**—manufacturing, retail, healthcare, logistics, finance, or others.

Sample Query

- “We are designing an online exam system for 20k students concurrently submitting answers. Where do you store the data, and how do you handle peak load?”

1. Clarify requirements (forces you to ask questions)
2. Sketch high-level DB role in the architecture
3. Identify 2–3 options
4. Discuss trade-offs
5. Make a justified choice

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

Learning Goals

By completing this project you will:

- Develop the ability to **translate vague business requirements into database needs**.
- Apply advanced database concepts (schema design, indexing, transactions, normalization/denormalization, query optimization).
- Compare and justify different **database models** (relational, document, vector, graph, time-series).
- Demonstrate **system design thinking**: requirements gathering, high-level architecture, trade-off analysis.
- Evaluate a RAG system with **measurable metrics** (retrieval quality, latency, cost).
- Communicate design choices in a way that decision-makers can understand.

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

Capstone Project Grading

Component	Weight	Description
Requirements & Use Cases	15%	Analysis of SME's starting situation, mapped to broader industry needs. At least 3 example use cases across different maturity levels (e.g., Excel-only, ERP integration, AI-driven analytics).
Database Architecture Design	25%	Proposed schemas and architectures that cover different database models. Clear reasoning for model choice(s), indexing, normalization, and scaling.
RAG System Implementation	20%	Prototype advisor that retrieves knowledge and generates advisory conversations. Must "think aloud" and show trade-offs.
Trade-off Analysis	20%	At least 3 significant trade-offs (e.g., SQL vs NoSQL, push vs pull updates, caching vs recomputation). Each backed by corpus evidence or experimental results.
Evaluation	10%	Retrieval quality (recall@k, latency, cost) and at least one ablation or comparison (e.g., different chunking, index types).
Communication	10%	Clear documentation (design doc + ADRs) and final demo showing the advisor in action with realistic business scenarios.

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

How to write a bachelor-thesis, or “Portfolioprüfung”, or…

1. Die Motivation: *eigentlich das, was wir versprochen haben (s. LOOPi-Antragsdokument)*
2. Stakeholderanalyse: *WER soll unser Tool nutzen: wir suchen die 80-20 Maßnahmen, die “am meisten bringen”*
    1. *Landwirt*innen. z.B. Henrik sagt: ein tool mit Photosyntheseleistung für Schläge und/oder Betrieb*
    2. *Forscher (landwirtschaftlich, und Data-Nerds)*
    3. *Berater*
3. Anforderungen der verschiedenen Stakeholder (Was brauchen die genau)
4. Lastenheft/Anforderungen: *das hier sind nicht (nur) die “Wünsche” der Landwirt*innen, sondern die Anforderungen an das TOOL!*
    1. Funktional
    2. Nicht funktional
    3. Nach irgendeinem Kriterien geordnet (nach Wichtigkeit)
    4. SMART Ziele
5. Risikoanalyse: Projektmanagement der Ziele/Umsetzung der Anforderungen
6. Wenn mehrere Möglichkeiten, dann Nutzwertanalyse (auch bei einzelnen Entscheidungen)
7. Design
8. Umsetzung
9. Testen, Abgleichen mit Anforderungen (mit Hilfe von SMART Metriken)

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

Critical Conditions

- Use only synthetic/industry-provided corpora, never real company data.
- Your SME is just the *starting point*; the final system must be **generalizable across industries and maturity levels**.
- The **database focus** is central: the LLM layer is a **presentation and reasoning interface**, not the main subject.

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

eduScrum in practice

- **Sprint = short cycle** with clear *goal* and *end product*.
- **Roles** (each team of ~4–5 students assigns these right away):
  - Facilitator (keeps group on track)
  - Timekeeper (reminds of deadlines)
  - Quality Guard (checks Definition of Done is met)
- **Flip/Board**: visual workflow (To Do → Doing → Done).
  - Physical or online (lots of tools like Miro etc)
- **Definition of Done (DoD)**: checklist that defines when something is really finished.
- **Review** = show your product to everyone.
- **Retrospective** = reflect on teamwork, find one improvement for next time.

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

Micro-Sprint: The Perfect Pizza

Story

- As hungry students, we want to design the perfect pizza recipe so that our class could enjoy it at a dream pizza night.

Lernziel (Learning Goal)

- You will experience how a self-organized team works with eduScrum to achieve a concrete outcome.
- By the end of the sprint, your team will have:
  - Produced a clear, agreed pizza recipe,
  - Worked with roles, a flip (task board), and a Definition of Done,
  - Presented your result to the class.

Acceptance Criteria

- Your final product must include:
  - Ingredient list with quantities
  - Dough process (mixing, fermentation)
  - Baking instructions (temperature, duration, oven type)
  - Sketch or photo of the pizza
  - Short explanation of your choices (why these ingredients / method)

DHBW Loerrach
Baden-Wuerttemberg
Cooperative State University

Micro-Sprint: The Perfect Pizza ctd

Roles (choose in your team)

- Facilitator → keeps the team on task
- Timekeeper → manages time
- Quality Guard → checks Definition of Done
- (Optional) Note taker/Communicator

Definition of Done (DoD)

- A task is only 'done' when your team agrees it meets clear standards.
- Example:
  - Task = 'Research flour types'
  - DoD = At least 3 flour options identified, each with pros/cons documented, and one choice agreed.
- Your team must define DoD for at least one other task yourselves.

Artefacts

- Flip board (To Do → Doing → Done)
- Definition of Done for at least 2 tasks
- Final product = pizza recipe (meeting acceptance criteria)
- 3-minute pitch in the review

Timing

- 15 min → Sprint Planning (roles, tasks, DoD)
- 90 min → Sprint Execution (recipe, board, deliverables)
- 30 min → Sprint Review (presentations)
- 20 min → Retrospective (reflect on teamwork)
- Important: This sprint is not about pizza perfection — it is about practicing teamwork with eduScrum.

eduScrum

- Z.B.
- https://ilias.uni-marburg.de/ilias.php?baseClass=ilwikihandlergui&cmdNode=18j:te&cmdClass=ilobjwikigui&cmd=viewPage&ref_id=1857582&wpg_id=13183