

Data Technologies – Projectcases

Versie: 1.0, 2 februari 2026

1. Doel en randvoorwaarden

Binnen dit project werken studenten in teamverband aan **een werkende applicatie** waarin verschillende database-technologieën aantoonbaar en onderbouwd worden toegepast. De applicatie is het middel; de **focus ligt op datamodellering, databasekeuzes, query's, optimalisatie, beveiliging en documentatie.**

Randvoorwaarden

- **Applicatie verplicht:** er moet een werkende applicatie worden opgeleverd die de gekozen databases gebruikt.
- **Vrije technologiekeuze:**
 - Programmeertaal, libraries en frameworks zijn vrij te kiezen.
 - **ORM's zijn niet toegestaan;** database-interactie moet zichtbaar zijn via expliciete query's.
- **Projectmethodiek:**
 - Teams werken volgens **Scrum of Kanban**.
 - Een **product backlog met user stories** wordt in de eerste fase door het team zelf opgesteld en onderhouden.
- **Teamgrootte:** maximaal 4 studenten.
- **Databasetechnieken:** alle in de cursus behandelde database-technieken moeten aantoonbaar aan bod komen:
 - Relationale database (SQL)
 - Geavanceerde SQL (o.a. views, CTE's, aggregaties)
 - Normalisatie & database design
 - Database-optimalisatie (indexen, query-analyse)
 - NoSQL (document store)
 - Key-value database (caching)
 - Graph database
 - Database-beveiliging en privacy

Opmerking over ontwerpvolgorde

In de professionele praktijk wordt een applicatie vaak gestart met een volledig uitgewerkt datamodel. In dit project is het **databaseontwerp bewust iteratief**.

In de eerste weken stellen teams een **globaal conceptueel datamodel** op, gebaseerd op hun initiële analyse en user stories. Dit model is nadrukkelijk **niet definitief**. Naarmate in de workshops meer kennis wordt opgedaan over database design, normalisatie en optimalisatie, wordt het datamodel **herzien, verfijnd en onderbouwd**.

Het kunnen **bijstellen en verantwoorden van ontwerpkeuzes op basis van voortschrijdend inzicht** is een expliciet onderdeel van dit project en wordt meegenomen in de beoordeling.

Toelichting bij uitbreidingen (richtinggevend)

De genoemde uitbreidingen bij de cases zijn richtinggevend. Ze geven voorbeelden van mogelijke verdieping van de functionaliteit, maar zijn niet verplicht en niet uitputtend. Studenten zijn vrij om deze uitbreidingen geheel of gedeeltelijk toe te passen, of andere uitbreidingen te kiezen, mits de gemaakte keuzes inhoudelijk worden onderbouwd en passen binnen de randvoorwaarden van de cursus.

2. Case 1 – Online Webshop

Context

Je ontwikkelt een online webshop waarin gebruikers producten kunnen bekijken en bestellen. De webshop dient als experimenteeromgeving om verschillende database-technologieën te onderzoeken en te integreren.

Kernfunctionaliteiten

- Gebruikers kunnen een account aanmaken en inloggen
- Producten kunnen worden bekeken, gefilterd en gezocht
- Producten kunnen aan een winkelwagen worden toegevoegd
- Bestellingen kunnen worden geplaatst en ingezien

Uitbreidingen (richtinggevend)

- Productreviews en beoordelingen
 - Overzichten zoals top-verkopen of omzet per periode
 - Aanbevelingen (bijv. "klanten kochten ook")
 - Caching van veel geraadpleegde productoverzichten
-

3. Case 2 – Intranet CMS met Forum

Context

Je ontwikkelt een Content Management Systeem (CMS) voor een intranetomgeving. Medewerkers plaatsen content en gaan met elkaar in discussie via reacties en een forum.

Kernfunctionaliteiten

- Contentbeheerders kunnen artikelen aanmaken, wijzigen en publiceren
- Lezers kunnen artikelen bekijken
- Lezers kunnen reacties plaatsen onder artikelen
- Forumfunctionaliteit met onderwerpen en discussies
- Zoekfunctionaliteit over artikelen en discussies

Uitbreidingen (richtinggevend)

- Versiebeheer van artikelen
- Moderatie van reacties en forumberichten
- Overzichten van populaire artikelen of actieve discussies
- Aanbevelingen op basis van interesse of leesgedrag

4. Planning en deliverables per week (18 weken)

Week 1 – Kick-off & analyse

- Teams vormen en case kiezen
- Casus analyseren
- Eerste product backlog met user stories
- **Globaal conceptueel datamodel (eerste schets)**

Deliverable: initiële product backlog + globale ERD (conceptueel)

Week 2 – Relationale databases & basis SQL

- Identificeren kernentiteiten
- Eerste SQL-schema (DDL)
- Basis CRUD-query's

Deliverable: SQL-schema v1 + seed data

Week 3 – Geavanceerde SQL

- JOINS, aggregaties
- Toepassen van view, CTE of window function

Deliverable: query-portfolio v1

Week 4 – Database design

- **Herzien en uitwerken van het datamodel**
- Conceptueel → logisch → fysiek ontwerp
- Constraints (PK, FK, UNIQUE)

Deliverable: uitgewerkte ERD (definitieve versie) + toelichting

Week 5 – Normalisatie

- Normaliseren tot minimaal 3NF
- Motiveren van ontwerpkeuzes

Deliverable: genormaliseerd schema + uitleg

Week 6 – Database-optimalisatie

- Indexen toevoegen
- Query-analyse met EXPLAIN

Deliverable: query-portfolio v2 + performance-analyse

Week 7 – Tentamenfase

- Schriftelijke toets
 - Stabiliseren relationele database
-

Week 8 – NoSQL databases (introductie)

- Analyse geschikte data voor NoSQL
- Ontwerp documentmodel

Deliverable: NoSQL-datamodel + motivatie

Week 9 – NoSQL toepassen & Herkansing tentamen

- Implementatie NoSQL
- Voorbeeldquery's

Deliverable: werkende NoSQL-integratie

Week 10 – Key-value databases

- Bepalen cache-use-case
 - Ontwerp key-structuur
-

Week 11 – Key-value toepassen

- Implementatie caching
- Cache-invalidering beschrijven

Deliverable: werkende cache + motivatie

Week 12 – Graph databases

- Ontwerp graafmodel
 - Keuze aanbevelingslogica
-

Week 13 – Graph toepassen & tussentijdse presentatie

- Implementatie graph database
- Cypher-query's

Deliverable: tussentijdse presentatie

Week 14 – Database security & privacy

- Rollen en rechten
- Privacymaatregelen

Deliverable: security- en privacy-sectie (technisch rapport)

Week 15–16 – Integratie & documentatie

- Integratie alle databases
 - Afronden rapportages
-

Week 17 – Afronding

- Final checks
 - Opleveren alle artefacten
-

Week 18 – Demo day

- Eindpresentatie en demonstratie
-