



Relationale Datenbanken

Vom fachlichen Datenmodell zur sicheren SQL-Auswertung

Daten modellieren · SQL formulieren · Ergebnisse prüfen

BKWI · BPE 3 · 45 Unterrichtsstunden

Didaktisch erweiterte Kursfassung · A5 · Schuljahr 2026/2027 · v1.0 · Serienmaster R7

Christine Janischek · Lizenz Custom License NC School-Only 1.0 · emotionalspirit.de

*Dieses Lernskript wurde mit Unterstützung einer KI erstellt und anschließend redaktionell geprüft.
Trotz sorgfältiger Bearbeitung können Fehler enthalten sein. Bitte melde Fehler und
Verbesserungsvorschläge über die angegebene Kontaktmöglichkeit.*

LERN- UND ARBEITSSKRIPT

Relationale Datenbanken

Vom fachlichen Datenmodell zur sicheren SQL-Auswertung



Bildungsgang	Bildungsplaneinheit	Zeitrictwert
Berufskolleg Wirtschaftsinformatik (BKWI), Schuljahr 1	BPE 3 Relationale Datenbanken	45 Unterrichtsstunden

Arbeitsweise

Dieses Skript verbindet Erklären, Modellieren, Implementieren, Testen und Reflektieren. Es kann vollständig in Word bearbeitet werden. Praktische Aufgaben werden mit MySQL Workbench und dem bereits bekannten VS Code durchgeführt. Modelle, SQL-Dateien, Übungen und Selbstkontrollen können durch die Lehrkraft lokal oder über das Kurs-Repository bereitgestellt werden.

Hinweis zur Erstellung

Dieses Unterrichtsmaterial wurde mit Unterstützung Künstlicher Intelligenz erstellt und fachlich strukturiert. Trotz sorgfältiger Prüfung können Fehler oder Unvollständigkeiten enthalten sein. Hinweise bitte an die Autorin Christine Janischek melden.

Stand: September 2026 · Finale Pilotfassung 1.0 · Bildungsplan BKWI
Wirtschaftsinformatik vom 13. März 2025

So arbeitest du mit diesem Skript

Visueller Impuls

SQL-Arbeitsbereich · Lernagent-Datenbank

```

SELECT l.vorname, a.titel, b.punkte
FROM lernende l
JOIN bearbeitungen b ON b.lernende_id = l.lernende_id
JOIN aufgaben a ON a.aufgabe_id = b.aufgabe_id
WHERE b.punkte >= 8;

```

Ausführen

vorname	titel	punkte
Aylin	Relative Pfade prüfen	9
Ben	Flexbox anwenden	8
Cem	SQL-Abfrage erklären	10

Prüffragen

- Welche Tabellen?
- Welche JOIN-Bedingung?
- Ist der Filter prüfbar?

Abbildung: SQL-Abfrage und Ergebnisansicht. Ordne sichtbare Spalten den Tabellen und JOIN-Bedingungen zu.

Symbol / Block	Bedeutung
Lernziel	Das kannst du nach dem Abschnitt erklären oder praktisch umsetzen.
Merksatz	Kernaussage, die du sicher beherrschen solltest.
Grundlagen	Verbindliche Aufgabe zum Aufbau der Basiskompetenz.
Vertiefung	Anspruchsvollere Anwendung, Analyse oder Transfer.
Checkpoint	Kurze Selbstkontrolle vor dem Weiterarbeiten.
Repo-Praxis	Datei oder Lernangebot aus dem GitHub-Repository.

Technische Arbeitsumgebung

- MySQL Server und MySQL Workbench aus der digitalen Schultasche
- VS Code zum Öffnen der HTML-Lernseiten und SQL-Dateien
- ein Browser für lokale HTML-Übungen und das bestehende Website-Projekt
- optional ein von der Lehrkraft bereitgestellter Zugang zum Kurs-Repository

Vorkenntnisse und Anschluss

Du hast bereits mit Tabellenkalkulation gearbeitet, kennst VS Code und entwickelst eine eigene Website. Datenbanken werden in diesem Skript von Grund auf eingeführt. GitHub, Templates, GitHub Education und Copilot sind nicht Voraussetzung; dafür ist ein eigenes Einführungsskript vorgesehen.

Inhaltsverzeichnis

Die Einträge führen direkt zu den Kapiteln und Hauptabschnitten. In den Überschriften bringt dich „↔ Inhalt“ hierher zurück.

- 01 Datenbanken im betrieblichen Kontext**
- 02 Fachliche Modellierung mit dem EERM**
- 03 Kardinalitäten und Beziehungen**
- 04 Vom EERM zum relationalen Schema**
- 05 Redundanzen, Anomalien und Normalisierung**
- 06 MySQL Workbench: Modell und Datenbank erstellen**
- 07 Daten erfassen und verändern: INSERT, UPDATE, DELETE**
- 08 Daten abfragen: SELECT, WHERE und ORDER BY**
- 09 Mehrere Tabellen verbinden: JOIN**
- 10 Daten verdichten: Funktionen, GROUP BY und HAVING**

- 11** Unterabfragen und komplexe Auswertungen
- 12** Datenqualität, Massendaten und Verantwortung
- 13** Lernmaterialien nutzen und Arbeit dokumentieren
- 14** Lernprojekt: Datenbank für den KI-Lernagenten
- 15** Selbsttest und Prüfungsvorbereitung
- 16** Wochenplan für 45 Unterrichtsstunden
- 17** Repo-Ressourcen und Einsatz im Unterricht
- 18** Quellen und Weiterarbeit
- 19** Lizenz und Kontakt

1. Datenbanken im betrieblichen Kontext

↪ Inhalt

Kapitelziel

Du unterscheidest Daten, Informationen, Datenbanken und Datenbankmanagementsysteme und begründest den Einsatz relationaler Datenbanken.

1.1 Von Daten zu Informationen

Daten sind zunächst Zeichen oder Werte ohne zwingenden Zusammenhang. Erst durch Kontext und Auswertung entstehen Informationen. Ein einzelner Wert 8 ist kaum aussagekräftig; als „8 von 10 Punkten bei der Aufgabe Normalisierung“ kann er dem Lernagenten zur Auswahl einer passenden nächsten Übung dienen.

Brücke zu deinem Website-Projekt

Dein aktuelles Webprojekt funktioniert zunächst ohne Datenbank und kann so vollständig abgegeben werden. HTML und CSS beschreiben Aufbau und Gestaltung; eine Datenbank ist zu Beginn ausdrücklich nicht verpflichtend. Im Datenbankunterricht lernst du jedoch bereits, wie sich die Inhalte deiner Galerie, deines Automaten oder einer anderen kleinen Webanwendung später strukturiert speichern ließen.

Technische Trennung

Eine reine HTML-/CSS-Seite greift nicht direkt auf MySQL zu. Für eine sichere Verbindung wird später eine serverseitige oder objektorientierte Anwendung beziehungsweise eine Programmierschnittstelle benötigt. Welche objektorientierte Sprache dafür verwendet wird, bleibt offen und wird erst im späteren OOP-Kurs festgelegt.

Begriff	Kurzbeschreibung	Beispiel
Daten	gespeicherte Werte oder Zeichen	2026-09-01; Aufgabe 104; 8 Punkte
Information	interpretierte Daten mit Kontext	Aufgabe 104 wurde mit 8 von 10 Punkten gelöst
Datenbank	strukturierte, dauerhaft gespeicherte Datenmenge	Kurse, Aufgaben und Bearbeitungen
DBMS	Software zum Definieren, Speichern, Schützen und Abfragen	MySQL
Anwendung	nutzt das DBMS für einen Zweck	KI-Lernagent oder Kurstrainer

Merksatz

Eine Datenbank speichert strukturierte Daten. Das Datenbankmanagementsystem stellt Regeln und Werkzeuge bereit, damit mehrere Anwendungen zuverlässig damit arbeiten können.

RDB-K01-A01 · Grundlagen: Daten oder Information?

Ordne fünf Beispiele aus einem Onlineshop den Begriffen Daten oder Information zu und begründe zwei Zuordnungen.

Material und Abgabe: Skript; Tabelle mit fünf Zuordnungen und zwei Begründungen.

Prüfe: Alle fünf Beispiele sind zugeordnet; zwei Begründungen verwenden die Begriffe Kontext und Information.

RDB-K01-A02 · Vertiefung: Tabellenkalkulation oder Datenbank?

Vergleiche beide Werkzeuge für viele Lernende, parallele Bearbeitungen, Beziehungen zwischen Kursen und Aufgaben sowie Zugriffsschutz. Formuliere eine begründete Empfehlung.

Material und Abgabe: Skript; Vergleich in vier Kriterien und begründete Empfehlung.

Prüfe: Datenmenge, Mehrbenutzerbetrieb, Beziehungen und Zugriffsschutz werden korrekt verglichen.

2. Fachliche Modellierung mit dem EERM

↪ Inhalt

Kapitelziel

Du leitest Entitätstypen, Attribute, Schlüssel und Beziehungen aus einem betrieblichen Sachverhalt ab.

2.1 Abstraktion und Reduktion

Ein Datenmodell bildet nicht die gesamte Wirklichkeit ab. Es wählt genau die Merkmale aus, die für den Zweck des Informationssystems erforderlich sind. Diese bewusste Vereinfachung heißt Abstraktion und Reduktion.

Element	Bedeutung	Beispiel KI-Lernagent
Entität	ein konkretes einzelnes Objekt	Aufgabe mit ID 104
Entitätstyp	Menge gleichartiger Entitäten	AUFGABE
Attribut	Eigenschaft eines Entitätstyps	Titel, Schwierigkeit
Primärschlüssel	identifiziert jeden Datensatz eindeutig	aufgabe_id
Beziehungstyp	fachlicher Zusammenhang	LERNENDE bearbeiten AUFGABE

2.2 Modellierungsroutine

1. Sachverhalt lesen und Zweck klären.
2. Nomen als Kandidaten für Entitätstypen markieren.
3. Eigenschaften als Attribute zuordnen.
4. Für jeden Entitätstyp einen stabilen Primärschlüssel wählen.
5. Verben als mögliche Beziehungen prüfen.
6. Kardinalitäten sprachlich begründen.
7. Modell auf Vollständigkeit, Eindeutigkeit und Redundanz prüfen.

Achtung

Namen, E-Mail-Adressen oder Telefonnummern sind meist keine guten Primärschlüssel: Sie können sich ändern oder mehrfach vorkommen. Künstliche IDs sind häufig stabiler.

RDB-K02-A01 · Grundlagen: Lernagent modellieren

Markiere im folgenden Sachverhalt Entitätstypen, Attribute und Beziehungen: Lernende bearbeiten Aufgaben. Jede Aufgabe gehört zu einem Kurs und kann mehrere gestufte Hinweise besitzen.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K02-A01_modellierung.drawio; EERM mit Entitätstypen, Attributen und Beziehungen.

Prüfe: Mindestens LERNENDE, KURS, AUFGABE, HINWEIS und BEARBEITUNG sind fachlich nachvollziehbar modelliert.

RDB-K02-A02 · Vertiefung: Modellierungsentscheidung begründen

Prüfe, ob der Bearbeitungsstatus als Attribut der Aufgabe oder der Bearbeitung modelliert werden muss. Beziehe mehrere Lernende in deine Begründung ein.

Material und Abgabe: Skript; Begründung mit mindestens drei vollständigen Sätzen.

Prüfe: Der Status liegt bei BEARBEITUNG; die Begründung berücksichtigt mehrere Lernende und mehrere Versuche.

3. Kardinalitäten und Beziehungen [↔ Inhalt](#)

Kapitelziel

Du formulierst Eine-Sätze und bestimmst 1:1-, 1:n- und m:n-Beziehungen einschließlich Optionalität.

3.1 Die Zwei-Satz-Methode

Für jede Beziehung formulierst du zwei Sätze, jeweils aus Sicht eines Entitätstyps. So werden Mengenverhältnisse nachvollziehbar statt nur „gefühlte“ eingetragen.

Beziehung	Eine-Sätze	Umsetzung
1:1	Eine Person besitzt höchstens einen Mitarbeiterausweis; ein Ausweis gehört genau einer Person.	Fremdschlüssel auf einer fachlich passenden Seite

Beziehung	Eine-Sätze	Umsetzung
1:n	Ein Kurs enthält viele Aufgaben; eine Aufgabe gehört zu genau einem Kurs.	Fremdschlüssel der 1-Seite in der n-Tabelle
m:n	Ein Lernender bearbeitet viele Aufgaben; eine Aufgabe wird von vielen Lernenden bearbeitet.	eigene Zwischentabelle BEARBEITUNG

Merksatz

Eine m:n-Beziehung wird im relationalen Modell durch eine Zuordnungstabelle aufgelöst. Diese enthält die Fremdschlüssel beider beteiligten Tabellen und häufig zusätzliche Vorgangsdaten.

RDB-K03-A01 · Grundlagen: Eine-Sätze formulieren

Formuliere zu KURS – AUFGABE und LERNENDE – BEARBEITUNG jeweils zwei Eine-Sätze. Bestimme die Kardinalität.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K03-A01_eine_saetze.docx; Vier Eine-Sätze und zwei Kardinalitäten.

Prüfe: Beide Beziehungsrichtungen, Minimum und Maximum sind fachlich korrekt.

RDB-K03-A02 · Vertiefung: Optionalität analysieren

Kann ein neu angelegter Lernender zunächst ohne Bearbeitung existieren? Kann eine Bearbeitung ohne Aufgabe existieren? Begründe fachlich und leite Minimal-/Maximalkardinalitäten ab.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K03-A01_eine_saetze.docx; Begründete Min-/Max-Kardinalitäten.

Prüfe: Optionalität wird aus den Geschäftsregeln und nicht aus Vermutungen abgeleitet.

4. Vom EERM zum relationalen Schema ↩

Inhalt

Kapitelziel

Du überführst Entitätstypen und Beziehungen in Tabellen mit geeigneten Datentypen, Primär- und Fremdschlüsseln.

4.1 Transformationsregeln

- Entitätstyp → Tabelle
- Attribut → Spalte
- Primärschlüssel → PRIMARY KEY
- 1:n-Beziehung → Fremdschlüssel in der n-Tabelle
- m:n-Beziehung → neue Zuordnungstabelle
- fachliche Pflichtangabe → NOT NULL
- eindeutiger fachlicher Wert → UNIQUE, wenn begründet

4.2 Geeignete Datentypen

Fachlicher Inhalt	MySQL-Beispiel	Hinweis
kurzer Text	VARCHAR(100)	maximale sinnvolle Länge festlegen
Ganzzahl	INT	für IDs und Zählwerte
Geldbetrag	DECIMAL(10,2)	nicht FLOAT verwenden
Datum	DATE	ohne Uhrzeit
Zeitpunkt	DATETIME	Datum und Uhrzeit
Wahrheitswert	BOOLEAN	in MySQL intern meist TINYINT(1)

LERNENDE(lernende_id PK, vorname, nachname, registriert_am, aktiv)
 KURS(kurs_id PK, titel, niveau)
 AUFGABE(aufgabe_id PK, kurs_id FK, titel, schwierigkeit, max_punkte)
 BEARBEITUNG(bearbeitung_id PK, lernende_id FK, aufgabe_id FK, begonnen_am, status, erreichte_punkte)

RDB-K04-A01 · Grundlagen: Schema notieren

Überführe dein EERM aus Kapitel 2 in Relationenschreibweise. Kennzeichne PK und FK.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K04-A01_relationenschema.docx; Relationenschreibweise mit PK- und FK-Kennzeichnung.

Prüfe: Jeder Entitätstyp ist übertragen; alle Fremdschlüssel verweisen auf den passenden Primärschlüssel.

RDB-K04-A02 · Vertiefung: Datentypen bewerten

Begründe für Punktzahl, Schwierigkeitsstufe, Bearbeitungszeitpunkt und Aktiv-Status jeweils einen geeigneten Datentyp. Nenne eine typische Fehlentscheidung.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K04-A02_datentypen.docx; Vier Datentypentscheidungen mit Begründung.

Prüfe: Wertebereich, Genauigkeit, NULL und typische Fehlentscheidung werden berücksichtigt.

5. Redundanzen, Anomalien und Normalisierung ↔ Inhalt

Kapitelziel

Du erkennst problematische Abhängigkeiten und überführst ein Tabellenmodell schrittweise bis zur dritten Normalform.

5.1 Warum normalisieren?

Redundanz bedeutet, dass dieselbe fachliche Tatsache mehrfach gespeichert wird. Sie führt zu höherem Pflegeaufwand und kann widersprüchliche Daten erzeugen.

Anomalie	Problem	Beispiel Lernagent
Änderungsanomalie	eine Tatsache muss mehrfach geändert werden	Kurstitel steht in jeder Bearbeitung
Einfügeanomalie	eine Information lässt sich ohne fremde Daten nicht speichern	neue Aufgabe erst mit Bearbeitung speicherbar
Löschanomalie	beim Löschen geht unbeabsichtigt Wissen verloren	letzte Bearbeitung löscht einzige Aufgabendaten

5.2 Die drei Normalformen

Normalform	Prüffrage	Korrektur
1NF	Sind alle Werte atomar und gibt es keine Wiederholungsgruppen ?	Listen und Mehrfachwerte auflösen
2NF	Hängt jedes Nichtschlüsselattribut vom gesamten zusammengesetzten Schlüssel ab?	partielle Abhängigkeiten auslagern
3NF	Hängt kein Nichtschlüsselattribut indirekt von einem anderen Nichtschlüsselattribut ab?	transitive Abhängigkeiten auslagern

Lernroutine

Bestimme zuerst den Schlüssel und notiere funktionale Abhängigkeiten. Zerlege danach schrittweise. Prüfe nach jeder Zerlegung, ob sich die ursprünglichen Informationen durch JOIN wieder vollständig zusammensetzen lassen.

RDB-K05-A01 · Grundlagen: Bis 3NF zerlegen

Normalisiere LERNPROTOKOLL(lernende_id, lernendenname, aufgabe_id, aufgabentitel, kurs_id, kurstitel, status, erreichte_punkte). Dokumentiere jeden Schritt.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K05-A01_normalisierung.docx; Dokumentierte Zerlegung von 1NF bis 3NF.

Prüfe: Primärschlüssel und funktionale Abhängigkeiten sind angegeben; partielle und transitive Abhängigkeiten werden entfernt.

RDB-K05-A02 · Vertiefung: Grenzen diskutieren

Erkläre, warum in Auswertungssystemen manchmal bewusst redundante Strukturen verwendet werden. Trenne operative Datenbank und Analysezweck.

Material und Abgabe: Skript; Abwägung in 120 bis 180 Wörtern.

Prüfe: Operative Datenhaltung und Analysezweck werden getrennt; Nutzen und Risiken bewusster Redundanz sind benannt.

6. MySQL Workbench: Modell und Datenbank erstellen [↩ Inhalt](#)

Kapitelziel

Du erstellst ein EERM, leitest ein Schema ab und implementierst Tabellen mit Schlüsseln und Integritätsregeln.

6.1 Workbench-Arbeitsablauf

1. MySQL Workbench starten und Verbindung zum lokalen Server öffnen.
2. File → New Model und Add Diagram wählen.
3. Tabellen, Spalten, Datentypen und Primärschlüssel anlegen.
4. Beziehungen mit passenden Werkzeugen setzen.

5. Database → Forward Engineer ausführen und SQL-Vorschau prüfen.
6. Schema erzeugen und Tabellen im Navigator kontrollieren.
7. Testdaten einfügen und Regeln gezielt testen.

```
CREATE DATABASE lernagent CHARACTER SET utf8mb4;
USE lernagent;

CREATE TABLE kurse (
  kurs_id INT PRIMARY KEY AUTO_INCREMENT,
  titel VARCHAR(100) NOT NULL,
  niveau VARCHAR(30) NOT NULL
);

CREATE TABLE aufgaben (
  aufgabe_id INT PRIMARY KEY AUTO_INCREMENT,
  kurs_id INT NOT NULL,
  titel VARCHAR(120) NOT NULL,
  schwierigkeit INT NOT NULL,
  max_punkte DECIMAL(5,2) NOT NULL,
  aktiv BOOLEAN NOT NULL DEFAULT TRUE,
  CONSTRAINT fk_aufgabe_kurs
    FOREIGN KEY (kurs_id) REFERENCES kurse(kurs_id)
    ON UPDATE CASCADE ON DELETE RESTRICT
);
```

Referentielle Integrität

Ein Fremdschlüssel darf nur auf einen vorhandenen Primärschlüsselwert verweisen – oder NULL enthalten, wenn dies fachlich und technisch erlaubt ist. RESTRICT, CASCADE und SET NULL müssen fachlich begründet werden.

RDB-K06-A01 · Grundlagen: Forward Engineering

Erstelle aus deinem Modell ein Schema. Sichere das Modell als .mwb und das erzeugte SQL als .sql. Dokumentiere einen erfolgreichen Integritätstest.

Material und Abgabe: 02_SQL/01_lernagent_struktur.sql und 01_Aufgaben_und_Vorlagen/RDB-K06-A01_workbench_checkliste.md; .mwb-Datei, Struktur-SQL und dokumentierter Integritätstest.

Prüfe: Sieben Tabellen, PK/FK, UNIQUE- und CHECK-Regeln sind vorhanden; mindestens ein gültiger und ein ungültiger Test sind dokumentiert.

RDB-K06-A02 · Vertiefung: Löschregel entscheiden

Bewerte ON DELETE RESTRICT, CASCADE und SET NULL für das Löschen eines Kurses mit zugeordneten Aufgaben und bereits vorhandenen Bearbeitungen.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K06-A02_loeschregeln.docx; Entscheidungstabelle und Empfehlung.

Prüfe: RESTRICT, CASCADE und SET NULL werden fachlich verglichen; die Empfehlung schützt bestehende Bearbeitungen.

7. Daten erfassen und verändern: INSERT, UPDATE, DELETE ↩ Inhalt

Kapitelziel

Du manipulierst Daten gezielt und sicher und erkennst Risiken fehlender Bedingungen.

7.1 INSERT

```
INSERT INTO kurse (titel, niveau)
VALUES ('Relationale Datenbanken', 'Grundlagen');
```

```
INSERT INTO aufgaben (kurs_id, titel, schwierigkeit,
max_punkte)
VALUES (1, 'Entitätstypen erkennen', 1, 10.00);
```

7.2 UPDATE und DELETE

```
UPDATE aufgaben
SET max_punkte = 12.00
WHERE aufgabe_id = 1;
```

```
DELETE FROM aufgaben
WHERE aufgabe_id = 1;
```

Sicherheitsroutine

Vor UPDATE oder DELETE zuerst dieselbe WHERE-Bedingung mit SELECT testen. In einer Transaktion arbeiten, Ergebnis prüfen und erst danach COMMIT ausführen. Ohne WHERE kann die gesamte Tabelle verändert oder gelöscht werden.

```
START TRANSACTION;
SELECT * FROM aufgaben WHERE aktiv = FALSE;
DELETE FROM aufgaben WHERE aktiv = FALSE;
-- Ergebnis prüfen
ROLLBACK; -- oder bewusst COMMIT;
```

RDB-K07-A01 · Grundlagen: DML ausführen

Füge zwei Kurse und sechs Aufgaben ein. Ändere die maximale Punktzahl einer Aufgabe und deaktiviere eine weitere Aufgabe. Prüfe jeden Schritt mit SELECT.

Material und Abgabe: 02_SQL/01_lernagent_struktur.sql, 02_lernagent_testdaten_basis.sql und RDB-K07-A01_arbeitsdatei.sql; Ausführbare DML-Datei mit SELECT-Nachweisen.

Prüfe: Zwei Kurse und sechs Aufgaben werden konsistent eingefügt; UPDATE und Deaktivierung betreffen genau die vorgesehenen Zeilen.

RDB-K07-A02 · Vertiefung: Fehlerdiagnose

Erzeuge kontrolliert eine Fremdschlüsselverletzung. Notiere Fehlermeldung, Ursache und korrekte Lösung. Lösche keine Constraints zur „Reparatur“.

Material und Abgabe: 04_Testmaterial/RDB-K07-A02_fremdschluesselfehler.sql; Fehlerprotokoll mit Ursache und korrigierter Anweisung.

Prüfe: Die FK-Verletzung wird reproduziert, erklärt und ohne Entfernen des Constraints behoben.

8. Daten abfragen: SELECT, WHERE und ORDER BY [↩ Inhalt](#)

Kapitelziel

Du wählst Spalten gezielt aus, filterst Datensätze logisch korrekt und sortierst Ergebnisse.

8.1 Grundstruktur einer Abfrage

```
SELECT titel, schwierigkeit, max_punkte  
FROM aufgaben  
WHERE aktiv = TRUE  
      AND schwierigkeit <= 2  
ORDER BY schwierigkeit ASC, titel ASC;
```

Baustein	Aufgabe	Merksatz
SELECT	Spalten auswählen (Projektion)	SELECT * nur zur Erkundung
FROM	Datenquelle festlegen	Tabellenrolle verstehen
WHERE	Zeilen filtern (Selektion)	NULL mit IS NULL prüfen
ORDER BY	Ergebnis sortieren	ASC aufsteigend, DESC absteigend

8.2 Logische Bedingungen

```
SELECT aufgabe_id, titel, schwierigkeit, max_punkte
FROM aufgaben
WHERE aktiv = TRUE
      AND (schwierigkeit = 1 OR schwierigkeit = 2)
      AND max_punkte BETWEEN 5.00 AND 15.00;
```

Operatorrangfolge

AND wird vor OR ausgewertet. Setze Klammern, wenn Bedingungen gemischt werden. Das macht die fachliche Absicht eindeutig und verhindert schwer erkennbare Fehler.

RDB-K08-A01 · Grundlagen: Filtern und sortieren

Gib alle aktiven Aufgaben der Schwierigkeitsstufen 1 und 2 bis maximal 15 Punkte aus. Sortiere nach Schwierigkeit und Titel.

Material und Abgabe: 02_SQL/03_lernagent_testdaten_training.sql und RDB-K08-A01_arbeitsdatei.sql; SELECT-Abfrage und Ergebniskontrolle.

Prüfe: Nur aktive Aufgaben der Stufen 1 und 2 bis 15 Punkte erscheinen; Sortierung ist zweistufig korrekt.

RDB-K08-A02 · Vertiefung: Testfälle entwickeln

Formuliere Testdaten, mit denen du zeigen kannst, dass eine gemischte AND-/OR-Bedingung ohne Klammern ein falsches Ergebnis liefert.

Material und Abgabe: 04_Testmaterial/RDB-K08-A02_operatorentest.sql; Testdaten und zwei verglichene Abfragen.

Prüfe: Mindestens ein Datensatz weist den Unterschied zwischen ungeklammerter und geklammerter Bedingung nach.

9. Mehrere Tabellen verbinden: JOIN ↩

Inhalt

Kapitelziel

Du leitest JOIN-Pfade aus PK-/FK-Beziehungen ab und vermeidest kartesische Produkte und Scheinduplikate.

9.1 JOIN-Routine

1. Ausgabeattribute in der Aufgabe markieren.
2. Start- und Zieltabellen bestimmen.
3. PK-/FK-Pfad im EERM verfolgen.
4. Jede Tabelle mit Alias notieren.
5. JOIN ... ON schrittweise ergänzen.
6. Erst danach WHERE, GROUP BY und ORDER BY hinzufügen.
7. Ergebniszahl und Beispielzeilen fachlich prüfen.

```
SELECT l.vorname, l.nachname, a.titel, b.begonnen_am
FROM bearbeitungen b
JOIN lernende l ON l.lernende_id = b.lernende_id
JOIN aufgaben a ON a.aufgabe_id = b.aufgabe_id
WHERE b.status = 'abgeschlossen'
ORDER BY b.begonnen_am;
```

9.2 INNER JOIN und LEFT JOIN

INNER JOIN liefert nur passende Kombinationen. LEFT JOIN behält alle Zeilen der linken Tabelle und ergänzt NULL, wenn rechts kein Treffer existiert. Damit lassen sich zum Beispiel Lernende ohne Bearbeitung finden.

```
SELECT l.lernende_id, l.vorname, l.nachname
FROM lernende l
LEFT JOIN bearbeitungen b ON b.lernende_id = l.lernende_id
WHERE b.bearbeitung_id IS NULL;
```

Duplikate verstehen

Mehrere Ergebniszeilen sind bei 1:n- oder m:n-Beziehungen häufig fachlich korrekt. DISTINCT darf erst eingesetzt werden, nachdem der JOIN-Pfad geprüft und die Ursache verstanden wurde.

RDB-K09-A01 · Grundlagen: Drei Tabellen verbinden

Gib Nachname, Aufgabentitel und Beginn aller Bearbeitungen aus. Zeichne vorher den JOIN-Pfad.

Material und Abgabe: 02_SQL/03_lernagent_testdaten_training.sql und RDB-K09-A01_arbeitsdatei.sql; JOIN-Abfrage plus skizzierter PK-/FK-Pfad.

Prüfe: LERLENDE, BEARBEITUNG und AUFGABE sind korrekt verbunden; kein kartesisches Produkt entsteht.

RDB-K09-A02 · Vertiefung: Lernpfad auswerten

Verbinde LERNENDE, BEARBEITUNG, AUFGABE und KURS. Gib alle abgeschlossenen Aufgaben eines Lernenden mit Kurs und erreichter Punktzahl aus.

Material und Abgabe: 02_SQL/03_lernagent_testdaten_training.sql; Vier-Tabellen-JOIN und fachliche Ergebniskontrolle.

Prüfe: Der JOIN-Pfad ist vollständig; Filter und Sortierung entsprechen dem Arbeitsauftrag; Mehrfachzeilen werden fachlich erklärt.

10. Daten verdichten: Funktionen, GROUP BY und HAVING [↔ Inhalt](#)

Kapitelziel

Du setzt Aggregat- und Datumsfunktionen ein, bildest Gruppen und filterst Gruppenergebnisse.

10.1 Aggregatfunktionen

Funktion	Bedeutung	Beispiel
COUNT	Anzahl	COUNT(*)
SUM	Summe	SUM(erreichte_punkte)
AVG	arithmetischer Mittelwert	AVG(erreichte_punkte)
MIN	kleinster Wert	MIN(begonnen_am)
MAX	größter Wert	MAX(max_punkte)

```

SELECT a.schwierigkeit, COUNT(*) AS anzahl_bearbeitungen
FROM bearbeitungen b
JOIN aufgaben a ON a.aufgabe_id = b.aufgabe_id
WHERE b.status = 'abgeschlossen'
GROUP BY a.schwierigkeit
HAVING COUNT(*) >= 3
ORDER BY anzahl_bearbeitungen DESC;

```

WHERE oder HAVING?

WHERE filtert einzelne Ausgangszeilen vor der Gruppierung.

HAVING filtert bereits gebildete Gruppen nach der Aggregation.

10.2 Datum und Zeitraum

```

SELECT bearbeitung_id, begonnen_am
FROM bearbeitungen
WHERE begonnen_am >= '2026-04-01'
  AND begonnen_am < '2026-05-01';

```

Halboffene Intervalle sind für Zeiträume oft robuster als
 MONTH(begonnen_am)=4, insbesondere wenn mehrere Jahre oder
 Indizes berücksichtigt werden.

RDB-K10-A01 · Grundlagen: Gruppieren

Ermittle Anzahl der Aufgaben und durchschnittliche Maximalpunktzahl je Schwierigkeitsstufe. Sortiere nach Anzahl absteigend.

Material und Abgabe: 02_SQL/03_lernagent_testdaten_training.sql und RDB-K10-A01_arbeitsdatei.sql; Gruppierte Abfrage.

Prüfe: COUNT und AVG werden je Schwierigkeitsstufe korrekt berechnet und nach Anzahl absteigend sortiert.

RDB-K10-A02 · Vertiefung: Lernstand auswerten

Berechne je Kurs die durchschnittlich erreichten Punkte abgeschlossener Bearbeitungen. Zeige nur Kurse mit mindestens fünf Bearbeitungen und erkläre die Reihenfolge der SQL-Klauseln.

Material und Abgabe: 02_SQL/03_lernagent_testdaten_training.sql; Auswertung mit JOIN, GROUP BY, HAVING und Erklärung.

Prüfe: Nur Gruppen mit mindestens fünf Datensätzen erscheinen; WHERE und HAVING werden korrekt unterschieden.

11. Unterabfragen und komplexe Auswertungen [← Inhalt](#)

Kapitelziel

Du nutzt Unterabfragen, wenn ein Zwischenergebnis Teil einer Bedingung ist, und vergleichst sie mit JOIN-Lösungen.

11.1 Skalarer Vergleich

```
SELECT aufgabe_id, titel, max_punkte
FROM aufgaben
WHERE max_punkte > (
    SELECT AVG(max_punkte)
    FROM aufgaben
);
```

11.2 Mengenbedingung mit IN

```
SELECT vorname, nachname
FROM lernende
WHERE lernende_id IN (
    SELECT lernende_id
    FROM bearbeitungen
    WHERE status = 'offen'
);
```

Entscheidungshilfe

Eine Unterabfrage ist geeignet, wenn die äußere Abfrage ein einzelnes Vergleichsergebnis oder eine Menge benötigt. JOIN ist häufig gut lesbar, wenn Attribute aus mehreren Tabellen ausgegeben werden. Entscheidend ist eine korrekte, nachvollziehbare Lösung.

RDB-K11-A01 · Grundlagen: Über dem Durchschnitt

Gib Aufgaben aus, deren maximale Punktzahl über dem Durchschnitt liegt.

Material und Abgabe: 02_SQL/03_lernagent_testdaten_training.sql und RDB-K11-A01_arbeitsdatei.sql; Abfrage mit skalarer Unterabfrage.

Prüfe: Die Unterabfrage liefert genau einen Durchschnittswert; nur Werte oberhalb dieses Ergebnisses werden ausgegeben.

RDB-K11-A02 · Vertiefung: Zwei Lösungswege

Finde Lernende mit abgeschlossenen Bearbeitungen einmal mit IN-Unterabfrage und einmal mit JOIN. Vergleiche Lesbarkeit und mögliche Duplikate.

Material und Abgabe: 02_SQL/03_lernagent_testdaten_training.sql; IN- und JOIN-Lösung mit Vergleich in 80 bis 120 Wörtern.

Prüfe: Beide Lösungen liefern dieselbe Personenmenge; mögliche JOIN-Duplikate und DISTINCT werden fachlich erklärt.

12. Datenqualität, Massendaten und Verantwortung [↩ Inhalt](#)

Kapitelziel

Du diskutierst Chancen und Risiken großer personalisierter Datenbestände und entwickelst begründete Schutzmaßnahmen.

12.1 Chancen und Risiken

Perspektive	Chance	Risiko
Lernende	passende Aufgaben und rechtzeitige Hilfen	Überwachung, Etikettierung, Kontrollverlust
Lehrkraft	Lernstände erkennen und gezielt unterstützen	vorschnelle oder verzerrte Bewertung
Schule	Lernangebote verbessern	zu weitreichende Leistungsprofile

Perspektive	Chance	Risiko
Technik	Muster in Bearbeitungen erkennen	falsche Korrelationen, Datenlecks

Bearbeitungszeiten, Fehlversuche und erreichte Punkte können dem Lernagenten helfen, passende Hinweise oder nächste Aufgaben vorzuschlagen. Gleichzeitig entstehen sensible Lernprofile, aus denen vorschnell Leistungsfähigkeit oder Verhalten abgeleitet werden könnten. Eine bloße technische Machbarkeit ist deshalb noch keine ausreichende Begründung für Speicherung und Auswertung.

12.2 Bewertungsraster

1. Zweck: Welches konkrete Problem soll gelöst werden?
2. Erforderlichkeit: Welche Daten werden wirklich benötigt?
3. Datenminimierung: Kann die Aufgabe mit weniger oder anonymisierten Daten gelöst werden?
4. Zugriff und Schutz: Wer darf was sehen oder verändern?
5. Transparenz: Verstehen betroffene Personen Nutzung und Folgen?
6. Folgen: Welche Fehlentscheidungen, Benachteiligungen oder Missbräuche sind möglich?

RDB-K12-A01 · Grundlagen: Lerndaten bewerten

Nenne je drei Chancen und Risiken der Speicherung von Bearbeitungsdauer, Fehlversuchen, Punkten und abgerufenen Hinweisen.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K12-A01_datenverantwortung.docx; Tabelle mit je drei Chancen und Risiken.

Prüfe: Alle vier Datenarten werden einbezogen; Aussagen sind konkret und aus Sicht Betroffener formuliert.

RDB-K12-A02 · Vertiefung: Personalisierung diskutieren

Bewerte einen Lernagenten, der auf Grundlage gespeicherter Lernaktivitäten automatisch Aufgaben und Hilfen auswählt. Formuliere eine Empfehlung aus Sicht der Lernenden und der Schule.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K12-A02_schutzkonzept.docx; Beurteilung in 180 bis 250 Wörtern.

Prüfe: Zweck, Datenminimierung, Rollen, Transparenz, Aufbewahrung und mindestens zwei Schutzmaßnahmen fließen in ein begründetes Urteil ein.

13. Lernmaterialien nutzen und Arbeit dokumentieren [← Inhalt](#)

Kapitelziel

Du arbeitest mit SQL-Dateien, Modellen und HTML-Übungen in VS Code und dokumentierst Ergebnisse nachvollziehbar.

13.1 Empfohlener Arbeitsablauf

1. Aufgabenstellung und benötigte Dateien in VS Code öffnen.
2. Struktur-SQL und anschließend die zugehörigen Testdaten in MySQL Workbench importieren.
3. Datenmodell und Tabellenrollen analysieren.
4. Eigene Abfragen in einer eindeutig benannten .sql-Datei bearbeiten.
5. Jede Abfrage ausführen und das Ergebnis fachlich prüfen.
6. Fehler, Korrekturen und Erkenntnisse kurz im Lernjournal dokumentieren.
7. Arbeitsdateien im vorgesehenen persönlichen Ordner sichern.

Artefakt	Empfohlene Ablage	Nicht ablegen
SQL-Abfragen	eigene .sql-Datei je Lernauftrag	Passwörter oder Zugangsdaten
Modell	exportierte Grafik; .mwb nach Vorgabe	personenbezogene Echtdaten
Ergebnisse	Screenshot oder kurze Ergebnistabelle nach Vorgabe	ungeprüfte Zufallsergebnisse
Reflexion	Word-Skript oder Lernjournal	unreflektierte KI-Komplettlösung

KI und Eigenleistung

KI darf als Lernassistent eingesetzt werden, wenn der Arbeitsauftrag dies erlaubt. Du musst jede übernommene Lösung erklären, testen und Fehler selbst eingrenzen können. Die konkrete Nutzung von Copilot oder anderen GitHub-Diensten wird erst festgelegt, wenn die schulischen Lizenzen und Rahmenbedingungen geklärt sind.

13.2 Optionale Bereitstellung über GitHub

Die Lehrkraft kann dieselben Materialien später über ein Kurs-Repository bereitstellen. Die Einführung in Git, GitHub, Templates, GitHub Education, Branches, Commits und Pull Requests erfolgt jedoch in einem gesonderten Skript. Für die Bearbeitung dieses Datenbankskripts reichen zunächst die digitale Schultasche, VS Code, MySQL Workbench und ein lokaler Ressourcenordner aus.

RDB-K13-A01 · Grundlagen: SQL-Lernjournal

Dokumentiere zu einer Abfrage: Ziel, verwendete Tabellen, JOIN-Pfad, SQL-Lösung, Testergebnis und eine mögliche Fehlerquelle.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K13-A01_lernjournal.docx; Vollständig ausgefüllter Journaleintrag.

Prüfe: Ziel, Tabellen, JOIN-Pfad, SQL, Soll-/Ist-Ergebnis und Fehlerquelle sind nachvollziehbar dokumentiert.

RDB-K13-A02 · Vertiefung: Qualitätsprüfung

Prüfe eine SQL-Lösung eines Teammitglieds anhand von Fachlogik, Lesbarkeit, Testfällen und Datenschutz. Formuliere konstruktives Feedback – unabhängig davon, über welchen technischen Weg die Datei ausgetauscht wurde.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K13-A02_reviewbogen.docx; Ausgefüllter Reviewbogen.

Prüfe: Feedback belegt Fachlogik, Lesbarkeit, Testfälle und Datenschutz mit mindestens einem konkreten Verbesserungsvorschlag.

14. Lernprojekt: Datenbank für den KI-Lernagenten [↔ Inhalt](#)

Kapitelziel

Du durchläufst den vollständigen Entwicklungsweg von der Anforderung über Modell und SQL bis zur fachlichen Bewertung.

14.1 Ausgangssituation

Ein schulischer KI-Lernagent soll Kurse, Aufgaben, Hinweise, Lernende, Bearbeitungen und Rückmeldungen verwalten. Er benötigt verlässliche Daten, um Lernstände sichtbar zu machen und passende Hilfen bereitzustellen. Gleichzeitig müssen personenbezogene Lerndaten sparsam, transparent und verantwortungsvoll verarbeitet werden.

14.2 Meilensteine und Abgaben

Meilenstein	Produkt	Qualitätskriterium
M1 Analyse	Begriffe, Regeln, offene Fragen	fachlicher Zweck klar
M2 EERM	Modell + Eine-Sätze	Schlüssel und Kardinalitäten begründet
M3 3NF	Relationenschema + Abhängigkeiten	Redundanzen und Anomalien reduziert
M4 Implementierung	.mwb + Struktur-SQL	Datentypen, PK/FK, Constraints korrekt
M5 Daten	.sql mit Testdaten	realistisch, konsistent, keine Echtdaten
M6 Auswertung	mindestens 8 Abfragen	JOIN, Aggregat, Datum, Unterabfrage

Meilenstein	Produkt	Qualitätskriterium
M7 Reflexion	Kurzbericht + Präsentation	Chancen/Risiken und Lernweg bewertet

14.3 Verbindliche Abfragen

- aktive Aufgaben mit Kurs und Schwierigkeitsstufe
- abgeschlossene Bearbeitungen in einem Zeitraum
- Anzahl der Bearbeitungen je Aufgabe und Schwierigkeitsstufe
- durchschnittlich erreichte Punkte je Kurs
- Lernende ohne abgeschlossene Bearbeitung mit LEFT JOIN
- Aufgaben über der durchschnittlichen Maximalpunktzahl
- Kurse mit mindestens fünf abgeschlossenen Bearbeitungen
- abgerufene Hinweise je Aufgabe und Hinweisstufe

RDB-K14-A01 · Grundlagen: Projekt umsetzen

Bearbeite M1 bis M6. Führe für jede Abfrage mindestens einen positiven und einen negativen Testfall durch.

Material und Abgabe: 02_SQL und 01_Aufgaben_und_Vorlagen/RDB-K14-Projektauftrag.docx; Projektprodukte M1 bis M6.

Prüfe: Alle Meilensteine sind vorhanden; jede SQL-Abfrage besitzt einen positiven und einen negativen beziehungsweise leeren Testfall.

RDB-K14-A02 · Vertiefung: Datenethik und Schutzkonzept

Entwickle ein minimales Datenkonzept für hilfreiche Lernstandsauswertungen ohne unnötig umfassende oder dauerhafte personenbezogene Lernprofile. Begründe Aufbewahrung, Rollen und mögliche Pseudonymisierung.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K14-A02_schutzkonzept.docx; Schutzkonzept in 250 bis 350 Wörtern.

Prüfe: Erforderliche Daten, Rollen, Aufbewahrung, Pseudonymisierung und Risiken sind begründet und auf das Modell bezogen.

14.4 Freiwilliger Transfer zum eigenen Webprojekt

Wenn du schneller vorankommst oder deine Projektidee bereits weiterentwickeln möchtest, kannst du eine spätere Datenbankerweiterung konzipieren. Diese Aufgabe ist eine Vertiefung

und keine Voraussetzung für die erste Abgabe des Webprojekts Ende März.

Webprojekt-Idee	Mögliche spätere Tabellen	Beispiel für eine Auswertung
Galerie	BILD, KATEGORIE, SCHLAGWORT, BILD_SCHLAGWORT	Bilder einer Kategorie nach Datum
Automat / Simulation	PRODUKT, SCHACHT, VORGANG	verkaufte Produkte und Umsatz
kleine Webanwendung	BENUTZER, EINTRAG, KATEGORIE	Einträge je Kategorie oder Zeitraum

RDB-K14-A03 · Vertiefung: Datenbank-Erweiterung entwerfen

Beschreibe, welche veränderlichen Daten dein Webprojekt später speichern könnte. Entwirf ein kleines EERM mit zwei bis vier Entitätstypen und drei sinnvollen SQL-Auswertungen. Eine technische Verbindung zwischen Website und Datenbank ist noch nicht erforderlich.

Material und Abgabe: 01_Aufgaben_und_Vorlagen/RDB-K14-A03_webprojekt_transfer.docx; EERM mit zwei bis vier Entitätstypen und drei Auswertungen.

Prüfe: Veränderliche Daten, Schlüssel, Beziehungen und drei fachlich sinnvolle SQL-Fragen sind vollständig beschrieben.

Ausblick auf den OOP-Kurs

Nach Einführung in eine objektorientierte Programmiersprache kann dieses Datenmodell in einem späteren Softwareprojekt wieder aufgegriffen werden. Denkbar ist ein gemeinsamer Kurstrainer oder die Weiterentwicklung des KI-Lernagenten. Erst dort kommen Verbindungsaufbau, Datenzugriffsschicht, Abfragen und Datenmanipulation innerhalb der Anwendung sowie die Verarbeitung der Ergebnisse hinzu. Die konkrete Programmiersprache ist nicht Bestandteil dieses RDB-Skripts.

Lernstufe	Schwerpunkt	Möglicher Transfer
BKWI 1: Webprojekt	HTML/CSS, Gestaltung und Interaktion	spätere Datenbankerweiterung nur konzipieren
BKWI 1: Relationale Datenbanken	EERM, 3NF, MySQL und SQL	Datenstrukturen und Auswertungen sicher beherrschen
BKWI 2: OOP	Klassen, Objekte, Schichten und Datenzugriff	Kurstrainer oder KI-Lernagent mit Datenhaltung
Weiterentwicklung	Agentenlogik und Lernunterstützung	Lernstände auswerten und passende Hilfen verwalten

15. Selbsttest und Prüfungsvorbereitung

↩ Inhalt

Kapitelziel

Du überprüfst deine Kompetenz entlang der Bildungsplanbereiche BPE 3.1 bis 3.5.

Bereich	Ich kann ...	sicher / üben
BPE 3.1 EERM	Ich kann Entität, Entitätstyp, Attribut, Beziehung, Kardinalität, Datentyp, PK und FK erklären und modellieren.	☐ / ☐
BPE 3.2 Normalisierung	Ich erkenne Redundanzen und Anomalien und normalisiere bis 3NF.	☐ / ☐

Bereich	Ich kann ...	sicher / üben
BPE 3.3 Implementierung	Ich erstelle Tabellen und führe CREATE, INSERT, UPDATE und DELETE konsistent aus.	☐ / ☐
BPE 3.4 Auswertung	Ich löse mehrtabellige SELECT-Abfragen mit Bedingungen, Sortierung, Funktionen, Datum, Unterabfragen und Gruppierung.	☐ / ☐
BPE 3.5 Massendaten	Ich diskutiere Chancen und Risiken personalisierter Massendaten anhand begründeter Kriterien.	☐ / ☐

15.1 Gemischte Prüfungssimulation RDB-K15-A01

1. Extrahiere aus einem betrieblichen Text Entitätstypen, Attribute und Regeln.
2. Zeichne ein EERM und begründe alle Kardinalitäten mit Einesätzen.
3. Überführe das Modell in 3NF und kennzeichne PK/FK.
4. Implementiere zwei verbundene Tabellen einschließlich Integritätsregel.
5. Füge konsistente Testdaten ein und ändere einen Datensatz sicher.
6. Formuliere eine Abfrage über mindestens drei Tabellen mit Filter und Sortierung.
7. Erstelle eine gruppierte Auswertung mit Aggregatfunktion und HAVING.
8. Löse eine Bedingung mit einer Unterabfrage.
9. Bewerte eine personalisierte Datenauswertung mit mindestens vier Kriterien.

Prüfstrategie

Lies zuerst Operator und gefordertes Produkt. Zeichne bei JOIN-Aufgaben den Tabellenpfad, bei Normalisierung die Abhängigkeiten und bei Bewertungen ein Kriterienraster. Prüfe SQL nicht nur syntaktisch, sondern anhand des erwarteten fachlichen Ergebnisses.

16. Wochenplan für 45 Unterrichtsstunden [← Inhalt](#)

Kapitelziel

Der Lernweg ist für drei Doppelstunden pro Woche angelegt. Sieben volle Wochen ergeben 42 Stunden; in Woche 8 werden drei weitere Stunden verbindlich für BPE 3 genutzt. Die verbleibenden drei Stunden dienen als Puffer, Diagnose, Leistungsfeststellung oder individuelle Vertiefung.

16.1 Systematischer Lernweg: drei Doppelstunden pro Woche

Woche	Doppelstunde 1	Doppelstunde 2	Doppelstunde 3
1	DS 1: Daten, Informationen, DBMS; Lernagent als Kontext	DS 2: Entitäten, Attribute und Schlüssel	DS 3: Beziehungen, Eine-Sätze und Kardinalitäten
2	DS 4: relationales Schema und Datentypen	DS 5: Redundanzen und Anomalien	DS 6: 1NF, 2NF und 3NF einführen
3	DS 7: Normalisierung anwenden und begründen	DS 8: EERM in MySQL Workbench erstellen	DS 9: CREATE, PK/FK, Constraints und Integrität
4	DS 10: INSERT und konsistente Testdaten	DS 11: UPDATE, DELETE und sichere Transaktionen	DS 12: SELECT, WHERE und ORDER BY

Woche	Doppelstunde 1	Doppelstunde 2	Doppelstunde 3
5	DS 13: logische Bedingungen und Testfälle	DS 14: INNER JOIN über mehrere Tabellen	DS 15: LEFT JOIN, NULL und Duplikate
6	DS 16: Aggregatfunktionen	DS 17: GROUP BY und HAVING	DS 18: Datumsbereiche und Unterabfragen
7	DS 19: Chancen und Risiken personalisierter Lerndaten	DS 20: Lernagent-Projekt – Analyse und EERM	DS 21: Projekt – 3NF und Implementierung
8	DS 22: Projekt – SQL-Auswertungen und Tests	DS 23: 1 Stunde Reflexion/Abschluss; 1 Stunde Puffer	DS 24: 2 Stunden Diagnose, Leistung oder Vertiefung

Stundenrechnung

Wochen 1 bis 7: $7 \times 6 = 42$ Stunden. Woche 8, DS 22: 2 verbindliche Stunden. Woche 8, erste Stunde von DS 23: 1 verbindliche Stunde. Damit sind die 45 Stunden der BPE 3 erfüllt. Die restlichen 3 Stunden bleiben flexibel und gleichen Ausfälle, unterschiedliches Lerntempo oder Leistungsfeststellungen aus.

16.2 Drei Vertiefungsstufen

Stufe	Ziel	Typische Unterstützung / Aufgabe
A – Fundament	verbindliche Mindestkompetenzen sicher erreichen	geführtes Beispiel, markierter JOIN-Pfad, Satzstarter, Teilmodell, SQL-Gerüst und Ergebniskontrolle

Stufe	Ziel	Typische Unterstützung / Aufgabe
B - Anwendung	Lehrplananforderungen selbstständig anwenden	vollständige Modellierungs- und SQL-Aufgaben im Lernagent-Kontext sowie Repo-Übungen
C - Vertiefung	Zusammenhänge übertragen, beurteilen und weiterentwickeln	komplexere JOIN-Ketten, alternative Lösungen, Fehleranalyse, Webprojekt-Transfer und Lernagent-Erweiterung

Einsatzregel

Alle Lernenden bearbeiten die Fundamentaufgabe. Stufe B ist das reguläre Zielniveau. Stufe C wird nach sicherer Bearbeitung oder für schneller voranschreitende Lernende eingesetzt. Vertiefung ersetzt keine noch fehlende Grundlage.

16.3 Ergebniskontrollen je Woche

Woche	Verbindlicher Nachweis	Mögliche Vertiefung
1	Begriffe, Entitätstypen, Schlüssel und Kardinalitäten korrekt	alternatives Modell vergleichen
2	Relationenschema und Normalisierung bis 3NF	Grenzen der Normalisierung beurteilen
3	funktionsfähiges Workbench-Modell mit Integritätsregeln	Löschregeln fachlich vergleichen
4	sichere DML-Befehle und einfache SELECT-Abfragen	Testfälle und Transaktionen erweitern

Woche	Verbindlicher Nachweis	Mögliche Vertiefung
5	korrekte mehrtabellige JOIN-Abfragen	LEFT JOIN und Duplikatsursachen analysieren
6	Gruppierung, Datum und Unterabfrage anwenden	zwei Lösungswege vergleichen
7	Lerndaten bewerten; Projektmodell in 3NF	Schutzkonzept und zusätzliche Entitätstypen
8	Projektabfragen testen und Lernweg reflektieren	Transfer zum Webprojekt oder späteren OOP-Kurs

17. Repo-Ressourcen und Einsatz im Unterricht [← Inhalt](#)

Kapitelziel

Du findest schnell die passende Datei und weißt, welche Materialien für Lernende, Lehrkraft oder Leistungsfeststellung gedacht sind.

Bereich im Repo	Nutzen im Unterricht	Einsatz
generated/README.md	zentraler Schüler-Einstieg	verbindlicher Startpunkt
generated/informationen/teil-b	EERM, Kardinalitäten, 3NF, Redundanz, Integrität	Erarbeitung und Wiederholung
generated/informationen/teil-c	JOIN, SELECT, WHERE, Funktionen, Datum, GROUP BY, DML	SQL-Nachschlageblätter
generated/uebungen	FoodTruckNetz und Stadtfahrradverleih mit Selbstkontrolle	Grundlagen und Vertiefung

Bereich im Repo	Nutzen im Unterricht	Einsatz
generated/ klassenarbeiten	Varianten, Modelle, Struktur- und Datendumps	Lehrkraft / Prüfungsvorbereitung
generated/lehrplaene	importierter Bildungsplan und Schlagworte	Lehrplanabgleich
.devcontainer / VS Code	reproduzierbare Entwicklungsumgebung	optional bei GitHub-/Codespaces-Weg

Qualitätshinweis zum aktuellen Repo

Einige automatisch erzeugte Markdown-Dateien enthalten beschädigte Codeblock-Markierungen oder veraltete Zielgruppen-/BPE-Bezeichnungen. Für das Skript wurden die fachlichen Inhalte neu gesetzt und korrigiert. Vor einer unmittelbaren Schülerfreigabe der Markdown-Seiten sollte die Generierung repariert und erneut validiert werden.

17.1 Finale Offline-Bereitstellung im Unterricht

Baustein	Empfehlung
Master-Kursrepo	edu-code-course-rdb bleibt fachliche, geschützte Quelle für gepflegte Unterrichtsmaterialien.
Aktueller Unterrichtsstart	Ressourcen als lokales Paket für VS Code und MySQL Workbench bereitstellen.
Spätere GitHub-Lösung	erst nach Klärung von GitHub Education, schulischer Organisation und Lizenzen festlegen

Baustein	Empfehlung
Schülerarbeit	nicht direkt im öffentlichen Master-Repository bearbeiten lassen
Eigenes Einführungsskript	Git, GitHub, Templates, Zusammenarbeit und gegebenenfalls Copilot gesondert aufbauen
Fallback	identisches ZIP mit generated/, SQL-Dumps und Modellgrafiken

Quellen und Weiterarbeit [↩ Inhalt](#)

- [Kurs-Repository edu-code-course-rdb](#)
- [Zentraler Schüler-Einstieg](#)
- [GitHub Education](#)

Lehrplangrundlage: Bildungsplan zur Erprobung für das zweijährige Berufskolleg Wirtschaftsinformatik, Wirtschaftsinformatik, 13. März 2025, insbesondere BPE 3 Relationale Datenbanken (45 Stunden). Die im Repository hinterlegte Lehrplanfassung wurde für den Kompetenzabgleich ausgewertet.

Abschluss

Du bist lernzielbereit, wenn du ein fachliches Problem modellieren, bis 3NF überführen, in MySQL implementieren, mit SQL auswerten, systematisch testen und die Folgen der Datenspeicherung begründet beurteilen kannst.

Lizenz und Kontakt [↩ Inhalt](#)

Lizenz: Creative Commons Namensnennung Nicht kommerziell Weitergabe unter gleichen Bedingungen 4.0 International CC BY NC SA 4.0.

Autorin und Rückmeldung: Christine Janischek · Hinweise zu Fehlern und Verbesserungsvorschlägen über <https://edublog.emotionalspirit.de>

Quellenhinweis: Bildungsplan BKWI Wirtschaftsinformatik vom 13. März 2025; MySQL-Dokumentation und lokal erstellte, vollständig fiktive Unterrichtsdaten. Externe Materialien sind für den Pflichtpfad nicht erforderlich.