

Datenbanken Übung

Abgegeben von:

David ...
Maxim Maul
Jonas Veit



**UNIVERSITÄT
HEIDELBERG**
ZUKUNFT
SEIT 1386

Universität Heidelberg
Sommersemester 2025

Datenbanken Blatt 1

Jonas Veit, ...

26. April 2025

Aufgabe 1.2

1. Gib die Spalte 'STNr' in der Relation 'Einschreibungen' zurück.
Ergebnis: Spalte 'STNr' in 'Einschreibungen' ohne duplikate: s1,s2,s3,s4
2. Entferne alle Studenten(STNr) aus der Relation 'Studenten', welche auch in der Relation 'Einschreibungen' vorkommen.
Ergebnis: Spalte 'STNr' von 'Student' ohne s1, s2, s3, s4 (ohne duplikate) => output: s5
3. Verknüpfe die Relationen 'Professorinnen' und 'Kursleitungen' über gleichbenannte Spalten, falls diese dort gleiche Werte aufweisen. Anschließend wird die zusammengefügte Spalte 'Name' zurückgegeben.
Ergebnis: Alle Namen von Professorinnen, welche einen Kurs leiten: Dr. Klein, Dr. Weiß, Dr. Braun
4. Selektiere alle Einträge in 'Prüfungen', welche im Raum 'R101' stattfinden und füge diese mit der Relation 'Einschreibungen' +ber gleichbenannte Spalten zusammen, falls diese die gleichen Werten aufweisen. Anschließend gib die Spalte 'STNr' zurück.
Ergebnis: s1, s3
5. Entferne alle Bezeichnungen in der Relation 'Kurse' von eingeschriebenen Informatik Studenten und gib dort dann die Kursbezeichnungen zurück.
Ergebnis: BWL Grundlagen, Maschinendesign
6. Alle Namen von eingeschriebenen Studenten, welche eine Note kleiner als 2.0 erhalten haben.
Ergebnis: Müller, Meyer
7. Gib die Namen und Bezeichnungen der Studenten die in einem Kurs

eingeschriebenen waren, welche im Raum 'R101' eine Prüfung geschrieben haben

Ergebnis: (Müller, Datenbanken) (Müller, Programmierung), (Meyer, Datenbanken), (Meyer, Programmierung)

8. Selektiere Kurse, in die mehr als ein Student eingeschrieben ist. Ergebnis: Datenbanken, Programmierung

Datenbanken Blatt 1

Jonas Veit, ...

26. April 2025

Aufgabe 1.3

a) Zuordnung der Anforderungen

- **A.1 → B.6 Anfragesprache**
Einfache Abfragen ohne Programmierung sollen möglich sein.
- **A.2 → B.4 Persistenz, B.9 Wiederherstellung, B.11 Datensicherung**
Schutz vor Datenverlust durch dauerhafte Speicherung und Wiederherstellung im Fehlerfall.
- **A.3 → B.3 Integritätssicherung**
Gewährleistung, dass nur gültige Punktwerte ($\text{Punkte} \geq 0$) gespeichert werden.
- **A.4 → B.10 Integration**
Verwaltung und Zugriff auf große Datenmengen.
- **A.5 → B.4 Persistenz**
Langfristige Speicherung der Übungsdaten.
- **A.6 → B.1 Datenunabhängigkeit, B.3 Integritätssicherung**
Vermeidung mehrfacher Speicherung durch eindeutige Identifikation.
- **A.7 → B.5 Synchronisation, B.7 Transaktionskonzept**
Ermöglicht parallelen Zugriff und gleichzeitige Änderungen ohne Datenkonflikte.
- **A.8 → B.4 Persistenz, B.8 Operationen**
Zugriff auf die Datenbank ist jederzeit möglich.

- **A.9 → B.2 Datenschutz, B.12 Sichtenkonzept**
Steuerung des Datenzugriffs auf einzelne Übungsgruppen.

b) Erfüllung durch klassische Dateisysteme (NTFS, FAT32)

- **A.1** Kein Abfragesystem, nur Dateioperationen möglich.
- **A.2** Keine garantierte Datensicherung oder Wiederherstellung ohne zusätzliche Mechanismen.
- **A.3** Keine Integritätsprüfung, beliebige Inhalte möglich.
- **A.4** Große Datenmengen können gespeichert werden.
- **A.5** Dateien bleiben bei ordentlicher Nutzung langfristig erhalten.
- **A.6** Keine Sicherstellung von Eindeutigkeit oder Vermeidung redundanter Daten.
- **A.7** Kein Mechanismus für gleichzeitigen Zugriff (Locking nur sehr rudimentär).
- **A.8** Dateien sind grundsätzlich jederzeit zugreifbar.
- **A.9** Keine gezielte Zugriffsbeschränkung auf Teilmengen von Daten (nur Dateiberechtigungen, keine Sichten).

Fazit: Dateisysteme wie NTFS oder FAT32 erfüllen nur Basisfunktionen wie Speicherung großer Datenmengen und Zugriff rund um die Uhr. Für komplexere Anforderungen wie Integrität, Synchronisation, Datenschutz oder Anfragesprachen sind relationale Datenbanksysteme notwendig.