



### KN03: Datenmanipulation und Abfragen I

- A) Daten hinzufügen (25%)
- B) Daten löschen (25%)
- C) Daten abfragen (25%)
- D) Daten verändern (25%)

## KN03: Datenmanipulation und Abfragen I

---

Beachten Sie die [allgemeinen Informationen zu den Abgaben](#).

Sie werden hier nun Daten in Ihrer Datenbank erfassen, ändern, löschen und auslesen (CRUD Operationen). In dieser Kompetenz behandeln wir **einfache** Abfragen noch ohne Verknüpfungen zwischen Collections.

Die Grundlage zu den folgenden Aufgaben finden [Sie in der Theorie](#), die **ersten beiden Kapitel**.

### A) Daten hinzufügen (25%)

---

Fügen Sie für jede Collection ein sinnvolle Menge an Datensätzen hinzu. Im Schnitt sollte dies 3-5 Datensätze sein pro Collection. Bedingungen:

- Verwenden Sie `objectId`, um das Feld `_id` zu setzen (keine Hart-kodierte Werte).
- Verwenden Sie Variablen für die ObjectIds.
- Verwenden Sie für mindestens eine Collection die `insertOne()`-Anweisung.
- Verwenden Sie für mindestens eine Collection die `insertMany()`-Anweisung.

#### Abgaben:

- Eine Skript-Datei (.js), die die Befehle enthält. Die Datei sollte einfach ausführbar sein, ohne dass Änderungen notwendig sind. Die Skript-Datei ist so formatiert, dass Sie von Auge lesbar ist (z.B. nicht alles auf einer Zeile)

### B) Daten löschen (25%)

---

Zuerst werden Sie ganze Collections rauslöschen. Dieses Skript können Sie dann als Grundlage verwenden, um "aufzuräumen" und erlauben sich damit andere Skripts auf einer leeren Datenbank auszuführen.

Erstellen Sie ein **neues** Skript (.js), welches alle Collections löscht. Verwenden Sie dazu den `collection.drop()` Befehl.

Erstellen Sie wieder ein neues Skript (.js), welches einzelne Einträge löscht. **Bedingungen:**

- Löschen Sie **ein** Datensatz aus einer Collection raus mit dem Befehl `deleteOne()`. Verwenden Sie die `_id` in der Filterung.
- Löschen Sie mehrere Datensätze aus einer Collection raus mit dem Befehl `deleteMany()`, aber Ihr Befehl darf **nicht** alle Datensätze rauslöschen. Verwenden Sie eine ODER-Verknüpfung um mehrere `_id`'s zu filtern.

**Abgaben:**

- Skript 1, welches alle Collections löscht.
- Skript 2, welches Daten teilweise löscht.

## C) Daten abfragen (25%)

---

Nun werden wir auch Daten lesen. Verwenden Sie den `find()`-Befehl für folgende Abfragen. Räumen Sie zuerst auf und löschen sie alle Collections, um dann die Daten wieder hinzuzufügen. Verwenden Sie **nie** das Feld `_id` für die Abfragen. Das haben Sie bereits in Teil B) gemacht und funktioniert hier genau gleich.

- Mindestens eine Abfrage pro Collection
- Verwenden Sie mindestens einmal eine Filterung auf ein DateTime Feld
- Verwenden Sie mindestens einmal eine ODER-Verknüpfung in der Filterung
- Verwenden Sie mindestens einmal eine UND-Verknüpfung in der Filterung, aber nicht auf der gleichen Collection wie die ODER-Verknüpfung.
- Verwenden Sie mindestens einmal eine *Regex* um einen Teilstring zu finden.
- Verwenden Sie mindestens einmal eine Projektion, bei der die `_id` **auch** ausgegeben wird
- Verwenden Sie mindestens einmal eine Projektion, bei der die `_id` **nicht** ausgegeben wird.

**Abgaben:**

- **Eine Skript**-Datei, mit den Abfragen.

## D) Daten verändern (25%)

---

Im letzten Teil werden Sie noch Daten verändern. Räumen Sie zuerst auf und löschen sie alle Collections, um dann die Daten wieder hinzuzufügen. Nun werden Sie Daten verändern mit den folgenden Bedingungen:

- Die folgenden drei Befehle werden auf unterschiedlichen Collections ausgeführt.
- Verwenden Sie mindestens einmal den Befehl `updateOne()` und die `_id` als Filterung.
- Verwenden Sie mindestens einmal den Befehl `updateMany()`, aber ohne die `_id` zu verwenden. Die Filterung sollte eine ODER-Verknüpfung verwenden und auch tatsächlich mehr als einen Datensatz verändern.
- Verwenden Sie mindestens einmal den Befehl `replaceOne()`, um ein Dokument einer Collection zu verändern.

**Abgaben:**

- **Eine Skript**-Datei, mit den drei Befehlen.