|  |  |  |  |
| --- | --- | --- | --- |
| **Übungsprotokoll**  **INSY - Informationssysteme** | | | |
|  | **Übungsdatum:**  KW 04/2021 –  KW 13/2021 | **Klasse:**  3AHIT | **Name:**  Felix Schneider |
| **Abgabedatum:**  31.03.2022 | **Gruppe:**  INSY\_2 | **Note:** |
| **Leitung:**  DI (FH) Alexander MESTL | **Mitübende:**  - | | |
| **Übungsbezeichnung**:  MongoDB | | | |

**Inhaltsverzeichnis:**

[1 Aufgabenstellung 3](#_Toc99607616)

[2 Abstract (English) 3](#_Toc99607617)

[3 Theoretische Grundlagen 3](#_Toc99607618)

[4 Übungsdurchführung 3](#_Toc99607619)

[4.1 Installation 3](#_Toc99607620)

[4.2 Service verwalten 4](#_Toc99607621)

[4.3 Nutzung 4](#_Toc99607622)

[4.4 Überprüfung 4](#_Toc99607623)

[4.5 Robomongo 4](#_Toc99607624)

[4.6 Arbeiten mit MongoDB (Teil 1) 7](#_Toc99607625)

[5 Ergebnisse 28](#_Toc99607626)

[6 Kommentar 29](#_Toc99607627)

# Aufgabenstellung

Protokolliere den Ablauf der Installation und die Verwendungsmethoden von MongoDB.

# Abstract (English)

Log the installation process and usage methods of MongoDB.

# Theoretische Grundlagen

Bei Linux gibt es einen Spiegelserver, dem wir sagen müssen, dass wir MongoDB hinzufügen wollen.

NoSQL-Grundlagen, MongoDB-Grundlagen: <https://htlkrems3500-my.sharepoint.com/personal/f_schneider_htlkrems_at/Documents/Schule/#Scripts/INSY_NoSQL.pdf> (NoSQL-Script INSY)

Die wichtigsten Begriffe in Verbindung mit NoSQL: [wichtige Begriffe INSY](https://htlkrems3500-my.sharepoint.com/personal/f_schneider_htlkrems_at/Documents/Schule/INSY/wichtige%20Begriffe%20INSY.docx?web=1)

# Übungsdurchführung

## Installation

Folgen Sie den Anweisungen von <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/#import-the-public-key-used-by-the-package-management-system>, wenn Sie MongoDB auf Debian installieren wollen. Andernfalls gibt es auf dieser Website auch die Installationsdokumentationen anderer Betriebssysteme.

Trotzdem hier die wichtigsten Schritte:

Mit folgendem Befehl kann man sich den MongoDB public GPG Key importieren.

wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add –

Mit dem Befehl können Sie eine File List erstellen:

echo "deb http://repo.mongodb.org/apt/debian buster/mongodb-org/5.0 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list

Laden Sie die Pakete vom Spiegelserver neu:

sudo apt-get update

Und installieren Sie anschließend MongoDB:

sudo apt-get install -y mongodb-org

## Service verwalten

Um MongoDB zu starten:

sudo systemctl start mongod // Dienst starten

Diese Befehle können Sie auch durchführen:

sudo systemctl status mongod // Status ansehen

sudo systemctl enable mongod // Dienst enabeln

sudo systemctl stop mongod // Dienst stoppen

sudo systemctl restart mongod // Dienst neu starten

## Nutzung

Um über das Terminal einsteigen zu können, nutzen Sie folgenden Befehl:

mongosh

## Überprüfung

Mit diesem Befehl kann man nachsehen, ob ein Dienst läuft:

sudo systemctl status mongod

Mithilfe dieses Befehls kann man nachsehen, ob die Datenbank läuft (Waiting for connections):

tail -f /var/log/mongodb/mongod.log

## Robomongo

### Robomongo installieren

Gehen Sie auf folgende Website und installieren Sie die gratis Version Robo 3T: <https://robomongo.org/download>. Entzippen Sie die Datei.

### Robomongo ausführen

Öffnen Sie die exe-Datei im /bin-Ordner im ungezippten Ordner.

### Connection erstellen

Erstellen Sie eine neue Connection mit den Standardwerte (sollte richtige Connection sein):

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Robo3T nutzen

#### neue Database anlegen und Collection hinzufügen

Dies kann mithilfe der Shell hinzufügen:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

db.createCollection(„test3“)

Oder in Robo3T:

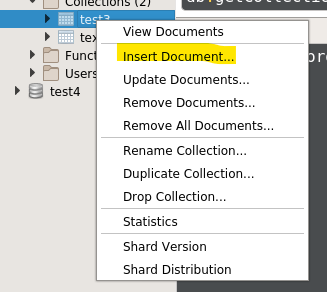
Ein Bild, das Text enthält.

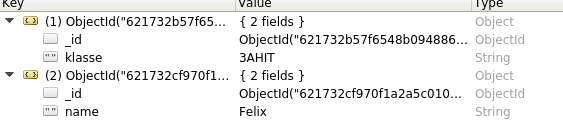
Automatisch generierte Beschreibung

#### insertOne()

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung





#### load()



#### drop()



#### Probleme mit NumberLong

Seit geraumer Zeit muss man bei NumberLong-Datentypen Anführungszeichen vor und nach der Zahl setzen, damit diese als String erkannt wird.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

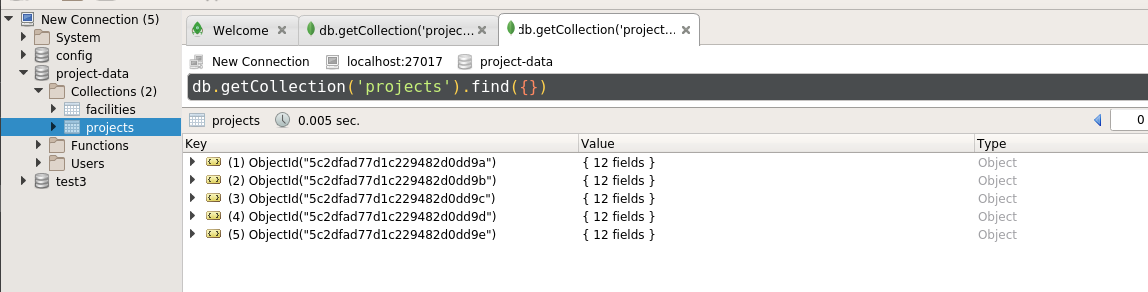
Automatisch generierte Beschreibung

#### Validation Error

Validation Errors werden nicht gut beschrieben, die Fehlermeldung ist eher mangelhaft… Trotzdem haben wir den Fehler gefunden. Die Max Length wurde überschritten, deswegen haben wir diese bei der Validation auf 100 gesetzt.



Nun endlich laden alle 5 Objekte:



## Arbeiten mit MongoDB (Teil 1) – Modellierung

[Angabe wird laufend angepasst/aktualisiert - sowohl Inhalt als auch Abgabetermin!]

In einer neuen virtuellen Maschine (vorzugsweise Ubuntu 20.04. LTS) wird MongoDB in der Version 5 installiert, der Zugriff erfolgt über die grafische Oberfläche Robo3T (von Robomongo). In weiterer Folge werden Musterdaten importiert und damit gearbeitet.

Die einzelnen Steps:

* Installation Betriebssystem und MongoDB - bis 03.02.
* Installation Robo3T und Import Musterdaten - bis 17.02.
* DDL- und DML-Aufgaben aus Dokument InsyExercise, Aufgabe 4.1 - 1. bis 3. Beispiel

### Modellierung (Schema)

#### Angabe

JS-Dateien (debitor und subproject) erstellen und modellieren.

#### Theorie

* NoSQL-Script S. 152ff

#### project.js

//-------------------------------------------------------------------------------------------

//      schema

//-------------------------------------------------------------------------------------------

db.createCollection("projects", {

     validationLevel  : "strict",

     validationAction : "error",

     validator : {

           $jsonSchema : {

                bsonType : "object",

                required : [

                    "title",

                    "projectType",

                    "projectState",

                    "description",

                    "projectBegin",

                    "isFWFSponsered",

                    "isFFGSponsered",

                    "isEUSponsered",

                    "isSmallProject",

                    "subprojects",

                    "fundings"

                ],

                additionalProperties : true,

                properties : {

                     title : {

                         bsonType  : "string",

                         minLength : 3,

                         maxLength : 50,

                         description : "the title of the project"

                     },

                     projectType : {

                        enum : [

                            "REQUEST\_FUNDING\_PROJECT",

                            "RESEARCH\_FUNDING\_PROJECT",

                            "MANAGEMENT\_PROJECT"

                        ],

                        description : "enum to describe projecttypes"

                     },

                     projectState : {

                         enum : [

                             "CREATED",

                             "IN\_APPROVEMENT",

                             "APPROVED"

                         ],

                         description : "enum to describe the project state"

                     },

                     description : {

                         bsonType  : "string",

                         minLength : 0,

                         maxLength : 4000,

                         description : "a short description of the project"

                     },

                     projectBegin : {

                        bsonType : "date"

                     },

                     isFWFSponsered : {

                        bsonType : "bool",

                        description : "indicates if the project is sponsered by the fwf"

                     },

                     isFFGSponsered : {

                        bsonType : "bool",

                        description : "indicates if the project is sponsered by the ffg"

                     },

                     isEUSponsered : {

                        bsonType : "bool",

                        description : "indicates if the project is sponsered by the eu"

                     },

                     isSmallProject : {

                        bsonType : "bool",

                        description : "indicates that the funding of the project don't exceeds 5000 Euro"

                     },

                     subprojects : {

                         bsonType : "array",

                         items : {

                            bsonType : "object",

                            required : ["\_id","title"],

                            additionalProperties : false,

                            properties : {

                               \_id : {

                                  bsonType : "objectId"

                               },

                               title : {

                                  bsonType  : "string",

                                  minLength : 3,

                                  maxLength : 100,

                                  description : "the description of the subproject"

                               }

                            }

                         }

                     },

                     fundings : {

                        bsonType : "array",

                        items : {

                           bsonType : "object",

                           required : ["\_id", "debitorName", "amount"],

                           additionalProperties : false,

                           properties : {

                              \_id : {

                                 bsonType : "objectId"

                              },

                              debitorName : {

                                 bsonType  : "string",

                                 minLength : 5,

                                 maxLength : 100

                              },

                              amount : {

                                 bsonType : "long",

                                 minimum : 0

                              }

                           }

                        }

                     }

                }

           }

     }

});

#### facilities.js

db.createCollection("facilities", {

     validationLevel  : "strict",

     validationAction : "error",

     validator : {

           $jsonSchema : {

                bsonType : "object",

                required : [

                   "\_id",

                   "name",

                   "code",

                   "projects"

                ],

                properties : {

                     \_id   : {

                         bsonType : "objectId"

                     },

                     name : {

                         bsonType  : "string",

                         minLength : 3,

                         maxLength : 100

                     },

                     code : {

                         bsonType  : "string",

                     },

                     projects : {

                        bsonType : "array",

                        items : {

                           bsonType : "object",

                           required : ["project\_id", "title"],

                           properties : {

                              project\_id : {

                                 bsonType : "objectId"

                              },

                              title : {

                                 bsonType : "string"

                              }

                           }

                        }

                     }

                }

           }

     }

});

#### subproject.js

//-------------------------------------------------------------------------------------------

//      schema

//-------------------------------------------------------------------------------------------

db.createCollection("subprojects", {

     validationLevel  : "strict",

     validationAction : "error",

     validator : {

           $jsonSchema : {

                bsonType : "object",

                required : [

                    "title",

                    "description",

            "appliedResearch",

            "theoreticalResearch",

            "focusResearch",

                    "project",

            "facilities"

                ],

                additionalProperties : true,

                properties : {

                     title : {

                         bsonType  : "string",

                         minLength : 3,

                         maxLength : 100,

                         description : "the title of the subproject"

                     },

                     description : {

                         bsonType  : "string",

                         minLength : 0,

                         maxLength : 4000,

                         description : "a short description of the subproject"

                     },

                     appliedResearch : {

                        bsonType   : "int",

            minimum    : 0,

            maximum    : 100

             },

                     theoreticalResearch : {

                        bsonType   : "int",

            minimum    : 0,

            maximum    : 100

             },

                     focusResearch : {

                        bsonType   : "int",

            minimum    : 0,

            maximum    : 100

             },

             project : {

            bsonType   : "object",

            required   : [ "\_id", "title" ],

            additionalProperties: false,

            properties: {

                \_id: {

                bsonType: "objectId"

                },

                title: {

                bsonType: "string",

                minLength: 3,

                maxLength: 100

                }

            }

             },

             facilities : {

            bsonType   : "object",

            required   : [ "\_id", "name" ],

            additionalProperties: false,

            properties: {

                \_id: {

                bsonType: "objectId"

                },

                name: {

                bsonType: "string",

                minLength: 3,

                maxLength: 100

                }

            }

             }

                }

           }

     }

});

#### debitor.js

//-------------------------------------------------------------------------------------------

//      schema

//-------------------------------------------------------------------------------------------

db.createCollection("debitors", {

     validationLevel  : "strict",

     validationAction : "error",

     validator : {

           $jsonSchema : {

                bsonType : "object",

                required : [

                    "name",

                    "description",

            "fundings"

                ],

                additionalProperties : true,

                properties : {

                     name : {

                         bsonType  : "string",

                         minLength : 3,

                         maxLength : 100,

                         description : "the title of the subproject"

                     },

                     description : {

                         bsonType  : "string",

             minLength : 0,

             maxLength : 4000,

                         description : "a short description of the subproject"

                     },

                     fundings : {

                        bsonType : "array",

                        items : {

                           bsonType : "object",

                           required : ["\_id", "title", "amount", "date"],

                           additionalProperties : false,

                           properties : {

                              \_id : {

                                 bsonType : "objectId"

                              },

                  title : {

                 bsonType : "string",

                 minLength : 3,

                 maxLength : 50

                  },

                              amount : {

                                 bsonType : "long",

                 minimum  : 0

                              },

                              date : {

                                 bsonType : "date"

                              }

                           }

                        }

                     }

                }

           }

     }

});

### Dokumente einfügen

#### Angabe

Für jede JS-Datei 5 Dokumente erstellen.

#### Theorie

* NoSQL-Script S. 170f

#### project.js

//-------------------------------------------------------------------------------------------

//       insert statements

//-------------------------------------------------------------------------------------------

db.projects.insertMany([

//1. Project

{

  \_id   : ObjectId("5c2dfad77d1c229482d0dd9a"),

  title :        "Project 1",

  projectType  : "REQUEST\_FUNDING\_PROJECT",

  projectState : "APPROVED",

  description  : "Projekt zur Planung von Produktionsplanungssystemen",

  projectBegin : new Date ('Jun 23, 2012'),

  isFWFSponsered : true,

  isFFGSponsered : false,

  isEUSponsered  : false,

  isSmallProject : false,

  subprojects : [

  {

      \_id  : ObjectId("874632936283294250329326"),

      title: "Subproject 3"

  },{

      \_id  : ObjectId("874632936283294250329327"),

      title: "Subproject 4"

  }

  ],

  fundings    : [

    {

            \_id  :  ObjectId("974632936283294250329328"),

            debitorName : "Debitor 3",

            amount : NumberLong(600000)

    },

    {

            \_id  :  ObjectId("974632936283294250329330"),

            debitorName : "Debitor 5",

            amount : NumberLong(30000)

    }

  ]

}

//2. Project

,{

  \_id   : ObjectId("5c2dfad77d1c229482d0dd9b"),

  title :        "Project 2",

  projectType  : "REQUEST\_FUNDING\_PROJECT",

  projectState : "APPROVED",

  description  : "Projekt zur Planung von Produktionsplanungssystemen",

  projectBegin : new Date ('Jun 23, 2012'),

  isFWFSponsered : true,

  isFFGSponsered : false,

  isEUSponsered  : false,

  isSmallProject : false,

  subprojects : [

  {

      \_id  : ObjectId("874632936283294250329328"),

      title: "Subproject 5"

  },{

      \_id  : ObjectId("874632936283294250329332"),

      title: "Subproject 9"

  }

  ],

  fundings    : [

    {

            \_id  :  ObjectId("974632936283294250329329"),

            debitorName : "Debitor 4",

            amount : NumberLong(150000)

    },

    {

            \_id  :  ObjectId("974632936283294250329326"),

            debitorName : "Debitor 1",

            amount : NumberLong(120000)

    }

  ]

}

//3. Project

,{

  \_id   : ObjectId("5c2dfad77d1c229482d0dd9c"),

  title :        "Project 3",

  projectType  : "REQUEST\_FUNDING\_PROJECT",

  projectState : "APPROVED",

  description  : "Projekt zur Planung von Produktionsplanungssystemen",

  projectBegin : new Date ('Jun 23, 2012'),

  isFWFSponsered : true,

  isFFGSponsered : false,

  isEUSponsered  : false,

  isSmallProject : false,

  subprojects : [

  {

      \_id  : ObjectId("874632936283294250329324"),

      title: "Subproject 1"

  },{

      \_id  : ObjectId("874632936283294250329331"),

      title: "Subproject 8"

  }

  ],

  fundings    : [

    {

            \_id  :  ObjectId("974632936283294250329326"),

            debitorName : "Debitor 1",

            amount : NumberLong(67000)

    },

    {

            \_id  :  ObjectId("974632936283294250329328"),

            debitorName : "Debitor 3",

            amount : NumberLong(300000)

    },

    {

            \_id  :  ObjectId("974632936283294250329330"),

            debitorName : "Debitor 5",

            amount : NumberLong(350000)

    }

  ]

}

//4. Project

,{

  \_id   : ObjectId("5c2dfad77d1c229482d0dd9d"),

  title :        "Project 4",

  projectType  : "REQUEST\_FUNDING\_PROJECT",

  projectState : "APPROVED",

  description  : "Projekt zur Planung von Produktionsplanungssystemen",

  projectBegin : new Date ('Jun 23, 2012'),

  isFWFSponsered : true,

  isFFGSponsered : false,

  isEUSponsered  : false,

  isSmallProject : false,

  subprojects : [{

      \_id  : ObjectId("874632936283294250329330"),

      title: "Subproject 7"

  },{

      \_id  : ObjectId("874632936283294250329333"),

      title: "Subproject 10"

  }

  ],

  fundings    : [

    {

            \_id  :  ObjectId("974632936283294250329327"),

            debitorName : "Debitor 2",

            amount : NumberLong(12000)

    },

    {

            \_id  :  ObjectId("974632936283294250329330"),

            debitorName : "Debitor 5",

            amount : NumberLong(1000000)

    }

  ]

}

//5. Project

,{

  \_id   : ObjectId("5c2dfad77d1c229482d0dd9e"),

  title :        "Project 5",

  projectType  : "REQUEST\_FUNDING\_PROJECT",

  projectState : "APPROVED",

  description  : "Projekt zur Planung von Produktionsplanungssystemen",

  projectBegin : new Date ('Jun 23, 2012'),

  isFWFSponsered : true,

  isFFGSponsered : false,

  isEUSponsered  : false,

  isSmallProject : false,

  subprojects : [{

      \_id  : ObjectId("874632936283294250329325"),

      title: "Subproject 2"

  },{

      \_id  : ObjectId("874632936283294250329329"),

      title: "Subproject 6"

  }

  ],

  fundings    : [

    {

            \_id  :  ObjectId("974632936283294250329326"),

            debitorName : "Debitor 1",

            amount : NumberLong(300000)

    },

    {

            \_id  :  ObjectId("974632936283294250329327"),

            debitorName : "Debitor 2",

            amount : NumberLong(10000)

    },

    {

            \_id  :  ObjectId("974632936283294250329329"),

            debitorName : "Debitor 4",

            amount : NumberLong(90000)

    }

  ]

},

]);

#### facilities.js

db.facilities.insertMany([{

     \_id  : ObjectId("874632936283294250329002"),

     name : "Institut für Angewandte Mathematik",

     code : "123.456.231",

     projects  : [{

        project\_id   : ObjectId("5c2dfad77d1c229482d0dd9c"),

        title : "Project 3"

     }]

}, {

     \_id  : ObjectId("874632936283294250329001"),

     name : "Institut für Wirtschaftsinformatik",

     code : "123.456.789",

     projects  : [{

        project\_id   : ObjectId("5c2dfad77d1c229482d0dd9a"),

        title : "Project 1"

     },{

         project\_id   : ObjectId("5c2dfad77d1c229482d0dd9d"),

         title : "Project 4"

     },{

         project\_id   : ObjectId("5c2dfad77d1c229482d0dd9e"),

         title : "Project 5"

     }]

}, {

     \_id  : ObjectId("874632936283294250329003"),

     name : "Institut für Softwareentwicklung",

     code : "123.456.789",

     projects  : [{

        project\_id   : ObjectId("5c2dfad77d1c229482d0dd9a"),

        title : "Project 1"

     },{

        project\_id   : ObjectId("5c2dfad77d1c229482d0dd9b"),

        title : "Project 2"

     },{

        project\_id   : ObjectId("5c2dfad77d1c229482d0dd9c"),

        title : "Project 3"

     }]

}, {

      \_id  : ObjectId("874632936283294250329004"),

      name : "Institut für Analysis",

      code : "123.321.789",

      projects  : [{

         project\_id   : ObjectId("5c2dfad77d1c229482d0dd9b"),

         title : "Project 2"

      },{

         project\_id   : ObjectId("5c2dfad77d1c229482d0dd9d"),

         title : "Project 4"

      },{

    project\_id   : ObjectId("5c2dfad77d1c229482d0dd9e");

    title : "Project 5";

      }]

}

]);

#### subproject.js

// InsertMany

db.subprojects.insertMany([

    {

        \_id:        ObjectId("874632936283294250329324"),

        title:      "Subproject 1",

        description:    "Once upon a time some dude thought he should make this subproject...",

        appliedResearch:    92,

        theoreticalResearch:    82,

        focusResearch:      58,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9c"),

            title:  "Project 3"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329002"),

            name:   "Institut für Angewandte Mathematik"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329325"),

        title:      "Subproject 2",

        description:    "Once upon a time some guy thought he should make this subproject...",

        appliedResearch:    19,

        theoreticalResearch:    93,

        focusResearch:      28,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9e"),

            title:  "Project 5"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329001"),

            name:   "Institut für Wirtschaftsinformatik"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329326"),

        title:      "Subproject 3",

        description:    "Once upon a time some person thought he should make this subproject...",

        appliedResearch:    14,

        theoreticalResearch:    8,

        focusResearch:      38,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9a"),

            title:  "Project 1"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329001"),

            name:   "Institut für Wirtschaftsinformatik"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329327"),

        title:      "Subproject 4",

        description:    "Once upon a time some schurke thought he should make this subproject...",

        appliedResearch:    28,

        theoreticalResearch:    40,

        focusResearch:      29,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9a"),

            title:  "Project 1"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329003"),

            name:   "Institut für Softwareentwicklung"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329328"),

        title:      "Subproject 5",

        description:    "Once upon a time some monte thought he should make this subproject...",

        appliedResearch:    93,

        theoreticalResearch:    82,

        focusResearch:      83,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9b"),

            title:  "Project 2"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329003"),

            name:   "Institut für Softwareentwicklung"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329329"),

        title:      "Subproject 6",

        description:    "Once upon a time some monte thought he should make this subproject...",

        appliedResearch:    45,

        theoreticalResearch:    56,

        focusResearch:      87,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9e"),

            title:  "Project 5"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329004"),

            name:   "Institut für Analysis"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329330"),

        title:      "Subproject 7",

        description:    "Once upon a time some monte thought he should make this subproject...",

        appliedResearch:    15,

        theoreticalResearch:    94,

        focusResearch:      54,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9d"),

            title:  "Project 4"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329004"),

            name:   "Institut für Analysis"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329331"),

        title:      "Subproject 8",

        description:    "Once upon a time some monte thought he should make this subproject...",

        appliedResearch:    37,

        theoreticalResearch:    72,

        focusResearch:      82,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9c"),

            title:  "Project 3"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329003"),

            name:   "Institut für Softwareentwicklung"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329332"),

        title:      "Subproject 9",

        description:    "Once upon a time some monte thought he should make this subproject...",

        appliedResearch:    15,

        theoreticalResearch:    22,

        focusResearch:      11,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9b"),

            title:  "Project 2"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329004"),

            name:   "Institut für Analysis"

        }

    },

    {

        \_id:        ObjectId("874632936283294250329333"),

        title:      "Subproject 10",

        description:    "Once upon a time some monte thought he should make this subproject...",

        appliedResearch:    19,

        theoreticalResearch:    75,

        focusResearch:      38,

        project: {

            \_id:    ObjectId("5c2dfad77d1c229482d0dd9d"),

            title:  "Project 4"

        },

        facilities: {

            \_id:    ObjectId("874632936283294250329001"),

            name:   "Institut für Wirtschaftsinformatik"

        }

    }

]);

#### debitor.js

db.debitors.insertMany([

    //1. Debitor

    {

        \_id: ObjectId("974632936283294250329326"),

        name: "Debitor 1",

        description: "some debitor",

        fundings: [

            {

                \_id: ObjectId("5c2dfad77d1c229482d0dd9b"),

                title: "Project 2",

                amount: NumberLong(120000),

                date: new Date('Oct 26, 2012')

            },{

                \_id: ObjectId("5c2dfad77d1c229482d0dd9c"),

                title: "Project 3",

                amount: NumberLong(67000),

                date: new Date('Sep 9, 2017')

            },{

                \_id: ObjectId("5c2dfad77d1c229482d0dd9e"),

                title: "Project 5",

                amount: NumberLong(300000),

                date: new Date('Aug 17, 2018')

            }

        ]

    },

    //2. Debitor

    {

        \_id: ObjectId("974632936283294250329327"),

        name: "Debitor 2",

        description: "some other debitor",

        fundings: [

            {

                \_id: ObjectId("5c2dfad77d1c229482d0dd9d"),

                title: "Project 4",

                amount: NumberLong(12000),

                date: new Date('Sep 4, 2010')

            },{

                \_id: ObjectId("5c2dfad77d1c229482d0dd9e"),

                title: "Project 5",

                amount: NumberLong(10000),

                date: new Date('Jan 09, 2019')

            }

        ]

    },

    //3. Debitor

    {

        \_id: ObjectId("974632936283294250329328"),

        name: "Debitor 3",

        description: "this debitor",

        fundings: [

            {

                \_id: ObjectId("5c2dfad77d1c229482d0dd9a"),

                title: "Project 1",

                amount: NumberLong(600000),

                date: new Date('Oct 5, 2015')

            },{

                \_id: ObjectId("5c2dfad77d1c229482d0dd9c"),

                title: "Project 3",

                amount: NumberLong(300000),

                date: new Date('Feb 20, 2005')

            }

        ]

    },

    //4. Debitor

    {

        \_id: ObjectId("974632936283294250329329"),

        name: "Debitor 4",

        description: "that other debitor",

        fundings: [

            {

                \_id: ObjectId("5c2dfad77d1c229482d0dd9b"),

                title: "Project 2",

                amount: NumberLong(150000),

                date: new Date('Feb 6, 2019')

            },{

                \_id: ObjectId("5c2dfad77d1c229482d0dd9e"),

                title: "Project 5",

                amount: NumberLong(90000),

                date: new Date('Jan 13, 2013')

            }

        ]

    },

    //5. Debitor

    {

        \_id: ObjectId("974632936283294250329330"),

        name: "Debitor 5",

        description: "one debitor",

        fundings: [

            {

                \_id: ObjectId("5c2dfad77d1c229482d0dd9a"),

                title: "Project 1",

                amount: NumberLong(30000),

                date: new Date('Jul 04, 2022')

            },{

                \_id: ObjectId("5c2dfad77d1c229482d0dd9c"),

                title: "Project 3",

                amount: NumberLong(350000),

                date: new Date('Sep 29, 2021')

            },{

                \_id: ObjectId("5c2dfad77d1c229482d0dd9d"),

                title: "Project 4",

                amount: NumberLong(1000000),

                date: new Date('Oct 2, 2019')

            }

        ]

    }

]);

### Dokumente bearbeiten

#### Angabe

Bestimmte Attribute bzw. Felder verändern und updaten, löschen oder erstellen.

#### Theorie

* NoSQL-Script S. 171ff

#### update1.js

Jedem Projekt ein Enddatum geben, das Rating ersetzen und einen Partner setzen (erstellen).

db.projects.updateMany(

    {},

    {

        $set: {

            projectEnd: new Date('Jan 01, 2020'),

            rating: 5,

            partners: [ { name: "TU Wien" } ]

        }

    }

);

#### update2.js

Alle REQUEST\_FUNDING\_PROJECT – Projekte EU-gesponsert markieren.

db.projects.updateMany(

    {

        projectType : "REQUEST\_FUNDING\_PROJECT"

    },

    {

        $set: { isEUSponsered: true }

    }

);

#### update3.js

Das rating zu projectRating renamen, weil da ein Fehler beim Schema passiert ist.

db.projects.updateMany(

    {},

    {

        $rename: { rating: "projectRating" }

    }

);

#### update4.js

Zwei Partner hinzufügen und einen wieder entfernen, weil der doch nicht hinzugehört…

db.projects.updateMany(

    {},

    {

        $push: {

            partners: {

                $each: [

                    { name: "HTL Krems" },

                    { name: "TU Graz" }

                ]

            }

        }

    }

);

db.projects.updateMany(

    {},

    {

        $pull: {

            partners: {

                name: "HTL Krems"

            }

        }

    }

);

## Arbeiten mit MongoDB (Teil 2) – Abfragen

### limit, sort, find – der easy Anfang

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Person, Screenshot enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Mit den Updates hinzugefügt bekommt man ebenfalls keine Ergebnisse:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Mit den Updates hinzugefügt bekommt man sogar Ergebnisse:

Ein Bild, das Text enthält.

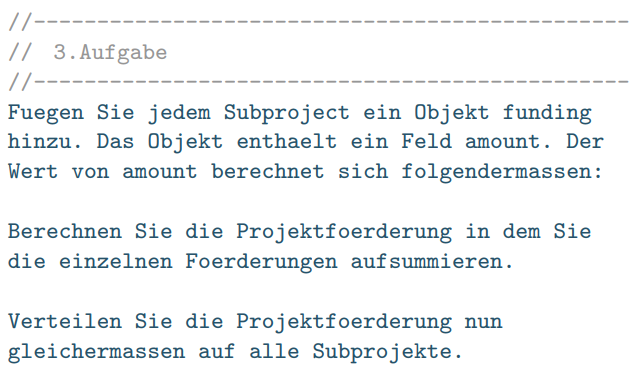
Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



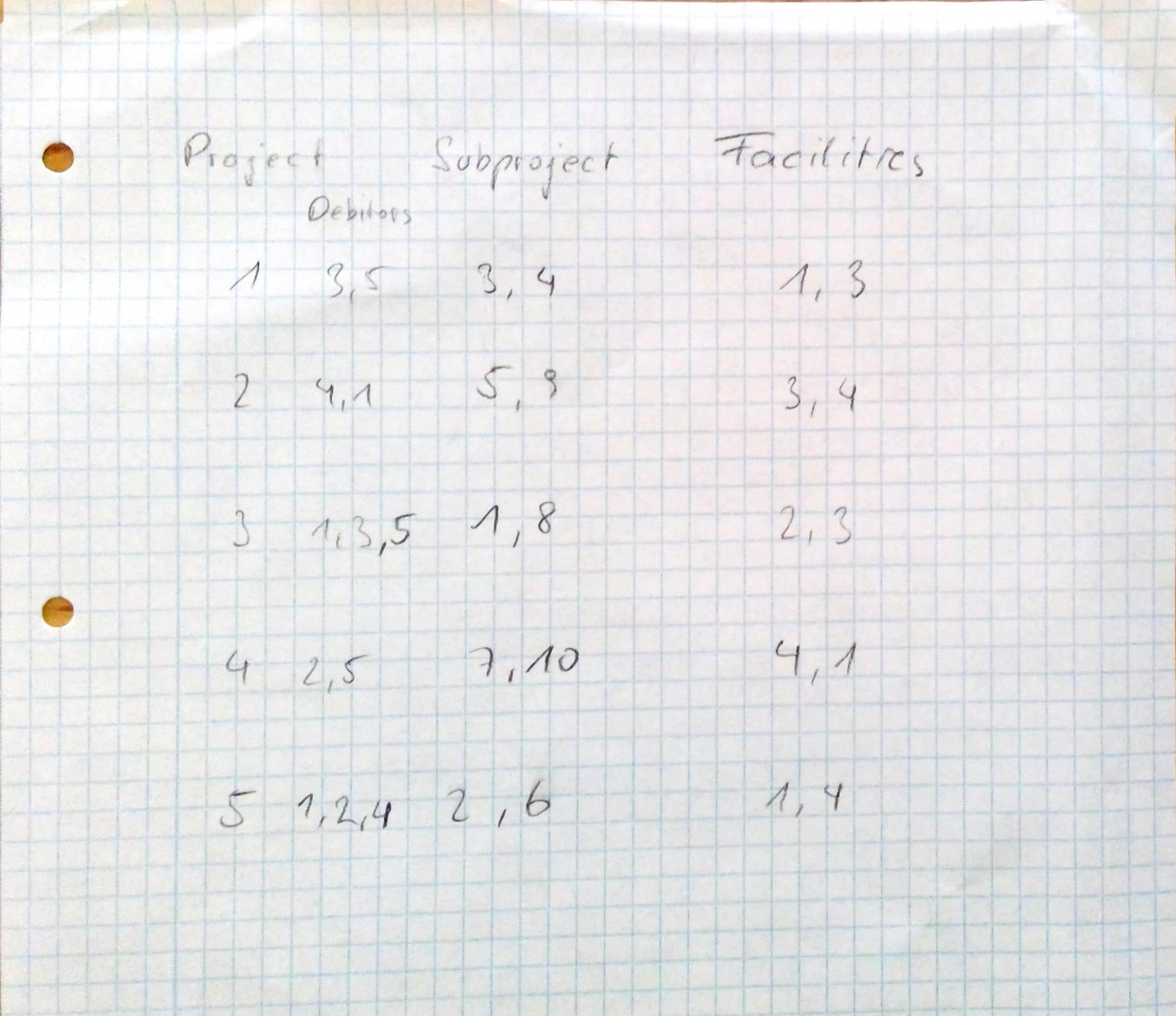
# Ergebnisse

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Die Aufgaben wurden erfüllt und die Datenbank enthält die richtigen Dokumente (TU Graz kommt 2x vor, weil die JS-Datei (update4.js) 2x ausgeführt wurde…).

# Kommentar



Falls Sie nachvollziehen wollen, welche Projects welche Subprojects, Facilities und so haben…