

Name:

Vorname:

Geburtsdatum:

Matrikelnummer:

Platz:

Klausur
Informatische Werkzeuge in den
Geistes- und Sozialwissenschaften 2

27. Juli 2023

	To be used for grading, do not write here								
prob.	1.1	2.1	2.2	2.3	3.1	3.2	4.1	Sum	grade
total	13	6	8	8	8	9	8	60	
reached									

Die „Lösungen“ der Aufgaben in diesem Dokument sollen den Studierenden als Anfangspunkt für die Beantwortung der Aufgaben dienen. Trotz aller Bemühungen kann es zu Unvollständigkeiten oder sogar Fehlern in den „Lösungen“ kommen. Da die Korrektur und Benotung der gestellten Aufgaben niemals lediglich auf einen „Vergleich mit der Musterlösung“ hinausläuft, ist dies auch nicht so schlimm. In jedem Fall sollten die Studierenden die Lösungen nachvollziehen und im Prinzip mittels des Lehrstoffs selbst verifizieren können.

Sollten Sie „Lösungen“ finden, die Sie nicht verstehen oder sogar für fehlerhaft halten diskutieren Sie diese am besten mit den Tutor*innen oder auf dem Kursforum und benachrichtigen Sie die Lehrenden; wir werden sie dann gegebenenfalls baldigst korrigieren.

1 Datenbanken

Aufgabe 1.1 (Verwaltung einer Datenbank für Energieunternehmen)

13 Pkt

Gegeben sind die SQL-Tabellen Anlagen und Firmen wie unten dargestellt. Die Spalten Anlage und Firma sind dabei *Primärschlüssel* ("Primary Key"). In der Tabelle Anlagen ist Betreiber ein *Fremdschlüssel* ("Foreign Key") der auf die Spalte Firma in der Tabelle Firmen verweist.

13 Min

Anlage	Betreiber	Typ	Output
ZWX-012	A-Corp	Wind	340
A56BK33	C-Energy	Solar	2
C001BQ0	D Gmbh	Solar	780
B759A09	Firma B	Kohle	330
...

Firma	Gegründet	Mitarbeitende
A-Corp	2001	44273
Firma B	1980	8777
C-Energy	2022	4
D GmbH	1999	509
...

1. Geben Sie einen SQL-Befehl an, mit dem die Tabelle Anlagen komplett (d.h. inklusive aller Einträge) gelöscht werden kann.

Lösung: Der Befehl lautet wie folgt:

```
DROP TABLE Anlagen;
```

2. Geben Sie einen SQL-Befehl an, mit dem die Tabelle Anlagen wie beschrieben wieder angelegt werden kann. Sie müssen die Tabelle nicht befüllen.

Lösung: Der Befehl lautet wie folgt:

```
CREATE TABLE Anlagen (
  Anlage TEXT PRIMARY KEY,
  Betreiber TEXT,
  Typ TEXT,
  Output INTEGER,
  FOREIGN KEY (Betreiber) REFERENCES Firmen(Firma)
);
```

3. Strom aus fossilen Brennstoffen wird verboten und Sie müssen Ihre Datenbank anpassen. Geben Sie einen SQL-Befehl an, mit dem *alle* Einträge mit dem Typ Kohle aus der Tabelle Anlagen gelöscht werden.

Lösung: Der Befehl lautet wie folgt:

```
DELETE FROM TABLE Anlagen WHERE Typ = 'Kohle';
```

4. Sie sind nun aus einem nicht näher genannten Grund an dem Output einer Anlage pro Mitarbeitenden der Betreiberfirma interessiert. Geben Sie einen SQL-Befehl an, der jeweils (nur) den Output einer Anlage und die Anzahl der Mitarbeitenden der Firma abfragt. Führen Sie dafür in Ihrer Anfrage einen angemessenen „Inner Join“ mit den beiden Tabellen durch.

Lösung:

Der Befehl lautet (zum Beispiel) wie folgt:

```
SELECT A.Output, F.Mitarbeitende FROM ANLAGEN A INNER JOIN FIRMEN F ON A.Betreiber = F.Firma;
```

2 Bild

Aufgabe 2.1 (Abgesicherter Zugriff auf Pixel)

6 Pkt

Schreiben Sie eine Python-Funktion `getPixelSafe()`, die ein (Pillow-)Bild `img`, sowie zwei Integer (für X- und Y-Koordinate) als Parameter entgegen nimmt. 6 Min

Sind beide Koordinaten passend für das Bild, so sollen die RGB-Werte des Bildes an der gegebenen Koordinate zurück gegeben werden.

Ist jedoch mindestens eine der Koordinaten zu groß oder zu klein für das gegebene Bild (z.B. eine X-Koordinate von -5 oder 400 bei einem Bild, das nur 200 Pixel breit ist), so soll `(0, 0, 0)` zurück gegeben werden.

Hinweis: Ein paar nützliche Informationen finden Sie im Folgenden:

- Sie können den Wert eines Pixels in Pillow-Bildern wie folgt auslesen:
`img.getpixel((x, y))` Gibt ein 3-Tupel `(r, g, b)` zurück.
 - Die Breite bzw. Höhe eines Pillow-Bildes können Sie mit `img.width` bzw. `img.height` abfragen.
-

Lösung: Hier ist eine mögliche *Implementation*:

```
def getPixelSafe(img, x, y):
    """Returns a pixel from an image,
    or (0,0,0) if coordinates are out of bounds"""
    if x >= img.width or y >= img.height or x < 0 or y < 0:
        return (0,0,0)
    else:
        return img.getpixel((x, y))
```

Aufgabe 2.2 (Farbe der Nachbarschaft eines Pixels)

8 Pkt

Schreiben Sie eine Python-Funktion `neighbourhood_average()`, die ein Bild `img`, sowie zwei Integer (für X- und Y-Koordinate) als Parameter entgegen nimmt und die durchschnittliche Farbe der Nachbarschaft um die gegebene Koordinate zurück gibt. 8 Min

Unter der “durchschnittlichen Farbe” verstehen wir hier ein Tripel aus den jeweils gemittelten Rot-, Grün- und Blauwerten aller Nachbarschaftspixel.

Die Nachbarschaft eines Pixels ist das Pixel selbst und alle benachbarten Pixel, horizontal, vertikal und diagonal. Wenn eine Nachbarschaft eines Pixels über die Grenzen des eigentlichen Bildes hinaus geht, so haben diese Pixel eine Farbe von (0, 0, 0).

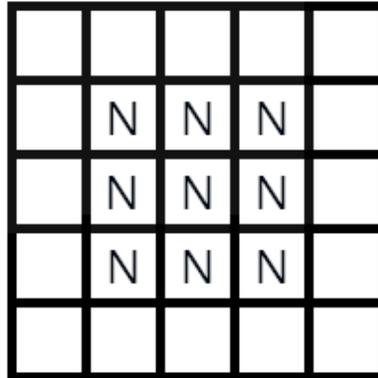


Abbildung 1: Raster mit Koordinaten (0,0) bis (4,4). Für die Koordinate (2,2) ist die Nachbarschaft genau die Felder, die mit N markiert sind.

Hinweis: Sie können für die Lösung dieser Aufgabe davon ausgehen, dass Ihnen die Funktion `getPixelSafe()` aus der vorherigen Aufgabe zur Verfügung steht (auch wenn Sie diese nicht erfolgreich bearbeitet haben).

Lösung: Hier ist eine mögliche *Implementation:*

```
def neighbourhood_average(img, x, y):
    """Returns the average colour in the neighbourhood
    of the given coordinate in the given image"""

    n_red = 0
    n_green = 0
    n_blue = 0

    for x_prime in [x-1,x,x+1]:
        for y_prime in [y-1,y,y+1]:

            r, g, b = getPixelSafe(img, x_prime, y_prime)
            n_red += r
            n_green += g
            n_blue += b

    return (int(n_red / 9), int(n_green / 9), int(n_blue / 9))
```

Aufgabe 2.3 (Pixelisierung eines Bildes)

Schreiben Sie eine Python-Funktion `pixelise()`, die ein (Pillow-)Bild `img` als Parameter entgegen nimmt. Der Rückgabewert soll eine verpixelte Version des Bildes sein, in der jedem Pixel die durchschnittliche Farbe der Nachbarschaft im Original zugewiesen wurde.

8 Pkt

8 Min

Hinweis: Sie können für die Lösung dieser Aufgabe davon ausgehen, dass Ihnen die Funktion `neighbourhood_average()` aus der vorherigen Aufgabe zur Verfügung steht (auch wenn Sie diese nicht erfolgreich bearbeitet haben).

Sie können den Wert eines Pixels in einem Pillow-Bild wie folgt setzen:

```
img.putpixel((x, y), value)
```

Sie können für diese Aufgabe davon ausgehen, dass Ihr Bild Dimensionen von 200x200 Pixeln hat.

Lösung: Hier ist eine mögliche *Implementation*:

```
def pixelise(img):
    """Returns a pixelised version of the given image."""
    img2 = img.copy()

    for x in range(1, img2.width, 3) :
        for y in range(1, img2.height, 3) :
            colour = neighbourhood_average(img, x, y)

            for x_prime in [x-1,x,x+1]:
                for y_prime in [y-1,y,y+1]:
                    # These checks are important, otherwise we run out of bounds.
                    if x <= x_prime and x_prime < img2.width and 0 <= y_prime and y_prime < img2.height:
                        img2.putpixel((x_prime, y_prime), colour)

    return img2
```

3 Semantic Web & Kulturelles Erbe

Aufgabe 3.1 (Semantische Netze)

8 Pkt

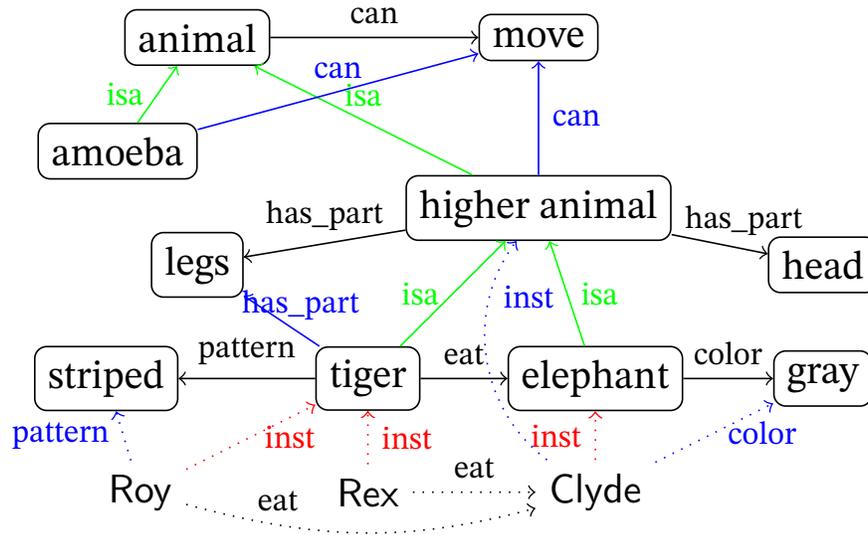
Zeichnen Sie ein semantisches Netz nach Ihren eigenen Vorstellungen. Das Netz soll mindestens 8 Knoten und 10 Kanten enthalten, davon mindestens jeweils 3 Inklusionen (*isa*-Kanten) und Instanziierungen (*inst*-Kanten).

8 Min

Geben Sie außerdem jeweils drei Subjekt/Prädikat/Objekt-Tripel aus der TBox und der ABox Ihres Netzes an.

Lösung:

Ein bekanntes semantisches Netz, was die Bedingungen erfüllt ist das folgende:



Hier sind die entsprechenden TBox/ABox-Einträge (Beispiele):

TBox	ABox
animal/can/move	Rex/inst/tiger
tiger/pattern/striped	Rex/eat/Clyde
amoeba/isa/animal	Roy/eat/Clyde

Aufgabe 3.2 (SPARQL-Anfragen)

9 Pkt

Geben Sie jeweils eine vollständige SPARQL Query für DBpedia oder eine vergleichbare Quelle an, die...

9 Min

1. (2 Punkte) ... alle bekannten Musikstücke (Songs) abfragt,
2. (3 Punkte) ... alle offiziellen Sprachen von Orten abfragt, an denen auch Deutsch eine offizielle Sprache ist,
3. (4 Punkte) ... alle Städte abfragt, sortiert nach Höhe über dem Meeresspiegel (niedrigste zuerst), aber insgesamt nur 10 Ergebnisse zurück gibt.

Hinweis: Wir erwarten nicht, dass Sie alle relevanten Ontologien für Properties wie "offizielleSprache" oder dergleichen auswendig können oder mitgebracht haben.

Sie können also die fiktive Ontologie UCO (für "Universally Convenient Ontology") benutzen und sich relevante Object Properties dafür ausdenken. So könnte zum Beispiel die offizielle Sprache eines Ortes als die Property `uco:officialLanguage` modelliert sein.

Lösung: Hier ist jeweils eine mögliche Lösung (mit echten Ontologien statt UCO):

(Funktionieren mit <https://dbpedia.org/snorql/>)

```
1. SELECT DISTINCT ?music WHERE {
    ?music rdf:type dbo:Song .
}
```

- ```

2. SELECT DISTINCT ?language WHERE {
 ?place dbo:officialLanguage ?language .
 ?place dbo:officialLanguage dbr:German_language .
}

3. SELECT DISTINCT ?city WHERE {
 ?city rdf:type dbo:City .
 ?city dbo:elevation ?altitude .
}
ORDER BY ?altitude
LIMIT 10

```
- 

## 4 Geistiges Eigentum

### Aufgabe 4.1 (Einbindung von CC-lizenzierten Werken)

8 Pkt

Sie haben die beiden unten abgebildeten Kunstwerke gefunden, welche unter den angegebenen Creative Commons Lizenzen<sup>1</sup> herausgegeben wurden.

8 Min

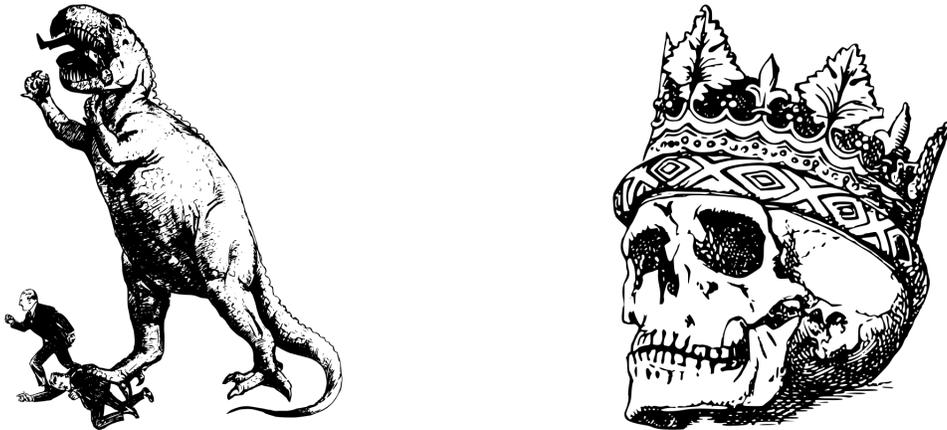


Abbildung 2: Dinosaurier hier unter CC-BY, Schädel hier unter CC-BY-NC-SA

Sie möchten die Kunstwerke gerne in Ihrem eigenen Projekten benutzen. Jetzt ist allerdings die Frage, welche Lizenz Sie für Ihr eigenes Projekt wählen können, wenn Sie dies tun.

Nennen Sie an, Ihr eigenes Projekt bindet jeweils eines der beiden Kunstwerke ein. Geben Sie dann zu jedem Kunstwerk eine Creative-Commons-Lizenz an, unter der Sie Ihr Projekt veröffentlichen dürften erklären Sie kurz (!) welche Bedingungen darüber hinaus erfüllt sein müssen.

*Lösung:*

- **Dinosaurier:**

Projekt darf zum Beispiel ebenfalls unter CC-BY oder sogar CC0 veröffentlicht werden. Solange die Urheber\*In des eingebundenen Bildes benannt wird, gibt es keine weiteren Einschränkungen.

<sup>1</sup>Wir nehmen die gegebenen Lizenzen nur für diese Aufgabe an, die wahren Lizenzen dieser Bilder sind anders.

- **Schädel mit Krone:**

Darf ausschließlich unter CC-BY-NC-SA veröffentlicht werden, also der gleichen Lizenz wie das Bild selbst. Dies ist durch die ShareAlike (SA) Klausel nicht zu umgehen. Das Projekt darf folglich auch nicht kommerziell verarbeitet werden. Und auch hier muss die Urheber\*In des eingebundenen Bildes benannt werden.

---