

Name:

Vorname:

Geburtsdatum:

Matrikelnummer:

Platz:

Nachklausur
Informatische Werkzeuge in den Geistes-
und Sozialwissenschaften 2

13. Oktober 2022

	To be used for grading, do not write here										
prob.	1.1	2.1	2.2	2.3	2.4	3.1	3.2	3.3	4.1	Sum	grade
total	15	4	6	8	6	7	4	5	5	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 Spieledatenbank)

15 Pkt

Gegeben ist die folgende SQL-Tabellen `Mitarbeitende`. Die Spalte `PersonID` ist dabei ein *Primärschlüssel* (“Primary Key”). 15 min

PersonID	Name	Abteilung	Gehalt
0	Gertrud Gründer	Management	80000
1	Franziska Finanz	Management	90000
2	Detlef Designer	Produktentwicklung	40000
3	Sabrina Server	IT	35000
4	Beatrice Beispiel	Praktikum	0

- (3 Punkte) Geben Sie einen SQL-Befehl an, der die Namen aller festen Mitarbeitenden (`Abteilung` darf also nicht “Praktikum” sein) abfragt.
- (3 Punkte) Geben Sie einen SQL-Befehl an, mit dem die Gehälter aller der Mitarbeitenden auf 0 gesetzt werden, die aktuell ein Gehalt von mehr als 60000 haben.
- (3 Punkte) Geben Sie einen SQL-Befehl an, mit dem *alle* Einträge in der Tabelle gelöscht werden. Die Tabelle selbst soll bestehen bleiben.
- (3 Punkte) Geben Sie einen SQL-Befehl an, mit dem eine neue Tabelle `Produkte` angelegt wird, falls diese noch nicht existiert. Diese Tabelle soll die Spalten `ProductID` (Integer, Primärschlüssel), `Bezeichnung` (Text), `Preis` (Integer) und `Verantwortung` (Integer) haben.
Dabei soll die Spalte `Verantwortung` ein *Fremdschlüssel* (“Foreign Key”) sein, der auf `PersonID` in `Mitarbeitende` verweist.
- (3 Punkte) Gehen Sie nun davon aus, dass die Tabelle `Produkte` einige Einträge erhalten hat und geben Sie einen SQL-Befehl an, der alle Bezeichnungen von Produkten und den Namen der jeweils verantwortlichen Person abfragt. Führen Sie dafür einen „Inner Join“ von `Produkte` und `Mitarbeitende` auf der Spalte `PersonID/Verantwortung` durch.

Solution:

1. Der Befehl lautet wie folgt:

```
SELECT Name FROM Mitarbeitende WHERE Abteilung != 'Praktikum';
```

2. Der Befehl lautet wie folgt:

```
UPDATE Mitarbeitende SET Gehalt=0 WHERE Gehalt > 60000;
```

3. Der Befehl lautet wie folgt:

```
DELETE FROM Mitarbeitende;
```

4. Der Befehl lautet wie folgt:

```
CREATE TABLE IF NOT EXISTS Produkte (  
    ProductID INTEGER PRIMARY KEY,  
    Bezeichnung TEXT,  
    Preis INTEGER,  
    Verantwortung INTEGER,  
    FOREIGN KEY (Verantwortung) REFERENCES Mitarbeitende(PersonID)  
);
```

5. Der Befehl lautet wie folgt:

```
SELECT Produkte.Bezeichnung, Mitarbeitende.Name  
FROM Produkte  
INNER JOIN Mitarbeitende  
ON Produkte.Verantwortung = Mitarbeitende.PersonID;
```

2 Bild

Aufgabe 2.1

4 Pkt

Wenn wir ein Farbbild (RGB) in Graustufen überführen, verwenden wir in der Regel eine gewichtete Summe der einzelnen Farbkanäle wie z.B.:

4 min

$$G = 0.21 \cdot R + 0.71 \cdot G + 0.08 \cdot B$$

Beschreiben Sie kurz, warum die einzelnen Kanäle hier unterschiedlich stark gewichtet werden, und warum gerade der Grüne Kanal so stark dominiert.

Solution: Menschliche Augen sind besonders sensitiv gegenüber grünem Licht, dieses hat also den größten Einfluss auf unsere Farbwahrnehmung. Um ein Bild so in Graustufen umzuwandeln, wie wir es wahrnehmen würden, gewichten wir deshalb den grünen Farbkanal stärker.

Aufgabe 2.2 (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 Bildes können Sie mit `img.width` bzw. `img.height` abfragen.

Solution: 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 2.3 (Helligkeit der Nachbarschaft eines Pixels)

8 Pkt

Schreiben Sie eine python-Funktion `neighbourhood_brightness()`, die ein Bild `img`, sowie zwei Integer (für X- und Y-Koordinate) als Parameter entgegen nimmt und die Helligkeit der Nachbarschaft um die gegebene Koordinate zurück gibt.

8 min

Wir verwenden hier der Einfachheit halber eine sehr naive Interpretation von „Helligkeit“. Um die Helligkeit eines Pixels zu ermitteln, addieren Sie bitte einfach die Werte der RGB-Kanäle des Pixels auf. Die Helligkeit einer Nachbarschaft entspricht der Summe der Helligkeit aller Pixel.

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 Helligkeit von 0.

Hinweis: Sie können für die Lösung dieser Aufgabe davon ausgehen, dass Ihnen die Funktion `getPixelSafe()` aus Aufgabe 2.2 zur Verfügung steht (auch wenn Sie diese nicht erfolgreich bearbeitet haben).

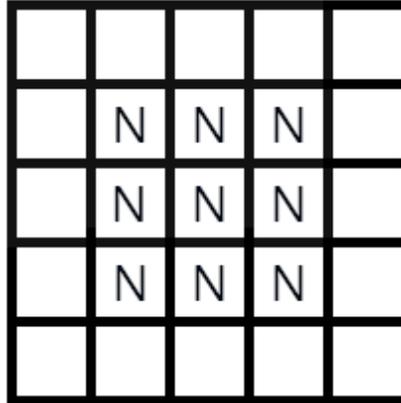


Abb. 1: Raster mit Koordinaten (0,0) bis (4,4). Für die Koordinate (2,2) ist die Nachbarschaft genau die Felder, die mit N markiert sind.

Solution: Hier ist eine mögliche Implementation:

```
def neighbourhood_brightness(img, x, y):  
    """Returns the sum total of "brightness" in the  
       neighbourhood of the given coordinate in the given image"""  
    neighbourhood = 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)  
            neighbourhood += sum([r,g,b])  
    return neighbourhood
```

Aufgabe 2.4 (Finden der hellsten Nachbarschaft)

6 Pkt

Schreiben Sie eine python-Funktion `brightest_area()`, die ein (Pillow-)Bild `img` als Parameter entgegen nimmt. Der Rückgabewert soll die Koordinate des Pixels mit der hellsten Nachbarschaft des Bildes sein, in der Form eines Tupels `(x,y)`.

6 min

Sollte der hellste Wert von mehr als einer Koordinate erreicht werden, ist egal, welche davon Sie zurück geben.

Hinweis: Sie können für die Lösung dieser Aufgabe davon ausgehen, dass Ihnen die Funktion `neighbourhood_brightness()` aus Aufgabe 2.3 zur Verfügung steht (auch wenn Sie diese nicht erfolgreich bearbeitet haben).

Erinnerung: Die Breite bzw. Höhe eines Bildes können Sie mit `img.width` bzw. `img.height` abfragen.

Solution: Hier ist eine mögliche Implementation:

```
def brightest_area(img):
    """Returns X and Y coordinates of the brightest
       region in the given image."""

    # Default values to start with.
    threshold = 0
    champion = (0,0)

    for x in range(0, img.width) :
        for y in range(0, img.height) :

            bright = neighbourhood_brightness(img, x, y)

            if bright > threshold:
                champion = (x,y)
                threshold = bright

    return champion
```

3 Semantic Web & Kulturelles Erbe

Aufgabe 3.1 (Semantische Netze)

7 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 Instanzierungen (*inst*-Kanten).

7 min

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

Hinweis: Wir erwarten nicht, dass Sie alle relevanten Ontologien auswendig können oder mitgebracht haben.

Sie können also die fiktive Ontologie UCO (für *“Universally Convenient Ontology”*) benutzen und relevante Object Properties etc. dafür erfinden. So könnte zum Beispiel Berlin als `uco:Berlin` modelliert sein.

Solution: Hier ist eine mögliche Lösung (mit echten Ontologien statt UCO):
(Funktioniert mit <https://dbpedia.org/snorql/>)

```
PREFIX : <http://dbpedia.org/resource/>
SELECT ?predicate WHERE {
  :Kreuzberg ?predicate :Berlin .
}
```

4 Geistiges Eigentum

Aufgabe 4.1 (Kombination von CC-lizenzierten Werken)

5 Pkt

Beatrice hat einen bösen Brief von einem Anwalt bekommen. Sie hat ein Buch geschrieben und es unter der Lizenz „Creative Commons - Attribution“ (CC-BY) veröffentlicht. In diesem Buch verwendet sie ein Bild eines Faultiers, welches unter „Creative Commons - Attribution NonCommercial ShareAlike“ (CC-BY-NC-SA) veröffentlicht wurde. Der Brief vom Anwalt sagt nun, dass sie dies nicht hätte tun dürfen. Beatrice ist verwirrt und sagt, sie hätte vorher extra in folgender Tabelle nachgesehen, dass die Lizenzen kompatibel wären.

5 min

Erklären Sie, wie es sein kann, dass Beatrice rechtlichen Ärger für ihre Verwendung des Faultierbildes bekommt, obwohl die Tabelle die beiden relevanten Lizenzen als kompatibel listet.

Solution: Die Lizenzen sind tatsächlich kompatibel, die Tabelle hat Recht. Das Problem entstand erst durch die Lizenz, die Beatrice für das Resultat gewählt hat. Ein Haken in der Tabelle bedeutet, dass die Werke grundsätzlich kombiniert werden können, es ist aber nicht egal, unter welcher Lizenz. Es muss immer mindestens die restriktivere der Lizenzen gewählt werden (in der Tabelle weiter rechts/unten). Es ist nicht möglich, um die NC/SA-Klauseln der Lizenz des Faultierbildes herum zu kommen, indem eine weniger restriktive Lizenz wie CC-BY gewählt wird.

	 PUBLIC DOMAIN	 PUBLIC DOMAIN	 CC BY	 CC BY SA	 CC BY NC	 CC BY ND	 CC BY NC SA	 CC BY NC ND
 PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
 PUBLIC DOMAIN	✓	✓	✓	✓	✓	✗	✓	✗
 CC BY	✓	✓	✓	✓	✓	✗	✓	✗
 CC BY SA	✓	✓	✓	✓	✗	✗	✗	✗
 CC BY NC	✓	✓	✓	✗	✓	✗	✓	✗
 CC BY ND	✗	✗	✗	✗	✗	✗	✗	✗
 CC BY NC SA	✓	✓	✓	✗	✓	✗	✓	✗
 CC BY NC ND	✗	✗	✗	✗	✗	✗	✗	✗

Abb. 2: Kompatibilität von Creative Commons Lizenzen