, Yvette
,
Krügerke, Maximilian

Task 1: Entwickeln einer Projekt Idee

1. A
2. **Project idea:**
   After finding the source of the given dataset, we know that it contains data of a literature review on a specific topic. If someone were to conduct research on this topic, this dataset would be useful for finding past studies. To make their citations more reliable, we found a second dataset containing impact factor data on various scientific journals. When compared with the literature review dataset, it enables users to identify which articles are published in authoritative journals. We assume that there is a positive correlation between a journal's impact factor and its reliability.

   In our database, each of the two datasets is represented by a table.

   Datasets used

- Given dataset: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2478
- Second dataset from https://www.kaggle.com/datasets/mostafafaramin/scientific-journals-ranking-sjr. This dataset ranks scientific journals based on their impact factor (IF) in 2020.

Below we visualize the two datasets as tables:

Dataset 1:

In [1]:
```
import pandas as pd
```

In [2]:
```
# given dataset
lit_df = pd.read_csv('Literature-data_TU-Darmstadt.txt', sep = '\t', header = 1, skiprows = [3])
lit_df.head(5)
```

Out[2]:

	Quelle	Autor	Titel	Jahr	Journal	Typ	DOI	Gelesen?	Empirisch?	Ausschlusspunkt	...	Cognitive / Memory tasks	Spatial Perception	Quanti
0	WOS	Chaudhary, Ayesha Hoor; Bukhari, Faisal; Iqbal...	Laparoscopic Training Exercises Using HTC VIVE	2020	INTELLIGENT AUTOMATION AND SOFT COMPUTING	NaN	10.31209/2019.100000149	Görge	NaN	Titel	...	NaN	NaN	
1	WOS	Kim, Soo-Kyun; Lee, Chang-Hee; Kim, Sun-Jeong;...	Implementation of Local Area VR Environment us...	2020	INTELLIGENT AUTOMATION AND SOFT COMPUTING	NaN	10.31209/2019.100000131	Görge	NaN	Titel	...	NaN	NaN	
2	WOS	Lee, Byong Kwon; Lee, Yang Sun	Distinction Between Real Faces and Photos by A...	2020	INTELLIGENT AUTOMATION AND SOFT COMPUTING	NaN	10.31209/2019.100000134	Görge	NaN	Titel	...	NaN	NaN	
3	WOS	Frederiksen, Joakim Grant; Sorensen, Stine May...	Cognitive load and performance in immersive vi...	2020	SURGICAL ENDOSCOPY AND OTHER INTERVENTIONAL TE...	NaN	10.1007/s00464-019-06887-8	Görge	Ja	Sample	...	NaN	x	
4	WOS	Rafique, Muhammad Usman; Cheung, Sen-ching S.	Tracking Attacks on Virtual Reality Systems	2020	IEEE CONSUMER ELECTRONICS MAGAZINE	NaN	10.1109/MCE.2019.2953741	Görge	Nein	Abstract	...	NaN	NaN	

5 rows × 41 columns

Dataset 2:

In [3]:
```
# dataset of journal impact factors
jif_df = pd.read_csv('journal_impact_factors_2020.csv')
jif_df.head(5)
```

Out[3]:

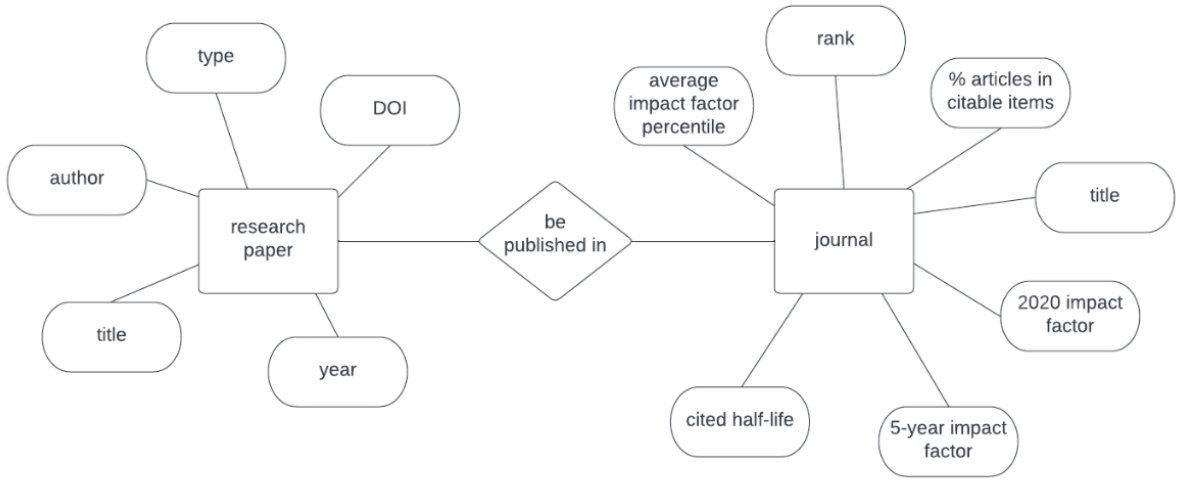
	Rank	Full Journal Title	Journal Impact Factor	Impact Factor without Journal Self Cites	5-Year Impact Factor	Cited Half-Life	Citing Half-life	% Articles in Citable Items	Average Journal Impact Factor Percentile
0	1	CA-A CANCER JOURNAL FOR CLINICIANS	292.278	291.481	225.87	3.4	4.6	77.27	99.795
1	2	NEW ENGLAND JOURNAL OF MEDICINE	74.699	73.983	72.098	8.7	4.9	84.45	99.697
2	3	Nature Reviews Materials	71.189	70.968	84.972	2.8	5.5	2.27	99.678
3	4	NATURE REVIEWS DRUG DISCOVERY	64.797	63.905	60.796	8.2	5.5	11.11	99.747
4	5	LANCET	60.392	59.208	59.345	8.6	4.2	69.82	99.091

Task 2: Data schema and databse set up

1.

Data schema

ERM



2. **Relational model:**

   literature (Titel, Autor, Jahr, Journal, Typ, DOI)
   journals (Title, Rank, Impact_Factor, IF_5_Yr, Half_Life, Percentage_Citable, Avg_IF_Percentile)

3. **Database set up**

   Steps

- Install PostgreSQL and create a database locally
- Install and import psycopg2
- Open connection to database
- Read in data files and create two tables ("literature" and "journals")
- Close connection when done

   Challenges

- Since our files are CSVs, the copy_from function in the given example failed to correctly parse values - it didn't ignore commas in quotes, which resulted in more columns than there

actually were. We tried using TSV files instead, but reading in individual lines became more difficult because the readline function didn't work in this case. In the end we decided on the copy_expert function:

- f1 = open('literature_dataset.csv')
- cur.copy_expert("""COPY literature FROM STDIN WITH CSV HEADER DELIMITER AS ','""", f1)
- After failing to create tables on the first try, we encountered another error on the second try: "DatabaseError: current transaction is aborted, commands ignored until end of transaction block". We had to roll back the transaction before trying again

Task 3: Pre-processing and import data

1. Steps

- Keep the attributes we need according to our relational model.
- Convert some string columns to lowercase for consistency. For example, some journal titles were all capitalized while others were only capitalized on the first letter ("BRAIN SCIENCES" vs "Brain Sciences").
- Deal with null values
- Literature dataset: identify and remove duplicates (marked with Ausschlusspunkt = "Dopplung")

The processed datasets we imported to the database are shown below:

2.

The processed datasets we imported to the database are shown below:

In [4]:
```
literature_df = pd.read_csv('literature_dataset.csv')
literature_df.head(5)
```

Out[4]:

	Titel	Autor	Jahr	Journal	Typ	DOI
0	laparoscopic training exercises using htc vive	chaudhary, ayesha hoor; bukhari, faisal; iqbal...	2020	intelligent automation and soft computing	NaN	10.31209/2019.100000149
1	implementation of local area vr environment us...	kim, soo-kyun; lee, chang-hee; kim, sun-jeong;...	2020	intelligent automation and soft computing	NaN	10.31209/2019.100000131
2	distinction between real faces and photos by a...	lee, byong kwon; lee, yang sun	2020	intelligent automation and soft computing	NaN	10.31209/2019.100000134
3	cognitive load and performance in immersive vi...	frederiksen, joakim grant; sorensen, stine may...	2020	surgical endoscopy and other interventional te...	NaN	10.1007/s00464-019-06887-8
4	tracking attacks on virtual reality systems	rafique, muhammad usman; cheung, sen-ching s.	2020	ieee consumer electronics magazine	NaN	10.1109/MCE.2019.2953741

In [5]:
```
journals_df = pd.read_csv('journal_dataset.csv')
journals_df.head(5)
```

Out[5]:

	Title	Rank	Impact_Factor	IF_5_Yr	Half_Life	Percentage_Citable	Avg_IF_Percentile
0	ca-a cancer journal for clinicians	1	292.278	225.870	3.4	77.27	99.795
1	new england journal of medicine	2	74.699	72.098	8.7	84.45	99.697
2	nature reviews materials	3	71.189	84.972	2.8	2.27	99.678
3	nature reviews drug discovery	4	64.797	60.796	8.2	11.11	99.747
4	lancet	5	60.392	59.345	8.6	69.82	99.091

Challenges:

- Initially there were some difficulties with importing the given dataset because the first few lines caused the columns to be parsed incorrectly.
- Even though duplicate rows in the literature dataset were marked, there were still differences between them in a few cases, so we had to delete the one not marked

with "Dopplung". In the image below, the row marked with "Dopplung" contains a DOI value, while the other does not.

	Autor	Titel	Jahr	Journal	Typ	DOI	Ausschlusspunkt
782	riva, g; wiederhold, bk	the new dawn of virtual reality in health care...	2015	annual review of cybertherapy and telemedicine	Article	NaN	Abstract
800	riva, g; wiederhold, bk	the new dawn of virtual reality in health care...	2015	annual review of cybertherapy and telemedicine...	Article; Book Chapter	10.3233/978-1-61499-595-1-3	Dopplung

-
- We couldn't find duplicates for two rows marked with "Dopplung", so we kept them in the dataset.
- Deciding what to do with null values
  - Since we are using research paper titles for the index in the literature dataset, we decided to replace null values in the DOI column with zeroes.
  - Null values in the impact factor dataset were strings ("Not Available"). We changed them to NaN instead

Task 4: Develop a Web application

1. a
   Decisions

- For easier and more systematic creation of tables, we defined all columns as type varchar according to the given project example. This made conditions involving integers (e.g. sorting by year) more complicated - we had to explicitly convert those columns to type INT.
- For some sorting and filtering functions, multiple options are provided for the user to toggle between, for example ascending/descending sorting order:
- order = 'ASC'
- # order = 'DESC'
- We make it easy to switch between queries by defining them beforehand and changing the value of a designated variable used in the execute function. For example:
- query1 = """SELECT * FROM literature"""
- query2 = """SELECT * FROM journals"""
- sql_query = query1 # change the value of this variable
- cur.execute(sql_query)

2.

3.

Sorting functions
Sort by journal ranking

- Useful when initially searching for papers to read - find research articles published by the most impactful journals.
- Ascending/descending order can be changed.

SQL query used:

```
SELECT L.Titel, Autor, J.Title, CAST (Rank AS INT), Impact_Factor, IF_5_Yr

FROM literature AS L, journals AS J
WHERE L.Journal = J.Title
ORDER BY Rank ASC;
```

Next we use the read_sql_query function to execute the query and retrieve the results in a pandas dataframe:
df = pd.read_sql_query(sql_query, conn)
The dataframe is displayed as follows:

Task 5: Present your solution

Task 6: Documentation

Weitere Teile der Dokumentation, sowie vollständiges Projekt:

https://github.com/thirdlongitude/dbs_projekt.git