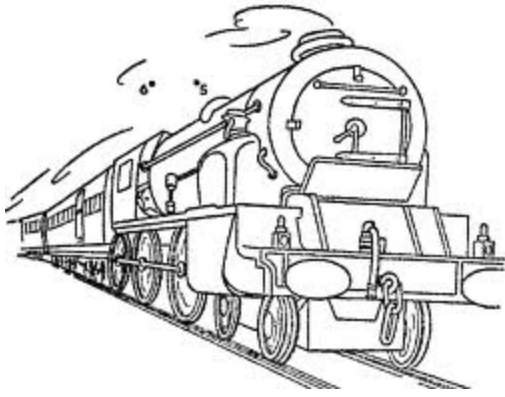


Übung Datenbanken: Ein Zugauskunftssystem



In [1]:

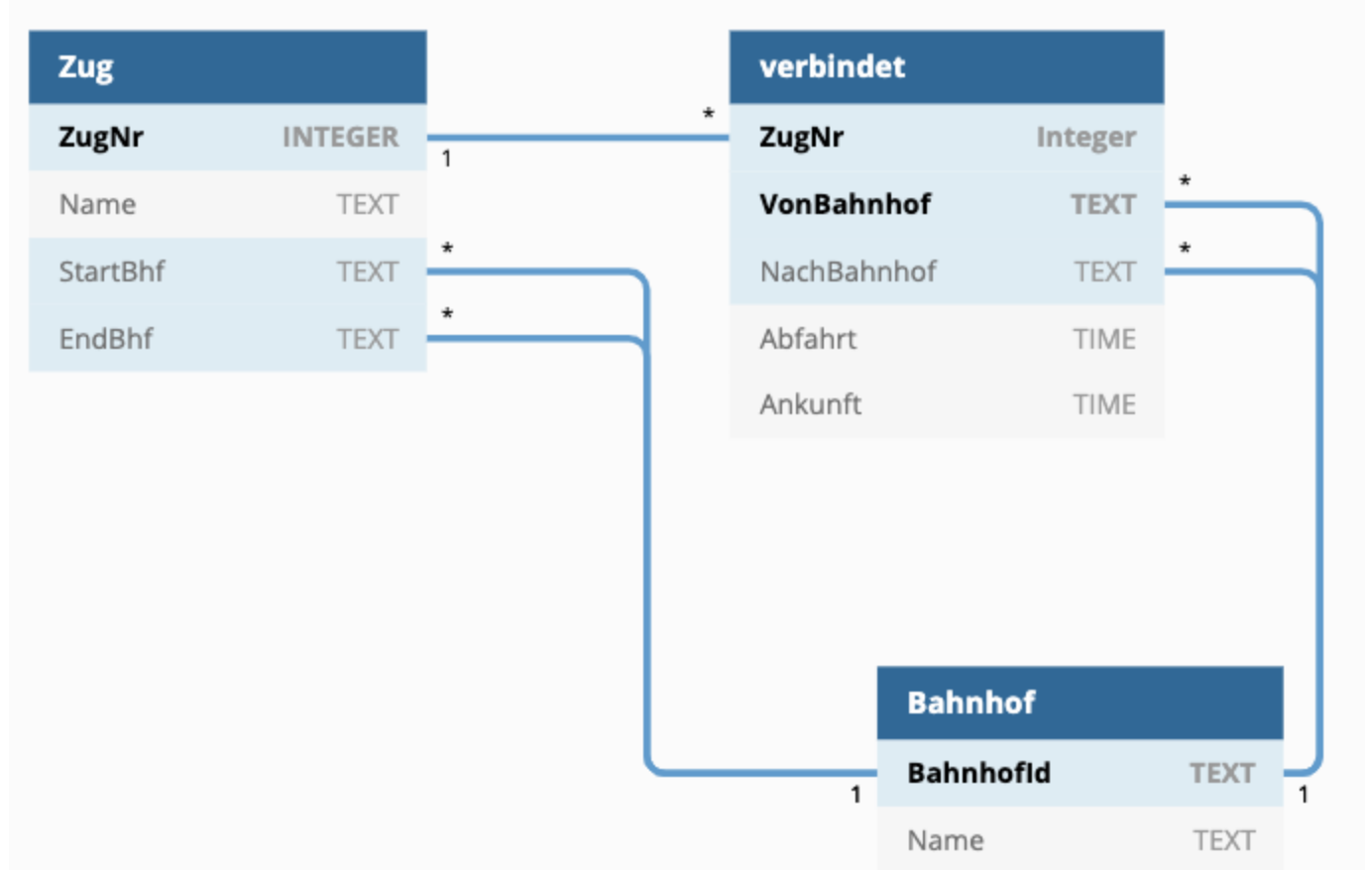
```
# Hier ist nur Code zum Initialisieren der Umgebung, bitte gehen Sie weiter, es gibt nicht  
# Keine langen Fehlermeldungen  
import sys  
ipython = get_ipython()  
  
def exception_handler(exception_type, exception, traceback):  
    print("%s: %s" % (exception_type.__name__, exception), file=sys.stderr)  
  
ipython._showtraceback = exception_handler  
  
!wget -qOzugauskunft-kemper.db 'https://nextcloud.th-deg.de/s/Kc7m486Z3KbZyB4/download?pat  
# Lade die Erweiterung, damit wir SQL Befehle nutzen können  
%reload_ext sql  
%sql sqlite:///zugauskunft-kemper.db  
%sql PRAGMA foreign_keys = ON
```

```
* sqlite:///zugauskunft-kemper.db  
Done.
```

Out[1]: []

Aufgabe

Nachdem wir in der letzten Aufgabe unsere Daten in folgendes Datenmodell importiert haben:



Wollen wir in dieser Aufgabe einige Abfragen auf diesem Modell entwerfen.

In [2]: `!echo ".schema" | sqlite3 zugauskunft-kemper.db`

```

CREATE TABLE Zug( ZugNr INTEGER NOT NULL,
    Name TEXT NOT NULL,
    StartBhf TEXT NOT NULL,
    EndBhf TEXT NOT NULL,
    FOREIGN KEY(StartBhf) REFERENCES Bahnhof(BahnhofId),
    FOREIGN KEY(EndBhf) REFERENCES Bahnhof(BahnhofId),
    PRIMARY KEY(ZugNr)
);
CREATE TABLE Bahnhof(
    BahnhofId TEXT NOT NULL,
    Name TEXT NOT NULL,
    PRIMARY KEY(BahnhofId)
);
CREATE TABLE verbindet(
    ZugNr Integer NOT NULL,
    VonBahnhof TEXT NOT NULL,
    NachBahnhof TEXT,
    Abfahrt TIME NOT NULL,
    Ankunft TIME NOT NULL,
    FOREIGN KEY(VonBahnhof) REFERENCES Bahnhof(BahnhofId),
    FOREIGN KEY(NachBahnhof) REFERENCES Bahnhof(BahnhofId),
    FOREIGN KEY(ZugNr) REFERENCES Zug(ZugNr),
    PRIMARY KEY(VonBahnhof, ZugNr)
);
  
```

Abfrage 1: Abfahrten in Deggendorf

Entwerfen Sie eine Abfrage, die alle Abfahrten in Deggendorf anzeigt. Geben Sie die Zugnummer, den Namen, den Start- und Endbahnhof sowie die Abfahrtszeit aus.

In [3]:

```
%%sql
SELECT Zug.ZugNr, Zug.Name, start.Name as Startbahnhof, end.Name as Endbahnhof
FROM Zug
JOIN Bahnhof AS start
ON(Zug.StartBhf = start.BahnhofId)
JOIN Bahnhof AS end
ON(Zug.EndBhf = end.BahnhofId)
WHERE Startbahnhof LIKE 'Deggendorf%' LIMIT 10;
```

```
* sqlite:///zugauskunft-kemper.db
Done.
```

Out[3]:

	ZugNr	Name	Startbahnhof	Endbahnhof
	1003122052	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122056	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122057	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122061	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122062	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122063	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122064	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122087	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122088	WBA1 Plattling	Deggendorf Hbf	Plattling
	1003122120	WBA1 Plattling	Deggendorf Hbf	Plattling

Nett, aber wir haben keine Abfahrtszeit!

In [4]:

```
%%sql SELECT ZugNr, Abfahrt as Abfahrt FROM verbindet LIMIT 10;
```

```
* sqlite:///zugauskunft-kemper.db
Done.
```

Out[4]:

	ZugNr	Abfahrt
	1003122052	5:30:00
	1003122052	5:35:00
	1003122052	5:39:00
	1003122056	24:38:00
	1003122056	24:44:00
	1003122056	24:49:00
	1003122057	23:29:00
	1003122057	23:34:00
	1003122057	23:39:00
	1003122058	22:04:00

In [7]:

```
%%sql SELECT ZugNr, MIN(Abfahrt) as Abfahrt FROM verbindet GROUP BY ZugNr LIMIT 10;
```

```
* sqlite:///zugauskunft-kemper.db
Done.
```

Out[7]:

	ZugNr	Abfahrt
	1003122052	5:30:00

1003122056	24:38:00
1003122057	23:29:00
1003122058	22:04:00
1003122059	21:41:00
1003122061	18:16:00
1003122062	17:16:00
1003122063	13:16:00
1003122064	12:16:00
1003122065	19:59:00

In [8]:

```

%%sql
SELECT Zug.ZugNr, Zug.Name, verbindet.Abfahrt, start.Name as Startbahnhof, end.Name as Endbahnhof
FROM Zug
JOIN Bahnhof AS start
ON(Zug.StartBhf = start.BahnhofId)
JOIN Bahnhof AS end
ON(Zug.EndBhf = end.BahnhofId)
JOIN (SELECT ZugNr,MIN(Abfahrt) as Abfahrt FROM verbindet GROUP BY ZugNr)
AS verbindet
USING(ZugNr)
WHERE Startbahnhof LIKE 'Deggendorf%'
ORDER BY Abfahrt ASC;

* sqlite:///zugauskunft-kemper.db
Done.

```

Out[8]:

ZugNr	Name	Abfahrt	Startbahnhof	Endbahnhof
1015320361	WBA1 Plattling	10:45:00	Deggendorf Hbf	Plattling
1015320360	WBA1 Plattling	11:45:00	Deggendorf Hbf	Plattling
1003122064	WBA1 Plattling	12:16:00	Deggendorf Hbf	Plattling
1015320364	WBA1 Plattling	12:45:00	Deggendorf Hbf	Plattling
1003122063	WBA1 Plattling	13:16:00	Deggendorf Hbf	Plattling
1015320359	WBA1 Plattling	13:45:00	Deggendorf Hbf	Plattling
1015320371	WBA1 Plattling	14:45:00	Deggendorf Hbf	Plattling
1015320357	WBA1 Plattling	15:45:00	Deggendorf Hbf	Plattling
1003122087	WBA1 Plattling	16:16:00	Deggendorf Hbf	Plattling
1015320356	WBA1 Plattling	16:45:00	Deggendorf Hbf	Plattling
1003122062	WBA1 Plattling	17:16:00	Deggendorf Hbf	Plattling
1015320363	WBA1 Plattling	17:45:00	Deggendorf Hbf	Plattling
1003122061	WBA1 Plattling	18:16:00	Deggendorf Hbf	Plattling
1015320362	WBA1 Plattling	18:45:00	Deggendorf Hbf	Plattling
1015320355	WBA1 Plattling	19:45:00	Deggendorf Hbf	Plattling
1015320354	WBA1 Plattling	20:45:00	Deggendorf Hbf	Plattling
1015320353	WBA1 Plattling	21:45:00	Deggendorf Hbf	Plattling
1015320347	WBA1 Plattling	22:51:00	Deggendorf Hbf	Plattling
1003122057	WBA1 Plattling	23:29:00	Deggendorf Hbf	Plattling

1003122056	WBA1 Plattling	24:38:00	Deggendorf Hbf	Plattling
1003122148	WBA1 Zwiesel (Bay)	25:27:00	Deggendorf Hbf	Zwiesel (Bay)
1015320340	WBA1 Plattling	5:00:00	Deggendorf Hbf	Plattling
1003122052	WBA1 Plattling	5:30:00	Deggendorf Hbf	Plattling
1015320344	WBA1 Plattling	6:14:00	Deggendorf Hbf	Plattling
1003122120	WBA1 Plattling	6:45:00	Deggendorf Hbf	Plattling
1015320332	WBA1 Plattling	7:09:00	Deggendorf Hbf	Plattling
1003122088	WBA1 Plattling	7:38:00	Deggendorf Hbf	Plattling
1015320383	WBA1 Plattling	8:15:00	Deggendorf Hbf	Plattling
1015320334	WBA1 Plattling	8:45:00	Deggendorf Hbf	Plattling
1015320348	WBA1 Plattling	9:45:00	Deggendorf Hbf	Plattling

Sie finden in den Ergebnissen sicher eine Zeilen mit : Abfahrt 25:27. Das ist typisch für Bahn Systeme, kann aber manche Zeitfunktion Ihrer Datenbank aus dem Tritt bringen.

Aufgabe 2: Sortierng

Ihre Daten sind warscheinlich falsch sortiert, weil in den Daten '5:00:00' angegeben ist. Der `TIME` Datentyp in SQLite versteht aber nur '05:00:00'. Das können Sie dann über einen LIKE Match und einen UPDATE fixen. Entwerfen Sie die entsprechende Query

In [30]: `%sql UPDATE verbindet SET Abfahrt = "0" || Abfahrt WHERE Abfahrt LIKE "_: %";`

```
* sqlite:///zugauskunft-kemper.db
0 rows affected.
```

Out[30]: `[]`

In [31]: `%sql UPDATE verbindet SET Ankunft = "0" || Ankunft WHERE Ankunft LIKE "_: %";`

```
* sqlite:///zugauskunft-kemper.db
0 rows affected.
```

Out[31]: `[]`

Abfrage 3: Laufweg einer zufälligen Fahrt

Geben Sie alle Halte einer zufälligen (Hint: Subselect mit `RANDOM()`) Fahrt aus.

In [25]: `%%sql
SELECT Zug.ZugNr, Zug.Name, verbindet.Ankunft, verbindet.Abfahrt, start.Name as VonBahnhof
FROM Zug, verbindet, Bahnhof AS start, Bahnhof AS end
WHERE
verbindet.ZugNr = Zug.ZugNr AND
verbindet.VonBahnhof = start.BahnhofId AND
verbindet.NachBahnhof = end.BahnhofId AND
Zug.ZugNr = (SELECT ZugNr from Zug ORDER BY RANDOM() LIMIT 1);`

```
* sqlite:///zugauskunft-kemper.db
Done.
```

Out[25]:

	ZugNr	Name	Ankunft	Abfahrt	VonBahnhof	NachBahnhof
--	-------	------	---------	---------	------------	-------------

1003122243	WBA2 Zwiesel (Bay)	20:29:00	20:29:00	Bodenmais	Böhmhof
------------	--------------------	----------	----------	-----------	---------

1003122243	WBA2 Zwiesel (Bay)	20:32:00	20:32:00	Böhmhof	Langdorf
1003122243	WBA2 Zwiesel (Bay)	20:40:00	20:40:00	Langdorf	Außenried
1003122243	WBA2 Zwiesel (Bay)	20:43:00	20:43:00	Außenried	Zwiesel (Bay)

Abfrage 3: Von Deggendorf nach Grafenau

Wie komme ich von Deggendorf nach Grafenau? Gibt es einen einfachen Weg das in einer SQL Abfrage abzubilden?

In [12]:

```
%%sql
SELECT ZugNr, start.Name as StartBhf, end.Name as EndBhf
FROM Zug, Bahnhof as start, Bahnhof as end
WHERE
    start.BahnhofId = Zug.StartBhf
AND
    end.BahnhofId = Zug.EndBhf
AND
    start.Name = "Deggendorf Hbf"
AND
    end.Name = "Grafenau";
```

```
* sqlite:///zugauskunft.db
Done.
```

Out[12]: ZugNr StartBhf EndBhf

Lösung: Das geht nicht so einfach mit einer relationalen Datenbank. Dafür gibt es verschiedene Erweiterungen, z.B. pgRouting für Postgres.