

#### 1 Wdh: Klassen und Objekte



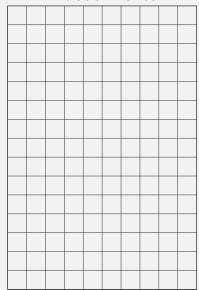
\_ repräsentieren **Gegenstände** in einem Computerprogramm.

 $\_$  sind der  $\sf Bauplan$ , der festlegt, welche  $\sf Eigenschaften$  (

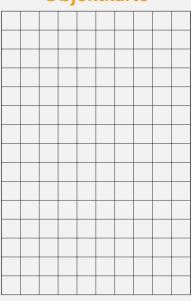
\_) und **Fähigkeiten** ( \_\_\_\_\_\_\_\_\_\_\_\_) einer bestimmten

Objektart gespeichert werden sollen. Man stellt sie dar mit:

#### Klassenkarte



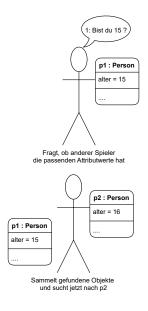
#### **Objektkarte**

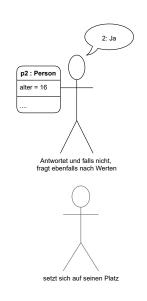


**1** Generell kann man Objektkarten mit oder ohne Methoden zeichnen, solange man es insgesamt einheitlich macht. Wir zeichnen sie daher immer <u>ohne Methoden</u>.

# Objektkarten Memory

- Erstelle auf einem Blatt eine Objektkarte der Klasse Person zu dir selbst. → 3x falten
- Gib deine Objektkarte bei der Lehrkraft ab. → Objektkarten werden gemischt.
- Ziehe eine Objektkarte und versuche, das zugehörige Objekt zu finden.
  - Frage deine:n Gegenüber dafür, ob die Attributwerte auf deiner gezogenen Karte auf sie/ihn zutreffen.
  - Ihr dürft euch nicht gegenseitig die Objektkarten zeigen!
  - Wer gefunden wurde, gibt seine aktuelle Objektkarte weiter und setzt sich.
  - Der/Die Finder:in sammelt alle gefundenen Objekte.

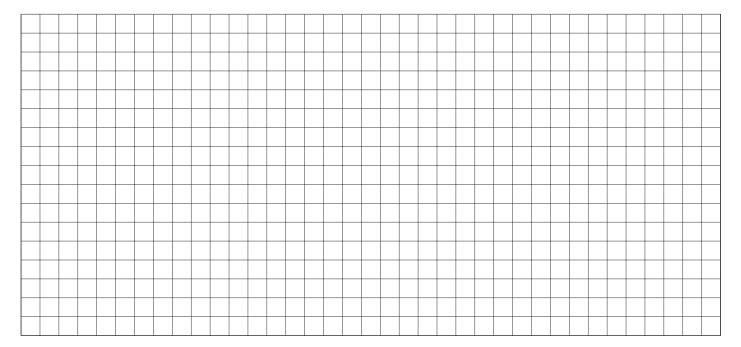






# Wdh: Von der Klasse zur Tabelle

- Zeichnet zu zweit eine Tabelle, in der man alle Objekte der Klasse Person sammeln kann.
- Tragt eure beiden Objekte (vom Objektkarten-Memory) in die Tabelle ein.
- Ordnet die folgenden Begriffe den Teilen der Tabelle zu. Achtung: Nicht alle Begriffe passen und manches hat mehrere Begriffe! Datensatz Tabelle Zelle Klasse Objekt Parameter Attribut Spalte Feld Methode Board Zeile Datentyp Attributwert



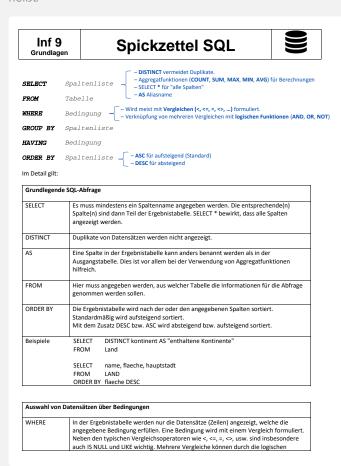
2 Wdh: Aufbau von (relationalen	) Datenbanken
Datenbanken speichern Datensätze in	Die
repr	räsentieren die <b>Attribute</b> (Synonym: Feld) und
bilden zusammen eine Klasse. Die	(=Zeilen) entsprechen
Objekten und in den Spalten stehen die A	Attributwerte. Jede Tabelle hat einen
(oft auch	"ID"), der Datensätze eindeutig identifiziert. Oft
werden die Datensätze hiermit einfach du	rchnummeriert. Im Tabellenschema wird er
unterstrichen und im Klassendiagramm in	nmer als erstes Attribut aufgelistet.
Der Aufbau einer Tabelle kann mit	oder
dargestellt we	rden. Dessen Aufbau ist:
${\bf TABELLENNAME}(\underline{\bf Datentyp\ Prim\"{a}rschl\"{u}ss}$	sel , Datentyp Spalte1, Datentyp Spalte2,)
Zum Beispiel:	

#### 3 SQL Spickzettel



Folgender SQL-Spickzettel enthält alle SQL-Grundlagen der 9. Klasse. Ihr dürft (sollt!) ihn bei allen SQL-Aufgaben benutzen. Über das Vorlagensymbol 🚨 oben könnt ihr den Spickzettel als eigenes PDF öffnen.

**1** Übrigens: **SQL** ist die Abkürzung für **S**tructured **Q**uery **L**anguage, was auf Deutsch etwa Strukturierte Abfrage Sprache heißt.



	Funktionen AND, OR und NOT verknüpft werden. Ggf. müssen die einzelnen Ausdrücke dabei sinnvoll geklammert werden
	Beispiel         WHERE jahr > 2015           AND laufzeit <= 90           AND NOT fsk = 18
LIKE	Kann in einer Bedingung zur Mustererkennung von Einträgen verwendet werden. Folgende zwei Platzhalter (wildcards) werden häufig eingesetzt:
	<ul> <li>% steht für beliebig viele Zeichen, auch keines (* bei MS Access)</li> <li>_ für genau ein beliebiges Zeichen (? bei MS Access)</li> </ul>
	Beispiele:
	WHERE titel LIKE "You%" – findet alle Titel die mit "You" beginnen Groß-/Kleinschreibung wird nicht berücksichtigt WHERE titel LIKE "%love%" – findet alle Titel die "love" enthalten WHERE titel LIKE "L" – findet alle Titel die mit L beginnen und genau 4 Zeichen lang sind
NULL	Bedeutet, dass kein Wert in einer Zelle eingetragen ist.
IS NULL	Überprüft (in einer Bedingung), ob kein Wert in einer Zelle eingetragen ist.

Aggregatfunktio	onen		
AVG	Berechnet	den Durchschnitt aller Werte einer Spalte.	
COUNT	Gibt die Ar	nzahl der Einträge einer Spalte aus.	
MAX bzw. MIN	Gibt das M	laximum bzw. Minimum aller Werte einer Spalte aus.	
SUM	Berechnet	die Summe aller Werte einer Spalte.	
Beispiel	SELECT FROM WHERE	COUNT(*) AS "Anzahl afrikanischer Länder " Land kontinent = "Afrika"	

Gruppierung	
GROUP BY	Datensätze mit demselben Wert in der angegeben Spalte werden gruppiert. Gruppierungen sind nur in Kombination mit Aggregatfunktionen sinnvoll.
HAVING	An gruppierte Datensätze werden Bedingungen mit HAVING formuliert.
Beispiel	SELECT fsk, MIN(laufzeit) FROM Film WHERE genre1="Filmkomödie" OR genre2="Filmkomödie" GROUP BY fsk HAVING fsk < 16

### SQL keywords should be in **lower case!**



select name, id
from products
where discount = 0
order by price asc;

## Noooo, they must be in **upper case!**



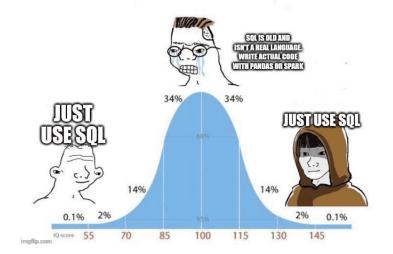
SELECT name, id FROM products WHERE discount = 0 ORDER BY price ASC;



sElEcT nAmE, iD fRoM PrOdUcTs WhErE dIsCoUnT = 0 OrDeR by pRiCe AsC;

'Sarcastic Query Language' • by u/casperdewith

**1** SQL Schlüsselwörter wie SELECT, WHERE etc. sind nicht case-sensitive. Groß-/Kleinschreibung ist also egal.







sql-island.informatik.uni-kl.de/

→ Pri	Not mär	nd tiert rsch Dat	da lüs	s vo sel)	olls	tän	dig	e T	abe	elle	nsc	her	ma	de	r Da	ateı	nba	ank	vor	n S(	QL	Isla	nd	(m	it D	ate	enty	pe	n u			
Ste	ellt o	die -	Гаb	elle	en c	der	Dat	ten	bar	nk n	nit l	<las    </las 	sse	nka	rter	n da	ar.															
	+	+																														
	+	-																														
	+																															
	+	+																														
	+																															
	+																														-	
	+																															
	+	_																														
	+	+																														
1																																

Für Schnelle: Spielt SQL Island, der SQL-Spickzettel hilft euch dabei.



In dieser Aufgabe geht es immer um die Tabelle land, deren erste Datensätze du hier siehst:

id	name	einwohner	flaeche	hauptstadt
1	Deutschland	83.24	358	Berlin
2	Frankreich	67.39	544	Paris
3	Brasilien	212.60	8516	Rio de Janeiro
	•••			

Welche SQL-Abfrage (rechte Seite) führt zu welcher Ergebnistabelle (linke Seite)? Ordne richtig zu!

<ol> <li>Zeige alle Spalten</li> </ol>
der Tabelle land.

- 2) Zeige die Spalten name und hauptstadt der Tabelle land.
- 3) Zeige die durchschnittliche Einwohnerzahl aller Länder.
  - **4)** Zeige die Namen aller Länder in alphabetisch absteigender Reihenfolge.
- 5) Zeige die Hauptstädte der Länder, deren Einwohnerzahl größer als 50 Mio ist.
- **6)** Zeige die Anzahl aller Länder, deren Name mit 'land' endet.
- 7) Zeige die Namen aller Länder, deren Fläche zwischen 100 und 999 Tausend km² liegt.
  - **8)** Zeige die Namen der Länder, die mit 'D' beginnen oder mit 'd' aufhören.
    - 9) Zeige die Namen der drei Länder mit der größten Einwohnerzahl.

i) SELECT name FROM land ORDER BY name DESC;

ii) SELECT name FROM land WHERE name LIKE 'D%' OR name LIKE '%d';

iii) SELECT COUNT(\*)
FROM land
WHERE name LIKE '%land';

iv) SELECT \*
FROM land;

v) SELECT name FROM land WHERE flaeche >= 100 AND flaeche <= 999;

vi) SELECT name, einwohner FROM land ORDER BY einwohner DESC LIMIT 3;

vii) SELECT AVG(einwohner) FROM land;

viii) SELECT name, hauptstadt FROM land;

ix) SELECT hauptstadt FROM land WHERE einwohner > 50;



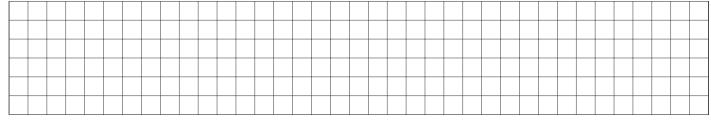
#### Wdh: SQL Basics



Bearbeite die Aufgabe Wdh - SQL Basics auf artemis.tum. de. Artemis gibt dir immer, wenn du auf Submit drückst, die ersten Zeilen der Ergebnistabelle und ob deine SQL-Abfrage (bzw. welche Teile von ihr) richtig sind, aus. Wenn du eine Abfrage richtig hast, notiere sie unten im Skript.

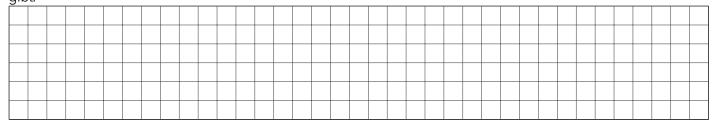
Falls du bei Gruppierung und Aggregatfunktionen Schwierigkeiten hast, hilft dir dieses Video (bitte Kopfhörer verwenden!): bycs.link/simpleclub-group-sort-aggregat

1) Vervollständige die SQL-Abfrage so, dass sie ID, Name, Art und URL aller Freibäder ausgibt.



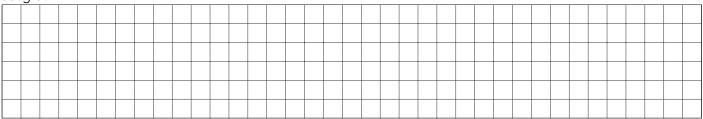
**2)** Schreibe eine SQL-Abfrage, die ausgibt, wie viele Gemeinden es im Regierungsbezirk Oberbayern

gibt.



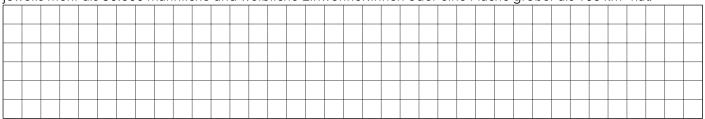
3) Schreibe eine SQL-Abfrage, die Name, Straße und URL (also die Internetadresse) alle Zoos in der Gemeinde mit Schluessel 09162000

ausgibt.

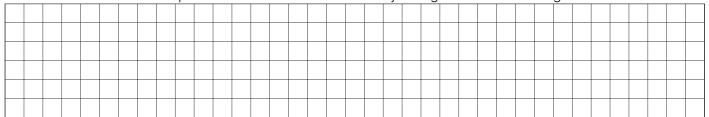


	_																																	
																		+		+														
1																		_	+															
mer Acht	n u tung	nd g: [	Reg Du	gie kan	run ınst	gsk be	ezi i de	i <b>rk</b> er \	des /erv	jev ven	wei dur	ilige ng v	en L on	an Gru	dkr upp	eis ieru	es a ung	nze nur	i <b>gt.</b> Spa	der ( Sort alten, u die	iere na	e di ch	e A der	usç nen	gab gru	e n	iach iert	n N wi	lam rd,	ne c	les	Lar	ndk	re
																																		L
																				en al	ler	Ger	mei	inde	e, d	lie	me	hr a	als	100	0.00	00 n	nän	ın
																	ohr, au			en al	ler	Ger	mei	inde	e, d	lie	me	hra	als	100	0.00	00 n	nän	ın
																				en al	ler	Ger	mei	inde	e, d	lie	me	hra	als	100	0.00	00 n	nän	ın
																				en al	ler	Ger	mei	inde	e, d	lie	me	hra	als	100	0.00	00 n	nän	in
																				en al	ler	Ger	mei	ind	e, d	lie	me	hra	als	100	0.00	00 n	nän	ın
																				en al	ler	Ger	mei	inde	e, d	lie	me	hr a	als	100	0.00	00 r	nän	in
																				en al	ler	Ger	mei	inde	e, d	lie	me	hra	als	100	).00	00 n	nän	in

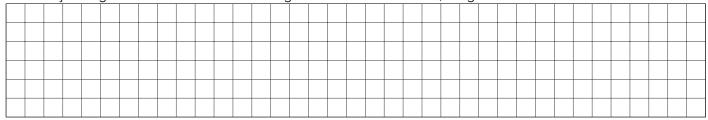
8) Schreibe eine SQL-Abfrage, die Name, Landkreis, Fläche und die Einwohnerzahlen aller Gemeinden ausgibt, die jeweils mehr als 50.000 männliche und weibliche Einwohner:innen oder eine Fläche größer als 100 km² hat.



9) Schreibe eine SQL-Abfrage, die die durchschnittlichen männlichen und weiblichen Einwohnerzahlen aller Gemeinde mit mehr als 100 km² Fläche pro Landkreis und den Namen des jeweiligen Landkreises ausgibt.



10) Schreibe eine SQL-Abfrage, die die Anzahl von Wanderwegen, die zu einer Gemeinde führen in einer Spalte Anzahl und den jeweiligen Gemeindeschlüssel absteigend nach Anzahl sortiert, ausgibt.





- 1. Visualisiere (mit Bleistift), wer Häuptling in welchem Dorf ist.
- 2. Überlege, wie du allgemein für diese zwei Tabellen darstellen kannst, wie sie (und ihre Spalten) miteinander in Beziehung stehen.



SELECT * FRO	M Bewohner					
bewohnernr	name	dorfnr	geschlecht	beruf	gold	status
1	Paul Backmann	1	m	Baecker	850	friedlich
2	Ernst Peng	3	m	Waffenschmied	280	friedlich
3	Rita Ochse	1	w	Baecker	350	friedlich
4	Carl Ochse	1	m	Kaufmann	250	friedlich
5	Dirty Dieter	3	m	Schmied	650	boese
6	Gerd Schlachter	2	m	Metzger	4850	boese
7	Peter Schlachter	3	m	Metzger	3250	boese
8	Arthur Schneiderpaule	2	m	Pilot	490	gefangen

#### Tabellenbeziehung im Klassendiagramm

- 1. Ergänze das Klassendiagramm entsprechend den beiden Tabellen oben.
- 2. Wie kann man die Beziehungen zwischen den beiden Tabellen im Klassendiagramm darstellen? Tipp: Unsere Überlegungen von oben helfen dabei.

Dorf	

Bewohner

### 4 Tabellenbeziehungen: Fremdschlüssel



### 5 Tabellenbeziehungen im Klassendiagramm



TabelleA
int id String spalte1
•••

TabelleB
int id String spalte1
•••

#### 6 Kardinalitäten



Die Kardinalität beschreibt, wie viele Objekte auf jeder Seite einer Beziehung stehen können.

Es gibt folgende Arten:

- 1:1, z.B. \_\_\_\_\_ Häuptling pro Dorf, der auch nur in einem Dorf Häuptling ist.
- 1:n, z.B. jeder Bewohner wohnt in einem Dorf, das aber \_\_\_\_\_\_ Bewohner

hat.

• m:n, z.B. \_\_\_\_\_\_ Lehrer pro Schulklasse + \_\_\_\_\_

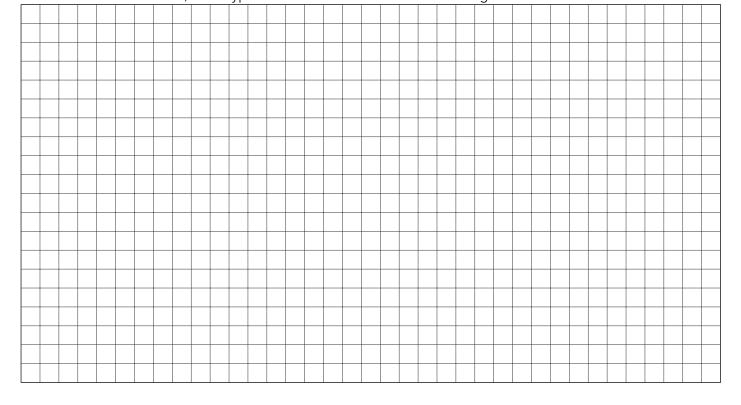
Schulklassen pro Lehrer (in Datenbanken nicht direkt umsetzbar, dazu später mehr).



# Klassendiagramm Flugverspätung



Bearbeite diese Aufgabe auf artemis.tum.de.
Erstelle ein Klassendiagramm für die Datenbank unter dbiu.de/flugverspaetungen/.
Damit du weniger schreiben musst, kannst du die letzten 6 Spalten der Tabelle Flug durch ... ersetzen.
Achte auf korrektes Format, Datentypen und Kardinalitäten. Zeichne das Diagramm anschließend unten auf:







Wir kennen jetzt Tabellen, die miteinander über Fremd- und Primärschlüssel in Beziehung stehen. Nun möchten wir aus diesen Tabellen auch zusammengehörende Datensätze abfragen.

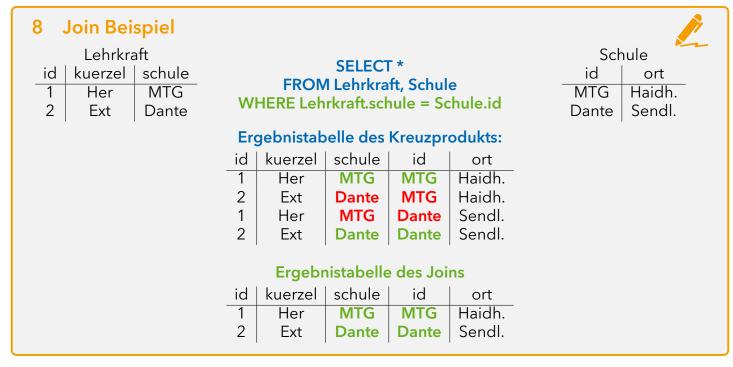
Öffne dafür www.dbiu.de/flugverspaetungen und führe folgende SQL-Abfrage aus:

# SELECT \* FROM Fluggesellschaft, Flug

Was beobachtest du? Werden nur zusammengehörende werden die beiden Tabellen miteinander kombiniert?	Datensätze	angezeigt?	Falls nicht,	nach	welchem	Muste

7 Kreuzprodukt / Join				
Möchte man Daten aus zwei Tabellen mit Beziehung zueinander abfragen, gibt man beide				
Tabellen mit Komma getrennt nach FROM an.				
Die SQL-Abfrage bildet dann das der Tabellen. Die				
Ergebnistabelle enthält von Datensätzen beider				
Tabellen (Merkregel:).				
Um nur zusammengehörige Datensätze (also solche, die miteinenader in Beziehung stehen,				
z.B. eine Bewohner mit seinem Dorf) auszuwählen, ergänzt man als Selektion eine				
Gleichheitsbedingung zwischen Fremd- und zugehörigem				
Dann spricht man von einem				
·				
Zum Beispiel kann man in SQL-Island die Daten aller Dörfer und ihrer zugehörigen				
Häuptlinge so ausgeben:				
SELECT *				
FROM Dorf, Bewohner				

WHERE Dorf.haeuptling = Bewohner.bewohnernr

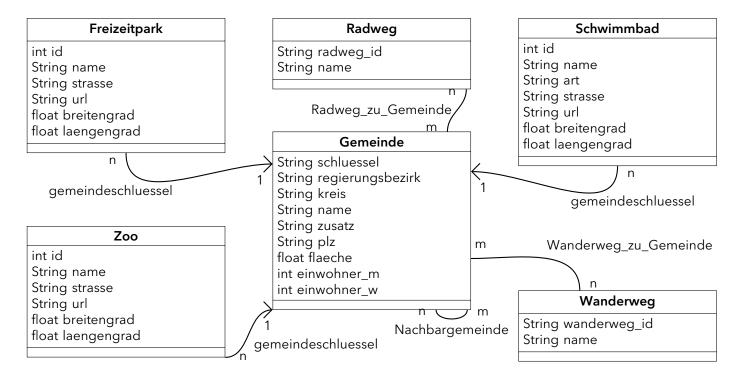




## **SQL** mit Kreuzprodukt und Join

Bearbeite diese Aufgabe auf artemis.tum.de. Du bekommst eine automatische Rückmeldung, ob deine Abgabe korrekt ist. Alle Aufgaben beziehen sich auf die Datenbank mit untem stehendem Klassendiagramm. Eine Online-Version gibt es unter www.dbiu.de/bayern/, dort ist auch das Tabellenschema zu finden.

Gib immer genau die geforderten Daten aus und nicht mehr. Sortiere nicht, wenn du nicht dazu aufgefordert wirst. Notiere unten anschließend deine korrekten SQL-Abfragen unten.



Verändere die SQL-Abfrage so, dass die Namen und Internetadressen (=url) aller Zoos und der Name und Regierungsbezirk der jeweiligen Gemeinde ausgegeben wird:

SELECT Zoo.name, Gemeinde.name

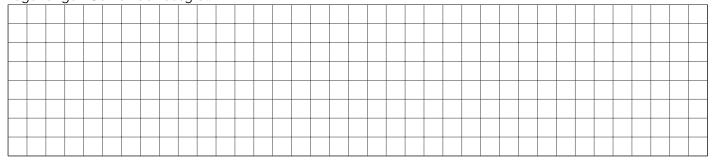
#### FROM Zoo, Gemeinde

Verändere die SQL-Abfrage so, dass die Namen und Straßen aller Freizeitparks und die Namen der jeweils zugehörigen Gemeinde ausgegeben wird.

SELECT Freizeitpark.name, Gemeinde.name

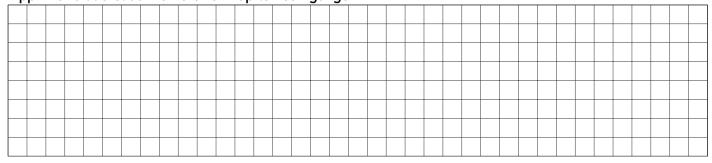
#### FROM Freizeitpark, Gemeinde

Schreibe eine SQL-Abfrage, die Namen und Art aller Schwimmbäder und den Namen und alle Einwohnerzahlen der zugehörigen Gemeinden ausgibt.



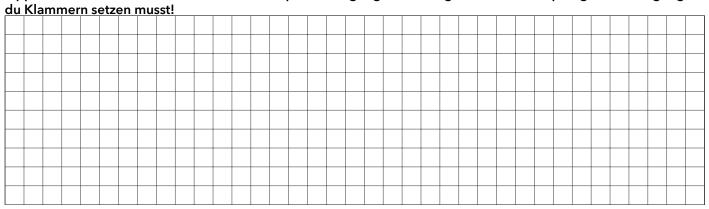
Schreibe eine SQL-Abfrage, die die Anzahl an Schwimmbädern in Gemeinden mit **mehr** als 1000 weiblichen Einwohnerinnen ausgibt.

Tipp: Hier brauchst du mehrere verknüpfte Bedingungen

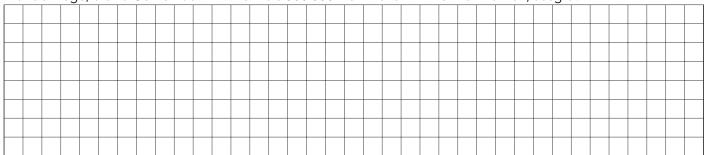


Schreibe eine SQL-Abfrage, die die Namen aller Gemeinde in Oberbayern oder Niederbayern, zu denen ein Wanderweg führt, ausgibt. Dopplungen dürfen auftreten und sollte nicht entfernt werden!

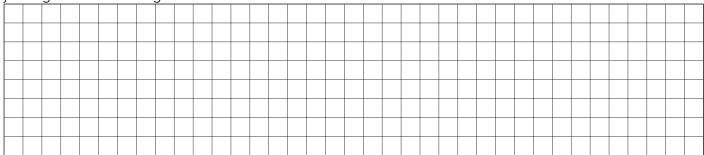
Tipp: Hier brauchst du wieder mehrere verknüpfte Bedingungen. Überlege bei der Verknüpfung von Bedingungen, ob



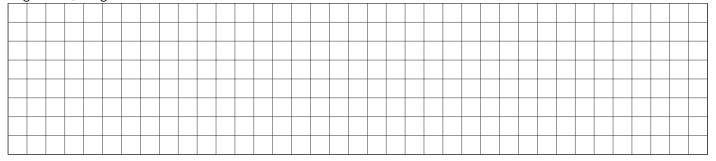
Schreibe eine SQL-Abfrage, die aus den Tabellen Gemeinde und Wanderweg\_zu\_Gemeinde die Anzahl der Wanderwege, die zu Gemeinden mit mehr als 500 000 männlichen Einwohnern führen, ausgibt.



Schreibe eine SQL-Abfrage, die eine Liste mit den Namen aller Gemeinden, die ein 'Freibad' haben, und die Namen der jeweiligen Freibäder ausgibt.



Schreibe eine SQL-Abfrage, die die Anzahl an Radwegen, die an Gemeinden im PLZ-Bereich größer als 96400 angrenzen, ausgibt.



Schreibe eine SQL-Abfrage, die die Namen aller Zoos in einer Gemeinde namens 'Erlangen' ausgibt.

Schreibe eine SQL-Abfrage, die die IDs aller Radwege, die zu Gemeinden in Oberfranken oder Unterfranken führen, ausgibt. Dopplungen sollen nicht entfernt werden.

