

Informatik 10: Datenbanken (Teil 1)



1 Wdh: Klassen und Objekte



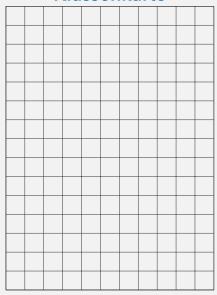
__ repräsentieren **Gegenstände** in einem Computerprogramm.

_ sind der Bauplan, der festlegt, welche Eigenschaften (

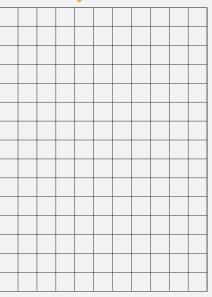
.) und **Fähigkeiten** (_______) einer bestimmten

Objektart gespeichert werden sollen. Man stellt sie dar mit:

Klassenkarte



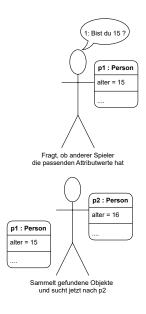
Objektkarte

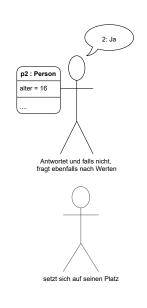


1 Generell kann man Objektkarten mit oder ohne Methoden zeichnen, solange man es insgesamt einheitlich macht. Wir zeichnen sie daher immer <u>ohne Methoden</u>.

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.

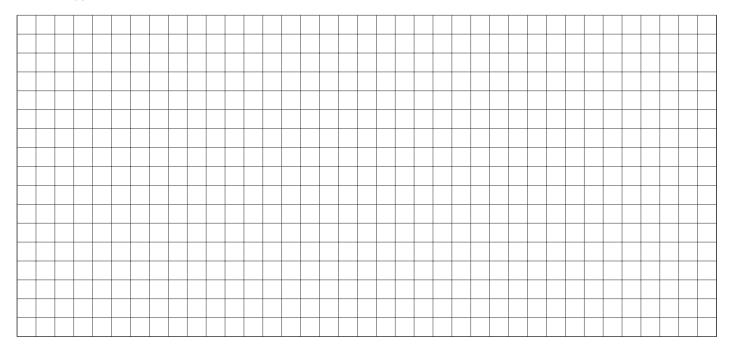






Wdh: Von der Klasse zur Tabelle

- 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



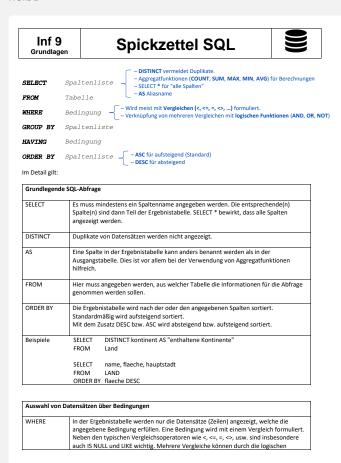
2 Wdh: Aufbau von (relationalen) Datenbanken
Datenbanken speichern Datensätze in Die
repräsentieren die Attribute (Synonym: Feld) und
bilden zusammen eine Klasse. Die (=Zeilen) entsprechen
Objekten und in den Spalten stehen die Attributwerte. Jede Tabelle hat einen
(oft auch "ID"), der Datensätze eindeutig identifiziert. Oft
werden die Datensätze hiermit einfach durchnummeriert. Im Tabellenschema wird er
unterstrichen und im Klassendiagramm immer als erstes Attribut aufgelistet.
Der Aufbau einer Tabelle kann mit oder
dargestellt werden. Dessen Aufbau ist:
${\bf TABELLENNAME}(\underline{{\bf Datentyp\ Prim\"{a}rschl\"{u}ssel}}\ ,\ {\bf Datentyp\ Spalte1},\ {\bf 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.



IS NULL	Überprüft (in einer Bedingung), ob kein Wert in einer Zelle eingetragen ist.
NULL	Bedeutet, dass kein Wert in einer Zelle eingetragen ist.
	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
	Beispiele:
	 % steht für beliebig viele Zeichen, auch keines (* bei MS Access) _ für genau ein beliebiges Zeichen (? bei MS Access)
LIKE	Kann in einer Bedingung zur Mustererkennung von Einträgen verwendet werden. Folgende zwei Platzhalter (wildcards) werden häufig eingesetzt:
	AND laufzeit <= 90 AND NOT fsk = 18
	WHERE jahr > 2015
	Beispiel
	Ausdrücke dabei sinnvoll geklammert werden
	Funktionen AND, OR und NOT verknüpft werden. Ggf. müssen die einzelnen

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!**



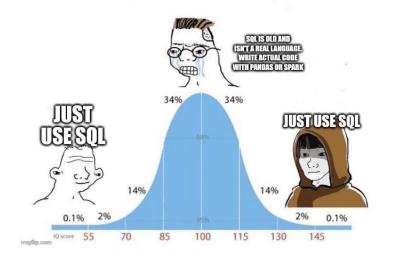
SELECT name, id FROM products WHERE discount = 0 ORDER BY price ASC;



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/

→ N Prin	s sir Noti närs Jen [ert sch	da lüss	s vo sel)	olls	tän	dig	e T																			
																											_
Stel	llt d	ie T	Гab	elle	n c	der	Dat	tenl	oar	ık n	nit l	Klas	sser	nka	rte	n di	ar.										
																											L
																											L
																											L
																											L
																											L
																											L
	-																										L
																											L
	-																										_
																											H
	-																										\vdash
																											\vdash
																											\vdash
																											H
																											H
																											T
																											T
																											T
	1													_	_	_				 		_		_		-	-

3. 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!

- 1) Zeige alle Spalten 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;





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

				Ш																															
																																			Г
																																			t
_																																			
							_					_			_																				
c	rei	be e	eine	e SC	2L-,	Abf	rag	ge,	die	aus	sgi	bt, ۱	wie	vie	ele (Gei	mei	nde	en (es ii	m F	Reg	ierı	ıng	jsbe	ezir	k ′C	Dbe	erba	aye	rn'	gib	t.		
																																			t
_	-			\vdash	-																									\vdash	\vdash	-			\vdash
_	-			\square							_																			_	-				1
	-	1																								_									
ic.	hrei	be	ein	e S	QL.	-Ab	fra	ae.	di	e N	lam	ne.	Stra	aße	ur	nd I	URL	(a	lso	die	. In	ter	net	adr	ess	e) a	alle	Zc	os	in	dei	r G	eme	ein	d
				620								- ,						(-								-,						_			
_	T					0.0.	- <u>-</u>																							П					Т
_																														-	-				H
	+			\vdash	_				-								_								-					-	+-	-	_		+
			l																																
	-																																		
_																																			
_																																			
				ne S																													m	änr	hli
				ne S																													ma	änr	nli
																																	- ma	änr	nli
																																	- ma	änr	nli
																																	- ma	änr	nli
																																	- ma	änr	nli
																																	- ma	änr	nli
																																	- ma	änr	nli
																																	- ma	änr	nli
																																	- ma	änr	nli
W(ohn	er g	rup	ppie	ert r	nac	h R	egi	eru	ngs	sbe	ezirl	< ur	nd c	den	Na	ame	en c	des	jew	veil	ige	n R	egi	ieru	ing	sbe	zirl	KS &	aus	gib	t.			
wc Sc	hrei	er g	ein	ne S	ert r	-Ak	h R	egi	, di	e c	sbe	du	c ur	nd o	nitt	Na	ne f	en c	des	jew	veil	ige	n R	egi	e ei	nes	sbe Kr	ezirl	es	(=L	gib	t.	eis)	un	ıd
Scne	hrei	be and	ein	ne S	GQL	nac	h R	egi	, di	e c	sbe die wei	du	c urr	sch	nittt dkr	llicheis	ne f	en c	des	de	reil	ige	neir	egi	e ei	ng mes	s Kr	reis	es n N	(=L	and the control of th	dkro	eis) Lar	um	id
Sc	hrei untun	ibe and g: [ein Re	ne S gier kan	GOL nost	-Ak	ofra oezii de	egi inge irk	, di	e c	die wei	duilige varieties	rch:	sch Gru	nitt dkr	lich reis	ne fes a	=läc anz	che eig	de t. Se	er Corti	ige Sen ere	neir e di	egi	e ei	nes gab	sbe Kr e r	reischach	es n N	(=Llam	and the control of th	dkro	eis) Lar	um	id
Sc me	hrei untun	ibe and g: [ein Re	ne S	GOL nost	-Ak	ofra oezii de	egi inge irk	, di des	e c	die wei	duilige varieties	rch:	sch Gru	nitt dkr	lich reis	ne fes a	=läc anz	che eig	de t. Se	er Corti	ige Sen ere	neir e di	egi	e ei	nes gab	sbe Kr e r	reischach	es n N	(=Llam	and the contract of the contra	dkro	eis) Lar	um	id
Sc me	hrei untun	ibe and g: [ein Re	ne S gier kan	GOL nost	-Ak	ofra oezii de	egi inge irk	, di des	e c	die wei	duilige varieties	rch:	sch Gru	nitt dkr	lich reis	ne fes a	=läc anz	che eig	de t. Se	er Corti	ige Sen ere	neir e di	egi	e ei	nes gab	sbe Kr e r	reischach	es n N	(=Llam	and the contract of the contra	dkro	eis) Lar	um	id
Scme	hrei untun	ibe and g: [ein Re	ne S gier kan	GOL nost	-Ak	ofra oezii de	egi inge irk	, di des	e c	die wei	duilige varieties	rch:	sch Gru	nitt dkr	lich reis	ne fes a	=läc anz	che eig	de t. Se	er Corti	ige Sen ere	neir e di	egi	e ei	nes gab	sbe Kr e r	reischach	es n N	(=Llam	and the contract of the contra	dkro	eis) Lar	um	id

T																		sgik	, c.	1								T						Π
																			_									+						
	+																		+									+			-			+
																			_									+						
	-																											4						
																												4						L
																		sgib		en a	lier	Ge	·me	eina	е, с	ale	meh	ır a	ais	75.		Jm	ian	
																											er G						gil	b
CI	15 11	IGIII	ais	\sim \sim																	HI													
	Т				.00	0 11	iaiii	IIIC	ine	un	a w	/eic	lici	ie i	_1111	VOI	nnei	r:ınr	en		T		lac		910	اعرا	ais		UK	m²	Tiat	·		Τ
					.00	011	ıarıı	IIIC	ne	une	a w	/erc	JIICI	је і		WOI	nei	r:Inr		-					910		ais		UK	m²	III	•		I
						-	14111	IIIC	ne	une	a w	/eic	DIICI	ie i	_1110	WOI	nnei	r:inr	en						910	i)Ci	ais		UK	m²				
					.00		lam	IIIC	ne	une	a w	/eic	oner	ie i		WOI	nnei	r:inr	end						gio	ije i	ais		UK	m²		•		
					.00		lanı	IIIC	ne	une	a w	/eic	DIICI	Te i	_1110	WOI	nnei	r:inr							910	i)ei	ais		UK	m²				
							latif	1110	ne	une	a w	/eic	DIICI	ie i		WOI	nnei	r:inr							910	i de la companya de l	ais		UK	m²				
					.00		lann	1110	ne	und	a w	/eic	DIICI	ie i		VOI	nei	r:inr							910	ije!	als		O K	m²				
	nrei		eine	e SO	ΩL	Abt	rag	e, «	die	die	e du	urch	nsch	nnit	ttlic	her	n mä	ännl	iche	en ur	nd v	weik	olic	hen	Ein	nwa	ohne gibt.						em	<u> </u>
	nrei		eine	e SO	ΩL	Abt	rag	e, «	die	die	e du	urch	nsch	nnit	ttlic	her	n mä	ännl	iche	en ur	nd v	weik	olic	hen	Ein	nwa	ohne						em	n
S	nrei	als	eine 10	e SC 0 kr	ΩL m² l	Abti	-rag the	e, (pro	die o La	die and	e du kre	urcheis u	nschund	nnitde	ttlicen N	herlam	n ma	ännlides	iche jew	en ur eilig	nd v en	weikLan	olic dkı	hen	Eiı es a	าพณ	ohne	rza	ahle	en a	alle	r G		
m	nrei	als	eine 10	e SC 0 kr	ΩL m² l	Abti	-rag the	e, (pro	die o La	die and	e du kre	urcheis u	nschund	nnitde	ttlicen N	herlam	n ma	ännlides	iche jew	en ur eilig	nd v en	weikLan	olic dkı	hen	Eiı es a	าพณ	bhne gibt.	rza	ahle	en a	alle	r G		
m S	nrei	als	eine 10	e SC 0 kr	ΩL m² l	Abti	-rag the	e, (pro	die o La	die and	e du kre	urcheis u	nschund	nnitde	ttlicen N	herlam	n ma	ännlides	iche jew	en ur eilig	nd v en	weikLan	olic dkı	hen	Eiı es a	าพณ	bhne gibt.	rza	ahle	en a	alle	r G		
) S	nrei	als	eine 10	e SC 0 kr	ΩL m² l	Abti	-rag the	e, (pro	die o La	die and	e du kre	urcheis u	nschund	nnitde	ttlicen N	herlam	n ma	ännlides	iche jew	en ur eilig	nd v en	weikLan	olic dkı	hen	Eiı es a	าพณ	bhne gibt.	rza	ahle	en a	alle	r G		



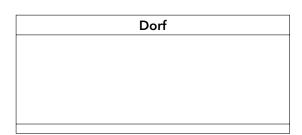
- 1. Visualisiere (mit Bleistift), wer Häuptling in welchem Dorf ist.
- 2. Überlege, wie du allgemein für diese zwei Tabellen darstellen kannst, wie sie (und ihre Spalten) miteinander in Beziehung stehen.





Tabellenbeziehung im Klassendiagramm

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



	Bewo	ohner		

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

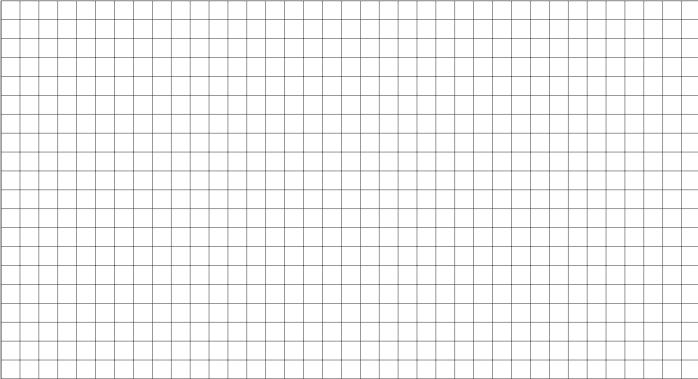


Bearbeite diese Aufgabe auf artemis.tum.de.

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:





SQL: Tabellen verbinden



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	Muster

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

______,

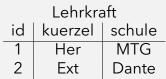
Zum Beispiel kann man in SQL-Island die Daten aller Dörfer und ihrer zugehörigen Häuptlinge so ausgeben:

SELECT*

FROM Dorf, Bewohner

WHERE Dorf.haeuptling = Bewohner.bewohnernr

8 Join Beispiel







Ergebnistabelle des Kreuzprodukts:

id	kuerzel	schule	id	ort
1	Her	MTG	MTG	Haidh.
2	Ext	Dante	MTG	Haidh.
1	Her	MTG	Dante	Sendl.
2	Ext	Dante	Dante	Sendl.

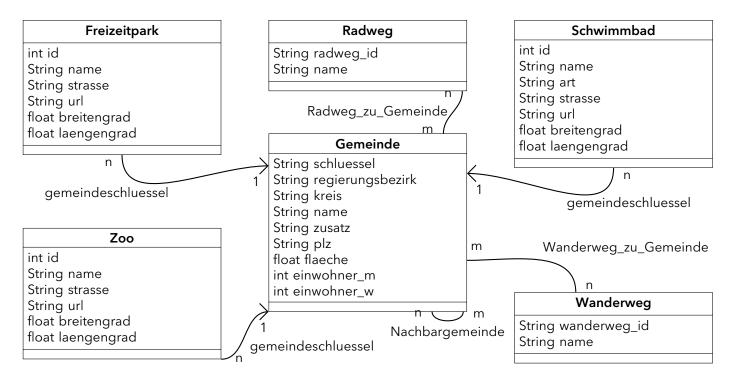
Ergebnistabelle des Joins

id	kuerzel	schule	id	ort
1	Her	MTG	MTG	Haidh.
2	Ext	Dante	Dante	Sendl.



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 du Klammern setzen musst! 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.

