

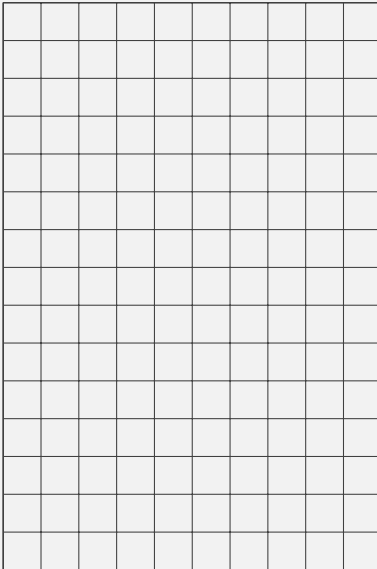


1 Wdh: Klassen und Objekte

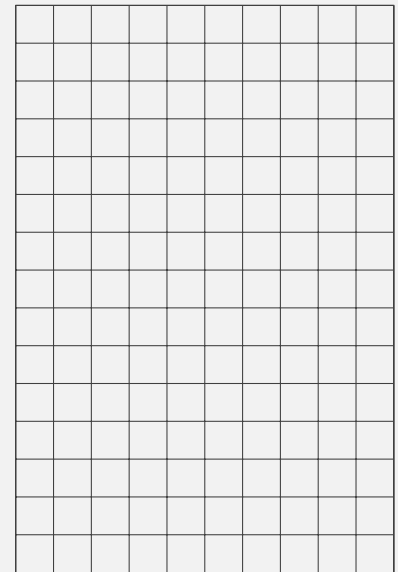


_____ repräsentieren **Gegenstände** in einem Computerprogramm.
_____ sind der **Bauplan**, der festlegt, welche **Eigenschaften** (
_____) und **Fähigkeiten** (_____) einer bestimmten
Objektart gespeichert werden sollen. Man stellt sie dar mit:

Klassenkarte



Objektkarte

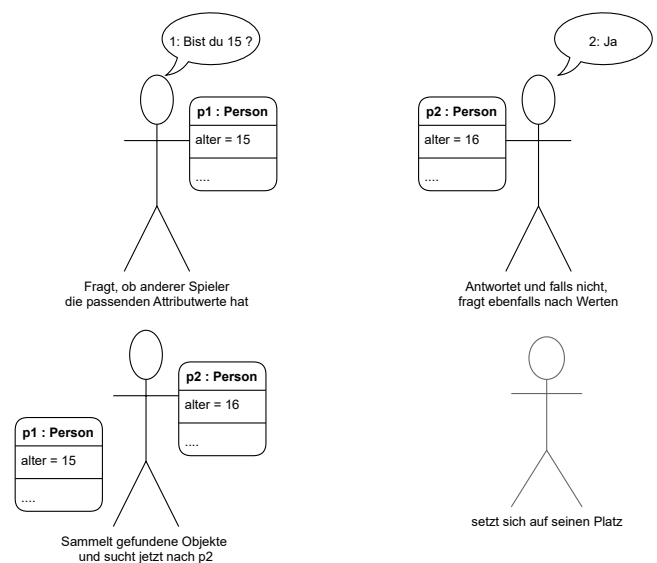


❗ Generell kann man Objektkarten mit oder ohne Methoden zeichnen, solange man es insgesamt einheitlich macht. Wir zeichnen sie daher immer ohne Methoden.



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.




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.

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

Inf 9 Grundlagen	Spickzettel SQL	
----------------------------	------------------------	-----------------------------------------------------------------------------------

SELECT	Spaltenliste	<ul style="list-style-type: none"> - DISTINCT vermeidet Duplikate. - Aggregatfunktionen (COUNT, SUM, MAX, MIN, AVG) für Berechnungen - SELECT * für "alle Spalten" - AS Aliasname
FROM	Tabelle	
WHERE	Bedingung	<ul style="list-style-type: none"> - Wird meist mit Vergleichen (<, <=, =, >, >=) formuliert. - Verknüpfung von mehreren Vergleichen mit logischen Funktionen (AND, OR, NOT)
GROUP BY	Spaltenliste	
HAVING	Bedingung	
ORDER BY	Spaltenliste	<ul style="list-style-type: none"> - ASC für aufsteigend (Standard) - DESC für absteigend

Im Detail gilt:

Grundlegende SQL-Abfrage	
SELECT	Es muss mindestens ein Spaltenname angegeben werden. Die entsprechende(n) Spalte(n) sind dann Teil der Ergebnistabelle. SELECT * bewirkt, dass alle Spalten angezeigt werden.
DISTINCT	Duplikate von Datensätzen werden nicht angezeigt.
AS	Eine Spalte in der Ergebnistabelle kann anders benannt werden als in der Ausgangstabelle. Dies ist vor allem bei der Verwendung von Aggregatfunktionen hilfreich.
FROM	Hier muss angegeben werden, aus welcher Tabelle die Informationen für die Abfrage genommen werden sollen.
ORDER BY	Die Ergebnistabelle wird nach der oder den angegebenen Spalten sortiert. Standardmäßig wird aufsteigend sortiert. Mit dem Zusatz DESC bzw. ASC wird absteigend bzw. aufsteigend sortiert.
Beispiele	<pre> SELECT DISTINCT kontinent AS "enthaltene Kontinente" FROM Land SELECT name, flaeche, hauptstadt FROM LAND ORDER BY flaeche DESC </pre>

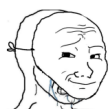
Auswahl von Datensätzen über Bedingungen	
WHERE	In der Ergebnistabelle werden nur die Datensätze (Zeilen) angezeigt, welche die angegebene Bedingung erfüllen. Eine Bedingung wird mit einem Vergleich formuliert. Neben den typischen Vergleichsoperatoren wie <, <=, =, >, >=, usw. sind insbesondere auch IS NULL und LIKE wichtig. Mehrere Vergleiche können durch die logischen

	Funktionen AND, OR und NOT verknüpft werden. Ggf. müssen die einzelnen Ausdrücke dabei sinnvoll geklammert werden
	Beispiel WHERE jahr > 2015 AND lauffzeit <= 90 AND NOT fsk = 18
LIKE	Kann in einer Bedingung zur Mustererkennung von Einträgen verwendet werden. Folgende zwei Platzhalter (wildcards) werden häufig eingesetzt: <ul style="list-style-type: none"> • % steht für beliebig viele Zeichen, auch keines (* bei MS Access) • _ für genau ein beliebiges Zeichen (? bei MS Access) Beispiele: <ul style="list-style-type: none"> • 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.

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

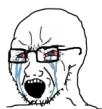
Gruppierung	
GROUP BY	Datensätze mit demselben Wert in der angegebenen Spalte werden gruppiert. Gruppierungen sind nur in Kombination mit Aggregatfunktionen sinnvoll.
HAVING	An gruppierte Datensätze werden Bedingungen mit HAVING formuliert.
Beispiel	SELECT fsk, MIN(laufzeit) FROM Film WHERE genre1="Filmkomödie"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



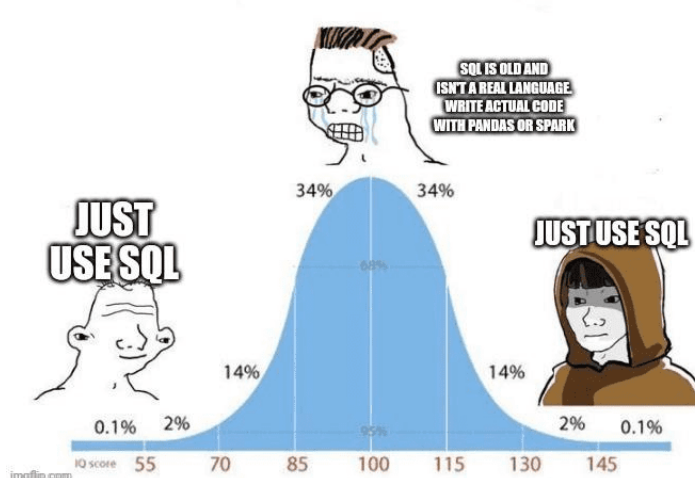
```
SELECT name, id
FROM products
WHERE discount = 0
ORDER BY price ASC;
```



```
sELeCt nAmE, iD
fRoM PrOdUcTs
WhErE dIsCoUnT = 0
OrDeR bY pRiCe AsC;
```

'Sarcastic Query Language' • by u/casperdewith

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





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.

i) SELECT name
FROM land
ORDER BY name DESC;

2) Zeige die Spalten name und hauptstadt der Tabelle land.

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

3) Zeige die durchschnittliche Einwohnerzahl aller Länder.

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

4) Zeige die Namen aller Länder in alphabetisch absteigender Reihenfolge.

iv) SELECT *
FROM land;

5) Zeige die Hauptstädte der Länder, deren Einwohnerzahl größer als 50 Mio ist.

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

6) Zeige die Anzahl aller Länder, deren Name mit 'land' endet.

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

7) Zeige die Namen aller Länder, deren Fläche zwischen 100 und 999 Tausend km² liegt.

vii) SELECT AVG(einwohner)
FROM land;

8) Zeige die Namen der Länder, die mit 'D' beginnen oder mit 'd' aufhören.

viii) SELECT name, hauptstadt
FROM land;

9) Zeige die Namen der drei Länder mit der größten Einwohnerzahl.

ix) SELECT hauptstadt
FROM land
WHERE einwohner > 50;

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

[illegible][illegible][illegible]

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



Wenn Datensätze mittels Primärschlüssel in einer anderen Tabelle verwendet werden, spricht man dort von einem Fremdschlüssel. Im **Tabellenschema** werden die

_____ durch **überstreichen** (manchmal auch) markiert. Ein Beispiel in SQL-Island ist der Häuptling eines Dorfes, der in der Tabelle Dorf mittels bewohnernr eingetragen wird. Die **bewohnernr** ist hierbei _____ in der **Tabelle Bewohner** und _____ in der **Tabelle Dorf** (heißt hier aber **haeuptling**).

5 Tabellenbeziehungen im Klassendiagramm



TabelleA
int id
String spalte1
...

TabelleB
int id
String spalte1
...



Erstelle ein Klassendiagramm für die Datenbank unter dbiu.de/flugverspaetungen/.

Achte auf korrektes Format, Datentypen und Kardinalitäten. Zeichne das Diagramm anschließend unten auf:

A full-page sheet of white graph paper with a light gray grid. The grid consists of small squares, approximately 10 units wide by 10 units high. There are no margins or additional markings on the page.



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 Datensätze angezeigt? Falls nicht, nach welchem Muster werden die beiden Tabellen miteinander kombiniert?

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 miteinander 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



Lehrkraft		
id	kuerzel	schule
1	Her	MTG
2	Ext	Dante

SELECT *
FROM Lehrkraft, Schule
WHERE Lehrkraft.schule = Schule.id

Schule	
id	ort
MTG	Haidh.
Dante	Sendl.

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.

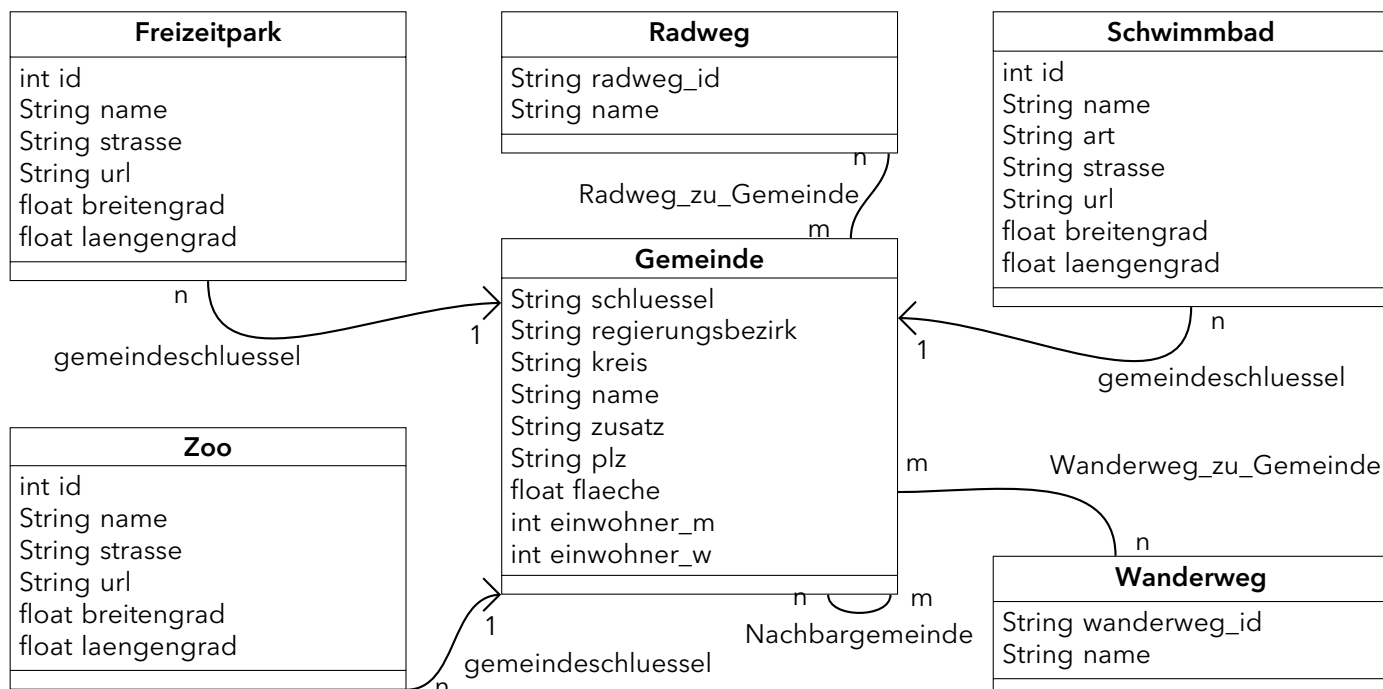


SQL mit Kreuzprodukt und Join

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 unten 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.



[illegible][illegible]