

Informatik Q2 Abels

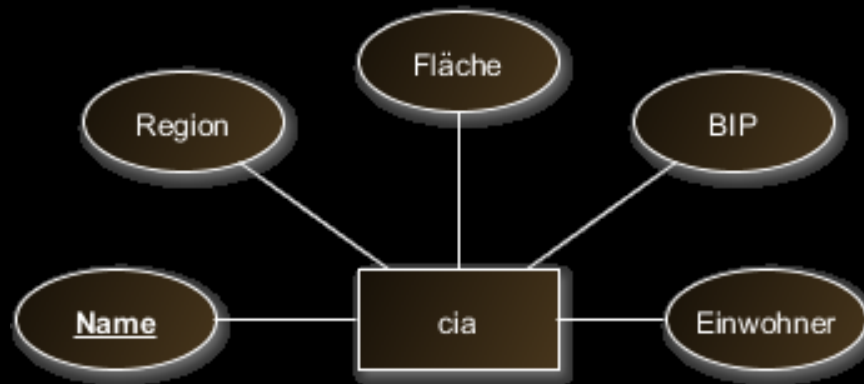


SQL

SQL

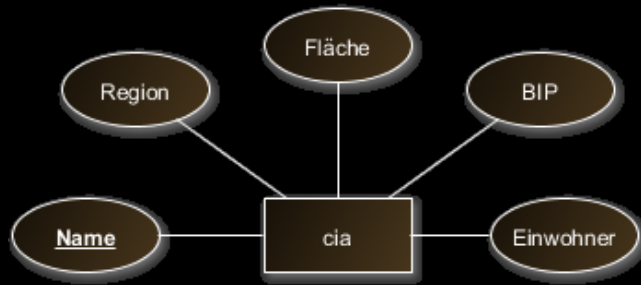
Structured Query Language

SQL



Name	Region	Fläche	Einwohner	BIP
Afghanistan	Asien	652.000	25.838.797	21.000.000.000
Albanien	Europa	28.748	3.490.435	5.600.000.000
Algerien	Afrika	2.381.740	31.193.917	147.600.000.000
Amerikanische Samoa-Inseln	Ozeanien	199	65.446	150.000.000
Andorra	Europa	468	66.824	1.200.000.000
Angola	Afrika	1.246.700	10.145.267	11.600.000.000
Anguilla	Mittelamerika	91	11.797	88.000.000
Antarktik	Antarktis	14.000.000	0	0
Antigua und Barbuda	Mittelamerika	442	66.422	524.000.000
Argentinien	Südamerika	2.766.890	36.955.182	367.000.000.000
Armenien	Osteuropa	29.800	3.344.336	9.900.000.000
Aruba	Mittelamerika	193	69.539	1.600.000.000
Ashmore und Cartier Inseln	Südostasien	5	0	0
Australien	Ozeanien	7.686.850	19.169.083	416.200.000.000
Österreich	Europa	83.858	8.131.111	190.600.000.000
Azerbajjan	Osteuropa	86.600	7.748.163	14.000.000.000
Bahamas	Mittelamerika	13.940	294.982	5.580.000.000
Bahrain	Mittlerer Osten	620	634.137	8.600.000.000
Baker Island	Ozeanien	1	0	0
Bangladesh	Asien	144.000	129.194.224	187.000.000.000

SQL



Name	Region	Fläche	Einwohner	BIP
Afghanistan	Asien	652.000	25.838.797	21.000.000.000
Albanien	Europa	28.748	3.490.435	5.600.000.000
Algerien	Afrika	2.381.740	31.193.917	147.600.000.000
Amerikanische Samoa-Inseln	Ozeanien	199	65.446	150.000.000
Andorra	Europa	468	66.824	1.200.000.000
Angola	Afrika	1.246.700	10.145.267	11.600.000.000
Anguilla	Mittelamerika	91	11.797	88.000.000
Antarktik	Antarktis	14.000.000	0	0
Antigua und Barbuda	Mittelamerika	442	66.422	524.000.000
Argentinien	Südamerika	2.766.890	36.955.182	367.000.000.000
Armenien	Osteuropa	29.800	3.344.336	9.900.000.000
Aruba	Mittelamerika	193	69.539	1.600.000.000
Ashmore und Cartier Inseln	Südostasien	5	0	0
Australien	Ozeanien	7.686.850	19.169.083	416.200.000.000
Österreich	Europa	83.858	8.131.111	190.600.000.000
Azerbajjan	Osteuropa	86.600	7.748.163	14.000.000.000
Bahamas	Mittelamerika	13.940	294.982	5.580.000.000
Bahrain	Mittlerer Osten	620	634.137	8.600.000.000
Baker Island	Ozeanien	1	0	0
Bangladesh	Asien	144.000	129.194.224	187.000.000.000

```
SELECT Name  
FROM cia
```

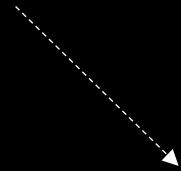


Name
Afghanistan
Albanien
Algerien
Amerikanische Samoa-Inseln
Andorra
Angola
Anguilla
Antarktik
Antigua und Barbuda
Argentinien
Armenien
Aruba
Australien
Österreich
Azerbajjan
Bahamas
Bahrain
Bangladesh

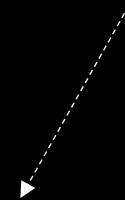
SELECT

allgemein

Eine bestimmte Spalte aus einer bestimmten Tabelle selektieren



Mehrere Spalten aus einer bestimmten Tabelle selektieren



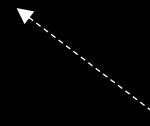
```
SELECT Spalte  
FROM Tabelle
```



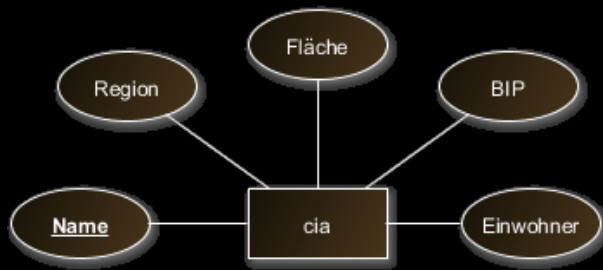
```
SELECT *  
FROM Tabelle
```



```
SELECT Spalte1, Spalte2  
FROM Tabelle
```



Alle Spalten aus einer bestimmten Tabelle selektieren



SELECT

Beispiel



```
SELECT Name
FROM cia
```

Name
Afghanistan
Albanien
Algerien
Amerikanische Samoa-Inseln
Andorra
Angola
Anguilla
Antarktik
Antigua und Barbuda
Argentinien
Armenien
Aruba
Australien
Österreich
Azerbaidjan
Bahamas
Bahrain
Bangladesh



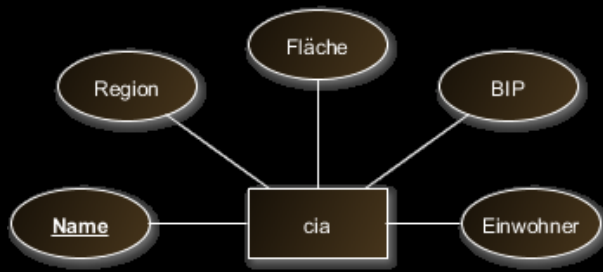
```
SELECT *
FROM cia
```

Name	Region	Fläche	Einwohner	BIP
Afghanistan	Asien	652.000	25.838.797	21.000.000.000
Albanien	Europa	28.748	3.490.435	5.600.000.000
Algerien	Afrika	2.381.740	31.193.917	147.600.000.000
Amerikanische Samoa-Inseln	Ozeanien	199	65.446	150.000.000
Andorra	Europa	468	66.824	1.200.000.000
Angola	Afrika	1.246.700	10.145.267	11.600.000.000
Anguilla	Mittelamerika	91	11.797	88.000.000
Antarktik	Antarktis	14.000.000	0	0
Antigua und Barbuda	Mittelamerika	442	66.422	524.000.000
Argentinien	Südamerika	2.766.890	36.955.182	367.000.000.000
Armenien	Osteuropa	29.800	3.344.336	9.900.000.000
Aruba	Mittelamerika	193	69.539	1.600.000.000
Ashmore und Cartier Inseln	Südostasien	5	0	0
Australien	Ozeanien	7.686.850	19.169.083	416.200.000.000
Österreich	Europa	83.858	8.131.111	190.600.000.000
Azerbaidjan	Osteuropa	86.600	7.748.163	14.000.000.000
Bahamas	Mittelamerika	13.940	294.982	5.580.000.000
Bahrain	Mittlerer Osten	620	634.137	8.600.000.000
Baker Island	Ozeanien	1	0	0
Bangladesh	Asien	144.000	129.194.224	187.000.000.000



```
SELECT Name, Region
FROM cia
```

Name	Region
Afghanistan	Asien
Albanien	Europa
Algerien	Afrika
Amerikanische Samoa-Inseln	Ozeanien
Andorra	Europa
Angola	Afrika
Anguilla	Mittelamerika
Antarktik	Antarktis
Antigua und Barbuda	Mittelamerika
Argentinien	Südamerika
Armenien	Osteuropa
Aruba	Mittelamerika
Australien	Ozeanien
Österreich	Europa
Azerbaidjan	Osteuropa
Bahamas	Mittelamerika
Bahrain	Mittlerer Osten
Bangladesh	Asien



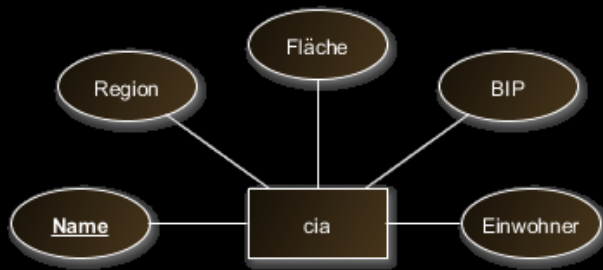
SELECT

+ - * /

```
SELECT Name, Einwohner/1E6  
FROM cia
```

$1E6 = 1 \cdot 10^6$

Name	Einwohner/1E6
Afghanistan	25,84
Albanien	3,49
Algerien	31,19
Amerikanische Samoa-Inseln	0,07
Andorra	0,07
Angola	10,15
Anguilla	0,01
Antarktik	NULL
Antigua und Barbuda	0,07
Argentinien	36,96
Armenien	3,34
Aruba	0,07
Australien	19,17
Österreich	8,13
Azerbaidjan	7,75
Bahamas	0,29
Bahrain	0,63
Bangladesh	129,19

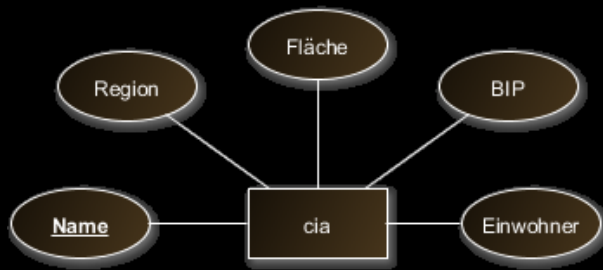


SELECT

AS

```
SELECT Name, BIP/Einwohner AS ProKopf  
FROM cia
```

Name	ProKopf
Afghanistan	812,73
Albanien	1.604,38
Algerien	4.731,69
Amerikanische Samoa-Inseln	2.291,97
Andorra	17.957,62
Angola	1.143,39
Anguilla	7.459,52
Antarktik	NULL
Antigua und Barbuda	7.888,95
Argentinien	9.930,95
Armenien	2.960,23
Aruba	23.008,67
Australien	21.712,05
Österreich	23.440,83
Azerbaidjan	1.806,88
Bahamas	18.916,41
Bahrain	13.561,74
Bangladesh	1.447,43



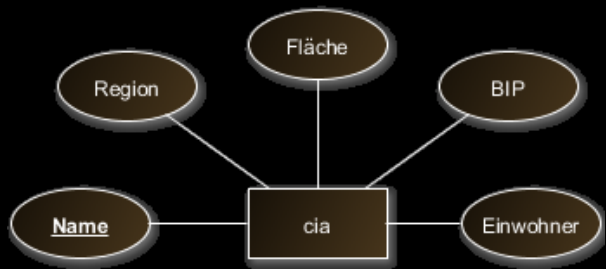
SELECT

DISTINCT



```
SELECT DISTINCT Region  
FROM cia
```

Region
Asien
Europa
Afrika
Ozeanien
Mittelamerika
Antarktis
Südamerika
Osteuropa
Mittlerer Osten
Nordamerika
Südostasien
Welt
Arktik-Region



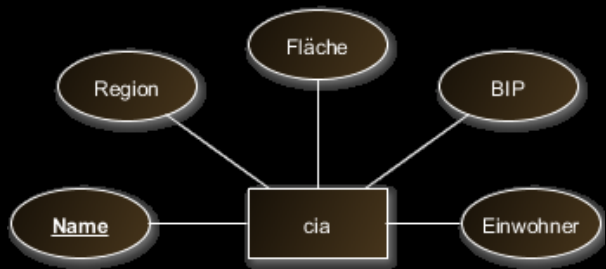
SELECT

WHERE

```
SELECT Name, Einwohner  
FROM cia  
WHERE Einwohner > 2E8
```

Name	Einwohner
China	1.261.832.482
Indien	1.014.003.817
Indonesien	224.784.210
Vereinigte Staaten von Amerika	275.562.673

- = gleich
- > größer als
- < kleiner als
- >= größer als oder gleich
- <= kleiner als oder gleich
- <> nicht gleich



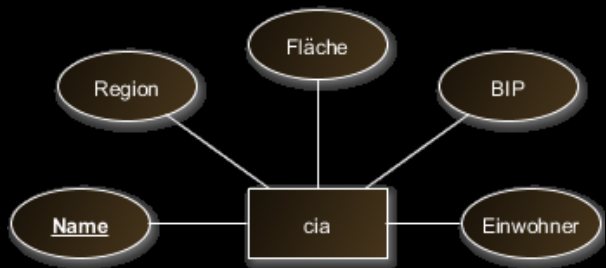
SELECT

BETWEEN

Name	BIP
Albanien	5.600.000.000
Bahamas	5.580.000.000
Bosnien und Herzegovina	6.200.000.000
Botswana	5.700.000.000
Brunei	5.600.000.000
Zentral-Afrikanische Republik	5.800.000.000
Tschad	7.600.000.000
Estland	7.900.000.000
Fidschi	5.900.000.000
Gabun	7.900.000.000
Island	6.420.000.000
Laos	7.000.000.000
Macau	7.650.000.000
Macedonia, The Former Yugoslav Republic of	7.600.000.000
Mongolei	6.100.000.000
Namibia	7.100.000.000
Rwanda	5.900.000.000
Tajikistan	6.200.000.000
Turkmenistan	7.700.000.000

```
SELECT Name, BIP
FROM cia
WHERE BIP BETWEEN 5E09 AND 8E09
```

kann auch Text sein



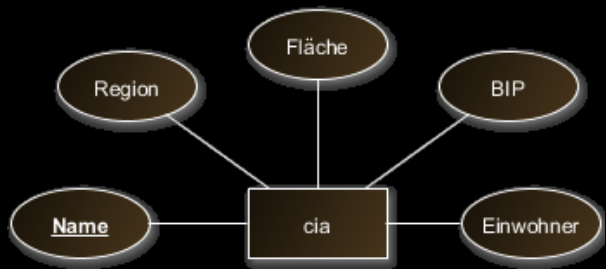
SELECT

LIKE

```
SELECT Name
FROM cia
WHERE Name LIKE '%land%'
```

– 1 Zeichen
% bel. viele Zeichen

Name
British Virgin Islands
Christmas Island
Cocos (Keeling) Islands
Cook Islands
Estland
Falkland Islands (Islas Malvinas)
Faroe Islands
Finnland
Deutschland
Griechenland
Grönland
Island
Irland
Niederlande
Niederländische Antillen
Neuseeland
Norfolk Island
Northern Mariana Islands
Pitcairn Islands
Russland
Solomon Islands
Swaziland
Thailand
Virgin Islands

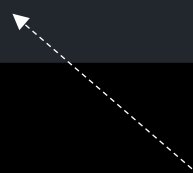


SELECT

IN

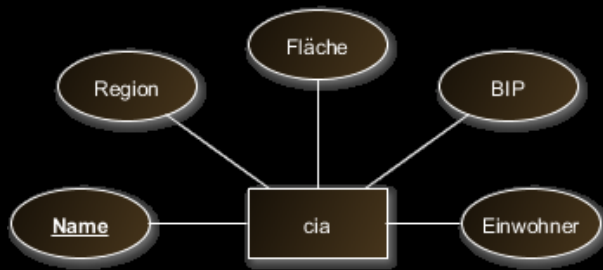


```
SELECT Name, Einwohner  
FROM cia  
WHERE Name IN ('Albanien', 'Estland', 'China')
```



auch möglich: **NOT IN**

Name	Einwohner
Albanien	3.490.435
China	1.261.832.482
Estland	1.431.471



SELECT

AND OR NOT



```
SELECT Name
FROM cia
WHERE Einwohner > 2e7 AND Region = "Europa"
```

auch möglich: **OR**, **NOT**

Name

Frankreich

Deutschland

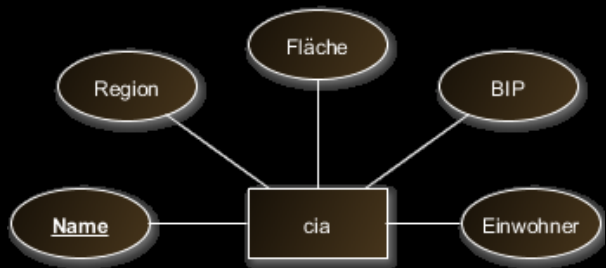
Italien

Polen

Rumänien

Spanien

Groß-Britannien



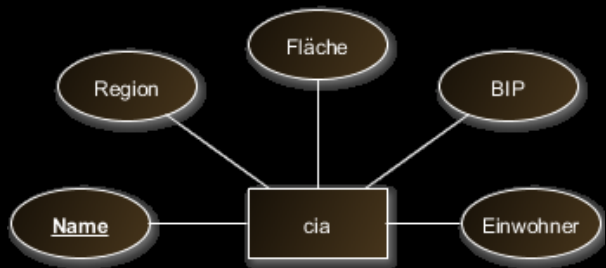
SELECT

IS NULL

```
SELECT *  
FROM cia  
WHERE Einwohner IS NULL
```

Name	Region	Fläche	Einwohner	BIP
Antarktik	Antarktis	14.000.000	NULL	NULL

auch möglich: **IS NOT NULL**



SELECT

ORDER BY

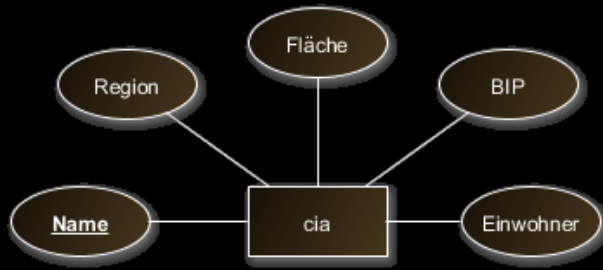
```

SELECT Name, BIP/Einwohner AS ProKopf
FROM cia
ORDER BY ProKopf DESC
    
```

ASC | aufsteigend
DESC | absteigend

auch möglich:
ORDER BY Attribut1 **DESC**, Attribut2 **ASC**, ...

Name	ProKopf
Luxemburg	33.608,53
Vereinigte Staaten von Amerika	33.585,83
Bermuda	31.747,54
Monaco	27.450,86
Schweiz	27.126,12
Cayman-Inseln	26.752,58
Norwegen	24.837,31
Jersey	24.742,73
Dänemark	23.930,02
Belgien	23.766,04
Singapur	23.607,27
Österreich	23.440,83
Japan	23.310,95
Island	23.230,15
Frankreich	23.141,87
Kanada	23.090,63
Aruba	23.008,67
Niederlande	22.973,48
Kuwait	22.699,96
Liechtenstein	22.665,88
Deutschland	22.512,78



SELECT

LIMIT

```
SELECT Name, BIP/Einwohner AS ProKopf
FROM cia
ORDER BY ProKopf DESC, Name ASC
LIMIT 5
```

Name	ProKopf
Luxemburg	33.608,53
Vereinigte Staaten von Amerika	33.585,83
Bermuda	31.747,54
Monaco	27.450,86
Schweiz	27.126,12



Übung 1 – SELECT

Input

<https://sql-tutorial.de/home/lektionen.php?lektion=1>

Übungen

<https://sql-tutorial.de/home/uebungen.php?lektion=1>





Übung 1 – SELECT



1
SELECT *
FROM cia;

2
SELECT Name
FROM cia;

3
SELECT Name
FROM cia
WHERE Einwohner > 1E08;

4
SELECT Name
FROM cia
WHERE (BIP BETWEEN 1E09 AND 100E09) AND Region = „Asien“;

5
SELECT Name, Einwohner
FROM cia
WHERE Name IN („Frankreich“, „Deutschland“, „Polen“);

6
SELECT Name, Einwohner/1E06, Region
FROM cia
WHERE Region = "Südamerika";

7
SELECT Name, BIP/Einwohner AS ProKopf
FROM cia
WHERE Einwohner > 2E08;

8
SELECT Name, BIP/Einwohner AS ProKopf
FROM cia
WHERE Einwohner > 2E08;
ORDER BY ProKopf DESC;

9
SELECT Name
FROM cia
WHERE Name LIKE "%Vereinigte%";

10
SELECT DISTINCT Region
FROM cia
GROUP BY Region;

11
SELECT *
FROM cia
WHERE BIP is NULL;



Tagebucheintrag

SQL

select ... from ... where

<, <=, =, <>, >=, >

and, or, not

between

like

in, not in

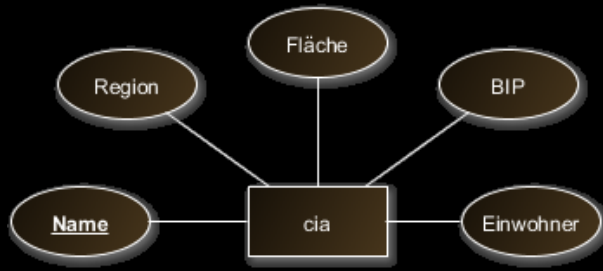
is null, not is null

distinct

as

order by

limit



SELECT

Aggregatfunktionen

```
SELECT SUM(Einwohner), COUNT(Einwohner), MIN(Einwohner), MAX(Einwohner), AVG(Einwohner)
FROM cia
WHERE Region = "Asien"
```

SUM(Einwohner)	COUNT(Einwohner)	MIN(Einwohner)	MAX(Einwohner)	AVG(Einwohner)
3.054.634.511,00	18	301.475	1.261.832.482	169.701.917,28



Übung 2 – Aggregatfunktionen

Input

<https://sql-tutorial.de/home/lektionen.php?lektion=2>

Übungen

<https://sql-tutorial.de/home/uebungen.php?lektion=2>





Übung 2 – Aggregatfunktionen



1

```
SELECT COUNT(*)  
FROM cia
```



2

```
SELECT SUM(Einwohner)  
FROM cia
```



3

```
SELECT AVG(BIP)  
FROM cia
```



4

```
SELECT SUM(Einwohner), SUM(BIP)  
FROM cia  
WHERE Region = 'Europa'
```



5

```
SELECT MIN(Fläche), MAX(Fläche)  
FROM cia
```



6

```
SELECT COUNT(DISTINCT Region)  
FROM cia
```





Tagebucheintrag



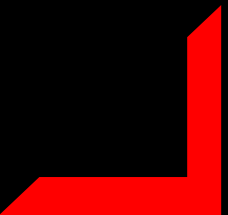
sum(...)

count(...)

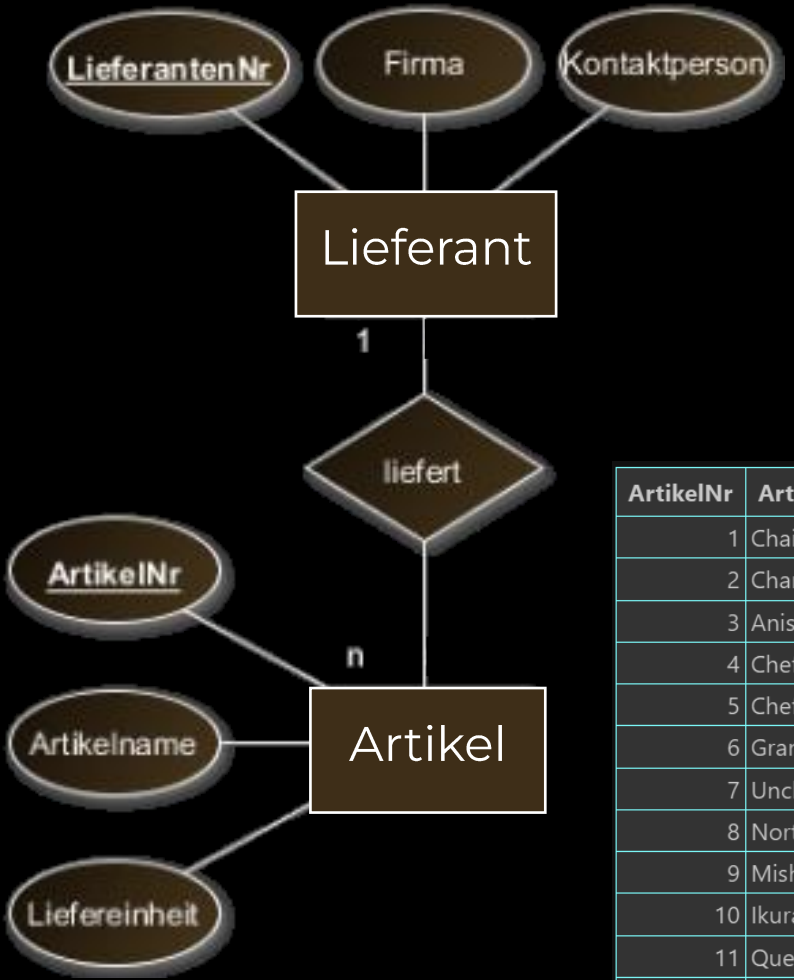
min(...)

max(...)

avg(...)



SELECT



Lieferant

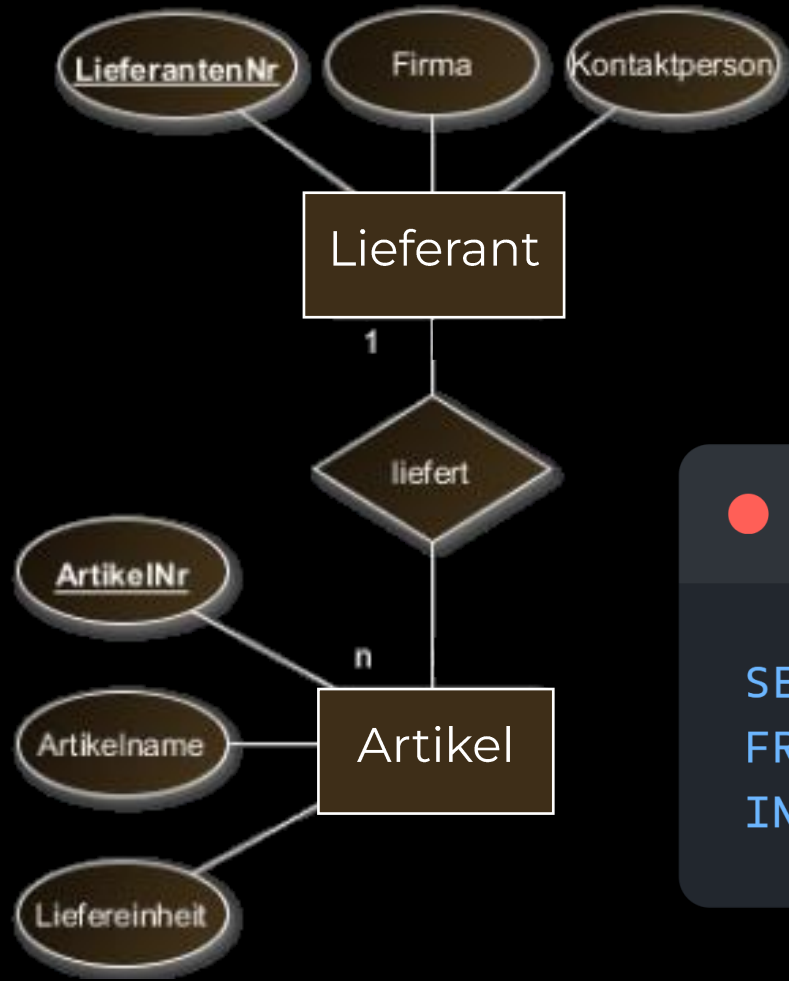
Artikel

LieferantenNr	Firma	Kontaktperson
1	Exotic Liquids	Charlotte Cooper
2	New Orleans Cajun Delights	Shelley Burke
3	Grandma Kelly's Homestead	Regina Murphy
4	Tokyo Traders	Yoshi Nagase
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra
6	Mayumi's	Mayumi Ohno
7	Pavlova, Ltd.	Ian Devling
8	Specialty Biscuits, Ltd.	Peter Wilson
9	PB Knäckebröd AB	Lars Peterson
10	Refrescos Americanas LTDA	Carlos Diaz

ArtikelNr	Artikelname	Liefereinheit	LieferantenNr
1	Chai	10 Kartons x 20 Beutel	1
2	Chang	24 x 12-oz-Flaschen	1
3	Aniseed Syrup	12 x 550-ml-Flaschen	1
4	Chef Anton's Cajun Seasoning	48 x 6-oz-Gläser	2
5	Chef Anton's Gumbo Mix	36 Kartons	2
6	Grandma's Boysenberry Spread	12 x 8-oz-Gläser	3
7	Uncle Bob's Organic Dried Pears	12 x 1-lb-Packungen	3
8	Northwoods Cranberry Sauce	12 x 12-oz-Gläser	3
9	Mishi Kobe Niku	18 x 500-g-Packungen	4
10	Ikura	12 x 200-ml-Gläser	4
11	Queso Cabrales	1-kg-Paket	5
12	Queso Manchego La Pastora	10 x 500-g-Packungen	5
13	Konbu	2-kg-Karton	6
14	Tofu	40 x 100-g-Packungen	6
15	Genen Shouyu	24 x 250-ml-Flaschen	6
16	Pavlova	32 x 500-g-Kartons	7
17	Alice Mutton	20 x 1-kg-Dosen	7
18	Carnarvon Tigers	16-kg-Paket	7
19	Teatime Chocolate Biscuits	10 Kartons x 12 Stück	8
20	Sir Rodney's Marmalade	30 Geschenkkartons	8

SELECT

JOIN / INNER JOIN



```
SELECT A.Artikelname, L.Kontaktperson
FROM Lieferant L
INNER JOIN Artikel A ON L.LieferantenNr = A.LieferantenNr
```

auch möglich:

```
INNER JOIN Tabelle1 ON Tabelle1.Attribut1 = Tabelle0.Attribut1
INNER JOIN Tabelle2 ON Tabelle2.Attribut2 = Tabelle0.Attribut2
```

...

Artikelname	Kontaktperson
Chai	Charlotte Cooper
Chang	Charlotte Cooper
Aniseed Syrup	Charlotte Cooper
Chef Anton's Cajun Seasoning	Shelley Burke
Chef Anton's Gumbo Mix	Shelley Burke
Grandma's Boysenberry Spread	Regina Murphy



Übung 3 – Join

Input

<https://sql-tutorial.de/home/lektionen.php?lektion=3>

Übungen

<https://sql-tutorial.de/home/uebungen.php?lektion=3>





Übung 3 – Join



```
1
SELECT K.Kategoriename
FROM Kategorie K
JOIN Artikel A ON K.KategorieNr = A.KategorieNr
WHERE A.Artikelname = "Teatime Chocolate Biscuits";
```

```
2
SELECT A.Artikelname, A.ArtikelNr, L.Firma
FROM Artikel A
JOIN Lieferant L ON A.LieferantenNr = L.LieferantenNr;
```

```
3
SELECT V.Firma
FROM Versandfirma V
JOIN Bestellung B ON V.FirmenNr = B.FirmenNr
WHERE B.BestellNr = 10266;
```

```
4
SELECT P.Nachname
FROM Personal P
JOIN Bestellung B
ON P.PersonalNr = B.PersonalNr
WHERE B.BestellNr = 10266;
```

```
5
SELECT A.Artikelname, A.ArtikelNr, L.Firma, K.Kategoriename
FROM Artikel A
JOIN Lieferant L ON A.LieferantenNr = L.LieferantenNr
JOIN Kategorie K ON A.KategorieNr = K.KategorieNr;
```

```
6
SELECT B.ArtikelNr, B.Einzelpreis, B.Einzelpreis*B.Anzahl AS Gesamtpreis
FROM Bestelldetails B
WHERE B.BestellNr = 11031;
```

```
7
SELECT COUNT(B.KundenNr)
FROM Bestellung B
JOIN Personal P ON P.PersonalNr = B.PersonalNr
JOIN Kunde K ON P.KundenCode = K.KundenCode
WHERE K.Nachname = "Buchanan";
```

```
8
SELECT A.Artikelname, A.Lagerbestand, BD.Anzahl
FROM Artikel A
JOIN Bestelldetails BD ON A.ArtikelNr = BD.ArtikelNr
JOIN Bestellung B ON BD.BestellNr = B.BestellNr
JOIN Kunde K ON B.KundenCode = K.KundenCode
WHERE K.Firma = "Rattlesnake Canyon Grocery" AND B.Bestelldatum = "1998-05-06" AND A.Lagerbestand < BD.Anzahl;
```

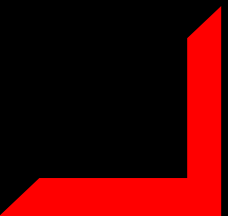
```
9
SELECT A.Artikelname
FROM Artikel A
JOIN Bestelldetails BD ON A.ArtikelNr = BD.ArtikelNr
JOIN Bestellung B ON BD.BestellNr = B.BestellNr
JOIN Kunde K ON B.KundenCode = K.KundenCode
WHERE K.Firma = "Richter Supermarkt";
```

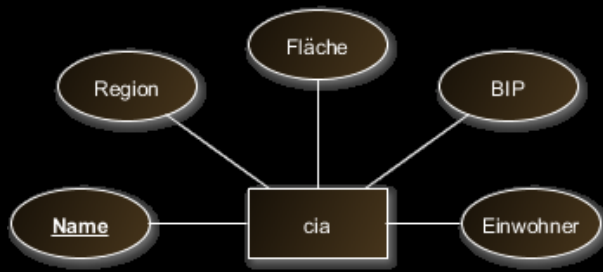
```
10
SELECT K.Firma
FROM Lieferant L
JOIN Artikel A ON L.LieferantenNr = A.LieferantenNr
JOIN Bestelldetails BD ON A.ArtikelNr = BD.ArtikelNr
JOIN Bestellung B ON BD.BestellNr = B.BestellNr
JOIN Kunde K ON B.KundenCode = K.KundenCode
WHERE L.Firma = "Escargots Nouveaux";
```



Tagebucheintrag

(inner) join ... on





SELECT

GROUP BY

```
SELECT Region, COUNT(*) AS Anzahl_Länder
FROM cia
GROUP BY Region
HAVING Anzahl_Länder > 10
```

WHERE bei gruppierten Werten **nicht möglich**

Region	Anzahl_Länder
Afrika	54
Asien	18
Europa	43
Mittelamerika	30
Mittlerer Osten	16
Ozeanien	25
Südamerika	14
Südostasien	15



Übung 4 – Gruppierung

Input

<https://sql-tutorial.de/home/lektionen.php?lektion=4>

Übungen

<https://sql-tutorial.de/home/uebungen.php?lektion=4>

The logo for SQL Tutorial, featuring the letters 'SQL' in a large, red, cursive font, with the word 'Tutorial' in a smaller, black, cursive font below it. The logo is centered within a white rounded rectangle.

SQL
Tutorial





Übung 4 – Gruppierung



```
1
SELECT Region, COUNT(Name)
FROM cia
GROUP BY Region
```

```
2
SELECT Region, COUNT(*)
FROM cia
WHERE Einwohner > 10E06
GROUP BY Region
```

```
3
SELECT Region, SUM(Einwohner)
FROM cia
GROUP BY Region
HAVING SUM(Einwohner) > 1E08
```

```
4
SELECT Region, SUM(Einwohner) AS Einwohnerzahl, SUM(Fläche)
FROM cia
GROUP BY Region
ORDER BY 2 DESC
```

```
5
SELECT Region, SUM(Einwohner) AS Einwohnerzahl , SUM(Fläche)
FROM cia
GROUP BY Region
HAVING Region LIKE '%Amerika%'
ORDER BY 2 DESC
```

```
6
SELECT Fahrerweltmeister, SUM(WM_Punkte)
FROM WMTitel
GROUP BY Fahrerweltmeister
ORDER BY 2 DESC
```

```
7
SELECT KonstrukteursWM, SUM(Team_Punkte)
FROM WMTitel
WHERE Saison BETWEEN "1990" AND "1999"
GROUP BY KonstrukteursWM
```

```
8
SELECT KonstrukteursWM, SUM(Team_Punkte)
FROM WMTitel
WHERE Saison BETWEEN "1990" AND "1999"
GROUP BY KonstrukteursWM
HAVING SUM(Team_Punkte) >= 200
```

```
9
SELECT KonstrukteursWM, AVG(Team_Punkte)
FROM WMTitel
WHERE Fahrerweltmeister = "Michael Schumacher"
GROUP BY KonstrukteursWM
ORDER BY 2 DESC
```

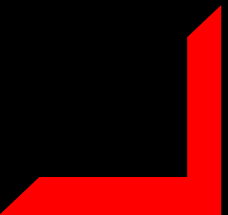
```
10
SELECT KonstrukteursWM, SUM(Team_Punkte)
FROM WMTitel
WHERE Saison >= "1995"
GROUP BY KonstrukteursWM
HAVING SUM(Team_Punkte) >= 200
ORDER BY 2 DESC
```

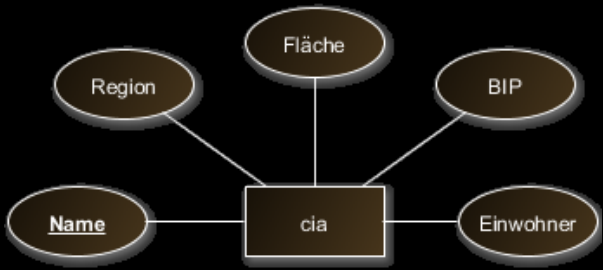


Tagebucheintrag



*group by
having*





SELECT

Unterabfragen

```

SELECT Name
FROM cia
WHERE Einwohner > (
  SELECT Einwohner
  FROM cia
  WHERE Name = "Vereinigte Staaten von Amerika"
);
  
```

Vergleichsoperator
1 Wert

Name
China
Indien

Einwohner
275.562.673

```

SELECT Name
FROM cia
WHERE Region IN (
  SELECT Region
  FROM cia
  WHERE Name LIKE "%vereinigte%"
);
  
```

IN
1 Liste

Region
Mittlerer Osten
Nordamerika

Name
Bahrain
Zypern
Gazastreifen
Iran
Irak
Israel
Jordanien
Kuwait
Libanon
Oman
Qatar
Saudi-Arabien
Syrien
Vereinigte Arabische Emirate
West Bank
Jemen
Bermuda
Kanada
Mexiko
Saint Pierre und Miquelon
Vereinigte Staaten von Amerika



Übung 5 – Unterabfragen

Input

<https://sql-tutorial.de/home/lektionen.php?lektion=5>

Übungen

<https://sql-tutorial.de/home/uebungen.php?lektion=5>





Übung 5 – Unterabfragen



```
1
SELECT Name
FROM cia
WHERE Einwohner > (
  SELECT MAX(Einwohner)
  FROM cia
  WHERE Region = "Südamerika"
);
```

```
2
SELECT Name
FROM cia
WHERE Einwohner > (
  SELECT SUM(Einwohner)
  FROM cia
  WHERE Region = "Europa"
);
```

```
3
SELECT Name
FROM cia
WHERE Region = "Europa" AND BIP/Einwohner > (
  SELECT BIP/Einwohner
  FROM cia
  WHERE Name = "Groß-Britannien"
);
```

```
4
SELECT *
FROM cia
WHERE Region IN (
  SELECT Region
  FROM cia
  WHERE Name IN ("Armenien", "Iran")
);
```

```
5
SELECT Name, Einwohner/Fläche
FROM cia
WHERE Region = "Südostasien" AND Einwohner/Fläche > (
  SELECT AVG(Einwohner/Fläche)
  FROM cia
  WHERE Region = "Südostasien"
);
```

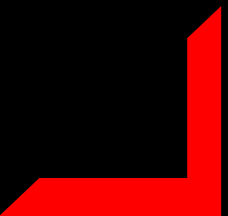
```
6
SELECT Name
FROM cia
WHERE Einwohner > (
  SELECT Einwohner
  FROM cia
  WHERE Name = "Kenia"
)
AND Einwohner < (
  SELECT Einwohner
  FROM cia
  WHERE Name = "Kanada"
);
```



Tagebucheintrag

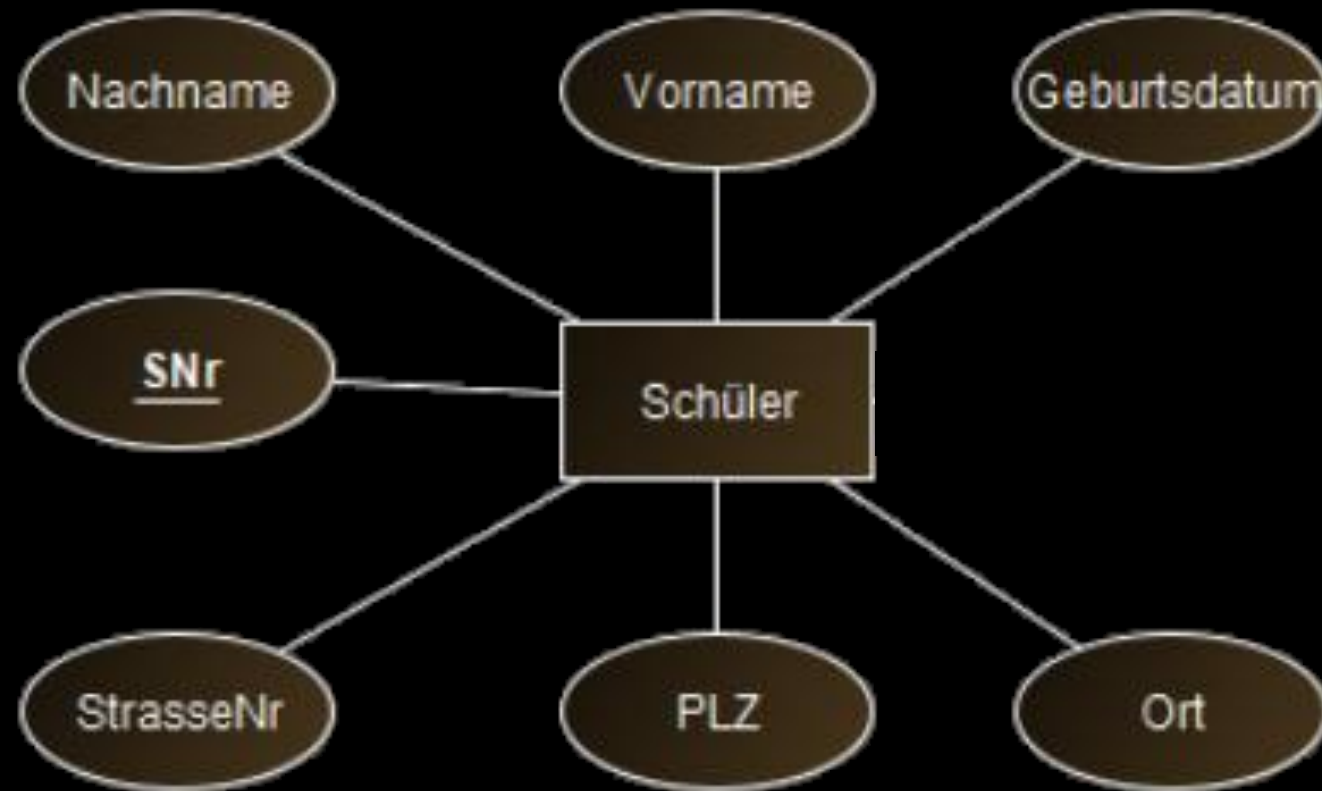
... > < = >= <= <> (select ...)

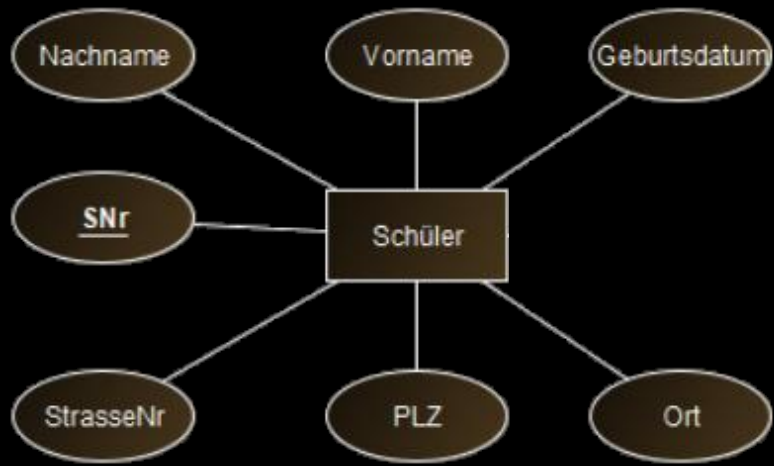
... in (select ...)



DML

Data Manipulation Language





DML

CREATE

INT(n)

Ganzzahl mit max. n Ziffern

VARCHAR(n)

Text mit max. n Zeichen

DATE

Datum der Form YYYY-MM-DD

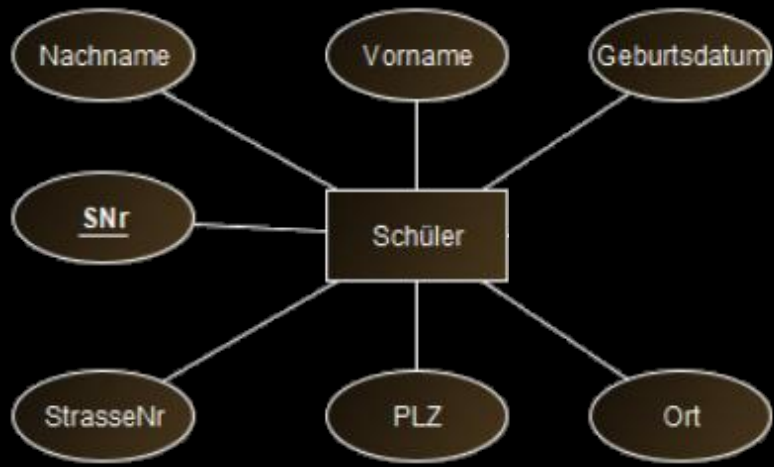
TIME

Uhrzeit der Form HH:MM:SS

falls Primärschlüssel aus mehreren Attributen besteht (z. B. bei n-m-Beziehung):

primary key (Schlüssel1, Schlüssel2)

```
CREATE TABLE SchueLER (
    SNr INT(5) PRIMARY KEY,
    Nachname VARCHAR(40),
    Vorname VARCHAR(30),
    Geburtsdatum DATE,
    StrasseNr VARCHAR(40),
    PLZ VARCHAR(7),
    Ort VARCHAR(40)
);
```



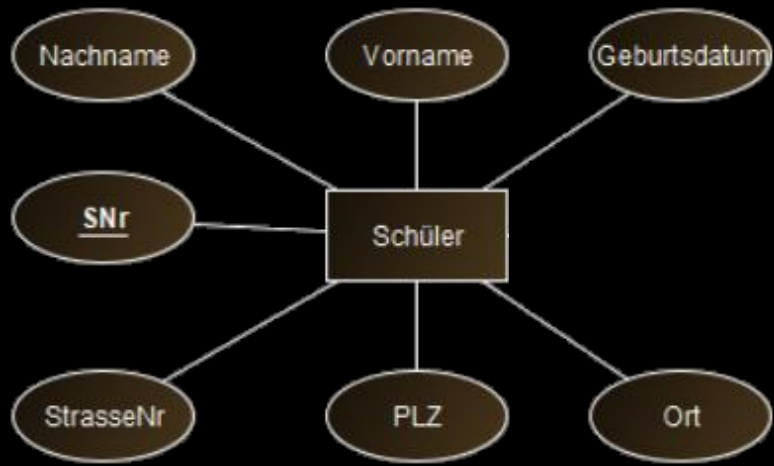
DML

DROP



```
DROP TABLE Schue1er;  
DROP TABLE IF EXISTS Schue1er;
```

= „Lösche es nur, wenn es da ist – sonst mache nichts.“



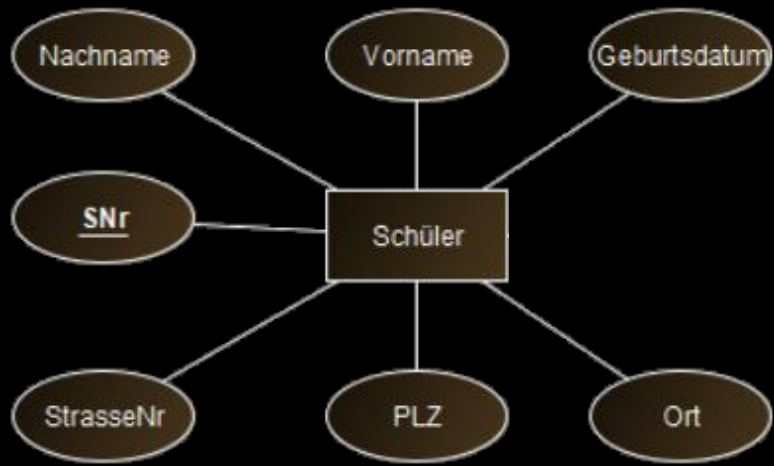
DML

INSERT

nicht notwendig wenn alle Attribute in der richtigen Reihenfolge angegeben werden (insb. bei AUTO_INCREMENT zwingend)

```
INSERT INTO Schueler (SNr, Nachname, Vorname, Geburtsdatum, StrasseNr, PLZ, Ort) VALUES
(1, 'Mueller', 'Heinz', '1988-08-14', 'Hüttenweg 6', '64536', 'Oberdorf'),
(2, 'Schmidt', 'Laura', '1992-03-21', 'Bergstraße 12', '70173', 'Stuttgart'),
(3, 'Weber', 'Daniel', '1985-11-09', 'Hauptstraße 45', '50667', 'Köln'),
(4, 'Fischer', 'Anna', '1990-06-17', 'Gartenweg 8', '80331', 'München'),
(5, 'Meyer', 'Tobias', '1983-01-30', 'Schulstraße 27', '28195', 'Bremen'),
(6, 'Wagner', 'Sophie', '1995-09-12', 'Bahnhofstraße 3', '01067', 'Dresden');
```

immer ein Komma, nur am Ende ein Semikolon



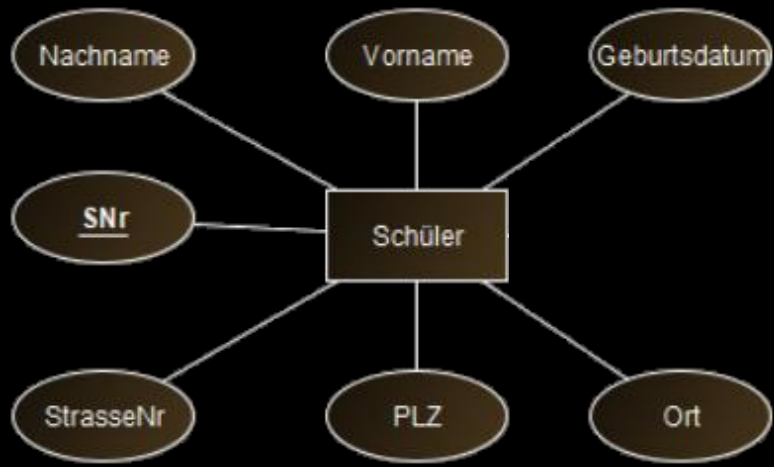
DML

UPDATE

auch mehrere Änderungen mit Komma getrennt möglich



```
UPDATE SchueLER SET Nachname = 'Müller' WHERE SNr = 1;
```



DML

DELETE

spezifisch sein, sonst werden ALLE Zeilen mit entsprechender Bedingung gelöscht



```
DELETE FROM Schue1er WHERE SNr = 1024;
```



Übung 6 – DML

Input

<https://sql-tutorial.de/home/lektionen.php?lektion=6>

Übungen

<https://sql-tutorial.de/home/uebungen.php?lektion=6>





Übung 6 – DML



```
1  
DELETE FROM Schueler;
```

```
2  
INSERT INTO Schueler  
VALUES(1024, "Mueller", "Heinz", "1988-08-14", "Hüttenweg 6", "64536", "Oberdorf");
```

```
3  
SELECT * FROM Kurs;  
  
INSERT INTO Belegt  
VALUES (1024, "InfoQ2Abe", NULL);  
  
INSERT INTO Kurs  
VALUES ("InfoQ2Abe", "Info", "Datenbanken", "GK", "2527/2", 2);
```

```
4  
UPDATE Kurs  
SET Stunden = 3  
WHERE Stunden = 2
```

```
5  
UPDATE Kurs  
SET Stunden = Stunden - 1  
WHERE Fach = "D"
```

```
6  
DELETE FROM Belegt  
WHERE Punkte < 5 AND SNr = (  
    SELECT SNr  
    FROM Schueler  
    WHERE Vorname = "Heinz" AND Nachname = "Müller"  
);
```



Tagebucheintrag



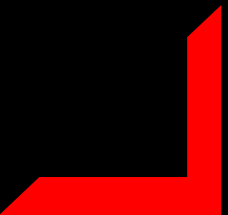
Insert

Update

Delete

Create

Drop





Wochenübung



SQL spielerisch lernen



Übungen LuoSQL



Quiz W3Schools



Exercises W3Schools

Cheatsheet Online

<https://devhints.io/mysql>

<https://gist.github.com/bradtraversy/c831baaad44343cc945e76c2e30927b3>

<https://www.mysqltutorial.org/mysql-cheat-sheet/>

<https://learnsql.com/blog/mysql-cheat-sheet/>

