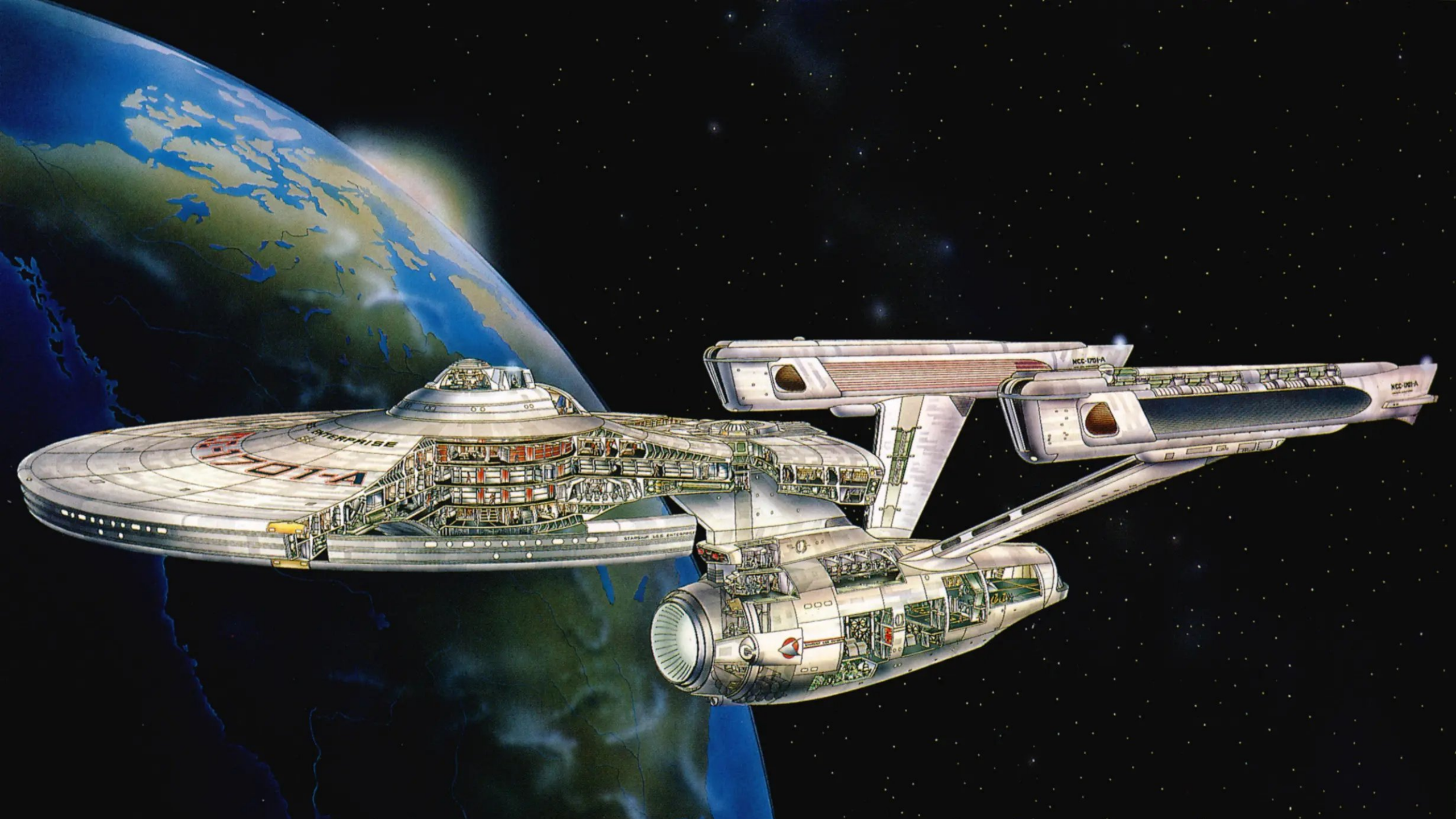
HTL anichstraße

Biomedizintechnik · Elektronik · Elektrotechnik · Maschinenbau · Wirtschaftsingenieure

SQL-ABFRAGEN TEIL 2

AND / OR / NOT --- XOR --- ORDER BY --- COUNT --- MIN --- SUM --- GROUP BY --- JOINS --- INNER JOINS


ENTERPRISE
NCC-1701-A
NCC-1701-A

VERKNÜPFUNGEN AND/OR/NOT

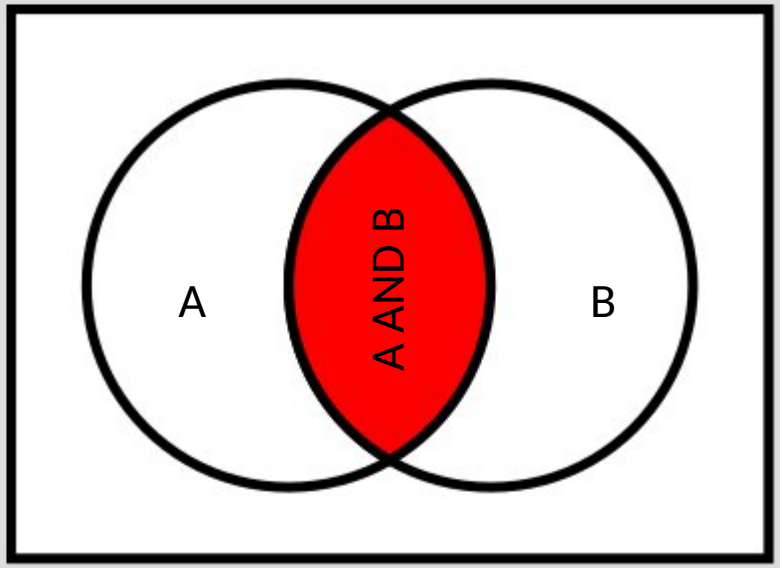
- Wie kann ich **Bedingungen** miteinander **verbinden**?
  - AND / OR / NOT
- **Wahrheitstabellen** bzw. Venn-Diagramm zeigen uns die Logik an

VERKNÜPFUNGEN AND/OR/NOT

- Wahrheitstabellen **AND** (auch: Konjunktion)

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Venn-Diagramm



VERKNÜPFUNGEN AND/OR/NOT

- Beispiel Konjunktion: Wann gehen wir ins Schwimmbad?
  - Nur wenn es warm ist **und** die Sonne scheint!

Es ist warm	Die Sonne scheint	1 oder 0?
0	0	
0	1	
1	0	
1	1	

VERKNÜPFUNGEN AND/OR/NOT

- Beispiel Konjunktion: Wann gehen wir ins Schwimmbad?
  - Nur wenn es warm ist **und** die Sonne scheint!

Es ist warm	Die Sonne scheint	1 oder 0?
0	0	0
0	1	0
1	0	0
1	1	1

VERKNÜPFUNGEN AND/OR/NOT

- Wie bekomme ich alle aktiven Raumschiffe, die einen Schaden > 50 haben?

```
mysql> SELECT raumschiffname, schaden, aktiv FROM raumschiff WHERE schaden>50
    -> AND
    -> aktiv=1;
+----------------+---------+-------+
| raumschiffname | schaden | aktiv |
+----------------+---------+-------+
| Enterprise     |      55 |     1 |
| Voyager        |      80 |     1 |
| Weisser Stern  |      90 |     1 |
+----------------+---------+-------+
```

ÜBUNGEN

- Zeige alle **kampfstarken Raumschifftypen** mit hoher Geschwindigkeit. Sie müssen mindestens **100 Geschütze** und eine **Schildstärke von 1000** haben. Außerdem müssen sie mindestens die **Geschwindigkeit 1000** aufbringen können.
- Wir möchten alle **Planeten** sehen, deren **Bevölkerung zwischen 10000 und 100000** liegt.
    - Einmal mit BETWEEN, einmal mit AND

VERKNÜPFUNGEN AND/OR/NOT

- Beispiel Disjunktion: Kuchen oder Eis?
  - Du darfst Kuchen **oder** Eis essen (oder beides)!

Ich nehme Kuchen	Ich nehme Eis	1 oder 0?
0	0	
0	1	
1	0	
1	1	

VERKNÜPFUNGEN AND/OR/NOT

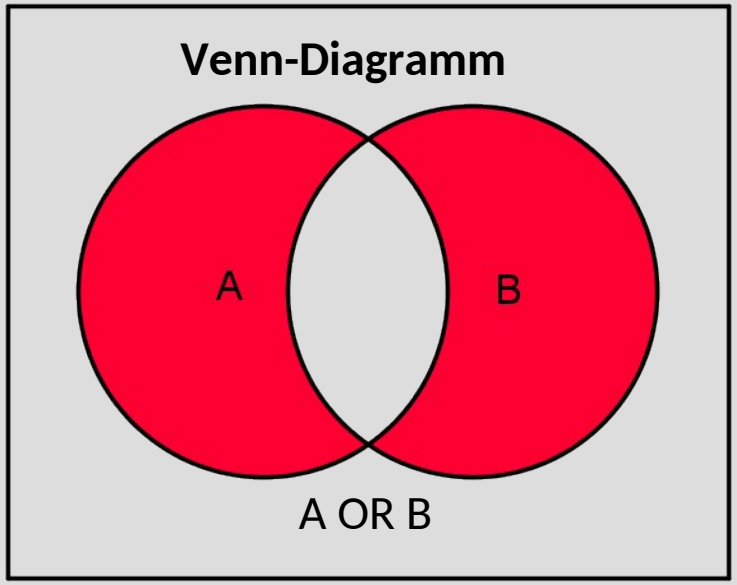
- Beispiel Disjunktion: Kuchen oder Eis?
  - Du darfst Kuchen **oder** Eis essen (oder beides)!

Ich nehme Kuchen	Ich nehme Eis	1 oder 0?
0	0	0
0	1	1
1	0	1
1	1	1

VERKNÜPFUNGEN AND/OR/NOT

- Wahrheitstabellen **OR** (auch: Disjunktion)

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



VERKNÜPFUNGEN AND/OR/NOT

- Wie finde ich alle Raumschiffe, deren Besatzung aus mindestens 100 Personen besteht oder deren Lagerraum mindestens 1000 Einheiten umfasst?

- Mit „**OR** “:

Disjunktion

```
mysql> SELECT * FROM raumschifftyp WHERE besatzung >=100
    -> OR
    -> lagerraum >=1000;
+-----------------+-----------------+-----------+---------+-------------------+---------------+-----------+-----------------+
| raumschifftypid | bezeichnung     | besatzung | groesse | anzahlgeschuetze  | schildstaerke | lagerraum | geschwindigkeit |
+-----------------+-----------------+-----------+---------+-------------------+---------------+-----------+-----------------+
|               1 | Fregatte        |      1000 |     500 |               150 |       1567.12 |        50 |            1123 |
|               2 | Forschungsschiff |       100 |     200 |                20 |         987.4 |       100 |            2000 |
|               3 | Raumstation     |     10000 |    7000 |              1234 |       2345.67 |      9878 |               0 |
|               4 | Transportschiff |       200 |    3000 |               120 |        800.67 |      9000 |             400 |
+-----------------+-----------------+-----------+---------+-------------------+---------------+-----------+-----------------+
```

ÜBUNGEN

- Zeige die **Bezeichnung** und **Besatzung** aller Raumschifftypen, wenn sie größer als 300 sind oder mehr als 100 Geschütze haben;
- Zeige alle **Planeten**, deren **Bevölkerung** kleiner als 500 und größer als 10000 ist.
- Zeige alle **Raumschiffe**, die **gar keinen Schaden** haben oder von einem **Kapitän** mit der **ID 1 oder 3** geflogen werden;

ORDER BY

- Nach einer bestimmten Spalte aufsteigend/absteigend sortieren
- Mit „**ORDER BY** “:
  - Optional mit DESC (absteigend, ***descending***)

```
MariaDB [spaceships]> SELECT * FROM raumschiff ORDER BY raumschiffname;
+-------------+-----------------------------+---------+-------+--------------------+--------------+-------------+
| raumschiffId | raumschiffname              | schaden | aktiv | raumschifftypid_fk | kapitaenId_fk | flotteId_fk |
+-------------+-----------------------------+---------+-------+--------------------+--------------+-------------+
|           5 | Babylon 5                   |      50 |     1 |                  3 |           11 |        NULL |
|           3 | DeepSpace 9                 |      20 |     1 |                  3 |           10 |        NULL |
|           9 | Defiant                     |       0 |     1 |                  2 |            5 |           6 |
|           1 | Enterprise                  |      55 |     1 |                  1 |            1 |           1 |
|           8 | Millenium Falke             |       0 |     1 |                  2 |            3 |        NULL |
|           4 | Sehnsucht nach Unendlichkeit |       0 |     1 |                  4 |            4 |           3 |
|           7 | Serenity                    |       0 |     1 |                  4 |            6 |           3 |
|           2 | Voyager                     |      80 |     1 |                  2 |            5 |           2 |
|           6 | Weisser Stern               |      90 |     1 |                  1 |            9 |           1 |
+-------------+-----------------------------+---------+-------+--------------------+--------------+-------------+
9 rows in set (0.001 sec)
```

ORDER BY

- Absteigend mit DESC

```
MariaDB [spaceships]> SELECT * FROM raumschiff ORDER BY raumschiffname DESC;
+--------------+-----------------------------+---------+-------+--------------------+---------------+-------------+
| raumschiffId | raumschiffname              | schaden | aktiv | raumschifftypid_fk | kapitaenId_fk | flotteId_fk |
+--------------+-----------------------------+---------+-------+--------------------+---------------+-------------+
|            6 | Weisser Stern               |      90 |     1 |                  1 |             9 |           1 |
|            2 | Voyager                     |      80 |     1 |                  2 |             5 |           2 |
|            7 | Serenity                    |       0 |     1 |                  4 |             6 |           3 |
|            4 | Sehnsucht nach Unendlichkeit|       0 |     1 |                  4 |             4 |           3 |
|            8 | Millenium Falke             |       0 |     1 |                  2 |             3 |        NULL |
|            1 | Enterprise                  |      55 |     1 |                  1 |             1 |           1 |
|            9 | Defiant                     |       0 |     1 |                  2 |             5 |           6 |
|            3 | DeepSpace 9                 |      20 |     1 |                  3 |            10 |        NULL |
|            5 | Babylon 5                   |      50 |     1 |                  3 |            11 |        NULL |
+--------------+-----------------------------+---------+-------+--------------------+---------------+-------------+
9 rows in set (0.001 sec)
```

AGGREGATFUNKTIONEN

- **Bilden Zusammenfassungen von einzelnen Tabellen**
- Werden auf eine Gruppe von Zeilen in einer Tabelle angewendet
  - COUNT(*) zählt die Anzahl der Zeilen
  - SUM(x) berechnet die Summe der Spalte x
  - AVG(x) berechnet den Durchschnitt der Spalte x
  - analog: MIN(x) MAX(x)

ÜBUNGEN AGGREGATFUNKTIONEN

```
MariaDB [spaceships]> SELECT COUNT(*) FROM Planet;
+----------+
| COUNT(*) |
+----------+
|       11 |
+----------+
1 row in set (0.000 sec)
```

```
MariaDB [spaceships]> SELECT SUM(galaxie) FROM Planet;
+--------------+
| SUM(galaxie) |
+--------------+
|           49 |
+--------------+
1 row in set (0.000 sec)
```

```
MariaDB [spaceships]> SELECT planetname, MAX(galaxie) FROM Planet;
+------------+--------------+
| planetname | MAX(galaxie) |
+------------+--------------+
| Erde       |           12 |
+------------+--------------+
1 row in set (0.000 sec)
```

ÜBUNGEN AGGREGATFUNKTIONEN

```
MariaDB [spaceships]> SELECT max(bevoelkerung) FROM Planet WHERE galaxie = 1;
+--------------------+
| max(bevoelkerung) |
+--------------------+
|            7000000 |
+--------------------+
1 row in set (0.000 sec)
```

GROUP BY

- Gruppiere Daten nach Kriterien
- z.B. gruppiere Daten nach Galaxienummer, innerhalb der Gruppe soll der Planet mit der größten Bevölkerung angezeigt werden

```
MariaDB [spaceships]> SELECT * FROM Planet;
+----------+----------------+-------------+---------+-------------+------------------+------------+--------------+--------------------+
| planetid | planetname     | bevoelkerung | galaxie | sonnensystem | planetenposition | metallLager | kristallLager | dunkleMaterieLager |
+----------+----------------+-------------+---------+-------------+------------------+------------+--------------+--------------------+
|        1 | Erde           |      7000000 |       1 |           1 |                3 |          0 |            0 |                  0 |
|        2 | Mars           |         7000 |       1 |           1 |                4 |          0 |            0 |                  0 |
|        3 | Vulkan         |       123456 |       3 |           4 |                4 |          0 |            0 |                  0 |
|        4 | Romulus        |        12236 |       6 |           2 |                2 |          0 |            0 |                  0 |
|        5 | Sirius         |         1226 |       3 |           4 |                2 |          0 |            0 |                  0 |
|        6 | Beteigeuze     |           12 |       4 |           7 |                2 |          0 |            0 |                  0 |
|        7 | Centauri Prime |      8123876 |      12 |           6 |                2 |          0 |            0 |                  0 |
|        8 | Narn           |        47851 |      12 |           9 |                2 |          0 |            0 |                  0 |
|        9 | Bajor          |      4741851 |       1 |           9 |                5 |          0 |            0 |                  0 |
|       10 | Caprica        |      1234332 |       4 |           5 |                6 |          0 |            0 |                  0 |
|       11 | Tatooine       |         1432 |       2 |           1 |                1 |          0 |            0 |                  0 |
+----------+----------------+-------------+---------+-------------+------------------+------------+--------------+--------------------+
11 rows in set (0.000 sec)

MariaDB [spaceships]> SELECT planetname, MAX(bevoelkerung) FROM PLANET GROUP BY galaxie;
+----------------+-------------------+
| planetname     | MAX(bevoelkerung) |
+----------------+-------------------+
| Erde           |           7000000 |
| Tatooine       |              1432 |
| Vulkan         |            123456 |
| Beteigeuze     |           1234332 |
| Romulus        |             12236 |
| Centauri Prime |           8123876 |
+----------------+-------------------+
6 rows in set (0.001 sec)
```

max. Bevölkerung in galaxie 1

max. Bevölkerung in galaxie 12

LIMIT

- Beschränke die Abfrage auf bestimmte Datensätze
- LIMIT x,y
  - x gibt den Offset an
  - y gibt die Anzahl der Datensätze an
- **Achtung:** LIMIT 3,2 gibt 2 Datensätze aus, beginnend mit dem Datensatz mit **Index 4** (0,1,2,3,4)

Offset: 3 (Start mit 4. Datensatz)

Anzahl: 2 Datensätze

```
MariaDB [spaceships]> SELECT * FROM raumschiff LIMIT 3,2;
+--------------+-------------------------------+---------+-------+--------------------+--------------+-------------+
| raumschiffId | raumschiffname                | schaden | aktiv | raumschifftypid_fk | kapitaenId_fk | flotteId_fk |
+--------------+-------------------------------+---------+-------+--------------------+--------------+-------------+
|            4 | Sehnsucht nach Unendlichkeit |       0 |     1 |                  4 |             4 |           3 |
|            5 | Babylon 5                     |      50 |     1 |                  3 |            11 |        NULL |
+--------------+-------------------------------+---------+-------+--------------------+--------------+-------------+
2 rows in set (0.000 sec)
```

ÜBUNGEN

- Zeigen Sie alle Gebäude an, deren Name mit K beginnt.
- Zeigen Sie alle Kapitäne an, absteigend alphabetisch sortiert nach ihrem Namen.
- Zeigen Sie den Namen, die Galaxie, das Sonnensystem und die Position aller Planeten an, die in den Galaxien in den Galaxien 2,3,7 oder 9 liegen.
- Welche drei Kapitäne haben das höchste Gehalt?
- Welche Durchschnittsgeschwindigkeit haben unsere Raumschifftypen?
- Geben Sie das Durchschnittsgehalt unserer Kapitäne aus und zwar gruppiert nach Heimatplaneten

JOINS

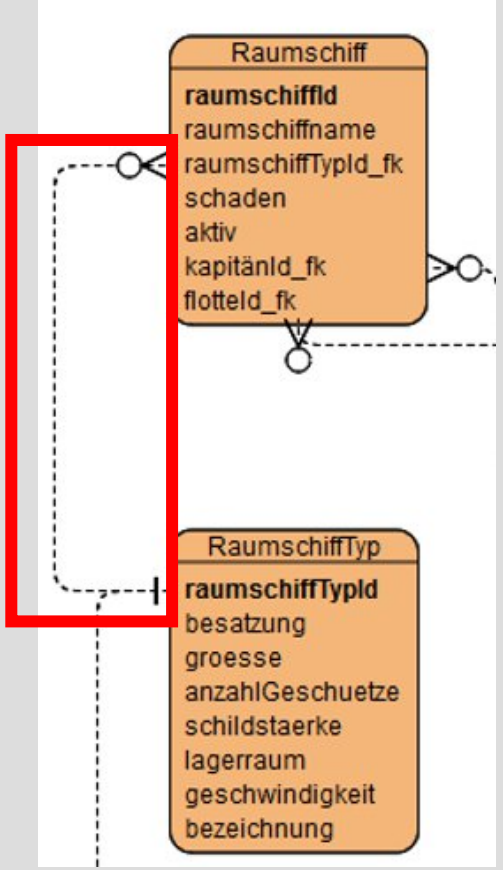
- JOINS sind Abfragen über mehrere Tabellen
- **Hintergrund:** In „guten“ Datenbanken sind Daten auf viele verschiedene Tabellen aufgeteilt (vgl. Normalformen)
- Unterschiedliche Arten:
  - (Cross Join)
  - INNER JOIN
  - LEFT JOIN (LEFT OUTER JOIN)
  - RIGHT JOIN (RIGHT OUTER JOIN)
  - FULL JOIN (FULL OUTER JOIN)

JOINS

- 1:n Beziehung **raumschiff** - **raumschifftyp**

```
MariaDB [spaceships]> SELECT * FROM raumschiff;
+--------------+-------------------------------+---------+-------+---------------------+---------------+-------------+
| raumschiffId | raumschiffname                | schaden | aktiv | raumschifftypid_fk  | kapitaenId_fk | flotteId_fk |
+--------------+-------------------------------+---------+-------+---------------------+---------------+-------------+
|            1 | Enterprise                    |      55 |     1 |                   1 |             1 |           1 |
|            2 | Voyager                       |      80 |     1 |                   2 |             5 |           2 |
|            3 | DeepSpace 9                   |      20 |     1 |                   3 |            10 |        NULL |
|            4 | Sehnsucht nach Unendlichkeit  |       0 |     1 |                   4 |             4 |           3 |
|            5 | Babylon 5                     |      50 |     1 |                   3 |            11 |        NULL |
|            6 | Weisser Stern                 |      90 |     1 |                   1 |             9 |           1 |
|            7 | Serenity                      |       0 |     1 |                   4 |             6 |           3 |
|            8 | Millenium Falke               |       0 |     1 |                   2 |             3 |        NULL |
|            9 | Defiant                       |       0 |     1 |                   2 |             5 |           6 |
+--------------+-------------------------------+---------+-------+---------------------+---------------+-------------+
9 rows in set (0.000 sec)

MariaDB [spaceships]> SELECT * FROM raumschifftyp;
+-----------------+-----------------+-----------+---------+-------------------+--------------+-----------+-----------------+
| raumschifftypid | bezeichnung     | besatzung | groesse | anzahlgeschuetze  | schildstaerke | lagerraum | geschwindigkeit |
+-----------------+-----------------+-----------+---------+-------------------+--------------+-----------+-----------------+
|               1 | Fregatte        |      1000 |     500 |               150 |      1567.12 |        50 |            1123 |
|               2 | Forschungsschiff |       100 |     200 |                20 |        987.4 |       100 |            2000 |
|               3 | Raumstation     |     10000 |    7000 |              1234 |      2345.67 |      9878 |               0 |
|               4 | Transportschiff |       200 |    3000 |               120 |       800.67 |      9000 |             400 |
|               5 | Spionageschiff  |        10 |      10 |                 5 |           30 |         0 |            3000 |
|               6 | Passagierschiff |        10 |     500 |                10 |           30 |         0 |            1600 |
+-----------------+-----------------+-----------+---------+-------------------+--------------+-----------+-----------------+
```



CROSS JOIN

- Zeigt das **kartesische (=kreuz) Produkt** der Zeilen an
- Kartesisches Produkt A x B für A = x,y,z und B = 1,2,3 ist
  - AxB=(x,1), (x,2),(x,3),(y,1),(y,2,),(y,3),(z,1),(z,2),(z,3)
- SQL Befehl für Cross Join **raumschiff** und **raumschifftyp**

```
MariaDB [spaceships]> SELECT * FROM raumschiff CROSS JOIN raumschifftyp;
```

  - **Problem:** Sehr viele Datensätze – wird kaum angewendet

```
MariaDB [spaceships]> SELECT COUNT(*) FROM raumschiff CROSS JOIN raumschifftyp;
+----------+
| COUNT(*) |
+----------+
|       54 |
+----------+
1 row in set (0.001 sec)
```

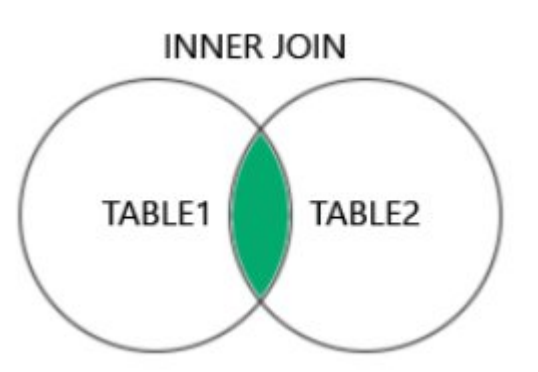
Quelle: https://www.w3schools.com/sql/sql_join.asp [2.1.2024]

ARTEN VON JOINS

- **INNER JOIN:** Zeigt Datensätze mit übereinstimmenden Werten in beiden Tabellen.
- **LEFT JOIN (LEFT OUTER JOIN):** Zeigt alle Datensätze aus der linken Tabelle und die übereinstimmenden Datensätze aus der rechten Tabelle.
- **RIGHT JOIN (RIGHT OUTER JOIN):** Zeigt alle Datensätze aus der rechten Tabelle und die übereinstimmenden Datensätze aus der linken Tabelle.
- **FULL JOIN (FULL OUTER JOIN):** Zeigt alle Datensätze, wenn es eine Übereinstimmung in entweder der linken oder der rechten Tabelle gibt.

INNER JOIN

- Zeigt Datensätze mit **übereinstimmenden Werten in beiden Tabellen**.
- Entspricht dem „gefilterten CROSS JOIN“



RaumschiffName	RaumschiffTypId_fk
Enterprise	1
Voyager	2
Serenity	3
Weißer Stern	3

RaumschiffTypId	Bezeichnung
1	Fregatte
2	Transportschiff
3	Spionageschiff

Quelle: https://www.w3schools.com/sql/sql_join.asp [2.1.2024]

INNER JOIN: VARIANTE A

- Zeige nur die Datensätze, die im **Primärschlüssel** und im **Fremdschlüssel gleiche Werte** haben
  - Spalte der Tabelle muss angegeben werden (Alias möglich)

```
MariaDB [spaceships]> SELECT r.raumschiffname, t.bezeichnung
    -> FROM raumschiff AS r
    -> INNER JOIN raumschifftyp AS t ON r.raumschifftypid_fk = t.raumschifftypid;
+-----------------------------+------------------+
| raumschiffname              | bezeichnung      |
+-----------------------------+------------------+
| Enterprise                  | Fregatte         |
| Voyager                     | Forschungsschiff |
| DeepSpace 9                 | Raumstation      |
| Sehnsucht nach Unendlichkeit | Transportschiff  |
| Babylon 5                   | Raumstation      |
| Weisser Stern               | Fregatte         |
| Serenity                    | Transportschiff  |
| Millenium Falke             | Forschungsschiff |
| Defiant                     | Forschungsschiff |
+-----------------------------+------------------+
9 rows in set (0.001 sec)
```

OHNE „ABKÜRZUNG“

- ohne as t bzw. as r

```
MariaDB [spaceships]> select raumschiff.raumschiffname, raumschifftyp.bezeichnung from raumschiff inner join raumschifftyp on raumschiff.raumschifftypid_fk=raumschifftyp.raumschifftypid;
+-----------------------------+------------------+
| raumschiffname              | bezeichnung      |
+-----------------------------+------------------+
| Enterprise                  | Fregatte         |
| Voyager                     | Forschungsschiff |
| DeepSpace 9                 | Raumstation      |
| Sehnsucht nach Unendlichkeit | Transportschiff  |
| Babylon 5                   | Raumstation      |
| Weisser Stern               | Fregatte         |
| Serenity                    | Transportschiff  |
| Millenium Falke             | Forschungsschiff |
| Defiant                     | Forschungsschiff |
+-----------------------------+------------------+
9 rows in set (0.000 sec)
```

INNER JOIN: VARIANTE B

- Gleicher Output mit „Where“
- Zeige nur die Datensätze, die im **Primärschlüssel** und im **Fremdschlüssel gleiche Werte** haben:

```
MariaDB [spaceships]> SELECT raumschiffname, bezeichnung FROM raumschiff, raumschifftyp
    -> WHERE
    -> raumschifftypID = raumschifftypID_fk;
+------------------------------+------------------+
| raumschiffname               | bezeichnung      |
+------------------------------+------------------+
| Enterprise                   | Fregatte         |
| Voyager                      | Forschungsschiff |
| DeepSpace 9                  | Raumstation      |
| Sehnsucht nach Unendlichkeit | Transportschiff  |
| Babylon 5                    | Raumstation      |
| Weisser Stern                | Fregatte         |
| Serenity                     | Transportschiff  |
| Millenium Falke              | Forschungsschiff |
| Defiant                      | Forschungsschiff |
+------------------------------+------------------+
9 rows in set (0.001 sec)
```

ÜBUNG

- Wir möchten wissen, welche Raumschiffe sind von einem Raumschifftyp, der eine Geschwindigkeit > 1000 hat.
- Vorüberlegungen:

```
MariaDB [spaceships]> select * from raumschifftyp;
+-----------------+------------------+-----------+---------+------------------+---------------+-----------+-----------------+
| raumschifftypid | bezeichnung      | besatzung | groesse | anzahlgeschuetze | schildstaerke | lagerraum | geschwindigkeit |
+-----------------+------------------+-----------+---------+------------------+---------------+-----------+-----------------+
|               1 | Fregatte         |      1000 |     500 |              150 |       1567.12 |        50 |            1123 |
|               2 | Forschungsschiff |       100 |     200 |               20 |         987.4 |       100 |            2000 |
|               3 | Raumstation      |     10000 |    7000 |             1234 |       2345.67 |      9878 |               0 |
|               4 | Transportschiff  |       200 |    3000 |              120 |        800.67 |      9000 |             400 |
|               5 | Spionageschiff   |        10 |      10 |                5 |            30 |         0 |            3000 |
|               6 | Passagierschiff  |        10 |     500 |               10 |            30 |         0 |            1600 |
+-----------------+------------------+-----------+---------+------------------+---------------+-----------+-----------------+
```

```
MariaDB [spaceships]> select bezeichnung from raumschifftyp where geschwindigkeit > 1000;
+------------------+
| bezeichnung      |
+------------------+
| Fregatte         |
| Forschungsschiff |
| Spionageschiff   |
| Passagierschiff  |
+------------------+
4 rows in set (0.002 sec)
```

LÖSUNG

```
MariaDB [spaceships]> SELECT raumschiffname, bezeichnung, geschwindigkeit FROM raumschifftyp rt, raumschiff r WHERE rt.raumschifftypId = r. raumschifftypId_fk AND geschwindigkeit > 1000;
+-----------------+------------------+-----------------+
| raumschiffname  | bezeichnung      | geschwindigkeit |
+-----------------+------------------+-----------------+
| Enterprise      | Fregatte         |            1123 |
| Voyager         | Forschungsschiff |            2000 |
| Weisser Stern   | Fregatte         |            1123 |
| Millenium Falke | Forschungsschiff |            2000 |
| Defiant         | Forschungsschiff |            2000 |
+-----------------+------------------+-----------------+
5 rows in set (0.001 sec)
```

ÜBUNG

- Wir möchten alle **Auftragsnamen** mit der **Bezeichnung des Auftragstyps** sehen.

```
MariaDB [spaceships]> select * from auftrag;
+-----------+-----------------------------------------+-------------------------------------------------+----------------+-----------------+-----------------+-----------------+-------------+
| auftragId | auftragName                             | beschreibung                                    | erteilungsdatum | beendigungsdatum | folgeauftragId_fk | auftragTypId_fk | flotteId_fk |
+-----------+-----------------------------------------+-------------------------------------------------+----------------+-----------------+-----------------+-----------------+-------------+
|         1 | Sektor X erkunden                       |                                                 | 2301-04-04     | NULL            |            NULL |               1 |           1 |
|         2 | Raumpiraten jagen                       |                                                 | 2301-05-06     | NULL            |            NULL |               2 |           2 |
|         3 | Dunkle Materie nach Caprica bringen     |                                                 | 2301-05-03     | 2301-06-01      |               4 |               3 |           3 |
|         4 | Personen zur Erde bringen               |                                                 | 2301-06-02     | NULL            |            NULL |               3 |           3 |
|         5 | Grenze überwachen                       |                                                 | 2300-09-09     | 2301-01-04      |            NULL |               4 |           4 |
|         6 | Invasion verhindern                     |                                                 | 2299-09-09     | 2300-01-04      |               7 |               2 |           5 |
|         7 | Territorium zurück erobern              | Holen Sie sich das verlorene Gebiet zurück      | 2300-01-05     | 2300-03-04      |               8 |               2 |           2 |
|         8 | Soldaten ins Kampfgebiet bringen        |                                                 | 2300-03-05     | 2300-04-04      |               9 |               3 |           2 |
|         9 | Rohstoffe zum Wiederaufbau transportieren | Vieles wurde zerstört. Bauen wir es wieder auf. | 2300-05-05     | 2300-06-04      |            NULL |               3 |           2 |
+-----------+-----------------------------------------+-------------------------------------------------+----------------+-----------------+-----------------+-----------------+-------------+
9 rows in set (0.001 sec)
```

```
MariaDB [spaceships]> select * from auftragtyp;
+--------------+-------------+--------------------------------+
| auftragtypId | bezeichnung | beschreibung                   |
+--------------+-------------+--------------------------------+
|            1 | Forschung   |                                |
|            2 | Kampf       | Bösen Aliens Arschtritte geben |
|            3 | Transport   | Rohstoffe von A nach B bringen |
|            4 | Patrouille  |                                |
+--------------+-------------+--------------------------------+
4 rows in set (0.000 sec)
```

LÖSUNG

```
MariaDB [spaceships]> SELECT a.auftragName, at.bezeichnung FROM auftrag as a
INNER JOIN auftragtyp as at ON at.auftragtypId = a.auftragTypId_fk;
+-----------------------------------------+-------------+
| auftragName                             | bezeichnung |
+-----------------------------------------+-------------+
| Sektor X erkunden                       | Forschung   |
| Raumpiraten jagen                       | Kampf       |
| Dunkle Materie nach Caprica bringen     | Transport   |
| Personen zur Erde bringen               | Transport   |
| Grenze überwachen                       | Patrouille  |
| Invasion verhindern                     | Kampf       |
| Territorium zurück erobern              | Kampf       |
| Soldaten ins Kampfgebiet bringen        | Transport   |
| Rohstoffe zum Wiederaufbau transportieren | Transport   |
+-----------------------------------------+-------------+
9 rows in set (0.001 sec)
```

ARTEN VON JOINS

- **INNER JOIN:** Zeigt Datensätze mit übereinstimmenden Werten in beiden Tabellen.
- **LEFT JOIN (LEFT OUTER JOIN):** Zeigt alle Datensätze aus der linken Tabelle und die übereinstimmenden Datensätze aus der rechten Tabelle.
- **RIGHT JOIN (RIGHT OUTER JOIN):** Zeigt alle Datensätze aus der rechten Tabelle und die übereinstimmenden Datensätze aus der linken Tabelle.
- **FULL JOIN (FULL OUTER JOIN):** Zeigt alle Datensätze, wenn es eine Übereinstimmung in entweder der linken oder der rechten Tabelle gibt.

OUTER JOIN

- Was würde bei einem Inner Join mit „Raumstation“ passieren?
  - Keine Beziehung = keine Ausgabe

RaumschiffName	RaumschiffTypId_fk
Enterprise	1
Voyager	2
Serenity	3
Weißer Stern	3

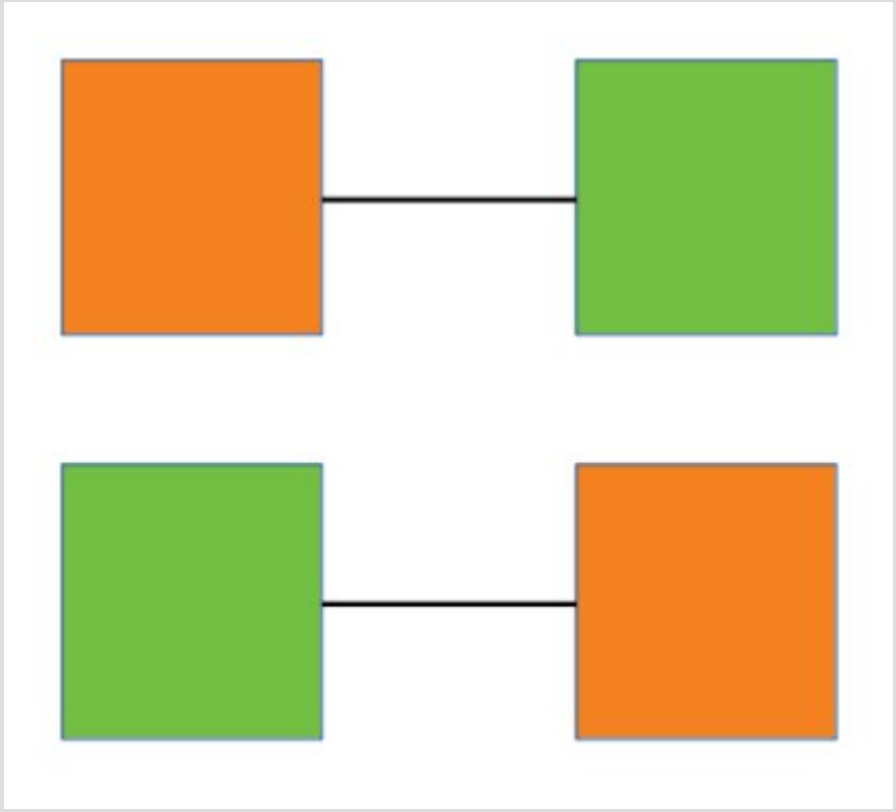
RaumschiffTypId	Bezeichnung
1	Fregatte
2	Transportschiff
3	Spionageschiff
4	Raumstation

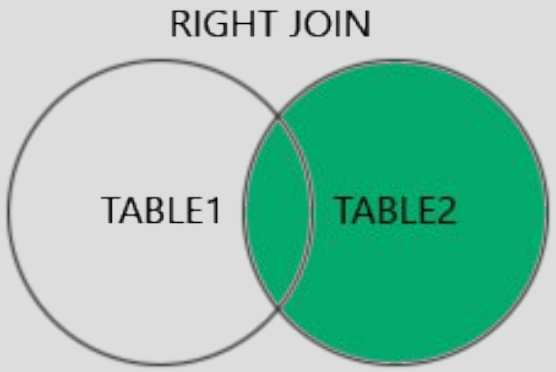
Tabelle bei einem Join

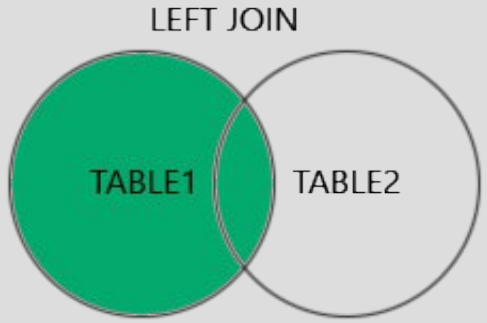
- Deshalb: Outer Join

OUTER JOIN

- Left / Right Outer Join
  - Bestimmt, bei welcher Tabelle alle Datensätze im Ergebnis angezeigt werden



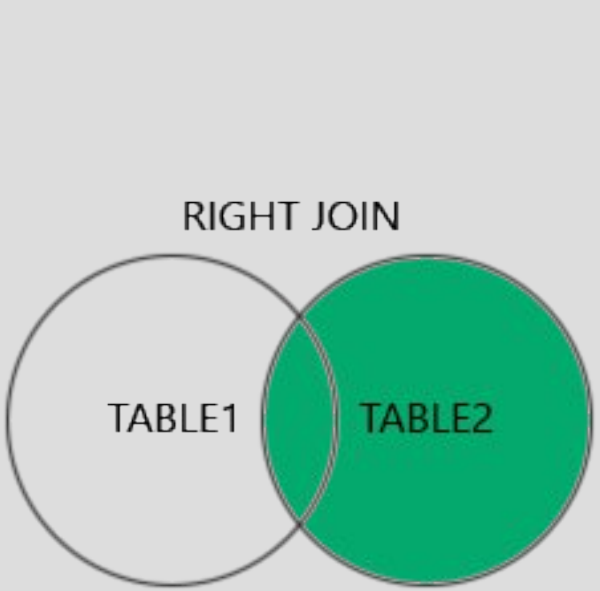




RIGHT OUTER JOIN

RaumschiffName	RaumschiffTyp Id_fk
Enterprise	1
Voyager	2
Serenity	3
Weißer Stern	3

RaumschiffTypId	Bezeichnung
1	Fregatte
2	Transportschiff
3	Spionageschiff
4	Raumstation



RaumschiffName	Bezeichnung
Enterprise	Fregatte
Voyager	Transportschiff
Serenity	Spionageschiff
Weißer Stern	Spionageschiff
null	Raumstation

RIGHT/LEFT OUTER JOIN

```
MariaDB [spaceships]> SELECT bezeichnung, raumschiffname
    -> FROM raumschiff
    -> LEFT OUTER JOIN raumschifftyp ON
    -> raumschifftypId_fk = raumschifftypId;
+------------------+------------------+----------+
| bezeichnung      | raumschiffname   |          |
+------------------+------------------+----------+
| Fregatte         | Enterprise       |          |
| Forschungsschiff | Voyager          |          |
| Raumstation      | DeepSpace 9      |          |
| Transportschiff  | Sehnsucht nach Une dlichkeit |
| Raumstation      | Babylon 5        |          |
| Fregatte         | Weisser Stern    |          |
| Transportschiff  | Serenity         |          |
| Forschungsschiff | Millenium Falke  |          |
| Forschungsschiff | Defiant          |          |
+------------------+------------------+----------+
9 rows in set (0.000 sec)
```

LEFT OUTER JOIN

```
MariaDB [spaceships]> SELECT bezeichnung, raumschiffname FROM raumschiff
    -> RIGHT OUTER JOIN raumschifftyp ON
    -> raumschifftypId_fk=raumschifftypId;
+------------------+------------------------------+
| bezeichnung      | raumschiffname               |
+------------------+------------------------------+
| Fregatte         | Enterprise                   |
| Forschungsschiff | Voyager                      |
| Raumstation      | DeepSpace 9                  |
| Transportschiff  | Sehnsucht nach Unendlichkeit |
| Raumstation      | Babylon 5                    |
| Fregatte         | Weisser Stern                |
| Transportschiff  | Serenity                     |
| Forschungsschiff | Millenium Falke              |
| Forschungsschiff | Defiant                      |
| Spionageschiff   | NULL                         |
| Passagierschiff  | NULL                         |
+------------------+------------------------------+
11 rows in set (0.001 sec)
```

RIGHT OUTER JOIN

LEFT/RIGHT OUTER JOIN

- Es gilt: Beim LEFT OUTER JOIN ist immer die als erstes angegebene Tabelle jene, die als Ganzes ausgegeben wird.
  - **Ausgabe**: INNER JOIN + nicht gematchte Zeilen der Tabelle LINKS vom Join
- Beim RIGHT OUTER JOIN ist es umgekehrt.

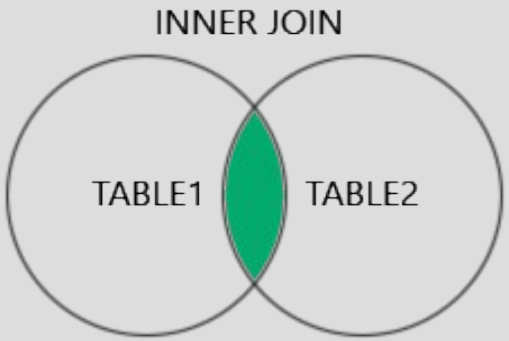
  - **Ausgabe**: INNER JOIN + nicht gematchte Zeilen der Tabelle RECHTS vom Join

ZUSAMMENFASSENDES BEISPIEL

- Inner Join

```
MariaDB [spaceships]> SELECT * FROM raumschifftyp;
+-----------------+-----------------+-----------+---------+-----------------+--------------+-----------+-----------------+
| raumschifftypid | bezeichnung     | besatzung | groesse | anzahlgeschuetze | schildstaerke | lagerraum | geschwindigkeit |
+-----------------+-----------------+-----------+---------+-----------------+--------------+-----------+-----------------+
|               1 | Fregatte        |      1000 |     500 |             150 |       1567.12 |        50 |            1123 |
|               2 | Forschungsschiff |       100 |     200 |              20 |         987.4 |       100 |            2000 |
|               3 | Raumstation     |     10000 |    7000 |            1234 |       2345.67 |      9878 |               0 |
|               4 | Transportschiff |       200 |    3000 |             120 |        800.67 |      9000 |             400 |
|               5 | Spionageschiff  |        10 |      10 |               5 |            30 |         0 |            3000 |
|               6 | Passagierschiff |        10 |     500 |              10 |            30 |         0 |            1600 |
+-----------------+-----------------+-----------+---------+-----------------+--------------+-----------+-----------------+
6 rows in set (0.001 sec)
```

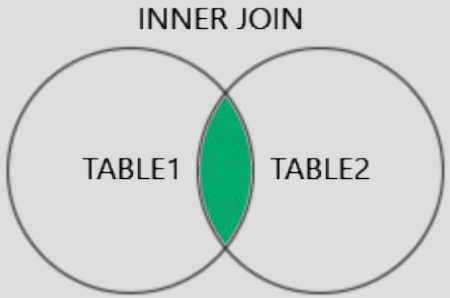
```
MariaDB [spaceships]> SELECT * FROM raumschiff;
+--------------+-----------------------------+---------+-------+-------------------+--------------+-------------+
| raumschiffId | raumschiffname              | schaden | aktiv | raumschifftypid_fk | kapitaenId_fk | flotteId_fk |
+--------------+-----------------------------+---------+-------+-------------------+--------------+-------------+
|            1 | Enterprise                  |      55 |     1 |                 1 |            1 |           1 |
|            2 | Voyager                     |      80 |     1 |                 2 |            5 |           2 |
|            3 | DeepSpace 9                 |      20 |     1 |                 3 |           10 |        NULL |
|            4 | Sehnsucht nach Unendlichkeit |       0 |     1 |                 4 |            4 |           3 |
|            5 | Babylon 5                   |      50 |     1 |                 3 |           11 |        NULL |
|            6 | Weisser Stern               |      90 |     1 |                 1 |            9 |           1 |
|            7 | Serenity                    |       0 |     1 |                 4 |            6 |           3 |
|            8 | Millenium Falke             |       0 |     1 |                 2 |            3 |        NULL |
|            9 | Defiant                     |       0 |     1 |                 2 |            5 |           6 |
|           13 | Todesstern                  |      34 |     1 |              NULL |            1 |        NULL |
+--------------+-----------------------------+---------+-------+-------------------+--------------+-------------+
10 rows in set (0.001 sec)
```



ZUSAMMENFASSENDES BEISPIEL

- Inner Join

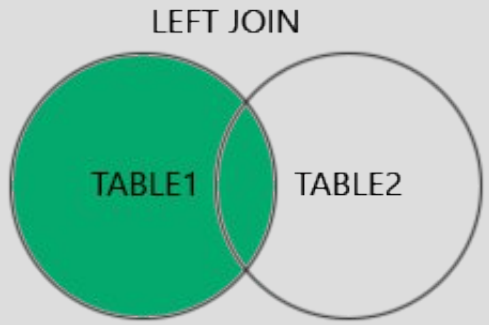
```
MariaDB [spaceships]> SELECT bezeichnung, raumschiffname FROM raumschiff
INNER JOIN raumschifftyp ON raumschifftypId=raumschifftypId_fk;
+------------------+------------------------------+
| bezeichnung      | raumschiffname               |
+------------------+------------------------------+
| Fregatte         | Enterprise                   |
| Forschungsschiff | Voyager                      |
| Raumstation      | DeepSpace 9                  |
| Transportschiff  | Sehnsucht nach Unendlichkeit |
| Raumstation      | Babylon 5                    |
| Fregatte         | Weisser Stern                |
| Transportschiff  | Serenity                     |
| Forschungsschiff | Millenium Falke              |
| Forschungsschiff | Defiant                      |
+------------------+------------------------------+
9 rows in set (0.001 sec)
```



ZUSAMMENFASSENDES BEISPIEL

- Left Outer Join

```
MariaDB [spaceships]> SELECT * FROM raumschiff;
+--------------+-----------------------------+---------+------+---------------------+--------------+-------------+
| raumschiffId | raumschiffname              | schaden | aktiv | raumschifftypid_fk | kapitaenId_fk | flotteId_fk |
+--------------+-----------------------------+---------+------+---------------------+--------------+-------------+
|            1 | Enterprise                  |      55 |    1 |                   1 |            1 |           1 |
|            2 | Voyager                     |      80 |    1 |                   2 |            5 |           2 |
|            3 | DeepSpace 9                 |      20 |    1 |                   3 |           10 |        NULL |
|            4 | Sehnsucht nach Unendlichkeit |       0 |    1 |                   4 |            4 |           3 |
|            5 | Babylon 5                   |      50 |    1 |                   3 |           11 |        NULL |
|            6 | Weisser Stern               |      90 |    1 |                   1 |            9 |           1 |
|            7 | Serenity                    |       0 |    1 |                   4 |            6 |           3 |
|            8 | Millenium Falke             |       0 |    1 |                   2 |            3 |        NULL |
|            9 | Defiant                     |       0 |    1 |                   2 |            5 |           6 |
|           13 | Todesstern                  |      34 |    1 |                NULL |            1 |        NULL |
+--------------+-----------------------------+---------+------+---------------------+--------------+-------------+
10 rows in set (0.001 sec)
```



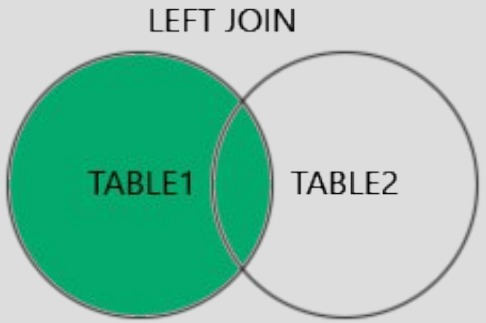

```
MariaDB [spaceships]> SELECT * FROM raumschifftyp;
+-----------------+----------------+-----------+---------+-----------------+--------------+-----------+-----------------+
| raumschifftypid | bezeichnung    | besatzung | groesse | anzahlgeschuetze | schildstaerke | lagerraum | geschwindigkeit |
+-----------------+----------------+-----------+---------+-----------------+--------------+-----------+-----------------+
|               1 | Fregatte       |      1000 |     500 |             150 |      1567.12 |        50 |            1123 |
|               2 | Forschungsschiff |       100 |     200 |              20 |        987.4 |       100 |            2000 |
|               3 | Raumstation    |     10000 |    7000 |            1234 |      2345.67 |      9878 |               0 |
|               4 | Transportschiff |       200 |    3000 |             120 |       800.67 |      9000 |             400 |
|               5 | Spionageschiff |        10 |      10 |               5 |           30 |         0 |            3000 |
|               6 | Passagierschiff |        10 |     500 |              10 |           30 |         0 |            1600 |
+-----------------+----------------+-----------+---------+-----------------+--------------+-----------+-----------------+
6 rows in set (0.001 sec)
```

ZUSAMMENFASSENDES BEISPIEL

- Left Outer Join

LEFT ✉ LINKS vom Schlüsselwort „JOIN“

```
MariaDB [spaceships]> SELECT bezeichnung, raumschiffname FROM raumschiff
LEFT OUTER JOIN raumschifftyp ON raumschifftypId=raumschifftypId_fk;
+-----------------+-----------------------------+
| bezeichnung     | raumschiffname              |
+-----------------+-----------------------------+
| Fregatte        | Enterprise                  |
| Forschungsschiff | Voyager                    |
| Raumstation     | DeepSpace 9                 |
| Transportschiff | Sehnsucht nach Unendlichkeit |
| Raumstation     | Babylon 5                   |
| Fregatte        | Weisser Stern               |
| Transportschiff | Serenity                    |
| Forschungsschiff | Millenium Falke            |
| Forschungsschiff | Defiant                    |
| NULL            | Todesstern                  |
+-----------------+-----------------------------+
10 rows in set (0.001 sec)
```

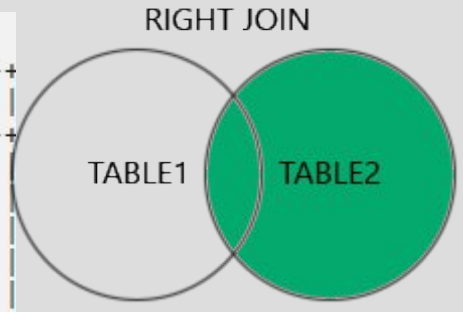


ZUSAMMENFASSENDES BEISPIEL

- RIGHT Outer Join

```
MariaDB [spaceships]> SELECT * FROM raumschifftyp;
+-----------------+-----------------+-----------+---------+-----------------+---------------+-----------+-----------------+
| raumschifftypid | bezeichnung     | besatzung | groesse | anzahlgeschuetze | schildstaerke | lagerraum | geschwindigkeit |
+-----------------+-----------------+-----------+---------+-----------------+---------------+-----------+-----------------+
|               1 | Fregatte        |      1000 |     500 |             150 |       1567.12 |        50 |            1123 |
|               2 | Forschungsschiff |       100 |     200 |              20 |         987.4 |       100 |            2000 |
|               3 | Raumstation     |     10000 |    7000 |            1234 |       2345.67 |      9878 |               0 |
|               4 | Transportschiff |       200 |    3000 |             120 |        800.67 |      9000 |             400 |
|               5 | Spionageschiff  |        10 |      10 |               5 |            30 |         0 |            3000 |
|               6 | Passagierschiff |        10 |     500 |              10 |            30 |         0 |            1600 |
+-----------------+-----------------+-----------+---------+-----------------+---------------+-----------+-----------------+
6 rows in set (0.001 sec)
```

```
MariaDB [spaceships]> SELECT * FROM raumschiff;
+--------------+-----------------------------+---------+-------+---------------------+---------------+-------------+
| raumschiffId | raumschiffname              | schaden | aktiv | raumschifftypid_fk  | kapitaenId_fk | flotteId_fk |
+--------------+-----------------------------+---------+-------+---------------------+---------------+-------------+
|            1 | Enterprise                  |      55 |     1 |                   1 |             1 |           1 |
|            2 | Voyager                     |      80 |     1 |                   2 |             5 |           2 |
|            3 | DeepSpace 9                 |      20 |     1 |                   3 |            10 |        NULL |
|            4 | Sehnsucht nach Unendlichkeit |       0 |     1 |                   4 |             4 |           3 |
|            5 | Babylon 5                   |      50 |     1 |                   3 |            11 |        NULL |
|            6 | Weisser Stern               |      90 |     1 |                   1 |             9 |           1 |
|            7 | Serenity                    |       0 |     1 |                   4 |             6 |           3 |
|            8 | Millenium Falke             |       0 |     1 |                   2 |             3 |        NULL |
|            9 | Defiant                     |       0 |     1 |                   2 |             5 |           6 |
|           13 | Todesstern                  |      34 |     1 |                NULL |             1 |        NULL |
+--------------+-----------------------------+---------+-------+---------------------+---------------+-------------+
10 rows in set (0.001 sec)
```

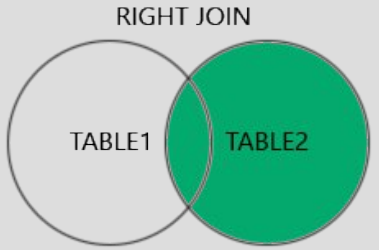


ZUSAMMENFASSENDES BEISPIEL

- Right Outer Join

RIGHT —> Rechts vom Schlüsselwort „JOIN“

```
MariaDB [spaceships]> SELECT bezeichnung, raumschiffname FROM raumschiff
RIGHT OUTER JOIN raumschifftyp ON raumschifftypId=raumschifftypId_fk;
+------------------+------------------------------+
| bezeichnung      | raumschiffname               |
+------------------+------------------------------+
| Fregatte         | Enterprise                   |
| Forschungsschiff | Voyager                      |
| Raumstation      | DeepSpace 9                  |
| Transportschiff  | Sehnsucht nach Unendlichkeit |
| Raumstation      | Babylon 5                    |
| Fregatte         | Weisser Stern                |
| Transportschiff  | Serenity                     |
| Forschungsschiff | Millenium Falke              |
| Forschungsschiff | Defiant                      |
| Spionageschiff   | NULL                         |
| Passagierschiff  | NULL                         |
+------------------+------------------------------+
11 rows in set (0.001 sec)
```



ARTEN VON JOINS

- **INNER JOIN:** Zeigt Datensätze mit übereinstimmenden Werten in beiden Tabellen.
- **LEFT JOIN (LEFT OUTER JOIN):** Zeigt alle Datensätze aus der linken Tabelle und die übereinstimmenden Datensätze aus der rechten Tabelle.
- **RIGHT JOIN (RIGHT OUTER JOIN):** Zeigt alle Datensätze aus der rechten Tabelle und die übereinstimmenden Datensätze aus der linken Tabelle.
- **FULL JOIN (FULL OUTER JOIN):** Zeigt alle Datensätze, wenn es eine Übereinstimmung in entweder der linken oder der rechten Tabelle gibt.

ZUSAMMENFASSUNG

Left Outer Join	Right Outer Join	Full Outer Join
Fetches all the rows from the table on the left	Fetches all the rows from the table on the right	Fetches all the rows from both the tables
Inner Join + all the unmatched rows from the left table	Inner Join + all the unmatched rows from the right table	Inner Join + all the unmatched rows from the left table + all the unmatched rows from the right table
Unmatched data of the right table is lost	Unmatched data of the left table is lost	No data is lost
SELECT [column1, column2, ...] FROM table1 LEFT OUTER JOIN table2 ON table1.matching_column = table2.matching_column	SELECT [column1, column2, ...] FROM table1 RIGHT OUTER JOIN table2 ON table1.matching_column = table2.matching_column	SELECT [column1, column2, ...] FROM table1 FULL OUTER JOIN table2 ON table1.matching_column = table2.matching_column

Quelle: https://www.geeksforgeeks.org/difference-between-left-right-and-full-outer-join/ [12.01.2024]