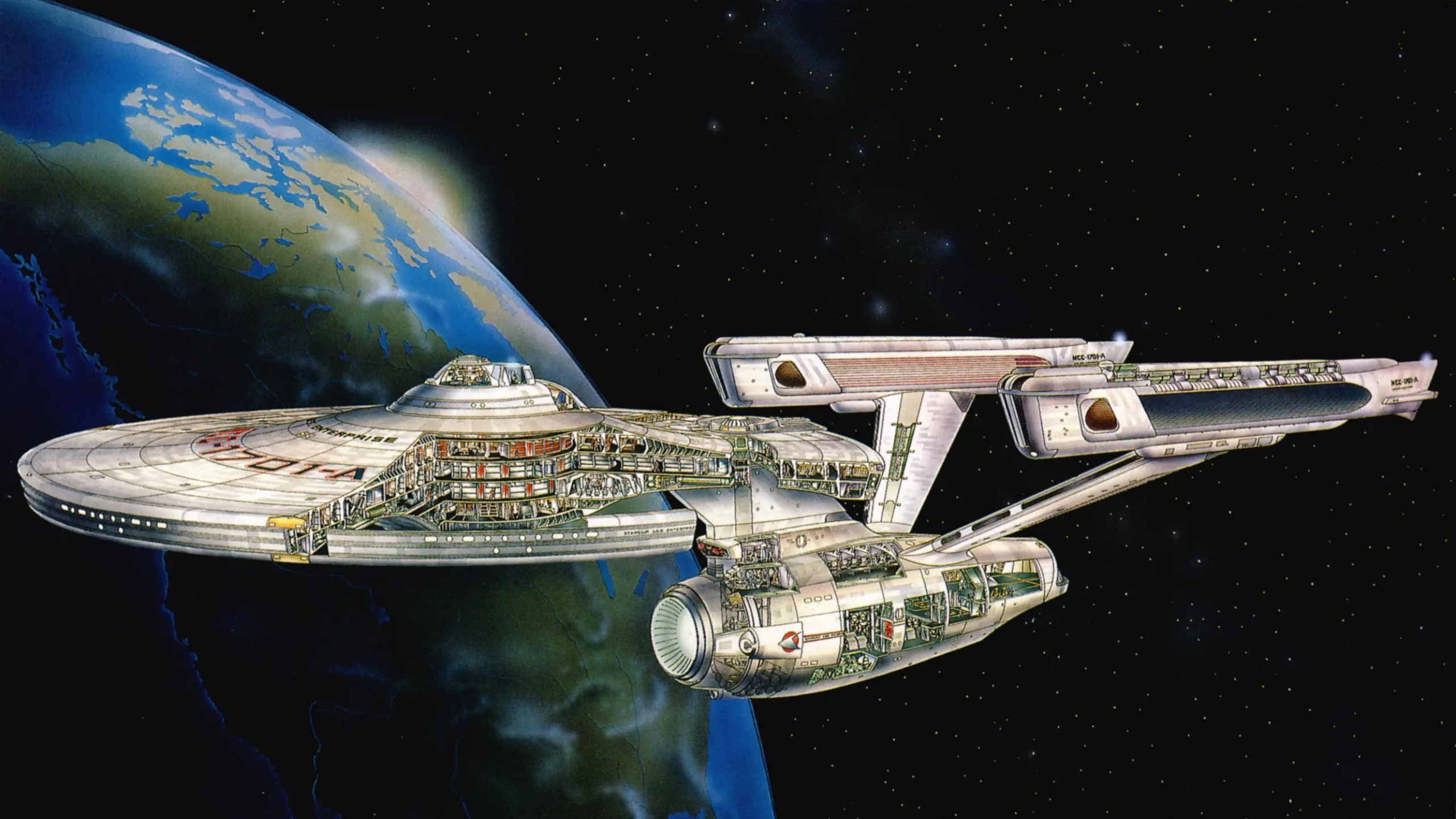




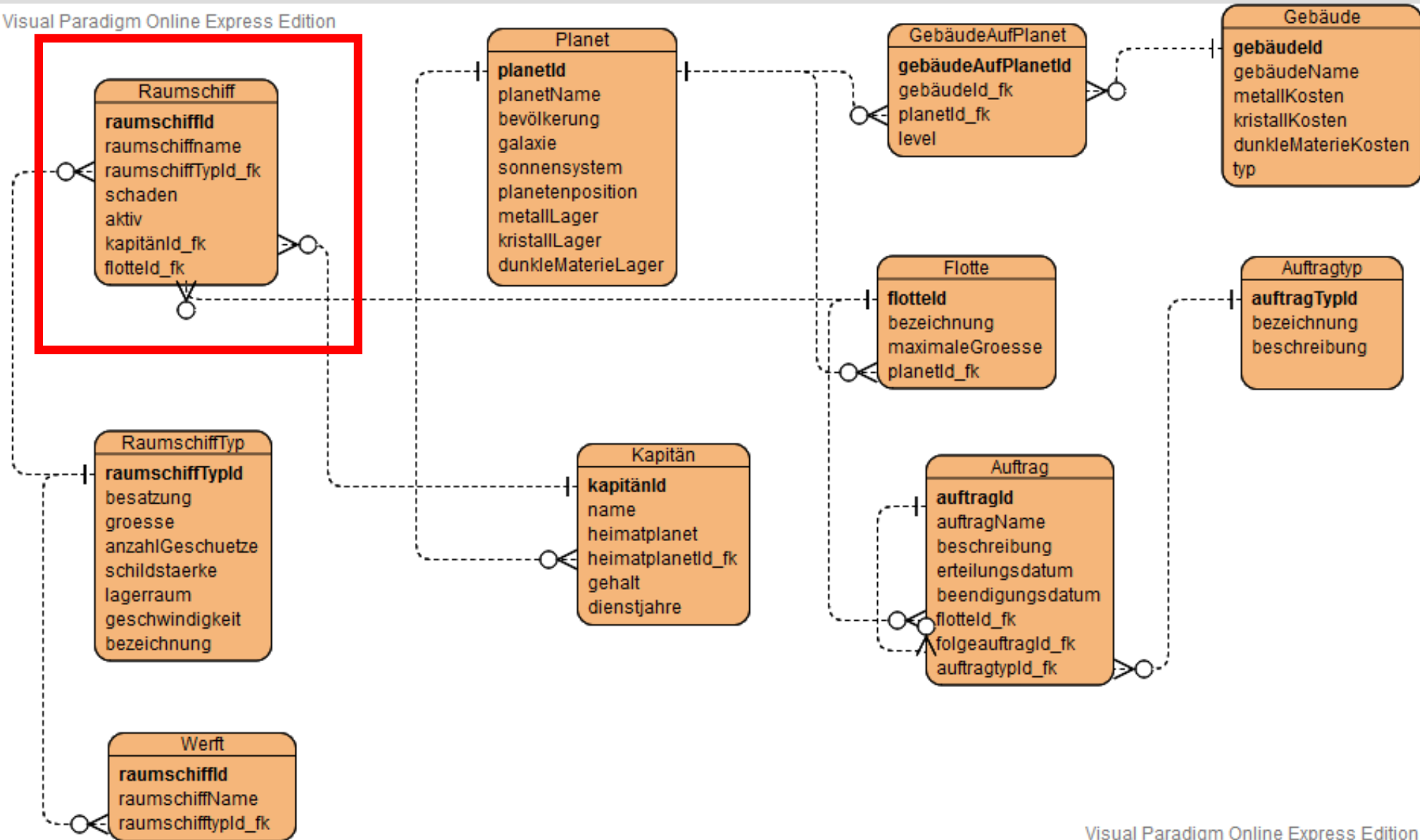
SQL-ABFRAGEN TEIL 1

SELECT --- WHERE --- RECHNEN --- BETWEEN --- LIKE --- IN



ÜBUNGSDATENBANK „RAUMSCHIFFE“

Visual Paradigm Online Express Edition



Visual Paradigm Online Express Edition

DATENBANK „RAUMSCHIFFE“

- Welche **Tabellen** sind vorhanden?
- Welche **Felder** sind in einer Tabelle vorhanden?
- Welche **Datensätze** sind gespeichert?
- Wie kann ich nach bestimmten **Daten suchen**?

SQL-Befehle helfen uns, diese Fragen zu beantworten!

```
mysql> show tables;
+-----+
| Tables_in_raumschiffe |
+-----+
| auftrag                |
| auftragtyp             |
| gebaeude               |
| gebaeudeaufplanet      |
| kapitaen               |
| logbuch                |
| planet                 |
| raumschiff              |
| raumschiffotyp         |
| schadensbericht        |
| werft                  |
| zeit                   |
+-----+
```

VORBEREITUNGEN (CMD)

- Öffne ein Eingabefenster (CMD) und wechsle zum Binary-Verzeichnis von mysql in xampp
- Logge dich mit deinen Userdaten ein

Eingabeaufforderung - mysql -u root -p

```
C:\Users\Patrick>cd C:\xampp\mysql\bin
```

```
C:\xampp\mysql\bin>mysql -u root -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 441
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

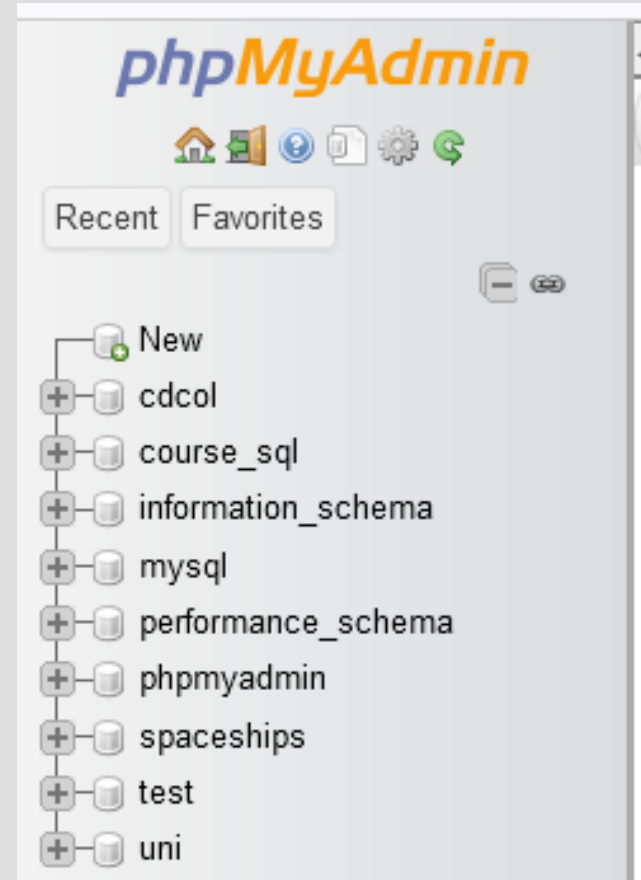
MariaDB [(none)]>
```


DATENBANKEN ANZEIGEN

```
MariaDB [(none)]> show databases;
```

Database
cdcol
course_sql
information_schema
mysql
performance_schema
phpmyadmin
spaceships
test
uni

9 rows in set (0.001 sec)



USE DATABASE

```
MariaDB [spaceships]> USE spaceships;  
Database changed  
MariaDB [spaceships]>
```

phpMyAdmin

Recent Favorites

Server: 127.0.0.1 » Database: spaceships

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> auftrag	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> auftragtyp	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> flotte	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	48.0 KiB	-
<input type="checkbox"/> gebaeude	★ Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> gebaeudeaufplanet	★ Browse Structure Search Insert Empty Drop	35	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> kapitaen	★ Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> logbuch	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> planet	★ Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> raumschiff	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> raumschifftyp	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> schadensbericht	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> werft	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
12 tables	Sum	110	InnoDB	utf8mb4_general_ci	224.0 KiB	0 B

⬆ ☐ Check all With selected:

SHOW TABLES

```
MariaDB [spaceships]> SHOW TABLES;
```

```
+-----+
| Tables_in_spaceships |
+-----+
| auftrag               |
| auftragtyp            |
| flotte                |
| gebaeude              |
| gebaeudeaufplanet     |
| kapitaen              |
| logbuch               |
| planet                |
| raumschiff             |
| raumschiffotyp        |
| schadensbericht       |
| werft                 |
+-----+
12 rows in set (0.001 sec)
```

The screenshot shows the phpMyAdmin interface for the 'spaceships' database. The left sidebar displays the database structure tree, and the main panel shows the 'Structure' tab with a list of tables and their properties.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> auftrag	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> auftragtyp	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> flotte	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	48.0 KiB	-
<input type="checkbox"/> gebaeude	★ Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> gebaeudeaufplanet	★ Browse Structure Search Insert Empty Drop	35	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> kapitaen	★ Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> logbuch	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> planet	★ Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> raumschiff	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> raumschiffotyp	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> schadensbericht	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> werft	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
12 tables	Sum	110	InnoDB	utf8mb4_general_ci	224.0 KiB	0 B

At the bottom, there is a 'Check all' checkbox and a 'With selected:' dropdown menu.

SELECT

- Ziel: **Projektion** (Anzeige bestimmter Felder)
- Felder (also Spalten) nach SELECT werden angezeigt
- Vier Bestandteile
 - 1. SELECT
 - II. Feldname(n)
 - III. FROM
 - Tabelle

SELECT

raumschiffname, schaden

FROM

raumschiff



SELECT

- Bsp: Wir möchten alle Felder der Tabelle Raumschiff sehen:

```
mysql> 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

SELECT

- Bsp: Wir möchten alle Raumschiffnamen mit dem Schaden sowie dem Feld „aktiv“ sehen:

```
mysql> SELECT raumschiffname, schaden, aktiv FROM raumschiff;
```

raumschiffname	schaden	aktiv
Enterprise	55	1
Voyager	80	1
DeepSpace 9	20	1
Sehnsucht nach Unendlichkeit	0	1
Babylon 5	50	1
Weisser Stern	90	1
Serenity	0	1
Millenium Falke	0	1
Defiant	0	1

SELECT

- Bsp: Gleicher Aufruf mit vertauschten Feldern:

```
mysql> SELECT aktiv, schaden, raumschiffname FROM raumschiff;
```

aktiv	schaden	raumschiffname
1	55	Enterprise
1	80	Voyager
1	20	DeepSpace 9
1	0	Sehnsucht nach Unendlichkeit
1	50	Babylon 5
1	90	Weisser Stern
1	0	Serenity
1	0	Millenium Falke
1	0	Defiant

- Die **Reihenfolge** im SQL-Befehl steuert die Ausgabe – hat keinen Einfluss auf die Tabelle.
- Für die Anzeige wird eine **temporäre Tabelle** erstellt und angezeigt

ÜBUNGEN

- Zeige von **allen Gebäuden** nur die **ID** und den **Namen** an
- Wir möchten **alle Datensätze** der **Tabelle kapitaen** sehen
- Zeige von **allen Kapitänen** nur die **ID**, den **Namen** und das **Gehalt**
- Zeige von **allen Raumschifftypen** **alle Felder außer der Geschwindigkeit** an
- Erzeuge und erkläre folgende **Fehlermeldungen** bzw. **falsche Ausgaben**:
 - SELECT FROM raumschiff;
 - SELECT raumschiffname;
 - SELECT raumschiffname raumschiffid FROM raumschiff;
 - SELECT raumschiffname raumschiffid raumschiff;
 - SELECT raumschiffname raumschiffid FROM raumschiff

ÜBUNGEN - LÖSUNGEN

Antwort

```
SELECT gebaeudeld, gebaeudeName FROM gebaeude;
```

Antwort

```
SELECT * FROM kapitaen;
```

Antwort

```
SELECT kapitaenId, kapitaenName, gehalt FROM kapitaen;
```

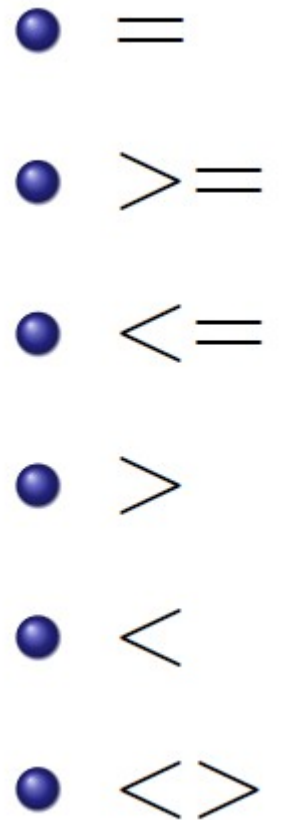
Antwort

```
SELECT raumschifftypId, bezeichnung, besatzung, groesse,  
anzahlGeschuetze, schildstaerke, lagerraum;
```


SELECT ... WHERE

- Ziel: **Selektionen** (Auswahl bestimmter Datensätze)
- Datensatz wird nur angezeigt, wenn er die **Bedingungen** erfüllt
- Drei Bestandteile:
 - I. DATENFELD raumschiffID_fk
 - II. OPERATOR =
 - III. WERT 3

Mehr Operatoren



SELECT ... WHERE

- Bsp: Wir möchten alle Datensätze sehen, die eine „3“ als raumschiffotyp haben:

```
mysql> SELECT * FROM Raumschiff WHERE raumschiffotypid_fk=3;
```

raumschiffId	raumschiffname	schaden	aktiv	raumschiffotypid_fk	kapitaenId_fk	flotteId_fk
3	DeepSpace 9	20	1	3	10	NULL
5	Babylon 5	50	1	3	11	NULL

- Auswahl („Selektion“) der Datensätze nach einer Bedingung

ÜBUNGEN

- Zeige alle Raumschiffnamen und die Höhe des Schadens an, wenn der Schaden größer als 20 ist.
- Zeige alle Raumschiffnamen, die ID des Kapitäns und die Höhe des Schadens an, wenn der Schaden zwischen 40 und 60 ist.
- Zeige alle Raumschiffnamen sowie den Schaden der Flotte mit der ID „3“ an.
- Zeige alle Raumschiffnamen der Kapitäne mit einer ID kleiner als 4 an.

RECHENOPERATIONEN

- SQL kann auch als „Taschenrechner“ verwendet werden

```
mysql> SELECT 4+5;
+-----+
| 4+5 |
+-----+
|    9 |
+-----+
```

- Anwendung: z.B. Temporäre **Manipulation von Werten**:

```
mysql> SELECT besatzung + 5 FROM raumschiffotyp;
```

besatzung + 5
1005
105
10005
205
15
15

```
mysql> SELECT besatzung+5, besatzung FROM raumschiffotyp;
```

besatzung+5	besatzung
1005	1000
105	100
10005	10000
205	200
15	10
15	10



Werte in der
Datenbank
bleiben
unverändert!

RECHENOPERATIONEN

- SQL-Rechenoperatoren

+ - * / DIV MOD

- DIV und MOD für **ganzzahlige Division**
 - Was ist 10 geteilt durch 3?
 - ✓ 3 passt dreimal in 10
 - ✓ Rest ist 1
 - $3*3 + 1 = 10$

RECHENOPERATIONEN

- SQL-Rechenoperatoren

+ - * / DIV MOD

- DIV und MOD in SQL

```
mysql> SELECT 10 DIV 3;
```

```
+-----+
| 10 DIV 3 |
+-----+
|          3 |
+-----+
```

```
mysql> SELECT 10 MOD 3;
```

```
+-----+
| 10 MOD 3 |
+-----+
|          1 |
+-----+
```

- Sehr hilfreich für **Datenanalyse** (z.B. „jeder 3. Datensatz...”)

RECHENOPERATIONEN

- Prioritäten bei SQL-Rechenoperationen

1. - (unärer Operator, z.B. -3)
2. *, /, DIV, MOD
3. +, -

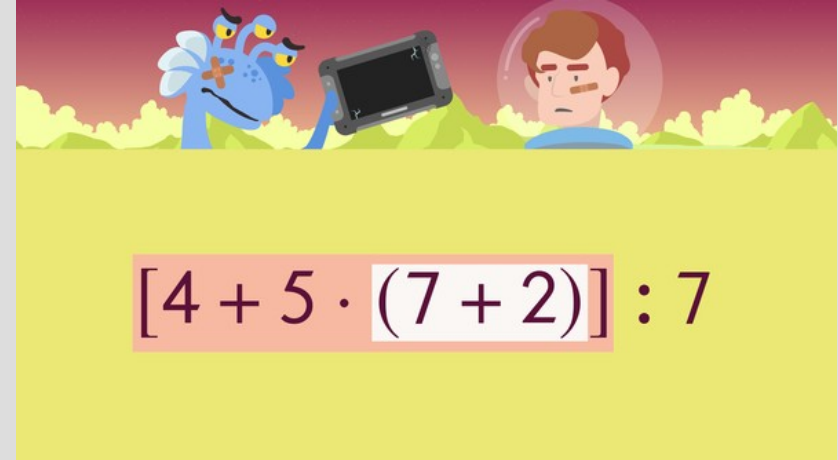
```
mysql> SELECT 4+6*2;
```

+-----+
4+6*2
+-----+
16
+-----+

```
mysql> SELECT (4+6)*2;
```

+-----+
(4+6)*2
+-----+
20
+-----+

Klammern für
andere
Prioritäten



ÜBUNGEN

Ermittle zuerst im Kopf den erwarteten Output und überprüfe danach mit dem SQL-Befehl:

- `SELECT 2 + 4 - + 7;`
- `SELECT (4+5) * (8-3);`
- `SELECT 4+5 * 8-3;`
- `SELECT 4 MOD 2 + 5;`
- `SELECT 10 MOD 3 * 10 DIV 3;`

BETWEEN

- Erweiterung von Select ... Where

```
mysql> SELECT * FROM Raumschifftyp WHERE groesse BETWEEN 5000 and 10000;
```

raumschiffotypid	bezeichnung	besatzung	groesse	anzahlgeschuetze	schildstaerke	lagerraum	geschwindigkeit
3	Raumstation	10000	7000	1234	2345.67	9878	0

- Funktioniert auch mit **Gleitkommazahlen**:

```
mysql> SELECT raumschiffname, schaden FROM raumschiff WHERE schaden BETWEEN 50.5 and 99.5;
```

raumschiffname	schaden
Enterprise	55
Voyager	80
Weisser Stern	90

DESCRIBE

- Schneller Überblick über die Attribute und Datentypen:

```
MariaDB [spaceships]> describe kapitaen;
```

Field	Type	Null	Key	Default	Extra
kapitaenid	int(11)	YES		NULL	
kapitaenname	varchar(100)	YES		NULL	
heimatplanet	varchar(100)	YES		NULL	
gehalt	double	YES		NULL	
dienstjahre	int(11)	YES		NULL	
heimatplanetId_fk	int(11)	YES		NULL	

6 rows in set (0.022 sec)

BETWEEN

- Funktioniert auch mit **Zeichenketten**:

```
mysql> SELECT * FROM raumschiff WHERE raumschiffname BETWEEN "Enterprise" and "Voyager";
```

raumschiffId	raumschiffname	schaden	aktiv	raumschiffotypid_fk	kapitaenId_fk	flotteId_fk
1	Enterprise	55	1	1	1	1
2	Voyager	80	1	2	5	2
4	Sehnsucht nach Unendlichkeit	0	1	4	4	3
7	Serenity	0	1	4	6	3
8	Millenium Falke	0	1	2	3	NULL

LIKE

- Wie finde ich alle Raumschiffe, deren Name mit „F“ beginnt?
 - So nicht: `mysql> SELECT * FROM Raumschiffotyp WHERE bezeichnung="F";`
Empty set (0.00 sec)
- Mit „**LIKE**“ und einem **Muster**
 - Muster: % oder _

`mysql> SELECT * FROM Raumschiffotyp WHERE bezeichnung LIKE "F%";`

% steht für alle
möglichen
Zeichenfolgen

raumschiffotypid	bezeichnung	besatzung	groesse	anzahlgeschuetze	schildstaerke	lagerraum	geschwindigkeit
1	Fregatte	1000	500	150	1567		1123
2	Forschungsschiff	100	200	20	987		2000

_ steht für genau ein
Zeichen

`mysql> SELECT * FROM Raumschiff WHERE raumschiffname LIKE "Serenit_";`

raumschiffId	raumschiffname	schaden	aktiv	raumschiffotypid_fk	kapitaenId_fk	flotteId_fk
7	Serenity	0	1	4	6	3

LIKE

- Wie finde ich ein Raumschiff, dessen Name „nach“ enthält?

Suchwort und
davor/danach ein %

```
mysql> SELECT * FROM Raumschiff WHERE raumschiffname LIKE "%nach%";
```

raumschiffId	raumschiffname	schaden	aktiv	raumschifftypeid_fk	kapitaenId_fk	flotteId_fk
4	Sehnsucht nach Unendlichkeit	0	1	4	4	3

IN

- Wie finde ich alle Raumschiffe, deren Typ 1, 2 oder 3 ist?
- Der IN-Operator durchsucht eine Liste aus Werten

Liste (1,2,3)

```
mysql> SELECT * FROM Raumschiff WHERE raumschifftypid_fk IN (1,2,3);
```

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
5	Babylon 5	50	1	3	11	NULL
6	Weisser Stern	90	1	1	9	1
8	Millenium Falke	0	1	2	3	NULL
9	Defiant	0	1	2	5	6

Liste („Enterprise“, „Voyager“, „Babylon 5“)

```
mysql> SELECT * FROM Raumschiff WHERE raumschiffname IN ("Enterprise", "Voyager", "Babylon 5");
```

raumschiffId	raumschiffname	schaden	aktiv	raumschifftypid_fk	kapitaenId_fk	flotteId_fk
1	Enterprise	55	1	1	1	1
2	Voyager	80	1	2	5	2
5	Babylon 5	50	1	3	11	NULL