
PSQL-Datenbank „Projektwoche“

a) Präzise textuelle Beschreibung des zu modellierenden Weltausschnitts bzw. der Anforderungen an die Datenbank (DB) ohne Vorgriff auf Entwurfsentscheidungen des Designers.

Die Datenbank soll die Organisation einer Projektwoche an einer Schule ermöglichen bzw. erleichtern. Es wird davon ausgegangen, dass Listen der Angestellten und Listen der SchülerInnen vorliegen. Es gibt zwei Personengruppen. Erstens die SchülerInnen und zweitens die Angestellten, welches hauptsächlich Lehrkräfte aber z.B. auch Sozialpädagogen sein können. Falls Eltern den Wunsch haben Projekte anzubieten, müssten diese für die Projektwoche in die Liste der Angestellten aufgenommen werden.

Zunächst tragen einige der Angestellten Projekte als Leitung ein (es muss genau eine Projektleitung geben). Dabei können für jeden Tag der Projektwoche die vorher eingetragenen default-Werte, von welcher Uhrzeit bis zu welcher Uhrzeit das Projekt stattfindet, geändert werden. Außerdem muss für jeden Tag festgelegt werden, an welchem Ort das Projekt stattfindet, was leider etwas aufwendig ist aber mehr Flexibilität ermöglicht. Ein Projekt kann zu unterschiedlichen Zeiten an unterschiedlichen Orten stattfinden. Orte können auch gleichzeitig für mehrere Projekte genutzt werden. Da alle Orte (insbesondere Räume im Schulgebäude) eine begrenzte Anzahl an Plätze haben, ist dies auch ein Attribut der Orte, welches ggf. mit der Anzahl der teilnehmenden SchülerInnen verglichen werden sollte.

Nach der Eintragung der Projekte durch die projektleitende Person, sollen sich die übrigen Lehrkräfte einem oder zwei Projekten als weitere Leitung zuweisen oder sie werden einem oder zwei Projekten zugewiesen.

Die SchülerInnen weisen sich dann einem Projekt zu. Sie können dabei auch die Rolle eines Assistenten wählen, wenn Sie zusammen mit der projektleitenden Lehrkraft das Projekt organisieren und durchführen. SuS die sich keinem Projekt zuweisen können von den Lehrkräften einem Projekt zugewiesen werden. Projekte haben eine Mindestzahl und eine Maximalzahl an Teilnehmenden pro Lehrkraft. Während die Maximalzahl durch die maximale Platzanzahl des Raumes und ggf. die Materialmenge beschränkt ist, hat die Mindestgröße eines Projektes den Sinn, dass die Anzahl der SchülerInnen pro Lehrkraft ungefähr ähnlich ist, was für mehr gefühlte Gerechtigkeit sorgt. Da erst nach der Zuweisung der SchülerInnen zu den Projekten klar ist, welche Projekte sehr voll und welche sehr leer sind, sollte erst dann die Zuweisung von Mitleitenden Lehrkräften oder die Zuweisung von SchülerInnen zu Projekten geändert werden.

Projekte können auch Klassen- oder Kursfahrten sein. Das Startdatum und Enddatum eines Projektes kann auch um ein paar Tage von dem regulären Startdatum und Enddatum abweichen.

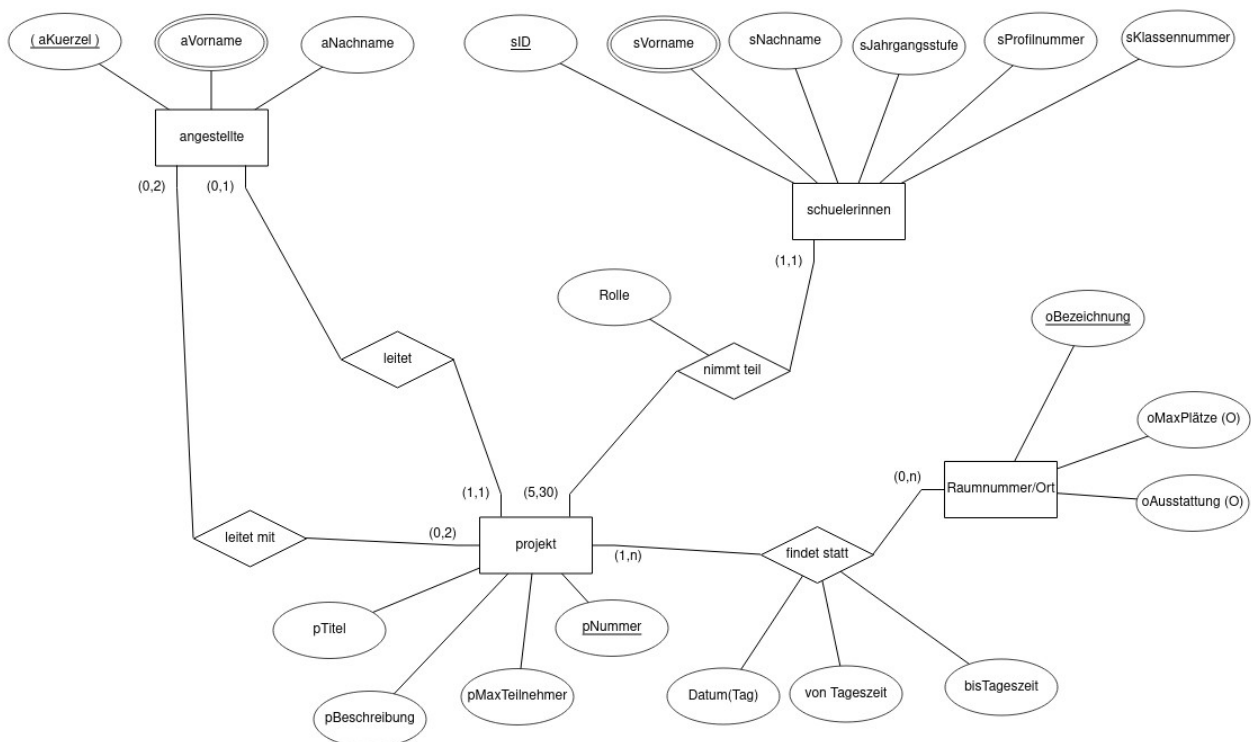
Datum des regulären Startprojektwochen werden. Ebenso lässt sich auch die Tageszeit zu der das Projekt täglich beginnt und endet kann

Grundlegende Anforderungen an die Datenbank, d.h. :

Die DB soll folgende Entitäten enthalten:, Angestellte, SchuelerInnen, Projekte, Orte

Sie DB soll folgende Relationen enthalten: leitet, leitet mit, nimmt teil, findet statt

b) Erstellung eines ER-Modells mit vollständiger Attributierung und Angabe von Schlüsseln und Komplexitäten.

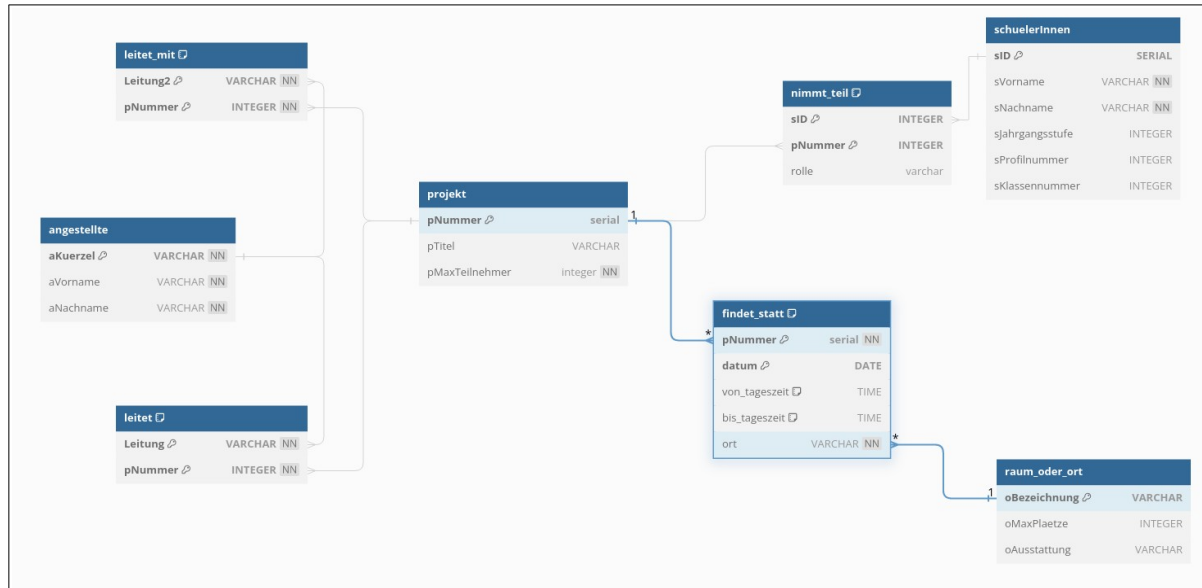


Erläuterungen zu diesem ER-Modell¹:

- Entitäten sind an Rechtecken, Beziehungen an Rauten, und Attribute an Ovalen zu erkennen
- die Angaben (z.B. (1,1) oder (1,n)) sind die Komplexitäten in der min-max-Notation
- Primärschlüsselattribute sind unterstrichen
- Mehrwertige Attribute sind doppelt umkreist
- Optionale Attribute sind an (O) zu erkennen

1 - Diese Grafik wurde erstellt mit: <https://erdplus.com/>

Hier noch eine andere Version des ER-Modells, welches später aus dem Inhalt der create_projektwoche.sql Datei erstellt wurde²:



c) Angabe von statischen und ggf. dynamischen Integritätsbedingungen, die nicht im „normalen“ ERM erfasst werden.

- die Datentypen der Attribute, Entitäten und Beziehungen
- Einschränkungen für die zulässigen Wertebereiche (z.B. für die Jahrgangsstufennummer, die bei einer Projektwoche für die Sekundarstufe 1 nur Werte von 7 bis 10 haben darf)
- das Datums und das Uhrzeitformat
- Nicht-Null Bedingungen für Attribute
- Regeln für das Updaten oder Löschen von Einträgen
- Regeln zur Vermeidung falscher Eintragungen (Angestellte sollten z.B. nicht versehentlich als SchülerIn eingetragen werden können)
- bedingte Einschränkungen für gewissen Einträge (z.B. kann die Rolle der SchülerInnen bei einem Projekt nur teilnehmend oder assistierend sein)
- Speicherung des Verlaufs bei Änderungen von Einträgen

d) Transformation des ER-Modells in eine Menge von Relationen (incl. Schlüssel und Fremdschlüssel

siehe Inhalt der Datei: create_projektwoche.sql

e) Angabe der verwendeten Transformationsregeln.

- Jede Entität im ER-Modell wird zu einer Relation (Tabelle).
- Jedes Attribut der Entität wird zu einer Spalte der Relation.
- Jede Beziehung im ER-Modell wird durch Hinzufügen von Fremdschlüsseln zu einer Relationen (Tabelle)
- Die einzelnen Attribute erhalten Bedingungen, d.h. ihnen wird ein Wertbereich zugewiesen

f) Angabe von funktionalen Abhängigkeiten in den Relationen (zur möglichen Prüfung der Normalformen).

Entität/ Relation	Primärschlüssel	Funktionale Abhängigkeit „ \rightarrow “ bedeutet hier „bestimmt“
angestellte	aKuerzel	aKuerzel \rightarrow aVorname, aNachname
schuelerInnen	sID	sID \rightarrow sVorname, sNachname, sJahrgangsstufe, sProfilnummer, sKlassennummer
projekt	pNummer	pNummer \rightarrow pTitel, pMaxTeilnehmer
nimmt_teil	{sID, pNummer}	
leitet	{Leitung, pNummer}	
leitet_mit	{Leitung2, pNummer}	
raum_oder_ort	oBezeichnung	oBezeichnung \rightarrow oMaxPlaetze, oAusstattung
findet_statt	{pNummer, datum}	{pNummer, datum} \rightarrow von_tageszeit, bis_tageszeit, ort

Alle Tabellen, d.h. sowohl die Entitäten als auch die Relationen, entsprechen der dritten Normalform, da sie alle der zweiten Normalform entsprechen und alle Attribute vollständig und nicht-transitiv bzw. direkt von dem jeweiligen Primärschlüssel abhängen.

g) Konkrete Umsetzung des Relationenentwurfs in einen Datenentwurf für die Implementierung in SQL (postgresql).

i. Umsetzung des Relationenentwurfs in DDL und DML, die als Dateien in die Datenbank eingelesen werden können.

Datendefinition (DDL) siehe create_projektwoche.sql

Datenmanipulation (DML) siehe insert_projektwoche.sql

ii. Begründung der Wahl besonderer Attributdatentypen (z. B. NUMERIC, BOOL, oder feste VARCHAR-Längen).

Um mehr Flexibilität zu ermöglichen wurde die varchar-Länge fast aller varchar-Werte nicht begrenzt, da davon auszugehen ist, dass die meisten Namen oder Bezeichnungen/Titel sowieso kurz gewählt werden. Lediglich die Projektbeschreibung wird auf 1000 Zeichen begrenzt, um zu lange Texte zu vermeiden.

Bei den INTEGER Werten wurden Vorgaben für die Länge und den Bereich gemacht, um Tippfehler zu vermeiden.

h) Entwurf einer Beispielpopulation der Tabellen: realitätsnah, variationsreich, aber klein.

siehe schuelerInnen.csv, angestelltei.csv und insert_projektwoche.sql

i) Konstruktion einer Aufgabensequenz von insgesamt ca. 10 einfachen bis zu schwierigen Anfragen für eine Umsetzung in...

- i. Relationenalgebra**
- ii. TRC und DRC (optional)**
- iii. SQL.**

Die Beispielanfragen in Relationenalgebra und SQL befinden sich auch in der Datei „Relationenalgebra Anfragen an die DB.odt“ die SQL-Beispielanfragen befinden sich auch in der Datei “SQL Anfragen an die DB.txt“.

SQL-Beispielanfragen an die PostgreSQL-Datenbank projektwoche:

1 - Alle Schülerinnen der Jahrgangsstufe 10:

```
SELECT sVorname, sNachname, sJahrgangsstufe
FROM schuelerInnen
WHERE sJahrgangsstufe = 10;
```

2 - Alle SchülerInnen aus dem Profil 1:

```
SELECT sVorname, sNachname, sJahrgangsstufe, sProfilnummer  
FROM schuelerInnen  
WHERE sProfilnummer = 1;
```

3 - Alle SchülerInnen der Klasse 721:

```
SELECT sVorname, sNachname, sJahrgangsstufe, sProfilnummer, sKlassennummer  
FROM schuelerInnen  
WHERE sJahrgangsstufe = 7  
AND sProfilnummer = 1  
AND sKlassennummer = 2;
```

4 - Alle Schülerinnen und Schüler anzeigen, die als Teilnehmer an dem Projekt 2 teilnehmen:

```
SELECT * FROM schuelerInnen  
WHERE sid IN  
(SELECT sid FROM nimmt_teil WHERE rolle = 'teilnehmer' AND pNummer = 2);
```

5 - Alle Projektnummern und Projekttitel mit den Nachnamen der leitenden und der mitleitenden Lehrkräften:

```
SELECT p.pNummer, p.pTitel, a1.aNachname AS Leiter, a2.aNachname AS MitLeiter  
FROM projekt p  
LEFT JOIN leitet l ON p.pNummer = l.pNummer  
LEFT JOIN angestellte a1 ON l.Leitung = a1.aKuerzel  
LEFT JOIN leitet_mit lm ON p.pNummer = lm.pNummer  
LEFT JOIN angestellte a2 ON lm.Leitung2 = a2.aKuerzel;
```

6 - Anzahl der schuelerInnen die an dem Projekt 5 teilnehmen:

```
SELECT COUNT(*) AS Anzahl_Teilnehmer
```

```
FROM nimmt_teil  
WHERE pNummer = 5;
```

7 - Daten aller schuelerInnen, die noch keinem Projekt zugeordnet sind:

```
SELECT sVorname, sNachname, sJahgangsstufe, sProfilnummer, sKlassennummer  
FROM schuelerInnen s  
LEFT JOIN nimmt_teil nt ON s.sID = nt.sID  
WHERE nt.sID IS NULL;
```

8 - Projektnummer und Projekttitel mit Anzahl der Teilnehmenden und der maximal zulässigen Anzahl an Teilnehmenden:

(folgende Anfrage funktioniert nicht optimal, da die Leitungen nicht in einer Zeile angegeben werden)

```
SELECT p.pNummer, p.pTitel, p.pMaxPers,  
COUNT(nt.sID) AS Anzahl_SchuelerInnen  
FROM projekt p  
LEFT JOIN nimmt_teil nt ON p.pNummer = nt.pNummer  
GROUP BY p.pNummer, p.pTitel, p.pMaxPers  
ORDER BY p.pNummer;
```

9 - Projektnummer und Projekttitel mit dem Nachnamen der dazugehörigen Leitung:

```
SELECT p.pNummer, p.pTitel, a.aNachname AS Leitung  
FROM projekt p  
JOIN leitet l ON p.pNummer = l.pNummer  
JOIN angestellte a ON l.L Leitung = a.aKuerzel  
ORDER BY p.pNummer, a.aNachname;
```

10 - Projektnummer und Projekttitel mit den Nachnamen der Leitung und den Nachnamen der Angestellten, die noch an dem Projekt beteiligt sind:

```
SELECT p.pNummer, p.pTitel, a1.aNachname AS Leitung, a2.aNachname AS MitLeitungen
```

```
FROM projekt p
JOIN leitet l ON p.pNummer = l.pNummer
JOIN angestellte a1 ON l.Leitung = a1.aKuerzel
LEFT JOIN leitet_mit lm ON p.pNummer = lm.pNummer
LEFT JOIN angestellte a2 ON lm.MitLeitungen = a2.aKuerzel
ORDER BY p.pNummer;
```

11 - Alle Raumbezeichnungen, die noch keinem Projekt zugewiesen sind:

```
SELECT ro.oBezeichnung
FROM raum_oder_ort ro
LEFT JOIN findet_statt fs ON ro.oBezeichnung = fs.ort
WHERE fs.ort IS NULL;
```

12 - Alle Räume die doppelt belegt sind mit Angabe des Projektes und der leitenden angestellten Person:

```
SELECT p1.pNummer, p1.pTitel, l1.Leitung, fs1.ort
FROM projekt p1
JOIN leitet l1 ON p1.pNummer = l1.pNummer
JOIN findet_statt fs1 ON p1.pNummer = fs1.pNummer
JOIN findet_statt fs2 ON fs1.ort = fs2.ort AND fs1.pNummer != fs2.pNummer
GROUP BY p1.pNummer, p1.pTitel, l1.Leitung, fs1.ort
ORDER BY fs1.ort;
```

13 - Projekttitel und die Kürzel der dazugehörigen Projektleitung von den Projekten, die am 27.06.2024 auf dem Schulhof stattfinden:

```
SELECT projekt.pTitel, angestellte.aKuerzel
FROM projekt
JOIN leitet ON projekt.pNummer = leitet.pNummer
JOIN angestellte ON leitet.Leitung = angestellte.aKuerzel
JOIN findet_statt ON projekt.pNummer = findet_statt.pNummer
```



```
WHERE findet_statt.datum = '2024-06-27'  
AND findet_statt.ort = 'Schulhof';
```

14 - Projekttitle und Projektleitungskürzel zusammen mit dem Datum und der Endzeit, von Projekten, die nach 16 Uhr noch stattfinden:

```
SELECT pTitel, angestellte.aKuerzel, findet_statt.datum, findet_statt.bis_tageszeit  
FROM projekt  
JOIN leitet ON projekt.pNummer = leitet.pNummer  
JOIN angestellte ON leitet.Leutung = angestellte.aKuerzel  
JOIN findet_statt ON projekt.pNummer = findet_statt.pNummer  
WHERE findet_statt.bis_tageszeit >= '16:00:00';
```

15 - Projektnummer, Projekttitle sowie Vorname und Name der SchülerInnen, die als assistent bei dem jeweiligen Projekten mitwirken:

```
SELECT schuelerInnen.sVorname, schuelerInnen.sNachname, projekt.pNummer, projekt.pTitel  
FROM schuelerInnen  
JOIN nimmt_teil ON schuelerInnen.sID = nimmt_teil.sID  
JOIN projekt ON nimmt_teil.pNummer = projekt.pNummer  
WHERE nimmt_teil.rolle = 'assistent'  
ORDER BY projekt.pNummer;
```

16 - Übersicht über ein Projekt (pro_übersicht.php):

```
SELECT *  
FROM projekt NATURAL JOIN  
(SELECT * from leitet, angestellte  
WHERE akuerzel = leitung) as xyz  
WHERE ptitel = :projektname;
```

17 - Teilnehmerdaten abrufen:

```
SELECT s.svorname, s.snachname, s.sjahrgangsstufe, s.sprofilnummer, s.sklassennummer
FROM schuelerinnen s
INNER JOIN nimmt_teil n ON s.sid = n.sid
WHERE n.pnummer = :pnummer AND n.rolle = 'teilnehmer';
```

19 Daten eines Projekts abfragen:

```
SELECT svorname, snachname, ptitel
FROM schuelerinnen INNER JOIN nimmt_teil
ON schuelerinnen.sid = nimmt_teil.sid
INNER JOIN projekt
ON nimmt_teil.pnummer = projekt.pnummer
WHERE schuelerinnen.sid = :sid AND projekt.ptitel = :projektname;
```

j) Erstellung von Musterlösungen zu diesen Anfragen in den genannten Sprachen (ggf. auch Alternativlösungen, Dokumentation der Anfrageergebnisse)

projektwoche=>			
projektwoche=> SELECT sVorname, sNachname, sJahrgangsstufe, sProfilnummer			
FROM schuelerInnen			
WHERE sProfilnummer = 1;			
svorname	snachname	sjahrgangsstufe	sprofilnummer
-----+-----+-----+-----			
Max	Müller	7	1
Julia Maria	Schmidt	7	1
David	Wagner	7	1
Sarah	Fischer	7	1
Lukas	Becker	7	1
Anna Maria	Schneider	7	1
Tim	Meier	7	1
Laura	Schulz	7	1
Jonas	Keller	7	1
Lisa	Lehmann	7	1
Kevin Justin	Koch	7	1
Marie	Richter	7	1
Paul	Wolf	7	1
Lena	Braun	7	1
Simon	Hofmann	7	1
Laura	Berger	7	1
Felix	Schäfer	7	1
Emily Sophie	Hartmann	7	1
...

```

projektwoche=> SELECT sVorname, sNachname, sJahrgangsstufe
FROM schuelerInnen
WHERE sJahrgangsstufe = 10;
  svorname | snachname | sjahrgangsstufe
-----+-----+-----
Lukas      | Sommer   | 10
Anna       | Schmitz   | 10
Tim        | Wagner    | 10
Laura      | Mayer     | 10
Felix      | Schuster  | 10
Emily      | Frank     | 10
Nico       | Braun     | 10
Julia      | Richter   | 10
David      | Berger    | 10
Sarah      | Schulz    | 10
Lukas      | Meier     | 10
Anna       | Wolf      | 10
Tim        | Hofmann   | 10
Laura      | Becker    | 10
Simon Ole  | Lehmann   | 10
Maria      | Koloeffel | 10
David Torsten | Hartmann | 10
Lina Sarah | Herrmann  | 10
Lukas      | Keller    | 10
Anna       | Sommer   | 10
Timon      | Schmitz   | 10
Laura Luisa | Wagner    | 10
Simon      | Huber     | 10
Julia      | Mayer     | 10
Tolga      | Can       | 10
Sarah      | Frank     | 10
Lukas      | Braun     | 10
Anna       | Richter   | 10
Timo       | Müller    | 10
Laura Mia  | Berger    | 10
(30 Zeilen)

```

```

projektwoche=> SELECT sVorname, sNachname, sJahrgangsstufe, sProfilnummer, sKlassennummer
FROM schuelerInnen
WHERE sJahrgangsstufe = 7
AND sProfilnummer = 1
AND sKlassennummer = 2;
  svorname | snachname | sjahrgangsstufe | sprofilnummer | sklassennummer
-----+-----+-----+-----+-----
Kevin Justin | Koch      | 7 | 1 | 2
Marie       | Richter   | 7 | 1 | 2
Paul        | Wolf      | 7 | 1 | 2
Lena        | Braun     | 7 | 1 | 2
Simon       | Hofmann   | 7 | 1 | 2
Laura       | Berger    | 7 | 1 | 2
Felix       | Schäfer   | 7 | 1 | 2
Emily Sophie | Hartmann  | 7 | 1 | 2
Nico        | Herrmann  | 7 | 1 | 2
Julia       | Keller     | 7 | 1 | 2
(10 Zeilen)

```

```
projektwoche=> SELECT * FROM schuelerInnen
WHERE sID IN (SELECT sID FROM nimmt_teil WHERE rolle = 'teilnehmer' AND pNummer = 2);
sid |   svorname   | snachname | sjahrgangsstufe | sprofilnummer | sklassennummer
-----+-----+-----+-----+-----+-----
 2 | Julia Maria  | Schmidt   | 7 | 1 | 1
11 | Kevin Justin | Koch      | 7 | 1 | 2
15 | Simon        | Hofmann   | 7 | 1 | 2
25 | Timo         | Huber     | 7 | 2 | 1
35 | Jonas        | Berger    | 8 | 1 | 1
45 | Nico         | Schäfer   | 8 | 1 | 2
48 | Sarah        | Keller    | 8 | 1 | 2
55 | David        | Schuster  | 8 | 2 | 1
65 | Lukas        | Becker    | 9 | 1 | 1
67 | Tim          | Schneider | 9 | 1 | 1
75 | Simon        | Mayer     | 9 | 1 | 2
77 | Felix        | Frank     | 9 | 1 | 2
85 | Timo         | Becker    | 9 | 2 | 1
95 | Felix        | Schuster  | 10 | 1 | 1
(14 Zeilen)
```

```
projektwoche=> SELECT
    ro.oBezeichnung
FROM
    raum_oder_ort ro
LEFT JOIN
    findet_statt fs ON ro.oBezeichnung = fs.ort
WHERE
    fs.ort IS NULL;
oBezeichnung
-----
002
003 Werkstatt
005
007
009
010
012
101
105
107
109
110
112
201
204
205
207
212
301
304
305
307
309
310
312
(25 Zeilen)
```

```

projektwoche=> SELECT
  p1.pNummer,
  p1.pTitel,
  l1.Leitung,
  fs1.ort
FROM
  projekt p1
JOIN
  leitet l1 ON p1.pNummer = l1.pNummer
JOIN
  findet_statt fs1 ON p1.pNummer = fs1.pNummer
JOIN
  findet_statt fs2 ON fs1.ort = fs2.ort AND fs1.pNummer != fs2.pNummer
GROUP BY
  p1.pNummer,
  p1.pTitel,
  l1.Leitung,
  fs1.ort
ORDER BY
  fs1.ort;

```

pnummer	ptitel	leitung	ort
3	Filmproduktion von Kurzfilmen	Schul	209
6	Diskutieren und Argumentieren	Leh	209

(2 Zeilen)

```

projektwoche=> SELECT sVorname, sNachname, sJahrgangsstufe, sProfilnummer, sKlassennummer
FROM schuelerInnen s
LEFT JOIN nimmt_teil nt ON s.sID = nt.sID
WHERE nt.sID IS NULL;

```

svorname	snachname	sjahrgangsstufe	sprofilnummer	sklassennummer
Sarah	Fuchs	7	2	1
Anna	Sommer	8	2	1
Lisa	Herrmann	9	1	1
Simon	Koch	9	2	1
Emily	Frank	10	1	1

(5 Zeilen)

```

projektwoche=> SELECT
  projekt.pTitel,
  angestellte.aKuerzel
FROM
  projekt
JOIN
  leitet ON projekt.pNummer = leitet.pNummer
JOIN
  angestellte ON leitet.Leitung = angestellte.aKuerzel
JOIN
  findet_statt ON projekt.pNummer = findet_statt.pNummer
WHERE
  findet_statt.datum = '2024-06-27'
AND
  findet_statt.ort = 'Schulhof';

```

ptitel	akuerzel
Kochen und Backen: Kulinarische Experimente	Fis
Diskutieren und Argumentieren	Leh
Neue Sportarten kennenlernen	Lim
Gemeinsam musizieren	Merk

(4 Zeilen)

```

projektwoche=> SELECT
  p.pNummer,
  p.pTitel,
  p.pMaxPers,
  COUNT(nt.sID) AS Anzahl_SchuelerInnen
FROM
  projekt p
LEFT JOIN
  nimmt_teil nt ON p.pNummer = nt.pNummer
GROUP BY
  p.pNummer, p.pTitel, p.pMaxPers
ORDER BY
  p.pNummer;

```

pnummer	ptitel	pmaxpers	anzahl_schuelerinnen
1	Roboter programmieren	15	21
2	Upcycling: Kreative Ideen für alte Dinge	12	17
3	Filmproduktion von Kurzfilmen	17	12
4	Kochen und Backen: Kulinarische Experimente	25	7
5	Theaterworkshop: Schauspiel und Improvisation	20	6
6	Diskutieren und Argumentieren	24	10
7	Gartenteich anlegen	28	10
8	Neue Sportarten kennenlernen	23	6
9	Gemeinsam musizieren	23	6
10	Abschlussfahrt der Klasse 1012	12	10
11	Abschlussfahrt der Klasse 1021	12	10

(11 Zeilen)

k) Auswahl und Implementierung einer Sicht auf die erstellten Daten in Form einer interaktiven Anwendung (webbasiert in PHP)

(i) Begründung und Motivation dieser Sicht

Die Motivation hinter der Entwicklung einer datenbankgestützten, interaktiven Anwendung für die Projektwoche besteht in der effizienten Verwaltung und Präsentation von Projektdaten.

Die Anwendung ermöglicht eine schnelle und effiziente Verwaltung von Projekten, was den Administrationsaufwand erheblich reduziert. Lehrer können neue Projekte eintragen. Sie haben Zugriff auf alle Projektdetails. Schüler können die verfügbaren Projekte durchsuchen und sich für Projekte anmelden. Sie können die Details der Projekte und die Verfügbarkeit der Plätze einsehen. Alle Beteiligten haben jederzeit Einblick in die Projektdetails, was die Transparenz und die Nachverfolgung der Projektauswahl erhöht. So wird die Verwaltung der Teilnehmerzahlen und die Zuweisung zu Projekten automatisiert, was menschliche Fehler minimiert.

(ii) Beschreibung der interaktiven Anwendungsmöglichkeiten

Die Anwendung bietet verschiedene interaktiven Möglichkeiten:

1. Projektverwaltung: Lehrer und Administratoren können neue Projekte erstellen und die Details wie Titel, Beschreibung, maximale Teilnehmerzahl und Leitung eingeben. Eine Übersicht über die bestehende Projekte kann erstellt werden.
2. Projektanmeldung: Schüler können alle verfügbaren Projekte einsehen und sich dafür anmelden, solange die maximale Teilnehmerzahl nicht erreicht ist.
3. Übersicht über die Projekte: eine Übersicht über die Teilnehmerzahlen, die Verteilung der Schüler auf die Projekte und andere relevante Statistiken kann generiert werden. Der Echtzeit-Status der Projekte, einschließlich der verbleibenden Plätze und der Anmeldungen, kann abgerufen werden.

(iii) Beschreibung optionaler Möglichkeiten der Erweiterung

Die Anwendung kann in verschiedenen Bereichen erweitert werden, um zusätzliche Funktionen und Verbesserungen zu bieten:

1. Erweiterte Such- und Filterfunktionen: Möglichkeit, Projekte nach mehreren Kriterien (z.B. Themenbereich, Lehrer, verfügbare Plätze) zu filtern.
2. Benachrichtigungssystem: Automatische Benachrichtigungen über Email an Schüler und Lehrer über Anmeldungen, Änderungen oder wichtige Informationen zu Projekten.
3. Feedback-System: Möglichkeit für Schüler, Feedback zu Projekten zu geben, was die Qualität der zukünftigen Projekte verbessern kann. Auch die Implementierung eines Bewertungssystems für Projekte und Lehrer/Leiter.

(iv) Erstellung einer datenbankgestützten, interaktiven Anwendung in PHP auf der Basis der erstellten Daten

Siehe Anhang.