

Datenbankpraktikum

Dozent: Hr. Herker

AG-Manager

¹Erstellt von:

Rayk von Ende

Thomas Grell

Sebastian Rösch

Inhaltsverzeichnis

<i>Beschreibung der Miniwelt Kursfahrten.....</i>	<i>2</i>
<i>Das ER-Modell.....</i>	<i>2</i>
<i>Im ER-Modell nicht dargestellte semantische Bezüge.....</i>	<i>3</i>
<i>Funktionale Abhängigkeiten.....</i>	<i>3</i>
<i>Das Relationenmodell mit allgemeinen Typen- und Schlüsselangaben.....</i>	<i>3</i>
<i>Die Transformation des ER-Modells in das Relationenmodell.....</i>	<i>3</i>
<i>Textvorlage: Aufgaben zur Modellierung.....</i>	<i>4</i>
<i>Daten der Tabellen.....</i>	<i>4</i>
<i>Anfragen an die Datenbank.....</i>	<i>6</i>
<i>SQL: Create, Insert, Drop.....</i>	<i>6</i>
<i>SQL: Anfragen.....</i>	<i>9</i>
<i>Textvorlage: Aufgaben zu Anfragen.....</i>	<i>11</i>
<i>DES: Kursfahrten-Schema.....</i>	<i>13</i>
<i>DES: Daten zu den Relationen.....</i>	<i>13</i>
<i>DES: Anfragen in Relationenalgebra.....</i>	<i>13</i>
<i>DES: Anfragen im Tupelkalkül.....</i>	<i>14</i>
<i>DES: Antworten zu den Anfragen im Tupelkalkül.....</i>	<i>15</i>
<i>DES: Anfragen im Domänenkalkül.....</i>	<i>16</i>
<i>Beschreibung und Motivation des View.....</i>	<i>17</i>
<i>Basismöglichkeiten und Erweiterung des Views.....</i>	<i>17</i>
<i>Textvorlage: Aufgaben bzgl. des Views.....</i>	<i>17</i>

1 Als Vorlage (Format, Überschriften, Aufgaben etc.) diente [1]

Didaktisch-Methodische Hinweise

Angaben zum Schwierigkeitsgrad, Umfang und Zeitbedarf der bis hierher geleisteten Arbeit:

Das Projekt AG-Planer ist inzwischen eine voll funktionsfähige Datenbank, auf die von Schülern, Lehrern und Administratoren per Browser zugegriffen werden kann. Der Arbeitsumfang und Schwierigkeitsgrad variiert zwischen den einzelnen Entwicklungsstufen. Den höchsten Arbeitsaufwand und Schwierigkeitsgrad würden wir der Entwicklung der php-Views zuordnen. Eine Übersicht zu den einzelnen Sichten findet sich im Dokument „Sichten und Anwendungsprogramm“. Der Zeitbedarf für einen einigermaßen erfahrenen php-Programmierer liegt hier bei etwa 42 Stunden. Das entspricht ca. 1,5 Stunden pro php-Skript, in der natürlich Entwicklung, Fehlersuche und Testung enthalten sind. Sollte ein ähnliches Projekt in der Schule verwirklicht werden, müssen die php-Skripte der Views durch die Lehrkraft geliefert werden. Eine andere Möglichkeit ist natürlich die Verringerung der Anwenderszenarien, die durch Views unterstützt werden sollen.

Die Erstellung der SQL-Abfragen ist eigentlich nicht von der Initialisierung der Views zu trennen. Dennoch sollen Sie hier noch einmal gesondert betrachtet werden. Denn SQL Abfragen könnten auch von Schülern entwickelt werden. Denkbar wäre eine Art Projektarbeit, in der die von den Schüler*innen entwickelten SQL-Abfragen durch die Lehrkraft in die php-Skripte aufgenommen werden. Das Finden von SQL-Abfragen zu einer bestimmten Problemstellung kann je nach Aufgabenstellung in allen möglichen Schwierigkeitsbereichen liegen. Angefangen von der Anzeige einzelner Spalten aus bestimmten Tabellen bis hin zu so komplexen Abfragen, wie sie teilweise für die in diesem Projekt verwirklichten Views nötig waren. Ein vergleichsweise geringerer Aufwand stellte die Erstellung des ER-Modells dar. Die Erstellung des ER-Modells ist eine Aufgabe, die auch Schüler*innen übernehmen könnten. Der zeitliche Aufwand zur Erstellung, Prüfung und Verbesserung sowie dem Eintragen der Kardinalitäten betrug ungefähr 5-8 Stunden. Zusätzlich mussten auch die Integritätsbedingungen und funktionale Abhängigkeiten diskutiert werden.

Das Erstellen der Datenbank mit postgresql stellt auch einen eher geringeren Aufwand dar. Die zur Erstellung der Tabellen und zum Einfügen der Daten notwendigen SQL Befehle stellen für Schüler*innen eine gute Fingerübung dar, vorausgesetzt das zugrunde liegende ER-Modell war gut durchdacht. Dennoch sollte der zeitliche Aufwand dafür nicht unterschätzt werden. Hier würde sich für eine etwaige Umsetzung im Unterricht eine kooperative Lernform anbieten. Die Schüler*innen implementieren in Teilgruppen jeweils einen Teil der SQL-Befehle, die dann zusammengeführt werden.

Angaben zur Bedeutsamkeit im Kontext des Informatikunterrichts der bis hierher geleisteten Arbeit:

Das Projekt AG-Planer greift alle wichtigen Inhalte einer Unterrichtssequenz zum Thema Datenbank auf. Das zugrunde liegende ER-Modell ist nicht zu komplex, sodass es von Schüler*innen erstellt werden kann. Sollte ein solches Projekt, eventuell etwas reduziert, im Unterricht verwirklicht werden, sollte den Schüler*innen schnell bewusst werden, wie wichtig ein gut erstelltes ER-Modell mit zugehörigen Kardinalitäten für die spätere Verwirklichung ist. Die häufig als trocken empfundene Theorie wird hier zu einer Notwendigkeit, die spätere Entwicklungsschritte erheblich strukturiert und erleichtert. Abgesehen von dem offensichtlichen Datenbankbezug wird auch das Thema Rechnernetze tangiert, zumindest falls angestrebt wird einen Zugriff auf die Datenbank über Internetbrowser zu verwirklichen. Dabei können oder müssten die php-Skripte von der Lehrkraft vorgegeben werden. Die Begriffe Server, IP-Adressen und Ports

können hier im Sinne eines Spiralprinzips thematisiert werden. Entweder dient das Thema als erster Berührungspunkt mit dem Thema oder es können Inhalte des Themas Rechnernetze gezielt wiederholt werden.

Bei welchen Inhalten ist eine didaktische Reduktion nötig bzw. angebracht? - Fakultative und Unverzichtbare Teile des AG-Planers

Die Funktionalität des AG-Planers lässt sich natürlich reduzieren, falls im Unterricht nur eine geringere Komplexität umgesetzt werden kann bzw. soll. Mögliche Gründe für eine Reduktion können zum Beispiel das geringe Leistungsniveau des Kurses oder eine zu geringe Stundenanzahl für die Unterrichtsreihe Datenbanken sein. Bei der Reduktion muss natürlich aufgepasst werden, dass das Ergebnis des Projekts immer noch ein AG-Planer ist. Manche Inhalte sind fakultativ und könnten bei Bedarf auch weggelassen werden. Andere Inhalte sind unverzichtbar. Als fakultativ würden wir die Verwirklichung der Sichten beschreiben. Die dem AG-Planer zugrundeliegende Datenbank könnte auch in postgresql durch einen Admin bearbeitet und gepflegt werden. Natürlich ist ein Benutzerzugang über einen Browser praktisch besser einsetzbar. Jedoch reicht es unter Umständen bei der Umsetzung eines Projekts im Informationsunterricht nur eine AG-Verwaltung als Datenbank via postgresql anzustreben.

Außerdem ist es möglich das ER-Modell zu reduzieren. Fakultativ ist (insbesondere, wenn man keine Sichten per php verwirklichen will) im ER-Modell die Entität Session und Termine mit den zugehörigen Relationen. Ohne durch php verwirklichte Sichten, macht es keinen Sinn die Session der Benutzer zu verwalten, um bei einem erneuten login keine Zugangsdaten mehr angeben zu müssen.

Wann eine AG stattfindet wird in der Entität Termine gespeichert, das ist natürlich schön, wenn Benutzer gucken könnten, wann eine bestimmte AG stattfinden soll oder welche AGs zu einem bestimmten Termin angeboten werden. Dennoch ist dieses Feature für einen AG-Planer nicht essenziell. Es würde ja schon ausreichen, dass der reduzierte AG-Planer AGs mit den zugehörigen AG-Leitern und Teilnehmenden Schülern verwaltet. Diese Datenbank kann dann natürlich bei Bedarf um neue Funktionalitäten erweitert werden. Unverzichtbar sind daher die Verwirklichung der Entitäten Benutzer, Andere, Schüler, Lehrer und AGs mit den jeweiligen Relationen. Dazu zählt nicht nur die Erstellung des ER-Modells, sondern auch das Erstellen von Tabellen und der insert-Dateien.

Vorschläge zur methodischen Umsetzung dieses oder eines ähnlichen Projekts im Informatikunterricht:

Es bietet sich an, die Vielzahl an Aufgaben auf mehrere Gruppen aufzuteilen und die Arbeitsergebnisse der Einzelgruppen zu einem Großen und Ganzen zusammen zu fügen. Dabei könnten die Gruppenergebnisse und Gruppenarbeiten wie in einer fiktiven Firma organisiert werden. In regelmäßigen Abständen wird der Fortschritt der einzelnen Gruppen in einer Art „Kurzmeeting“ mitgeteilt. In diesen Meetings geht es vorrangig um Erfahrungsaustausch und darum, dass die Schüler*innen das Gefühl haben zusammen an einem Projekt zu arbeiten und nicht jede Gruppe für sich. Außerdem kann die Lehrkraft so auch den Lernfortschritt beobachten. Bevor die Teilaufgaben auf einzelne Gruppen verteilt werden können muss sich die Lerngruppe auf ein gemeinsames ER-Modell einigen. Die einzelnen Gruppen könnten Vorschläge entwickeln und im Plenum vorstellen. Gemeinschaftlich wird sich dann auf ein ER-Modell geeinigt, das natürlich auch aus mehreren Vorschlägen zusammengesetzt werden kann. Nachdem das ER-Modell feststeht und

das Ziel der Projektarbeit festgelegt ist, müssen Teilaufgaben identifiziert werden und auf die Gruppen verteilt werden und dann kann die kooperative Gruppenarbeit beginnen.

Weiterführende sinnvolle Fragestellungen

Weiterführende sinnvolle Fragestellungen könnten sein: Wie sollen Schüler, Lehrer und Externe Zugriff auf die Datenbank erlangen?

Gibt es einen Schulserver, auf dem der AG-Planer abgelegt werden kann?

Beschreibung der Miniwelt AG-Manager

Leiter bieten AGs an, wählt SuS aus. Leiter können Lehrer, Schüler oder Externe sein.

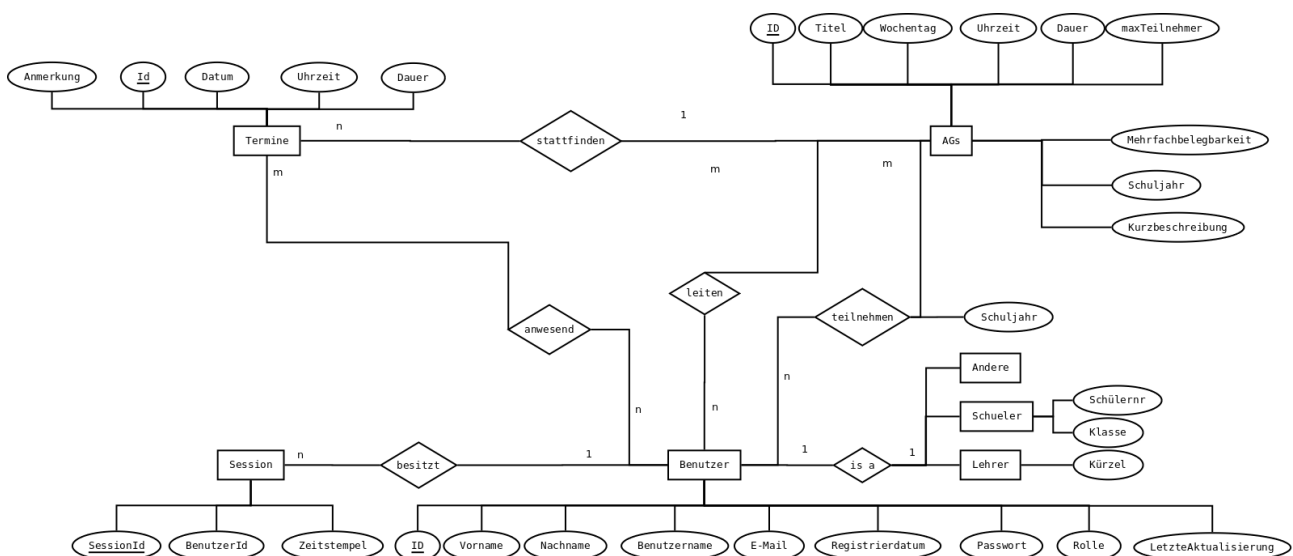
Leiter & SuS sind an den Terminen anwesend, an denen die AGs stattfinden. Sie können sich auf dem AG-Portal mit einer Session anmelden.

Leiter können dort AGs erstellen. Sie tragen die Anwesenheit dort ein.

SuS können AGs sehen. Sie können auch die Eintragung ihrer Anwesenheit prüfen.

Klassenlehrer sehen die Anwesenheit und können bei entsprechend häufiger Anwesenheit einen Vermerk auf dem Zeugnis notieren.

Das ER-Modell



7. Klasse hat immer 2 Flexzeiten (Zeit in der Hausaufgaben gemacht werden können oder für ein beliebiges Fach gelernt werden kann) und 1 AG
8. Klasse kann wählen: 1 FZ & 2 AGs oder 2 FZ und 1 AG
Viele AGs nur in einmal in 7./8. Klasse belegbar, damit auch andere zum Zuge kommen (z.B. Lego-Roboter)
Chor: Schülerzeitung etc. mehrfach belegbar.
AG-Slots: Di und Do, 90min
Flexzeit-Slots: 7, Dauer ?
AG-Teilnehmerzahl: 10-12 (Kochen: 9, Chor: viele)
Klassen können für AGs gemischt werden

Im ER-Modell nicht dargestellte semantische Bezüge

Statische Integritätsbedingungen:

Domain- bzw. Attributbedingungen:

- AGs müssen eine positive Dauer haben
- Die Teilnehmerzahl von AGs muss positiv sein
- Termine müssen eine positive Dauer haben
- Schuljahre müssen in der Form XXXX/YY angegeben werden
- Benutzer können nur die Rollen admin, viewer, editor oder user haben

Relationenbedingungen:

- Schlüsselbedingungen:
 - in Benutzer, Lehrer und Schüler: Benutzer_ID
 - in AGs: AG_ID
 - in Termine: Termin_ID
 - in Sessions: Session_ID
 - in Leiten: AG_ID, Benutzer_ID
 - in Anwesenheit: Termin_ID, Benutzer_ID
 - in Teilnahme: AG_ID, Benutzer_ID
- Aggregatbedingungen: keine
- Rekursive Bedingungen: keine

Referentielle Bedingungen:

- Ein Lehrer kann nur angelegt werden, wenn er bereits als Benutzer existiert
- Ein Schüler kann nur angelegt werden, wenn er bereits als Benutzer existiert
- Nur existierende Nutzer können Sessions haben
- Nur existierende AGs können Termine haben
- Nur existierende AGs können nur von existierenden Benutzern geleitet werden
- Nur existierende Nutzer können nur an existierenden Terminen anwesend sein
- Nur existierende Nutzer können nur an existierenden AGs teilnehmen

Dynamische

Zu Schuljahresbeginn werden alle Schüler und Lehrer der Schule mit den in der Datenbank vorhandenen abgeglichen. Falls sie noch an der Schule sind, wird das Datum ihrer letzten Aktualisierung neu gesetzt. Auf diese Weise können „Karteileichen“ gefunden und entfernt werden.

Funktionale Abhängigkeiten

Benutzer:

Benutzer_ID \rightarrow {Benutzername, Passwort, Vorname, Nachname, E-Mail, Rolle, Registrierdatum, aktiv, letzte_Aktualisierung}

Benutzername \rightarrow {Benutzer_ID, Passwort, Vorname, Nachname, E-Mail, Rolle, Registrierdatum, aktiv, letzte_Aktualisierung}

E-Mail \rightarrow {Benutzer_ID, Benutzername, Passwort, Vorname, Nachname, E-Mail, Rolle, Registrierdatum, aktiv, letzte_Aktualisierung}

Lehrer:

Benutzer_ID \rightarrow Kürzel

Kürzel \rightarrow Benutzer_ID

Schüler:

Benutzer_ID \rightarrow {Schülernummer, Klasse}

Schülernummer \rightarrow {Benutzer_ID, Klasse}

AGs:

AG_ID \rightarrow {Titel, Wochentag, Uhrzeit, Dauer, maxTeilnehmer, Mehrfachbelegbarkeit, Schuljahr, Kurzbeschreibung}

Termine:

Termin_ID \rightarrow {Zeitstempel, Dauer, Anmerkung, AG_ID}

AG_ID \rightarrow {Termin_ID, Zeitstempel, Dauer, Anmerkung}

Sessions:

Session_ID \rightarrow {Loginzeitstempel, Benutzer_ID}

Benutzer_ID \rightarrow {Session_ID, Loginzeitstempel}

Leiten:

trivial

Anwesenheit:

trivial

Teilnahme:

{AG_ID, Benutzer_ID} → Schuljahr

Das Relationenmodell mit allgemeinen Typen- und Schlüsselangaben

users (user_id, username, password, firstname, lastname, email, roll, registrationtime, enabled, last_update)

teachers (user_id, shorthand)

students (user_id, studentnumber, class)

wgs (wg_id, title, day, time, duration, max_num, multiple, schoolyear, description)

events (event_id, datetime, duration, annotation, wg_id)

sessions (session_id, logintime, user_id)

lead (wg_id, user_id)

present (event_id, user_id)

participate (wg_id, user_id, schoolyear)

Deutsche Übersetzung:

Benutzer (Benutzer_ID, Benutzername, Passwort, Vorname, Nachname, E-Mail, Rolle, Registrierdatum, aktiv, letzte_Aktualisierung)

Lehrer (Benutzer_ID, Kürzel)

Schüler (Benutzer_ID, Schülernummer, Klasse)

AGs (AG_ID, Titel, Wochentag, Uhrzeit, Dauer, maxTeilnehmer, Mehrfachbelegbarkeit, Schuljahr, Kurzbeschreibung)

Termine (Termin_ID, Zeitstempel, Dauer, Anmerkung, AG_ID)

Sessions (Session_ID, Loginzeitstempel, Benutzer_ID)

Leiten (AG_ID, Benutzer_ID)

Anwesenheit (Termin ID, Benutzer ID)

Teilnahme (AG_ID, Benutzer_ID, Schuljahr)

Die Transformation des ER-Modells in das Relationenmodell

Transformation der Entitätstypen AGs, Termine, Benutzer, Sessions:

1:1 aus ER-Modell übernommen.

Transformation der Subtypbeziehungen Lehrer, Schueler, Andere:

Jeder Subtyp wird zu einem neuen Relationsschema, das alle Attribute des jeweiligen Subtyps als auch den Primärschlüssel des Supertyps als Fremdschlüssel enthält. Der Fremdschlüssel wird zum Primärschlüssel des Subtyps. Da „Andere“ keine eigenen Attribute besitzt, ist die Relation nicht notwendig zu implementieren. Jeder Benutzer, der kein Schüler oder Lehrer ist, ist automatisch ein „Anderer“.

Transformation „besitzt“:

Die einseitig optionale Beziehung (ein Benutzer muss keine Session haben) wird in den Entitätstyp „Session“ durch Hinzunahme der BenutzerID integriert.

Semantikverlust, da Optionalität nicht mehr erkennbar. Integritätsbedingung (eine Session muss einen Benutzer haben) muss hinzugefügt werden.

Transformation „anwesend“:

Es handelt sich um einen beidseitig optionalen m:n Beziehungstyp. Er wird in eine eigene Relation „Anwesenheit“ transformiert, die die Fremdschlüssel der beteiligten Entitäten (Termine und Benutzer) enthält, die gemeinsam den Primärschlüssel bilden.

Semantikverlust, da Optionalität nicht mehr erkennbar ist. Integritätsbedingung (ein Benutzer kann nicht an mehreren Terminen gleichzeitig anwesend sein) muss hinzugefügt werden.

Transformation „stattfinden“:

Der n:1 Beziehungstyp (Termin – AG) wird in den Entitätstyp Termine durch Hinzunahme der AG_ID integriert.

Semantikverlust, weil einseitige Optionalität (eine AG muss nicht zwangsläufig einen Termin haben) nicht mehr erkennbar. Integritätsbedingung notwendig, weil ein Termin genau einer AG zugeordnet werden muss (NOT NULL REFERENCES wgs(wg_id)).

Transformation „leiten“:

Es handelt sich um einen einseitig optionalen m:n Beziehungstyp. Er wird in eine eigene Relation „Leiten“ transformiert, die die Fremdschlüssel der beteiligten Entitätstypen (AGs und Benutzer) enthält, die gemeinsam den Primärschlüssel bilden.

Semantikverlust, da Optionalität (nicht jeder Benutzer muss eine AG leiten) nicht mehr erkennbar. Keine Integritätsbedingungen notwendig.

Transformation „teilnehmen“:

Es handelt sich um einen beidseitig optionalen m:n Beziehungstyp. Er wird in eine eigene Relation „Teilnahme“ transformiert, die die Fremdschlüssel der beteiligten Entitätstypen (AGs und Benutzer) enthält, die gemeinsam den Primärschlüssel bilden.

Semantikverlust, da Optionalität (nicht jeder Benutzer muss an einer AG teilnehmen, eine AG muss keine Teilnehmer haben) nicht mehr erkennbar.

Integritätsbedingung notwendig, um „maxTeilnehmer“ nicht zu überschreiten.

Textvorlage: Aufgaben zur Modellierung²

Szenario:

Die Schule plant die Einführung einer Datenbank zur Verwaltung von AGs. Leiter bieten AGs an, wählen SuS aus. Leiter können Lehrer, Schüler oder Externe sein.

Leiter & SuS sind an den Terminen anwesend, an denen die AGs stattfinden. Sie können sich auf dem AG-Portal mit einer Session anmelden.

Leiter können dort AGs erstellen. Sie tragen die Anwesenheit dort ein.

SuS können AGs sehen. Sie können auch die Eintragung ihrer Anwesenheit prüfen.

Klassenlehrer sehen die Anwesenheit und können bei entsprechend häufiger Anwesenheit einen Vermerk auf dem Zeugnis notieren.

Aufgabe 1)

Setzt euch zu dritt zusammen und überlegt euch ein ER Modell und beschriftet die Kardinalitäten.

Aufgabe 2)

a) Überführt euer Modell in ein Relationenmodell.

b) Überlegt euch Zusammenhänge, die gesondert geprüft werden müssen und sich nicht aus euren Tabellen ergeben. Kann es zu Überschneidungen oder Widersprüchen kommen?

Aufgabe 3)

Teilt euch auf und trifft euch mit einem_r Schüler_In einer anderen Gruppe und stellt euer Modell vor und vergleicht es mit dem Modell der anderen Gruppen.

Aufgabe 4)

Verbessert euer Modell nach den Erkenntnissen aus Aufgabe 3).

Daten der Tabellen

Die Tabelle ist zu finden in: ../modell/ag_manager-Daten.csv

Anfragen an die Datenbank

```
--*****  
--Daten mit SELECT auswählen  
--*****  
--Gesucht ist jeweils eine Tabelle aller ...
```

² Stark angelehnt an [1].

- 1. Arbeitsgemeinschaften.
- 2. Benutzerdaten (Benutzername, Vorname, Nachname).
- 3. AGs (Titel, Wochentag) dieses Schuljahres.
- 4. Klassen aufsteigend sortiert.
- 5. Benutzer (Vorname, Nachname, Benutzername) mit "Nina" im Vornamen.
- 6. AGs (Titel, Tag), die nicht am Mittwoch stattfinden.
- 7. AGs (Id, Titel) mit dem groß oder klein geschriebenen Wort "Theater" in der Beschreibung.
- 8. Events (Id, Datum, Bemerkung) der letzten 7 Tage absteigend nach Datum sortiert.

--*****

--Daten mit SELECT auswählen unter Verwendung von Joins:

--*****

--Gesucht ist jeweils eine Tabelle aller ...

- 9. Lehrer (Vorname, Nachname, Kürzel).
- 10. Schüler (Vorname, Nachname, Klasse, Schülernummer).
- 11. Veranstaltungen (= Events) (Datum, Bemerkung) der Näh-AG seit Beginn des Jahres.
- 12. Schüler (Vorname, Nachname) mit Tag und Zeit der AGs an denen sie teilnehmen.
- 13. User (Id, Vorname, Nachname) die innerhalb des letzten Jahres keine Session hatten.

--*****

--Daten mit SELECT auswählen unter Verwendung von Joins, Subqueries

--und Aggregatfunktionen:

--*****

- 14. Gesucht ist die Gesamtzahl an AG-Plätzen in AGs mit beschränkter Platzzahl
- 15. Gesucht ist eine Tabelle der Gesamtdauern aller AG-Termine pro AG (AG-Id), die bisher stattgefunden haben.
- 16. Gesucht ist eine Tabelle aller AGs mit mehr als einem Leiter.
- 17. Gesucht ist die AG mit den meisten eingeschriebenen Teilnehmern.
- 18. SuS (Nachname,Rolle), die an einer AG teilnehmen.
- 19. Gesucht ist eine Tabelle aller AGs (Titel, Schuljahr, Beschreibung) der Schülerin mit dem Benutzernamen "schneeflocke". Die Sortierung erfolgt absteigend nach Schuljahr und aufsteigend nach Titel.
- 20. Gesucht ist eine Tabelle aller Schüler der Klasse 7a, in welcher die AGs und die Anzahl der besuchten Veranstaltungen (= Events) seit Schuljahresbeginn 2020 aufgelistet werden. Die Sortierung erfolgt nach Nachname, Vorname und Titel aufsteigend.
- 21. Gesucht ist eine Tabelle aller Schüler (Vorname, Nachname) mit Tag und Zeit und der Anzahl der AGs an denen sie teilnehmen, die am selben Tag zur selben Zeit stattfinden.
- 22. Gesucht ist eine Tabelle der prozentualen Teilnahme der SuS an der AG Basketball absteigend geordnet nach Teilnahme.
- 23. SuS (Name, Vorname), die an allen Terminen einer AG (Titel) anwesend waren aufsteigend sortiert nach AG-Titel.

SQL: Create, Insert, Drop

(Die vollständigen Daten sind zu finden in: ../sql/)

CREATE DOMAIN schoolyear AS VARCHAR(7) CHECK (value SIMILAR TO '[0-9]{4}/[0-9]{2}');

```
CREATE TYPE weekday AS ENUM ('Montag', 'Dienstag', 'Mittwoch', 'Donnerstag', 'Freitag', 'Samstag', 'Sonntag');
```

```
CREATE TABLE IF NOT EXISTS users (  
  user_id          SERIAL PRIMARY KEY,  
  firstname        VARCHAR(30) NOT NULL,  
  lastname         VARCHAR(30) NOT NULL,  
  username         VARCHAR(30) UNIQUE NOT NULL,  
  password         VARCHAR(255) NOT NULL,  
  email            VARCHAR(100),  
  roll             VARCHAR(6) DEFAULT 'user' CHECK (roll='admin' OR roll='viewer' OR roll='editor' OR roll='user'),  
  registrationtime TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,  
  enabled          BOOLEAN NOT NULL DEFAULT TRUE,  
  last_update      TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL  
);
```

```
CREATE OR REPLACE FUNCTION update_timestamp()  
RETURNS TRIGGER AS $$  
BEGIN  
  NEW.last_update = now();  
  RETURN NEW;  
END;  
$$ language 'plpgsql';
```

```
/*  
CREATE OR REPLACE FUNCTION chk_max_num(id INTEGER)  
RETURNS BOOLEAN AS $$  
BEGIN  
  RETURN (SELECT COUNT(user_id) FROM wgs WHERE id=wgs.wg_id)<(SELECT max_num FROM wgs WHERE wgs.wg_id=id);  
END;  
$$ language 'plpgsql';  
*/
```

```
CREATE TRIGGER last_userupdate BEFORE UPDATE ON users FOR EACH ROW EXECUTE  
PROCEDURE update_timestamp();
```

```
CREATE TABLE IF NOT EXISTS teachers (  
  shorthand VARCHAR(10) NOT NULL UNIQUE,  
  user_id INTEGER PRIMARY KEY NOT NULL REFERENCES users(user_id) ON DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS students (  
  studentnumber VARCHAR(20) NOT NULL UNIQUE,  
  class VARCHAR(10) NOT NULL,
```

```
user_id INTEGER PRIMARY KEY NOT NULL REFERENCES users(user_id) ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS sessions (
session_id VARCHAR(255) PRIMARY KEY,
user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE ON UPDATE
CASCADE,
logintime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TABLE IF NOT EXISTS wgs (
wg_id SERIAL PRIMARY KEY,
title VARCHAR(100) NOT NULL,
day weekday NOT NULL,
time TIME NOT NULL,
duration INTEGER DEFAULT 45 CHECK(duration>=0),
max_num INTEGER DEFAULT 0 CHECK (max_num>=0),
multiple BOOLEAN DEFAULT TRUE,
schoolyear schoolyear,
description TEXT
);
```

```
CREATE TABLE IF NOT EXISTS events (
event_id SERIAL PRIMARY KEY,
date DATE DEFAULT CURRENT_DATE,
time TIME DEFAULT CURRENT_TIME,
duration INTEGER DEFAULT 45 CHECK (duration>=0),
annotation TEXT,
wg_id INTEGER NOT NULL REFERENCES wgs(wg_id) ON DELETE CASCADE ON UPDATE
CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS lead (
wg_id INTEGER NOT NULL REFERENCES wgs(wg_id) ON DELETE CASCADE ON UPDATE
CASCADE,
user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE ON UPDATE
CASCADE,
PRIMARY KEY (wg_id, user_id)
);
```

```
CREATE TABLE IF NOT EXISTS present (
event_id INTEGER NOT NULL REFERENCES events(event_id) ON DELETE CASCADE ON UPDATE
CASCADE,
user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE ON UPDATE
CASCADE,
PRIMARY KEY (event_id, user_id)
);
```

```

CREATE TABLE IF NOT EXISTS participate (
    wg_id INTEGER NOT NULL REFERENCES wgs(wg_id) ON DELETE CASCADE ON UPDATE CASCADE,
    user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE ON UPDATE CASCADE,
    schoolyear schoolyear,
    PRIMARY KEY (wg_id, user_id)
);

```

```

GRANT ALL PRIVILEGES ON DATABASE ag_manager TO ag_admin;
GRANT USAGE ON SCHEMA public TO ag_admin;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO ag_admin;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO ag_admin;

```

```

GRANT CONNECT ON DATABASE ag_manager TO ag_user;
GRANT USAGE ON SCHEMA public TO ag_user;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO ag_user;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO ag_user;

```

```

INSERT INTO users (username,password,firstname,lastname,enabled,roll,email)
VALUES
('admin','$2y$10$g5rEA6MdkIm.YvlgvbrDkucvEytY8bAhUtinpv4Sc.SQ0SG88JHke','Adam','Adler',TRUE,
'admin',''),
('sonnenschein','$2y$10$sQYHCXBpamR9d.p//
7j6Fe9a9mtJpsp4tB66ftr1drr9jhxxk8jA.','Edwin','Edison',TRUE,'editor',''),
('herbstnebel','$2y$10$dD2.3nwlnewnqX/MgpczjOJgMka2k7y6ijKMQoBNvUOqyBw4JG/
XO','Viki','Vigor',TRUE,'viewer',''),
('frühlingsblume','$2y$10$CjCqCHAmXn4Ymlc0VaA11eBkeOMS8qwuLtc089BKSoTmrV6nPGqhW','Usch
i','Usus',TRUE,'user',''),
('schneeflocke','$2y$10$frQYeAqITH5WFMiyswS/x.MHb/
7uf87WMikkQI5wX5PhCa97pI7Va','Nina','Nadel',TRUE,'user',''),
('luftballon','$2y$10$10aSVrGbeHXmMUzufaWtG.u56DQCUH7wowtndocYmsGPVjM0V2A3G','Gregor','
Müller',TRUE,'user',''),
('weintraube','$2y$10$.M/0kk/NXMz5ODJJtOQRPeDccDxQc82V3IkwEsucBXl/
nbnCvI432','Walter','Dremel',TRUE,'user',''),
('eisenhut','$2y$10$xEB//3wlrPGCJzgbeRhVB.u5be9agVWz.Of/
DM6lhAzVUY2mrc4xm','Fidora','Flanell',TRUE,'user',''),
('rosskastanie','$2y$10$4JB9hFNJCL2WmN0dgiId5ewWJejycSRnfjKZLKRJQlBsp0iEMUHJK','Liselotte','
Müller',TRUE,'user',''),
('silberstreif','$2y$10$SOIMoE7IM2vWBxv2y4z5H.9KcIXOTFHoOZWYGbtAGGhIGFL1vdRVm','Jana','J
ensen',TRUE,'editor',''),
('ostseebrise','$2y$10$7noloL/
vpWM7PQeZZClyZ.YLvYv7sy9Z1b5JyZzOTz748rHuRsUm.','Bernd','Brot',TRUE,'editor',''),
('sabt','$2y$10$.yPt1AZLQbA/
cp0FokIW1.PIy3oYL1LHupkisXT4HjVZyJuJlMrNK','Stephan','Abt',TRUE,'user',''),
('lauda','$2y$10$XFvKMUEiTTb9CEiji26uc.4meRUfinRLFNrCv8W0TLGXqRZntOiB2','Laura','Dresdner',
TRUE,'user','lauradresdner@ag-manager.de'),

```

('selowe','\$2y\$10\$G4Hsq8dL6F.6i97kWmpZ1e9XQHbRP10igwFPObuy7H2xgjF14tuiG','Sebastian','Lowe',
TRUE,'user','sebastianlowe@ag-manager.de'),
('dodec','\$2y\$10\$WII5NIye02XWTm2QQZf3KOHAAZzXjSEGt0nh0Eh7KhgKLtnCIMxNTW','Doreen','Dec
ker',TRUE,'user','doreendecker@ag-manager.de'),
('jewex','\$2y\$10\$RJaPziCDlB/uQCPQW/TYme69Ckesu2t8csmrfphtvSX/
LgB1ReXQ6','Jens','Wexler',TRUE,'user','jenswexler@ag-manager.de'),
('laumue','\$2y\$10\$unizPYhH3IH7vALCtWBbeVFJD8bTvP/jI42Gb.rWNCQP/
nqxiJD6','Laura','Müller',TRUE,'user','derwendler@istegal.de'),
('kamue','\$2y\$10\$0zzVmR8.d5ZiY9w2WWEJwesQQmtKRkY2OKGQaLwejnacBVox1Trba','Katja','Müller',
TRUE,'user','katjamueller@ag-manager.de'),
('frappa','\$2y\$10\$w1IwaKUV04YuSh3vSJ94T.jK/Q4AkyfNH9H1okypv/
umW2N4XDtGS','Frank','Zappa',TRUE,'user','frankzappa@ag-manager.de'),
('herrbaer','\$2y\$10\$.1ysq2BbcL3Z1yiuzWyNQoEgZUPLR/
K0LUjAbUr9zq0CH6rRq4AG','Herbert','Bähr',TRUE,'user','herbertbaehr@ag-manager.de'),
('almey','\$2y\$10\$Wk5jg0PQP7eQ.NOWxUMA2upeK3KNMW44XOAGC2A9mLkZhinWm2TAC','Alexan
der','Meyer',TRUE,'user','alexandermeyer@ag-manager.de'),
('hamarb','\$2y\$10\$tsSNwqtBeiUaA9mEOUoX6eceNRyeX9JOADfEuDXrc0DoVA2yzZT4y','Hamudi','Harb',
TRUE,'user','hamudiharb@ag-manager.de'),
('habarb','\$2y\$10\$n6lA913lIFcbOb2vTu1uh.6tL3gjTnwrH7bghNfkEx0DUh0Xytppte','Habudi','Harb',TRUE,'
user','habudiharb@ag-manager.de'),
('wischam','\$2y\$10\$9Pa6H8Z4oz4pXg0QCE.S3u/uGTzGTZZJPWDi4AW9DWBUTEaqYViUG','Wisaam
Aram','Cham',TRUE,'user','wisaamcham@ag-manager.de'),
('chutri','\$2y\$10\$wnWJKe0apgfc8gUElIuSyekj5H9fU/jqlmybW0sDZ37A.p5.5wLK','Chu
Thành','Tri',TRUE,'user','chutri@ag-manager.de'),
('tonnhi','\$2y\$10\$SEst1EDJFQlVndtvBSUJeb60ZpsyvUuBEi5MWFIPVpCgN7/Um9rO','Tôn
Ngoc','Nhi',TRUE,'user','tonnhi@ag-manager.de'),
('wiskwi','\$2y\$10\$fnDYph4iNLa85m.IAxaUzePWqu27EKVyF3HDRFc9i.nw25DiVvxMy','Wisława','Kwiat
kowska',TRUE,'user','wislawakwiatkowska@ag-manager.de'),
('nazajac','\$2y\$10\$3nj8Y3t.D6C6M4zh5BP1ju5Sj.ZwYQiN5lwAzHITnp0N0/
XW7hL9W','Natasza','Zajac',TRUE,'user','nataszazajac@ag-manager.de'),
('wawis','\$2y\$10\$p15XWJubR0EL1.T6XeX5hua8D1Txa1SyrM9AKG.vjTWeKMmz2LY8W','Walery','Wiśn
iewski',TRUE,'user','walerywisniewski@ag-manager.de'),
('wolwi','\$2y\$10\$hsmDYOFDim33s09ayFTbSuFbVcU4zqZc7nkB5AezSZMhGrb9EoGDC','Wolfgang','Wo
wereit',TRUE,'user','wolfgangworeit@ag-manager.de'),
('jabab','\$2y\$10\$64r33U1khW/VCzkVplGpcuyd/
mRlHrRrEj35X6gSYrH8S0SFF5Xy','Jakub','Jabłoński',TRUE,'user','jakubjablonski@ag-manager.de'),
('wiszc','\$2y\$10\$OjRWYz/iIGq2hCsXuGhGI.pBt.O/mZbinoyrnaFieSCVdLAPVi/
XK','Wisla','Szczepańska',TRUE,'user','wisiaszczepanska@ag-manager.de'),
('admin2','\$2y\$10\$WgKWY7Nt.tcMGAY.qHt6COMQQY7tghjEjg4Cn3CxFBtFLdyzHo6.y','Bertha','Bentz',
TRUE,'admin','berthabentz@ag-manager.de'),
('admin3','\$2y\$10\$Jy3JT9Q1.DVdrxJpSSVife/k0UQPMmhFXLutjehTkkHB/
psKgx2Sq','Claus','Clever',TRUE,'admin','clausclever@ag-manager.de'),
('regenhimmel','\$2y\$10\$PLOSrH4R67xyHYW84l6RMeSZ72NRXC9EJCCI1GVMXgsgjT/
SQCvrK','Moritz','Macker',TRUE,'editor','moritzmacker@ag-manager.de'),
('winterweite','\$2y\$10\$vjwhQBJL67Vpl4BYapVBF.nUY9ws.YrDjWFtdQ2pa/
1moOWn0dOhq','Elisabeth','Engel',TRUE,'editor','elisabethengel@ag-manager.de'),

```
(
'moossteinbrech','$2y$10$SSziJY2i4cKKzve/
BxDAK.IECctgUGznDUVhBPiajIsHEWkMX5lIC','Eduard','Möhrike',TRUE,'editor','eduardmoehrike@ag-
manager.de'),
('kaulquappe96','$2y$10$e5OMSqQPsDIUEc0g/
nAI.eqMsxMRlRXhS8MYPiAEQKibWrvplzLy','Hannah','Merkur',TRUE,'editor','hannahmerkur@ag-
manager.de'),
('rotbaum','$2y$10$x9nOWCKrOdMSNXoz3CN03u0JztJ3X2sjBA0Y9xdjoVf9PuHZkq2bO','Waldemar','W
acker',TRUE,'viewer','waldemarwacker@ag-manager.de'),
('kartoffelkraut','$2y$10$v09W5342kMVMocsQz2nuEOC/dNC7joy.CmeH51N1BtB97/
lAVUhT2','Xaver','Xanthopoulos',TRUE,'viewer','xaverxanthopoulos@ag-manager.de'),
('stoppelfeld','$2y$10$YHc4lUuAWOz9Sjgsx7MvIugdoFssrZP4LD9nNcI82x.fPrX/
aLr8u','Yannis','Yoghurt',TRUE,'viewer','yannis yoghurt@ag-manager.de');
```

```
INSERT INTO wgs (title, day, time, duration, max_num, multiple, schoolyear, description)
VALUES ('Nähen','Montag','15:00:00',90,10,FALSE,'2020/21','Wir nähen alle einen Teddybären.'),
('Schach','Montag','15:00:00',90,10,TRUE,'2020/21','Klassisches Schachspiel für jung und alt. Beginn: 29.
März 2021'),
('Karate','Mittwoch','15:30:00',90,0,TRUE,'2020/21','Shotokan-Karate für die persönliche Entwicklung.'),
('Technik','Freitag','14:00:00',90,6,TRUE,'2020/21','Auf- und Abbau der Theatertechnik. Betreuung während
der Proben und Aufführungen.'),
('Chemie','Dienstag','16:00:00',90,8,TRUE,'2019/20','Experimente, die Puffen und Stinken.'),
('Basketball','Mittwoch','16:00:00',90,20,TRUE,'2020/21','Was gibt es da zu erklären?');
```

```
INSERT INTO teachers (user_id, shorthand) VALUES
(1,'ADL'),
(2,'EDI'),
(3,'VIG'),
(10,'JEN'),
(11,'BRO'),
(33,'BNZ'),
(34,'CLR'),
(35,'MAC'),
(39,'WCK'),
(40,'XTS'),
(41,'YOG');
```

```
INSERT INTO students (class, studentnumber, user_id) VALUES
('7a','usus01',4),
('7b','nina01',5),
('7a','grmü01',6),
('7b','wadr01',7),
('8a','fifl01',8),
('8a','limü01',9),
('8b','abt01',12),
('8b','dresdner01',13),
('7a','lowe01',14),
('7a','decker01',15),
('7a','wexler01',16),
```

```
(
'7a','mueller01',17),
('7a','mueller02',18),
('8a','zappa01',19),
('7a','baehr01',20),
('7a','meyer01',21),
('7a','harb01',22),
('7a','harb02',23),
('7a','cham01',24),
('7a','tri01',25),
('7a','nhi01',26),
('7a','kwiatkowska01',27),
('7a','zajac01',28),
('7b','wisn01',29),
('7b','wowi01',30),
('7b','jabl01',31),
('7b','szcz01',32);
```

```
INSERT INTO lead (user_id, wg_id) VALUES
(4,2),
(2,3),
(11,6),
(10,4),
(1,1),
(35,1),
(36,2),
(38,5);
```

```
INSERT INTO events (date, time, duration, annotation, wg_id) VALUES
('2021-03-22','15:00:00',90,'Gemeinsames Kennenlernen',1),
('2021-03-28','15:00:00',90,'Schnittmuster vorbereiten',1),
('2021-04-05','15:00:00',60,'Stoff ausschneiden',1),
('2021-03-24','16:00:00',90,'Verhaltensregeln und Kennenlernen',3),
('2021-03-31','16:00:00',90,'Wurftechniken',6),
('2021-04-07','16:00:00',90,'Taktik 1',6),
('2021-04-14','16:00:00',90,'Taktik 2',6),
('2021-04-21','16:00:00',90,'Fairplay',6),
('2021-03-31','15:30:00',90,'Taikyoku Shodan',3),
('2021-05-01','16:00:00',90,'Feiertags-AG',6),
('2021-04-28','15:30:00',90,'Eintragung am WE nach dem Termin',3),
('2021-05-03','15:00:00',90,'Datum in der Zukunft',1),
('2021-04-26','15:00:00',90,' ',2),
('2021-04-30','11:00:00',60,'Fand heute früher statt wegen Elternsprechtag',4);
```

```
INSERT INTO participate (user_id, wg_id, schoolyear) VALUES
(4,1,'2020/21'),
(4,2,'2019/20'),
(4,6,'2020/21'),
(5,3,'2020/21'),
```


(7,6,'2020/21'),
(6,6,'2020/21'),
(5,6,'2020/21'),
(9,3,'2020/21'),
(8,3,'2020/21'),
(5,1,'2020/21'),
(5,2,'2020/21'),
(6,2,'2020/21'),
(4,4,'2020/21'),
(7,4,'2020/21'),
(12,6,'2020/21'),
(15,6,'2020/21'),
(8,6,'2020/21'),
(14,6,'2020/21'),
(9,6,'2020/21'),
(26,6,'2020/21'),
(13,6,'2020/21'),
(19,6,'2020/21'),
(30,3,'2020/21'),
(20,3,'2020/21'),
(12,3,'2020/21'),
(15,3,'2020/21'),
(12,1,'2020/21'),
(12,2,'2020/21'),
(12,4,'2020/21'),
(20,4,'2020/21'),
(24,4,'2020/21'),
(15,4,'2020/21'),
(19,2,'2020/21'),
(8,1,'2020/21'),
(23,1,'2020/21'),
(31,1,'2020/21'),
(22,3,'2020/21'),
(21,3,'2020/21'),
(17,3,'2020/21'),
(28,3,'2020/21'),
(25,3,'2020/21');

INSERT INTO present (event_id,user_id) VALUES

(1,4),
(2,4),
(3,4),
(4,5),
(5,7),
(5,6),
(5,5),
(5,4),
(6,6),

(6,5),
(6,4),
(7,7),
(7,6),
(7,5),
(7,4),
(8,7),
(8,6),
(8,5),
(8,4),
(9,8),
(9,9),
(9,5),
(10,12),
(10,15),
(10,7),
(10,13),
(10,8),
(10,14),
(10,6),
(10,9),
(10,5),
(10,26),
(10,4),
(10,19),
(11,12),
(11,20),
(11,15),
(11,8),
(11,22),
(11,21),
(11,17),
(11,9),
(11,5),
(11,25),
(11,30),
(11,28),
(12,12),
(12,8),
(12,23),
(12,31),
(12,5),
(12,4),
(13,12),
(13,6),
(13,5),
(13,19),
(14,12),

(14,20),
(14,24),
(14,15),
(14,7);

DROP TABLE IF EXISTS sessions;
DROP TABLE IF EXISTS teachers;
DROP TABLE IF EXISTS students;
DROP TABLE IF EXISTS lead;
DROP TABLE IF EXISTS present;
DROP TABLE IF EXISTS participate;
DROP TABLE IF EXISTS events;
DROP TABLE IF EXISTS wgs;
DROP TRIGGER IF EXISTS last_userupdate ON users;
DROP TABLE IF EXISTS users;
DROP FUNCTION update_timestamp();
DROP DOMAIN schoolyear;
DROP TYPE weekday;

SQL: Anfragen

--*****

--Daten mit SELECT auswählen

--*****

--Gesucht ist jeweils eine Tabelle aller ...

--1. Arbeitsgemeinschaften:

SELECT * FROM wgs;

--2. Benutzerdaten (Benutzername, Vorname, Nachname):

SELECT username, firstname, lastname FROM users;

--3. AGs (Titel, Wochentag) dieses Schuljahres:

SELECT title, day FROM wgs WHERE schoolyear='2020/21';

--4. Klassen aufsteigend sortiert:

SELECT DISTINCT class FROM students ORDER BY class ASC;

--5. Benutzer (Vorname, Nachname, Benutzername) mit "Nina" im Vornamen:

SELECT firstname, lastname, username FROM users
WHERE firstname LIKE '%Nina%';

--6. AGs (Titel, Tag), die nicht am Mittwoch stattfinden:

SELECT title, day
FROM wgs
WHERE NOT day = 'Mittwoch';

--7. AGs (Id, Titel) mit dem groß oder klein geschriebenen Wort "Theater"

```
--in der Beschreibung:
SELECT wg_id, title
FROM wgs
WHERE description SIMILAR TO '%(T|t)heater%';
```

```
--8. Events (Id, Datum, Bemerkung) der letzten 7 Tage absteigend nach Datum sortiert:
SELECT event_id, date, annotation FROM events
WHERE date >= (now() - INTERVAL '7 days')
ORDER BY date DESC;
```

```
--*****
```

```
--Daten mit SELECT auswählen unter Verwendung von Joins:
```

```
--*****
```

```
--Gesucht ist jeweils eine Tabelle aller ...
```

```
--9. Lehrer (Vorname, Nachname, Kürzel):
SELECT firstname, lastname, shorthand
FROM users AS u, teachers AS t
WHERE u.user_id=t.user_id;
```

```
--10. Schüler (Vorname, Nachname, Klasse, Schülernummer):
SELECT firstname, lastname, class, studentnumber
FROM users AS u, students AS s
WHERE u.user_id=s.user_id;
```

```
--11. Veranstaltungen (= Events) (Datum, Bemerkung) der Näh-AG seit Beginn des Jahres:
SELECT date, annotation FROM events, wgs
WHERE title LIKE 'Nähen' AND wgs.wg_id=events.wg_id
AND date>='2021-01-01';
```

```
--12. Schüler (Vorname, Nachname) mit Tag und Zeit der AGs an denen sie teilnehmen:
SELECT firstname, lastname, wgs.day, wgs.time
FROM users NATURAL JOIN students NATURAL JOIN participate NATURAL JOIN wgs;
```

```
--13. User (Id, Vorname, Nachname) die innerhalb des letzten Jahres keine Session hatten:
SELECT user_id, firstname, lastname
FROM sessions NATURAL JOIN users
WHERE logintime > now() - INTERVAL '1 year';
```

```
--*****
```

```
--Daten mit SELECT auswählen unter Verwendung von Joins, Subqueries
```

```
--und Aggregatfunktionen:
```

```
--*****
```

```
--14. Gesucht ist die Gesamtzahl an AG-Plätzen in AGs mit beschränkter Platzzahl
SELECT SUM(max_num) AS anzahl_plaetze_gesamt
```

FROM wgs;

--15. Gesucht ist eine Tabelle der

--Gesamtdauern aller AG-Termine pro AG (AG-Id), die bisher stattgefunden haben:

SELECT SUM(events.duration) AS Gesamtdauer, wg_id

FROM events NATURAL JOIN wgs

GROUP BY wg_id;

--16. Gesucht ist eine Tabelle aller

--AGs mit mehr als einem Leiter:

SELECT COUNT(wg_id) as AnzLeiter, wg_id, title

FROM wgs NATURAL JOIN lead

GROUP BY wg_id

HAVING COUNT(wg_id) > 1;

--17. Gesucht ist die AG mit den meisten eingeschriebenen Teilnehmern:

SELECT title, COUNT(p.user_id) AS count FROM wgs, participate AS p

WHERE wgs.wg_id = p.wg_id

GROUP BY wgs.wg_id

ORDER BY count DESC LIMIT 1;

--18. SuS (Nachname,Rolle), die an einer AG teilnehmen:

SELECT u.lastname, u.roll

FROM users u

WHERE EXISTS

(SELECT 1

FROM participate p

WHERE u.user_id=p.user_id) AND roll = 'user';

--19. Gesucht ist eine Tabelle aller AGs (Titel, Schuljahr, Beschreibung)

--der Schülerin mit dem Benutzernamen "schneeflocke".

--Die Sortierung erfolgt absteigend nach Schuljahr und aufsteigend nach Titel.

SELECT title, p.schoolyear, description FROM wgs, participate AS p

WHERE p.user_id = (SELECT user_id FROM users WHERE username='schneeflocke')

AND p.wg_id = wgs.wg_id

ORDER BY p.schoolyear DESC, wgs.title ASC;

--20. Gesucht ist eine Tabelle aller

--Schüler der Klasse 7a, in welcher die AGs und die Anzahl der besuchten

--Veranstaltungen (= Events) seit Schuljahresbeginn 2020 aufgelistet werden.

--Die Sortierung erfolgt nach Nachname, Vorname und Titel aufsteigend:

SELECT firstname, lastname, title, COUNT(e.event_id)

FROM students AS s, users AS u, wgs AS w, present AS p, events AS e

WHERE class='7a' AND e.date>='2020-08-10' AND u.user_id=s.user_id

AND s.user_id=p.user_id AND p.event_id=e.event_id AND e.wg_id=w.wg_id

GROUP BY u.user_id, w.wg_id

ORDER BY lastname,firstname,title ASC;

--21. Gesucht ist eine Tabelle aller

--Schüler (Vorname, Nachname) mit Tag und Zeit und der Anzahl der AGs an denen sie teilnehmen,

--die am selben Tag zur selben Zeit stattfinden:

```
SELECT COUNT(user_id) AS Teilnahmen, firstname, lastname, wgs.day, wgs.time
FROM users NATURAL JOIN students NATURAL JOIN participate NATURAL JOIN wgs
GROUP BY user_id, wgs.day, wgs.time
HAVING count(user_id) > 1;
```

--22. Gesucht ist eine Tabelle der

--prozentualen Teilnahme der SuS an der AG Basketball absteigend geordnet nach Teilnahme:

```
SELECT (COUNT(wg_id)::FLOAT/anz_termine) AS teilnahme_prozentual, firstname,lastname,title
FROM (SELECT COUNT(wg_id) AS anz_termine, wg_id
      FROM events
      GROUP BY wg_id) AS stattgefunden NATURAL JOIN present NATURAL JOIN events
NATURAL JOIN participate NATURAL JOIN wgs NATURAL JOIN users
GROUP BY title,anz_termine,firstname,lastname
HAVING title = 'Basketball'
ORDER BY teilnahme_prozentual DESC;
```

--23. SuS (Name, Vorname), die an allen Terminen einer AG (Titel)

--anwesend waren aufsteigend sortiert nach AG-Titel.

```
SELECT u3.fn AS vorname,u3.ln AS name, u3.canw AS anwesend, u3.title AS ag_name,
COUNT(events.event_id)
FROM
  (SELECT u2.firstname AS fn, u2.lastname AS ln,u2.c1 AS canw,wgs.wg_id AS agname,wgs.title
   FROM
     (SELECT u1.firstname,u1.lastname,count(events.event_id) AS c1,wg_id
      FROM
        (SELECT firstname,lastname,event_id
         FROM users, present
         WHERE users.user_id=present.user_id) AS u1,events
        WHERE u1.event_id = events.event_id
        GROUP BY u1.firstname,u1.lastname, events.wg_id) AS u2, wgs
     WHERE wgs.wg_id= u2.wg_id) AS u3,events
WHERE events.wg_id = u3.agname
GROUP BY u3.title,u3.fn,u3.ln,u3.canw
HAVING u3.canw = COUNT(events.event_id)
ORDER BY ag_name ASC;
```

Textvorlage: Aufgaben zu Anfragen³

Aufgabe 1): Partnerarbeit

Hinweise:

- Nutzt die einlaminierten Tabellen!
- Führt eure Anfragen schriftlich durch!
- Vergleicht eure Ergebnisse mit der Musterlösung (am Lehrerpult)

³ Stark angelehnt an [1].

Suche 4 Anfragen heraus und notiere sie in Relationaler Algebra:

%--*****

%--Daten auswählen

%--*****

%--Gesucht ist jeweils eine Tabelle aller ...

%--1. Arbeitsgemeinschaften:

%--2. Benutzerdaten (Benutzername, Vorname, Nachname):

%--3. AGs (Titel, Wochentag) dieses Schuljahres:

%--4. Benutzer (Vorname, Nachname, Benutzername) mit Vorname "Nina":

%--5. User (Id, Vorname, Nachname) die eine Session haben:

%--*****

%--Daten mit SELECT auswählen unter Verwendung von Joins:

%--*****

%--Gesucht ist jeweils eine Tabelle aller ...

%--6. Lehrer (Vorname, Nachname, Kürzel):

%--7. Schüler (Vorname, Nachname, Klasse, Schülernummer):

%--8. Schüler (Vorname, Nachname) mit Tag und Zeit der AGs an denen sie teilnehmen:

Aufgabe 2): Partnerarbeit

Melde dich auf dem Raspberry Pi mit Putty an (IP-Adresse und Passwort an der Tafel). Gib dann psql ein.

- Durch \dt werden alle Tabellen angezeigt, die sich in der Datenbank befinden. Notiert euch zu jeder Tabelle die dazugehörigen Attribute, die Schlüssel sowie die Zusammenhänge zwischen den Tabellen.
- Wähle 10 Anfragen aus Aufg.1 aus und notiere sie in SQL-Syntax. Vergleicht eure Ergebnisse mit den Musterlösungen, die sich am Lehrerpult befinden.
- Überlegt euch noch eigene Anfragen und notiert die Ergebnisse.
- Trefft euch mit einem_r anderen Schüler_In und lasst diese_n jeweils zwei eurer Anfragen aus c) durchführen.

DES: Daten zu den Relationen

(Die vollständigen Daten sind zu finden in: ../relational/ag_manager-query.ra)

DES: Anfragen in Relationenalgebra

%--*****

%--Daten auswählen

%--*****

%--Gesucht ist jeweils eine Tabelle aller ...

%--1. Arbeitsgemeinschaften:

/listing wgs

%--2. Benutzerdaten (Benutzername, Vorname, Nachname):

project username, firstname, lastname (users);

%--3. AGs (Titel, Wochentag) dieses Schuljahres:

project title, day (select (schoolyear='2020/21') (wgs));

%--4. Benutzer (Vorname, Nachname, Benutzername) mit Vorname "Nina":

project firstname, lastname, username (select firstname='Nina' (users));

%--5. User (Id, Vorname, Nachname) die eine Session haben:

project user_id, firstname, lastname, logintime (sessions njoin users);

%--*****

%--Daten mit SELECT auswählen unter Verwendung von Joins:

%--*****

%--Gesucht ist jeweils eine Tabelle aller ...

%--6. Lehrer (Vorname, Nachname, Kürzel):

project firstname, lastname, shorthand (select users.user_id = teachers.user_id (users product teachers));

%--7. Schüler (Vorname, Nachname, Klasse, Schülernummer):

project firstname, lastname, class, studentnumber (users njoin students);

%--8. Schüler (Vorname, Nachname) mit Tag und Zeit der AGs an denen sie teilnehmen:

project firstname, lastname, wgs.day, wgs.time (users NJOIN students NJOIN participate NJOIN wgs);

DES: Antworten in Relationaler Algebra

%--*****

%--Daten auswählen

%--*****

DES> %--Gesucht ist jeweils eine Tabelle aller ...

1.

DES> %--Arbeitsgemeinschaften:

DES> /listing wgs

wgs(1,'Nähen','Montag',time(15,0,0),90,10,'0','2020/21','Wir nähen alle einen Teddybären.').

wgs(2,'Schach','Montag',time(15,0,0),90,10,'1','2020/21','Klassisches Schachspiel für jung und alt. Beginn: 29. März 2021').

wgs(3,'Karate','Mittwoch',time(15,30,0),90,0,'1','2020/21','Shotokan-Karate für die persönliche Entwicklung.').

wgs(4,'Technik','Freitag',time(14,0,0),90,6,'1','2020/21','Auf- und Abbau der Theatertechnik. Betreuung während der Proben und Aufführungen.').

wgs(5,'Chemie','Dienstag',time(16,0,0),90,8,'1','2019/20','Experimente, die Puffen und Stinken.').

wgs(6,'Basketball','Mittwoch',time(16,0,0),90,20,'1','2020/21','Was gibt es da zu erklären?').

Info: 6 rules listed.

2.

%--Benutzerdaten (Benutzername, Vorname, Nachname):

DES> project username, firstname, lastname (users);

answer(users.username:varchar(30),users.firstname:varchar(30),users.lastname:varchar(30)) ->

```
{
  answer(admin,'Adam','Adler'),
  answer(admin2,'Bertha','Bentz'),
  answer(admin3,'Claus','Clever'),
  answer(almey,'Alexander','Meyer'),
  answer(chutri,'Chu Thành','Tri'),
  answer(dodec,'Doreen','Decker'),
  answer(eisenhut,'Fidora','Flanell'),
  answer(frappa,'Frank','Zappa'),
  answer(frühlingsblume,'Uschi','Usus'),
  answer(habarb,'Habudi','Harb'),
  answer(hamarb,'Hamudi','Harb'),
  answer(herbstnebel,'Viki','Vigor'),
  answer(herrbaer,'Herbert','Bähr'),
  answer(jajab,'Jakub','Jabłoński'),
  answer(jewex,'Jens','Wexler'),
  answer(kamue,'Katja','Müller'),
  answer(kartoffelkraut,'Xaver','Xanthopoulos'),
```

```

answer(kaulquappe96,'Hannah','Merkur'),
answer(lauda,'Laura','Dresdner'),
answer(laumue,'Laura','Müller'),
answer(luftballon,'Gregor','Müller'),
answer(moossteinbrech,'Eduard','Möhrike'),
answer(nazajac,'Natasza','Zajac'),
answer(ostseebrise,'Bernd','Brot'),
answer(regenhimmel,'Moritz','Macker'),
answer(rosskastanie,'Liselotte','Müller'),
answer(rotbaum,'Waldemar','Wacker'),
answer(sabt,'Stephan','Abt'),
answer(schneeflocke,'Nina','Nadel'),
answer(selowe,'Sebastian','Lowe'),
answer(silberstreif,'Jana','Jensen'),
answer(sonnenschein,'Edwin','Edison'),
answer(stoppelfeld,'Yannis','Yoghurt'),
answer(tonnhi,'Tôn Ngoc','Nhi'),
answer(wawis,'Walery','Wiśniewski'),
answer(weintraube,'Walter','Dremel'),
answer(winterweite,'Elisabeth','Engel'),
answer(wischam,'Wisaam Aram','Cham'),
answer(wiskwi,'Wisława','Kwiatkowska'),
answer(wiszc,'Wisla','Szczepańska'),
answer(wolwi,'Wolfgang','Wowereit')
}

```

Info: 41 tuples computed.

3.

%--AGs (Titel, Wochentag) dieses Schuljahres:

project title, day (select (schoolyear='2020/21') (wgs));

```

answer(wgs.title:varchar(100),wgs.day:varchar(255)) ->
{
  answer('Basketball','Mittwoch'),

```

```

answer('Karate','Mittwoch'),
answer('Nähen','Montag'),
answer('Schach','Montag'),
answer('Technik','Freitag')
}

```

Info: 5 tuples computed.

4.

%--Benutzer (Vorname, Nachname, Benutzername) mit Vorname "Nina":

```
project firstname, lastname, username (select firstname='Nina' (users));
```

```

answer(users.firstname:string,users.lastname:varchar(30),users.username:varchar(30)) ->
{
  answer('Nina','Nadel',schneeflocke)
}

```

Info: 1 tuple computed.

5.

%--User (Id, Vorname, Nachname) die eine Session haben:

```
DES> project user_id, firstname,lastname, logintime (sessions njoin users);
```

```

answer(user_id:int,firstname:varchar(30),lastname:varchar(30),logintime:datetime(datetime)) ->
{
  answer(2,'Edwin','Edison',timestamp(2021,5,25,15,33,19,620020000))
}

```

Info: 1 tuple computed.

%--*****

%--Daten mit SELECT auswählen unter Verwendung von Joins:

%--*****

%--Gesucht ist jeweils eine Tabelle aller ...

6.

%--Lehrer (Vorname, Nachname, Kürzel):

project firstname, lastname, shorthand (select users.user_id = teachers.user_id (users product teachers));

answer(users.firstname:varchar(30),users.lastname:varchar(30),teachers.shorthand:varchar(10)) ->

```
{
  answer('Adam','Adler','ADL'),
  answer('Bernd','Brot','BRO'),
  answer('Bertha','Bentz','BNZ'),
  answer('Claus','Clever','CLR'),
  answer('Edwin','Edison','EDI'),
  answer('Jana','Jensen','JEN'),
  answer('Moritz','Macker','MAC'),
  answer('Viki','Vigor','VIG'),
  answer('Waldemar','Wacker','WCK'),
  answer('Xaver','Xanthopoulos','XTS'),
  answer('Yannis','Yoghurt','YOG')
}
```

Info: 11 tuples computed.

7.

%--Schüler (Vorname, Nachname, Klasse, Schülernummer):

project firstname, lastname, class, studentnumber (users njoin students);

answer(firstname:varchar(30),lastname:varchar(30),class:varchar(10),studentnumber:varchar(20)) ->

```
{
  answer('Alexander','Meyer','7a',meyer01),
  answer('Chu Thành','Tri','7a',tri01),
  answer('Doreen','Decker','7a',decker01),
  answer('Fidora','Flanell','8a',fifl01),
  answer('Frank','Zappa','8a',zappa01),
  answer('Gregor','Müller','7a',grmü01),
  answer('Habudi','Harb','7a',harb02),
}
```

```

answer('Hamudi','Harb','7a',harb01),
answer('Herbert','Bähr','7a',baehr01),
answer('Jakub','Jabłoński','7b',jabl01),
answer('Jens','Wexler','7a',wexler01),
answer('Katja','Müller','7a',mueller02),
answer('Laura','Dresdner','8b',dresdner01),
answer('Laura','Müller','7a',mueller01),
answer('Liselotte','Müller','8a',limü01),
answer('Natasza','Zajac','7a',zajac01),
answer('Nina','Nadel','7b',nina01),
answer('Sebastian','Lowe','7a',lowe01),
answer('Stephan','Abt','8b',abt01),
answer('Tôn Ngoc','Nhi','7a',nhi01),
answer('Uschi','Usus','7a',usus01),
answer('Walery','Wiśniewski','7b',wisn01),
answer('Walter','Dremel','7b',wadr01),
answer('Wisaam Aram','Cham','7a',cham01),
answer('Wisla','Szczepańska','7b',szcz01),
answer('Wisława','Kwiatkowska','7a',kwiatkowska01),
answer('Wolfgang','Wowereit','7b',wowi01)
}

```

Info: 27 tuples computed.

8.

%--Schüler (Vorname, Nachname) mit Tag und Zeit der AGs an denen sie teilnehmen:

```

DES> project firstname, lastname, wgs.day, wgs.time (users NJOIN students NJOIN participate NJOIN
wgs);

```

```

answer(firstname:varchar(30),lastname:varchar(30),wgs.day:varchar(255),wgs.time:datetime(time)) ->
{
answer('Alexander','Meyer','Mittwoch','15:30:00'),
answer('Chu Thành','Tri','Mittwoch','15:30:00'),
answer('Doreen','Decker','Freitag','14:00:00'),
answer('Doreen','Decker','Mittwoch','15:30:00'),

```

answer('Doreen','Decker','Mittwoch','16:00:00'),
answer('Fidora','Flanell','Mittwoch','15:30:00'),
answer('Fidora','Flanell','Mittwoch','16:00:00'),
answer('Fidora','Flanell','Montag','15:00:00'),
answer('Frank','Zappa','Mittwoch','16:00:00'),
answer('Frank','Zappa','Montag','15:00:00'),
answer('Gregor','Müller','Mittwoch','16:00:00'),
answer('Gregor','Müller','Montag','15:00:00'),
answer('Habudi','Harb','Montag','15:00:00'),
answer('Hamudi','Harb','Mittwoch','15:30:00'),
answer('Herbert','Bähr','Freitag','14:00:00'),
answer('Herbert','Bähr','Mittwoch','15:30:00'),
answer('Jakub','Jabłoński','Montag','15:00:00'),
answer('Laura','Dresdner','Mittwoch','16:00:00'),
answer('Laura','Müller','Mittwoch','15:30:00'),
answer('Liselotte','Müller','Mittwoch','15:30:00'),
answer('Liselotte','Müller','Mittwoch','16:00:00'),
answer('Natasza','Zajac','Mittwoch','15:30:00'),
answer('Nina','Nadel','Mittwoch','15:30:00'),
answer('Nina','Nadel','Mittwoch','16:00:00'),
answer('Nina','Nadel','Montag','15:00:00'),
answer('Sebastian','Lowe','Mittwoch','16:00:00'),
answer('Stephan','Abt','Freitag','14:00:00'),
answer('Stephan','Abt','Mittwoch','15:30:00'),
answer('Stephan','Abt','Mittwoch','16:00:00'),
answer('Stephan','Abt','Montag','15:00:00'),
answer('Tôn Ngoc','Nhi','Mittwoch','16:00:00'),
answer('Uschi','Usus','Freitag','14:00:00'),
answer('Uschi','Usus','Mittwoch','16:00:00'),
answer('Uschi','Usus','Montag','15:00:00'),
answer('Walter','Dremel','Freitag','14:00:00'),
answer('Walter','Dremel','Mittwoch','16:00:00'),
answer('Wisaam Aram','Cham','Freitag','14:00:00'),
answer('Wolfgang','Wowereit','Mittwoch','15:30:00')

}

Info: 38 tuples computed.

Die Views

Zu den Views siehe [../view/view_basics_and_extended.pdf](#)

Quelle:

[1] Hartung, Thomas, Herker, Sebastian, Nordmann, Thomas, Vogt, Alisa: Kursfahrten. Berlin 2017