**Aufgabe „2021\_So\_FIAE\_GA1\_BearbeitungsHinweise\_DB.docx“**

Die Handlungsschritte 1 bis 5 beziehen sich auf die folgende Ausgangssituation:

Der gemeinnützige Verein Helferlein e. V. vermittelt Dienstleistungen von freiwilligen Helfern an hilfesuchende Bürger. Bislang wurden diese Dienstleistungen telefonisch kommuniziert. Nun möchte der Helferlein e. V. die Verwaltung der Dienstleistungen digitalisieren.

Sie sollen vier der folgenden fünf Aufgaben in diesem Projekt erledigen:

~~1. UML-Anwendungsfall-Diagramm entwickeln~~

~~2. Prozedur für Entgeltauflistung schreiben~~

~~3. Objektorientiertes Modell für geschützten Methodenzugriff entwickeln~~

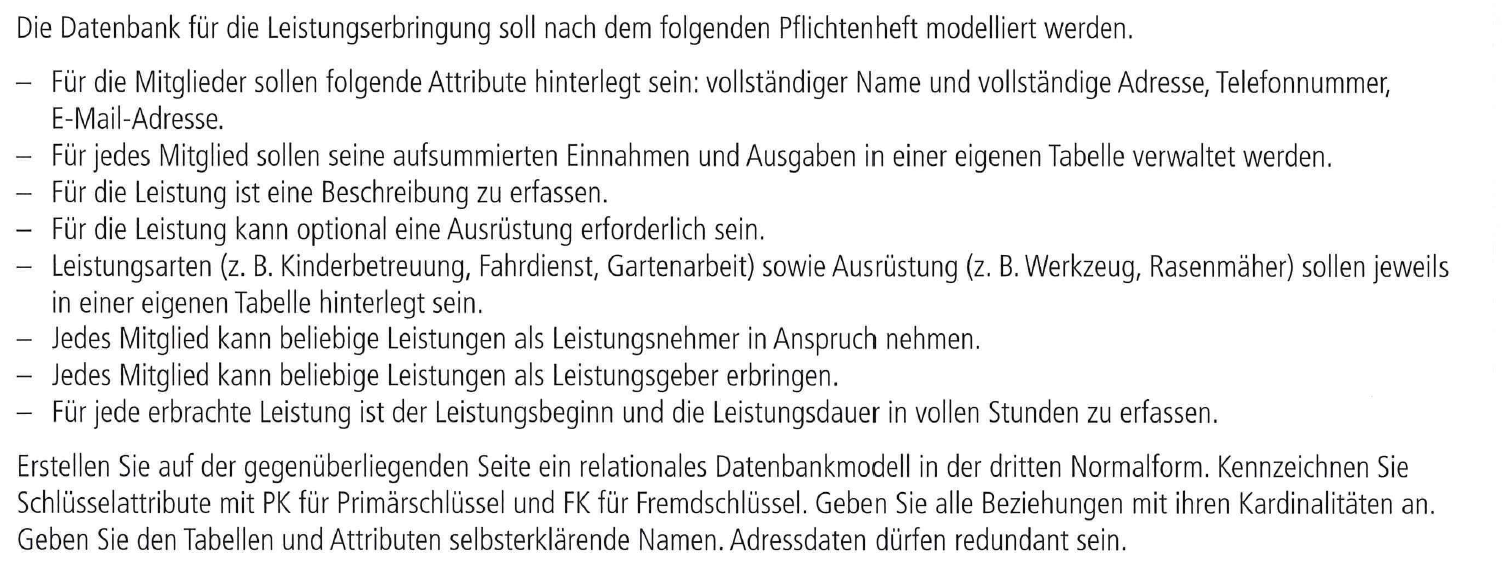
4. Relationales Datenbankmodell für Leistungserbringung anfertigen

5. SQL-Abfragen zu Mitgliedern, Angeboten und Bewertungen erstellen

Im Belegsatz finden Sie dazu:

* SQL
* Sequenzdiagramm
* Klassendiagramm
* USE-Case

**Handlungsschritt 4**





Lösungs- bzw. Bewertungsvorschlag:







**Tabelle Lstg\_Mitglied:**



offizielle Variante:

* hier wurde ein zusammengesetzter PK aus 4 Spalten gebildet ----> die Kombinationen aus diesen 4 Spalten müssen immer eindeutig sein, sprich: sich unterscheiden --> es kann kein Mehrfacheintrag für die selbe Leistungserbringung erfolgen
* die ersten 3 Spalten sind jede für sich Fremdschlüsselfeld

Variante 2:

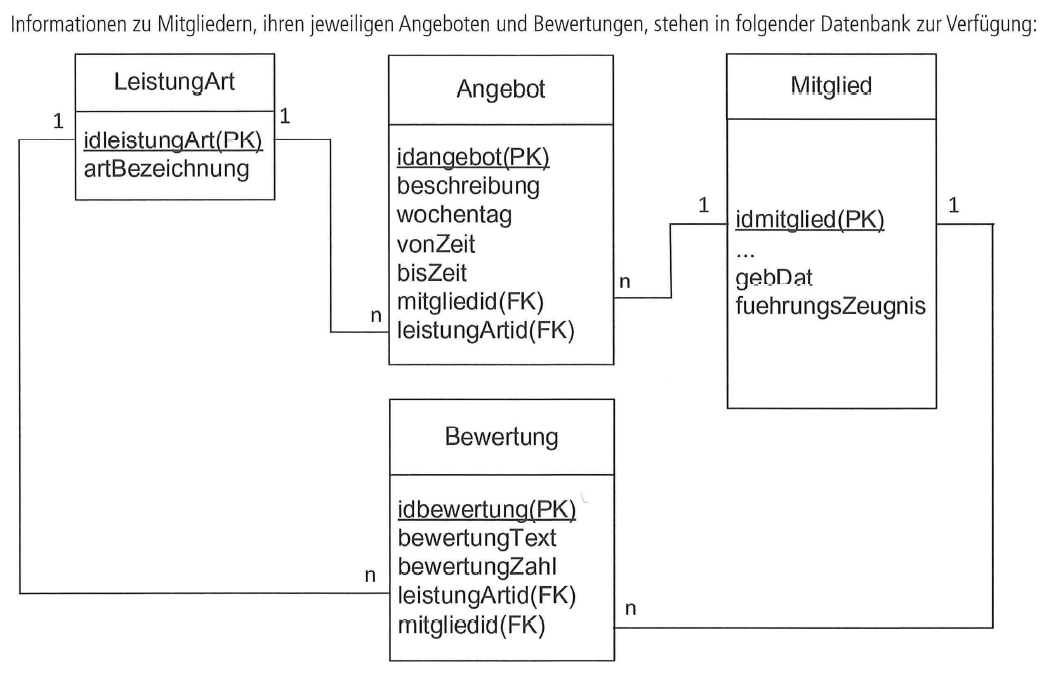
diese Tabelle könnte auch so aussehen (also kein 4fach zusammengesetzter PK, sondern nur eine einfache ID als PK):  
Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung



* hier wurde ein künstlicher Schlüssel verwendet, um als ID Eindeutigkeit herzustellen ---> dieser Schlüssel kann aber keine Mehrfacheinträge bei den 4 ersten Spalten verhindern

**Handlungsschritt 5**



Formulieren Sie SQL-Abfragen für die nachstehenden Aufgabenstellungen.

a) Geben Sie alle Attribute des jüngsten Mitglieds aus. (4 Punkte)

VARIANTE 1:

select \* from Mitglied  
where gebDat = (select max(gebDat) from Mitglied)



**BEACHTE:**

* 'jüngstes Mitglied' ---> Funktion max()

VARIANTE 2:

select \* from Mitglied  
order by gebDat desc limit 1;



**BEACHTE:**

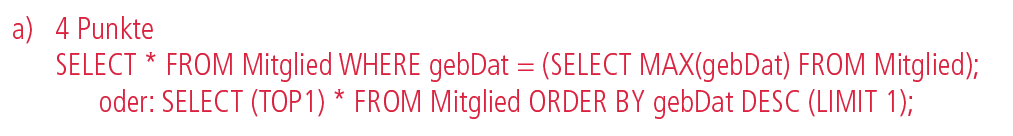


* die Aufgabe und diese Lösung gehen davon aus, dass es nur 1 jüngstes Mitglied gibt ---> das ist praktisch nicht gut



* !!! Abfragen immer so formulieren, dass sie **allgemeingültig** sind







b) Ermitteln Sie eine Mitgliederliste aufsteigend sortiert nach der durchschnittlichen Bewertung für die Leistungsart "Kinderbetreuung". (6 Punkte )





select  
 idmitglied,  
 mitgliedName,  
 avg(bewertungZahl) as 'Durchschnitt'  
from Mitglied   
 join Bewertung on idmitlied = mitgliedid  
 join LeistungsArt on leistungArtid = idleistungArt  
where artBezeichnung = 'Kinderbetreuung'  
group by idmitglied, mitgliedName  
order by avg(bewertungZahl) asc;





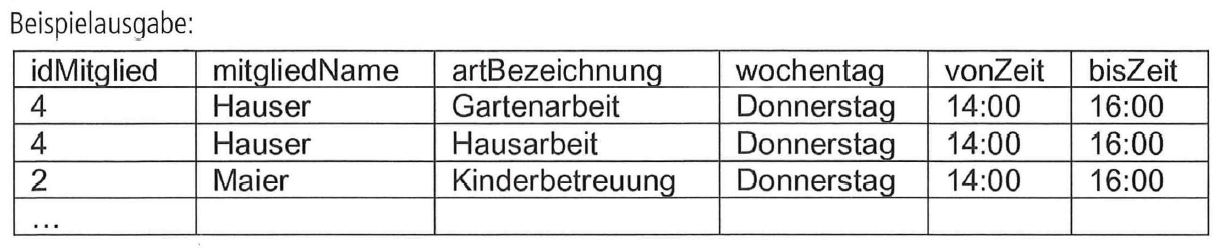
// bei der Musterlösung fehlt die Sortierung

// wenn in der Spaltenliste gleichzeitig Spalten und Ergebnisse von Aggregatfunktionen verwendet werden sollen, **muss** es eine gruppierte Liste sein  
// alle Spalten, die in der Spaltenliste verwendet werden sollen, **müssen** nach dem GROUP BY aufgeführt werden  
select  
 idmitglied,  
 mitgliedName,  
 avg(bewertungZahl) as 'Durchschnitt'  
from Mitglied   
 join Bewertung on idmitlied = mitgliedid  
 join LeistungsArt on leistungArtid = idleistungArt  
where artBezeichnung = 'Kinderbetreuung'  
group by idmitglied, mitgliedName  
order by avg(bewertungZahl) asc;

// das folgende ist KEINE Lösung:

// diese Anweisung liefert den Durchschnitt **aller Mitglieder** bei der Kinderbetreuung  
// hier fehlt die Gruppierung  
select  
 avg(bewertungZahl) as 'Durchschnitt'  
from Mitglied   
 join Bewertung on idmitlied = mitgliedid  
 join LeistungsArt on leistungArtid = idleistungArt  
where artBezeichnung = 'Kinderbetreuung'  
order by avg(bewertungZahl) asc;

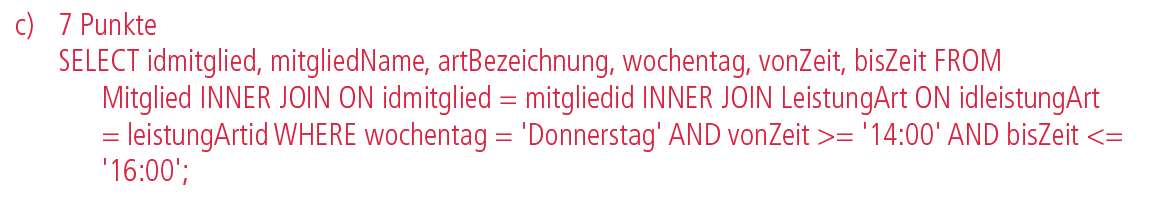
c) Erstellen Sie eine Angebotsliste, die alle Mitglieder und die entsprechende Leistungsart des Angebots ausgibt, welche donnerstags von 14:00 Uhr bis 16:00 Uhr zur Verfügung stehen. (7 Punkte)



select  
 idmitglied as idMitglied // ??? muss das sein  
 mitgliedName  
 artBezeichnung  
 wochentag  
 vonZeit  
 bisZeit  
from Mitglied  
 join Angebot on idmitglied = mitgliedid  
 join LeistungArt on leistungArtid = idleistungArt  
where wochentag = 'Donnerstag' and  
 vonZeit = '14:00' and  
 bisZeit = '16:00'

**HINWEIS:**

* die Aussagen zum Zeitfenster sind nicht exakt ---> genau 14 bis 16 Uhr? Oder?  
  vonZeit = '14:00' ODER vonZeit >= '14:00'  
  ist der Zeitraum von 14 bis 16 gemeint oder die Leistung genau von Zeitpunkt 1 bis Zeitpunkt 2????
* in der Prüfung statt JOIN besser INNER JOIN verwenden



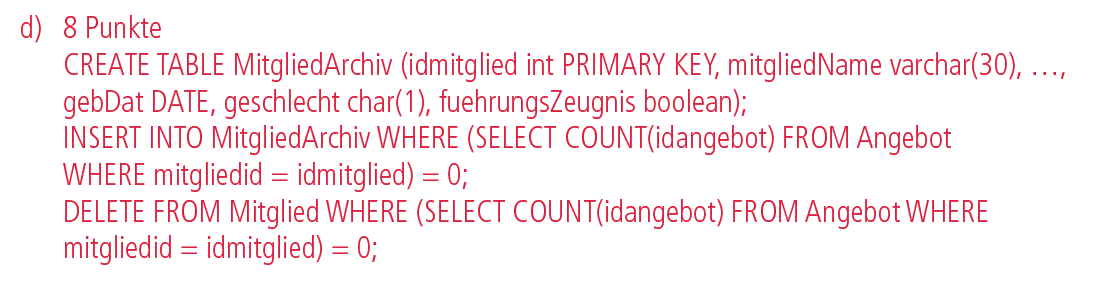
d) Erstellen Sie eine neue Tabelle MitgliedArchiv. Transferieren Sie alle Mitglieder, die kein Angebot eingestellt haben, in diese Tabelle. Löschen Sie diese inaktiven Mitglieder aus der Tabelle Mitglied. (8 Punkte)

create table MitgliedArchiv (  
 idmitglied int primary key,  
 mitgliedName char(50),  
 gebDat date,  
 fuehrungsZeugnis char(1))

insert into MitgliedArchiv(idmitglied, mitgliedName, gebDat, fuehrungsZeugnis)  
 select idmitglied, mitgliedName, gebDat, fuehrungsZeugnis  
 from Mitglied full outer join Angebot on idmitglied = mitgliedid  
 where idangebot is NULL

*-- das Schlüsselwort VALUES entfällt, wenn die einzufügenden Daten von einem SELECT geliefert werden*

delete from Mitglied   
 where idmitglied in (select idmitglied from Mitglied full outer join Angebot on idmitglied = mitgliedid where idangebot is NULL)





**HINWEIS:** Anweisung 1 ok; Anweisungen 2 und 3 halte ich für nicht funktionierend