



Einführung in Datenbanken — Klausur 1 —

Name: _____

Studiengang: _____

Matrikelnummer: _____

Fühlen Sie sich gesundheitlich in der Lage die Prüfung abzulegen? ☐ Ja, ☐ Nein

Unterschrift: _____

(Melden Sie sich unbedingt bei der Aufsicht, wenn Sie sich krank fühlen sollten – eine Krankmeldung muss vor Ende der Prüfung erfolgen.)

Aufgabe	Punkte	Max. Punkte	Zeit
1 (SQL-Anfragen [ILIAS])		25	50 min
2 (Äquivalenz von Anfragen [ILIAS])		6	15 min
3 (Relationale Algebra [ILIAS])		4	5 min
4 (ER-Entwurf)		6	15 min
5 (Relationales Modell [ILIAS])		5	15 min
6 (Bonusaufgabe: BCNF [ILIAS])		4	10 min
7 (Bonusaufgabe: Evaluation [ILIAS])		1	1 min
Summe		51	111 min

Hinweise:

- Bearbeitungsdauer: 120 Minuten
- Wegen der Bonusaufgaben zählen 46 Punkte als 100%.
- Die Klausur hat 21 Seiten. Bitte prüfen Sie die Vollständigkeit.
- Nur die Aufgabe 4 (Entity-Relationship-Entwurf) ist hier zu beantworten (Bitte benutzen Sie den vorgegebenen Platz und verwenden Sie weder Rot- noch Bleistift.). Alle anderen Aufgaben müssen am Rechner in ILIAS bearbeitet werden.
- Bei Ankreuzaufgaben sollten Sie notfalls raten. Es gibt keine negativen Punkte.
- Mobiltelefone müssen ausgeschaltet werden, oder stumm geschaltet in Ihrer Tasche verstaut werden. Wenn Sie erreichbar sein müssen, sprechen Sie mit der Aufsicht.
- Fragen Sie, wenn etwas nicht klar ist. Sie können auch probieren, bei völlig unverständlichen Fehlermeldungen von PostgreSQL (Adminer) zu fragen. Eventuell bekommen Sie aber keine Antwort (wenn wir denken, dass Sie es wissen müssen).
- Drei DIN-A4 Blätter (Vorder- und Rückseite) mit Notizen sind erlaubt.
- Wichtige Keyboard-Shortcuts (für Wechsel zwischen Adminer und ILIAS):
 - **Alt+Tab**: Fenster wechseln
 - **STRG+A**: Alles markieren
 - **STRG+C**: Kopieren (in Zwischenablage)
 - **STRG+V**: Einfügen
 - **SHIFT+STRG+V**: Unformatiert einfügen
 - **STRG+Z**: Undo
 - **SHIFT+STRG+Z** oder **STRG+Y**: Redo
 - **SHIFT+Enter**: Normale neue Zeile in ILIAS (kein Paragraph).
- Es empfiehlt sich, nur einmal auf dem Adminer-Link zu klicken, und damit ein neues Fenster zu öffnen. Später können Sie dann mit **Alt+Tab** zwischen den Fenstern wechseln. Wenn Sie oft auf den Link klicken, öffnen Sie viele Fenster.
- Wenn die Tastatur unerwartete Zeichen erzeugt, kontrollieren Sie, dass Sie die deutsche Tastaturbelegung gewählt haben (unten rechts gibt es Deutsch und Englisch).
- Sie können Ihre Antworten bis zum Ende der Klausur noch ändern. Es zählt die letzte gespeicherte Antwort. Alle Antworten werden auf dem Server gespeichert. Wenn Ihr Rechner ein Problem haben sollte, das die Aufsicht nicht beseitigen kann, können Sie an einen anderen Rechner wechseln und die Klausur dort fortsetzen.
- Keine Aufzeichnungen über die Klausur sollen den Raum verlassen. Wenn Sie auf Ihre Notizblätter etwas schreiben, müssen Sie diese abgeben.
- Es bringt nichts, SQL-Anfragen zu basteln, die ausschließlich im Beispielszustand das richtige Ergebnis liefern, aber mit der Aufgabenstellung überhaupt nichts zu tun haben. Extreme Fälle werden mit 0 Punkten bewertet. Alle Anfragen werden auch manuell angeschaut, nicht nur automatisch korrigiert. Auch minimale Syntaxfehler kosten allerdings wenigstens einen Punkt, vermutlich mehr.

Beispiel-Datenbank

Der Stundenplan von Informatik-Lehrveranstaltungen und Anmeldungen von Studierenden zu diesen Lehrveranstaltungen seien in folgender Datenbank gespeichert:

- WOCHENTAGE(TAG, NR, WOCHENENDE)
- LV(VID, TITEL, TAG → WOCHENTAGE, STUNDE, UEBUNG_ZU° → LV, MAX_TN)
- STUDENTEN(SID, VORNAME, NACHNAME, EMAIL°, FACH)
- TEILNEHMER(SID → STUDENTEN, VID → LV)

Zur Vereinfachung sei angenommen, dass alle Lehrveranstaltungen nur einen Termin haben, und alle Termine Doppelstunden sind, die in einem festen Raster 8, 10, 12, ... starten. Ein Terminkonflikt kann also nur bestehen, wenn TAG und STUNDE übereinstimmen.

Die Tabellen sind mit den folgenden Inhalten gefüllt. Die Inhalte der Tabellen dienen nur als Beispiele, um die Beziehungen zwischen den Tabellen zu illustrieren.

- Die Tabelle WOCHENTAGE definiert die Reihenfolge der Wochentage. Primärschlüssel sind jeweils unterstrichen, hier ist also TAG Primärschlüssel. Der Wert in der Spalte NR ist ebenfalls eindeutig (Alternativschlüssel). Die Spalte WOCHENENDE enthält "Y" für Tage am Wochenende, und "N" für die üblichen Arbeitstage.

WOCHENTAGE		
<u>TAG</u>	NR	WOCHENENDE
MO	1	N
DI	2	N
⋮	⋮	⋮
SA	6	Y
SO	7	Y

- In der Lehrveranstaltungs-Tabelle LV ist TAG ein Fremdschlüssel, der auf WOCHENTAG verweist. Das ist in der Übersicht oben mit dem Pfeil gekennzeichnet. Der Vorlesungstitel TITEL ist eindeutig (Alternativschlüssel). UEBUNG_ZU ist ein Fremdschlüssel, der auf die Tabelle selbst verweist. In Einträgen für Übungen enthält er die VID der zugehörigen Vorlesung, ansonsten einen Nullwert. Es kann mehrere alternative Übungsgruppen zu einer Vorlesung geben. MAX_TN ist die maximale Teilnehmerzahl (natürlich nicht negativ).

LV					
<u>VID</u>	TITEL	TAG	STUNDE	UEBUNG_ZU	MAX_TN
1	OOP	DI	8		200
2	Übung 1 zu OOP	DI	10	1	3
3	Übung 2 zu OOP	DI	12	1	3
4	Datenbanken	MO	10		100
5	DB-Übung	DI	12	4	20
6	XML+DBen	DO	10		25

- (Die Tabellen STUDENTEN und TEILNEHMER sind auf der nächsten Seite gezeigt.)

- Die Studenten-Tabelle ist wie aus der Vorlesung bekannt, nur mit einer zusätzlichen Spalte für das Studienfach. VORNAME und NACHNAME ist ein Alternativschlüssel.

STUDENTEN				
<u>SID</u>	VORNAME	NACHNAME	EMAIL	FACH
101	Lisa	Weiss	weiss@acm.org	Informatik
102	Michael	Grau	NULL	Mathematik
103	Daniel	Sommer	daniel@gmail.com	Informatik
104	Iris	Winter	winter@hotmail.com	Bioinformatik

- Die Tabelle Teilnehmer enthält die Information, welcher Student für welche Lehrveranstaltung registriert ist:

TEILNEHMER	
<u>SID</u>	<u>VID</u>
101	1
101	2
101	4
101	5
102	1
102	2
103	1
103	2
104	2
104	3
104	4
104	5

Aufgabe 1 (SQL-Anfragen [ILIAS])**25 Punkte**

Die Aufgaben a, b, c, d und e sind im Ilias zu finden und zu bearbeiten. Formulieren Sie die folgenden Anfragen in SQL. Natürlich sollen Ihre Anfragen nicht nur mit den obigen Beispiel-Daten funktionieren, sondern für beliebige Tabelleninhalte. Beachten Sie, dass auch für unnötige Komplikationen Punkte abgezogen werden können. Die Anfragen sollen keine Duplikate liefern, aber für ein unnötiges **DISTINCT** oder ein unnötiges **UNION** statt **UNION ALL** werden auch Punkte abgezogen. Um Missverständnisse zu vermeiden, sind jeweils die Ergebnisse der gesuchten Anfragen im Beispiel-Zustand angegeben. Die Spaltenüberschriften Ihrer Anfrage sollen mit dem Beispiel übereinstimmen und in der angegebenen Reihenfolge sein.

- a) Geben Sie für die Studentin “Lisa Weiss” einen Stundenplan aus, der im Beispielzustand so aussieht:

Ergebnis				
TAG	VON	BIS	TITEL	ART
MO	10	12	Datenbanken	V
DI	8	10	OOP	V
DI	10	12	Übung 1 zu OOP	Ü
DI	12	14	DB-Übung	Ü

Beachten Sie dabei Folgendes:

- Das Ergebnis soll nach Wochentagen sortiert sein (in der üblichen Reihenfolge, nicht etwa alphabetisch), und bei gleichem Wochentag nach der Start-Uhrzeit. Sollte Frau Weiss für zwei gleichzeitige Veranstaltungen (gleicher Tag und gleiche Start-Uhrzeit) gebucht sein, sollen diese Veranstaltungen nach dem Titel sortiert werden.
- Die Start-Uhrzeit (aus der Spalte **STUNDE** der Tabelle **LV**) soll in einer Ausgabe-Spalte mit dem Namen **VON** stehen.
- In der Spalte “**BIS**” soll die Start-Uhrzeit plus 2 (Stunden) ausgegeben werden. Sie können davon ausgehen, dass alle Lehrveranstaltungen spätestens um 20 Uhr beginnen, so dass ein Überlauf über Mitternacht hinaus nicht berücksichtigt werden muss.
- Die Spalte “**ART**” soll den Wert “Ü” enthalten, wenn es sich um eine Übung handelt, (erkennbar daran, dass die Spalte **UEBUNG_ZU** keinen Nullwert enthält). Ansonsten soll “V” ausgegeben werden.
- Selbstverständlich können Sie nicht voraussetzen, dass die Studentin eine bestimmte **SID** hat. In der Aufgabe gegeben sind nur **VORNAME** und **NACHNAME**.

Falls Sie die Spalte “**ART**” nicht hinbekommen, können Sie diese weglassen (gegen Punktabzug).

Lösung: *Lösung mit CASE:*

```
SELECT V.TAG, V.STUNDE AS VON, V.STUNDE+2 AS BIS, V.TITEL,  
       CASE WHEN V.UEBUNG_ZU IS NULL THEN 'V' ELSE 'Ü' END AS ART  
FROM   STUDENTEN S, TEILNEHMER T, LV V, WOCHENTAGE W  
WHERE  S.VORNAME = 'Lisa' AND S.NACHNAME = 'Weiss'  
       AND S.SID = T.SID AND T.VID = V.VID  
       AND V.TAG = W.TAG  
ORDER BY W.NR, VON, V.TITEL
```

Mögliche Bepunktung:

- 1 Punkt für syntaktisch korrekte Anfrage ungefähr in der richtigen Größe (die mit der Aufgabe etwas zu tun hat) ohne offensichtliche, schwere semantische Fehler.
- 1 Punkt für die SELECT-Klausel ohne die spezielle Spalte ART, aber inklusive dem Term zur Berechnung von BIS und der Umbenennung der Spalten.
- 1 Punkt für die spezielle Spalte ART.
- 1 Punkt für den Join der vier Tabellen und die Selektion von Lisa Weiss.
- 1 Punkt für das ORDER BY.

Lösung mit UNION. Dies ist etwas erschwert, weil nach dem UNION auf die Spalte W.NR nicht mehr zugegriffen werden kann, sofern diese nicht unter SELECT gelistet ist.

```
WITH LVX AS (  
  SELECT VID, TAG, STUNDE, TITEL, 'V' AS ART  
  FROM   LV  
  WHERE  UEBUNG_ZU IS NULL  
  UNION  
  SELECT VID, TAG, STUNDE, TITEL, 'Ü' AS ART  
  FROM   LV  
  WHERE  UEBUNG_ZU IS NOT NULL  
)  
SELECT V.TAG, V.STUNDE AS VON, V.STUNDE+2 AS BIS, V.TITEL, V.ART  
FROM   STUDENTEN S, TEILNEHMER T, LVX V, WOCHENTAGE W  
WHERE  S.VORNAME = 'Lisa' AND S.NACHNAME = 'Weiss'  
       AND S.SID = T.SID AND T.VID = V.VID  
       AND V.TAG = W.TAG  
ORDER BY W.NR, VON, V.TITEL
```

Vereinfachte Variante ohne die Spalte ART:

```
SELECT V.TAG, V.STUNDE AS VON, V.STUNDE+2 AS BIS, V.TITEL  
FROM   STUDENTEN S, TEILNEHMER T, LV V, WOCHENTAGE W  
WHERE  S.VORNAME = 'Lisa' AND S.NACHNAME = 'Weiss'  
       AND S.SID = T.SID AND T.VID = V.VID  
       AND V.TAG = W.TAG  
ORDER BY W.NR, VON, V.TITEL
```

Ich würde für die vereinfachte Variante 4/5 Punkten geben.

- b) Geben Sie die Titel von Übungen aus, die nicht den Titel der Vorlesung als Teilzeichenkette enthalten. (Hinweis: Falls Sie eine String-Konkatenation benötigen, verwenden Sie den Operator “||”).

Ergebnis	
UEBUNG	VORLESUNG
DB-Übung	Datenbanken

Lösung: *Die wesentliche Schwierigkeit ist das Vergleichsmuster für LIKE zusammenzusetzen. Es ist aber auch eine Anfrage mit Selbst-Verbund.*

```
SELECT U.TITEL AS UEBUNG, V.TITEL AS VORLESUNG
FROM LV U, LV V
WHERE U.UEBUNG_ZU = V.VID
AND U.TITEL NOT LIKE '%' || V.TITEL || '%'
```

Mögliche Bepunktung:

- 1 Punkt für syntaktisch korrekte Anfrage ungefähr in der richtigen Größe (die mit der Aufgabe etwas zu tun hat) ohne offensichtliche, schwere semantische Fehler.
- 1 Punkt für die Deklaration der zwei Tupelvariablen (FROM-Klausel)
- 1 Punkt für die richtige Join-Bedingung (unter FROM oder WHERE)
- 1.5 Punkte für die NOT LIKE-Bedingung mit dem berechneten Muster.
- 0.5 Punkte für die SELECT-Klausel mit Umbenennung der Spalten.

Abzüge z.B.

- 0.5 Punkte für DISTINCT (TITEL ist Alternativschlüssel)
- 0.5 Punkte für Join zusätzlicher Tabellen, sofern das Ergebnis dadurch nicht falsch wird.

- c) Gesucht sind Lehrveranstaltungen, die mehr Teilnehmer haben, als es MAX_TN erlaubt. Geben Sie Folgendes aus:

- Den TITEL der Lehrveranstaltung,
- Die tatsächliche Teilnehmerzahl (Ausgabespalte “ANGEMELDET”),
- Die maximale Teilnehmerzahl (Spalte MAX_TN der Tabelle LV)

Ergebnis		
TITEL	ANGEMELDET	MAX_TN
Übung 1 zu OOP	4	3

Lösung: *Dies ist ein Beispiel für eine Anfrage mit GROUP BY und HAVING:*

```
SELECT L.TITEL, COUNT(*) AS ANGEMELDET, L.MAX_TN
FROM LV L, TEILNEHMER T
WHERE L.VID = T.VID
GROUP BY L.TITEL, L.MAX_TN
HAVING COUNT(*) > L.MAX_TN
```

Mögliche Bepunktung:

- 1 Punkt für syntaktisch korrekte Anfrage ungefähr in der richtigen Größe (die mit der Aufgabe etwas zu tun hat) ohne offensichtliche, schwere semantische Fehler.
- 1 Punkt für Tupelvariablen und Join-Bedingung (FROM- und WHERE-Klausel)
- 1 Punkt für GROUP BY
- 1 Punkt für HAVING
- 1 Punkt für die korrekte SELECT-Klausel mit COUNT.

- d) Welche 8-Uhr und 10-Uhr Termine an Arbeitstagen sind noch frei? Arbeitstage sind Wochentage, bei denen in der Spalte WOCHENENDE der Wert N steht. Frei bedeutet, dass dort noch keine Lehrveranstaltung eingetragen ist. Sortieren Sie das Ergebnis nach dem Wochentag (in der üblichen Reihenfolge nach den Nummern) und bei gleichem Wochentag nach der Stunde (8 oder 10). Geben Sie die Nummer des Wochentags mit aus, damit das Sortieren problemlos möglich ist.

Ergebnis		
TAG_NR	TAG	ZEIT
1	MO	8
3	MI	8
3	MI	10
4	DO	8
5	FR	8
5	FR	10

Wenn Sie es mit beiden Stunden (8 und 10 Uhr) nicht hinbekommen, wäre es ein Anfang, nur die 8-Uhr Termine an allen Wochentagen (außer Wochenende) zu berücksichtigen. Natürlich würde es nur einen Teil der Punkte dafür geben.

Lösung: *Die Anfrage braucht ein UNION ALL:*

```
SELECT NR as TAG_NR, TAG, 8 AS ZEIT
FROM WOCHENTAGE W
WHERE W.WOCHENENDE = 'N'
```

```
AND      NOT EXISTS(SELECT * FROM LV V
                      WHERE V.TAG=W.TAG AND V.STUNDE=8)
UNION ALL
SELECT NR AS TAG_NR, TAG, 10 AS ZEIT
FROM      WOCHENTAGE W
WHERE     W.WOCHENENDE = 'N'
AND      NOT EXISTS(SELECT * FROM LV V
                      WHERE V.TAG=W.TAG AND V.STUNDE=10)
ORDER BY TAG_NR, ZEIT
```

Alternativ:

```
WITH KANDIDATEN AS
(SELECT NR, TAG, 8 as ZEIT, WOCHENENDE FROM WOCHENTAGE
 UNION ALL
 SELECT NR, TAG, 10 as ZEIT, WOCHENENDE FROM WOCHENTAGE)
SELECT NR AS TAG_NR, TAG, ZEIT
FROM      KANDIDATEN K
WHERE     WOCHENENDE = 'N'
AND      NOT EXISTS(SELECT * FROM LV V
                      WHERE V.TAG=K.TAG AND V.STUNDE=K.ZEIT)
ORDER BY TAG_NR, ZEIT
```

Mögliche Bepunktung:

- 0.5 Punkte für syntaktisch korrekte Anfrage ungefähr in der richtigen Größe (die mit der Aufgabe etwas zu tun hat) ohne offensichtliche, schwere semantische Fehler.
- 1 Punkt für Verwendung von NOT EXISTS (syntaktisch korrekt) mit Tabelle LV (plus ggf. weiteren Tabellen)
- 1 Punkt für nur die Tabelle LV im NOT EXISTS und korrekte WHERE-Bedingung im NOT EXISTS
- 1.5 Punkte für die korrekte SELECT-Klausel mit Konstante und ORDER BY
- 1 Punkt für UNION ALL.

Anfrage für die abgemagerte Version (gibt 4 von 5 Punkten):

```
SELECT NR as TAG_NR, TAG, 8 AS ZEIT
FROM      WOCHENTAGE W
WHERE     W.WOCHENENDE = 'N'
AND      NOT EXISTS(SELECT * FROM LV V
                      WHERE V.TAG=W.TAG AND V.STUNDE=8)
ORDER BY TAG_NR, ZEIT
```

e) Geben Sie für jede Lehrveranstaltung Folgendes aus:

- den Titel der Lehrveranstaltung,
- die Anzahl der Studiengänge (Spalte FACH in STUDENTEN), aus denen es Teilnehmer gibt,
- die Gesamtanzahl der Teilnehmer, sowie
- die Anzahl der Teilnehmer, die weder Informatik noch Bioinformatik studieren (also von außerhalb des Instituts kommen).

Dabei sollen auch Lehrveranstaltungen mit ausgegeben werden, die 0 Teilnehmer haben, bzw. 0 Teilnehmer von außerhalb der Informatik.

Ergebnis			
TITEL	ANZ_FAECHER	ANZ_TN	OHNE_INF
OOP	2	3	1
Übung 1 zu OOP	3	4	1
Übung 2 zu OOP	1	1	0
Datenbanken	2	2	0
DB-Übung	2	2	0
XML+DBen	0	0	0

Im Beispieldzustand ist der eine Teilnehmer unter OHNE_INF Michael Grau, der Mathematik studiert.

Falls Sie die letzte Spalte nicht hinbekommen, lassen Sie diese weg. Natürlich kostet das Punkte.

Lösung: Diese Anfrage benötigt COUNT DISTINCT, einen Outer Join oder UNION oder Unteranfragen unter SELECT, und ggf. CASE oder einen zusätzlichen Join mit einer Hilfstabelle, um über einer Teilmenge zu zählen.

```

SELECT V.TITEL,
       (SELECT COUNT(DISTINCT S.FACH)
        FROM TEILNEHMER T, STUDENTEN S
        WHERE T.VID=V.VID AND S.SID=T.SID)
       AS ANZ_FAECHER,
       (SELECT COUNT(*)
        FROM TEILNEHMER T
        WHERE T.VID=V.VID)
       AS ANZ_TN,
       (SELECT COUNT(*)
        FROM TEILNEHMER T, STUDENTEN S
        WHERE T.VID=V.VID AND T.SID=S.SID
        AND S.FACH<>'Informatik' AND S.FACH<>'Bioinformatik')
       AS OHNE_INF
FROM LV V

```

Alternativ:

```
SELECT V.TITEL, COUNT(DISTINCT S.FACH) AS ANZ_FAECHER,  
       COUNT(T.SID) AS ANZ_TN,  
       COUNT(CASE WHEN S.FACH='Informatik' THEN NULL  
                WHEN S.FACH='Bioinformatik' THEN NULL  
                ELSE S.FACH END) AS OHNE_INF  
FROM   LV V LEFT JOIN TEILNEHMER T USING(VID)  
       LEFT JOIN STUDENTEN S USING(SID)  
GROUP BY V.TITEL
```

Mögliche Bepunktung:

- 0.5 Punkte für syntaktisch korrekte Anfrage ungefähr in der richtigen Größe (die mit der Aufgabe etwas zu tun hat) ohne offensichtliche, schwere semantische Fehler.
- 1 Punkt für korrektes GROUP BY (bzw. geschachtelte Anfragen) und SELECT-Klausel mit TITEL und ohne DISTINCT.
- 1 Punkt für korrektes COUNT DISTINCT
- 0.5 Punkte für korrekte Berechnung von ANZ_TN
- 1 Punkt für korrekten Outer Join oder geschachtelte Anfragen oder UNION ALL
- 1 Punkt für die korrekte Berechnung der Spalte OHNE_INF

Abgemagerte Version (gibt 4 von 5 Punkten):

```
SELECT V.TITEL, COUNT(DISTINCT S.FACH) AS ANZ_FAECHER,  
       COUNT(T.SID) AS ANZ_TN  
FROM   LV V LEFT JOIN TEILNEHMER T USING(VID)  
       LEFT JOIN STUDENTEN S USING(SID)  
GROUP BY V.TITEL
```

Aufgabe 2 (Äquivalenz von Anfragen [ILIAS])**6 Punkte**

Gegeben sind jeweils drei Anfragen. Welche davon sind äquivalent, d.h. liefern die gleiche Antwort in jedem Datenbank-Zustand, der die Integritätsbedingungen (Schlüssel, Fremdschlüssel, NOT NULL) erfüllt? Es sei natürlich auch vorausgesetzt, dass die Vergleichsoperatoren die übliche Semantik haben.

a) **Anfrage 1:**

```
SELECT *  
FROM   LV  
WHERE  TAG NOT IN (SELECT TAG FROM WOCHENTAGE)
```

Anfrage 2:

```
SELECT *  
FROM   LV  
WHERE  TAG = 'MI' OR TAG IS NULL
```

Anfrage 3:

```
SELECT *  
FROM   LV  
WHERE  MAX_TN < 50 AND MAX_TN > 100
```

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ Alle drei sind äquivalent
- ☐ 1 und 2 sind äquivalent, 3 nicht
- ☐ 1 und 3 sind äquivalent, 2 nicht
- ☐ 2 und 3 sind äquivalent, 1 nicht
- ☐ Alle drei sind unterschiedlich (keine zwei sind äquivalent)

b) **Anfrage 1:**

```
SELECT SID, VORNAME, NACHNAME
FROM   STUDENTEN
WHERE  SID IN (SELECT SID FROM TEILNEHMER WHERE VID = 1)
```

Anfrage 2:

```
SELECT S.SID, S.VORNAME, S.NACHNAME
FROM   STUDENTEN S, TEILNEHMER T
WHERE  S.SID = T.SID AND T.VID = 1
```

Anfrage 3:

```
SELECT S.SID, S.VORNAME, S.NACHNAME
FROM   STUDENTEN S, TEILNEHMER X, TEILNEHMER Y, LV V
WHERE  S.SID = X.SID AND S.SID = Y.SID
AND    X.VID = Y.VID AND Y.VID = V.VID AND V.VID = 1
```

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ Alle drei sind äquivalent
- ☐ 1 und 2 sind äquivalent, 3 nicht
- ☐ 1 und 3 sind äquivalent, 2 nicht
- ☐ 2 und 3 sind äquivalent, 1 nicht
- ☐ Alle drei sind unterschiedlich (keine zwei sind äquivalent)

c) **Anfrage 1:**

```
SELECT V.VID, V.TITEL
FROM   LV V
WHERE  V.UEBUNG_ZU IS NOT NULL
AND    NOT EXISTS(SELECT MAX_TN
                    FROM    LV X
                    WHERE    X.UEBUNG_ZU IS NOT NULL
                    AND      X.MAX_TN > V.MAX_TN)
```

Anfrage 2:

```
SELECT DISTINCT V.VID, V.TITEL
FROM   LV V, LV X
WHERE  V.UEBUNG_ZU IS NOT NULL
AND    X.UEBUNG_ZU IS NOT NULL
AND    V.MAX_TN > X.MAX_TN
```

Anfrage 3:

```
SELECT V.VID, V.TITEL
FROM   LV V
WHERE  V.UEBUNG_ZU IS NOT NULL
AND    NOT V.MAX_TN < ANY (SELECT MAX_TN
                           FROM    LV
                           WHERE    UEBUNG_ZU = UEBUNG_ZU)
```

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ Alle drei sind äquivalent
- ☐ 1 und 2 sind äquivalent, 3 nicht
- ☐ 1 und 3 sind äquivalent, 2 nicht
- ☐ 2 und 3 sind äquivalent, 1 nicht
- ☐ Alle drei sind unterschiedlich (keine zwei sind äquivalent)

Aufgabe 3 (Relationale Algebra [ILIAS])**4 Punkte**

- a) Wie heisst der Operator σ in der relationalen Algebra?

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ Selektion
- ☐ Projektion
- ☐ Kartesisches Produkt
- ☐ Join

- b) Gegeben seien zwei Relationen $R(A, B)$ und $S(B, C)$. Welchem Ausdruck der relationalen Algebra entspricht der natürliche Verbund $R \bowtie S$?

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ $\sigma_{R.B=S.B}(\rho_R(R) \times \rho_S(S))$
- ☐ $\pi_{A,B,C}(\sigma_B(R) \times \sigma_B(S))$
- ☐ $\pi_{A,B,C}(\sigma_{B=B'}(R \times \pi_{B' \leftarrow B,C}(S)))$
- ☐ $\sigma_{A=B}(R \times S)$

- c) Gegeben sei eine Relation $R(A, B, C)$. Was ist der einfachste Ausdruck der relationalen Algebra, der `SELECT * FROM R` entspricht?

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ $\pi_*(R)$
- ☐ $\pi_{A,B,C}(R)$
- ☐ $\sigma_{A,B,C}(R)$
- ☐ R

- d) Nehmen Sie an, die Tabellen Vorlesungen und Übungen waren ursprünglich in getrennten Tabellen gespeichert:

VORLESUNG(VID, TITEL, TAG, STUNDE, MAX_TN)

UEBUNG(VID, TITEL, TAG, STUNDE, UEBUNG_ZU->VORLESUNG, MAX_TN)

Später entdeckt man die vielen gemeinsamen Spalten und andere Schwierigkeiten dieses Entwurfs und möchte zur Tabelle LV kommen, wie sie im vorn Beispiel gezeigt wird. Mit welchem Ausdruck der relationalen Algebra kommt man von den Tabellen VORLESUNG und UEBUNG zur Tabelle LV?

Bitte nicht hier, sondern in ILIAS antworten!

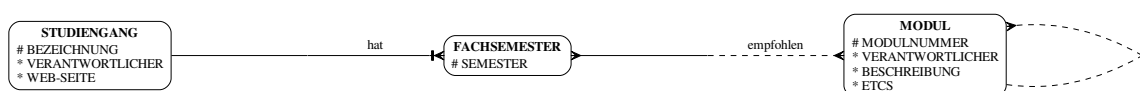
- ☐ $UEBUNG \bowtie VORLESUNG$
- ☐ $UEBUNG \cup \pi_{VID,TITEL,TAG,STUNDE,UEBUNG_ZU<-NULL,MAX_TN}(VORLESUNG)$
- ☐ $\pi_{VID,TITEL,TAG,STUNDE,MAX_TN}(UEBUNG) \cup VORLESUNG$
- ☐ $UEBUNG \bowtie \pi_{VID,TITEL,TAG,STUNDE,UEBUNG_ZU<-NULL,MAX_TN}(VORLESUNG)$

Aufgabe 4 (ER-Entwurf)**6 Punkte**

Entwerfen Sie ein ER-Diagramm in der ER-Notation aus der Vorlesung (Barker-Notation). Das gesuchte ER-Diagramm soll die Regelstudienpläne verschiedener Studiengänge, deren Module und die Empfehlungen für Fachsemester modellieren.

- Ein Studiengang wird durch eine **BEZEICHNUNG** identifiziert. Er hat immer einen **VERANTWORTLICHEN** und eine **WEB-SEITE**.
- Ein Fachsemester eines Studiengangs wird durch die **BEZEICHNUNG** des zugehörigen Studiengangs und das **SEMESTER** (1, 2, 3, usw.) identifiziert. Ein Fachsemester gehört zu genau einem Studiengang. Ein Studiengang hat mindestens ein Fachsemester. Natürlich haben die meisten Studiengänge mehrere Fachsemester.
- Ein Modul wird durch eine **MODULNUMMER** identifiziert. Es hat immer einen **VERANTWORTLICHEN**, eine **BESCHREIBUNG** und **ETCS-PUNKTE**.
- Ein Modul kann für mehrere Fachsemester empfohlen sein (dann lässt auch der Regelstudienplan den Studierenden freie Wahl). Ein Modul kann aber auch für kein Fachsemester empfohlen sein (damit der Regelstudienplan nicht völlig überfrachtet wird, z.B. Anwendungsfach-Module nicht Semestern zuzuordnen). Umgekehrt ist jedem Fachsemester mindestens ein Modul zugeordnet, meist auch mehrere.
- Ein Modul kann die Voraussetzung für viele Module sein, aber es muss nicht Voraussetzung für ein anderes Modul sein. Ein Modul muss kein anderes Modul als Voraussetzung haben, aber es kann mehrere Module als Voraussetzungen haben.

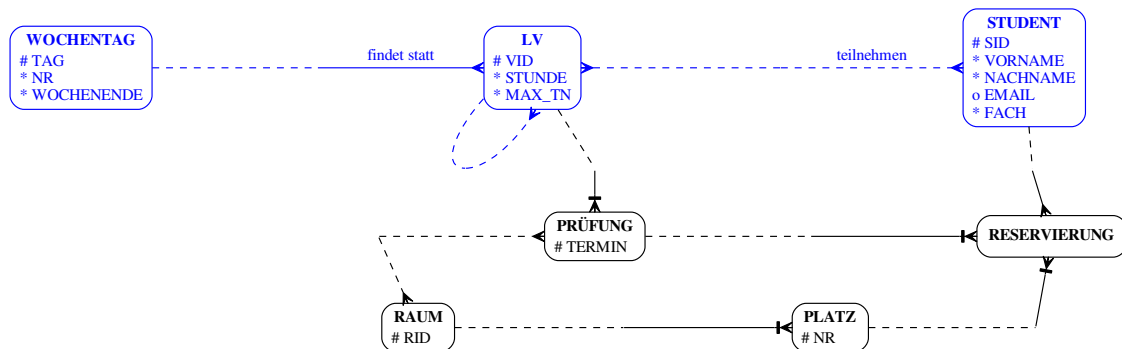
Geben Sie die Primärschlüssel der Entity-Typen und die minimalen bzw. maximalen Kardinalitäten der Relationships entsprechend der Notation an! Rollennamen der Relationships dürfen weggelassen werden.

Lösung:

Aufgabe 5 (Relationales Modell [ILIAS])

5 Punkte

Das folgende erweiterte ER-Diagramm modelliert die Verwaltung von elektronischen Prüfungen.



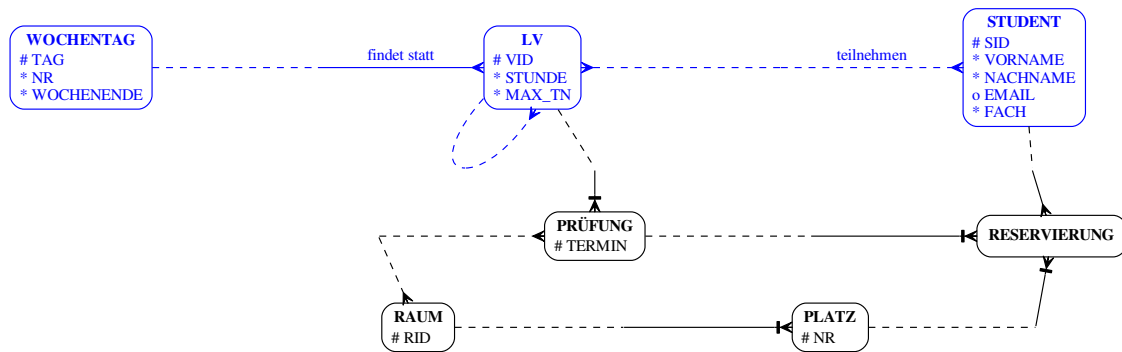
Die Übersetzung der Entities WOCHENTAG, LV und STUDENT sowie der Relationships zwischen diesen Entities in blauer Farbe entspricht den gegebenen Relationen

- WOCHENTAGE(TAG, NR, WOCHENENDE)
- LV(VID, TITEL, TAG → WOCHENTAGE, STUNDE, UEBUNG_ZU° → LV, MAX_TN)
- STUDENTEN(SID, VORNAME, NACHNAME, EMAIL°, FACH)
- TEILNEHMER(SID → STUDENTEN, VID → LV)

Übersetzen Sie die restlichen Entities und Relationships des ER-Schema in schwarzer Farbe ins Relationale Modell.

- Dafür finden Sie in der Tabelle auf der nächsten Seite in der linken Spalte eine Liste von möglichen Relationen. Die Liste enthält Relationen für die Entities und für alle Relationships. Aber aufgepasst: nicht alle Relationen in der Liste werden für die Übersetzung benötigt.
- Auf der rechten Seite der Tabelle finden Sie eine Liste von Attributen, die manchmal auch als Primär- bzw. Fremdschlüssel markiert sind. Ziehen Sie die Attribute mit den Markierungen auf die linke Seite zur richtigen Relation. Aber auch hier wieder aufgepasst: nicht alle Attribute in der Liste werden für die Übersetzung benötigt. Manche Attribute müssen jedoch mehrfach verwendet werden.
- Wenn Sie eine Relation auf der linken Seite leer lassen, d.h. kein Attribut zuordnen, dann drücken Sie damit aus, dass diese Relation für die Übersetzung nicht benötigt wird. Hinweis: mehrere Relationen müssen für die richtige Lösung leer bleiben.
- Wegen der Singular- und Pluralform der Tabellennamen machen Sie sich keine Sorgen.
- Für richtige Zuordnungen erhalten Sie 0.5 Punkte. Für falsche Zuordnungen werden Ihnen 0.25 Punkte abgezogen. Die Gesamtpunktzahl kann jedoch nicht unter Null sinken.

Das ER-Diagramm ist auf der nächsten Seite wiederholt, damit Sie es zusammen mit den möglichen Relationen und Attributen anschauen können.



Notation (Beispiele):

- #NR markiert, dass das Attribut NR Teil des Primärschlüssels ist.
- SID → STUDENTEN bedeutet, dass das Attribut SID ein Fremdschlüssel ist, der die Tabelle STUDENTEN referenziert.
- (#SID, #VID) → TEILNEHMER gibt an, dass die Attribute SID und VID Teil des Primärschlüssels sind und die Kombination (SID, VID) ein Fremdschlüssel auf die Relation TEILNEHMER ist.

Relationen	Attribute
LV_PRÜFUNG	#NR #NR → Platz
PLATZ	#RID #RID → PLATZ
PLATZ_RESEVIERUNG	#RID → RAUM
PRÜFUNG	(#RID, #NR) → PLATZ
PRÜFUNG_RAUM	SID #SID SID → STUDENTEN #SID → STUDENTEN
PRÜFUNG_RESERVIERUNG	#TERMIN #TERMIN → PRÜFUNG
RAUM	VID #VID #VID → LV #VID → PRÜFUNG
RAUM_PLATZ	(#VID, #TERMIN) → PRÜFUNG
RESERVIERUNG	
STUDENT_RESERVIERUNG	

Lösung:

- PRÜFUNG(VID → LV, TERMIN, ANMELDUNG, RÜCKTRITT)
- RAUM(RID, BESCHREIBUNG)
- PLATZ(RID → RAUM, NR, NOTEBOOK-MODELL)
- ZUTEILUNG((VID, TERMIN) → PRÜFUNG, (RID, NR) → PLATZ, SID → STUDENTEN)

- $\text{PRÜFUNG_RAUM}(\langle \underline{\text{VID}}, \underline{\text{TERMIN}} \rangle \rightarrow \text{PRÜFUNG}, \underline{\text{RID}} \rightarrow \text{RAUM})$

Aufgabe 6 (Bonusaufgabe: BCNF [ILIAS])**4 Punkte**

Es wurde eine neue Tabelle für Lehrveranstaltungen entworfen, die auch mehrere Termine zu einer Veranstaltung erlaubt. Im Prinzip brauchen Sie die Anwendung aber nicht zu verstehen, da die funktionalen Abhängigkeiten (FAen) gegeben sind, und dann alles ganz formal abläuft. Die Tabelle ist:

NEU(VID, TITEL, TAG, STUNDE, RAUM, MAX_TN)

Folgende funktionalen Abhängigkeiten sollen gelten:

- $VID \rightarrow TITEL, MAX_TN$
- $TITEL \rightarrow VID$
- $VID, TAG, STUNDE \rightarrow RAUM$
- $TAG, STUNDE, RAUM \rightarrow VID$

a) Was ist die Attributhülle $\{TITEL\}^+$?

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ $\{TITEL\}$
- ☐ $\{VID\}$
- ☐ $\{TITEL, VID\}$
- ☐ $\{TITEL, VID, MAX_TN\}$
- ☐ $\{VID, MAX_TN\}$
- ☐ $\{TITEL, VID, MAX_TN, TAG, STUNDE, RAUM\}$

b) Ist $\{VID, TAG, STUNDE\}$ ein Schlüssel?

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ Ja, ein minimaler Schlüssel
- ☐ Ja, ein Schlüssel, aber nicht minimal
- ☐ Nein

c) Welche der folgenden FAen werden von den gegebenen FAen logisch impliziert? Kreuzen Sie die erste FA an, die diese Bedingung erfüllt.

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ $RAUM \rightarrow TAG, STUNDE$
- ☐ $VID \rightarrow TAG, STUNDE$
- ☐ $TAG, STUNDE, RAUM \rightarrow MAX_TN$
- ☐ Keine dieser funktionalen Abhängigkeiten wird impliziert

d) Ist die Relation in BCNF?

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ Ja, die Relation ist in BCNF.
- ☐ Nein, die ersten zwei FAen verletzen beide BCNF.
- ☐ Nein, die letzten zwei FAen verletzen beide BCNF.
- ☐ BCNF ist verletzt, aber von einer anderen Teilmenge der FAen.

Aufgabe 7 (Bonusaufgabe: Evaluation [ILIAS])**1 Punkt**

An dieser Vorlesung soll eine neue Modul-Evaluation getestet werden. Die TANs für die Online-Umfrage werden am 26.02.2020 verschickt und die Umfrage läuft dann bis zum 10.03.2020.

Werden Sie an dieser Umfrage teilnehmen?

Die richtige Antwort, also die, die einen Punkt gibt, lautet “ja”. Vermutlich werden wir nicht erfahren, wer tatsächlich teilgenommen hat. Sie sollten aber einen gewissen moralischen Druck verspüren, auch bei der Umfrage teilzunehmen, wenn Sie sich den Bonuspunkt holen.

Bitte nicht hier, sondern in ILIAS antworten!

- ☐ Ja, ich werde an der Umfrage teilnehmen.
- ☐ Nein, ich werde nicht an der Umfrage teilnehmen.