Lehrstuhl für Datenbanksysteme
Fakultät für Informatik
Technische Universität München

Esolution
Sticker mit SRID hier einkleben

**Hinweise zur Personalisierung:**

- Ihre Prüfung wird bei der Anwesenheitskontrolle durch Aufkleben eines Codes personalisiert.
- Dieser enthält lediglich eine fortlaufende Nummer, welche auch auf der Anwesenheitsliste neben dem Unterschriftenfeld vermerkt ist.
- Diese wird als Pseudonym verwendet, um eine eindeutige Zuordnung Ihrer Prüfung zu ermöglichen.

# Einsatz und Realisierung von Datenbanksystemen

| | | | |
|---|---|---|---|
| **Klausur:** | IN2031 / Probe | **Datum:** | Donnerstag, 23. Juli 2020 |
| **Prüfer:** | Prof. Dr. Alfons Kemper | **Uhrzeit:** | 11:00 – 12:30 |



## Bearbeitungshinweise

- Diese Klausur umfasst **12 Seiten** mit insgesamt **11 Aufgaben**.
  Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Die Gesamtpunktzahl in dieser Prüfung beträgt 90 Punkte.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- Als Hilfsmittel sind zugelassen:
  - ein **analoges Wörterbuch** Deutsch ↔ Muttersprache **ohne Anmerkungen**
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Ergebnisse vorheriger Teilaufgaben lösbar.
- **Es werden nur solche Ergebnisse gewertet, bei denen der Lösungsweg erkennbar ist.** Auch Textaufgaben sind **grundsätzlich zu begründen**, sofern es in der jeweiligen Teilaufgabe nicht ausdrücklich anders vermerkt ist.
- Schreiben Sie weder mit roter / grüner Farbe noch mit Bleistift.
- Schalten Sie alle mitgeführten elektronischen Geräte vollständig aus, verstauen Sie diese in Ihrer Tasche und verschließen Sie diese.

Hörsaal verlassen von _____ bis _____ / Vorzeitige Abgabe um _____

## Aufgabe 1 Recovery (6 Punkte)

0 1 2

a)* Für was steht ACID? (4 Stichworte)

Atomicity, Consistency, Isolation, Durability

0 1

b)* Für was steht WAL-Prinzip? (Langnamen der Abkürzung angeben)

Write Ahead Logging

0 1

c)* Vor welchen Ereignissen muss das, was das WAL-Prinzip besagt, vollständig abgeschlossen sein? (Stichwort)

TX Ende / Commit oder Page Write

0 1 2

d)* Nennen Sie die drei Phasen des Wiederanlaufs in chronologischer Reihenfolge. (3 Stichworte)

Analyse Phase, Redo, Undo

## Aufgabe 2 Historien (8 Punkte)

Gegeben sei die folgende Historie:

$$r_1(y) \rightarrow w_1(y) \rightarrow r_1(x) \rightarrow w_2(y) \rightarrow w_1(y) \rightarrow c_1 \rightarrow c_2 \rightarrow r_3(x) \rightarrow w_3(x) \rightarrow c_3$$

a)* Kreuzen Sie an, ob die Eigenschaften von der Historie erfüllt werden.

- [x] Vermeidet kaskadierendes Zurücksetzen (ACA)
- [ ] Strikt (ST)
- [ ] Diese Historie kann von einem 2PL-Scheduler erzeugt worden sein.
- [ ] Diese Historie kann von einem strikten 2PL-Scheduler erzeugt worden sein.
- [ ] Serialisierbar (SR)
- [x] Rücksetzbar (RC)

0 1 2

b)* Fügen Sie Commits in die folgende Historie H so ein, dass die Historie RC aber nicht ACA erfüllt:

$$H = w_1(x) \rightarrow r_2(x) \rightarrow w_2(y)$$

$w_1(x) \rightarrow r_2(x) \rightarrow w_2(y) \rightarrow c_1 \rightarrow c_2$

## Aufgabe 3 IT-Sicherheit (6 Punkte)

Gegeben sei die Tabelle `PrAnmeldung` mit einigen Beispielwerten. Die Tabelle speichert für alle Studenten, zu welchen Prüfungen sie angemeldet sind.

`PrAnmeldung`

| MatrNr | Name | Pruefung |
|---|---|---|
| 03642000 | Thuy | NetSec |
| 03654321 | Alex | Modern DBs |
| 03636363 | Anna | Robotics |
| ⋮ | ⋮ | ⋮ |

Es gibt eine Website, in der man sich nach Eingabe der Matrikelnummer alle seine Prüfungsanmeldungen auflisten lassen kann. Leider hat das Entwicklungsteam vergessen, die Benutzereingabe auf SQL-Injections zu prüfen. Die im Formular genutzte Anfrage lautet:

```
SELECT Pruefung FROM PrAnmeldung WHERE MatrNr={Benutzereingabe};
```

Schreiben Sie folgende SQL-Injections für das Feld `{Benutzereingabe}`:

a)* Sie haben vergessen, sich rechtzeitig für ERDB anzumelden. Schreiben Sie eine SQL-Injection, um sich mit Ihrer Matrikelnummer (`MatrNr`) und Ihrem Namen (`Name`) zur Prüfung (`Pruefung`) „ERDB“ anzumelden.

0 1 2 3 4

```
0;INSERT INTO PrAnmeldung (MatrNr, Name, Pruefung) VALUES (\{MatrNr\}, "\{Name\}", "ERDB");--
```

b)* Die Manipulation ist aufgefallen und Sie verwischen Ihre Spuren. Schreiben Sie eine SQL-Injection, um die gesamte Tabelle zu löschen.

0 1 2

```
0;DROP TABLE PrAnmeldung;--
```

## Aufgabe 4 Datalog (7 Punkte)

Gegeben seien die Fakten `voraussetzen` und `vorlesungen` aus der Universitätsdatenbank (hier nur in Ausschnitten dargestellt):

```
%%voraussetzen(VorgNr,NachfNr)
voraussetzen(5001,5041).
voraussetzen(5001,5043).
voraussetzen(5043,5022).
...
%%Vorlesungen(VorlNr, Titel, SWS, gelesenVon)
vorlesungen(5001,grundzuege,4,2137).
vorlesungen(5022,wissenschaftstheorie,3,2126).
vorlesungen(5041,ethik,4,2125).
vorlesungen(5043,erkenntnistheorie,3,2126).
...
```

Gegeben sei der folgende Ausdruck in Domänenkalkül:

$$\{[t] \mid \exists\, v,s,g([v,t,s,g] \in \text{Vorlesungen} \wedge \exists\, n([v,n] \in \text{voraussetzen} \wedge \exists\, s2,g2([n,\text{'Ethik'},s2,g2] \in \text{Vorlesungen})))\}$$

0 1

a)* Nennen Sie den Zweck des Ausdrucks.

`Vorlesungen`, die Ethik voraussetzt (Vorgänger von Ethik).

0 1 2 3

b)* Übersetzen Sie den Ausdruck in Datalog.

```
ethikBraucht(T) :- vorlesungen(V,T,_,_), voraussetzen(V,N),
                   vorlesungen(N,'ethik',_,_).
```

0 1 2 3

c)* Gegeben seien die folgenden Datalog-Prädikate:

```
vorgaenger(V,N) :- voraussetzen(V,N).
vorgaenger(V,N) :- vorgaenger(V,N),voraussetzen(V,N).
nichtVoraussetzen(V,NV) :- vorlesungen(V,_,_,_),vorlesungen(NV,_,_,_),
                           not(vorgaenger(V,NV)).
```

Begründen Sie stichpunktartig, ob das Prädikat `nichtVoraussetzen` stratifiziert ist.

`nichtVoraussetzen` hat genau ein negiertes Prädikat `vorgaenger`
`vorgaenger` hängt nicht von `nichtVoraussetzen` ab
=> stratifiziert

## Aufgabe 5 Fragmentierung (8 Punkte)

Gegeben sei folgende Relation `Klausur` mit Schlüssel `MatrNr`:

| MatrNr | Name | Note | Standort |
|---|---|---|---|
| 10101 | Philipp | 1,0 | München |
| 10102 | Magdalena | 1,0 | Garching |
| 10103 | Erik | 1,0 | Garching |
| 10104 | Josef | 1,0 | Garching |
| 10105 | Alex | 1,0 | Garching |
| 10106 | Maxmilian | 1,0 | München |

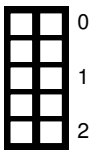
Für eine verteilte Datenbank soll die Tabelle geeignet fragmentiert werden. Ziel ist, Namen mit Standort der Studenten lokal und die Noten getrennt abzuspeichern.

Fragmentieren Sie die Relation geeignet *vertikal*.

a)* Geben Sie das Schema für die zwei resultierenden Relationen $KlausurV_1$ und $KlausurV_2$ an.


$KlausurV_1 : \{[\underline{MatrNr}, Note]\}, KlausurV_2\{[\underline{MatrNr}, Name, Standort]\}$

b) * Geben Sie in SQL-92 die zwei resultierenden Relationen `KlausurV1` und `KlausurV2` als Hilfstabellen (mittels `with`) an.



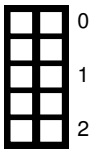

```
with KlausurV1 as (SELECT MatrNr,Note FROM Klausur),
     KlausurV2 as (SELECT MatrNr,Name,Standort FROM Klausur)
```

Die geeignetere der beiden resultierenden Relationen soll *horizontal* fragmentiert werden.

c)* Geben Sie das Prädikat der Selektion an, mit dem fragmentiert wird.

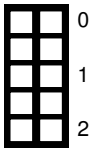

Standort='Garching' oder Standort='München'

d)* Geben Sie in SQL-92 die zwei resultierenden Relationen `KlausurH1` und `KlausurH2` als Hilfstabellen (mittels `with`) an.




```
with KlausurH1 as(SELECT * FROM KlausurV2 WHERE Standort='Garching'),
     KlausurH2 as(SELECT * FROM KlausurV2 WHERE Standort<>'Garching')
```

e) Schreiben Sie eine SQL-Abfrage, die die Ursprungsrelation aus den Teilrelationen zusammensetzt.




```
select KlausurV2.*, KlausurV1.Note
from KlausurV1,
  (select * from KlausurH1 union select * from KlausurH2) as KlausurV2
where KlausurV1.MatrNr=KlausurV2.MatrNr
```

## Aufgabe 6 Window-Functions (10 Punkte)

Betrachten Sie die folgende Tabelle `Waren` mit verkauften Produkten in einem Supermarkt. Die Spalte `verkauft` besagt, wieviele Einheiten des jeweiligen Produktes verkauft worden sind.

| Name | Preis | Kategorie | Verkauft |
|---|---|---|---|
| Brot | 1.00 | Backwaren | 8128 |
| Butter | 0.80 | Kühlwaren | 496 |
| Grill | 60.00 | Haushalt | 6 |
| Steak | 8.00 | Kühlwaren | 28 |
| ... | ... | ... | ... |

0 1 2 3

a)* Ermitteln Sie in SQL mittels Fensterfunktionen (Windowfunctions) den prozentualen Umsatzanteil jedes Produktes innerhalb seiner Kategorie.

```
select name,kategorie,
       verkauft * preis * 100.0 / sum(verkauft * preis)
       over (partition by kategorie)
from Waren;
```

0 1 2 3

b)* Ermitteln Sie in SQL mittels Fensterfunktionen (Windowfunctions) für jedes Produkt das Mittel der Verkaufszahlen aus den 5 besser verkauften (höhere Verkaufszahlen) Produkten geordnet nach Verkaufszahlen.

```
select name, avg(verkauft) over (order by verkauft desc
           rows between 5 preceding and current row)
from Waren;
```

0 1 2 3 4

c)* Ermitteln Sie in SQL mittels Fensterfunktionen (Windowfunctions) die drei Produkte mit dem meisten Umsatz pro Kategorie.

```
select *
from ( select *, rank() over (partition by kategorie
                              order by (verkauft * preis) desc)
       from waren)
where rank <= 3;
```

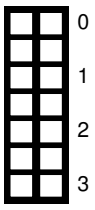
## Aufgabe 7 Skyline (6 Punkte)

Gegeben sei die Relation Zuege:

| Baureihe | Gewicht | Vmax |
|---|---|---|
| 401 | 795 | 280 |
| 402 | 455 | 280 |
| 403 | 459 | 330 |
| 406 | 488 | 330 |
| 407 | 495 | 320 |
| 411 | 402 | 230 |
| 412 | 670 | 250 |
| 415 | 311 | 230 |

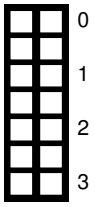
Wir betrachten die Skyline über das Minimum des Attributs *Gewicht* und das Maximum über das Attribut *Vmax*.

a)* Geben Sie die Anfrage, die die oben genannte Skyline in SQL-92 berechnet, an (d.h. ohne Skyline-Operator).




```
SELECT * FROM Zuege z WHERE NOT EXISTS (
  SELECT * FROM Zuege dom WHERE
  (dom.Gewicht <= z.Gewicht AND dom.Vmax >= z.Vmax) AND
  (dom.Gewicht < z.Gewicht OR dom.Vmax > z.Vmax)
)
```

b)* Geben Sie alle Tupel an, die in der Skyline enthalten sind. Es reicht, wenn Sie die zutreffenden Baureihen angeben.




402,403,415

## Aufgabe 8 Hauptspeicher-Datenbanken (9 Punkte)

Sie sollen für die Alexander-Maximilians-Universität (AMU) ein Hauptspeicherdatenbanksystem optimieren. In dem System sind die Daten aller Studenten gespeichert.

**Relationen**

*Studenten*: MatrNr (8 Byte), Name (48 Byte), Studiengang (4 Byte), Semester (4 Byte)
MatrNr ist der Primärschlüssel der indiziert ist.
Schätzen Sie für jede der untenstehenden Anfragen einzeln, ob ein Row- oder Column-Store besser geeignet ist.

a)* `select * from Studenten;`

- [x] Column-Store
- [x] Row-Store

b)* `select Semester, count(*) from Studenten group by Semester;`

- [ ] Row-Store
- [x] Column-Store

c)* `select Name, Studiengang, Semester from Studenten where MatrNr = 42;`

- [x] Row-Store
- [ ] Column-Store

d)* `select Studiengang from Studenten where MatrNr = 42;`

- [ ] Row-Store
- [x] Column-Store

e)* `select * from Studenten where Semester < 5;`

- [x] Row-Store
- [x] Column-Store

f)* `select * from Studenten where Semester = 25;`

- [ ] Row-Store
- [x] Column-Store

g)* `insert into studenten values(4242, Max Meyer, Info, 7);`

- [ ] Column-Store
- [x] Row-Store

## Aufgabe 9 XQuery (12 Punkte)

Als Grundlage für die Aufgabe wird eine leichte Abwandlung des Uni Schemas genutzt, von dem hier ein Ausschnitt geben ist. `Fakultaeten` ist der Wurzelknoten des Dokuments `unifak`:

```
<Fakultaeten>
 <Fakultaet>
   <FakName>Theologie</FakName>
   <ProfessorIn PersNr="P2134">
      <Name>Augustinus</Name>
      <Rang>C3</Rang>
      <Raum>309</Raum>
      <Vorlesungen>
         <Vorlesung VorlNr="V5022">
            <Titel>Glaube und Wissen</Titel>
            <SWS>2</SWS>
         </Vorlesung>
         ...
      </Vorlesungen>
   </ProfessorIn>
 </Fakultaet>
 <Fakultaet>
   <FakName>Philosophie</FakName>
   <ProfessorIn ID="P2126" PersNr="P2126">
        <Name>Russel</Name>
        <Rang>C4</Rang>
        <Raum>232</Raum>
        <Vorlesungen>
          <Vorlesung ID="V5043" VorlNr="V5043" Voraussetzungen="V5001">
            <Titel>Erkenntnistheorie</Titel>
            <SWS>3</SWS>
          </Vorlesung>
          ...
```

a)* Geben Sie in XPath die Namen der Professoren aus, die eine Vorlesung mit dem Titel `Maeeutik` halten.

```
doc('unifak')//ProfessorIn[.//Vorlesung/Titel='Maeeutik']/Name
```

0 1 2 3

b)* Geben Sie in XPath alle Professoren aus, die mindestens zwei Vorlesungen halten.

```
doc('unifak')//ProfessorIn[.//Vorlesung[2]]
```

0 1 2 3

c)* Erstellen Sie in XQuery ein Vorlesungsverzeichnis geordnet nach Vorlesungstitel (aufsteigend) wie nachfolgend gezeigt.

```
<Vorlesungsverzeichnis>
   <Vorlesung Titel="Erkenntnistheorie"/>
   <Vorlesung Titel="Glaube und Wissen"/>
    ...
</Vorlesungsverzeichnis>
```

```
<Vorlesungsverzeichnis> {
   for $v in doc('unifak')//Vorlesung
   order by $v/Titel (: 1 Punkt :)
   return <Vorlesung Titel="{$v/Titel}" />
} </Vorlesungsverzeichnis>
```

0 1 2 3 4 5 6

## Aufgabe 10 TF-IDF (9 Punkte)

0 1 2 3 4 5 6

a)* Berechnen Sie für die folgenden drei Dokumente die TF-IDF Werte. Dabei sind alle Worte relevant.

1. ERDB macht echt viel Spaß
2. Die Klausur ist sicher machbar
3. Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

ERDB IDF = log(3/2) = 1/6

| | D1 | D2 | D3 |
|---|---|---|---|
| TF | 1/5 | 0 | 1/10 |
| TF-IDF | 1/30 | 0 | 1/60 |

Klausur IDF = log(3/2) = 1/6

| | D1 | D2 | D3 |
|---|---|---|---|
| TF | 0 | 1/5 | 1/10 |
| TF-IDF | 0 | 1/30 | 1/60 |

0 1 2 3

b) Welches Ranking ergibt sich für die Anfrage: “ERDB Klausur“? Berechne Sie die Werte auf 3 Nachkommastellen genau.
Hilfswerte (gerundet):

$$log(3) = 1/2$$
$$log(2,5) = 2/5$$
$$log(2) = 1/3$$
$$log(1,5) = 1/6$$
$$log(1) = 0$$

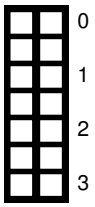
Ranking:
D1: 1/30 = 0,033
D2: 1/30 = 0,033
D3: 1/60 + 1/60 = 1/30 = 1/10 * 1/3 = 0.033

## Aufgabe 11 SPARQL (9 Punkte)

Im Folgenden ist schematisch eine RDF-Buchdatenbank dargestellt. Jedes Tripel (Subjekt, Prädikat, Objekt) enthält eine Information, z.B. Erscheinungsjahr oder Autorenname, zu einem Buch. Jedes Märchen ist Teil eines Buches.

```
@prefix ex:<http://maerchen.example.org/>.
ex:KHG1 ex:hatAutor ex:Grimm.
ex:KHG1 ex:erschienen 1812.
ex:Rapunzel ex:teilVon ex:KHG1.
ex:DerGestiefelteKater ex:teilVon ex:KHG1.
ex:KHG2 ex:hatAutor ex:Grimm.
ex:KHG2 ex:erschienen 1837.
ex:SchneeweißchenUndRosenrot ex:teilVon ex:KHG2.
ex:Kindermaerchen ex:hatAutor ex:Anderson.
ex:Kindermaerchen ex:erschienen 1837.
ex:KleineMeerjungfrau ex:teilVon ex:Kindermaerchen.
```

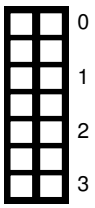
a)* Werten Sie aus:




```
PREFIX ex:<http://maerchen.example.org/>
SELECT ?book
WHERE {?book ex:hatAutor ex:Grimm. ?book ex:erschienen ?jahr FILTER (?jahr > 1836)}
```

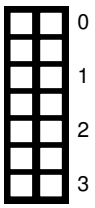
```
ex:KHG2
```

b)* Geben Sie jedes Märchen mit Erscheinungsjahr aus.




```
PREFIX ex:<http://maerchen.example.org/>
SELECT ?maerchen ?jahr
WHERE { ?maerchen ex:teilVon ?book.
        ?book ex:erschienen ?jahr .}
```

c)* Geben Sie jeweils den Namen der Märchen aus Büchern von `ex:Grimm` mit einem Erscheinungsjahr nach 1836 aus.




```
PREFIX ex:<http://maerchen.example.org/>
SELECT ?maerchen
WHERE {?maerchen ex:teilVon ?book.
       ?book ex:hatAutor ex:Grimm.
       ?book ex:erschienen ?jahr FILTER (?jahr > 1836).}
```

**Zusätzlicher Platz für Lösungen. Markieren Sie deutlich die Zuordnung zur jeweiligen Teilaufgabe. Vergessen Sie nicht, ungültige Lösungen zu streichen.**