Hinweise zur Bearbeitung der Klausur zum Kurs 1666 Datenbanken in Rechnernetzen

Bitte lesen Sie sich diese Hinweise vollständig und aufmerksam durch, bevor Sie mit der Bearbeitung der Klausur beginnen.

1. Die Klausurdauer beträgt 3 Stunden.
2. Prüfen Sie bitte die Vollständigkeit Ihrer Unterlagen. Die Klausur umfasst:
   - 2 Deckblätter
   - diese Hinweise
   - 1 Formblatt für eine Teilnahmebescheinigung zur Vorlage beim Finanzamt
   - 6 Aufgaben auf den Seiten 1–9
3. Bevor Sie mit der Bearbeitung der Klausuraufgaben beginnen, füllen Sie bitte die folgenden Teile der Klausur aus:
   (a) sämtliche Deckblätter mit Name, Anschrift sowie Matrikelnummer. Markieren Sie vor der Abgabe auf allen Deckblättern die von Ihnen bearbeiteten Aufgaben.
   (b) die Teilnahmebescheinigung, falls Sie diese wünschen.
4. Schreiben Sie Ihre Lösungen auf Ihr eigenes Papier (DIN A4) und nicht auf die Seiten mit den Aufgabenstellungen. Heften Sie vor Abgabe der Klausur die Deckblätter (und evtl. die Teilnahmebescheinigung) an Ihre Bearbeitung.
5. Schreiben Sie bitte auf jedes Blatt oben links Ihren Namen und oben rechts Ihre Matrikelnummer. Nummerieren Sie Ihre Seiten bitte durch.
6. Als Hilfsmittel sind nur unbeschriebenes Konzeptpapier und Schreibzeug zugelassen. Die Reinschrift der Klausur darf **nicht mit Bleistift** erfolgen.
7. Durch Lösen der Aufgaben sind maximal 100 Punkte erreichbar. Sie dürfen damit rechnen einen Übungsschein bzw. ein Zertifikat zu erhalten, wenn Sie insgesamt mindestens 50% Prozent der Gesamtpunkte erreichen.

Aufgabe 1 (Multiple Choice) (10 Punkte)

Welche der folgenden Aussagen sind korrekt? Kreuzen Sie die Wahr (W) oder die Falsch (F) Spalte an.

W	F	Das Zwei-Phasen-Commit-Protokoll ...
☒	☐	... garantiert globale Serialisierbarkeit von verteilten Transaktionen.
☐	☒	... garantiert eine serielle Ausführung von verteilten Transaktionen.
		Garantiert eine serialisierbare Ausführung ...
☒	☐	... verhindert ein fortgepflanztes Rollback.
☐	☒	... garantiert den gleichzeitigen Start der Subtransaktionen.
		Garantiert das quasi gleichzeitige Commit der Subtransaktionen.

W	F	Was macht eine Teiltransaktion, wenn sie den ready-to-commit Zustand erreicht?
☒	☐	Sie hält alle Sperren.
☐	☒	Sie gibt ihre Sperren frei.
☐	☒	Sie wartet auf das prepare-to-commit von der globalen Transaktion.
		Sie wartet auf das commit von der globalen Transaktion.
☐	☒	Sie sendet ein prepare-to-commit an die globale Transaktion.
		Prepare-to-commit hat bereits vorher stattgefunden. Außerdem wird das prepare-to-commit von der globalen Transaktion aus geschickt.

W	F	Welche der folgenden Aussagen zu den Replikationsverfahren sind wahr, welche sind falsch?
☐	☒	Durch die redundante Speicherung von Daten wird auf jeden Fall die Verfügbarkeit erhöht, unabhängig vom Replikationsverfahren.
		ROWA behindert die Verfügbarkeit, da Updates bei Ausfall eines Knotens nicht mehr durchgeführt werden können.
☒	☐	Wenn die Primary Copy ausfällt, kann ein anderer Knoten zur Primary Copy ernannt werden.
☐	☒	Diese neue Primary Copy synchronisiert sich zunächst mit der bisherigen Primary Copy, damit sie auch tatsächlich auf dem aktuellen Stand ist.
		Da die alte Primary Copy ausgefallen ist, müsste sie dann warten, bis die alte Primary Copy wieder da ist. Dann bräuchte sie erst gar nicht anzufangen, da dann auch die alte Primary Copy weitermachen könnte.
☒	☐	Das ROWA-Verfahren garantiert den Lese-Transaktionen, dass sie immer einen konsistenten Datenbankzustand sehen
☐	☒	Beim Primary Copy Verfahren kann durch die Einbeziehung von „Schnappschuss-Versionen" erreicht werden, dass die Lese-Transaktionen auf den verschiedenen Knoten den aktuellen Datenbankzustand sehen.
		Das zwar nicht, aber zumindest, dass bei veralteten Daten der sichtbare Datenbank-Zustand konsistent ist.

W	F	Fragen zur globalen Serialisierbarkeit:
☒	☐	Wenn in einer globalen Schedule eine der lokalen Schedules nicht serialisierbar ist, dann ist die globale Schedule auch nicht serialisierbar.
☐	☒	Wenn alle lokalen Schedules serialisierbar sind, dann ist auch globale Serialisierbarkeit gegeben. *Hinzukommen muss, dass es eine serielle Reihenfolge der Transaktionen gibt, die bei allen lokalen Schedules gilt.*

W	F	Fragen zur Partitionierung und Allokation:
☒	☐	Eine Relation kann horizontal partitioniert werden durch eine Selektion.
☐	☒	Um sicherzustellen, dass die horizontalen Partitionen durch einen Join wieder zur Ausgangsrelation zusammengeführt werden können, muss in jeder Partition der Primärschlüssel vorkommen. *Horizontale Partitionen werden durch eine Vereinigung zusammengeführt, nicht durch einen Join.*
☒	☐	Bei einer abgeleiteten horizontalen Partitionierung braucht für die Zusammenführung der Partitionen zur Ausgangsrelation kein Join mit der externen Relation durchgeführt werden, die in der Partitionierungsbedingung vorkommt.
☐	☒	Um beim Ausfall eines Knotens sicherzustellen, dass die Daten trotzdem zugreifbar bleiben, werden üblicherweise mehrere Partitionen gemeinsam allokiert. *Es ist umgekehrt: einzelne Partitionen werden mehrfach an unterschiedliche Knoten allokiert.*
☒	☐	Um eine vertikal partitionierte Relation wieder zusammenzufügen, wird in der Regel ein Join über den Primärschlüssel der Ursprungsrelation durchgeführt. Für die Durchführung dieses Joins ist der Semi-Join schlechter geeignet als eine Übertragung aller Teilrelationen an den Zielknoten. *Das ist richtig. Da die Joinbedingung lautet: Primärschlüssel = Primärschlüssel, müssen sowieso alle Tupel übertragen werden. Damit sind in diesem Falle die zusätzlichen Schritte des Semi-Join-Verfahrens nur Overhead, sie bringen aber keinen Nutzen.*

Hinweise zur Bewertung:

Da durch ein zufälliges Ankreuzen leicht ein paar Punkte erworben werden können, werden folgende Regeln bei der Bewertung angewendet:

1. Jede richtige Lösung gibt einen halben Punkt.
2. Für eine falsche Lösung wird ein halber Punkt abgezogen.
3. Falls nichts angekreuzt ist (weder W noch F), so gibt es 0 Punkte (d.h., es wird auch nichts abgezogen)
4. Pro Teilaufgabe (eigenständige Box zum Ankreuzen) ist die minimale Punktzahl 0 (es gibt also keine negativen Punkte und falsche Antworten werden nicht mit richtigen Antworten aus anderen Teilaufgaben verrechnet).
5. Die Gesamtpunktzahl wird auf die nächste ganze Zahl aufgerundet.

Aufgabe 2: Serialisierbarkeit von Schedules

(12 Punkte)

Gegeben seien die folgenden lokalen Schedules S_A und S_B der globalen Transaktionen T_1 und T_2 an den Knoten A und B. Prüfen Sie jeweils, ob globale Serialisierbarkeit gegeben ist.

Bewertung: 4 Punkte je Teilaufgabe

a) S_A: $< r_1[a]\ r_2[a]\ w_2[a]\ r_3[b]\ w_3[b]\ w_1[a] >$
   S_B: $< r_1[c]\ w_1[c]\ r_2[d]\ w_2[d]\ r_3[e]\ w_3[e] >$

b) S_A: $< r_2[a]\ r_1[a]\ w_2[b]\ w_1[a]\ r_3[a]\ r_3[b]\ w_3[b] >$
   S_B: $< r_2[c]\ w_2[c]\ r_3[d]\ w_3[d]\ r_1[e]\ w_1[e] >$

c) S_A: $< r_1[a]\ r_3[b]\ w_1[a]\ w_3[b]\ r_2[a]\ w_2[a] >$
   S_B: $< r_3[c]\ w_3[c]\ r_2[d]\ w_2[d]\ r_1[d]\ w_1[d] >$

Aufgabe 3: Verteilter Join (24 Punkte)

Gegeben seien die beiden Tabellen

Projekt (PNR, P-Name, P-Leiter, ...) K_P
Mitarbeiter (MNR, M-Name, Adresse, ...) K_M

Folgende Zahlen beschreiben die Größenordnungen der abgespeicherten Tabellen:

- Ca. 4.000 Projekte, ca. 50.000 Mitarbeiter
- Ca. 10 KB werden pro Projekt abgespeichert. Ca. 5 KB werden pro Mitarbeiter abgespeichert.
- Es gibt ca. 1.000 P-Leiter (einzelne Projektleiter können mehrere Projekte leiten).
- Das Feld P_Leiter nimmt 6 Byte ein.

Die Tabellen werden an unterschiedlichen Knoten gespeichert, Projekt an Knoten K_P und Mitarbeiter an Knoten K_M.

Das Ziel ist es im Folgenden, einen Join zwischen den beiden Tabellen über die Join-Bedingung (P-Leiter = MNR) durchzuführen und das Ergebnis an Knoten K_P abzuliefern.

(a) Beschreiben Sie für die folgenden Fälle, welche Datenübertragungen stattfinden müssen und welche Daten dabei zu übertragen sind.
(b) Berechnen Sie für die folgenden Fälle, welche Datenmengen jeweils übertragen werden müssen.

Fall 1:
Kompletter lokaler Join findet am Knoten K_M statt.

Fall 2:
Kompletter lokaler Join findet am Knoten K_P statt.

Fall 3:
Semi-Join findet am Knoten K_M statt.

Fall 4:

Join mit Hashfilter findet am Knoten K_M statt.

Einige weitere Angaben für den Fall 4:

- Der Bitvektor habe eine Länge von 1013, das Hashverfahren füllt ca. die Hälfte der Bits mit einer 1.
- Wir gehen davon aus, dass die Mitarbeiter-Nummern gleichverteilt auf den Bitvektor abgebildet werden.
- In der Rechnung können Sie mit 1000 rechnen statt mit der Primzahl 1013. Ebenso können Sie bei KB mit 1000 statt mit 1024 rechnen.

Punktevergabe:

Aufgabe a): Je Fall 3 Punkte
Aufgabe b): Je Fall 3 Punkte

Aufgabe 4: Partitionierung und Allokation (22 Punkte)

Ein Unternehmen ist auf drei Standorte in Deutschland verteilt. An jedem Standort gibt es eine Dispositionsabteilung, die das rechtzeitige Eintreffen von Lieferungen von Zulieferern aus dem Umland überwacht. Die drei Abteilungen überwachen jeweils einen Postleitzahlbereich mit folgender Aufteilung:

Abteilung an Standort 1: PLZ 01000 bis 39999 (Der Norden, Osten und die Mitte)
Abteilung an Standort 2: PLZ 40000 bis 69999 (Der Westen)
Abteilung an Standort 3: PLZ 70000 bis 99999 (Der Süden)

Die Verwaltung der Lager des Unternehmens wird über die drei Relationen Lieferant, Teil, und Bestellung realisiert. Der Primärschlüssel ist jeweils unterstrichen:

Lieferant(	Teil(	Bestellung(
LiefNr,	TeilNr,	LiefNr,
LiefName,	TeilName,	TeilNr,
LiefPLZ,	LagerNr,	Anz,
LiefStrasse)	RegalNr,	Termin,
	Bestand)	Status)

Anmerkung: Bei den folgenden beiden Aufgaben in a) und b) geht es nicht darum, die geeigneten Partitionen und Allokationen durch Berechnungen zu ermitteln, sondern durch Überlegungen unter Einbeziehung der im Folgenden genannten Bedingungen. Daher ist es wichtig, Ihre Überlegungen zu dokumentieren und die Ergebnisse zu begründen.

a) (16 Punkte)

Die drei Relationen sollen sinnvoll partitioniert werden. Es wird dabei von folgenden Bedingungen ausgegangen:

- Jede Abteilung überwacht nur die Bestellungen der Lieferanten ihres PLZ-Bereiches
- Es gibt drei Materiallager: Lager1, Lager2 und Lager3, die eigene Verbundrechner haben. Die Abteilungen sind von diesen Lagern weit entfernt und greifen über die TeilNr der Teil-Relation meist nur auf den aktuellen Bestand zu.
- Die Materiallager greifen hauptsächlich auf die Attribute TeilNr, TeilName, LagerNr und RegalNr der Teil-Relation zu. Lager1 verwaltet alle Teile mit LagerNr = 1 und so weiter.

Definieren Sie alle Partitionen über die relationalen Operatoren, durch die sie aus den Basis-Relationen gebildet werden. Begründen Sie, warum Sie diese Partitionen benötigen.

b) (6 Punkte)

Wie könnte eine geeignete zugehörige Allokation aussehen, die folgenden Bedingungen entspricht:

- Globale Zugriffe finden eher selten statt.
- Die Materiallager greifen auch häufig lesend auf Bestand in Teil zu.

Führen Sie diese Allokation einmal nicht redundant und einmal mit Redundanzen durch. Begründen Sie, warum Sie Partitionen an mehreren Knoten speichern.

Aufgabe 5: Tree-Quorum (18 Punkte)

Die Knoten, auf denen ein Objekt repliziert wird, seien in einem Baum angeordnet. Dieser Baum hat die Höhe 4 und einen Verzweigungsgrad von 3.

a) (6 Punkte)

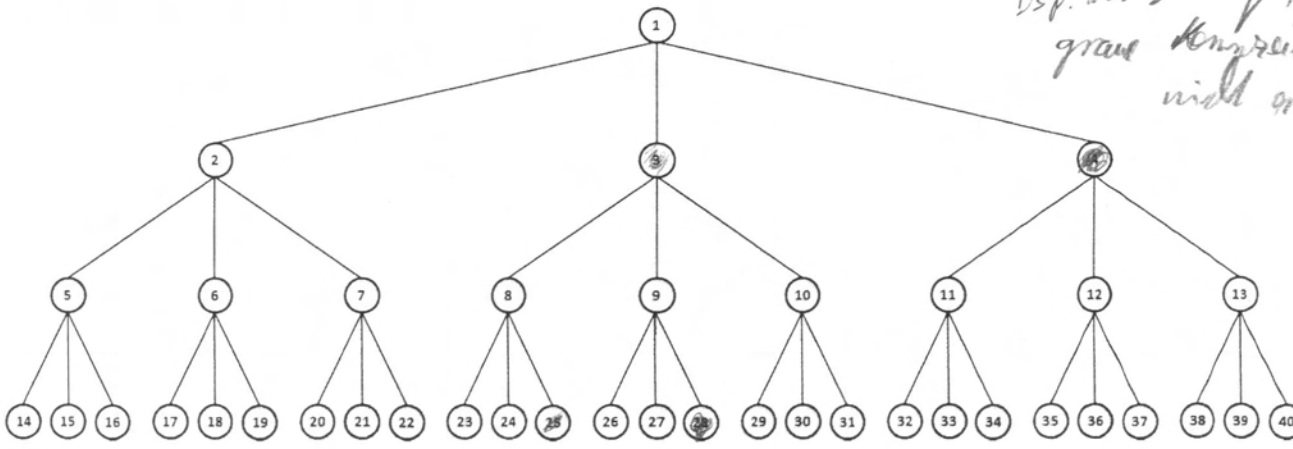
Bestimmen Sie das Schreibquorum, also die Bedingungen, die erfüllt sein müssen, damit eine Transaktion Änderungen einbringen kann.

b) (je 6 Punkte)

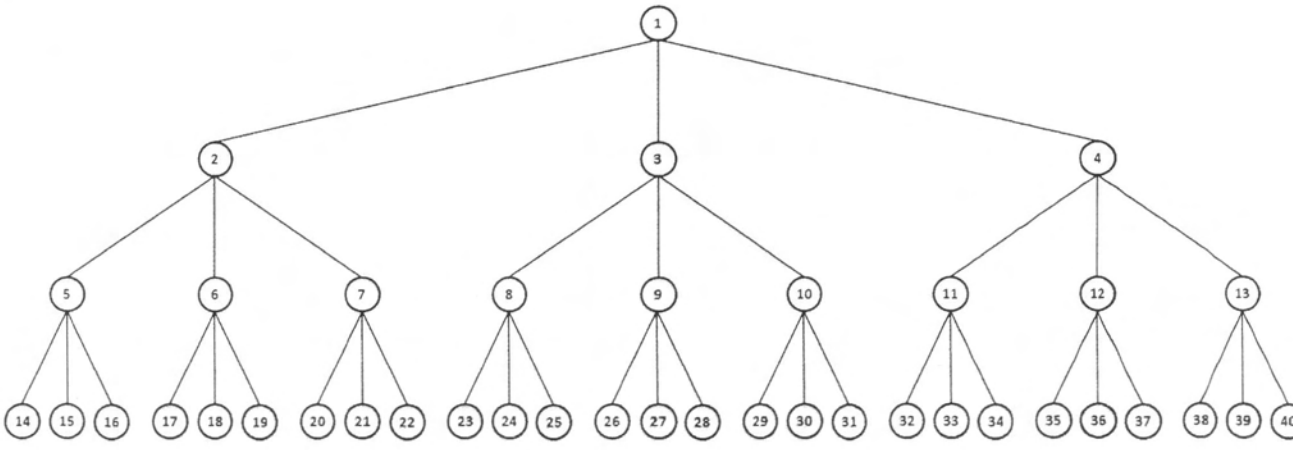
Betrachten Sie die beiden folgenden Bäume, in denen die nicht erreichbaren Knoten grau gekennzeichnet sind. Bestimmen Sie für jeden Baum, ob die Transaktion eine Änderung durchführen kann, d.h. ob sie das erforderliche Quorum bekommen kann.

Begründen Sie, warum ein Quorum nicht erreicht wird bzw. beschreiben Sie, durch welche Knoten das Quorum erfüllt wird.

b1)



b2)



Aufgabe 6: Globale Schema-Architektur (14 Punkte)

Bei der Integration von unterschiedlichen Datenbanken sind häufig zwei scheinbar entgegengesetzte Anforderungen zu erfüllen:

(1) Die lokalen Anwendungen sollen weiterhin benutzt werden, ohne dass der Programm-Code dieser Anwendungen geändert werden muss.

(2) Es sollen globale Anwendungen entstehen, die Daten aus den unterschiedlichen integrierten Datenbanken in einheitlicher Weise bearbeiten.

a) (6 Punkte)
Skizzieren Sie für den Fall von 2 zu integrierenden Datenbanken die Schema-Architektur, mit der diese Anforderungen erfüllt werden können.

b) (4 Punkte)
Beschreiben die Funktionen der Repräsentations-Schemata in diesem Kontext.

c) (4 Punkte)
Beschreiben Sie die Schichten des Globalen Schemas und deren Inhalte.