Gruppe **A**

Bitte tragen Sie **SOFORT** und **LESERLICH** Namen und Matrikelnr. ein, und legen Sie Ihren Ausweis bereit.

PRÜFUNG AUS			26.08.2020
◯ DATENMODELLIERUNG 2 (**184.790**)	◯ DATENBANKSYSTEME (**184.686**)		**GRUPPE A**
Matrikelnr.	Familienname	Vorname	

Arbeitszeit: 90 Minuten. Lösen Sie die Aufgaben auf den vorgesehenen Blättern; Lösungen auf Zusatzblättern werden nicht gewertet. **Viel Erfolg!**

Achtung!

Für sämtliche Fragen mit Ankreuzmöglichkeiten gilt: Ankreuzen alleine gibt keine Punkte!

Punkte gibt es nur in Zusammenhang mit geforderter Erklärung/Beispiel/...!

Notation:

In den Aufgaben 1 – 3 wird die folgende (aus der Vorlesung bekannte) Notation für Transaktionen T_i verwendet:

- $r_i(O)$ und $w_i(O)$: Lese- bzw. Schreibzugriff von T_i auf Objekt O.
- b_i, c_i, a_i: Beginn (`BEGIN OF TRANSACTION`), Commit (`COMMIT`) bzw. Abbruch (`ABORT`/`ROLLBACK`) von T_i.

Die Indizes $_i$ können weggelassen werden, wenn klar ist zu welcher Transaktion eine Operation gehört.

Des weiteren wird das aus der Vorlesung bekannte Format für Logeinträge verwendet:

`[LSN, TA, PageID, Redo, Undo, PrevLSN]` für "normale" Einträge, bzw.
`[LSN, TA, BOT, PrevLSN]` für `BOT` Log-Einträge und
`[LSN, TA, COMMIT, PrevLSN]` für `COMMIT` Einträge.

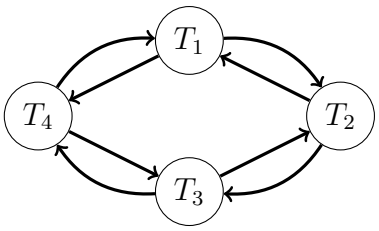
Kompensations Logeinträge (Compensation Log Records) haben das Format
⟨`LSN, TA, PageID, Redo, PrevLSN, UndoNextLSN`⟩ bzw.
⟨`LSN, TA, BOT, PrevLSN`⟩

Dabei stellt `LSN` die Log-Sequence Nummer dar, `TA` die Transaktion, `PageID` ist die veränderte Seite, `Redo` und `Undo` die für das Redo bzw. Undo benötigten Informationen, `UndoNextLSN` ist die LSN des nächsten Logeintrags der selben Transaktion welcher zurückgesetzt werden soll, und `PrevLSN` die LSN des vorherigen Logeintrags derselben Transaktion.

Im Falle logischer Protokollierung sollen die Änderungen zum aktuellen Datenbestand nur mittels *Addition* bzw. *Subtraktion* angegeben werden, z.B. [·, ·, ·, X+=d_1, X-=d_2, ·].

Aufgabe 1: Eigenschaften von Transaktionen (14)

a) Betrachten Sie den rechts dargestellten Serialisierbarkeitsgraph. Geben Sie eine Historie der vier Transaktionen T_1, T_2, T_3 und T_4 an, welche *exakt* diesen Serialisierbarkeitsgraph erzeugt.
Die Anzahl und Namen der verwendeten Datensätze sind frei wählbar.
Geben Sie für jede Transaktion ihr Beginn (b_i) und Ende (c_i) an.



b) Ist jede richtige Lösung zu Aufgabe a) **konfliktserialisierbar**?
Falls ja, geben Sie eine konfliktäquivalente, serielle Ausführungsreihenfolge der Transaktionen an.
Falls nein, geben Sie eine möglichst kleine Menge an Transaktionen an, welche man entfernen müsste damit jede richtige Lösung konfliktserialisierbar wäre. Geben Sie anschließend für die verbleibenden Transaktionen eine konfliktäquivalente, serielle Ausführungsreihenfolge der Transaktionen an.

alle korrekten Historien sind konfliktserialisierbar: ◯ ja ◯ nein

(falls "nein") Transaktionen entfernen: .....

Konfliktäquivalente serielle Ausführung der (verbleibenden) Transaktionen: .....

c) Betrachten Sie die unten gegebene Historie der vier Transaktionen T_1, T_2, T_3 und T_4. Bestimmen Sie, ob die Historie **strikt** ist, und geben Sie eine kurze Begründung (1 Satz) an.

T_1	b					$r(B)$	$w(A)$	$w(B)$			$w(D)$	c												
T_2	b	$r(A)$												$w(A)$		a								
T_3	b			$r(B)$						$r(D)$									$w(B)$			$w(A)$		
T_4	b		$w(C)$										$r(B)$		$r(A)$			$w(C)$						

Historie ist strikt: ◯ ja ◯ nein

Begründung: .....

.....

.....

Aufgabe 2: Protokollierung und Wiederanlauf (Logging und Recovery) (11)

Betrachten Sie die folgende Historie, welche durch eine Abfolge von Elementaroperationen der drei Transaktionen T_1, T_2 und T_3 auf den Datensätzen A, B, C und D gegeben ist. Nehmen Sie an, dass jeder Datensatz X jeweils auf der Seite P_X gespeichert ist, und dass zu Beginn die Werte $A = B = C = D = 20$ sind.

Schritt	T_1	T_2	T_3
1	`BOT`		
2	$r(A, a_1)$		
3			`BOT`
4			$w(C, 4)$
5		`BOT`	
6		$r(B, b_2)$	
7		$r(D, d_2)$	
8	$w(A, a_1 - 1)$		
9			$r(B, b_3)$
10	`ROLLBACK`		
11			$r(A, a_3)$
12		$w(B, b_2 + d_2)$	
13			$w(C, a_3 - 1)$
14		$w(A, b_2 + d_2)$	

a) Zwischen Schritt 12 und 13 wird ein aktionskonsistenter Sicherungspunkt ("Checkpoint") erstellt.
i) Geben Sie die Inhalte der Seiten P_B und P_C (LSN, Werte von B und C) im Hintergrundspeicher **direkt nach Erstellung des Sicherungspunktes** an.
Annahme: Jeder Logeintrag erhält als LSN die Nummer des Schrittes der zugehörigen Aktion.

P_B	LSN:	
$B =$		

P_C	LSN:	
$C =$		

ii) Was ist der späteste Schritt, bei dem ein Redo nun beginnen könnte, und bis zu welchem Schritt müsste ein Undo zurückgehen?

Redo: ..... Undo: .....

b) Geben Sie alle Logeinträge an, welche (unter Verwendung des in der Vorlesung vorgestellten Logging-Protokolls) bei Abarbeitung dieser Historie erzeugt werden. Alle nötigen Aktionen für das `ROLLBACK` in Schritt 10 werden vor der Aktion in Schritt 11 ausgeführt. Verwenden Sie die logische Protokollierung.

(Diese Aufgabe ist unabhängig von a), d.h. es gibt keinen Sicherungspunkt und gültige LSN sind frei wählbar.)

`[#1,` T_1`, BOT, #0]`, `[#2,` T_3`, BOT, #0]`	.....
.....	.....
.....	.....
.....	.....
.....	.....

c) Nehmen Sie an, im Anschluss an Schritt 14 wird nach einem Verlust des Datenbankpuffers ein Wiederanlauf (nach dem ARIES Verfahren) durchgeführt. Nehmen Sie weiter an, dass es direkt nach dem *erfolgreichen* Abschluss des Wiederanlaufs erneut zu einem Verlust des Datenbankpuffers und einem anschließenden Wiederanlauf kommt.

Wie viele Logeinträge werden im Rahmen dieses erneuten Wiederanlaufs erstellt?

Anzahl der neuen Logeinträge: .....

(Sie können davon ausgehen, dass beim Absturz keine Logeinträge verloren gehen.)

Aufgabe 3: Sperrprotokolle (Locking) (10)

a) Zwei Transaktionen T_1 und T_2 sind durch die folgenden Elementaroperationen gegeben:

- T_1: $b_1, r_1(D), w_1(A), r_1(B), r_1(A), w_1(C), c_1$
- T_2: $b_2, w_2(B), r_2(C), w_2(C), r_2(A), w_2(D), c_2$

Beide Transaktionen arbeiten nach folgendem Sperrprotokoll: Für Exclusive-Locks wird ein "normales" Zwei-Phasen Sperrprotokoll verwendet (2PL). Dabei werden Sperren so spät wie möglich angefragt und so früh wie möglich wieder freigegeben.

Share-Locks werden jeweils direkt vor einer Leseoperation angefragt, und sofort nach der Operation wieder freigegeben (kein 2PL). Die Freigabe von Share-Locks hat dabei keinen Einfluss auf das 2PL für Exclusive-Locks.

Geben Sie eine teilweise Historie (=Abfolge der Elementaroperationen) der beiden Transaktionen T_1 und T_2 an, welche zu *einem Deadlock* führt. Geben Sie die Historie nur bis zum Auftreten des Deadlocks an, und achten Sie darauf, dass eine blockierte Transaktion keine weiteren Operationen ausführt (geben Sie bitte jedoch die Operation noch an, auf Grund der die Transaktion blockiert).

(Würde T_1 zum Beispiel bei $r_1(D)$ blockieren, sollte $r_1(D)$ noch angegeben werden, $w_1(A)$ aber nicht mehr.)

b) *Die unvermeidbare, Covid-19 inspirierte, Aufgabe:*

In einer Werkhalle werden Werkstücke zusammengebaut. Normalerweise teilen sich die Mitarbeiter_innen dabei das Werkzeug. Als Maßnahme gegen das Coronavirus muss Werkzeug nun gründlich desinfiziert werden, bevor es an eine andere Person weitergegeben werden darf. Um die Anzahl der Desinfektionsvorgänge einzuschränken wurden folgende Regeln eingeführt:

- Jedes Werkstück wird von genau einer Person zusammengebaut.
- Wurde die Arbeit an einem Werkstück einmal begonnen, wird diese **nicht mehr abgebrochen oder pausiert**.
- Werkzeuge dürfen nur weitergegeben werden, wenn sie für das Werkstück nicht mehr benötigt werden.

Beim ersten Versuch diese Strategie umzusetzen traten immer wieder Deadlocks auf. Welche(s) der in der Vorlesung vorgestellten Sperrprotokolle kann verwendet werden, um dieses Problem zu lösen? Beschreiben Sie auch möglicherweise nötige Vorbedingungen, sowie eine alternative Lösung falls diese nicht gegeben sind.

(Es darf davon ausgegenagen werden, dass alle benötigten Werkzeuge prinzipiell vorhanden sind – nur eben möglicherweise in Verwendung.)

Für die Aufgaben 4 – 6 gilt die Datenbankbeschreibung auf diesem Blatt.

Aufgabe 4: Erstellen eines Datenbankschemas mittels SQL und Schlüsselabhängigkeiten (11)

a) Folgendes Schema sei gegeben:

`artist`	(<u>name</u>, <u>stern</u>, hauptBewegung: *bewegung.titel*)
`bewegung`	(<u>titel</u>, repName: *artist.name*, repStern: *artist.stern*)
`folgt`	(<u>name</u>: *artist.name*, <u>stern</u>: *artist.stern*, <u>bewegung</u>: *bewegung.titel*)
`beeinflusst`	(<u>von</u>: *bewegung.titel*, <u>zu</u>: *bewegung.titel*)

Jeder Künstler und jede Künstlerin (`artist`) hat einen Namen und ein Sternzeichen, die Kombination davon ist immer eindeutig. Darüber hinaus wird auch vermerkt welches die wichtigste (Kunst-)Bewegung ist (`hauptBewegung`). Sternzeichen werden als Enum umgesetzt, wobei in der Datenbank nur die Werte "Widder", "Jungfrau" und "Loewe" vorkommen dürfen. Jede Kunstbewegung hat einen eindeutigen Namen. Zusätzlich hat jede Bewegung auch einen Repräsentanten (`repName`, `repStern`). Jeder Künstler und jede Künstlerin kann einer oder mehreren Bewegungen folgen, dies wird in einer eigenen Tabelle (`folgt`) festgehalten. Die Kombination aus Künstler bzw. Künstlerin und Bewegung ist eindeutig. Zuletzt wird auch für Bewegungen gespeichert (`beeinflusst`), welche Einflüsse es zwischen verschiedenen Bewegungen gibt: eine Bewegung (`von`) kann eine andere Bewegung (`zu`) beeinflussen. Die Kombination ist hier jeweils eindeutig.

Geben Sie die nötigen SQL Statements an, um obiges Schema (inklusive aller Konsistenzbedingungen) anzulegen. Wählen Sie passende Typen für Attribute. **Die Abkürzung VC statt VARCHAR(100) ist erlaubt.**

Hinweis: Achten Sie bei den Statements auf die Reihenfolge.

b) Berücksichtigen Sie folgende Datenbank-Instanz:

Bewegung

titel	repName	repStern
Post-Impressionismus	Paul Cézanne	Widder
Impressionismus	Claude Monet	Widder
Fauvismus	Henri Matisse	Jungfrau
Kubismus	Pablo Picasso	Widder
Futurismus	Gino Severini	Widder
Expressionismus	Edvard Munch	Loewe

Artist

name	stern	hauptBewegung
Claude Monet	Widder	Expressionismus
Edvard Munch	Loewe	Impressionismus
Gino Severini	Widder	Futurismus
Henri Matisse	Jungfrau	Kubismus
Pablo Picasso	Widder	Expressionismus
Paul Cézanne	Loewe	Fauvismus

Kreuzen Sie an ob diese Datenbank-Instanz das Schema aus a) erfüllt. Falls nicht, geben Sie bitte die Tupel an die das Schema verletzen sowie eine **einfache** Erklärung (maximal 1 Satz!) was genau das Problem ist.

Datenbank-Instanz erfüllt das Schema aus a) ◯ ja ◯ nein

Aufgabe 5: Rekursive Abfragen (12)

Erstellen Sie eine Rekursive SQL Abfrage

Realisieren Sie, basierend auf dem Schema von Aufgabe 4 a), folgende Abfrage:

Die Tabelle `beeinflusst` definiert welche (Kunst-)Bewegung von anderen Bewegungen geprägt wurde. Gesucht sind alle Bewegungen welche – direkt oder indirekt – von der Hauptbewegung des Künstlers ('Klimt', 'Loewe') beeinflusst wurden (inkl. der Hauptbewegung von ('Klimt', 'Loewe') selbst). Mit Ausnahme der Hauptbewegung von ('Klimt', 'Loewe') sollen dabei generell nur jene Bewegungen betrachtet (und ausgegeben) werden, welche die Hauptbewegung mindestens zweier Künstler_innen sind.

Die Ausgabe soll aus dem Titel der Kunstbewegung bestehen, wobei ein mehrfaches Auftreten desselben Tupels ausgeschlossen werden soll.

Geben Sie die nötigen SQL Statements an, um eine Rekursive Abfrage zu schreiben welche die beschriebene Abfrage implementiert.

Hinweis: Achten Sie darauf, dass ihre Abfrage terminiert.

Aufgabe 6: PL/SQL Trigger (12)

Nehmen Sie an, dass die **Funktionen und Trigger** wie auf **Seite T** (am vorletzten Blatt dieser Prüfung) definiert wurden. Die Aufgaben beziehen sich auf die Beispielinstanz auf **Seite B**.

In jedem der folgenden tasks ist ein SQL Statement gegeben das **über die Beispielinstanz** ausgeführt wird. (Das `UPDATE` in Aufgabe a hat keinen Einfluss auf Aufgabe b usw.) Geben Sie die Ausgabe der `SELECT`-Statements an. **Falls ein Fehler auftreten würde, geben Sie an welcher Fehler auftritt.**

Sie dürfen die Werte in ihren Antworten beliebig (soweit noch eindeutig zuordenbar) abkürzen!

a)

```
UPDATE bewegung SET repname='Mucha' WHERE titel='Jugendstil';

SELECT * from bewegung WHERE titel='Jugendstil';
SELECT name, bewegung from folgt;
```

---

b)

```
BEGIN;
UPDATE bewegung SET titel='Romantizismus' WHERE repname='Goya';
DELETE FROM folgt WHERE bewegung = 'Glas';
COMMIT;

SELECT * from bewegung WHERE repname='Goya';
SELECT * from folgt WHERE bewegung = 'Glas';
```

c)

```
UPDATE bewegung SET repname = 'Titian', repStern = 'Jungfrau';

SELECT * from bewegung;
SELECT count(*) from folgt;
```

Gesamtpunkte: 70

Viel Erfolg!

Sie können diesen Zettel abtrennen und brauchen ihn nicht abgeben!

Diesen Zettel daher bitte nicht beschriften! (Lösungen auf diesem Zettel werden nicht gewertet!)

Trigger für Aufgabe 6:

```
CREATE FUNCTION upB() RETURNS TRIGGER AS $$
BEGIN
    IF (OLD.repStern = 'Widder') THEN
       RETURN OLD;
    END IF;

    IF (OLD.repName != NEW.repName OR
        OLD.repStern != NEW.repStern) THEN
       DELETE FROM folgt WHERE name = OLD.repName AND stern = OLD.repStern;
       RETURN NEW;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;


CREATE TRIGGER trB BEFORE UPDATE ON bewegung
       FOR EACH ROW EXECUTE PROCEDURE upB();
```

```
CREATE FUNCTION delF() RETURNS TRIGGER AS $$
BEGIN
    IF OLD.stern = 'Jungfrau' THEN
      RETURN NULL;
    END IF;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trF BEFORE DELETE ON folgt
       FOR EACH ROW EXECUTE PROCEDURE delF();
```

Seite B

Beispielinstanz für Aufgabe 6:

Sie können diesen Zettel abtrennen und brauchen ihn nicht abgeben!

Diesen Zettel daher bitte nicht beschriften! (Lösungen auf diesem Zettel werden nicht gewertet!)

Artist

name	stern	hauptBewegung
Klimt	Loewe	Jugendstil
Mucha	Loewe	Jugendstil
Bresson	Loewe	Surrealismus
Da Vinci	Widder	Renaissance
Goya	Widder	Romantik
Botticelli	Widder	Renaissance
Chihuly	Jungfrau	Glas
Titian	Jungfrau	Renaissance

Bewegung

titel	repName	repStern
Jugendstil	Klimt	Loewe
Surrealismus	Bresson	Loewe
Renaissance	Da Vinci	Widder
Glas	Chihuly	Jungfrau
Romantik	Goya	Widder

Folgt

name	stern	bewegung
Klimt	Loewe	Romantik
Klimt	Loewe	Jugendstil
Mucha	Loewe	Jugendstil
Titian	Jungfrau	Romantik
Da Vinci	Widder	Jugendstil
Bresson	Loewe	Glas
Chihuly	Jungfrau	Glas