Fachbereich Informatik

Prof. Dr. Nikolaus Augsten
Jakob-Haringer-Str. 2
5020 Salzburg, Austria
Telefon: +43 662 8044 6347
E-Mail: nikolaus.augsten@plus.ac.at

PARIS LODRON UNIVERSITÄT SALZBURG

Datenbanken II — Prüfung
Wintersemester 2022/2023 — 10.03.2023

Name: _____ **Matrikelnummer:** _____

Hinweise

- Bitte überprüfen Sie die Vollständigkeit des Prüfungsbogens (11 nummerierte Seiten).
- Schreiben Sie Ihren Namen und Ihre Matrikelnummer auf jedes Blatt des Prüfungsbogens und geben Sie alle Blätter ab.
- Grundsätzlich sollten Sie alle Antworten auf den Prüfungsbogen schreiben.
- Sollten Sie mehr Platz für eine Antwort benötigen, bitte einen klaren Verweis neben die Frage auf die Seitennummer des zusätzlichen Blattes setzen.
- Keinen Bleistift verwenden. Keinen roten Stift verwenden.
- Verwenden Sie die Notation und die Lösungsansätze, die während der Vorlesung besprochen wurden.
- Aufgaben mit mehr als einer Lösung werden nicht bewertet.
- Als Unterlage ist ein beliebig (auch beidseitig) beschriftetes A4-Blatt erlaubt.
- Zeit für die Prüfung: 90 Minuten

Unterschrift _____

Korrekturabschnitt — Bitte frei lassen

Aufgabe	1	2	3	4	5	6	7	8	9	10	Summe
Maximale Punkte	1	1	1	1	1	1	1	1	1	1	10
Erreichte Punkte											

Aufgabe 1 - *Slotted Page*. 1 Punkt

In die folgende Tabelle werden Werte eingefügt:

```
CREATE TABLE books (
  bid INTEGER,
  btitle VARCHAR(20)
);
```

Das Speichern eines `INTEGER` benötigt 4 Byte. Strings, die als `VARCHAR` gespeichert werden, benötigen ein Byte pro Zeichen und zusätzlich ein Byte zur Terminierung. Beispielsweise benötigt die Zeichenfolge `DBMS` 5 Bytes zur Speicherung. Die Tupel werden in eine Slotted Page mit folgenden Eigenschaften eingefügt:

- Größe: $2^{13} = 8192$ Bytes,
- Adressierungstyp: **Byte-Adressierung** (es kann jedes Byte adressiert werden)

Die folgenden Operationen werden in dieser Reihenfolge durchgeführt:

```
INSERT INTO books VALUES (42, 'Categoriae');               -- Tupel A
INSERT INTO books VALUES (13, 'Analytica_priora');         -- Tupel B
INSERT INTO books VALUES (37, 'De_sophisticis_elenchis');  -- Tupel C
```

Ergänzen Sie die Slotted Page um die **fehlenden Werte/Adressen**, wobei p_i und g_i sich auf den jeweiligen Datensatz d_i beziehen. **(0.125 Punkte pro Wert/Adresse)**

a	f	g_1	p_1	g_2	p_2	g_3	p_3	$\ldots$	d_3	d_2	d_1
									C	B	A

Aufgabe 2 - *Indexstrukturen.* 1 Punkt

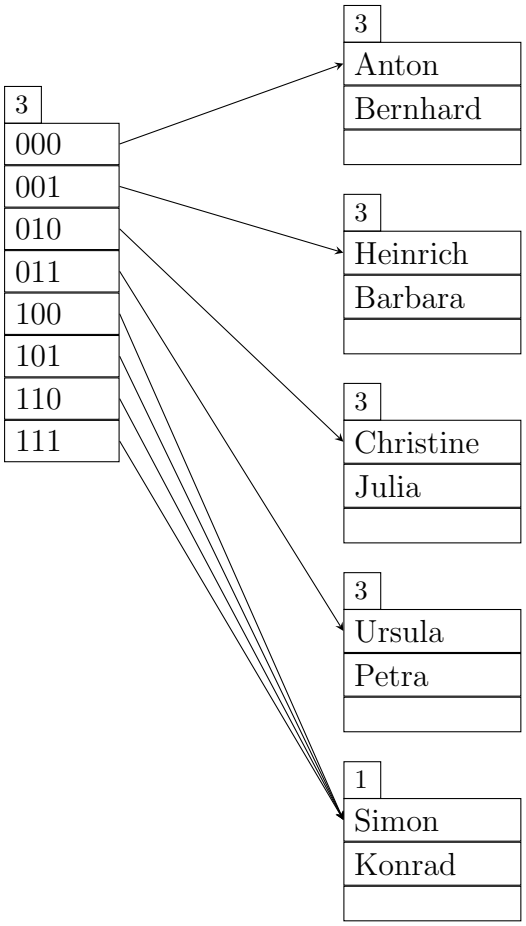
Zeichnen Sie für die folgende Tabelle einen **2-stufigen Clustered Index** auf dem Attribut `Player`. **Beide Indexstufen** sollen **sparse** sein. Ein **Datenblock** kann **2 Tupel** speichern. In einen **Indexblock** können **5 Einträge** gespeichert werden.

`Player`	`Team`
Bell	Steelers
Brees	Saints
Brown	Steelers
Donald	Rams
Edelman	Patriots
Fournette	Jaguars
Gurley	Rams
Houston	Chiefs
Jones	Falcons
Kuechly	Panthers
Lattimore	Saints
Leonard	Colts
Mack	Bears
Michel	Colts
Sanu	Falcons
Thomas	Saints

Aufgabe 3 - *Erweiterbares Hashing*. 1 Punkt

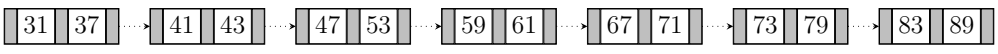
Die Hashfunktion $h(x)$ liefert die in der Tabelle angegebenen Binärwerte. Es soll das Tupel *Heinrich* aus dem gegebenen Hashcontainer gelöscht werden. Ein Bucket im Hashcontainer kann bis zu 3 Tupel speichern. Das Verzeichnis soll so klein wie möglich gehalten werden. **Illustrieren Sie den resultierenden Hashcontainer**.

x	$h(x)$
Anton	0000
Barbara	0010
Bernhard	0001
Christine	0100
Heinrich	0011
Julia	0101
Konrad	1110
Petra	0111
Simon	1100
Ursula	0110



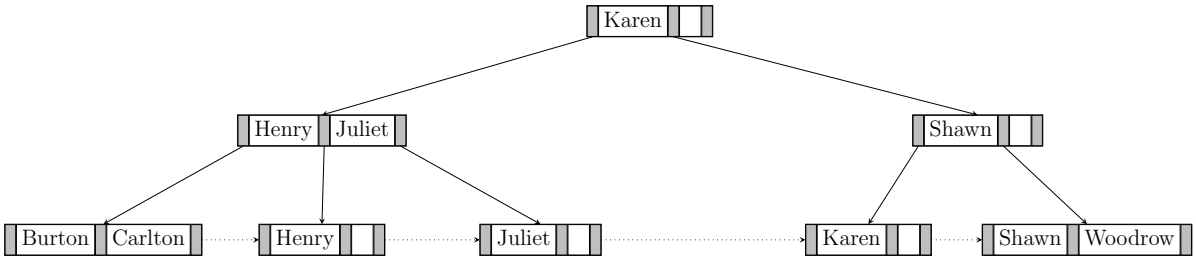
Aufgabe 4 - *B⁺-Baum-Konstruktion.* 1 Punkt

Gegeben sind die Blätter eines B$^+$-Baumes ($m = 3$). Konstruieren Sie die **darüberliegenden Ebenen** (d.h. die inneren Knoten) so, dass ein **gültiger B$^+$-Baum mit minimaler Höhe** entsteht.



Aufgabe 5 - *B^+-Baum-Einfügen.* 1 Punkt

Gegeben ist ein B^+-Baum mit $m = 3$. Zeichnen Sie den B^+-Baum der nach dem Einfügen von **Dexter** entsteht.



Aufgabe 6 - *Effiziente Anfragebearbeitung.* 1 Punkt

Gegeben sei eine Relation $R[A, B, C]$ mit folgenden Eigenschaften:

- $|R| = 8 \cdot 10^9$ Tupel,
- Pro Datenblock werden $16 \cdot 10^3$ Tupel gespeichert,
- Attribut A hat ganzzahlige Werte gleichverteilt im Bereich $[1; 8 \cdot 10^6]$,
- Attribut B hat ganzzahlige Werte gleichverteilt im Bereich $[1; 8 \cdot 10^9]$,
- Attribut C hat ganzzahlige Werte gleichverteilt im Bereich $[1; 8 \cdot 10^3]$,
- Duplikate werden mittels Tuple Identifier (TID) aufgelöst,
- Folgende Indizes existieren:
  - flacher sparse Index auf Attribut A mit 12.500 Einträgen pro Indexblock,
  - dense B^+-Baum-Index auf Attribut B, $m = 4.096$, minimale Höhe

Es soll folgende Anfrage beantwortet werden:

$$\sigma_{B<7\cdot10^3 \wedge B>7\cdot10^3}(R)$$

Geben Sie die **Strategie (0.5 Punkte)** an und berechnen Sie die **Anzahl der Blockzugriffe (0.5 Punkte)** um die Anfrage **möglichst effizient** zu beantworten (1 Knotenzugriff im B^+-Baum entspricht 1 Blockzugriff).

Aufgabe 7 - *Effiziente Anfragebearbeitung*. 1 Punkt

Gegeben ist die Relation $R[A]$. Auf $R.A$ existiert ein sparse B^+-Baum Index. Was ist die **effizienteste Strategie** um Bereichsanfragen von folgendem Typ zu beantworten?

$$\sigma_{a<A<b}(R)$$

Geben Sie **alle notwendigen Schritte** an.

Aufgabe 8 - *Join-Algorithmen.* 1 Punkt

Die Relation $R[\underline{A, B}, C]$ hat $n_R = 3000$ Tupel auf $b_R = 400$ Blöcken, die Relation $S[\underline{B}, D]$ hat $n_S = 2000$ Tupel auf $b_S = 450$ Blöcken. Es stehen $M = 15$ Blöcke im Puffer zur Verfügung. Es soll ein natürlicher Join mittels eines Hash-Join-Algorithmus auf die Relationen durchgeführt werden.

Beantworten Sie die folgenden Fragen und begründen Sie.

- Welche Relation wird als Probe-Input verwendet, welche als Build-Input?
- Was ist die minimale Anzahl von Partitionen, die erstellt werden müssen?
- Schätzen Sie ab, wieviele Block-Zugriffe ein Hash Join braucht.

Aufgabe 9 - *Join Kardinalität.* 1 Punkt

Gegeben seien 3 Relationen $R[A, B, C]$, $S[B, C, D, F]$ und $T[A, D, E, F]$ mit folgenden Eigenschaften:

- $|R| = 2500$ Tupel, $V(R, A) = 50$, $V(R, B) = 10$, $V(R, C) = 20$
- $|S| = 500$ Tupel, $V(S, B) = 50$, $V(S, C) = 40$, $V(S, D) = 10$, $V(S, F) = 100$
- $|T| = 4000$ Tupel, $V(T, A) = 25$, $V(T, D) = 50$, $V(T, E) = 200$, $V(T, F) = 20$

Schätzen Sie $|R \bowtie S \bowtie T|$ ab.

Aufgabe 10 - *Anfrageoptimierung, Join-Reihenfolge.* 1 Punkt

Gegeben seien 3 Relationen $R[A, B, C]$, $S[A, D, E]$ und $T[A, F, G]$:

- $|R| = 1.200$ Tupel, $V(R, A) = 50$, $V(R, B) = 100$, $V(R, C) = 200$
- $|S| = 3.000$ Tupel, $V(S, A) = 20$, $V(S, D) = 1.000$, $V(S, E) = 600$
- $|T| = 5.000$ Tupel, $V(T, A) = 100$, $V(T, F) = 1.200$, $V(T, G) = 1.800$

Weiters sei die folgende SQL-Anfrage gegeben:

```
SELECT      R.A, S.D, T.G
FROM        R, S, T
WHERE       R.A = S.A AND R.A = T.A
```

1. Zeichnen Sie die **algebraische Normalform als Operatorbaum** für die gegebene SQL-Anfrage. **(0.5 Punkte)**
2. Wenden Sie die heuristische Optimierung an, um den Operatorbaum zu optimieren. **Es soll auch die Join-Reihenfolge optimiert werden! (0.5 Punkte)**