



Übung zur Vorlesung *Einsatz und Realisierung von Datenbanksystemen* im SoSe19

Maximilian {Bandle, Schüle} (i3erdb@in.tum.de)
<http://db.in.tum.de/teaching/ss19/impldb/>

Blatt Nr. 08

Hausaufgabe 1

Wie ändert sich die Bedeutung des Redo-Log und Undo-Log in Hauptspeicherdatenbanken im Vergleich zu klassischen Datenbanken? Wo werden sie gespeichert?

Hausaufgabe 2

HyPer schafft 120.000 Transaktionen pro Sekunde. Pro Transaktion werden 120 Byte in die Log geschrieben. Berechnen Sie den benötigten Durchsatz zum Schreiben der Log.

Die Datenbank läuft für einen Monat und stürzt dann ab. Es wurde kein Snapshot erstellt. Berechnen Sie die Recoveryzeit. Gehen Sie davon aus, dass die Recovery durch die Festplatte limitiert ist (100 MiB / s). Wieviel Log Einträge werden pro Sekunde reconvert?

Hausaufgabe 3

Gegeben eine Tabelle *Produkte* mit folgendem Schema und 10000 Einträgen:

Id (8 Byte) | Name (32 Byte) | Preis (8 Byte) | Anzahl (8 Byte)

Wieviele Daten werden für folgende Queries in die CPU-Caches geladen? Unterscheiden sie jeweils zwischen Row und Column Store.

1. *select * from Produkte*
2. *select Anzahl from Produkte*

Hausaufgabe 4

Rekonstruieren Sie die ursprüngliche SQL-Anfrage aus dem folgenden (Pseudo-)Code eines codegenerierenden Datenbanksystems. Welche Art von Join wurde benutzt? Handelt es sich um Column- oder Row-Store?

```
struct Student { int matrnr; std::string name; int semester; };
struct Hoeren { int matrnr; int vorlnr; };
struct Result { int vorlnr; int a; };

std::vector<Result> compute(std::vector<Student>&ses, std::vector<Hoeren>&hs){
    std::unordered_multimap<int, Hoeren*> h_map;
    std::unordered_map<int, Student*> s_map;
    for (auto &h : hs)
        h_map.insert(std::make_pair(h.matrnr, &h));
    for (auto &s : ses)
        s_map.insert(std::make_pair(s.matrnr, &s));
```

```

std::unordered_map<int, int> count_map;
std::unordered_map<int, int> sum_map;
for (auto &h : hs) {
    count_map.insert(std::make_pair(h.vorlnr, 0));
    sum_map.insert(std::make_pair(h.vorlnr, 0));
}
for (auto &h : h_map) {
    sum_map[h.second->vorlnr] += s_map[h.first]->semester;
    count_map[h.second->vorlnr]++;
}
std::vector<Result> res;
for (auto &r : sum_map)
    res.push_back({ r.first, r.second / count_map[r.first] });
return res;
}

```

Hinweis Beantworten Sie die folgenden Anfragen mit Hilfe von SQL und den TPC-H Daten.

Hausaufgabe 5

1. Wie viele Kunden mit einer Bestellung, deren Kommentar (`o_comment`) das Wort „packages“ enthält, gibt es?
2. Was ist die durchschnittliche Anzahl von Dezimalstellen in `l_orderkey`?
3. Was sind die Namen aller Kunden und Zulieferer?

Hausaufgabe 6

1. Berechnen Sie für jede Bestellung den Rang (`rank`) nach `o_totalprice`, wobei der höchste Preis dem Rang 1 entspricht.
2. Berechnen Sie für jede Bestellung die wachsende Summe von `o_totalprice` nach Datum (`o_orderdate`). Die Summe soll jedes Jahr neu beginnen.
3. Was ist der Median von `o_totalprice`?