



Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken* im SoSe20

Maximilian {Bandle, Schüle}, Josef Schmeißer (i3erdb@in.tum.de)

<http://db.in.tum.de/teaching/ss20/impldb/>

Blatt Nr. 07

Hausaufgabe 1

Analysieren wir die Gehälter von Professoren mittels Windowfunctions und führen Sie die Abfragen unter hyper-db.de aus. Dazu orientieren wir uns an der Relation *Professoren* des erweiterten Universitätsschemas:

```
with Professoren (persnr, name, rang, raum, gehalt, steuerklasse) as (  
  values (2125, 'Sokrates', 'C4', 226, 85000, 1),  
         (2126, 'Russel', 'C4', 232, 80000, 3),  
         (2127, 'Kopernikus', 'C3', 310, 65000, 5),  
         (2128, 'Aristoteles', 'C4', 250, 85000, 1),  
         (2133, 'Popper', 'C3', 52, 68000, 1),  
         (2134, 'Augustinus', 'C3', 309, 55000, 5),  
         (2136, 'Curie', 'C4', 36, 95000, 3),  
         (2137, 'Kant', 'C4', 7, 98000, 1)  
)
```

1. Ermitteln Sie zu jedem Professor das Durchschnittsgehalt aller Professoren.
2. Ermitteln Sie zu jedem Professor das Durchschnittsgehalt aller Professoren partitioniert nach Rang.
3. Ermitteln Sie nun die wachsende Summe (das Quantil) des Gehaltes aller Professoren partitioniert nach Rang und absteigend sortiert nach ihrem Gehalt. Gleich verdienende Professoren sind im selben Quartil.
4. Ermitteln Sie nun die wachsende Summe des Gehaltes aller Professoren partitioniert nach Rang und absteigend (total) sortiert nach ihrem Gehalt (reihenweise, nicht als Range-Query).
5. Ermitteln Sie nun das gleitende Durchschnittsgehalt aus genau zwei mehr bzw. weniger verdienenden Professoren sortiert nach Gehalt und partitioniert nach Rang.
6. Ermitteln Sie nun das gleitende Durchschnittsgehalt aus den 500 Einheiten mehr bzw. weniger verdienenden Professoren sortiert nach Gehalt und partitioniert nach Rang.
7. Geben sie zu jedem Professor das Gehalt des eins besser wie eins schlechter verdienenden.
8. Ermitteln Sie die drei bestverdienendsten Professoren einmal mit und einmal ohne Windowfunctions.

Hausaufgabe 2

Lösen Sie folgende Anfrage mit SQL basierend auf dem bekannten Universitätschema.

1. Bestimmen Sie die Durchschnittsnote für jeden Studenten.

2. Basierend auf dieser Durchschnittsnote, bestimmen Sie für alle Studenten ihren Rangplatz innerhalb ihrer Kohorte (Studenten desselben Semesters).
3. Berechnen Sie zusätzlich für jeden Studenten auch noch die **Abweichung** seiner Durchschnittsnote von der Durchschnittsnote der Kohorte (also vom Durchschnitt der Durchschnittsnote der Studenten der Kohorte) ausgegeben werden.

Lösen Sie Teilaufgaben 2 und 3 jeweils einmal mit und einmal ohne Nutzung von Windowfunktionen. Ihre Anfragen können Sie auf hyper-db.de testen. Nutzen Sie folgende erweiterte *pruefen* Relation:

```
with mehr_pruefen(MatrnNr, VorlNr, PersNr, Note) as (
  select * from pruefen
  union
  values (29120,0,0,3.0), (29555,0,0,2.0), (29555,0,0,1.3), (29555,0,0,1.0)
)
```

Hausaufgabe 3

Betrachten wir das bekannte Uni-Schema mit den Faktentabellen **hoeren** und **pruefen** .

1. Ermitteln Sie in SQL mittels Fensterfunktionen (Windowfunctions) die Top-3 Studenten pro Vorlesung und geben Sie deren Namen aus.
2. Ermitteln Sie mittels SQL-92, um wieviele Notenstufen Studenten, die die Vorlesung gehört haben, in der Prüfung besser abgeschnitten haben.

Hausaufgabe 4

Wenden wir nun Fensterfunktionen an dem fiktiven TPC-H Schema an.

1. Erstellen Sie in SQL eine Abfrage nach dem jährlichen Produktionsvolumen pro Jahr und Land. Verwenden Sie diese Abfrage als Hilfstabelle (**with-Statement**) in den folgenden Abfragen, orientieren Sie sich an TPC-H Anfrage 7.
2. Ranken Sie Länder anhand ihres jährlichen Produktionsvolumens, auf Platz eins ist das Land mit dem höchsten Volumen, das je in einem Jahr getätigt worden ist.
3. Küren Sie nun die Jahressieger. Ranken Sie dazu die Länder partitioniert nach Jahr.
4. Ermitteln Sie nun das laufende Produktionsmittel der Länge drei (Vorjahr, Nachjahr, falls erfasst).

Hausaufgabe 5

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
...

- 1 Ermitteln Sie in SQL mittels Fensterfunktionen (Windowfunctions) den prozentualen Umsatzanteil jedes Produktes innerhalb seiner Kategorie.
- 2 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.
- 3 Ermitteln Sie in SQL mittels Fensterfunktionen (Windowfunctions) die drei Produkte mit dem meisten Umsatz pro Kategorie.