TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Alfons Kemper, Ph.D.

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. 08

Hausaufgabe 1

Der Datenbanken-Lehrstuhl möchte wissen, mit welchem Eis der Gefrierschrank bestückt werden soll. Die Kosten sollen möglichst gering sein, aber die Schleckzeit möglichst groß. Hierfür wurde ein Test mit handelsüblichen Eissorten durchgeführt.

`Eis`

id	Name	Schleckzeit (*min*)	Kosten (*ct*)
D	Double-Stieleis	5	45
E	Eiskonfekt	7	50
F	Frucht-Stieleis	4	30
G	Großes Stieleis	5	35
M	Mini-Stieleis	2	15
Q	Quetschtüte	3	25
S	Sandwich-Eis	5	35
W	Waffeltüte	4	25

Wir betrachten die Skyline über das **Maximum** des Attributs *Schleckzeit* sowie das **Minimum** des Attributs *Kosten* der Tabelle `Eis`.

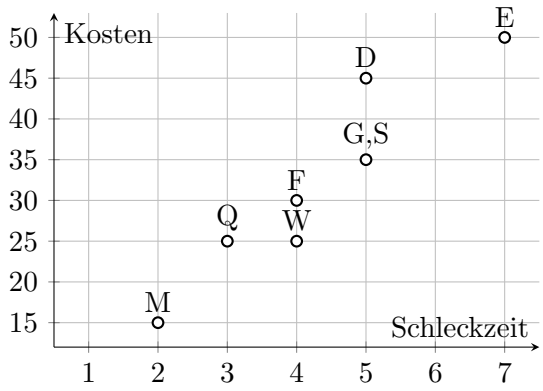
a) Geben Sie die Anfrage, die die oben genannte Skyline mithilfe des Skyline-Operators berechnet.

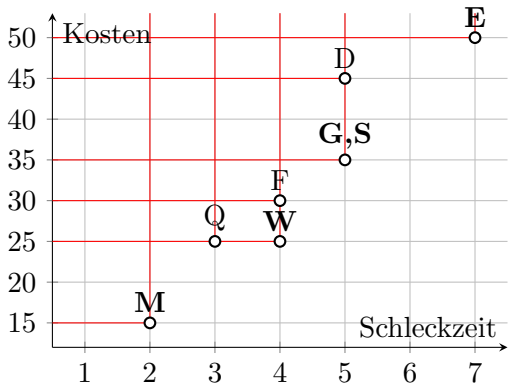
```
SELECT id FROM Eis e
SKYLINE of e.Schleckzeit max, e.Kosten min
```

b) Geben Sie die Anfrage, die die oben genannte Skyline in SQL-92 berechnet, an (d.h. ohne Skyline-Operator).

```
SELECT id FROM Eis e WHERE NOT EXISTS (
  SELECT * FROM Eis dom WHERE
  (dom.Kosten <= e.Kosten AND dom.Schleckzeit >= e.Schleckzeit) AND
  (dom.Kosten < e.Kosten OR dom.Schleckzeit > s.Schleckzeit)
)
```

c) Vervollständigen Sie das unten gezeigte Diagramm. Zeichnen Sie alle Dominanzachsen ein.

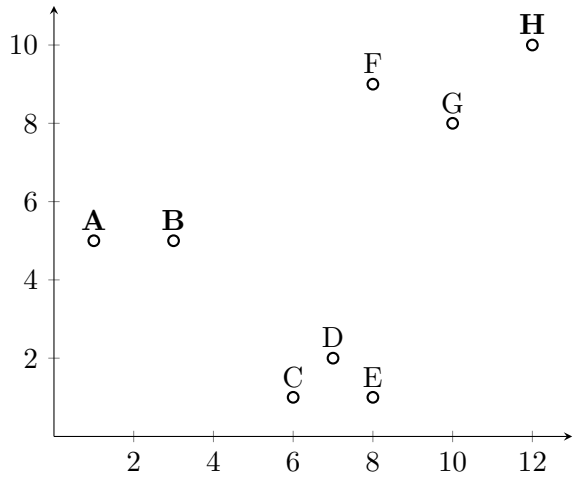




d) Geben Sie die Kürzel aller in der Skyline enthalten Tupel an.
M, W, G, S, E
Q, F werden von W dominiert; D wird von G,S dominiert

Hausaufgabe 2

Gegeben seien folgende Datenpunkte, die im Plot und der Tabelle dargestellt sind.
Die Punkte sollen mithilfe des k-Means-Algorithmus in drei Cluster aufgeteilt werden.



Punkt	X	Y
A	1	5
B	3	5
C	6	1
D	7	2
E	8	1
F	8	9
G	10	8
H	12	10

Als initiale Clusterzentren werden dabei folgende Punkte gewählt:
Cluster $(C_1) \rightarrow A$; Cluster $(C_2) \rightarrow B$; Cluster $(C_3) \rightarrow H$.

a) Führen Sie die Zuordnung für die erste Iteration qualitativ durch, indem sie das zugehörige Feld ankreuzen. Eine Rechnung oder Begründung ist nicht erforderlich.

	A	B	C	D	E	F	G	H
C_1	✔							
C_2		✔	✔	✔	✔			
C_3						✔	✔	✔

b) Berechnen Sie den Mittelpunkt M_3 von Cluster C_3 für die erste Iteration (Rechenweg angeben).
x = $(8 + 10 + 12)/3 = 10$
y = $(9 + 8 + 10)/3 = 9$
$M_3 = (10, 9)$

c) Nennen Sie die Bedingung, nach der k-Means das Clustering optimiert.

Abstände von Punkt zu Clusterzentren sind minimal.

d) Geben Sie die Terminierungsbedingung von k-Means an.

Keine Änderung der Zuordnung zu den Clustern zwischen den Iterationen.

Hausaufgabe 3

Gegeben seien Datenpunkte, welche im nachfolgenden Listing aufgeführt sind. Die Punkte sollen mithilfe des k-Means-Algorithmus in drei Cluster aufgeteilt werden. Als initiale Clusterzentren werden hierbei die jeweiligen Datenpunkte aus der `clusters_0`-Hilfsrelation gewählt.

```
with points (pid, x, y) as (values('A',1,5), ('B',3,5), ('C',6,1),
  ('D',7,2), ('E',8,1), ('F',8,9), ('G',10,8), ('H',12,10)
), clusters_0 (cid,x,y) as (values ('1',1e0,5e0), ('2',3e0,5e0),
  ('3',12e0,10e0))
```

a) Formulieren Sie eine Iteration des k-Means-Algorithmus in SQL, die Ihnen die Clusterzentren zurückgibt. Nutzen Sie dazu eine Unterabfrage, die das Kreuzprodukt aus Clustern und Punkten berechnet und mit einer Window-Funktion pro Punkt ein Ranking der Cluster anhand der euklidischen Distanz erstellt.

```
[...]
clusters_1(cid, x,y, count) as (
  select cid, avg(px), avg(py), count(*) from (
    select cid, p.x as px, p.y as py, rank() over (partition by pid
      order by (p.x-c.x)*(p.x-c.x)+(p.y-c.y)*(p.y-c.y) asc,
        (c.x*c.x+c.y*c.y) asc)
    from points p, clusters_0 c) x
  where x.rank=1 group by cid
)
```

b) Geben Sie anschließend die neuen Clusterzentren aus.

```
[...]
select * from clusters_1
```

c) Berechnen Sie nun auf Grundlage Ihrer vorgehenden Anfrage die Zuordnung der Datenpunkte zu den jeweiligen Clusterzentren.

```
[...]
select cid,pid from (
    select cid, pid, rank() over (partition by pid
        order by (p.x-c.x)*(p.x-c.x)+(p.y-c.y)*(p.y-c.y) asc,
            (c.x*c.x+c.y*c.y) asc)
    from points p, clusters_1 c) x
where x.rank=1
```

d) Formulieren Sie nun Clusterberechnung als rekursive SQL-Anfrage mit folgendem Schema: `clusters_n (cid,step,x,y,delta)`. Nehmen Sie initial die gegebenen Clusterzentren. Verwenden Sie als Vorlage im Rekursionsschritt Ihre Anfrage aus Teilaufgabe a, welche die Clusterzentren pro Iteration neuberechnet (`assign`). Hinweis: Nutzen Sie für die Fixpunkiteration `delta` als die Summe aller Änderungen in Schritt

`step`, um die Terminierungsbedingung des k-Means-Algorithmus zu formulieren. Ihre Anfrage soll terminieren, wenn die neu zugewiesenen Zentren gleich den vorherigen sind: `delta = 0`.

```
with recursive
[...]
clusters_n (cid, x, y, step, delta) as (
  select c.cid, c.x, c.y, 0 as step, 1e0 as delta
  from clusters_0 c
union all
  select cp.cid,
    avg(assign.x) as cx, avg(assign.y) as cy, curr_step.step,
    (avg(assign.x)-cp.x)*(avg(assign.x)-cp.x) +
    (avg(assign.y)-cp.y)*(avg(assign.y)-cp.y) as delta
  from (
      select c.cid as cid, p.x as x, p.y as y,
        rank() over (partition by p.pid order by
          (c.x - p.x)*(c.x - p.x) + (c.y - p.y)*(c.y - p.y) asc, cid asc)
      from points p, clusters_n c
      where c.step = (select max(step) from clusters_n)
    ) as assign,
    ( select max(step)+1 as step from clusters_n ) as curr_step,
    ( select sum(delta) as s from clusters_n ) as delta_sum,
  clusters_n cp
  where rank = 1 and cp.cid = assign.cid and delta_sum.s > 0
  group by cp.cid, cp.x, cp.y, curr_step.step, delta
)
```

Die Clusterzentren können wiederum mit folgender Abfrage ausgegeben werden:

```
[...]
select * from clusters_n
```

Hausaufgabe 4

Alex und Max möchten sich für ihre neue Firma ein Fortbewegungsmittel zulegen. Hilf ihnen, die drei günstigsten bei 40.000 km Fahrleistung pro Jahr zu finden, wenn sie das Auto 5 Jahre lang nutzen wollen. Wende den NRA- und Threshold-Algorithmus an und bilde eine Skyline.

Einheit	Treibstoff	Preis
1l	Diesel	1,00€
1l	Benzin	1,50€
1l	Kerosin	1,00€
1kWh	Strom	0,10€

Kosten

Gefährt	Kosten
Privatjet	2.500.000€
Elektroauto	80.000€
Cabrio	40.000€
Limousine	35.000€
Transporter	20.000€
Combi	25.000€
Sport-Coupé	25.000€

Verbrauch

Gefährt	Verbrauch
Privatjet	0,2l/km (Kerosin)
Elektroauto	20kWh/100km (Strom)
Cabrio	4l/100km (Diesel)
Limousine	5l/100km (Diesel)
Transporter	6l/100km (Benzin)
Combi	5l/100km (Benzin)
Sport-Coupé	4l/100km (Benzin)

Kosten sortiert

Gefährt	Kosten
Transporter	20.000€
Sport-Coupé	25.000€
Combi	25.000€
Limousine	35.000€
Cabrio	40.000€
Elektroauto	80.000€
Privatjet	2.500.000€

Spritkosten für 5 Jahre: Gesamtleisung 200.000km

Gefährt	Kosten
Elektroauto	20kWh/100km * 200.000km * 0,1€/kWh (Strom) = 4.000€
Cabrio	4l/100km * 200.000km * 1€/l (Diesel) = 8.000€
Limousine	5l/100km * 200.000km * 1€/l (Diesel) = 10.000€
Sport-Coupé	4l/100km * 200.000km * 1,5€/l (Benzin) = 12.000€
Combi	5l/100km * 200.000km * 1,5€/l (Benzin) = 15.000€
Transporter	6l/100km * 200.000km * 1,5€/l (Benzin) = 18.000€
Privatjet	0,2l/km * 200.000km * 1€/l (Kerosin) = 40.000€

NRA

Zw. Ergebnis: Phase 1		
Transporter	24.000€	↗
Elektroauto	24.000€	↗

Zw. Ergebnis: Phase 2		
Transporter	28.000€	↗
Elektroauto	29.000€	↗
Sport-Coupé	33.000€	↗
Cabrio	33.000€	↗

Zw. Ergebnis: Phase 3		
Elektroauto	29.000€	↗
Transporter	30.000€	↗
Cabrio	33.000€	↗
Sport-Coupé	35.000€	↗
Combi	35.000€	↗
Limousine	35.000€	↗

Zw. Ergebnis: Phase 4		
Transporter	32.000€	↗
Combi	37.000€	↗
Sport-Coupé	37.000€	√
Elektroauto	39.000€	↗
Cabrio	43.000€	↗
Limousine	45.000€	√

Zw. Ergebnis: Phase 5		
Transporter	35.000€	↗
Sport-Coupé	37.000€	√
Combi	40.000€	√
Elektroauto	44.000€	↗
Limousine	45.000€	√
Cabrio	48.000€	√

Zw. Ergebnis: Phase 6		
Sport-Coupé	37.000€	√
Transporter	38.000€	√
Combi	40.000€	√
Limousine	45.000€	√
Cabrio	48.000€	√
Elektroauto	84.000€	√

Threshold

Zw. Ergebnis: Phase 1	
Threshold	24.000€
Transporter	38.000€
Elektroauto	84.000€

Zw. Ergebnis: Phase 2	
Threshold	33.000€
Sport-Coupé	37.000€
Transporter	38.000€
Cabrio	48.000€
Elektroauto	84.000€

Zw. Ergebnis: Phase 3	
Threshold	35.000€
Sport-Coupé	37.000€
Transporter	38.000€
Combi	40.000€
Limousine	45.000€
Cabrio	48.000€
Elektroauto	84.000€

Zw. Ergebnis: Phase 4	
Sport-Coupé	37.000€
Transporter	38.000€
Combi	40.000€
Limousine	45.000€
Threshold	47.000€
Cabrio	48.000€
Elektroauto	84.000€

Skyline
Alle Fortbewegungsmittel ausser Combi und Privatjet sind in Skyline enthalten.

Combi Von Sport-Coupé dominiert

Privatjet Von allen dominiert

Hausaufgabe 5

Zeigen Sie die weiteren Phasen des Apriori-Algorithmus für unser Beispiel in Abbildung 1 (hier ist lediglich bis inkl. 2. Phase dargestellt). Damit eine Menge von Produkten ein Frequentitemset ist, muss sie in mindestens 3/5 aller Verkäufe enthalten sein, d.h. $minsupp = s_0 = 3/5$. Gehen Sie für die Assoziationsregeln von einer minimalen Konfidenz von $k_0 = 0$ aus und berechnen Sie die Konfidenz der Assoziationsregel {Drucker}⇒ {Papier, Toner}.

VerkaufsTransaktionen	
TransID	Produkt
111	Drucker
111	Papier
111	PC
111	Toner
222	PC
222	Scanner
333	Drucker
333	Papier
333	Toner
444	Drucker
444	PC
555	Drucker
555	Papier
555	PC
555	Scanner
555	Toner

Zwischenergebnisse	
FI-Kandidat	Anzahl
{Drucker}	4
{Papier}	3
{PC}	4
{Scanner}	2
{Toner}	3
{Drucker, Papier}	3
{Drucker, PC}	3
{*Drucker, Scanner*}	
{Drucker, Toner}	3
{Papier, PC}	2
{*Papier, Scanner*}	
{Papier, Toner}	3
{*PC, Scanner*}	
{PC, Toner}	2
{*Scanner, Toner*}	

Abbildung 1: Ausgangssituation für den Apriori-Algorithmus

Vgl. Übungsbuch 17.6. Frequentitemsets sind alle nicht gestrichenen (wegen zu geringem Supports) bzw. nicht kursiv gesetzten (wegen nicht häufig auftretender Teilmenge).

Iteration	Item-Menge X	$\sigma(X)$	s(X)
1	{Drucker}	4	4/5
1	{Papier}	3	3/5
1	{PC}	4	4/5
1	~~{Scanner}~~	2	2/5
1	{Toner}	3	3/5
2	{Drucker, Papier}	3	3/5
2	{Drucker, PC}	3	3/5
2	{*Drucker, Scanner*}		
2	{Drucker, Toner}	3	3/5
2	~~{Papier, PC}~~	2	2/5
2	{*Papier, Scanner*}		
2	{Papier, Toner}	3	3/5
2	{*PC, Scanner*}		
2	~~{PC, Toner}~~	2	2/5
2	{*Scanner, Toner*}		
3	{*Drucker, Papier, PC*}		
3	{Drucker, Papier, Toner}	3	3/5
3	{*Drucker, PC, Toner*}		
3	{*Papier, PC, Toner*}		

Der Vollständigkeit halber im Nachfolgenden alle möglichen Assoziationsregeln.

Item-Menge X	$\sigma(X)$	$s(X)$	$c(X)$
$\emptyset \Rightarrow$ {Drucker}	4	4/5	4/5
$\emptyset \Rightarrow$ {Papier}	3	3/5	3/5
$\emptyset \Rightarrow$ {PC}	4	4/5	4/5
$\emptyset \Rightarrow$ {Toner}	3	3/5	3/5
$\emptyset \Rightarrow$ {Drucker, Papier}	3	3/5	3/5
{Drucker}$\Rightarrow$ {Papier}	3	3/5	3/4
{Papier}$\Rightarrow$ {Drucker}	3	3/5	3/3
$\emptyset \Rightarrow$ {Drucker, PC}	3	3/5	3/5
{Drucker}$\Rightarrow$ {PC}	3	3/5	3/4
{PC}$\Rightarrow$ {Drucker}	3	3/5	3/4
$\emptyset \Rightarrow$ {Drucker, Toner}	3	3/5	3/5
{Drucker}$\Rightarrow$ {Toner}	3	3/5	3/4
{Toner}$\Rightarrow$ {Drucker}	3	3/5	3/3
$\emptyset \Rightarrow$ {Papier, Toner}	3	3/5	3/5
{Papier}$\Rightarrow$ {Toner}	3	3/5	3/3
{Toner}$\Rightarrow$ {Papier}	3	3/5	3/3
$\emptyset \Rightarrow$ {Drucker, Papier, Toner}	3	3/5	3/5
{Drucker}$\Rightarrow$ {Papier, Toner}	3	3/5	3/4
{Drucker, Papier}$\Rightarrow$ {Toner}	3	3/5	3/3
{Drucker, Toner}$\Rightarrow$ {Papier}	3	3/5	3/3
{Papier}$\Rightarrow$ {Drucker, Toner}	3	3/5	3/3
{Papier, Toner}$\Rightarrow$ {Drucker}	3	3/5	3/3
{Toner}$\Rightarrow$ {Drucker, Papier}	3	3/5	3/3