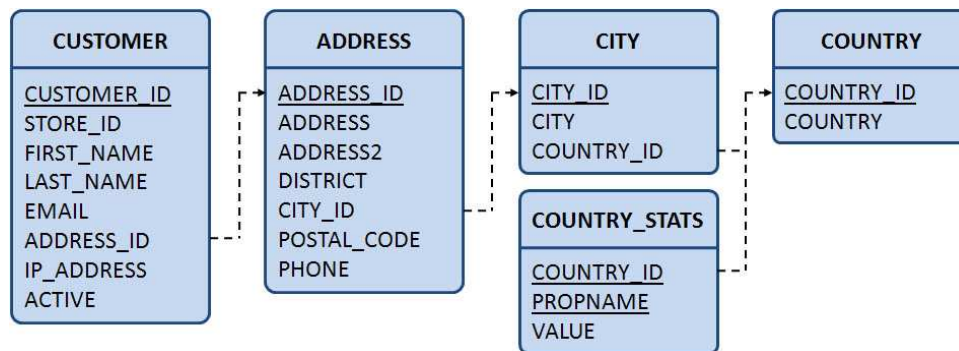

EXAMEN: Scriptingtalen

Prof. Dr. Peter Dawyndt
1^e Bachelor Informatica
groep 1

dinsdag 18-08-2009, 14:00h
academiejaar 2008-2009
tweede zittijd

Opgave 1

De DVD-verhuurketen heeft een aanpassing doorgevoerd aan de manier waarop de adressen van haar klanten worden bijgehouden in de Sakila databank. Bovendien werden extra statistieken over landen toegevoegd om een betere marktpolitiek te kunnen uitstippelen. Hiervoor werden drie extra tabellen aangemaakt in de oorspronkelijke databank. Onderstaande figuur geeft een overzicht van de nieuwe en aangepaste tabellen, waarbij (samengestelde) primaire sleutels onderlijnd zijn en vreemde sleutels worden aangegeven met onderbroken pijlen.



De extra statistieken over landen worden bijgehouden in de tabel **COUNTRY_STATS**, waarbij elke record de naam (kolom **PROPNAME**) en de waarde (kolom **VALUE**) van een statistiek van een bepaald land (kolom **COUNTRY_ID**) bevat. De eigenschappen die momenteel worden bijgehouden zijn: onderwijsbudget (%; **educexp**), bruto nationaal product(\$; **gdpcapit**), toename bruto nationaal product (%; **gdpgrowth**), koopkrachtpariteit (\$; **gdpppp**), inflatiepeil (%; **inflation**), defensiebudget (%; **militaryexp**), bevolkingstoename (%; **popgrowth**), bevolkingsaantal (**population**), totale oppervlakte (km²; **surface**) en werkeloosheidsgraad (%; **unemploy**). Hierbij staat tussen ronde haakjes telkens de gebruikte eenheid en de engelstalige naam van de eigenschap zoals die in de databank gebruikt wordt.

De uitgebreide Sakila databank werd ingeladen in een MySQL RDBMS en is bevaagbaar via de phpMyAdmin webapplicatie (<http://users.ugent.be/~dhollevo/scrita.sql/>). Inloggen doe je in deze databank met je UGent account. Gevraagd wordt om SQL zoekopdrachten te formuleren die een antwoord bieden op onderstaande vragen. Zorg er telkens voor dat de kolommen van de resulterende tabel een zinvolle naam krijgen. Geef in je antwoordbestand ook aan hoeveel records de resultatentabel bevat.

1. Geef een lijst van landen met hun bruto nationaal product, waarvoor het bruto nationaal product minstens 20% boven het gemiddelde bruto nationaal product van alle landen ligt. Sorteer deze lijst in dalende volgorde van bruto nationaal product.
2. Geef het/de land(en) met de grootste bevolkingsdichtheid. Hou hierbij rekening met eventuele ex-aequo's. De bevolkingsdichtheid van een land wordt berekend als het totaal aantal inwoners gedeeld door de totale oppervlakte van het land.
3. Geef alle koppels van landen waarvan alle klanten samen meer dan 500 online bestellingen geplaatst hebben. Sorteer de bekomen lijst volgens totaal aantal bestellingen.

4. Geef een lijst van namen (voornaam + familienaam) van klanten die in een land wonen waar de werkloosheidsgraad ten hoogste 5% bedraagt. Gebruik voor het opvragen van de klanten van de DVD-verhuurketen de tabel `customer` en niet de tabel `online_customer` of de view `customer_list`.

Opmerking: Het is niet toegelaten om gebruik te maken van MySQL-specifieke uitbreidingen die niet voldoen aan de SQL standaard. Dit geldt bijvoorbeeld voor het sleutelwoord `LIMIT`.

Opgave 2

Bij het spelletje scrabble komt het erop aan om bestaande woorden te vormen met een gegeven reeks letters. We willen daarom een `bash` shell script `scrabble` schrijven dat in een gegeven woordenlijst alle woorden opzoekt die kunnen gevormd worden met de lettercombinatie die als argument aan het shell script wordt doorgegeven. Daarbij is het niet noodzakelijk dat de letters uit de lettercombinatie allemaal gebruikt worden, maar elke letter uit de lettercombinatie mag hoogstens één keer gebruikt worden. Hoofdletters en kleine letters mogen als verschillende letters beschouwd worden. Hieronder hebben we reeds het ontwerp gemaakt van het shell script `scrabble`.

```
#!/bin/sh

WOORDENLIJST='~/woordenboek.txt'

echo $1 | cat - ${WOORDENLIJST} | gsed -n '
...
;
```

Hierbij verwijst de shell variabele `WOORDENLIJST` naar een tekstbestand dat een woordenlijst bevat, waarbij elk woord op een afzonderlijke regel staat. In bovenstaand ontwerp hebben we hiervoor het bestand `woordenboek.txt` gebruikt dat in de home directory van de gebruiker staat, maar je kunt evengoed een andere woordenlijst gebruiken (bv. `/usr/dict/words`).

Het shell script zorgt er eerst voor dat de lettercombinatie gevolgd door de woordenlijst als invoer naar een pipe worden gestuurd. Daarna is het de bedoeling dat een `gsed`-script enkel die woorden uit de woordenlijst filtert die kunnen gevormd worden met letters uit de lettercombinatie, en deze wegschrijft naar standaard output. Onderstaande sessie illustreert hoe het shell script kan gebruikt worden

```
$ scrabble scriptingtalen
actie
acties
agent
ai
al
alen
alge
algen
alpinist
alpiniste
alpinisten
...
```

Het is jouw opdracht om een dergelijk `gsed`-script te schrijven op de plaats waar de drie puntjes staan in het ontwerp. Je bent vrij om het `gsed`-script te schrijven zoals je zelf wilt, maar een mogelijke strategie wordt hieronder beschreven:

1. Sla het eerste woord (de opgegeven lettercombinatie) op in de hold buffer, zodat je het in een latere fase te allen tijde kunt hergebruiken.
2. Voor alle andere woorden maak je eerst een kopie van het woord in de pattern space, waarbij het duplicaat van het origineel wordt gescheiden door een verticale streep. Indien het huidige woord `actie` is, dan bevat de pattern space hierna de tekst `actie|actie`.
3. Voeg nu de inhoud van de hold space achteraan toe aan de pattern space.
4. Gebruik een lus in `gsed` waarbij je telkens de eerste letter in het duplicaat van het huidige woord en een corresponderende letter in de opgegeven lettercombinatie schrapt, totdat alle letters van het duplicaat zijn opgebruikt of totdat er geen corresponderende letter meer is.
5. Schrijf de originele versie van het huidige woord (stuk voor de verticale streep in de pattern space) enkel uit indien alle letters van het duplicaat zijn opgebruikt.

Voor deze opgave moet je het volledige shell script `scrabble` indienen, niet enkel het `gsed`-script dat je hieraan hebt toegevoegd.

Opgave 3

Voor de informatie-eenheden *bytes*, *bits*, en *bits per seconde* worden vaak voorvoegsels gebruikt om grote hoeveelheden te kunnen kwantificeren. Traditioneel werden hiervoor de SI-veelvouden kilo, mega, giga, tera etc. gebruikt. In de computerwereld werden ze echter anders gebruikt dan officieel voorgeschreven. Omdat opslagcapaciteiten vaak in machten van 2 uitgedrukt worden en $2^{10} = 1024$ zo dicht bij 1000 ligt, werd ooit bedacht dat het handig zou zijn niet 1000 byte maar 1024 byte een kilobyte te noemen. Dit werd verwarrend toen de opslagcapaciteiten groter werden.

Zo heeft een floppy van 1,44 megabyte niet een capaciteit van $1,44 \times 1024 \times 1024$ byte, en ook niet van $1,44 \times 1000 \times 1000$ byte, maar van $1,44 \times 1024 \times 1000$. Het RAM-geheugen in een computer wordt traditioneel in megabytes van 1024×1024 byte aangegeven, maar de capaciteit van harde schijven wordt in gigabytes van $1000 \times 1000 \times 1000$ byte opgegeven. Telecommunicatiesnelheden worden ook in machten van 1000 uitgedrukt, een lijn van 34 megabit/seconde transporteert dus 34.000.000 bit per seconde.

In 1998 is door de International Electrotechnical Commission (IEC) besloten deze verwarring uit de wereld te helpen door een nieuwe standaard (IEEE 1541-2002) voor binaire voorvoegsels te maken. Sindsdien is het niet meer gewenst de gewone voorvoegsels kilo, mega, giga, tera, peta en exa te gebruiken voor machten van twee. Een megabyte is dus 1.000.000 bytes. Als men een hoeveelheid van 1.048.576 bytes wil benoemen heet dat nu een mebibyte. Zoals duidelijk te zien is in onderstaande tabel wordt het verschil tussen de SI-waarde en de reële waarde groter bij hogere machten.

SI				IEC				verschil
prefix	symbool			prefix	symbool			
-	-	10^0	1	-	-	2^0	1	0%
kilo	K	10^3	1.000	kibi	Ki	2^{10}	1024	2,4%
mega	M	10^6	1.000.000	mebi	Mi	2^{20}	1.048.576	4,9%
giga	G	10^9	1.000.000.000	gibi	Gi	2^{30}	1.073.741.824	7,4%
tera	T	10^{12}	1.000.000.000.000	tebi	Ti	2^{40}	1.099.511.627.776	9,9%
peta	P	10^{15}	1.000.000.000.000.000	pebi	Pi	2^{50}	1.125.899.906.842.624	12,6%
exa	E	10^{18}	1.000.000.000.000.000.000	exbi	Ei	2^{60}	1.152.921.504.606.846.976	15,3%
zetta	Z	10^{21}	1.000.000.000.000.000.000.000	zebi	Zi	2^{70}	1.180.591.620.717.411.303.424	18,1%
yotta	Y	10^{24}	1.000.000.000.000.000.000.000.000	yobi	Yi	2^{80}	1.208.925.819.614.629.174.706.176	20,9%

De voorvoegsels *zebi* en *yobi* zijn op dit moment grotendeels theoretisch, want gegevenshoeveelheden van deze grootteorde komen (nagenoeg) niet voor. De informatie die een yottabyte kan opslaan komt overeen met een stapel diskettes ter grootte van "2 keer de afstand van de aarde naar de zon".

Gevraagd wordt om een `bash` shell script `hrf` (staat voor *human-readable form*) te schrijven dat als filter kan gebruikt worden in pipes. Dit shell script moet gegevens inlezen van standaard input,

bepaalde numerieke waarden omzetten naar een voor de mens leesbaarder formaat en het resultaat uitschrijven naar standaard output. Het shell script moet hierbij volgende opties ondersteunen:

- Optie `-t` zorgt ervoor dat grondtal 1024 wordt gebruikt bij het vereenvoudigen van numerieke waarden, daar waar standaard grondtal 1000 wordt gebruikt indien deze optie niet gebruikt wordt. In beide gevallen worden gepaste SI of IEC symbolen gebruikt indien de oorspronkelijke numerieke waarde groter of gelijk is aan het grondtal.
- Optie `-f` heeft als verplicht argument een lijst van de velden die moeten aangepast worden, waarbij veldindices van elkaar worden gescheiden door een komma. Indien deze optie niet gebruikt wordt, dan worden de numerieke waarden in alle velden aangepast.
- Optie `-d` heeft als verplicht argument het te gebruiken veldscheidingsteken. Indien deze optie niet gebruikt wordt, dan wordt het standaard veldscheidingsteken van `awk` gebruikt om velden te scheiden.
- Optie `-s` heeft als argument een natuurlijk getal dat de eerste regel aangeeft waarop de aanpassingen worden doorgevoerd. Standaard gebeurt de verwerking vanaf de eerste regel indien deze optie niet gebruikt wordt. Deze optie laat dus toe om een aantal regels over te slaan bij de verwerking.

Zorg ervoor dat de commandolijn-argumenten door het `bash` shell script zo verwerkt worden dat bijvoorbeeld de volgende commandolijnen allemaal hetzelfde resultaat geven

```
$ cat test | hrf -t -f3,4 -d: -s3
$ cat test | hrf -d : -f 3,4 -t -s 3
$ cat test | hrf -tf3,4 -s 3 -d :
```

Zorg ook dat het shell script een gepaste foutmelding geeft indien niet-ondersteunde opties gebruikt worden, of indien verplichte argumenten ontbreken. De eigenlijke verwerking van de numerieke velden naar een voor de mens leesbaarder formaat moet gebeuren door een `awk` script dat volledig binnen het `bash` shell script zit ingebed. Daarbij moeten niet-numerieke velden steeds onaangeroerd blijven. Numerieke waarden kleiner dan tien moeten in het voor de mens leesbare formaat weergegeven worden met 1 cijfer na de komma, terwijl de weergave van grotere getallen geen cijfers na de komma moet bevatten. De functionaliteit van het `bash` shell script `hrf` wordt nog eens geïllustreerd in onderstaande sessie.

```
$ cat test
xxx:256:1048:xxx:xxx:18739:1234567:9876543210:12345678901234:98765432101234567:12345678909876543210:xxx
xxx:256:1048:xxx:xxx:18739:1234567:9876543210:12345678901234:98765432101234567:12345678909876543210:xxx
xxx:256:1048:xxx:xxx:18739:1234567:9876543210:12345678901234:98765432101234567:12345678909876543210:xxx
xxx:256:1048:xxx:xxx:18739:1234567:9876543210:12345678901234:98765432101234567:12345678909876543210:xxx
xxx:256:1048:xxx:xxx:18739:1234567:9876543210:12345678901234:98765432101234567:12345678909876543210:xxx
$ cat test | hrf -d:
xxx:256:1.0K:xxx:xxx:18K:1.2M:9.9G:12T:98P:12E:xxx
xxx:256:1.0K:xxx:xxx:18K:1.2M:9.9G:12T:98P:12E:xxx
xxx:256:1.0K:xxx:xxx:18K:1.2M:9.9G:12T:98P:12E:xxx
xxx:256:1.0K:xxx:xxx:18K:1.2M:9.9G:12T:98P:12E:xxx
xxx:256:1.0K:xxx:xxx:18K:1.2M:9.9G:12T:98P:12E:xxx
$ cat test | hrf -td:
xxx:256:1.0Ki:xxx:xxx:18Ki:1.2Mi:9.2Gi:11Ti:87Pi:10Ei:xxx
xxx:256:1.0Ki:xxx:xxx:18Ki:1.2Mi:9.2Gi:11Ti:87Pi:10Ei:xxx
xxx:256:1.0Ki:xxx:xxx:18Ki:1.2Mi:9.2Gi:11Ti:87Pi:10Ei:xxx
xxx:256:1.0Ki:xxx:xxx:18Ki:1.2Mi:9.2Gi:11Ti:87Pi:10Ei:xxx
xxx:256:1.0Ki:xxx:xxx:18Ki:1.2Mi:9.2Gi:11Ti:87Pi:10Ei:xxx
$ cat test | hrf -td: -s3 -f1,3,10
xxx:256:1048:xxx:xxx:18739:1234567:9876543210:12345678901234:98765432101234567:12345678909876543210:xxx
xxx:256:1048:xxx:xxx:18739:1234567:9876543210:12345678901234:98765432101234567:12345678909876543210:xxx
xxx:256:1.0Ki:xxx:xxx:18739:1234567:9876543210:12345678901234:87Pi:12345678909876543210:xxx
xxx:256:1.0Ki:xxx:xxx:18739:1234567:9876543210:12345678901234:87Pi:12345678909876543210:xxx
xxx:256:1.0Ki:xxx:xxx:18739:1234567:9876543210:12345678901234:87Pi:12345678909876543210:xxx
```

Opgave 4

In een *woordzoeker* zitten woorden verborgen die horizontaal en verticaal in beide richtingen kunnen gelezen worden. Gevraagd wordt om een subprocedure `maak_woordzoeker` te schrijven in VBA die op een nieuw werkblad een woordzoeker opbouwt op basis van een gegeven woordenlijst. De subprocedure moet de woordenlijst uitlezen uit de eerste kolom van het actieve werkblad in Excel, waarbij de woordenlijst loopt tot aan de eerste lege cel in de eerste kolom. De subprocedure moet het aantal rijen en het aantal kolommen van de woordzoeker aan de gebruiker opvragen via boodschapvensters (woordzoeker is rechthoekig, zodat aantal rijen niet noodzakelijk gelijk is aan aantal kolommen). Onderstaande figuur toont een voorbeeld van een 25×25 woordzoeker zoals die door de subprocedure `maak_woordzoeker` gegenereerd moet worden, vertrekkende van een lijst van alle chemische elementen.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	M	U	I	R	E	C	P	R	O	M	E	T	H	I	U	M	R	N	A	A	G	N	A	M	T
2	N	X	S	A	M	A	R	I	U	M	G	P	E	D	V	D	L	M	U	I	T	E	T	U	L
3	O	E	W	M	C	T	E	L	L	U	U	R	N	O	M	U	I	L	L	Y	R	E	B	I	A
4	G	N	J	O	D	I	U	M	U	I	N	O	L	O	P	T	A	N	I	T	A	L	P	M	B
5	R	O	O	L	H	C	M	U	I	N	O	T	U	L	P	U	B	Y	C	T	H	E	G	D	O
6	A	N	S	Y	M	L	L	R	N	E	H	A	F	N	I	U	M	S	N	E	E	S	R	A	K
7	D	K	T	B	U	M	M	O	O	R	H	C	A	L	C	I	U	M	I	R	L	X	E	C	O
8	I	R	I	D	I	U	M	X	B	Y	T	T	R	I	U	M	I	C	N	B	I	F	V	Y	O
9	U	Y	K	E	L	I	U	F	E	R	B	I	U	M	F	U	R	O	D	I	U	M	L	Y	L
10	M	P	S	E	L	D	I	O	L	K	Q	N	R	U	L	I	T	H	I	U	M	U	I	M	S
11	U	T	T	N	A	I	N	T	I	G	M	I	O	I	X	D	A	D	U	M	Z	I	Z	U	T
12	I	O	O	K	G	B	I	S	U	O	U	U	U	R	A	A	N	F	M	Z	I	N	K	I	O
13	V	N	F	P	Z	U	M	R	M	U	I	M	L	O	H	L	I	M	U	H	R	R	D	S	F
14	E	S	I	S	Z	R	U	E	U	D	D	A	F	H	R	L	Y	O	I	N	K	O	P	E	R
15	L	I	J	Z	E	R	L	T	I	T	A	A	N	T	R	A	D	O	N	E	O	F	M	C	O
16	E	L	A	N	T	H	A	N	A	N	R	L	N	G	P	A	R	I	P	N	I	U	U	F	
17	D	I	M	V	N	L	I	W	E	N	A	F	T	M	F	V	Z	B	E	T	I	L	I	M	S
18	N	C	E	F	E	B	P	B	H	T	V	L	K	U	O	H	W	L	T	U	U	A	M	U	O
19	E	I	R	O	O	B	S	I	T	A	F	O	N	I	T	I	A	U	S	N	M	C	R	I	F
20	M	U	I	D	N	A	C	S	U	A	D	W	N	S	S	Z	V	X	N	I	K	K	E	L	P
21	U	M	C	J	J	R	Q	M	R	L	U	L	B	E	R	K	E	L	I	U	M	J	F	L	L
22	K	W	I	K	K	I	C	U	R	I	U	M	U	N	U	B	L	F	E	M	U	I	L	A	K
23	Q	M	U	I	L	U	H	T	Q	B	B	S	T	G	U	E	U	R	O	P	I	U	M	H	W
24	J	B	M	U	I	M	S	O	T	A	A	T	S	A	Z	F	R	A	N	C	I	U	M	T	E
25	Q	H	P	R	A	S	E	O	D	Y	M	I	U	M	U	I	T	N	O	R	T	S	W	M	M

Om de woordzoeker zo compact mogelijk te houden, moet bij het opbouwen van de woordzoeker elk nieuw woord op een positie met *maximale overlap* geplaatst worden. Dit is een positie waarop het woord kan geplaatst worden zodat het een maximaal aantal letters gemeenschappelijk heeft met de woorden die reeds in de woordzoeker geplaatst zijn. Indien er meerdere mogelijke posities met maximale overlap zijn, dan mag één van deze posities willekeurig gekozen worden.

Stel bijvoorbeeld dat eerst het woord MANGAAN in de woordzoeker moet geplaatst worden. Dan zijn alle posities waarop het woord binnen de grenzen van het rooster kan geplaatst worden maximaal, aangezien elk van deze posities 0 letters gemeenschappelijk heeft met de woorden die reeds op het rooster staan (er zijn er namelijk nog geen). We kiezen een willekeurige positie waarin bijvoorbeeld het woord MANGAAN van links naar rechts wordt geschreven.

Stel nu dat als tweede woord het chemische element LUTETIUM in de woordzoeker moet geplaatst worden. Zoals aangegeven in onderstaande figuur zijn er drie mogelijke posities waarop het woord LUTETIUM kan geplaatst worden zodat het 1 letter gemeenschappelijk heeft met het woord MANGAAN. Alle andere posities zijn ofwel ongeldig (vallen buiten het rooster of hebben een andere letter op een plaats waar reeds een letter van het woord MANGAAN stond) of hebben geen enkele letter gemeenschappelijk met het woord MANGAAN. Om het woord LUTETIUM te plaatsen moet de subprocedure dus willekeurig één van de drie mogelijke posities kiezen, rekening houdend dat sommige van deze posities ook ongeldig kunnen zijn omdat het woord LUTETIUM dan buiten de grenzen van het rooster zou vallen.

(a) L
 U
 T
 E
 T
 I
 U
 M A N G A A N

(b) M A N G A A N
 U
 I
 T
 E
 T
 U
 L

(c) L U T E T I U M A N G A A N

Terwijl het zoekrooster gevuld raakt, kan het voorkomen dat voor sommige woorden geen enkele geldige positie binnen het zoekrooster kan gevonden worden. Indien dit het geval is moet de subprocedure finaal een lijst van niet geplaatste woorden weergeven in een boodschapvenster dat is opgemaakt zoals in onderstaand voorbeeld (let hierbij op de titel van het boodschapvenster en het gebruik van een informatie-icoontje).



Als alle woorden uit de woordenlijst al dan niet in het zoekrooster geplaatst zijn, dan moeten alle lege posities van het zoekrooster worden opgevuld met willekeurige letters. De subprocedure moet een nieuw werkblad aanmaken in Excel, waarop het zoekrooster kan uitgeschreven worden. Hierbij moet een opmaak gebruikt worden zoals aangegeven in het voorbeeldrooster met de chemische elementen hierboven. De nummering in de rij- en kolomhoofding moet hierbij in een lichte voorgrondkleur op een donkere achtergrond geplaatst worden. Alle ingevulde cellen moeten een passende grootte krijgen en moeten zowel horizontaal als verticaal in het midden uitgelijnd worden.

Opmerking: Deze opgave moet worden ingediend als het Excel bestand *opgave4.xls* (Excel 2003 formaat) met ingesloten macro's.