Bestandsuitwisseling en afwijkende karaktersets

Gebruik van speciale tekens in de database

Toon Loonen

Als een letter met een accent, bijvoorbeeld een é of een ë wordt ingetoetst op een invoerstation, bijvoorbeeld een VT220-terminal, dan kan men verwachten dat bij het opvragen op een ander station, bijvoorbeeld een PC of een printer, weer hetzelfde teken terug komt. Dit gebeurt helaas niet altijd. In dit artikel worden de eisen beschreven die nodig zijn om dit wel te realiseren. Verder komen enkele extra mogelijkheden aan bod, zoals het bij elkaar sorteren van de E, e en ë of zoeken zonder verschil te maken tussen hoofd- versus kleine letter en accenten. Het geheel is uitgevoerd met behulp van Sybase op een HP9000 onder HP-UX, maar elk volwassen RDBMS heeft deze mogelijkheden in de ene of andere vorm.

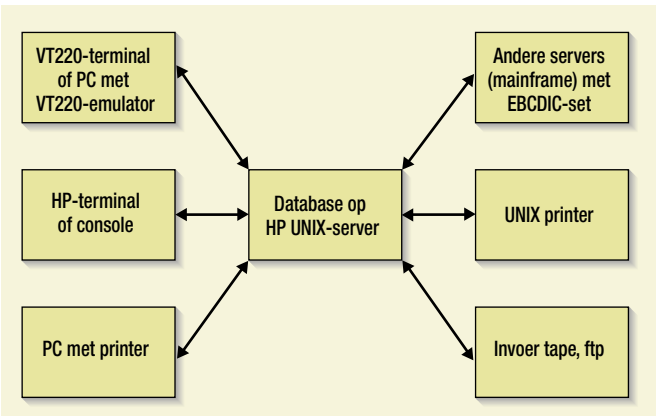
Bij diverse projecten bestaan problemen met het gebruik van diakritische tekens, vooral in een *mixed hardware*-omgeving. Denk hierbij aan de meeste client/server-systemen met bijvoorbeeld een UNIX-databaseserver, enkele VT220-terminals en verder PC-clients met VT220- of X-Windows-terminal-emulatie en 4GL (Uniface, Powerbuilder) toepassingen.

Deze systemen gebruiken allen een variant van de ASCII-karakterset. Hierin wordt elk teken voorgesteld door een combinatie van 8 bits en heeft dus 256 (= 2^8) mogelijke waarden.
De ASCII-set bevat op positie 0 t/m 31 stuurtekens (bijvoorbeeld 9 = TAB, 10 = linefeed, 12 = formfeed, 13 = Carriage Return, enzovoort) en op positie 32 t/m 127 vast gedefinieerde tekens (bijvoorbeeld 49 = "1", 65 = "A", 97 = "a").

Positie 128 en hoger zijn echter geheel verschillend gedefinieerd:

- ISO 8859/x Latin1 (UNIX) en CP1252 (MS Windows): een ë heeft ASCII-code 235;
- Roman8 (op HP3000 en HP9000, HP-UX): een ë heeft ASCII-code 205;
- Op de PC (CP850 of PC8-Code page 437) heeft een ë de ASCII-code 137.

Er zijn van deze sets verder nog nationale versies voor bijvoorbeeld Turks, Roemeens of Scandinavische talen.



Figuur 1: Voorbeeld van een mixed hardware-omgeving waarin het besproken probleem zich kan voordoen.

EEN CONVERSIE-VOORBEELD

Laten we het teken ë eens nader bekijken. Op een VT220-terminal wordt door middel van een programma (bijvoorbeeld een eenvoudige SQL-editor, text-editor voor bulk-import of een specifieke gebruikerstoepassing) een SQL-insertstatement samengesteld. Een van de woorden bevat een speciaal teken, bijvoorbeeld de genoemde ë. Dit teken heeft hier de ASCII-waarde 235. De databaseserver staat ook ingesteld op Latin1 en het teken kan zonder conversie worden opgeslagen in de database.

Een client/server-toepassing op een PC vraagt nu de gegevens op uit deze database. Op de PC heeft het teken ë de waarde 137. Er moet dus een conversie worden uitgevoerd op het teken. Bij het aanloggen op het RDBMS kan deze toepassing aangeven welke tekenset wordt gebruikt, bijvoorbeeld CP850. Het RDBMS zal nu alle tekens boven de 127 van de lokale (Latin1) set naar de door de client gewenste set (CP850) vertalen en daarna de gegevens naar de client versturen. Op de PC worden deze tekens dus weer correct getoond.

Op deze PC kan ook een VT220-emulator draaien, waarop een UNIX-programma (bijvoorbeeld een SQL-editor) is gestart. Nu is er geen vertaling nodig in het RDBMS. De VT220-emulator ontvangt tekens van de UNIX-server net als een echte VT220-terminal. Als hierin een ë voorkomt zal de terminal-emulator een ë

op het scherm tonen. Interpretatie en vertaling worden nu dus door de terminal-emulator verzorgd.

Dezelfde situatie, maar nu bekijken we een HP9000 met HP-UX als operating system. Deze gebruikt niet Latin1 maar de karakterset Roman8. Het RDBMS (Sybase) kan hier geïnstalleerd worden met de Roman8 (Default) karakterset of met de Latin1-set. Op een VT220-terminal wordt weer een SQL-insertstatement samengesteld dat een ë bevat. Dit teken heeft hier de ASCII-waarde 235.
De databaseserver staat ingesteld op Roman8 en het teken moet met conversie van 235 naar 205 in de database worden opgeslagen. Het programma kan hiervoor bij het aanloggen aangeven welke tekenset wordt gebruikt, in het onderhavige geval dus Latin1. Het RDBMS (in Roman8) weet nu dat elk teken boven de 127 van deze client vertaald moet worden van Latin1 naar Roman8.

Een ander client-programma op een PC geeft ook weer de gebruikte karakterset door bij het aanloggen, bijvoorbeeld CP850. Het RDBMS weet nu dat voor deze client de tekens van Roman8 naar CP850 vertaald moeten worden. Bij het ontvangen van gegevens van de PC moet de conversie natuurlijk de andere kant op worden uitgevoerd.

Om de tekens van de verschillende clients goed te kunnen vertalen moet het RDBMS dus kennis hebben van deze karaktersets. Alleen indien alle clients van één type zijn of tenminste allemaal dezelfde karakterset hebben, zou het mogelijk zijn om bij het opslaan in of ophalen uit de database geen vertaling uit te voeren. Bijvoorbeeld bij Windows-clients en een RDBMS op UNIX (Latin1) kan het teken ë zonder vertaling in de database worden opgesla-

Foutief eindexamen

Het meest aansprekende probleem met het foutief weergeven van accenten heeft zich onlangs voorgedaan bij het HAVO-eindexamen Frans. Daar zijn de tekens met accenten verkeerd afgedrukt. Dit is niet onderkend bij een controle en de examens zijn naar de scholen gestuurd. Nog net voor het examen is de fout ontdekt en zijn nieuwe examens gestuurd. Op negen scholen is echter nog de oude foutieve versie van het examen aan de leerlingen uitgedeeld. Iedereen die een verkeerde tekst heeft gekregen tijdens zijn examen, heeft het examen ongeldig kunnen laten verklaren om het later over te doen. Doordat het examen ongeldig is verklaart, heeft dit geen herkansing tot gevolg.

Hex	dec	0	1	2	3	4	5	6	7
		8	9	A	B	C	D	E	F
0	0		☺	●	♥	♦	♣	♠	•
	8	◘	○	■	♂	♀	♪	♫	☼
1	16	►	◄	↕	‼	¶	§	▬	↨
	24	↑	↓	→	←	∟	↔	▲	▼
2	32	spatie	!	"	#	$	%	&	'
	40	(	)	*	+	,	-	.	/
3	48	0	1	2	3	4	5	6	7
	56	8	9	:	;	<	=	>	?
4	64	@	A	B	C	D	E	F	G
	72	H	I	J	K	L	M	N	O
5	80	P	Q	R	S	T	U	V	W
	88	X	Y	Z	[	\	]	^	_
6	96	`	a	b	c	d	e	f	g
	104	h	i	j	k	l	m	n	o
7	112	p	q	r	s	t	u	v	w
	120	x	y	z	{	|	}	~	⌂
8	128	Ç	ü	é	â	ä	à	å	ç
	136	ê	ë	è	ï	î	ì	Ä	Å
9	144	É	æ	Æ	ô	ö	ò	û	ù
	152	ÿ	Ö	Ü	¢	£	¥	Pts	ƒ
A	160	á	í	ó	ú	ñ	Ñ	ª	º
	168	¿	⌐	¬	½	¼	¡	«	»
B	176	▒	▒	▒	|	┤	╣	╗	┐
	184	╕	╣	╗	╕	╜	╗	╝	┘
C	192	└	┴	┬	├	─	┼	╞	╔
	200	╚	╔	╩	╦	╠	═	╬	═
D	208	╨	╤	╥	╙	╘	╒	╓	╫
	216	╪	┘	┌	▒	▄	▌	▐	▀
E	224	α	ß	Γ	π	Σ	σ	µ	τ
	232	Φ	Θ	Ω	δ	∞	φ	ε	∩
F	240	≡	±	≥	≤	⌠	⌡		≈
	248	°	∙	·	√	ⁿ	²	■	

PC-karakterset PC8-Code Page 437.

HEX	DEC	0	1	2	3	4	5	6	7
		8	9	A	B	C	D	E	F
2	32		!	"	#	$	%	&	'
	40	(	)	*	+	,	-	.	/
3	48	0	1	2	3	4	5	6	7
	56	8	9	:	;	<	=	>	?
4	64	@	A	B	C	D	E	F	G
	72	H	I	J	K	L	M	N	O
5	80	P	Q	R	S	T	U	V	W
	88	X	Y	Z	[	\	]	^	_
6	96	‘	a	b	c	d	e	f	g
	104	h	i	j	k	l	m	n	o
7	112	p	q	r	s	t	u	v	w
	120	x	y	z	{	|	}	~	
8	128	€		‚	*f*	„	…	†	‡
	136	ˆ	‰	Š	‹	Œ		Ž	
9	144		‘	’	“	”	•	–	—
	152	˜	™	š	›	œ		ž	Ÿ
A	160		¡	¢	£	¤	¥	¦	§
	168	¨	©	ª	«	¬	-	®	¯
B	176	°	±	²	³	´	µ	¶	·
	184	¸	¹	º	»	¼	½	¾	¿
C	192	À	Á	Â	Ã	Ä	Å	Æ	Ç
	200	È	É	Ê	Ë	Ì	Í	Î	Ï
D	208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×
	216	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	224	à	á	â	ã	ä	å	æ	ç
	232	è	é	ê	ë	ì	í	î	ï
F	240	ð	ñ	ò	ó	ô	õ	ö	÷
	248	ø	ù	ú	û	ü	ý	þ	ÿ

CP1252. Deze set is gelijk aan ISO 8859/1 Latin1 behalve de tekens 0x80-0x9F, waar in Latin1 dezelfde stuurtekens staan als in 0x00-0x1F. Verder is in versie ISO 8859/15 het teken voor de waarde 164 vervangen door het Euro-teken €.

Afwijkende karaktersets

Naast de ASCII-karakterset hanteert IBM de (eveneens 8bit) EBCDIC-set op de IBM mainframe en AS400. Deze wijkt geheel af van de ASCII-set, ook voor de gewone letters en cijfers. Groot verschil is dat bij het sorteren in ASCII de cijfers voor de letters komen en in EBCDIC juist andersom. We gaan hier verder niet in detail op in, maar de algemene conversie-principes die hier vermeld zijn blijven ook bij deze conversies van toepassing.

Naast de 8bit-karaktersets kennen we UNICODE, een 16bits-karakterset, die alle bijzondere tekens bevat evenals de tekens van bijvoorbeeld het Griekse en Cyrillische alfabet. Alhoewel UNICODE conceptueel gezien de beste opslag is, is het voor de meeste toepassingen in Nederland niet nodig. Alleen als men alle namen van personen van bijvoorbeeld Zweedse, Turkse en Roemeense afkomst correct gespeld in de database wilt opnemen is UNICODE waarschijnlijk nodig. De databases hebben hiervoor het datatype *nchar* en *nvarchar* (naast *char* en *varchar*). Verdere details zijn te vinden op internet (http://www.unicode.org) en in de handboeken van de RDBMS leverancier.

gen. Immers, als het weer wordt opgevraagd op hetzelfde type station, zonder vertaling, wordt ook weer een ë getoond.

Een probleem bij het vertalen is dat een teken soms wel in de ene karakterset bestaat en niet in de andere. Bijvoorbeeld een š

UNICODE is waarschijnlijk alleen nodig bij Zweedse of Turkse namen

komt wel voor in Roman8 (waarde 235) maar niet in Latin1. Bij een insert, dus het vertalen van dit teken naar een Latin1-set, in de databaseserver zal een fout optreden. Bij een select, dus het vertalen van de server naar de clientset, zal dit teken door een vraagteken worden vervangen en een "waarschuwing" worden gegeven.

ZOEKEN EN SORTEREN

Om in de database een klant te vinden met de naam Jansen, is het vaak gewenst te zoeken op een wijze waarbij niet gelet wordt op hoofdletters, kleine letters of accenten. Hierbij wordt dus zowel Jansen, JANSEN, jansen als Jänsen gevonden. Ook bij het sorteren kan gevraagd worden om deze namen bij elkaar te sorteren. Dit is mogelijk door aan het RDBMS op te geven dat bij sorteringen en vergelijkingen alle letters onafhankelijk van accenten en hoofd- of kleine letter bij elkaar gesorteerd moeten worden.

Dit werkt dan voor de gesorteerde resultset maar ook bij zoeken met een where clause

```
where name = "Jansen"
```

zal ook JANSEN, jansen en Jänsen teruggeven. Verder zal een unieke index op naam de waarde "JANSEN" niet accepteren als er al een "Jänsen" in de database staat. Ook een tabelnaam "KLANT" is niet meer mogelijk als er al een tabel "klant" is.
Voor de vergelijkingen en het opbouwen van indexen moet het RDBMS natuurlijk weten welke ASCII-waarde een A, a of ä voorstelt. Vandaar dat het installeren van de goede karakterset en het correct converteren van tekens naar de karakterset van de server een vereiste is.

Wordt een bestaande database van de standaard binaire sortering omgezet naar de hierboven beschreven sortering, dan moeten ook alle indexen opnieuw worden opgebouwd. Hierbij kan het voorkomen dat er *duplicate keys* ontstaan voor waarden, die hiervoor wel uniek waren (JANSEN en Jansen). Men moet hiermee in de planning goed rekening houden en de inhoud van de database op dit soort mogelijke problemen controleren.

In Sybase zijn enige testen uitgevoerd om te zien of er gevolgen voor de performance waren vanwege deze afwijkende sortering, maar er is hierbij geen performanceteruggang van betekenis gevonden.

HET EURO-TEKEN €

Het Euro-teken € kan op het toetsenbord van de PC worden ingetikt door de combinatie Ctrl-Alt-5. In de nieuwste versies van de verschillende karaktersets heeft dit teken ook een eigen waarde gekregen:

MS Windows, CP1252	128
ISO 8859/15 Latin1 (dus versie 15 van deze set)	164
Roman8-1999 dus een subversie van de Roman8-set	186

De nieuwste versies van de verschillende RDBMS'en zullen deze sets ook ondersteunen. Voor oudere versies is waarschijnlijk een patch nodig. De werkwijze voor conversie van dit teken is verder geheel gelijk aan wat hiervoor is geschreven, voor een letter met accent, bijvoorbeeld de ë of ä. Let bij gebruik van het €-teken wel op of alle andere software en hardware (terminals, printers, labelprinters, barcode printers, plotters, enzovoort) dit teken, en ook de andere speciale tekens, ondersteunen.

BESTANDSUITWISSELING TUSSEN VERSCHILLENDE HARDWARE

Met FTP, tape of een messaging systeem (MQ) is het mogelijk om bestanden uit te wisselen tussen systemen met verschillende

karaktersets, bijvoorbeeld van de PC naar een UNIX-database-server of een internet-webserver. FTP heeft hiervoor twee modes: ASCII en binary.

In ASCII-mode worden eenvoudige tekstbestanden zonder opmaak (bijvoorbeeld programmacode, gegevensbestanden) overgezet. Hierbij vertaalt FTP de karakterset van de verzendende (bijvoorbeeld CP850) naar de ontvangende (Latin1) machine.

In binary mode worden bestanden binair overgezet, dus zonder vertaling van karakters of andere wijzigingen. Deze mode moet gebruikt worden voor niet-ASCII (niet tekst) bestanden, bijvoorbeeld executables of plaatjes (JPEG-bestanden) maar ook voor WORD- of PDF-bestanden.

Nog net voor het examen is de fout ontdekt en zijn nieuwe examens gestuurd

Een regel in een ASCII-bestand wordt op de PC afgesloten met twee tekens: <Carriage Return> en <Line Feet> (^M^J). Ook kan het bestand worden afgesloten met een Ctrl-Z-teken (^Z). Op een UNIX-machine ontbreekt het Ctrl-Z-teken aan het einde van het bestand en wordt een regel afgesloten met alleen een <Line Feet> (^J). Het FTP-programma moet de extra tekens bij het overhalen in ASCII-mode naar een UNIX-systeem verwijderen. Gebeurt dat niet en bekijken we het bestand op een VT220-terminal dan zullen we deze tekens mogelijk ook op het scherm zien.

Bijvoorbeeld: ^M
De tweede regel begint hier ^M
En de derde hier. ^M

Bij het overhalen van UNIX naar een PC moeten de extra tekens worden toegevoegd. Wordt dit vergeten en een tekst afgedrukt, dan komt deze mogelijk niet goed uit de printer vanwege het ontbreken van het <Carriage Return> teken. Elke volgende regel begint op de verticale positie waar de vorige geëindigd is:

Bijvoorbeeld:

De tweede regel begint hier

En de derde hier.

Dezelfde conversies zijn nodig, wanneer een bestand op een andere manier wordt overgezet, bijvoorbeeld met tape of een *message handler*.

Indien een UNIX-schijf als netwerkdrive voor een PC-netwerk wordt gebruikt en de bestanden in PC-formaat op deze schijf worden opgeslagen, dan kunnen bovenstaande problemen zich ook voordoen bij het bekijken van deze PC-bestanden vanuit UNIX.

Praktische oplossingen

Als een naam gezocht moet worden zonder op hoofd- of kleine letters te letten en de beschreven methode is niet mogelijk, dan kan er in de tabel een tweede kolom "naam" worden opgenomen met de naam in hoofdletters. Deze tweede (redundante) kolom wordt door de programmatuur bijgehouden. Bij het zoeken moet de door de gebruiker ingetikte naam ook eerst in hoofdletters worden omgezet waarna de query kan worden afgevuurd. Dit betekent meer werk, het kost meer opslag, het werkt dan nog maar voor één kolom en waarschijnlijk nog niet voor accenten. Daarom verdient de methode met de RDBMS-instellingen de voorkeur.

Een geheel andere methode om te zoeken op een naam "die lijkt op" ofwel "klinkt als", is de functie *soundex*. Hiervoor bestaan verschillende algoritmes, die eventueel zelfs afhankelijk van de gebruikte taal zijn. De Sybase-functie soundex("Jansen") zal een string teruggeven die bestaat uit letters en cijfers. Hierbij krijgen letters die op elkaar lijken dezelfde waarde en dubbele waardes worden verwijderd. Bij Sybase zal bijvoorbeeld soundex("Jansen") en soundex("Jenssen") dezelfde waarde teruggeven en bij het zoeken worden beide waardes teruggegeven. Zie voor meer informatie ook http://www.hpl.lib.tx.us/clayton/soundex.html.

CONCLUSIE

In een omgeving, waarin hardware wordt gebruikt waarop verschillende karaktersets in gebruik zijn, moet in de architectuur rekening gehouden worden met conversie van de karaktersets, wanneer gegevens naar een andere processor worden gestuurd, bijvoorbeeld voor opslag in een database of het overzetten van bestanden. Alleen dan zal een ingevoerde waarde op een ander onderdeel van deze omgeving weer correct worden getoond.

Verder kan men overwegen om voor zoeken en sorteren, de waarden met en zonder teken, en met hoofd- of kleine letter bij elkaar te groeperen. Hiervoor moet het RDBMS wel weten welke ASCII-waarde een bepaald teken voorstelt. ●

Toon Loonen (toon.loonen@cgey.nl, toon.loonen@inter.nl.net) is als consultant werkzaam bij Cap Gemini Ernst & Young.