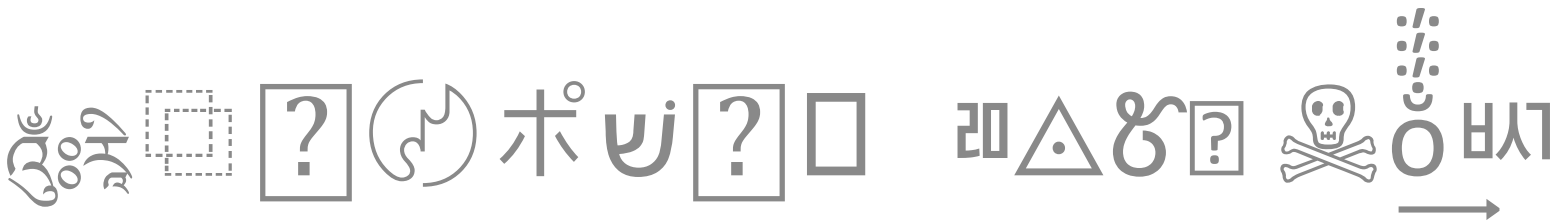
Lost In Translation:



Your host: **Solomon Rutzky**

Version: 2.0 (20180414)

Sql Quantum Lift

C:\> whoami

- Founder of Sql Quantum Lift:
  - **SQL#** (SQLsharp) : SQLCLR library of functions
  - OmniExec : Multi-threaded, multi-server & DB query tool
- Working in IT and with databases since 1996:
  - Variety of Roles, OSes, Languages, and DBs
  - SQL Server (since 2002), SQLCLR (since 2006)
- Areas of interest / concentration:
  - Collations & Encodings (https://Collations.Info/)
  - SQLCLR (https://SQLCLR.org/)
  - Module Signing (https://ModuleSigning.Info/)
- Articles:
  - Sql Quantum Leap (blog)
  - SQL Server Central (incl. Stairway to SQLCLR series)
  - Simple-Talk

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Agenda

- Character Sets and Encodings
- Collations
- Wrap-up / Q & A

- **Please see "Extra Notes" (Slide 27)**

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

CHARACTER SETS AND ENCODINGS

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

LCID / LANGID / Locale

- LCID =

```
+-------------+---------+-------------------------+
|   Reserved  | Sort ID |      Language ID        |
+-------------+---------+-------------------------+
31          20 19     16 15                       0   bit
```

- LANGID (i.e. "Language ID" from LCID) =

```
+--------------------------+--------------------------+
|  Secondary Language ID   |   Primary Language ID    |
+--------------------------+--------------------------+
15                       10  9                        0   bit
```

- LANGID = 0x0409 (Decimal / INT = 1033)
  - 0000010000001001
  - 000001
  - 0000001001 (8 + 1 = 9)

- Primary = English (en) 0x09
- Secondary = United States (US) 0x01

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

ASCII

- 7 bits
- 128 items defined
- Decimal: 0 – 127
- Hex: 0x00 – 0x7F

- ```
  Bit # -- Decimal Value / # of symbols
  0     -- 1
  1     -- 2
  2     -- 4
  3     -- 8
  4     -- 16
  5     -- 32
  6     -- 64
  7     -- 128   <-- 7 bit ASCII
  ```

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Image taken from http://ascii-table.com/

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/
  ```

Extended ASCII

- 8 bits
- 256 items defined
  - First 128 items are always the same
  - Second 128 items depend on Code Page
- All items:
  - Decimal: 0 – 255
  - Hex: 0x00 – 0xFF
- Variable items:
  - Decimal: 128 – 255
  - Hex: 0x80 – 0xFF

- ```
  Bit # -- Decimal Value / # of symbols
  8     -- 256   <-- 8 bit (i.e. 1 byte) Extended ASCII (code pages)
  ```

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/


  ```

Common Code Page

- Windows 1252 (CP1 in SQL Server)
- Latin1_General
- Mistakenly referred to as:
  - ISO-8859-1
  - ANSI

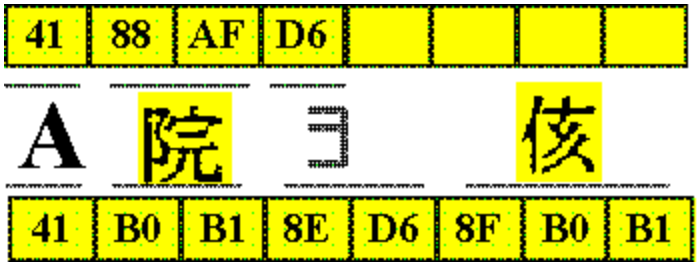
- Differ only in items 0x80 – 0x9F

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Double-Byte Character Set (DBCS)

These are single-byte (half-width) characters.
ｱｲｳｴｵｶｷｸｹｺｻｼｽｾｿ

Ｔｈｅｓｅ　ａｒｅ　ｄｏｕｂｌｅ－ｂｙｔｅ　ｃｈａｒａｃｔｅｒｓ．
アイウエオカキクケコサシスセソ
あいうえおかきくけこさしすせそ
一二三四五六七八区十



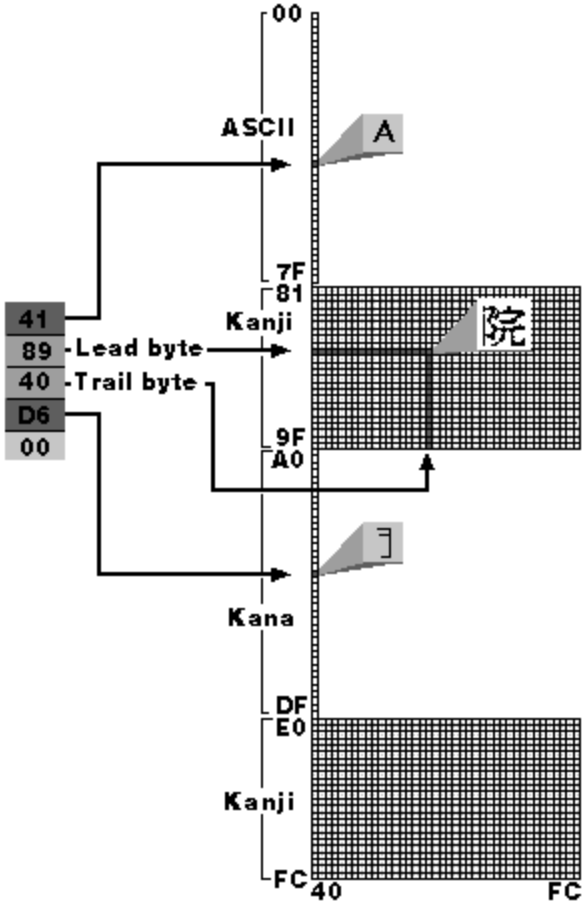
Shift-JIS
MS-DOS®, Windows

EUC (UJIS) UNIX

Images taken from https://msdn.microsoft.com/en-us/library/cc194788.aspx

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Double-Byte Character Set (DBCS)



[Lead Byte 87-88]

[Lead Byte 97-98]

Image taken from
https://msdn.microsoft.com/en-us/library/cc194788.aspx

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Code Page Conversions

```
SELECT ASCII('Ç' COLLATE Latin1_General_CI_AS) AS [CP1252 Value],
   'Ç' COLLATE SQL_Latin1_General_CP850_CI_AS AS
   [CharacterInCP850], ASCII('Ç' COLLATE
   SQL_Latin1_General_CP850_CI_AS) AS [CP850 Value];

CP1252 Value  CharacterInCP850  CP850 Value
199           Ç                 128
```

- First string is parsed using DB's default Collation
- Next, string might be converted again if used with column or COLLATE using different Code Page

- Character might exist with a different value in new Code Page
- Character might not exist, but might have an equivalent via "Best fit" mappings

- Which Collation is Used to Convert NVARCHAR to VARCHAR in a WHERE Condition?
  - (Part A of 2: "Duck")
  - (Part B of 2: "Rabbit")

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Unicode

- UCS-2
  - Always 2 bytes
  - No Supplementary Characters
  - Identical to first 65,536 Code Points of UTF-16 (i.e. the BMP)
- UTF-8
  - Variable length
  - 1 – 4 bytes
- UTF-16
  - 2 bytes
  - Can be used in dual-2 bytes sets for Supplementary Characters
- UTF-32
  - Always 4 bytes

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Unicode (cont.): Encodings

Figure 2-11. Unicode Encoding Forms

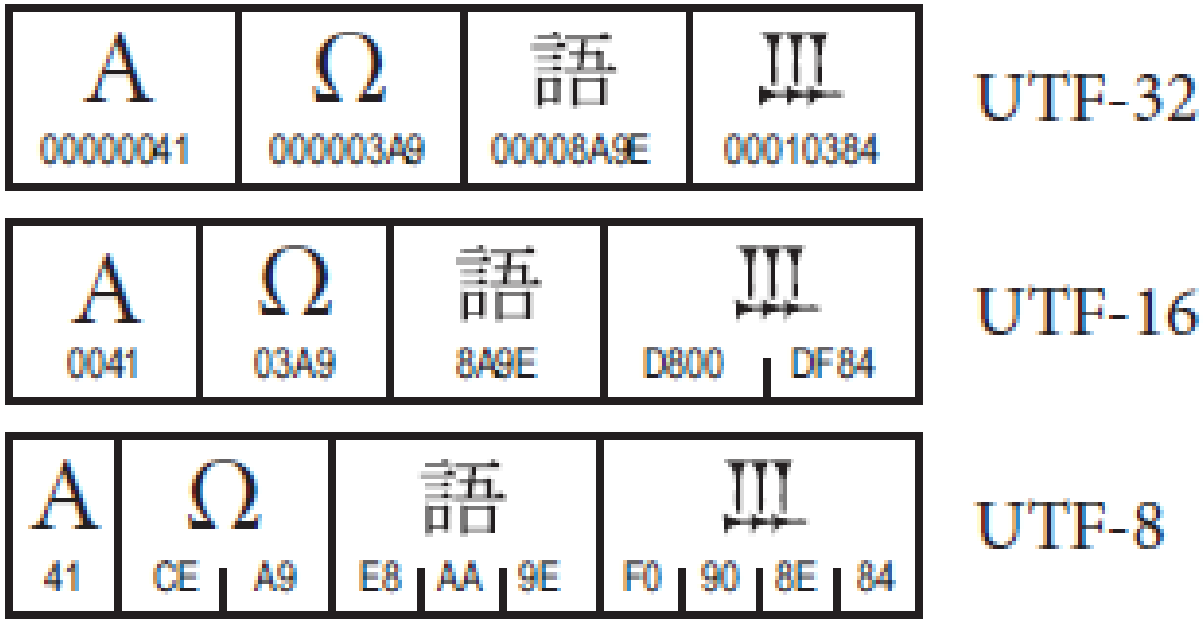


Image taken from https://www.unicode.org/versions/Unicode10.0.0/ch02.pdf

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

- Bit # -- Decimal Value / # of symbols
- 9 -- 512
  10 -- 1024
  11 -- 2048
  12 -- 4096
  13 -- 8192
  14 -- 16,384
  15 -- 32,768
  16 -- 65,536 <-- 16 bit limit

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Unicode (cont.): Endianess

Figure 2-12. Unicode Encoding Schemes

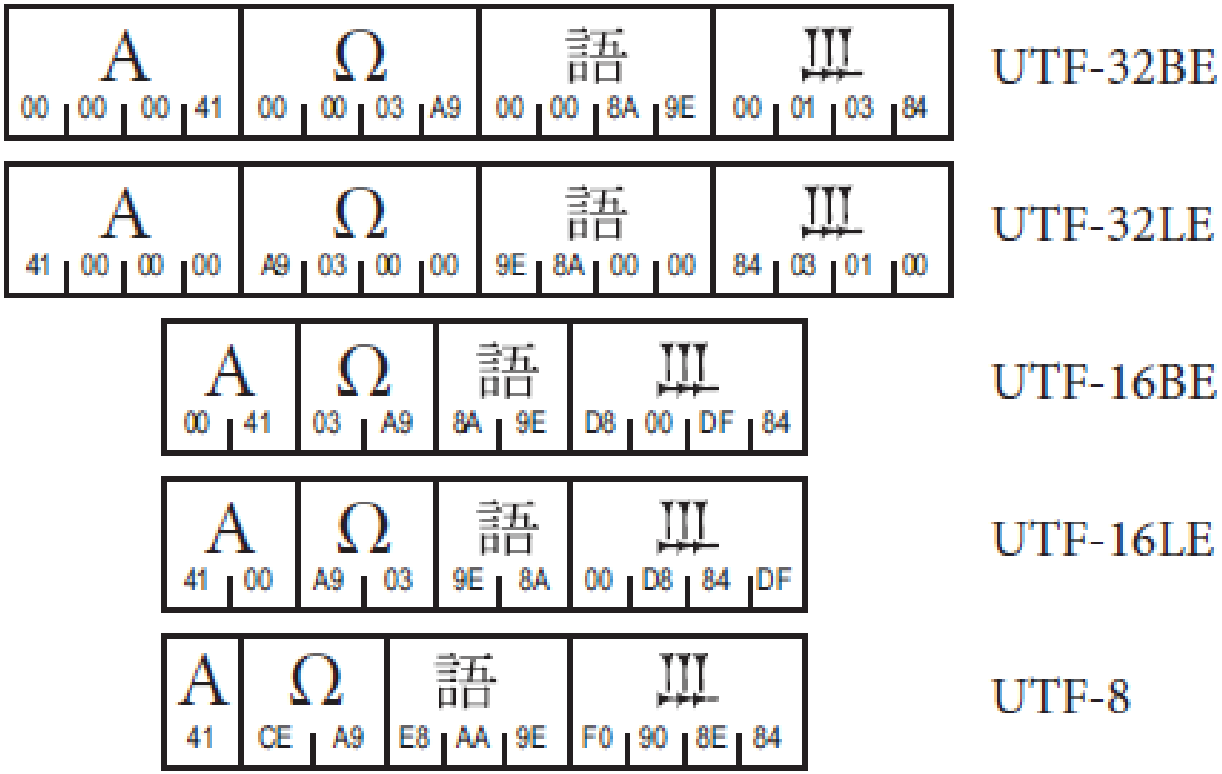


Image taken from https://www.unicode.org/versions/Unicode10.0.0/ch02.pdf

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

COLLATIONS

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Hierarchy: Instance-level

- Determines system DB Collation (master, model, msdb, tempdb**)
  - BUT, msdb (at least) can be restored from another server with a different Collation, resulting in F.U.N.!
  - ** tempdb due to being created from model
    - DATABASE_DEFAULT
    - CATALOG_DEFAULT
  - ** default collation for temporary tables (not table variable columns, except for their meta-data) without COLLATE clause
- Variable *names* (not their contents)
- Cursor names
- GOTO label names
- Default Collation for DBs created without COLLATE clause
- Instance-level meta-data (Login names, Database names, etc)
- "sysname" used to be affected by case-sensitive / binary collation (2005 = yes, 2008 R2 and newer = no, unsure about 2008)

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Hierarchy: Database-level

- String literals / constants
- Variables (their contents)
- Parameters
- OUTPUT parameters & return values
- Default Collation for new columns created (CREATE TABLE & ALTER TABLE ... ADD) without COLLATE clause
- Default Collation for table *variables* (but not their meta-data) without COLLATE clause
- Operation of CHAR() and NCHAR() functions
  - Code Page for CHAR(), and Supplementary Characters for NCHAR()
- Database-level meta-data (User names, Object names, etc)
- Contained DBs have exceptions

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Collation Precedence

- Coercible Default (weakest)
  - Literal
  - Variable
  - Non-string value converted to string
    - UNIQUEIDENTIFIER
    - DATETIME
- Implicit
  - Table Column / View Field
- Explicit (strongest)
  - COLLATE clause

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Sorting (Normalization)

- Unicode® Technical Standard #10: UNICODE COLLATION ALGORITHM (https://www.unicode.org/reports/tr10/)
- Unicode® Standard Annex #15: UNICODE NORMALIZATION FORMS (https://www.unicode.org/reports/tr15/)
  - NFD = Canonical Decomposition = "when correctly displayed should always have the same visual appearance and behavior" (step 1 of UCA)
  - NFKD = Compatibility Decomposition = "may have distinct visual appearances or behaviors"

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Sorting (Normalization - NFD)

Figure 1. Examples of Canonical Equivalence

Subtype	Examples
Combining sequence	Ç ⟷ C+◌̧
Ordering of combining marks	q+◌̇+◌̣ ⟷ q+◌̣+◌̇
Hangul & conjoining jamo	가 ⟷ ㄱ + ㅏ
Singleton equivalence	Ω ⟷ Ω

Image taken from https://www.unicode.org/reports/tr15/

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Sorting (Normalization - NFKD)

Subtype	Examples
Font variants	ℌ → H
	ℍ → H
Linebreaking differences	[NBSP] → [SPACE]
Positional variant forms	ε → ε
	ϵ → ε
	⋲ → ε
	⋹ → ε
Circled variants	① → 1

Width variants	ｶ → カ
Rotated variants	⁀ → {
	‿ → }
Superscripts/subscripts	i⁹ → i9
	i₉ → i9
Squared characters	ｱﾊﾟｰﾄ → アパート
Fractions	¼ → 1/4
Other	dž → dž

Images taken from https://www.unicode.org/reports/tr15/

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Unicode DUCET (Default Sort Weights)

```
1D5CF ; [.185B.0020.0005.1D5CF] # MATHEMATICAL SANS-SERIF SMALL V
1D603 ; [.185B.0020.0005.1D603] # MATHEMATICAL SANS-SERIF BOLD SMALL V
1D637 ; [.185B.0020.0005.1D637] # MATHEMATICAL SANS-SERIF ITALIC SMALL V
1D66B ; [.185B.0020.0005.1D66B] # MATHEMATICAL SANS-SERIF BOLD ITALIC SMALL V
1D69F ; [.185B.0020.0005.1D69F] # MATHEMATICAL MONOSPACE SMALL V
24E5  ; [.185B.0020.0006.24E5] # CIRCLED LATIN SMALL LETTER V
0056  ; [.185B.0020.0008.0056] # LATIN CAPITAL LETTER V
...
1F77  ; [.1932.0020.0002.03B9][.0000.0024.0002.0301] # GREEK SMALL LETTER IOTA WITH OXIA
038A  ; [.1932.0020.0008.0399][.0000.0024.0002.0301] # GREEK CAPITAL LETTER IOTA WITH TONOS
1FDB  ; [.1932.0020.0008.0399][.0000.0024.0002.0301] # GREEK CAPITAL LETTER IOTA WITH OXIA
1F76  ; [.1932.0020.0002.03B9][.0000.0025.0002.0300] # GREEK SMALL LETTER IOTA WITH VARIA
1FDA  ; [.1932.0020.0008.0399][.0000.0025.0002.0300] # GREEK CAPITAL LETTER IOTA WITH VARIA
1FD0  ; [.1932.0020.0002.03B9][.0000.0026.0002.0306] # GREEK SMALL LETTER IOTA WITH VRACHY
1FD8  ; [.1932.0020.0008.0399][.0000.0026.0002.0306] # GREEK CAPITAL LETTER IOTA WITH VRACHY
1FD6  ; [.1932.0020.0002.03B9][.0000.002A.0002.0342] # GREEK SMALL LETTER IOTA WITH PERISPOMENI
03CA  ; [.1932.0020.0002.03B9][.0000.002B.0002.0308] # GREEK SMALL LETTER IOTA WITH DIALYTIKA
```

File fragment taken from http://www.unicode.org/Public/UCA/latest/allkeys.txt

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Sorting (Multi-level)

- Each step applied to the entire string, *not* character by character:
  - Standard: sort base characters (regardless of accent and case differences)
  - IF Accent-sensitive, apply accent / diacritic weights
  - IF Case-sensitive, apply casing weights

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Sorting (Multi-level 2)

Figure 3. Comparison of Sort Keys

	String	Sort Key
1	cab	0706 06D9 06EE 0000 0020 0020 0020 0000 0002 0002 0002
2	Cab	0706 06D9 06EE 0000 0020 0020 0020 0000 0008 0002 0002
3	cáb	0706 06D9 06EE 0000 0020 0020 0021 0020 0000 0002 0002 0002 0002
4	dab	0712 06D9 06EE 0000 0020 0020 0020 0000 0002 0002 0002

In *Figure 3*, "cab" $<_3$ "Cab" $<_2$ "cáb" $<_1$ "dab". The differences that produce the ordering are shown by the **bold underlined** items:

- For strings 1 and 2, the first difference is in 0002 versus 0008 (Level 3).
- For strings 2 and 3, the first difference is in 0020 versus 0021 (Level 2).
- For strings 3 and 4, the first difference is in 0706 versus 0712 (Level 1).

Image taken from https://www.unicode.org/reports/tr10/

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Recommendations

- Use Windows Collation over SQL_ collations
  - Latin1_General* over SQL_Latin1_General*
- Use highest level Collation:
  - 140 over 100 over 90 over unspecified (i.e. 80)
- Use BIN2 instead of BIN as it sorts properly

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Extra Notes

- Slide 5 (LCID / Locale):
  - COLLATIONPROPERTY (https://docs.microsoft.com/en-us/sql/t-sql/functions/collation-functions-collationproperty-transact-sql)
  - LCID Structure (https://msdn.microsoft.com/en-us/library/cc233968.aspx)
  - Primary Language IDs (https://msdn.microsoft.com/en-us/library/cc195085.aspx)
- Slide 11 (Code Page Conversions):
  - Can't update “CO2” to “CO_2” in table row (https://dba.stackexchange.com/a/191619/30859)
  - Collate issues with wrong characters (https://dba.stackexchange.com/a/197315/30859)
- Side 12 (Unicode):
  - UTF-8, UTF-16, UTF-32 & BOM (https://www.unicode.org/faq/utf_bom.html)
- Slide 23 (Default Sort Weights)
  - Unicode® Technical Standard #10: UNICODE COLLATION ALGORITHM (https://www.unicode.org/reports/tr10/)
- Slide 25 (Multi-level Sorting 2)
  - ICU Collation Demo (http://demo.icu-project.org/icu-bin/collation.html)

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/

Hiding in Plain Sight

- Collation Resources:
  - https://Collations.Info/
- Articles:
  - https://www.SqlServerCentral.com/author/solomon-rutzky
  - https://www.SqlServerCentral.com/stairways/stairway-to-sqlclr (Stairway to SQLCLR)
  - https://www.simple-talk.com/author/solomon-rutzky/
- SQLsharp.com
  - https://SQLsharp.com/
- StackOverflow.com & DBA.StackExchange.com
  - https://StackExchange.com/users/281451/solomon-rutzky
- LinkedIn
  - http://www.LinkedIn.com/in/srutzky/
- Email:
  - SRutzky@SqlQuantumLift.com

Solomon Rutzky
Email: SRutzky@SqlQuantumLift.com
Company: https://SqlQuantumLift.com
Blog: https://SqlQuantumLeap.com
Twitter: @SqlQuantumLeap
https://Collations.Info/