Citus Con: An Event for Postgres

Hosted by Microsoft

Localization on PostgreSQL

Keisuke Takahashi

Cloud Solution Architect (Data & Analytics),
Microsoft

#CITUSCON | April 12-13, 2022

Agenda

- **Introduction** (5min)
- **A number of different character sets** (5min)
  - Character Set Support
- **Locale-specific collation order** (15min)
  - Locale Support
  - Collation Support

Agenda

- **Introduction**
- A number of different character sets
  - Character Set Support
- Locale-specific collation order
  - Locale Support
  - Collation Support


WILLKOMMEN
欢迎
स्वागत
BIENVENIDA
WELCOME
BIENVENUE
ようこそ
환영하다
добро пожаловать
ترحيب
BEM-VINDO
歡迎
Chào mừng

Who I am

- Keisuke Takahashi
- Cloud Solution Architect (Data&AI) at Microsoft Japan
- Open Source Software enthusiast
- English as a Second Language
- Learned Chinese for two years
- Have some Korean friends

This presentation is for whom ...

- Already knows what character encoding is.
- Uses PostgreSQL for data that may include non-ASCII characters.
- Be interested in both PostgreSQL and foreign languages.

Today's Goal

- Learn the basics of Character Sets, Locale and Collation in PostgreSQL
- Learn which locale / collation you should choose

Azure Database for PostgreSQL deployment options

Single Server

Fully-managed, single-node PostgreSQL database service with built-in HA

Example use cases

- Transactional and operational analytics workloads
- Apps requiring JSON, geospatial support, or full-text search
- Cloud-native apps built with modern frameworks

Flexible Server NEW

Maximum control for your database with a simplified developer experience

Example use cases

- Support for a variety of workloads with a new simplified architecture
- High-performance apps utilizing zone co-location for low latency

Hyperscale (Citus)

Worry-free PostgreSQL in the cloud with an architecture built to scale out

Example use cases

- Scaling PostgreSQL multi-tenant, SaaS applications
- Real-time operational analytics
- Building high throughput transactional apps

The benefits of Azure Database for PostgreSQL

Innovate with open-source tools and extensions

Stay productive with full compatibility with community PostgreSQL and support for your favorite PostgreSQL extensions.

Maximize performance with a fully managed Azure service

Focus on your application innovation, not database management. Enjoy AI-powered performance optimization and advanced security.

Ultimate control and flexibility for databases

Enjoy maximum control and flexibility with custom maintenance windows, zone redundant high availability and additional configuration parameters for fine grained database tuning.

Build massively scalable PostgreSQL applications

Scale with ease to hundreds of nodes, with no app rewrites. Save time by running transactions and analytics in one database and avoid the costs of manual sharding.

Single Server

Flexible Server NEW

Hyperscale (Citus)

Agenda

- Introduction
- **A number of different character sets**
  - Character Set Support
- Locale-specific collation order
  - Locale Support
  - Collation Support

Default settings on Azure PostgreSQL

```
SELECT name, setting FROM pg_settings WHERE name IN
('server_encoding', 'dynamic_shared_memory_type')
```

Single Server

```
postgres=> SELECT name, setting FROM pg_settings WHERE name IN ('server_encoding', 'dynamic_shared_memory_type');
            name             | setting
-----------------------------+---------
 dynamic_shared_memory_type | windows
 server_encoding            | UTF8
```

Flexible Server

```
postgres=> SELECT name, setting FROM pg_settings WHERE name IN ('server_encoding', 'dynamic_shared_memory_type');
            name             | setting
-----------------------------+---------
 dynamic_shared_memory_type | posix
 server_encoding            | UTF8
```

Hyperscale (Citus)

```
citus=> SELECT name, setting FROM pg_settings WHERE name IN ('server_encoding', 'dynamic_shared_memory_type');
            name             | setting
-----------------------------+---------
 dynamic_shared_memory_type | posix
 server_encoding            | UTF8
```

Default settings on Azure PostgreSQL

```
SELECT datname, datcollate, datctype FROM pg_database
WHERE datname IN ('template1','template0')
```

Single Server

```
postgres=> SELECT datname, datcollate, datctype FROM pg_database WHERE datname IN ('template1','template0');
  datname  |         datcollate         |          datctype
-----------+----------------------------+----------------------------
 template1 | English_United States.1252 | English_United States.1252
 template0 | English_United States.1252 | English_United States.1252
```

Flexible Server

```
postgres=> SELECT datname, datcollate, datctype FROM pg_database WHERE datname IN ('template1','template0');
  datname  | datcollate |  datctype
-----------+------------+------------
 template1 | en_US.utf8 | en_US.utf8
 template0 | en_US.utf8 | en_US.utf8
```

Hyperscale (Citus)

```
citus=> SELECT datname, datcollate, datctype FROM pg_database WHERE datname IN ('template1','template0');
  datname  | datcollate  |  datctype
-----------+-------------+-------------
 template1 | en_US.UTF-8 | en_US.UTF-8
 template0 | en_US.UTF-8 | en_US.UTF-8
```

Supported Character Sets (Server-side) #1-10

Name	Language	ICU?	Bytes/Char
`UTF8`	*all*	Yes	1–4
`SQL_ASCII`	*any*	No	1
`WIN1256`	Arabic	Yes	1
`LATIN7`	Baltic	Yes	1
`WIN1257`	Baltic	Yes	1
`LATIN8`	Celtic	Yes	1
`LATIN2`	Central European	Yes	1
`WIN1250`	Central European	Yes	1
`WIN866`	Cyrillic	Yes	1
`WIN1251`	Cyrillic	Yes	1

Supported Character Sets (Server-side) #11-20

Name	Language	ICU?	Bytes/Char
KOI8R	Cyrillic (Russian)	Yes	1
KOI8U	Cyrillic (Ukrainian)	Yes	1
WIN1253	Greek	Yes	1
WIN1255	Hebrew	Yes	1
EUC_JP	Japanese	Yes	1–3
EUC_JIS_2004	Japanese	No	1–3
EUC_KR	Korean	Yes	1–3
ISO_8859_6	Latin/Arabic	Yes	1
ISO_8859_5	Latin/Cyrillic	Yes	1
ISO_8859_7	Latin/Greek	Yes	1

Supported Character Sets (Server-side) #21-30

Name	Language	ICU?	Bytes/Char
ISO_8859_8	Latin/Hebrew	Yes	1
LATIN9	LATIN1 with Euro and accents	Yes	1
MULE_INTERNAL	Multilingual Emacs	No	1–4
LATIN6	Nordic	Yes	1
LATIN4	North European	Yes	1
LATIN10	Romanian	No	1
EUC_CN	Simplified Chinese	Yes	1–3
LATIN3	South European	Yes	1
WIN874	Thai	No	1
EUC_TW	Traditional Chinese, Taiwanese	Yes	1–3

Supported Character Sets (Server-side) #31-35

Name	Language	ICU?	Bytes/Char
LATIN5	Turkish	Yes	1
WIN1254	Turkish	Yes	1
WIN1258	Vietnamese	Yes	1
LATIN1	Western European	Yes	1
WIN1252	Western European	Yes	1

Setting the Character Set

* This operation is not available on Azure PostgreSQL Hyperscale(Citus) because Citus is a DB level extension.

```
CREATE DATABASE new_database TEMPLATE template0 ENCODING
'encoding' LC_CTYPE 'collation.encoding'
```

Example:

```
CREATE DATABASE my_db TEMPLATE template0 ENCODING 'UTF-8'
LC_CTYPE 'ja_JP.UTF-8';
```

Result:

```
postgres=> SELECT datname, datcollate, datctype FROM pg_database WHERE datname = 'my_db';
 datname | datcollate |  datctype
---------+------------+------------
 my_db   | en_US.utf8 | ja_JP.UTF-8
```

Setting the Character Set

* This operation is not available on Azure PostgreSQL Hyperscale(Citus) because Citus is a DB level extension.

```
CREATE DATABASE new_database TEMPLATE template0 ENCODING
'encoding' LC_CTYPE 'collation.encoding'
```

Example:

```
CREATE DATABASE my_db TEMPLATE template0 ENCODING 'UTF-8'
LC_CTYPE 'ja_JP.UTF-8';
```

Alternative:

```
$ createdb my_db --locale=japanese --template=template0
$ initdb --encoding=UTF=8
```

Agenda

- Introduction
- A number of different character sets
  - Character Set Support
- **Locale-specific collation order**
  - Locale Support
  - Collation Support

How to check your locale settings

```
SELECT name, setting, context FROM pg_settings WHERE name
LIKE 'lc%'
```

Result on Azure Database for PostgreSQL Hyperscale (Citus):

```
citus=> SELECT name, setting, context FROM pg_settings WHERE name LIKE 'lc%';
    name      |   setting   |  context
--------------+-------------+-----------
 lc_collate   | en_US.UTF-8 | internal
 lc_ctype     | en_US.UTF-8 | internal
 lc_messages  | en_US.utf8  | superuser
 lc_monetary  | en_US.utf8  | user
 lc_numeric   | en_US.utf8  | user
 lc_time      | en_US.utf8  | user
```

Parameters

Name	Description	Note
LC_COLLATE	String sort order	
LC_CTYPE	Character classification (What is a letter? Its upper-case equivalent?)	
LC_MESSAGES	Language of messages	
LC_MONETARY	Formatting of currency amounts	This can also be configured with Azure Portal
LC_NUMERIC	Formatting of numbers	This can also be configured with Azure Portal
LC_TIME	Formatting of dates and times	

Default settings on Azure PostgreSQL

Single Server

```
postgres=> SELECT name, setting, context FROM pg_settings WHERE name LIKE 'lc%';
    name     |          setting           |  context
-------------+----------------------------+-----------
 lc_collate  | English_United States.1252 | internal
 lc_ctype    | English_United States.1252 | internal
 lc_messages | English_United States.1252 | superuser
 lc_monetary | English_United States.1252 | user
 lc_numeric  | English_United States.1252 | user
 lc_time     | English_United States.1252 | user
```

Flexible Server

```
postgres=> SELECT name, setting, context FROM pg_settings WHERE name LIKE 'lc%';
    name     |   setting   |  context
-------------+-------------+-----------
 lc_collate  | en_US.utf8  | internal
 lc_ctype    | en_US.utf8  | internal
 lc_messages | en_US.utf8  | superuser
 lc_monetary | en_US.utf-8 | user
 lc_numeric  | en_US.utf-8 | user
 lc_time     | en_US.utf8  | user
```

Hyperscale (Citus)

```
citus=> SELECT name, setting, context FROM pg_settings WHERE name LIKE 'lc%';
    name     |   setting   |  context
-------------+-------------+-----------
 lc_collate  | en_US.UTF-8 | internal
 lc_ctype    | en_US.UTF-8 | internal
 lc_messages | en_US.utf8  | superuser
 lc_monetary | en_US.utf8  | user
 lc_numeric  | en_US.utf8  | user
 lc_time     | en_US.utf8  | user
```

How to set your locale

* This operation is not available on Azure PostgreSQL Hyperscale(Citus) because Citus is a DB level extension.

```
CREATE DATABASE new_database TEMPLATE template0 ENCODING
'encoding' LC_COLLATE 'collation.encoding' LC_CTYPE
'collation.encoding'
```

Example:

```
CREATE DATABASE my_db TEMPLATE template0 ENCODING 'UTF-8'
LC_COLLATE 'ja_JP.UTF-8' LC_CTYPE 'ja_JP.UTF-8';
```

Result:

```
postgres=> SELECT datname, datcollate, datctype FROM pg_database WHERE datname = 'my_db';
 datname | datcollate  |  datctype
---------+-------------+-------------
 my_db   | ja_JP.UTF-8 | ja_JP.UTF-8
```

How to set your locale

* This operation is not available on Azure PostgreSQL Hyperscale(Citus) because Citus is a DB level extension.

```
CREATE DATABASE new_database TEMPLATE template0 ENCODING
'encoding' LC_COLLATE 'collation.encoding' LC_CTYPE
'collation.encoding'
```

Example:

```
CREATE DATABASE my_db TEMPLATE template0 ENCODING 'UTF-8'
LC_COLLATE 'ja_JP.UTF-8' LC_CTYPE 'ja_JP.UTF-8';
```

Alternative:

```
$ createdb ja --locale=japanese --template=template0
$ initdb --encoding=UTF=8 --locale=ja_JP.UTF=8
```

How to get Predefined Collations

```
SELECT * FROM pg_collation
```

Example:

```
SELECT collname, collprovider FROM pg_collation WHERE
collname LIKE 'ja%';
```

Alternative:

```
\dOS+
```

Predefined "ja" Collations on Azure PostgreSQL

```
SELECT collname, collprovider FROM pg_collation WHERE
collname LIKE 'ja%';
```

Single Server

```
postgres=> SELECT collname, collprovider FR
OM pg_collation WHERE collname LIKE 'ja%';
  collname  | collprovider
------------+-------------
 ja         | c
 ja.utf8    | c
 ja_JP      | c
 ja_JP.utf8 | c
(4 rows)
```

Flexible Server

```
postgres=> SELECT collname, collprovider FR
OM pg_collation WHERE collname LIKE 'ja%';
  collname   | collprovider
-------------+-------------
 ja_JP.eucjp | c
 ja_JP.utf8  | c
 ja_JP       | c
 ja_JP       | c
 ja-x-icu    | i
 ja-JP-x-icu | i
(6 rows)
```

Hyperscale (Citus)

```
citus=> SELECT collname, collprovider FROM
pg_collation WHERE collname LIKE 'ja%';
   collname   | collprovider
--------------+-------------
 ja_JP        | c
 ja_JP.eucjp  | c
 ja_JP.ujis   | c
 ja_JP.utf8   | c
 japanese     | c
 japanese.euc | c
 ja_JP        | c
 ja-x-icu     | i
 ja-JP-x-icu  | i
(9 rows)
```

Predefined "zh" Collations on Azure PostgreSQL

Single Server

```
postgres=> SELECT collname, collprovider FROM pg_collation WHERE collname LIKE 'zh%';
    collname     | collprovider
-----------------+--------------
 zh              | c
 zh.utf8         | c
 zh_CHS          | c
 zh_CHS.utf8     | c
 zh_CHT          | c
 zh_CHT.utf8     | c
 zh_CN           | c
 zh_CN.utf8      | c
 zh_HK           | c
 zh_HK.utf8      | c
 zh_Hans         | c
 zh_Hans.utf8    | c
 zh_Hans_HK      | c
 zh_Hans_HK.utf8 | c
 zh_Hans_MO      | c
 zh_Hans_MO.utf8 | c
 zh_Hant         | c
 zh_Hant.utf8    | c
 zh_MO           | c
 zh_MO.utf8      | c
 zh_SG           | c
 zh_SG.utf8      | c
 zh_TW           | c
 zh_TW.utf8      | c
(24 rows)
```

Flexible Server

```
postgres=> SELECT collname, collprovider FROM pg_collation WHERE collname LIKE 'zh%';
     collname     | collprovider
------------------+--------------
 zh_CN            | c
 zh_CN.utf8       | c
 zh_HK.utf8       | c
 zh_SG            | c
 zh_SG.utf8       | c
 zh_TW.euctw      | c
 zh_TW.utf8       | c
 zh_CN            | c
 zh_HK            | c
 zh_SG            | c
 zh_TW            | c
 zh_TW            | c
 zh-x-icu         | i
 zh-Hans-x-icu    | i
 zh-Hans-CN-x-icu | i
 zh-Hans-HK-x-icu | i
 zh-Hans-MO-x-icu | i
 zh-Hans-SG-x-icu | i
 zh-Hant-x-icu    | i
 zh-Hant-HK-x-icu | i
 zh-Hant-MO-x-icu | i
 zh-Hant-TW-x-icu | i
(22 rows)
```

Hyperscale (Citus)

```
citus=> SELECT collname, collprovider FROM pg_collation WHERE collname LIKE 'zh%';
     collname     | collprovider
------------------+--------------
 zh_CN            | c
 zh_CN.gb2312     | c
 zh_CN.utf8       | c
 zh_HK.utf8       | c
 zh_SG            | c
 zh_SG.gb2312     | c
 zh_SG.utf8       | c
 zh_TW.euctw      | c
 zh_TW.utf8       | c
 zh_CN            | c
 zh_HK            | c
 zh_SG            | c
 zh_TW            | c
 zh_TW            | c
 zh-x-icu         | i
 zh-Hans-x-icu    | i
 zh-Hans-CN-x-icu | i
 zh-Hans-HK-x-icu | i
 zh-Hans-MO-x-icu | i
 zh-Hans-SG-x-icu | i
 zh-Hant-x-icu    | i
 zh-Hant-HK-x-icu | i
 zh-Hant-MO-x-icu | i
 zh-Hant-TW-x-icu | i
(24 rows)
```

libc vs. ICU

- **libc**
  - Old-school locale/collation provider of PostgreSQL
- **ICU**
  - International Components for Unicode
  - Have been supported since PostgreSQL 10
  - User-defined locale/collation support

Create ICU Database

* This operation is not available on Azure PostgreSQL Hyperscale(Citus) because Citus is a DB level extension.

```
CREATE DATABASE new_database ENCODING 'UTF-8' TEMPLATE
template0 LC_COLLATE 'collation' LC_CTYPE 'collation'
```

Example:

```
CREATE DATABASE my_icu_db ENCODING 'UTF-8' LC_COLLATE
'ja-x-icu' LC_CTYPE 'ja-x-icu';
```

Create ICU Template DB

* This operation is not available on Azure PostgreSQL Hyperscale(Citus) because Citus is a DB level extension.

```
CREATE DATABASE new_template ENCODING 'UTF-8' TEMPLATE
template0 LC_COLLATE 'collation' LC_CTYPE 'collation'
```

```
CREATE DATABASE new_database TEMPLATE new_template
```

Example:

```
CREATE DATABASE template_ja_x_icu ENCODING 'UTF-8'
TEMPLATE template0 LC_COLLATE 'ja-x-icu' LC_CTYPE 'ja-x-
icu';
CREATE DATABASE my_db TEMPLATE template_ja_x_icu;
```

Set Collation for the Column

```
CREATE TABLE tablename (
    colname1 type COLLATE "collation",
    colname2 type COLLATE "collation",
    ...
)
```

Example:

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) COLLATE "ja-x-icu");
```

Specify Collation on SELECT

```
SELECT * FROM table ORDER BY column COLLATE "collation"
```

Example:

```
SELECT * FROM mytable ORDER BY mycol COLLATE "ja-x-icu";
```

Sorting Experiments

- **Environment**
  - Azure Database for PostgreSQL Hyperscale (Citus)
    - PostgreSQL 14
    - Citus 10.2
    - Basic tier
    - West US 2

Sorting Experiments

- **10 collations**
  - English
    - en_US.utf8, en-US-x-icu
  - Japanese
    - ja_JP.utf8, ja-x-icu, ja-JP-x-icu
  - Chinese
    - zh_CN.utf8, zh-x-icu, zh-Hans-x-icu, zh-Hant-x-icu

Sorting Experiments

- **165 unicode characters / combinations**
  - Latin
  - Japanese (Hiragana, Katakana, Kanji, Simbol, etc.)
  - Chinese (Traditional, Simplified)
  - Hangul
  - Chữ Nôm
  - Arabic
  - Cyrillic

Tested Characters

Character	UTF-8 (hex)	Description
a	0x61	Latin Small Letter a
z	0x7A	Latin Small Letter z
A	0x41	Latin Capital Letter A
Z	0x5A	Latin Capital Letter Z
ａ	0xEF 0xBD 0x81	Fullwidth Latin Capital Letter a
ｚ	0xEF 0xBD 0x9A	Fullwidth Latin Capital Letter z
Ａ	0xEF 0xBC 0xA1	Fullwidth Latin Capital Letter A
Ｚ	0xEF 0xBC 0xBA	Fullwidth Latin Capital Letter Z

Tested Characters

Character	UTF-8 (hex)	Description
0	0x30	ASCII Digit Zero
9	0x39	ASCII Digit Nine
０	0xEF 0xBC 0x90	Fullwidth Digit Zero
９	0xEF 0xBC 0x99	Fullwidth Digit Nine
⓪	0xE2 0x93 0xAA	Circled Digit Zero
㊿	0xE3 0x8A 0xBF	Circled Number Fifty
🈐	0xE3 0x89 0x8C	Circled Number Fifty on Black Square

Tested Characters

Character	UTF-8 (hex)	Description
①	0xE2 0x91 0xA0	Circled Digit One
❶	0xE2 0x9D 0xB6	Dingbat Negative Circled Digit One
Ⅰ	0xE2 0x85 0xA0	Roman Numeral One
ⅰ	0xE2 0x85 0xB0	Small Roman Numeral One
⑴	0xE2 0x91 0xB4	Parenthesized Digit One
(1)	0x28 0x31 0x29	Digit One in Parenthesisses
一	0xE4 0xB8 0x80	CJK Ideograph, First
壱	0xE5 0xA3 0xB1	Number One

Tested Characters

Character	UTF-8 (hex)	Description
⑨	0xE2 0x91 0xA8	Circled Digit Nine
❾	0xE2 0x9D 0xBE	Dingbat Negative Circled Digit Nine
Ⅸ	0xE2 0x85 0xA8	Roman Numeral Nine
ⅸ	0xE2 0x85 0xB8	Small Roman Numeral Nine
⑼	0xE2 0x91 0xBC	Parenthesized Digit Nine
(9)	0x28 0x39 0x29	Digit Nine in Parenthesisses
九	0xE4 0xB9 0x9D	Nine
9.	0xE2 0x92 0x90	Digit Nine Full Stop
9.	0x91 0x2E	Digit Nine and Full Stop

Tested Characters

Character	UTF-8 (hex)	Description
Ⅶ	0xE2 0x85 0xA6	Roman Numeral Seven
ⅶ	0xE2 0x85 0xB6	Small Roman Numeral Seven
VII	0x56 0x49 0x49	Latin Capital Letter V, I, I
vii	0x76 0x69 0x69	Latin Small Letter v, i, i
Ⅻ	0xE2 0x85 0xAB	Roman Numeral Twelve
ⅻ	0xE2 0x85 0xBB	Small Roman Numeral Twelve
XII	0x58 0x49 0x49	Latin Capital Letter X, I, I
xii	0x78 0x69 0x69	Latin Small Letter x, i, i

Tested Characters

Character	UTF-8 (hex)	Description
α	0xCE 0xB1	Greek Small Letter Alpha
ω	0xCF 0x89	Greek Small Letter Omega
A	0xCE 0x91	Greek Capital Letter Alpha
Ω	0xCE 0xA9	Greek Capital Letter Omega

Tested Characters

Character	UTF-8 (hex)	Description
(a)	0x28 0x61 0x29	"a" in parenthesisses
(z)	0x28 0x7A 0x29	"z" in parenthesisses
{a}	0x7B 0x61 0x7D	"a" in Curly Brackets
[a]	0x5B 0x61 0x5D	"a" in Square Brackets
（a）	0xEF 0xBC 0x88 0x61 0xEF 0xBC 0x89	"a" in Fullwidth parenthesisses
（z）	0xEF 0xBC 0x88 0x7A 0xEF 0xBC 0x89	"z" in Fullwidth parenthesisses

Tested Characters

Character	UTF-8 (hex)	Description
｛a｝	0xEF 0xBD 0x9B 0x61 0xEF 0xBD 0x9D	"a" in Fullwidth Curly Brackets
［a］	0xEF 0xBC 0xBB 0x61 0xEF 0xBC 0xBD	"a" in Fullwidth Square Brackets
【a】	0xE3 0x80 0x90 0x61 0xE3 0x80 0x91	"a" in Black Lenticular Brackets

Tested Characters

Character	UTF-8 (hex)	Description
a	0x20 0x61	"a" after a space
a	0xE3 0x80 0x80 0x61	"a" after an Ideographic space
_a	0x5F 0x61	"a" after an underscore
__a	0xEF 0xBC 0xBF 0x61	"a" after a Fullwidth Low Line
/a	0x2F 0x61	"a" after a Solidus
／a	0xEF 0xBC 0x8F 0x61	"a" after a Fullwidth Solidus

Tested Characters

Character	UTF-8 (hex)	Description
あ	0xE3 0x81 0x82	Hiragana Letter A
ア	0xE3 0x82 0xA2	Katakana Letter A
ｧ	0xEF 0xBD 0xB1	Halfwidth Katakana Letter A
ん	0xE3 0x82 0x93	Hiragana Letter N
ン	0xE3 0x83 0xB3	Katakana Letter N
ﾝ	0xEF 0xBE 0x9D	Halfwidth Katakana Letter N

Tested Characters

Character	UTF-8 (hex)	Description
は	0xE3 0x81 0xAF	Hiragana Letter Ha
ば	0xE3 0x81 0xB0	Hiragana Letter Ba
ば	0xE3 0x81 0xB0 0xE3 0x82 0x9B	Hiragana Letter Ha and Voiced Sound Mark
ぱ	0xE3 0x81 0xB1	Hiragana Letter Pa
ぱ	0xE3 0x81 0xB0 0xE3 0x82 0x9C	Hiragana Letter Ha and Semi-voiced Sound Mark

Tested Characters

Character	UTF-8 (hex)	Description
ﾉﾊ	0xE3 0x83 0x8F	Katakana Letter Ha
ﾉﾊﾞ	0xE3 0x83 0x90	Katakana Letter Ba
ﾉﾊﾞ	0xE3 0x83 0x8F 0xE3 0x82 0x9B	Katakana Letter Ha and Voiced Sound Mark
ﾉﾊﾟ	0xE3 0x83 0x91	Katakana Letter Pa
ﾊ	0xEF 0xBE 0x8A	Halfwidth Katakana Letter Ha
ﾊﾞ	0xEF 0xBE 0x8A 0xEF 0xBE 0x9E	Halfwidth Katakana Letter Ha and Halfwidth Katakana Voiced Sound Mark
ﾊﾟ	0xEF 0xBE 0x8A 0xEF 0xBE 0x9F	Halfwidth Katakana Letter Ha and Halfwidth Katakana Semi-voiced Sound Mark

Tested Characters

Character	UTF-8 (hex)	Description
あつ	0xE3 0x81 0x82 0xE3 0x81 0xA4	Hiragana Letter A and Tu
あっ	0xE3 0x81 0x82 0xE3 0x81 0xA3	Hiragana Letter A and Small Tu
ああ	0xE3 0x81 0x82 0xE3 0x81 0x82	Hiragana Letter A and A
あぁ	0xE3 0x81 0x82 0xE3 0x81 0x81	Hiragana Letter A and Small A
あゝ	0xE3 0x81 0x82 0xE3 0x82 0x9D	Hiragana Letter A and Hiragana Iteration Mark

Tested Characters

Character	UTF-8 (hex)	Description
あー	0xE3 0x81 0x82 0xE3 0x83 0xBC	Hiragana Letter A and Prolonged Sound Mark
あ-	0xE3 0x81 0x82 0x2D	Hiragana Letter A and Hyphen/Minus
あ〜	0xE3 0x81 0x82 0xE3 0x80 0x9C	Hiragana Letter A and Wave Dash (macOS)
あ～	0xE3 0x81 0x82 0xEF 0xBD 0x9E	Hiragana Letter A and Fullwidth Tilde (Windows)
あ~	0xE3 0x81 0x82 0x7E	Hiragana Letter A and Tilde

Tested Characters

Character	UTF-8 (hex)	Description
あ…	0xE3 0x81 0x82 0xE2 0x80 0xA6	Hiragana Letter A and Horizontal Ellipsis
あ゙	0xE3 0x81 0x82 0xE3 0x82 0x9B	Hiragana Letter A and Voiced Sound Mark

Tested Characters

Character	UTF-8 (hex)	Description
キロ	0xE3 0x82 0xAD 0xE3 0x83 0xAD	Katakana Letter Ki and Ro
ｷﾛ	0xEF 0xBD 0xB7 0xEF 0xBE 0x9B	Halfwidth Katakana Letter Ki and Ro
㌔	0xE3 0x8C 0x94	Square Kiro
km	0x6B 0x6D	Latin Small Letter k and m
ｋｍ	0xEF 0xBD 0x8B 0xEF 0xBD 0x8D	Fullwidth Latin Small Letter k and m
㎞	0xE3 0x8E 0x9E	Squre km

Tested Characters

Character	UTF-8 (hex)	Description
キロメートル	0xE3 0x8C 0x96	Squre Kiromeetoru
粁	0xE7 0xB2 0x81	Kilometre (Japanese)
cm	0x63 0x6D	Latin Small Letter c and m
ｃ ｍ	0xEF 0xBD 0x83 0xEF 0xBD 0x8D	Fullwidth Latin Small Letter c and m
㎝	0xE3 0x8E 0x9D	Squre cm
センチ	0xE3 0x8C 0xA2	Squre Senti
糎	0xE7 0xB3 0x8E	Centimetre (Japanese)

Tested Characters

Character	UTF-8 (hex)	Description
kg	0x6B 0x67	Latin Small Letter k and g
ｋ ｇ	0xEF 0xBD 0x8B 0xEF 0xBD 0x87	Fullwidth Latin Small Letter k and g
㎏	0xE3 0x8E 0x8F	Squre kg
㌔	0xE3 0x8C 0x95	Squre Kiroguramu
瓩	0xE7 0x93 0xA9	Kilogram (Japanese)

Tested Characters

Character	UTF-8 (hex)	Description
(株)	0x28 0xE6 0xA0 0xAA 0x29	numerary adjunct for trees; root, in Parenthesisses
（株）	0xEF 0xBC 0x88 0xE6 0xA0 0xAA 0xEF 0xBC 0x89	numerary adjunct for trees; root, in Fullwidth parenthesisses
㈱	0xE3 0x88 0xB1	Parenthesized Ideograph Stock

Tested Characters

Character	UTF-8 (hex)	Description
畑	0xE7 0x95 0x91	dry (as opposed to rice) field; used in Japanese names
畠	0xE7 0x95 0xA0	garden, field, farm, plantation
働	0xE5 0x83 0x8D	labor; work
匂	0xE5 0x8C 0x82	fragrance, smell
♡	0xE2 0x99 0xA1	White Heart Suit
💓	N/A	

Tested Characters

Character	UTF-8 (hex)	Description
!	0x21	Exclamation Mark
！	0xEF 0xBC 0x81	Fullwidth Exclamation Mark
@	0x40	Commercial At
＠	0xEF 0xBC 0xA0	Fullwidth Commercial At
#	0x23	Number Sign
＃	0xEF 0xBC 0x83	Fullwidth Number Sign
$	0x24	Doller Sign
＄	0xEF 0xBC 0x84	Fullwidth Doller Sign
?	0x3F	Question Mark

Tested Characters

Character	UTF-8 (hex)	Description
？	0xEF 0xBC 0x9F	Fullwidth Question Mark
,	0x2C	Comma
，	0xEF 0xBC 0x8C	Fullwidth Comma
.	0x2E	Full Stop
．	0xEF 0xBC 0x8E	Fullwidth Full Stop
、	0xE3 0x80 0x81	Ideographic Comma
。	0xE3 0x80 0x82	Ideographic Full Stop
､	0xEF 0xBD 0xA4	Halfwidth Ideographic Comma
｡	0xEF 0xBD 0xA1	Halfwidth Ideographic Full Stop

Tested Characters

Character	UTF-8 (hex)	Description
高	0xE9 0xAB 0x98	high, tall; lotfy elevated
髙	0xE9 0xAB 0x99	Variant of 高 U+9ADB, high, tall; lotfy elevated
変	0xE5 0xA4 0x89	change, transform, alter (Japanese)
變	0xE8 0xAE 0x8A	change, transform, alter (Traditional Chinese)
变	0xE5 0x8F 0x98	change, transform, alter (Simplified Chinese)
総	0xE7 0xB7 0x8F	collect; overall, altogether (Japanese)
總	0xE7 0xB8 0xBD	collect; overall, altogether (Traditional Chinese)
总	0xE6 0x80 0xBB	collect; overall, altogether (Simplified Chinese)

Tested Characters

Character	UTF-8 (hex)	Description
가	0xEA 0xB0 0x80	Hangul ga
히	0xED 0x9E 0x88	Hangul hi
까	0xEA 0xB9 0x8C	Hangul kka
찌	0xEC 0xB0 0x8C	Hangul jji
𡨸	N/A	Vietnamese Chữ Nôm
喃	0xE5 0x96 0x83	Vietnamese Chữ Nôm
𣍐	N/A	Vietnamese Chữ Nôm
𦓡	N/A	Vietnamese Chữ Nôm

Tested Characters

Character	UTF-8 (hex)	Description
È	0xD0 0x80	Cyrillic Capital Letter Ie with Grave
Ë	0xD0 0x81	Cyrillic Capital Letter Io
Ⱨ	0xD3 0xBE	Cyrillic Capital Letter Ha with Stroke
ⱨ	0xD3 0xBF	Cyrillic Small Letter Ha with Stroke
Ä	0xC3 0x84	Latin Capital Letter A with Diaresis
ä	0xC3 0xA4	Latin Small Letter a with Diaresis
Ü	0xC3 0x9C	Latin Capital Letter U with Diaresis
ü	0xC3 0xBC	Latin Small Letter u with Diaresis

Tested Characters

Character	UTF-8 (hex)	Description
¡	0xC2 0xA1	Inverted Exclamation Mark
¿	0xC2 0xBF	Inverted Question Mark
ٹ	0xD9 0xB9	Arabic Letter Tteh
ٹ	0xD9 0xBA	Arabic Letter Tteheh
ے	0xDB 0x92	Arabic Letter Barree
ۓ	0xDB 0x93	Arabic Letter Barree with Hamza Above

Setup (on Citus Database)

```
CREATE TABLE t_en_US_utf8 (t text COLLATE "en_US.utf8");
CREATE TABLE t_en_US_x_icu (t text COLLATE "en-US-x-icu");
CREATE TABLE t_ja_JP_utf8 (t text COLLATE "ja_JP.utf8");
CREATE TABLE t_ja_x_icu (t text COLLATE "ja-x-icu");
CREATE TABLE t_ja_JP_x_icu (t text COLLATE "ja-JP-x-icu");
CREATE TABLE t_zh_CN_utf8 (t text COLLATE "zh_CN.utf8");
CREATE TABLE t_zh_x_icu (t text COLLATE "zh-x-icu");
CREATE TABLE t_zh_Hans_x_icu (t text COLLATE "zh-Hans-x-
icu");
CREATE TABLE t_zh_Hant_x_icu (t text COLLATE "zh-Hant-x-
icu");
```

SELECT t FROM t_en_US_utf8 ORDER BY t;

(en_US.utf8)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
,	#	ⅰ	9.	は	ン	kg	#	@	ﾊ	あ…
!	✖	ⅶ	⓪	ば	㈱	cm	$	A	ﾝ	あ～
¡	✖	ⅸ	♡	ぱ	🆒	km	,	Z	あ～	あぁ
?	ﾞ	ⅻ	❶	ん	㊿	가	.	a	あ-	ああ
¿	Ⅰ	①	❾	ア	ｷﾛ	까	0	z	💗	あっ
.	Ⅶ	⑨	、	ハ	ｷﾛｸﾞﾗﾑ	찌	1	。	宁字	あつ
@	Ⅸ	⑴	。	バ	ｷﾛﾒｰﾄﾙ	히	9	、	離	あゝ
$	Ⅻ	⑼	あ	パ	ｾﾝﾁ	!	?	ｱ	鞘	あー

SELECT t FROM t_en_US_utf8 ORDER BY t;

(en_US.utf8)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
あ～	ｋ ｍ	9.	a	ä	Ⅶ	ۓ	一	总	総	
あ゛	ｷﾛ	a	／a	Ä	xii	ے	九	(株)	總	
は゛	ﾊ゛	a	__a	cm	XII	A	働	（株）	變	
は゜	ﾊ゜	_a	【a】	kg	z	α	匂	砥	高	
キロ	0	/a	(a)	km	(z)	Ω	変	畑	髙	
ハ゛	1	(a)	[a]	ü	（z）	ω	喃	畠		
ｃ ｍ	9	[a]	{a}	Ü	Z	È	壱	粁		
ｋ ｇ	(9)	{a}	A	vii	ٹ	Ë	変	糎		

SELECT t FROM t_en_US_x_icu ORDER BY t;

(en-US-x-icu)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
a	!	。	(z)	【a】	💗	①	9.	cm	k g	vii
a	!	。	(株)	@	$	❶	9.	c m	kg	VII
_a	¡	(1)	（株）	@	$	🆖	a	cm	km	Vll
__a	?	(9)	㈱	/a	0	㊿	a	i	k m	xii
,	?	⑼	[a]	／a	0	9	A	I	km	xii
,	¿	(a)	［a］	#	⓪	9	A	ix	ü	XII
、	.	(a)	{a}	#	1	⑨	ä	IX	Ü	Xll
、	.	(z)	{a}	♡	1	❾	Ä	kg	vii	z

SELECT t FROM t_en_US_x_icu ORDER BY t;

(en-US-x-icu)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
z	Ë	까	あ～	あっ	は	パ	九	砥	高	
Z	⁎	찌	あ…	あつ	ハ	ﾊﾟ	働	畑	髙	
Z	⁑	히	あ~	キロ	ﾊ	ばﾞ	匂	畠	孱	
α	ٹ	あ	あ～	ｷﾛ	ば	ハ゛	変	粁	離	
A	ٹ	ア	あゝ	ｷﾛ	バ	ん	喃	糎	鞄	
ω	ے	ｱ	あー	ｷﾛｸﾞﾗﾑ	ﾊﾞ	ン	壱	総		
Ω	ۓ	あ゛	あぁ	ｷﾛﾒｰﾄﾙ	ぱ	ﾝ	変	總		
È	가	あ-	ああ	ｾﾝﾁ	はﾟ	―	总	變		

SELECT t FROM t_ja_JP_utf8 ORDER BY t;

(ja_JP.utf8)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
È	Ⅶ	⑨	㈱	cm	ㅎ|	$	0	Ⅻ	vii	ハ
✖	Ⅸ	(1)	50	km	💗	(9)	1	Z	xii	ハ゛
✖	Ⅻ	(9)	㊿	変	孖	(a)	9	[a]	z	ハ゜
ٹ	ⅰ	9.	キロ	总	離	(z)	9.	_a	{a}	ン
ٹ	vii	⓪	キログラム	高	齟	(株)	?	a	。	a
ے	ix	♡	キロメートル	가	a	,	@	cm	、	、
ے	xii	❶	センチ	까	!	.	A	kg	ｱ	。
ا	①	❾	kg	찌	#	/a	Ⅶ	km	ｷﾛ	,

SELECT t FROM t_ja_JP_utf8 ORDER BY t;

(ja_JP.utf8)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
.	［a］	9	あ	あぁ	ば	パ	壱	畑	¿	
?	{a}	A	あ-	ああ	ぱ	ン	粁	畠	Ä	
!	【a】	Z	あ～	あっ	ん	A	九	変	Ü	
__a	$	a	あ゛	あつ	ア	Ω	高	喃	ä	
／a	#	c m	あゝ	あ～	キロ	α	糎	變	ü	
（a）	@	k g	あー	は	ハ	ω	総	砥		
（z）	0	k m	あ～	は゜	ハ゛	Ë	働	總		
（株）	1	z	あ…	ば゛	バ	—	匂	¡		

SELECT t FROM t_ja_x_icu ORDER BY t;

(ja-x-icu)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
a	!	。	(z)	【a】	💗	①	9.	cm	k g	vii
a	!	。	(株)	@	$	❶	9.	c m	kg	VII
_a	¡	(1)	（株）	@	$	🆙	a	cm	km	Ⅶ
__a	?	(9)	㈱	/a	0	㊿	a	i	k m	xii
,	?	⑼	[a]	／a	0	9	A	I	㎞	xii
,	¿	(a)	［a］	#	⓪	9	A	ix	ü	XII
、	.	(a)	{a}	#	1	⑨	ä	IX	Ü	Ⅻ
、	.	(z)	{a}	♡	1	❾	Ä	kg	vii	z

SELECT t FROM t_ja_x_icu ORDER BY t;

(ja-x-icu)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
z	Ë	까	あ～	あっ	は	はﾟ	壱	畑	总	
Z	⁎	찌	あ…	あつ	ハ	パ	粁	畠	髙	
Z	⁑	히	あ~	キロ	ﾊ	はﾞ	九	変	孖	
α	ـت	あ	あ～	ｷﾛ	ば	ハﾞ	高	喃	靡	
A	ـث	ア	あー	ｷﾛ	バ	ん	糎	變	鞦	
ω	ـے	ｱ	あぁ	キログラム	ﾊﾞ	ン	総	砥		
Ω	ـۓ	あ゛	あゝ	キロメートル	ぱ	ﾝ	働	總		
È	가	あ-	ああ	センチ	パ	―	匂	変		

SELECT t FROM t_ja_JP_x_icu ORDER BY t;

(ja-JP-x-icu)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
a	!	。	(z)	【a】	💗	①	9.	cm	k g	vii
a	!	。	(株)	@	$	❶	9.	c m	kg	VII
_a	¡	(1)	（株）	@	$	🆖	a	cm	km	Ⅶ
__a	?	(9)	㈱	/a	0	㊿	a	i	k m	xii
,	?	⑼	[a]	／a	0	9	A	I	㎞	xii
,	¿	(a)	[a]	#	⓪	9	A	ix	ü	XII
、	.	(a)	{a}	#	1	⑨	ä	IX	Ü	Ⅻ
、	.	(z)	{a}	♡	1	❾	Ä	kg	vii	z

(The same as ja-x-icu)

SELECT t FROM t_ja_JP_x_icu ORDER BY t;

(ja-JP-x-icu)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
z	Ë	까	あ〜	あっ	は	は゜	壱	畑	总	
Z	⁎	찌	あ…	あつ	ハ	パ゜	粁	畠	髙	
Z	⁑	히	あ~	キロ	ﾊ	ば゛	九	変	孖	
α	ٹ	あ	あ〜	ｷﾛ	ば	バ゛	高	喃	離	
A	ٹ	ア	あー	㌔	バ	ん	糎	變	鄗	
ω	ے	ｱ	あぁ	キログラム	ﾊﾞ	ン	総	砥		
Ω	ۓ	あ゛	あゝ	キロメートル	ぱ	ﾝ	働	總		
È	가	あ-	ああ	センチ	パ	—	匂	変		

(The same as ja-x-icu)

SELECT t FROM t_zh_CN_utf8 ORDER BY t;

(zh_CN.utf8)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
,	#	ⅰ	9.	は	ン	kg	ㅎ|	9	、	離
!	✖	ⅶ	⓪	ば	㈱	cm	！	？	ァ	鼬
¡	⁎	ⅸ	♡	ぱ	🈐	km	#	@	ハ	あ…
?	؞	ⅻ	❶	ん	㊿	匂	$	A	ヽ	あ～
¿	｜	①	❾	ア	ｷﾛ	畠	,	Z	あ~	あぁ
.	Ⅶ	⑨	、	ハ	ｷﾛｸﾞﾗﾑ	가	.	a	あ-	ああ
@	Ⅸ	⑴	。	バ	ｷﾛﾒｰﾄﾙ	까	0	z	💗	あっ
$	Ⅻ	⑼	あ	パ	ｾﾝﾁ	찌	1	。	字	あつ

(Similar to en_US.utf8)

SELECT t FROM t_zh_CN_utf8 ORDER BY t;

(zh_CN.utf8)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
あゝ	c m	9	[a]	{a}	Ü	Z	È	髙	壱	
あー	k g	(9)	{a}	A	vii	ٹ	Ë	九	(株)	
あ～	k m	9.	a	ä	VII	ۓ	变	糎	（株）	
あ゛	ｷﾛ	a	／a	Ä	xii	ے	変	喃	总	
は゛	ﾊ゛	a	__a	cm	XII	A	變	砥	總	
は゜	ﾊ゜	_a	【a】	kg	z	α	総	粁		
キロ	0	/a	(a)	km	(z)	Ω	働	畑		
ハ゛	1	(a)	[a]	ü	(z)	ω	高	一		

(Similar to en_US.utf8)

SELECT t FROM t_zh_x_icu ORDER BY t;

(zh-x-icu)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
a	!	。	(z)	【a】	💗	①	9.	cm	k g	vii
a	!	。	㈱	@	$	❶	9.	c m	kg	VII
_a	¡	(1)	(株)	@	$	🆙	a	cm	km	Ⅶ
__a	?	(9)	（株）	/a	0	㊿	a	i	k m	xii
,	?	⑼	[a]	／a	0	9	A	I	㎞	xii
,	¿	(a)	[a]	#	⓪	9	A	ix	ü	XII
、	.	(a)	{a}	#	1	⑨	ä	IX	Ü	Ⅻ
、	.	(z)	{a}	♡	1	❾	Ä	kg	vii	z

(Similar to ja-x-icu)

SELECT t FROM t_zh_x_icu ORDER BY t;

(zh-x-icu)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
z	Ë	까	あ～	あっ	は	パ	変	喃	総	
Z	⁎	찌	あ…	あつ	ハ	ﾊﾟ	変	砥	總	
Z	⁑	히	あ~	キロ	ﾊ	はﾞ	變	粁	脬	
α	ـٹ	あ	あ～	キロ	ば	ハﾞ	働	畑	靡	
A	ـٹ	ア	あゝ	キロ	バ	ん	高	畠	鼬	
ω	ـے	ｱ	あー	キログラム	ﾊﾞ	ン	髙	一		
Ω	ـۓ	あﾞ	あぁ	キロメートル	ぱ	ﾝ	九	壱		
È	가	あ-	ああ	センチ	はﾟ	匂	糎	总		

(2/2)

SELECT t FROM t_zh_Hans_x_icu ORDER BY t;

(zh-Hans-x-icu)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
a	!	。	(z)	【a】	💗	①	9.	cm	k g	vii
a	!	。	㈱	@	$	❶	9.	c m	kg	VII
_a	¡	⑴	(株)	@	$	🆒	a	cm	km	Ⅶ
__a	?	⑼	（株）	/a	0	㊿	a	i	k m	xii
,	？	⑼	[a]	／a	0	9	A	I	km	xii
,	¿	⑻	[a]	#	⓪	9	A	ix	ü	XII
、	.	(a)	{a}	#	1	⑨	ä	IX	Ü	Ⅻ
、	.	(z)	{a}	♡	1	❾	Ä	kg	vii	z

SELECT t FROM t_zh_Hans_x_icu ORDER BY t;

(zh-Hans-x-icu)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
z	Ë	까	あ～	あっ	は	パ	変	喃	総	
Z	⁎	찌	あ…	あつ	ハ	ﾊﾟ	変	砥	總	
Z	⁑	히	あ～	キロ	ﾊ	はﾞ	變	粁	錚	
α	ٹ	あ	あ～	ｷﾛ	ば	ハﾞ	働	畑	靡	
A	ٹ	ア	あゝ	キロ	バ	ん	高	畠	鞠	
ω	ے	ｱ	あー	キログラム	ﾊﾞ	ン	髙	一		
Ω	ۓ	あﾞ	あぁ	キロメートル	ぱ	ﾝ	九	壱		
È	가	あ-	ああ	センチ	はﾟ	匂	糎	总		

SELECT t FROM t_zh_Hant_x_icu ORDER BY t;

(zh-Hant-x-icu)

#001 -008	#009 -016	#017 -024	#025 -032	#033 -040	#041 -048	#049 -056	#057 -064	#065 -072	#073 -080	#081 -088
a	!	。	(z)	【a】	💗	①	9.	cm	k g	vii
a	!	。	㈱	@	$	❶	9.	c m	kg	VII
_a	¡	⑴	(株)	@	$	🆒	a	cm	km	Ⅶ
__a	?	⑼	（株）	/a	0	㊿	a	i	k m	xii
,	？	⑼	[a]	／a	0	9	A	I	㎞	xii
,	¿	⑻	[a]	#	⓪	9	A	ix	ü	XII
、	.	（a）	{a}	#	1	⑨	ä	IX	Ü	Xll
、	.	⑻	{a}	♡	1	❾	Ä	kg	vii	z

(Similar to zh-Hans-x-icu)

SELECT t FROM t_zh_Hant_x_icu ORDER BY t;

(zh-Hant-x-icu)

#089 -096	#097 -104	#105 -112	#113 -120	#121 -128	#129 -136	#137 -144	#145 -152	#153 -160	#161 -168	#169 -176
z	Ë	까	あ～	あっ	は	パ	九	粁	總	
Z	⁎	찌	あ…	あつ	ハ	ﾊﾟ	匂	畠	變	
Z	⁑	히	あ~	キロ	ﾊ	はﾞ	壱	高	孿	
α	ㅊ	あ	あ～	ｷﾛ	ば	ハﾞ	変	髙	靡	
A	ㄴ	ア	あゝ	ｷﾛ	バ	ん	砥	喃	鞠	
ω	ㅷ	ｱ	あー	キログラム	ﾊﾞ	ン	変	働		
Ω	ㅹ	あﾞ	あぁ	キロメートル	ぱ	ﾝ	总	総		
È	가	あ-	ああ	センチ	はﾟ	—	畑	糎		

(Similar to zh-Hans-x-icu)

libc vs. ICU (English)

libc (en_US.utf8)

```
citus=> EXPLAIN SELECT t FROM t_en_us_utf8 ORDER BY t;
                              QUERY PLAN
-----------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4)
   Sort Key: t COLLATE "en_US.utf8"
   ->  Seq Scan on t_en_us_utf8  (cost=0.00..2.65 rows=165 width=4)
```

ICU (en-US-x-icu)

```
citus=> EXPLAIN SELECT t FROM t_en_us_x_icu ORDER BY t;
                              QUERY PLAN
-----------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4)
   Sort Key: t COLLATE "en-US-x-icu"
   ->  Seq Scan on t_en_us_x_icu  (cost=0.00..2.65 rows=165 width=4)
```

libc vs. ICU (Japanese)

libc (ja_JP.utf8)

```
citus=> EXPLAIN SELECT t FROM t_ja_JP_utf8 ORDER BY t;
                              QUERY PLAN
-----------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4)
   Sort Key: t COLLATE "ja_JP.utf8"
   ->  Seq Scan on t_ja_jp_utf8  (cost=0.00..2.65 rows=165 width=4)
```

ICU (ja-x-icu)

```
citus=> EXPLAIN SELECT t FROM t_ja_x_icu ORDER BY t;
                              QUERY PLAN
-----------------------------------------------------------------------
 Sort  (cost=9.73..10.14 rows=165 width=4)
   Sort Key: t COLLATE "ja-x-icu"
   ->  Seq Scan on t_ja_x_icu  (cost=0.00..3.65 rows=165 width=4)
```

ja-x-icu vs. ja-JP-x-icu

ja-x-icu

```
citus=> EXPLAIN SELECT t FROM t_ja_x_icu ORDER BY t;
                            QUERY PLAN
-------------------------------------------------------------------
 Sort  (cost=9.73..10.14 rows=165 width=4)
   Sort Key: t COLLATE "ja-x-icu"
   ->  Seq Scan on t_ja_x_icu  (cost=0.00..3.65 rows=165 width=4)
```

ja-JP-x-icu

```
citus=> EXPLAIN SELECT t FROM t_ja_JP_x_icu ORDER BY t;
                            QUERY PLAN
-------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4)
   Sort Key: t COLLATE "ja-JP-x-icu"
   ->  Seq Scan on t_ja_jp_x_icu  (cost=0.00..2.65 rows=165 width=4)
```

en-US-x-icu vs. ja-JP-x-icu

en-US-x-icu

```
citus=> EXPLAIN SELECT t FROM t_en_us_x_icu ORDER BY t;
                              QUERY PLAN
-----------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4)
   Sort Key: t COLLATE "en-US-x-icu"
   ->  Seq Scan on t_en_us_x_icu  (cost=0.00..2.65 rows=165 width=4)
```

ja-JP-x-icu

```
citus=> EXPLAIN SELECT t FROM t_ja_JP_x_icu ORDER BY t;
                              QUERY PLAN
-----------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4)
   Sort Key: t COLLATE "ja-JP-x-icu"
   ->  Seq Scan on t_ja_jp_x_icu  (cost=0.00..2.65 rows=165 width=4)
```

ja-x-icu vs. ja-JP-x-icu with COLLATE clause

COLLATE "ja-x-icu"

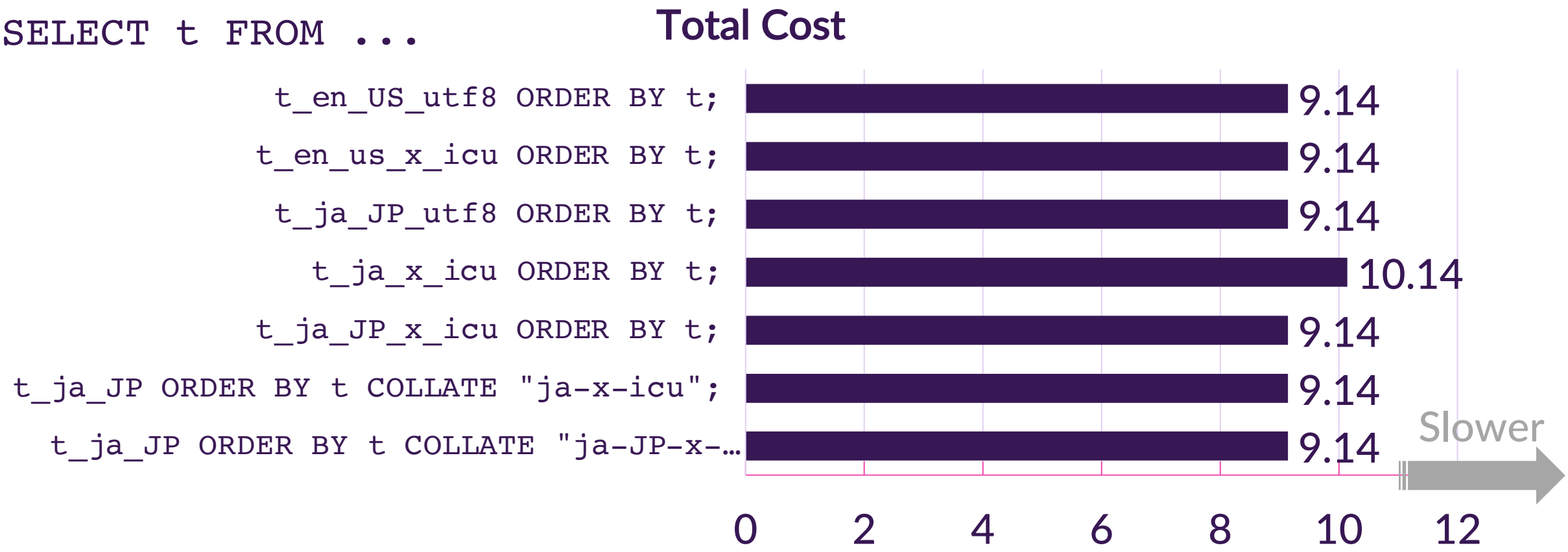
```
citus=> EXPLAIN SELECT t FROM t_ja_JP ORDER BY t COLLATE "ja-x-icu";
                            QUERY PLAN
-------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=36)
   Sort Key: t COLLATE "ja-x-icu"
   ->  Seq Scan on t_ja_jp  (cost=0.00..2.65 rows=165 width=36)
```

COLLATE "ja-JP-x-icu"

```
citus=> EXPLAIN SELECT t FROM t_ja_JP ORDER BY t COLLATE "ja-JP-x-icu";
                            QUERY PLAN
-------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=36)
   Sort Key: t COLLATE "ja-JP-x-icu"
   ->  Seq Scan on t_ja_jp  (cost=0.00..2.65 rows=165 width=36)
```

Summary of Total Cost by EXPLAIN

```
SELECT t FROM ...
```



libc vs. ICU (English)

libc (en_US.utf8)

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_en_US_utf8 ORDER BY t;
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4) (actual time=0.990..1.006 rows=165 loops=1)
   Sort Key: t COLLATE "en_US.utf8"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_en_us_utf8  (cost=0.00..2.65 rows=165 width=4) (actual time=0.016..0.109 rows=165 loops
 Planning Time: 0.153 ms
 Execution Time: 1.046 ms
```

ICU (en-US-x-icu)

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_en_us_x_icu ORDER BY t;
                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4) (actual time=0.535..0.551 rows=165 loops=1)
   Sort Key: t COLLATE "en-US-x-icu"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_en_us_x_icu  (cost=0.00..2.65 rows=165 width=4) (actual time=0.016..0.042 rows=165 loops=1
 Planning Time: 0.152 ms
 Execution Time: 0.591 ms
```

libc vs. ICU (Japanese)

libc (ja_JP.utf8)

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_ja_JP_utf8 ORDER BY t;
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4) (actual time=2.498..2.514 rows=165 loops=1)
   Sort Key: t COLLATE "ja_JP.utf8"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_ja_jp_utf8  (cost=0.00..2.65 rows=165 width=4) (actual time=0.017..0.044 rows=165 loops
 Planning Time: 0.205 ms
 Execution Time: 2.553 ms
```

ICU (ja-x-icu)

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_ja_x_icu ORDER BY t;
                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------
 Sort  (cost=9.73..10.14 rows=165 width=4) (actual time=0.667..0.688 rows=165 loops=1)
   Sort Key: t COLLATE "ja-x-icu"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_ja_x_icu  (cost=0.00..3.65 rows=165 width=4) (actual time=0.011..0.044 rows=165 loops=1)
 Planning Time: 0.150 ms
 Execution Time: 0.742 ms
```

ja-x-icu vs. ja-JP-x-icu

ja-x-icu

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_ja_x_icu ORDER BY t;
                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Sort  (cost=9.73..10.14 rows=165 width=4) (actual time=0.667..0.688 rows=165 loops=1)
   Sort Key: t COLLATE "ja-x-icu"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_ja_x_icu  (cost=0.00..3.65 rows=165 width=4) (actual time=0.011..0.044 rows=165 loops=1)
 Planning Time: 0.150 ms
 Execution Time: 0.742 ms
```

ja-JP-x-icu

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_ja_JP_x_icu ORDER BY t;
                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=4) (actual time=0.493..0.509 rows=165 loops=1)
   Sort Key: t COLLATE "ja-JP-x-icu"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_ja_jp_x_icu  (cost=0.00..2.65 rows=165 width=4) (actual time=0.016..0.044 rows=165 loops=1)
 Planning Time: 0.146 ms
 Execution Time: 0.547 ms
```

ja-x-icu vs. ja-JP-x-icu with COLLATE clause

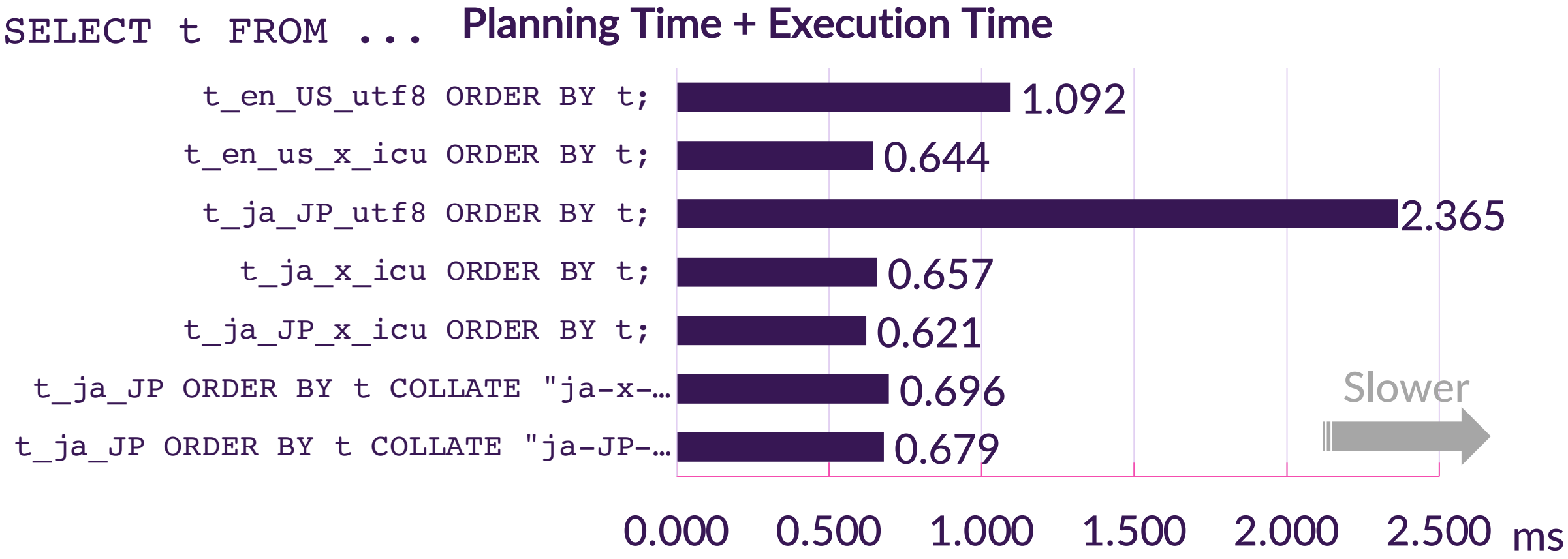
COLLATE "ja-x-icu"

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_ja_JP ORDER BY t COLLATE "ja-x-icu";
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=36) (actual time=0.472..0.488 rows=165 loops=1)
   Sort Key: t COLLATE "ja-x-icu"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_ja_jp  (cost=0.00..2.65 rows=165 width=36) (actual time=0.017..0.049 rows=165 loops=1)
 Planning Time: 0.171 ms
 Execution Time: 0.536 ms
```

COLLATE "ja-JP-x-icu"

```
citus=> EXPLAIN ANALYZE SELECT t FROM t_ja_JP ORDER BY t COLLATE "ja-JP-x-icu";
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Sort  (cost=8.73..9.14 rows=165 width=36) (actual time=0.551..0.567 rows=165 loops=1)
   Sort Key: t COLLATE "ja-JP-x-icu"
   Sort Method: quicksort  Memory: 32kB
   ->  Seq Scan on t_ja_jp  (cost=0.00..2.65 rows=165 width=36) (actual time=0.014..0.050 rows=165 loops=1)
 Planning Time: 0.106 ms
 Execution Time: 0.614 ms
```

Summary of Actual Time by ANALYZE



* Each value is the median of 10-time ANALYZE execution results.

Conclusion

- Using Azure Database for PostgreSQL Hyperscale (Citus), you can specify locale and collation for each table (not database).
- ICU collation is better than libc not only for the sort order but also the performance.
- Setting up ICU collation for each table is better than using COLLATE clause with ICU collation for libc tables.

References

- PostgreSQL: Documentation
  - https://www.postgresql.org/docs/
- ロケール(国際化と地域化) | Let's POSTGRES
  - https://lets.postgresql.jp/documents/technical/text-processing/2
- PostgreSQL 10のICUコレーションを使うと日本語を普通にソートでき、更に文字順序までカスタマイズできる – yohgaki's blog
  - https://blog.ohgaki.net/postgresql-10-icu-locale-collation-enables-natural-japanese-sorting
- PostgreSQL Collations | Vertabelo Database Modeler
  - https://vertabelo.com/blog/collations-in-postgresql/

References

- Azure Database for PostgreSQLでCollation(照合順序)を指定してCREATE DATABASEしたい - 浅草橋青空市場
  - https://asazure.hatenablog.jp/entry/2018/04/13/203114
- Don't let collation versions corrupt your PostgreSQL indexes - Microsoft Tech Community
  - https://techcommunity.microsoft.com/t5/azure-database-for-postgresql/don-t-let-collation-versions-corrupt-your-postgresql-indexes/ba-p/1978394

Agenda

- **Introduction**
- **A number of different character sets**
  - Character Set Support
- **Locale-specific collation order**
  - Locale Support
  - Collation Support

THANK YOU AND HAPPY HACKING!

Q&A and discussion
in Microsoft Open Source Discord

#cituscon

Keep in touch with the speaker?

@kske_t

@k14i