Character Set & Globalization

Martin Hoermann
ORDIX AG
Paderborn

Schlüsselworte
Character Set, Globalization, Unicode, UTF8, UTF16, DMU

Einleitung

Die Auswahl des Character Set beim Anlegen einer Datenbank entscheidet über die zur Verfügung stehenden Zeichen der Datenbank. Die richtige Verwendung und die Änderung des Zeichensatzes, z.B. auf einen Unicode-Zeichensatz, stellt den Administrator und Anwendungsentwickler allerdings vor verschiedene Herausforderungen.

Der Vortrag beleuchtet die Grundlagen von Character Sets, die Migration des Datenbank Character Set und die verschiedenen Ausprägungen der unter Oracle verfügbaren Unicode-Zeichensätze.

Eine Übersicht über weitere Aspekte der Globalisierung - oder des National Language Support - rundet den Vortrag ab.

Character Set

Das Character Set einer Datenbank legt prinzipiell die zur Verfügung stehenden Zeichen einer Oracle-Datenbankanwendung fest. In Westeuropa waren früher die Datenbankzeichensätze WE8ISO8859P1, WE8ISO8859P15 und WE8MSWIN1252 weit verbreitet. Diese Zeichensätze haben höchstens 256 Zeichen, sodass jedes Zeichen mit einem Byte kodiert werden kann. Diese so genannten Single-Byte-Zeichensätze haben aber den Nachteil, dass nicht definierte Zeichen - z.B. die Buchstaben des griechischen Alphabets - damit nicht definiert gespeichert werden können. Es ist zwar prinzipiell möglich die Bytes anders als definiert zu interpretieren, dies führt jedoch zu einer sehr unübersichtlichen Datenspeicherung. Hiervon ist dringend abzuraten.

Um das Problem des eingeschränkten Zeichensatzes zu beheben wurde mit Unicode ein sehr mächtiger Standard definiert, der die Zeichen der meisten lebenden Sprachen und vieler ausgestorbener Sprachen definiert. Die am meisten verwendeten Kodierungen der Datenbank sind UTF8 (veraltet), AL32UTF8 und AL16UTF16. Mit dem Unicode-Standard 6.1 werden 110.181 Zeichen in Oracle 12c zur Verfügung gestellt. Die Unicode-Zeichensätze sind sogenannte Multibyte-Zeichensätze, die je nach Kodierung und Zeichen zwischen einem und vier Byte Speicherplatz benötigen.

Vom Client zur Datenbank und Zurück

Dieses Kapitel beschreibt den Weg eines Zeichens von der Tastatur bis hin zur Datenbank und umgekehrt von der Datenbank zum Bildschirm des Anwenders. Wenn eine Datei auf einem Client erzeugt wird, so hat diese Datei eine bestimmte Kodierung (engl. Encoding) oder auch einen bestimmten Zeichensatz. Idealerweise sollte die NLS_LANG-Variable genau diesen Zeichensatz

benennen, z.B. NLS_LANG=. WE8MSWIN1252 wenn es um eine auf Windows mit einem Standard-Editor erzeugte Datei geht.

Unterscheidet sich der Zeichensatz in der Definition von NLS_LANG und dem Datenbank-zeichensatz, so wird über einen festgelegten Algorithmus eine Konvertierung durchgeführt. Existiert dasselbe Zeichen in beiden Zeichensätzen, so wird es entsprechend auf die Kodierung des Ziels - also der Datenbank - konvertiert. Existiert das Zeichen nicht, aber ein äquivalentes Zeichen, so wird das Zeichen ersetzt. So könnte beispielsweise aus einem „ä“ ein „a“ werden. Gibt es weder das genaue Zeichen noch ein äquivalentes Zeichen, so wird ein Ersatzzeichen (Replacement Character) verwendet. Äquivalente Zeichen und Ersatzzeichen können nicht wieder zurückgewandelt werden. Idealerweise sollten die Zeichensätze zwischen Client und Datenbank also entweder identisch sein, oder die Datenbank sollte eine Obermenge (Superset) der Zeichen zur Verfügung stellen. Dies ist dann in der Regel ein Unicode-Zeichensatz. Sind die Kodierung auf dem Client, der Client-Zeichensatz und Datenbankzeichensatz identisch gehen weder Zeichen verloren noch werden Zeichen ersetzt. Dies ist der Idealzustand einer jeden Datenbankanwendung.

Wird nun ein Zeichen aus der Datenbank zurückgelesen, passiert prinzipiell genau das zuvor Beschriebene nur in umgekehrter Richtung. Unterscheiden sich also NLS_LANG des Client und der Zeichensatz der Datenbank können Zeichen entweder in ein äquivalentes Zeichen oder ein Ersatzzeichen konvertiert werden. Alle Daten die verlustfrei vom Client in die Datenbank gelangt sind, kommen bei gleicher Konfiguration auch verlustfrei wieder zurück. Ob die Zeichen allerdings korrekt dargestellt werden können, hängt vom Zeichensatz des Frontend und dem zur Verfügung stehenden Font zur Darstellung ab. In einer DOS-Box (Codepage 850) gelingt es meist nicht, trotz korrekter Einstellung, die deutschen Umlaute darzustellen. Schreibt man hingegen aus einer DOS-Box eine Spool-Datei und öffnet diese wiederum mit einem Standard-Editor (Codepage 1252), so sind die Zeichen dort sichtbar.

Invalid Data & Conversion Errors

Bei der oben beschriebenen Verbindung kann es zu zwei Arten von Problemen kommen, die Oracle als Invalid Data und Conversion Errors bezeichnet. Beide Phänomene wurde bereits zuvor erklärt und werden nun an einem Beispiel veranschaulicht.

Ist die Kodierung einer Datei beispielsweise in WE8ISO8859P7 und enthält Zeichen des griechischen Alphabets und die NLS_LANG-Variable sowie der Datenbankzeichensatz ist auf WE8ISO8859P1 eingestellt, so werden die Zeichen an die entsprechenden Positionen 0xA0..0xFF geschrieben. Wenn die Daten mit derselben Konfiguration ausgelesen und als P7-kodiert interpretiert werden, sind die griechischen Zeichen wieder sichtbar. Ohne dieses Wissen sind die Daten jedoch bedeutungslos, da es sich im P1-Zeichensatz um Sonderzeichen handelt. Wird die Datenbank nun beispielsweise in einen anderen Zeichensatz konvertiert, werden natürlich semantisch die P1-Daten konvertiert, was ein Auslesen der griechischen Zeichen praktisch unmöglich macht. Oracle spricht bei diesen Daten von Invalid Data.

Ist die Kodierung einer Datei beispielsweise WE8ISO8859P7 und enthält Zeichen des griechischen Alphabets und die NLS_LANG-Variable ist auf P7 und der Datenbankzeichensatz ist auf P1 eingestellt, so konvertiert die Datenbank die Zeichen. Dabei werden bis auf das kleine Beta alle Zeichen durch den Replacement Character des P1, das umgekehrte Fragezeichen, ersetzt. Das kleine Beta wird zum deutschen „ß“. Die ursprünglichen Daten sind damit nicht mehr reproduzierbar. Oracle spricht von Conversion Errors.

Encoding I

Bei Single-Byte-Datensätzen ist die Kodierung relativ einfach. Jeder Buchstabe bekommt einen Wert zwischen 0x00 und 0xFF zugewiesen. Die Zuweisung erfolgt über Character-Set-Tabellen, die im Internet - z.B. bei Wikipedia - einfach zu recherchieren sind.

Bei Unicode-Zeichensätzen gestaltet sich die Kodierung etwas schwieriger. Jedes Unicode-Zeichen bekommt eine Nummer. Diese Nummer wird in der Regel Hex-kodiert mit führenden „U+“ geschrieben. So ist ein „Ä“ beispielsweise ein „U+00C4“. Im Internet stehen umfangreiche Unicode-Tabellen zur Verfügung.

Jetzt gibt es für verschiedene Anwendungszwecke unterschiedliche Kodierungen. Für den westeuropäischen Raum eignet sich die Kodierung AL32UTF8. Dabei werden die 128 ASCII-Zeichen identisch wie in der ASCII-Tabelle kodiert und kommen daher mit 7 Bit plus eines führenden 0-Bit aus. Alle anderen Zeichen werden mit zwei bis vier Byte kodiert. Hilfreich kann der Algorithmus zum Auslesen sein, um z.B. festzustellen ob ein „Ä“ tatsächlich ein „Ä“ ist. Mit Hilfe der `dump`-Funktion auf ein einzelnes Zeichen ergibt sich z.B.:

```
SELECT dump(  zuklaerendeszeichen ) FROM tabelle

Typ=1 Len=2 195, 132
```

Im ersten Schritt bietet es sich an, die Zahlen in Binär-Schreibweise aufzustellen:

11000011 **10**000100

Die führende „110“ des ersten Byte beschreibt, dass es sich um einen Zweibyte-Zeichen handelt. Die führende „10“ des zweiten Byte besagt, dass es ein Folge-Byte ist. Alle anderen Zahlen werden nun zusammengefasst:

00011000100 = 196^{10} = U+00C4 = Ä

Bei dem zu klärenden Zeichen handelt es sich also tatsächlich um ein „Ä“.

Encoding II

Wie bereits oben erläutert sind vielfältige Komponenten auf dem Weg von der Tatstatur bis zur Datenbank beteiligt. Der Oracle-Datenbankadministrator legt beim Anlegen der Datenbank, i.d.R. nach Vorgabe des Applikationsverantwortlichen, den Zeichensatz der Datenbank fest. Dieser ist im Nachhinein nur mit einigen Aufwand zu ändern.

Auf der Seite des Client sind die Oracle-NLS-Einstellungen und die Kodierung der Anwendung relevant. Im Falle einer Datei kann diese von verschiedenen Werkzeugen gelesen und manipuliert werden. SQL*Developer, TOAD und Ultra-Compare verfügen zwar über Einstellungen für die Default-Kodierung, interpretieren aber häufig zur Laufzeit anhand des Dateninhalts die Kodierung der Datei um. Gelangt z.B. ein „Schmierzeichen“ in eine Datei so wird von vielen Tools eine Unicode-Kodierung angenommen. Dies kann fatale Auswirkungen bei der Arbeit mit den Dateien haben. Für die Arbeit mit einer Anwendung gilt prinzipiell dasselbe.

Unicode in der Datenbank

Die Datenbank verfügt über zwei Zeichensätze (v$nls_parameters). Der Datenbankzeichensatz (NLS_CHARACTERSET) legt die Kodierung für die Spalten mit den Datentypen VARCHAR2, CHAR und CLOB fest. Der NLS-Datenbank-Zeichensatz (NLS_NCHAR_CHARACTERSET) legt die Kodierung für die Spalten mit den Datentypen NVARCHAR2, NCHAR und NCLOB fest.

Bis Oracle 8 war der Name des UTF-8 Zeichensatzes UTF8, ab der Version 9 ist der Name AL32UTF8. Der Präfix „AL“ steht dabei für All Languages. Abhängig von der Oracle-Version wird mit diesem Zeichensatz eine jeweils aktuelle Unicode-Spezifikation unterstützt, beispielsweise in Oracle 11 die Spezifikation 5.0 und in Oracle 12.1 die Spezifikation 6.1.

In früheren Versionen war es noch möglich, sowohl für den Datenbankzeichensatz also auch den NLS-Datenbankzeichensatz einheitlich UTF8 zu verwenden. Bei Migrationen stehen bei dem Datenbank-Zeichensatz AL32UTF8, für den NLS-Datenbank-Zeichensatz ausschließlich die Zeichensätze AL16UTF16 und UTF8 zur Verfügung. Der Zeichensatz AL16UTF16 ist unter anderen optimiert auf chinesische, koreanische und japanische Zeichen von denen die meisten nur zwei Byte benötigen, in AL32UTF8 benötigen diese Zeichen in der Regel drei Byte.

Bei der Migration des Zeichensatzes kann es aufgrund der unterschiedlichen Kodierung natürlich zu einem Datenverlust kommen, da die Kodierungen für ein und dieselben Zeichen unterschiedlich viele Bytes benötigen. Insbesondere die Einstellung von NLS_LENGTH_SEMANTIC auf Bytes statt Character verursacht hier potenziell Probleme. Aber auch mit der Einstellung auf Character kann das Problem entstehen, wenn die maximale Größe des Datentyps CHAR und VARCHAR2 bei der Konvertierung 4.000 Bytes überschreitet.

Character Set Migration

Oracle bietet verschiedene Möglichkeiten, um den Zeichensatz einer Datenbank zu migrieren oder die Migration zu prüfen: Export/Import, CSSCAN/CSALTER und DMU.

Das Database Migration Utility (DMU) steht seit der Oracle-Version 12.1 zur Verfügung und soll die Arbeit bei der Migration des Zeichensatzes vereinfachen.

Bei der Migration auf einen anderen Zeichensatz besteht aufgrund der Kodierung immer die Möglichkeit des Datenverlustes (Data Truncation), weil Charater-Felder eine für die neue Kodierung nicht ausreichende Länge haben (siehe oben). Weiterhin kann es zu Konvertierungsfehlern kommen, wenn ein Zeichen des ursprünglichen Zeichensatzes im neuen Zeichensatz nicht zur Verfügung steht (Replacement Errors). In diesem Fall werden die Zeichen durch den sogenannten Replacement Character ersetzt (siehe oben). Sowohl DMU also auch der Vorgänger CSSCAN prüfen die Daten vor der Migration auf potenzielle Fehler.

Sortierung

Der Datenbankzeichensatz legt die zur Verfügung stehenden Zeichen fest. Dies ist jedoch nur die Basis der Globalisierung einer Datenbankanwendung. Im nächsten Schritt muss für die Anwendung eine Sortierung festgelegt werden. Dies geschieht in der Regel über den Parameter NLS_SORT, der wiederum abhängig vom Parameter NLS_LANGUAGE ist. Für die „gängigen“ Sprachen stehen die dort typischweise verwendeten Sortiermöglichkeiten zur Verfügung. Dies sind für den deutschen Sprachraum beispielsweise GERMAN, XGERMAN, GERMAN_DIN und XGERMAN_DIN.

Ab der Version 12 stellt Oracle eine Implementierung des Unicode Collation Algorithm (UCA) zur Verfügung. Über dieses Verfahren können auch mehrsprachige Sortierungen eingestellt und sogar selbst implementiert werden.

Fazit

Dieser Artikel zeigt die zahlreichen Facetten des Datenbankzeichensatzes. Mit einem Unicode-Zeichensatz stehen zwar alle gängigen Zeichen dieser Welt zur Verfügung, nichtsdestotrotz kann es bei der Anwendung zahlreiche Hürden geben.

Neben der Sortierung ist das Thema der Globalisierung natürlich erheblich weitreichender. Themen wie Monats- und Wochentagsnamen, Dezimal- und Tausendertrenner, erster Tag der Woche und Währung sind hier einige Stichpunkte.

Kontaktadresse:
Martin Hoermann
ORDIX AG
Westernmauer 12-16
D-33098 Paderborn

Telefon:	+49 (0) 5251 / 1063 - 0
Fax:	+49 (0) 180 / 1673 490
E-Mail	info@ordix.de
Internet:	www.ordix.de