Processing data in Java in Oracle

Is Oracle always the best tool or application to handle data? The answer lies inside what you do. Typically, data is transformed through the data processing language, but it can also be transformed through processing strings or numbers.

Is Oracle always the best tool or application to handle data? The answer lies inside what you do. Typically, data is transformed through the data processing language, but it can also be transformed through processing strings or numbers. To distinguish between these types of processing, assume that the data is stored in the DML scope, what the output or display data is or how the saved data is displayed to the user. A simple example for this case is the date. What is saved may be required to display in different ways. Examples are March 8th, 2007 as well as August 3, 2007; It depends on what you or your users want to see.

The main focus of this article relates to how you can get from things saved to show it somehow. This focus relates to problems handled around a string or a number. Assume you have an application that supports statistics or provides votes to users via the Internet. Clicking to submit in an online statistics form, statistics by ID or by account number, to be passed as one of the name - value pairs in a URL, this is a commonly done problem. in your identifiers.

Another use for this 'back-to-back' transmission mechanism is to validate or limit your use. Votes, whether in paper or electronic forms, are represented by digital formats. Very large numbers make them more difficult to forge when their density is more spaced.
The problem though is that the transmission of a 22-digit number of barcodes, for example, consumes space or length in the URL (limit of 255 or 256 characters). What is needed here is how to shorten a large number into a smaller number.

One method to solve this problem is to take numbers in the base 10 system and convert them into numbers in the base-36 system, that way you can save space. In Oracle, how would you do that?

PL / SQL method

Consider the PL / SQL method. The code is optimized into bits, in which factors have a jump in the form of repetition or repetition used in accounts. It also relies on current input with a minimum of 22 digits.

```
CREATE OR REPLACE FUNCTION gen_barcode36 (i_coupon_number NUMBER)
RETURN VARCHAR2 IS
v_number number;
v_curval number;
v_curinc number: = power (36,14);
v_true number: = 0;
```

```
v_pos number: = 15;
v_dec_count number: = 0;
v_pos_val number: = 0;
v_cur_num number: = 0;
v_test_num number: = 0;
v_new_num number: = 0;
v_cur_digit char: = '';
v_new_val varchar2 (15): = '';
BEGIN
v_number: = i_coupon_number;
while (v_true = 0) loop
v_curinc: = v_curinc * 36;
v_curval: = v_number / v_curinc;
if (v_curval 1) then
v_true: = 1;
else
v_pos: = v_pos + 1;
end if;
end loop;
v_dec_count: = v_pos;
v_new_val: = NULL;
v_cur_num: = v_number;
WHILE (v_dec_count> 0) LOOP
v_pos_val: = power (36, v_dec_count - 1);
v_test_num: = trunc (v_cur_num / v_pos_val);
select decode (v_test_num, 35, 'z', 34, 'y', 33, 'x',
32, 'w', 31, 'v', 30, 'u',
29, 't', 28, 's', 27, 'r',
26, 'q', 25, 'p', 24, 'o',
23, 'n', 22, 'm', 21, 'l',
20, 'k', 19, 'j', 18, 'i',
17, 'h', 16, 'g', 15, 'f',
14, 'e', ??13, 'd', 12, 'c',
11, 'b', 10, 'a', 9, '9',
8, '8', 7, '7', 6, '6',
5, '5', 4, '4', 3, '3',
2, '2', 1, '1', '0')
INTO v_cur_digit FROM DUAL;
IF (v_new_val IS NOT NULL) THEN
v_new_val: = v_new_val || v_cur_digit;
ELSE
v_new_val: = v_cur_digit;
END IF;
v_cur_num: = v_cur_num - (v_pos_val * v_test_num);
v_dec_count: = v_dec_count - 1;
END LOOP;
RETURN v_new_val;
END gen_barcode36;
```

/

Another version of this method has output attached to it so you can see how the numbers are reduced.

```
SQL> select gen_barcode36 (770000000000000000000000) from dual;
GEN_BARCODE36 (7.7 billion trillion)
---------------------------------------------- ---
1950zn8fqxjygow
Starting with 7.7 billion billion
Position: 6140942214464815497216 current number: 1
Position 14 current value: 1
Position: 170581728179578208256 current number: 9
Position 13 current value: 19
Position: 4738381338321616896 current number: 5
Position 12 current value: 195
Position: 131621703842267136 current number: 0
Position 11 current value: 1950
Position: 3656158440062976 current number: 35
Position 10 current value: 1950z
Position: 101559956668416 current number: 23
Position 9 current value: 1950zn
Position: 2821109907456 current number: 8
Position 8 current value: 1950zn8
Position: 78364164096 current number: 15
Position 7 current value: 1950zn8f
Position: 2176782336 current number: 26
Position 6 current value: 1950zn8fq
Position: 60466176 current number: 33
Position 5 current value: 1950zn8fqx
Position: 1679616 current number: 19
Position 4 current value: 1950zn8fqxj
Position: 46656 current number: 34
Position 3 current value: 1950zn8fqxjy
Position: 1296 current number: 16
Position 2 current value: 1950zn8fqxjyg
Position: 36 current number: 24
Position 1 current value: 1950zn8fqxjygo
Position: 1 current number: 32
Position 0 current value: 1950zn8fqxjygow
Final value: 1950zn8fqxjygow
```

That does not mean security through obscure status. However, a strange string will make it difficult for most users to calculate what is displayed.

If a value is taken into account, the data processing performance is 'OK', but how long will it take to create a million values? Create a table with 3 columns. The first column is the base column 10, the second column is the value processed by base number 36 using the PL / SQL Server code above and the third column is reserved for

output by using a different meaning to variable change the numbers. Create a table and include a million records. Adjust the time to see how long it will take to create these 36 base values.

```
SQL> create table base36
2 (barcode number,
3 plsql_ver varchar2 (15),
4 other_ver varchar2 (15));
T?o b?ng.
SQL> create sequence barcode_seq start with 7.7 billion trillion;
T?o b?n ??.

SQL> begin
2 for i in 1.1000000 loop
3 insert into base36 (barcode) values ??(barcode_seq.nextval);
4 end loop;
5 end;
6 /
PL / SQL procedure was successfully completed.
SQL> commits;
Complete commit.
SQL> set timing on
SQL> update base36 set plsql_ver = gen_barcode36 (barcode);
1000000 rows updated.
Elapsed: 00: 33: 09.01
```

Working speed is over 500 copies / s

Could this be a quick way of PLATFORM / SQL to accomplish this task? This problem is a good example as we use language better in communication, while PL / SQL performs data transformation. A good method in this case will be used to handle strings in Java and another method attached (for strings).

How to put Java code into the database?

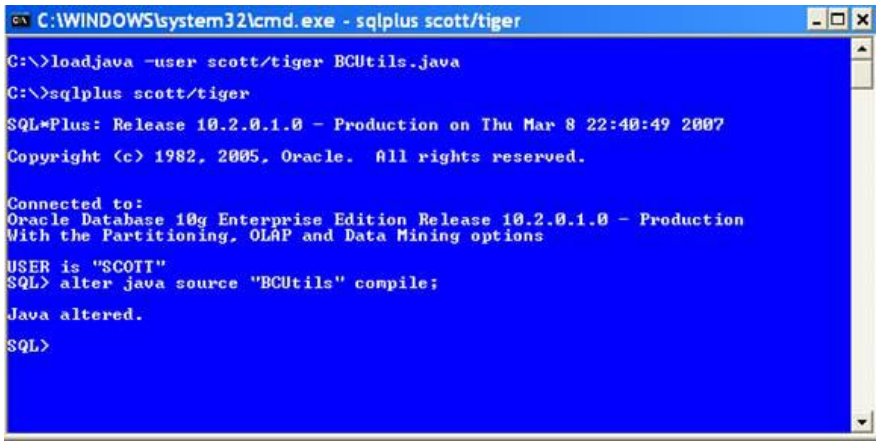
First, we need to have a Java file with the appropriate code in it. The second is to upload the code (source or compiled version) into the database. The third is to compile the code, and then the final step is to create a wrapper function or procedure around Java code. That function is called the Oracle name object, it will call the Java name object.

Source code.

```
import java.math.BigInteger;
import java.lang.String;
public class BCUtils
{
public static String getBarcode36 (String barcode10)
{
String value = new String (barcode10);
```

```
BigInteger bigI = new BigInteger (value.toString ());
StringBuffer result = new StringBuffer (bigI.toString (36));
return result.toString ();
}
}
```

Download the source file and compile it.



Create wrapper / publish function

```
SQL> create or replace function get_bc36 (bc_10 varchar2)
2 return varchar2
3 as language java name
4 'BCUtils.getBarcode36 (java.lang.String) return java.lang.String';
5 /
Hàm t?o.
```

Check the difference

```
SQL> update base36 set other_ver = get_bc36 (barcode);
1000000 rows updated.
Elapsed: 00: 13: 56.22
```

Working speed reaches 1200 / s, twice as fast as the PLATFORM / SQL version. Why is that? Oracle acknowledges that PL / SQL is not the fastest language in the world or best for string processing.

List of important steps introduced in the section titled 'Java Stored Procedure Steps' in the Java Developer tutorial.

Conclude

PL / SQL can do many good things, but other languages ??can do much better. That can be verified in the form of speed measurements. If you find that saved data must be processed to display at the output, you should choose a different language to support for example Oracle.

guides. Thank you for reading and for following us regularly.