Working with Date and Time Values

This chapter deals with recipes for date, time, and related temporal datatypes. Queries involving dates and times are very common in almost all databases. Working with dates and times is also very strange when compared to other forms of data, due to the long history (pardon the pun) of how our concepts of dates, hours, minutes, seconds, days, months and years came about. For instance, our division of minutes into 60 seconds, hours into sixty minutes, and days into two 12-hour blocks is derived from ancient Sumerian and Babylonian tradition. And they got the notion of a base-12 for counting hours and other things by using the segments of the four fingers (not the thumb) of one hand.

Calendars around the world stem from various religious influences, the whims of emperors and kings in different regions, and more. To put the icing on the cake, when you think you understand dates and times at a fundamental level, just ask astronomers for their perspective. They’ll tell you there are at least three different measurements for the length of a day—depending on whether you’re measuring rotation of the Earth; the Earth’s time to return to point to the same fixed stars in the night sky; and, strangest of all, simply considering the day as an equation relating to angles in space.

It is therefore no surprise that dealing with date and time values, performing temporal arithmetic, and answering supposedly simple date and time questions can often lead to very tricky problems—with or without Oracle.

One of the key aspects of this chapter is to get you thinking about your own date and time problems and queries in a new light, and to spark your imagination as to how Oracle’s many and varied date and time functions and tools can solve some of your trickiest chronological problems.

6-1. Converting Datetime Values into Readable Strings

Problem

You want to convert the short dates provided by Oracle’s default date format into a full date for inclusion in a form letter or report.
Solution

Use Oracle’s `TO_CHAR` function to format date, time, and other temporal data in the desired format. The `TO_CHAR` function supports a wide range of date-specific formatting options that handle day and month names, number notation and formatting, and similar tasks. Our recipe assumes you want to display the date on which employees were hired in a fully descriptive format. Specifically, you’d want 1/1/2009 reported as Thursday, January 1st, 2009.

The next SQL statement performs the relevant calculations and formatting for all employees.

```sql
select first_name, last_name,
       to_char(hire_date, 'Day, Month DDTH, YYYY') formatted_hire_date
from hr.employees;
```

Results from our `TO_CHAR` formatting appear with the correct day name, month name and other desired formatting in place.

<table>
<thead>
<tr>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
<th>FORMATTED HIRE_DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Donald</td>
<td>OConnell</td>
<td>Monday, June 21ST, 1999</td>
</tr>
<tr>
<td>Douglas</td>
<td>Grant</td>
<td>Thursday, January 13TH, 2000</td>
</tr>
<tr>
<td>Jennifer</td>
<td>Whalen</td>
<td>Thursday, September 17TH, 1987</td>
</tr>
<tr>
<td>Michael</td>
<td>Hartstein</td>
<td>Saturday, February 17TH, 1996</td>
</tr>
<tr>
<td>Pat</td>
<td>Fay</td>
<td>Sunday, August 17TH, 1997</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How It Works

Our recipe uses a very straightforward structure to select the `FIRST_NAME` and `LAST_NAME` values from the `HR.EMPLOYEE` table. When selecting the `HIRE_DATE`, we employ the `TO_CHAR` function with specific formatting codes to change the default format of the data, instructing Oracle to emit the full day and month name, ordinal date number, and full year.

The general format of the `TO_CHAR` function for converting dates takes this form.

```
to_char(<date/time expression>, '<format options>')
```

In our recipe, we used several options to format the day. We led with the `Day` option, which returns the full day name, such as Monday. We also use `DD`, together with the ordinal formatting code `TH`, to return the ordinal day number, such as 21st and 17th. The `MONTH` code returns the full month name, such as January. The year is returned by the `YYYY` code, which instructs Oracle to return the full four-digit year. We assume a future upgrade of Oracle will support five `Y` formatting characters here at some point before the year 10000.

Oracle includes numerous other day, month, year, and other formatting options to return date and time information in digits, words, and regionally sensitive strings (although the key words that are used in formatting commands are always in English). There are five more formatting characters in the string, though they may not be immediately obvious. The two comma characters and three spaces are themselves included as format options. These, together with other common punctuation such as dashes and colons, are all valid options in the `TO_CHAR` format string.