Database Programming

Date Functions

ACADEMY

Objectives

This lesson covers the following objectives:

- Select and apply the single-row functions MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, and TRUNC that operate on date data
- Explain how date functions transform dates into date data or numeric values
- Demonstrate proper use of arithmetic operators with dates

Objectives (cont.)

This lesson covers the following objectives:

- Demonstrate the use of SYSDATE and date functions
- State the implications for world businesses to be able to easily manipulate data stored in date format

Purpose

Have you ever wondered how many days remain in the school year or how many weeks there are until graduation? Because the Oracle database stores dates as numbers, you can easily perform calculations on dates using addition, subtraction, and other mathematical operators.

Businesses depend on being able to use date functions to schedule payrolls and payments, track employee performance reviews and years of service, or keep track of orders and shipments. All of these business needs are easily handled using simple SQL date functions.

Displaying Dates

The default display format for dates is

`DD/MON/YYYY` for example: **`02/DEC/2012`**.

However, the Oracle database stores dates internally with a numeric format representing the century, year, month, day, hour, minute, and second.

Displaying Dates (cont.)

The default display and input format for any date is **`DD/MON/YYYY`**. Valid Oracle dates are between January 1, 4712 B.C., and December 31, 9999 A.D. This represents the range of dates that you can store successfully in an Oracle database.

SYSDATE

When a record with a date column is inserted into a table, the century information is picked up from the SYSDATE function. SYSDATE is a date function that returns the current database server date and time.

SYSDATE

To display the current date, use the DUAL table.

```
SELECT SYSDATE
FROM DUAL;
```

DATE Data Type

The DATE data type always stores year information as a four-digit number internally: two digits for the century and two digits for the year. For example, the Oracle database stores the year as 1996 or 2004, not just as 96 or 04.

In previous versions, the century component was not displayed by default. However, due to changing business requirements around the world, the 4-digit year is now the default display.

Working with Dates

```
SELECT last_name, hire_date + 60
FROM employees;

SELECT last_name, (SYSDATE – hire_date)/7
FROM employees;

SELECT employee_id, (end_date – start_date)/365 AS “Tenure in last
job”
FROM job_history;
```

Date Functions

The date functions shown in the table operate on Oracle dates. All of the date functions return a value with a DATE data type except the MONTHS_BETWEEN function, which returns a numeric data type value.

Function	Description
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Next day of the date specified
LAST_DAY	Last day of the month
ROUND	Round date
TRUNC	Truncate date

Date Functions (cont.)

The following query shows how the date functions are used.

Date Functions

Function	Description
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Next day of the date specified
LAST_DAY	Last day of the month
ROUND	Round date
TRUNC	Truncate date

```
SELECT employee_id, hire_date,
ROUND(MONTHS_BETWEEN(SYSDATE, hire_date)) AS TENURE,
ADD_MONTHS (hire_date, 6) AS REVIEW,
NEXT_DAY(hire_date, 'FRIDAY'),
LAST_DAY(hire_date)
FROM employees
WHERE MONTHS_BETWEEN (SYSDATE, hire_date) > 36;
```

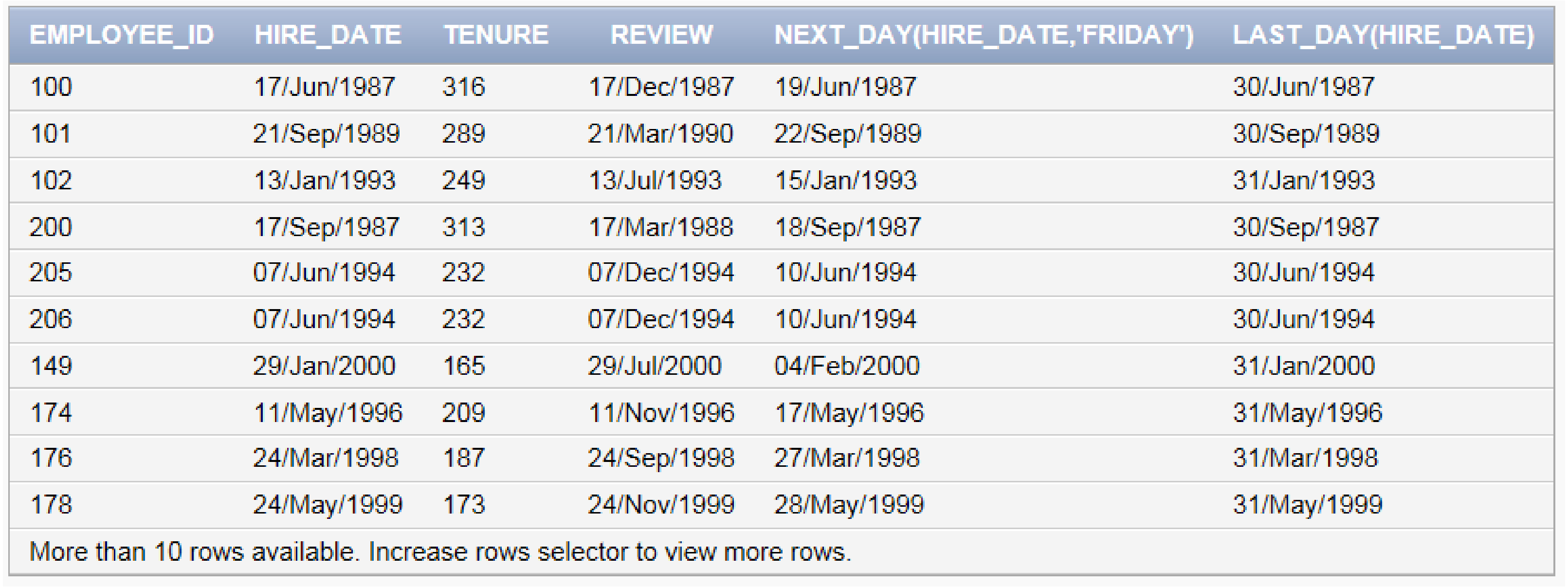
Date Functions (cont.)

Here is another example of a query using multiple date functions.

```
SELECT employee_id, hire_date,
ROUND(MONTHS_BETWEEN(SYSDATE, hire_date)) AS TENURE,
ADD_MONTHS (hire_date, 6) AS REVIEW,
NEXT_DAY(hire_date, 'FRIDAY'),
LAST_DAY(hire_date)
FROM employees
WHERE MONTHS_BETWEEN (SYSDATE, hire_date) > 36;
```

Date Functions (cont.)

The result set from this query returns 20 rows including:



EMPLOYEE_ID	HIRE_DATE	TENURE	REVIEW	NEXT_DAY(HIRE_DATE,'FRIDAY')	LAST_DAY(HIRE_DATE)
100	17/Jun/1987	316	17/Dec/1987	19/Jun/1987	30/Jun/1987
101	21/Sep/1989	289	21/Mar/1990	22/Sep/1989	30/Sep/1989
102	13/Jan/1993	249	13/Jul/1993	15/Jan/1993	31/Jan/1993
200	17/Sep/1987	313	17/Mar/1988	18/Sep/1987	30/Sep/1987
205	07/Jun/1994	232	07/Dec/1994	10/Jun/1994	30/Jun/1994
206	07/Jun/1994	232	07/Dec/1994	10/Jun/1994	30/Jun/1994
149	29/Jan/2000	165	29/Jul/2000	04/Feb/2000	31/Jan/2000
174	11/May/1996	209	11/Nov/1996	17/May/1996	31/May/1996
176	24/Mar/1998	187	24/Sep/1998	27/Mar/1998	31/Mar/1998
178	24/May/1999	173	24/Nov/1999	28/May/1999	31/May/1999

More than 10 rows available. Increase rows selector to view more rows.

Date Functions (cont.)

Below are the results from queries using ROUND and TRUNC date functions with SYSDATE (assume **`SYSDATE = '23/Oct/2013'`**).

Function	Result
`ROUND (SYSDATE, 'MONTH')`	`01/Nov/2013`
`ROUND (SYSDATE, 'YEAR')`	`01/Jan/2014`
`TRUNC (SYSDATE, 'MONTH')`	`01/Oct/2013`
`TRUNC (SYSDATE, 'YEAR’)`	`01/Jan/2013`

Terminology

Key terms used in this lesson included:

- ADD_MONTHS
- LAST_DAY
- MONTHS_BETWEEN
- NEXT_DAY
- SYSDATE

Summary

In this lesson, you should have learned how to:

- Select and apply the single-row functions MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, and TRUNC that operate on date data
- Explain how date functions transform Oracle dates into date data or numeric values
- Demonstrate proper use of the arithmetic operators with dates

Summary (cont.)

In this lesson, you should have learned how to:

- Demonstrate the use of SYSDATE and date functions
- State the implications for world businesses to be able to easily manipulate data stored in date format