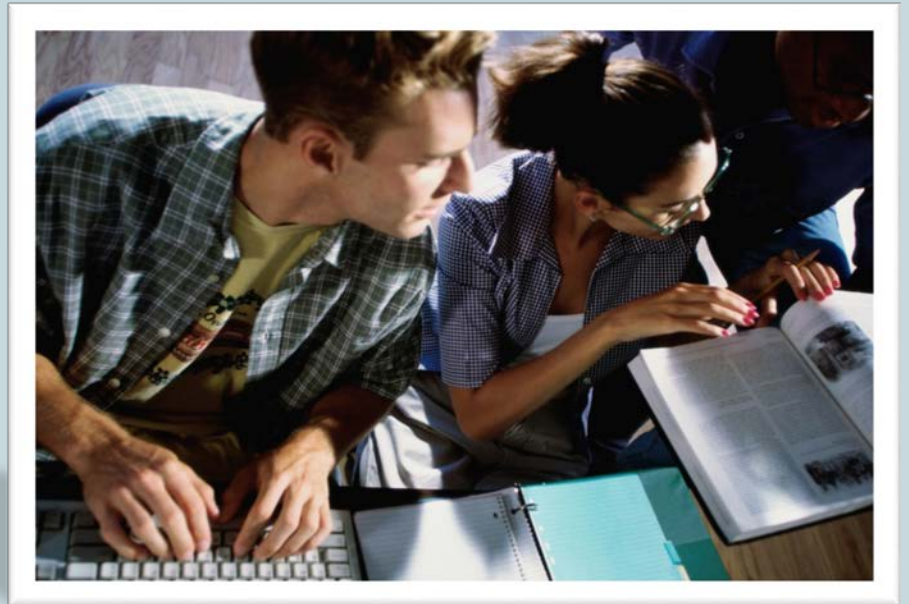




Programación de bases de datos con SQL

4-3

Funciones de Fecha



Objetivos

En esta lección se abordan los siguientes objetivos:

- Mostrar el uso de SYSDATE y las funciones de fecha
- Establecer las consecuencias de negocio mundial a fin de poder manipular fácilmente los datos almacenados en formato de fecha

Objetivo

- ¿Se ha preguntado alguna vez cuántos días quedan del año escolar o el número de semanas que falta hasta la graduación?
- Debido a que la base de datos de Oracle almacena las fechas como números, puede realizar cálculos en las fechas con los operadores matemáticos de suma y resta, entre otros.
- Las empresas dependen de poder utilizar funciones de fecha para planificar las nóminas y los pagos, realizar un seguimiento de las revisiones de rendimiento de los empleados y años de servicio, o realizar un seguimiento de los pedidos y envíos.
- Todas estas necesidades de negocio se realizan fácilmente mediante sencillas funciones de fecha de SQL.



Visualización de Fechas

- La visualización por defecto y el formato de entrada de las fechas es DD-Mon-YYYY.
- Por ejemplo: 02-Dec-2014.
- Sin embargo, las bases de datos Oracle almacena fechas de modo interno en un formato numérico que representa el siglo, el año, el mes, el día, la hora, el minuto y el segundo.
- Las fechas de Oracle válidas son del 1 de enero de 4712 A.C. y el 31 de diciembre de 9999 D.C.
- Esto representa el rango de fechas que puede almacenar correctamente en una base de datos Oracle.

SYSDATE

- SYSDATE es una función de fecha que devuelve la fecha y hora actuales del servidor de base de datos.
- Utilice SYSDATE para mostrar la fecha actual, utilice la tabla DUAL.

```
SELECT SYSDATE  
FROM dual;
```

SYSDATE
01-Jul-2015

Tipo de Dato DATE

- El tipo de dato DATE siempre almacena internamente la información del año como un número de cuatro dígitos: dos dígitos para el siglo y dos dígitos para el año.
- Por ejemplo, Oracle Database almacena el año como 1996 o 2004, no solo como 96 o 04.
- En las versiones anteriores, el componente de siglo no se mostraba por defecto.
- Sin embargo, debido a las cambiantes necesidades de los negocios en todo el mundo, el año de 4 dígitos es ahora la visualización por defecto.

Trabajar con fechas

Ejemplos:	resultado
<pre>SELECT last_name, hire_date + 60 FROM employees;</pre>	Agrega 60 días a hire_date.
<pre>SELECT last_name, (SYSDATE - hire_date)/7 FROM employees;</pre>	Muestra el número de semanas desde que se contrató al empleado.
<pre>SELECT employee_id, (end_date - start_date)/365 AS "Tenure in last job" FROM job_history;</pre>	Busca el número de días que mantuvo trabajo un empleado y, a continuación, lo divide entre 365 para mostrarlo en años.

Funciones de Fecha

- Las funciones de fecha que se muestran en la tabla funcionan en fechas de Oracle.
- Todas las funciones de fecha devuelven un valor del tipo de dato DATE excepto la función MONTHS_BETWEEN, que devuelve un valor de tipo de dato numérico.

Función	Descripción
MONTHS_BETWEEN	Número de meses entre dos fechas
ADD_MONTHS	Agregar meses de calendario a fecha
NEXT_DAY	Fecha de la siguiente incidencia del día de la semana especificado
LAST_DAY	Último día del mes
ROUND	Redondear fecha
TRUNC	Truncar fecha

Funciones de Fecha

- MONTHS_BETWEEN: toma 2 argumentos DATE y devuelve el número de meses de calendario entre las 2 fechas.
- Si el primer argumento es una fecha anterior a la segunda, el número devuelto es negativo.

Ejemplos de Función de Fecha:	Resultado	
<pre>SELECT last_name, hire_date FROM employees WHERE MONTHS_BETWEEN (SYSDATE, hire_date)>240;</pre>	King	17-Jun-1987
	Kochhar	21-Sep-1989
	De Haan	13-Jan-1993

Funciones de Fecha

- **ADD_MONTHS:** toma 2 argumentos, una fecha y un número. Devuelve un valor de fecha con el argumento numérico agregado al componente mensual de la fecha.
- Si el número proporcionado es negativo, la función restará ese número de meses del argumento de fecha.

Ejemplos de Función de Fecha:	Resultado
<pre>SELECT ADD_MONTHS (SYSDATE, 12) AS "Next Year" FROM dual;</pre>	01-Jul-2016

Funciones de Fecha

- **NEXT_DAY**: toma 2 argumentos, una fecha y un día de la semana y devuelve la fecha de la siguiente incidencia de ese día de la semana después del argumento DATE.

Ejemplos de Función de Fecha:	Resultado
<pre>SELECT NEXT_DAY (SYSDATE, 'Saturday') AS "Next Saturday" FROM dual;</pre>	04-Jul-2015

Funciones de Fecha

- **LAST_DAY**: toma un argumento DATE y devuelve la fecha del último día del mes del argumento DATE.

Ejemplos de Función de Fecha:	Resultado
<pre>SELECT LAST_DAY (SYSDATE) AS "End of the Month" FROM dual;</pre>	31-Jul-2015

Funciones de Fecha

- ROUND: devuelve una fecha redondeada a la unidad especificada en el segundo argumento.

Ejemplos de Función de Fecha:	Resultado	
<pre>SELECT hire_date, ROUND (hire_date, 'Month') FROM employees WHERE department_id=50;</pre>	16-Nov-1999	01-Dec-1999
	17-Oct-1995	01-Nov-1995
	29-Jan-1997	01-Feb-1997

<pre>SELECT hire_date, ROUND (hire_date, 'Year') FROM employees WHERE department_id=50;</pre>	16-Nov-1999	01-Jan-2000
	17-Oct-1995	01-Jan-1996
	29-Jan-1997	01-Jan-1997

Funciones de Fecha

- TRUNC: devuelve una fecha truncada a la unidad especificada en el segundo argumento.

Ejemplos de Función de Fecha:	Resultado	
<pre>SELECT hire_date, TRUNC(hire_date, 'Month') FROM employees WHERE department_id=50;</pre>	16-Nov-1999 17-Oct-1995 29-Jan-1997 ...	01-Nov-1999 01-Oct-1995 01-Jan-1997 ...
<pre>SELECT hire_date, TRUNC(hire_date, 'Year') FROM employees WHERE department_id=50;</pre>	16-Nov-1999 17-Oct-1995 29-Jan-1997 ...	01-Jan-1999 01-Jan-1995 01-Jan-1997 ...

Funciones de Fecha

- A continuación se muestra un ejemplo de una consulta con varias funciones de fecha.
- La salida se muestra en la siguiente diapositiva.

```
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;
```


Funciones de Fecha

- El juego de resultados de esta consulta devuelve 20 filas en las que se incluye:

EMPLOYEE_ID	HIRE_DATE	TENURE	REVIEW	NEXT_DAY(HIRE_DATE,'FRIDAY')	LAST_DAY(HIRE_DATE)
100	17-Jun-1987	348	17-Dec-1987	19-Jun-1987	30-Jun-1987
101	21-Sep-1989	321	21-Mar-1990	22-Sep-1989	30-Sep-1989
102	13-Jan-1993	281	13-Jul-1993	15-Jan-1993	31-Jan-1993
200	17-Sep-1987	345	17-Mar-1988	18-Sep-1987	30-Sep-1987
205	07-Jun-1994	265	07-Dec-1994	10-Jun-1994	30-Jun-1994
206	07-Jun-1994	265	07-Dec-1994	10-Jun-1994	30-Jun-1994
149	29-Jan-2000	197	29-Jul-2000	04-Feb-2000	31-Jan-2000
174	11-May-1996	241	11-Nov-1996	17-May-1996	31-May-1996
176	24-Mar-1998	219	24-Sep-1998	27-Mar-1998	31-Mar-1998
178	24-May-1999	205	24-Nov-1999	28-May-1999	31-May-1999
More than 10 rows available. Increase rows selector to view more rows.					

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- ADD_MONTHS
- LAST_DAY
- MONTHS_BETWEEN
- NEXT_DAY
- SYSDATE
- ROUND
- TRUNC



Resumen

En esta lección, debe haber aprendido lo siguiente:

- Seleccionar y aplicar las funciones de una sola fila MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND y TRUNC que funcionan en los datos de fecha
- Explicar cómo transforman las funciones de fecha las fechas de Oracle en datos de fecha o en valores numéricos
- Mostrar un uso adecuado de los operadores aritméticos con fechas

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Mostrar el uso de SYSDATE y las funciones de fecha
- Establecer las consecuencias de negocio mundial a fin de poder manipular fácilmente los datos almacenados en formato de fecha

