

第 5 章 日期时间型数据的处理

日期时间是数据库中常用的数据类型，Oracle 和 SQL Server 有多种日期时间类型，各自有不同的范围和精度。与普通的数值型和字符型数据相比，其处理方式相对较为复杂，Oracle 和 SQL Server 提供了多个函数操作日期时间数据。

本章主要包括：

- 主要日期时间类型
- 处理日期时间常量
- 日期时间处理函数

5.1 日期时间类型

关于日期时间数据，SQL:2011 标准中规定了两种数据类型，一种表示日期时间值，另一种表示日期时间间隔。

前一类数据类型包括三种：

- **date**：以年、月、日表示的日期值。
- **time**：以小时、分钟、秒表示的时间值。
- **timestamp**：以年、月、日、小时、分钟、秒表示的日期日期值。

time 和 **timestamp** 类型可以包括时区，也可以包括一定精度的、以小数表示的秒。下面的介绍，我们只涉及常用的不包括时区的情形。

第二类包括：

- **year-month** 间隔：以年、月之一或两者表示间隔。
- **day-time** 间隔：以年月之外的单位表示间隔。

5.1.1 Oracle 中的日期时间类型

Oracle 中的日期时间类型包括 **date**、**timestamp** 以及表示时间间隔的类型 **interval year to month** 和 **interval day to second**。**date** 与 **timestamp** 对应 SQL:2011 中的 **timestamp** 类型。

与 SQL:2011 中的 **date** 类型不同，Oracle 中的 **date** 类型除了日期外，也包括时间信息。而 Oracle 的 **timestamp** 与 SQL:2011 的 **timestamp** 类型基本相同。

Oracle 中的 **date** 和 **timestamp** 类型的主要属性如下表所示：

表 5-1 Oracle 的主要日期时间类型

数据类型	格式	范围	精确度
date	yyyy-mm-dd hh:mi:ss	-4712-01-01 00:00:00~9999-12-31 23:59:59	秒
timestamp(p)	yyyy-mm-dd hh:mi:ss.	-4712-01-01 00:00:00~9999-12-31 23:59:59.9[p]	与 p 相关

yyyy 表示四位年份，mm 表示两位月份，dd 表示两位日期，hh 表示两位小时，mi 表示两位分钟，ss 表示两位秒。p 表示秒的小数位数，其范围为 0 到 9，默认为 6。-4712 表示公元前 4712 年。

Oracle 的 **date** 与 **timestamp** 类型的区别在于精度不同，**date** 类型只精确到秒，另外 **timestamp** 类型可以包括时区信息，即 **timestamp with time zone** 和 **timestamp with local time zone**。

5.1.2 SQL Server 中的日期时间类型

SQL Server 支持的所有日期时间类型如下表所示：

表 5-2 SQL Server 的主要日期时间类型

数据类型	格式	范围	精确度
time(p)	hh:mm:ss[.nnnnnnn]	00:00:00.0000000~23:59:59.9999999	100 纳秒
date	yyyy-mm-dd	0001-01-01~9999-12-31	天
smalldatetime	yyyy-mm-dd hh:mm:ss	1900-01-01 00:00:00~2079-06-06 23:59:59	分钟
datetime	yyyy-mm-dd hh:mm:ss[.nnn]	1753-01-01 00:00:00~9999-12-31 23:59:59.997	0.00333 秒
datetime2(p)	yyyy-mm-dd hh:mm:ss[.nnnnnnn]	0001-01-01 00:00:00~9999-12-31 23:59:59.9999999	100 纳秒
datetimeoffset	yyyy-mm-dd hh:mm:ss[.nnnnnnn][+ -]hh:mm	0001-01-01 00:00:00~9999-12-31 23:59:59.9999999	100 纳秒

yyyy 表示四位年份，mm 表示两位月份或两位分钟(Oracle 使用 mi 表示两位分钟)，dd 表示两位日期，hh 表示两位小时，ss 表示两位秒。p 表示秒的小数位数，其范围为 0 到 7，默认为 7。

SQL Server 2008 之前的版本只支持以上 6 种类型中的 datetime 和 smalldatetime，SQL Server 2008 增加了另外 4 种。SQL Server 建议在描述日期时间数据时，使用 4 种新类型，以与 SQL 标准更好地兼容。

date 类型对应 SQL:2011 标准的 date 类型，time 类型对应 SQL:2011 的无时区 time 类型，datetime 和 datetime2 和对应 SQL:2011 标准的无时区 timestamp 类型，datetimeoffset 类型对应 SQL:2011 标准的有时区 timestamp 类型。

datetime 的小数秒精度精确到三百分之一秒（相当于 3.33 毫秒或 0.00333 秒），值舍入到 .000、.003 或 .007 秒三个增量。datetime2 是 SQL Server 2008 引入的新类型，比 datetime 表示的范围更大，精度更高，用以代替 datetime 类型。

datetimeoffset 类型的时区值从-14:00 至 +14:00。此类型常量值在日期时间值后面附加时区信息，如 1912-10-25 12:24:32.000 +10:00。

另外要注意，虽然 SQL Server 也有 timestamp 类型，但它是 rowversion 的同义词，与日期时间类型无关。

5.2 处理日期时间常量

日期时间常量主要用于以下几个方面：

- select、update、delete 语句中的 where 条件中涉及日期时间值
- insert 语句添加日期时间字段值
- update 语句修改日期时间字段值

使用日期时间常量主要是下面两种方法：

- 使用指定格式的字符串常量
- 把指定格式的字符串常量使用数据类型转换函数转换为日期时间型数据

另外，年份可以使用四位，也可以使用两位，为了避免引起混淆(如 50 可以表示 2050 年，也可以表示 1950 年)，本书统一使用四位。

5.2.1 Oracle 的情形

在 Oracle 中使用日期时间常量，主要有两种方式。

一是用 date 或 timestamp 关键字标识字符串，若只包含年月日数据则使用 date，若除日

期外还包括时间数据，则使用 `timestamp` 关键字。

二是用 `to_date()`或 `to_timestamp()`函数把指定格式的字符串转换为 `date` 或 `timestamp` 类型常量。两种方式不受操作系统和数据库相关参数设置的影响。

使用 `date` 和 `timestamp` 关键字时，日期时间的各个部分要遵从固定顺序。

- 日期型格式: `date 'yyyy-mm-dd'`
- 日期时间格式: `timpstamp 'yyyy-mm-dd hh24:mi:ss.fff '`

下面示例演示其具体用法。

创建测试表:

```
SQL> create table t(a int, b date);
```

添加日期型数据:

```
SQL> insert into t values(1, date'2016-03-19');
```

添加日期时间型数据:

```
SQL> insert into t values(2, timestamp'2016-03-19 16:28:00');
```

执行查询，以上两行数据显示为:

```
SQL> select a,to_char(b, 'yyyy-mm-dd hh24:mi:ss') as "B" from t;
```

A	B
1	2016-03-19 00:00:00
2	2016-03-19 16:28:00

在 `update` 语句中使用日期型常量:

```
SQL> update t set b=date'2016-03-20' where a=1;
```

`to_date()`和 `to_timestamp()`函数的用法请参考 5.3 节相关内容。

5.2.2 SQL Server 的情形

涉及日期时间常量时，SQL Server 建议使用与语言环境设置无关的标准字符串格式，一方面省去了使用类型转换函数的繁琐，另一方面也可以避免引起混淆。使用标准形式时，不需要像 Oracle 一样附加 `date` 或 `timestamp` 前缀。

- 日期型格式: `'yyyy-mm-dd'`
- 时间型格式: `'hh:mm:ss[.nnnnnnnn]'`
- 日期时间型格式: `' yyyy-mm-dd hh:mm:ss[.nnnnnnnn] '`

日期之间的分隔符可以省略，日期和时间之间也可以用 `T` 分隔。下面通过几个实例说明其用法。

创建测试表 t:

```
1> create table t(a int, b datetime)
2> go
```

添加日期型数据:

```
1> insert into t
2> values(1, '2016-03-19')
3> go
```

添加日期时间型数据:

```

1> insert into t
2> values(2, '2016-03-19 19:23:10')
3> go

```

5.2.3 设置默认日期时间环境参数

Oracle 和 SQL Server 分别用 `nls_date_format` 和 `dateformat` 参数设置 SQL*Plus 和 `sqlcmd` 使用的默认日期时间格式。

Oracle 的 SQL*Plus 客户端使用 `nls_date_format` 参数规定字符串转换为日期时间常量、以及显示日期时间值时的顺序，这样使用字符串表示日期时间常量时，不需再附加 `date` 或 `timestamp` 关键字。

执行下面命令设置 `nls_date_format` 为 'mm-dd-yyyy hh24:mi:ss'：

```
SQL> alter session set nls_date_format='mm-dd-yyyy hh24:mi:ss';
```

按照上述指定顺序(即月日年)添加两行记录,此时不需要附加 `timestamp` 及 `date` 关键字：

```

SQL> insert into t values(3, '03-20-2016 11:33:10');
已创建 1 行。
SQL> insert into t values(4, '03-21-2016');
已创建 1 行。

```

查询 `t` 表，日期时间的显示格式也符合 `nls_date_format` 参数指定的顺序：

```
SQL> select * from t where a>=3;
```

	A	B
3	03-20-2016	11:33:10
4	03-21-2016	00:00:00

在 SQL Server 的客户端中，`dateformat` 参数只规定字符串转换为日期型数据时，日期的年月日默认顺序，并不影响其中的时间部分。另外，`dateformat` 参数也不影响查询语句中的日期时间值显示格式，这与 Oracle 的 `nls_date_format` 参数的功能不同。

设置 `dateformat` 可以使用 `set` 命令，有效值为 `mdy`、`dmy`、`ymd`、`ydm`、`myd` 和 `dym`，`m`、`d`、`y` 三个字母分别表示年月日。

下面命令把 `dateformat` 设置为 `dmy`：

```

1> set dateformat dmy
2> go

```

继续使用上节的 `t` 表，对其添加一行记录，`b` 列只包括日期部分：

```

1> insert into t values(3, '03-19-2016')
2> go

```

再添加一行，`b` 列也可以包含时间数据：

```

1> insert into t values(4, '03-20-2016 11:58:30')
2> go

```

执行查询，可以发现 `b` 列的显示格式未受影响：

```

1> select * from t where a>=3
2> go
a          b
-----
3 2016-03-19 00:00:00.000

```

而如果字符串常量不符合 `dateformat`，则会报错：

```
1> insert into t values(5, '20-03-2016')
2> go
消息 242, 级别 16, 状态 3, 服务器 LAW_X240, 第 1 行
从 varchar 数据类型到 datetime 数据类型的转换产生一个超出范围的值。
语句已终止。
```

日期各个部分的分隔符可以为斜杠 “/”、连字符 “-” 或句点 “.”：

```
1> insert into t values(5, '20-03-2016')
2> insert into t values(6, '20/03/2016')
3> insert into t values(7, '20.03.2016')
4> go
```

如果使用上节内容中的标准日期格式，则各个日期部分是固定的，不受 `dateformat` 参数设置的影响，请读者自行验证，这里不再赘述。

5.3 日期时间处理函数

Oracle 和 SQL Server 提供了丰富的函数处理日期时间数据，本节讲述常用日期时间处理函数的用法。

下面列表是 Oracle 和 SQL Server 的日期时间处理函数的一个总体对比。

功能	Oracle	SQL Server
字符串转换为日期时间值	<code>to_date()</code>	<code>cast()</code>
		<code>convert()</code>
日期时间值转换为字符串	<code>to_char</code>	<code>cast()</code>
		<code>convert()</code>
获取当前日期时间	<code>sysdate</code>	<code>getdate()</code>
	<code>localtimestamp(p)</code>	<code>sysdatetime()</code>
	<code>current_timestamp(p)</code>	<code>sysdatetimeoffset()</code>
	<code>systimestamp</code>	
抽取日期时间指定部分	<code>extract()</code>	<code>datepart()</code>
获得指定日期所在月份的最后一天	<code>last_day()</code>	<code>eomonth()</code>

5.3.1 类型转换函数

类型转换函数的功能主要是日期型数据和字符串数据的相互转换。

Oracle 使用 `to_date()` 和 `to_timestamp()` 函数把字符串转换为日期型常量，使用 `to_char()` 函数把日期型常量转换为字符串。

Oracle 的 `to_date()` 函数把字符串常量转换为 `date` 类型的值。

- `to_date()` 函数的用法为：`to_date (string, format)`

string 是要转换的字符串，*format* 是格式码，用于指定第一个参数中的年月日时等各部分。格式码中的日期时间各部分的分隔符要与字符串中的分隔符一致，常用的分隔符一般为 “-” 或 “/”。格式码用 `yyyy` 表示 4 位年份，`mm` 表示 2 位月份，`dd` 表示 2 位日期，`hh24` 表示两位小时，`mi` 表示两位分钟，`ss` 表示两位秒。

下面示例对 t 表添加一行记录，其日期型字段的值为“2016 年 3 月 19 日”：

```
SQL> insert into t values(5, to_date('2016-03-19', 'yyyy-mm-dd'));
```

下面示例使用另外一种分隔符添加同一个日期时间值：

```
SQL> insert into t values(6, to_date('2016/03/19 20:35:10', 'yyyy/mm/dd hh24:mi:ss'));
```

- to_timestamp()函数的用法如下：to_timestamp(string, format)

to_timestamp()函数与 to_date()函数用法相似，只是秒的精度更高，格式码部分使用 ff 表示秒的小数部分，格式码其他部分与 to_date()函数用法相同。

下面示例重建 t 表：

```
SQL> create table t(a int, b timestamp);
```

然后对其添加一行记录：

```
SQL> insert into t values
2  (1, to_timestamp('2016-03-19 15:30:10.12300', 'yyyy-mm-dd hh24:mi:ss.ff'))
3  /
```

Oracle 的 to_char()函数功能是把日期时间常量转换为指定格式的字符串。

- to_char()函数的用法如下：to_char(date, format)

date 是要转换的日期型常量，format 是指定的格式码，其用法与 to_date()函数相同。

下面查询以“-”为分隔符分隔日期部分，以“:”为分隔符分隔时间部分，日期和时间之间以空格分隔：

```
SQL> select a, to_char(b, 'yyyy-mm-dd hh24:mi:ss') as b
2  from t
3  where a=6
4  /
```

A	B
6	2016-03-19 20:35:10

下面的查询只要求显示 hiredate 中的年和月，并且以/分隔：

```
SQL> select a, to_char(b, 'yyyy/mm') as b
2  from t
3  where a=6
4  /
```

A	B
6	2016/03

下面查询要求得到 hiredate 中的年月日，并且以汉字年月日分隔：

```
SQL> select a, to_char(b, 'yyyy') || '年'
2          || to_char(b, 'mm') || '月'
3          || to_char(b, 'dd') || '日'
4  as b
5  from t
6  where a=6
7  /
```

A	B
---	---

SQL Server 的 `cast()`和 `convert()`函数可以实现两种类型数据的相互转换，我们这里只介绍与 Oracle 的 `to_date()`和 `to_char()`函数用法相似的 `convert()`函数。

使用 `convert` 函数把字符串转换为日期型数据的用法为：

`convert(datetime, str, fmt)`

其中第一个参数 `datetime` 为类型名称，用于指定把第二个字符串参数转换为日期型数据，第三个参数为格式码，用于指定第二个参数中年月日的各部分。

若转换的字符串包含日期和时间，使用格式码 120：

```
1> insert into t values(8, convert(datetime, '2016-03-31 15:41:00', 120))
2> go
```

若转换的字符串只包括日期，使用格式码 101, 103, 110, 112：

```
1> insert into t values(9, convert(datetime, '03/31/2016', 101))
2> insert into t values(10, convert(datetime, '31/03/2016', 103))
3> insert into t values(11, convert(datetime, '03-31-2016', 110))
4> insert into t values(12, convert(datetime, '20160331', 112))
5> go
```

日期各部分的分隔符可以使用“-”，“/”或“.”，时间各部分的分隔符使用“:”。

若使用日期或时间的标准字符串格式，可以省略格式码，附加格式码也会失效，如下面示例所示：

```
1> insert into t values(13, convert(datetime, '2016-04-04'))
2> insert into t values(14, convert(datetime, '2016-04-04', 103))
3> insert into t values(15, convert(datetime, '2016-04-04', 110))
4> insert into t values(16, convert(datetime, '2016-04-04', 120))
5> insert into t values(17, convert(time, '16:05:00', 103))
6> insert into t values(18, convert(time, '16:05:00', 108))
7> insert into t values(19, convert(time, '16:05:00', 120))
8> go
```

SQL Server 使用 `convert()`函数除了完成与 Oracle 的 `to_date()`函数类似的功能以外，也可以完成与 Oracle 的 `to_char()`函数类似的功能。

`convert()`函数把日期型数据转换为字符串的用法为：

`convert(varchar, date, str_format)`

第一个参数为转换的目标类型 `varchar`，第二个参数是一个日期型表达式，第三个参数是表示格式码的整数，与 Oracle 的 `to_char()`函数指定格式码的方式不同。

最常用的格式码为 120，其效果相当于 Oracle 中的 'yyyy-mm-dd hh24:mi:ss' 的形式：

```
1> select a, convert(varchar, b, 120)
2> from t
3> where a=8
4> go
a
-----
      8 2016-03-31 15:41:00
```

下面几个示例演示其他几种格式码的效果。

使用 101~105，只包括日期部分：

```
1> select a,
2> convert(varchar, b, 101) as "101",
3> convert(varchar, b, 102) as "102",
```

```

4> convert(varchar, b, 103) as "103",
5> convert(varchar, b, 104) as "104",
6> convert(varchar, b, 105) as "105"
7> from t
8> where a=8
9> go
a          101          102          103          104          105
-----
8 03/31/2016    2016. 03. 31    31/03/2016    31. 03. 2016    31-03-2016

```

使用 109、113、120 包括日期和时间两部分：

```

1> select a,
2> convert(varchar, b, 109) as "109",
3> convert(varchar, b, 113) as "113",
4> convert(varchar, b, 120) as "120"
5> from t
6> where a=8
7> go
a          109          113          120
-----
8 03 31 2016    3:41:00:000PM 31 03 2016 15:41:00:000    2016-03-31 15:41:00

```

使用 108 和 114 只包括时间部分：

```

1> select a,
2> convert(varchar, b, 108) as "108",
3> convert(varchar, b, 114) as "114"
4> from t
5> where a=8
6> go
a          108          114
-----
8 15:41:00    15:41:00:000

```

Oracle 与 SQL Server 常用的日期时间格式码对比，请参考下表：

表 5-3 SQL Server 与 Oracle 的日期时间格式码对比

SQL Server	Oracle
101	mm/dd/yyyy
103	dd/mm/yyyy
110	mm-dd-yyyy
112	yyyymmdd
120	yyyy-mm-dd
	hh24:mi:ss

5.3.2 获得当前日期时间

Oracle 使用 sysdate, localtimestamp(p), current_date, current_timestamp(p) 及 systimestamp 获取当前日期时间。SQL Server 使用 getdate()、current_timestamp 和 sysdatetime() 获取当前日期时间，使用 getutcdate() 和 sysutcdatetime() 返回 UTC 时间(世界标准时间，即 0 度经度的时间)，使用 sysdatetimeoffset() 返回包含时区信息的当前日期时间。current_timestamp 是 SQL 标准形式。

Oracle 的 sysdate 返回值的类型为 date，localtimestamp(p) 返回值的类型为 timestamp，参数 p 为精确度，即小数位数，默认为 6。

简单示例如下：

```
SQL> select sysdate, localtimestamp
2   from dual
3   /
```

SYSDATE	LOCALTIMESTAMP
04-4月 -16	04-4月 -16 07.15.43.498000 下午

如要指定时间格式，可以使用 to_char()函数：

```
SQL> select to_char(sysdate,'yyyy-mm-dd hh24:mi:ss') as sdate,
2   to_char(localtimestamp,'yyyy-mm-dd hh24:mi:ss.ff') as localtimestamp
3   from dual
4   /
```

SDATE	LOCALTIMESTAMP
2016-04-04 19:18:42	2016-04-04 19:18:42.719000

current_timestamp 及 systimestamp 函数的返回值类型为 timestamp with time zone，除了包括 localtimestamp 返回的结果外，还包括当前时区信息。

```
SQL> select current_timestamp, systimestamp
2   from dual
3   /
```

CURRENT_TIMESTAMP	SYSTIMESTAMP
04-4月 -16 07.23.45.317000 下午 +08:00	04-4月 -16 07.23.45.317000 下午 +08:00

current_date 的返回数据受时区影响，如下面示例所示。

设置日期时间的默认格式：

```
SQL> alter session set nls_date_format='yyyy-mm-dd hh24:mi:ss';
会话已更改。
```

查询当前时区，当前时间和 current_date：

```
SQL> select sessiontimezone,sysdate, current_date from dual;
```

SESSIONTIMEZONE	SYSDATE	CURRENT_DATE
+08:00	2016-04-04 22:28:19	2016-04-04 22:28:19

把时区设置为“-8:0”后，再次查询：

```
SQL> alter session set time_zone='-8:0';
会话已更改。
```

```
SQL> select sysdate, current_date from dual;
```

SYSDATE	CURRENT_DATE
2016-04-04 22:23:53	2016-04-04 06:23:53

恢复时区为“+8:0”后查询：

```
SQL> alter session set time_zone='+8:0';
会话已更改。
```

```
SQL> select sysdate, current_date from dual;
```

SYSDATE	CURRENT_DATE
2016-04-04 22:24:16	2016-04-04 22:24:16

SQL Server 的 `getdate()`、`current_timestamp`、`sysdatetime()`都返回系统时间，`getdate()`与 `current_timestamp` 的效果等价，后者是 SQL 标准用法(Oracle 也支持)，返回值为 `datetime` 类型。`sysdatetime()`的返回值类型为 `datetime2`，精度比前两者高。

```
1> select
2> getdate() as "getdate()",
3> current_timestamp as "current_timestamp",
4> sysdatetime() as "sysdatetime()"
5> go
```

getdate()	current_timestamp	sysdatetime()
2016-04-04 19:35:27.890	2016-04-04 19:35:27.890	2016-04-04 19:35:27.8903748

`getutcdate()`和 `sysutcdatetime()`返回 UTC 时间，后者精度更高：

```
1> select
2> getdate() as "getdate()",
3> getutcdate() as "getutcdate()",
4> sysutcdatetime() as "sysutcdatetime()"
5> go
```

getdate()	getutcdate()	sysutcdatetime()
2016-04-04 21:53:52.247	2016-04-04 13:53:52.247	2016-04-04 13:53:52.2480098

`sysdatetimeoffset()`返回包括时区信息的当前时间：

```
1> select
2> getdate() as "getdate()",
3> sysdatetimeoffset() as "sysdatetimeoffset()"
4> go
```

getdate()	sysdatetimeoffset()
2016-04-04 21:54:31.777	2016-04-04 21:54:31.7697566 +08:00

5.3.3 抽取日期时间的指定部分

Oracle 使用 `extract()`函数抽取一个日期值的年、月、日、小时、分钟、秒，SQL Server 使用 `datepart()`函数来完成相同的功能。

Oracle 的 `extract()`函数的用法为：

`extract(datepart from expr)`

其中的 `datepart` 依据用户要抽取的内容，可以为 `year`、`month`、`day`、`hour`、`minute` 以及 `second`，第二个参数为日期型表达式。

对 `date` 类型的数据使用 `extract()`函数时，Oracle 将其按照 SQL 标准中的 `date` 类型看待，而 SQL 标准中的 `date` 类型数据只包含日期，不包含时间部分，从而 `extract()`函数的第一个参数只能为 `year`、`month` 及 `day`。对于 `timestamp`、`timestamp with time zone` 以及 `timestamp with local time zone` 类型的数据，除了 `year`、`month` 和 `day` 外，`extract()`函数也支持 `hour`、`minute` 及 `second` 选项，以抽取时间的指定部分。

如查询 `emp` 表中，1981 年 5 月入职的人数，可以使用下面查询：

```
SQL> select count(*) from emp
2  where extract(year from hiredate)=1981
3  and extract(month from hiredate)=5
```

```

4 /

COUNT(*)
-----
1

```

emp 表中的 hiredate 字段是 date 类型，支持抽取其日期各部分。若对 date 类型数据抽取时间部分，如 hour，可以发现 extract() 函数不支持：

```

SQL> select extract(hour from to_date('2016-04-04 10:25:30', 'yyyy-mm-dd'))
2   from dual
3   /
select extract(hour from to_date('2016-04-04 10:25:30', 'yyyy-mm-dd'))
                                     *

```

第 1 行出现错误：

ORA-30076: 对析出来源无效的析出字段

若对 timestamp 类型的数据抽取其时间部分，则不会发生问题：

```

SQL> select extract(hour from timestamp'2016-04-04 10:25:30')
2   from dual
3   /

EXTRACT (HOURFROMTIMESTAMP'2016-04-0410:25:30')
-----
10

```

SQL Server 的 datepart() 函数用法为：

datepart(datepart, expr)

其中的 datepart 参数可以取值为 year、month、day、week、weekday、hour、minute、second、millisecond、microsecond、nanosecond 等，week 作为参数抽取的是当年的第几周，weekday 作为参数表示抽取的是当周的星期几，第二个参数为日期型表达式。

Oracle 使用 to_char 函数实现这里的 week 及 weekday 参数的功能，格式码分别取 D 及 WW，若使用单个 W 作为格式码，则表示提取当月的第几周，Oracle 的 extract 函数并未提供类似功能。

抽取年月日部分，SQL Server 还提供了 year(date)、month(date)、day(date) 三个函数，与 datepart() 函数的第一个参数指定为 year、month、day 的效果相同。

下面几个示例使用 datepart() 函数完成与上节内容相同的查询。

查询 emp 表中，1981 年 5 月入职的人数：

```

1> select count(*) from emp
2> where datepart(year, hiredate)=1981
3> and datepart(month, hiredate)=5
4> go

-----
1

```

5.3.4 获取日期和时间差

SQL Server 提供了 datediff() 函数用于得出两个日期时间值的差，Oracle 并未提供与 datediff() 功能类似的函数。

Oracle 中可以把结束时间值与开始时间值相减求出其差值，即 enddate - startdate，这里的 enddate 与 startdate 都是 date 数据类型，得出的差是两个值相差的天数，要注意这个天数是附带小数的。

如果要获得其他指定单位(如小时、分钟或秒)的时间段，则可以使用下面方法处理：

- 相差天数：round(enddate - startdate)
- 相差小时数：round((enddate - startdate)*24)
- 相差分钟数：round((enddate - startdate)*24*60)
- 相差秒数：round((enddate - startdate)*24*60*60)

round 函数用四舍五入的方法对小数取整。

下面我们用一个简单的例子说明其用法。

删除之前的 t 表后重建：

```
SQL> create table t(startdate date, enddate date);
```

然后对其添加记录：

```
SQL> insert into t values
2  (
3      to_date('2016-04-04 12:00:00','yyyy-mm-dd hh24:mi:ss'),
4      to_date('2016-04-04 12:05:30','yyyy-mm-dd hh24:mi:ss')
5  )
6  /
```

最后我们求这条记录两个字段值相差的秒数：

```
SQL> select round((enddate - startdate)*24*60*60)
2  from t
3  /
```

```
ROUND((ENDDATE-STARTDATE)*24*60*60)
-----
330
```

也可以求出其相差的分钟数：

```
SQL> select round((enddate - startdate)*24*60)
2  from t
3  /
```

```
ROUND((ENDDATE-STARTDATE)*24*60)
-----
6
```

SQL Server 使用 datediff()函数得到两个日期数据的时间差。SQL Server 的 datediff()函数的用法为：

datediff(datepart, startdate, enddate)

其中的参数含义如下：

- startdate：日期时间起始值，可以为 SQL Server 支持的任何日期时间型数据。
- enddate：日期时间结束值，可以为 SQL Server 支持的任何日期时间型数据。。
- datepart：返回的差值单位，其取值可以为 year、month、day、week、hour、minute、second、millisecond、microsecond、nanosecond 等，也可以使用其缩写形式，分别为 yy、mm 或 m、dd 或 d、wk 或 ww、hh、mi 或 n、ss 或 s、ms、mcs、ns 等。

下面我们使用使用 datediff 函数完成与上节 Oracle 功能类似的示例，参数的数据类型使用 datetime。

删除之前的 t 表后，重新创建：

```
1> create table t(startdate datetime, enddate datetime)
2> go
```

添加测试数据:

```
1> insert into t values('2016-04-04 12:00:00', '2016-04-04 12:05:30')
2> go
```

查询表中的两个字段值相差的秒数:

```
1> select datediff(second, startdate, enddate) from t
2> go

-----
330
```

查询表中的两个字段值相差分钟数:

```
1> select datediff(minute, startdate, enddate) from t
2> go

-----
5

(1 行受影响)
```

查询相差的小时数:

```
1> select datediff(hour, startdate, enddate) from t
2> go

-----
0
```

使用缩写查询相差的毫秒数:

```
1> select datediff(ms, startdate, enddate) from t
2> go

-----
330000
```

使用缩写查询相差的分钟数:

```
1> select datediff(mi, startdate, enddate) from t
2> go

-----
5
```

5.3.5 获取指定日期所在月份的最后一天

Oracle 使用 last_day()函数获取指定日期所在月份的最后一天:

```
SQL> select last_day(sysdate) as last_day_of_month from dual;

LAST_DAY_OF_MO
-----
31-8月 -16
```

SQL Server 使用 eomonth()函数获取指定日期所在月份的最后一天:

```
1> select eomonth(getdate()) as last_day_of_month
2> go
```

last_day_of_month

2016-08-31