

## 8.1

SQL File 4\* x

```
1 select invoice_total,
2     format(invoice_total, 1) as total_format,
3     convert(invoice_total, signed) as total_convert,
4     cast(invoice_total as signed) as total_cast
5 from invoices;
6
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

| invoice_total | total_format | total_convert | total_cast |
|---------------|--------------|---------------|------------|
| 3813.33       | 3,813.3      | 3813          | 3813       |
| 40.20         | 40.2         | 40            | 40         |

Result 2 x

Output

Action Output

| # | Time     | Action  | Message             |
|---|----------|---|---------------------|
| 1 | 20:45:02 | select invoice_total, format(invoice_total, 1) as total_format, convert(invoice_total, signed) as total_conv... | 115 row(s) returned |

## 9.1

SQL File 6\* x SQL File 7\* x

```
1 select invoice_total, round(invoice_total, 1) as one_digit,
2     round(invoice_total, 0) as zero_digits
3 from invoices;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

| invoice_total | one_digit | zero_digits |
|---------------|-----------|-------------|
| 3813.33       | 3813.3    | 3813        |
| 40.20         | 40.2      | 40          |

Result 1 x

Output

## 9.2

SQL File 6\* x SQL File 7\* x

```
1 select start_date,
2     date_format(start_date, '%b/%d/%y') as format1,
3     date_format(start_date, '%c/%e/%y') as format2,
4     date_format(start_date, '%l:%i %p') as twelve_hour,
5     date_format(start_date, '%c/%e/%y %l:%i %p') as format3
6 from date_sample;
```

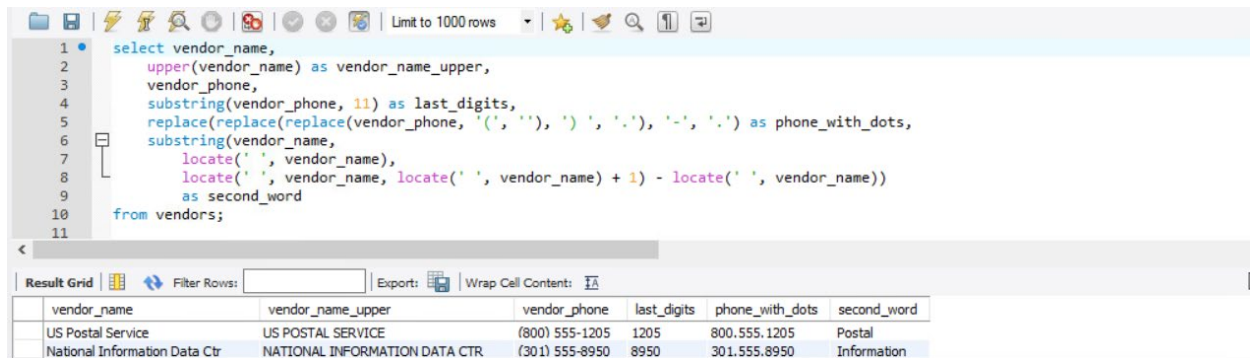
Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

| start_date          | format1   | format2 | twelve_hour | format3          |
|---------------------|-----------|---------|-------------|------------------|
| 1982-03-01 00:00:00 | Mar/01/82 | 3/1/82  | 12:00 AM    | 3/1/82 12:00 AM  |
| 2002-02-28 00:00:00 | Feb/28/02 | 2/28/02 | 12:00 AM    | 2/28/02 12:00 AM |

Result 2 x

Output

## 9.3

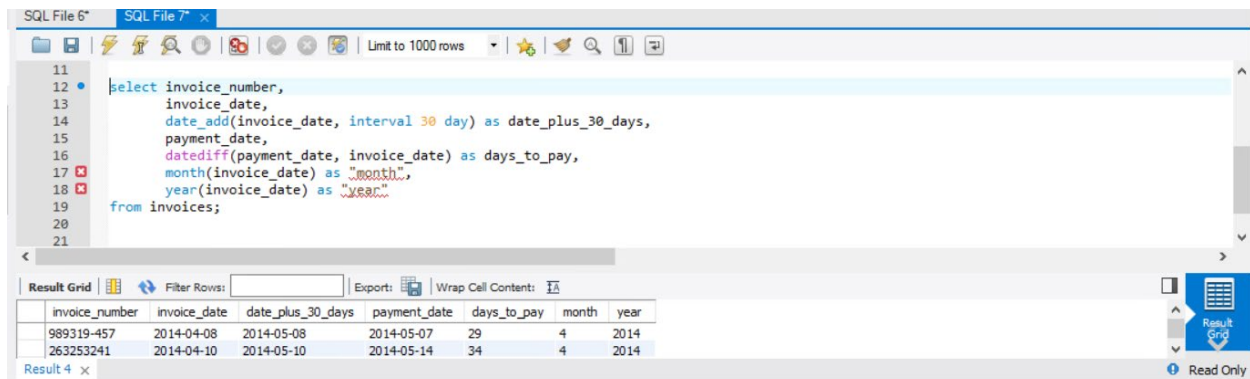


The screenshot shows a SQL query in an IDE. The query selects vendor information and formats the phone number. The results table has 6 columns: vendor\_name, vendor\_name\_upper, vendor\_phone, last\_digits, phone\_with\_dots, and second\_word. Two rows are displayed: US Postal Service and National Information Data Ctr.

```
1 select vendor_name,
2       upper(vendor_name) as vendor_name_upper,
3       vendor_phone,
4       substring(vendor_phone, 11) as last_digits,
5       replace(replace(replace(vendor_phone, '(', ''), ')', '.'), '-', '.') as phone_with_dots,
6       substring(vendor_name,
7               locate(' ', vendor_name),
8               locate(' ', vendor_name, locate(' ', vendor_name) + 1) - locate(' ', vendor_name))
9       as second_word
10  from vendors;
```

| vendor_name                   | vendor_name_upper             | vendor_phone   | last_digits | phone_with_dots | second_word |
|-------------------------------|-------------------------------|----------------|-------------|-----------------|-------------|
| US Postal Service             | US POSTAL SERVICE             | (800) 555-1205 | 1205        | 800.555.1205    | Postal      |
| National Information Data Ctr | NATIONAL INFORMATION DATA CTR | (301) 555-8950 | 8950        | 301.555.8950    | Information |

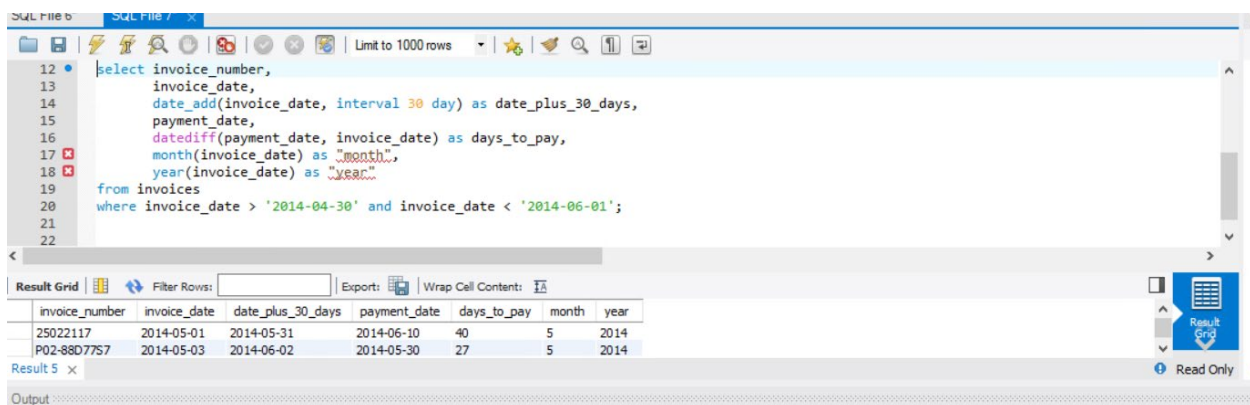
## 9.4



The screenshot shows a SQL query in an IDE. The query selects invoice information and calculates the date plus 30 days, days to pay, month, and year. The results table has 7 columns: invoice\_number, invoice\_date, date\_plus\_30\_days, payment\_date, days\_to\_pay, month, and year. Two rows are displayed.

```
11 select invoice_number,
12        invoice_date,
13        date_add(invoice_date, interval 30 day) as date_plus_30_days,
14        payment_date,
15        datediff(payment_date, invoice_date) as days_to_pay,
16        month(invoice_date) as month,
17        year(invoice_date) as year
18  from invoices;
```

| invoice_number | invoice_date | date_plus_30_days | payment_date | days_to_pay | month | year |
|----------------|--------------|-------------------|--------------|-------------|-------|------|
| 989319-457     | 2014-04-08   | 2014-05-08        | 2014-05-07   | 29          | 4     | 2014 |
| 263253241      | 2014-04-10   | 2014-05-10        | 2014-05-14   | 34          | 4     | 2014 |



The screenshot shows a SQL query in an IDE, similar to the previous one but with a filter. The query selects invoice information and calculates the date plus 30 days, days to pay, month, and year, filtered by invoice\_date between '2014-04-30' and '2014-06-01'. The results table has 7 columns: invoice\_number, invoice\_date, date\_plus\_30\_days, payment\_date, days\_to\_pay, month, and year. Two rows are displayed.

```
12 select invoice_number,
13        invoice_date,
14        date_add(invoice_date, interval 30 day) as date_plus_30_days,
15        payment_date,
16        datediff(payment_date, invoice_date) as days_to_pay,
17        month(invoice_date) as month,
18        year(invoice_date) as year
19  from invoices
20  where invoice_date > '2014-04-30' and invoice_date < '2014-06-01';
```

| invoice_number | invoice_date | date_plus_30_days | payment_date | days_to_pay | month | year |
|----------------|--------------|-------------------|--------------|-------------|-------|------|
| 25022117       | 2014-05-01   | 2014-05-31        | 2014-06-10   | 40          | 5     | 2014 |
| P02-8807757    | 2014-05-03   | 2014-06-02        | 2014-05-30   | 27          | 5     | 2014 |