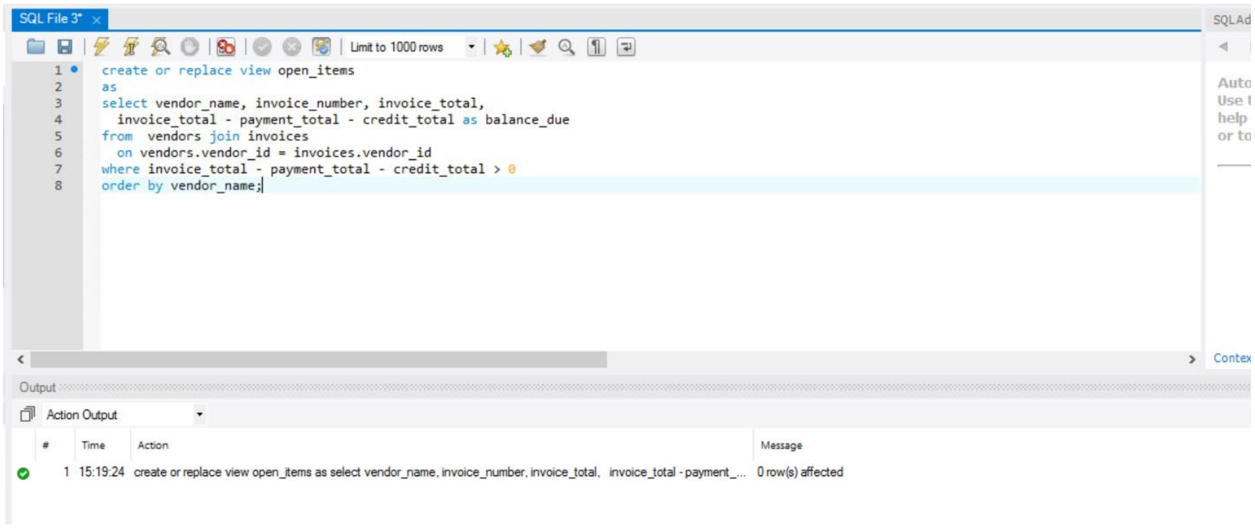
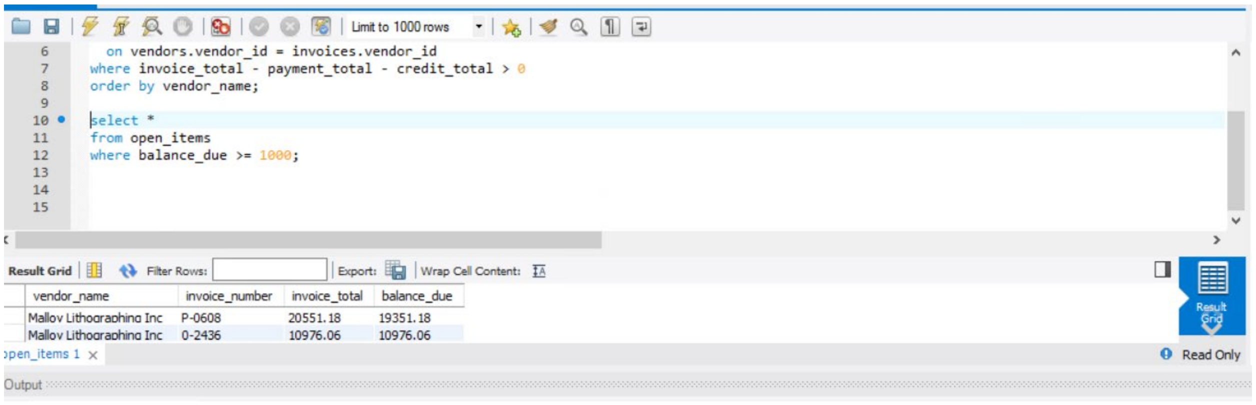
12.1



```
create or replace view open_items
as
select vendor_name, invoice_number, invoice_total,
  invoice_total - payment_total - credit_total as balance_due
from  vendors join invoices
  on vendors.vendor_id = invoices.vendor_id
where invoice_total - payment_total - credit_total > 0
order by vendor_name;
```

```
#   Time      Action                                                                                                   Message
1   15:19:24  create or replace view open_items as select vendor_name, invoice_number, invoice_total,  invoice_total - payment_...  0 row(s) affected
```

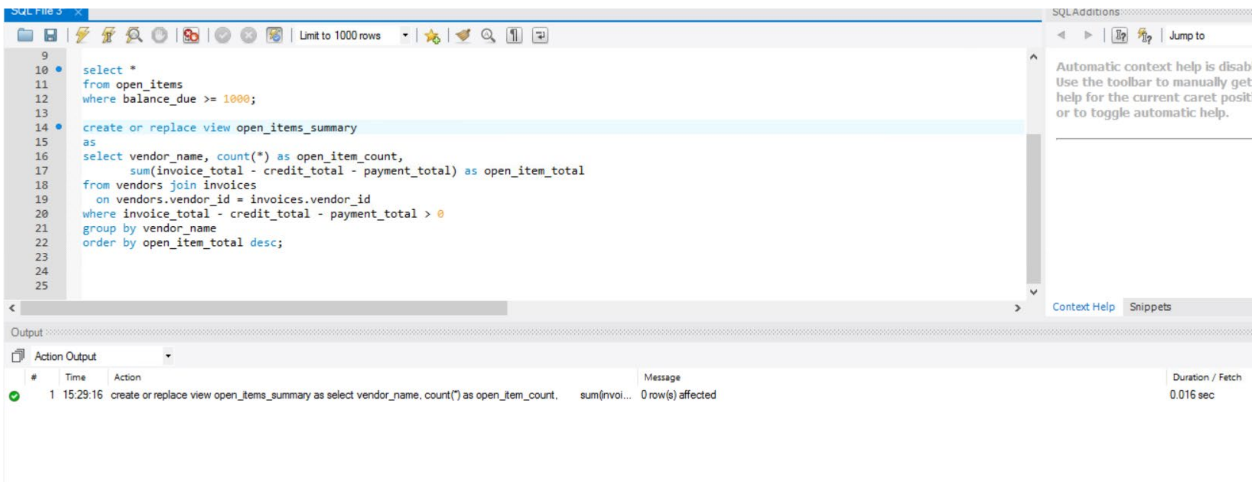
12.2



```
  on vendors.vendor_id = invoices.vendor_id
where invoice_total - payment_total - credit_total > 0
order by vendor_name;

select *
from open_items
where balance_due >= 1000;
```

vendor_name	invoice_number	invoice_total	balance_due
Mallov Lithographing Inc	P-0608	20551.18	19351.18
Mallov Lithographing Inc	0-2436	10976.06	10976.06

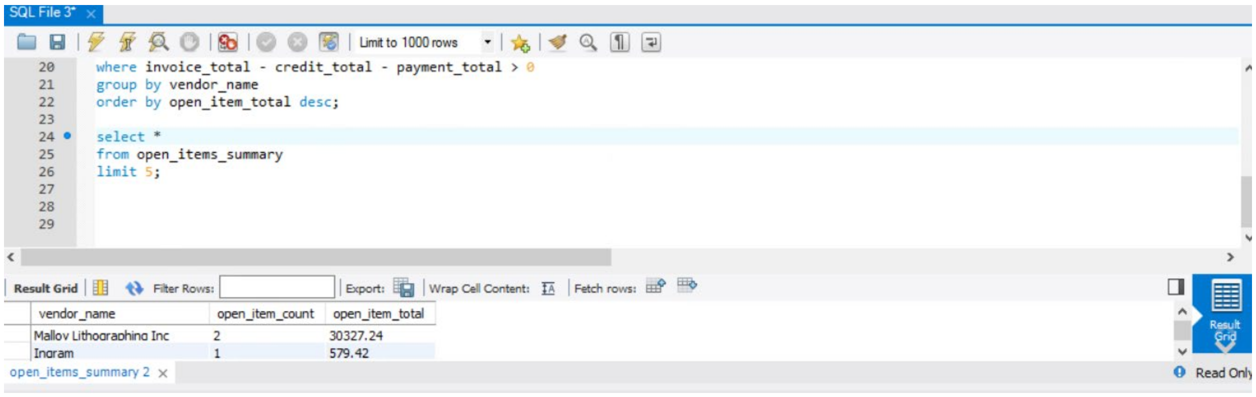
12.3



```
select *
from open_items
where balance_due >= 1000;

create or replace view open_items_summary
as
select vendor_name, count(*) as open_item_count,
       sum(invoice_total - credit_total - payment_total) as open_item_total
from vendors join invoices
  on vendors.vendor_id = invoices.vendor_id
where invoice_total - credit_total - payment_total > 0
group by vendor_name
order by open_item_total desc;
```

```
#   Time      Action                                                                                            Message            Duration / Fetch
1   15:29:16  create or replace view open_items_summary as select vendor_name, count(*) as open_item_count,   sum(invoi...  0 row(s) affected  0.016 sec
```

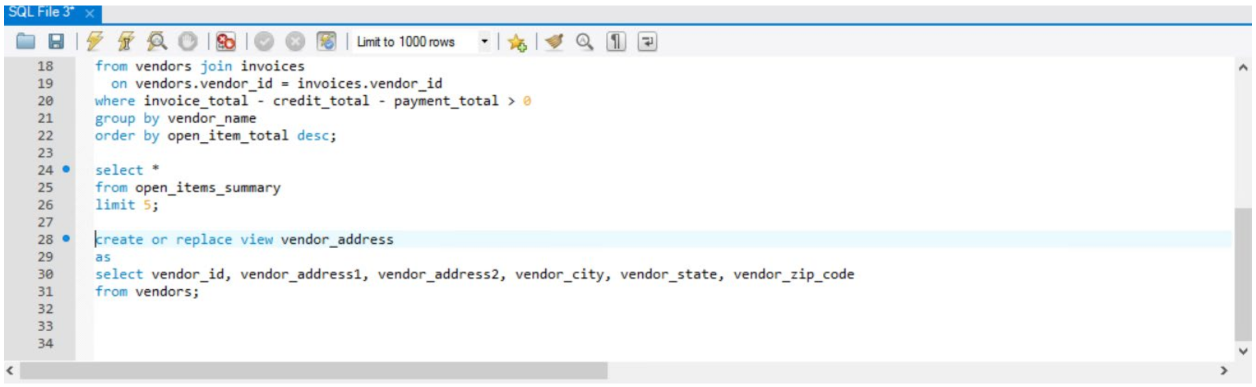
12.4



```
where invoice_total - credit_total - payment_total > 0
group by vendor_name
order by open_item_total desc;

select *
from open_items_summary
limit 5;
```

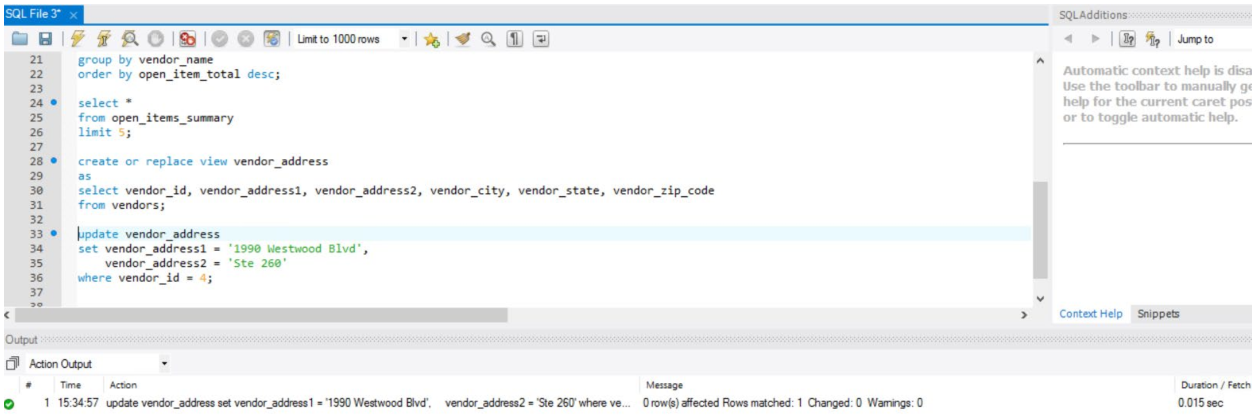
vendor_name	open_item_count	open_item_total
Malloy Lithographing Inc	2	30327.24
Ingram	1	579.42

12.5



```
from vendors join invoices
  on vendors.vendor_id = invoices.vendor_id
where invoice_total - credit_total - payment_total > 0
group by vendor_name
order by open_item_total desc;

select *
from open_items_summary
limit 5;

create or replace view vendor_address
as
select vendor_id, vendor_address1, vendor_address2, vendor_city, vendor_state, vendor_zip_code
from vendors;
```

12.6

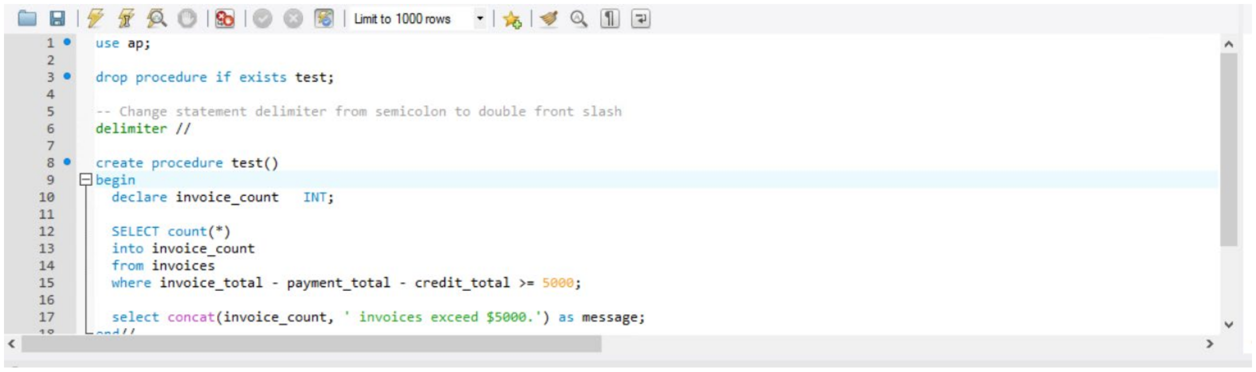


```
group by vendor_name
order by open_item_total desc;

select *
from open_items_summary
limit 5;

create or replace view vendor_address
as
select vendor_id, vendor_address1, vendor_address2, vendor_city, vendor_state, vendor_zip_code
from vendors;

update vendor_address
set vendor_address1 = '1990 Westwood Blvd',
    vendor_address2 = 'Ste 260'
where vendor_id = 4;
```

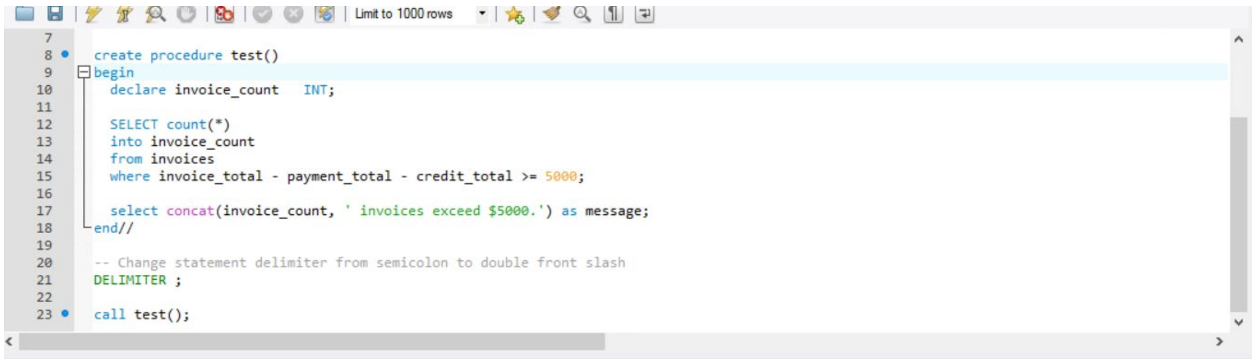
Automatic context help is disa
Use the toolbar to manually ge
help for the current caret pos
or to toggle automatic help.

```
#  Time      Action                                                                                    Message                                                        Duration / Fetch
1  15:34:57  update vendor_address set vendor_address1 = '1990 Westwood Blvd',  vendor_address2 = 'Ste 260' where ve...  0 row(s) affected Rows matched: 1  Changed: 0  Warnings: 0  0.015 sec
```

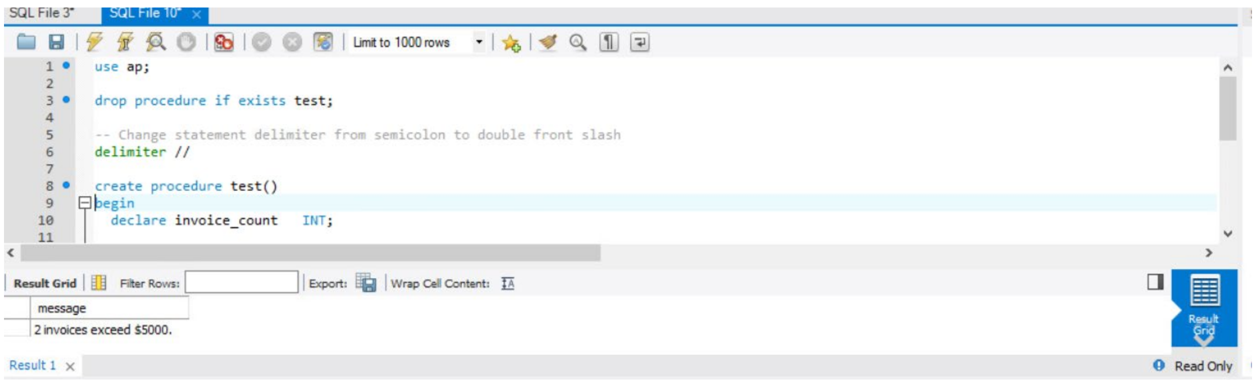
13.1


Limit to 1000 rows

```
use ap;

drop procedure if exists test;

-- Change statement delimiter from semicolon to double front slash
delimiter //

create procedure test()
begin
  declare invoice_count   INT;

  SELECT count(*)
  into invoice_count
  from invoices
  where invoice_total - payment_total - credit_total >= 5000;

  select concat(invoice_count, ' invoices exceed $5000.') as message;
```


Limit to 1000 rows

```

create procedure test()
begin
  declare invoice_count   INT;

  SELECT count(*)
  into invoice_count
  from invoices
  where invoice_total - payment_total - credit_total >= 5000;

  select concat(invoice_count, ' invoices exceed $5000.') as message;
end//

-- Change statement delimiter from semicolon to double front slash
DELIMITER ;

call test();
```

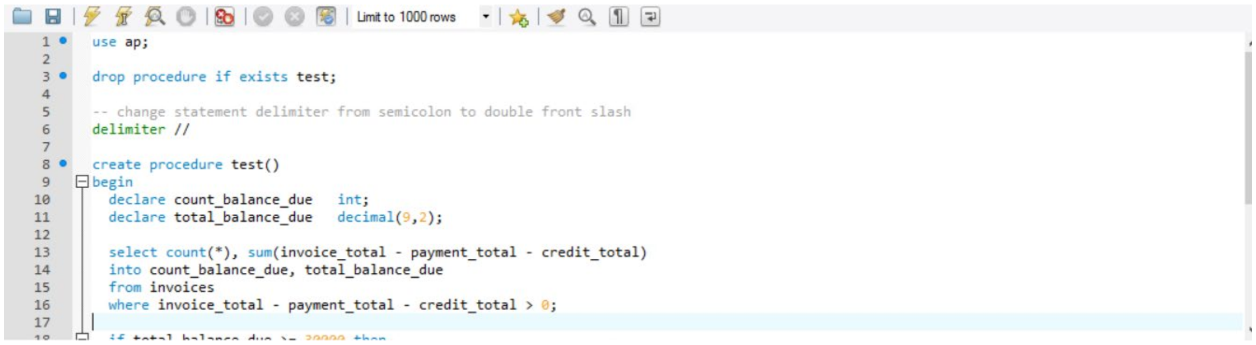


Limit to 1000 rows

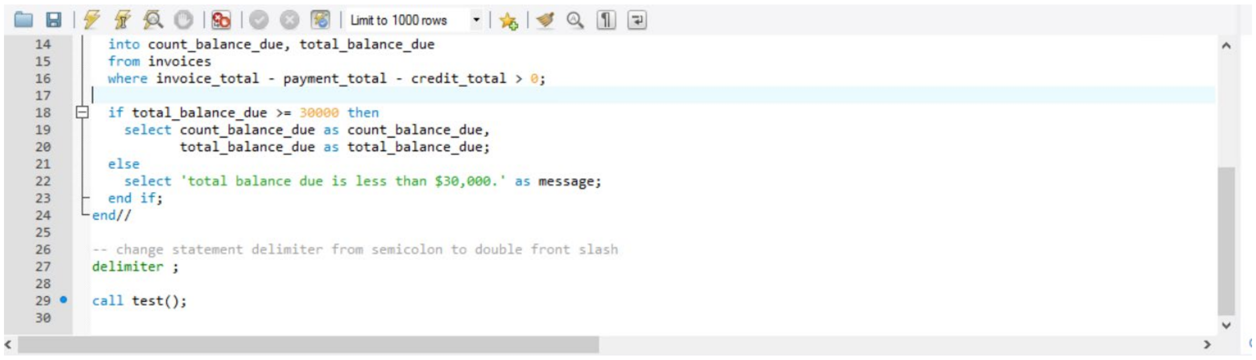
```
use ap;

drop procedure if exists test;

-- Change statement delimiter from semicolon to double front slash
delimiter //

create procedure test()
begin
  declare invoice_count   INT;
```

```
message
2 invoices exceed $5000.
```

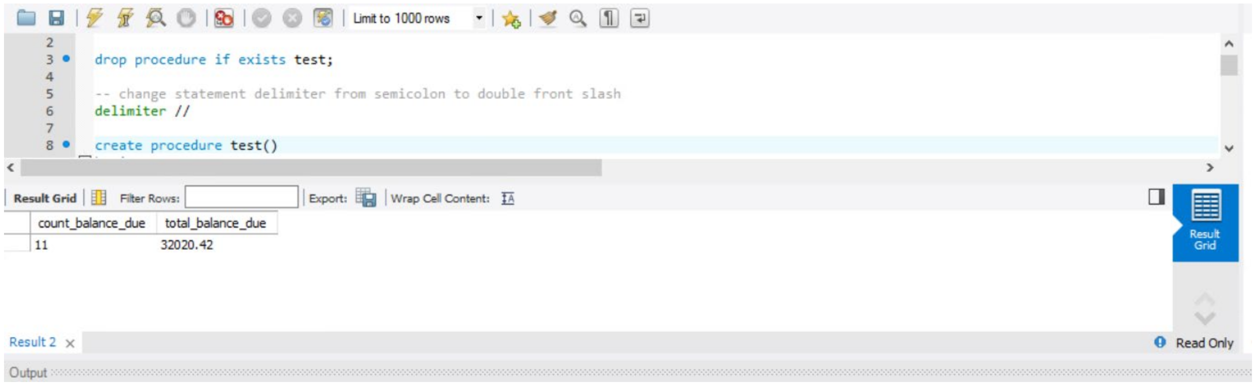
13.2


Limit to 1000 rows

```
use ap;

drop procedure if exists test;

-- change statement delimiter from semicolon to double front slash
delimiter //

create procedure test()
begin
  declare count_balance_due   int;
  declare total_balance_due   decimal(9,2);

  select count(*), sum(invoice_total - payment_total - credit_total)
  into count_balance_due, total_balance_due
  from invoices
  where invoice_total - payment_total - credit_total > 0;

```



```
  into count_balance_due, total_balance_due
  from invoices
  where invoice_total - payment_total - credit_total > 0;

  if total_balance_due >= 30000 then
    select count_balance_due as count_balance_due,
           total_balance_due as total_balance_due;
  else
    select 'total balance due is less than $30,000.' as message;
  end if;
end//

-- change statement delimiter from semicolon to double front slash
delimiter ;

call test();
```

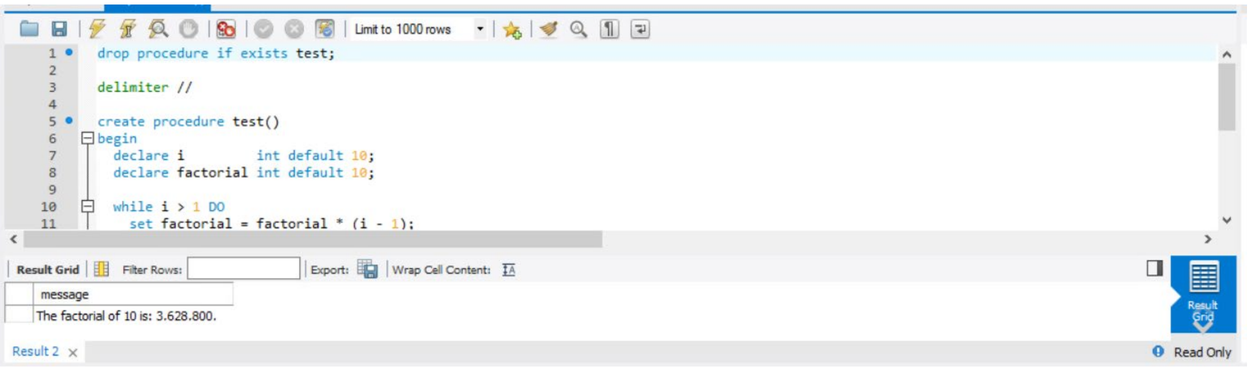


```

drop procedure if exists test;

-- change statement delimiter from semicolon to double front slash
delimiter //

create procedure test()
```

```
count_balance_due   total_balance_due
11                  32020.42
```

Result 2

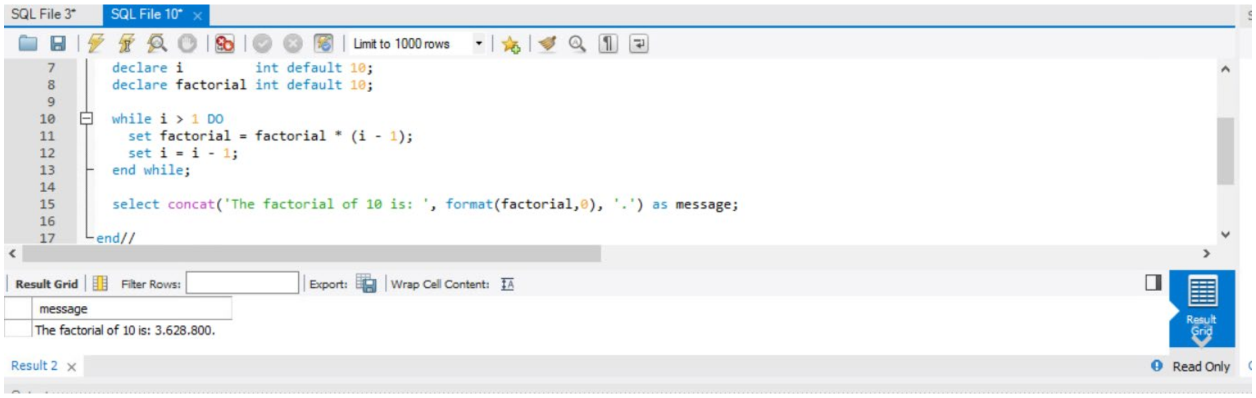
13.3



```
drop procedure if exists test;

delimiter //

create procedure test()
begin
  declare i         int default 10;
  declare factorial int default 10;

  while i > 1 DO
    set factorial = factorial * (i - 1);
```

```
message
The factorial of 10 is: 3.628.800.
```

Result 2



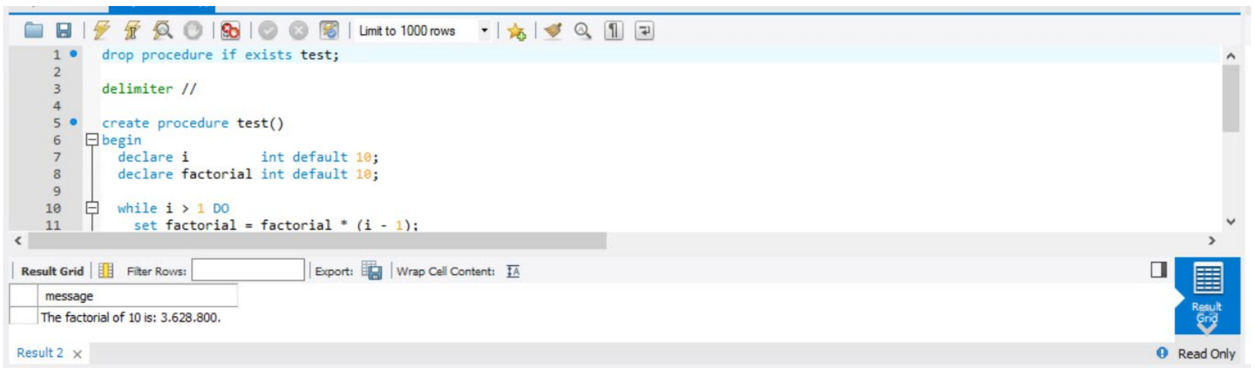
SQL File 3* SQL File 10*

```
  declare i         int default 10;
  declare factorial int default 10;

  while i > 1 DO
    set factorial = factorial * (i - 1);
    set i = i - 1;
  end while;

  select concat('The factorial of 10 is: ', format(factorial,0), '.') as message;

end//
```

```
message
The factorial of 10 is: 3.628.800.
```

Result 2

13.3



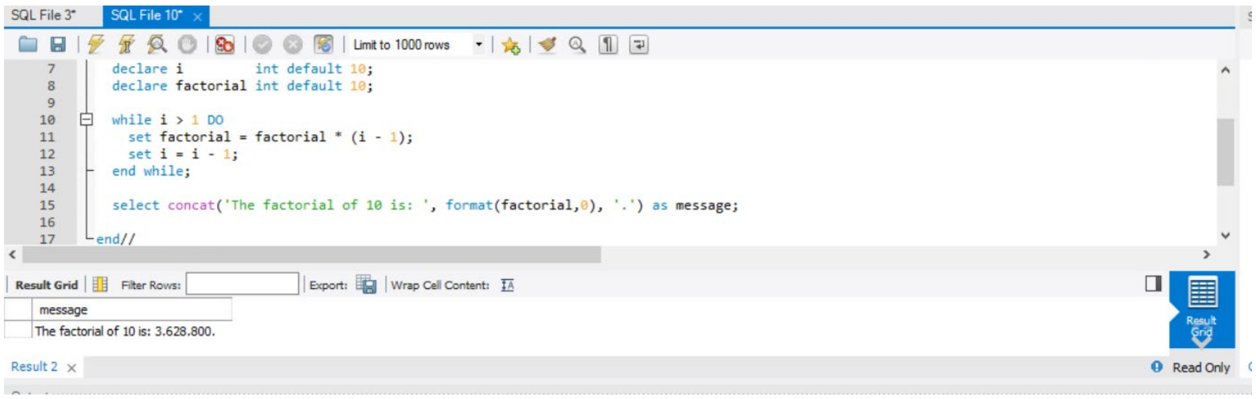
Limit to 1000 rows

```
drop procedure if exists test;

delimiter //

create procedure test()
begin
  declare i         int default 10;
  declare factorial int default 10;

  while i > 1 DO
    set factorial = factorial * (i - 1);
```

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

```
message
The factorial of 10 is: 3.628.800.
```

Result 2 Read Only



SQL File 3* SQL File 10*

Limit to 1000 rows

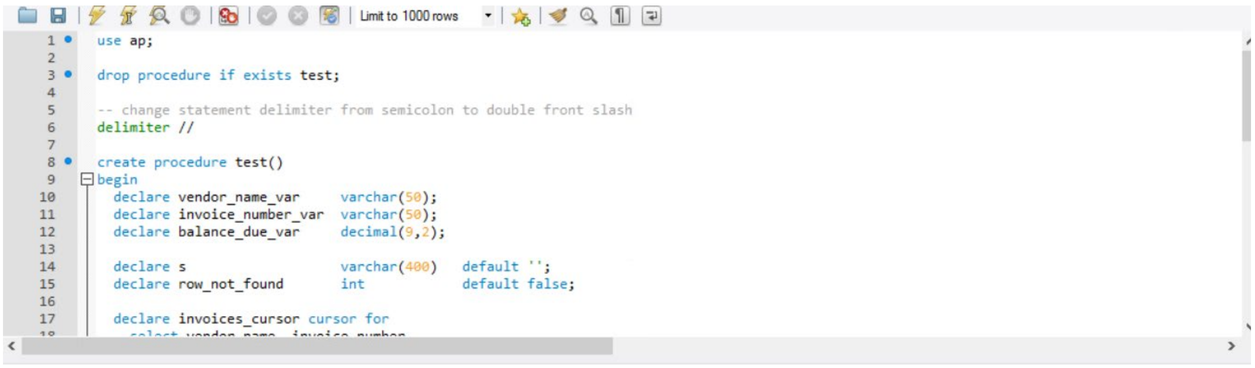
```
  declare i         int default 10;
  declare factorial int default 10;

  while i > 1 DO
    set factorial = factorial * (i - 1);
    set i = i - 1;
  end while;

  select concat('The factorial of 10 is: ', format(factorial,0), '.') as message;

end//
```

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

```
message
The factorial of 10 is: 3.628.800.
```

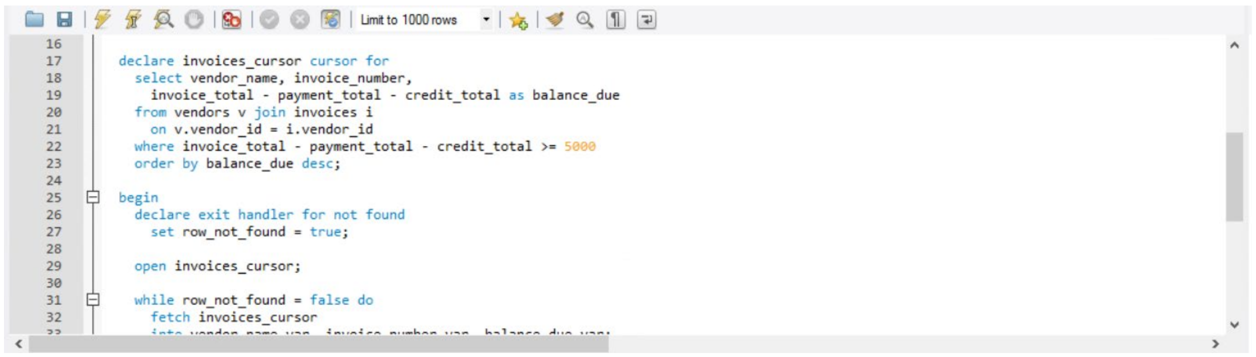
Result 2 Read Only

13.4

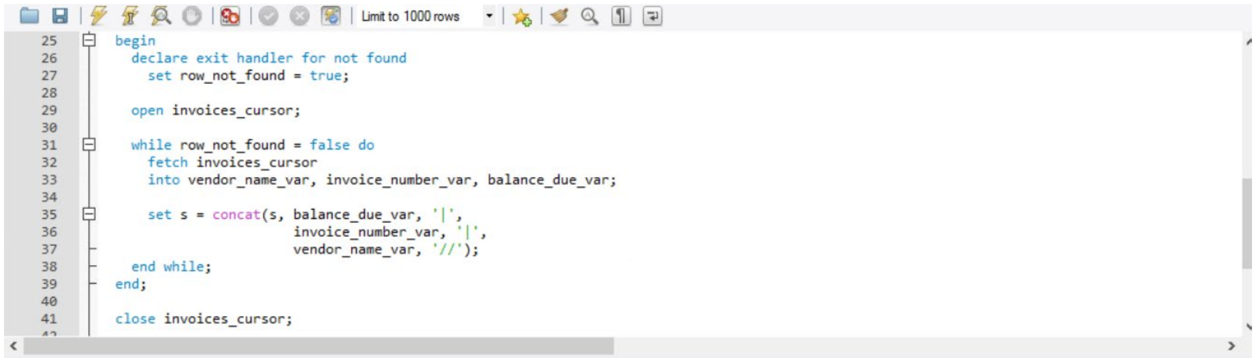


Limit to 1000 rows

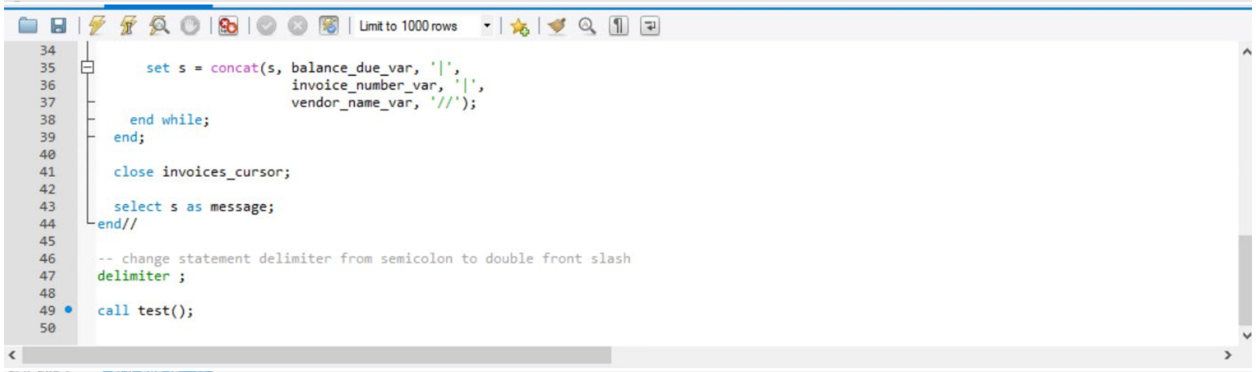
```
use ap;

drop procedure if exists test;

-- change statement delimiter from semicolon to double front slash
delimiter //

create procedure test()
begin
  declare vendor_name_var     varchar(50);
  declare invoice_number_var  varchar(50);
  declare balance_due_var     decimal(9,2);

  declare s                   varchar(400)   default '';
  declare row_not_found       int            default false;

  declare invoices_cursor cursor for
```



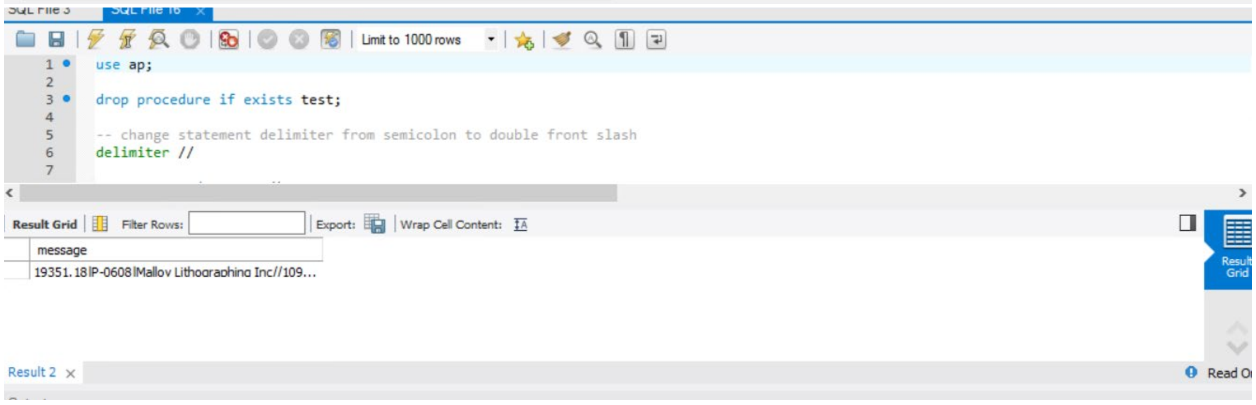
```

  declare invoices_cursor cursor for
    select vendor_name, invoice_number,
      invoice_total - payment_total - credit_total as balance_due
    from vendors v join invoices i
      on v.vendor_id = i.vendor_id
    where invoice_total - payment_total - credit_total >= 5000
    order by balance_due desc;

  begin
    declare exit handler for not found
      set row_not_found = true;

    open invoices_cursor;

    while row_not_found = false do
      fetch invoices_cursor
```



```
  begin
    declare exit handler for not found
      set row_not_found = true;

    open invoices_cursor;

    while row_not_found = false do
      fetch invoices_cursor
      into vendor_name_var, invoice_number_var, balance_due_var;

      set s = concat(s, balance_due_var, '|',
                        invoice_number_var, '|',
                        vendor_name_var, '//');
    end while;
  end;

  close invoices_cursor;
```



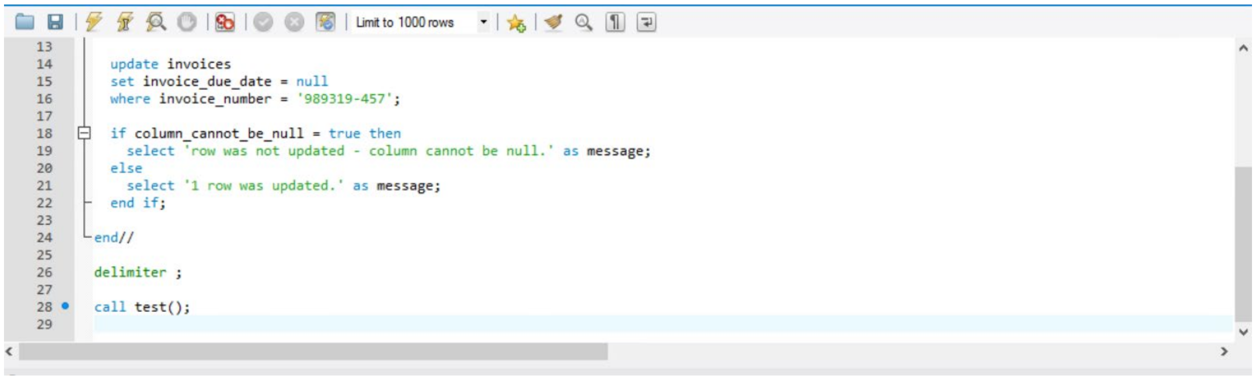
```

      set s = concat(s, balance_due_var, '|',
                        invoice_number_var, '|',
                        vendor_name_var, '//');
    end while;
  end;

  close invoices_cursor;

  select s as message;
end//

-- change statement delimiter from semicolon to double front slash
delimiter ;

call test();

```



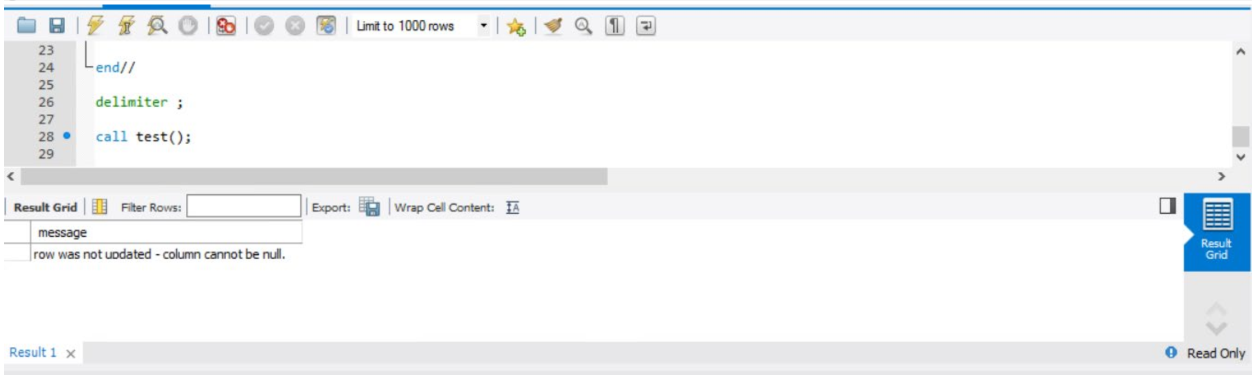
```
use ap;

drop procedure if exists test;

-- change statement delimiter from semicolon to double front slash
delimiter //
```

```
message
19351.18|P-0608|Malloy Lithographing Inc//109...
```

13.5

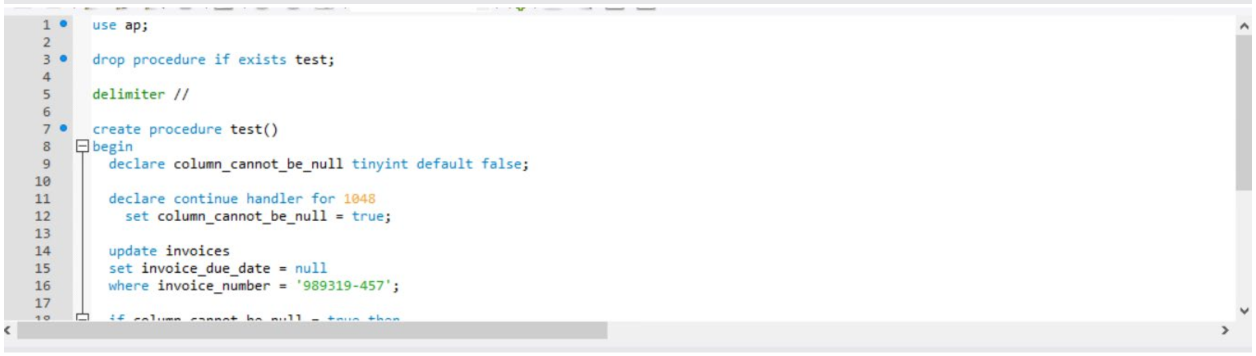


```
  update invoices
  set invoice_due_date = null
  where invoice_number = '989319-457';

  if column_cannot_be_null = true then
    select 'row was not updated - column cannot be null.' as message;
  else
    select '1 row was updated.' as message;
  end if;

end//

delimiter ;

call test();
```



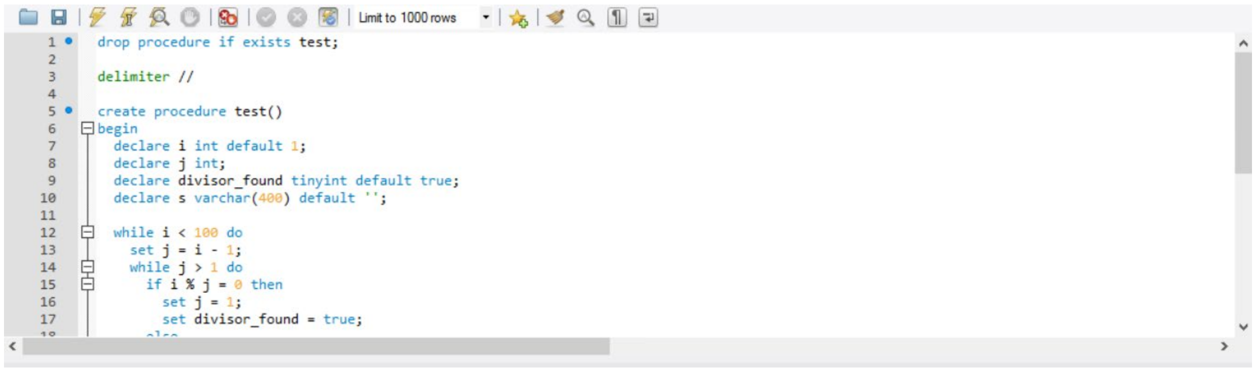
```
end//

delimiter ;

call test();
```

```
message
row was not updated - column cannot be null.
```

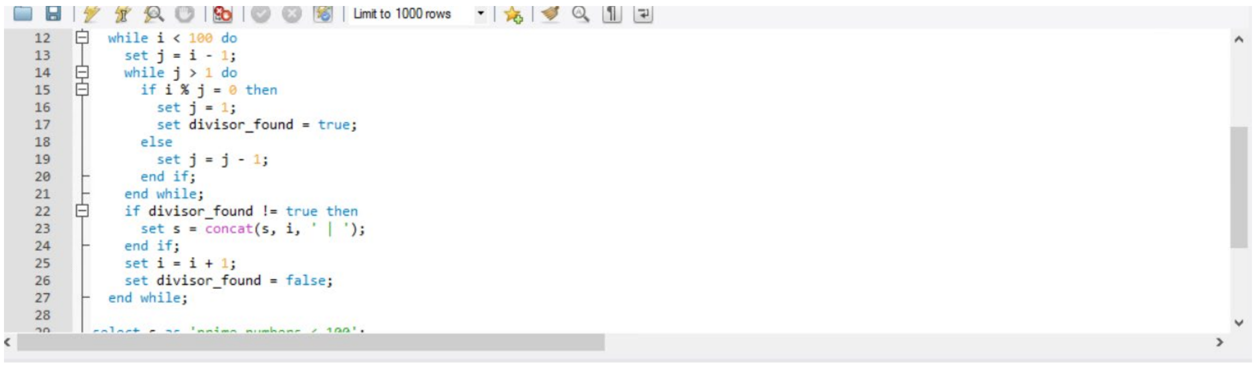


```
use ap;

drop procedure if exists test;

delimiter //

create procedure test()
begin
  declare column_cannot_be_null tinyint default false;

  declare continue handler for 1048
    set column_cannot_be_null = true;

  update invoices
  set invoice_due_date = null
  where invoice_number = '989319-457';

```

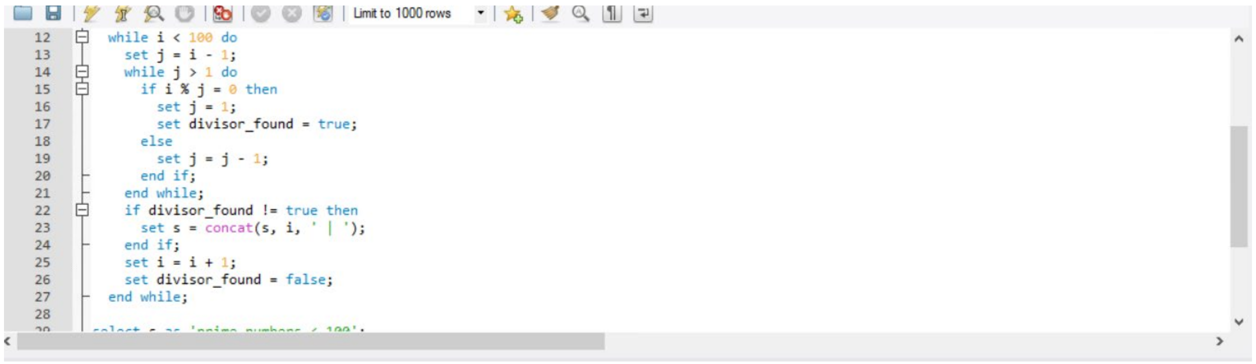
13.6



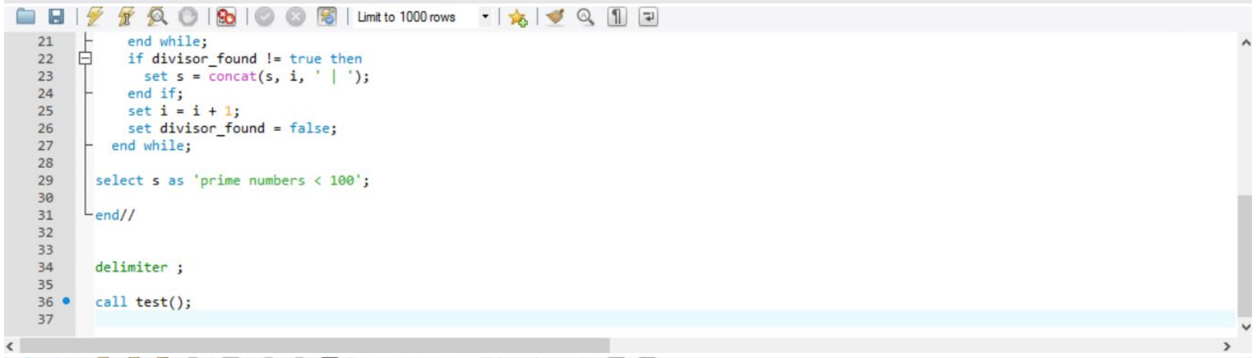
```
drop procedure if exists test;

delimiter //

create procedure test()
begin
  declare i int default 1;
  declare j int;
  declare divisor_found tinyint default true;
  declare s varchar(400) default '';

  while i < 100 do
    set j = i - 1;
    while j > 1 do
      if i % j = 0 then
        set j = 1;
        set divisor_found = true;
```



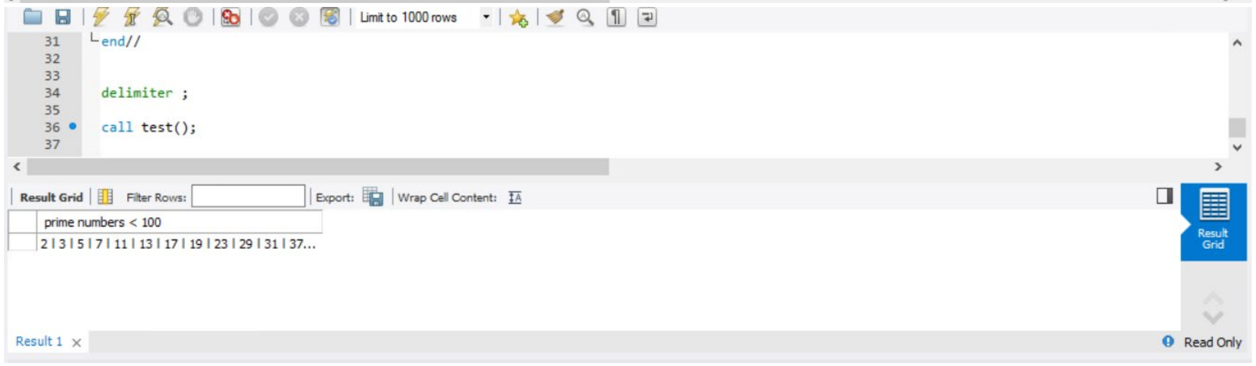
```
  while i < 100 do
    set j = i - 1;
    while j > 1 do
      if i % j = 0 then
        set j = 1;
        set divisor_found = true;
      else
        set j = j - 1;
      end if;
    end while;
    if divisor_found != true then
      set s = concat(s, i, ' | ');
    end if;
    set i = i + 1;
    set divisor_found = false;
  end while;

```



```
  while i < 100 do
    set j = i - 1;
    while j > 1 do
      if i % j = 0 then
        set j = 1;
        set divisor_found = true;
      else
        set j = j - 1;
      end if;
    end while;
    if divisor_found != true then
      set s = concat(s, i, ' | ');
    end if;
    set i = i + 1;
    set divisor_found = false;
  end while;

```



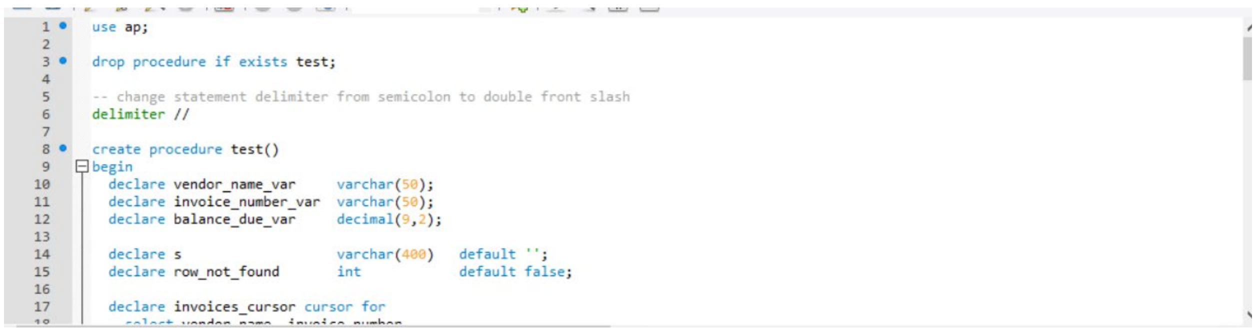
```
    end while;
    if divisor_found != true then
      set s = concat(s, i, ' | ');
    end if;
    set i = i + 1;
    set divisor_found = false;
  end while;

select s as 'prime numbers < 100';

end//


delimiter ;

call test();

```



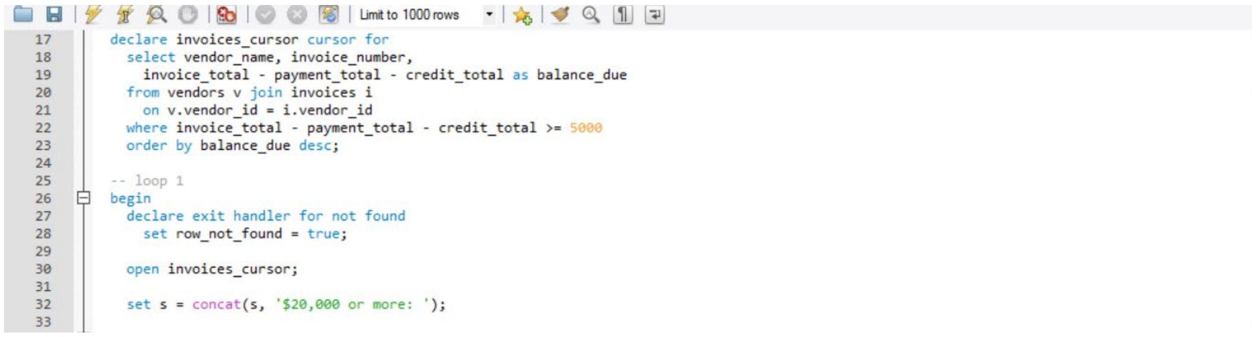
```
end//


delimiter ;

call test();

```

```
prime numbers < 100
2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 | 37...
```

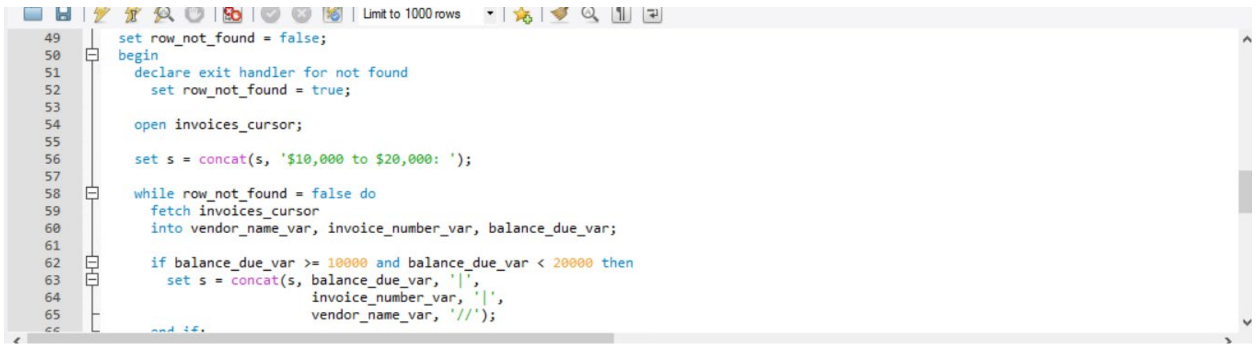
13.7



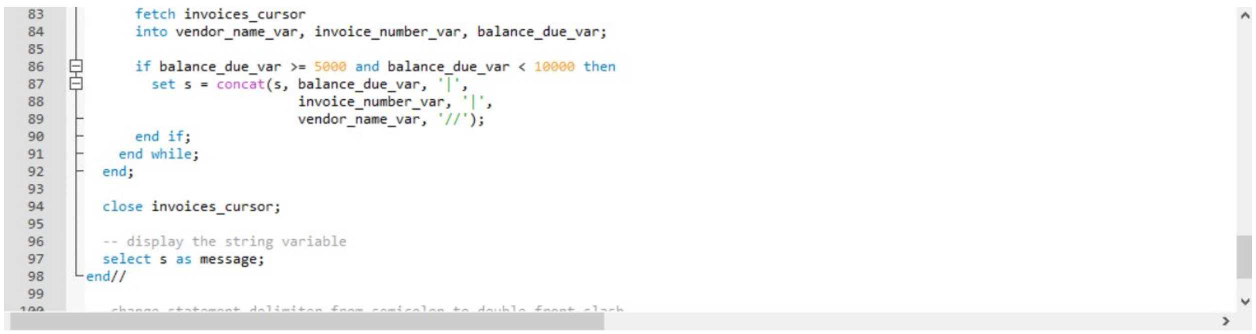
```
use ap;

drop procedure if exists test;

-- change statement delimiter from semicolon to double front slash
delimiter //

create procedure test()
begin
  declare vendor_name_var     varchar(50);
  declare invoice_number_var  varchar(50);
  declare balance_due_var     decimal(9,2);

  declare s                   varchar(400)   default '';
  declare row_not_found       int            default false;

  declare invoices_cursor cursor for
```



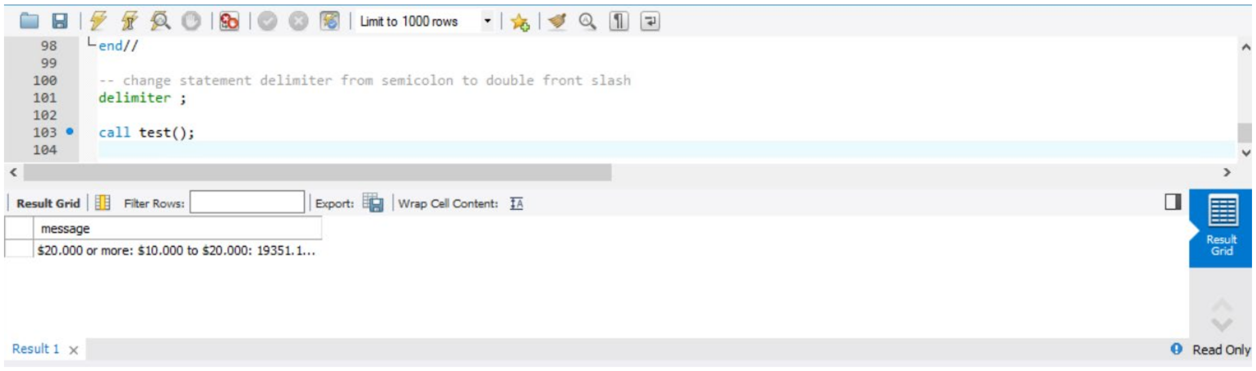
```
  declare invoices_cursor cursor for
    select vendor_name, invoice_number,
      invoice_total - payment_total - credit_total as balance_due
    from vendors v join invoices i
      on v.vendor_id = i.vendor_id
    where invoice_total - payment_total - credit_total >= 5000
    order by balance_due desc;

  -- loop 1
  begin
    declare exit handler for not found
      set row_not_found = true;

    open invoices_cursor;

    set s = concat(s, '$20,000 or more: ');

```



```
  set row_not_found = false;
  begin
    declare exit handler for not found
      set row_not_found = true;

    open invoices_cursor;

    set s = concat(s, '$10,000 to $20,000: ');

    while row_not_found = false do
      fetch invoices_cursor
      into vendor_name_var, invoice_number_var, balance_due_var;

      if balance_due_var >= 10000 and balance_due_var < 20000 then
        set s = concat(s, balance_due_var, '|',
                       invoice_number_var, '|',
                       vendor_name_var, '//');
```



```
    fetch invoices_cursor
    into vendor_name_var, invoice_number_var, balance_due_var;

    if balance_due_var >= 5000 and balance_due_var < 10000 then
      set s = concat(s, balance_due_var, '|',
                     invoice_number_var, '|',
                     vendor_name_var, '//');
    end if;
  end while;
end;

close invoices_cursor;

-- display the string variable
select s as message;
end//

```



```
end//

-- change statement delimiter from semicolon to double front slash
delimiter ;

call test();
```

```
message
$20.000 or more: $10.000 to $20.000: 19351.1...
```