**PRELAB**

**1. How many triggers can be applied to a table?**

A) There are 6 types of triggers in mysql

1. Before Insert
2. After Insert
3. Before Update
4. After Update
5. Before Delete
6. After Delete

**2. Show the two PL/SQL cursor exceptions**

A) There are 2 types of PL/SQL cursor exceptions

1. CURSOR\_ALREADY\_OPEN - Reason for this exception is when you open a cursor that is already open
2. INVALID\_CURSOR - Reason for this exception is When you perform an invalid operation on a cursor like closing a cursor, fetch data from a cursor that is not opened.

**3.Explain 3 basic parts of trigger**

A) A trigger has 3 basic parts

1. Triggering event or statement
2. Trigger Restriction
3. Trigger Action

**4.What are character function?**

A) A character function is a function that takes one or more character values as parameters and returns either a character value or a number value. The Oracle Server and PL/SQL provide a number of different character datatypes, including CHAR, VARCHAR, VARCHAR2, LONG, RAW, and LONG RAW.

**5.Explain TTITLE and BTITLE**

A) **SQL**\*Plus substitution variables (& variables) are expanded before **BTITLE** is executed. The resulting string is stored as the **BTITLE** text.

You can avoid this double substitution in a TTITLE command by not using the & prefix for variables that are to be substituted on each page of results. If you want to use a substitution variable to insert unchanging text in a TTITLE, enclose it in quotes so that it is only substituted once.

**6. What are the uses of SYSDATE and USER keywords?**

A) SYSDATE refers to the current server system date. It is a pseudo column. USER is also a pseudo column but refers to the current user logged onto the session. They are used to monitor changes happening in table.

**7. How does ROWID help in running a query faster?**

A) ROWID s are the fastest way to access a row of data, but if you can do an operation in a single DML statement, that is faster than selecting the data first, then supplying the ROWID to the DML statement. If rows are moved, the ROWID will change. Rows can move due to maintenance operations like shrinks and table moves.

**INLAB**

**Implement PL/SQL Programs on Case Study 8 (SAINT GOBAIN)**

1. **Write a cursor to select the five Expected amount from Quotation tables.**

delimiter @

create procedure exp\_amount()

begin

declare exp\_amount int;

declare count int default 0;

declare q\_finished int default 0;

declare c1 cursor for select exp\_amt from quotation;

declare continue handler for not found set q\_finished=1;

open c1;

q\_details:loop

fetch c1 into exp\_amount;

select exp\_amount;

set count=count+1;

if count=5 then

leave q\_details;

elseif q\_finished=1 then

leave q\_details;

end if;

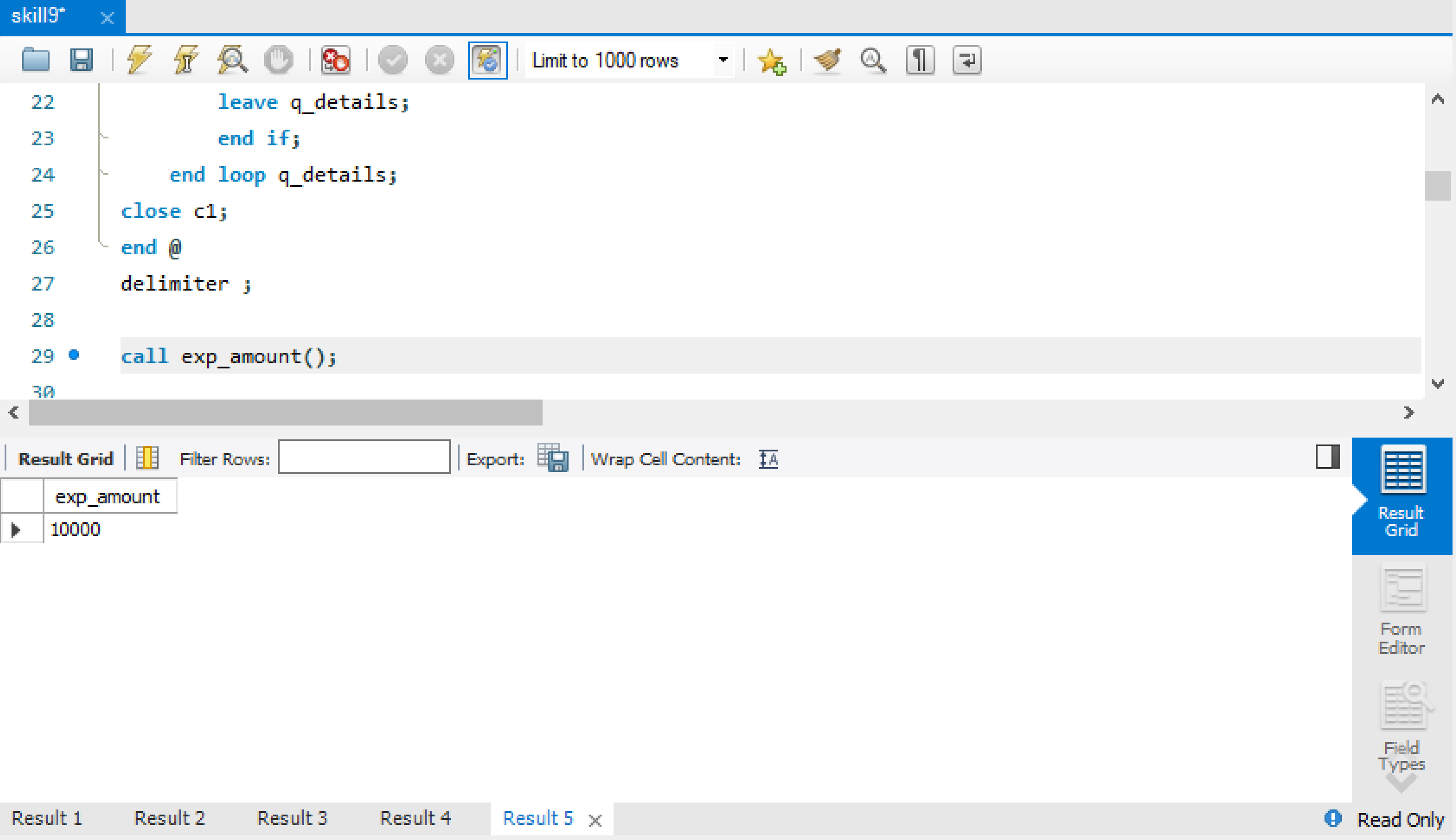
end loop q\_details;

close c1;

end @

delimiter ;

call exp\_amount();



1. **Write the Pl/SQL program provides information on the customers who paid the highest Advance amount from Quotation table**

delimiter $

create procedure Adv\_amount()

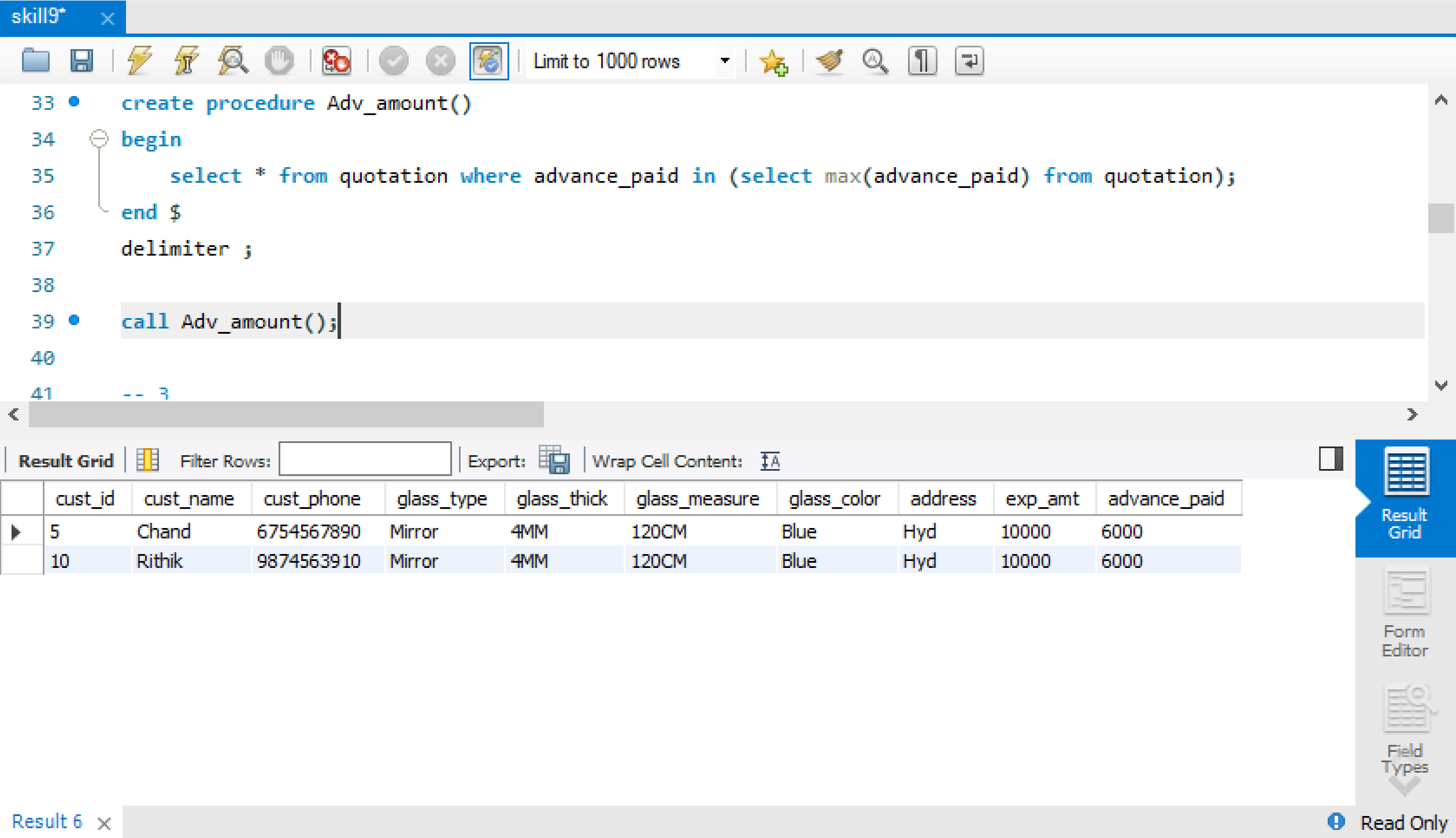
begin

select \* from quotation where advance\_paid in (select max(advance\_paid) from quotation);

end $

delimiter ;

call Adv\_amount();



1. **Write the procedure to Display the Maximum to Minimum Exp\_amt in the Quotation table.**

Delimiter $$

create procedure sorted\_quotation()

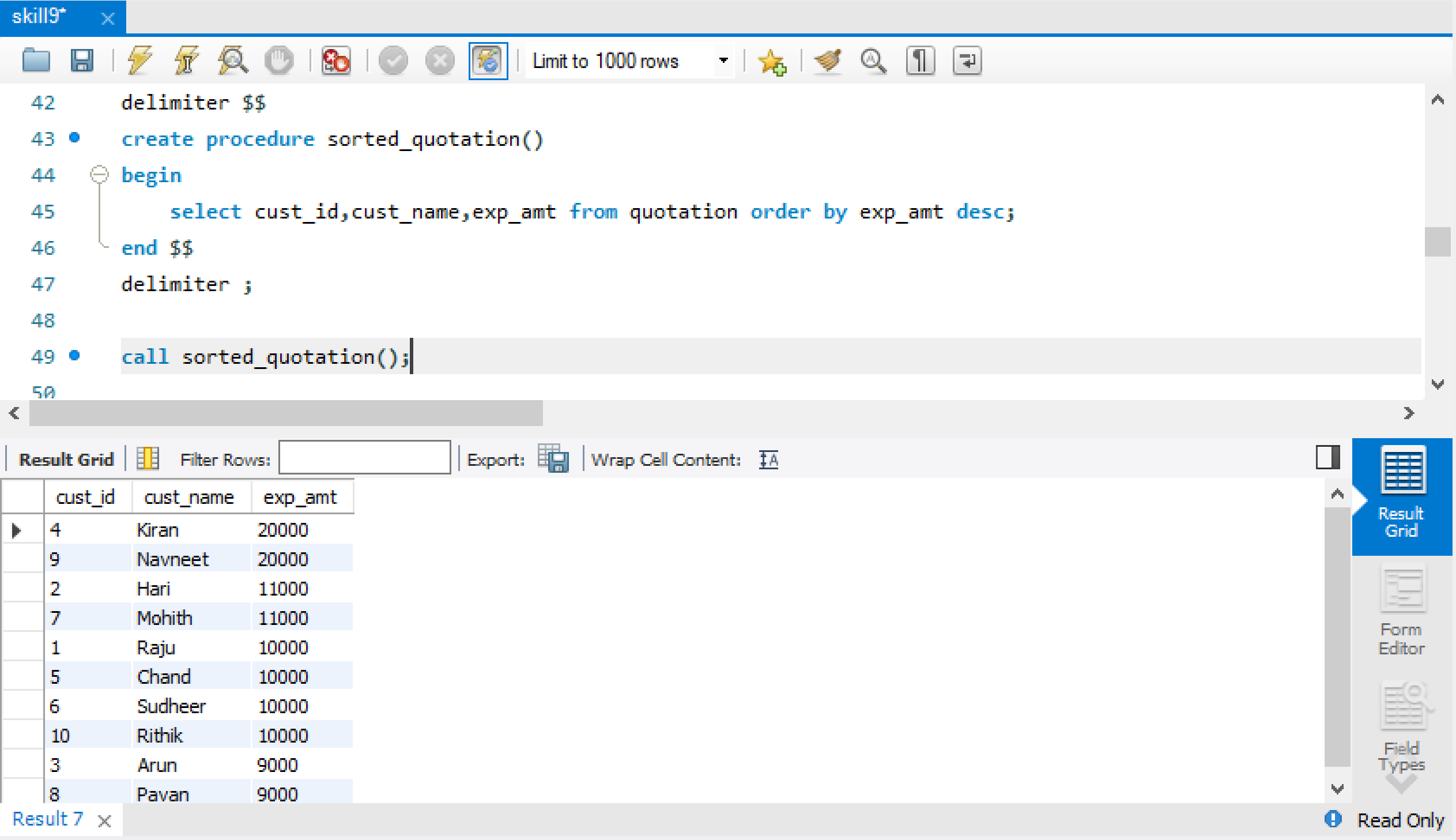
begin

select cust\_id,cust\_name,exp\_amt from quotation order by exp\_amt desc;

end $$

delimiter ;

call sorted\_quotation();



1. **Create a function that takes Cust\_Id and returns the name of the customer**

delimiter @@

create function Customer\_name(c\_id int) returns varchar(20)

begin

declare c\_name varchar(20);

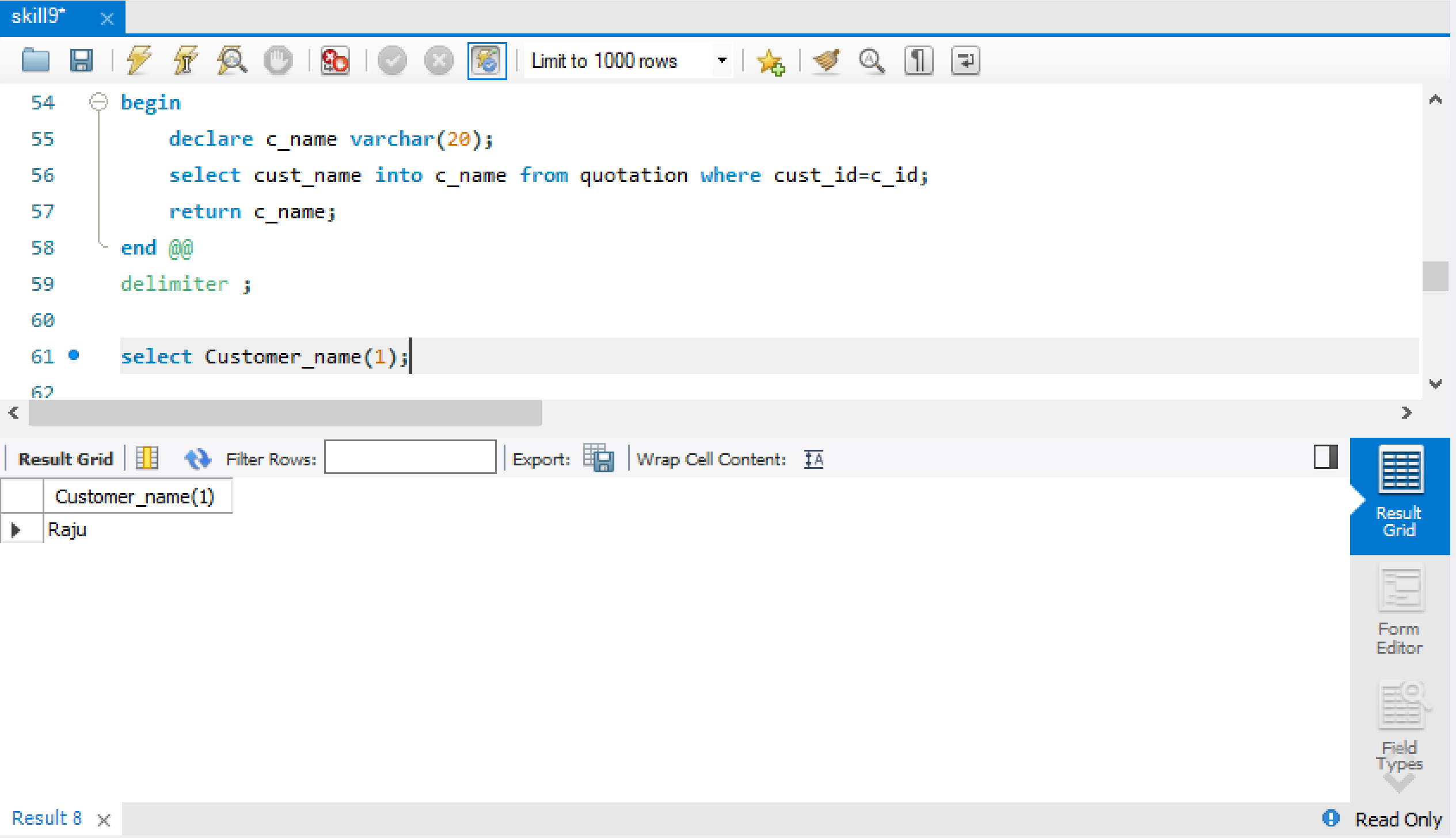
select cust\_name into c\_name from quotation where cust\_id=c\_id;

return c\_name;

end @@

delimiter ;

select Customer\_name(1);



1. **Write a function to list the Glass\_type and Glass\_feature in Quotation and Bill Table.**

delimiter $@$

create function glass(cus\_id int) returns varchar(50)

begin

declare g\_type varchar(20);

declare g\_feature varchar(20);

select glass\_type into g\_type from quotation where cust\_id=cus\_id;

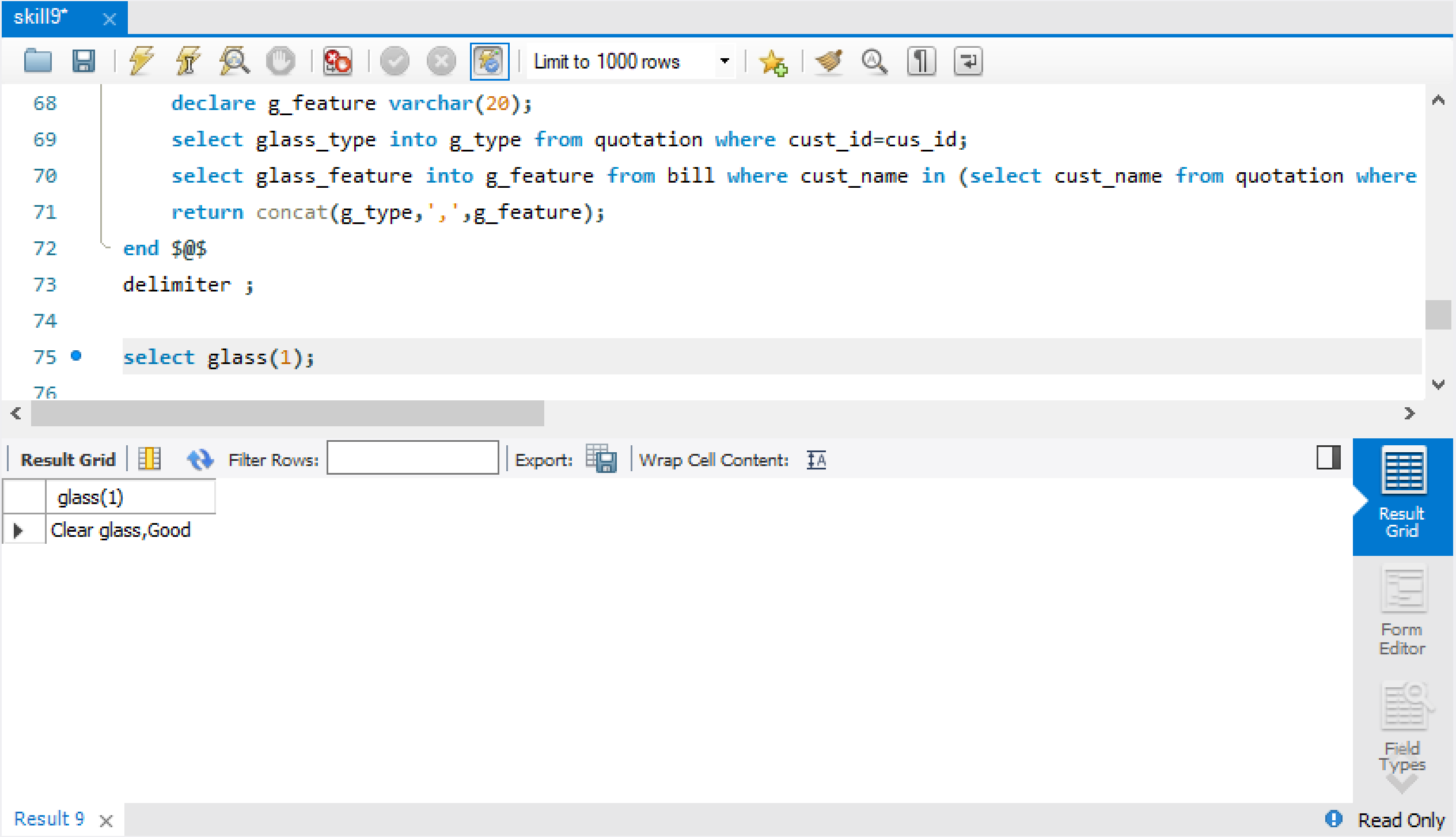
select glass\_feature into g\_feature from bill where cust\_name in (select cust\_name from quotation where cust\_id=cus\_id);

return concat(g\_type,',',g\_feature);

end $@$

delimiter ;

select glass(1);



1. **Write a procedure to delete, customers not paid any advance**

delimiter @$

create procedure delete\_noadvance()

begin

delete from quotation where advance\_paid is NULL or advance\_paid = 0;

end @$

delimiter ;

insert into quotation values(11,'RK',7286009239,'Mirror','4MM','120CM','Blue','Hyd',10000,0);

-- before delete

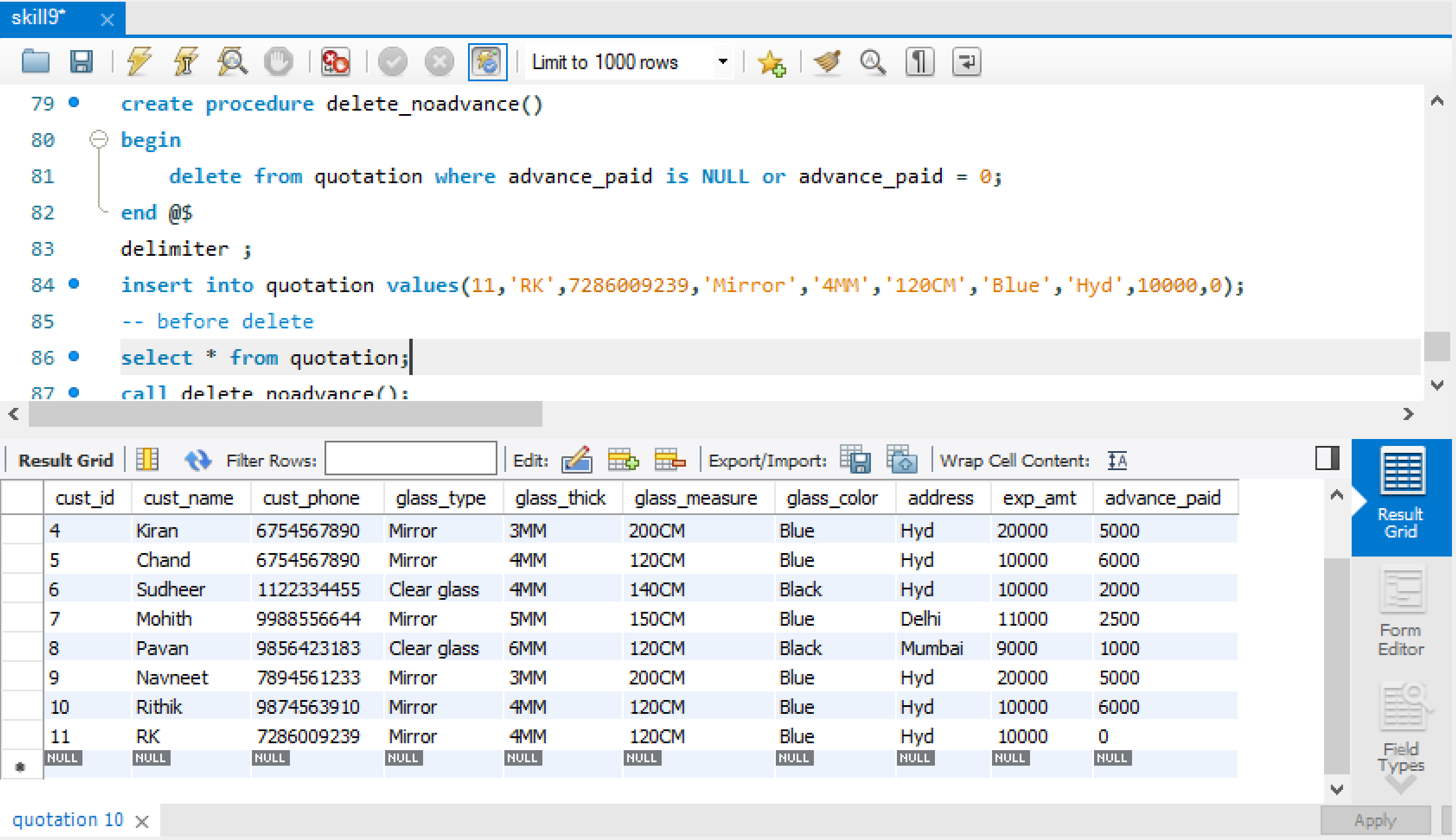
select \* from quotation;

call delete\_noadvance();

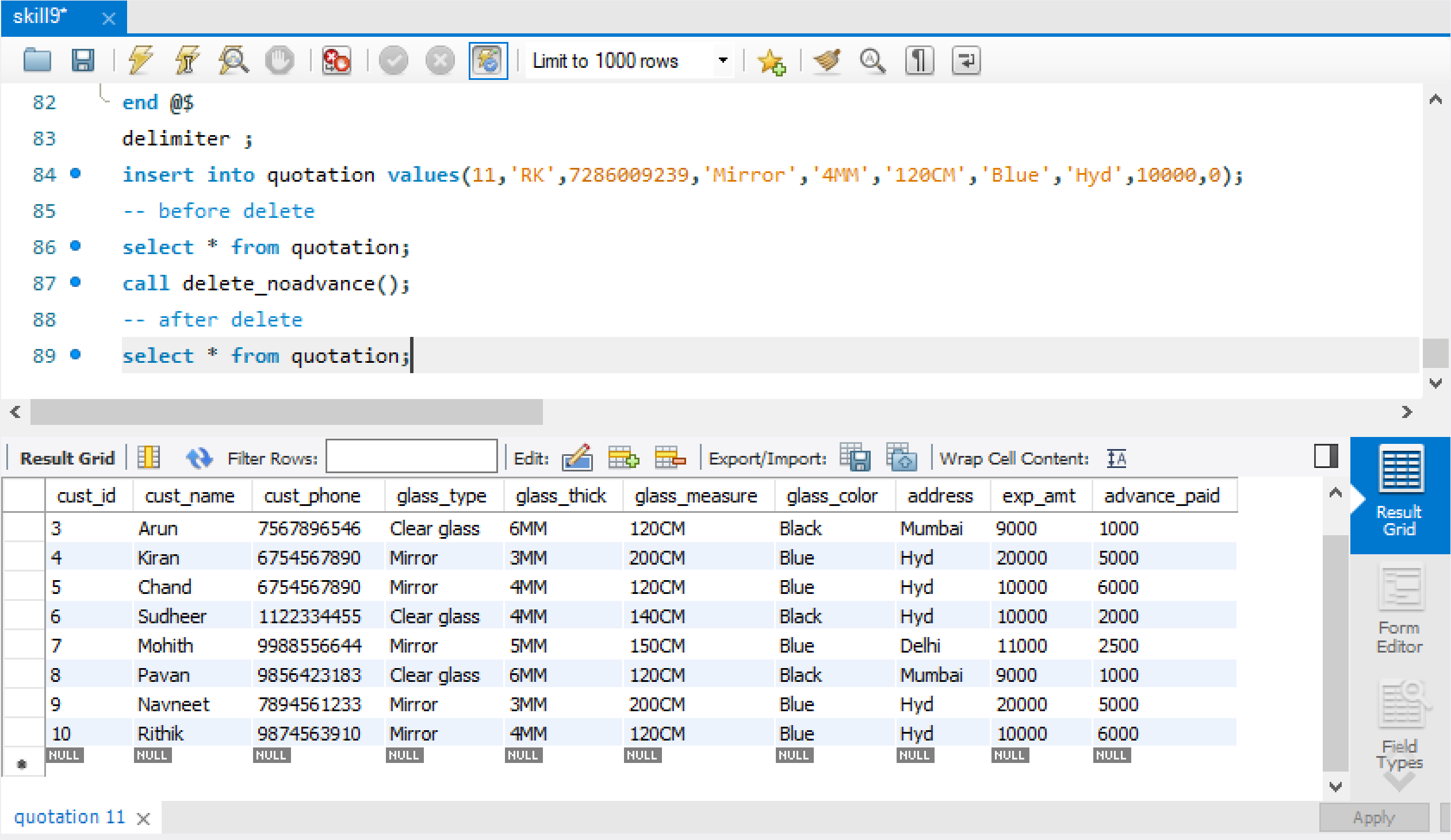
-- after delete

select \* from quotation;

**Before Delete**



**After Delete**

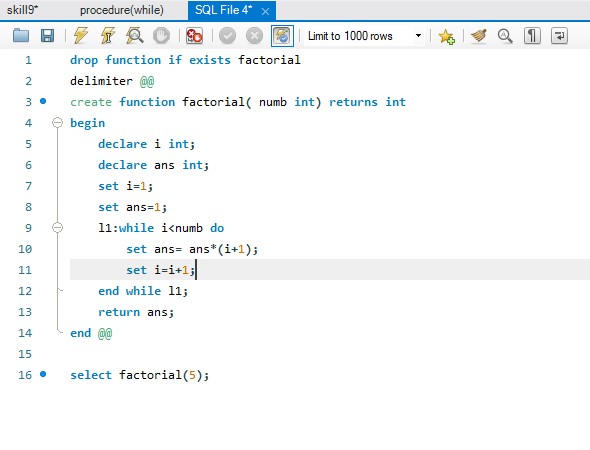


**POSTLAB**

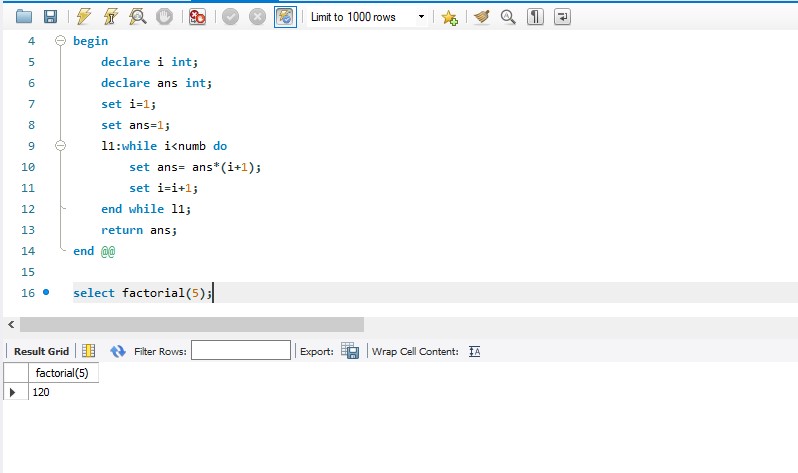
**1)** **Write a PL/SQL block to show single and multiline comments.**

A) Single-line comments begin with a double hyphen ( - - ) anywhere on a line and extend to the end of the line. Multi-line comments begin with a slash-asterisk ( /\* ), end with an asterisk-slash ( \*/ ), and can span multiple lines.

1. **Write a PL/SQl program to display the factorial of a number**



Output:

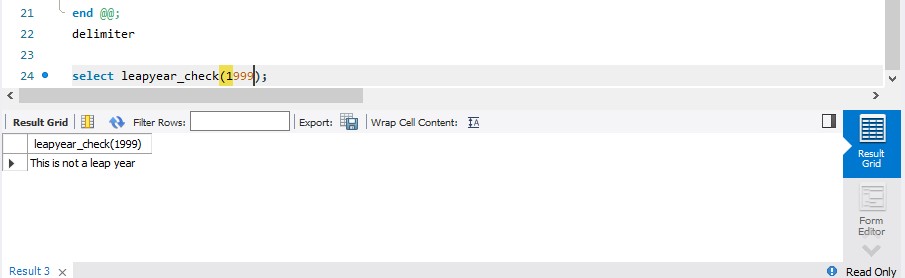


1. **Given a year, report if it is a leap year. The tricky thing here is that a leap year in the Gregorian calendar occurs: on every year that is evenly divisible by 4 except every year that is evenly divisible by 100 unless the year is also evenly divisible by 400**

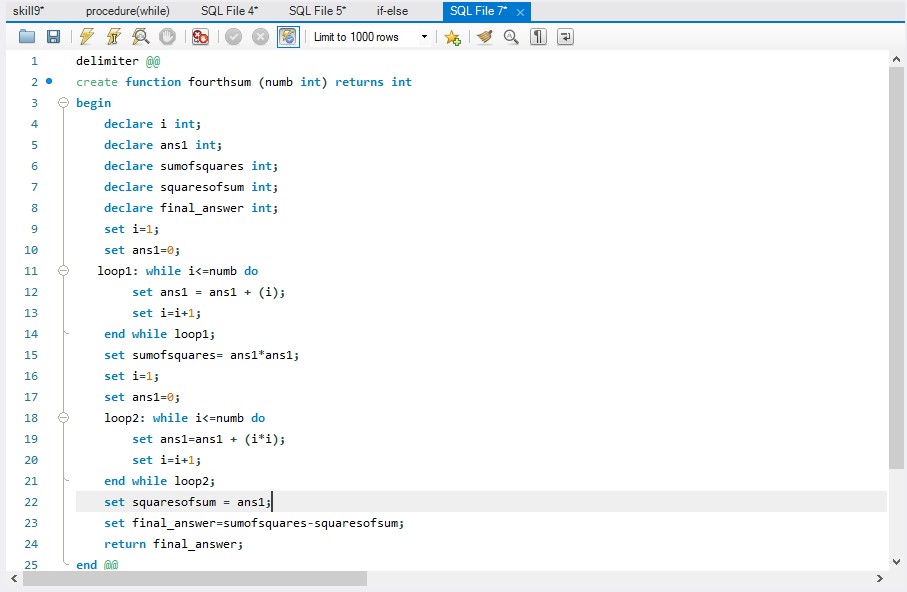
**For example, 1997 is not a leap year, but 1996 is. 1900 is not a leap year, but 2000 is. If your language provides a method in the standard library that does this look-up, pretend it doesn't exist and implement it yourself. Find the solution using pl/sql**



Output:



**4) Find the difference between the square of the sum and the sum of the squares of the first N natural numbers. The square of the sum of the first ten natural numbers is (1 + 2 + ... + 10)² = 55² = 3025. The sum of the squares of the first ten natural numbers is 1² + 2² + ... + 10² = 385. Hence the difference between the square of the sum of the first ten natural numbers and the sum of the squares of the first ten natural numbers is 3025 - 385 = 2640. Find the solution using pl/sql.**



Output:

