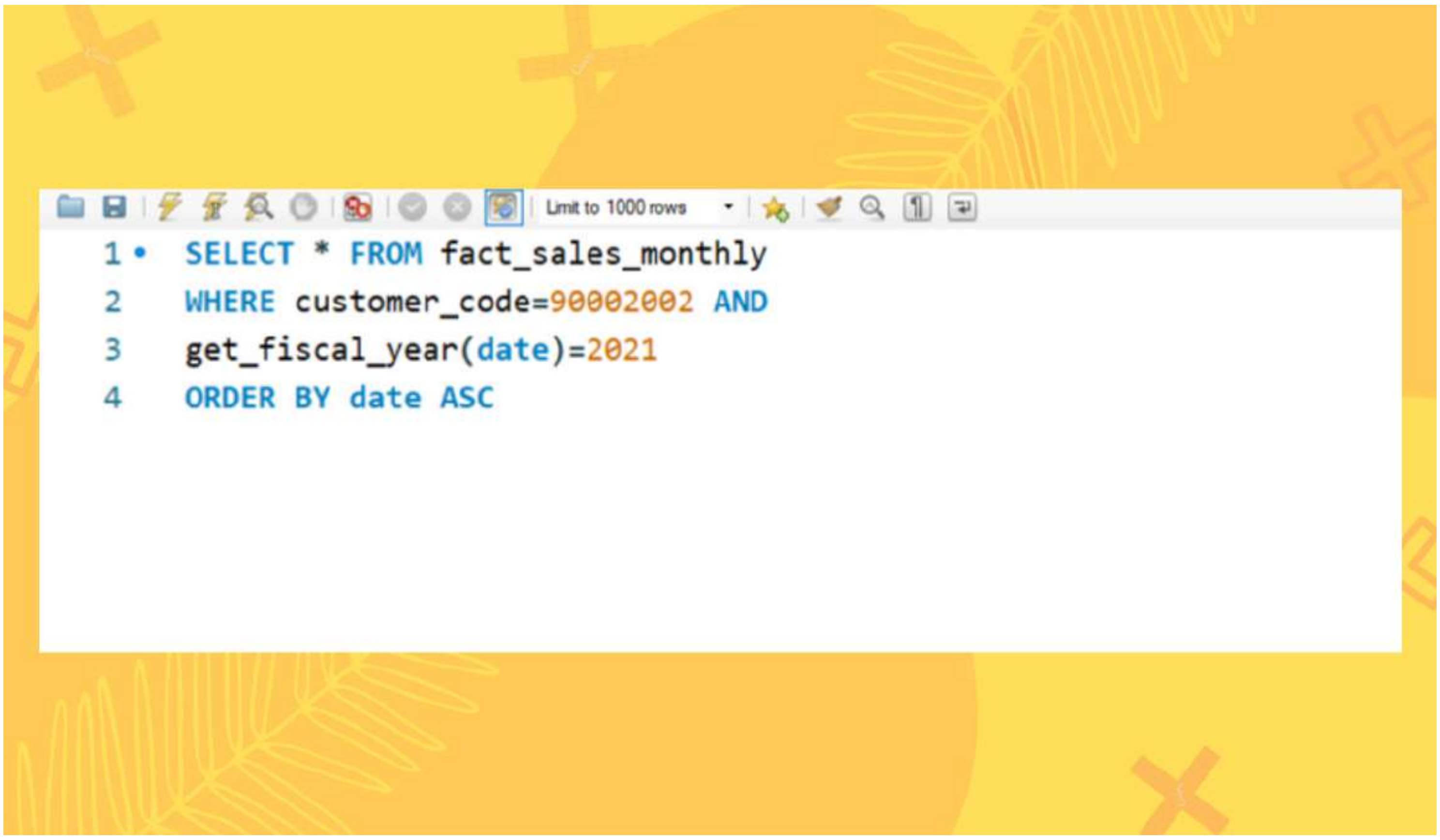
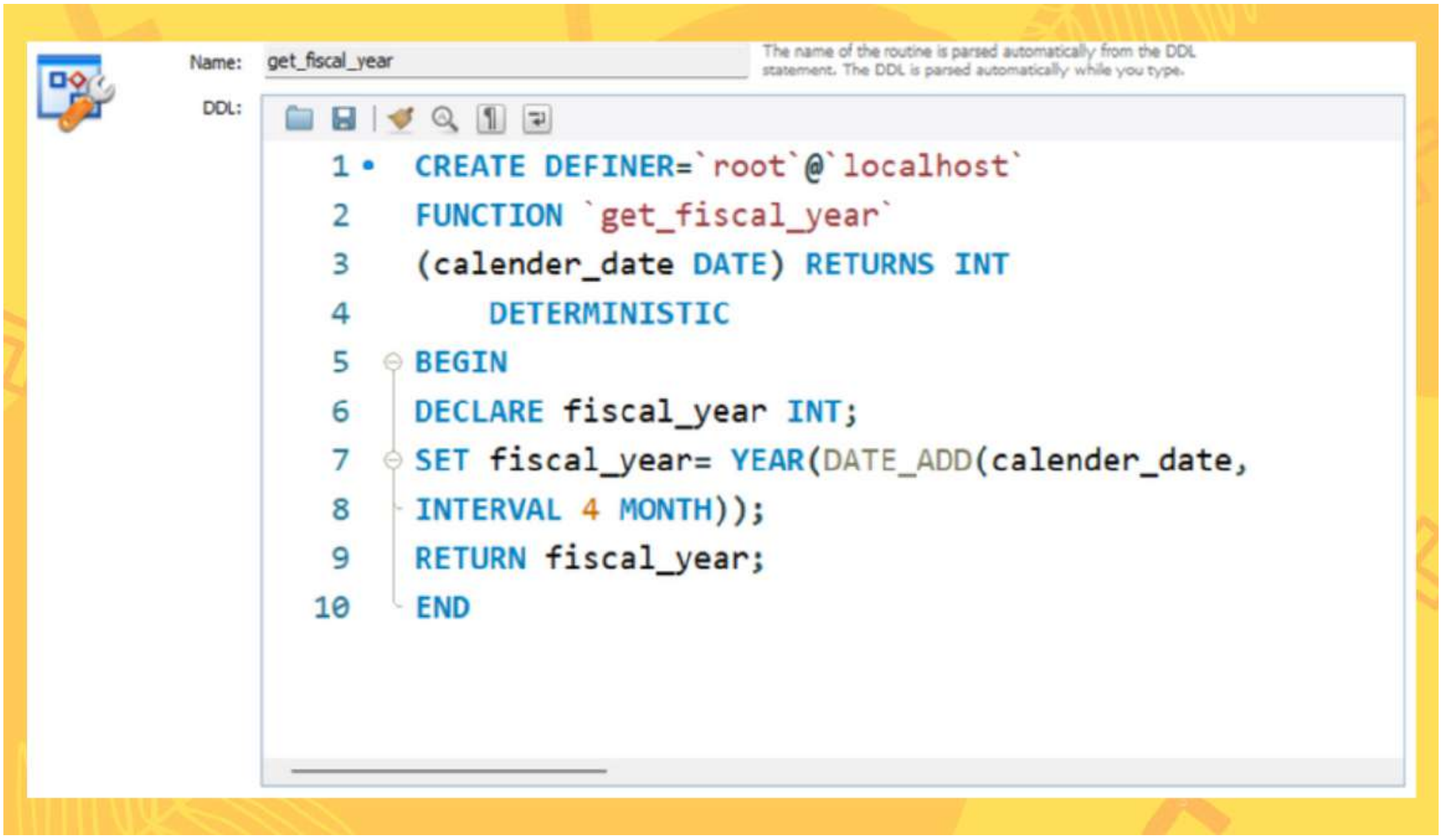
Generate a report of individual product sales (aggregated on a monthly basis at the product code level) for Fiscal Year 2021



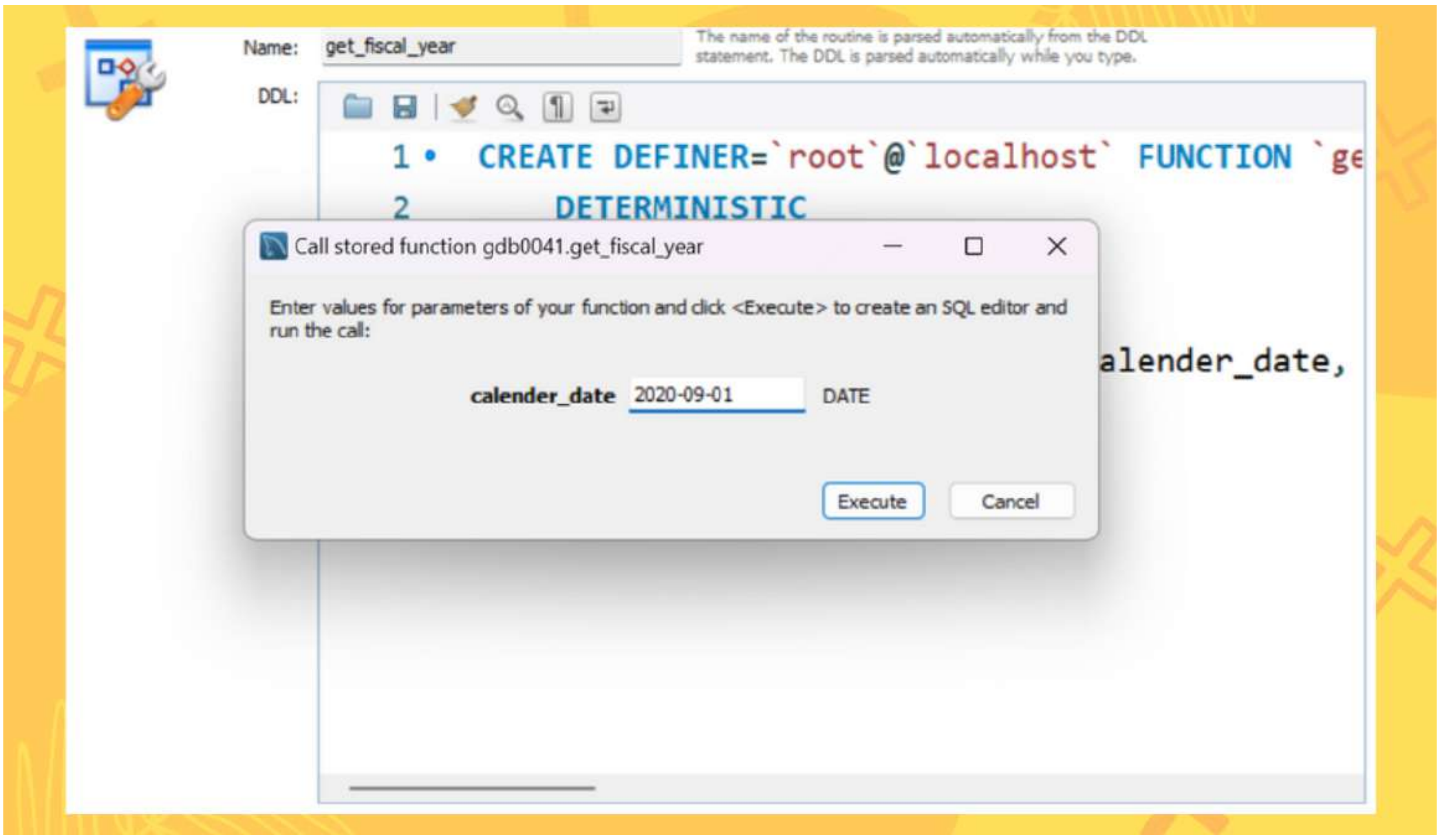
```sql
SELECT * FROM fact_sales_monthly
WHERE customer_code=90002002 AND
get_fiscal_year(date)=2021
ORDER BY date ASC
```


Name: get_fiscal_year

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```sql
CREATE DEFINER=`root`@`localhost`
FUNCTION `get_fiscal_year`
(calender_date DATE) RETURNS INT
    DETERMINISTIC
BEGIN
DECLARE fiscal_year INT;
SET fiscal_year= YEAR(DATE_ADD(calender_date,
INTERVAL 4 MONTH));
RETURN fiscal_year;
END
```



Name: get_fiscal_year

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

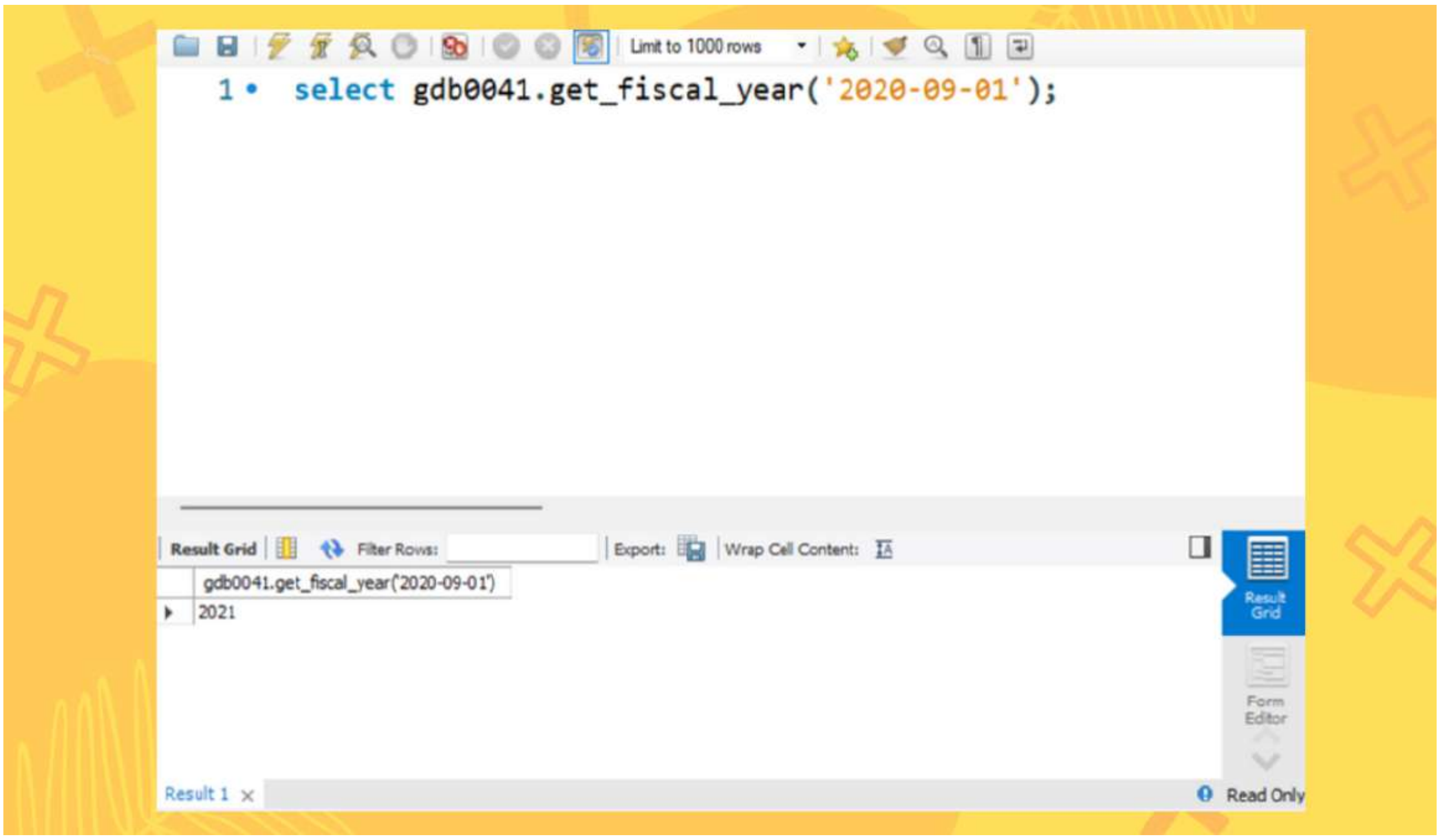
DDL:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `ge
        DETERMINISTIC
                                                alender_date,
```

Call stored function gdb0041.get_fiscal_year

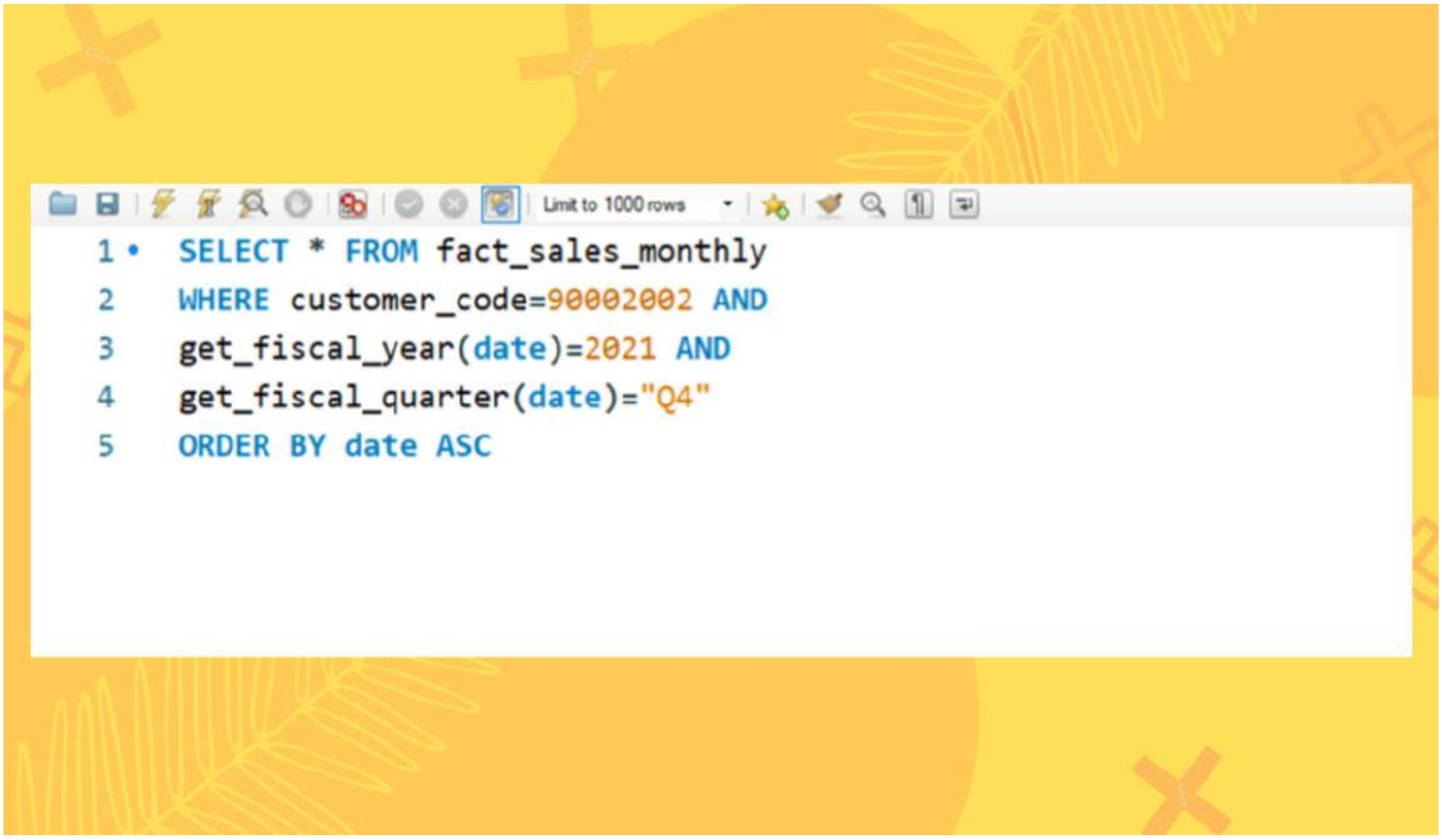
Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

calender_date 2020-09-01 DATE

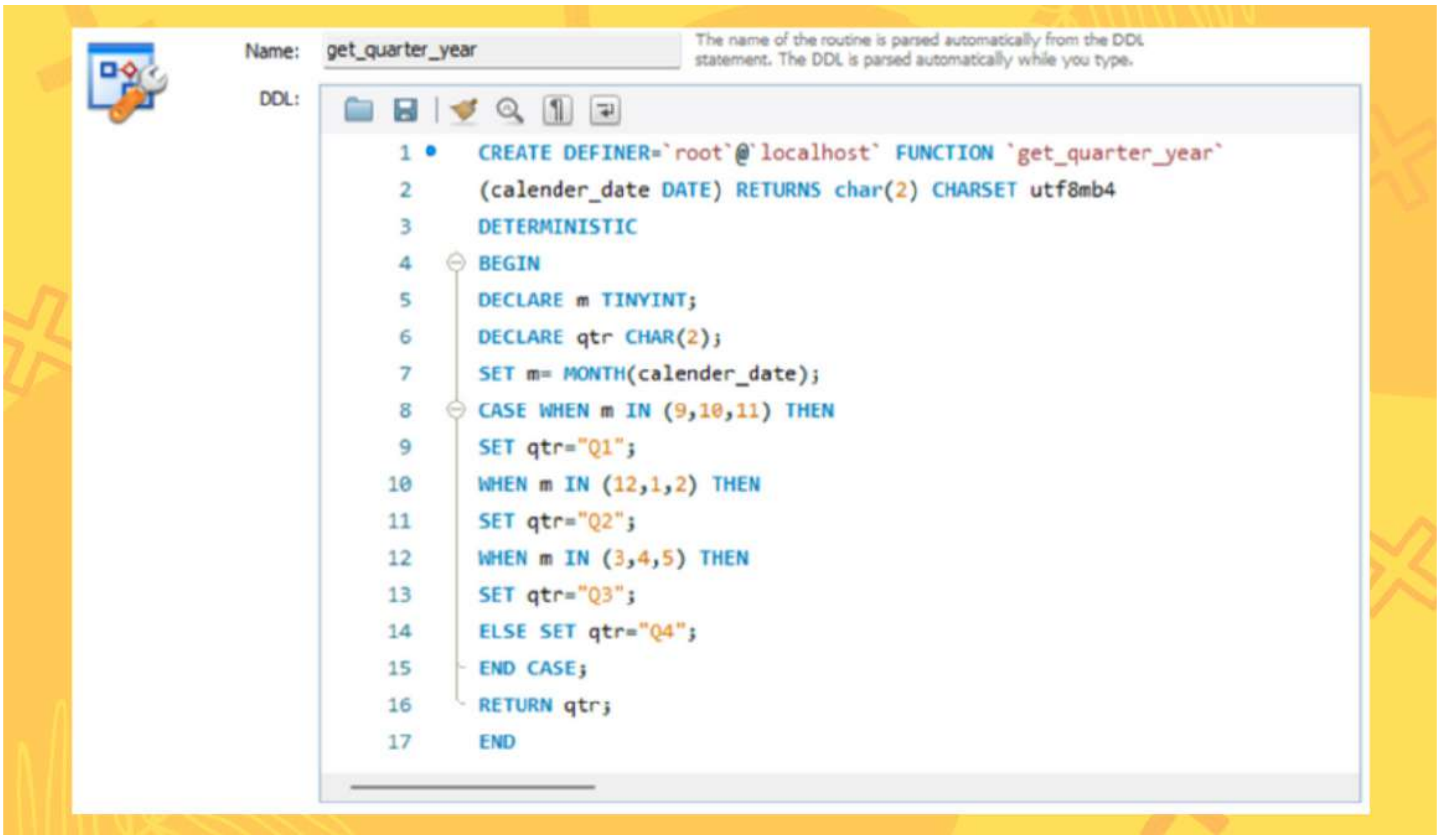


```
select gdb0041.get_fiscal_year('2020-09-01');
```

```
gdb0041.get_fiscal_year('2020-09-01')
2021
```



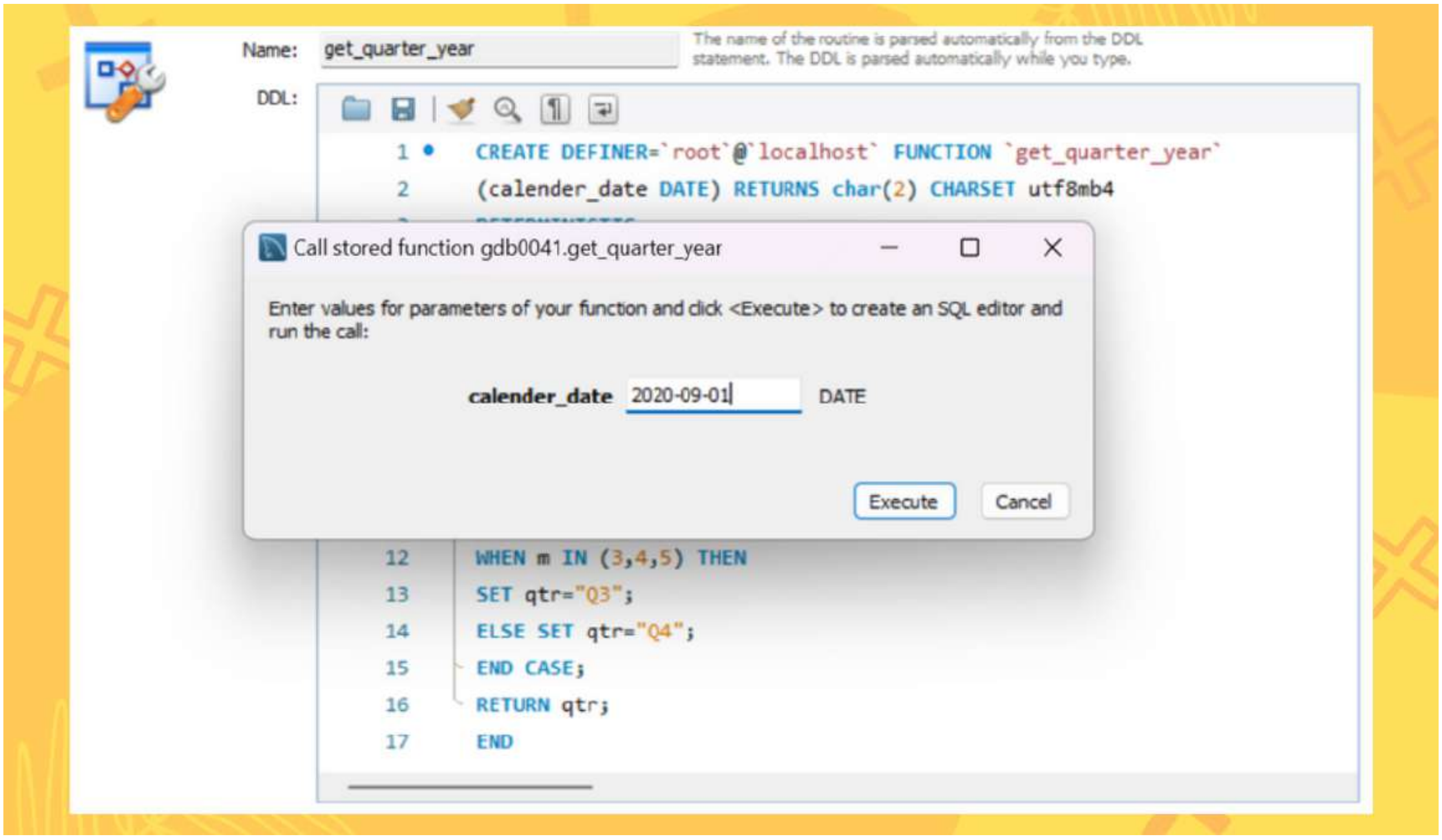
```sql
SELECT * FROM fact_sales_monthly
WHERE customer_code=90002002 AND
get_fiscal_year(date)=2021 AND
get_fiscal_quarter(date)="Q4"
ORDER BY date ASC
```


Name: get_quarter_year

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_quarter_year`
(calender_date DATE) RETURNS char(2) CHARSET utf8mb4
DETERMINISTIC
BEGIN
DECLARE m TINYINT;
DECLARE qtr CHAR(2);
SET m= MONTH(calender_date);
CASE WHEN m IN (9,10,11) THEN
SET qtr="Q1";
WHEN m IN (12,1,2) THEN
SET qtr="Q2";
WHEN m IN (3,4,5) THEN
SET qtr="Q3";
ELSE SET qtr="Q4";
END CASE;
RETURN qtr;
END
```


Name: get_quarter_year

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

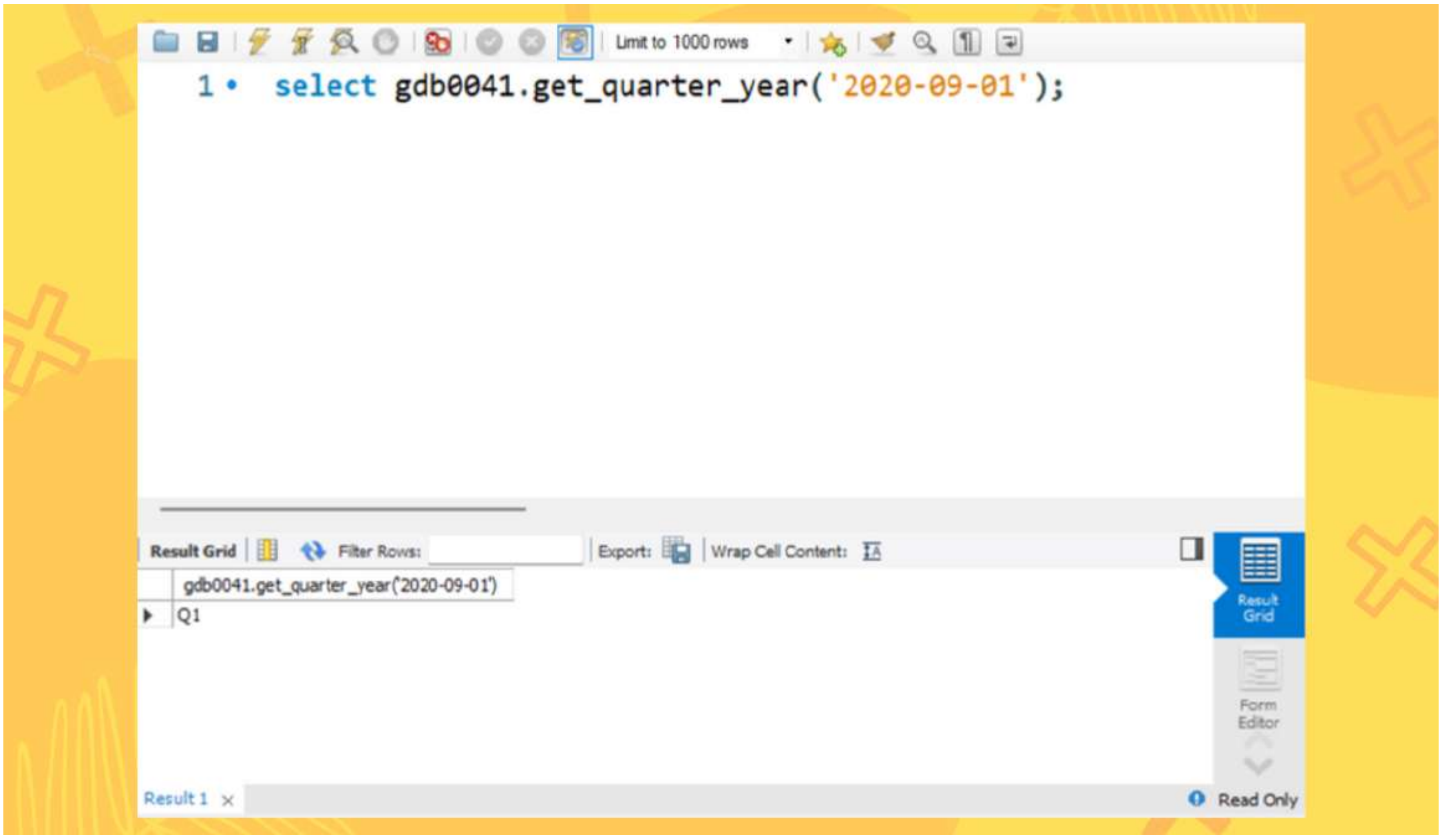
```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_quarter_year`
(calender_date DATE) RETURNS char(2) CHARSET utf8mb4
```

Call stored function gdb0041.get_quarter_year

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

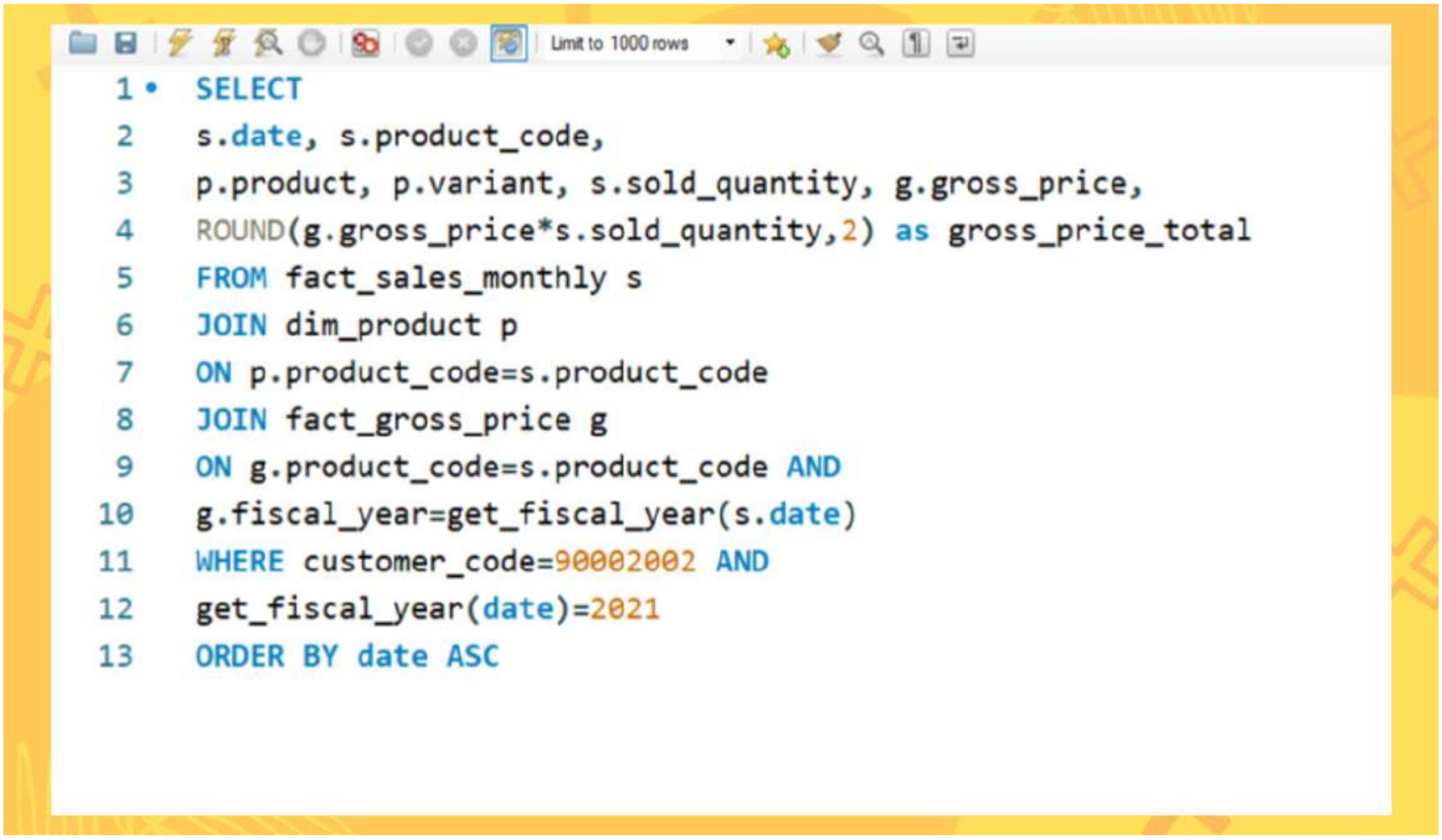
calender_date 2020-09-01 DATE

```
WHEN m IN (3,4,5) THEN
SET qtr="Q3";
ELSE SET qtr="Q4";
END CASE;
RETURN qtr;
END
```

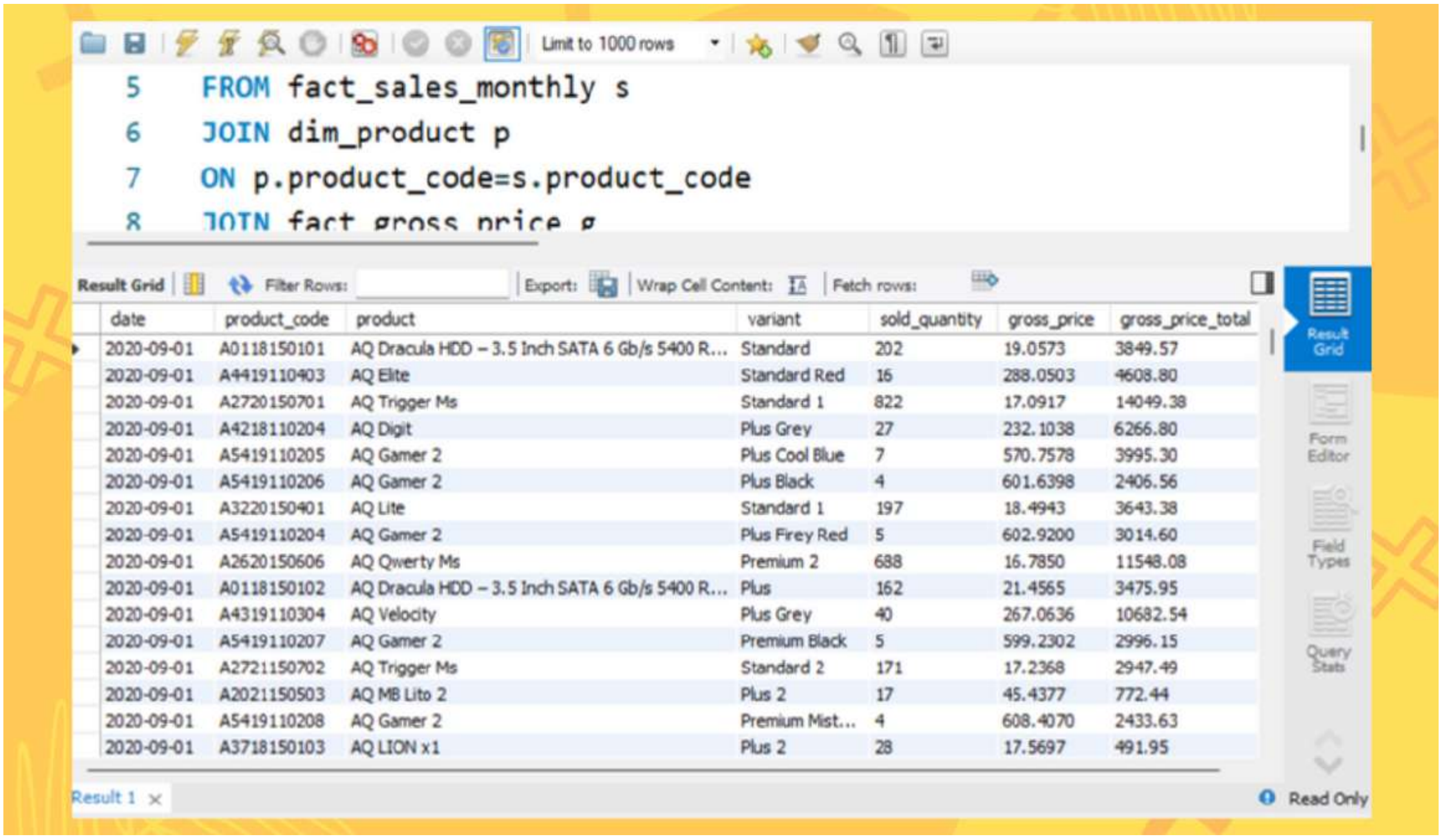


```
select gdb0041.get_quarter_year('2020-09-01');
```

```
gdb0041.get_quarter_year('2020-09-01')
Q1
```



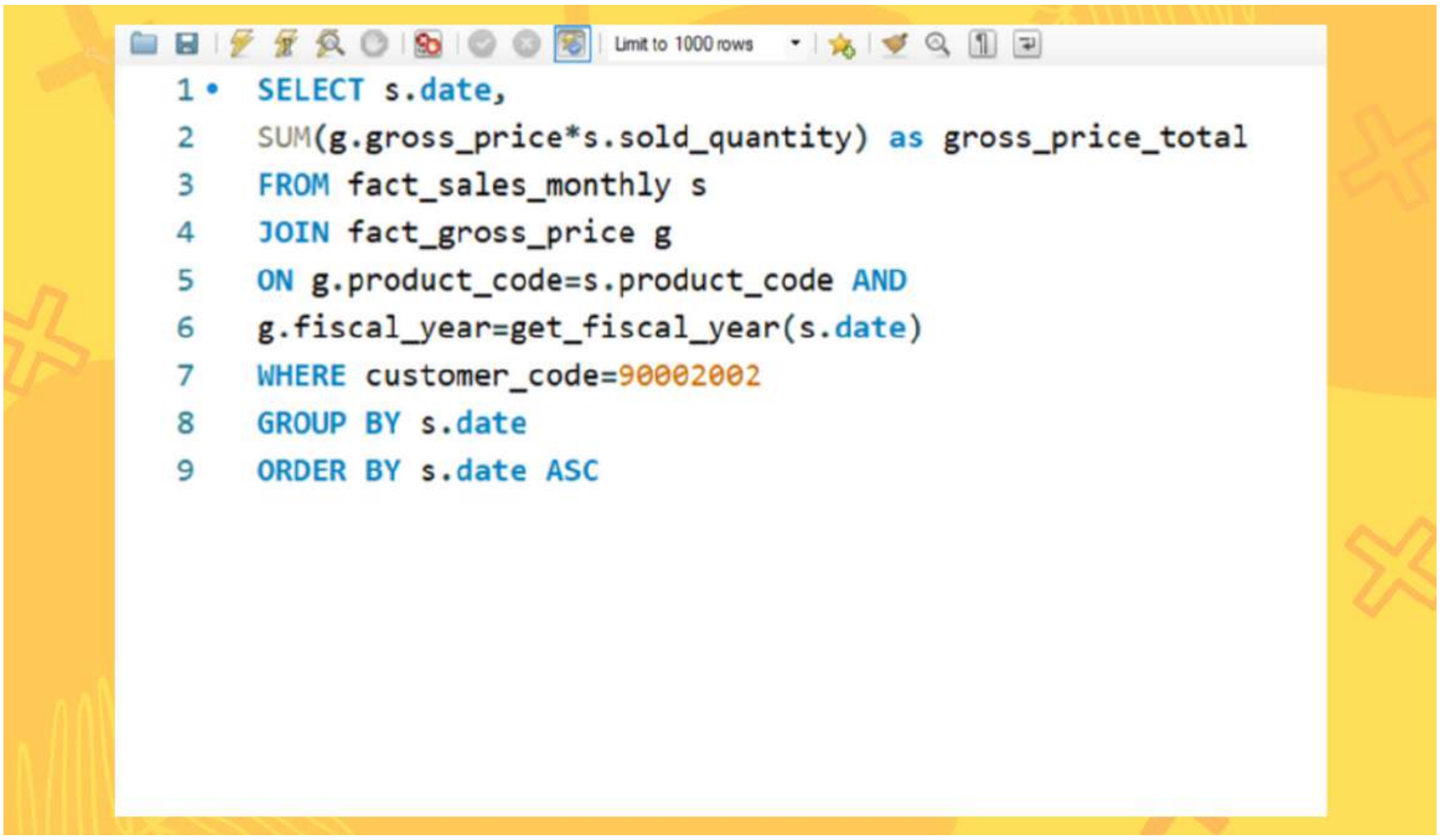
```sql
SELECT
s.date, s.product_code,
p.product, p.variant, s.sold_quantity, g.gross_price,
ROUND(g.gross_price*s.sold_quantity,2) as gross_price_total
FROM fact_sales_monthly s
JOIN dim_product p
ON p.product_code=s.product_code
JOIN fact_gross_price g
ON g.product_code=s.product_code AND
g.fiscal_year=get_fiscal_year(s.date)
WHERE customer_code=90002002 AND
get_fiscal_year(date)=2021
ORDER BY date ASC
```



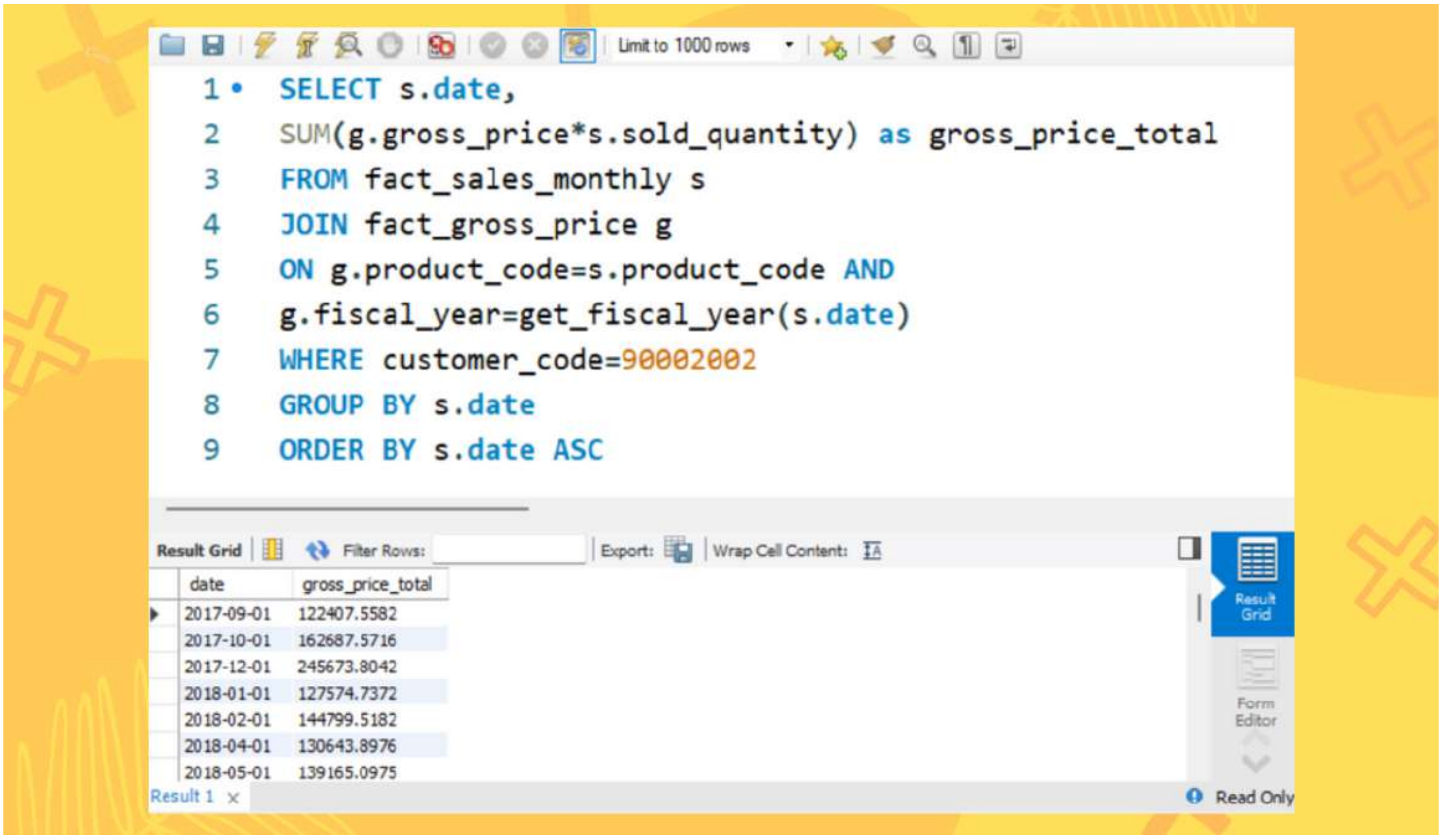
```sql
FROM fact_sales_monthly s
JOIN dim_product p
ON p.product_code=s.product_code
JOIN fact gross price g
```

date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2020-09-01	A0118150101	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Standard	202	19.0573	3849.57
2020-09-01	A4419110403	AQ Elite	Standard Red	16	288.0503	4608.80
2020-09-01	A2720150701	AQ Trigger Ms	Standard 1	822	17.0917	14049.38
2020-09-01	A4218110204	AQ Digit	Plus Grey	27	232.1038	6266.80
2020-09-01	A5419110205	AQ Gamer 2	Plus Cool Blue	7	570.7578	3995.30
2020-09-01	A5419110206	AQ Gamer 2	Plus Black	4	601.6398	2406.56
2020-09-01	A3220150401	AQ Lite	Standard 1	197	18.4943	3643.38
2020-09-01	A5419110204	AQ Gamer 2	Plus Firey Red	5	602.9200	3014.60
2020-09-01	A2620150606	AQ Qwerty Ms	Premium 2	688	16.7850	11548.08
2020-09-01	A0118150102	AQ Dracula HDD – 3.5 Inch SATA 6 Gb/s 5400 R...	Plus	162	21.4565	3475.95
2020-09-01	A4319110304	AQ Velocity	Plus Grey	40	267.0636	10682.54
2020-09-01	A5419110207	AQ Gamer 2	Premium Black	5	599.2302	2996.15
2020-09-01	A2721150702	AQ Trigger Ms	Standard 2	171	17.2368	2947.49
2020-09-01	A2021150503	AQ MB Lito 2	Plus 2	17	45.4377	772.44
2020-09-01	A5419110208	AQ Gamer 2	Premium Mist...	4	608.4070	2433.63
2020-09-01	A3718150103	AQ LION x1	Plus 2	28	17.5697	491.95

Generate an aggregate monthly gross sales report

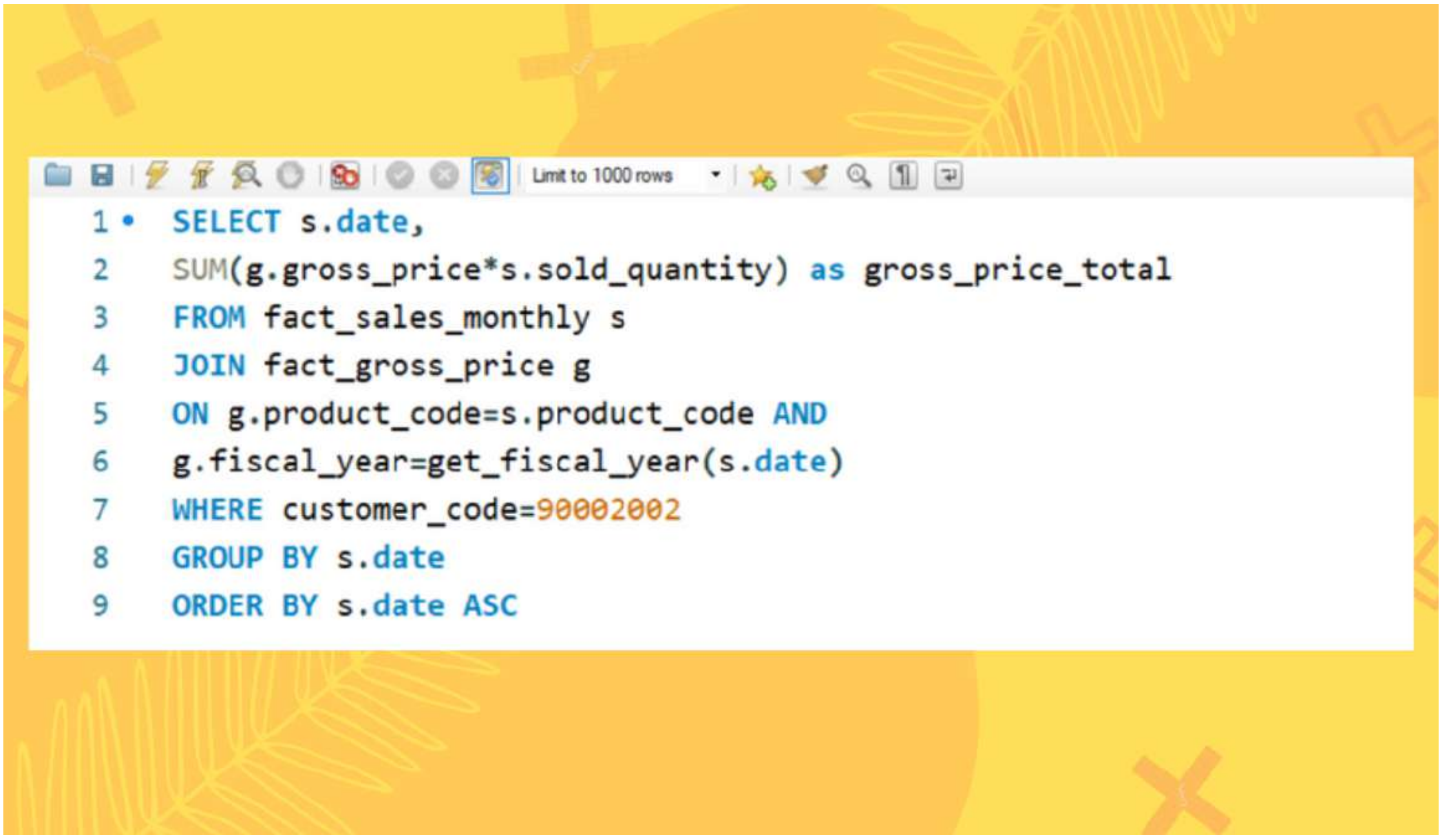


```sql
SELECT s.date,
SUM(g.gross_price*s.sold_quantity) as gross_price_total
FROM fact_sales_monthly s
JOIN fact_gross_price g
ON g.product_code=s.product_code AND
g.fiscal_year=get_fiscal_year(s.date)
WHERE customer_code=90002002
GROUP BY s.date
ORDER BY s.date ASC
```

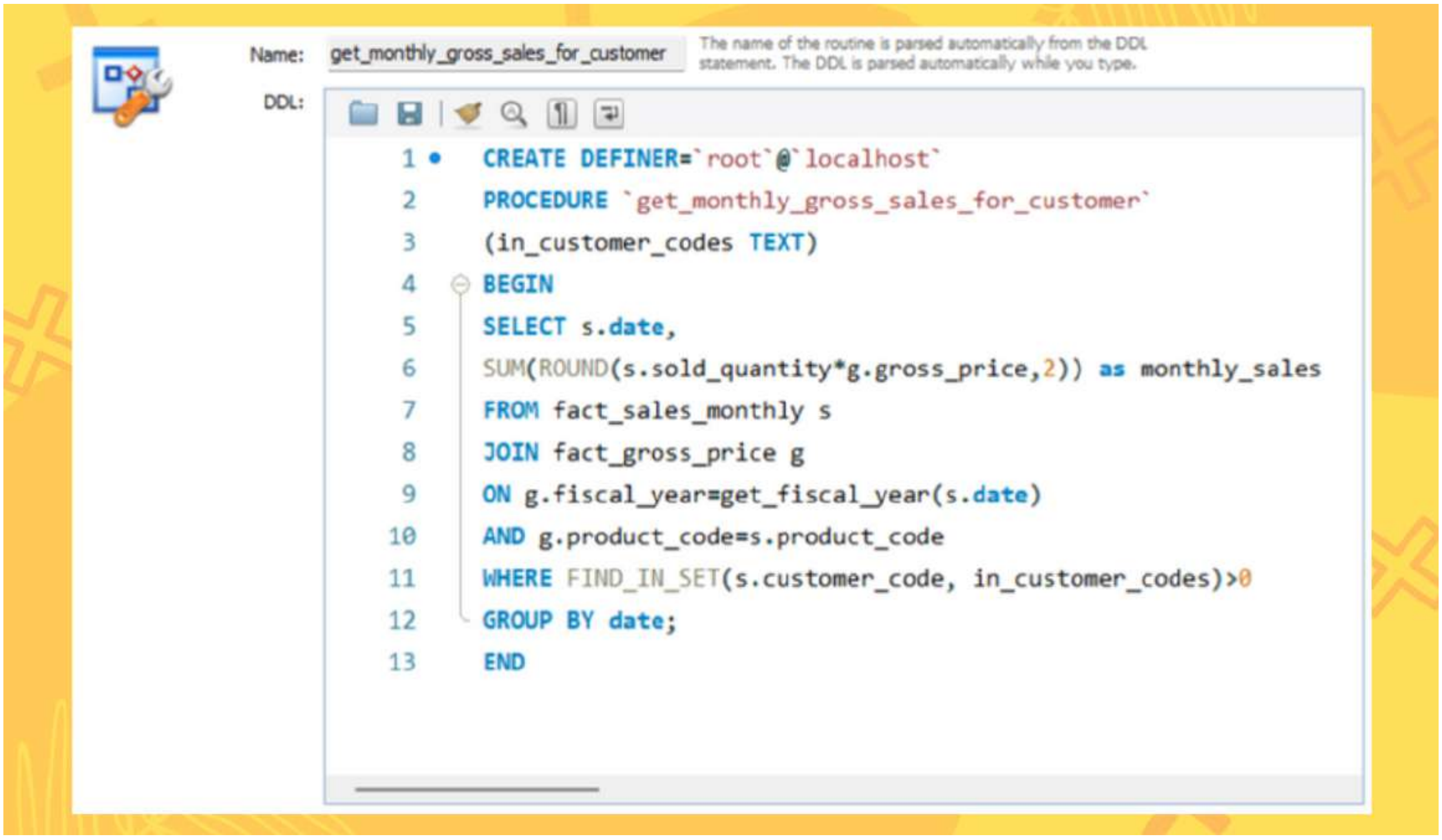


```
SELECT s.date,
SUM(g.gross_price*s.sold_quantity) as gross_price_total
FROM fact_sales_monthly s
JOIN fact_gross_price g
ON g.product_code=s.product_code AND
g.fiscal_year=get_fiscal_year(s.date)
WHERE customer_code=90002002
GROUP BY s.date
ORDER BY s.date ASC
```

date	gross_price_total
2017-09-01	122407.5582
2017-10-01	162687.5716
2017-12-01	245673.8042
2018-01-01	127574.7372
2018-02-01	144799.5182
2018-04-01	130643.8976
2018-05-01	139165.0975



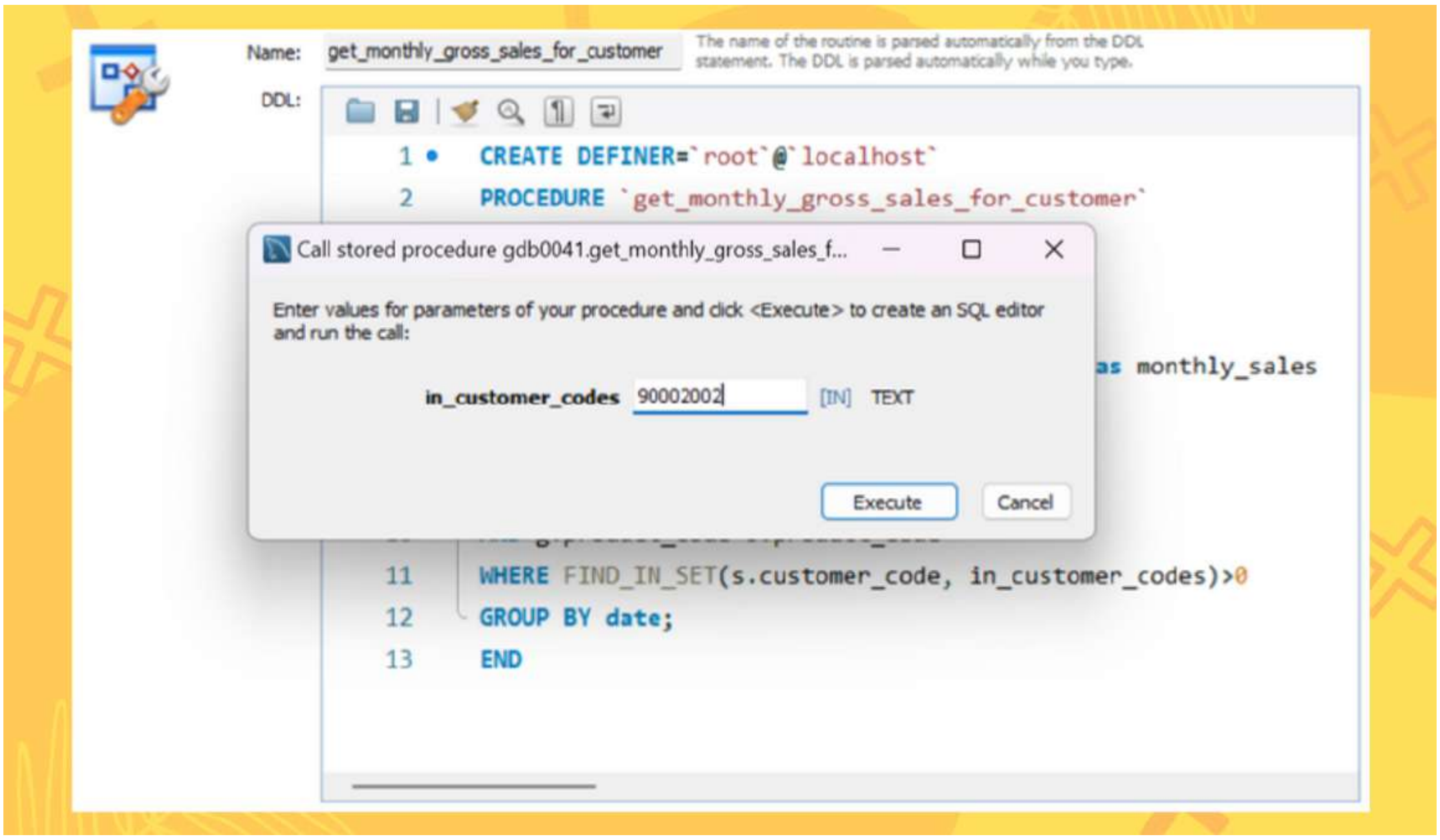
```sql
SELECT s.date,
SUM(g.gross_price*s.sold_quantity) as gross_price_total
FROM fact_sales_monthly s
JOIN fact_gross_price g
ON g.product_code=s.product_code AND
g.fiscal_year=get_fiscal_year(s.date)
WHERE customer_code=90002002
GROUP BY s.date
ORDER BY s.date ASC
```



get_monthly_gross_sales_for_customer

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```sql
CREATE DEFINER=`root`@`localhost`
PROCEDURE `get_monthly_gross_sales_for_customer`
(in_customer_codes TEXT)
BEGIN
SELECT s.date,
SUM(ROUND(s.sold_quantity*g.gross_price,2)) as monthly_sales
FROM fact_sales_monthly s
JOIN fact_gross_price g
ON g.fiscal_year=get_fiscal_year(s.date)
AND g.product_code=s.product_code
WHERE FIND_IN_SET(s.customer_code, in_customer_codes)>0
GROUP BY date;
END
```



Name: get_monthly_gross_sales_for_customer

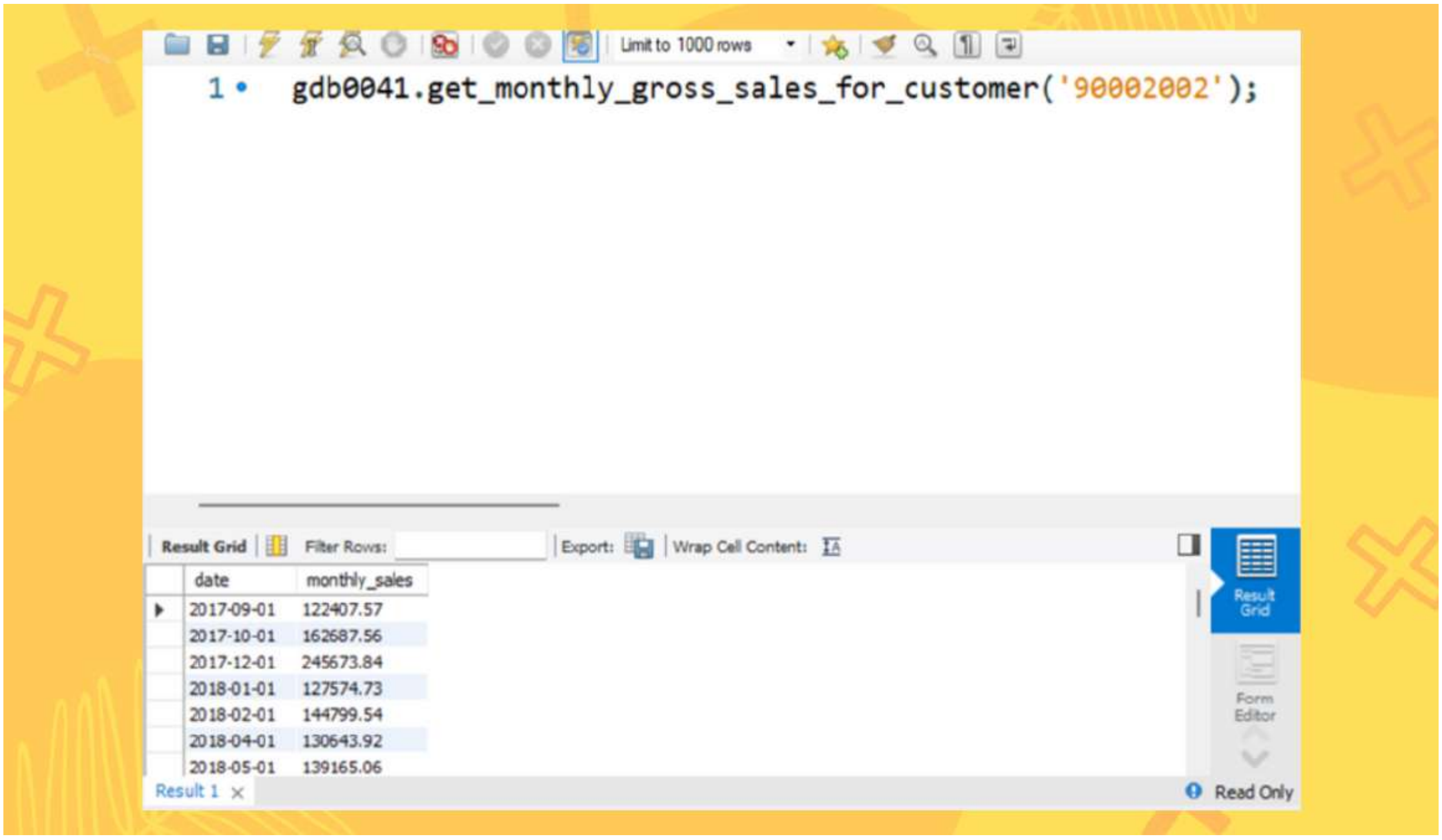
The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```
CREATE DEFINER=`root`@`localhost`
PROCEDURE `get_monthly_gross_sales_for_customer`
                                                  as monthly_sales
WHERE FIND_IN_SET(s.customer_code, in_customer_codes)>0
GROUP BY date;
END
```

Call stored procedure gdb0041.get_monthly_gross_sales_f...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_customer_codes 90002002 [IN] TEXT

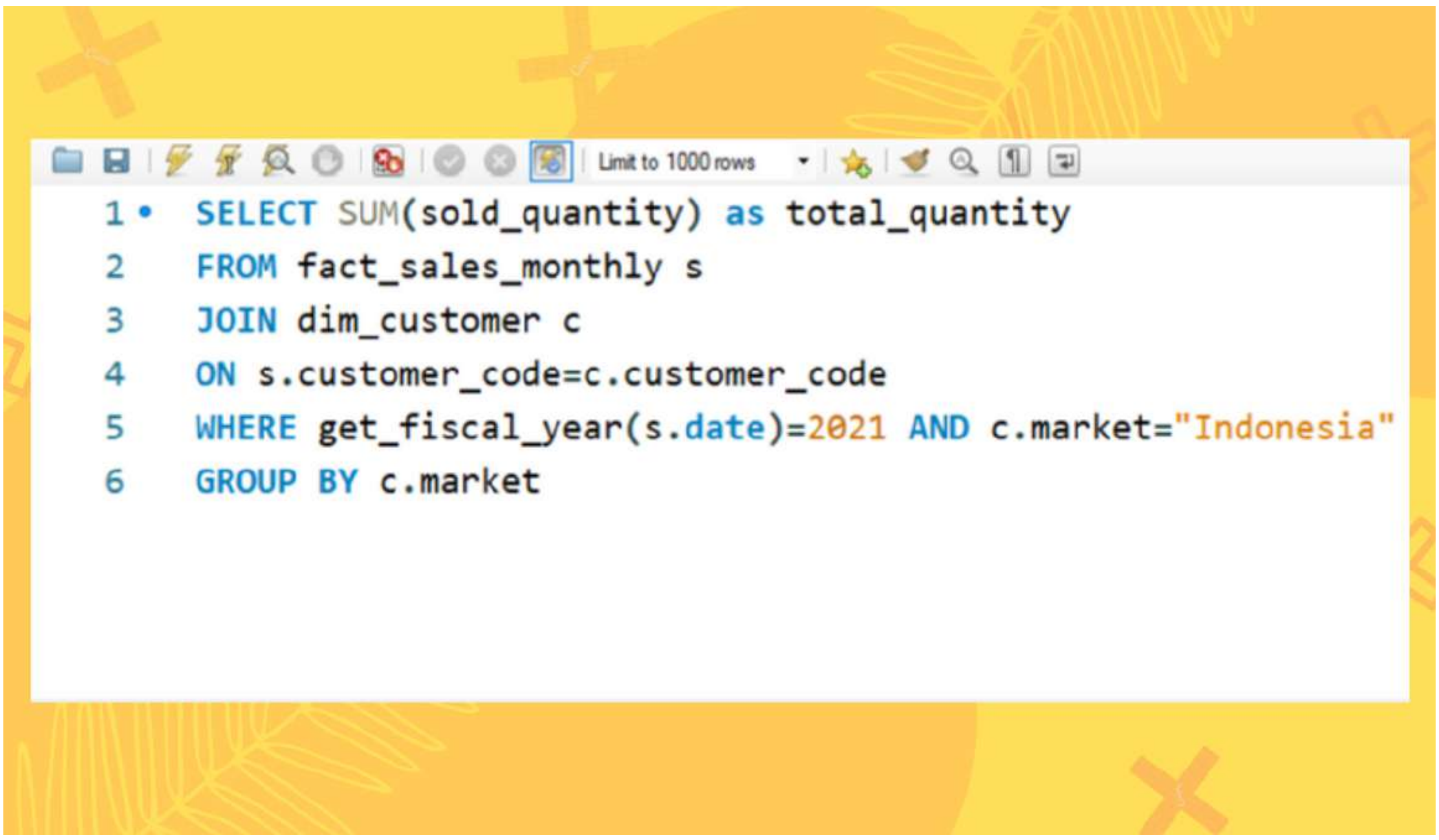


```
gdb0041.get_monthly_gross_sales_for_customer('90002002');
```

```
date        monthly_sales
2017-09-01  122407.57
2017-10-01  162687.56
2017-12-01  245673.84
2018-01-01  127574.73
2018-02-01  144799.54
2018-04-01  130643.92
2018-05-01  139165.06
```

Create a stored procedure thatt can determine the market badge on following logic.

If total quantity > 5 million that market is considered Gold else it is Silver



```sql
SELECT SUM(sold_quantity) as total_quantity
FROM fact_sales_monthly s
JOIN dim_customer c
ON s.customer_code=c.customer_code
WHERE get_fiscal_year(s.date)=2021 AND c.market="Indonesia"
GROUP BY c.market
```