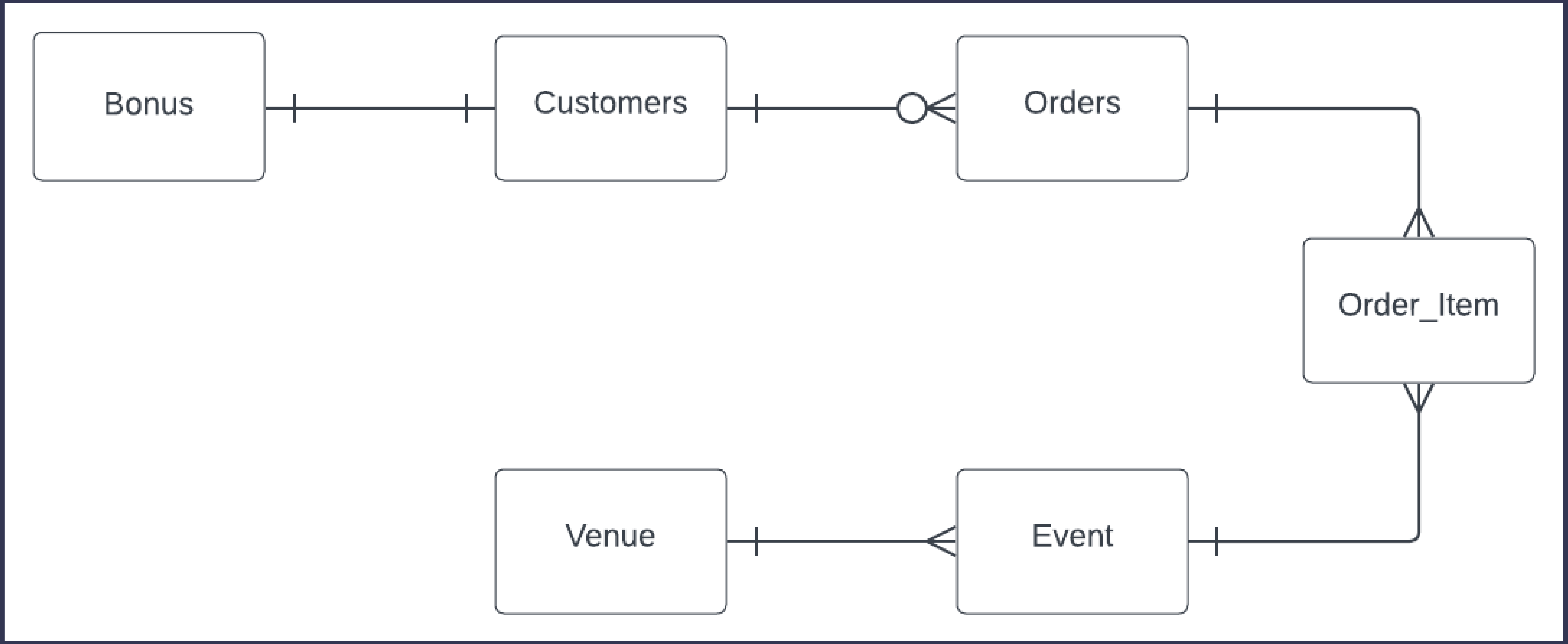
LARANA, INC.

FINAL PROJECT

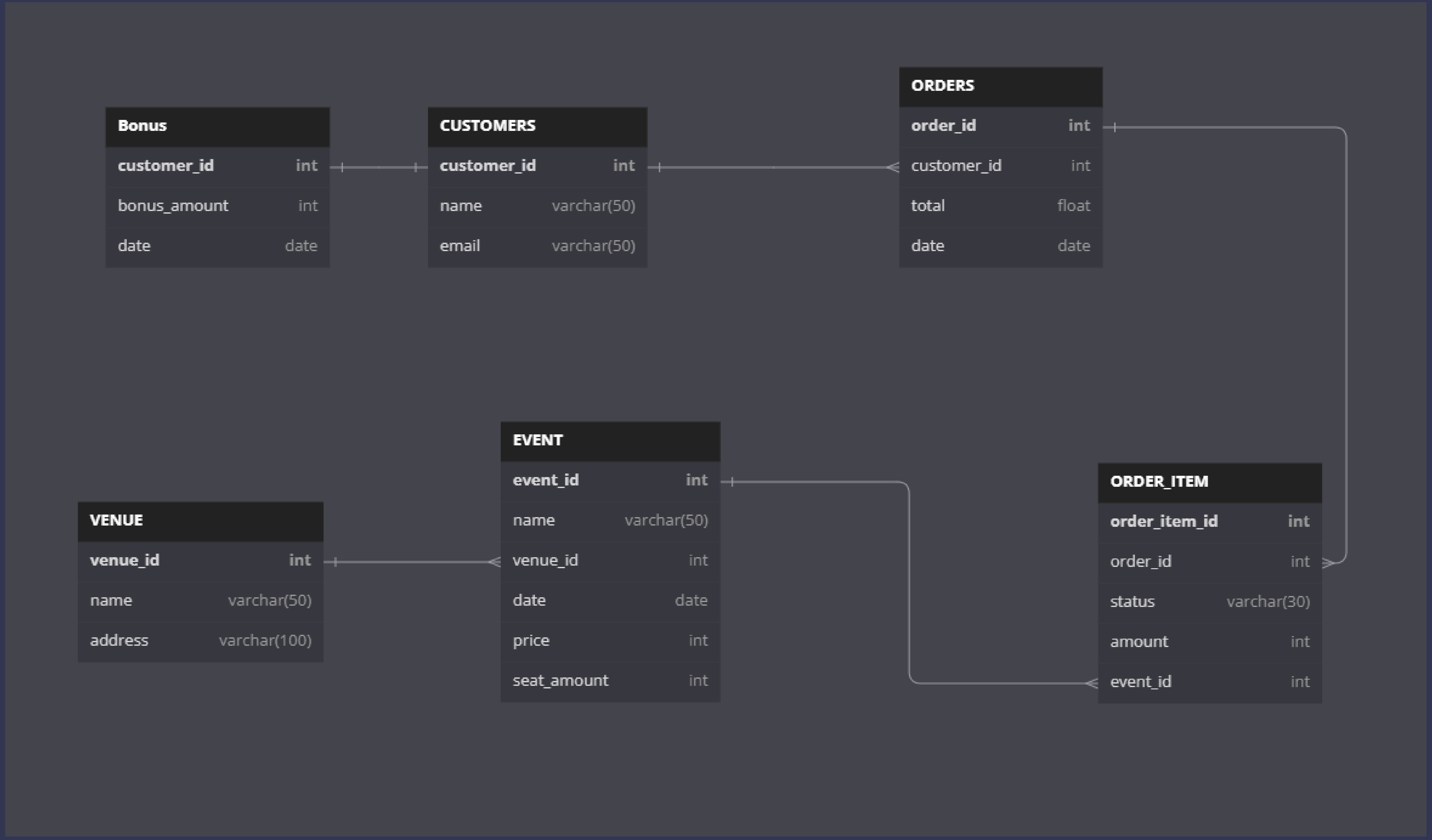
TOPIC - TICKETON.KZ

ALIMGOZHA ADIL
IBRAGIM ALDIAR
SALYKBAYEV BEKZHAN
ORYNGALIEV BAUYRZHAN

ERD



Conceptual Design



Bonus	
customer_id	int
bonus_amount	int
date	date

CUSTOMERS	
customer_id	int
name	varchar(50)
email	varchar(50)

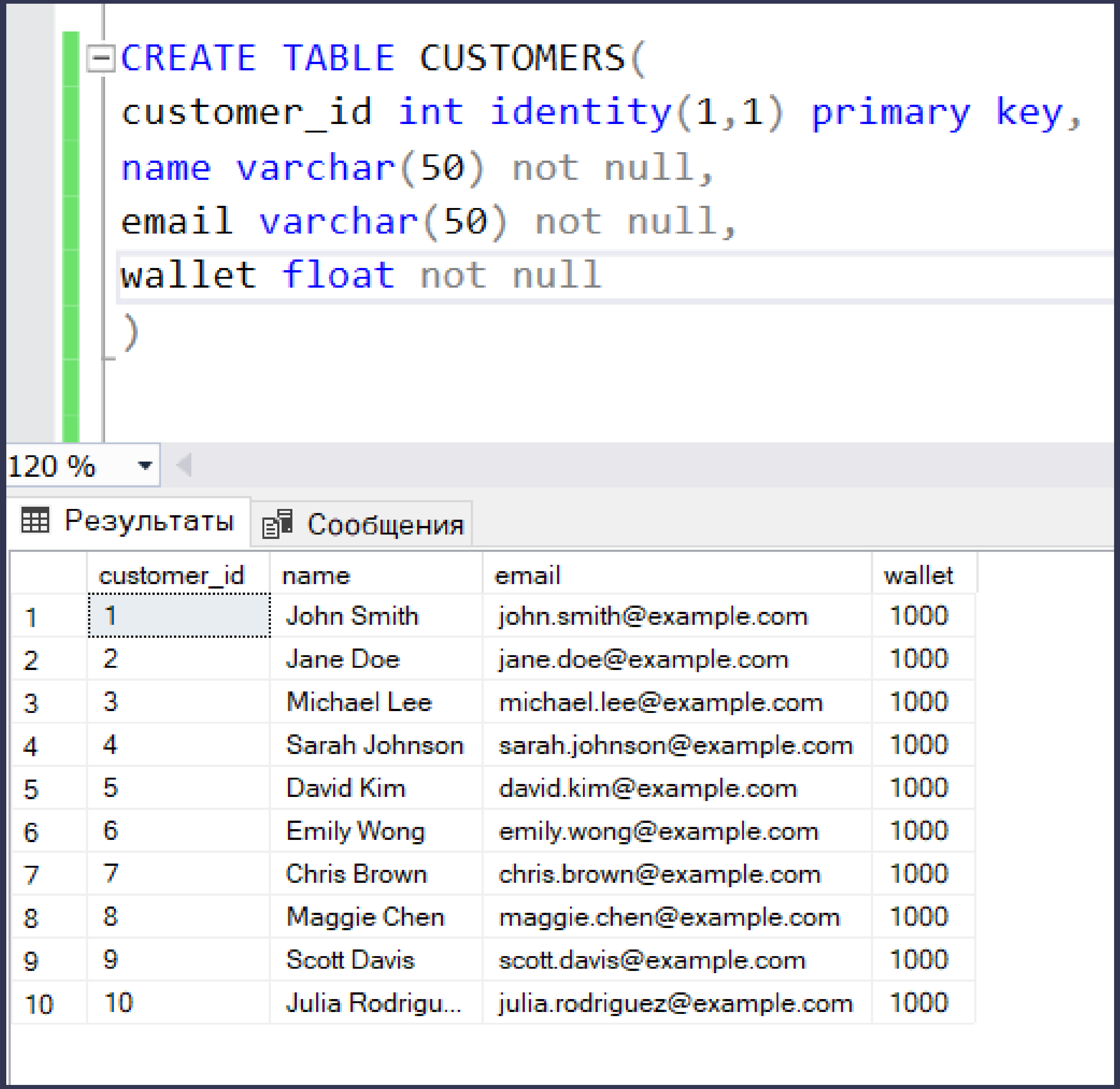
ORDERS	
order_id	int
customer_id	int
total	float
date	date

VENUE	
venue_id	int
name	varchar(50)
address	varchar(100)

EVENT	
event_id	int
name	varchar(50)
venue_id	int
date	date
price	int
seat_amount	int

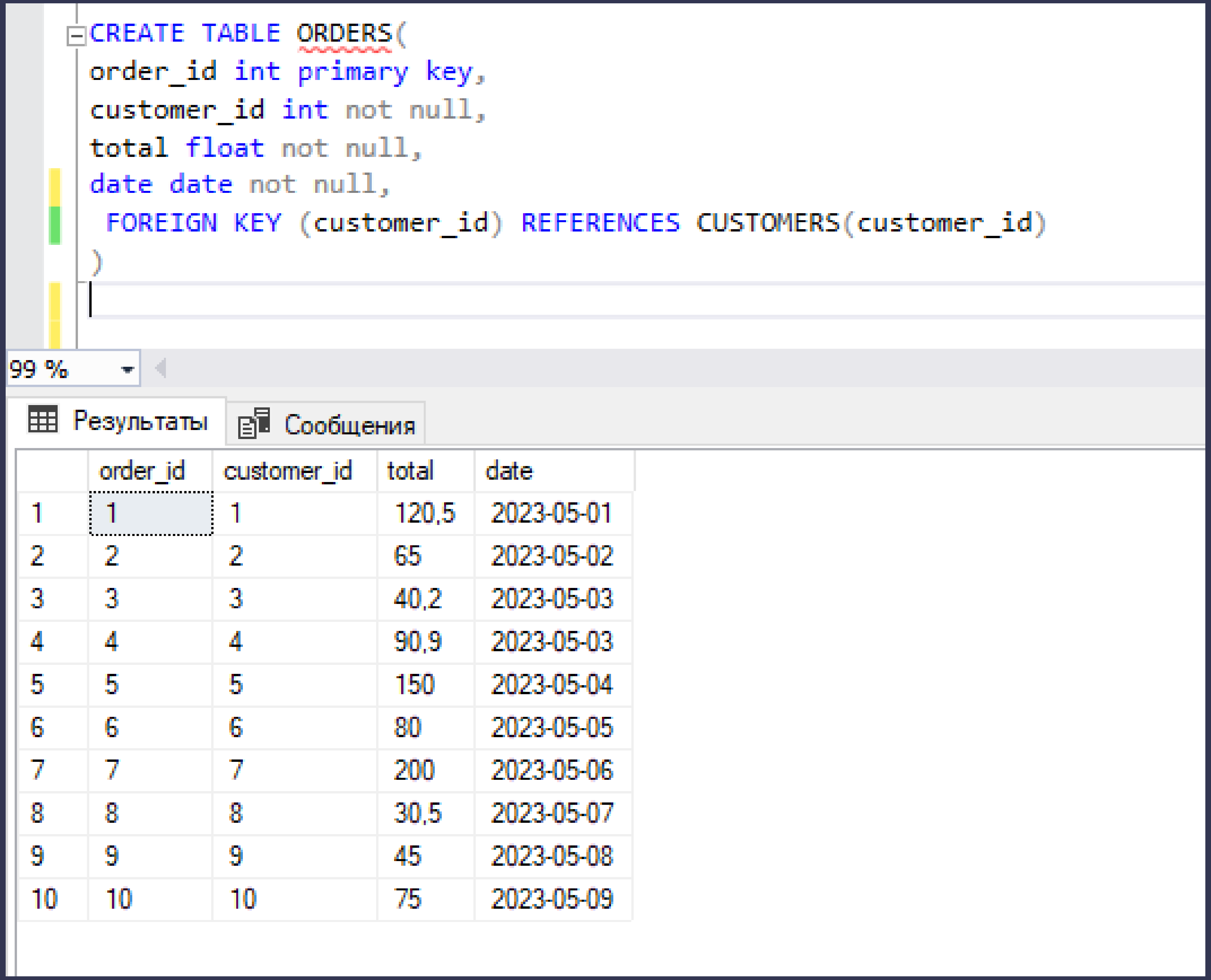
ORDER_ITEM	
order_item_id	int
order_id	int
status	varchar(30)
amount	int
event_id	int

TABLES



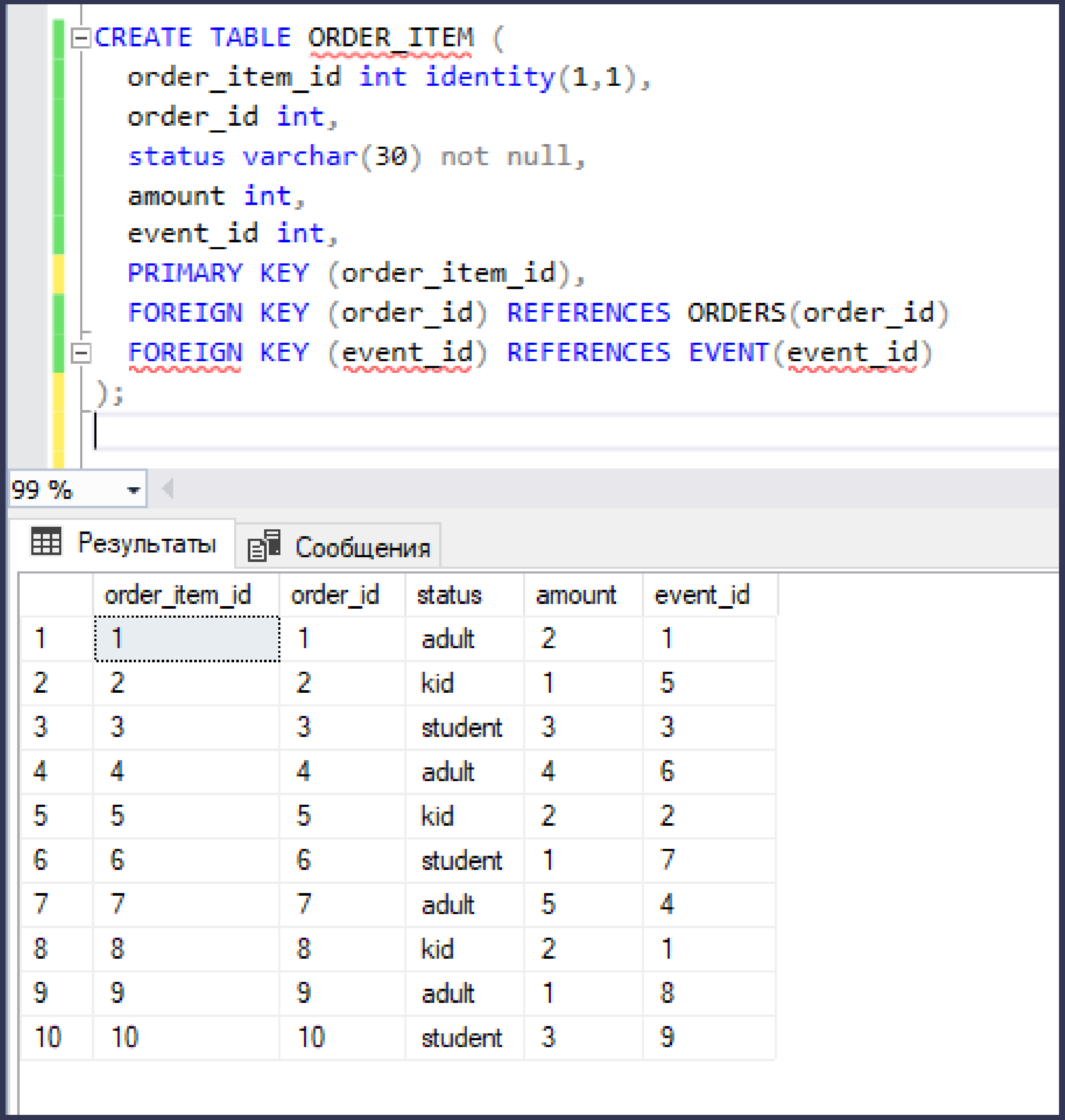
```sql
CREATE TABLE CUSTOMERS(
customer_id int identity(1,1) primary key,
name varchar(50) not null,
email varchar(50) not null,
wallet float not null
)
```

	customer_id	name	email	wallet
1	1	John Smith	john.smith@example.com	1000
2	2	Jane Doe	jane.doe@example.com	1000
3	3	Michael Lee	michael.lee@example.com	1000
4	4	Sarah Johnson	sarah.johnson@example.com	1000
5	5	David Kim	david.kim@example.com	1000
6	6	Emily Wong	emily.wong@example.com	1000
7	7	Chris Brown	chris.brown@example.com	1000
8	8	Maggie Chen	maggie.chen@example.com	1000
9	9	Scott Davis	scott.davis@example.com	1000
10	10	Julia Rodrigu...	julia.rodriguez@example.com	1000



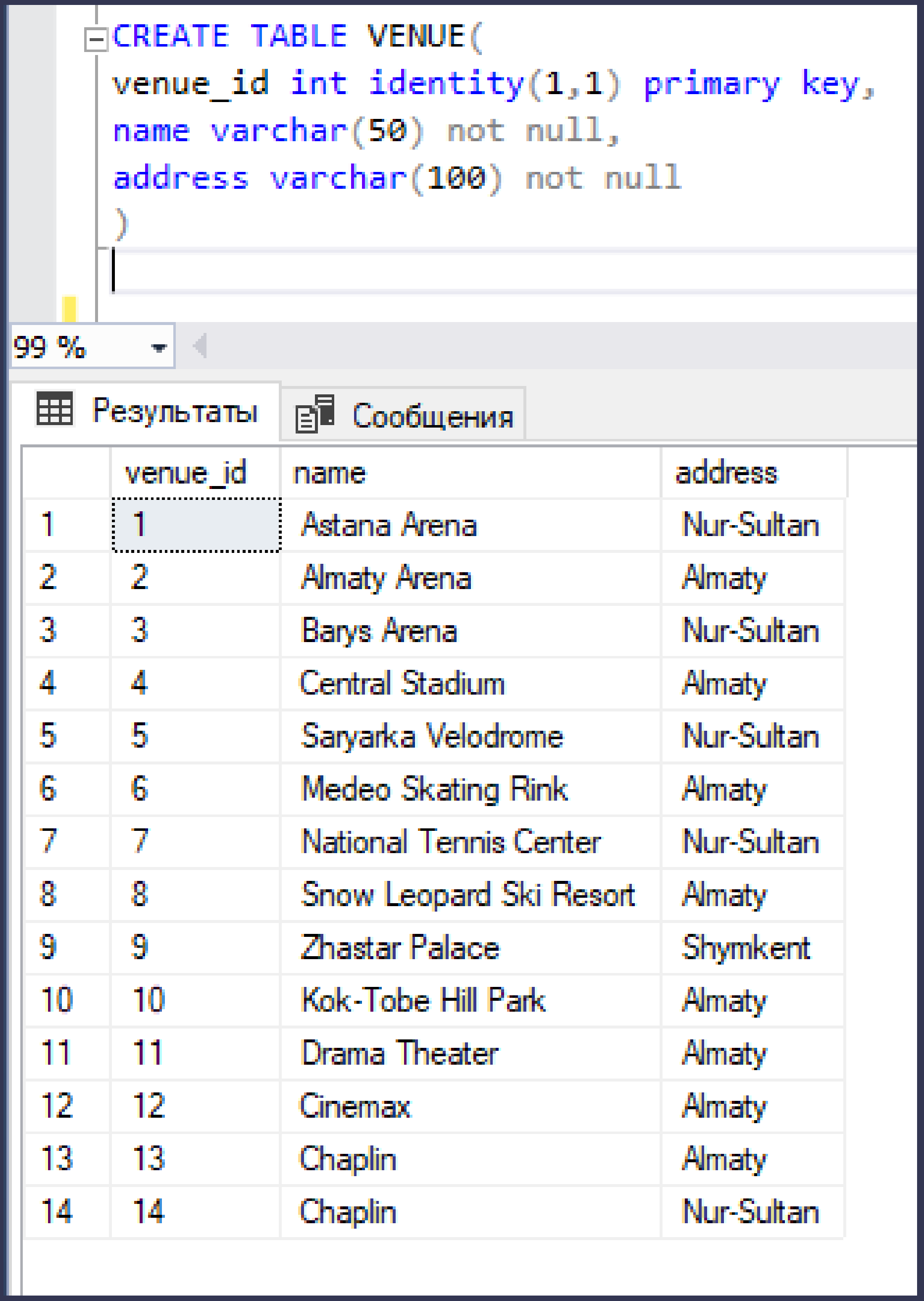
```sql
CREATE TABLE ORDERS(
order_id int primary key,
customer_id int not null,
total float not null,
date date not null,
 FOREIGN KEY (customer_id) REFERENCES CUSTOMERS(customer_id)
)
```

	order_id	customer_id	total	date
1	1	1	120,5	2023-05-01
2	2	2	65	2023-05-02
3	3	3	40,2	2023-05-03
4	4	4	90,9	2023-05-03
5	5	5	150	2023-05-04
6	6	6	80	2023-05-05
7	7	7	200	2023-05-06
8	8	8	30,5	2023-05-07
9	9	9	45	2023-05-08
10	10	10	75	2023-05-09



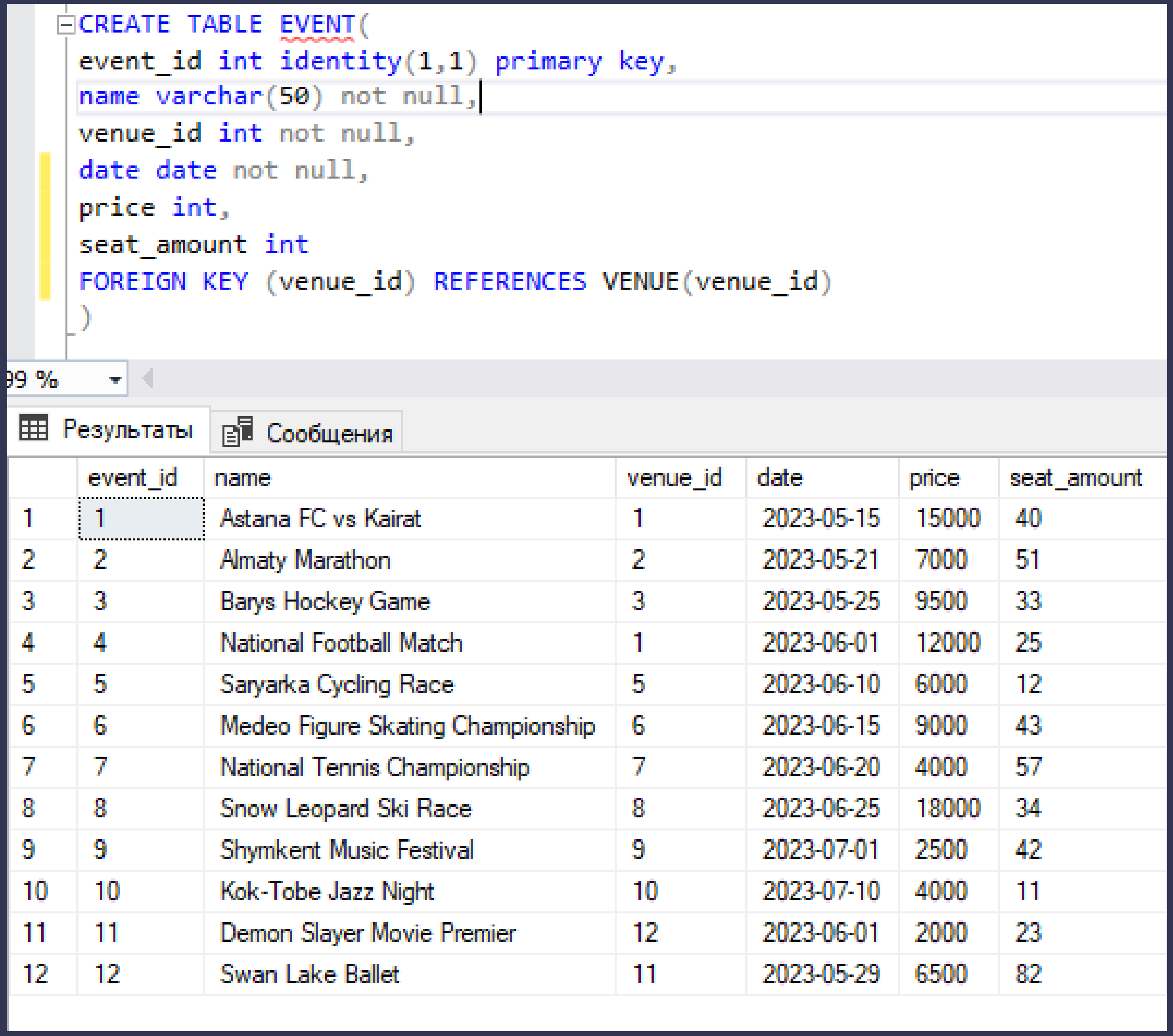
```
CREATE TABLE ORDER_ITEM (
    order_item_id int identity(1,1),
    order_id int,
    status varchar(30) not null,
    amount int,
    event_id int,
    PRIMARY KEY (order_item_id),
    FOREIGN KEY (order_id) REFERENCES ORDERS(order_id)
    FOREIGN KEY (event_id) REFERENCES EVENT(event_id)
);
```

	order_item_id	order_id	status	amount	event_id
1	1	1	adult	2	1
2	2	2	kid	1	5
3	3	3	student	3	3
4	4	4	adult	4	6
5	5	5	kid	2	2
6	6	6	student	1	7
7	7	7	adult	5	4
8	8	8	kid	2	1
9	9	9	adult	1	8
10	10	10	student	3	9



```
CREATE TABLE VENUE(
venue_id int identity(1,1) primary key,
name varchar(50) not null,
address varchar(100) not null
)
```

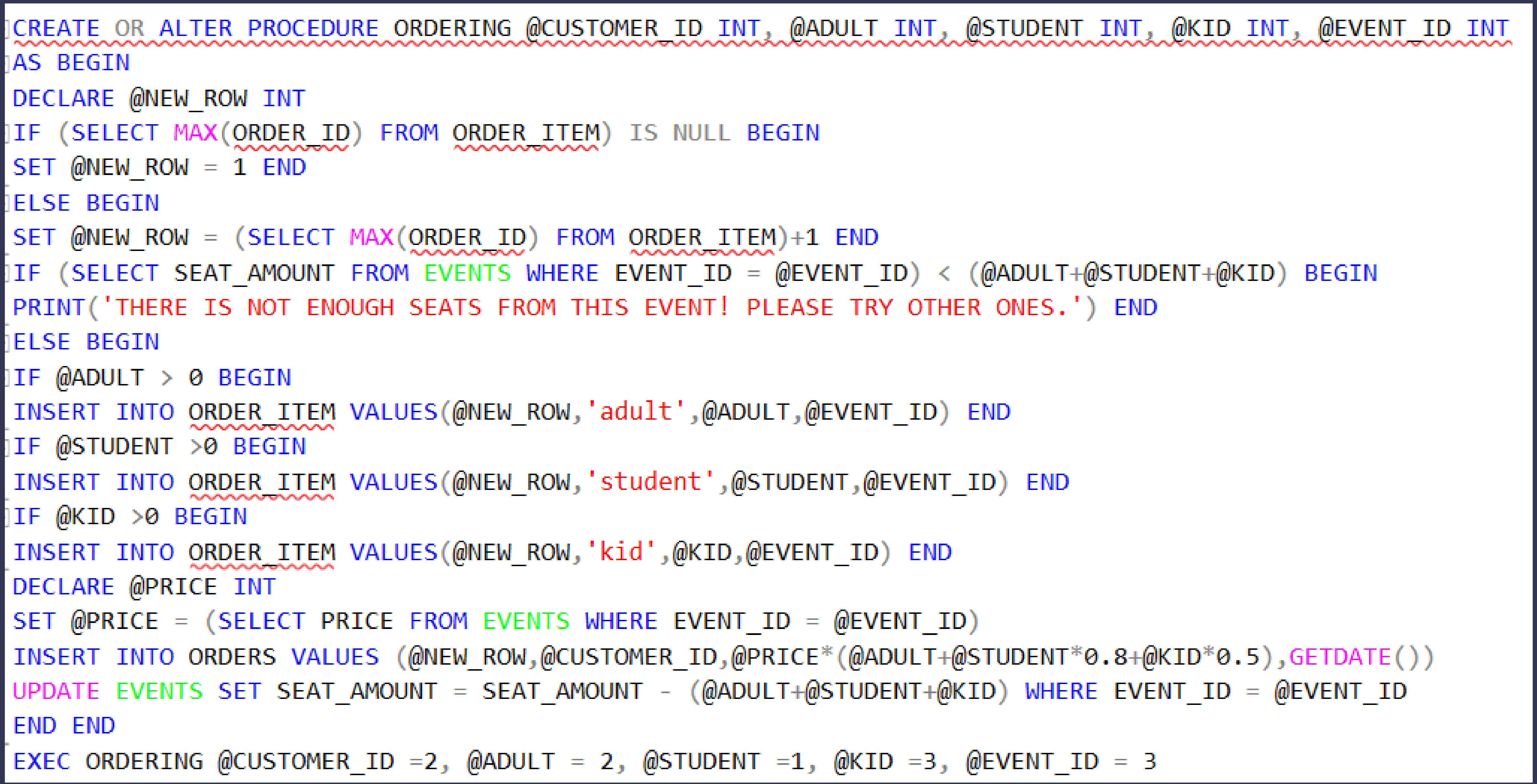
	venue_id	name	address
1	1	Astana Arena	Nur-Sultan
2	2	Almaty Arena	Almaty
3	3	Barys Arena	Nur-Sultan
4	4	Central Stadium	Almaty
5	5	Saryarka Velodrome	Nur-Sultan
6	6	Medeo Skating Rink	Almaty
7	7	National Tennis Center	Nur-Sultan
8	8	Snow Leopard Ski Resort	Almaty
9	9	Zhastar Palace	Shymkent
10	10	Kok-Tobe Hill Park	Almaty
11	11	Drama Theater	Almaty
12	12	Cinemax	Almaty
13	13	Chaplin	Almaty
14	14	Chaplin	Nur-Sultan



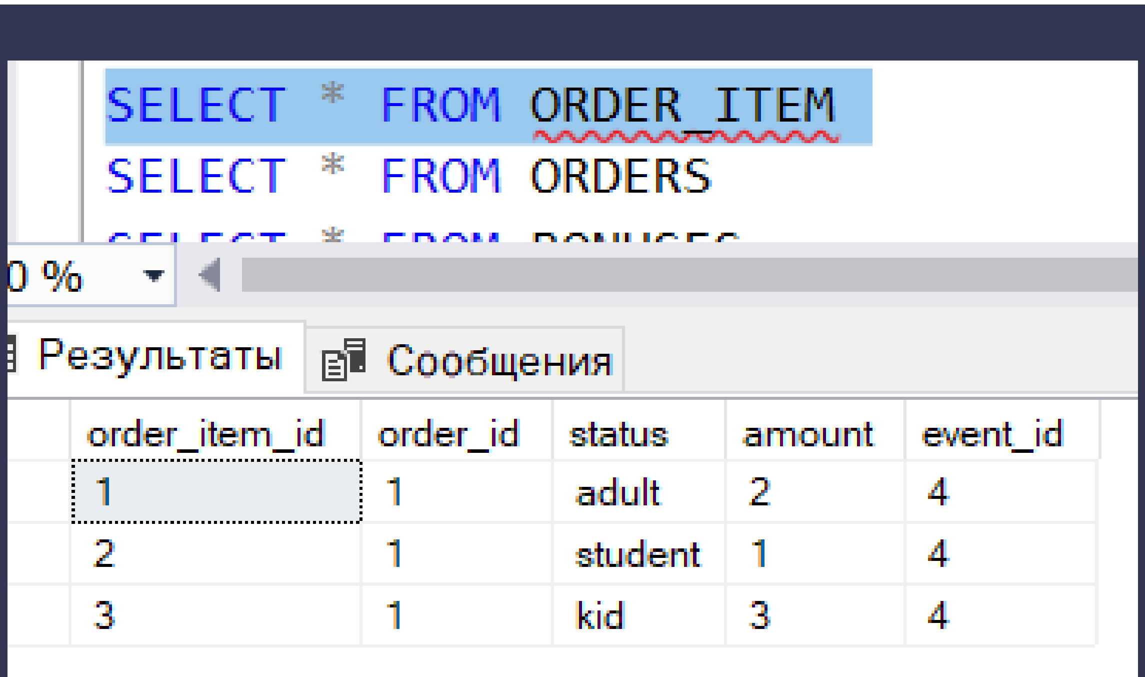
```
CREATE TABLE EVENT(
event_id int identity(1,1) primary key,
name varchar(50) not null,
venue_id int not null,
date date not null,
price int,
seat_amount int
FOREIGN KEY (venue_id) REFERENCES VENUE(venue_id)
)
```

	event_id	name	venue_id	date	price	seat_amount
1	1	Astana FC vs Kairat	1	2023-05-15	15000	40
2	2	Almaty Marathon	2	2023-05-21	7000	51
3	3	Barys Hockey Game	3	2023-05-25	9500	33
4	4	National Football Match	1	2023-06-01	12000	25
5	5	Saryarka Cycling Race	5	2023-06-10	6000	12
6	6	Medeo Figure Skating Championship	6	2023-06-15	9000	43
7	7	National Tennis Championship	7	2023-06-20	4000	57
8	8	Snow Leopard Ski Race	8	2023-06-25	18000	34
9	9	Shymkent Music Festival	9	2023-07-01	2500	42
10	10	Kok-Tobe Jazz Night	10	2023-07-10	4000	11
11	11	Demon Slayer Movie Premier	12	2023-06-01	2000	23
12	12	Swan Lake Ballet	11	2023-05-29	6500	82

1

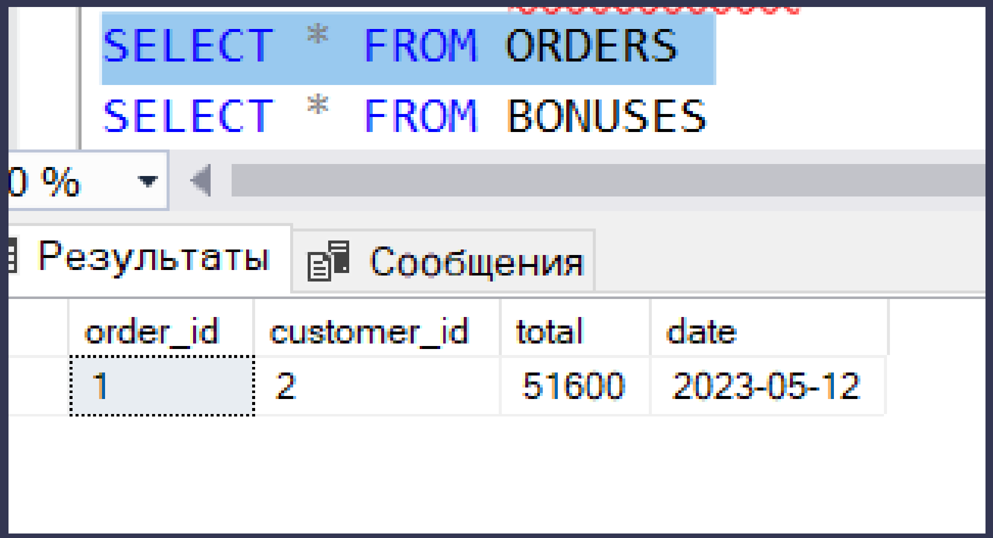


```
CREATE OR ALTER PROCEDURE ORDERING @CUSTOMER_ID INT, @ADULT INT, @STUDENT INT, @KID INT, @EVENT_ID INT
AS BEGIN
DECLARE @NEW_ROW INT
IF (SELECT MAX(ORDER_ID) FROM ORDER_ITEM) IS NULL BEGIN
SET @NEW_ROW = 1 END
ELSE BEGIN
SET @NEW_ROW = (SELECT MAX(ORDER_ID) FROM ORDER_ITEM)+1 END
IF (SELECT SEAT_AMOUNT FROM EVENTS WHERE EVENT_ID = @EVENT_ID) < (@ADULT+@STUDENT+@KID) BEGIN
PRINT('THERE IS NOT ENOUGH SEATS FROM THIS EVENT! PLEASE TRY OTHER ONES.') END
ELSE BEGIN
IF @ADULT > 0 BEGIN
INSERT INTO ORDER_ITEM VALUES(@NEW_ROW,'adult',@ADULT,@EVENT_ID) END
IF @STUDENT >0 BEGIN
INSERT INTO ORDER_ITEM VALUES(@NEW_ROW,'student',@STUDENT,@EVENT_ID) END
IF @KID >0 BEGIN
INSERT INTO ORDER_ITEM VALUES(@NEW_ROW,'kid',@KID,@EVENT_ID) END
DECLARE @PRICE INT
SET @PRICE = (SELECT PRICE FROM EVENTS WHERE EVENT_ID = @EVENT_ID)
INSERT INTO ORDERS VALUES (@NEW_ROW,@CUSTOMER_ID,@PRICE*(@ADULT+@STUDENT*0.8+@KID*0.5),GETDATE())
UPDATE EVENTS SET SEAT_AMOUNT = SEAT_AMOUNT - (@ADULT+@STUDENT+@KID) WHERE EVENT_ID = @EVENT_ID
END END
EXEC ORDERING @CUSTOMER_ID =2, @ADULT = 2, @STUDENT =1, @KID =3, @EVENT_ID = 3
```



```
SELECT * FROM ORDER_ITEM
SELECT * FROM ORDERS
SELECT * FROM BONUSES
```

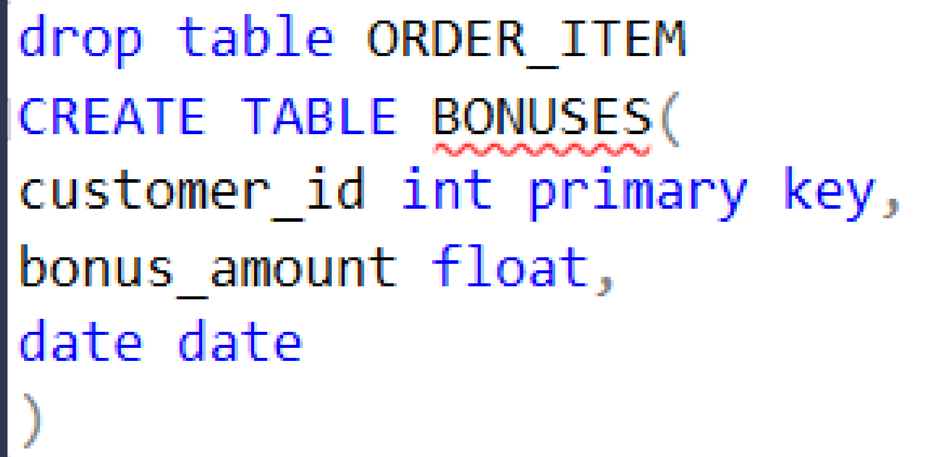
order_item_id	order_id	status	amount	event_id
1	1	adult	2	4
2	1	student	1	4
3	1	kid	3	4



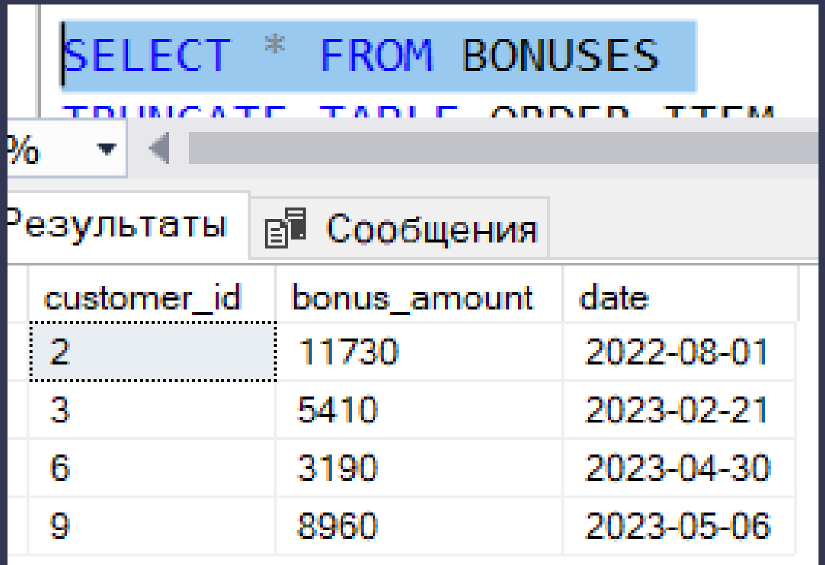
```
SELECT * FROM ORDERS
SELECT * FROM BONUSES
```

order_id	customer_id	total	date
1	2	51600	2023-05-12

2

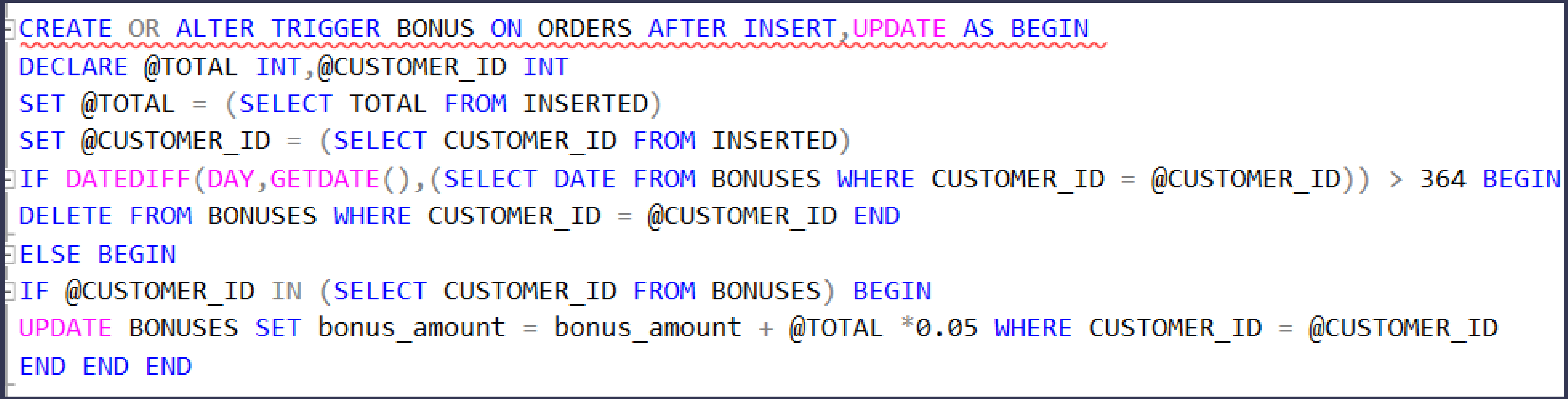


```
drop table ORDER_ITEM
CREATE TABLE BONUSES(
customer_id int primary key,
bonus_amount float,
date date
)
```

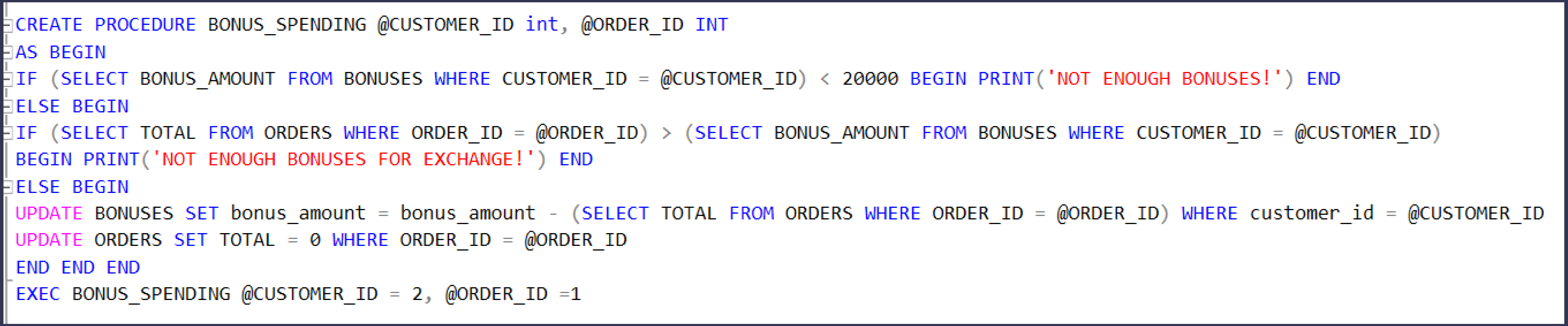


```
SELECT * FROM BONUSES
TRUNCATE TABLE ORDER_ITEM
```

customer_id	bonus_amount	date
2	11730	2022-08-01
3	5410	2023-02-21
6	3190	2023-04-30
9	8960	2023-05-06

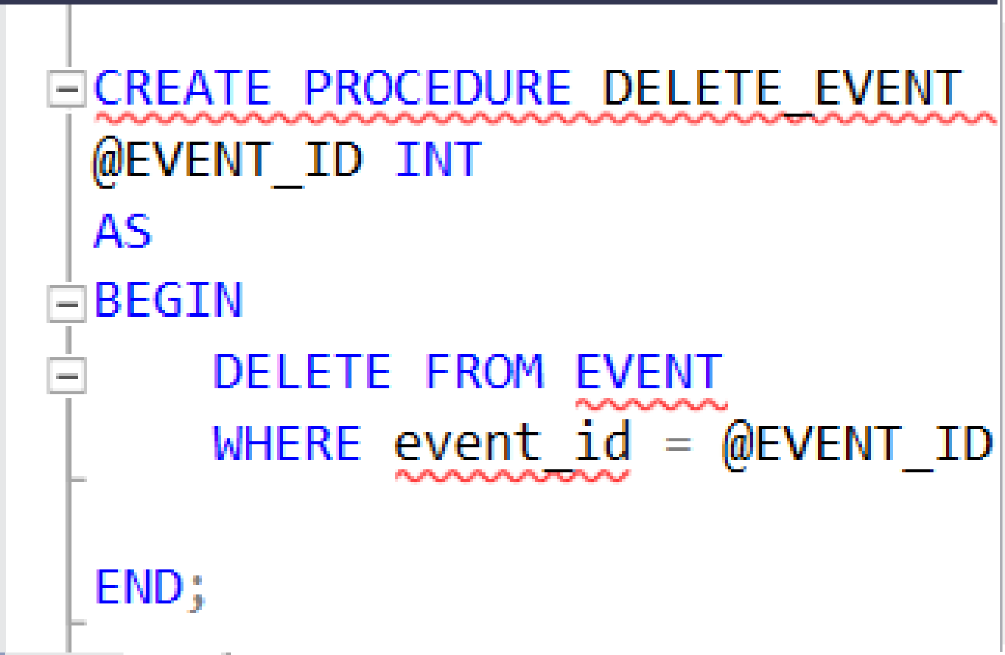


```
CREATE OR ALTER TRIGGER BONUS ON ORDERS AFTER INSERT,UPDATE AS BEGIN
DECLARE @TOTAL INT,@CUSTOMER_ID INT
SET @TOTAL = (SELECT TOTAL FROM INSERTED)
SET @CUSTOMER_ID = (SELECT CUSTOMER_ID FROM INSERTED)
IF DATEDIFF(DAY,GETDATE(),(SELECT DATE FROM BONUSES WHERE CUSTOMER_ID = @CUSTOMER_ID)) > 364 BEGIN
DELETE FROM BONUSES WHERE CUSTOMER_ID = @CUSTOMER_ID END
ELSE BEGIN
IF @CUSTOMER_ID IN (SELECT CUSTOMER_ID FROM BONUSES) BEGIN
UPDATE BONUSES SET bonus_amount = bonus_amount + @TOTAL *0.05 WHERE CUSTOMER_ID = @CUSTOMER_ID
END END END
```

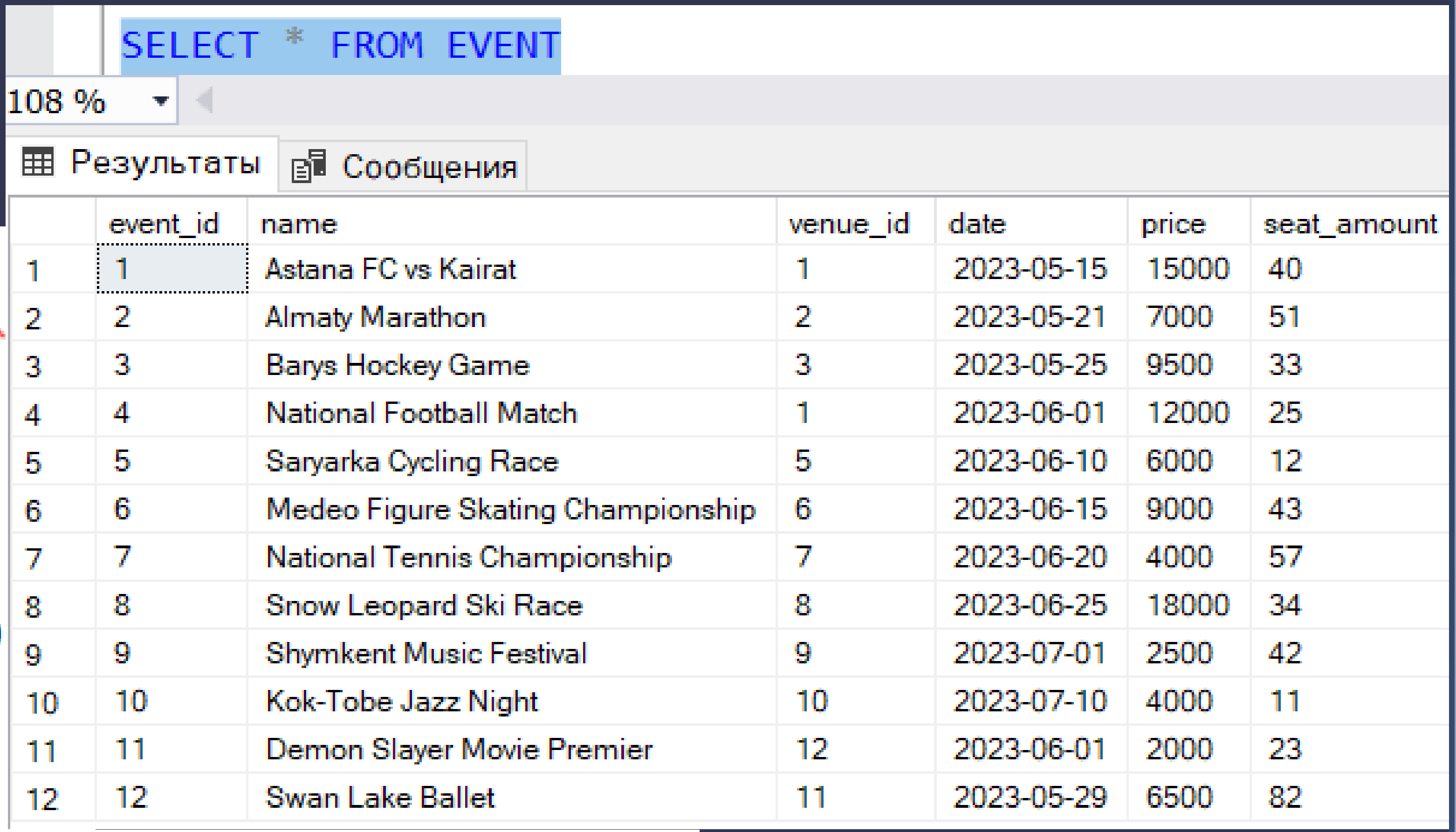


```
CREATE PROCEDURE BONUS_SPENDING @CUSTOMER_ID int, @ORDER_ID INT
AS BEGIN
IF (SELECT BONUS_AMOUNT FROM BONUSES WHERE CUSTOMER_ID = @CUSTOMER_ID) < 20000 BEGIN PRINT('NOT ENOUGH BONUSES!') END
ELSE BEGIN
IF (SELECT TOTAL FROM ORDERS WHERE ORDER_ID = @ORDER_ID) > (SELECT BONUS_AMOUNT FROM BONUSES WHERE CUSTOMER_ID = @CUSTOMER_ID)
BEGIN PRINT('NOT ENOUGH BONUSES FOR EXCHANGE!') END
ELSE BEGIN
UPDATE BONUSES SET bonus_amount = bonus_amount - (SELECT TOTAL FROM ORDERS WHERE ORDER_ID = @ORDER_ID) WHERE customer_id = @CUSTOMER_ID
UPDATE ORDERS SET TOTAL = 0 WHERE ORDER_ID = @ORDER_ID
END END END
EXEC BONUS_SPENDING @CUSTOMER_ID = 2, @ORDER_ID =1
```

3

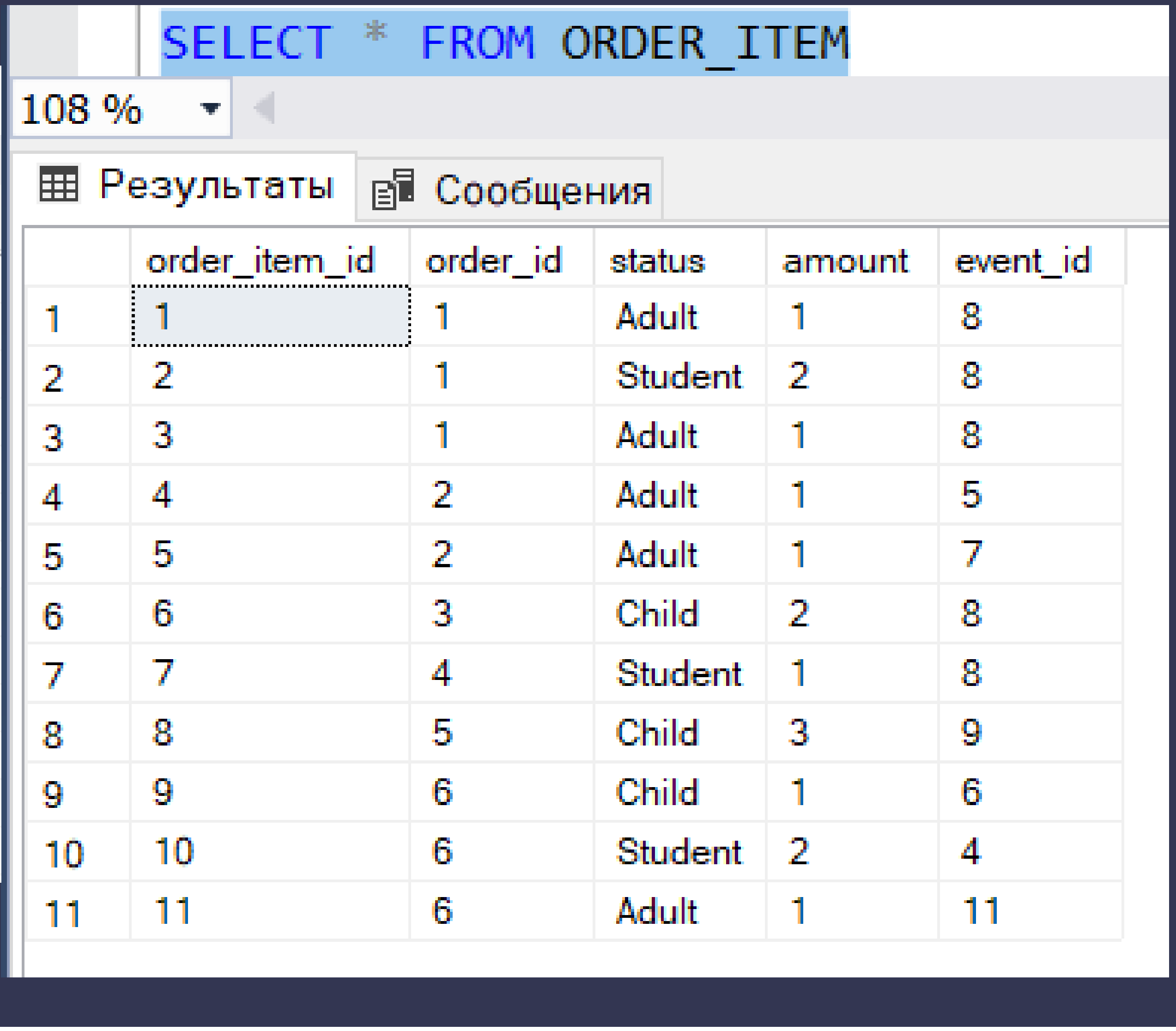


```
CREATE PROCEDURE DELETE_EVENT
@EVENT_ID INT
AS
BEGIN
    DELETE FROM EVENT
    WHERE event_id = @EVENT_ID

END;
```



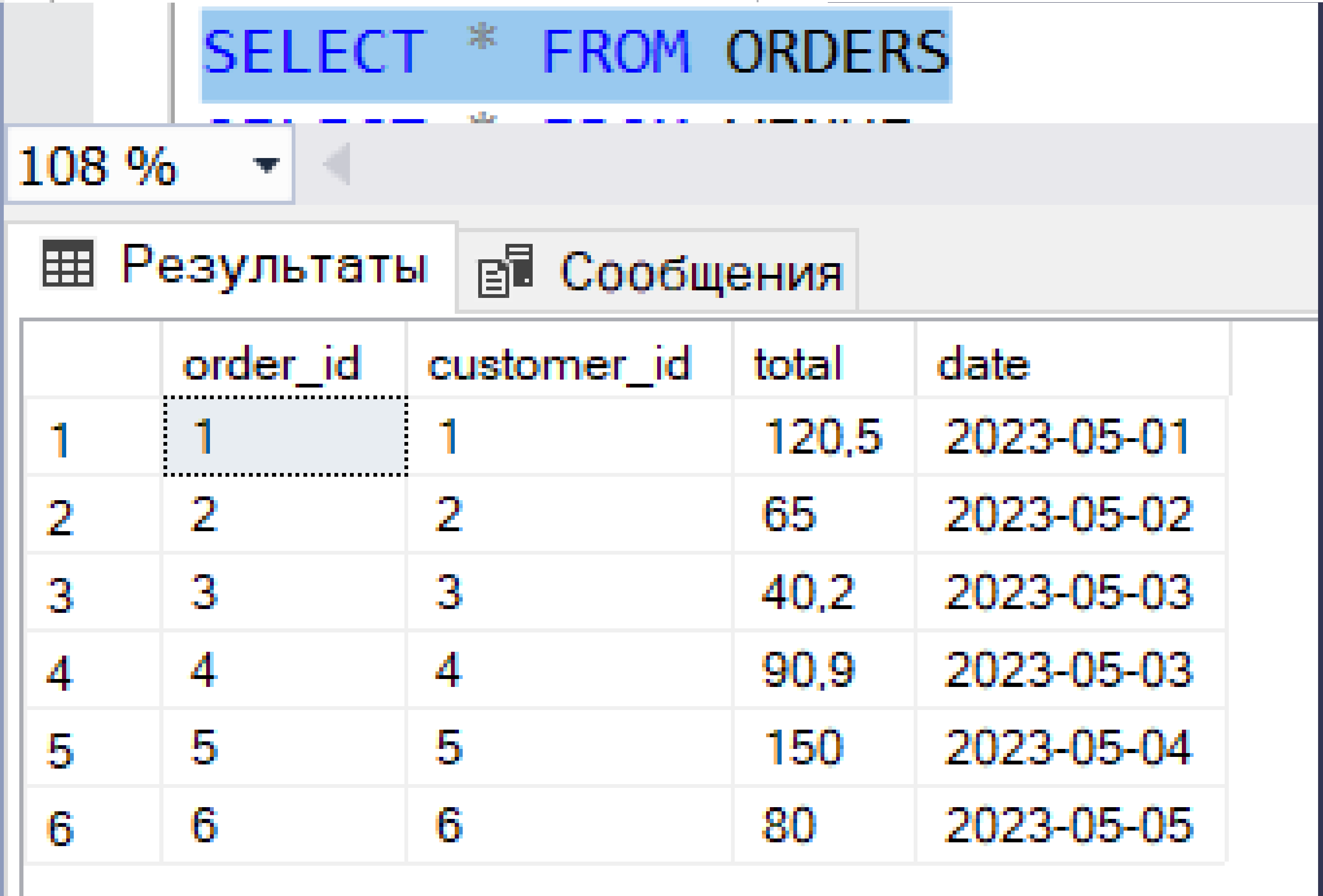
```
SELECT * FROM EVENT
```

	event_id	name	venue_id	date	price	seat_amount
1	1	Astana FC vs Kairat	1	2023-05-15	15000	40
2	2	Almaty Marathon	2	2023-05-21	7000	51
3	3	Barys Hockey Game	3	2023-05-25	9500	33
4	4	National Football Match	1	2023-06-01	12000	25
5	5	Saryarka Cycling Race	5	2023-06-10	6000	12
6	6	Medeo Figure Skating Championship	6	2023-06-15	9000	43
7	7	National Tennis Championship	7	2023-06-20	4000	57
8	8	Snow Leopard Ski Race	8	2023-06-25	18000	34
9	9	Shymkent Music Festival	9	2023-07-01	2500	42
10	10	Kok-Tobe Jazz Night	10	2023-07-10	4000	11
11	11	Demon Slayer Movie Premier	12	2023-06-01	2000	23
12	12	Swan Lake Ballet	11	2023-05-29	6500	82



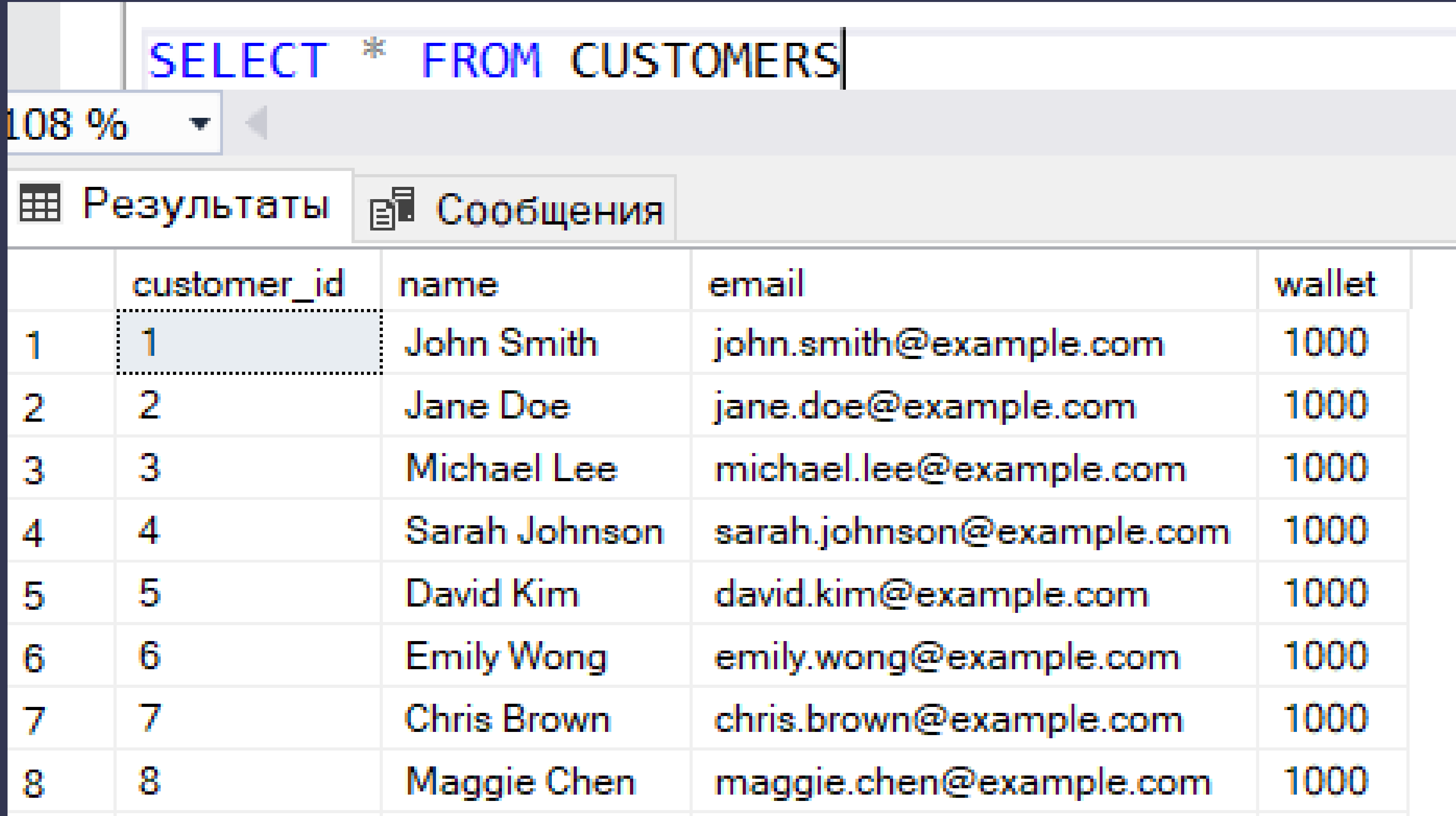
```
SELECT * FROM ORDER_ITEM
```

	order_item_id	order_id	status	amount	event_id
1	1	1	Adult	1	8
2	2	1	Student	2	8
3	3	1	Adult	1	8
4	4	2	Adult	1	5
5	5	2	Adult	1	7
6	6	3	Child	2	8
7	7	4	Student	1	8
8	8	5	Child	3	9
9	9	6	Child	1	6
10	10	6	Student	2	4
11	11	6	Adult	1	11



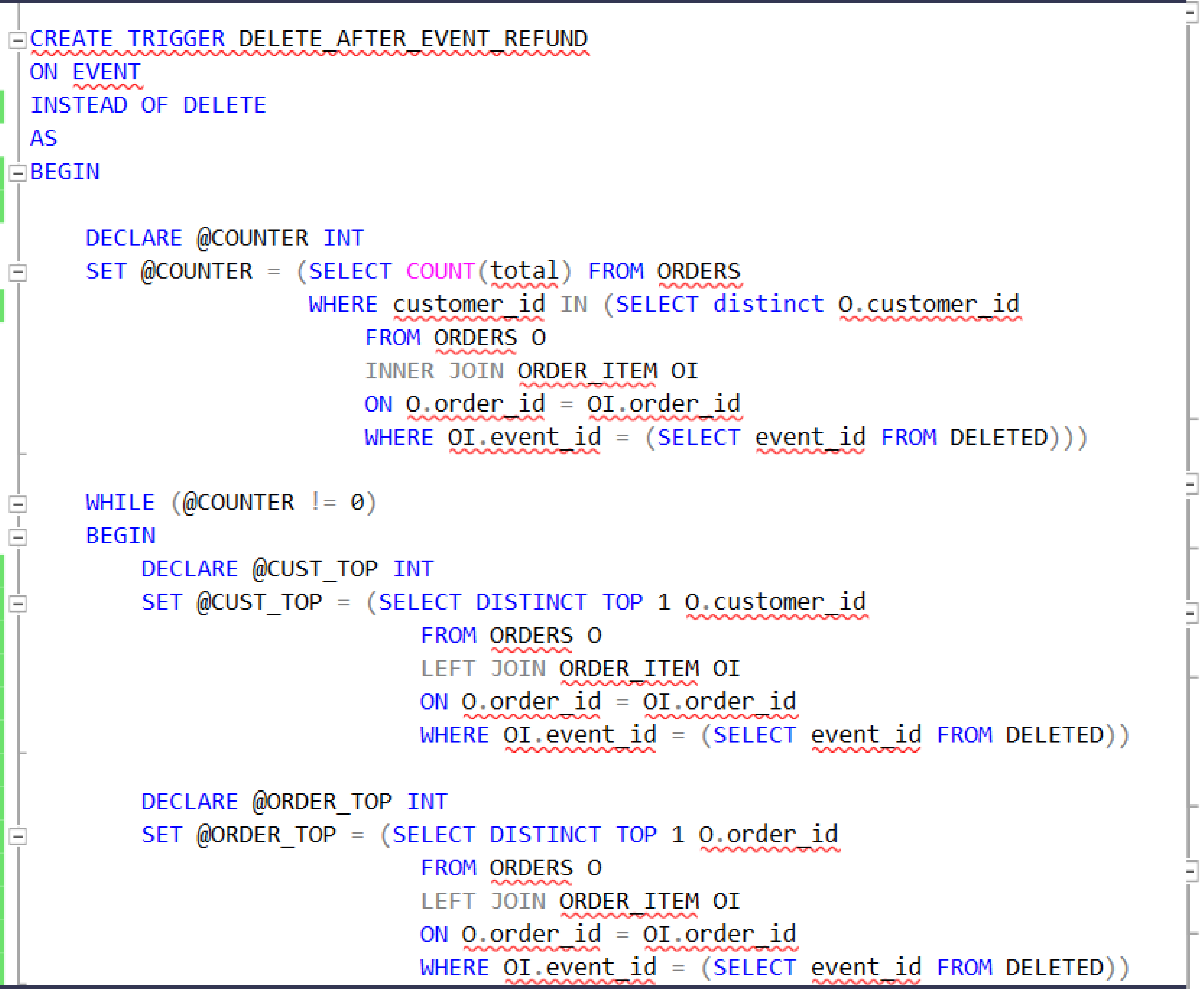
```
SELECT * FROM ORDERS
```

	order_id	customer_id	total	date
1	1	1	120,5	2023-05-01
2	2	2	65	2023-05-02
3	3	3	40,2	2023-05-03
4	4	4	90,9	2023-05-03
5	5	5	150	2023-05-04
6	6	6	80	2023-05-05

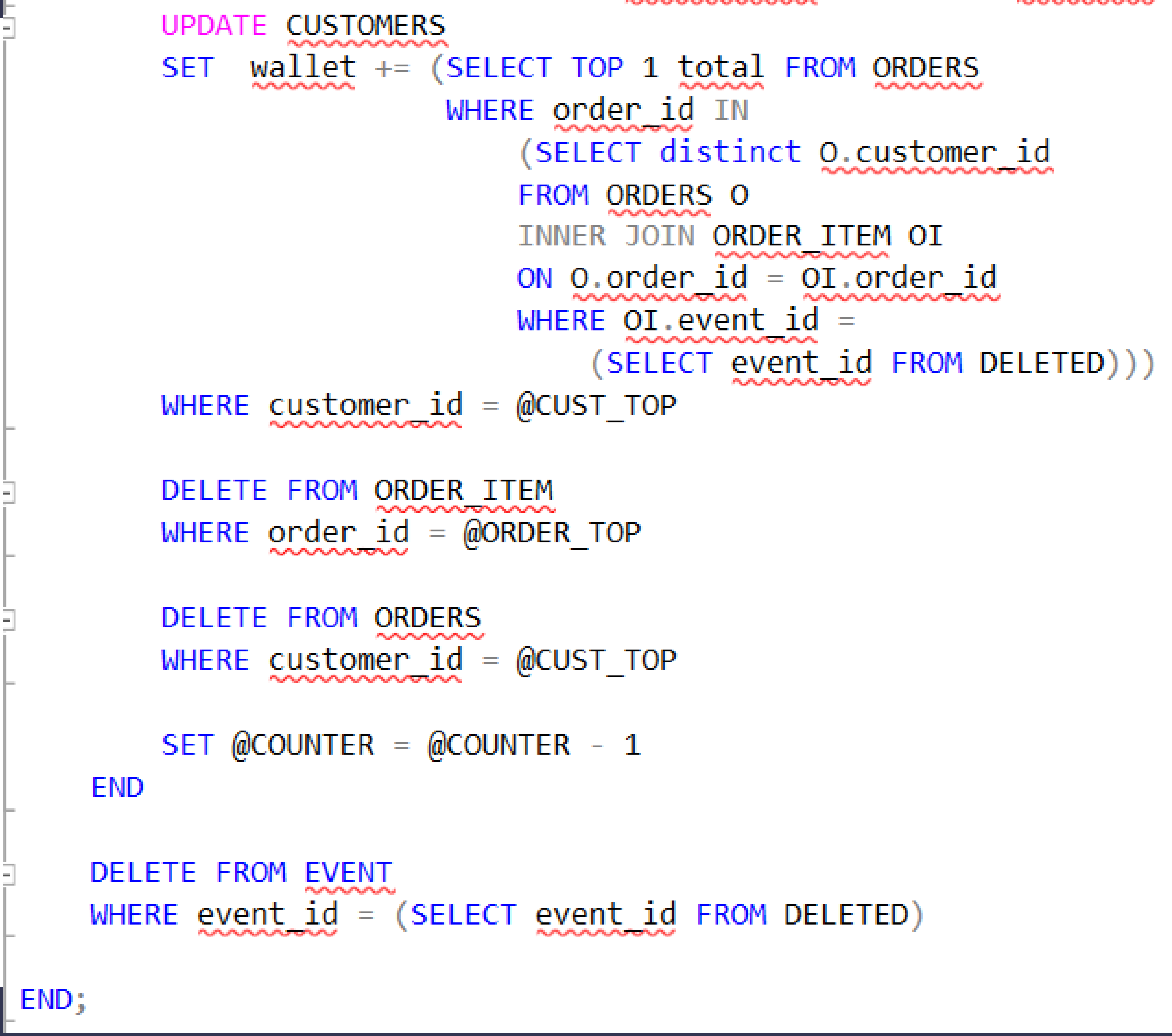


```
SELECT * FROM CUSTOMERS
```

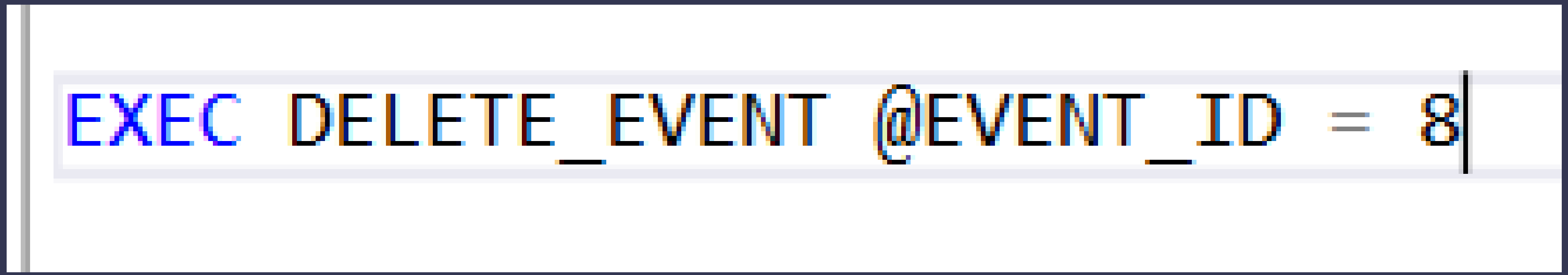
	customer_id	name	email	wallet
1	1	John Smith	john.smith@example.com	1000
2	2	Jane Doe	jane.doe@example.com	1000
3	3	Michael Lee	michael.lee@example.com	1000
4	4	Sarah Johnson	sarah.johnson@example.com	1000
5	5	David Kim	david.kim@example.com	1000
6	6	Emily Wong	emily.wong@example.com	1000
7	7	Chris Brown	chris.brown@example.com	1000
8	8	Maggie Chen	maggie.chen@example.com	1000



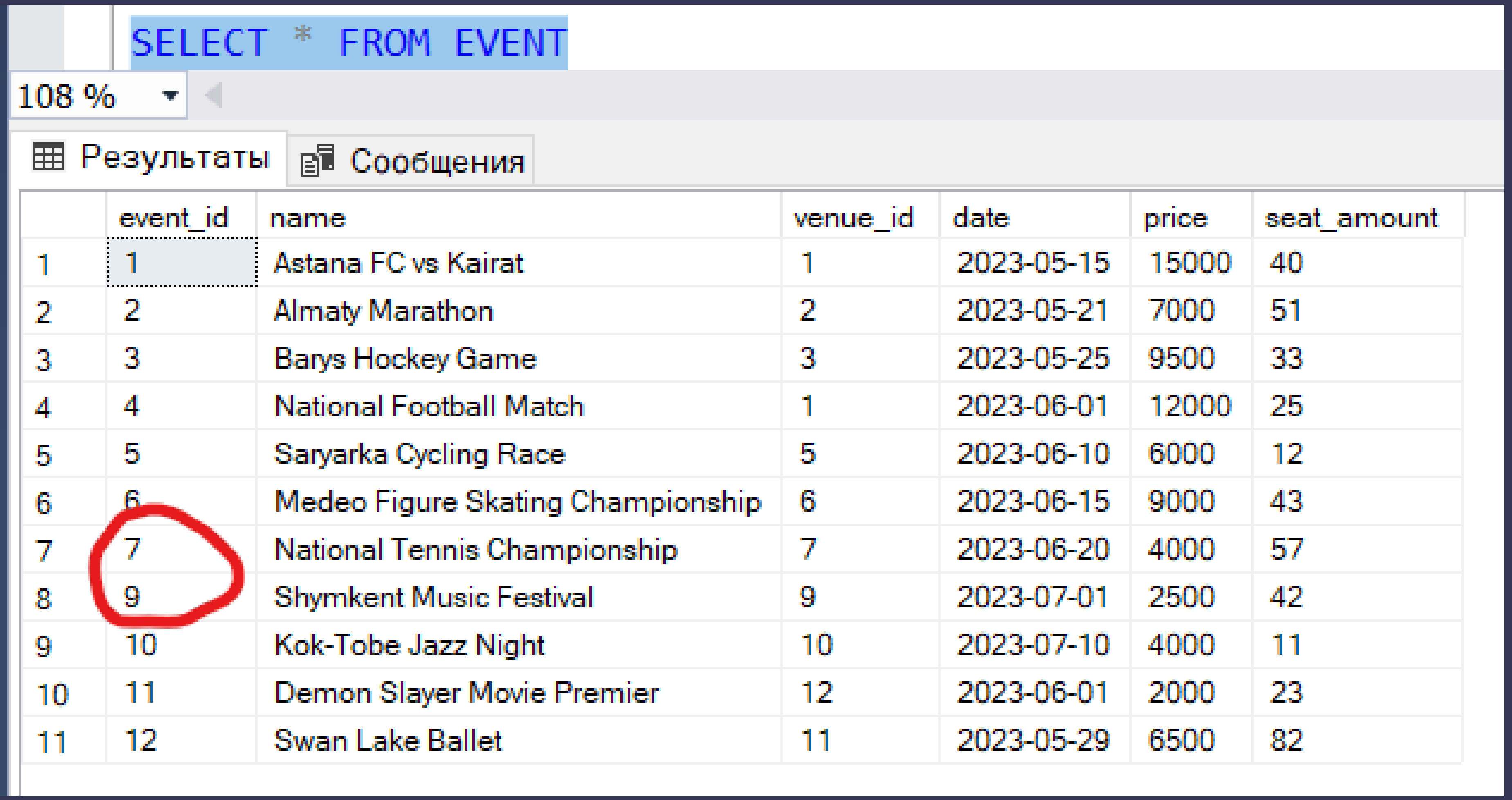
```
CREATE TRIGGER DELETE_AFTER_EVENT_REFUND
ON EVENT
INSTEAD OF DELETE
AS
BEGIN

    DECLARE @COUNTER INT
    SET @COUNTER = (SELECT COUNT(total) FROM ORDERS
                    WHERE customer_id IN (SELECT distinct O.customer_id
                        FROM ORDERS O
                        INNER JOIN ORDER_ITEM OI
                        ON O.order_id = OI.order_id
                        WHERE OI.event_id = (SELECT event_id FROM DELETED)))

    WHILE (@COUNTER != 0)
    BEGIN
        DECLARE @CUST_TOP INT
        SET @CUST_TOP = (SELECT DISTINCT TOP 1 O.customer_id
                            FROM ORDERS O
                            LEFT JOIN ORDER_ITEM OI
                            ON O.order_id = OI.order_id
                            WHERE OI.event_id = (SELECT event_id FROM DELETED))

        DECLARE @ORDER_TOP INT
        SET @ORDER_TOP = (SELECT DISTINCT TOP 1 O.order_id
                            FROM ORDERS O
                            LEFT JOIN ORDER_ITEM OI
                            ON O.order_id = OI.order_id
                            WHERE OI.event_id = (SELECT event_id FROM DELETED))
```



```
        UPDATE CUSTOMERS
        SET  wallet += (SELECT TOP 1 total FROM ORDERS
                        WHERE order_id IN
                            (SELECT distinct O.customer_id
                            FROM ORDERS O
                            INNER JOIN ORDER_ITEM OI
                            ON O.order_id = OI.order_id
                            WHERE OI.event_id =
                                (SELECT event_id FROM DELETED)))
        WHERE customer_id = @CUST_TOP

        DELETE FROM ORDER_ITEM
        WHERE order_id = @ORDER_TOP

        DELETE FROM ORDERS
        WHERE customer_id = @CUST_TOP

        SET @COUNTER = @COUNTER - 1
    END

    DELETE FROM EVENT
    WHERE event_id = (SELECT event_id FROM DELETED)

END;
```

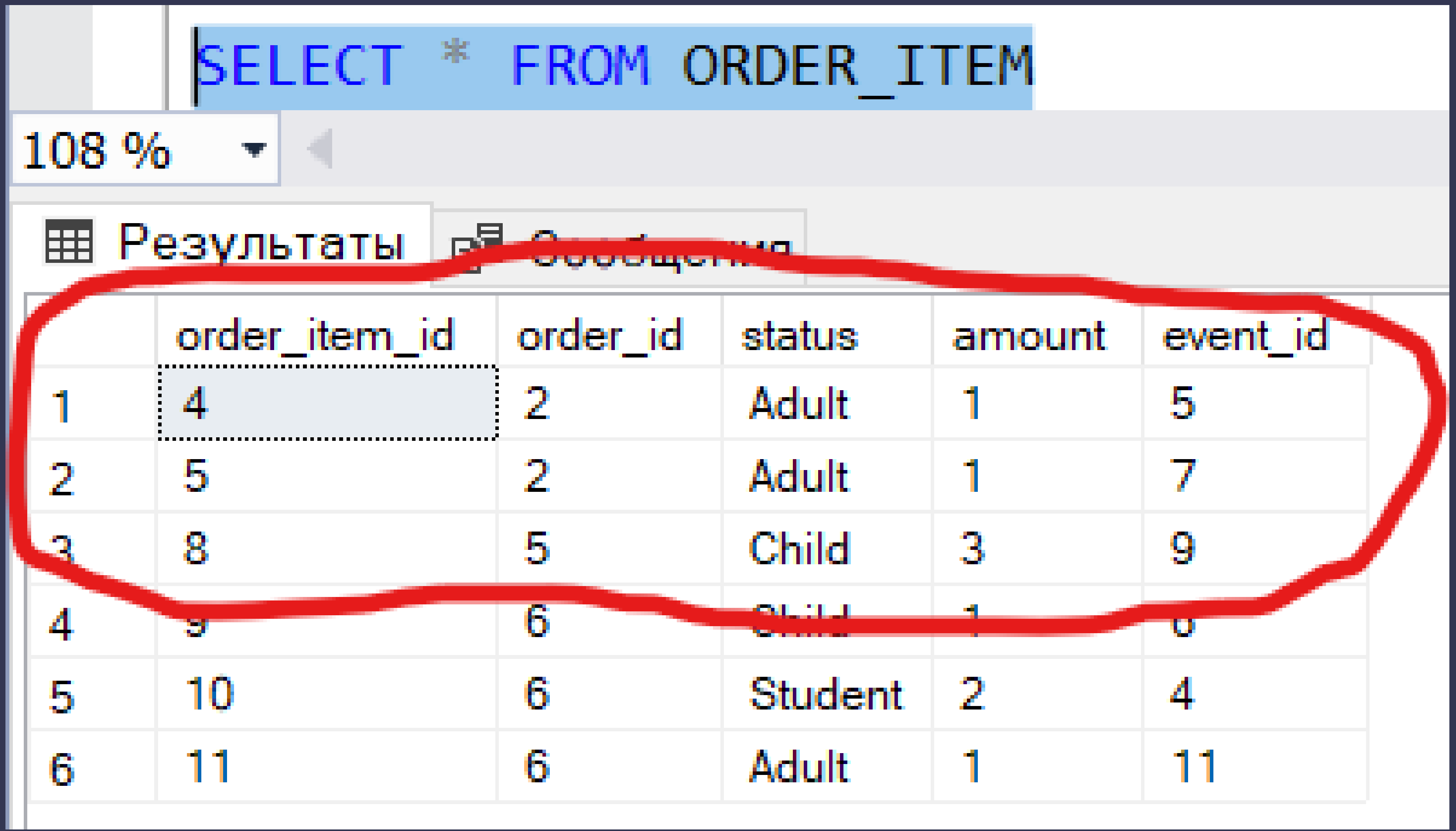


```
EXEC DELETE_EVENT @EVENT_ID = 8
```



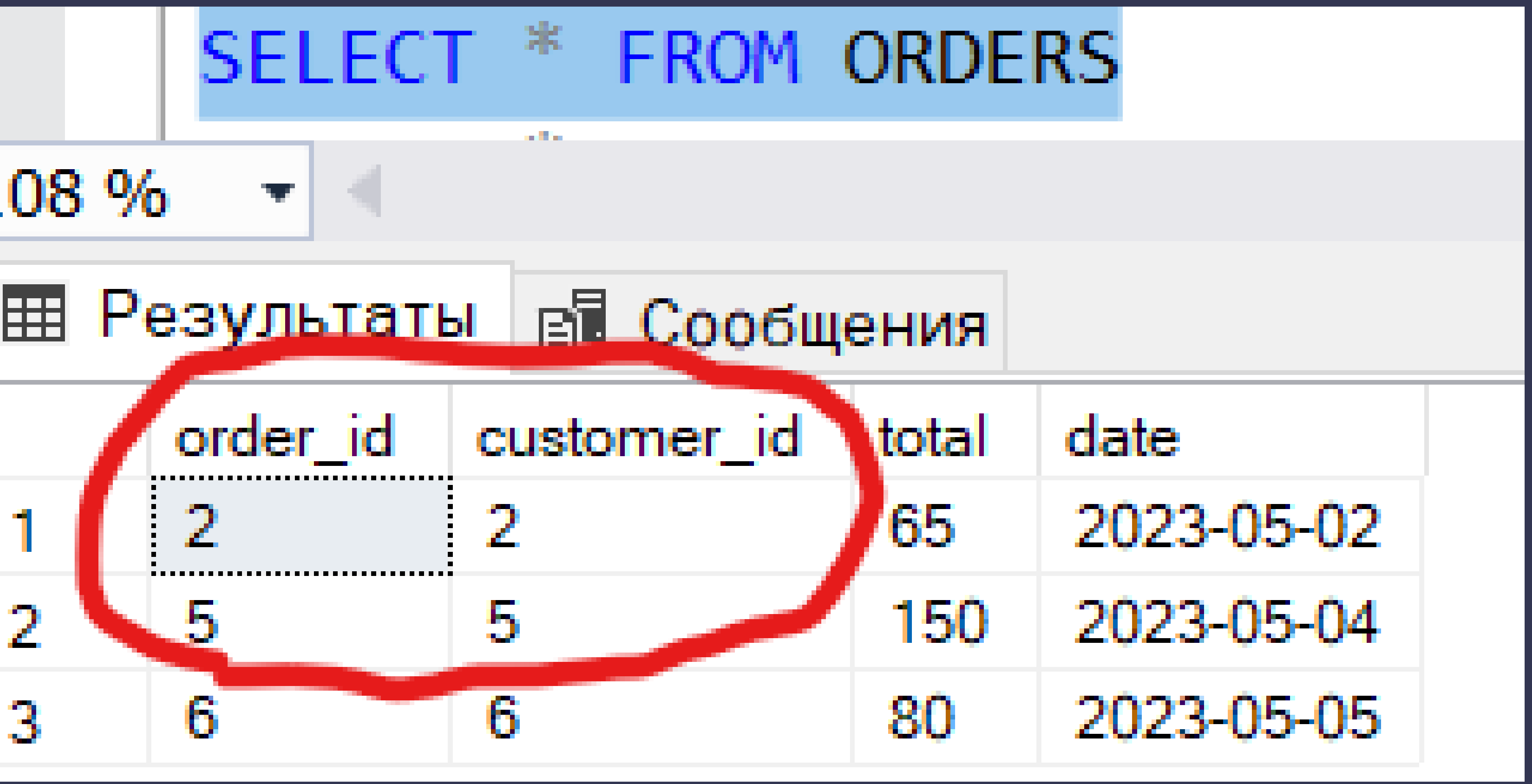
```
SELECT * FROM EVENT
```

	event_id	name	venue_id	date	price	seat_amount
1	1	Astana FC vs Kairat	1	2023-05-15	15000	40
2	2	Almaty Marathon	2	2023-05-21	7000	51
3	3	Barys Hockey Game	3	2023-05-25	9500	33
4	4	National Football Match	1	2023-06-01	12000	25
5	5	Saryarka Cycling Race	5	2023-06-10	6000	12
6	6	Medeo Figure Skating Championship	6	2023-06-15	9000	43
7	7	National Tennis Championship	7	2023-06-20	4000	57
8	9	Shymkent Music Festival	9	2023-07-01	2500	42
9	10	Kok-Tobe Jazz Night	10	2023-07-10	4000	11
10	11	Demon Slayer Movie Premier	12	2023-06-01	2000	23
11	12	Swan Lake Ballet	11	2023-05-29	6500	82



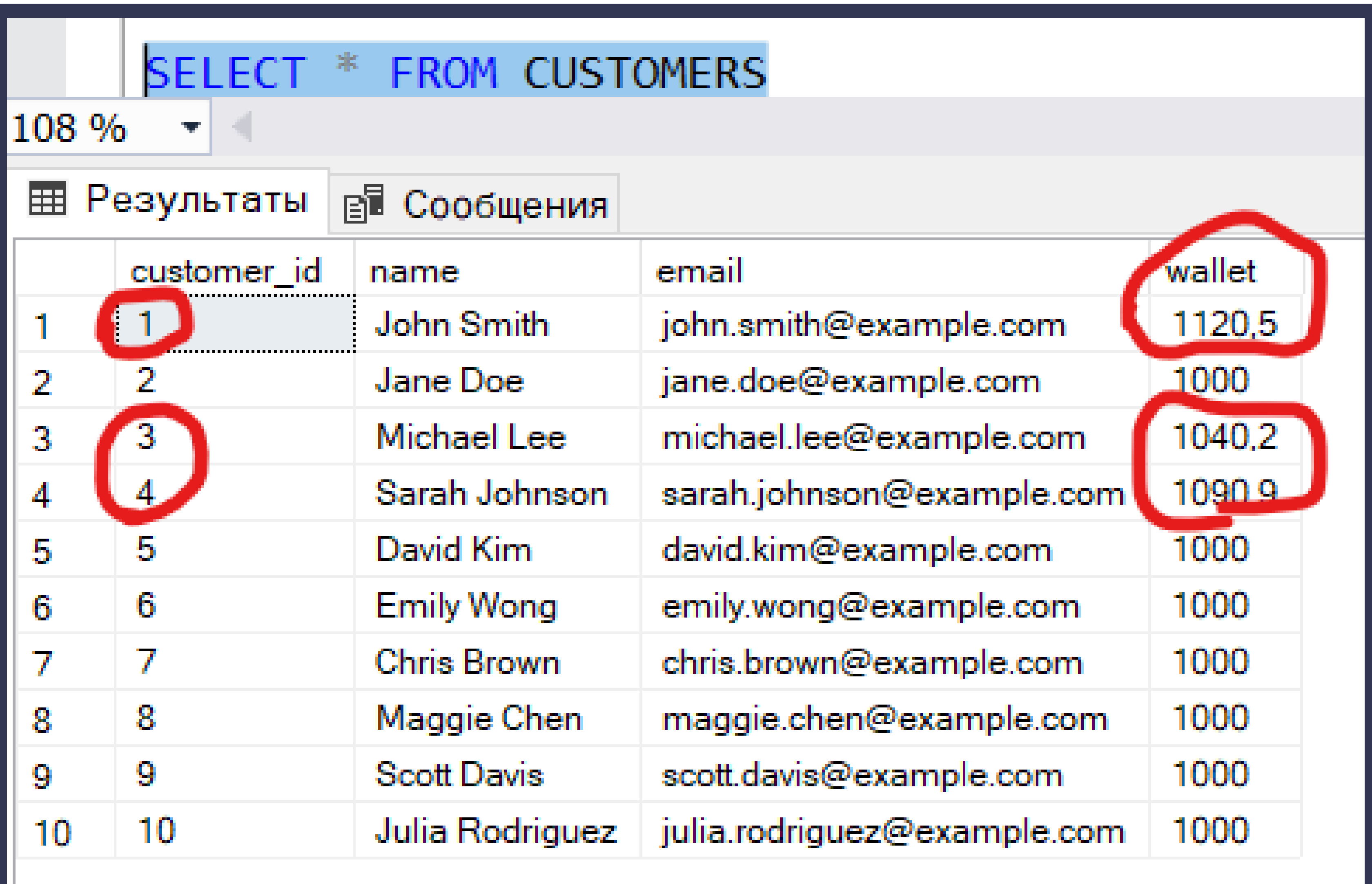
```
SELECT * FROM ORDER_ITEM
```

	order_item_id	order_id	status	amount	event_id
1	4	2	Adult	1	5
2	5	2	Adult	1	7
3	8	5	Child	3	9
4	9	6	Child	1	6
5	10	6	Student	2	4
6	11	6	Adult	1	11



```
SELECT * FROM ORDERS
```

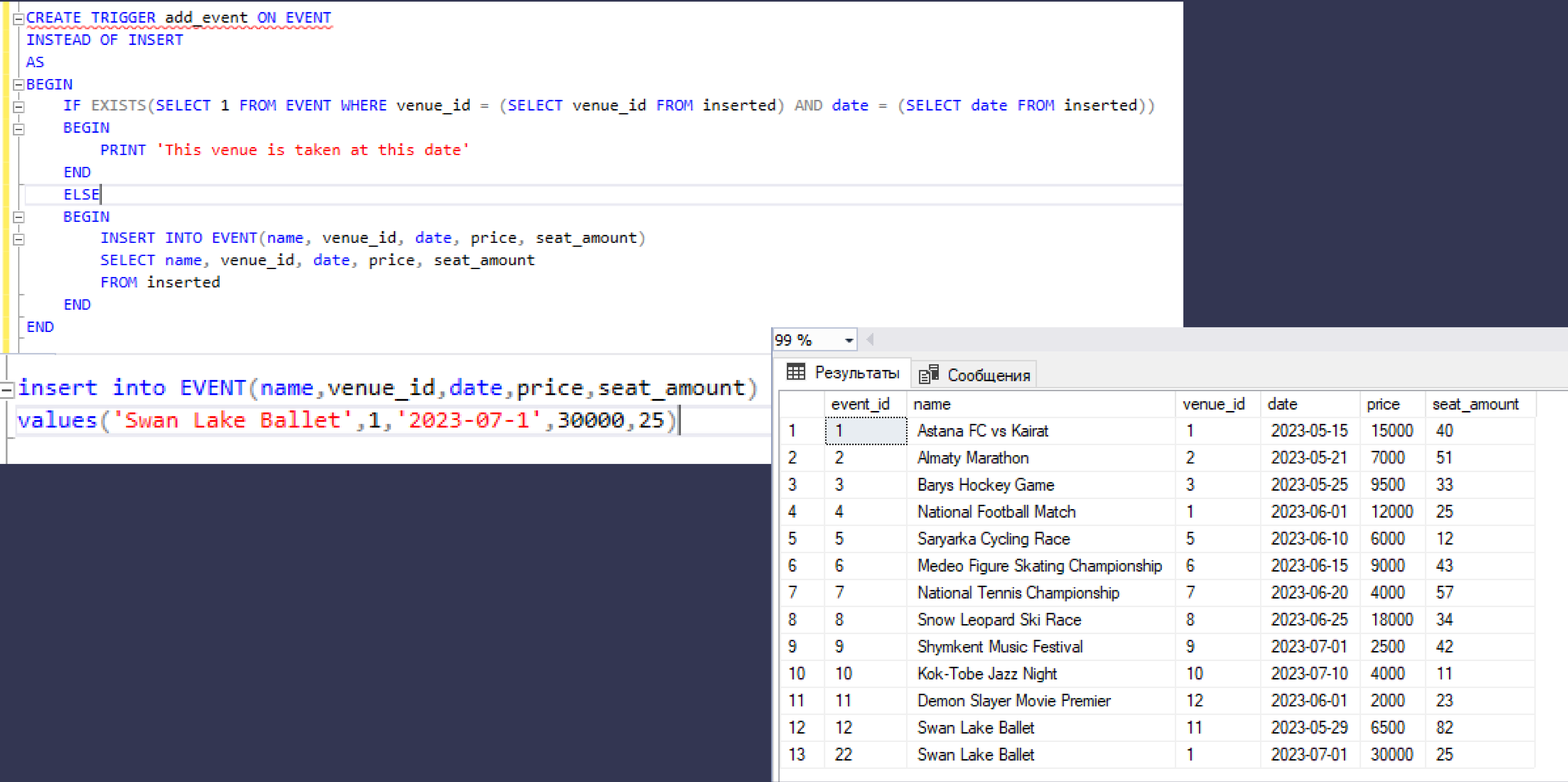
	order_id	customer_id	total	date
1	2	2	65	2023-05-02
2	5	5	150	2023-05-04
3	6	6	80	2023-05-05



```
SELECT * FROM CUSTOMERS
```

	customer_id	name	email	wallet
1	1	John Smith	john.smith@example.com	1120,5
2	2	Jane Doe	jane.doe@example.com	1000
3	3	Michael Lee	michael.lee@example.com	1040,2
4	4	Sarah Johnson	sarah.johnson@example.com	1090,9
5	5	David Kim	david.kim@example.com	1000
6	6	Emily Wong	emily.wong@example.com	1000
7	7	Chris Brown	chris.brown@example.com	1000
8	8	Maggie Chen	maggie.chen@example.com	1000
9	9	Scott Davis	scott.davis@example.com	1000
10	10	Julia Rodriguez	julia.rodriguez@example.com	1000

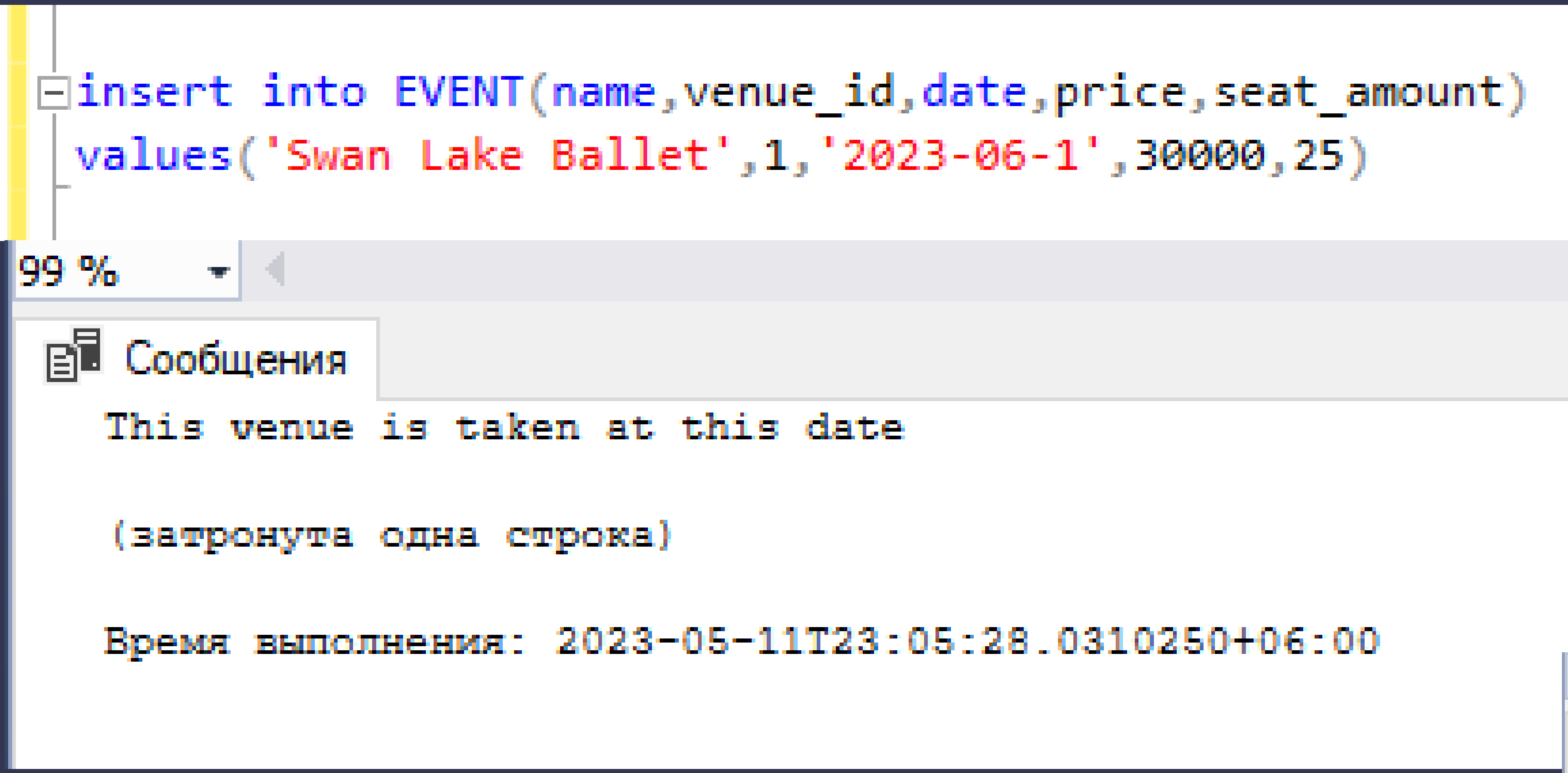
4



```
CREATE TRIGGER add_event ON EVENT
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS(SELECT 1 FROM EVENT WHERE venue_id = (SELECT venue_id FROM inserted) AND date = (SELECT date FROM inserted))
    BEGIN
        PRINT 'This venue is taken at this date'
    END
    ELSE
    BEGIN
        INSERT INTO EVENT(name, venue_id, date, price, seat_amount)
        SELECT name, venue_id, date, price, seat_amount
        FROM inserted
    END
END
```

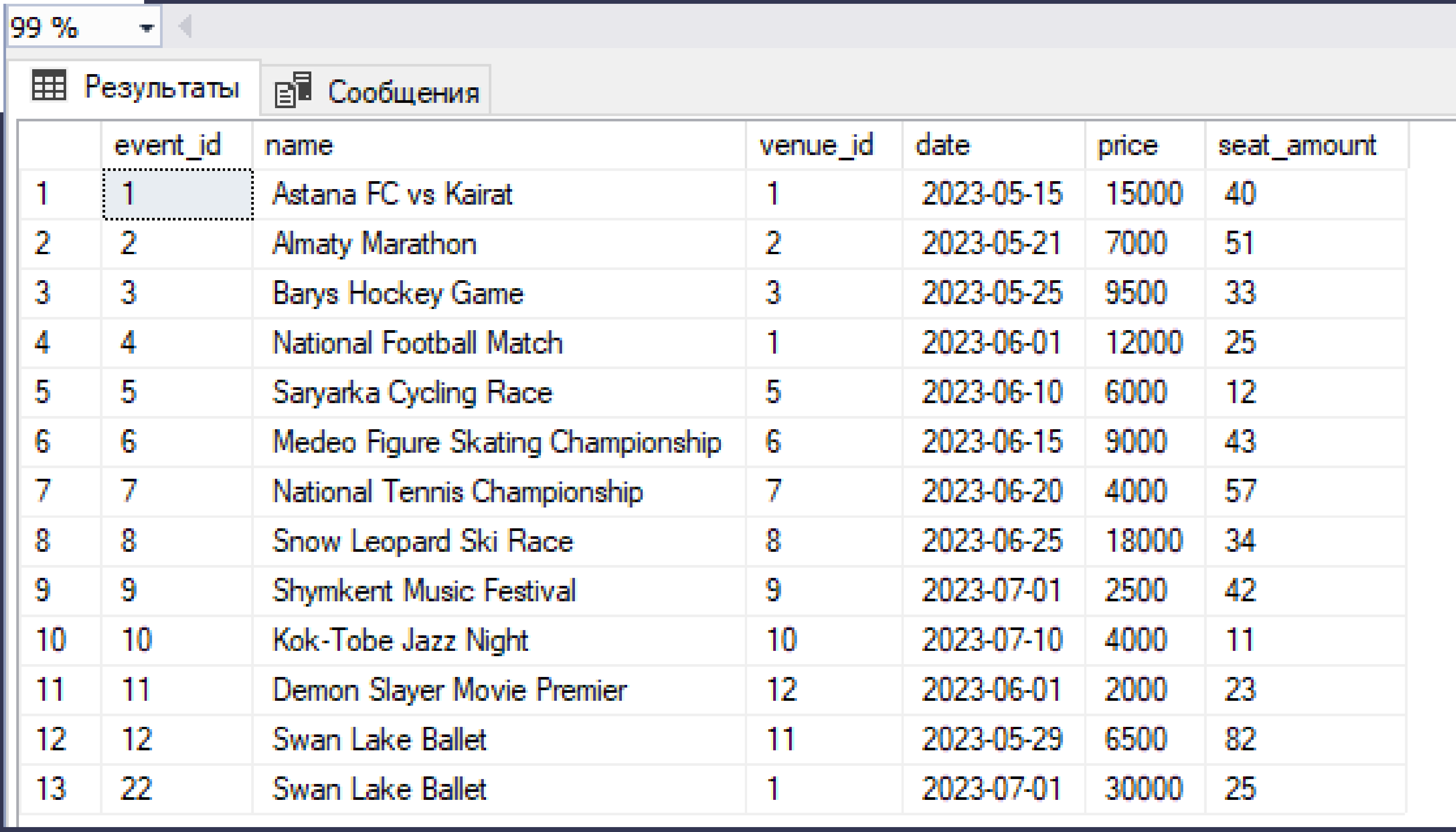
```
insert into EVENT(name,venue_id,date,price,seat_amount)
values('Swan Lake Ballet',1,'2023-07-1',30000,25)
```

	event_id	name	venue_id	date	price	seat_amount
1	1	Astana FC vs Kairat	1	2023-05-15	15000	40
2	2	Almaty Marathon	2	2023-05-21	7000	51
3	3	Barys Hockey Game	3	2023-05-25	9500	33
4	4	National Football Match	1	2023-06-01	12000	25
5	5	Saryarka Cycling Race	5	2023-06-10	6000	12
6	6	Medeo Figure Skating Championship	6	2023-06-15	9000	43
7	7	National Tennis Championship	7	2023-06-20	4000	57
8	8	Snow Leopard Ski Race	8	2023-06-25	18000	34
9	9	Shymkent Music Festival	9	2023-07-01	2500	42
10	10	Kok-Tobe Jazz Night	10	2023-07-10	4000	11
11	11	Demon Slayer Movie Premier	12	2023-06-01	2000	23
12	12	Swan Lake Ballet	11	2023-05-29	6500	82
13	22	Swan Lake Ballet	1	2023-07-01	30000	25



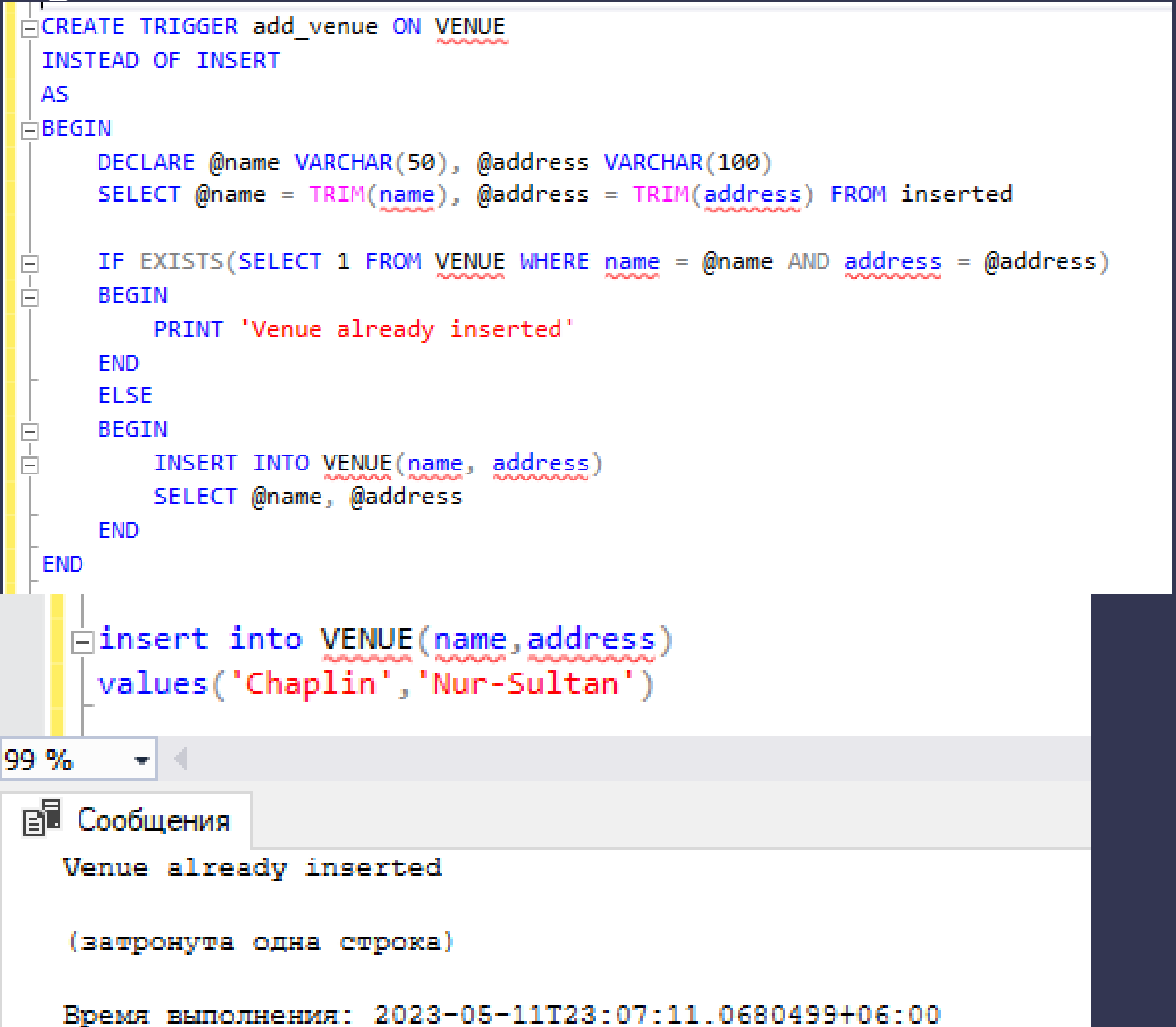
```sql
insert into EVENT(name,venue_id,date,price,seat_amount)
values('Swan Lake Ballet',1,'2023-06-1',30000,25)
```

```
This venue is taken at this date

(затронута одна строка)

Время выполнения: 2023-05-11T23:05:28.0310250+06:00
```



	event_id	name	venue_id	date	price	seat_amount
1	1	Astana FC vs Kairat	1	2023-05-15	15000	40
2	2	Almaty Marathon	2	2023-05-21	7000	51
3	3	Barys Hockey Game	3	2023-05-25	9500	33
4	4	National Football Match	1	2023-06-01	12000	25
5	5	Saryarka Cycling Race	5	2023-06-10	6000	12
6	6	Medeo Figure Skating Championship	6	2023-06-15	9000	43
7	7	National Tennis Championship	7	2023-06-20	4000	57
8	8	Snow Leopard Ski Race	8	2023-06-25	18000	34
9	9	Shymkent Music Festival	9	2023-07-01	2500	42
10	10	Kok-Tobe Jazz Night	10	2023-07-10	4000	11
11	11	Demon Slayer Movie Premier	12	2023-06-01	2000	23
12	12	Swan Lake Ballet	11	2023-05-29	6500	82
13	22	Swan Lake Ballet	1	2023-07-01	30000	25

5



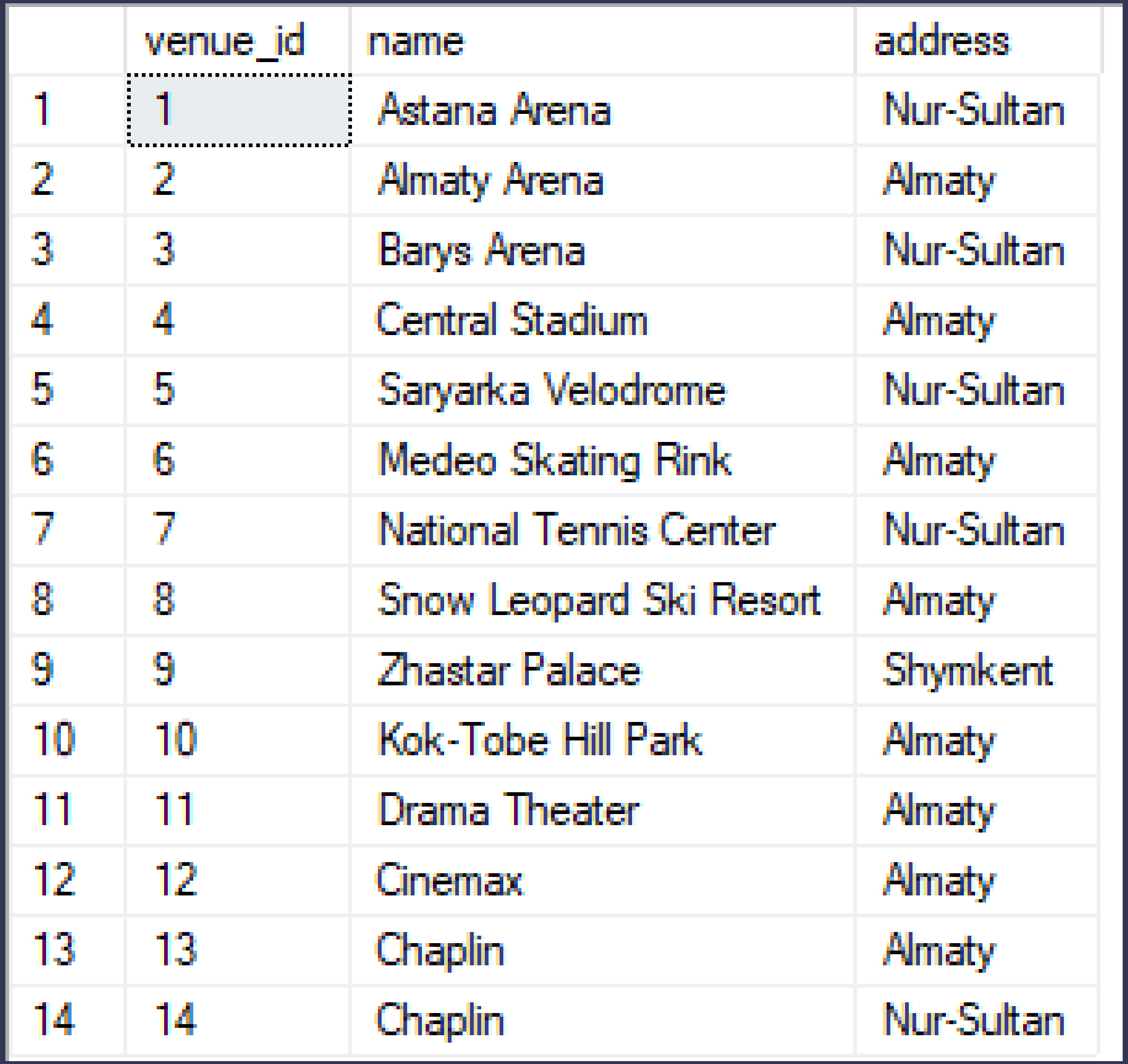
```
CREATE TRIGGER add_venue ON VENUE
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @name VARCHAR(50), @address VARCHAR(100)
    SELECT @name = TRIM(name), @address = TRIM(address) FROM inserted

    IF EXISTS(SELECT 1 FROM VENUE WHERE name = @name AND address = @address)
    BEGIN
        PRINT 'Venue already inserted'
    END
    ELSE
    BEGIN
        INSERT INTO VENUE(name, address)
        SELECT @name, @address
    END
END
```

```
insert into VENUE(name,address)
values('Chaplin','Nur-Sultan')
```

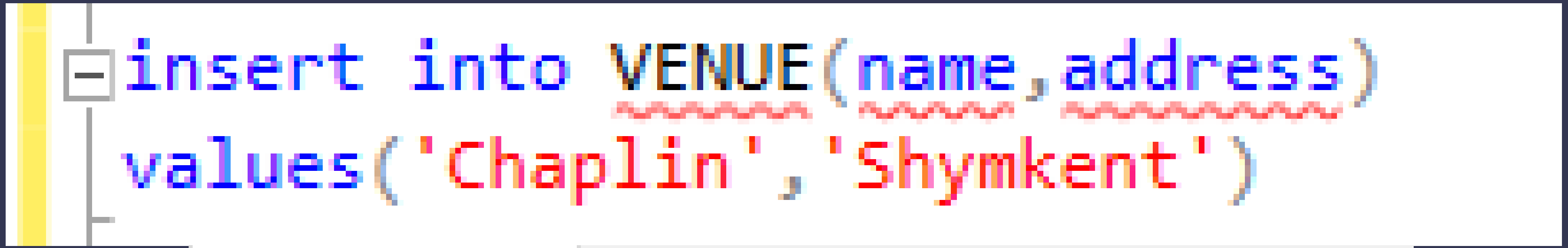
99 %

Сообщения

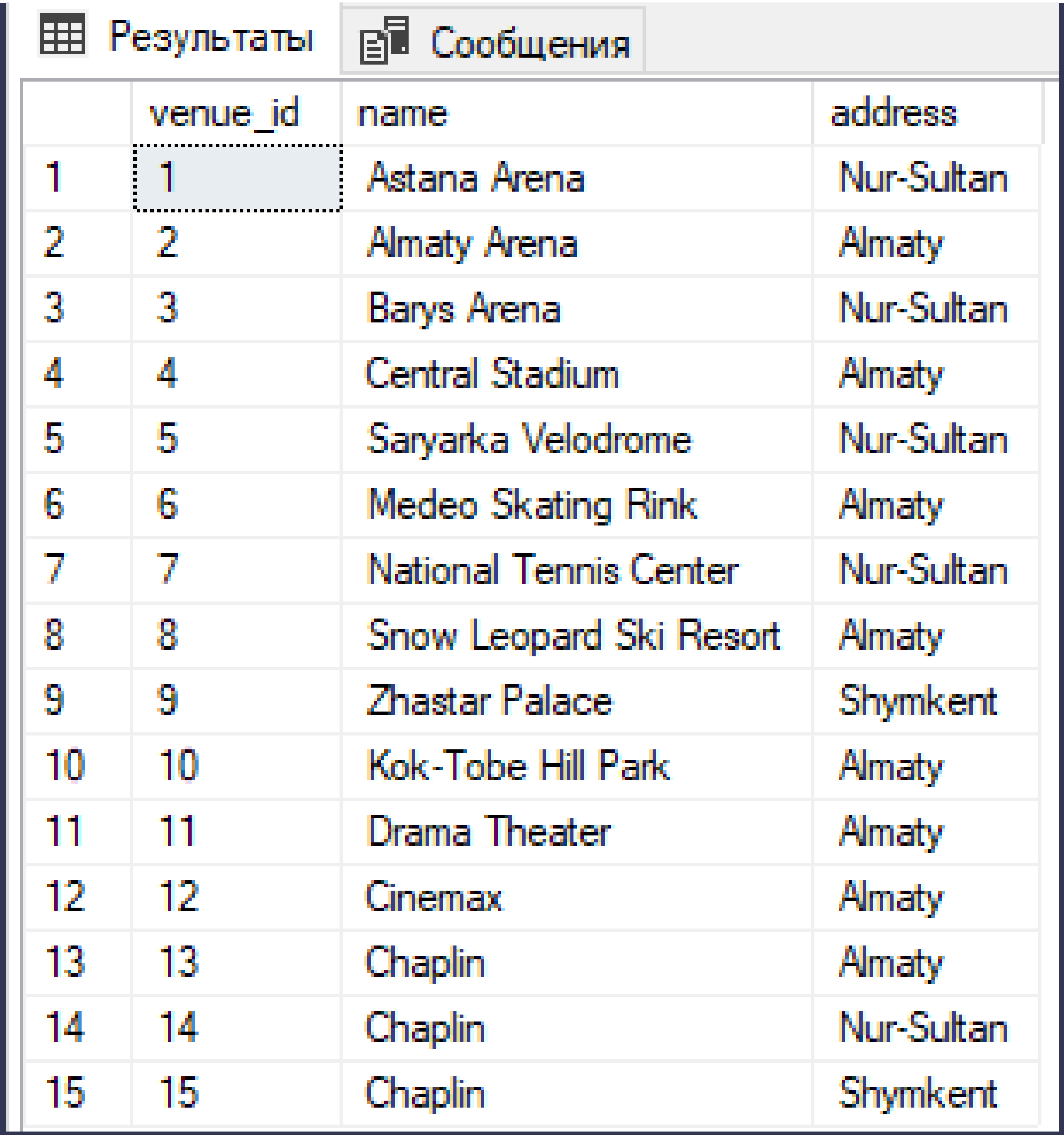
```
Venue already inserted

(затронута одна строка)

Время выполнения: 2023-05-11T23:07:11.0680499+06:00
```



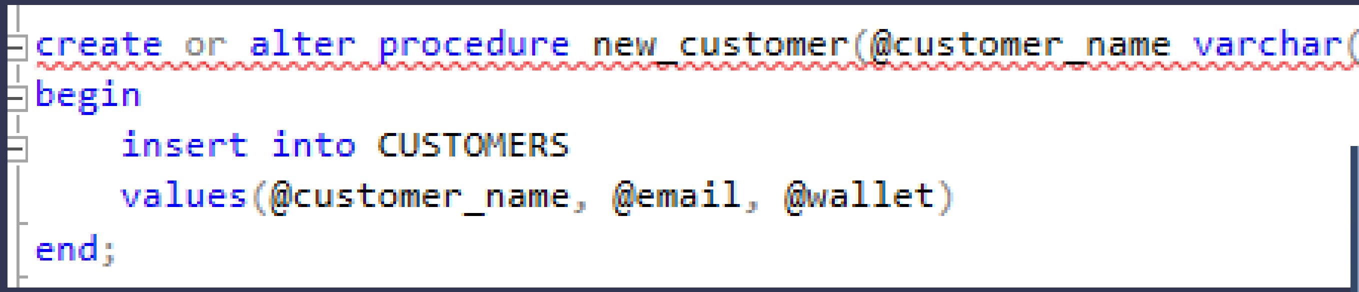
	venue_id	name	address
1	1	Astana Arena	Nur-Sultan
2	2	Almaty Arena	Almaty
3	3	Barys Arena	Nur-Sultan
4	4	Central Stadium	Almaty
5	5	Saryarka Velodrome	Nur-Sultan
6	6	Medeo Skating Rink	Almaty
7	7	National Tennis Center	Nur-Sultan
8	8	Snow Leopard Ski Resort	Almaty
9	9	Zhastar Palace	Shymkent
10	10	Kok-Tobe Hill Park	Almaty
11	11	Drama Theater	Almaty
12	12	Cinemax	Almaty
13	13	Chaplin	Almaty
14	14	Chaplin	Nur-Sultan



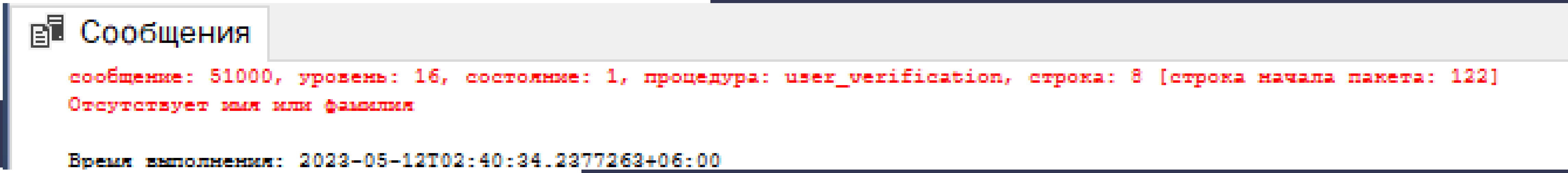
```sql
insert into VENUE(name,address)
values('Chaplin','Shymkent')
```



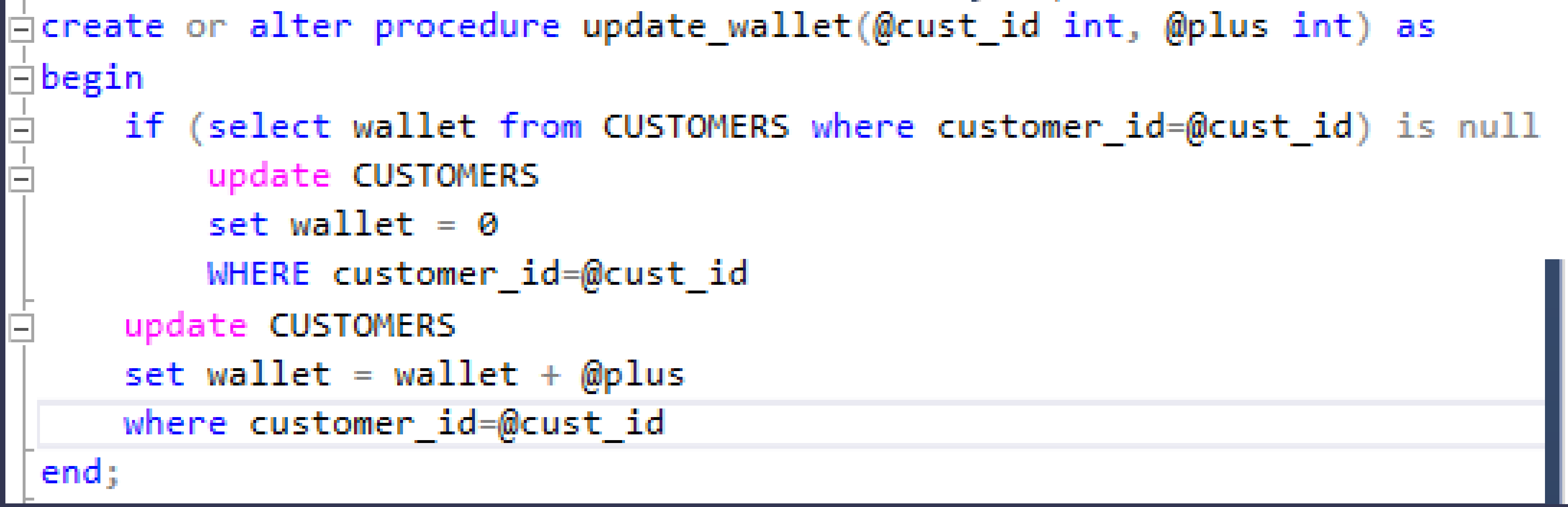
	venue_id	name	address
1	1	Astana Arena	Nur-Sultan
2	2	Almaty Arena	Almaty
3	3	Barys Arena	Nur-Sultan
4	4	Central Stadium	Almaty
5	5	Saryarka Velodrome	Nur-Sultan
6	6	Medeo Skating Rink	Almaty
7	7	National Tennis Center	Nur-Sultan
8	8	Snow Leopard Ski Resort	Almaty
9	9	Zhastar Palace	Shymkent
10	10	Kok-Tobe Hill Park	Almaty
11	11	Drama Theater	Almaty
12	12	Cinemax	Almaty
13	13	Chaplin	Almaty
14	14	Chaplin	Nur-Sultan
15	15	Chaplin	Shymkent



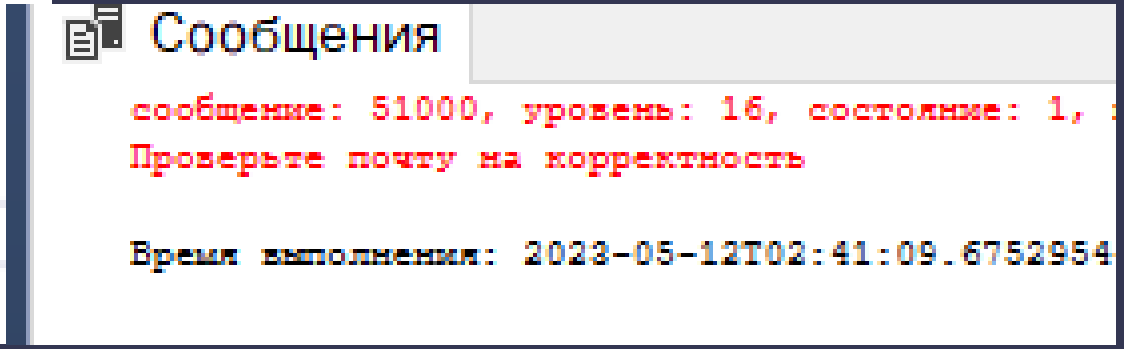
```sql
create or alter procedure new_customer(@customer_name varchar(30), @email varchar(40), @wallet int = null) as
begin
    insert into CUSTOMERS
    values(@customer_name, @email, @wallet)
end;
```



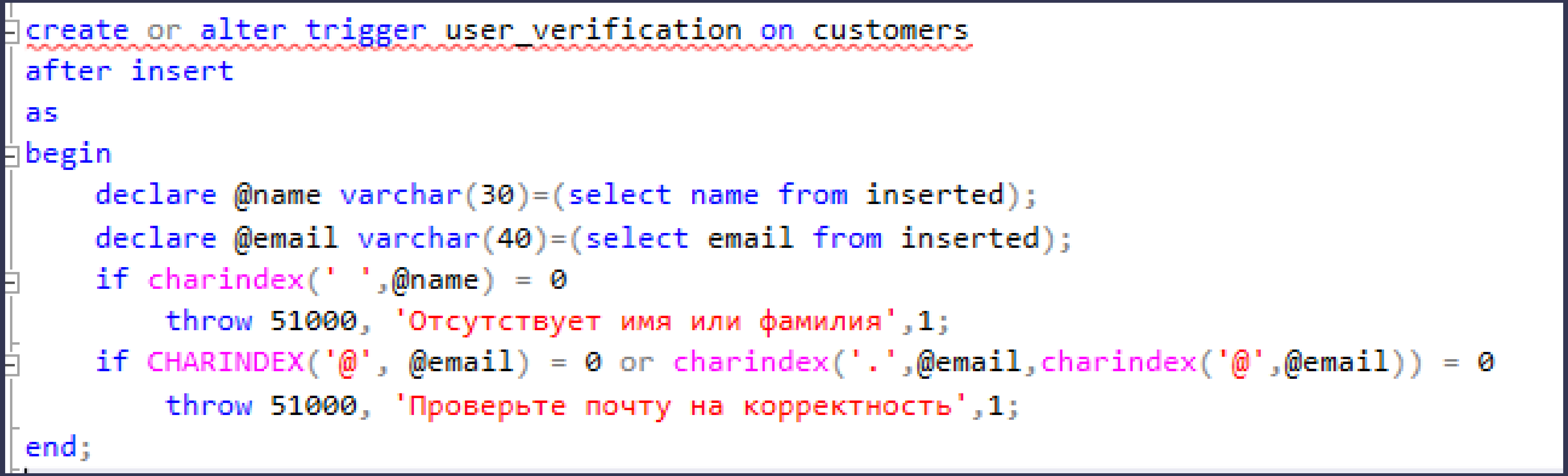
```
сообщение: 51000, уровень: 16, состояние: 1, процедура: user_verification, строка: 8 [строка начала пакета: 122]
Отсутствует имя или фамилия

Время выполнения: 2023-05-12T02:40:34.2377263+06:00
```



```sql
create or alter procedure update_wallet(@cust_id int, @plus int) as
begin
    if (select wallet from CUSTOMERS where customer_id=@cust_id) is null
        update CUSTOMERS
        set wallet = 0
        WHERE customer_id=@cust_id
    update CUSTOMERS
    set wallet = wallet + @plus
    where customer_id=@cust_id
end;
```

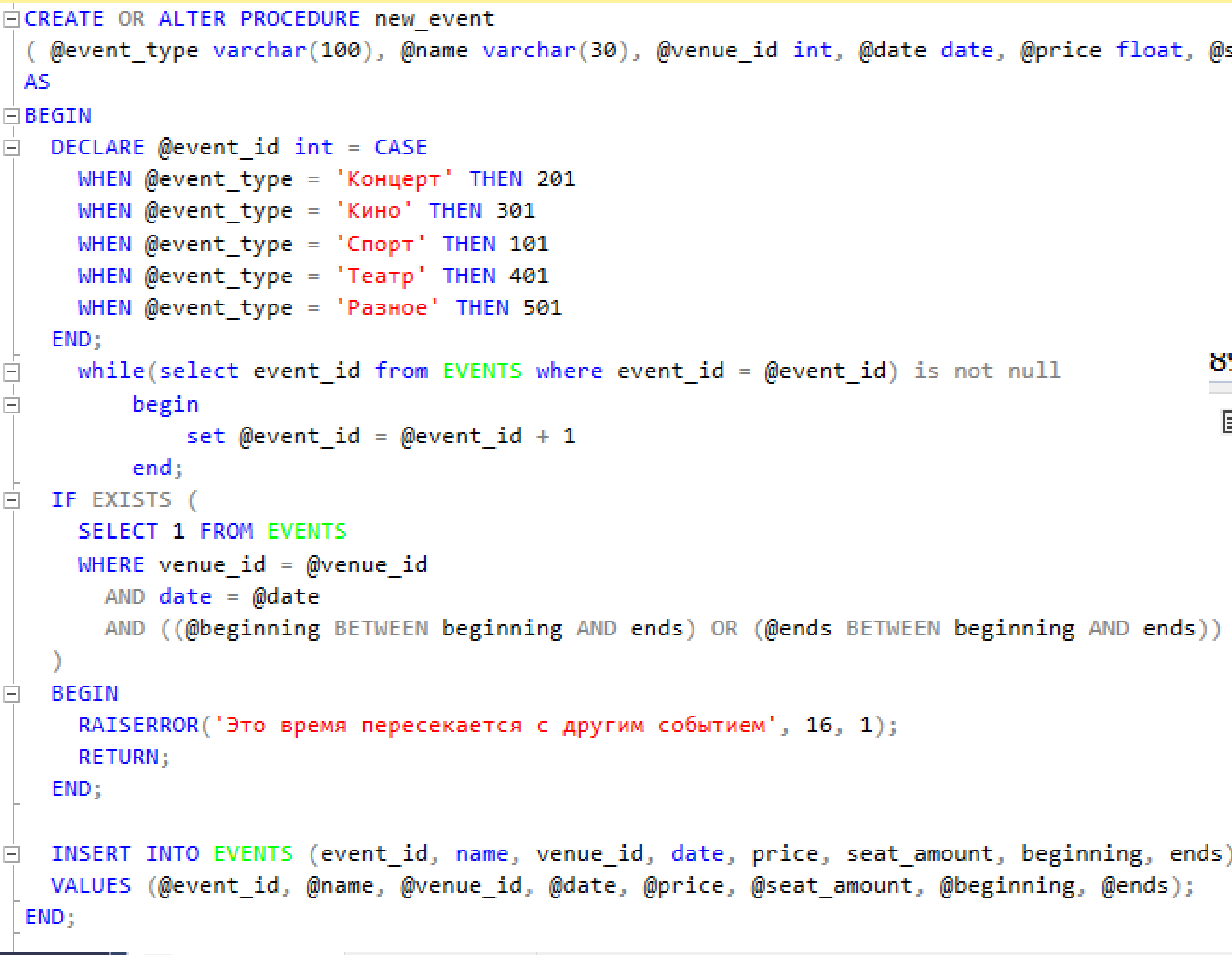


```
сообщение: 51000, уровень: 16, состояние: 1,
Проверьте почту на корректность

Время выполнения: 2023-05-12T02:41:09.6752954
```

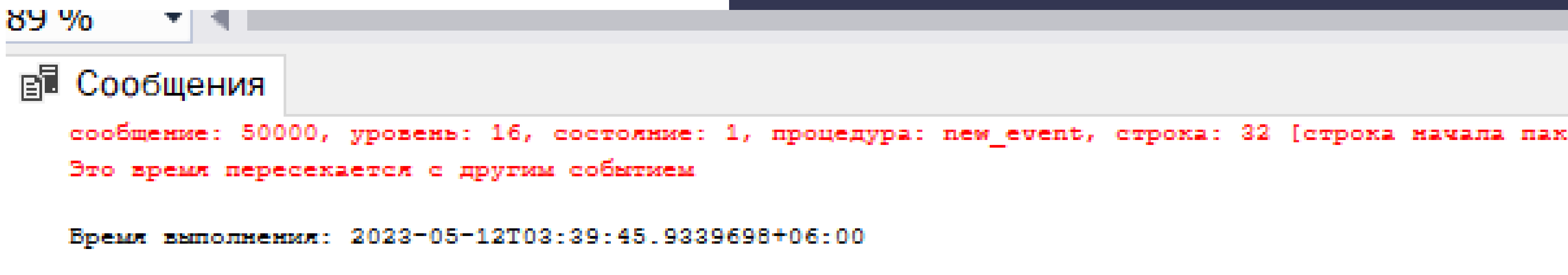


```sql
create or alter trigger user_verification on customers
after insert
as
begin
    declare @name varchar(30)=(select name from inserted);
    declare @email varchar(40)=(select email from inserted);
    if charindex(' ',@name) = 0
        throw 51000, 'Отсутствует имя или фамилия',1;
    if CHARINDEX('@', @email) = 0 or charindex('.',@email,charindex('@',@email)) = 0
        throw 51000, 'Проверьте почту на корректность',1;
end;
```

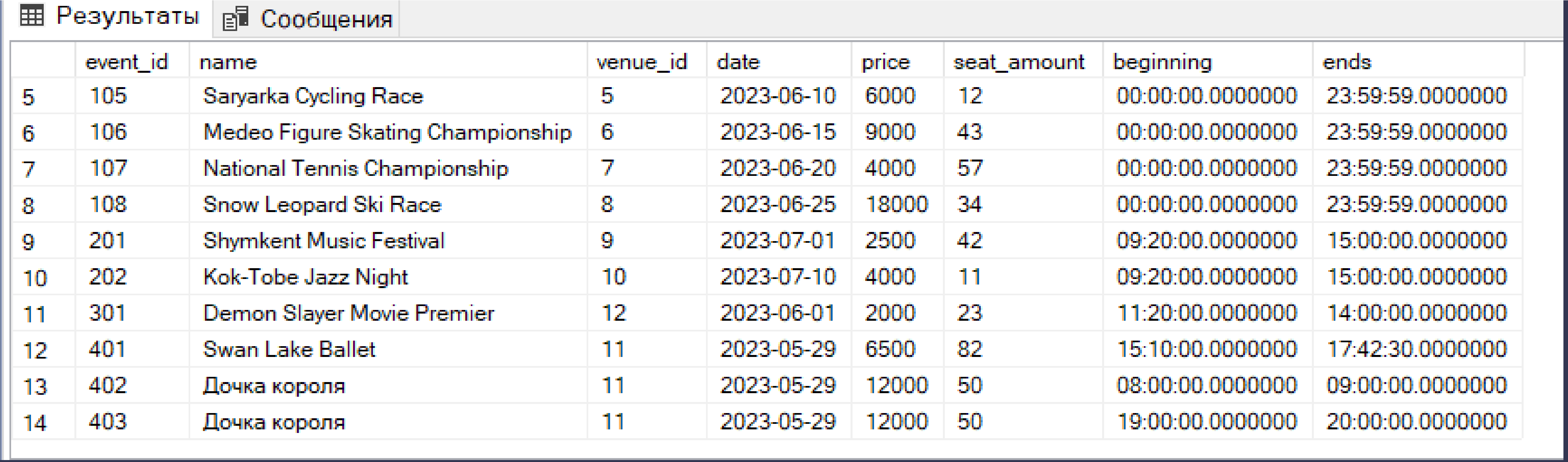
6



```sql
CREATE OR ALTER PROCEDURE new_event
( @event_type varchar(100), @name varchar(30), @venue_id int, @date date, @price float, @seat_amount int, @beginning time, @ends time)
AS
BEGIN
  DECLARE @event_id int = CASE
    WHEN @event_type = 'Концерт' THEN 201
    WHEN @event_type = 'Кино' THEN 301
    WHEN @event_type = 'Спорт' THEN 101
    WHEN @event_type = 'Театр' THEN 401
    WHEN @event_type = 'Разное' THEN 501
  END;
    while(select event_id from EVENTS where event_id = @event_id) is not null
        begin
            set @event_id = @event_id + 1
        end;
  IF EXISTS (
    SELECT 1 FROM EVENTS
    WHERE venue_id = @venue_id
      AND date = @date
      AND ((@beginning BETWEEN beginning AND ends) OR (@ends BETWEEN beginning AND ends))
  )
  BEGIN
    RAISERROR('Это время пересекается с другим событием', 16, 1);
    RETURN;
  END;

  INSERT INTO EVENTS (event_id, name, venue_id, date, price, seat_amount, beginning, ends)
  VALUES (@event_id, @name, @venue_id, @date, @price, @seat_amount, @beginning, @ends);
END;
```



```
сообщение: 50000, уровень: 16, состояние: 1, процедура: new_event, строка: 32 [строка начала пак
Это время пересекается с другим событием

Время выполнения: 2023-05-12T03:39:45.9339698+06:00
```



	event_id	name	venue_id	date	price	seat_amount	beginning	ends
5	105	Saryarka Cycling Race	5	2023-06-10	6000	12	00:00:00.0000000	23:59:59.0000000
6	106	Medeo Figure Skating Championship	6	2023-06-15	9000	43	00:00:00.0000000	23:59:59.0000000
7	107	National Tennis Championship	7	2023-06-20	4000	57	00:00:00.0000000	23:59:59.0000000
8	108	Snow Leopard Ski Race	8	2023-06-25	18000	34	00:00:00.0000000	23:59:59.0000000
9	201	Shymkent Music Festival	9	2023-07-01	2500	42	09:20:00.0000000	15:00:00.0000000
10	202	Kok-Tobe Jazz Night	10	2023-07-10	4000	11	09:20:00.0000000	15:00:00.0000000
11	301	Demon Slayer Movie Premier	12	2023-06-01	2000	23	11:20:00.0000000	14:00:00.0000000
12	401	Swan Lake Ballet	11	2023-05-29	6500	82	15:10:00.0000000	17:42:30.0000000
13	402	Дочка короля	11	2023-05-29	12000	50	08:00:00.0000000	09:00:00.0000000
14	403	Дочка короля	11	2023-05-29	12000	50	19:00:00.0000000	20:00:00.0000000