

## پروژه دیتابیس سامانه مدیریت رستوران

# DATA DOMAIN

نام اعضا :

فاطمه علی زاده

سانیا مسعودی

نام استاد : استاد حسین رحمانی

نام منتور : محمدعلی خسروآبادی

### شرح پروژه

برای مدیریت یک رستوران نیازمند یک سامانه میباشیم . بصورتی که برای ورود هر مشتری به رستوران و استفاده کردن از خدمات آن بتواند یک حساب باز کند . در خود رستوران علاوه بر مشتری ، تعدادی کارمند از جمله : مدیریت رستوران ، پیک ، آشپز، صندوقدار و انباردار داریم که همه باید برای خود پروفایلی ایجاد کنند.

کارمندان با توجه به عنوان شغلی و سطح دسترسی متفاوت ، از هم متمایز میشوند و باتوجه به ساعت کاری حقوق دریافت میکنند.

مشتری ها میتوانند در صورت تمایل اطلاعات اضافی راجب آدرس خود را در پروفایل وارد کنند تا در صورت مراجعه غیرحضورى ، سفارش به دستشان برسد.

با توجه به سوابق خرید هر مشتری، تخفیفاتی شامل او میشود (مثلا تخفیف 25 درصدی برای تعداد سفارش 4 ، بالای 4 تا).

این رستوران شامل انواع منو از جمله (فست فود،سالاد،غذای دریایی،دسر،غذای کودک،غذای وگان،صبحانه،غذای بدون شکر،ایتالیایی و مدیترانه ایی) است که هر منو دارای چندین غذا در خود است ، منو توسط یکی از کارمندان در نقش انباردار بعد از این که مقدار باقی مانده غذا ها 0 شد آپدیت شده تا مشتری ها بتوانند از منو سفارش بدهند .

مشتری ها میتوانند نظرات خود را درباره رستوران هم ثبت کنند تا به مدیریت بهتر ما کمک کنند.در اخرمشتری ها فاکتور خود را دریافت میکند و در صورت نهایی شدن تصمیمشان آن را پرداخت میکنند . همچنین فاکتور ها باید با تاریخ روز ساعت و اطلاعات جزئی ثبت شوند.

مشتری ها میتوانند میزها را برای مراسم خاص از قبل رزرو کنند و اطلاعات فعلی هر میز باید مشخص باشد تا به اشتباه میزی رزرو نشود .

## موجودیت ها

**مشتری** : یک موجودیت مستقل است که باید پروفایل خود را ثبت کند و ویژگی های زیر را دارد :

نام و نام خانوادگی / تاریخ تولد / رمز ورود / شماره تلفن / سن (derived) / نام کاربری ( Key attribute) / ادرس (نام شهر، نام خیابان، کد پستی)

**کارمند** : یک موجودیت مستقل است که با وارد کردن اطلاعات خود میتواند وارد سامانه شود و ویژگی های زیر را دارد:

نام و نام خانوادگی / تاریخ تولد / رمز ورود / شماره تلفن / سن (derived) / نام کاربری ( Key attribute) / ساعت کاری / حقوق / عنوان شغل / سطح دسترسی

**حساب کاربری** : موجودیتی مربوط به اعضا و کاربران است که میتوانند با استفاده از آن از برخی خدمات رستوران مانند رزرو و سفارش و پرداخت و .. استفاده کنند و شامل ویژگی های زیر است:

نام و نام خانوادگی / تاریخ تولد / رمز ورود / شماره تلفن / سن (derived) / نام کاربری ( Key attribute)

مشتری و کارمند ها از حساب کاربری ارث بری میکنند و رابطه (total disjoint) دارند

**فاکتور خرید** : موجودیتی مستقل است که مشتری ها میتوانند هزینه سفارش خود را در آن ببینند و پرداخت کنند و شامل ویژگی های زیر است:

تاریخ / نام مشتری یا کاربر / غذاهای انتخاب شده با قیمت و تعداد / آی دی فاکتور (key) / پرداخت شده یا نشده / ساعت / روز (derived) / هزینه کل

**میز**: یک موجودیت ضعیف مربوط به رستوران است که هر کاربر میتواند یک یا چند میز داشته باشد و شامل ویژگی های زیر را دارد:

نام کاربری کاربر در حال استفاده / شماره میز (key attribute) / رزرو شده یا نشده / ظرفیت میز / تاریخ رزرو / مدت زمان رزرو

**منو** : یک موجودیت مستقل است که انبار دار میتواند آن را اپدیت کند و داری چندین غذا است و شامل ویژگی های زیر است :

نوع منو / آی دی منو (key attr)

**غذا** : یک موجودیت ضعیف مربوط به منو رستوران است که دارای ویژگی های زیر است :

نام غذا / توضیحات راجب غذا / قیمت غذا / شناسه (key attr) / تعداد باقی مانده از آن غذا

**کامنت :** موجودیتی ضعیف مربوط به مشتریان است که میتوانند نظرات خود را ثبت کنند و شامل ویژگی های زیر است :

شناسه کاربر / متن توضیحات / امتیاز / آی دی کامنت (key attr)

**تخفیف ها :** موجودیتی ضعیف مربوط به فاکتور که با توجه به سوابق و تعداد دفعات خرید در نظر گرفته میشود :

نام کاربر / نوع تخفیف / شناسه تخفیف (key) / شناسه فاکتوری که این تخفیف بر آن اعمال میشود

## رابطه ها

**رزرو کردن :** هر مشتری میتواند یک یا چند میز رزرو کند و هر میز میتواند توسط یک مشتری رزرو شده

**سفارش ثبت کردن :** هر مشتری میتواند سفارشی ثبت کند که شامل چند غذا باشد و هر چند غذا میتواند مربوط به سفارش مربوط به یک مشتری باشد

**مشمول بودن :** هر فاکتور میتواند مشمول دریافت چند تخفیف شود .

**حساب داشتن :** هر کاربر میتواند یک حساب و هر حساب میتواند متعلق به یک کاربر باشد .

**نظر دادن (ارسال کامنت) :** هر مشتری میتواند یک یا چند نظر بدهد

**دارا بودن (رابطه بین منو و غذا) :** هر منو دارای چند غذا میتواند باشد و هر چند غذا مربوط به یک منو است.

**دارا بودن (رابطه بین مشتری و فاکتور) :** هر مشتری دارای یک فاکتور است و هر فاکتور فقط متعلق به یک مشتری میتواند باشد.

**آپدیت کردن :** هر کارمند با عنوان شغل انباردار میتواند منوها را آپدیت کند و همه منوها توسط یک کارمند آپدیت میشود.

## کوئری ها

فیلتر کردن غذا بر اساس نوع منویی که غذا در آن قرار دارد (فست فود، سالاد، غذای دریایی، دسر، غذای کودک، غذای وگان، صبحانه، غذای بدون شکر، ایتالیایی و مدیترانه ای)

جستجوی غذا بر اساس نام آن

فیلتر کردن میز بر اساس ظرفیت

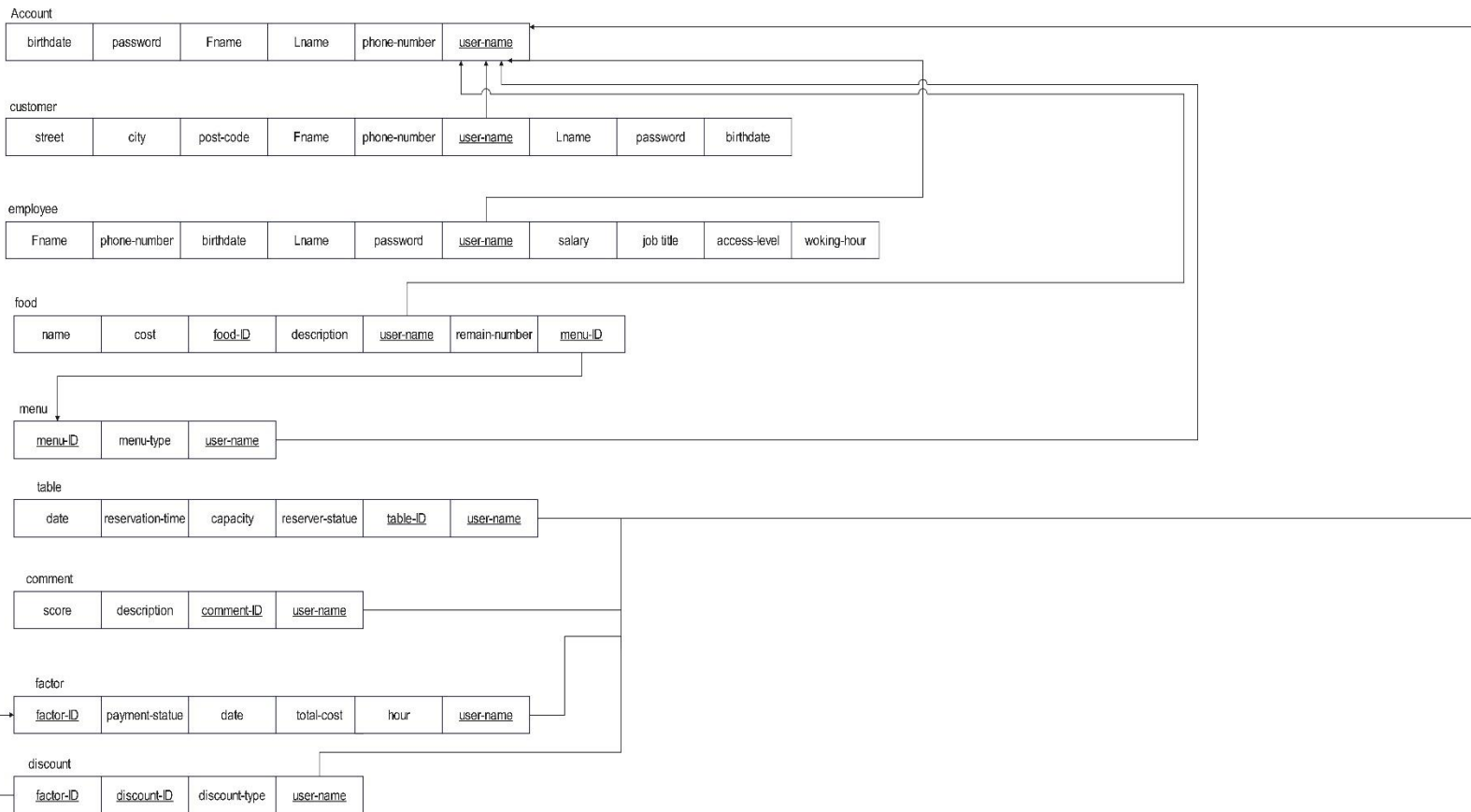
فیلتر کردن فاکتور بر اساس پرداخت شده یا نشده

جستجوی پرسنل بر اساس نام کاربری

دریافت حساب هایی که تلفن آنها با یک شروع میشود و تاریخ تولد آنها بعد از سال 2000 است



# TABLE MODEL



# QUERY

فیلتر کردن غذا بر اساس نوع منو

SELECT name, cost, description, menu\_type

FROM Food

JOIN Menu ON Food.menu\_ID = Menu.menu\_ID

ORDER BY menu\_type;

SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```
-- فیلتر کردن غذا بر اساس نوع
SELECT name, cost, description, menu_type
FROM Food
JOIN Menu ON Food.menu_ID = Menu.menu_ID
ORDER BY menu_type;
```

90 %

Results Messages

	name	cost	description	menu_type
1	Egg	12.00	Delicious pasta with your choice of sauce.	Brunch
2	Omelette	15.00	Grilled steak served with your choice of sides.	Brunch
3	Jelly	6.00	Sweet dessert to end your meal.	Dessert
4	Pizza	10.00	Delicious pizza with your choice of toppings.	FastFood
5	Hamburger	8.00	Juicy hamburger with your choice of toppings.	FastFood
6	French Fries	5.00	Crispy french fries served with your choice of dip...	FastFood
7	Chicken Wings	9.00	Spicy or mild chicken wings served with your cho...	FastFood
8	Salad Shirazi	7.00	Fresh salad with your choice of toppings.	Salad
9	Seafood	14.00	Fresh seafood cooked to perfection.	Seafood
10	Sushi	16.00	Delicious sushi made with fresh ingredients.	Seafood

جستجو غذا بر اساس نام --

```
SELECT name, cost, description
FROM Food
WHERE name='Pizza';
```

SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```
-- جستجو غذا بر اساس نام
SELECT name, cost, description
FROM Food
WHERE name='Pizza';
```

90 %

Results Messages

	name	cost	description
1	Pizza	10.00	Delicious pizza with your choice of toppings.

فیلتر کردن میز بر اساس ظرفیت --

```
SELECT *
FROM ReserveTable
WHERE reserve_status=1;
```

SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```
-- فیلتر کردن میز بر اساس ظرفیت
SELECT *
FROM ReserveTable
WHERE reserve_status=1;
```

90 %

Results Messages

	date	reservation_time	capacity	reserve_status	table_ID	user_name
1	2023-03-08	18:00:00.0000000	4	1	1	johndoe
2	2023-03-09	19:00:00.0000000	6	1	2	janesmith
3	2023-03-10	20:00:00.0000000	8	1	3	michaeljones
4	2023-03-11	21:00:00.0000000	4	1	4	sarahwilliams
5	2023-03-12	22:00:00.0000000	6	1	5	davidbrown

فیلتر کردن فاکتور بر اساس پرداخت شده یا نشده --

```
SELECT *
FROM Factor
WHERE payment_status=0 ;
```

SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```
-- فیلتر کردن فاکتور بر اساس پرداخت شده یا نشده
SELECT *
FROM Factor
WHERE payment_status=0 ;
```

90 %

Results Messages

	factor_ID	payment_status	date	total_cost	hour	user_name
1	2	0	2023-03-09	120.00	19:00:00.0000000	janesmith
2	4	0	2023-03-11	160.00	21:00:00.0000000	sarahwilliams
3	6	0	2023-03-13	200.00	18:00:00.0000000	jessicagarcia
4	8	0	2023-03-15	240.00	20:00:00.0000000	emilyperez
5	10	0	2023-03-17	280.00	22:00:00.0000000	annannguyen

جستجوی پرسنل بر اساس نام کاربری --

```
SELECT *
FROM Employee
WHERE user_name='abigailjohnson';
```

SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```
-- جستجوی پرسنل بر اساس نام کاربری
SELECT *
FROM Employee
WHERE user_name='abigailjohnson';
```

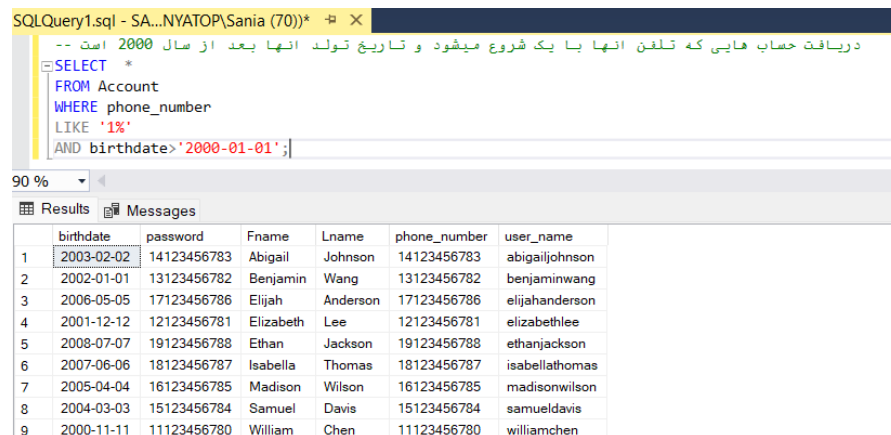
90 %

Results Messages

	salary	job_title	access_level	working_hour	birthdate	password	Fname	Lname	phone_number	user_name
1	26000.00	Keeper	2	12:00:00.0000000	2003-02-02	14123456783	Abigail	Johnson	14123456783	abigailjohnson

دریافت حساب هایی که تلفن آنها با یک شروع میشود و تاریخ تولد آنها بعد از سال 2000 است --

```
SELECT *  
  
FROM Account  
  
WHERE phone_number  
  
LIKE '1%'  
  
AND birthdate>'2000-01-01';
```

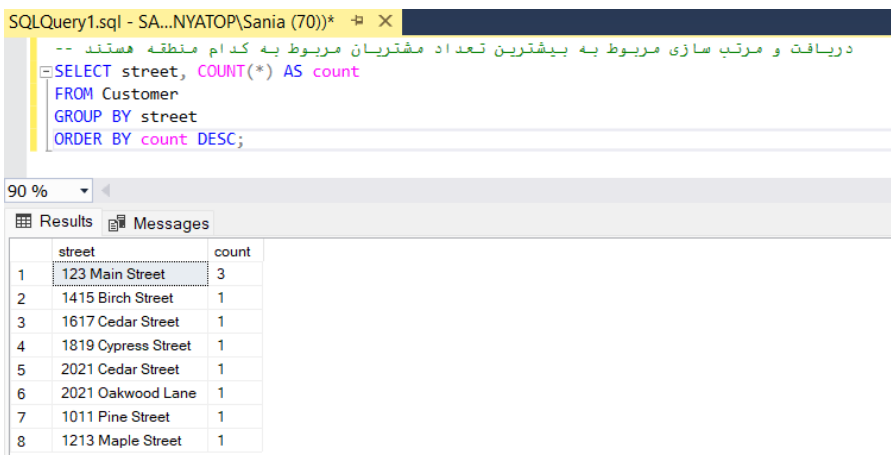


The screenshot shows a SQL query window with the following text: `-- دریافت حساب هایی که تلفن آنها با یک شروع میشود و تاریخ تولد آنها بعد از سال 2000 است --`  
`SELECT *`  
`FROM Account`  
`WHERE phone_number`  
`LIKE '1%'`  
`AND birthdate>'2000-01-01';`  
The results table has 9 rows and 7 columns: birthdate, password, Fname, Lname, phone\_number, user\_name. The first row is highlighted.

	birthdate	password	Fname	Lname	phone_number	user_name
1	2003-02-02	14123456783	Abigail	Johnson	14123456783	abigailjohnson
2	2002-01-01	13123456782	Benjamin	Wang	13123456782	benjaminwang
3	2006-05-05	17123456786	Elijah	Anderson	17123456786	elijahanderson
4	2001-12-12	12123456781	Elizabeth	Lee	12123456781	elizabethlee
5	2008-07-07	19123456788	Ethan	Jackson	19123456788	ethanjackson
6	2007-06-06	18123456787	Isabella	Thomas	18123456787	isabellathomas
7	2005-04-04	16123456785	Madison	Wilson	16123456785	madisonwilson
8	2004-03-03	15123456784	Samuel	Davis	15123456784	samueldavis
9	2000-11-11	11123456780	William	Chen	11123456780	williamchen

دریافت و مرتب سازی مربوط به بیشترین تعداد مشتریان مربوط به کدام منطقه هستند --

```
SELECT street, COUNT(*) AS count  
  
FROM Customer  
  
GROUP BY street  
  
ORDER BY count DESC;
```

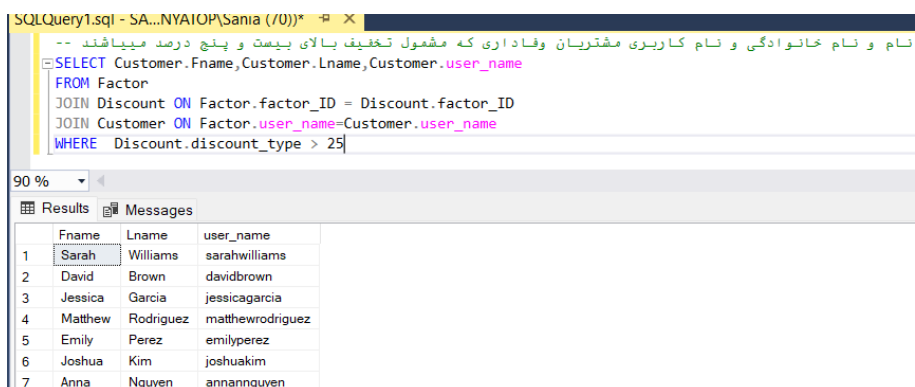


The screenshot shows a SQL query window with the following text: `-- دریافت و مرتب سازی مربوط به بیشترین تعداد مشتریان مربوط به کدام منطقه هستند --`  
`SELECT street, COUNT(*) AS count`  
`FROM Customer`  
`GROUP BY street`  
`ORDER BY count DESC;`  
The results table has 2 columns: street, count. The first row is highlighted.

	street	count
1	123 Main Street	3
2	1415 Birch Street	1
3	1617 Cedar Street	1
4	1819 Cypress Street	1
5	2021 Cedar Street	1
6	2021 Oakwood Lane	1
7	1011 Pine Street	1
8	1213 Maple Street	1

نام و نام خانوادگی و نام کاربری مشتریان وفاداری که مشمول تخفیف بالای بیست و پنج درصد میباشند

```
SELECT Customer.Fname, Customer.Lname, Customer.user_name  
  
FROM Factor  
  
JOIN Discount ON Factor.factor_ID = Discount.factor_ID  
  
JOIN Customer ON  
  
Factor.user_name=Customer.user_name  
  
WHERE Discount.discount_type > 25
```

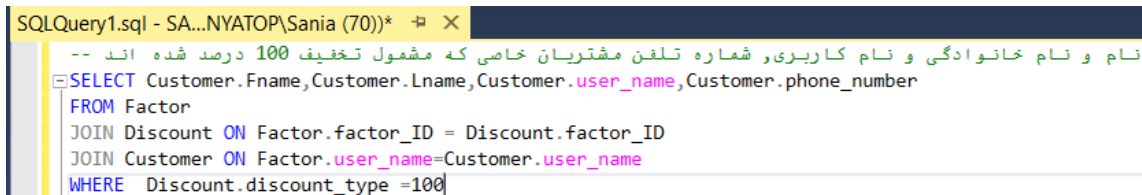


The screenshot shows a SQL query window with the following text: `-- نام و نام خانوادگی و نام کاربری مشتریان وفاداری که مشمول تخفیف بالای بیست و پنج درصد میباشند --`  
`SELECT Customer.Fname, Customer.Lname, Customer.user_name`  
`FROM Factor`  
`JOIN Discount ON Factor.factor_ID = Discount.factor_ID`  
`JOIN Customer ON Factor.user_name=Customer.user_name`  
`WHERE Discount.discount_type > 25`  
The results table has 3 columns: Fname, Lname, user\_name. The first row is highlighted.

	Fname	Lname	user_name
1	Sarah	Williams	sarahwilliams
2	David	Brown	davidbrown
3	Jessica	Garcia	jessicagarcia
4	Matthew	Rodriguez	matthewrodriguez
5	Emily	Perez	emilyperez
6	Joshua	Kim	joshuakim
7	Anna	Nguyen	annannguyen

نام و نام خانوادگی و نام کاربری, شماره تلفن مشتریان خاصی که مشمول تخفیف 100 درصد شده اند --

```
SELECT Customer.Fname, Customer.Lname, Customer.user_name, Customer.phone_number
FROM Factor
JOIN Discount ON Factor.factor_ID = Discount.factor_ID
JOIN Customer ON Factor.user_name=Customer.user_name
WHERE Discount.discount_type =100
```



SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```
-- نام و نام خانوادگی و نام کاربری, شماره تلفن مشتریان خاصی که مشمول تخفیف 100 درصد شده اند
SELECT Customer.Fname, Customer.Lname, Customer.user_name, Customer.phone_number
FROM Factor
JOIN Discount ON Factor.factor_ID = Discount.factor_ID
JOIN Customer ON Factor.user_name=Customer.user_name
WHERE Discount.discount_type =100
```

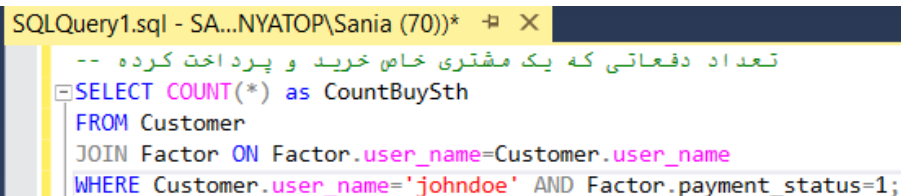
90 %

Results Messages

	Fname	Lname	user_name	phone_number
1	Joshua	Kim	joshuakim	9012345678
2	Anna	Nguyen	annannguyen	10123456789

تعداد دفعاتی که یک مشتری خاص خرید و پرداخت کرده --

```
SELECT COUNT(*) as CountBuySth
FROM Customer
JOIN Factor ON Factor.user_name=Customer.user_name
WHERE Customer.user_name='johndoe' AND Factor.payment_status=1;
```



SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```
-- تعداد دفعاتی که یک مشتری خاص خرید و پرداخت کرده
SELECT COUNT(*) as CountBuySth
FROM Customer
JOIN Factor ON Factor.user_name=Customer.user_name
WHERE Customer.user_name='johndoe' AND Factor.payment_status=1;
```

90 %

Results Messages

	CountBuySth
1	1



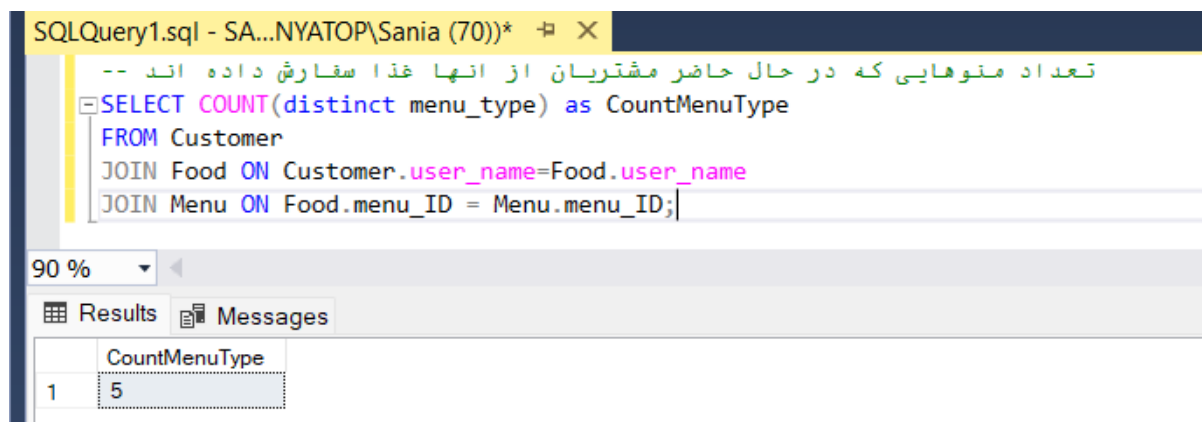
تعداد منوهایی که در حال حاضر مشتریان از آنها غذا سفارش داده اند --

```
SELECT COUNT(distinct menu_type) as CountMenuType
```

```
FROM Customer
```

```
JOIN Food ON Customer.user_name=Food.user_name
```

```
JOIN Menu ON Food.menu_ID = Menu.menu_ID;
```



The screenshot shows a SQL query window titled 'SQLQuery1.sql - SA...NYATOP\Sania (70))\*'. The query text is as follows:

```
-- تعداد منوهایی که در حال حاضر مشتریان از آنها غذا سفارش داده اند
SELECT COUNT(distinct menu_type) as CountMenuType
FROM Customer
JOIN Food ON Customer.user_name=Food.user_name
JOIN Menu ON Food.menu_ID = Menu.menu_ID;
```

Below the query window, the 'Results' tab is active, displaying a single row of data:

	CountMenuType
1	5

تعداد باقی مانده تمام غذا ها را اپدیت کرده و به اندازه یک واحد افزایش میدهد --

```
SELECT * FROM Food;
```

```
GO
```

```
CREATE PROCEDURE UPDATE_REMAIN_NUMBER_OF_FOOD
```

```
@FOOD_ID TINYINT,
```

```
@REMAIN_NUMBER TINYINT
```

```
AS
```

```
BEGIN
```

```
    UPDATE Food
```

```
    SET Food.remain_number= Food.remain_number + @REMAIN_NUMBER
```

```
    WHERE Food.food_ID=@FOOD_ID;
```

```
END;
```

```
GO
```

```
DECLARE FOOD_CURSOR CURSOR FOR
```

```

SELECT Food.food_ID FROM Food

OPEN FOOD_CURSOR;

DECLARE @FOOD_ID TINYINT;

FETCH NEXT FROM FOOD_CURSOR INTO @FOOD_ID;

WHILE @@FETCH_STATUS=0

BEGIN

    EXECUTE UPDATE_REMAIN_NUMBER_OF_FOOD @FOOD_ID,1;

    FETCH NEXT FROM FOOD_CURSOR INTO @FOOD_ID;

END;

CLOSE FOOD_CURSOR;

DEALLOCATE FOOD_CURSOR;

SELECT * FROM Food;

```

SQLQuery1.sql - SA...NYATOP\Sania (70))\*

```

-- تعداد باقی مانده تمام غذا ها را ایدیت کرده و به اندازه یک واحد افزایش میدهد
GO
DROP PROCEDURE UPDATE_REMAIN_NUMBER_OF_FOOD;
SELECT * FROM Food;
GO
CREATE PROCEDURE UPDATE_REMAIN_NUMBER_OF_FOOD
@FOOD_ID TINYINT,
@REMAIN_NUMBER TINYINT
AS
BEGIN
    UPDATE Food
    SET Food.remain_number= Food.remain_number + @REMAIN_NUMBER
    WHERE Food.food_ID=@FOOD_ID;
END;
GO
DECLARE FOOD_CURSOR CURSOR FOR
SELECT Food.food_ID
FROM Food;

OPEN FOOD_CURSOR;
DECLARE @FOOD_ID TINYINT;
FETCH NEXT FROM FOOD_CURSOR INTO @FOOD_ID;
WHILE @@FETCH_STATUS=0
BEGIN
    EXECUTE UPDATE_REMAIN_NUMBER_OF_FOOD @FOOD_ID,1;
    FETCH NEXT FROM FOOD_CURSOR INTO @FOOD_ID;
END;
CLOSE FOOD_CURSOR;
DEALLOCATE FOOD_CURSOR;
GO
SELECT * FROM Food

```

61 %

Results Messages

	name	cost	food_ID	description	user_name	remain_number	menu_ID
1	Pizza	10.00	1	Delicious pizza with your choice of toppings.	johndoe	16	1
2	Hamburger	8.00	2	Juicy hamburger with your choice of toppings.	janesmith	21	1
3	French Fries	5.00	3	Crispy french fries served with your choice of dip...	michaeljones	26	1
4	Chicken Wings	9.00	4	Spicy or mild chicken wings served with your cho...	sarahwilliams	18	1
5	Salad Shirazi	7.00	5	Fresh salad with your choice of toppings.	davidbrown	24	2
6	Egg	12.00	6	Delicious pasta with your choice of sauce.	jessicagarcia	16	3
7	Omelette	15.00	7	Grilled steak served with your choice of sides.	matthewrodriguez	14	3
8	Seafood	14.00	8	Fresh seafood cooked to perfection.	emilyperez	16	4

	name	cost	food_ID	description	user_name	remain_number	menu_ID
1	Pizza	10.00	1	Delicious pizza with your choice of toppings.	johndoe	17	1
2	Hamburger	8.00	2	Juicy hamburger with your choice of toppings.	janesmith	22	1
3	French Fries	5.00	3	Crispy french fries served with your choice of dip...	michaeljones	27	1
4	Chicken Wings	9.00	4	Spicy or mild chicken wings served with your cho...	sarahwilliams	19	1
5	Salad Shirazi	7.00	5	Fresh salad with your choice of toppings.	davidbrown	25	2
6	Egg	12.00	6	Delicious pasta with your choice of sauce.	jessicagarcia	17	3
7	Omelette	15.00	7	Grilled steak served with your choice of sides.	matthewrod...	15	3
8	Seafood	14.00	8	Fresh seafood cooked to perfection.	emilyperez	17	4

اخراج کردن اشیانی با سطح دسترسی پایین مساوی 3 و حذف کردن حساب کاربری مربوطشان --

```
SELECT * FROM Employee;
```

```
GO
```

```
CREATE FUNCTION GET_EMPLOYEE_TO_KICK_OUT()
```

```
RETURNS TABLE
```

```
AS
```

```
RETURN (
```

```
    SELECT user_name
```

```
    FROM Employee
```

```
    WHERE job_title = 'Chef' AND access_level = 3);
```

```
GO
```

```
DELETE FROM Employee
```

```
WHERE user_name IN (SELECT user_name FROM dbo.GET_EMPLOYEE_TO_KICK_OUT());
```

```
DELETE FROM Account
```

```
WHERE user_name IN (SELECT user_name FROM dbo.GET_EMPLOYEE_TO_KICK_OUT());
```

```
GO
```

```
SELECT * FROM Employee;
```

SQLQuery1.sql - SA...NYATOP\Sania (70)\* - H X

```
-- اخراج کردن اشیانی با سطح دسترسی پایین 3 و حذف کردن حساب کاربری مربوطشان
SELECT * FROM Employee;
DROP FUNCTION GET_EMPLOYEE_TO_KICK_OUT;
GO
|
|
|
CREATE FUNCTION GET_EMPLOYEE_TO_KICK_OUT()
RETURNS TABLE
AS
RETURN (
    SELECT user_name
    FROM Employee
    WHERE job_title = 'Chef' AND access_level = 3
);
GO
DELETE FROM Employee
WHERE user_name IN (SELECT user_name FROM GET_EMPLOYEE_TO_KICK_OUT());
DELETE FROM Account
WHERE user_name IN (SELECT user_name FROM GET_EMPLOYEE_TO_KICK_OUT());
GO
SELECT * FROM Employee;
```

75 %

Results Messages

	salary	job_title	access_level	working_hour	birthdate	password	Fname	Lname	phone_number	user_name
1	26000.00	Keeper	2	12:00:00.0000000	2003-02-02	14123456783	Abigail	Johnson	14123456783	abigailjohnson
2	24000.00	Delivery	2	11:00:00.0000000	2002-01-01	13123456782	Benjamin	Wang	13123456782	benjaminwang
3	32000.00	Cashier	1	03:00:00.0000000	2006-05-05	17123456786	Elijah	Anderson	17123456786	elijahanderson
4	22000.00	Keeper	1	10:00:00.0000000	2001-12-12	12123456781	Elizabeth	Lee	12123456781	elizabethlee
5	38000.00	Chef	1	05:00:00.0000000	2008-07-07	19123456788	Ethan	Jackson	19123456788	ethanjackson
6	34000.00	Dishwasher	2	04:00:00.0000000	2007-06-06	18123456787	Isabella	Thomas	18123456787	isabellathomas
7	30000.00	Chef	1	02:00:00.0000000	2005-04-04	16123456785	Madison	Wilson	16123456785	madisonwilson
8	38000.00	Chef	3	06:00:00.0000000	2009-08-08	20123456789	Olivia	White	20123456789	oliviawhite
9	28000.00	Manager	3	01:00:00.0000000	2004-03-03	15123456784	Samuel	Davis	15123456784	samueldavis
10	20000.00	Waiter	1	09:00:00.0000000	2000-11-11	11123456780	William	Chen	11123456780	williamchen

	salary	job_title	access_level	working_hour	birthdate	password	Fname	Lname	phone_number	user_name
1	26000.00	Keeper	2	12:00:00.0000000	2003-02-02	14123456783	Abigail	Johnson	14123456783	abigailjohnson
2	24000.00	Delivery	2	11:00:00.0000000	2002-01-01	13123456782	Benjamin	Wang	13123456782	benjaminwang
3	32000.00	Cashier	1	03:00:00.0000000	2006-05-05	17123456786	Elijah	Anderson	17123456786	elijahanderson
4	22000.00	Keeper	1	10:00:00.0000000	2001-12-12	12123456781	Elizabeth	Lee	12123456781	elizabethlee
5	38000.00	Chef	1	05:00:00.0000000	2008-07-07	19123456788	Ethan	Jackson	19123456788	ethanjackson
6	34000.00	Dishwasher	2	04:00:00.0000000	2007-06-06	18123456787	Isabella	Thomas	18123456787	isabellathomas
7	30000.00	Chef	1	02:00:00.0000000	2005-04-04	16123456785	Madison	Wilson	16123456785	madisonwilson
8	28000.00	Manager	3	01:00:00.0000000	2004-03-03	15123456784	Samuel	Davis	15123456784	samueldavis
9	20000.00	Waiter	1	09:00:00.0000000	2000-11-11	11123456780	William	Chen	11123456780	williamchen

اضافه کردن یک منو جدید به رستوران توسط انبار دار با سطح دسترسی بالا مساوی 1 --

```
SELECT * FROM Menu;
```

```
GO
```

```
DECLARE @ID TINYINT;
```

```
DECLARE @UserName NVARCHAR(256);
```

```
SELECT @ID = ISNULL(MAX(menu_ID), 0) + 1 FROM Menu;
```

```
SELECT TOP 1 @UserName = user_name FROM Employee
```

```
WHERE job_title = 'Keeper' AND access_level = 1;
```

```
INSERT INTO Menu (menu_ID, menu_type, user_name)
```

```
VALUES (@ID, 'new menu type', @UserName);
```

```
SELECT * FROM Menu;
```

SQLQuery1.sql - SA...NYATOP\Sania (70))\* ✕

```
-- اضافه کردن یک منو جدید به رستوران توسط انبار دار با سطح دسترسی بالا مساوی 1 --
SELECT * FROM Menu
GO
DECLARE @ID TINYINT;
DECLARE @UserName NVARCHAR(256);

SELECT @ID = ISNULL(MAX(menu_ID), 0) + 1 FROM Menu;

SELECT TOP 1 @UserName = user_name FROM Employee
WHERE job_title = 'Keeper' AND access_level = 1;

INSERT INTO Menu (menu_ID, menu_type, user_name)
VALUES (@ID, 'new menu type', @UserName);
SELECT * FROM Menu
```

100 %

Results Messages

	menu_ID	menu_type	user_name
1	1	FastFood	elizabethlee
2	2	Salad	elizabethlee
3	3	Brunch	elizabethlee
4	4	Seafood	elizabethlee
5	5	Dessert	elizabethlee
6	6	Kids	abigailjohnson
7	7	Vegan	abigailjohnson
8	8	Gluten-Free	abigailjohnson
9	9	Italian food	abigailjohnson
10	10	Mediterranean	abigailjohnson

	menu_ID	menu_type	user_name
1	1	FastFood	elizabethlee
2	2	Salad	elizabethlee
3	3	Brunch	elizabethlee
4	4	Seafood	elizabethlee
5	5	Dessert	elizabethlee
6	6	Kids	abigailjohnson
7	7	Vegan	abigailjohnson
8	8	Gluten-Free	abigailjohnson
9	9	Italian food	abigailjohnson
10	10	Mediterra...	abigailjohnson
11	11	new men...	elizabethlee

اعمال کردن تخفیف به روی همه فاکتورهای مشمول تخفیف شده و ابدیت کرن هزینه کل پرداخت --

```
SELECT * FROM Factor;
```

```
SELECT * FROM Discount;
```

```
GO
```

```
DECLARE FACTOR_CURSOR CURSOR FOR
```

```
SELECT factor_ID
```

```
FROM Factor;
```

```
OPEN FACTOR_CURSOR;
```

```
DECLARE @FACTOR_ID TINYINT;
```

```
DECLARE @DISCOUNT_TYPE TINYINT;
```

```
FETCH NEXT FROM FACTOR_CURSOR INTO @FACTOR_ID;
```

```
WHILE @@FETCH_STATUS = 0
```

```
BEGIN
```

```
    SELECT @DISCOUNT_TYPE = discount_type
```

```
    FROM Discount
```

```
    WHERE factor_ID = @FACTOR_ID;
```

```
    IF @DISCOUNT_TYPE IS NOT NULL
```

```
    BEGIN
```

```
        UPDATE Factor
```

```
        SET total_cost = (total_cost * (100 - @DISCOUNT_TYPE)) / 100
```

```
        WHERE factor_ID = @FACTOR_ID;
```

```
    END;
```

```
    FETCH NEXT FROM FACTOR_CURSOR INTO @FACTOR_ID;
```

```
END;
```

```
CLOSE FACTOR_CURSOR;
```

```
DEALLOCATE FACTOR_CURSOR;
```

```
SELECT * FROM Factor;
```

```

SQLQuery1.sql - SA...NYATOP\Sania (70))*
-- اعمال کردن تخفیف به روی همه فاکتور های مشمول تخفیف شده و اپدیت کردن هزینه کل پرداخت
SELECT * FROM Factor;SELECT * FROM Discount;
GO
DECLARE FACTOR_CURSOR CURSOR FOR
SELECT factor_ID
FROM Factor;
OPEN FACTOR_CURSOR;
DECLARE @FACTOR_ID TINYINT;
DECLARE @DISCOUNT_TYPE TINYINT;
FETCH NEXT FROM FACTOR_CURSOR INTO @FACTOR_ID;
WHILE @@FETCH_STATUS = 0
BEGIN
    SELECT @DISCOUNT_TYPE = discount_type
    FROM Discount
    WHERE factor_ID = @FACTOR_ID;
    IF @DISCOUNT_TYPE IS NOT NULL
    BEGIN
        UPDATE Factor
        SET total_cost = (total_cost * (100 - @DISCOUNT_TYPE)) / 100
        WHERE factor_ID = @FACTOR_ID;
    END;
    FETCH NEXT FROM FACTOR_CURSOR INTO @FACTOR_ID;
END;
CLOSE FACTOR_CURSOR;
DEALLOCATE FACTOR_CURSOR; SELECT * FROM Factor;

```

75 %

Results Messages

	factor_ID	payment_status	date	total_cost	hour	user_name
1	1	1	2023-03-08	100.00	18:00:00.0000000	johndoe
2	2	0	2023-03-09	120.00	19:00:00.0000000	janesmith
3	3	1	2023-03-10	140.00	20:00:00.0000000	michaeljones
4	4	0	2023-03-11	160.00	21:00:00.0000000	sarahwilliams
5	5	1	2023-03-12	180.00	22:00:00.0000000	davidbrown

	discount_ID	factor_ID	discount_type	user_name
1	1	1	25	johndoe
2	2	2	25	janesmith
3	3	3	25	michaeljones
4	4	4	50	sarahwilliams
5	5	5	50	davidbrown

	factor_ID	payment_status	date	total_cost	hour	user_name
1	1	1	2023-03-08	75.00	18:00:00.0000000	johndoe
2	2	0	2023-03-09	90.00	19:00:00.0000000	janesmith
3	3	1	2023-03-10	105.00	20:00:00.0000000	michaeljo...
4	4	0	2023-03-11	80.00	21:00:00.0000000	sarahwilli...
5	5	1	2023-03-12	90.00	22:00:00.0000000	davidbrown

جلوگیری از چندبار حساب باز کردن کاربران متفاوت با شماره تلفن های یکسان --

CREATE TRIGGER PREVENT\_DUPLICATE\_PHONENUMBER

ON Account

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS(

SELECT 1

FROM Account

JOIN inserted ON inserted.phone\_number=Account.phone\_number

AND Account.user\_name != inserted.user\_name)

BEGIN

ROLLBACK TRANSACTION;

END;

END;

GO

INSERT INTO Account (birthdate, password, Fname, Lname, phone\_number, user\_name)  
VALUES

('2000-01-01', 13123456783, 'Test', 'User', 10003456789, 'testuser');

INSERT INTO Account (birthdate, password, Fname, Lname, phone\_number, user\_name)  
VALUES

('2001-01-01', 12123456783, 'Another', 'User', 10003456789, 'anotheruser');

SQLQuery1.sql - SA...NYATOP\Sania (70))\*

-- جلوگیری از چندبار حساب باز کردن کاربران متفاوت با شماره تلفن های یکسان

DROP TRIGGER PREVENT\_DUPLICATE\_PHONENUMBER;

GO

CREATE TRIGGER PREVENT\_DUPLICATE\_PHONENUMBER

ON Account

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS(

SELECT 1

FROM Account

JOIN inserted ON inserted.phone\_number=Account.phone\_number

AND Account.user\_name != inserted.user\_name)

BEGIN

ROLLBACK TRANSACTION;

END;

END;

GO

INSERT INTO Account (birthdate, password, Fname, Lname, phone\_number, user\_name)

VALUES ('2000-01-01', 13123456783, 'Test', 'User', 10003456789, 'testuser');

GO

INSERT INTO Account (birthdate, password, Fname, Lname, phone\_number, user\_name)

VALUES ('2001-01-01', 12123456783, 'Another', 'User', 10003456789, 'anotheruser');

100 %

Messages

(1 row affected)

Msg 3609, Level 16, State 1, Line 23

The transaction ended in the trigger. The batch has been aborted.

Completion time: 2024-06-03T19:00:40.9671205+03:30

نمایش تخفیفاتی که براساس سوابق خرید مشتری شاملش میشود. مثلاً تخفیف 25 درصدی برای تعداد سفارش 4 و 4 به بالا

WITH OrderCounts AS (

```
SELECT user_name,  
COUNT(*) AS order_count  
FROM Factor  
GROUP BY user_name
```

)

```
SELECT oc.user_name,
```

```
CASE
```

```
WHEN oc.order_count >= 4
```

```
THEN 25
```

```
ELSE 0
```

```
END AS discount
```

```
FROM OrderCounts oc;
```

نمایش تخفیفاتی که براساس سوابق خرید مشتری شاملش میشود. مثلاً تخفیف 25 درصدی برای تعداد سفارش 4 و 4 به بالا

```
WITH OrderCounts AS (  
    SELECT user_name, COUNT(*) AS order_count  
    FROM Factor  
    GROUP BY user_name  
)  
  
SELECT oc.user_name,  
    CASE  
        WHEN oc.order_count >= 4 THEN 25  
        ELSE 0  
    END AS discount  
FROM OrderCounts oc;
```

110 %

	user_name	discount
1	annannguyen	0
2	davidbrown	0
3	emilyperez	0
4	janesmith	0
5	jessicagarcia	0
6	johndoe	25
7	joshuakim	0
8	matthewrodriguez	0
9	michaeljones	0
10	sarahwilliams	0

محاسبه سن یک فرد خاص

```
SELECT DATEDIFF(YEAR,Account.birthdate,GETDATE()) AS AGE
```

```
FROM Account
```

```
WHERE Account.user_name='johndoe';
```

```
-- محاسبه سن یک فرد خاص  
SELECT DATEDIFF(YEAR,Account.birthdate,GETDATE()) AS AGE  
FROM Account  
WHERE Account.user_name='johndoe';
```

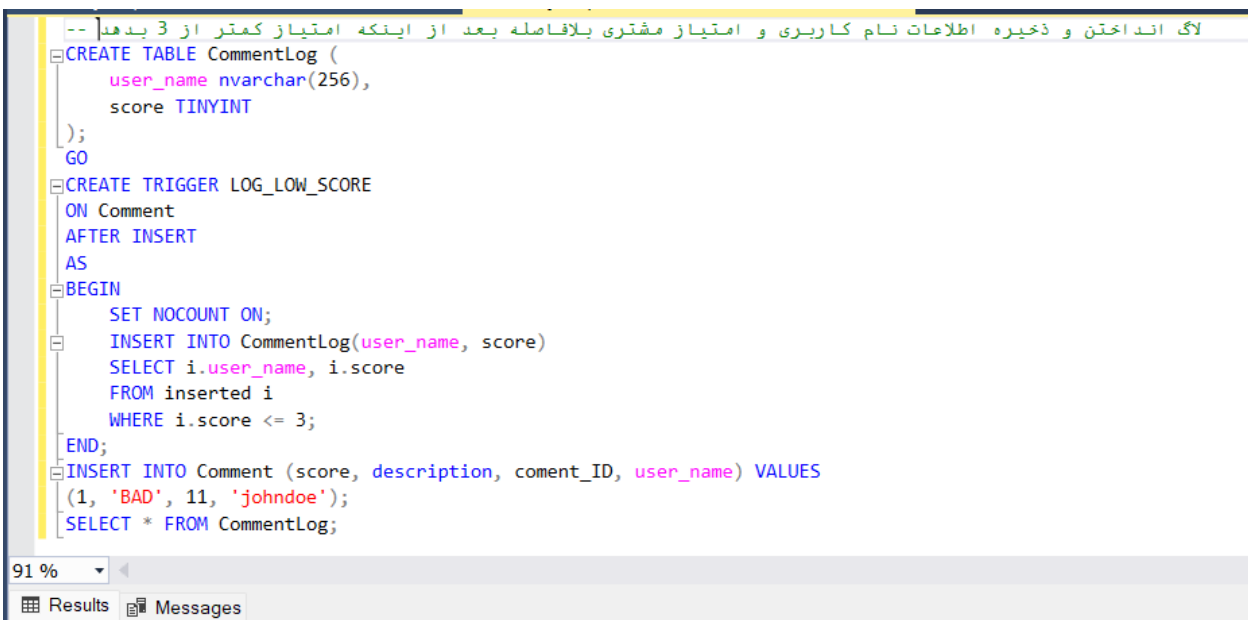
91 %

	AGE
1	34



لاگ انداختن و ذخیره اطلاعات نام کاربری و امتیاز مشتری بلافاصله بعد از اینکه امتیاز کمتر از 3 بدهد --

```
CREATE TABLE CommentLog (  
    user_name nvarchar(256),  
    score TINYINT  
);  
  
GO  
  
CREATE TRIGGER LOG_LOW_SCORE  
ON Comment  
AFTER INSERT  
AS  
BEGIN  
    SET NOCOUNT ON;  
  
    INSERT INTO CommentLog(user_name, score)  
    SELECT i.user_name, i.score  
    FROM inserted i  
    WHERE i.score <= 3;  
  
END;  
  
INSERT INTO Comment (score, description, coment_ID, user_name) VALUES  
(1, 'BAD', 11, 'johndoe');  
  
SELECT * FROM CommentLog;
```



```
-- لاگ انداختن و ذخیره اطلاعات نام کاربری و امتیاز مشتری بلافاصله بعد از اینکه امتیاز کمتر از 3 بدهد --  
CREATE TABLE CommentLog (  
    user_name nvarchar(256),  
    score TINYINT  
);  
GO  
CREATE TRIGGER LOG_LOW_SCORE  
ON Comment  
AFTER INSERT  
AS  
BEGIN  
    SET NOCOUNT ON;  
    INSERT INTO CommentLog(user_name, score)  
    SELECT i.user_name, i.score  
    FROM inserted i  
    WHERE i.score <= 3;  
END;  
INSERT INTO Comment (score, description, coment_ID, user_name) VALUES  
(1, 'BAD', 11, 'johndoe');  
SELECT * FROM CommentLog;
```