**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 ازخدمات آن بتواند یک حساب باز کند . در خود رستوران علاوه بر مشتری ، تعدادی کارمند از جمله : مدیریت رستوران ، پیک ، اشپز، 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 غیرحضوری ، 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 داری 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 جزیی ثبت شوند.

مشتری ها میتوانند میزهارا برای مراسم خاص از قبل رزرو کنند و اطلاعات فعلی هرمیز 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 میتوان 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 اعضاوکاربران است که میتوانند با استفاده از آن ازبرخی 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 ان ببینید 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 میتونه یک یا چند میز داشته باشه و شامل ویژگی های زیر را دارد:

نام کاربری کاربر درحال استفاده /شماره میز (key attribbute) / 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 میتونه یک یا چند میز رزرو کنه و هرمیز میتونه توسط یک مشتری رزرو شه

**سفارش ثبت کردن** :هر مشتری میتونه 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 میتونه یک حساب و هر حساب میتونه متعلق به یک کاربر باشد .

**نظر دادن(ارسال کامنت)** :هر مشتری میتونه یک یا چند نظر بده

**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 منو ها توسط یک کارمند اپدیت 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 ایی)

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 انها با یک شروع میشود و تاریخ تولد انها بعد از سال 2000 است

تعداد دفعاتی که یک مشتری خاص خرید و پرداخت کرده

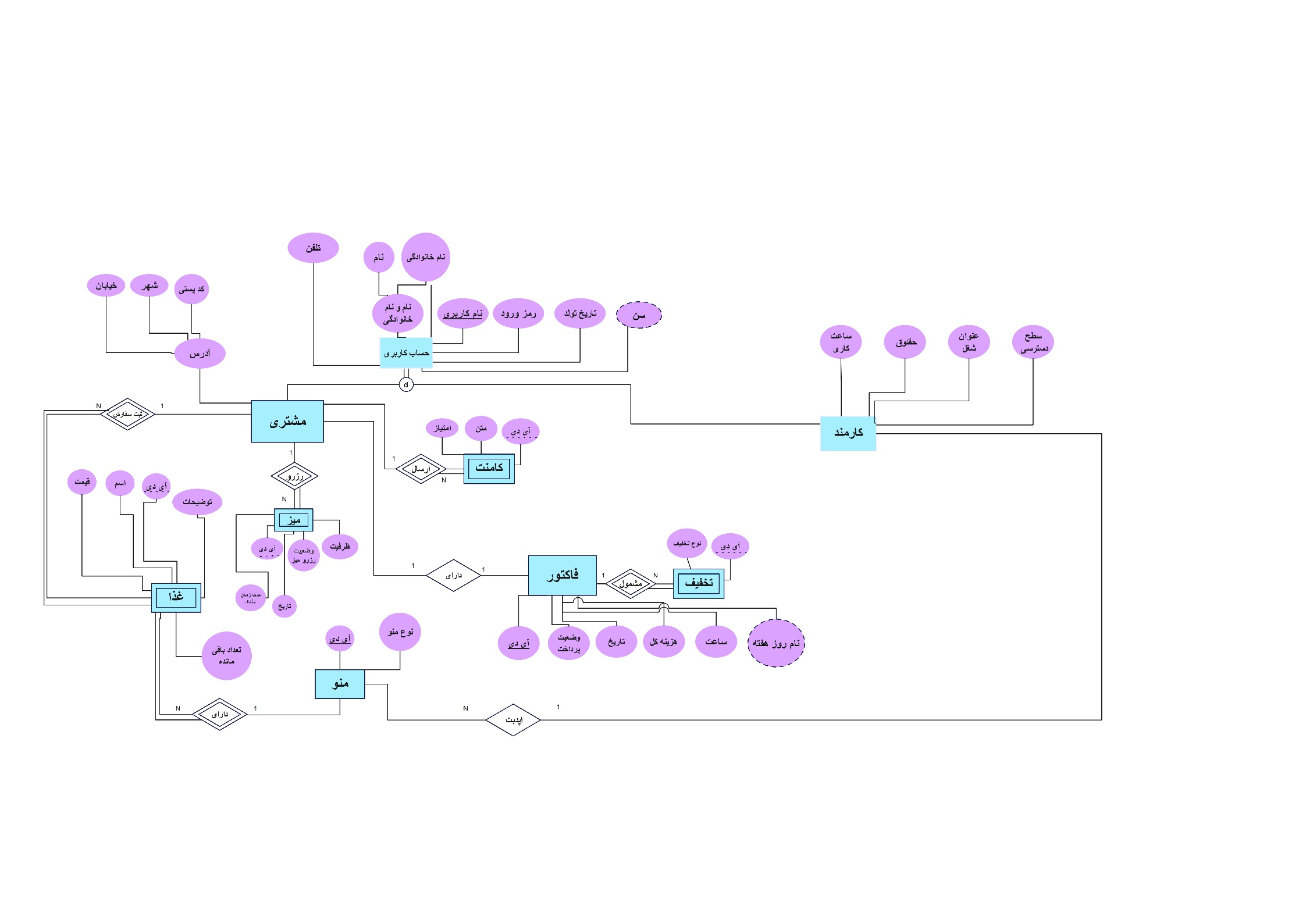
تعداد باقی مانده تمام غذا ها را اپدیت کرده و به اندازه یک واحد افزایش میدهد

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

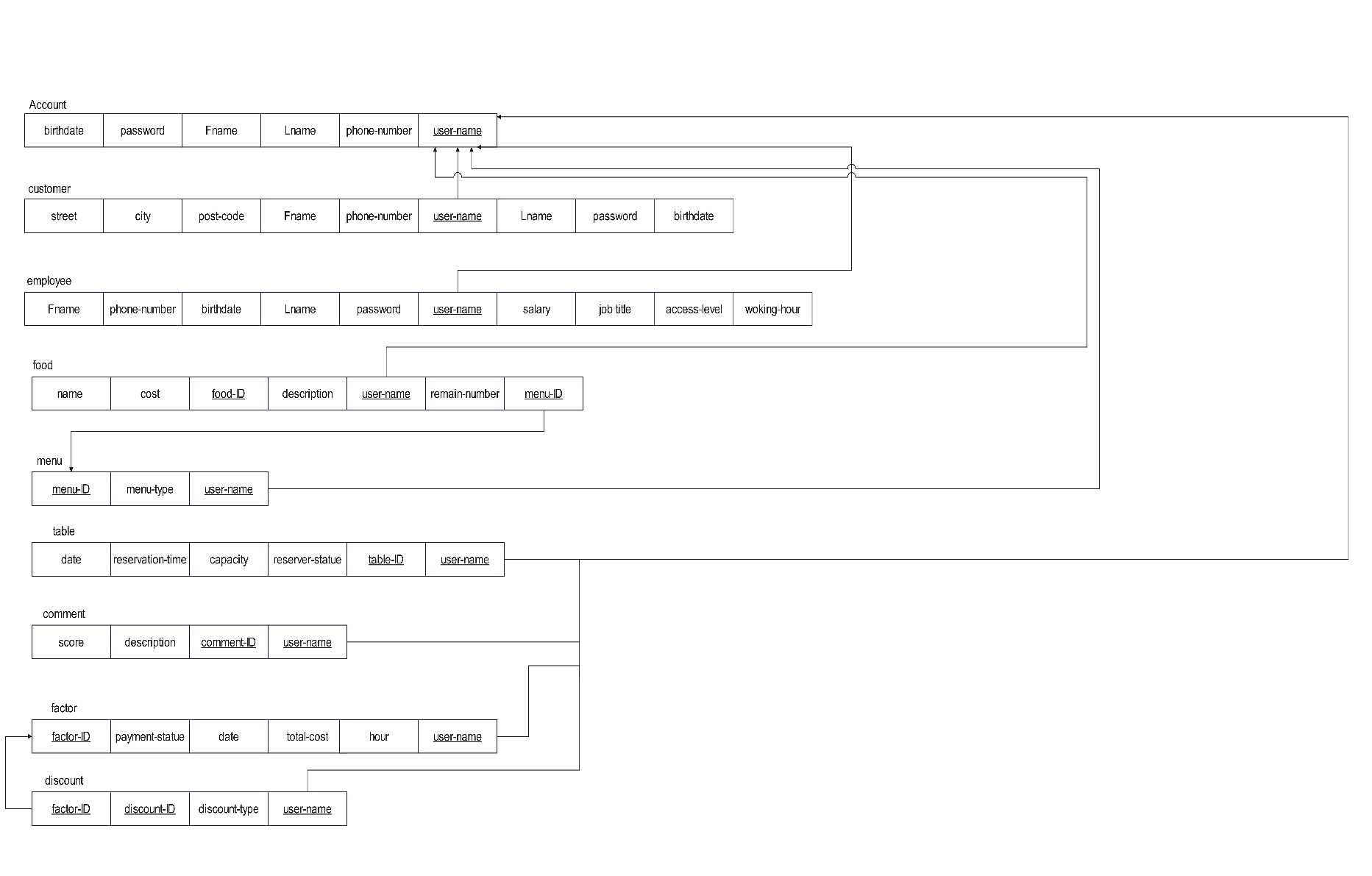
اخراج کردن اشپزانی با سطح دسترسی پایین مساوی 3 و حذف کردن حساب کاربری مربوطشان

-----------------------------------------------------------------------------------------------

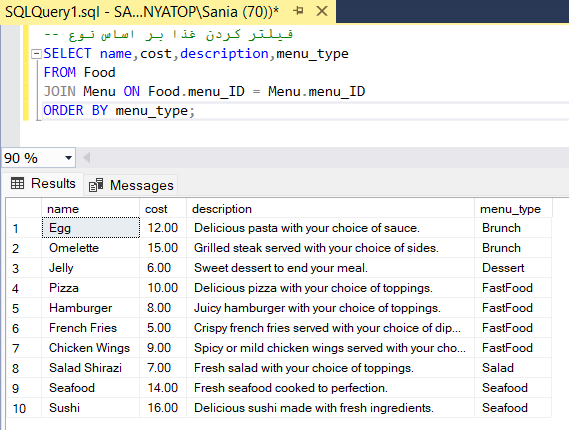
**ER**

****

**TABLE MODEL**

****

**QUERY**



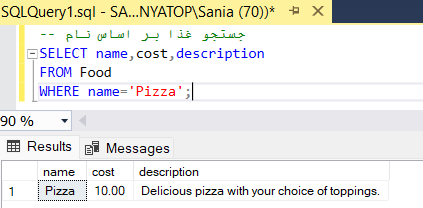
-- فیلتر کردن غذا بر اساس نوع منو

SELECT name,cost,description,menu\_type

FROM Food

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

ORDER BY menu\_type;

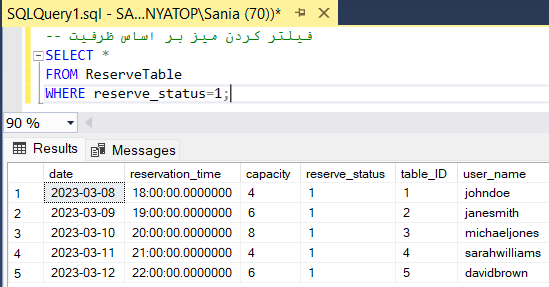
-- جستجو غذا بر اساس نام

SELECT name,cost,description

FROM Food

WHERE name='Pizza';

------------------------------------------------------------------------------------------------------------------------------------------



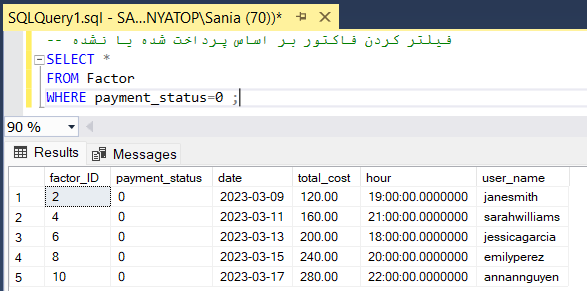
-- فیلتر کردن میز بر اساس ظرفیت

SELECT \*

FROM ReserveTable

WHERE reserve\_status=1;

------------------------------------------------------------------------------------------------------------------------------------------

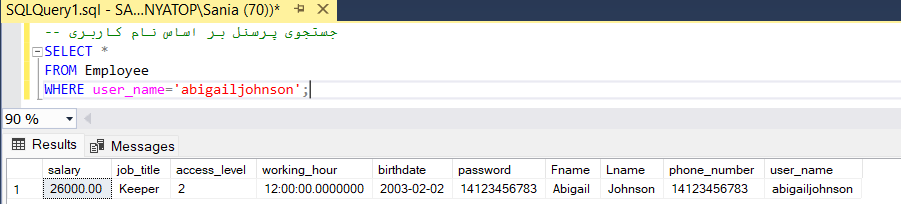
-- فیلتر کردن فاکتور بر اساس پرداخت شده یا نشده

SELECT \*

FROM Factor

WHERE payment\_status=0 ;

------------------------------------------------------------------------------------------------------------------------------------------

-- جستجوی پرسنل بر اساس نام کاربری

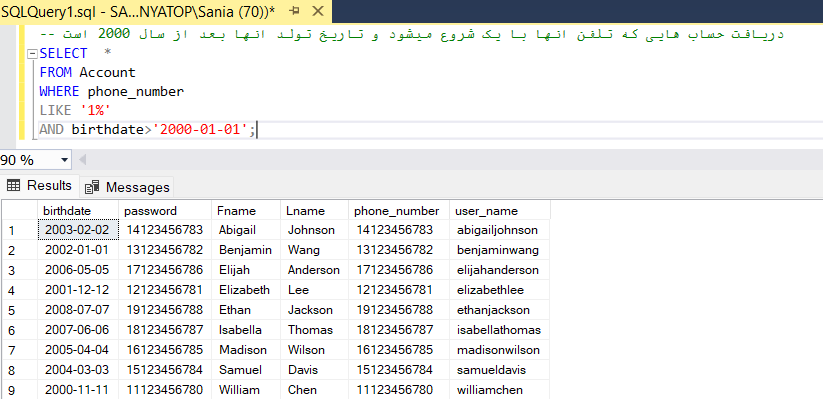
SELECT \*

FROM Employee

WHERE user\_name='abigailjohnson';

------------------------------------------------------------------------------------------------------------------------------------------

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

SELECT \*

FROM Account

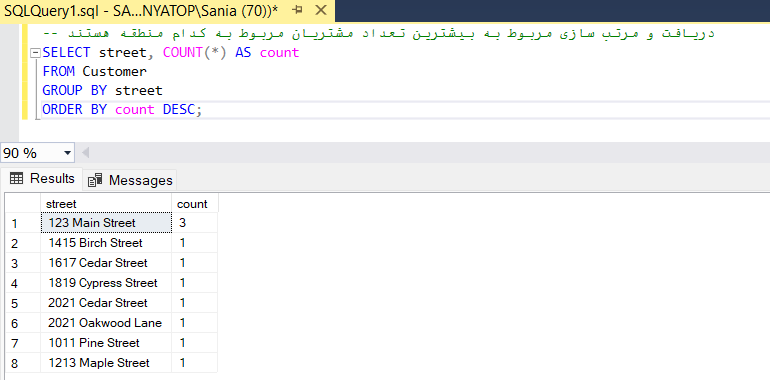
WHERE phone\_number

LIKE '1%'

AND birthdate>'2000-01-01';

------------------------------------------------------------------------------------------------------------------------------------------

-- 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;

------------------------------------------------------------------------------------------------------------------------------------------

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

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

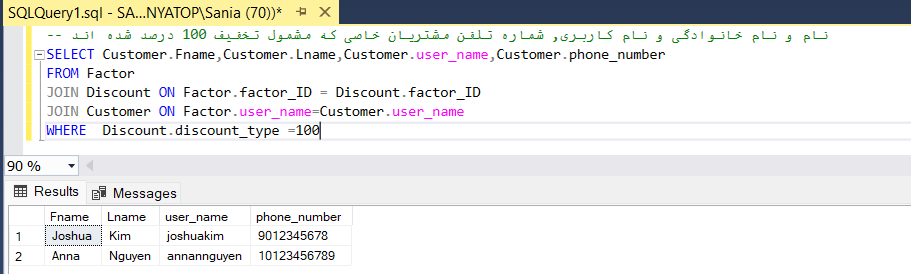
-- نام و نام خانوادگی و نام کاربری, شماره تلفن مشتریان خاصی که مشمول تخفیف 100 درصد شده اند

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

------------------------------------------------------------------------------------------------------------------------------------------

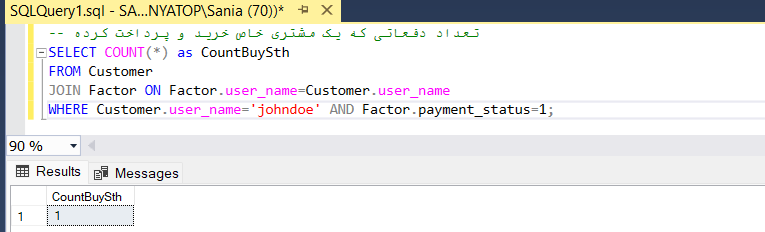
-- تعداد دفعاتی که یک مشتری خاص خرید و پرداخت کرده

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;



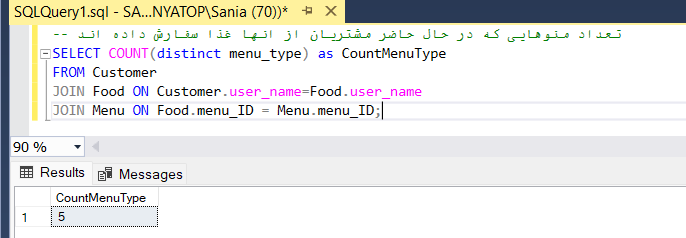
------------------------------------------------------------------------------------------------------------------------------------------

-- تعداد منوهایی که در حال حاضر مشتریان از انها 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;

------------------------------------------------------------------------------------------------------------------------------------------

-- 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

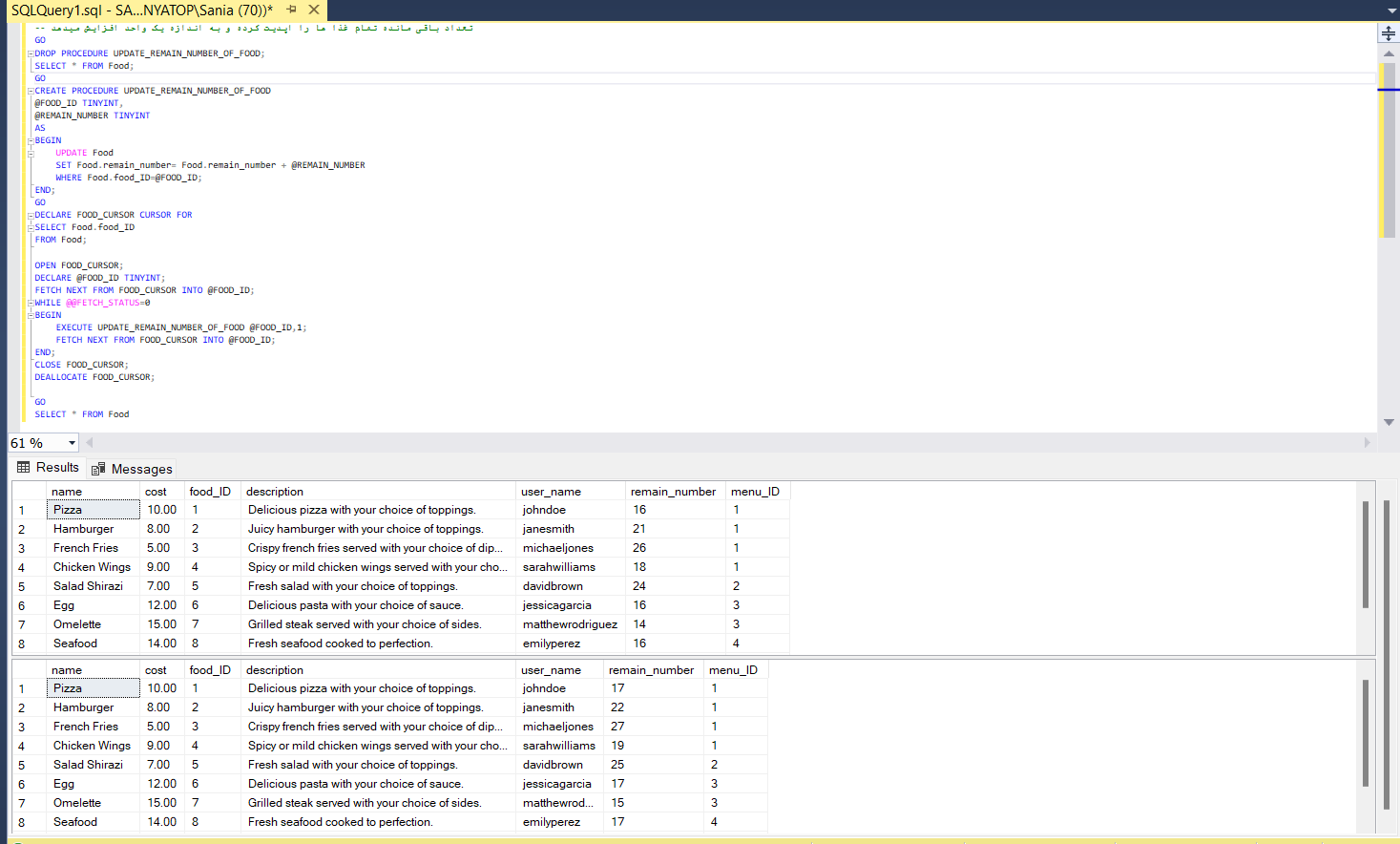
EXECUTE UPDATE\_REMAIN\_NUMBER\_OF\_FOOD @FOOD\_ID,1;

FETCH NEXT FROM FOOD\_CURSOR INTO @FOOD\_ID;

END;

CLOSE FOOD\_CURSOR;

DEALLOCATE FOOD\_CURSOR;

SELECT \* FROM Food;

-- اخراج کردن اشپزانی 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

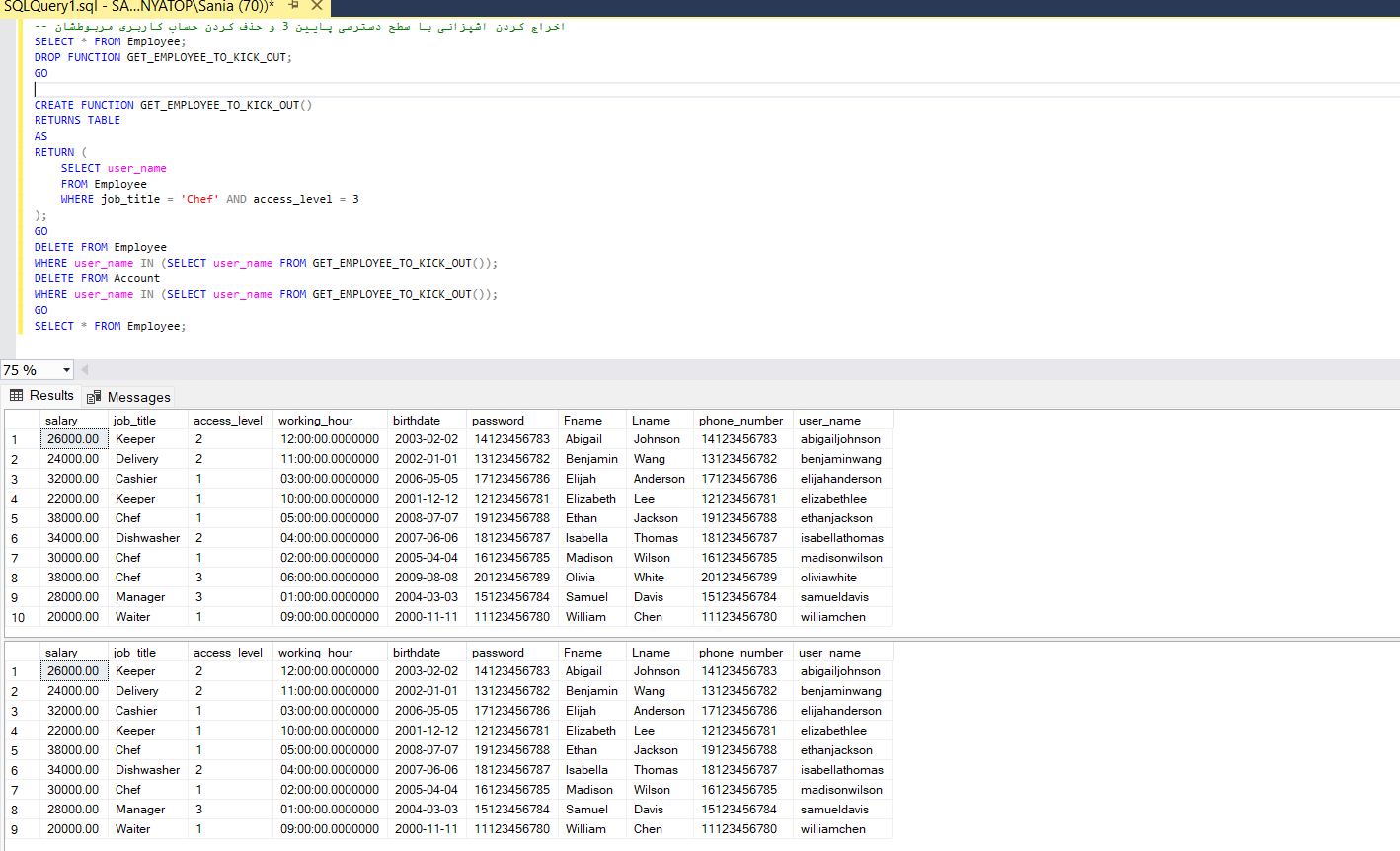
DELETE FROM Employee

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;

-- 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;

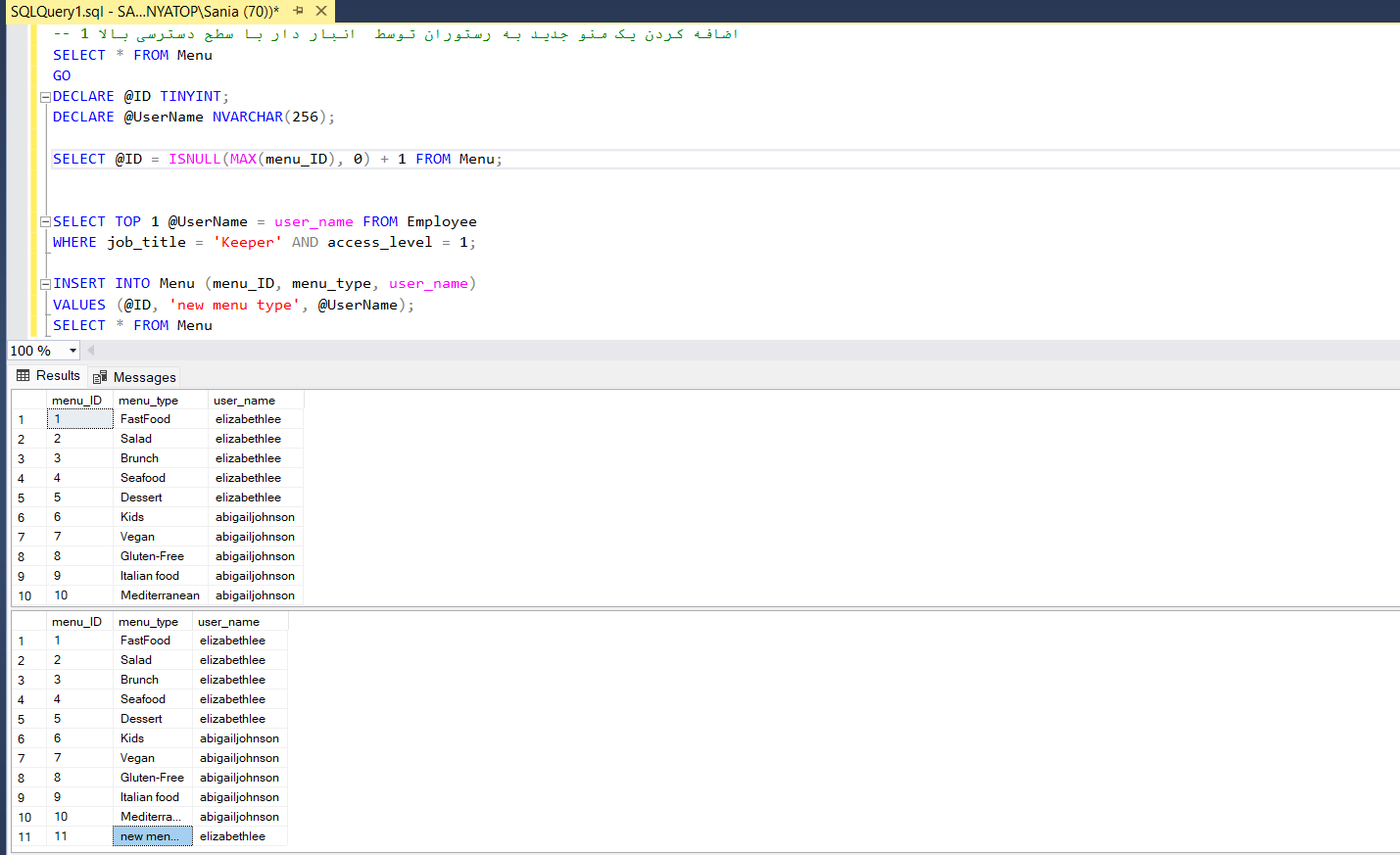
SELECT TOP 1 @UserName = user\_name FROM Employee

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;



-- اعمال کردن تخفیف به روی همه فاکتور های مشمول تخفیف شده و اپدیت 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;

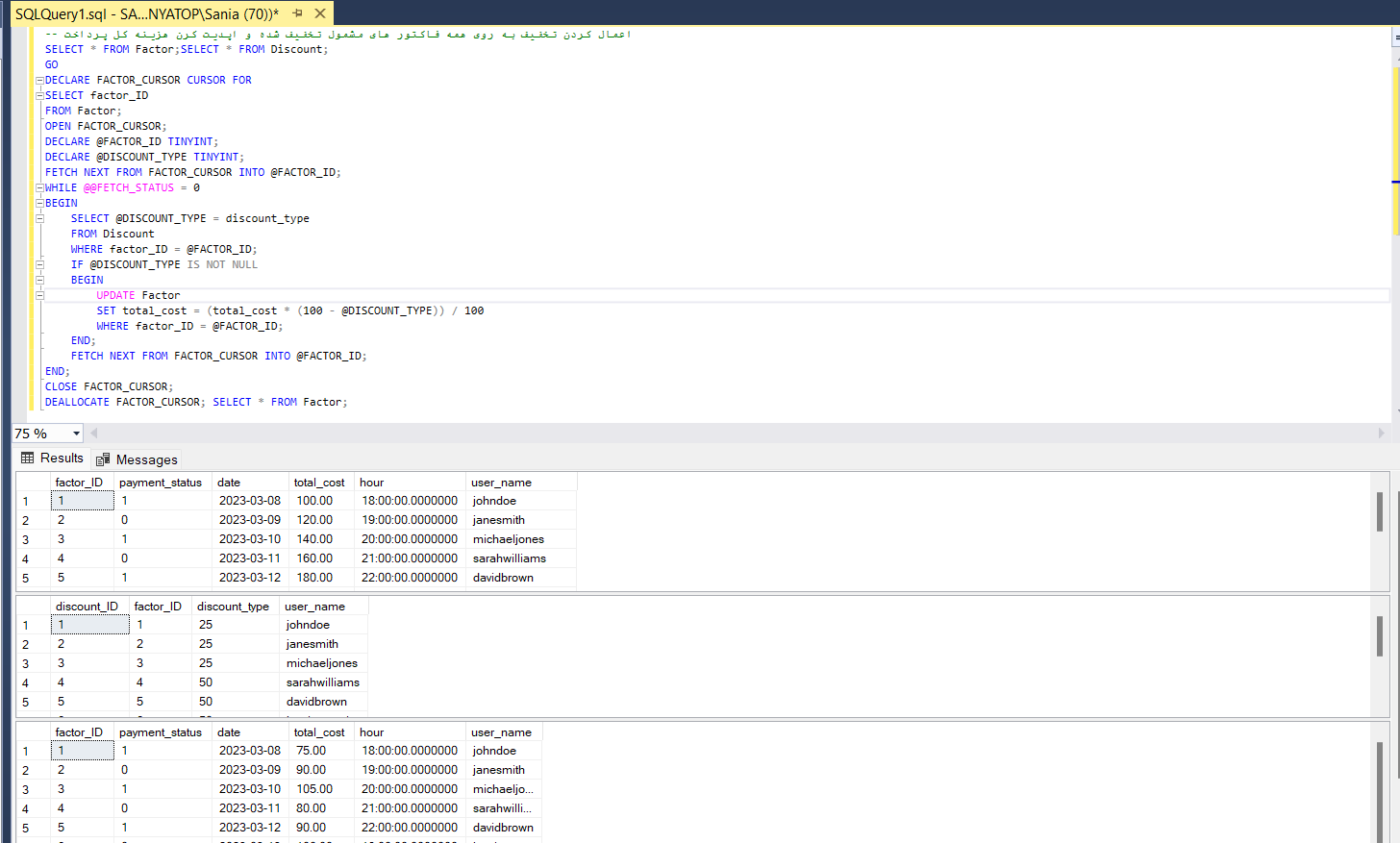
FETCH NEXT FROM FACTOR\_CURSOR INTO @FACTOR\_ID;

END;

CLOSE FACTOR\_CURSOR;

DEALLOCATE FACTOR\_CURSOR;

SELECT \* FROM Factor;



------------------------------------------------------------------------------------------------------------------------------------------

-- 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

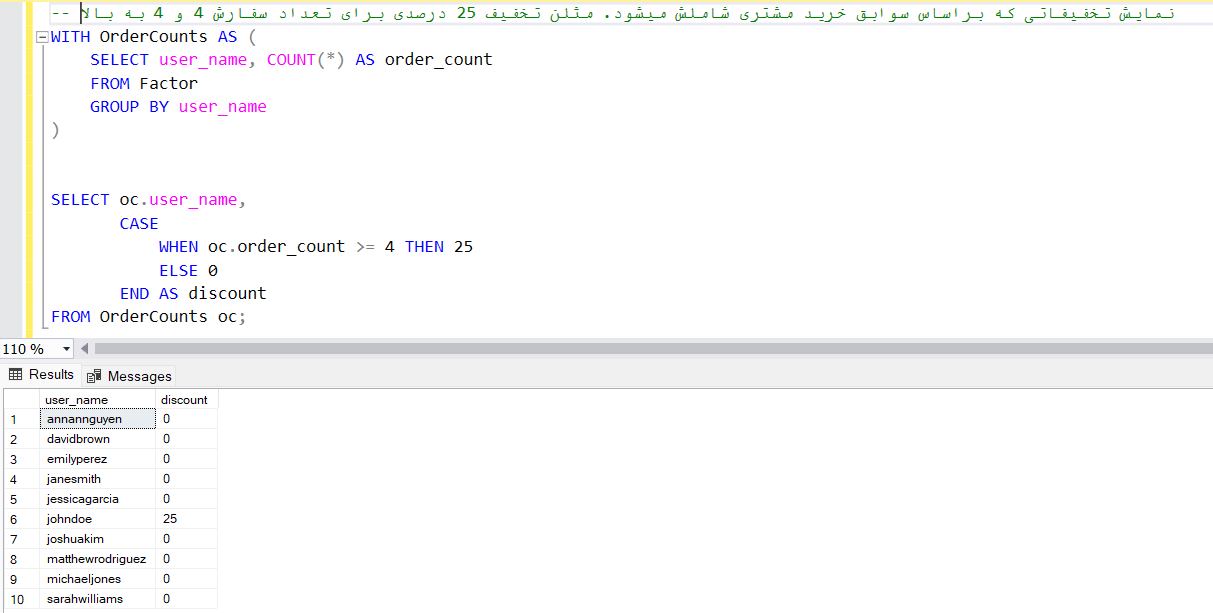
('2000-01-01', 13123456783, 'Test', 'User', 10003456789, 'testuser');

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

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

-- نمایش تخفیفاتی که براساس سوابق خرید مشتری شاملش میشود. مثلن 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;

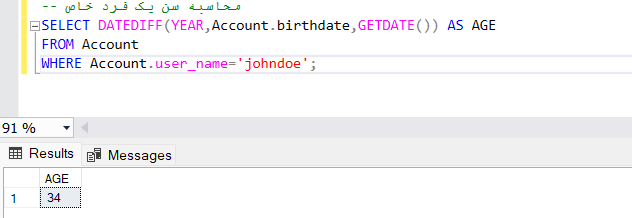
-------------------------------------------------------------------------------------------------------------------------------

-- محاسبه سن یک فرد خاص

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

FROM Account

WHERE Account.user\_name='johndoe';



-- 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

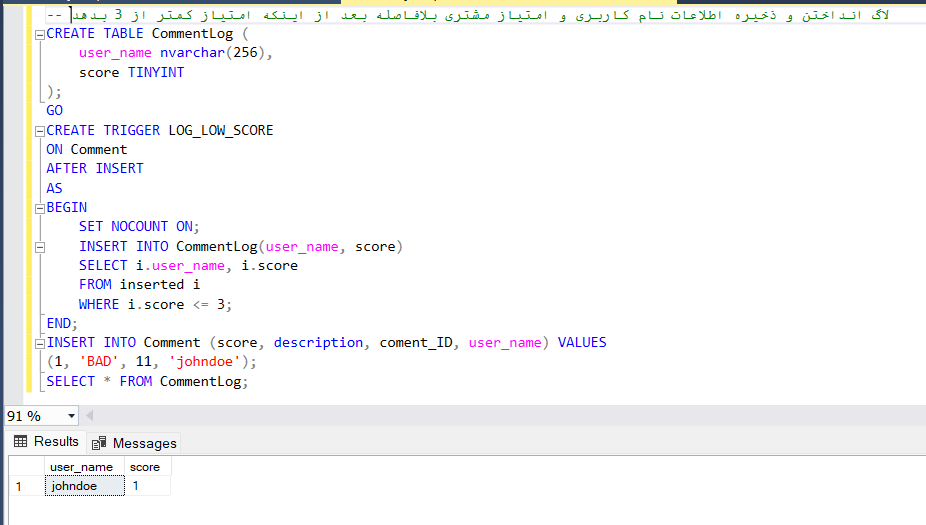
FROM inserted i

WHERE i.score <= 3;

END;

INSERT INTO Comment (score, description, coment\_ID, user\_name) VALUES

(1, 'BAD', 11, 'johndoe');

SELECT \* FROM CommentLog;