



DataBase | HW04
Fatemeh Nadi | 9636753

1.

برای آنکه برای هر `customer_id` یک `view` زده شود مدت زمان زیادی صرف خواهد شد
فرض میکنم که `view` را برای یک `customer` خاص نوشته میشود اما میتوان در حالت کلی `user` که لاگین کرده
را در نظر گرفت من برای یک مشتری با `customer_id = 29994` میسازم

```
create view customer_29994
```

سپس یک `user` برای اینکه با این وسیله کاربر شماره 29994 وارد شود میسازم و سپس با این `user` وارد میشوم
میخواهم یک فیلد از این `view` را تغییر دهم اما با ارور زیر مواجه شدم :

```
Query Editor
1 update customer_29994
2 set orderdate = current_date
```

Data Output Explain Messages Notifications

```
ERROR: permission denied for view customer_29994
SQL state: 42501
```

سپس دوباره با `postgres` وارد شدم و اجازه دسترسی به این `view` را به `user 29994` دادم

Query Editor

```
1
2 grant update on customer_29994 to "29994"
```

Data Output Explain **Messages** Notifications

GRANT

Query returned successfully in 101 msec.

✓ Query returned successfully in 101 msec.

حال با user 29994 وارد شده و رکوردی را تغییر میدهم و این تغییر اعمال میشود

Query Editor

```
1
2 update customer_29994
3 set orderdate = current_date
```

Data Output Explain **Messages** Notifications

UPDATE 0

Query returned successfully in 120 msec.

✓ Query returned successfully in 120 msec.

- دقت شود چون در مرحله نهایی متوجه شدم که باید ویو بسازم بدین شکل

```
create view customer_29994 as
)
select *
from sales.salesorderDetail as sod JOIN sales.salesorderheader as soh on
(soh.salesorderid = sod.salesorderid)
where customerid = 29994 and
current_date - soh.orderdate <= interval '1 day'
```

(

و در مرحله دادن مجوز فقط به فیلد های salesorderDetail از طریق زیر عمل میکنیم
grant update (orderqty , productid)on customer_29994 to "29994"

به جای (orderqty , productid) تمام ستون های جدول باید قرار بگیرد.
نتایج مانند قبل تکرار میشود.

b.

در ابتدا جدول production.InventoryDefaults را میسازیم (کد در فایل ضمیمه شده موجود است)
سپس مقادیری را به صورت پیش فرض وارد میکنیم : مثلا کالای شماره 991 با لوکیشن 6 در طبقه 12 و بین 2(این مقدار را از جدول قبلا پاک میکنم که موجود نباشد)

```
INSERT INTO production.inventorydefaults  
productid, locationid, shelf, bin(  
VALUES;(2 ,12 ,6 ,990)
```

حالا سطر دوم در تصویر زیر را به سطر اول اضافه کنم :

Query Editor








```
50
51 select *
52 from production.productinventory
53
```

Data Output

Explain

Messages

Notifications

	 productid [PK] integer	 locationid [PK] smallint	 shelf character varying (10)	 bin smallint	 quantity smallint	 rowguid uuid	 modifieddate timestamp without time zone	
1	1	1	1 A		1	408 47a24246-6...	2014-08-08 00:00:00	
2	1	6	6 B		5	324 d4544d7d-c...	2014-08-08 00:00:00	
3	1	50	1 A		5	353 bff7dc60-96...	2014-08-08 00:00:00	
4	2	1	6 B		2	427 f407c07a-ca...	2014-08-08 00:00:00	
5	2	6	50 A		1	318 ca1ff2f4-48f...	2014-08-08 00:00:00	
6	2	50	1 A		6	364 d38cfbee-63...	2014-08-08 00:00:00	
7	3	1	6 B		7	585 e18a519b-f...	2008-03-31 00:00:00	
8	3	6	50 A		9	443 3c860c96-1...	2008-03-31 00:00:00	

Query Editor

```

44
45
46
47 call transfer(1,6,1,324);
48
49

```

Data Output Explain Messages Notifications

CALL

Query returned successfully in 190 msec.

در نهایت جدول نهایی:

Query Editor

```

48
49
50
51
52 select *
53 from production.productinventory
54 where productid = 1
55

```

Data Output Explain Messages Notifications

	productid [PK] integer	locationid [PK] smallint	shelf character varying (10)	bin smallint	quantity smallint	rowguid uuid	modifieddate timestamp without time zone
1	1	50	A		5	353 bff7dc60-96...	2014-08-08 00:00:00
2	1	6	B		5	0 d4544d7d-c...	2014-08-08 00:00:00
3	1	1	A		2	732 47a24246-6...	2014-08-08 00:00:00

حال اگر کالا موجود نباشد نیز به صورت زیر عمل میکند

Query Editor

```

50
51
52 select *
53 from production.productinventory
54 where productid = 991
55
56
57

```

Data Output Explain Messages Notifications

	productid [PK] integer	locationid [PK] smallint	shelf character varying (10)	bin smallint	quantity smallint	rowguid uuid	modifieddate timestamp without time zone
--	---------------------------	-----------------------------	---------------------------------	-----------------	----------------------	-----------------	---

سپس

Query Editor

```
50  
51  
52  
53 call transfer(991,6,1,200);  
54  
55  
56
```

Data Output Explain Messages Notifications

CALL

Query returned successfully in 145 msec.

نتیجه:

Query Editor

```
55  
56  
57 select *  
58 from production.productinventory  
59 where productid = 991  
60  
61  
62
```

Data Output Explain Messages Notifications

	productid [PK] integer	locationid [PK] smallint	shelf character varying (10)	bin smallint	quantity smallint	rowguid uuid	modifieddate timestamp without time zone
1	991	1	12	2	200	89bfeeca-25...	2020-11-13 18:41:42.986993

.2

a.

Query Editor

```
1 create table RegistrationLog(  
2 StudentID      varchar(5) ,  
3 Semester character varying(6) COLLATE pg_catalog."default" NOT NULL,  
4 Year numeric(4,0) NOT NULL,  
5 Status         varchar(12) check (Status in ('Normal','NotReg','CheckUnder','CheckOver')),  
6 OverAttempts  int default 0,  
7 foreign key (StudentID) references Student (ID),  
8 primary key (StudentID,semester,year)  
9  
10 );
```

Data Output

Explain

Messages

Notifications

CREATE TABLE

Query returned successfully in 124 msec.

✓ Query returned successfully in 124 msec.

b.

✓ Query returned successfully in 130 msec.

Query Editor

```
1  create or replace procedure Register(  
2      temp_StudentID  varchar(5),  
3      incourse        varchar(8),  
4      temp_Semester   varchar(6),  
5      temp_sec        varchar(8),  
6      temp_Year       numeric(4,0),  
7      temp_Status     varchar(12),  
8      temp_OverAttempts integer  
9  )  
10 language plpgsql  
11 as $$
```

Data Output Explain **Messages** Notifications

CREATE PROCEDURE

Query returned successfully in 91 msec.

✓ Query returned successfully in 91 msec.

silber-large/postgres@PostgreSQL 12 ▾

Query Editor

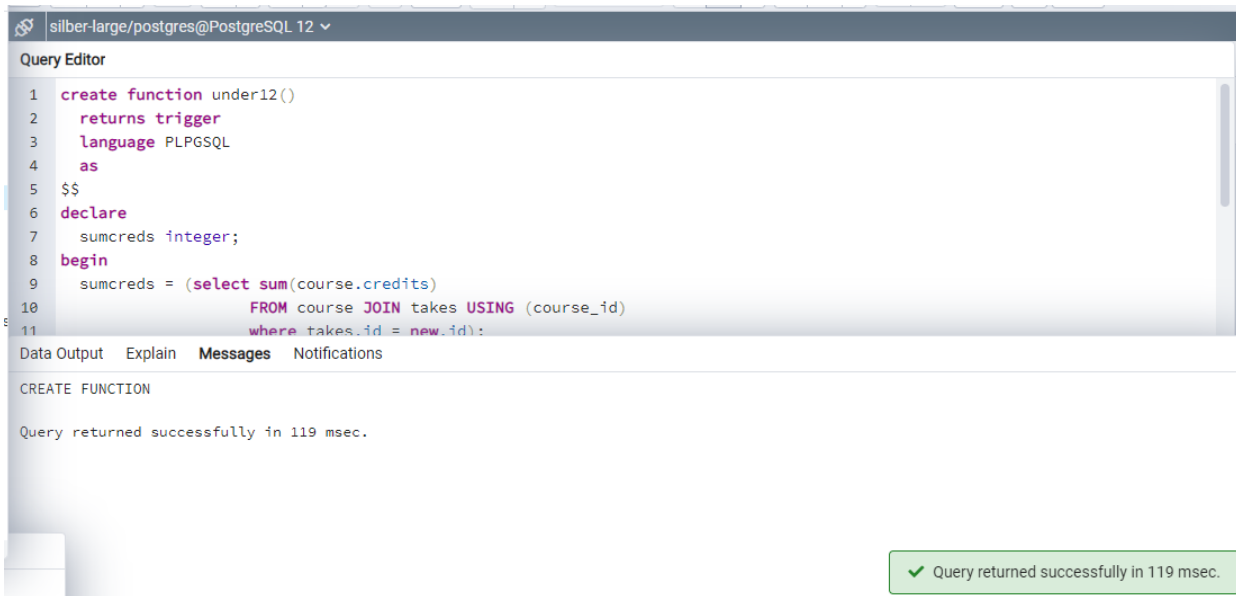
```
64  
65 call Register('38548','401','1','Fall','2003')  
66  
67  
68  
69  
70  
71  
72  
73  
74
```

Data Output Explain **Messages** Notifications

CALL

Query returned successfully in 122 msec.

C.



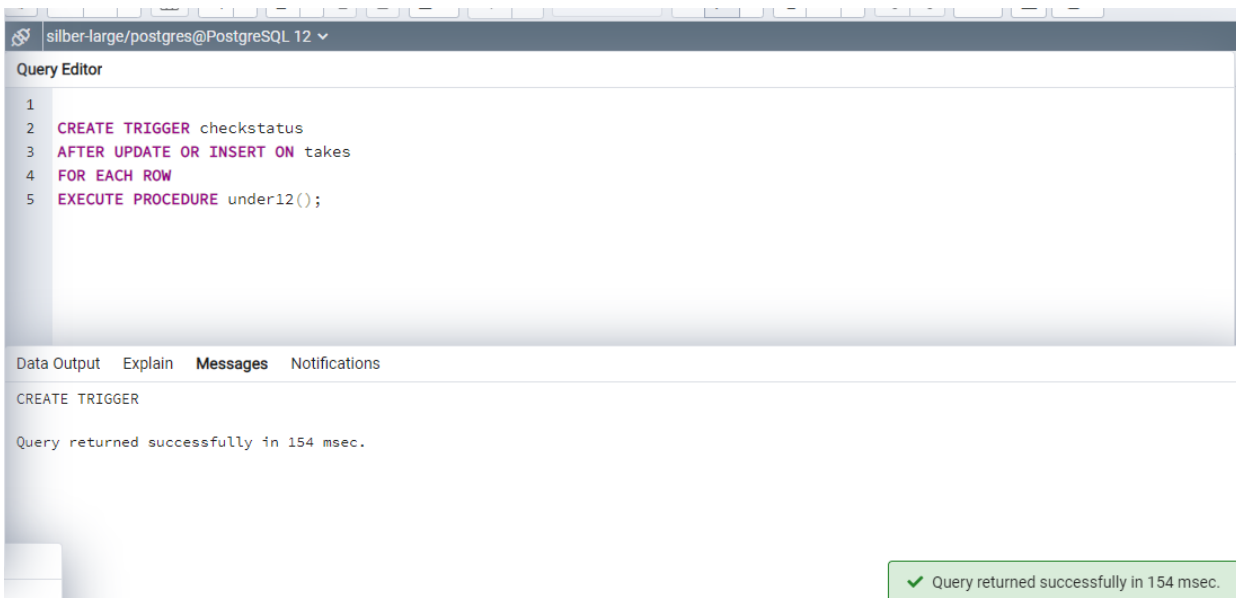
```
1 create function under12()
2   returns trigger
3   language PLPGSQL
4   as
5   $$
6   declare
7     sumcreds integer;
8   begin
9     sumcreds = (select sum(course.credits)
10                  FROM course JOIN takes USING (course_id)
11                  where takes.id = new.id);
```

Data Output Explain Messages Notifications

CREATE FUNCTION

Query returned successfully in 119 msec.

✓ Query returned successfully in 119 msec.



```
1
2 CREATE TRIGGER checkstatus
3 AFTER UPDATE OR INSERT ON takes
4 FOR EACH ROW
5 EXECUTE PROCEDURE under12();
```

Data Output Explain Messages Notifications

CREATE TRIGGER

Query returned successfully in 154 msec.

✓ Query returned successfully in 154 msec.

Test:

Query Editor

```
1 INSERT INTO public.takes(  
2   id, course_id, sec_id, semester, year, grade)  
3   VALUES ('38548', '974', '1', 'Fall', '2003', 'A');  
4
```

Data Output Explain Messages Notifications

INSERT 0 1

Query returned successfully in 124 msec.

✓ Query returned successfully in 124 msec.

Query Editor

```
17 select *  
18 from takes  
19 where id = '38548'
```

Data Output Explain Messages Notifications

	id [PK] character varying (5)	course_id [PK] character varying (8)	sec_id [PK] character varying (8)	semester [PK] character varying (6)	year [PK] numeric (4)	grade character varying (2)
1	38548	974	1	Fall	2003	A
2	38548	808	1	Fall	2003	A
3	38548	735	2	Spring	2010	C+
4	38548	631	1	Spring	2007	C-
5	38548	626	1	Fall	2006	A+
6	38548	486	1	Fall	2009	A+
7	38548	408	1	Spring	2007	A+
8	38548	400	1	Spring	2007	A+
9	38548	319	1	Spring	2003	A
10	38548	313	1	Fall	2010	C+
11	38548	200	2	Fall	2002	C
	38548	105	2	Fall	2002	B+
	38548	105	1	Fall	2009	C

پس از اضافه کردن چند کورس به این دانشجو نتیجه جدول RegistrationLog

silber-large/postgres@PostgreSQL 12

Query Editor

```
1 select *
2 from RegistrationLog
```

22:32:19

Data Output

Explain

Messages

Notifications

	studentid [PK] character varying (5)	semester [PK] character varying (6)	year [PK] numeric (4)	status character varying (12)	overattempts integer
1	38548	1	2003	CheckOver	1

✓ Successfully run. Total query runtime: 192 msec. 1 rows affected.

.3

a.

Query Editor

```
1 Create table category_rating(  
2 category_name character varying(25) COLLATE pg_catalog."default" NOT NULL,  
3 avg_rental_rate numeric(4,2) NOT NULL DEFAULT 4.99 ,  
4 avg_len smallint  
5 );  
6
```

Data Output Explain **Messages** Notifications

CREATE TABLE

Query returned successfully in 142 msec.

✓ Query returned successfully in 142 msec.

dvdrental/postgres@PostgreSQL 12 ▾

Query Editor

```
1 INSERT INTO category_rating  
2 select category.name , avg(film.rental_rate) , avg(length)  
3 from film  
4 JOIN film_category ON (film.film_id = film_category.film_id )  
5 JOIN category ON (film_category.category_id = category.category_id)  
6  
7 group by category.name  
8
```

Data Output Explain **Messages** Notifications

INSERT 0 16

Query returned successfully in 168 msec.

✓ Query returned successfully in 168 msec.

Query Editor

```

1 select *
2 from category_rating
3

```

Data Output Explain Messages Notifications

	category_name character varying (25)	avg_rental_rate numeric (4,2)	avg_len smallint
1	Family	2.76	115
2	Games	3.25	128
3	Animation	2.81	111
4	Classics	2.74	112
5	Documentary	2.67	109
6	New	3.12	111
7	Sports	3.13	128
8	Children	2.89	110
9	Music	2.95	114
10	Travel	3.24	113
11	Foreign	3.10	122
	Drama	3.02	121
	Horror	3.03	112

Query Editor

```

1 create or replace procedure dropping()
2 language plpgsql
3 as $$
4 declare
5     tri_len integer;
6 begin
7
8     tri_len = (
9         select avg(average_rental_rate)
10        from (

```

Data Output Explain Messages Notifications

CREATE PROCEDURE

Query returned successfully in 391 msec.

✓ Query returned successfully in 391 msec.

b.

Query Editor

```
1 create function Award ( tdate timestamp )
2 returns integer
3 language plpgsql
4 as
5 $$
6 declare
7     s integer;
8 begin
9     s = (
10         select count(*)
```

Data Output

Explain

Messages

Notifications

CREATE FUNCTION

Query returned successfully in 141 msec.

✓ Query returned successfully in 141 msec.

C.

Query Editor

```
1 create or replace function over7()
2   returns trigger
3   language PLPGSQL
4   as
5   $$
6   declare
7     storeID integer ;
8     invID integer ;
9     endrentalID integer;
10  begin
```

Data Output Explain Messages Notifications

CREATE FUNCTION

Query returned successfully in 152 msec.

✓ Query returned successfully in 152 msec.

Query Editor

```
53 create trigger gift
54   AFTER UPDATE or INSERT ON payment
55   FOR EACH ROW
56   EXECUTE PROCEDURE over7();
57
58
59
60
61
62
```

Data Output Explain Messages Notifications

CREATE TRIGGER

Query returned successfully in 143 msec.

✓ Query returned successfully in 143 msec.

d.

Query Editor

```
1 create view V1_inv as
2 (
3     select *
4     from inventory
5 );
6
```

Data Output Explain Messages

CREATE VIEW

Query returned successfully in 148 msec.

✓ Query returned successfully in 148 msec.

Query Editor

```
6
7 create role "staff_order";
8
9
10
11
```

Data Output Explain Messages

CREATE ROLE

Query returned successfully in 128 msec.

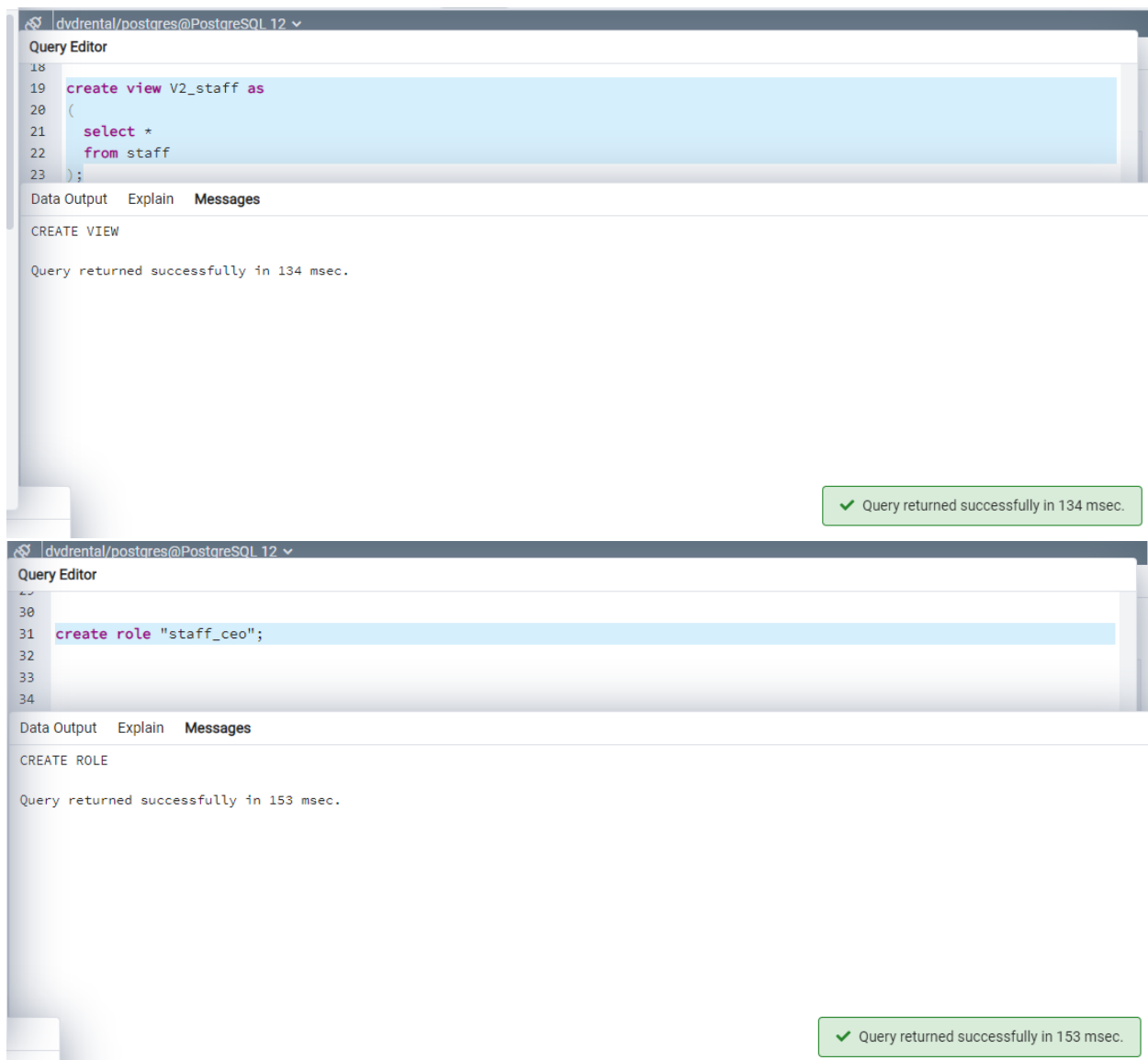
Query Editor

```
11
12 grant SELECT on V1_inv to "staff_order";
13
14
15
16
```

Data Output Explain Messages

GRANT

Query returned successfully in 117 msec.



e.

یک user بسازیم و به آن نقش staff_ceo بدهیم
سپس با استفاده از یک view نقش را به view بدهیم که فقط بتواند به فروشگاه ود دسترسی داشته باشد مانند سوال یک