

پروژه نهایی درس پایگاه داده فروشگاه اینترنتی

در این پروژه برای 2 نوع مشتری در نظر گرفته شده است:

1- مشتری که از قبل در سامانه ثبت نام کرده است

2- مشتری که جدید به سامانه وارد می شود.

برای هر مشتری که از قبل در سامانه ثبت نام کرده است اطلاعاتی از قبیل ID و name و password و family name و Email و credit و ... در جدول customer ذخیره می شود. هر مشتری که جدید وارد سیستم می شود اطلاعات مشابه مشتری قبلی به جز password و شماره تلفن را دارا است. (می توان شماره مشتری جدید را هم به اطلاعات جدول تلفن وارد کرد در صورت لزوم).

هر مشتری جدید که وارد سیستم می شود با استفاده از trigger اطلاعات آن به جدول customer با مقدار NULL password و credit برابر با 1- وارد جدول customer نیز می شود.

برای جلوگیری از تداخل ID ها ID مشتریان سابق از 1000 شروع می شود ولی مشتریان جدید از 2000

Trigger به صورت روبرو است:

Edit trigger

Details

Trigger name

before_insert_newcustomer

Table

newcustomer

Time

AFTER

Event

INSERT

Definition

```
1 BEGIN
2 INSERT INTO customer(ID, `password`, Email, Name ,
  Familyname, PostalCode, Gender, credit) VALUES (NEW.ID, "",
  NEW.Email, NEW.Name, NEW.Familyname, NEW.Postalcode,
  NEW.Gender,-1);
3 INSERT INTO customeraddress(customerid, Address) VALUES
  (NEW.ID, NEW.Address);
4 END
```

Definer

root@localhost

Go

Close

مشتریان می توانند با استفاده از تابع chargeaccount که به صورت زیر نوشته شده است حساب خود را شارژ کنند.

Edit routine

Details

Routine name
chargeaccount

Type
FUNCTION

	Name	Type	Length/Values	Options
Parameters	idc	INT	6	Drop
	price	FLOAT	7,2	Drop
Add parameter				

Return type
INT

Return length/values
2

Return options

Definition

```

1 BEGIN
2   DECLARE profit INT(2);
3   SET profit = 1;
4   UPDATE customer
5     set credit =credit + price
6     WHERE customer.id = idc;
7   RETURN profit;
8 END

```

Go Close

هر مشتری می تواند چندین شماره و آدرس داشته باشد به همین دلیل یک جدول برای ذخیره آدرس های مختلف و یک جدول برای ذخیره شماره های مختلف یک مشتری در نظر گرفته شده است. کلید اصلی جدول آدرس زوج(customerid,address) و کلید اصلی جدول شماره تلفن شماره تلفن است. در این 2 جدول شناسه مشتری کلید خارجی به جدول مشتری است.

سفارشات هر فروشگاه در یک جدول به اسم order ذخیره می شوند.

برای درست وارد کردن و چک کردن شرایطی از جمله موجودی کالا در فروشگاه و یک جدول به اسم temporary در نظر گرفته شده است که سفارشات مشتری ابتدا وارد این جدول می شوند و سپس اقدامات مناسبات برای قبول یا رد این درخواست در یک trigger انجام می شود(مانند چک کردن موجود بودن بیک و...).

Trigger این جدول به صورت مقابل است.

Edit trigger

Details

Trigger name

before_insert

Table

temporary

Time

BEFORE

Event

INSERT

Definition

```
1 BEGIN
2 DECLARE shopopent Time;
3 DECLARE shopcloset Time;
4 DECLARE crows INT;
5 DECLARE creq INT;
6 DECLARE cdelivery INT;
7 DECLARE stat char(100);
8 DECLARE stat2 char(100);
9 DECLARE paystat char(100);
10
11 SELECT starttime INTO shopopent
12 from shop
13 WHERE shop.id = NEW.shopID;
14
15 SELECT finishtime INTO shopcloset
16 from shop
```

Definer

root@localhost

Go

Close

در شکل روبرو که قسمتی از trigger این جدول است باز بون فروشگاه در ساعت درخواست چک شده است.

Edit trigger

Details

Trigger name

before_insert

Table

temporary

Time

BEFORE

Event

INSERT

Definition

15 SELECT finishtime INTO shopcloset

16 from shop

17 WHERE shop.ID = NEW.shopID;

18

19 IF(shopopent < NEW.time AND NEW.time < shopcloset)

20 THEN

21 SELECT count(*) INTO crows

22 FROM orderproduct

23 WHERE orderproduct.orderID = NEW.orderID;

24

25 SELECT count(*) INTO creg

26 FROM orderproduct,product

27 WHERE orderproduct.orderID = NEW.orderID AND

28 orderproduct.productID = product.productID AND

product.ShopID = NEW.ShopID AND

Definer

root@localhost

Go

Close

در قسمت بعد که در شکل مقابل نشان داده شده است:

```

5 SELECT count(*) INTO creq
6 FROM orderproduct,product
7 WHERE orderproduct.orderID = NEW.orderID AND
8 orderproduct.productID = product.productID AND product.ShopID = NEW.ShopID AND
9 orderproduct.num <= product.number;
10
11 SET stat = "denied";
12 IF(creq = crows)
13 THEN
14 SELECT count(*) INTO cdelivery
15 FROM shoppdelivry
16 WHERE shoppdelivry.ShopID = NEW.shopID AND
17 shoppdelivry.status = "free";
18 IF(cdelivery > 0)
19 THEN
20 SET stat = "registered";
21 SET stat2 = "delivered";
22 END IF;
23 END IF;
24 END IF;
25
26 IF(NEW.time < shopopent OR NEW.time > shopcloset)
27 THEN
28 SET stat = "denied" ;
29 END IF;
30
31 SET paystat = NEW.pay;
32 IF(NEW.orderID > 20000)
33 THEN
34 SET paystat = "bank";
35 END IF;
36 INSERT INTO orders(orderID, customerid,shopID,status,pay,
37 Date, time, Address)
38 VALUES (NEW.orderID, NEW.customerid, NEW.shopID, stat, paystat, NEW.Date, NEW.time, NEW.address);
39 IF(stat2 = "delivered")
40 THEN
41 UPDATE orders
42 SET status= stat2
43 WHERE orders.orderID= NEW.orderID;
44 CALL shopnewnum(NEW.orderID, NEW.shopID);
45 END IF;
46 END

```

موجودی کالاهای سفارش داده شده توسط مشتری در فروشگاه چک شده است و در صورت وجود کالا به مقدار کافی وضعیت به registered تغییر پیدا می کند و در غیر اینصورت به denied تبدیل می شود. در صورتی که وضعیت registered شود وجود یا عدم وجود پیک بررسی می شود و در صورت عدم وجود پیک به denied تغییر پیدا می کند. سپس در صورتی که مشتری، مشتری باشد که جدید وارد سامانه شده باشد نوع پرداخت آن به صورت bank مشخص می شود و سپس tuple مورد نظر وارد جدول order می شود. در مرحله بعد اگر وضعیت سفارش delivered باشد وضعیت جدول order برای سفارش مورد نظر تاره سازی می شود و وضعیت سفارش به delivered تغییر پیدا می کند و سپس procedure به اسم shopnewnum صدا زده می شود که مقادیر سفارش داده شده را از مقادیر کالاهای موجود در فروشگاه کم کند.

Procedure shopnewnum() به صورت مقابل نوشته شده است.

```

1 BEGIN
2 DECLARE n INT DEFAULT 0;
3 DECLARE i INT DEFAULT 0;
4 DECLARE numc INT DEFAULT 0;
5 DECLARE idp INT(10);
6
7 SELECT COUNT(*) FROM orderproduct INTO n;
8 SET i=0;
9 WHILE i<n DO
10 SET idp = -1;
11 SELECT orderproduct.productID
12 from orderproduct
13 WHERE orderproduct.ID = i AND orderID = idorder INTO idp;
14
15 SELECT orderproduct.num
16 from orderproduct
17 WHERE orderproduct.ID = i AND orderID = idorder INTO numc;
18
19 UPDATE product
20 SET number = number - numc
21 WHERE product.productID = idp AND product.ShopID = idshop;
22
23 SET i = i + 1;
24 END WHILE;
25 END

```

که یک حلقه for بر روی جدول order product می زند و تعداد کالاهای سفارش داده شده با order id مشخص را از کالاهای فروشگاه کم می کند.

در جدول order product کالاهای سفارش داده شده با یک ordered مشخص و تعداد آن ها ذخیره شده است.

بعد از اینکه tuple های مورد نظر وارد جدول order شده اند. توسط یک trigger به صورت زیر در صورتی که وضعیت آن ها delivered باشد یک پیک برایشان در نظر گرفته می شود و وضعیت پیک مورد نظر از free به onway تبدیل می شود.

```

1 BEGIN
2 DECLARE Iddelivery INT(20);
3 IF(NEW.status = "delivered")
4 THEN
5
6 SELECT shopdelivery.deliveryID INTO Iddelivery
7 FROM shopdelivery
8 WHERE shopdelivery.ShopID = NEW.shopID AND
9 shopdelivery.`status` = "free"
10 LIMIT 1;
11
12
13 UPDATE shopdelivery
14 SET `status` = "on-way"
15 WHERE shopdelivery.deliveryID = Iddelivery;
16
17 UPDATE shopdelivery
18 SET shopdelivery.orderID = NEW.orderID
19 WHERE shopdelivery.deliveryID = Iddelivery;
20
21 END IF;
22 END

```

سپس پیک بسته را به مشتری می‌رساند و تابع complete order را صدا می‌زند.

تابع completeorder به صورت زیر است

در این تابع ابتدا قیمت تمام کالاهای سفارش داده شده توسط مشتری حساب می‌شود و در صورتی که نوع پرداخت مشتری از نوع credit باشد این مقدار از حساب مشتری کم می‌شود. سپس 05. مبلغ به credit پیک اضافه می‌شود و وضعیت سفارش در جدول order به complete تغییر می‌کند.


```

SET pays = "";
SET pricetotal = 0;
SET profit = 1;
Select sum( p*n) INTO      pricetotal
from(
SELECT product.price AS p,orderproduct.num AS n
FROM `orders` , orderproduct , product
WHERE `orders`.`orderID` = idorder AND
orderproduct.orderid = idorder AND
orderproduct.productid = product.productID)AS W;

SELECT pay INTO pays
FROM orders
WHERE orders.orderID = idorder;

IF(pays = "credit")
THEN
UPDATE customer
SET credit = credit - pricetotal
WHERE customer.ID = idcustomer;
END IF;

UPDATE shopdelivery
SET credit = credit + .05*pricetotal
WHERE shopdelivery.orderID = idorder;

UPDATE shopdelivery
SET `status` = "free"
WHERE shopdelivery.orderID = idorder;

UPDATE `orders`
SET `status` = "completed"
WHERE `orders`.`orderID` = idorder;

DELETE FROM `temporary`
WHERE `temporary`.`orderID` = idorder;

RETURN profit;
END

```

کویری ها:

در این قسمت خروجی کویری های خواسته شده نمایش داده می شود . چون نسخه mysql از عمل with استفاده نمی کرد. در بعضی موارد ابتدا یک view مناسب ساخته شده است با توجه به جداول موجود و سپس داده های مورد نظر نمایش داده شده است.

1- کالاهایی که از فروشگاه سفارش داده شده اند و به دلیل عدم موجودی کافی رد شده اند نیز جز تعداد کالاهای سفارش داده شده برای محاسبه کالاهای پرفروش در نظر گرفته شده اند. فضای مجموعه کوچکتر فرض شده است و برای هر فروشگاه پرفروش ترین کالای آن نمایش داده شده است.

برای این قسمت ابتدا یک View به صورت زیر

```
CREATE view w as
(
SELECT orderproduct.productID , orders.shopID, sum(orderproduct.num)
FROM orderproduct,orders
WHERE orderproduct.orderID = orders.orderID
GROUP BY orderproduct.productID , orders.shopID);
```

که شامل تعداد کالاهای مختلفی که از فروشگاه های مختلف خریداری شده است می شود.

سپس همان طور که در زیر مشخص است این جدول در خودش ضرب شده است و تعداد کالاهایی که از هر کالا کمتر است محاسبه شده است و سپس کالاهایی که کمتر از یک کالا از آن ها بیشترین تعداد خرید را داشته است به عنوان خروجی در نظر گرفته شده است. (برای 5 کالا کافی است 1 را به 5 تبدیل کنیم)

```
SELECT w.productID , w.shopID
FROM w LEFT JOIN w as t
ON w.shopID = t.shopID AND w.snum < t.snum
GROUP BY w.productID , w.shopID
HAVING COUNT(t.snum) < 1
```

خروجی کویری در محیط sql به صورت زیر است:

Showing rows 0 - 1 (2 total, Query took 0.0120 seconds.)

```
SELECT w.productID , w.shopID FROM w LEFT JOIN w as t ON w.shopID = t.shopID AND w.snum < t.snum GROUP BY w.productID , w.shopID HAVING COUNT(t.snum) < 1
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

+ Options

productID	shopID
40003	50001
40004	50002

☐ Show all | Number of rows: 25 | Filter rows:

2- کویری این قسمت به صورت روبرو است

```
SECOND query
SELECT customertelephone.customerid, customertelephone.Telephone
FROM orders JOIN customertelephone
WHERE customertelephone.customerid = orders.customerid AND orders.status = "denied"
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0069 seconds.)

```
SELECT customertelephone.customerid, customertelephone.Telephone FROM orders JOIN customertelephone WHERE customertelephone.customerid = orders.customerid AND orders.status = "denied"
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: Filter rows:

Options

customerid	Telephone
10002	97653

☐ Show all | Number of rows: Filter rows:

3- برای این قسمت ابتدا یک view تشکیل شده است که مبلغ فروش را برای هر مشتری محاسبه میکند.

```
CREATE view moncust AS
SELECT orders.customerid, sum(orderproduct.num*product.price) as scust
FROM orderproduct, orders, product
WHERE `orders`.`orderid` = orderproduct.orderid AND
      orderproduct.productid = product.productid
GROUP BY orders.customerid
```

سپس با توجه به اینکه ID مشتریان قبلی کمتر از 2000 و ID مشتریان جدید بیشتر از 2000 در نظر گرفته شده است تفاوت مبلغ خرید به صورت زیر محاسبه می شود:

```
SELECT (SELECT AVG(moncust.scust)
FROM moncust, customer
WHERE moncust.customerid = customer.ID AND customer.ID > 20000) - (SELECT AVG(moncust.scust)
FROM moncust, customer
WHERE moncust.customerid = customer.ID AND customer.ID < 20000)
```

خروجی کویری در محیط sql به صورت زیر است:

Showing rows 0 - 0 (1 total, Query took 0.0131 seconds.)

```
SELECT (SELECT AVG(moncust.scust) FROM moncust,customer WHERE moncust.customerid = customer.ID AND customer.ID > 20000) - (SELECT AVG(moncust.scust) FROM moncust,customer WHERE moncust.customerid = customer.ID AND customer.ID < 20000)
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

+ Options

```
(SELECT AVG(moncust.scust) FROM moncust,customer WHERE moncust.customerid = customer.ID AND customer.ID > 20000) - (SELECT AVG(moncust.scust) FROM moncust,customer WHERE moncust.customerid = customer.ID AND customer.ID < 20000)
```

-6000.0000

☐ Show all | Number of rows: 25 | Filter rows:

4-با توجه با اینکه پیک ها با مبلغ credit صفر وارد سیستم می شوند و پیک از هر ارسال بسته 05. قیمت را برای خود بر میدارد پیکی که بیشترین credit را دارد مجموع بسته های حمل شده توسط آن بیشترین است که این مساله با استفاده از کویری روبرو مشخص شده است

```
.....
FORTH Query
SELECT *
FROM shopdelivery
WHERE shopdelivery.credit = (SELECT MAX(shopdelivery.credit)
                             FROM shopdelivery
                             )
.....
```

خروجی در محیط mysql به صورت زیر است:

Showing rows 0 - 0 (1 total, Query took 0.0080 seconds.)

```
SELECT * FROM shopdelivery WHERE shopdelivery.credit = (SELECT MAX(shopdelivery.credit) FROM shopdelivery )
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

+ Options

	ShopID	deliveryID	orderID	name	familyname	phone	address	status	credit
<input type="checkbox"/> Edit Copy Delete	50001	30001	60004	ehsan	ebrahimi	93176442		free	1050

[↑](#) ☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

5-کویری این قسمت به صورت زیر است:

```
SELECT *
FROM shop
WHERE (shop.FinishTime - shop.StartTime) =
(SELECT MAX(shop.FinishTime - shop.StartTime)
FROM shop)
```

خروجی کویری در محیط mysql به صورت زیر است:

Showing rows 0 - 0 (1 total, Query took 0.0100 seconds.)

`SELECT * FROM shop WHERE (shop.FinishTime - shop.StartTime) = (SELECT MAX(shop.FinishTime - shop.StartTime) FROM shop)`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	ID	Name	City	Address	Phone	Manager	StartTime	FinishTime
<input type="checkbox"/> Edit Copy Delete	50001	refah	shiraz	zargari	3825615	anna	08:00:00	20:00:00

↑ ☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table

نمودار entity relationship به صورت یه فایل png در پوشه قرار دار. مقادیر جدول ها نیز در یک فایل pdf در پوشه قرار دارد

کوثر بهنیا

9331710