

به نام خدا

گزارش پروژه پایانی پایگاه داده

اعضای گروه :

مهدیس صفری – فاطمه غلام زاده

شماره دانشجویی اعضای گروه :

۹۵۳۱۰۶۰-۹۵۳۱۰۵۱

مراحل انجام پروژه :

ابتدا موجودیت ها و جداول مورد نیاز را مشخص کردیم. سپس جداول را در پایگاه داده add کردیم. در این مرحله کلیدهای اصلی و کلیدهای خارجی هم مشخص می شوند.

بعد از ایجاد پایگاه داده به جداول موجود در آن data اضافه می کنیم.

۱۵ موجودیت قوی برای این پایگاه داده شناخته شدند که عبارتند از :

1-customer

2-admin

3-delivery

4-agent

5-store

6-product

7-customer_pn

8-customer_id_pc

9-orders

10-ordered_products

11-store's_deliveries

12-store's_agent

13-store's_admin

14-store's_product

15-charge

برای برآورده شدن شروط ذکر شده در صورت پروژه ، از trigger استفاده کردیم و تریگر هایی را به جداول مختلف اضافه نمودیم.

نمونه هایی از تریگر ها که روی جدول customer زده شده :

BEGIN

CREATE TEMPORARY TABLE caddress

select address

from customer_id_pc as sa

where id = new.customer_id ;

CREATE TEMPORARY TABLE admins

select sa.admin_id as aid

from store's_admins as sa

where sa.store_id = new.store_id ;

set @admin :=

(

select aid

from admins

ORDER BY RAND()

LIMIT 1

```
);
```

```
set
```

```
@ctime := CURRENT_TIME();
```

```
set
```

```
@t_open := (select t_open  
from store  
where store_id = new.store_id);
```

```
set
```

```
@t_close := (select t_close  
from store  
where store_id = new.store_id);
```

```
set
```

```
@reg := (select registered  
from customer  
where id = new.customer_id);
```

```

IF @ctime > @t_close or @ctime < @t_open THEN

    insert into failed_log values(new.customer_id,'store was closed',@admin,now());

    SET NEW.statuuss = 'failed';

end if;

if @reg = 'no' THEN

    SET NEW.payment = 'portal bank';

    #####age register has addressso check kone

ELSEIF not exists(select address

                    from caddress

                    where address = new.address) THEN

    insert into failed_log values(new.customer_id,'wrong address',@admin,now());

    SET NEW.statuuss = 'failed';

END IF;

END

```

برای هندل کردن مشتریان جدید یک ستون به نام registered به جدول customer اضافه کردیم. اگر این ستون 'yes' باشد نشان دهنده مشتری ثبت نام شده است و اگر 'no' باشد نشان دهنده مشتری جدید است. پیش فرض ما بر آن بوده است که آیدی هر مشتری شماره ملی آن مشتری است در نتیجه منحصر به فرد است. برای مشتری های جدید پسورد لازم نیست و به صورت دیفالت برابر '----' قرار داده می شود. هم چنین اعتبار حساب مشتری جدید به طور پیش فرض برابر با ۰ قرار داده می شود و پرداخت های مشتری جدید فقط با پرتال بانک انجام می شود .

برای شارژ کردن حساب یک موجودیت (جدول) به نام charge اضافه کردیم.

نمونه هایی از شروطی که در تریگر ها چک می شود :

- ۱- وقتی یک سفارش اضافه می شود چک میکنیم که آن سفارش در بازه ی زمانی که فروشگاه باز است ثبت شود وگرنه سفارش با شکست مواجه می شود.
- ۲- هنگام اضافه شدن سفارش چک می کنیم که محصول سفارش داده شده جزو محصول های آن فروشگاه باشد و اینکه آن محصول در آن فروشگاه موجودی داشته باشد وگرنه سفارش fail می شود.
- ۳- وقتی سفارشی داده می شود باید چک کنیم که فروشگاه مربوطه حداقل یک پیک آزاد داشته باشد تا بتواند محصول را برساند در غیر اینصورت سفارش fail می شود.
- ۴- وقتی مشتری ثابت سفارشی میدهد چک می کنیم که موجودی حسابش به قدر کافی باشد.
- ۵-

Log گرفتن :

هم چنین تمامی تغییراتی که در جداول اتفاق می افتد را در جدول های دیگری همراه با زمان آن تغییر log میگیریم. از جمله logهایی که میگیریم می توان به این موارد اشاره کرد :

جدول fail_log : شامل علت fail شدن سفارش ها به همراه تاریخ و ساعت

جدول delivery_log : شامل credit های قدیم و جدید پیک

جدول customer_log : شامل credit های قدیم و جدید مشتری

جدول order_log : شامل وضعیت های قدیم و جدید سفارش

جدول store's_product_log : شامل مقدار قدیم و جدید تعداد محصول

و

کوئری ها

کوئری اول :

```
SELECT store_id,product_id,order_rate FROM `store's_products` as sp WHERE order_rate =  
( SELECT Max(order_rate) FROM `store's_products` as f WHERE f.store_id = sp.store_id )  
  
UNION  
  
SELECT store_id,product_id,order_rate  
  
FROM (  
  
SELECT store_id,product_id,order_rate,  
  
@rn := IF(@eID = store_id, @rn + 1,  
  
IF(@eID := store_id, 1, 1)) AS rn  
  
FROM `store's_products`  
  
CROSS JOIN (SELECT @rn := 0, @eID := 0) AS vars  
  
ORDER BY store_id, order_rate DESC) AS t  
  
WHERE t.rn = 2
```

✓ Showing rows 0 - 5 (6 total, Query took 0.0128 seconds.)

```
SELECT store_id,product_id,order_rate FROM `store's_products` as sp WHERE order_rate = ( SELECT Max(order_rate) FROM `store's_products` as f WHERE f.store_id =  
sp.store_id ) UNION SELECT store_id,product_id,order_rate FROM ( SELECT store_id,product_id,order_rate, @rn := IF(@eID = store_id, @rn + 1, IF(@eID := store_id, 1, 1))  
AS rn FROM `store's_products` CROSS JOIN (SELECT @rn := 0, @eID := 0) AS vars ORDER BY store_id, order_rate DESC) AS t WHERE t.rn = 2
```

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

☐ Show all | Number of rows: 25 ▼ Filter rows: Sort by key: None ▼

+ Options

store_id	product_id	order_rate
00001	00001	3
00002	00008	2
00003	00011	2
00001	00006	1
00002	00005	1
00003	00003	1

کوٹری دوم :

```
SELECT orders.customer_id,phone_number FROM orders,`customer_pn`as  
customer_phone
```

```
WHERE customer_phone.id=orders.customer_id AND status='failed' ;
```

✓ Showing rows 0 - 1 (2 total, Query took 0.0058 seconds.)

```
SELECT orders.customer_id,phone_number FROM orders,`customer_pn`as customer_phone WHERE customer_phone.id=orders.customer_id AND status='failed'
```

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

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

+ Options

customer_id	phone_number
00001	09107776655
00001	09108877877

کوٹری سوم :

```
SELECT (R1.avg1-R2.avg2) as difference
```

```
FROM (SELECT sum(price-off)/(SELECT COUNT(customer_id) from orders,customer WHERE  
customer.id = orders.customer_id and customer.registered = 'yes' and orders.status <>  
'failed') as avg1
```

```
FROM orders,product,customer
```

```
WHERE customer.registered = 'yes' and customer.id=orders.customer_id AND  
orders.store_id = product.store_id)as R1 ,(SELECT sum(price-off)/(SELECT  
COUNT(customer_id) from orders,customer WHERE customer.id = orders.customer_id and  
customer.registered = 'no' and orders.status <> 'failed') as avg2
```

```
FROM orders,product,customer
```


WHERE customer.registered = 'no' and customer.id=orders.customer_id AND
orders.store_id = product.store_id)as R2

✓ Showing rows 0 - 0 (1 total, Query took 0.0187 seconds.)

```
SELECT (R1.avg1-R2.avg2) as difference FROM (SELECT sum(price-off)/(SELECT COUNT(customer_id) from orders,customer WHERE customer.id = orders.customer_id and customer.registered = 'yes' and orders.statuss <> 'failed') as avg1 FROM orders,product,customer WHERE customer.registered = 'yes' and customer.id=orders.customer_id AND orders.store_id = product.store_id)as R1 ,(SELECT sum(price-off)/(SELECT COUNT(customer_id) from orders,customer WHERE customer.id = orders.customer_id and customer.registered = 'no' and orders.statuss <> 'failed') as avg2 FROM orders,product,customer WHERE customer.registered = 'no' and customer.id=orders.customer_id AND orders.store_id = product.store_id )as R2
```

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

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

+ Options
difference
1526.6667

کوٹری چہارم:

SELECT * FROM delivery WHERE credit/order_rate = (SELECT MAX(credit/order_rate)
FROM delivery

✓ Showing rows 0 - 0 (1 total, Query took 0.0051 seconds.)

```
SELECT * FROM delivery WHERE credit/order_rate = (SELECT MAX(credit/order_rate) FROM delivery )
```

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

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

+ Options

	delivery_id	name	fname	phone_number	statuss	order_id	current_order_cost	credit	order_rate
<input type="checkbox"/> Edit Copy Delete	00004	milad	amini	09138887543	sending	00003	2530	126.5	1

↑ ☐ Check all With selected: Edit Copy Delete Export

کوٹری پنجم:

SELECT store_id,store_name FROM store WHERE (t_close - t_open) =
(SELECT Max(t_close - t_open)

;(FROM store

Showing rows 0 - 0 (1 total, Query took 0.0061 seconds.)

SELECT store_id,store_name FROM store WHERE (t_close - t_open) = (SELECT Max(t_close - t_open) FROM store)

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

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

Options

store_idstore_name

Edit

Copy

Delete

00003zara

Check all

With selected:

Edit

Copy

Delete

Export

نمودار ER در صفحه بعدی آورده شده است :

