

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



HỆ CƠ SỞ DỮ LIỆU (CO2013)

Báo cáo bài tập lớn 2 - L01 - HK202

“Hiện thực bảng dữ liệu về chuỗi nhà hàng kinh doanh thực phẩm”

GV ra đề và HD: Trương Quỳnh Chi
Lớp: L01
Nhóm: TVDQ
Sinh viên: Trần Long Vĩ - 1814804
Nguyễn Thế Duy - 1912912
Nguyễn Lê Minh Quân - 1914830
Nguyễn Thành Nhân - 1937030
Nguyễn Lương Trọng - 2120073

TP. HỒ CHÍ MINH, THÁNG 11/2021



Mục lục

1	Phần chung	2
1.1	Các câu lệnh tạo bảng và ràng buộc	2
1.2	Các câu lệnh tạo chỉ mục	6
1.3	Các câu lệnh insert dữ liệu	7
1.3.1	Bảng "DISHES"	8
1.3.2	Bảng "ORDER_DISHES"	8
1.3.3	Bảng "DISHES_LIST"	9
1.3.4	Bảng "ORDER_BILLS"	10
2	Phần riêng	10
2.1	Thành viên 1	10
2.1.1	Thủ tục Insert dữ liệu	10
2.1.2	Trigger	14
2.1.3	Thủ tục chứa câu SQL	18
2.1.4	Hàm	19
2.1.5	Giao diện ứng dụng và các hình ảnh minh họa	21

1 Phần chung

1.1 Các câu lệnh tạo bảng và ràng buộc

```
1      use db_assignment2;
2      create table CUSTOMERS
3      (
4          account_id char(7) not null,
5          username VARCHAR(50) not null,
6          password varchar(50) not null,
7          email VARCHAR(50) null,
8          phone_number char(11) not null,
9          Fname VARCHAR(50) not null,
10         Minit VARCHAR(50) null,
11         Lname VARCHAR(50) not null,
12         Bdate DATE null,
13         gender CHAR(8) not null check(gender='Male' or gender='Female' or
14             gender='LGBT'),
15         registration_date smalldatetime,
16         bonus_point INT not null check(bonus_point >= 0),
17         constraint login_customer PRIMARY KEY(account_id, username),
18         unique (phone_number)
19     );
20
21 -----
22 create table EMPLOYEES
23 (
24     account_id char(7) not null,
25     username VARCHAR(50) not null,
26     password varchar(50) not null,
27     email VARCHAR(50) null,
28     phone_number char(11) not null,
29     Fname VARCHAR(50) not null,
30     Minit VARCHAR(50) null,
31     Lname VARCHAR(50) not null,
32     Bdate DATE not null,
33     gender CHAR(8) not null check(gender='Male' or gender='Female' or
34         gender='LGBT'),
35     registration_date smalldatetime,
36     working_date_per_month INT not null check(working_date_per_month >= 0 and
37         working_date_per_month < 401),
38     constraint login_employee PRIMARY KEY(account_id, username)
39 );
40
41 -----
42 create table CASHIERS
43 (
44     account_id char(7) not null,
45     username VARCHAR(50) not null,
46     accounting_certification varchar(255) not null,
47     primary key (account_id, username),
48     foreign key(account_id, username) REFERENCES EMPLOYEES(account_id,
49         username) on update CASCADE
50 );
```



```
45 -----
46 create table SERVICE_EMPLOYEES
47 (
48     account_id char(7) not null,
49     username VARCHAR(50) not null,
50     high_school_certification varchar(255) not null,
51     primary key (account_id, username),
52     foreign key(account_id, username) REFERENCES EMPLOYEES(account_id,
53         username) on update CASCADE
54 );
55 -----
56 create table CHEFS
57 (
58     account_id char(7) not null,
59     username VARCHAR(50) not null,
60     chef_certification varchar(255) not null,
61     primary key (account_id, username),
62     foreign key(account_id, username) REFERENCES EMPLOYEES(account_id,
63         username) on update CASCADE
64 );
65 -----
66 create table MANAGERS
67 (
68     account_id char(7) not null,
69     username VARCHAR(50) not null,
70     college_certification varchar(255) not null,
71     toeic_certification varchar(255) not null,
72     primary key (account_id, username),
73     foreign key(account_id, username) REFERENCES EMPLOYEES(account_id,
74         username) on update CASCADE
75 );
76 -----
77 create table RESTAURANT_BRANCH
78 (
79     branch_name varchar(50) not null,
80     branch_addr varchar(100) not null,
81     percent_warehouse_available decimal(5, 4) not null
82     check(percent_warehouse_available >= 0 and percent_warehouse_available <=
83         1),
84     primary key(branch_name)
85 );
86 -----
87 create table MANAGE
88 (
89     manager_id char(7) not null,
90     manager_username VARCHAR(50) not null,
91     branch_name varchar(50) not null,
92     first_working_day smalldatetime not null,
93     primary key (manager_id),
94     foreign key(manager_id, manager_username) REFERENCES MANAGERS(account_id,
95         username) on update CASCADE,
96     foreign key (branch_name) REFERENCES RESTAURANT_BRANCH(branch_name) on
97         update CASCADE
```



```
91         );
92         -----
93     create table PROMOTIONS
94     (
95         promotion_id char(8) not null,
96         promotion_name varchar(50) not null,
97         promo_description varchar(250) not null,
98         begin_day smalldatetime not null,
99         end_day smalldatetime not null,
100         PRIMARY key(promotion_id)
101     );
102     -----
103     create table VOUCHERS
104     (
105         voucher_id char(7) not null,
106         promotion_id char(8) not null,
107         voucher_status varchar(10) not null check(voucher_status = 'Available' OR
108             voucher_status = 'Expired' or voucher_status = 'Used'),
109         expired_day smalldatetime not null,
110         voucher_value int not null CHECK(voucher_value > 0),
111         primary key(voucher_id, promotion_id),
112         foreign key(promotion_id) REFERENCES PROMOTIONS(promotion_id)
113     );
114     -----
115     create table SHIPPERS
116     (
117         shipper_id char(7) not null,
118         shipper_name varchar(50) not null,
119         phonenumber char(11) not null,
120         primary key(shipper_id)
121     );
122     -----
123     create table DISHES_CATEGORIES
124     (
125         category_id char(7) not null,
126         category_name varchar(50) not null,
127         category_description varchar(255) not null,
128         num_of_dishes int not null check(num_of_dishes >= 0),
129         primary key(category_id),
130         unique(category_name)
131     );
132     -----
133     create table DISHES
134     (
135         dish_id char(7) not null,
136         dish_name varchar(50) not null,
137         dish_image varchar(255) not null,
138         dish_description varchar(255) not null,
139         price int not null check(price >= 0),
140         category_id char(7) not null,
141         primary key(dish_id),
142         unique(dish_name),
```



```
142         foreign key(category_id) REFERENCES DISHES_CATEGORIES(category_id) on
143         update CASCADE
144     );
145     -----
146     create table ORDER_DISHES
147     (
148         order_id char(7) not null,
149         voucher_added char(7) null,
150         promotion_id char(8) null,
151         customer_id char(7) not null,
152         customer_username varchar(50) not null,
153         order_status char(10) not null
154         check(order_status='Unpaid' or order_status='Paid' or
155             order_status='Delivered' or order_status='Cancelled'),
156         total_price int not null check(total_price >= 0),
157         order_method char(10) not null
158         check(order_method='Live' or order_method='App' or order_method='Website'),
159         branch_name varchar(50) not null,
160         shipper_id char(7) null,
161         primary key(order_id),
162         foreign key(voucher_added, promotion_id) REFERENCES VOUCHERS(voucher_id,
163             promotion_id),
164         foreign key(shipper_id) REFERENCES SHIPPERS(shipper_id) on update CASCADE,
165         foreign key(customer_id, customer_username) REFERENCES
166             CUSTOMERS(account_id, username),
167         foreign key(branch_name) REFERENCES RESTAURANT_BRANCH(branch_name)
168     );
169     -----
170     create table DISHES_LIST
171     (
172         id int IDENTITY(1,1) not null,
173         order_id char(7) not null,
174         dish_id char(7) not null,
175         dish_quantity int not null check(dish_quantity >= 0),
176         primary key(order_id, dish_id),
177         foreign key(order_id) REFERENCES ORDER_DISHES(order_id),
178         foreign key(dish_id) REFERENCES DISHES(dish_id)
179     );
180     -----
181     create table ORDER_BILLS
182     (
183         bill_id char(7) not null,
184         payment_time smalldatetime not null,
185         payment_method char(10) not null
186         check(payment_method='Cash' or payment_method='Bank card' or
187             payment_method='E-wallet'),
188         total_price int not null check(total_price >= 0),
189         order_id char(7) not null,
190         primary key(bill_id, order_id),
191         foreign key(order_id) REFERENCES ORDER_DISHES(order_id)
192     );
193     -----
194     create table SUPPLIERS
```



```
190      (
191          supplier_name varchar(50) not null,
192          supplier_addr varchar(100) not null,
193          supplier_email varchar(50) not null,
194          primary key(supplier_name)
195      );
196      -----
197      create table CONSIGNMENTS
198      (
199          consignment_id char(7) not null,
200          supplier varchar(50) not null,
201          consigned_day smalldatetime not null,
202          chef_id char(7) not null,
203          chef_username varchar(50) not null,
204          primary key(consignment_id),
205          foreign key(chef_id, chef_username) REFERENCES CHEFS(account_id, username)
206              on UPDATE CASCADE,
207          foreign key(supplier) REFERENCES SUPPLIERS(supplier_name) on UPDATE CASCADE
208      );
209      -----
210      create table DISHES_MATERIALS
211      (
212          material_name varchar(50) not null,
213          consignment_id char(7) not null,
214          expired_day date not null,
215          quantity decimal(38,2) not null default 0 check(quantity >= 0),
216          primary key(material_name, consignment_id),
217          foreign key(consignment_id) REFERENCES CONSIGNMENTS(consignment_id) on
              update CASCADE
218      );
```

1.2 Các câu lệnh tạo chỉ mục

```
1      -- Get customer name
2      create index idx_customer_name
3      on CUSTOMERS(Fname, Minit, Lname);
4
5      -- Get dish name
6      create index idx_dish_name
7      on DISHES(dish_name);
8
9      -- Get dish price
10     create index idx_dish_price
11     on DISHES(price);
12
13     -- Get voucher value
14     create index idx_voucher_value
15     on VOUCHERS(voucher_value);
```



1.3 Các câu lệnh insert dữ liệu

```
1  -----
2  INSERT INTO CUSTOMERS
3  VALUES
4  ('111111', 'vitran', '1814804', 'vitran2000_vi@hcmut.edu.vn', '0322432411',
   'Tran', 'Long', 'Vi', '1-1-2000', 'Male', '2021-11-18 18:07:03', 0);
```

```
1  -----
2  INSERT INTO RESTAURANT_BRANCH
3  VALUES
4  ('Ly Thuong Kiet', '130 Ly Thuong Kiet, TP. Ho Chi Minh, Vietnam', 0.43);
```

```
1  -----
2  INSERT INTO PROMOTIONS
3  VALUES
4  ('01112021', 'Black Friday', 'Friday 13th, November 2021', '2021-11-01
   00:00:00', '2021-11-29 00:00:00');
```

```
1  -----
2  INSERT INTO VOUCHERS
3  VALUES
4  ('1234567', '01112021', 'Available', '2021-11-29 00:00:00', 500000);
5  INSERT INTO VOUCHERS
6  VALUES
7  ('7654321', '01112021', 'Expired', '2021-11-29 00:00:00', 200000);
8  INSERT INTO VOUCHERS
9  VALUES
10 ('2233445', '01112021', 'Used', '2021-11-29 00:00:00', 100000);
```

```
1  -----
2  INSERT INTO SHIPPERS
3  VALUES
4  ('0000001', 'Vo Thi An', '0883181332');
```

```
1  -----
2  INSERT INTO DISHES_CATEGORIES
3  VALUES
4  ('0000001', 'Appetizer', 'The first dish served in the meal', 1);
5  INSERT INTO DISHES_CATEGORIES
6  VALUES
7  ('0000002', 'Bbq', 'Grilled meat', 2);
8  INSERT INTO DISHES_CATEGORIES
9  VALUES
10 ('0000003', 'Dessert', 'The Last dish served in the meal', 2);
```

1.3.1 Bảng "DISHES"

Đây là các lệnh dùng để thêm dữ liệu vào bảng "DISHES":

```
1  -----
2  INSERT INTO DISHES
3  VALUES
4  ('111111', 'Mixed vegetable', 'm_veget.png', 'Vegetables mixed with sweet
5      and sour sauce', 19000, '0000001');
6  INSERT INTO DISHES
7  VALUES
8  ('222222', 'Beef stack', 'beef_s.png', 'Kobe beef is grilled on the stone',
9      49000, '0000002');
10 INSERT INTO DISHES
11 VALUES
12 ('333333', 'Grilled chicken', 'g_chicken.png', 'Chicken is baked in a
13     pressure cooker', 60000, '0000002');
14 INSERT INTO DISHES
15 VALUES
16 ('444444', 'Ice cream', 'i_cream.png', 'Delicious cold chocolate ice cream
    ', 25000, '0000003');
17 INSERT INTO DISHES
18 VALUES
19 ('555555', 'Pudding', 'pudding.png', 'A cake like jelly', 21000,
20     '0000003');
```

Để hiển thị dữ liệu trong bảng "DISHES", chúng ta chạy câu lệnh dưới đây:

```
1  select * from DISHES;
```

Sau khi thực hiện truy vấn, kết quả thu được là:

RESULTS						CTRL+A
	dish_id	dish_name	dish_image	dish_description	price	category_id
1	111111	Mixed vegetable	m_veget.png	Vegetables mix...	19000	0000001
2	222222	Beef stack	beef_s.png	Kobe beef is gr...	49000	0000002
3	333333	Grilled chicken	g_chicken.png	Chicken is bake...	60000	0000002
4	444444	Ice cream	i_cream.png	Delicious cold ...	25000	0000003
5	555555	Pudding	pudding.png	A cake like jelly	21000	0000003

1.3.2 Bảng "ORDER_DISHES"

Đây là các lệnh dùng để thêm dữ liệu vào bảng "ORDER_DISHES":

```
1  -----
2  INSERT INTO ORDER_DISHES
3  VALUES
4  ('1231231', null, null, '111111', 'vitran', 'Unpaid', 157000, 'App', 'Ly
5      Thuong Kiet', '0000001');
6  INSERT INTO ORDER_DISHES
7  VALUES
```



```
7      ('4564564', '2233445', '01112021', '1111111', 'vitran', 'Delivered', 330000,  
      'Live', 'Ly Thuong Kiet', null);
```

Để hiển thị dữ liệu trong bảng "ORDER_DISHES", chúng ta chạy câu lệnh dưới đây:

```
1      select * from ORDER_DISHES;
```

Sau khi thực hiện truy vấn, kết quả thu được là:

RESULTS										CRLF	
	order_id	voucher_added	promotion_id	customer_id	customer_user...	order_status	total_price	order_method	branch_name	shipper_id	
1	1231231	NULL	NULL	1111111	vitran	Unpaid	136000	App	Ly Thuong Kiet	0000001	
2	4564564	2233445	01112021	1111111	vitran	Delivered	230000	Live	Ly Thuong Kiet	NULL	

1.3.3 Bảng "DISHES_LIST"

Đây là các lệnh dùng để thêm dữ liệu vào bảng "DISHES_LIST":

```
1      -----  
2      INSERT INTO DISHES_LIST  
3      VALUES  
4      ('1231231', '1111111', 2);  
5      INSERT INTO DISHES_LIST  
6      VALUES  
7      ('1231231', '2222222', 2);  
8      INSERT INTO DISHES_LIST  
9      VALUES  
10     ('1231231', '5555555', 1);  
11  
12     INSERT INTO DISHES_LIST  
13     VALUES  
14     ('4564564', '1111111', 4);  
15     INSERT INTO DISHES_LIST  
16     VALUES  
17     ('4564564', '2222222', 2);  
18     INSERT INTO DISHES_LIST  
19     VALUES  
20     ('4564564', '3333333', 1);  
21     INSERT INTO DISHES_LIST  
22     VALUES  
23     ('4564564', '4444444', 3);  
24     INSERT INTO DISHES_LIST  
25     VALUES  
26     ('4564564', '5555555', 1);
```

Để hiển thị dữ liệu trong bảng "DISHES_LIST", chúng ta chạy câu lệnh dưới đây:

```
1      select * from DISHES_LIST;
```

Sau khi thực hiện truy vấn, kết quả thu được là:



	id	order_id	dish_id	dish_quantity
1	1	1231231	1111111	2
2	2	1231231	2222222	2
3	4	4564564	1111111	4
4	5	4564564	2222222	2
5	6	4564564	3333333	1
6	7	4564564	4444444	3
7	8	4564564	5555555	1

1.3.4 Bảng "ORDER_BILLS"

Đây là các lệnh dùng để thêm dữ liệu vào bảng "ORDER_BILLS":

```
1 -----  
2 INSERT INTO ORDER_BILLS  
3 VALUES  
4 ('2021-11-20 19:00:34', 'Cash', 230000, '4564564');
```

Để hiển thị dữ liệu trong bảng "ORDER_BILLS", chúng ta chạy câu lệnh dưới đây:

```
1 select * from ORDER_BILLS;
```

Sau khi thực hiện truy vấn, kết quả thu được là:

RESULTS					
	bill_id	payment_time	payment_meth...	total_price	order_id
1	0000001	2021-11-20 19:...	Cash	230000	4564564

2 Phần riêng

2.1 Thành viên 1

- Họ tên: Trần Long Vĩ
- MSSV: 1814804
- DBMS: MSSQL server

2.1.1 Thủ tục Insert dữ liệu

2.1.1.1 Thủ tục Insert dữ liệu cho bảng "ORDER_DISHES"

Chức năng của procedure này dùng để tạo một hàng trong bảng "ORDER_DISHES".
Câu lệnh tạo thủ tục:



```
1 CREATE or alter PROCEDURE Insert_Tuple_2_ORDER_DISHES
2 @order_id char(7),
3 @voucher_added char(7),
4 @customer_username varchar(50),
5 @order_status char(10),
6 @order_method char(10),
7 @branch_name varchar(50),
8 @shipper_id char(7)
9 as
10 begin
11 DECLARE @flag AS int
12 set @flag = 0
13 IF @order_id=null
14 BEGIN
15 RAISERROR ('Order ID must not equal "NULL!"', 16, 1)
16 set @flag = -1
17 END
18 if @customer_username=null
19 BEGIN
20 RAISERROR ('Customer Username must not equal "NULL!"', 16, 1)
21 set @flag = -1
22 END
23 if @order_status=null
24 BEGIN
25 RAISERROR ('Order status must not equal "NULL!"', 16, 1)
26 set @flag = -1
27 END
28 if (@order_status != 'Unpaid' and @order_status != 'Paid' and
    @order_status != 'Delivered' and @order_status != 'Cancelled')
29 BEGIN
30 RAISERROR ('Invalid Order status value!', 16, 1)
31 set @flag = -2
32 END
33 if @order_method=null
34 BEGIN
35 RAISERROR ('Last name must not equal "NULL!"', 16, 1)
36 set @flag = -1
37 END
38 if (@order_method != 'Live' and @order_method != 'App' and @order_method != 'Website')
39 BEGIN
40 RAISERROR ('Invalid Order method value!', 16, 1)
41 set @flag = -2
42 END
43 if @branch_name=null
44 BEGIN
45 RAISERROR ('Gender must not equal "NULL!"', 16, 1)
46 set @flag = -1
47 END
48
49 IF @flag=0
50 BEGIN
51 declare @promotion_id char(8);
52 SET @promotion_id = (select promotion_id from VOUCHERS where voucher_id =
```

```
        @voucher_added);
53 declare @customer_id char(7);
54 set @customer_id = (select account_id from CUSTOMERS where username =
        @customer_username);
55 INSERT INTO ORDER_DISHES
56 VALUES
57 (@order_id, @voucher_added, @promotion_id, @customer_id, @customer_username,
        @order_status, 0, @order_method, @branch_name, @shipper_id)
58 END
59 end;
```

Procedure này nhận vào 7 đối số ứng với 7/10 thuộc tính trong bảng "ORDER_DISHES". Ba thuộc tính còn lại bao gồm:

- "promotion_id" là mã của sự kiện, sẽ được truy vấn trong bảng "VOUCHERS" dựa trên "voucher_id" với điều kiện voucher đó chưa hết hạn.
- "customer_id" là mã khách hàng, sẽ được truy vấn trong bảng "CUSTOMERS" dựa trên "customer_username".
- "total_price" là tổng số tiền khách hàng phải thanh toán. Giá trị này sẽ được cập nhật khi thêm các tuple tương ứng với "order_id" vào bảng "DISHES_LIST".

Câu lệnh thực thi thủ tục mẫu:

```
1 exec Insert_Tuple_2_ORDER_DISHES
2 '4554443', null, 'vitran', 'Unpaid', "Live", "Ly Thuong Kiet", null;
```

Sau khi chạy, một tuple mới sẽ được thêm vào "ORDER_DISHES" với *total_price* = 0:

	order_id	voucher_added	promotion_id	customer_id	customer_user...	order_status	total_price	order_method	branch_name	shipper_id
1	1231231	NULL	NULL	1111111	vitran	Unpaid	136000	App	ly Thuong Kiet	0000001
2	4554443	NULL	NULL	1111111	vitran	Unpaid	0	Live	ly Thuong Kiet	NULL
3	4564564	2233445	01112021	1111111	vitran	Delivered	230000	Live	ly Thuong Kiet	NULL

2.1.1.2 Thủ tục insert dữ liệu cho bảng "DISHES_LIST"

Chức năng của procedure này dùng để tạo một hàng hoặc chỉnh sửa số lượng món ăn trong bảng "DISHES_LIST". Tuple sẽ được chỉnh sửa nếu đã tồn tại tuple tương ứng, khi đó giá trị "dish_quantity" sẽ được cập nhật dựa trên dữ liệu nhập vào.

Câu lệnh tạo thủ tục:

```
1 CREATE or alter PROCEDURE Insert_Tuple_2_DISHES_LIST
2 @order_id char(7),
3 @dish_id char(7),
4 @dish_quantity int
5 as
6 begin
7 DECLARE @flag AS int
8 set @flag = 0
9 IF @order_id=null
10 BEGIN
11 RAISERROR ('Order ID must not equal "NULL"!', 16, 1)
```

```
12      set @flag = -1
13      END
14      if @dish_id=null
15      BEGIN
16      RAISERROR ('Dish ID must not equal "NULL"!', 16, 1)
17      set @flag = -1
18      END
19      if @dish_quantity=null
20      BEGIN
21      RAISERROR ('Dish quantity must not equal "NULL"!', 16, 1)
22      set @flag = -1
23      END
24      if @dish_quantity < 0
25      BEGIN
26      RAISERROR ('Dish quantity must not equal "NULL"!', 16, 1)
27      set @flag = -1
28      END
29
30      IF @flag=0
31      BEGIN
32      if not exists(select * from DISHES_LIST where order_id = @order_id and dish_id =
          @dish_id)
33      begin
34      INSERT INTO DISHES_LIST
35      VALUES
36      (@order_id, @dish_id, @dish_quantity)
37      end
38      else
39      BEGIN
40      UPDATE DISHES_LIST
41      SET dish_quantity = @dish_quantity
42      where order_id = @order_id and dish_id = @dish_id;
43      end
44      END
45      end;
```

Câu lệnh thực thi thử mẫu:

```
1      exec Insert_Tuple_2_DISHES_LIST '4564564', '5555555', 1;
2      select * from DISHES_LIST;
```

Sau khi chạy, một tuple mới sẽ được thêm vào "DISHES_LIST":

RESULTS				
	id	order_id	dish_id	dish_quantity
1	1	1231231	1111111	2
2	2	1231231	2222222	2
3	4	4564564	1111111	4
4	5	4564564	2222222	2
5	6	4564564	3333333	1
6	7	4564564	4444444	3
7	8	4564564	5555555	1

Nếu tiếp tục chạy lệnh trên với giá trị "dish_quantity" khác, ta sẽ thấy giá trị này thay đổi ở hàng tương ứng thay vì tạo một hàng mới:

	id	order_id	dish_id	dish_quantity
1	1	1231231	1111111	2
2	2	1231231	2222222	2
3	4	4564564	1111111	4
4	5	4564564	2222222	2
5	6	4564564	3333333	1
6	7	4564564	4444444	3
7	8	4564564	5555555	2

2.1.2 Trigger

2.1.2.1 Trigger kiểm soát các hành động của bảng "DISHES_LIST"

Trigger dùng để kiểm soát hành vi INSERT, DELETE và UPDATE trên bảng "DISHES_LIST" có nhiệm vụ kiểm tra "order_id" trong bảng "inserted" hoặc "deleted" tương ứng, nếu hợp lệ, sẽ tự động cập nhật lại giá trị "total_price" trong bảng "ORDER_DISHES".

Câu lệnh tạo trigger:

```
1 create or alter TRIGGER Insert_Update_DISH_LIST_trigger
2 on DISHES_LIST
3 after INSERT, UPDATE
4 AS
5 BEGIN
6 DECLARE @order_id INT;
7 SET @order_id = (select top 1 order_id from inserted);
8 if @order_id is null
9 BEGIN
10 RAISERROR ('Order ID must not equal "NULL"', 16, 1);
11 ROLLBACK;
```

```
12     end;
13     declare @payment_price int;
14     set @payment_price = (select dbo.cal_payment_price(@order_id));
15     if @payment_price < 0 or @payment_price = null
16     BEGIN
17         RAISERROR ('Oops, something happened!', 16, 1);
18         ROLLBACK;
19     end;
20     update ORDER_DISHES
21     set total_price = @payment_price
22     where order_id = @order_id;
23     end
24     go
25
26     create or alter TRIGGER Delete_DISH_LIST_trigger
27     on DISHES_LIST
28     after DELETE
29     AS
30     BEGIN
31         DECLARE @order_id INT;
32         SET @order_id = (select top 1 order_id from deleted);
33         if @order_id is null
34         BEGIN
35             RAISERROR ('Order ID must not equal "NULL"!', 16, 1);
36             ROLLBACK;
37         end;
38         declare @payment_price int;
39         set @payment_price = (select dbo.cal_payment_price(@order_id));
40         if @payment_price < 0 or @payment_price is null
41         BEGIN
42             RAISERROR ('Oops, something happened!', 16, 1);
43             ROLLBACK;
44         end;
45         update ORDER_DISHES
46         set total_price = @payment_price
47         where order_id = @order_id;
48     end
49     go
```

Để kiểm tra câu lệnh trên hoạt động, ta sử dụng procedure được giới thiệu ở phần trước để thêm tuple vào bảng "DISHES_LIST", sau đó so sánh bảng dữ liệu trước và sau câu lệnh:

```
1     exec Insert_Tuple_2_DISHES_LIST '4564564', '5555555', 1;
2     select dl.order_id, dl.dish_id, total_price from DISHES_LIST dl, ORDER_DISHES od
       where od.order_id = '4564564' and dl.order_id = '4564564';
```

Kết quả thu được cho thấy giá trị "total_price" trong bảng "ORDER_DISHES" đã giảm xuống một ứng với món có id "5555555" và giá 21.000đ:

	order_id	dish_id	dish_quantity	total_price
1	4564564	1111111	4	251000
2	4564564	2222222	2	251000
3	4564564	3333333	1	251000
4	4564564	4444444	3	251000
5	4564564	5555555	2	251000

Before

	order_id	dish_id	dish_quantity	total_price
1	4564564	1111111	4	230000
2	4564564	2222222	2	230000
3	4564564	3333333	1	230000
4	4564564	4444444	3	230000
5	4564564	5555555	1	230000

After

2.1.2.2 Trigger kiểm soát các hành động của bảng "ORDER_DISHES"

Trigger này dùng để kiểm soát hành động INSERT, UPDATE của bảng "ORDER_DISHES". Khi một tuple của bảng có giá trị "order_status" được thiết lập ở giá trị "Paid" hoặc "Delivered":

- Bảng "ORDER_BILLS" sẽ được INSERT thêm tuple mới tương ứng với "order_id" vừa được INSERT hoặc UPDATE.
- Nếu tuple tương ứng được áp dụng voucher, giá trị voucher đó sẽ được chuyển thành "Used".

Câu lệnh tạo trigger:

```

1  create or alter TRIGGER Update_ORDER_DISHES_trigger
2  on ORDER_DISHES
3  after INSERT, UPDATE
4  AS
5  BEGIN
6  DECLARE @order_id INT;
7  SET @order_id = (select order_id from inserted);
8  if @order_id is null
9  BEGIN
10 RAISERROR ('Order ID must not equal "NULL"!', 16, 1);
11 ROLLBACK;
12 end;
13 DECLARE @voucher_id char(7);
14 set @voucher_id = (select voucher_added from ORDER_DISHES where order_id =
    @order_id);
15 DECLARE @order_status CHAR(10);
16 set @order_status = (select order_status from ORDER_DISHES where order_id =
    @order_id);
17 if (@order_status='Paid' or @order_status='Delivered')
```

```
18     begin
19     if @voucher_id is not null
20     begin
21     DECLARE @voucher_status varchar(10);
22     set @voucher_status = (select voucher_status from VOUCHERS where voucher_id =
        @voucher_id);
23     if (@voucher_status != 'Expired')
24     update VOUCHERS
25     set voucher_status = 'Used'
26     where voucher_id = @voucher_id;
27     end
28     if not exists (select * from ORDER_BILLS where order_id = @order_id)
29     begin
30     declare @datetime smalldatetime;
31     set @datetime = (select CAST(GETDATE() AS smalldatetime));
32     declare @price int;
33     set @price = (select total_price
34     from ORDER_DISHES
35     where order_id = @order_id);
36     insert into ORDER_BILLS
37     values (@datetime, 'Cash', @price, @order_id);
38     end;
39     end;
40     end
41     go
```

Sau khi thực hiện tạo trigger, ta có thể test bằng cách thực hiện truy vấn tới 2 bảng "VOUCHERS" và "DISHES_LIST" để kiểm tra sự thay đổi:

```
1     -- UPDATE voucher status tester
2     update ORDER_DISHES set order_status = 'Paid' where order_id = '4564564';
3     select voucher_added, voucher_value, voucher_status from ORDER_DISHES od, VOUCHERS
        v
4     where voucher_added = voucher_id;
5     -- INSERT tuple to ORDER_BILLS tester
6     update ORDER_DISHES set order_status = 'Paid' where order_id = '5067984';
7     select * from ORDER_BILLS;
```

Kết quả thu được sau khi thực thi các lệnh truy vấn được mô tả dưới đây:



Voucher status and order_status

	order_status	voucher_added	voucher_value	voucher_status
1	Unpaid	2233445	100000	Available

ORDER_BILLS table

	bill_id	payment_time	payment_meth...	total_price	order_id
1	1	2021-11-20 19:...	Cash	230000	4564564

Before

Voucher status and order_status

	order_status	voucher_added	voucher_value	voucher_status
1	Delivered	2233445	100000	Used

ORDER_BILLS table

	bill_id	payment_time	payment_meth...	total_price	order_id
1	1	2021-11-20 19:...	Cash	230000	4564564
2	4	2021-11-26 22:...	Cash	189000	5067984

After

2.1.3 Thủ tục chứa câu SQL

2.1.3.1 Thủ tục truy vấn danh sách món ăn dựa trên order_id

Thủ tục này được thiết kế nhằm mục đích truy vấn các thông tin về danh sách món ăn thuộc một đơn hàng, bao gồm các thông tin về tên món ăn, giá tiền, số lượng và tổng giá. Thủ tục nhận vào một đối số là "order_id" là mã của đơn hàng cần truy vấn.

Các câu lệnh dùng để tạo thủ tục:

```
1 CREATE or alter PROCEDURE Display_dish_list_of_order
2   @order_id char(7)
3   as
4   begin
5     DECLARE @flag AS int
6     set @flag = 0
7     IF @order_id=null
8     BEGIN
9       RAISERROR ('Order ID must not equal "NULL"!', 16, 1)
10      set @flag = -1
11    END
12
13    IF @flag=0
14    BEGIN
15      select dish_name, price, dish_quantity, (price*dish_quantity) as 'Total price'
16      from DISHES d, DISHES_LIST l
17      where d.dish_id = l.dish_id and l.order_id = @order_id
18      order by dish_quantity ASC
19    END
20  end;
```

Để thực thi thủ tục trên, ta thực hiện lệnh dưới đây (ở dưới đây là ví dụ áp dụng cho order_id = '4564564'):

```
1 exec Display_dish_list_of_order '4564564';
```

Sau khi thực thi lệnh trên, kết quả thu được là:

	dish_name	price	dish_quantity	Total price
1	Grilled chicken	60000	1	60000
2	Beef stack	49000	2	98000
3	Pudding	21000	2	42000
4	Ice cream	25000	3	75000
5	Mixed vegetable	19000	4	76000

2.1.3.2 Thủ tục truy vấn các thông tin liên quan đến đơn hàng

Thủ tục này được thiết kế nhằm truy vấn các thông tin về mã đơn hàng, tên đăng nhập của khách hàng, giá gốc - giá thanh toán - giá được giảm cũng như các thông tin khác về hình thức thanh toán, trạng thái đơn hàng và chi nhánh đặt hàng.

Các câu lệnh dùng để tạo thủ tục:

```
1 CREATE or alter PROCEDURE Display_order_dishes_list
2 as
3 begin
4     select dl.order_id, customer_username as 'username',
5     sum(price*dish_quantity) as 'o_price', total_price as 'p_price',
6     dbo.cal_decreased_price(od.order_id) as 'd_price',
7     order_method, order_status, branch_name
8     from DISHES_LIST dl, DISHES d, ORDER_DISHES od
9     where dl.dish_id = d.dish_id and dl.order_id = od.order_id
10    group by dl.order_id, customer_username, total_price, od.order_id, order_method,
11            order_status, branch_name
12    order by sum(price*dish_quantity) asc;
end;
```

Sau khi khởi tạo, để thực thi thủ tục trên, ta chỉ cần thực thi câu lệnh dưới đây:

```
1 exec Display_order_dishes_list;
```

Kết quả thu được là:

order_id	username	o_price	p_price	d_price	order_method	order_status	branch_name
1231231	vitran	136000	136000	0	App	Unpaid	Ly Thuong Kiet
5067984	vitran	189000	189000	0	Live	Paid	Ly Thuong Kiet
4564564	vitran	351000	251000	100000	Live	Delivered	Ly Thuong Kiet

2.1.4 Hàm

2.1.4.1 Hàm tính giá được giảm bởi voucher

Hàm được thiết kế với mục đích tính giá trị giảm của đơn hàng được áp dụng voucher. Hàm nhận vào mã đơn hàng và trả về kết quả bằng 0 nếu không có voucher, hoặc bằng giá trị của

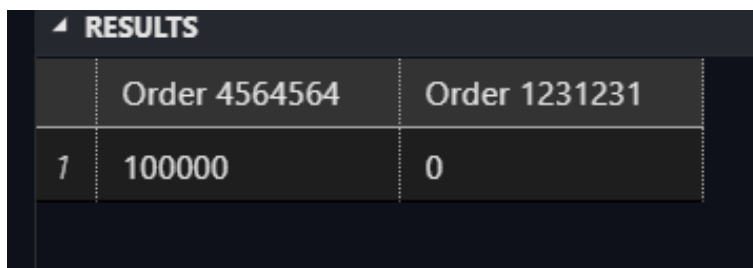
voucher nếu voucher ID hợp lệ.
Các câu lệnh dùng để tạo hàm

```
1      create or alter function cal_decreased_price(@order_id char(7))
2      returns INT
3      AS
4      BEGIN
5          if @order_id is NULL
6              return -1;
7          DECLARE @voucher_id char(7);
8          set @voucher_id = (select voucher_added from ORDER_DISHES where order_id =
                              @order_id);
9          DECLARE @promotion_id char(8);
10         set @promotion_id = (select promotion_id from order_dishes where order_id =
                               @order_id);
11         if (@voucher_id is NULL) or (@promotion_id is null)
12             return 0;
13
14         DECLARE @decreased_value int;
15         set @decreased_value = (select voucher_value from VOUCHERS where voucher_id =
                                  @voucher_id and promotion_id = @promotion_id);
16         if (@decreased_value is null) or (@decreased_value < 0)
17             return -2;
18         return @decreased_value
19     end
```

Để thực thi và kiểm tra hàm, ta dùng câu lệnh dưới đây với mã đơn hàng là '4564564' (có áp dụng voucher) và '1231231' (không áp dụng voucher):

```
1      select dbo.cal_decreased_price('4564564') as 'Order 4564564',
2      dbo.cal_decreased_price('1231231') 'Order 1231231';
```

Sau khi thực thi lệnh, kết quả thu được là:



	Order 4564564	Order 1231231
1	100000	0

2.1.4.2 Hàm tính số tiền phải thanh toán

Hàm này được thiết kế để tính tổng số tiền phải thanh toán của một đơn hàng, với đối số nhận vào là mã đơn hàng đó. Hàm sau khi được thực thi sẽ truy vấn các giá trị liên quan để tính tổng giá gốc của đơn hàng và giá được giảm bởi voucher (nếu có), sau đó trả về tổng số tiền phải thanh toán.

Các câu lệnh tạo hàm được hiển thị dưới đây:

```
1      create or alter FUNCTION cal_payment_price(@order_id char(7))
2      returns int
3      AS
4      BEGIN
5          if @order_id is NULL
6              return -1;
7          declare @total_price INT;
8          set @total_price = (select sum(price*dish_quantity) from DISHES d, DISHES_LIST l
9                               where l.order_id = @order_id and d.dish_id = l.dish_id);
10         if @total_price is NULL
11             return 0;
12         declare @decreased_value int;
13         set @decreased_value = (select dbo.cal_decreased_price(@order_id));
14         if @decreased_value <= 0
15             return @total_price;
16
17         declare @payment_value int;
18         set @payment_value = @total_price - @decreased_value;
19         if @payment_value < 0
20             return 0;
21         return @payment_value;
22     END
```

Để kiểm tra tính chính xác của hàm trên, ta có thể thực thi lệnh truy vấn dưới đây:

```
1      select dl.order_id, dbo.cal_payment_price(od.order_id) as 'payment price',
2             sum(price*dish_quantity) as 'original price',
3             dbo.cal_decreased_price(od.order_id) as 'decreased price'
4      from ORDER_DISHES od, DISHES_LIST dl, DISHES d
5      where od.order_id = dl.order_id and dl.dish_id = d.dish_id
6      group by od.order_id, dl.order_id;
```

Kết quả thu được là:

RESULTS				
	order_id	payment price	original price	decreased price
1	1231231	136000	136000	0
2	4564564	251000	351000	100000
3	5067984	189000	189000	0

2.1.5 Giao diện ứng dụng và các hình ảnh minh họa

Giao diện của ứng dụng là một bảng được liệt kê các mã đơn hàng và các thông tin liên quan tới mã đơn hàng đó. Dưới đây là hình ảnh giao diện của web: Trên giao diện của web, chúng ta có 3 nút thao tác:

- Nút dùng để thêm đơn hàng mới (New order): khi nút này được ấn, một form sẽ được hiển



New order

ID	Username	Original price	Payment price	Decreased price	Order method	Order status	Branch name	Manipulations
1231231	vitran	136000	136000	0	App	Unpaid	Ly Thuong Kiet	Del Edit
5067984	vitran	189000	189000	0	Live	Paid	Ly Thuong Kiet	Del Edit
4564564	vitran	351000	251000	100000	Live	Delivered	Ly Thuong Kiet	Del Edit

thì yêu cầu người dùng nhập các thông tin tương ứng để tạo đơn hàng. Ở đây, các trường

New order

username:

Dishes list:

<input type="checkbox"/> Mixed vegetable (19000 dong)	<input type="text" value="0"/>
<input type="checkbox"/> Beef stack (49000 dong)	<input type="text" value="0"/>
<input type="checkbox"/> Grilled chicken (60000 dong)	<input type="text" value="0"/>
<input type="checkbox"/> Ice cream (25000 dong)	<input type="text" value="0"/>
<input type="checkbox"/> Pudding (21000 dong)	<input type="text" value="0"/>

Voucher:

Status:

Method:

Branch:

ID	Username	Original price	Payment price	Decreased price	Order method	Order status	Branch name	Manipulations
1231231	vitran	136000	136000	0	App	Unpaid	Ly Thuong Kiet	Del Edit

của form đều sẽ được kiểm tra dữ liệu trước khi được INSERT vào table. Cụ thể, có hai trường hợp lỗi, là lỗi về format nếu bỏ trống hoặc nhập sai trường dữ liệu (voucher có thể bỏ trống) hoặc lỗi không tồn tại nếu xét giá trị của username hoặc voucher. Tất cả các lỗi đều sẽ được thông báo ra màn hình chính dưới dạng chuỗi.

Lỗi không tồn tại giá trị trong db

username is not exist!
Voucher had been used, expired, or no exists!
Invalid value! Please check again.

Lỗi format

Please Enter username!
Please choose order status!
Please choose order method!
Please choose order branch!
Invalid value! Please check again.

Nếu dữ liệu nhập hợp lệ, đơn hàng mới sẽ được INSERT vào database và hiển thị lại ở bảng dưới.

username: vitran

Dishes list:

- ☐ Mixed vegetable (19000 dong) 0
- ☒ Beef stack (49000 dong) 1
- ☒ Grilled chicken (60000 dong) 2
- ☐ Ice cream (25000 dong) 0
- ☐ Pudding (21000 dong) 0

Voucher: Enter voucher ID (Include 7 characters)

Status: Unpaid

Method: Website

Branch: Ly Thuong Kiet

OK

ID	Username	Original price	Payment price	Decreased price	Order method	Order status	Branch name	Manipulations
1231231	vitran	136000	136000	0	App	Unpaid	Ly Thuong Kiet	Del Edit
7914536	vitran	169000	169000	0	Website	Unpaid	Ly Thuong Kiet	Del Edit
5067984	vitran	189000	189000	0	Live	Paid	Ly Thuong Kiet	Del Edit

- Nút xóa được hiện ở cột Manipulations trong bảng, được dùng để xóa hàng tương ứng ra khỏi database.



ID	Username	Original price	Payment price	Decreased price	Order method	Order status	Branch name	Manipulations
1231231	vitran	136000	136000	0	App	Unpaid	Ly Thuong Kiet	Del Edit
7914536	vitran	169000	169000	0	Website	Unpaid	Ly Thuong Kiet	Del Edit
5067984	vitran	189000	189000	0	Live	Paid	Ly Thuong Kiet	Del Edit
4564564	vitran	351000	251000	100000	Live	Delivered	Ly Thuong Kiet	Del Edit

↓

ID	Username	Original price	Payment price	Decreased price	Order method	Order status	Branch name	Manipulations
1231231	vitran	136000	136000	0	App	Unpaid	Ly Thuong Kiet	Del Edit
5067984	vitran	189000	189000	0	Live	Paid	Ly Thuong Kiet	Del Edit
4564564	vitran	351000	251000	100000	Live	Delivered	Ly Thuong Kiet	Del Edit

- Nút Edit được hiện ở cột Manipulations trong bảng, được dùng để chỉnh sửa các giá trị của một đơn hàng.

Khi nhấn vào nút này, trang sẽ tự chuyển hướng tới trang edit dùng để lấy các thông tin cần chỉnh sửa. Các trường thông tin ở đây đều sẽ kiểm tra đầu vào tương tự như tạo mới đơn hàng.

ID	Username	Original price	Payment price	Decreased price	Order method	Order status	Branch name	Manipulations
1231231	vitran	136000	136000	0	App	Unpaid	Ly Thuong Kiet	Del Edit
5067984	vitran	189000	189000	0	Live	Paid	Ly Thuong Kiet	Del Edit
4564564	vitran	351000	251000	100000	Live	Delivered	Ly Thuong Kiet	Del Edit

Order ID:

Username:

Order status:

Order method:

Branch name:

Dishes list:

☒ Mixed vegetable (19000 dong)

☐ Beef stack (49000 dong)

☒ Grilled chicken (60000 dong)

☐ Ice cream (25000 dong)

☒ Pudding (21000 dong)

Total price = 100.000đ

[Edit](#)

ID	Username	Original price	Payment price	Decreased price	Order method	Order status	Branch name	Manipulations
1231231	vitran	19000	19000	0	App	Unpaid	Ly Thuong Kiet	Del Edit
5067984	vitran	100000	100000	0	Live	Delivered	Ly Thuong Kiet	Del Edit
4564564	vitran	351000	251000	100000	Live	Delivered	Ly Thuong Kiet	Del Edit