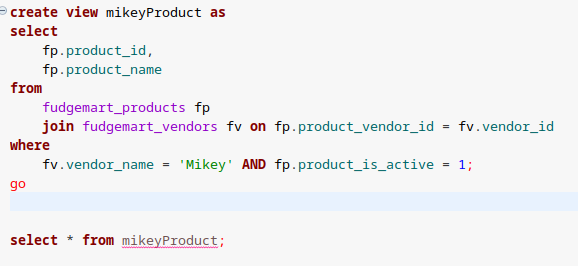
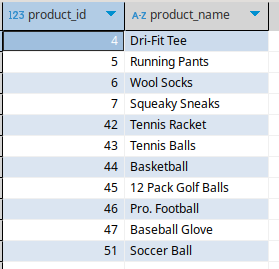
**QUESTION NO. 1. (2 MARKS)**

Write views to show

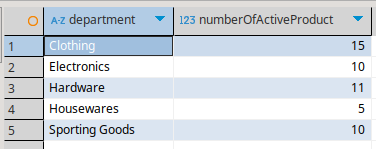
a. The list of active products (is\_active is 1) provided by the vendor Mikey





b. The list of product department and the number of active products of each product

department



**QUESTION NO. 2. (2 MARKS)**

Write one stored procedure with the following requirements

a. Check if the product is provided by the vendor Fudgeman, the retail price must be greater

than 30% of the wholesale price. If the product is provided by other vendors, the retail

price must be greater than 20% of the wholesale price. Otherwise, print the message

'Cannot insert this product due to the wrong price'

**create** **procedure** InsertProduct

**@ProductDepartment** **NVARCHAR**(20),

**@ProductName** **NVARCHAR**(100),

**@RetailPrice** **DECIMAL**(10, 2),

**@WholesalePrice** **DECIMAL**(10, 2),

**@ProductStatus** **BIT**,

**@ProductDate** **DATETIME**,

**@VendorID** **INT**

**as**

**begin**

**declare** **@VendorName** **NVARCHAR**(100);

**select** **@VendorName** = fv.vendor\_name

**from** fudgemart\_vendors fv

**where** fv.vendor\_id = **@VendorID**;

**if** **@VendorName** = 'Fudgeman'

**begin**

**if** **@RetailPrice** < (**@WholesalePrice** \* 1.3)

**begin**

**print** 'Cannot insert this product due to the wrong price';

**return**;

**end**

**end**

**else**

**begin**

**if** **@RetailPrice** < (**@WholesalePrice** \* 1.2)

**begin**

**print** 'Cannot insert this product due to the wrong price';

**return**;

**end**

**end**

-- Insert the product into the database if price is valid

**insert** **into** fudgemart\_products

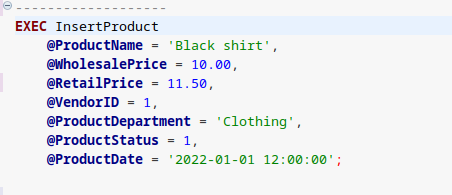
(product\_id, product\_department, product\_name, product\_retail\_price, product\_wholesale\_price, product\_is\_active, product\_add\_date, product\_vendor\_id, product\_description)

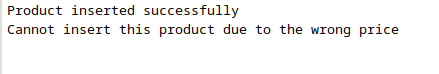
**values** (100, @ProductDepartment, @ProductName, @RetailPrice, @WholesalePrice, @ProductStatus, @ProductDate, @VendorID, **null**);

**print** 'Product inserted successfully';

**end**;

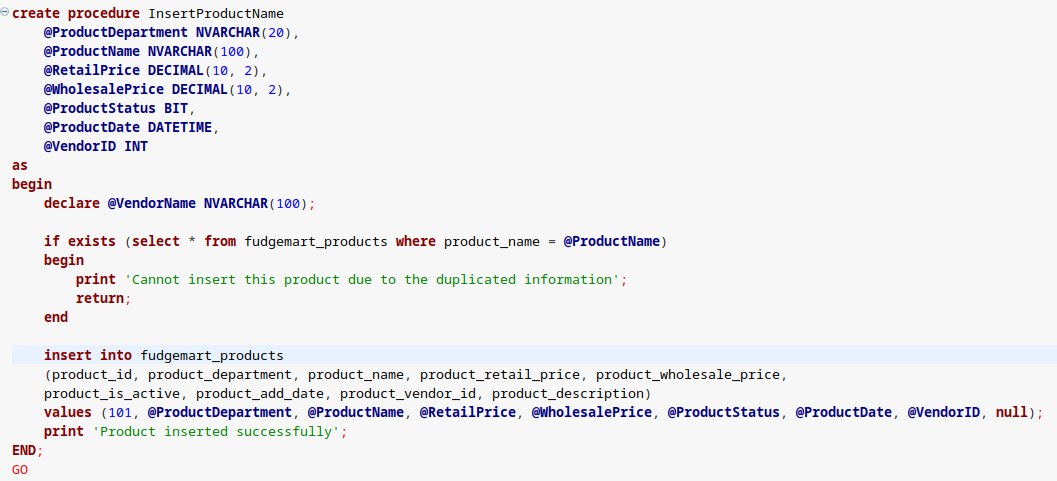
go

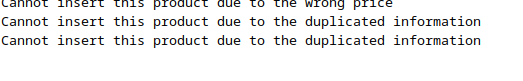
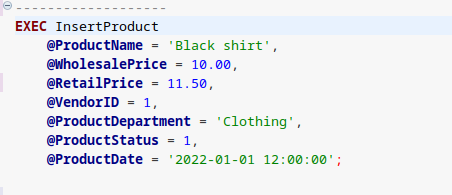




b. Check if product name have already existed in the database, print the message 'Cannot

insert this product due to the duplicated information’





c. If a and b are satisfied, insert into the fudgemart\_products with the values provided in the

parameters of the stored procedure

Hình như em là cái task này ở 2 câu trên rồi

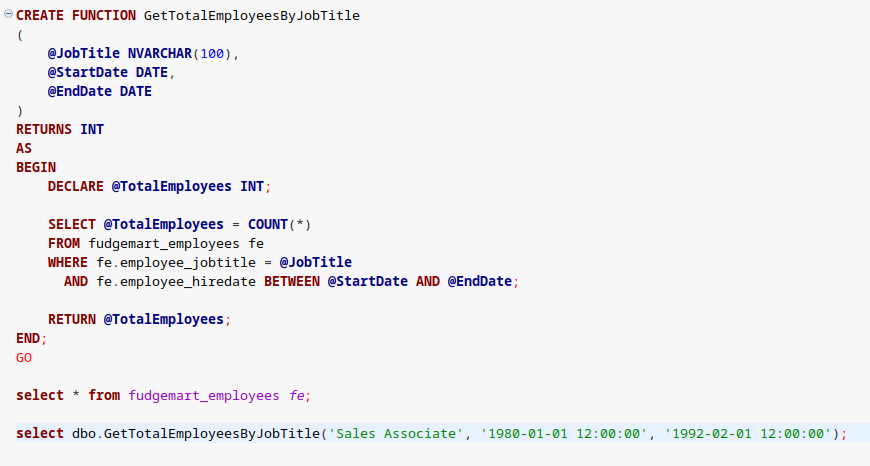
QUESTION NO.3. (2 MARKS)

Write one function with the following requirements

c. Input: employee job title, start\_date, end\_date

d. Output: the total employees are hired from the start\_date to the end\_date with the job title

given in the input parameters



Em không biết lại sao nhưng nó không kết quả là 0 :)  
Cô tha em bài này nhé  **\*nháy mắt\* \*nháy mắt\***

**QUESTION NO.4. (2.5 MARKS)**

Em khong biet :(

**QUESTION NO.5 (2.5 MARKS)**

Write one trigger with the following requirement:

a. Trigger happens on the table fudgemart\_products has the type instead of insert

b. Requirement:

if product department is Electronics, the retail price is calculated by the wholesale price \*

1.3

if product department is others, the retail price is calculated by the wholesale price \* 1.2

c. Instead of inserting with the values provided by users, the trigger will insert into the

fudgemart\_products with the new retail price calculated by the above requirement.

**CREATE** **TRIGGER** trg\_CalculateRetailPrice

**ON** fudgemart\_products

INSTEAD **OF** **INSERT**

**AS**

**BEGIN**

**DECLARE** **@ProductDepartment** **NVARCHAR**(100);

**DECLARE** **@WholesalePrice** **DECIMAL**(10, 2);

**DECLARE** **@RetailPrice** **DECIMAL**(10, 2);

-- Get the department and wholesale price from the inserted data

**SELECT**

**@ProductDepartment** = product\_department ,

**@WholesalePrice** = product\_wholesale\_price

**FROM** inserted;

-- Calculate the retail price based on the department

**IF** **@ProductDepartment** = 'Electronics'

**BEGIN**

**SET** **@RetailPrice** = **@WholesalePrice** \* 1.3;

**END**

**ELSE**

**BEGIN**

**SET** **@RetailPrice** = **@WholesalePrice** \* 1.2;

**END**

-- Insert the product with the new retail price

**INSERT** **INTO** fudgemart\_products (product\_name, product\_wholesale\_price, product\_retail\_price, product\_department, product\_vendor\_id)

**SELECT**

product\_name,

product\_wholesale\_price,

**@RetailPrice**,

product\_department,

product\_vendor\_id

**FROM** inserted;

**END**;

GO

