# Mô tả:

Một tiệm bánh cần quản lí việc mua bán và nhập hàng của tiệm. Sau đây là phần mô tả theo cá nghiệp vụ hằng ngày của tiệm:

Nhân viên bán hàng chỉ có thể bán những sản phẩm có mặt trong tiệm với tên riêng, đơn vị tính và mã định danh sản phẩm. Ví dụ: Bánh kem nhỏ với đơn vị là bánh, có mã định danh là BKN1234,…

Mỗi sản phẩm có mã hàng riêng biệt để phân biệt với nhau. Mã số là mã vạch của sản phẩm sẽ được in ra để người bán dễ dàng tạo thanh toán cho khách hàng. Mỗi sản phẩm đề có giá nhập và giá bán riêng và sẽ được lưu trữ trên cơ sở dữ liệu để Giám đốc có thể xem. Cần có các giảm giá % theo sự kiện. Bánh cần có hình ảnh minh họa.

Mỗi hàng hóa có đơn vị tính khác nhau: cái, bánh, hộp,…

Khi bán các đơn hàng sẽ có chi tiết của đơn hàng riêng cho biết khách hàng đã đặt hoặc mua những món nào có giá bán, giá mua, số lượng và thành tiền là bao nhiêu.

Nhân viên bán có thể xuất các biên nhận đặt trước và đặc cọc.

Nhân viên làm việc trong tiệm thì sẽ được quản lí về: Tên, địa chỉ, điện thoại,.. và phải cấp quyền để đăng nhập vào hệ thống với tên đăng nhập và mật khẩu.

Ngoài ra, Kế toán cũng có chức năng giống nhân viên nhưng họ có những nhiệm vụ cao hơn như: nhập hàng, kiểm kê hàng hóa và có thể xem một phần của quản lí của Giám đốc để có thể thống kê sau mỗi kỳ kinh doanh..

Khi nhập hàng, Giám đốc sẽ tạo phiếu với sản phẩm, số lượng và giá tiền, ngày tạo phiếu và mã tạo phiếu sẽ được tạo tự động.

Nhà cung cấp sẽ được lưu các thông tin: tên, số điện thoai (dùng để định danh các nhà cung cấp).

Khi Xuất bánh, có thể xuất đi các cơ sở khác, bán bánh, xuất trả về đối tác nếu bánh nhập bị lỗi, xuất hủy khi bánh bị hỏng,…

Ngoài bán bánh, tiệm sẽ có xuất bánh đi các chi nhanh khác khi học thiếu bánh haocjw nhà cung cấp giao bánh không kịp cho họ. Chúng ta sẽ lưu những dữ liệu giống như nhập bánh nhưng không có phần tiền.

Các chi nhánh cũng được lưu các thông tin như nhà cung cấp.

# Bảng

1. Bảng "DoiTacNhap" (Suppliers):

- Lưu trữ thông tin về các đối tác cung cấp hàng hóa cho cửa hàng.

- Mỗi đối tác có tên ("TenDoiTac"), số điện thoại ("SDT\_benNhap") và ghi chú ("GhiChu").

2. Bảng "DoiTacXuat" (Customers):

- Lưu trữ thông tin về các đối tác nhận hàng từ cửa hàng.

- Mỗi đối tác có tên ("TenDoiTac") và số điện thoại ("SDT\_benXuat").

3. Bảng "LoaiSanPham" (Product Categories):

- Lưu trữ thông tin về các loại sản phẩm.

- Mỗi loại sản phẩm có tên ("TenLoai"), mô tả ("MoTa") và hình ảnh ("HinhAnh").

4. Bảng "LoaiBienNhan" (Receipt Types):

- Lưu trữ thông tin về các loại biên nhận.

- Mỗi loại biên nhận có ID ("ID\_LoaiBienNhan") và tên loại ("TenLoai").

5. Bảng "LoaiBillXuat" (Export Bill Types):

- Lưu trữ thông tin về các loại hóa đơn xuất hàng.

- Mỗi loại hóa đơn xuất hàng có tên loại ("LoaiBill").

6. Bảng "Permission" (Permissions):

- Lưu trữ thông tin về các quyền hạn trong hệ thống.

- Mỗi quyền hạn có tên ("NameOfPermission"), ngày bắt đầu ("From") và ngày kết thúc ("To").

7. Bảng "SanPham" (Products):

- Lưu trữ thông tin về các sản phẩm trong cửa hàng.

- Mỗi sản phẩm có ID ("ID\_SanPham"), tên sản phẩm ("TenSanPham"), số lượng ("SoLuong"), giá ("Gia"), trạng thái sự kiện ("Event"), đơn vị tính ("DonViTinh"), giảm giá ("GiamGia"), trạng thái ("TrangThai"), ghi chú ("GhiChu") và thuộc loại sản phẩm ("TenLoai").

- Có quan hệ ngoại khoá với bảng "LoaiSanPham" thông qua trường "TenLoai".

8. Bảng "NhapBill" (Import Bills):

- Lưu trữ thông tin về các hóa đơn nhập hàng.

- Mỗi hóa đơn nhập hàng có ID ("ID\_bill"), ngày nhập hàng ("NgayNhap"), tổng giá trị hóa đơn ("TongBill"), ghi chú ("GhiChu") và số điện thoại của đối tác nhập hàng ("SDT\_benNhap").

- Có quan hệ ngoại khoá với bảng "DoiTacNhap" thông qua trường "SDT\_benNhap".

9. Bảng "NhapSanPham" (Imported Products):

- Lưu trữ thông tin về các sản phẩm đã nhập trong hóa đơn nhập hàng.

- Mỗi sản phẩm đã nhập có giá ("Gia"), số lượng ("SoLuong"), ghi chú ("GhiChu"), ID hóa đơn nhập hàng ("ID\_bill") và ID sản phẩm ("ID\_SanPham").

- Có quan hệ ngoại khoá với bảng "NhapBill" thông qua trường "ID\_bill" và với bảng "SanPham" thông qua trường "ID\_SanPham".

10. Bảng "BienNhan" (Receipts):

- Lưu trữ thông tin về các biên nhận hàng.

- Mỗi biên nhận hàng có ID ("ID\_bienNhan"), số tiền cọc ("TienCoc"), ghi chú ("GhiChu"), ID sản phẩm ("ID\_SanPham") và ID loại biên nhận ("ID\_LoaiBienNhan").

- Có quan hệ ngoại khoá với bảng "SanPham" thông qua trường "ID\_SanPham" và với bảng "LoaiBienNhan" thông qua trường "ID\_LoaiBienNhan".

11. Bảng "HinhAnh" (Images):

- Lưu trữ thông tin về hình ảnh của sản phẩm.

- Mỗi hình ảnh có đường dẫn ("Link") và ID sản phẩm ("ID\_SanPham").

- Có quan hệ ngoại khoá với bảng "SanPham" thông qua trường "ID\_SanPham".

12. Bảng "Roles" (Roles):

- Lưu trữ thông tin về các vai trò trong hệ thống.

- Mỗi vai trò có tên vai trò ("TenRole") và tên quyền hạn ("NameOfPermission").

- Có quan hệ ngoại khoá với bảng "Permission" thông qua trường "NameOfPermission".

13. Bảng "NhanVien" (Employees):

- Lưu trữ thông tin về các nhân viên trong cửa hàng.

- Mỗi nhân viên có mã số nhân viên ("MaSoNhanVien"), mật khẩu ("password"), tên vai trò ("TenRole").

- Có quan hệ ngoại khoá với bảng "Roles" thông qua trường "TenRole".

14. Bảng "XuatBill" (Export Bills):

- Lưu trữ thông tin về các hóa đơn xuất hàng.

- Mỗi hóa đơn xuất hàng có ID ("ID\_bill"), ngày xuất hàng ("NgayXuat"), tổng giá trị hóa đơn ("TongBill"), thông tin thẻ chặn ("TheChan"), ghi chú ("GhiChu"), số điện thoại của đối tác xuất hàng ("SDT\_benXuat"), ID biên nhận ("ID\_bienNhan"), mã số nhân viên ("MaSoNhanVien") và loại hóa đơn ("LoaiBill").

- Có quan hệ ngoại khoá với bảng "DoiTacXuat" thông qua trường "SDT\_benXuat", với bảng "BienNhan" thông qua trường "ID\_bienNhan", với bảng "NhanVien" thông qua trường "MaSoNhanVien" và với bảng "LoaiBillXuat" thông qua trường "LoaiBill".

15. Bảng "XuatSanPham" (Exported Products):

- Lưu trữ thông tin về các sản phẩm đã xuất trong hóa đơn xuất hàng.

- Mỗi sản phẩm đã xuất có số lượng ("SoLuong"), giá ("Gia"), ID sản phẩm ("ID\_SanPham") và ID hóa đơn xuất hàng ("ID\_bill").

- Có quan hệ ngoại khoá với bảng "SanPham" thông qua trường "ID\_SanPham" và với bảng "XuatBill" thông qua trường "ID\_bill".

16. Bảng "DeletedBill" (Deleted Bills):

- Lưu trữ thông tin về các hóa đơn đã bị xóa.

- Mỗi hóa đơn đã bị xóa có ngày hủy ("NgayHuy"), tổng giá trị hóa đơn ("TongBill"), ghi chú ("GhiChu") và ID hóa đơn ("ID\_bill").

- Có quan hệ ngoại khoá với bảng "XuatBill" thông qua trường "ID\_bill".

Đoạn mã tạo các bảng trên:

USE master

GO

CREATE DATABASE ABC\_bakery

GO

USE ABC\_bakery

GO

CREATE TABLE categories

(

id BIGINT NOT NULL IDENTITY(1,1),

name NVARCHAR(255) NOT NULL ,

prefix NVARCHAR(5) NOT NULL DEFAULT 'TL'

PRIMARY KEY (id)

);

CREATE TABLE products

(

id BIGINT NOT NULL IDENTITY(1, 1),

category\_id BIGINT NOT NULL ,

name nvarchar(255) NOT NULL ,

price float NOT NULL ,

is\_active bit NOT NULL ,

amount integer NOT NULL ,

expiration\_date date NOT NULL ,

created\_at date NOT NULL ,

updated\_at date NOT NULL ,

prefix NVARCHAR(5) NOT NULL DEFAULT 'SP'

PRIMARY KEY (id),

CONSTRAINT FK\_1 FOREIGN KEY (category\_id) REFERENCES categories(id)

);

CREATE TABLE images

(

id BIGINT NOT NULL IDENTITY(1, 1),

product\_id BIGINT NOT NULL ,

url VARCHAR(512) NOT NULL ,

PRIMARY KEY (id),

CONSTRAINT FK\_2 FOREIGN KEY (product\_id) REFERENCES products(id)

);

CREATE TABLE suppliers

(

id BIGINT NOT NULL IDENTITY(1,1),

name NVARCHAR(255) NOT NULL,

phone\_number VARCHAR(10) NOT NULL,

address NVARCHAR(255) NOT NULL,

created\_at DATE NOT NULL DEFAULT GETDATE(),

updated\_at DATE NOT NULL DEFAULT GETDATE(),

prefix NVARCHAR(5) NOT NULL DEFAULT 'KH' PRIMARY KEY (id)

);

CREATE TABLE roles

(

id bigint NOT NULL ,

name varchar NOT NULL ,

PRIMARY KEY (id)

);

CREATE TABLE permissions

(

id BIGINT NOT NULL IDENTITY(0, 1),

allow\_access VARCHAR(255) NOT NULL, PRIMARY KEY (id)

);

CREATE TABLE role\_permission

(

permission\_id BIGINT NOT NULL,

role\_id BIGINT NOT NULL, PRIMARY KEY (permission\_id, role\_id), CONSTRAINT FK\_3 FOREIGN KEY (permission\_id) REFERENCES PERMISSIONS(id), CONSTRAINT FK\_4 FOREIGN KEY (role\_id) REFERENCES roles(id)

);

CREATE TABLE users

(

id BIGINT NOT NULL IDENTITY(1, 1),

role\_id BIGINT NOT NULL ,

name NVARCHAR(255) NOT NULL ,

created\_at date NOT NULL DEFAULT GETDATE(),

updated\_at date NOT NULL DEFAULT GETDATE(),

PRIMARY KEY (id),

CONSTRAINT FK\_7 FOREIGN KEY (role\_id) REFERENCES roles(id)

);

CREATE TABLE imports

(

id BIGINT NOT NULL IDENTITY(1, 1),

supplier\_id BIGINT NOT NULL ,

admin\_id BIGINT NOT NULL ,

total integer NOT NULL ,

created\_at date NOT NULL ,

is\_inter bit NOT NULL ,

prefix NVARCHAR(5) NOT NULL DEFAULT 'D'

PRIMARY KEY (id),

CONSTRAINT FK\_5 FOREIGN KEY (supplier\_id) REFERENCES suppliers(id),

CONSTRAINT FK\_6 FOREIGN KEY (admin\_id) REFERENCES users(id)

);

CREATE TABLE import\_product

(

import\_id BIGINT NOT NULL ,

product\_id BIGINT NOT NULL ,

price float NOT NULL ,

amount integer NOT NULL ,

PRIMARY KEY (import\_id, product\_id),

CONSTRAINT FK\_8 FOREIGN KEY (import\_id) REFERENCES imports(id),

CONSTRAINT FK\_9 FOREIGN KEY (product\_id) REFERENCES products(id)

);

CREATE TABLE promotions

(

id BIGINT NOT NULL IDENTITY(1,1),

name NVARCHAR(255) NOT NULL,

is\_active BIT NOT NULL,

created\_at DATE NOT NULL DEFAULT GETDATE(),

code NVARCHAR(6) NOT NULL,

prefix NVARCHAR(5) NOT NULL DEFAULT 'KM',

ratio INT NOT NULL,

PRIMARY KEY (id),

CONSTRAINT CK\_1 CHECK (ratio BETWEEN 0 AND 101)

);

CREATE TABLE receipts

(

id BIGINT NOT NULL,

name NVARCHAR(255) NOT NULL,

receieved FLOAT NOT NULL,

expense FLOAT NOT NULL,

created\_at DATE NOT NULL DEFAULT GETDATE(),

prefix NVARCHAR(5) NOT NULL DEFAULT 'BN', PRIMARY KEY (id)

);

CREATE TABLE orders

(

id BIGINT NOT NULL IDENTITY(1,1),

promotion\_id BIGINT NOT NULL,

receipt\_id BIGINT NOT NULL,

cashier\_id BIGINT NOT NULL,

price FLOAT NOT NULL,

name NVARCHAR(255) NOT NULL,

address NVARCHAR(512) NOT NULL,

status INTEGER NOT NULL,

type INTEGER NOT NULL,

total INTEGER NOT NULL,

ship\_fee FLOAT NOT NULL,

payment\_method INTEGER NOT NULL,

is\_paid BIT NOT NULL,

prepay FLOAT NOT NULL,

ordered\_at DATE NOT NULL,

note TEXT NOT NULL,

change INTEGER NOT NULL,

receieved INTEGER NOT NULL,prefix NVARCHAR(5) NOT NULL DEFAULT 'HD', PRIMARY KEY (id), CONSTRAINT FK\_10 FOREIGN KEY (promotion\_id) REFERENCES promotions(id), CONSTRAINT FK\_11 FOREIGN KEY (receipt\_id) REFERENCES receipts(id), CONSTRAINT FK\_12 FOREIGN KEY (cashier\_id) REFERENCES users(id),

);

CREATE TABLE order\_detail

(

order\_id BIGINT NOT NULL IDENTITY(1,1),

product\_id BIGINT NOT NULL,

price FLOAT NOT NULL,

total INTEGER NOT NULL, PRIMARY KEY (order\_id, product\_id), CONSTRAINT FK\_13 FOREIGN KEY (order\_id) REFERENCES orders(id), CONSTRAINT FK\_14 FOREIGN KEY (product\_id) REFERENCES products(id));

# Các khóa chính, khóa ngoại

* Khóa chính (primary key):
  + Bảng "DoiTacNhap": "ID\_DoiTacNhap"
  + Bảng "DoiTacXuat": "SDT\_benXuat"
  + Bảng "LoaiSanPham": "TenLoai"
  + Bảng "LoaiBienNhan": "ID\_LoaiBienNhan"
  + Bảng "Permission": "NameOfPermission"
  + Bảng "SanPham": "ID\_SanPham"
  + Bảng "NhapBill": "ID\_bill"
  + Bảng "NhapSanPham": "ID\_bill", "ID\_SanPham"
  + Bảng "BienNhan": "ID\_bienNhan"
  + Bảng "HinhAnh": "ID\_SanPham"
  + Bảng "Roles": "TenRole"
  + Bảng "NhanVien": "MaSoNhanVien"
  + Bảng "XuatBill": "ID\_bill"
  + Bảng "XuatSanPham": "ID\_bill", "ID\_SanPham"
  + Bảng "DeletedBill": "ID\_bill"
* Khóa ngoại (foreign key):
  + Bảng "NhapSanPham": "ID\_SanPham" → Bảng "SanPham"
  + Bảng "NhapBill": "SDT\_benNhap" → Bảng "DoiTacNhap"
  + Bảng "BienNhan": "ID\_SanPham" → Bảng "SanPham"
  + Bảng "BienNhan": "ID\_LoaiBienNhan" → Bảng "LoaiBienNhan"
  + Bảng "XuatSanPham": "ID\_SanPham" → Bảng "SanPham"
  + Bảng "XuatBill": "SDT\_benXuat" → Bảng "DoiTacXuat"
  + Bảng "XuatBill": "ID\_bienNhan" → Bảng "BienNhan"
  + Bảng "XuatBill": "MaSoNhanVien" → Bảng "NhanVien"
  + Bảng "XuatBill": "LoaiBill" → Bảng "LoaiBillXuat"
  + Bảng "DeletedBill": "ID\_bill" → Bảng "XuatBill"

# Store Procedure, Function, Trigger

## Trigger

* Trigger AutoIncrementProductID

Chức năng: Tự động tăng giá trị của khóa chính ID cho bảng products khi có một sản phẩm mới được thêm vào.

Code:

DECLARE @NextProductID INT;

SELECT @NextProductID = ISNULL(MAX(ID), 0) + 1

FROM products;

CREATE TRIGGER AutoIncrementProductID

ON products

AFTER INSERT

AS

BEGIN

UPDATE p

SET ID = @NextProductID, @NextProductID = @NextProductID + 1

FROM products p

INNER JOIN inserted i ON p.ID IS NULL;

END;

* Trigger AutoIncrementImageID

Chức năng: Tự động tăng giá trị của khóa chính ID cho bảng images khi có một hình ảnh mới được thêm vào.

Code:

DECLARE @NextImageID INT;

SELECT @NextImageID = ISNULL(MAX(ID), 0) + 1

FROM images

WHERE ID LIKE 'IMG%';

CREATE TRIGGER AutoIncrementImageID

ON images

AFTER INSERT

AS

BEGIN

UPDATE i

SET ID = 'IMG' + CAST(@NextImageID AS NVARCHAR(50)), @NextImageID = @NextImageID + 1

FROM images i

INNER JOIN inserted ins ON i.ID IS NULL AND ins.ID IS NOT NULL;

END;

* Trigger AutoIncrementOrderID

Chức năng: Tự động tăng giá trị của khóa chính ID cho bảng orders khi có một đơn hàng mới được thêm vào.

Code:

DECLARE @NextOrderID INT;

SELECT @NextOrderID = ISNULL(MAX(ID), 0) + 1

FROM orders

WHERE ID LIKE 'ORDER\_%';

CREATE TRIGGER AutoIncrementOrderID

ON orders

AFTER INSERT

AS

BEGIN

UPDATE o

SET ID = 'ORDER\_' + CAST(@NextOrderID AS NVARCHAR(50)), @NextOrderID = @NextOrderID + 1

FROM orders o

INNER JOIN inserted ins ON o.ID IS NULL AND ins.ID IS NOT NULL;

END;

## Function

* Function GetProductsCreatedBetween

Chức năng: Hàm này trả về một bảng chứa thông tin của tất cả các sản phẩm được tạo trong khoảng thời gian từ @StartDate đến @EndDate.

Code:

CREATE FUNCTION [dbo].[GetProductsCreatedBetween](@StartDate DATE, @EndDate DATE)

RETURNS TABLE

AS

RETURN

(

SELECT products.id as ID, products.name AS TenBanh, categories.name AS Loai, products.amount as Ton

FROM products

INNER JOIN categories ON products.category\_id = categories.id

where products.created\_at between @StartDate and @EndDate

);

GO

## Store Procedure

* Stored Procedure ImportInToNhapBanhNoiBo

Chức năng: Nhập thông tin sản phẩm vào bảng nhập hàng

Tham số đầu vào: Không có

Giá trị trả về: Không có

Code:

USE [ABC\_bakery]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ImportInToNhapBanhNoiBo] Script Date: 10/30/2023 9:39:55 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[ImportInToNhapBanhNoiBo]

AS

BEGIN

SELECT id AS MaBanh, name AS TenBanh, amount AS Ton, amount AS SoLuong, price AS Gia

FROM products;

END;

GO

* Stored Procedure ShowOrderDetailsAndOrders

Chức năng: Lấy danh sách tất cả các chi tiết đơn hàng và đơn hàng

Tham số đầu vào: Không có

Giá trị trả về: Danh sách tất cả các chi tiết đơn hàng và đơn hàng

Code:

CREATE PROCEDURE ShowOrderDetailsAndOrders

AS

BEGIN

SELECT od.\*, o.\*

FROM order\_detail od

JOIN orders o ON od.order\_id = o.id;

END;

* Stored Procedure ShowAllSuppliers

Chức năng: Lấy danh sách tất cả các nhà cung cấp

Tham số đầu vào: Không có

Giá trị trả về: Danh sách tất cả các nhà cung cấp

Code:

CREATE PROCEDURE ShowAllSuppliers

AS

BEGIN

SELECT \* FROM suppliers;

END;