

**Discount (level, Discount\_Rate)**

create table Discount(

Level nvarchar(10) not null primary key,

Discount\_Rate int,

)

==>Điểm tích lũy theo bậc

Các cập bậc: Đồng: 10%. Bạc:12%. Vàng:15%. Bạch kim:20%

**Customer (name, Phone, Address, points, level)**

create table Customer(

ID int identity(1,1) not null,

FullName nvarchar(max),

Phone nvarchar(12) not null primary key,

Address nvarchar(max),

Points int,

Level nvarchar(10),

Constraint FK\_Customer\_Discount foreign key (Level) references Discount (Level)

)

=>Point là tính theo 10% bill của khách hàng trước khi tính chiết khấu(cộng dồn)

Level dựa vào point khách hàng đạt được

Đồng:100-499. Bạc: 500-699. Vàng: 700-999. Bạch kim: >=1000

**Staff (staff id, fullname, DoB, phone, Address, position)**

create table Staff(

Id\_Staff nvarchar(50) not null primary key,

FullName nvarchar(max),

DoB date,

Phone nvarchar (12),

Address nvarchar(max),

position nvarchar(50)

)

**==>Có 2 chức vụ là thu ngân và Pha chế**

**Thu ngân có tài khoản vào hệ thống để thanh toán đơn hàng cho khách**

Manager( manager id, fullname, DoB, Phone, Address)

create table Manager(

Id\_Manager nvarchar(50) not null primary key,

FullName nvarchar(max),

DoB date,

Phone nvarchar (12),

Address nvarchar(max)

)

==>Manager là chủ của cửa hàng

Manager là người có quyền trên tất cả, thêm, sửa , xóa... và là người tính lương cho nhân viên và doanh thu của cửa hàng

**Accounts (id\_Account,username, pass, id\_user, role)**

create table Accounts(

Id\_Account int identity(1,1) not null primary key,

UserName nvarchar(max),

PassWord nvarchar(max),

Id\_User nvarchar(50) not null,

Role nvarchar(50),

Constraint FK\_Accounts\_Staff foreign key (Id\_User) references Staff (Id\_Staff),

Constraint FK\_Accounts\_Manager foreign key (Id\_User) references Manager (Id\_Manager)

)

==>Tài khoản để đăng nhập vào hệ thống gồm tài khoản của manager và của nhân viên thu ngân để tính tiền cho khách

**Menu (Pro\_Id, Name, image, F&B, Category, Price, Status)**

create table Menu(

Id\_Product nvarchar(50) not null primary key,

Name nvarchar(max),

Image nvarchar(max),

FandB nvarchar(50),

Category nvarchar(100),

Price float,

Status bit

)

==>FandB là phân loại đồ ăn hay thức uống. Category là các loại nước (trà cà phê..)hay thức ăn...

Status để trạng thái còn hoặc hết hàng

**Temp (id\_order,product, quantity, price)**

create table Temp(

ID\_Order int not null,

ID\_Product nvarchar(50) not null,

Quantity int,

Price Float,

constraint PK\_Temp primary key (ID\_Order,ID\_Product),

Constraint FK\_Temp\_Menu foreign key (ID\_Product) references Menu (Id\_Product)

)

==>Chức năng giống như một giỏ hàng dùng để lưu các món khách đang order. Khi khách thanh toán thì sẽ tạo bill(ID\_Order là id cuối của bill+1) và bảng temp sẽ được chuyển sang bảng OrderDetail xóa dữ liệu trong bảng temp.

**Bill (id, phone, price, discount, total, Staff\_id, Date\_export)**

create table Bill(

ID int identity(1,1) not null primary key,

Phone nvarchar(12),

Price float,

Discount Float,

Total float,

Id\_Staff nvarchar(50),

Date\_export date,

Constraint FK\_Bill\_Customer foreign key (Phone) references Customer (Phone),

Constraint FK\_Bill\_Staff foreign key (Id\_Staff) references Staff (Id\_Staff)

)

==>Khi khách hàng thanh toán sẽ tiến hành tạo bill và id bill sẽ là id tự tăng, bill không được xóa

**OrderDetail (id, product, quantity, price)**

create table OrderDetail(

ID\_Order int not null,

ID\_Product nvarchar(50) not null,

Quantity int,

Price Float,

constraint PK\_OrderDetail primary key (ID\_Order,ID\_Product),

Constraint FK\_OrderDetail\_Bill foreign key (ID\_Order) references Bill (ID),

Constraint FK\_OrderDetail\_Menu foreign key (ID\_Product) references Menu (Id\_Product)

)

==>hiển thị chi tiết các món khách đã order

**KPI (staff\_id, Date, target, Result, Bonus)**

create table KPI(

Id\_Staff nvarchar(50) not null,

PayDay date not null,

Target float,

Result float,

Bonus float

constraint PK\_KPI primary key (Id\_Staff,PayDay)

)

==>Dùng để tính tiền KPI cho nhân viên dựa trên 10% theo số tiền nhân viên vượt qua chỉ tiêu 10%(result-target)

**Salary (staff\_id, Payday, Basic\_Salary, KPI, Fine, Salary)**

create table Salary(

Id\_Staff nvarchar(50) not null,

PayDay date not null,

Basic\_Salary float,

KPI float,

Fine float,

Salary float

constraint PK\_Salary primary key (Id\_Staff,PayDay)

)

==>Bảng tính lương

**Turnover (Date, earnings, Outgoing, profit)**

create table Turnover(

Date date not null primary key,

Earnings float,

Outgoing float,

Profit float

)

==>Tính doanh thu cho cửa hàng

MÔ TẢ

**BẢNG MENU**

-Procedure thêm món

-procedure update thông tin món / update category

-procedure xóa món

**BẢNG BILL/TEMP...**

**-**Trigger tính price trong temp

-Procedure Thêm món vào temp (id=id bill +1)

-Procedure xóa món trong temp

-Procedure update số lượng món trong temp (trigger price)

-Thanh toán=> Tạo procedure thêm vào bill, procedure chuyển bảng temp vào bảng orderdetail sau đó xóa bảng tenp

-Thanh toán => procedure update chiết khấu của khách ( trigger update level)

=>Procedure update Result cho nhân viên ( trigger update Bonus= (Result-Target)\*10%)

-Procedure thêm khách hàng ( nếu chưa có khách hàng đó)

-Procedure tính tổng bill

-Procedure update bảng level

**BẢNG ACCOUNT & BẢNG CUSTOMER/STAFF**

-Procedure đổi mật khẩu => sửa sp\_login

-Procedure đổi thông tin cá nhân

-Procedure xóa nhân viên => xóa account => xóa login => xóa user

-Procedure Thêm nhân viên => Thêm account => Thêm login => Thêm user

-Procedure Thêm khách hàng

**BẢNG LƯƠNG/KPI...**

-View lấy bảng lương

-View bảng doanh số

-Trigger tồn=thu-chi

-trigger lương=lương cơ bản + KPI -Fine

-Function lấy bảng KPI

-Procedure thêm vào bảng lương

-procedure thêm bảng doanh số