TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

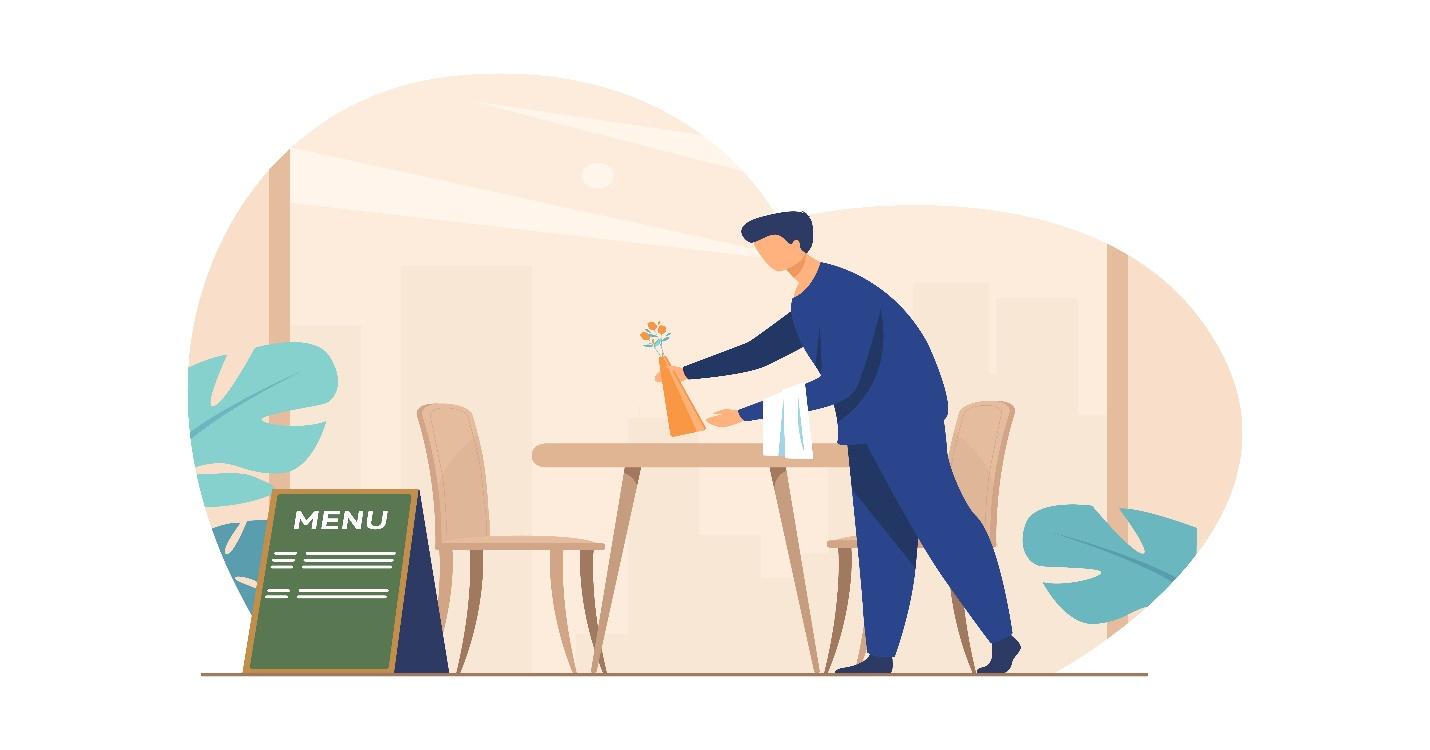
KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo bộ môn

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**QUẢN LÝ NHÀ HÀNG**

****

**Giáo Viên: TS. Nguyễn Thành Sơn**

**Sinh viên thực hiện*:***

|  |
| --- |
| **Khổng Gia Bình                       18110255**  **Lê Ngọc Đoan                           17133014**  **Nguyễn Thị Minh Hoàng         18110285**  **Lê Thị Phương Thảo                18110366** |

Mục lục

[**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN** 5](#_Toc60732873)

[1. Tổng quan về phần mềm 8](#_Toc60732874)

[1.1 Mục tiêu 8](#_Toc60732875)

[1.2 Phạm vi 8](#_Toc60732876)

[1.3 Mô tả 8](#_Toc60732877)

[1.4 Thông tin của phần mềm 8](#_Toc60732878)

[1.5 Chức năng của phần mềm 9](#_Toc60732879)

[1.5.1 Tạo, quản lý tài khoản (thêm, xóa, sửa). 9](#_Toc60732880)

[1.5.2 Đăng nhập. 9](#_Toc60732881)

[1.5.3 Quản lý bàn ăn (tình trạng trống, đầy). 9](#_Toc60732882)

[1.5.4 Bán hàng: tạo, sửa, thanh toán hóa đơn. 9](#_Toc60732883)

[1.5.5 Quản lý thông tin khách hàng, nhân viên. 9](#_Toc60732884)

[1.5.6 Quản lý loại món ăn, món ăn (thêm, xóa, sửa, tình trạng loại món ăn, món ăn). 9](#_Toc60732885)

[1.5.7 Thống kê: Tổng số doanh thu, số lượng hóa đơn, thanh toán. 9](#_Toc60732886)

[2. Thiết kế cơ sở dữ liệu 9](#_Toc60732887)

[2.1 Mô hình ERD 9](#_Toc60732888)

[2.2 Lược đồ quan hệ 10](#_Toc60732889)

[2.3 Thiết kế cơ sở dữ liệu mức chi tiết 11](#_Toc60732890)

[2.4 Lược đồ quan hệ 17](#_Toc60732891)

[3. Triển khai csdl (đổi tên lại cho pro) 17](#_Toc60732892)

[3.1 View 17](#_Toc60732893)

[create view getallBill select \* from Bill 17](#_Toc60732894)

[3.2 Trigger 18](#_Toc60732895)

[3.2.1 Trigger kiểm tra khi thay đổi trạng thái của bàn 18](#_Toc60732896)

[3.2.2 Xóa thông tin đặt món sau khi thanh toán 18](#_Toc60732897)

[3.2.3 Thay đổi trạng thái bàn sau khi xóa một order 19](#_Toc60732898)

[3.2.4 Bản doanh thu mỗi khi có bill được thanh toán 19](#_Toc60732899)

[3.2.5 Update ngưng bán các món ăn của loại món ăn nếu loại món ăn đó ngưng phục vụ 19](#_Toc60732900)

[3.2.6 Kiểm tra các món ăn bị disable có ảnh hưởng đến tình trạng của loại món ăn hay không 20](#_Toc60732901)

[3.3 Stored Procedure (Chức năng) 20](#_Toc60732902)

[3.3.1 Kiểm tra trạng thái tài khoản 20](#_Toc60732903)

[3.3.2 Lấy level của account 21](#_Toc60732904)

[3.3.3 Disable bàn ăn 21](#_Toc60732905)

[3.3.4 Enable lại bàn ăn 21](#_Toc60732906)

[3.3.5 Thay đổi state của seat 21](#_Toc60732907)

[3.3.6 Thêm 1 bàn 22](#_Toc60732908)

[3.3.7 Chuyển bàn 22](#_Toc60732909)

[3.3.8 Lấy order detail thông qua id bàn 22](#_Toc60732910)

[3.3.9 Update order detail 22](#_Toc60732911)

[3.3.10 Thêm item cho orderDetail 23](#_Toc60732912)

[3.3.11 Xóa 1 order detail sau khi gộp bàn 23](#_Toc60732913)

[3.3.12 Tạo order 23](#_Toc60732914)

[3.3.13 Tra cứu dữ liệu (lấy dữ liệu dòng cuối cùng của bảng order) 23](#_Toc60732915)

[3.3.14 Lấy dữ liệu của một order cụ thể 24](#_Toc60732916)

[3.3.15 Validate order mới 24](#_Toc60732917)

[3.3.16 Chỉnh sửa tên 1 khu vực bàn ăn 24](#_Toc60732918)

[3.3.17 Cập nhập lại idCustomer khi nhấn vào bếp 24](#_Toc60732919)

[3.3.18 Lấy dữ liệu lên từ các table 25](#_Toc60732920)

[3.3.19 Thêm nhân viên 26](#_Toc60732921)

[3.3.20 Cập nhật nhân viên 27](#_Toc60732922)

[3.3.21 Thêm khách hàng 27](#_Toc60732923)

[3.3.22 Lấy danh sách khách hàng theo ID 27](#_Toc60732924)

[3.3.23 Lấy điểm tích lũy của khach hàng 28](#_Toc60732925)

[3.3.24 Lấy thông tin khách hàng bằng ID 28](#_Toc60732926)

[3.3.25 Lấy danh sách các món đã được order của một khách hàng 28](#_Toc60732927)

[3.3.26 Thêm mới các món ăn trong chi tiết món 28](#_Toc60732928)

[3.3.27 Lấy tất cả thông tin của orderDetail bằng idOrder 29](#_Toc60732929)

[3.3.28 Lấy nhân viên bằng ID 29](#_Toc60732930)

[3.3.29 Lấy tên món ăn bằng id 29](#_Toc60732931)

[3.3.30 Update số lượng đặt món 29](#_Toc60732932)

[3.3.31 Thêm bill mới 30](#_Toc60732933)

[3.3.32 Best seller 30](#_Toc60732934)

[3.3.33 Tìm kiếm nhân viên 31](#_Toc60732935)

[3.3.34 Kiểm tra username có tồn tại 32](#_Toc60732936)

[3.3.35 Load loại sản phẩm 32](#_Toc60732937)

[3.3.36 Lấy loại món ăn bằng tên 33](#_Toc60732938)

[3.3.37 Lấy loại món ăn bằng tên và trạng thái 33](#_Toc60732939)

[3.3.38 Lấy danh sách món ăn của một loại món ăn cụ thể 33](#_Toc60732940)

[3.3.39 Chỉnh sửa loại món ăn 34](#_Toc60732941)

[3.3.40 Lấy thông tin cụ thể của một hóa đơn 34](#_Toc60732942)

[3.3.41 Lấy các món ăn của một hóa đơn 34](#_Toc60732943)

[3.3.42 Thêm một loại món ăn mới 35](#_Toc60732944)

[3.3.43 Thêm sự kiện khuyến mãi 35](#_Toc60732945)

[3.3.44 Chỉnh sửa sự kiện 35](#_Toc60732946)

[3.3.45 Thêm mới một khách hàng 35](#_Toc60732947)

[3.3.46 Chinh sửa thông tin khách hàng 35](#_Toc60732948)

[3.3.47 Thêm món ăn 36](#_Toc60732949)

[3.3.48 Chỉnh sửa món ăn 36](#_Toc60732950)

[3.4 Function 36](#_Toc60732951)

[3.4.1 Lấy danh sách nhân viên bằng id 36](#_Toc60732952)

[3.4.2 Lấy số lượng món ăn 37](#_Toc60732953)

[3.4.3 Lấy value discount trong Events 37](#_Toc60732954)

[3.4.4 Thống kê doanh thu,... theo loại thức ăn 37](#_Toc60732955)

[3.4.5 Lấy danh sách tất cả cả Bill 38](#_Toc60732956)

[3.4.6 Tìm Bill (hóa đơn) theo khoản thời gian 38](#_Toc60732957)

[3.4.7 Tìm kiếm hóa đơn theo trạng thái (hoàn thành- hủy) 38](#_Toc60732958)

[3.4.8 Tìm danh sách hóa đơn theo tên khách hàng 39](#_Toc60732959)

[3.4.9 Tìm kiếm danh sách hóa đơn theo tên nhân viên thu ngân 39](#_Toc60732960)

[3.4.10 Lấy danh sách các hóa đơn theo ID 39](#_Toc60732961)

[3.4.11 Thống kê doanh thu,... theo loại thứ ăn 40](#_Toc60732962)

[3.4.12 Lấy doanh thu theo ngày 40](#_Toc60732963)

[3.4.13 Lấy doanh thu theo khoảng thời gian 40](#_Toc60732964)

[3.4.14 Lấy loại món ăn 41](#_Toc60732965)

[3.5 Phân quyền 41](#_Toc60732966)

[3.5.1 Tạo user và role 41](#_Toc60732967)

[3.5.2 Phân quyền Chủ cửa hàng 42](#_Toc60732968)

[3.5.3 Phân quyền nhân viên 45](#_Toc60732969)

[4. Giao diện 47](#_Toc60732970)

[4.1 Kết nối 47](#_Toc60732971)

[4.2 Thực thi 48](#_Toc60732972)

[Đối với câu truy vấn có lấy dữ liệu 48](#_Toc60732973)

[Truy vấn không lấy dữ liệu trả về 50](#_Toc60732974)

[4.3 Mô hình triển khai 50](#_Toc60732975)

[5. Kết luận 51](#_Toc60732976)

[5.1 Những điều đã đạt được 51](#_Toc60732977)

[5.2 Hạn chế 51](#_Toc60732978)

[5.3 Những procedure, function đặc biệt 51](#_Toc60732979)

[5.3.1 Trigger cho bản doanh thu mỗi khi có bill được thanh toán 51](#_Toc60732980)

[5.3.2 Hàm lấy dữ liệu dùng chung 52](#_Toc60732981)

[5.3.3 Kiểm tra Order cuối 53](#_Toc60732982)

[5.3.4 Tìm kiếm nhân viên theo các yêu cầu có sẵn 55](#_Toc60732983)

[▪ 55](#_Toc60732984)

**Phó trưởng khoa: Lê Văn Vinh TS. Nguyễn Thành Sơn**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

----\*\*\*----

Tp. Hồ Chí Minh, ngày 17 tháng 11 năm 2019

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Nhóm 11

Ngành:

Tên đề tài:

Họ và tên Giáo viên hướng dẫn:

**NHẬN XÉT**

1. Về nội dung đề tài và khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm: (Bằng chữ: )

Tp. Hồ Chí Minh, ngày   tháng   năm 20

Giáo viên hướng dẫn

LỜI CAM KẾT

1. Những nội dung trong luận văn này là do nhóm tôi thực hiện dưới sự hướng dẫn trực tiếp của thầy Nguyễn Thành Sơn.

2. Mọi tham khảo dùng trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

1. Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.

Hồ Chí Minh, ngày 25 tháng 12 năm 2020

Lê Thị Phương Thảo

# Tổng quan về phần mềm

## Mục tiêu

Ngày nay với sự phát triển nhanh chóng của công nghệ, nó dường như đã có mặt trong mọi lĩnh vực và ngành nghề ,đem lại ứng dụng to lớn. Đặc biệt trong lĩnh vực quản lí, chúng ta không thể tiếp tục sử dụng cách thủ công -ghi chép để quản lí mọi thứ vì lượng dữ liệu và nhu cầu đang ngày càng to lớn. Phần mềm quản lí nhà hàng của nhóm chúng em cùng với mục tiêu đó là giúp việc quản lí trở nên đơn giản và dễ dàng hơn rất nhiều. Phần mềm quản lí nhà hàng nhằm không những giúp cho việc chọn món ăn để phục vụ ,thanh toán hóa đơn một cách tiện lợi nhanh chóng mà còn giúp quản lí những món ăn có trong nhà hàng, quản lí nhân viên, hóa đơn, khách hàng , doanh thu nhà hàng… một cách dễ dàng với giao diện thân thiện, dễ sử dụng đối với nhân viên trong việc phục vụ đặt món của khách hàng và việc quản lí các mảng trong nhà hàng của chủ cửa hàng.

## Phạm vi

Phần mềm quản lí nhà hàng không chỉ áp dụng trong pham vi cho các mô hình cửa hàng bán thức ăn (nhà hàng, quán ăn, ….) mà có thể thay đổi nhanh chóng, áp dụng cho các loại hình kinh doanh hàng hóa đa dạng khác (cửa hàng quần áo, đồ dùng gia đình, … ).

Phần mềm có thể sử dụng riêng lẻ (cá nhân) hoặc nhiều người dùng trên nhiều máy tính khác nhau (kết nối thông quan mạng nội bộ, cùng tương tác sử dụng phần mềm).

## Mô tả

Trong nhà hàng, mỗi món ăn sẽ có một mã duy nhất để phân biệt với các món ăn khác, mỗi món sẽ thuộc một loại món ăn, ngoài ra còn có giá ,tên món, trạng thái còn bán hay ngừng bán. Mỗi loại món cũng có một mã duy nhất , gồm nhiều món ăn, ngoài ra còn có tên món và trạng thái còn bán hay ngừng bán. Trong nhà hàng sẽ có nhiều bàn ở nhiều vị trí, mỗi bàn có một mã duy nhất và thuộc một khu vực, ngoài ra còn có trạng thái bàn đã có khách hay chưa. Mỗi khu vực cũng có mã duy nhất, gồm nhiều bàn, quy định tối đa 5 khu vực, ngoài ra còn có tên khu vực và trạng thái . Trong nhà hàng cũng có nhiều nhân viên, mỗi nhân viên có một mã duy nhất, ngoài ra còn có tên, số điện thoại, giới tính, ngày sinh, địa chỉ, lương , người quản lý. Mỗi nhân viên có một tài khoản riêng biệt để đăng nhập vào phần mềm gồm tên đăng nhập, mật khẩu không thể thay đổi . Mỗi khách hàng đến nhà hàng , lúc thanh toán sẽ có thể lưu thông tin để lưu điểm tích lũy và giảm giá trong những lần ăn tới, gồm có tên khách, số điện thoại, số điểm tích lũy và mã khách hàng duy nhất. Mỗi khách khi đặt món chưa thanh toán sẽ được lưu lại gồm các món đã đặt, vị trí ngồi , số lượng món và giá từng món, sau khi thanh toán sẽ tạo thành bill gồm có mã khách hàng, mã giảm giá nếu khách có dùng, số điểm dùng để giảm của khách, thời gian thanh toán tổng tiền,trạng thái. Sự kiện giảm giá được lưu lại gồm mã sự kiện là duy nhất, tên sự kiện, thời gian bắt đầu kết thúc.

## Thông tin của phần mềm

Tên phần mềm: Quản lý nhà hàng

Ngôn ngữ: C#

Giao diện lập trình ứng dụng: wpf

Phần mềm hỗ trợ: SQL server

## Chức năng của phần mềm

### Tạo, quản lý tài khoản (thêm, xóa, sửa).

### Đăng nhập.

### Quản lý bàn ăn (tình trạng trống, đầy).

### Bán hàng: tạo, sửa, thanh toán hóa đơn.

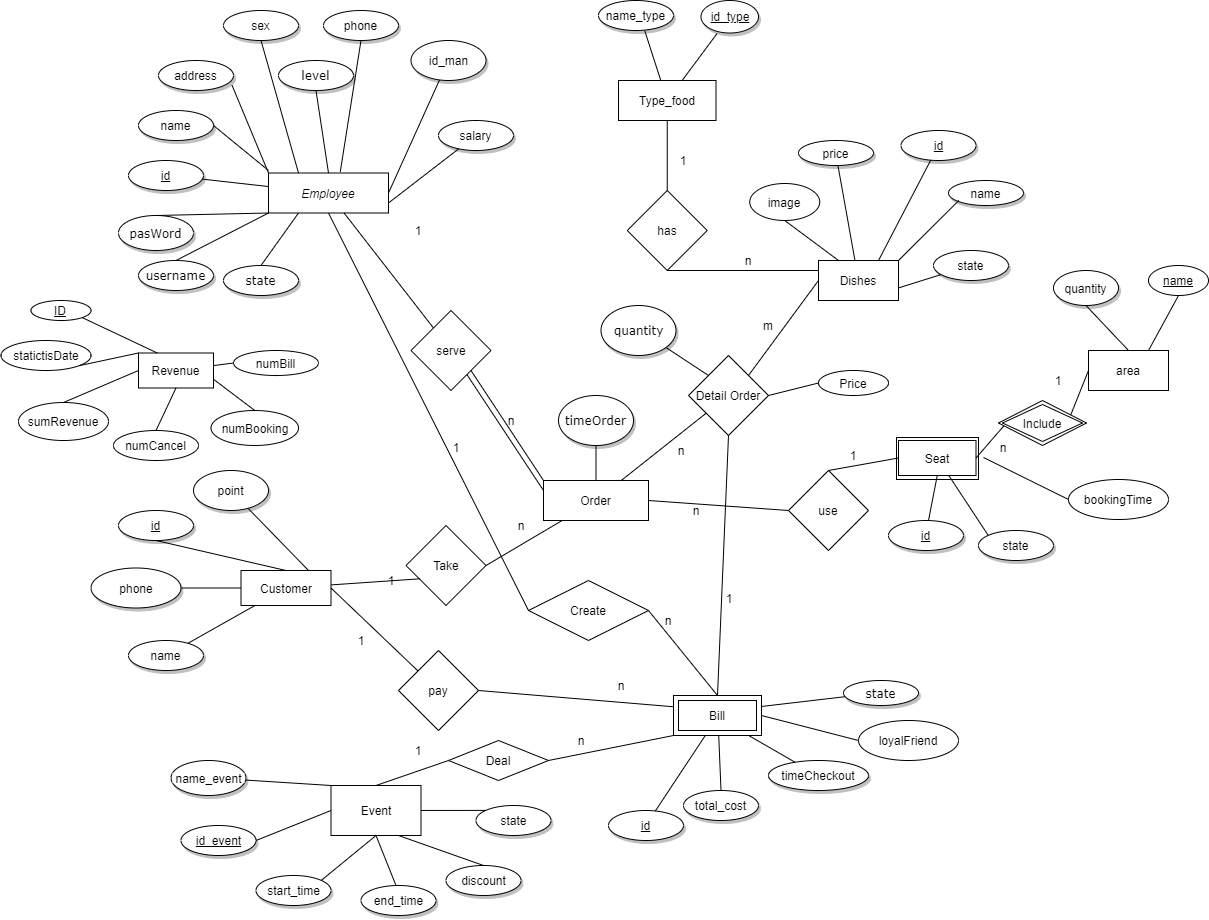
### Quản lý thông tin khách hàng, nhân viên.

### Quản lý loại món ăn, món ăn (thêm, xóa, sửa, tình trạng loại món ăn, món ăn).

### Thống kê: Tổng số doanh thu, số lượng hóa đơn, thanh toán.

# Thiết kế cơ sở dữ liệu

## Mô hình ERD



## Lược đồ quan hệ

Employee(id, idMan, userName, passWord, name, sex, phone, address, salary, level, state)

* Mỗi nhân viên được xác định bởi ID, có các thông tin liên lạc và có một mã người quản lý. Nhân viên được quản lý trực tiếp bởi người quản lí (ông chủ), người có cấp cao nhất sẽ quản lý chính mình.

Customer(id, phone, name, point)

* Mỗi khách hàng sẽ có số điện thoại (phone) riêng, có tên, điểm tích lũy. Cứ 100.000 vnd sử dụng khách hàng sẽ được tích một điểm. Cứ 100 điểm khách hàng sẽ được giảm giá, sau khi áp dụng giảm giá, số điểm tích lũy sẽ được trừ tương ứng.

Event(id,name, startDate, endDate, discount)

* + Mỗi khuyến mãi sẽ có mã xác đinh (id), tên chủ đề của từng khuyến mãi (name), startDate thời gian bắt đầu, endDate thời gian kết thúc khuyến mãi, discount là tỷ lệ phần trăm giảm giá trên sản phẩm.

Dished(id, name, price, idType, state, image)

* Mỗi món ăn sẽ có mã xác định (id), tên món ăn (name), thuộc một loại món ăn nhất định (idType), có giá price và tính trạng có đang được phục vụ ở cửa hàng hay không (state)

TypeFood(id, name, state)

* Mỗi loại món ăn sẽ gồm nhiều món ăn, được xác định bởi idType, và tình trạng có còn đang phục vụ tại cửa hàng hay không. Khi muốn ngưng bán một dòng sản phẩm (tất cả các món ăn của một loại cụ thể) có thể thay đổi tình trạng của loại món ăn tương ứng (điều ngược lại thì không đúng).

Bill(id, timeCheckOut, deal, totalCost ,idEmp, idCus, state, loyalFriend)

* Mỗi hóa đơn sẽ được phân biệt bằng mã (id), có giá trị thanh toán (totalCost), có thời gian thanh toán (timeCheckOut), có mã nhân viên phục vụ thu ngân (idEmp), khuyến mãi áp dụng (deal), mã khách hàng thanh toán (idCus), và số điểm tích lũy (hay khuyến mãi khách hàng thân thiết tương ứng).

Order (id, idSeat, idEmp, idCus, timeOrder)

* + Mỗi order sẽ được phân biệt bằng id, lưu thông tin về nhân viên đang thanh toán (idEmp), mã bàn của khách hàng đang sử dụng dịch vụ (idCus và idSet).
  + Sau khi hóa đơn được thanh toán, order sẽ bị hủy, mã của order tiếp theo tiếp tục tăng lên.

OrderDetail (id, idOrder, idBill, idDish, quantity)

* Nhiều orderDetail có thể thuộc cùng một bill giống nhau (giống nhau về idBill), phân biệt bởi mã Bill và món ăn tương ứng (idBill và idDishes), mã order sẽ bị xóa khi order đó được thanh toán (hoặc hủy).

Seat (id, state, idArea, bookingTime)

* Mỗi bàn sẽ phân biệt bởi mã bàn (id), tình trạng bàn còn trống hay không (state) với 1 biểu thị tình trạng bàn còn trống (active), 2 là bàn đã có khách, 3 bàn đã được đặt trước, và 4 là bàn hiện không còn hoạt động.

Area (id, name, quantity)

* Id là id của khu vực, mỗi khu vực sẽ có một id riêng khác với các khu vực còn lại để phân biệt, name là tên của mỗi khu vực, quantity là tổng số bàn của mỗi khu vực

Revenue (id, statisticDate, sumRevenue, numBill, numBooking)

* Doanh thu được tự động thống kê dựa theo ngày statisticDate, thống kê theo tổng doanh thu (sumRevenue), tổng số hóa đơn được lập (numBill), tổng số đặt trước (numBooking). Mỗi khi có hóa đơn mới được cập nhật, Revenue sẽ được cập nhật tương ứng.

## Thiết kế cơ sở dữ liệu mức chi tiết

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã sự kiện | | INT identity(1,1) | | Primary Key | |  | |
| name | Tên sự kiện | | NVARCHAR(100) | | NOT NULL | |  | |
| startDate | Ngày bắt đầu | | DATE | | NOT NULL | |  | |
| endDate | Ngày kết thúc | | DATE | | NOT NULL | |  | |
| discount | Giảm giá áp dụng | | DECIMAL(5,2) | | NOT NULL | | Là tỉ lệ 1-100% giảm giá trên sản phẩm | |
|  |  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã khách hàng | | INT identity(1,1) | | primary Key | |  | |
| phone | Số điện thoại | | VARCHAR(12) | |  | |  | |
| name | Tên khách hàng | | NVARCHAR(30) | | NOT NULL | |  | |
| point | Điểm khách hàng | | int | | NOT NULL  DEFAULT 0 | | Cứ 100,000 đồng trong hóa đơn, khách hàng sẽ được tích 1 điểm | |
|  |  | |  | |  | |  | |
|  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã nhân viên | | INT identity(1,1) | | Primary Key | |  | |
| idMan | Mã người quản lý | | INT | | Foreign Key | |  | |
| userName | Tài khoản | | VARCHAR(15) | | NOT NULL | |  | |
| passWord | Mật khẩu | | VARCHAR(15) | | NOT NULL | |  | |
| name | Tên nhân viên | | NVARCHAR(30) | | NOT NULL | |  | |
| sex | Giới tính | | char(1) | | NOT NULL | |  | |
| phone | Số điện thoại | | VARCHAR(12) | | NOT NULL | |  | |
| address | Địa chỉ | | NVARCHAR(100) | | NOT NULL | |  | |
| salary | Lương | | INT | |  | |  | |
| level | Cấp độ | | bit | | DEFAULT 0 | | 0,1: ông chủ, nhân viên | |
| state | Trạng thái nhân viên | | bit | | DEFAULT 0 | | 0,1: thôi việc, còn làm việc | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã loại thức ăn | | INT identity(1,1) | | Primary Key | |  | |
| name | Tên loại thức ăn | | NVARCHAR(30) | | NOT NULL | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã đĩa (món ăn) | | INT identity(1,1) | | Primary Key | |  | |
| name | Tên món ăn | | NVARCHAR(30) | | NOT NULL | |  | |
| price | Giá | | INT | | NOT NULL | |  | |
| idType | Mã loại món ăn | | INT | | FOREIGN KEY | |  | |
| state | Trạng thái đĩa | | NVARCHAR(15) | | NOT NULL | | 1,0: Còn phục vụ, ngưng phục vụ | |
| image |  | | NVARCHAR(100) | | DEFAULT 'dish.png' | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã Cancel | | INT | | Primary Key | |  | |
| idOrder | Mã đặt hàng | | INT | | Foreign Key | |  | |
| idEmp | Mã nhân viên | | INT | | foreign Key | |  | |
| idCus | Mã khách hàng | | INT | | Foreign Key | |  | |
| timeCancel | Thời gian hủy | | DATETIME | |  | |  | |
| totalCost | Tổng tiền | | INT | |  | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã chỗ ngồi | | INT identity(1,1) | | PRIMARY KEY | |  | |
| state | Trạng thái chỗ ngồi | | int | | DEFAULT 1 not null | | 1 biểu thị tình trạng bàn còn trống (active), 2 là bàn đã có khách, 3 bàn đã được đặt trước, và 4 là bàn hiện không còn hoạt động. | |
| idArea | Vị trí | | INT | | FOREIGN KEY | | Biểu diễn bởi 3 chiều a,b,c). Với a biểu diễn khu vực (lầu 1, lầu 2,..), (b,c) tọa độ hàng cột.  Vd: 123 => Lầu 1 bàn thứ 2 cột thứ 3 | |
| bookingTime | Thời gian đặt trước | | DateTime | |  | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã đơn đặt hàng | | INT identity(1,1) | | Primary Key | |  | |
| idSeat | Mã chỗ ngồi | | INT | | FOREIGN KEY | |  | |
| idEmp | Mã nhân viên | | INT | | FOREIGN KEY | |  | |
| idCus | Mã khách hàng | | INT | | FOREIGN KEY | | mã khách hàng. Người đầu tiên là guest. Dùng chung khi khách hàng không có tài khoản và chưa muốn tạo tài khoản | |
| timeOrder | Thời gian đặt hàng | | DATETIME | | NOT NULL | | GETDATE() | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã thông tin đơn | | INT identity(1,1) | | Primary Key | |  | |
| idOrder | Mã order | | int | | FOREIGN KEY | |  | |
| idBill | Mã bill | | int | | FOREIGN KEY | |  | |
| idDish | Mã món ăn | | INT | | FOREIGN KEY | |  | |
| quantity | Số lượng | | INT | | NOT NULL  DEFAULT 1 | |  | |
| price | Giá | | INT | | NOT NULL DEFAULT 0 | |  | |
| idEmp | Mã nhân viên | | INT | | FOREIGN KEY | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã hóa đơn | | INT identity(1,1) | | Primary Key | |  | |
| timeCheckOut | Thời gian thanh toán, hủy | | DATETIME | |  | | Thời gian thanh toán đối với đơn thanh toán hoặc thời gian hủy đối với order bị hủy | |
| deal | Thỏa thuận | | INT | | FOREIGN KEY | | Với hóa đơn bị hủy thì cột này là null, Khóa ngoại tới Events(id) | |
| totalCost | Tổng chi phí hóa đơn | | INT | |  | |  | |
| idEmp | Mã nhân viên | | INT | | FOREIGN Key | |  | |
| idCus | Mã khách hàng | | INT | | FOREOGN Key | |  | |
| state | Trạng thái | | BIT | | NOT NULL | | 0: Hủy,1: Đã thanh toán | |
| loyalFriend | Khách hàng thân thiết | | INT | | NOT NULL DEFAULT 0 | | Khách hàng có điểm trên 100, mà đã sử dụng thì trừ tiền tại đây | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã doanh thu | | INT identity | | PRIMARY KEY | |  | |
| statisticDate | Thời gian | | Date | | Not Null Default Cast(getDate() as Date) | | Giờ | |
| sumRevenue | Tổng doanh thu | | INT | | NOT NULL DEFAULT 0 | |  | |
| numCancel | Số lượng Order(Bill) bị hủy | | INT | | NOT NULL DEFAULT 0 | |  | |
| numBill | Số lượng hóa đơn trong ngày | | INT | | NOT NULL DEFAULT 0 | |  | |
| numBooking | Số lượng đặt hàng trong ngày | | INT | | NOT NULL DEFAULT 0 | |  | |
|  |  | |  | |  | |  | |
|  |  | |  | |  | |  | |
| **Thuộc tính** | **Diễn giải** | | **Kiểu dữ liệu** | | **Ràng buộc** | | **Ghi chú** | |
| id | Mã khu vực | | int | | primary key | |  | |
| name | Tên khu vực | | NVARCHAR(50) | |  | |  | |
| quantity | Định lượng | | NVARCHAR(50) | |  | | Diễn giải số bàn của mình | |

## Lược đồ quan hệ

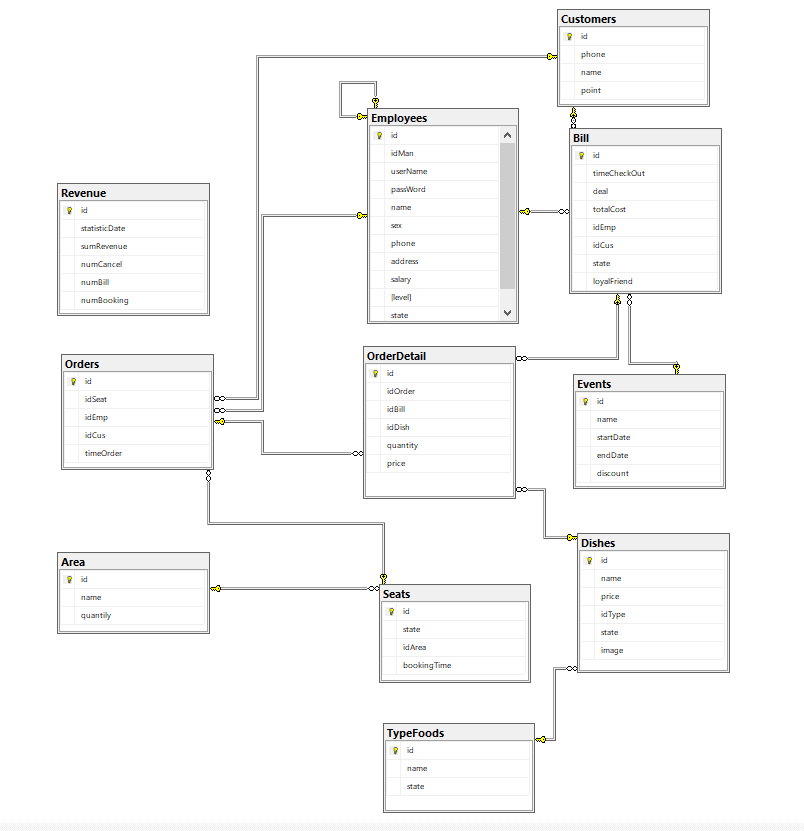


Figure : Mô hình quan hệ

(Lược đồ quan hệ giữa các bảng)

# Triển khai csdl (đổi tên lại cho pro)

## View

Lấy các Bill hiện có

|  |
| --- |
| create view getallBill select \* from Bill |

Lấy các Event hiện có

|  |
| --- |
| create view getallEvent select \* from Events |

Lấy danh sách các khách hàng hiện có

|  |
| --- |
| create view getallCus as Select \* from Customers |

## Trigger

### Trigger kiểm tra khi thay đổi trạng thái của bàn

create trigger tr\_ChangeStateTable on Seats for update

as

begin

declare @idban int

select @idban=id from inserted

IF (Select state from inserted) = 4

begin

if (Select state from deleted) = 1

begin

update Seats set state =4 where id=@idban

end

else

rollback tran

end

end

### Xóa thông tin đặt món sau khi thanh toán

CREATE OR ALTER TRIGGER DELETEOrder on OrderDetail for Update

as

BEGIN

Declare @idbill int, @idorder int, @olquant int, @nequant int;

select @idbill= i.idBill ,@idorder= i.idOrder ,@olquant= d.quantity, @nequant=i.quantity from inserted as i, deleted as d;

if(@idbill is not null and @idorder is not null and @olquant=@nequant)

begin

Update OrderDetail set idOrder=NULL where idOrder= @idOrder

Delete from Orders where id=@idorder;

end

END

GO

### Thay đổi trạng thái bàn sau khi xóa một order

CREATE OR ALTER TRIGGER tg\_changestateseat on Orders for Delete

as

BEGIN

Declare @idSeat int;

select @idSeat = idSeat

from deleted ;

Update Seats set state=1 where id=@idSeat;

END

GO

### Bản doanh thu mỗi khi có bill được thanh toán

create trigger tg\_updateRevenue on Bill after Insert

as

begin

if CAST( getDate() as DATE) not in (select r.statisticDate from Revenue r )

INSERT INTO Revenue VALUES (CAST( getDate() as DATE), 0,0,0,0)

DECLARE @day DATETIME,@mStateBill bit, @stateBill DATETIME, @cost int

select @day =cast( GETDATE() as DATE)

SELECT @stateBill = state From inserted

select @cost = totalCost FROM INSERTED

if (@stateBill =1)

BEGIN

update Revenue SET numCancel = numCancel +1 WHERE CAST(statisticDate as DATE)= @day

END

update Revenue SET numBill = numBill +1 WHERE cast (statisticDate as DATE) = @day

UPDATE Revenue SET sumRevenue = sumRevenue + @cost WHERE cast (statisticDate as DATE) = @day

end

go

### Update ngưng bán các món ăn của loại món ăn nếu loại món ăn đó ngưng phục vụ

Create trigger tg\_updateALlDish on TypeFoods for UPDATE

as

begin

declare @TypeState bit, @id int

Select @TypeState=i.state, @id= i.id from inserted i

if (@TypeState = 1)

update Dishes set state =cast( @TypeState as NVARCHAR(1))WHERE idType =@id

end

go

### Kiểm tra các món ăn bị disable có ảnh hưởng đến tình trạng của loại món ăn hay không

CREATE TRIGGER tg\_updateAllDishChangeType on Dishes AFTER INSERT, UPDATE

as

begin

Declare @ItemState NVARCHAR(15), @opItemState NVARCHAR(15), @idType INT

SELECT @ItemState= state, @idType=idType FROM INSERTED

if (@ItemState='0') --0: available

BEGIN

if (select t.state from TypeFoods t WHERE @idType= t.id) = 1

update TypeFoods set state = cast (@ItemState as BIT) WHERE id= @idType

END

ELSE

BEGIN

if (select count(d.id) from Dishes d WHERE d.state= '0' AND @idType= d.idType) <= 0

update TypeFoods set state = cast (@ItemState as BIT) WHERE id= @idType

END

end

## Stored Procedure (Chức năng)

### Kiểm tra trạng thái tài khoản

create procedure checkAccount (@userName VARCHAR(15),@passWord VARCHAR(15))

as

begin

Select \*

from Employees

where userName=@userName and passWord=@passWord and state=1;

end

### Lấy level của account

create procedure getLevelAccount (@userName VARCHAR(15),@passWord VARCHAR(15))

as

begin

Select \*

from Employees

where userName=@userName and passWord=@passWord and state=1 and level=1

end

### Disable bàn ăn

create procedure DisableTable @id int

as

begin

UPDATE Seats set state=4 where id=@id

end

go

### Enable lại bàn ăn

create procedure EnableTable @id int

as

begin

UPDATE Seats set state=1 where id=@id

end

go

### Thay đổi state của seat

create procedure ChangeStateSeat @id int,@state int

as

begin

UPDATE Seats set state=@state where id=@id

end

go

### Thêm 1 bàn

create procedure AddNewSeat @idArea int

as begin

INSERT INTO Seats(idArea) VALUES(@idArea);

end

### Chuyển bàn

create proc ChangeSeat @id1 int,@id2 int

as

begin

--chuyển trạng thái bàn 1 qua trống

update Seats set state=1 where id=@id1;

--chuyển trạng thái bàn 2 qua có khách

update Seats set state=2 where id=@id2;

--chuyển id seats trong order từ bàn 1 sang bàn 2

update Orders set idSeat=@id2 where idSeat=@id1

end

### Lấy order detail thông qua id bàn

create proc getOrderDetailBySeatId @id int

as

begin

select OrderDetail.idDish,OrderDetail.quantity,OrderDetail.idOrder

from OrderDetail, Orders

where OrderDetail.idOrder=Orders.id and Orders.idSeat=@id

end

### Update order detail

create proc updateOrderDetail @idDish int,@quantity int,@idOrder int

as

begin

update OrderDetail set quantity=quantity+@quantity

where idDish=@idDish and idOrder=@idOrder;

end

execute updateOrderDetail @idDish =5,@quantity =1,@idOrder=163

### Thêm item cho orderDetail

create proc addItemOrderDetail @idDish int,@quantity int,@idOrder int

as

begin

declare @price int;

select @price=price

from Dishes

where id=@idDish;

INSERT INTO OrderDetail(idOrder,idDish,quantity,price) VALUES (@idOrder,@idDish,@quantity,@price);

end

### Xóa 1 order detail sau khi gộp bàn

create proc deleteOrderDetail @idOrder int

as

begin

delete from OrderDetail where idOrder=@idOrder;

end

### Tạo order

alter proc addOrder @idSeat int,@idEmp int

as

begin

insert into Orders (idSeat,idEmp,timeOrder) values (@idSeat,@idEmp,GETDATE());

end

### Tra cứu dữ liệu (lấy dữ liệu dòng cuối cùng của bảng order)

create proc getLastOrder

as

begin

SELECT TOP 1 \* FROM Orders ORDER BY id DESC

end

### Lấy dữ liệu của một order cụ thể

create proc GetOrders @idSeat int

as

begin

Select \* from Orders where idSeat=@idSeat;

End

### Validate order mới

create proc check\_Orderdetail

as

begin

declare @idOrder int

select @idOrder=id

from (SELECT TOP 1 \* FROM Orders ORDER BY id DESC) as A

if NOT EXISTS (SELECT \*

FROM OrderDetail

WHERE idOrder=@idOrder)

begin

Delete from Orders where id=@idOrder;

end

end

### Chỉnh sửa tên 1 khu vực bàn ăn

create proc editAreaName @idArea int,@nameArea nvarchar(50)

as

begin

update Area set name=@nameArea

where id=@idArea;

end

### Cập nhập lại idCustomer khi nhấn vào bếp

alter proc updateCustomerOrder @phone varchar(12)

as

begin

declare @idOrder int,@idCustomer int

select @idOrder=id

from (SELECT TOP 1 \* FROM Orders ORDER BY id DESC) as A;

if (@phone='null')

set @idCustomer=1

else

begin

select @idCustomer=id

from Customers

where phone=@phone;

end

update Orders set idCus=@idCustomer where id=@idOrder;

end

### Lấy dữ liệu lên từ các table

CREATE OR ALTER PROC GETDATA (@table varchar(30))

AS

if (@table ='Dishes')

begin

select \* from Dishes

return

end

else if (@table ='TypeFoods')

begin

select \* from TypeFoods where state='False'

return

end

else if (@table ='Manager')

begin

select \* from Employees as a

where Exists(select \*

from Employees b

where a.id= b.idMan)

return

end

else if(@table='Events')

begin

select \* from Events

where endDate>GETDATE()

return

end

else if(@table ='Bill')

begin

select \* from Bill

return

end

else if(@table='Area')

begin

select \* from Area

return

end

else if (@table='Employees')

begin

select \* from Employees

return

end

else if (@table='Customers')

begin

select \* from Customers

return

end

GO

### Thêm nhân viên

CREATE PROC addEmployees(@idMan int,

@userName nvarchar(15),

@passWord varchar(15),

@name nvarchar(30),

@sex nvarchar(2),

@phone varchar(12),

@address nvarchar(100),

@salary int,

@level int,

@state nvarchar(15))

As

Begin

Insert into Employees values ( @idMan, @userName,

@passWord, @name, @sex,

@phone, @address, @salary,

@level,@state);

End

### Cập nhật nhân viên

CREATE OR ALTER PROC updateEmployees(@id int, @idMan int,

@userName nvarchar(15),

@passWord varchar(15),

@name nvarchar(30),

@sex CHAR(1),

@phone varchar(12),

@address nvarchar(100),

@salary int,

@level int,

@state nvarchar(15))

As

Begin

UPDATE Employees

SET

idMan=@idMan,

userName=@userName,

passWord=@passWord,

name=@name,

sex=@sex,

phone=@phone,

address=@address,

salary=@salary,

level= @level,

state=@state

where id=@id;

END

### Thêm khách hàng

CREATE PROC addCustomer (@phone varchar(10),

@name nvarchar(30),

@point int)

AS

BEGIN

INSERT INTO Customers VALUES (@phone, @name ,@point)

END

GO

### Lấy danh sách khách hàng theo ID

CREATE PROC getCustomerByID(@id int)

AS

BEGIN

SELECT \* FROM Customers where id=@id

END

GO

### Lấy điểm tích lũy của khach hàng

CREATE PROC getCustomerPoint (@id int)

AS

BEGIN

SELECT point from Customers where id=@id

END

go

### Lấy thông tin khách hàng bằng ID

CREATE PROC getCustomerByID(@id int)

AS

BEGIN

SELECT \* FROM Customers where id=@id

END

GO

### Lấy danh sách các món đã được order của một khách hàng

CREATE PROC getOrderDetail

as

BEGIN

SELECT \* FROM OrderDetail ;

END

go

### Thêm mới các món ăn trong chi tiết món

CREATE PROC addOrderDetail (

@idOrder int,

@idDish int,

@quantity int,

@price int)

AS

BEGIN

INSERT INTO OrderDetail VALUES (@idOrder,null,@idDish,@quantity, @price)

END

### Lấy tất cả thông tin của orderDetail bằng idOrder

CREATE PROC getDishesofaOrder (@idorder int )

AS

BEGIN

SELECT \* FROM OrderDetail

where idOrder=@idorder

END

go

### Lấy nhân viên bằng ID

Create function getIdByEmp (@userName varchar(15))

returns int

as

begin

declare @id int;

select @id=id from Employees where userName=@userName;

return @id

end

go

### Lấy tên món ăn bằng id

CREATE PROC getnameDishbyID (@id int)

as

BEGIN

SELECT name

FROM Dishes

WHERE id= @id

END

GO

### Update số lượng đặt món

CREATE PROC updatequantityOrder (@id int, @quantity int)

AS

BEGIN

UPDATE OrderDetail SET quantity=@quantity where id=@id;

END

### Thêm bill mới

CREATE OR ALTER PROC pro\_addNewBill ( @deal int, @totalcost int, @idem int, @idcus int, @state int, @loyalFriend int, @idOrder int)

AS

BEGIN

if(@state !=0 and @state!=1)

set @state=0;

if(@idcus=0)

set @idcus = null;

INSERT INTO Bill VALUES ( GETDATE(),@deal ,@totalcost,@idem,@idcus,@state, @loyalFriend );

Declare @idbill int ;

set @idbill= (SELECT TOP 1 id FROM Bill

ORDER BY id DESC);

update OrderDetail set idBill=@idbill where idOrder=@idOrder

END

GO

### Best seller

CREATE OR ALTER PROC getBestSeller (@idType int)

as

BEGIN

select top(5) id, name, price, idType, state, image, TongDia

from Dishes,(select sum(quantity) as TongDia, idDish

from OrderDetail , Dishes

where Dishes.id=idDish AND Dishes.idType= @idType and idBill IS not NULL

Group by idDish) as t

where Dishes.id =t.idDish

order by t.TongDia Desc

END

GO

### Tìm kiếm nhân viên

CREATE or ALTER PROC pro\_searchEmployee (@key nvarchar (50), @condition nvarchar(20))

as

BEGIN

if(@key ='null')

begin

if(@condition=N'Tất cả' or @condition='')

select \* from Employees

if(@condition= N'Nam')

select \* from Employees where sex='M'

if (@condition=N'Nữ')

select \* from Employees where sex='F'

if(@condition=N'Quản lí')

select \* from Employees where level=1

if(@condition=N'Nhân viên')

select \* from Employees where level=0

if(@condition=N'Đang làm việc')

select \* from Employees where state=1

if(@condition=N'Đã nghĩ' )

select \* from Employees where state=0

end

else

begin

if (@condition =N'Tất cả')

select \* from Employees where (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')

if(@condition= 'Nam')

select \* from Employees where sex='M' and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')

if (@condition=N'Nữ')

select \* from Employees where sex='F' and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')

if(@condition=N'Quản lí')

select \* from Employees where level=1 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')

if(@condition=N'Nhân viên')

select \* from Employees where level=0 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')

if(@condition=N'Đang làm việc')

select \* from Employees where state=1 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')

if(@condition=N'Đã nghĩ' )

select \* from Employees where state=0 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')

end

END

GO

### Kiểm tra username có tồn tại

CREATE OR ALTER proc isExistUsername(@username varchar(15))

as

BEGIN

SELECT userName From Employees

where userName= @username;

END

### Load loại sản phẩm

CREATE or ALTER PROCEDURE pro\_loadTypeFood @state NVARCHAR =NULL

as

begin

if (@state is NULL)

select t.id, t.name, COUNT(d.id) as quantity, t.state as stateService

from TypeFoods t LEFT JOIN Dishes d ON t.id = d.idType

GROUP by t.id, t.name, t.state

ELSE

select t.id, t.name, COUNT(d.id) as quantity, t.state as stateService

from TypeFoods t LEFT JOIN Dishes d ON t.id = d.idType

where t.state = cast(@state as bit)

GROUP by t.id, t.name, t.state

End

### Lấy loại món ăn bằng tên

create or alter function loadTypeFoodByName(@name nvarchar(30))

returns table

as

return SELECT t.id, t.name, COUNT(d.id) as quantity, t.state as stateService from TypeFoods t, Dishes d

WHERE (select charindex(@name, t.name))>0 and t.id = d.idType

group by t.id, t.name, t.state

go

### Lấy loại món ăn bằng tên và trạng thái

CREATE or alter PROCEDURE pro\_loadTypeFoodByName @name NVARCHAR(30), @state NVARCHAR =NULL

AS

BEGIN

if (@state is not NULL) SELECT t.id, t.name, COUNT(d.id) as quantity, t.state as stateService from TypeFoods t, Dishes d

WHERE (cast(@state as bit)) = t.state and (select charindex(@name, t.name))>0 and t.id = d.idType

group by t.id, t.name, t.state

ELSE

SELECT t.id, t.name, COUNT(d.id) as quantity, t.state as stateService from TypeFoods t, Dishes d

WHERE (select charindex(@name, t.name))>0 and t.id = d.idType

group by t.id, t.name, t.state

END

### Lấy danh sách món ăn của một loại món ăn cụ thể

create or alter PROCEDURE pro\_findTypeFoodByName(@id int)

as

BEGIN

SELECT d.id,d.name,d.price,t.id, d.state,t.name,d.image FROM Dishes d, TypeFoods t

WHERE t.id=@id AND t.id=d.idType

END

DECLARE @type INT

set @type= 4

EXECUTE pro\_findTypeFoodByName @type

go

### Chỉnh sửa loại món ăn

create or alter procedure pro\_editTypeFood(@id int, @name NVARCHAR(30), @state bit )

as

begin

update TypeFoods set name= @name, state= @state where @id = id

end

### Lấy thông tin cụ thể của một hóa đơn

CREATE OR ALTER procedure load\_Invoice (@id int)

as

BEGIN

select b.id, b.timeCheckOut , c.name as CusName, c.phone, e.id as EmpId, e.name EmpName, eve.name as PromName, eve.discount, b.loyalFriend, b.totalCost

FROM Bill b, Customers c, Employees e, Events eve

WHERE b.id =@id and b.idCus =c.id and b.idEmp =e.id and b.deal =eve.id

END

### Lấy các món ăn của một hóa đơn

CREATE or ALTER PROCEDURE pro\_loadDishInBill(@id int)

as

BEGIN

select d.name, odt.price, odt.quantity from Bill b, OrderDetail odt, Dishes d

where @id=b.id AND b.id= odt.idBill AND odt.idDish =d.id

END

go

### Thêm một loại món ăn mới

create or alter PROC pro\_addTypeFood(@name nvarchar(30), @state bit)

as

BEGIN

insert into TypeFoods VALUES (@name, @state);

END

go

### Thêm sự kiện khuyến mãi

CREATE PROC addEvents(@id INT, @name NVARCHAR(50), @startDate DATETIME, @endDate DATETIME, @discount DOUBLE PRECISION)

AS

BEGIN

INSERT INTO dbo.Events VALUES(@id,@name,@startDate,@endDate,@discount)

END

GO

### Chỉnh sửa sự kiện

CREATE PROC EditEvents(@id INT, @name NVARCHAR(50), @startDate DATETIME, @endDate DATETIME, @discount DOUBLE PRECISION)

AS

BEGIN

UPDATE dbo.Events SET name=@name, startDate=@startDate, endDate=@endDate,discount=@discount

WHERE id=@id;

END

### Thêm mới một khách hàng

CREATE PROC AddCustomers(@id INT, @phone VARCHAR(20), @name NVARCHAR(50), @point INT)

AS

BEGIN

INSERT dbo.Customers VALUES(@id,@phone,@name,@point)

END

### Chinh sửa thông tin khách hàng

CREATE PROC EditCustomers(@id INT, @name NVARCHAR(50), @phone VARCHAR(50), @point INT)

AS

BEGIN

UPDATE dbo.Customers SET name=@name, phone=@phone, point=@point

WHERE id=@id

END

### Thêm món ăn

CREATE PROC AddDishes( @name NVARCHAR(50), @price INT, @nameType NVARCHAR(50), @state BIT, @img\_source VARCHAR(50))

AS

BEGIN

DECLARE @idType INT

SELECT @idType=id FROM dbo.TypeFoods WHERE name=@nameType

INSERT dbo.Dishes VALUES ( @name,@price,@idType,@state,@img\_source)

END

GO

### Chỉnh sửa món ăn

CREATE PROC EditDishes(@id INT, @name NVARCHAR(50), @price INT, @nameType NVARCHAR(50), @state NVARCHAR(50), @img\_source VARCHAR(50))

AS

BEGIN

DECLARE @idType INT

SELECT @idType=dbo.TypeFoods.id FROM dbo.TypeFoods,dbo.Dishes WHERE Dishes.idType=dbo.TypeFoods.id AND @nameType=dbo.TypeFoods.name

UPDATE dbo.Dishes SET name=@name, price=@price, idType=@idType,state=@state,image=@img\_source

WHERE id=@id

END

GO

## Function

### Lấy danh sách nhân viên bằng id

CREATE OR ALTER FUNCTION fn\_getEmpByID (@id int)

RETURNS TABLE

AS

RETURN SELECT \* FROM Employees WHERE id=@id

GO

### Lấy số lượng món ăn

CREATE Or ALTER FUNCTION fn\_getQuantityDish (@idDish int, @idOrder int)

RETURNS INT

AS

BEGIN

DECLARE @QUAN INT;

SELECT @QUAN=quantity from OrderDetail where idDish= @idDish and idOrder= @idOrder;

RETURN @QUAN;

END

GO

### Lấy value discount trong Events

CREATE OR ALTER FUNCTION fn\_getDiscountpercent (@idEvent int)

RETURNS Decimal (5,2)

AS

BEGIN

DECLARE @a Decimal(5,2);

SELECT @a= discount

FROM Events WHERE id=@idEvent;

return @a;

END

go

### Thống kê doanh thu,... theo loại thức ăn

create or ALTER FUNCTION fn\_getStatisticByTypeFood(@start DATETIME, @end DATETIME)

RETURNS TABLE

AS

return select top 7 t.name, SUM(odt.quantity) as number,SUM(odt.price\*odt.quantity) as totalRevenue from TypeFoods t, Dishes d, Bill b, OrderDetail odt

WHERE CAST( b.timeCheckOut as DATE)>= CAST( @start AS DATE) and CAST( b.timeCheckOut as DATE)<=Cast( @end as date) and b.id=odt.idBill AND odt.idDish= d.id AND t.id = d.idType

GROUP by t.name

go

### Lấy danh sách tất cả cả Bill

create or alter function loadBill()

returns table

as

return select b.id as idBill, e.name as nameEmp, c.name as nameCus,count(o.id) as Quantity,

b.timeCheckOut,b.deal,

b.totalCost, b.state, b.loyalFriend from Bill b,

OrderDetail o, Employees e, Customers c

WHERE b.id = o.idBill and e.id =b.idEmp and b.idCus=c.id

GROUP BY b.id,e.name, c.name, b.timeCheckOut,b.totalCost,b.state ,b.deal,b.loyalFriend

go

### Tìm Bill (hóa đơn) theo khoản thời gian

create or alter function fn\_billByDateTime(@start DATETIME,@end DATETIME)

returns table

as RETURN (Select b.id as idBill,e.name as nameEmp, c.name as nameCus,COUNT(b.id) as Quantity,

b.timeCheckOut,b.deal,

b.totalCost, b.state, b.loyalFriend from Bill b,

OrderDetail o, Employees e, Customers c

WHERE b.timeCheckOut >= @start and b.timeCheckOut <= @end and b.id = o.idBill and e.id =b.idEmp and b.idCus=c.id

GROUP BY b.id, e.name, c.name, b.timeCheckOut,b.totalCost,b.state ,b.deal,b.loyalFriend)

go

### Tìm kiếm hóa đơn theo trạng thái (hoàn thành- hủy)

create or alter function fn\_billByState(@state BIT)

returns table

as RETURN select b.id as idBill, e.name as nameEmp, c.name as nameCus,COUNT(b.id) as Quantity,

b.timeCheckOut,b.deal,

b.totalCost, b.state, b.loyalFriend from Bill b,

OrderDetail o, Employees e, Customers c

WHERE b.state = @state and b.id = o.idBill and e.id =b.idEmp and b.idCus=c.id

GROUP BY b.id, e.name, c.name, b.timeCheckOut,b.totalCost,b.state ,b.deal,b.loyalFriend

go

select \* FROM dbo.fn\_billByState(1)

### Tìm danh sách hóa đơn theo tên khách hàng

CREATE or ALTER function fn\_billByNameCus(@name nvarchar(30))

returns table

as RETURN select b.id as idByBill, e.name as nameEmp, c.name as nameCus,COUNT(o.id) as Quantity,

b.timeCheckOut,b.deal,

b.totalCost, b.state, b.loyalFriend from Bill b,

OrderDetail o, Employees e, Customers c

WHERE (select CHARINDEX(@name, c.name))>0 and b.id = o.idBill

and e.id =b.idEmp and b.idCus=c.id

GROUP BY b.id, e.name, c.name, b.timeCheckOut,b.totalCost,b.state ,b.deal,b.loyalFriend

go

### Tìm kiếm danh sách hóa đơn theo tên nhân viên thu ngân

CREATE or ALTER function fn\_billByNameEmp(@name nvarchar(30))

returns table

as RETURN select b.id as idByBill, e.name as nameEmp, c.name as nameCus,COUNT(o.id) as Quantity,

b.timeCheckOut,b.deal,

b.totalCost, b.state, b.loyalFriend from Bill b,

OrderDetail o, Employees e, Customers c

WHERE (select CHARINDEX(@name, e.name))>0 and b.id = o.idBill

and e.id =b.idEmp and b.idCus=c.id and e.state=0

GROUP BY b.id, e.name, c.name, b.timeCheckOut,b.totalCost,b.state,b.deal,b.loyalFriend

go

### Lấy danh sách các hóa đơn theo ID

create or alter function fn\_billById(@id int)

returns table

as

return select b.id as idBill , e.name as nameEmp, c.name as nameCus,count(o.id) as Quantity,

b.timeCheckOut,b.deal,

b.totalCost, b.state, b.loyalFriend from Bill b,

OrderDetail o, Employees e, Customers c

WHERE (SELECT CHARINDEX(cast (@id as NVARCHAR), CAST (b.id as NVARCHAR)))>0 and b.id = o.idBill and e.id =b.idEmp and b.idCus=c.id

GROUP BY b.id, e.name, c.name, b.timeCheckOut,b.totalCost,b.state ,b.deal,b.loyalFriend

go

### Thống kê doanh thu,... theo loại thứ ăn

create or ALTER FUNCTION fn\_getStatisticBySpecificalFood(@start DATETIME, @end DATETIME)

RETURNS TABLE

AS

return select top 7 d.name, SUM(odt.quantity) as number,SUM(odt.quantity\* odt.price) as totalRevenue

from Dishes d, Bill b, OrderDetail odt

WHERE CAST( b.timeCheckOut as DATE)>= CAST( @start AS DATE) and CAST( b.timeCheckOut as DATE)<=Cast( @end as date) and b.id=odt.idBill AND odt.idDish= d.id

GROUP by d.name

### Lấy doanh thu theo ngày

CREATE or alter FUNCTION fn\_RevenurePerDay (@day DATETIME)

RETURNS TABLE

AS

return select \* FROM Revenue WHERE CAST(@Day as DATE) = Revenue.statisticDate

### Lấy doanh thu theo khoảng thời gian

CREATE or alter FUNCTION fn\_RevenuePerDay (@start DATETIME, @end DATETIME)

RETURNS TABLE

AS

return select id, statisticDate, sumRevenue, numCancel, numBill, numBooking

FROM Revenue

WHERE CAST(@start as DATE) <= Revenue.statisticDate and CAST(@end as DATE) >= Revenue.statisticDate

go

### Lấy loại món ăn

CREATE or ALTER function loadTypeFood ()

returns table

as

return select t.id, t.name, COUNT(d.id) as quantity, t.state as stateService

from TypeFoods t, Dishes d

WHERE t.id = d.idType

GROUP by t.id, t.name, t.state

go

select \* from dbo.loadTypeFood()

go

## Phân quyền

### Tạo user và role

|  |
| --- |
| -----------Tạo tài khoản nhân viên và thêm tài khoản vào Role Thu ngân  Create Login thungan With Password '123'  Create User thungan For Login thungan  EXEC sp\_addrole 'RoleThuNgan'  EXEC sp\_addrolemember 'RoleThuNgan', 'thungan' |

|  |
| --- |
| -------Tạo tài khoản quản lý nhà hàng và thêm tài khoản vào Role Chủ cửa hàng  EXEC sp\_addrole 'RoleChuCuaHang'  ----------------Tạo tài khoản chủ cửa hàng---------------  CREATE LOGIN quanlynhahang WITH PASSWORD=N'123', DEFAULT\_DATABASE=[master], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=ON  GO  USE ManagementRestaurant  GO  CREATE USER quanlynhahang FOR LOGIN quanlynhahang  GO  ALTER USER quanlynhahang WITH DEFAULT\_SCHEMA=[dbo]  GO  USE ManagementRestaurant  ALTER SERVER ROLE [sysadmin] ADD MEMBER quanlynhahang;  GO  USE ManagementRestaurant  ALTER SERVER ROLE [serveradmin] ADD MEMBER quanlynhahang;  GO  USE ManagementRestaurant  ALTER SERVER ROLE [setupadmin] ADD MEMBER quanlynhahang;  GO  USE ManagementRestaurant  ALTER SERVER ROLE [processadmin] ADD MEMBER quanlynhahang;  GO  USE ManagementRestaurant  ALTER SERVER ROLE [diskadmin] ADD MEMBER quanlynhahang;  GO  USE ManagementRestaurant  ALTER SERVER ROLE [securityadmin] ADD MEMBER quanlynhahang;  GO  USE ManagementRestaurant  ALTER SERVER ROLE [dbcreator] ADD MEMBER quanlynhahang;  GO  EXEC sp\_addrolemember 'RoleChuCuaHang', 'quanlynhahang' |

### Phân quyền Chủ cửa hàng

|  |
| --- |
| -----Phân Quyền cho chủ cửa hàng  GRANT EXECUTE ON dbo.pro\_searchEmployee TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.AddDishes TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.EditDishes TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.isExistUsername TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.addCustomer TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.AddDishes TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.AddCustomers TO RoleChuCuaHang WITH GRANT OPTION  grant EXECUTE ON dbo.addEmployees TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.addEvents TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.addItemOrderDetail TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.AddNewSeat TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.addOrder TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.addOrderDetail TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.ChangeSeat TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.ChangeStateSeat TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.check\_Orderdetail TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.checkAccount TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.CreateLoginAndUser TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.deleteEmployees TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.deleteOrderDetail TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.DisableTable TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.editAreaName TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.EditCustomers TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.EditDishes TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.EditEvents TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.EnableTable TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getCustomerByID TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getCustomerPoint TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.GETDATA TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getDishesofaOrder TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getlastid TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getLastOrder TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getLevelAccount TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getnameDishbyID TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getOrderDetail TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getOrderDetailBySeatId TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.GetOrders TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.load\_Invoice TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_addNewBill TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_addTypeFood TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_editTypeFood TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_findTypeFoodByName TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_loadDishInBill TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_loadTypeFood TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_loadTypeFoodByName TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.updateCustomerOrder TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.updateEmployees TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.updateOrderDetail TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.updatequantityOrder TO RoleChuCuaHang WITH GRANT OPTION  ------------------ Function  GRANT SELECT ON dbo.fn\_billByDateTime TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billById TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billByNameCus TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billByNameEmp TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billByState TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_getEmpByID TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_getStatisticBySpecificalFood TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_getStatisticByTypeFood TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_RevenuePerDay TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_RevenurePerDay TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.loadBill TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.loadTypeFood TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT ON dbo.loadTypeFoodByName TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.fn\_getDiscountpercent TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.fn\_getQuantityDish TO RoleChuCuaHang WITH GRANT OPTION  GRANT EXECUTE ON dbo.getlastidOrderDetail TO RoleChuCuaHang WITH GRANT OPTION  ---------- Phân quyền trên bảng  GRANT SELECT, INSERT, UPDATE, DELETE ON Area TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Bill TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Customers TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Dishes TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Employees TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Events TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON OrderDetail TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Orders TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Revenue TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Seats TO RoleChuCuaHang WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON TypeFoods TO RoleChuCuaHang WITH GRANT OPTION |

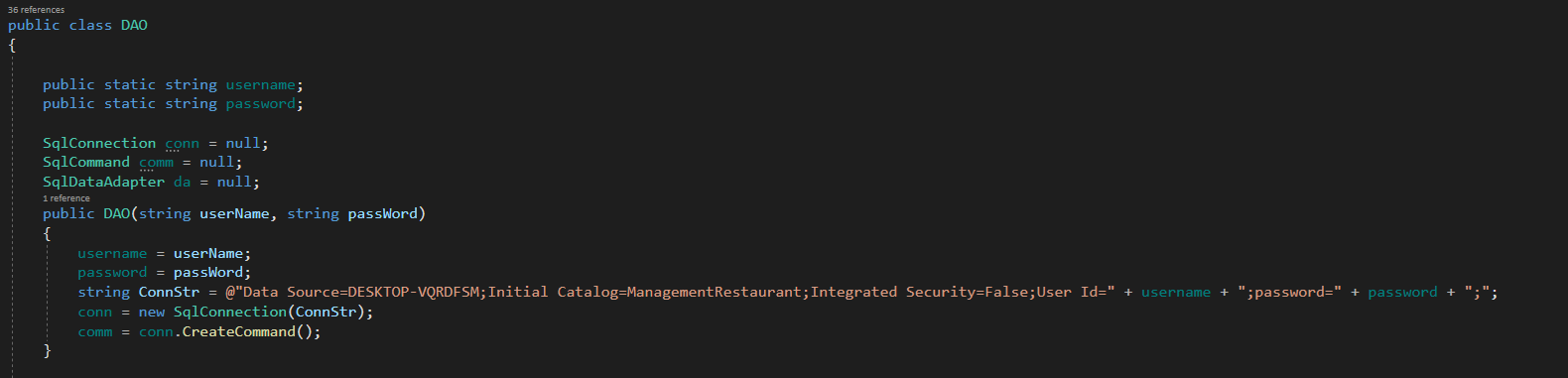
### Phân quyền nhân viên

|  |
| --- |
| ----- Phân Quyền cho thu ngân  GRANT EXECUTE ON dbo.pro\_searchEmployee TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.addCustomer TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.AddCustomers TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.addItemOrderDetail TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.addOrder TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.addOrderDetail TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.ChangeSeat TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.ChangeStateSeat TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.check\_Orderdetail TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.checkAccount TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.CreateLoginAndUser TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.deleteOrderDetail TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.EditCustomers TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getCustomerByID TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getCustomerPoint TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.GETDATA TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getDishesofaOrder TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getlastid TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getLastOrder TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getLevelAccount TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getnameDishbyID TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getOrderDetail TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getOrderDetailBySeatId TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.GetOrders TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.load\_Invoice TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_addNewBill TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_findTypeFoodByName TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_loadDishInBill TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_loadTypeFood TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.pro\_loadTypeFoodByName TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.updateCustomerOrder TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.updateOrderDetail TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.updatequantityOrder TO RoleThuNgan WITH GRANT OPTION  ---------Function  GRANT SELECT ON dbo.fn\_billByDateTime TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billById TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billByNameCus TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billByNameEmp TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_billByState TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_getEmpByID TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_getStatisticBySpecificalFood TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_getStatisticByTypeFood TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_RevenuePerDay TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.fn\_RevenurePerDay TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.loadBill TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.loadTypeFood TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON dbo.loadTypeFoodByName TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.fn\_getDiscountpercent TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.fn\_getQuantityDish TO RoleThuNgan WITH GRANT OPTION  GRANT EXECUTE ON dbo.getlastidOrderDetail TO RoleThuNgan WITH GRANT OPTION  ----------Phân quyền trên bảng  GRANT SELECT ON Area TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT, INSERT ON Bill TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON Customers TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON Dishes TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON Employees TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON Events TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON OrderDetail TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON Orders TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON Revenue TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON Seats TO RoleThuNgan WITH GRANT OPTION  GRANT SELECT ON TypeFoods TO RoleThuNgan WITH GRANT OPTION |

# Giao diện

*(Ứng dụng triển khai theo mô hình MVC)*

## Kết nối

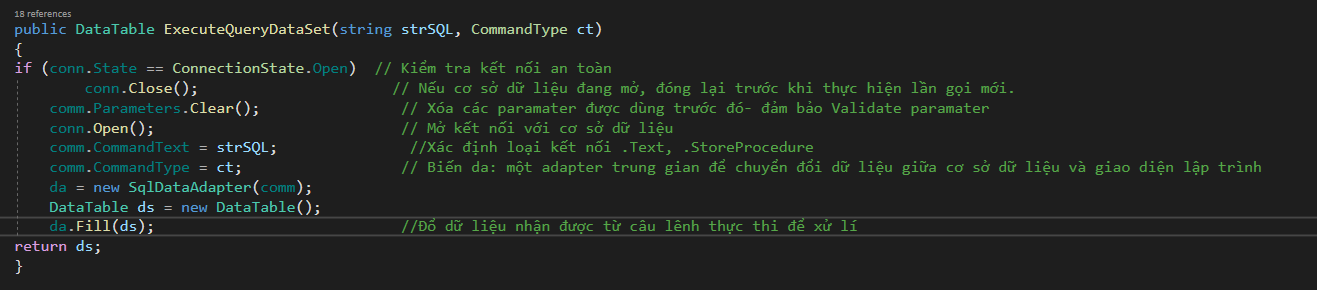
Đăng nhập ứng dụng- kết nối sql server sử dụng tài khoản phân quyền trên sql server (tên đăng nhập, mật khẩu sql server đồng thời cũng là tài khoản, mật khẩu của user đó trong cơ sở dữ liệu).

*Figure 1: Class DAO định dạng kết nối vào cơ sở dữ liệu*

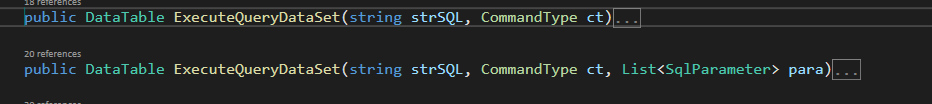
## Thực thi

Kết nối- thực thi query

#### Đối với câu truy vấn có lấy dữ liệu



*Figure 2: Template query truy vấn có trả về dữ liệu là một bảng, không có tham số truyền vào*



*Figure 3: Template cho query truy vấn có trả về dữ liệu, có tham số truyền vào.*

*CODE thực thi:*

public DataTable ExecuteQueryDataSet(string strSQL, CommandType ct)

{

if (conn.State == ConnectionState.Open) // Kiểm tra kết nối an toàn

conn.Close(); // Nếu cơ sở dữ liệu đang mở, đóng lại trước khi thực hiện lần gọi mới.

comm.Parameters.Clear(); // Xóa các paramater được dùng trước đó- đảm bảo Validate paramater

conn.Open(); // Mở kết nối với cơ sở dữ liệu

comm.CommandText = strSQL; //Xác định loại kết nối .Text, .StoreProcedure

comm.CommandType = ct; // Biến da: một adapter trung gian để chuyển đổi dữ liệu giữa cơ sở dữ liệu và giao .. diện lập trình

da = new SqlDataAdapter(comm);

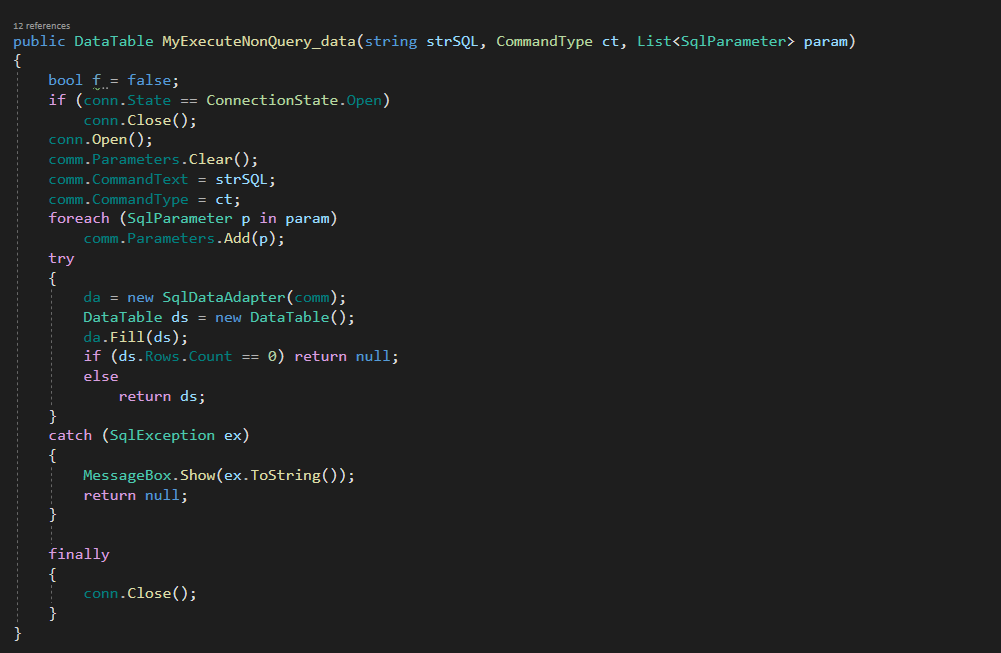
DataTable ds = new DataTable();

da.Fill(ds); //Đổ dữ liệu nhận được từ câu .. lệnh thực thi để xử lí

return ds;

}

Truy vấn chỉ trả về một giá trị duy nhất



*Figure 4: Template cho query truy vấn chỉ trả về một giá trị đơn lẻ.*

#### Truy vấn không lấy dữ liệu trả về

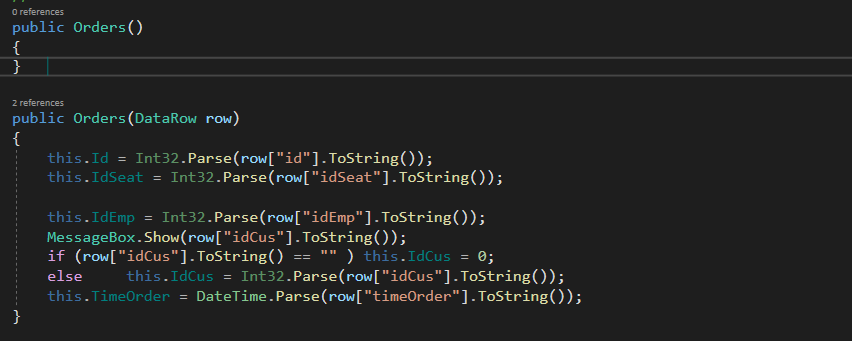


*Figure 5: Template cho query truy vấn không trả về dữ liệu.*

## Mô hình triển khai

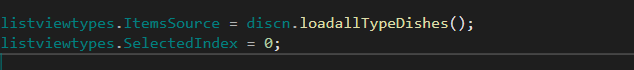
Hiển thị kết quả đối tượng dữ liệu trả về:

* Dữ liệu chứa trong Model



*Figure 6: Model Order*

* Hiển thị thông qua View



# Kết luận

## Những điều đã đạt được

Xây dựng được phần mềm quản lý nhà hàng đơn giản, thân thiện với người dùng. Trong quá trình thực hiện đề tài, nhóm đã đạt được một số thành quả:

* Nắm được phân tích, thiết kế hệ thống cơ sở dữ liệu phù hợp.
* Phân cấp phân quyền cho các đối tượng sử dụng cơ sở dữ liệu.
* Tìm hiểu và nắm rõ về một số khái niệm trong SQL Server, cách hoạt động của trigger, function, procedure.
* Hỗ trợ cho người dùng với giao diện khá thân thiện, giúp người quản trị dễ dàng quản lí.

## Hạn chế

* + - Chưa ứng dụng được view vào đề tài
    - Chưa tối ưu hóa truy vấn, cũng như dữ liệu nhớ phải khởi tạo để sử dụng.
    - Chưa kiểm thử hết các trường hợp có thể xảy ra trong thực tế để xử lý.

## Những procedure, function đặc biệt

### Trigger cho bản doanh thu mỗi khi có bill được thanh toán

|  |
| --- |
| -- Tạo Trigger cho bản doanh thu mỗi khi có bill được thanh toán  create trigger tg\_updateRevenue on Bill after Insert  as  begin    if CAST( getDate() as DATE) not in (select r.statisticDate from Revenue r )  INSERT INTO Revenue VALUES (CAST( getDate() as DATE), 0,0,0,0)  DECLARE @day DATETIME,@mStateBill bit, @stateBill DATETIME, @cost int  select @day =cast( GETDATE() as DATE)  SELECT @stateBill = state From inserted  select @cost = totalCost FROM INSERTED  if (@stateBill =1)  BEGIN  update Revenue SET numCancel = numCancel +1 WHERE CAST(statisticDate as DATE)= @day  END  update Revenue SET numBill = numBill +1 WHERE cast (statisticDate as DATE) = @day  UPDATE Revenue SET sumRevenue = sumRevenue + @cost WHERE cast (statisticDate as DATE) = @day  end  go |

### Hàm lấy dữ liệu dùng chung

|  |
| --- |
| CREATE OR ALTER PROC GETDATA (@table varchar(30))  AS  if (@table ='Dishes')  begin  select \* from Dishes  return  end  else if (@table ='TypeFoods')  begin  select \* from TypeFoods where state='False'  return  end  else if (@table ='Manager')  begin  select \* from Employees as a  where Exists(select \*  from Employees b  where a.id= b.idMan)  return  end  else if(@table='Events')  begin  select \* from Events  where endDate>GETDATE()  return  end  else if(@table ='Bill')  begin  select \* from Bill  return  end  else if(@table='Area')  begin  select \* from Area  return  end  else if (@table='Employees')  begin  select \* from Employees  return  end  else if (@table='Customers')  begin  select \* from Customers  return  end  GO |

### Kiểm tra Order cuối

(Trong thực tế, nhiều trường hợp chọn bàn nhưng vẫn chưa chọn món, thay đổi trước khi dữ liệu món ăn được cập nhật, procedure giúp giải quyết việc xác định order cuối)\_ Hàm tích hợp dùng chung cho các trường hợp

|  |
| --- |
| ALTER or CREATE FUNCTION getlastidOrderDetail (@table varchar(20))  returns int  as  BEGIN  Declare @id int;  if(@table ='Customers')  begin  if (EXISTS(select \* from Customers))  begin  select @id=MAX(id)  from Customers;  return @id;  end  else  return 0;  end  if(@table='OrderDetail')  begin  if(EXISTS(SELECT \* FROM OrderDetail))  begin  select @id= Max(id)  from OrderDetail;  return @id;  end  else return 0;  end  if(@table ='Employees')  begin  if(EXISTS(Select \* from Employees))  Begin  select @id= Max(id)  from Employees;  return @id;  end  else return 0;  end    if(@table='Orders')  begin  if(EXISTS (Select \* from Orders))  begin  select @id= Max(id)  from Orders;  return @id;  end  else return 0;  end    if(@table='Bill')  begin  if(EXISTS(Select \* from Bill))  begin  select @id= Max(id)  from Bill;  return @id;  end  else return 0;  end  if(@table='Dishes')  begin  if(EXISTS(Select \* from Dishes))  begin  select @id= Max(id)  from Dishes;  return @id;  end  else return 0;  end  if(@table='TypeFoods')  begin  if(EXISTS(Select \* from TypeFoods))  begin  select @id= Max(id)  from TypeFoods;  return @id;  end  else return 0;  end  if(@table='Events')  begin  if(EXISTS(Select \* from Events))  begin  select @id= Max(id)  from Events;  return @id;  end  else return 0;  end  return null;  END  GO |

### Tìm kiếm nhân viên theo các yêu cầu có sẵn

|  |
| --- |
| CREATE or ALTER PROC pro\_searchEmployee (@key nvarchar (50), @condition nvarchar(20))  as  BEGIN  if(@key ='null')  begin  if(@condition=N'Tất cả' or @condition='')  select \* from Employees  if(@condition= N'Nam')  select \* from Employees where sex='M'  if (@condition=N'Nữ')  select \* from Employees where sex='F'  if(@condition=N'Quản lí')  select \* from Employees where level=1  if(@condition=N'Nhân viên')  select \* from Employees where level=0  if(@condition=N'Đang làm việc')  select \* from Employees where state=1  if(@condition=N'Đã nghĩ' )  select \* from Employees where state=0  end  else  begin  if(@condition =N'Tất cả')  select \* from Employees where (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')  if(@condition= 'Nam')  select \* from Employees where sex='M' and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')  if (@condition=N'Nữ')  select \* from Employees where sex='F' and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')  if(@condition=N'Quản lí')  select \* from Employees where level=1 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')  if(@condition=N'Nhân viên')  select \* from Employees where level=0 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')  if(@condition=N'Đang làm việc')  select \* from Employees where state=1 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')  if(@condition=N'Đã nghĩ' )  select \* from Employees where state=0 and (name LIKE '%' +@key+'%' OR phone LIKE '%'+@key + '%' OR address LIKE '%' + @key + '%')  end  END  GO |

### 

* + 1. Hướng dẫn sử dụng demo

* Đăng nhập tài khoản mật khẩu của bạn: vào Tên đăng nhập và Mật khẩu

Các tài khoản:

* Chủ cửa hàng: tk: quanlynhahang ; mk: 123
* Nhân viên: tk: hoangg ; mk: 123

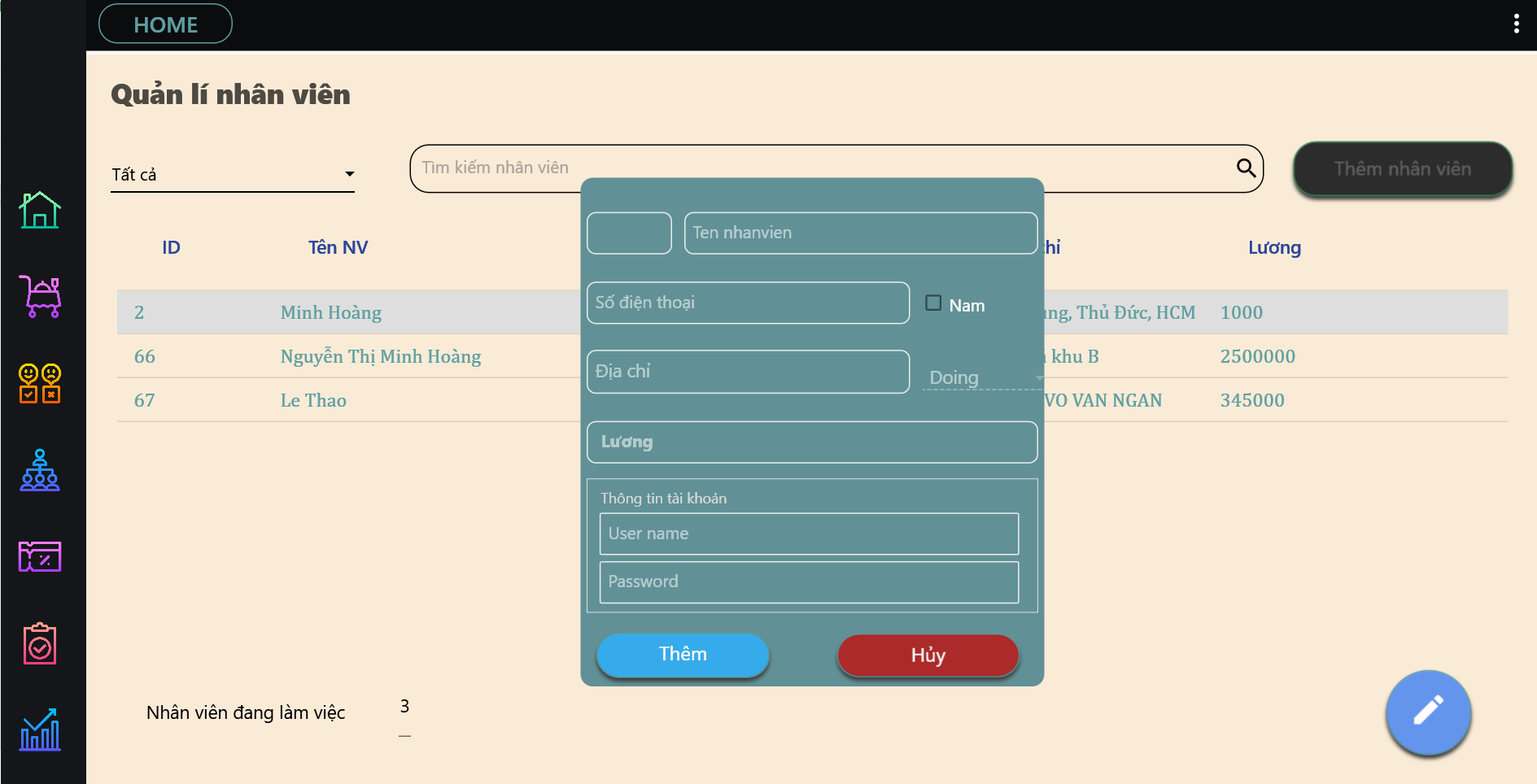
tk: thaobeo ; mk: 123

* Sau khi đăng nhập ta sẽ ra giao diện sau:



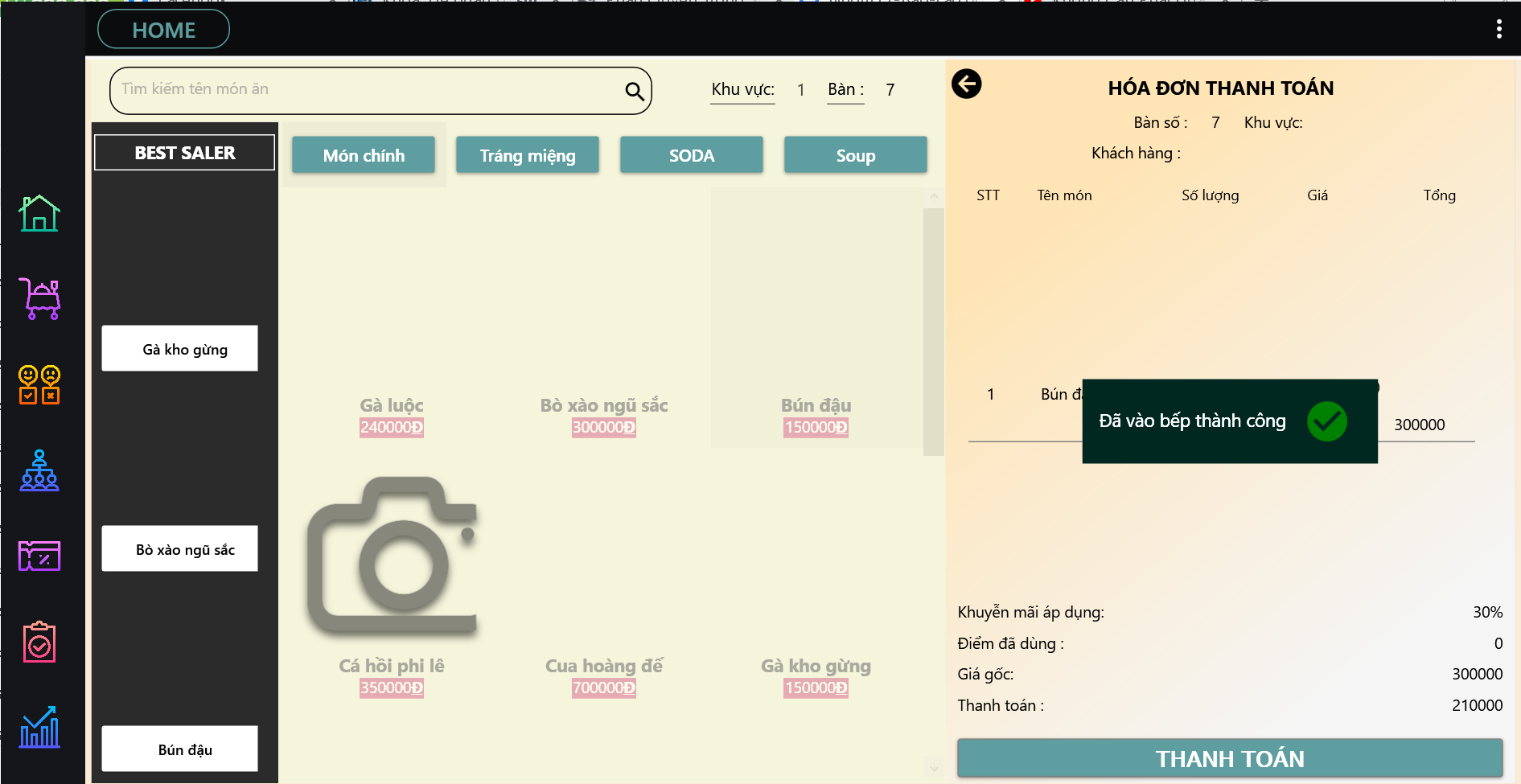
* Đưa con trỏ chuột sang phía bên trái ta sẽ thấy danh sách các mục, chọn lần lượt các mục tương ứng với chức năng cần dùng



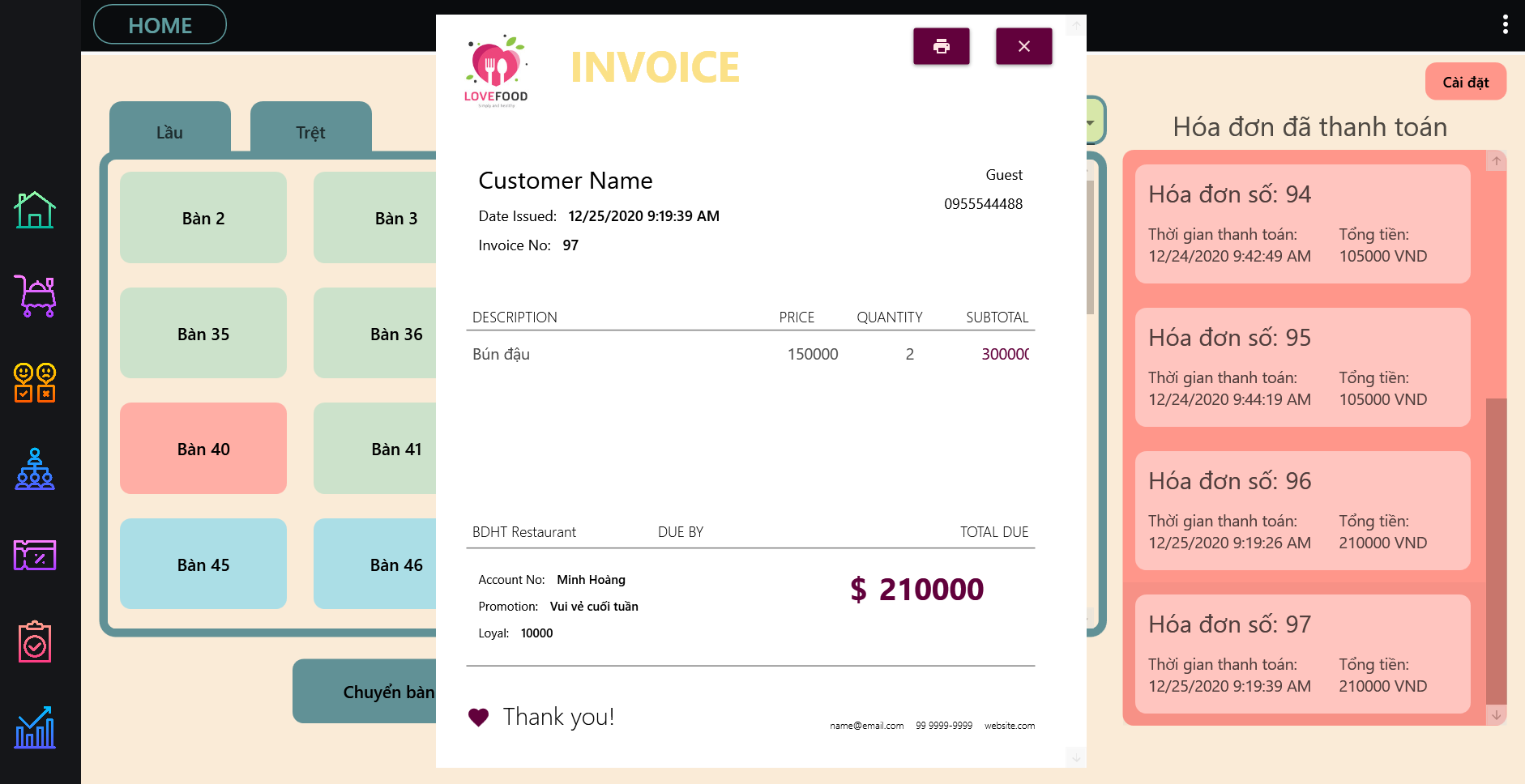
* Cách thêm tài khoản nhân viên trong mục chức năng nhân viên:
* Quá trình bán hàng: Vào mục bán hàng



* Chọn bàn khách hàng đặt > chọn món ăn > sau đó chọn vào bếp để bắt đầu quá trình nấu nướng phục vụ > sau đó chọn thanh toán khi khách hàng ăn xong và kết thúc ăn uống. Ngoài ra còn có chức năng như chuyển bàn, đặt trước và gộp bàn.



* Ta sẽ thấy hóa đơn ở phần Hóa đơn đã thanh toán phía bên phải, chọn vào đó ta sẽ thấy thông tin và có thể in hóa đơn cho khách hàng



* Ngoài ra còn có chức năng thú vị khác như quản lý món ăn, khách hàng, nhân viên, khuyến mãi, chi tiết hóa đơn, và thống kê doanh thu..