MỤC LỤC

[CHƯƠNG I: TỔNG QUAN VỀ HỆ THỐNG 1](#_Toc151595230)

[1.1. Đặc tả bài toán 1](#_Toc151595231)

[1.2. Nghiệp vụ bài toán 2](#_Toc151595232)

[1.3. Mô tả chức năng của bài toán 3](#_Toc151595233)

[1.4. Tính lương nhân viên 5](#_Toc151595234)

[1.5. Tiền sân 6](#_Toc151595235)

[CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 7](#_Toc151595236)

[2.1. Thiết kế cơ sở dữ liệu mức quan niệm 7](#_Toc151595237)

[2.2. Thiết kế cơ sở dữ liệu mức logic 8](#_Toc151595238)

[2.3. Các ràng buộc cần có 9](#_Toc151595239)

[2.4. Cài đặt CSDL và các ràng buộc 11](#_Toc151595240)

[2.5. Các View 15](#_Toc151595241)

[2.6. Các Trigger 16](#_Toc151595242)

[a. Trigger đặt trạng thái đơn hàng 16](#_Toc151595243)

[b. Trigger đặt trạng thái sân bóng thành Đang sử dụng: 16](#_Toc151595244)

[c. Kiểm tra xem số lượng sản phẩm còn lại còn đủ để đáp ứng số lượng sản phẩm khách hàng yêu cầu hay không nếu còn thì cập nhật lại số lượng kho 16](#_Toc151595245)

[d. Kiểm tra trùng lặp số điện thoại của khách hàng 17](#_Toc151595246)

[e. Kiểm tra tên sản phẩm có bị trùng không 18](#_Toc151595247)

[f. Trigger không cho phép tạo hóa đơn sau 22h và trước 5h 19](#_Toc151595248)

[g. Trigger trùng số điện thoại khi thêm nhân viên 20](#_Toc151595249)

[CHƯƠNG III: THIẾT KẾ CÁC CHỨC NĂNG 22](#_Toc151595250)

[3.1. Kết nối Cơ sở dữ liệu 22](#_Toc151595251)

[3.2. Xem thông tin danh mục 23](#_Toc151595252)

[3.2.1. Danh mục nhân viên 23](#_Toc151595253)

[3.2.2. Danh mục ca làm việc 23](#_Toc151595254)

[3.2.3. Danh mục bảng phân ca 23](#_Toc151595255)

[3.2.4. Danh mục hóa đơn 24](#_Toc151595256)

[3.2.5. Danh mục sản phẩm 24](#_Toc151595257)

[3.2.6. Danh mục sân bóng 25](#_Toc151595258)

[3.2.7. Danh mục khách hàng 25](#_Toc151595259)

[3.2.8. Danh mục nhà cung cấp 25](#_Toc151595260)

[3.2.9. Danh mục tài khoản 26](#_Toc151595261)

[3.3. Quản lý 27](#_Toc151595262)

[1. Quản lý khách hàng 27](#_Toc151595263)

[1.1. Tạo khách hàng mới (PROC) 27](#_Toc151595264)

[1.2. Tìm khách hàng bằng số điện thoại (PROC) 28](#_Toc151595265)

[2. Quản lý hóa đơn 29](#_Toc151595266)

[2.1. Thêm hóa đơn (PROC) 29](#_Toc151595267)

[2.2. Hóa đơn chưa thanh toán (VIEW) 29](#_Toc151595268)

[2.3. Thanh toán hóa đơn (PROC) 29](#_Toc151595269)

[3. Quản lý sản phẩm 31](#_Toc151595270)

[3.1. Thêm sản phẩm (PROC) 31](#_Toc151595271)

[3.2. Sửa sản phẩm (PROC) 32](#_Toc151595272)

[3.3. Xóa sản phẩm (PROC) 33](#_Toc151595273)

[3.4. Tìm kiếm sản phẩm (PROC) 34](#_Toc151595274)

[4. Quản lý sân bóng 34](#_Toc151595275)

[4.1. Thêm sân bóng (PROC) 34](#_Toc151595276)

[4.2. Xóa sân bóng (PROC) 35](#_Toc151595277)

[4.2. Sửa sân bóng (PROC) 36](#_Toc151595278)

[5. Thống kê doanh thu và chi phí 36](#_Toc151595279)

[5.1. Doanh thu theo ngày, tháng, năm (FUNC) 36](#_Toc151595280)

[6. Quản lý nhân viên 37](#_Toc151595281)

[6.1. Thêm nhân viên (PROC) 37](#_Toc151595282)

[6.2. Xóa nhân viên (PROC) 39](#_Toc151595283)

[6.3. Tìm kiếm nhân viên (FUNC) 40](#_Toc151595284)

[6.4. Tính lương nhân viên (PROC) 41](#_Toc151595285)

[6.5. Sửa thông tin nhân viên (PROC) 42](#_Toc151595286)

[6.6. Sửa ca làm việc nhân viên (PROC) 43](#_Toc151595287)

[CHƯƠNG IV: TẠO USER VÀ PHÂN QUYỀN 45](#_Toc151595288)

[CHƯƠNG V: THIẾT KẾ GIAO DIỆN HỆ THỐNG 49](#_Toc151595289)

[1. Form đăng nhập 49](#_Toc151595290)

[2. Form dành cho admin 49](#_Toc151595291)

[Thống kê doanh thu 49](#_Toc151595292)

[Xem, thêm, xóa, sửa, tìm, tính lương cho nhân viên 50](#_Toc151595293)

[Xem, thêm, xóa, sửa cho sân bóng 50](#_Toc151595294)

[Xem, thêm, sửa, tìm sản phẩm 51](#_Toc151595295)

[Xem, thêm, xóa tài khoản cho nhân viên 51](#_Toc151595296)

[Xem, tìm khách hàng 52](#_Toc151595297)

[Xem, sửa ca làm của nhân viên 52](#_Toc151595298)

[3. Form dành cho nhân viên 53](#_Toc151595299)

[Trang chủ 53](#_Toc151595300)

[Danh sách sân (nơi đặt sân) 53](#_Toc151595301)

[Tạo hóa đơn hoặc tạo mới khách hàng nếu là khách hàng mới 54](#_Toc151595302)

[Xem, tìm ca theo tên nhân viên 54](#_Toc151595303)

[Xem những hóa đơn chưa thanh toán 55](#_Toc151595304)

**CHƯƠNG I: TỔNG QUAN VỀ HỆ THỐNG**

**1.1. Đặc tả bài toán**

Một chuỗi các sân bóng cần một hệ thống để quản lý nhân viên của sân bóng cũng như việc đặt sân của khách hàng. Hệ thống được xây dựng cho người quản lý và nhân viên của hệ thống sân bóng với các quyền khác nhau. Đối với người dùng là quản lý thì người này có quyền quản lý sân, quản lý doanh thu, quản lý khách. Đối với người dùng là nhân viên thì có quyền đặt sân, xem lịch sân, và tạo hóa đơn đặt sân cho khách hàng, xem doanh thu cá nhân từ việc đặt sân cũng như bán các thứ như nước, thức ăn, … cho khách hàng. Đối với hệ thống quản lý chuỗi sân bóng thì cần lưu những dữ liệu sau:

Sân bóng gồm nhiều sân bóng, mỗi sân bóng phân biệt với nhau bằng mã sân bóng, mỗi sân bóng bào gồm các thông tin kích thước sân bóng, trạng thái sân bóng, mã giá thuê sân, … có nhiều giá thuê, tùy thuộc vào ngày cũng như khung giờ thuê, mỗi giá thuê phân biệt bằng mã giá thuê, gồm những thông tin như ngày thuê, khung giờ thuê, giá.

Sân bóng có nhiều nhân viên, một nhân viên chỉ làm một công việc, một công việc có thể được làm bởi nhiều nhân viên. Các công việc được phân biệt với nhau bằng mã công việc, mỗi công việc gồm các thông tin như mã công việc, tên công việc, mô tả chi tiết, lương. Các nhân viên phân biệt với nhau bằng mã nhân viên, thông tin nhân viên bao gồm mã nhân viên, tên nhân viên, ngày sinh, thưởng, giới tính, địa chỉ, số điện thoại, mã công việc, ngày tuyển dụng.

Mỗi khách hàng sẽ được phân biệt với nhau qua mã khách hàng, thông tin khách hàng bao gồm mã khách hàng, họ tên khách hàng, ngày sinh, giới tính, số điện thoại, địa chỉ.

Ngoài ra, con có thêm các loại sản phẩm khác như nước uống, thức ăn, khăn ướt, … Mỗi loại sản phẩm được phân biệt với nhau qua mã loại sản phẩm. Một sản phẩm chỉ thuộc một loại sản phẩm, một loại sản phẩm có thể có nhiều sản phẩm. Các sản phẩm phân biệt với nhau bằng mã sản phẩm, sản phẩm gồm các thông tin như mã sản phẩm, tên sản phẩm, đơn giá, số lượng còn, mã loại sản phẩm.

Một sản phẩm bán trong sân được cung cấp bởi một nhà cung cấp, một nhà cung cấp có thể cung cấp nhiều loại sản phẩm. Nhà cung cấp được phân biệt bởi mã nhà cung cấp, nhà cung cấp lưu những thông tin như mã nhà cung cấp, tên nhà cung cấp, số điện thoại.

Các nhân viên sẽ được chia ca để làm việc. Một nhân viên có thể làm nhiều ca, một ca có thể có nhiều nhân viên. Các ca làm việc được phân biệt với nhau qua mã ca, mỗi ca có các thông tin như mã ca, ngày, giờ bắt đầu, giờ kết thúc.

Bảng thống kê tài chính sẽ là điều tất yếu để duy trì các hoạt động khác của quán bao gồm mã giao dịch, loại giao dịch (doanh thu, chi phí), ngày giao dịch, số tiền, mô tả.

**1.2. Nghiệp vụ bài toán**

Bài toán quản lí sân bóng đá là một nghiệp quản lý kinh doanh trong lĩnh vực dịch vụ thể thao và giải trí. Nó liên quan đến việc quản lý nhân viên, quản lý sân bóng, quản lý hàng hóa bán trong sân, quản lý doanh thu và điều hành. Bài toán không quá chú trọng vào nhân viên, tuy nhiên nhân viên vẫn là người sẽ làm việc trực tiếp với khách hàng, vì vậy chúng ta cần xây dựng tài liệu đào tạo nghiệp vụ cho nhân viên về quy trình tác nghiệp, văn hóa phục vụ, cách ứng xử, xử lý tình huống, …

* Quản lý nhân viên:

Khi tuyển dụng nhân sự, quán cần đưa ra các yêu cầu cũng như các điều kiện (ví dụ: làm ca đêm, không đi trễ quá số buổi quy định, xin nghỉ trước số ngày quy định, …) mà nhân sự đồng ý trong suốt quá trình làm việc. Quán cần chi trả cho nhân sự mức lương hợp lý. Nhân viên đã được tuyển dụng sẽ trải qua đợt đào tạo về nghiệp vụ, quy trình ở vị trí của bản thân. Người quản lý cần theo dõi và quan sát các nhân sự của mình, chỉ ra những điểm còn thiếu sót để nhân sự khắc phục và lời khen cho những điểm tốt của nhân sự để nhân sự lấy đó làm động lực.

* Quy trình khi khách tới thuê sân:

Khi khách đến sân, đầu tiên nhân viên đứng quầy sẽ chào khách. Sau đó nhân viên sẽ tạo hóa đơn trên hệ thống. Tiếp theo nhân viên hỏi khách muốn thuê sân bao nhiêu, và cho khách biết sân nào còn trống để khách chọn. Sau đó nhân viên hỏi khách có muốn gọi thêm đồ uống hay gì không, khách có thể gọi ngay ban đầu hoặc trong lúc thuê có thể gọi sau. Lúc này nhân viên sẽ dẫn khách ra sân nếu khách chưa biết. Sau khi khách thuê xong ra quầy tính tiền, nhân viên in hóa đơn và báo giá cho khách hàng.

* Quản lý sân bóng, hàng hóa, doanh thu và điều hành sân bóng sẽ qua các giai đoạn:

Giai đoạn đầu, sân bóng phải tìm được được nguồn cung cấp hàng như giày, bóng, nước uống, thức ăn, … Quán phải tìm được đối tác, nhà sản xuất cung cấp hàng hóa uy tín từ đầu vì có chính sách giá cả, giao hàng, chứng từ rõ ràng để tiện cho việc kiểm soát. Sân bóng cần thống kê hàng hóa nhập vào và có kho hàng để lưu trữ, bảo quản và kiểm soát hàng hóa. Cần kiểm tra hàng hóa theo hóa đơn nhập hàng và nhập vào chương trình quản lý sân bóng.

Tiếp theo, ở giai đoạn vận hành, chúng ta cần xây dựng chi phí của từng sản phẩm và lên giá thành cho các sản phẩm đó. Ngoài ra, chúng ta cũng cần xây dựng quy trình làm việc, nghiệp vụ cho các nhân viên thu ngân, dọn vệ sinh.

Cuối cùng, ở giai đoạn kiểm soát, chúng ta cần kiểm soát nhiều thứ như về nghiệp vụ của nhân viên với khách hàng, về độ hài lòng về dịch vụ sân bóng cũng như, chất lượng sân bóng xem có cần cải thiện gì không. Ngoài ra, sân bóng phải kiểm soát được thu chi hằng ngày, nhân viên thu ngân có quyền xem bảng doanh thu theo ngày để tổng kết ca. Nhân viên sẽ phải kiểm tra số tiền thu chi có đúng với doanh thu ca đó không rồi mới bàn giao cho nhân viên ca khác. Người quản lý cần phải nắm các bảng báo cáo phân tích như báo cáo doanh thu (theo ngày, tháng, năm), báo cáo chi phí (chi tiết các khoản chi), báo cáo thất thoát (bị đổ vỡ, hư hỏng, mất mát). Chủ của chuỗi quán cà phê sẽ dựa trên các báo cáo để nắm rõ doanh thu, chi phí, lương nhân sự, … để có thể đưa ra điều chỉnh sao cho có lợi nhất.

**1.3. Mô tả chức năng của bài toán**

* **Chức năng cho nhân viên:**
* ***Đặt sân:***

Nhân viên có thể đặt sân cho khách hàng dựa trên yêu cầu về thời gian và sân bóng cụ thể.

* ***Ghi chú trạng thái đặt sân*** (đã xác nhận, chưa xác nhận).
* ***Xem lịch đặt sân:***

Nhân viên có thể xem lịch đặt sân để kiểm tra sân bóng nào đang được sử dụng và sân nào còn trống trong thời gian cụ thể.

* ***Tạo đơn đặt sân qua điện thoại:***

Nhân viên có thể tiếp nhận cuộc gọi điện thoại từ khách hàng và tạo đơn đặt sân dựa trên thông tin cung cấp.

* ***Xem thông tin cá nhân của khách hàng:***

Nhân viên có thể xem thông tin cá nhân của khách hàng, bao gồm thông tin liên hệ và lịch sử đặt sân.

* ***Báo cáo doanh thu cá nhân:***

Nhân viên có thể xem báo cáo về doanh thu cá nhân của họ từ việc đặt sân.

* **Chức năng cho quản lý của sân bóng:**
* ***Quản lý thông tin sân bóng:***

Thêm, chỉnh sửa, và xóa thông tin về các sân bóng, bao gồm trạng thái và giá cước sử dụng.

* ***Quản lý thông tin nhân viên:***

Thêm, chỉnh sửa, và xóa thông tin nhân viên, bao gồm mã nhân viên, họ tên, và vai trò (nếu có).

* ***Xem lịch đặt sân toàn bộ sân bóng:***

Quản lý có thể xem lịch đặt sân cho tất cả các sân bóng để kiểm tra tình trạng sử dụng và trống.

* ***Xác nhận đơn đặt sân:***

Quản lý có quyền xác nhận các đơn đặt sân để đảm bảo rằng lịch trình không bị trùng lặp.

* ***Quản lý tài chính:***

Theo dõi tài chính của sân bóng, bao gồm thu chi, lợi nhuận, và tạo báo cáo tài chính.

* ***Quản lý thông tin khách hàng:***

Thêm, chỉnh sửa, và xóa thông tin khách hàng, bao gồm mã khách hàng, họ tên, và thông tin liên hệ.

* ***Tạo báo cáo thống kê:***

Tạo các báo cáo thống kê về lịch đặt sân, doanh thu, và hoạt động của sân bóng.

**1.4. Tính lương nhân viên**

1. **Lương cơ bản:** đây là số tiền nhân viên được trả cho việc làm công việc cố định mà họ được giao. Đây là số tiền cố định mà nhân viên được trả hàng tháng (tính theo giờ).
2. **Thời gian làm việc:** đây là số giờ mà nhân viên đã làm việc trong tháng.
3. **Lương theo giờ:** được tính theo công thức: Lương cơ bản x giờ làm việc.
4. **Tiền thưởng:** đây là khoản tiền thưởng mà nhân viên được trả cho các ca làm việc vào ngày lễ, những ngày có nhân viên xin nghỉ đột xuất (mỗi ca làm việc này nhân viên sẽ được cộng một đơn vị vào ca thưởng) hoặc doanh thu quán đạt được chỉ tiêu nhất định (ở đây chỉ tiêu của sân bóng đề ra sẽ được thưởng thêm 5% doanh thu). Công thức tính tiền thưởng được tính như sau: tiền thưởng = doanh thu x 5% + ca thưởng x 4 giờ x 2 x lương cơ bản (sân bóng quy định mỗi ca làm là 4 giờ).

* **TIỀN LƯƠNG = LƯƠNG THEO GIỜ + TIỀN THƯỞNG**

***Ví dụ***: Giả sử sân bóng có 2 loại nhân viên: Nhân viên dọn sân và nhân viên thu ngân. Cả hai đều được trả lương theo giờ làm việc.

Lương nhân viên thu ngân: 25.000/giờ

Lương nhân viên dọn sân: 27.000/giờ

Vào tháng này, nhân viên thu ngân A làm 41 ca và 5 ca thưởng, còn nhân viên dọn sân làm 32 ca và 3 ca thưởng.

Do đó lương của nhân viên **A**:

Lương theo giờ = số giờ làm việc x lương cơ bản = 41ca x 4giờ x 25.000 đồng/giờ = 4.100.000 đồng

Lương thưởng = ca thưởng x 4giờ x 2 x lương cơ bản = 5 x 4 x 2 x 25.000 = 1.000.000 đồng

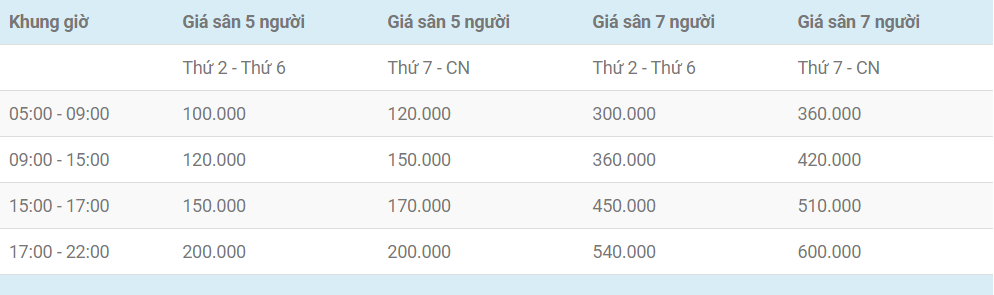
=> tổng lương = 5.100.000 đồng

Tương tự với nhân viên **B** ta có Tổng lương = 4.104.000 đồng.

***Trong trường hợp doanh thu quán đạt chỉ tiêu thì nhân viên trong quán được thưởng thêm 5% doanh thu của quán***

**1.5. Tiền sân**

Đây là số tiền khách hàng phải trả cho mỗi giờ thuê, sẽ tùy vào kích thước sân, mỗi sân sẽ có giá khác nhau.



* **Số giờ thuê:** Số giờ mà khách đã thuê sân.
* **Số tiền thanh toán:** Tổng tiền = Giá sân x Số giờ thuê

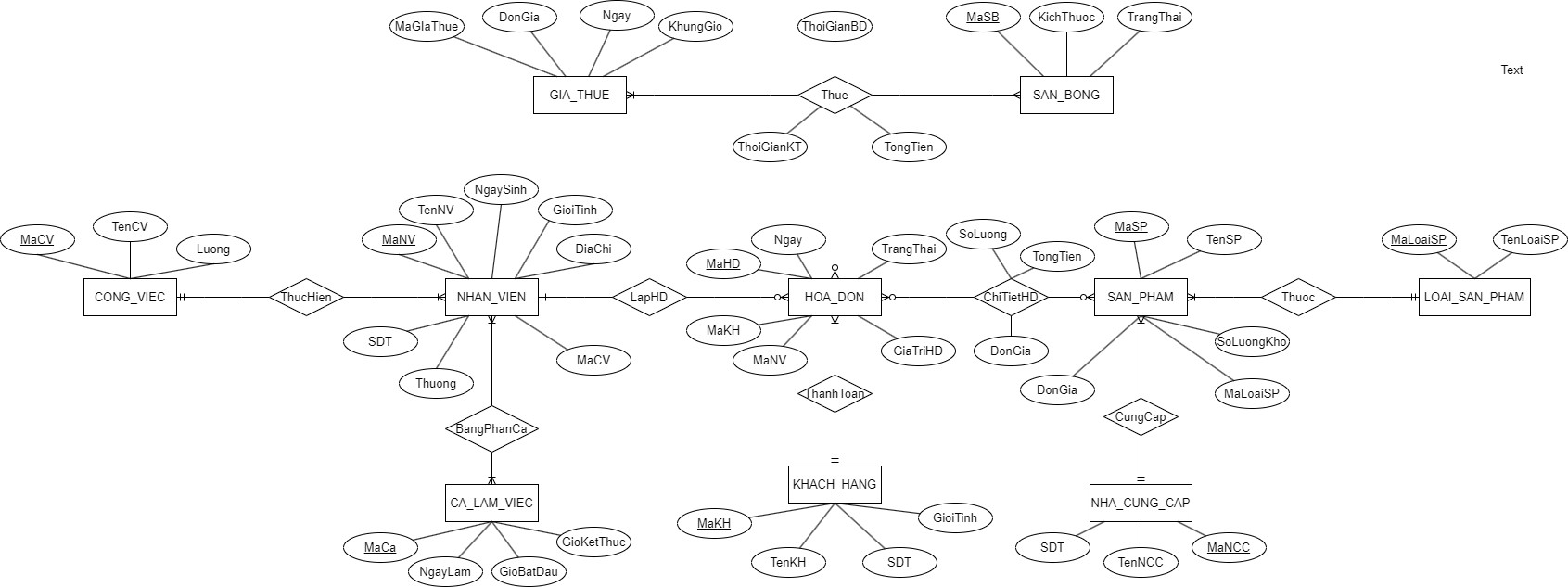
***Ví dụ:*** Khách thuê vào thứ 7, sân 5 người, thuê từ 16h đến 20h

Số tiền khách phải trả là: 100.000 + 120.000 x 3 = 460.000 đồng

**CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**2.1. Thiết kế cơ sở dữ liệu mức quan niệm**

Từ mô tả về dữ liệu cần có ở phần mô tả của bài toán ta hình thành được sơ đồ thực thể kết hợp (ERD).



**2.2. Thiết kế cơ sở dữ liệu mức logic**

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

1. CongViec(MaCV, TenCV, Luong)
2. NhanVien(MaNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, Thuong, MaCV, SoCa)
3. BangPhanCa(MaCa, MaNV, NgayLam)
4. CaLamViec(MaCa, NgayLam, GioBatDau, GioKetThuc)
5. HoaDon(MaHD, Ngay, TrangThai, GiaTriHD ,MaKH, MaNV)
6. Thue(MaHD, MaSB, MaGiaThue, ThoiGianBD, ThoiGianKT, TongTien)
7. SanBong(MaSB, KichThuoc, TrangThai)
8. GiaThue(MaGiaThue. KichThuoc, DonGia, NgayThue, KhungGio)
9. KhachHang(MaKH, TenKH, SDT, GioiTinh
10. ChiTietHoaDon(MaHD, MaSP, SL, DonGia, TongTien)
11. SanPham(MaSP, TenSP, DonGia, SoLuongKho, MaLoaiSP, MaNCC)
12. LoaiSanPham(MaLoaiSP, TenLoai)
13. NhaCungCap(MaNCC, TenNCC, DiaChi)

**2.3. Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| **STT** | **BẢNG** | **RÀNG BUỘC** |
| 1 | CongViec | Khóa chính MaCV, CHECK Luong > 0 |
| 2 | NhanVien | Khóa chính MaNV,  Khóa ngoại MaCV đến bảng CongViec,  NOT NULL TenNV, NOT NULL SDT,  CHECK NgaySinh (>=18 tuổi), CHECK SDT (=10), |
| 3 | KhachHang | Khoá chính MaKH, NOT NULL TenKH, NOT NULL SDT, |
| 4 | SanBong | Khóa chính MaSB,  NOT NULL KichThuoc,  NOT NULL TrangThai |
| 5 | LoaiSanPham | Khoá chính MaLoaiSP,  NOT NULL TenLoaiSP |
| 6 | SanPham | Khoá chính MaSP,  Khoá ngoại MaLoaiSP đến bảng LoaiSanPham,  NOT NULL TenSP,  NOT NULL DonGia |
| 7 | HoaDon | Khoá chính MaHD,  Khoá ngoại MaKH đến bảng KhachHang, Khoá ngoại MaNV đến bảng NhanVien, |
| 8 | ChiTietHoaDon | Khoá chính (MaHD, MaSP),  Khoá ngoại MaHD đến bảng HoaDon, Khoá ngoại MaSP đến bảng SanPham, CHECK SL(>0) |
| 9 | Thue | Khóa chính (MaHD, MaSB, MaGiaThue),  Khóa ngoại MaSB đến bảng SanBong,  Khóa ngoại MaHD đến bảng, HoaDon,  Khóa ngoại MaGiaThue đến bảng GiaThue |
| 10 | CaLamViec | Khóa chính (MaCa, NgayLam),  CHECK NgayLam(>=0) |
| 11 | BangPhanCa | Khóa chính (MaCa, NgayLam, MaNV),  Khóa ngoại (MaCa, NgayLam) đến bảng CaLamViec, Khóa ngoại MaNV đến bảng NhanVien |
| 12 | GiaThue | Khóa chính MaGiaThue,  NOT NULL KichThuoc,  NOT NULL DonGia,  NOT NULL NgayThue,  NOT NULL KhungGio, |
| 13 | NhaCungCap | Khoá chính MaNCC,  NOT NULL TenNCC,  NOT NULL SDT,  CHECK SDT(=10) |

**2.4. Cài đặt CSDL và các ràng buộc**

***Bảng công việc***

CREATE TABLE CongViec(

MaCV INT IDENTITY CONSTRAINT PK\_CongViec PRIMARY KEY,

TenCV nvarchar(50) NOT NULL,

Luong float check (Luong > 0)

)

***Bảng nhân viên***

CREATE TABLE NhanVien(

MaNV INT IDENTITY CONSTRAINT PK\_NhanVien PRIMARY KEY,

TenNV NCHAR(50) NOT NULL,

NgaySinh date check (DATEDIFF(year, NgaySinh, GETDATE())>=18),

GioiTinh nvarchar(3),

DiaChi nvarchar(100),

SDT nchar(11) check (len(SDT)=10),

MaCV INT CONSTRAINT FK\_NhanVien\_CV FOREIGN KEY REFERENCES CongViec(MaCV),

SoCa int,

Thuong int,

)

***Bảng khách hàng***

CREATE TABLE KhachHang(

MaKH INT IDENTITY CONSTRAINT PK\_KhachHang PRIMARY KEY,

TenKH nvarchar(50) NOT NULL,

SDT nchar(11) NOT NULL,

GioiTinh NVARCHAR(10)

)

***Bảng nhà cung cấp***

CREATE TABLE NhaCungCap(

MaNCC INT IDENTITY CONSTRAINT PK\_NhaCungCap PRIMARY KEY,

TenNCC NCHAR(50),

DiaChi NCHAR(50),

SDT NCHAR(10) CHECK (LEN(SDT)=10)

)

***Bảng loại sản phẩm***

CREATE TABLE LoaiSanPham(

MaLoaiSP INT IDENTITY CONSTRAINT PK\_LoaiSanPham PRIMARY KEY,

TenLoaiSP nvarchar(50) NOT NULL

)

***Bảng sản phẩm***

CREATE TABLE SanPham (

MaSP INT IDENTITY PRIMARY KEY,

TenSP NVARCHAR(255) NOT NULL,

DonGia FLOAT,

SoLuongKho INT DEFAULT 0,

MaLoaiSP INT,

MaNCC INT,

FOREIGN KEY (MaLoaiSP) REFERENCES LoaiSanPham(MaLoaiSP),

FOREIGN KEY (MaNCC) REFERENCES NhaCungCap(MaNCC)

);

***Bảng chi tiết thuê***

CREATE TABLE Thue (

MaHD INT,

MaSB INT,

MaGiaThue INT,

ThoiGianBD Time,

ThoiGianKT Time,

TongTien FLOAT,

PRIMARY KEY (MaHD, MaSB, MaGiaThue),

FOREIGN KEY (MaHD) REFERENCES HoaDon(MaHD),

FOREIGN KEY (MaSB) REFERENCES SanBong(MaSB),

FOREIGN KEY (MaGiaThue) REFERENCES GiaThue(MaGiaThue)

);

***Bảng sân bóng***

CREATE TABLE SanBong (

MaSB INT IDENTITY PRIMARY KEY,

KichThuoc NVARCHAR(50) NOT NULL,

TrangThai NVARCHAR(50) NOT NULL

);

***Bảng hoá đơn***

CREATE TABLE HoaDon (

MaHD INT IDENTITY PRIMARY KEY,

Ngay DATE,

TrangThai INT,

GiaTriHD FLOAT,

MaKH INT,

MaNV INT,

FOREIGN KEY (MaKH) REFERENCES KhachHang(MaKH),

FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)

);

***Bảng chi tiết hoá đơn***

CREATE TABLE ChiTietHoaDon(

MaHD INT CONSTRAINT FK\_ChiTietHD\_HD FOREIGN KEY REFERENCES HoaDon(MaHD),

MaSP INT CONSTRAINT FK\_ChiTietHD\_SP FOREIGN KEY REFERENCES SanPham(MaSP),

SL int check (SL > 0),

DonGia float,

TongTien float

CONSTRAINT PK\_ChiTietHD PRIMARY KEY (MaHD, MaSP)

)

***Bảng các ca làm việc***

CREATE TABLE CaLamViec(

MaCa INT IDENTITY,

NgayLam date check (DATEDIFF(day, NgayLam, GETDATE())>=0),

GioBatDau TIME,

GioKetThuc TIME,

CONSTRAINT PK\_CaLamViec PRIMARY KEY (MaCa, NgayLam)

)

***Bảng phân ca làm cho nhân viên***

CREATE TABLE BangPhanCa(

MaCa INT IDENTITY,

MaNV INT CONSTRAINT FK\_PhanCa\_NV FOREIGN KEY REFERENCES NhanVien(MaNV),

NgayLam date,

CONSTRAINT PK\_BangPhanCa PRIMARY KEY (MaCa, MaNV, NgayLam),

CONSTRAINT FK\_PhanCa\_Ca FOREIGN KEY (MaCa,NgayLam) REFERENCES CaLamViec(MaCa,NgayLam)

)

***Bảng giá thuê***

CREATE TABLE GiaThue (

MaGiaThue INT IDENTITY PRIMARY KEY,

KichThuoc VARCHAR(50) NOT NULL,

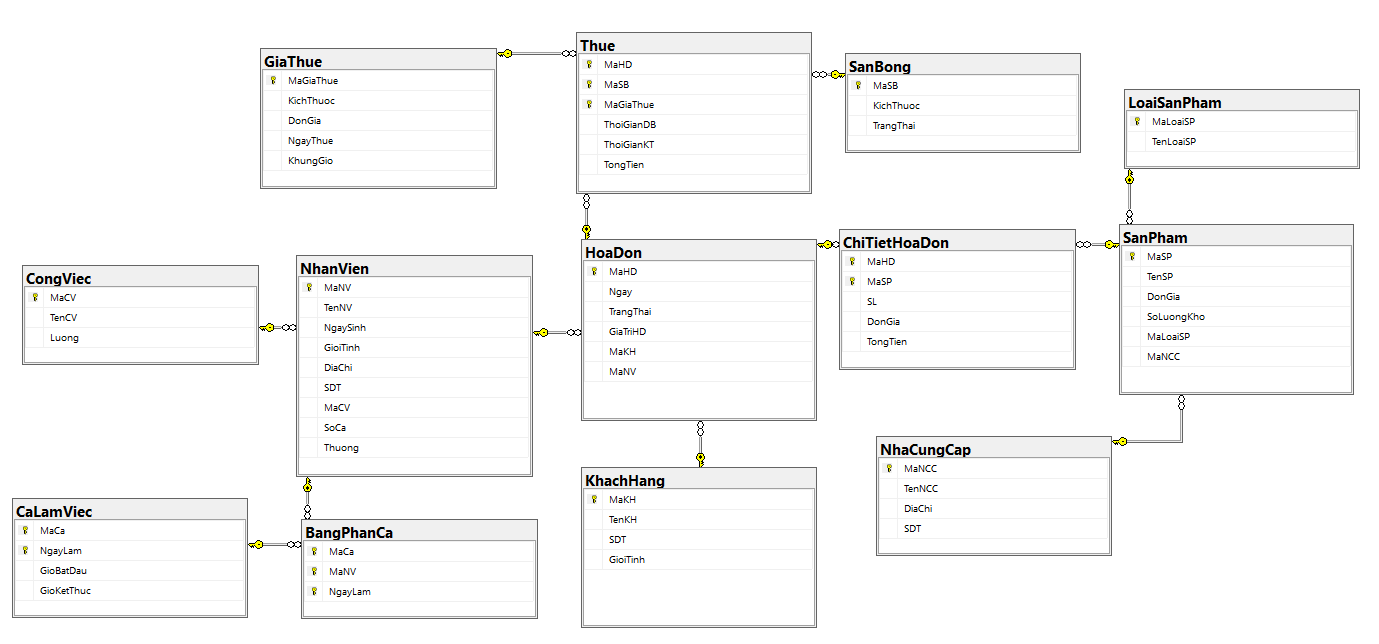
DonGia FLOAT NOT NULL,

NgayThue INT,

KhungGio Time,

KhungGioKT TIME

);



**2.5. Các View**

1. **Xem danh sách sản phẩm**

GO

CREATE VIEW V\_DanhSachSanPham AS

SELECT MaSP, TenSP, DonGia, SoLuongKho, MaLoaiSP, MaNCC

FROM dbo.SanPham

1. **Xem danh sách loại sản phẩm**

GO

CREATE VIEW V\_DanhSachLoaiSanPham AS

SELECT MaLoaiSP, TenLoaiSP

FROM dbo.LoaiSanPham

1. **Xem số lượng sản phẩm đã bán trong ngày**

GO

CREATE VIEW V\_SoLuongSanPhamDaBanTrongNgay AS

SELECT sp.MaSP, sp.TenSP, SUM(cthd.SL) AS SoLuongDaBan

FROM dbo.SanPham sp

JOIN dbo.ChiTietHoaDon cthd ON sp.MaSP = cthd.MaSP JOIN dbo.HoaDon hd ON cthd.mahd = hd.MaHD

GROUP BY sp.MaSP, sp.TenSP

1. **Xem thông tin nhân viên**

GO

CREATE VIEW V\_ThongTinNhanVien AS

SELECT nv.TenNV, nv.NgaySinh, nv.SDT, nv.SoCa, nv.Thuong, cv.TenCV

FROM dbo.NhanVien nv

JOIN dbo.CongViec cv ON nv.MaCV = cv.MaCV

1. **Xem danh sách sân bóng:**

GO

CREATE VIEW V\_DanhSachSanBong AS

SELECT MaSB, KichThuoc, TrangThai

FROM dbo.SanBong

1. **Xem danh sách khách hàng**

GO

CREATE VIEW V\_DanhSachKhachHang AS

SELECT MaKH, TenKH, GioiTinh, SDT

FROM dbo.KhachHang

1. **Xem danh sách nhà cung cấp:**

GO

CREATE VIEW V\_DanhSachNhaCungCap AS

SELECT MaNCC, TenNCC, DiaChi, SDT

FROM NhaCungCap

1. **Xem danh sách hóa đơn chưa thanh toán:**

GO

CREATE VIEW V\_DanhSachHoaDonChuaThanhToan AS

SELECT HoaDon.MaHD, HoaDon.Ngay, Thue.MaSB, KhachHang.TenKH

FROM HoaDon

JOIN Thue on Thue.MaHD = HoaDon.MaHD

JOIN KhachHang on KhachHang.MaKH = HoaDon.MaKH

WHERE TrangThai = 0

**2.6. Các Trigger**

**a. Trigger đặt trạng thái đơn hàng**

|  |
| --- |
| CREATE TRIGGER set\_HoaDon\_TrangThai ON HoaDon  FOR INSERT AS  BEGIN  UPDATE HoaDon  SET TrangThai = 0  WHERE MaHD IN (SELECT MaHD FROM INSERTED)  END; |

**b. Trigger đặt trạng thái sân bóng thành Đang sử dụng:**

|  |
| --- |
| CREATE TRIGGER TG\_SetTrangThaiSan  ON Thue  AFTER INSERT  AS  BEGIN  -- Cập nhật giá trị TrangThai trong bảng SanBong  UPDATE SanBong  SET TrangThai = 'Đang sử dụng'  FROM SanBong  INNER JOIN inserted i ON SanBong.MaSB = i.MaSB;  END; |

**c. Kiểm tra xem số lượng sản phẩm còn lại còn đủ để đáp ứng số lượng sản phẩm khách hàng yêu cầu hay không nếu còn thì cập nhật lại số lượng kho**

|  |
| --- |
| -- sau khi thêm các sản phẩm vào hóa đơn, nếu hết thì sẽ quăng lỗi, không thì sẽ cập nhật lại số lượng kho  GO  CREATE TRIGGER TG\_SPHetHangVaCapNhatSLSP  ON ChiTietHoaDon INSTEAD OF INSERT, UPDATE  AS  BEGIN  DECLARE @MaHD INT, @MaSP INT, @SL INT, @TongTien FLOAT, @SLExisted INT  --Gán giá trị đã insert hoặc sau khi update cho các biến  SELECT @MaHD = i.MaHD, @MaSP = i.MaSP, @SL = i.SL, @TongTien = i.TongTien  FROM inserted i  DECLARE @CurrSL INT, @DonGia FLOAT  -- Lấy ra số lượng sản phầm đang còn trong kho  SELECT @CurrSL = SanPham.SoLuongKho, @DonGia = SanPham.DonGia  FROM SanPham  WHERE SanPham.MaSP = @MaSP  -- xử lý số lượng dành cho update  SELECT @SLExisted = ChiTietHoaDon.SL  FROM inserted, ChiTietHoaDon  WHERE inserted.MaHD = ChiTietHoaDon.MaHD AND inserted.MaSP = ChiTietHoaDon.MaSP  IF NOT EXISTS (SELECT 1 FROM ChiTietHoaDon WHERE @MaHD = ChiTietHoaDon.MaHD AND @MaSP = ChiTietHoaDon.MaSP)  BEGIN  SET @SLExisted = 0  END  IF(@SL > @CurrSL + @SLExisted)  BEGIN  RAISERROR('Sản phẩm không đủ để thêm',16,1)  RETURN  END  ELSE  BEGIN  IF EXISTS (SELECT 1 FROM ChiTietHoaDon WHERE @MaHD = ChiTietHoaDon.MaHD AND @MaSP = ChiTietHoaDon.MaSP)  BEGIN  SET @TongTien = @SL\*@DonGia  UPDATE ChiTietHoaDon  SET SL = SL + @SL - @SLExisted,  TongTien = TongTien + @TongTien  WHERE ChiTietHoaDon.MaHD = @MaHD AND ChiTietHoaDon.MaSP = @MaSP  UPDATE SanPham  SET SoLuongKho = SoLuongKho - @SL+ @SLExisted  END  ELSE  BEGIN  INSERT INTO ChiTietHoaDon(MaHD, MaSP, SL, TongTien)  VALUES(@MaHD, @MaSP, @SL, @TongTien)  UPDATE SanPham  SET SoLuongKho = SanPham.SoLuongKho - i.SL  FROM SanPham  INNER JOIN inserted i ON SanPham.MaSP = i.MaSP;  END  END  END |

***Thực hiện bắt trigger ở C#***

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  try  {  int MaSP;  Product selected = comboBox2.SelectedItem as Product;  MaSP = (int)selected.MaSP;  int MaHD = BillDAO.GetBillIdByPitchId(MaSB);  ProductDAO PD = new ProductDAO();  int SoLuong = (int)numericUpDown1.Value;  PD.addProductToBillDetail(MaHD.ToString(), MaSP.ToString(), SoLuong);  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  catch(Exception ex1)  {  MessageBox.Show(ex1.Message);  }  showBillProduct(MaSB);  } |

**d. Kiểm tra trùng lặp số điện thoại của khách hàng**

|  |
| --- |
| CREATE TRIGGER TG\_TrungSDT ON KhachHang  INSTEAD OF INSERT, UPDATE AS  BEGIN  DECLARE @TenKH NVARCHAR(50), @SDT NVARCHAR(11), @GioiTinh NVARCHAR(10);  SELECT @TenKH = inserted.TenKH ,@SDT = inserted.SDT, @GioiTinh = inserted.GioiTinh FROM inserted;  -- Kiểm tra số điện thoại vừa thêm có bị trùng lặp  IF EXISTS (SELECT 1 FROM KhachHang WHERE SDT = @SDT)  BEGIN  -- Nếu trùng thì rollback  --PRINT N'Số điện thoại đã tồn tại'  RAISERROR('Số điện thoại đã tồn tại!',16,1)  ROLLBACK;  END  ELSE  BEGIN  INSERT INTO KhachHang(TenKH, SDT, GioiTinh)  VALUES(@TenKH, @SDT, @GioiTinh)  END  END |

***Thực hiện bắt trigger ở C#***

|  |
| --- |
| private void button2\_Click(object sender, EventArgs e)  {  string TenKH = textBox1.Text;  string SDT = textBox2.Text;  string GioiTinh = comboBox1.Text;  try  {  CustomerDAO CD = new CustomerDAO();  CD.addCustomer(TenKH, SDT, GioiTinh);  MessageBox.Show("Thêm thành công khách hàng: " + TenKH + ", Có số điện thoại " + SDT);  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

**e. Kiểm tra tên sản phẩm có bị trùng không**

|  |
| --- |
| CREATE TRIGGER TG\_TrungTenSP  ON dbo.SanPham  AFTER INSERT AS  BEGIN  -- Kiểm tra tên sản phẩm vừa thêm có bị trùng lặp  IF EXISTS (  SELECT \*  FROM inserted i WHERE EXISTS ( SELECT \*  FROM dbo.SanPham sp  WHERE sp.TenSP = i.TenSP  )  )  BEGIN  -- Nếu trùng thì rollback  RAISERROR ('Tên sản phẩm bị trùng', 16, 1)  ROLLBACK;  END  END |

***Thực hiện bắt trigger ở C#***

|  |
| --- |
| private void button13\_Click(object sender, EventArgs e)  {  string TenSP = textBox16.Text;  string SoLuong = textBox17.Text;  string DonGia = textBox8.Text;  int MaNCC = int.Parse(textBox14.Text);  int MaLoaiSP = int.Parse((comboBox2.SelectedItem as Category).MaLoaiSP);  ProductDAO PD = new ProductDAO();  try  {  if (PD.addProduct(TenSP, DonGia, SoLuong, MaLoaiSP, MaNCC)) ;  {  MessageBox.Show("Thêm sản phẩm thành công!");  LoadProductList();  }  }  catch(SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

**f. Trigger không cho phép tạo hóa đơn sau 22h và trước 5h**

|  |
| --- |
| CREATE TRIGGER TG\_ValidTimeToInsertBill  ON HoaDon  INSTEAD OF INSERT  AS  BEGIN  DECLARE @MaHD INT, @Ngay DATE, @TrangThai INT, @GiaTriHD FLOAT, @MaKH INT, @MaNV INT  DECLARE @CurrentTime TIME  SELECT @CurrentTime = FORMAT(SYSDATETIME(), 'HH:mm:ss');  IF (@CurrentTime >= '22:00:00' OR @CurrentTime <= '05:00:00')  BEGIN  RAISERROR('Không được thuê sân sau 22h và trước 5h',16,1);  RETURN  END  ELSE  BEGIN  SELECT @Ngay = i.Ngay, @TrangThai = i.TrangThai, @GiaTriHD = i.GiaTriHD, @MaKH = i.MaKH, @MaNV = i.MaNV  FROM inserted i  INSERT INTO HoaDon(Ngay, TrangThai, GiaTriHD, MaKH, MaNV)  VALUES(@Ngay, @TrangThai, @GiaTriHD, @MaKH, @MaNV)  END  END |

***Thực hiện bắt trigger ở C#***

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  int MaKH = CustomerDAO.GetCustomerIdByPhoneNum(textBox3.Text);  int MaNV = int.Parse(this.staff.MaNV);  try  {  BillDAO.addBill(DateTime.Now.ToString("yyyy-MM-dd"), "0", "0", MaKH.ToString(), MaNV.ToString());  int MaHD = BillDAO.GetNewestBillId();  RentDAO.addBillAndPitchToRent(MaHD.ToString(),this.MaSB.ToString());  MessageBox.Show("Thêm thành công hóa đơn cho sân " + this.MaSB.ToString());  this.Close();  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

**g. Trigger trùng số điện thoại khi thêm nhân viên**

|  |
| --- |
| GO  CREATE TRIGGER TG\_InsertStaff  ON NhanVien  INSTEAD OF INSERT  AS  BEGIN  DECLARE @TenNV NVARCHAR(20), @NgaySinh NVARCHAR(20), @GioiTinh NVARCHAR(3), @DiaChi NVARCHAR(50), @SDT NVARCHAR(11), @MaCV INT  -- Lấy số điện thoại từ bản ghi vừa thêm vào  SELECT @TenNV = I.TenNV, @NgaySinh = I.NgaySinh, @GioiTinh = I.GioiTinh, @DiaChi = I.DiaChi ,@SDT = I.SDT, @MaCV = I.MaCV  FROM inserted I    IF EXISTS (SELECT 1 FROM NhanVien WHERE SDT = @SDT)  BEGIN  RAISERROR('Số điện thoại đã tồn tại', 16,1)  ROLLBACK  END  INSERT INTO NhanVien(TenNV, NgaySinh, GioiTinh, DiaChi, SDT, SoCa, Thuong, MaCV)  VALUES(@TenNV, @NgaySinh, @GioiTinh, @DiaChi, @SDT, 0, 0,@MaCV)  END |

***Thực hiện bắt trigger ở C#***

|  |
| --- |
| private void button2\_Click(object sender, EventArgs e)  {  string TenNV = textBox2.Text;  string NgaySinh = textBox4.Text;  string GioiTinh = textBox3.Text;  string DiaChi = textBox5.Text;  string SDT = textBox6.Text;  string JobId = (comboBox1.SelectedItem as Job).MaCV;  StaffDAO SD = new StaffDAO();  try  {  if (SD.addStaff(TenNV, NgaySinh, GioiTinh, DiaChi, SDT, JobId))  {  MessageBox.Show("Thêm nhân viên thành công!");  LoadStaffList();  }  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }    } |

**CHƯƠNG III: THIẾT KẾ CÁC CHỨC NĂNG**

**3.1. Kết nối Cơ sở dữ liệu**

|  |
| --- |
| public class Database  {  public static string connectionString = @"Data Source=ACERNITRO5EAGLE;Initial Catalog=QLSANBONG;Persist Security Info=True;User ID=" + UserLoginInfo.username + ";Password=" + UserLoginInfo.password;  private static string connectionStringAdmin = @"Data Source=ACERNITRO5EAGLE;Initial Catalog=QLSANBONG;Integrated Security=True";  private SqlConnection conn = new SqlConnection(connectionString);  private SqlConnection connAdmin = new SqlConnection(connectionStringAdmin);  public SqlConnection getConnection  {  get  {  return conn;  }  }  public void openConnection()  {  if (conn.State == ConnectionState.Closed)  {  conn.Open();  }  }  public void closeConnection()  {  if (conn.State == ConnectionState.Open)  {  conn.Close();  }  }  public SqlConnection getConnectionAdmin  {  get  {  return connAdmin;  }  }  public void openConnectionAdmin()  {  if (conn.State == ConnectionState.Closed)  {  connAdmin.Open();  }  }  public void closeConnectionAdmin()  {  if (conn.State == ConnectionState.Open)  {  connAdmin.Close();  }  }  public static void updateConnectionString()  {  connectionString = @"Data Source=ACERNITRO5EAGLE;Initial Catalog=QLSANBONG;Persist Security Info=True;User ID=" + UserLoginInfo.username + ";Password=" + UserLoginInfo.password;  }  public Database()  {    }  } |

**3.2. Xem thông tin danh mục**

**3.2.1. Danh mục nhân viên**

|  |
| --- |
| CREATE VIEW V\_ThongTinNhanVien AS  SELECT nv.MaNV, nv.TenNV, nv.NgaySinh, nv.GioiTinh, nv.DiaChi, nv.SDT, nv.SoCa, nv.Thuong, cv.TenCV  FROM dbo.NhanVien nv JOIN dbo.CongViec cv ON nv.MaCV = cv.MaCV |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getStaff()  {  sql = "select \* from V\_ThongTinNhanVien";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  return dt;  } |

**3.2.2. Danh mục ca làm việc**

|  |
| --- |
| CREATE VIEW V\_CaLamViec AS  SELECT \*  FROM CaLamViec |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getShift()  {  sql = "select \* from V\_CaLamTrongNgay";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaCa"].ColumnName = "Mã Ca";  dt.Columns["NgayLam"].ColumnName = "Thứ";  dt.Columns["GioBatDau"].ColumnName = "Giờ bắt đầu";  dt.Columns["GioKetThuc"].ColumnName = "Giờ kết thúc";    return dt;  } |

**3.2.3. Danh mục bảng phân ca**

|  |
| --- |
| CREATE VIEW V\_BangPhanCa AS  SELECT BangPhanCa.MaCa, NhanVien.MaNV, NhanVien.TenNV  FROM BangPhanCa, NhanVien  WHERE BangPhanCa.MaNV = NhanVien.MaNV |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getDivision()  {  sql = "select \* from V\_BangPhanCa";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaNV"].ColumnName = "Mã Nhân Viên";  dt.Columns["TenNV"].ColumnName = "Tên Nhân Viên";  dt.Columns["MaCa"].ColumnName = "Mã Ca";  return dt;  } |

**3.2.4. Danh mục hóa đơn**

|  |
| --- |
| CREATE VIEW V\_DanhSachHoaDon AS  SELECT \*  FROM HoaDon |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getBill()  {  sql = "select \* from V\_DanhSachHoaDon";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaHD"].ColumnName = "Mã Hóa Đơn";  dt.Columns["Ngay"].ColumnName = "Ngày";  dt.Columns["TrangThai"].ColumnName = "Trạng Thái";  dt.Columns["GiaTriHD"].ColumnName = "Tổng tiền";  dt.Columns["MaKH"].ColumnName = "Khách Hàng";  dt.Columns["MaNV"].ColumnName = "Nhân Viên";  return dt; |

**3.2.5. Danh mục sản phẩm**

|  |
| --- |
| CREATE VIEW V\_DanhSachSanPham AS  SELECT MaSP, TenSP, DonGia, SoLuongKho  FROM dbo.SanPham |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getProduct()  {  sql = "select \* from V\_DanhSachSanPham";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaSP"].ColumnName = "Mã Sản Phẩm";  dt.Columns["TenSP"].ColumnName = "Tên Sản Phẩm";  dt.Columns["SoLuongKho"].ColumnName = "Số Lượng";  dt.Columns["DonGia"].ColumnName = "Đơn Giá";  return dt;  } |

**3.2.6. Danh mục sân bóng**

|  |
| --- |
| CREATE VIEW V\_DanhSachSanBong AS  SELECT MaSB, KichThuoc, TrangThai  FROM dbo.SanBong |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getPitch()  {  sql = "select \* from V\_DanhSachSanBong";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaSB"].ColumnName = "Mã sân";  dt.Columns["KichThuoc"].ColumnName = "Kích thước";  dt.Columns["TrangThai"].ColumnName = "Trạng Thái";  return dt;  } |

**3.2.7. Danh mục khách hàng**

|  |
| --- |
| CREATE VIEW V\_DanhSachKhachHang AS  SELECT MaKH, TenKH, GioiTinh, SDT  FROM dbo.KhachHang |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getCustomer()  {  sql = "select \* from V\_DanhSachKhachHang";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaKH"].ColumnName = "Mã Khách";  dt.Columns["TenKH"].ColumnName = "Tên Khách";  dt.Columns["GioiTinh"].ColumnName = "Giới Tính";  dt.Columns["SDT"].ColumnName = "Số Điện Thoại";  return dt;  } |

**3.2.8. Danh mục nhà cung cấp**

|  |
| --- |
| CREATE VIEW V\_DanhSachNhaCungCap AS  SELECT MaNCC, TenNCC, DiaChi, SDT  FROM NhaCungCap |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getProvider()  {  sql = "select \* from V\_DanhSachNhaCungCap";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaNCC"].ColumnName = "Mã Nhà Cung Cấp";  dt.Columns["TenNCC"].ColumnName = "Tên Nhà Cung Cấp";  dt.Columns["DiaChi"].ColumnName = "Địa Chỉ";  dt.Columns["SDT"].ColumnName = "Số Điện Thoại";  return dt;  } |

**3.2.9. Danh mục tài khoản**

|  |
| --- |
| CREATE VIEW V\_DanhSachTaiKhoan AS  SELECT \*  FROM TAIKHOAN |

***Thực thi trên C#***

|  |
| --- |
| public DataTable getAccounts()  {  sql = "select \* from V\_DanhSachTaiKhoan";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  return dt;  } |

**3.3. Quản lý**

**1. Quản lý khách hàng**

**1.1. Tạo khách hàng mới (PROC)**

|  |
| --- |
| CREATE PROCEDURE proc\_AddCustomer  @TenKH nvarchar(50),  @SDT nchar(11),  @GioiTinh NVARCHAR(10)  AS  BEGIN  INSERT INTO KhachHang (TenKH, SDT, GioiTinh)  VALUES (@TenKH, @SDT, @GioiTinh)  END |

***Thực thi ở C#***

|  |
| --- |
| public bool addCustomer(string TenKH, string SDT, string GioiTinh)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_AddCustomer", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure    cmd.Parameters.Add(new SqlParameter("@TenKH", SqlDbType.NVarChar, 50));  cmd.Parameters.Add(new SqlParameter("@GioiTinh", SqlDbType.NVarChar, 3));  cmd.Parameters.Add(new SqlParameter("@SDT", SqlDbType.NChar, 11));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@TenKH"].Value = TenKH;  cmd.Parameters["@GioiTinh"].Value = GioiTinh;  cmd.Parameters["@SDT"].Value = SDT;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**1.2. Tìm khách hàng bằng số điện thoại (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_SearchCustomerByPhoneNumber  @SDT NVARCHAR(50)  AS  BEGIN  SELECT \* FROM KhachHang  WHERE SDT LIKE N'%' + @SDT + '%';  END |

***Thực thi ở C#***

|  |
| --- |
| public static DataTable searchCustomerByPhoneNum(string SDT)  {  Database db = new Database();  List<Staff> staffList = new List<Staff>();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_SearchCustomerByPhoneNumber", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@SDT", SqlDbType.NVarChar, 50));  cmd.Parameters["@SDT"].Value = SDT; // Tên muốn tìm  // Thực hiện stored procedure  DataTable dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaKH"].ColumnName = "Mã Khách";  dt.Columns["TenKH"].ColumnName = "Tên Khách";  dt.Columns["GioiTinh"].ColumnName = "Giới Tính";  dt.Columns["SDT"].ColumnName = "Số Điện Thoại";  return dt;  }  }  } |

**2. Quản lý hóa đơn**

**2.1. Thêm hóa đơn (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_AddBill  @MaHD INT,  @Ngay DATE,  @TrangThai NVARCHAR(50),  @GiaTriHD DECIMAL(18, 2),  @MaKH INT,  @MaNV INT  AS  BEGIN  INSERT INTO HoaDon (MaHD, Ngay, TrangThai, GiaTriHD, MaKH, MaNV)  VALUES (@MaHD, @Ngay, @TrangThai, @GiaTriHD, @MaKH, @MaNV);  END |

***Thực thi ở C#***

|  |
| --- |
| public static bool addBill(string Ngay, string TrangThai, string GiaTriHD, string MaKH, string MaNV)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_AddBill", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@Ngay", SqlDbType.NVarChar, 50));  cmd.Parameters.Add(new SqlParameter("@TrangThai", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@GiaTriHD", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@MaKH", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.NVarChar, 10));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@Ngay"].Value = Ngay;  cmd.Parameters["@TrangThai"].Value = TrangThai;  cmd.Parameters["@GiaTriHD"].Value = GiaTriHD;  cmd.Parameters["@MaKH"].Value = MaKH;  cmd.Parameters["@MaNV"].Value = MaNV;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**2.2. Hóa đơn chưa thanh toán (VIEW)**

|  |
| --- |
| GO  CREATE VIEW V\_DanhSachHoaDonChuaThanhToan AS  SELECT HoaDon.MaHD, HoaDon.Ngay, Thue.MaSB, KhachHang.TenKH  FROM HoaDon  JOIN Thue on Thue.MaHD = HoaDon.MaHD  JOIN KhachHang on KhachHang.MaKH = HoaDon.MaKH  WHERE TrangThai = 0 |

***Thực thi ở C#***

|  |
| --- |
| public DataTable getUnPayedBill()  {  sql = "select \* from V\_DanhSachHoaDonChuaThanhToan";  cmd = new SqlCommand(sql, conn);  dt = new DataTable();  dt.Load(cmd.ExecuteReader());  dt.Columns["MaHD"].ColumnName = "Mã Hóa Đơn";  dt.Columns["Ngay"].ColumnName = "Ngày";  dt.Columns["MaSB"].ColumnName = "Sân bóng";  dt.Columns["TenKH"].ColumnName = "Khách hàng";  return dt;  } |

**2.3. Thanh toán hóa đơn (PROC)**

|  |
| --- |
| CREATE PROCEDURE proc\_PayBill(@MaHD INT,@MaSB INT, @GiaTriHD FLOAT)  AS  BEGIN  UPDATE HoaDon  SET HoaDon.TrangThai = 1, HoaDon.GiaTriHD = @GiaTriHD  WHERE MaHD = @MaHD  UPDATE SanBong  SET SanBong.TrangThai = N'Trống'  WHERE MaSB = @MaSB  UPDATE Thue  SET Thue.ThoiGianKT = FORMAT(SYSDATETIME(), 'HH:mm:ss')  WHERE Thue.MaHD = @MaHD  END |

***Thực thi ở C#***

|  |
| --- |
| public static bool PayBill(int MaHD, int MaSB, float GiaTriHD)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_PayBill", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaHD", SqlDbType.Int));  cmd.Parameters.Add(new SqlParameter("@MaSB", SqlDbType.Int));  cmd.Parameters.Add(new SqlParameter("@GiaTriHD", SqlDbType.Float));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@MaHD"].Value = MaHD;  cmd.Parameters["@MaSB"].Value = MaSB;  cmd.Parameters["@GiaTriHD"].Value = GiaTriHD;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**3. Quản lý sản phẩm**

**3.1. Thêm sản phẩm (PROC)**

|  |
| --- |
| --Thêm sản phẩm  GO  CREATE PROCEDURE proc\_AddProduct  @TenSP nvarchar(50),  @DonGia float,  @SoLuongKho int,  @MaLoaiSP nchar(10),  @MaNCC nchar(10)  AS  BEGIN  -- Thêm mới sản phẩm  INSERT INTO SanPham (MaSP, TenSP, DonGia, SoLuongKho, MaLoaiSP, MaNCC)  VALUES (@TenSP, @DonGia, @SoLuongKho, @MaLoaiSP, @MaNCC)  END |

***Thực thi ở C#***

|  |
| --- |
| public bool addProduct(string TenSP, string DonGia, string SoLuongKho, string MaLoaiSP, string MaNCC)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_AddProduct", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@TenSP", SqlDbType.NVarChar, 50));  cmd.Parameters.Add(new SqlParameter("@DonGia", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@SoLuongKho", SqlDbType.NVarChar, 3));  cmd.Parameters.Add(new SqlParameter("@MaLoaiSP", SqlDbType.NVarChar, 100));  cmd.Parameters.Add(new SqlParameter("@MaNCC", SqlDbType.NVarChar, 11));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@TenSP"].Value = TenSP;  cmd.Parameters["@DonGia"].Value = DonGia;  cmd.Parameters["@SoLuongKho"].Value = SoLuongKho;  cmd.Parameters["@MaLoaiSP"].Value = MaLoaiSP;  cmd.Parameters["@MaNCC"].Value = MaNCC;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**3.2. Sửa sản phẩm (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_UpdateProduct  @MaSP INT,  @TenSP NVARCHAR(255),  @DonGia INT,  @SoLuongKho INT,  @MaLoaiSP INT,  @MaNCC INT  AS  BEGIN  UPDATE SanPham  SET TenSP = @TenSP,  DonGia = @DonGia,  SoLuongKho = @SoLuongKho,  MaLoaiSP = @MaLoaiSP,  MaNCC = @MaNCC  WHERE MaSP = @MaSP;  END |

***Thực thi ở C#***

|  |
| --- |
| public bool updateProduct(string MaSP, string TenSP, string DonGia, string SoLuongKho, string MaLoaiSP, string MaNCC)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_UpdateProduct", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaSP", SqlDbType.Int));  cmd.Parameters.Add(new SqlParameter("@TenSP", SqlDbType.NVarChar, 50));  cmd.Parameters.Add(new SqlParameter("@DonGia", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@SoLuongKho", SqlDbType.NVarChar, 3));  cmd.Parameters.Add(new SqlParameter("@MaLoaiSP", SqlDbType.NVarChar, 100));  cmd.Parameters.Add(new SqlParameter("@MaNCC", SqlDbType.NVarChar, 11));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@MaSP"].Value = MaSP;  cmd.Parameters["@TenSP"].Value = TenSP;  cmd.Parameters["@DonGia"].Value = DonGia;  cmd.Parameters["@SoLuongKho"].Value = SoLuongKho;  cmd.Parameters["@MaLoaiSP"].Value = MaLoaiSP;  cmd.Parameters["@MaNCC"].Value = MaNCC;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**3.3. Xóa sản phẩm (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_DeleteProduct  @MaSP INT  AS  BEGIN  BEGIN TRANSACTION  BEGIN TRY  --Xoá sản phẩm theo @MaSP trong bảng SanPham  DELETE FROM dbo.SanPham WHERE SanPham.MaSP = @MaSP  COMMIT TRAN  END TRY  BEGIN CATCH  ROLLBACK  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  END CATCH  END |

***Thực thi ở C#***

|  |
| --- |
| public bool deleteProduct(string MaSP)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_deleteProduct", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaSP", SqlDbType.NVarChar));  // Thiết lập giá trị cho tham số  cmd.Parameters["@MaSP"].Value = MaSP; // Mã nhân viên bạn muốn xóa  // Bắt đầu giao dịch  SqlTransaction transaction = connection.BeginTransaction();  cmd.Transaction = transaction;  try  {  int result = cmd.ExecuteNonQuery(); // Thực hiện stored procedure để xóa nhân viên và ca làm việc  // Commit giao dịch nếu thành công  transaction.Commit();  return result > 0;  }  catch (Exception e)  {  // Rollback giao dịch nếu có lỗi  transaction.Rollback();  MessageBox.Show(e.ToString());  return false;  }  }  }  } |

**3.4. Tìm kiếm sản phẩm (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_SearchProductByName  @TenSP NVARCHAR(50)  AS  BEGIN  SELECT \* FROM SanPham  WHERE TenSP LIKE N'%' + @TenSP + '%';  END |

***Thực thi ở C#***

|  |
| --- |
| public static DataTable searchProductByName(string TenSP)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_SearchProductByName", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@TenSP", SqlDbType.NVarChar, 50));  cmd.Parameters["@TenSP"].Value = TenSP; // Tên muốn tìm  // Thực hiện stored procedure  DataTable dt = new DataTable();  dt.Load(cmd.ExecuteReader());  return dt;  }  }    } |

**4. Quản lý sân bóng**

**4.1. Thêm sân bóng (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_AddPitch  @KichThuoc VARCHAR(10),  @TrangThai NVARCHAR(20)  AS  BEGIN  INSERT INTO SanBong(KichThuoc, TrangThai)  VALUES (@KichThuoc, @TrangThai);  END |

***Thực thi ở C#***

|  |
| --- |
| public bool addPitch(string KichThuoc, string TrangThai)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_AddPitch", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@KichThuoc", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@TrangThai", SqlDbType.NVarChar, 20));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@KichThuoc"].Value = KichThuoc;  cmd.Parameters["@TrangThai"].Value = TrangThai;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**4.2. Xóa sân bóng (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_DeletePitch  @MaSB INT  AS  BEGIN  DELETE FROM SanBong  WHERE MaSB = @MaSB;  END |

***Thực thi ở C#***

|  |
| --- |
| public bool deletePitch(string MaSB)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_DeletePitch", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaSB", SqlDbType.NVarChar));  // Thiết lập giá trị cho tham số  cmd.Parameters["@MaSB"].Value = MaSB; // Mã nhân viên bạn muốn xóa  // Bắt đầu giao dịch  SqlTransaction transaction = connection.BeginTransaction();  cmd.Transaction = transaction;  try  {  int result = cmd.ExecuteNonQuery(); // Thực hiện stored procedure để xóa nhân viên và ca làm việc  // Commit giao dịch nếu thành công  transaction.Commit();  return result > 0;  }  catch (Exception e)  {  // Rollback giao dịch nếu có lỗi  transaction.Rollback();  MessageBox.Show(e.ToString());  return false;  }  }  }  } |

**4.2. Sửa sân bóng (PROC)**

|  |
| --- |
| -- Sửa sân  GO  CREATE PROCEDURE proc\_UpdatePitch  @MaSB INT,  @KichThuoc VARCHAR(10),  @TrangThai NVARCHAR(20)  AS  BEGIN  UPDATE SanBong  SET  KichThuoc = @KichThuoc, TrangThai = @TrangThai  WHERE MaSB = @MaSB;  END |

***Thực thi ở C#***

|  |
| --- |
| public bool updatePitch(string MaSB, string KichThuoc, string TrangThai)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_UpdatePitch", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaSB", SqlDbType.Int));  cmd.Parameters.Add(new SqlParameter("@KichThuoc", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@TrangThai", SqlDbType.NVarChar, 20));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@MaSB"].Value = MaSB;  cmd.Parameters["@KichThuoc"].Value = KichThuoc;  cmd.Parameters["@TrangThai"].Value = TrangThai;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**5. Thống kê doanh thu và chi phí**

**5.1. Doanh thu theo ngày, tháng, năm (FUNC)**

|  |
| --- |
| GO  CREATE FUNCTION getPayedBill(@NgayBD DATE, @NgayKT DATE)  RETURNS @BillPayed TABLE(Ngay DATE,GiaTriHD FLOAT,TenKH NVARCHAR(50),SDTKH NVARCHAR(11),MaNV INT ,TenNV NVARCHAR(50)  )  AS  BEGIN  INSERT INTO @BillPayed  SELECT HoaDon.Ngay, HoaDon.GiaTriHD, KhachHang.TenKH, KhachHang.SDT, NhanVien.MaNV, NhanVien.TenNV  FROM HoaDon  JOIN KhachHang on HoaDon.MaKH = KhachHang.MaKH  JOIN NhanVien on HoaDon.MaNV = NhanVien.MaNV  WHERE HoaDon.TrangThai = 1 AND (HoaDon.Ngay BETWEEN @NgayBD AND @NgayKT)  RETURN  END |

***Thực thi ở C#***

|  |
| --- |
| public static DataTable GetPayedBill(DateTime NgayBD, DateTime NgayKT)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("SELECT \* FROM dbo.getPayedBill(@NgayBD, @NgayKT)", connection))  {  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@NgayBD", SqlDbType.Date)).Value = NgayBD;  cmd.Parameters.Add(new SqlParameter("@NgayKT", SqlDbType.Date)).Value = NgayKT;  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  // Đổ dữ liệu vào DataTable  dt.Load(cmd.ExecuteReader());  dt.Columns["Ngay"].ColumnName = "Ngày thanh toán";  dt.Columns["GiaTriHD"].ColumnName = "Giá trị";  dt.Columns["TenKH"].ColumnName = "Khách hàng";  dt.Columns["SDTKH"].ColumnName = "Số điện thoại";  dt.Columns["MaNV"].ColumnName = "Mã nhân viên";  dt.Columns["TenNV"].ColumnName = "Nhân viên";  return dt;  }  }  } |

**6. Quản lý nhân viên**

**6.1. Thêm nhân viên (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_AddStaff  @MaNV INT,  @TenNV NCHAR(50),  @NgaySinh DATE,  @GioiTinh NCHAR(3),  @DiaChi NVARCHAR(100),  @SDT NCHAR(11),  @Thuong INT,  @MaCV NCHAR(10),  @SoCa INT  AS  BEGIN  INSERT INTO NhanVien (MaNV, TenNV, NgaySinh, GioiTinh, DiaChi, SDT, Thuong, MaCV, SoCa)  VALUES (@MaNV, @TenNV, @NgaySinh, @GioiTinh, @DiaChi, @SDT, @Thuong, @MaCV, @SoCa);  END |

***Thực thi ở C#***

|  |
| --- |
| public bool addStaff(string TenNV, string NgaySinh, String GioiTinh, String DiaChi, String SDT, String MaCV)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_AddStaff", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@TenNV", SqlDbType.NVarChar, 50));  cmd.Parameters.Add(new SqlParameter("@NgaySinh", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@GioiTinh", SqlDbType.NVarChar, 3));  cmd.Parameters.Add(new SqlParameter("@DiaChi", SqlDbType.NVarChar, 100));  cmd.Parameters.Add(new SqlParameter("@SDT", SqlDbType.NChar, 11));  cmd.Parameters.Add(new SqlParameter("@MaCV", SqlDbType.NVarChar, 10));  cmd.Parameters.Add(new SqlParameter("@SoCa", SqlDbType.Int, 11));  cmd.Parameters.Add(new SqlParameter("@Thuong", SqlDbType.Float, 10));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@TenNV"].Value = TenNV;  cmd.Parameters["@NgaySinh"].Value = NgaySinh;  cmd.Parameters["@GioiTinh"].Value = GioiTinh;  cmd.Parameters["@DiaChi"].Value = DiaChi;  cmd.Parameters["@SDT"].Value = SDT;  cmd.Parameters["@MaCV"].Value = MaCV;  cmd.Parameters["@SoCa"].Value = 0;  cmd.Parameters["@Thuong"].Value = 0;  // Thực hiện stored procedure  try  {  int result = cmd.ExecuteNonQuery();  return result > 0;  }  catch(SqlException ex)  {  MessageBox.Show(ex.Message);  return false;  }    }    }  } |

**6.2. Xóa nhân viên (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_deleteStaffAndShifts  @MaNV INT  AS  BEGIN  BEGIN TRANSACTION;  DECLARE @SQLstring VARCHAR(200), @UserName VARCHAR(12)  -- Xóa các ca làm việc của nhân viên từ bảng BangPhanCa  DELETE FROM BangPhanCa  WHERE MaNV = @MaNV;  -- Xóa nhân viên từ bảng NhanVien  DELETE FROM NhanVien  WHERE MaNV = @MaNV;  DELETE FROM TAIKHOAN  WHERE MaNV = @MaNV;  SELECT @UserName = UserName FROM TAIKHOAN  SET @SQLstring = 'DROP LOGIN ' + @UserName  EXEC(@SQLstring)  SET @SQLstring = 'DROP USER ' + @UserName  EXEC(@SQLstring)  COMMIT;  END |

***Thực thi ở C#***

|  |
| --- |
| public bool deleteStaff(string MaNV)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_deleteStaffAndShifts", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.NChar));  // Thiết lập giá trị cho tham số  cmd.Parameters["@MaNV"].Value = MaNV; // Mã nhân viên bạn muốn xóa  // Bắt đầu giao dịch  SqlTransaction transaction = connection.BeginTransaction();  cmd.Transaction = transaction;  try  {  int result = cmd.ExecuteNonQuery(); // Thực hiện stored procedure để xóa nhân viên và ca làm việc  // Commit giao dịch nếu thành công  transaction.Commit();  return result > 0;  }  catch (Exception e)  {  // Rollback giao dịch nếu có lỗi  transaction.Rollback();  MessageBox.Show(e.ToString());  return false;  }  }  }  } |

**6.3. Tìm kiếm nhân viên (FUNC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_SearchStaffByName  @TenNV NVARCHAR(50)  AS  BEGIN  SELECT \* FROM NhanVien  WHERE TenNV LIKE N'%' + @TenNV + '%';  END |

***Thực thi ở C#***

|  |
| --- |
| public List<Staff> searchStaffByName(string TenNV)  {  Database db = new Database();  List<Staff> staffList = new List<Staff>();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_SearchStaffByName", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@TenNV", SqlDbType.NChar, 50));  cmd.Parameters["@TenNV"].Value = TenNV; // Tên muốn tìm  // Thực hiện stored procedure  DataTable dt = new DataTable();  dt.Load(cmd.ExecuteReader());  foreach (DataRow item in dt.Rows)  {  Staff staff = new Staff(item);  staffList.Add(staff);  }  }  }  return staffList;  } |

**6.4. Tính lương nhân viên (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_CalculateSalary  @MaNV INT  AS  BEGIN  DECLARE @LuongCoBan INT;  DECLARE @SoGioLam INT;  DECLARE @LuongTheoGio INT;  DECLARE @SoCaThuong INT;  DECLARE @LuongThuong INT;  DECLARE @TongLuong INT;  -- Lấy lương cơ bản từ bảng CongViec  SELECT @LuongCoBan = Luong  FROM CongViec  WHERE MaCV = (SELECT MaCV FROM NhanVien WHERE MaNV = @MaNV);  -- Lấy số giờ làm việc từ bảng NhanVien  SELECT @SoGioLam = SoCa\*4 FROM NhanVien WHERE MaNV = @MaNV;  -- Tính lương theo giờ  SET @LuongTheoGio = @SoGioLam \* @LuongCoBan;  -- Lấy số ca thưởng từ bảng NhanVien  SELECT @SoCaThuong = Thuong\*4 FROM NhanVien WHERE MaNV = @MaNV;  -- Tính lương thưởng  SET @LuongThuong = @SoCaThuong \* 4 \* 2 \* @LuongCoBan;  -- Tính tổng lương  SET @TongLuong = @LuongTheoGio + @LuongThuong;  -- Trả về tổng lương  SELECT @TongLuong AS TongLuong;  END |

***Thực thi ở C#***

|  |
| --- |
| public static float getSalary(int MaNV)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("dbo.CalculateSalary", connection))  {  cmd.CommandType = CommandType.StoredProcedure;    // Thêm tham số cho function  cmd.Parameters.AddWithValue("@MaNV", MaNV); // Thay đổi giá trị cho phù hợp  // Thêm tham số đối với giá trị trả về  SqlParameter returnValue = new SqlParameter("@ReturnValue", SqlDbType.Float);  returnValue.Direction = ParameterDirection.ReturnValue;  cmd.Parameters.Add(returnValue);  // Thực hiện command  cmd.ExecuteNonQuery();  // Lấy giá trị trả về từ tham số  float result = Convert.ToSingle(cmd.Parameters["@ReturnValue"].Value);  return Math.Abs(result);  }  }  } |

**6.5. Sửa thông tin nhân viên (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_UpdateStaff  @MaNV INT,  @TenNV NCHAR(50),  @NgaySinh NCHAR(10),  @GioiTinh NCHAR(3),  @DiaChi NVARCHAR(100),  @SDT NCHAR(11),  @MaCV NCHAR(10)  AS  BEGIN  UPDATE NhanVien  SET  TenNV = @TenNV,  NgaySinh = @NgaySinh,  GioiTinh = @GioiTinh,  DiaChi = @DiaChi,  SDT = @SDT,  MaCV = @MaCV  WHERE MaNV = @MaNV;  END |

***Thực thi ở C#***

|  |
| --- |
| public bool updateStaff(string MaNV, string TenNV, string NgaySinh, String GioiTinh, String DiaChi, String SDT, String MaCV)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_UpdateStaff", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.Int));  cmd.Parameters.Add(new SqlParameter("@TenNV", SqlDbType.NChar, 50));  cmd.Parameters.Add(new SqlParameter("@NgaySinh", SqlDbType.NChar, 10));  cmd.Parameters.Add(new SqlParameter("@GioiTinh", SqlDbType.NChar, 3));  cmd.Parameters.Add(new SqlParameter("@DiaChi", SqlDbType.NVarChar, 100));  cmd.Parameters.Add(new SqlParameter("@SDT", SqlDbType.NChar, 11));  cmd.Parameters.Add(new SqlParameter("@MaCV", SqlDbType.NChar, 10));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@MaNV"].Value = MaNV;  cmd.Parameters["@TenNV"].Value = TenNV;  cmd.Parameters["@NgaySinh"].Value = NgaySinh;  cmd.Parameters["@GioiTinh"].Value = GioiTinh;  cmd.Parameters["@DiaChi"].Value = DiaChi;  cmd.Parameters["@SDT"].Value = SDT;  cmd.Parameters["@MaCV"].Value = MaCV;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } }  } |

**6.6. Sửa ca làm việc nhân viên (PROC)**

|  |
| --- |
| GO  CREATE PROCEDURE proc\_UpdateShiftAssignment  @MaCa INT,  @MaNV INT,  @NgayLam DATE  AS  BEGIN  UPDATE BangPhanCa  SET NgayLam = @NgayLam  WHERE MaCa = @MaCa AND MaNV = @MaNV;  END |

***Thực thi ở C#***

|  |
| --- |
| public static bool updateShiftDivision(string MaCa, string NgayLam, string MaNV, string MaNVold)  {  Database db = new Database();  using (SqlConnection connection = db.getConnection)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_UpdateShiftDivision", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@MaCa", SqlDbType.Int));  cmd.Parameters.Add(new SqlParameter("@NgayLam", SqlDbType.NVarChar, 20));  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.Int));  cmd.Parameters.Add(new SqlParameter("@MaNVold", SqlDbType.Int));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@MaCa"].Value = MaCa;  cmd.Parameters["@NgayLam"].Value = NgayLam;  cmd.Parameters["@MaNV"].Value = MaNV;  cmd.Parameters["@MaNVold"].Value = MaNVold;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  } |

**CHƯƠNG IV: TẠO USER VÀ PHÂN QUYỀN**

Ứng với 2 nhóm người dùng (Nhân viên và Quản lý) là các role riêng

* **Đối với Role Staff (Dành cho Nhân viên):**

USE QLSANBONG

CREATE ROLE Staff

--Gán các quyền trên table cho role Staff

GRANT EXECUTE TO Staff

GRANT SELECT TO Staff

GRANT SELECT, REFERENCES ON BangPhanCa TO Staff

GRANT SELECT, REFERENCES ON CaLamViec TO Staff

GRANT SELECT, REFERENCES ON ChiTietHoaDon TO Staff

GRANT SELECT, REFERENCES ON CongViec TO Staff

GRANT SELECT, REFERENCES ON GiaThue TO Staff

GRANT SELECT, REFERENCES ON HoaDon TO Staff

GRANT SELECT, REFERENCES ON KhachHang TO Staff

GRANT SELECT, REFERENCES ON LoaiSanPham TO Staff

GRANT SELECT, REFERENCES ON NhaCungCap TO Staff

GRANT SELECT, REFERENCES ON NhanVien TO Staff

GRANT SELECT, REFERENCES ON SanBong TO Staff

GRANT SELECT, REFERENCES ON SanPham TO Staff

GRANT SELECT, REFERENCES ON Thue TO Staff

* GRANT EXECUTE ON proc\_AddCustomer to Staff;

DENY EXECUTE ON proc\_AddAccount to Staff;

DENY EXECUTE ON proc\_AddPitch to Staff;

DENY EXECUTE ON proc\_AddProduct to Staff;

DENY EXECUTE ON proc\_AddStaff to Staff;

DENY EXECUTE ON proc\_DeleteAccount to Staff;

DENY EXECUTE ON proc\_DeletePitch to Staff;

DENY EXECUTE ON proc\_DeleteProduct to Staff;

DENY EXECUTE ON proc\_deleteStaffAndShifts to Staff;

DENY EXECUTE ON proc\_UpdatePitch to Staff;

DENY EXECUTE ON proc\_UpdateProduct to Staff;

DENY EXECUTE ON proc\_UpdateStaff to Staff;

* **Đối với Role Manager:** Ta sử dụng server role sysadmin.

***- Trigger tạo tài khoản:***

|  |
| --- |
| GO  CREATE TRIGGER TG\_CreateUser  ON TAIKHOAN  AFTER INSERT  AS  BEGIN  DECLARE @UserName varchar(12), @UserPassWord varchar(10), @MaNV INT  SELECT @UserName = inserted.UserName, @UserPassWord = inserted.UserPassWord, @MaNV = inserted.MaNV  FROM inserted  DECLARE @sqlString nvarchar(2000), @MaCV INT  ----  SET @sqlString= 'CREATE LOGIN [' + @UserName +'] WITH PASSWORD='''+ @UserPassWord  +''', DEFAULT\_DATABASE=[QLSANBONG], CHECK\_EXPIRATION=OFF,  CHECK\_POLICY=OFF'  EXEC (@sqlString)  ----  SET @sqlString= 'CREATE USER ' + @UserName +' FOR LOGIN '+ @UserName  EXEC (@sqlString)  SELECT @MaCV = MaCV  FROM NhanVien  WHERE MaNV = @MaNV  IF (@MaCV = 1)  SET @sqlString = 'ALTER SERVER ROLE sysadmin' + ' ADD MEMBER '  + @UserName;  ELSE  SET @sqlString = 'ALTER ROLE Staff ADD MEMBER ' + @UserName;  EXEC (@sqlString)  END |

**TẠO TÀI KHOẢN:**

***Code C#:***

|  |
| --- |
| public static bool addAccount(string username, string userpassword, string MaNV)  {  Database db = new Database();  using (SqlConnection connection = db.getConnectionAdmin)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_AddAccount", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@UserName", SqlDbType.VarChar, 12));  cmd.Parameters.Add(new SqlParameter("@UserPassWord", SqlDbType.VarChar, 10));  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.Int));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@UserName"].Value = username;  cmd.Parameters["@UserPassWord"].Value = userpassword;  cmd.Parameters["@MaNV"].Value = MaNV;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  }  private void button19\_Click(object sender, EventArgs e)  {  string username = textBox22.Text;  string userpassword = textBox23.Text;  string MaNV = textBox24.Text;  try  {  if (AccountDAO.addAccount(username, userpassword, MaNV)) ;  {  MessageBox.Show("Thêm tài khoản cho nhân viên "+ MaNV + " thành công!");  LoadAccountsList();  }  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

***Code SQL:***

|  |
| --- |
| GO  CREATE PROCEDURE proc\_AddAccount(@UserName VARCHAR(12), @UserPassWord VARCHAR(10), @MaNV INT)  AS  BEGIN  IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV)  BEGIN  RAISERROR(N'Nhân viên không tồn tại', 16, 1)  RETURN  END  ELSE  IF EXISTS (SELECT 1 FROM TAIKHOAN WHERE UserName = @UserName)  BEGIN  RAISERROR(N'Tài khoản bị trùng', 16, 1)  RETURN  END  ELSE  BEGIN  INSERT INTO TAIKHOAN(UserName, UserPassWord, MaNV)  VALUES(@UserName, @UserPassWord, @MaNV)  END  END |

* ***Trigger xóa tài khoản:***

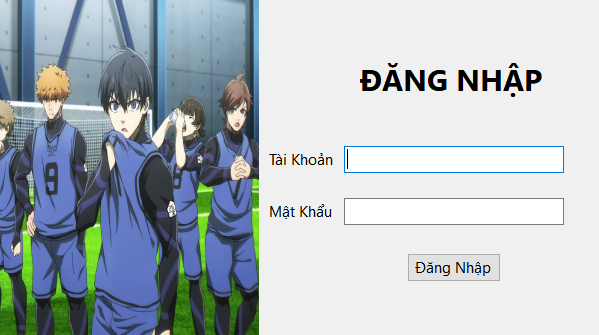
|  |
| --- |
| GO  CREATE TRIGGER TG\_DeleteUser  ON TAIKHOAN  AFTER DELETE AS  BEGIN  DECLARE @UserName VARCHAR(12), @SQLstring VARCHAR(50)  SELECT @UserName = del.UserName FROM deleted del  SET @SQLstring = 'DROP LOGIN ' + @UserName;  EXEC(@SQLstring)  SET @SQLstring = 'DROP USER ' + @UserName;  EXEC(@SQLstring)  END |

***Thực thi ở C#:***

|  |
| --- |
| public static bool deleteAccount(string username, string userpassword, string MaNV)  {  Database db = new Database();  using (SqlConnection connection = db.getConnectionAdmin)  {  connection.Open();  using (SqlCommand cmd = new SqlCommand("proc\_DeleteAccount", connection))  {  cmd.CommandType = CommandType.StoredProcedure;  // Thêm tham số cho stored procedure  cmd.Parameters.Add(new SqlParameter("@UserName", SqlDbType.VarChar, 12));  cmd.Parameters.Add(new SqlParameter("@UserPassWord", SqlDbType.VarChar, 10));  cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.Int));  // Thiết lập giá trị cho các tham số  cmd.Parameters["@UserName"].Value = username;  cmd.Parameters["@UserPassWord"].Value = userpassword;  cmd.Parameters["@MaNV"].Value = MaNV;  // Thực hiện stored procedure  int result = cmd.ExecuteNonQuery();  return result > 0;  }  }  }  private void button21\_Click(object sender, EventArgs e)  {  string username = textBox27.Text;  string userpassword = textBox26.Text;  string MaNV = textBox25.Text;  try  {  if (AccountDAO.deleteAccount(username, userpassword, MaNV)) ;  {  MessageBox.Show("Xóa tài khoản cho nhân viên " + MaNV + " thành công!");  LoadAccountsList();  }  }  catch (SqlException ex)  {  MessageBox.Show(ex.Message);  }  } |

**CHƯƠNG V: THIẾT KẾ GIAO DIỆN HỆ THỐNG**

## 1. Form đăng nhập

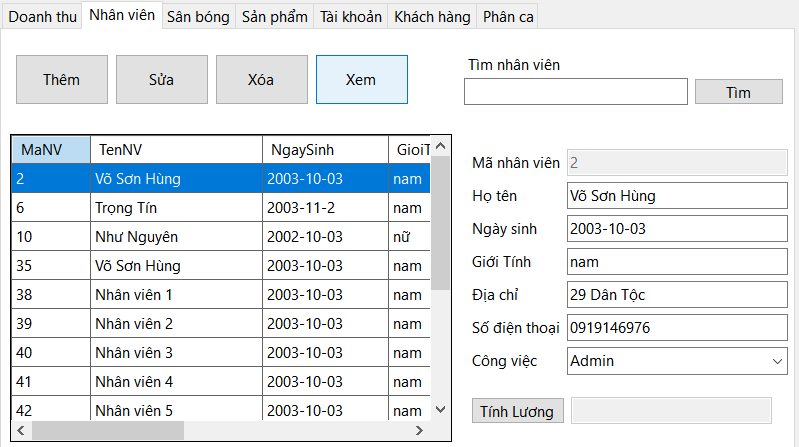


## 2. Form dành cho admin

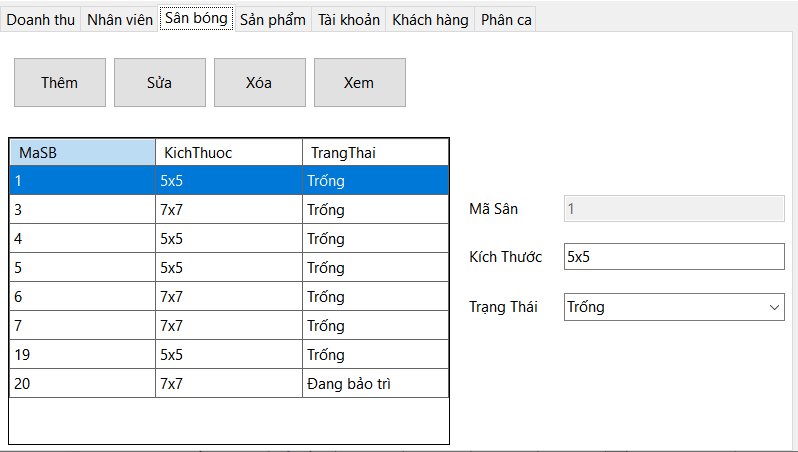
### **Thống kê doanh thu**



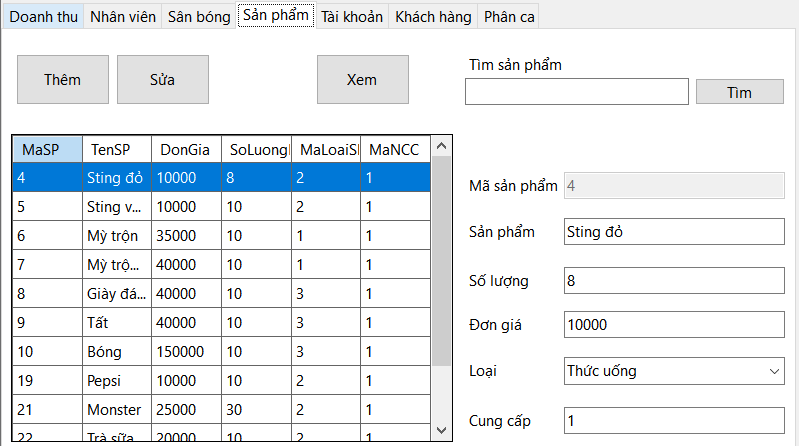
### **Xem, thêm, xóa, sửa, tìm, tính lương cho nhân viên**



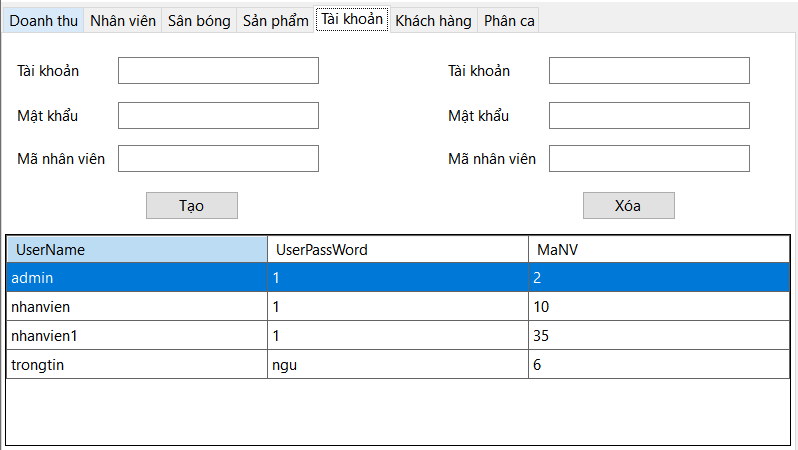
### **Xem, thêm, xóa, sửa cho sân bóng**



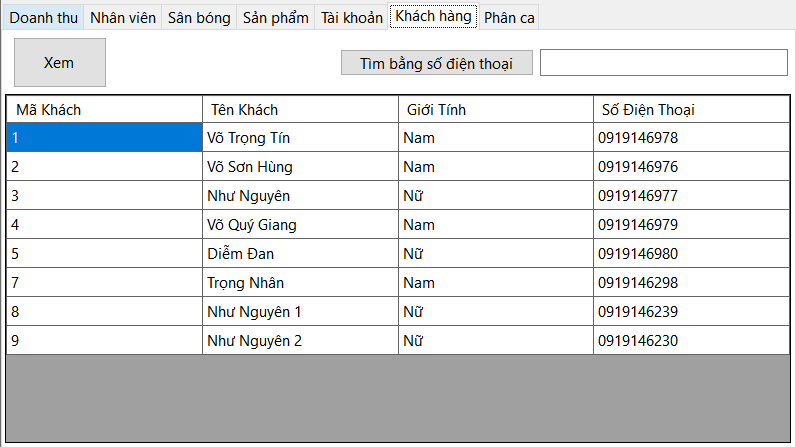
### **Xem, thêm, sửa, tìm sản phẩm**



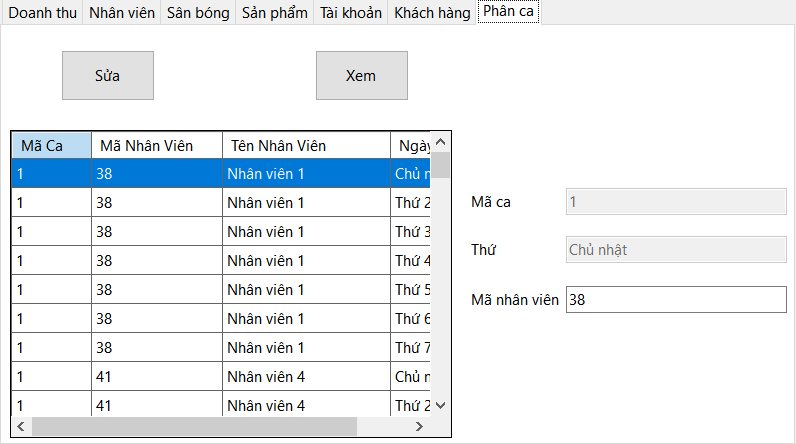
### **Xem, thêm, xóa tài khoản cho nhân viên**



### **Xem, tìm khách hàng**

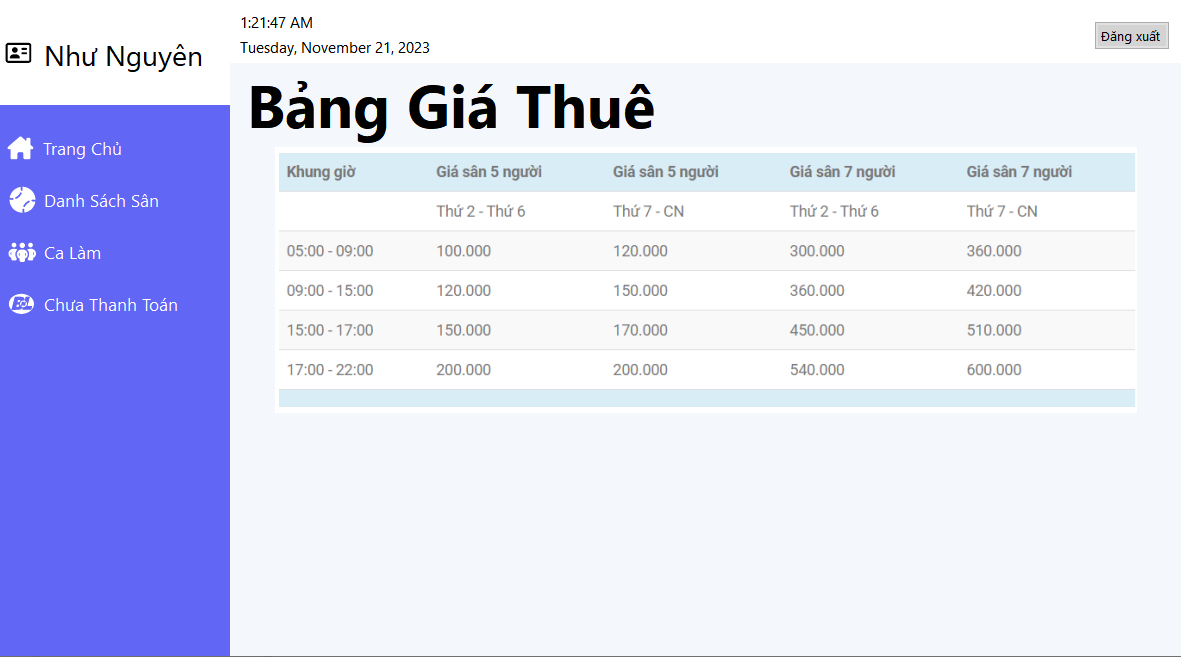


### **Xem, sửa ca làm của nhân viên**

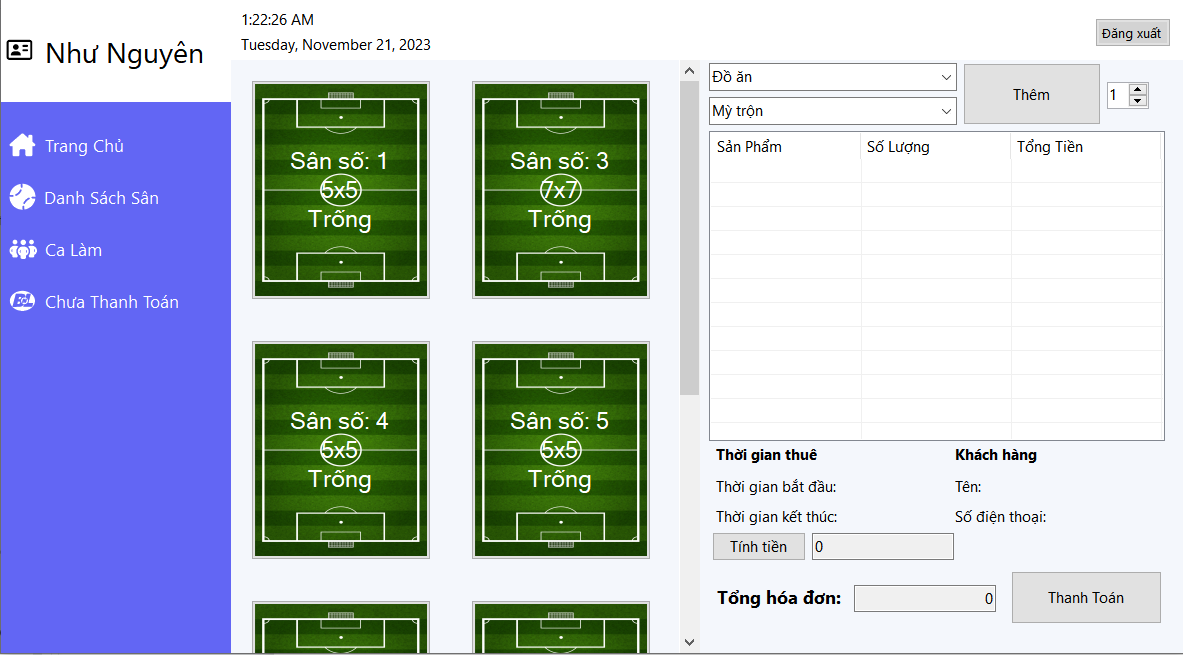


## 3. Form dành cho nhân viên

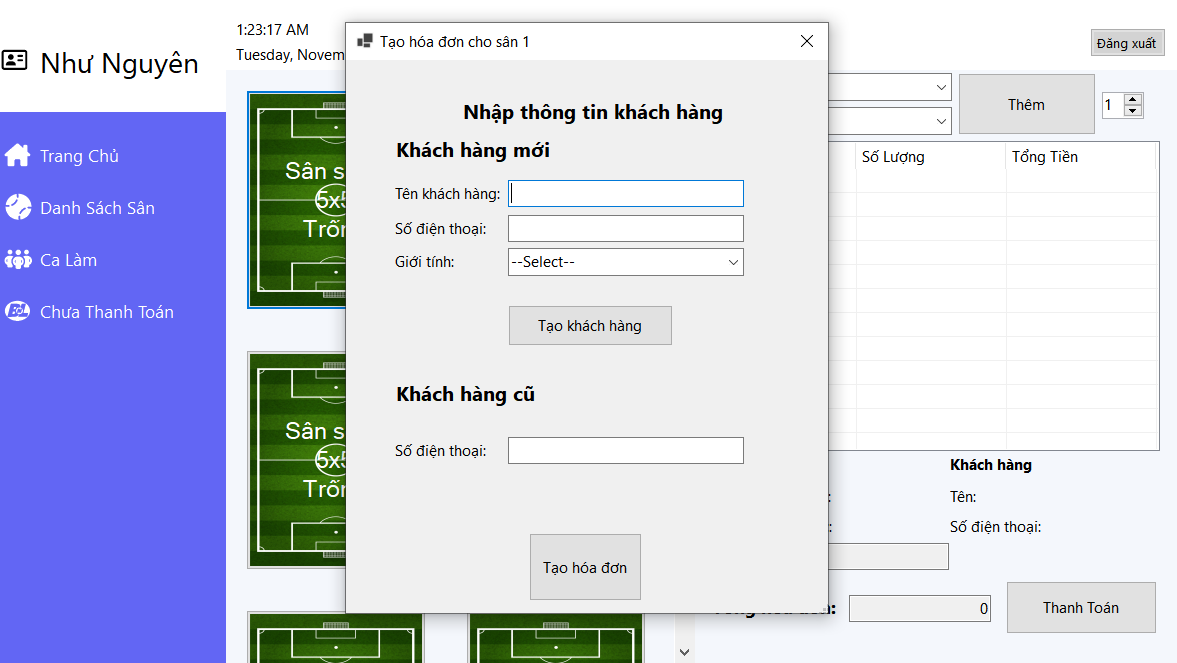
### **Trang chủ**



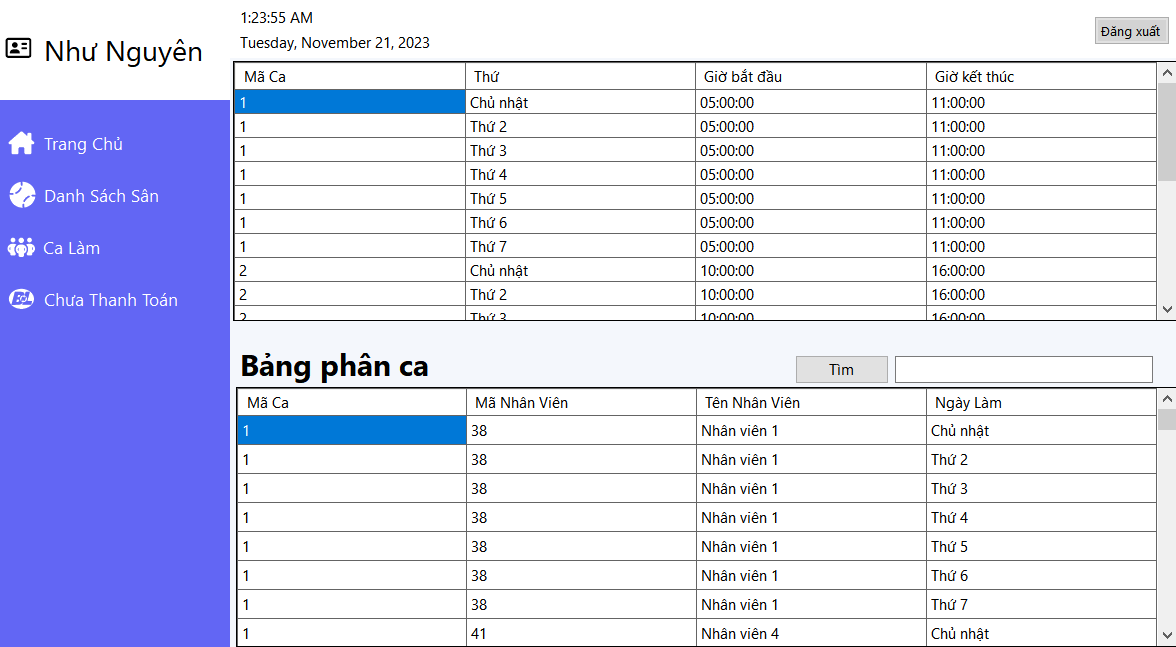
### **Danh sách sân (nơi đặt sân)**



### **Tạo hóa đơn hoặc tạo mới khách hàng nếu là khách hàng mới**



### **Xem, tìm ca theo tên nhân viên**



### **Xem những hóa đơn chưa thanh toán**

