ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*



**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**BÁO CÁO ĐỒ ÁN THỰC HÀNH**

**HỆ THỐNG ĐẶT VÀ GIAO THỨC ĂN ONLINE**

**Người thực hiện: Nhóm 2**

**Hướng dẫn thực hành:**

**ThS. Tiết Gia Hồng**

**ThS. Hồ Thị Hoàng Vy**

**ThS.Lương Hán Cơ**

*Thành phố Hồ Chí Minh, tháng 12/2022*

# Mục lục

[Mục lục 2](#_Toc121690146)

[Mục lục hình ảnh 4](#_Toc121690147)

[I. THÔNG TIN CHUNG 5](#_Toc121690148)

[1.1. Thông tin đồ án 5](#_Toc121690149)

[1.2. Thông tin nhóm và đánh giá 5](#_Toc121690150)

[1.3. Phân công công việc 5](#_Toc121690151)

[II. THIẾT KẾ CƠ SỞ DỮ LIỆU 6](#_Toc121690152)

[2.1. Lược đồ thực thể kết hợp 6](#_Toc121690153)

[2.2. Ràng buộc dữ liệu bổ sung 6](#_Toc121690154)

[2.3. Mô hình cơ sở dữ liệu quan hệ 9](#_Toc121690155)

[2.4. Mô tả bảng – thuộc tính 10](#_Toc121690156)

[2.5. Các chức năng của hệ thống 17](#_Toc121690157)

[2.5.1. Chức năng cho DÙNG CHUNG 17](#_Toc121690158)

[2.5.2. Phân hệ KHÁCH HÀNG 17](#_Toc121690159)

[2.5.3. Phân hệ ĐỐI TÁC 18](#_Toc121690160)

[2.5.4. Phân hệ TÀI XẾ 19](#_Toc121690161)

[2.5.4. Phân hệ NHÂN VIÊN 19](#_Toc121690162)

[2.5.5. Phân hệ QUẢN TRỊ 19](#_Toc121690163)

[III. PHÂN QUYỀN NGƯỜI DÙNG 20](#_Toc121690164)

[IV. THIẾT KẾ GIAO DIỆN HỆ THỐNG 20](#_Toc121690165)

[4.1. Dùng chung 20](#_Toc121690166)

[4.1.1. Chức năng đăng nhập 20](#_Toc121690167)

[4.1.2. Chức năng đăng kí 21](#_Toc121690168)

[4.2. Phân hệ KHÁCH HÀNG 21](#_Toc121690169)

[4.2.1. Chức năng thông tin cá nhân 21](#_Toc121690170)

[4.2.2. Chức năng xem đơn hàng đã đặt 22](#_Toc121690171)

[4.2.3. Chức năng đặt hàng 23](#_Toc121690172)

[4.3. Phân hệ ĐỐI TÁC 26](#_Toc121690173)

[4.3.1. Chức năng thông tin 26](#_Toc121690174)

[4.3.2. Chức năng chi nhánh 27](#_Toc121690175)

[4.3.3. Chức năng Món Ăn 29](#_Toc121690176)

[4.3.4. Chức năng lập hợp đồng 30](#_Toc121690177)

[4.3.5. Chức năng xem hợp đồng đã lập 31](#_Toc121690178)

[4.3.6. Chức năng đơn hàng 32](#_Toc121690179)

[4.4. Phân hệ TÀI XẾ 32](#_Toc121690180)

[4.4.1. Chức năng tài khoản 32](#_Toc121690181)

[4.4.2. Chức năng Danh sách đơn hàng 33](#_Toc121690182)

[4.4.3. Chức năng đơn hàng đã nhận 34](#_Toc121690183)

[4.4.4. Chức năng Thống kê 35](#_Toc121690184)

[4.5. Phân hệ NHÂN VIÊN 36](#_Toc121690185)

[4.5.1. Chức năng tài khoản 36](#_Toc121690186)

[4.5.2. Chức năng xem hợp đồng đã duyệt 37](#_Toc121690187)

[4.5.3. Chức năng hợp đồng chưa duyệt 38](#_Toc121690188)

[4.6. Phân hệ QUẢN TRỊ 39](#_Toc121690189)

[4.6.1 Chức năng xem tài khoản 39](#_Toc121690190)

[V. TÌNH HUỐNG TRANH CHẤP 40](#_Toc121690191)

[VI. TÀI LIỆU THAM KHẢO 84](#_Toc121690192)

# Mục lục hình ảnh

[Hình 1: Lược đồ thực thể kết hợp 5](#_Toc121689995)

[Hình 2: Mô hình quan hệ 9](#_Toc121689996)

[Hình 3: Giao diện đăng nhập 19](#_Toc121689997)

[Hình 4: Giao diện đăng ký 20](#_Toc121689998)

[Hình 5: Giao diện khách hàng 21](#_Toc121689999)

[Hình 6: Giao diện lịch sử đặt hàng 22](#_Toc121690000)

[Hình 7: Giao diện đặt hàng 23](#_Toc121690001)

[Hình 8: Giao diện danh sách món ăn 24](#_Toc121690002)

[Hình 9: Giao diện điền thông tin đặt hàng 25](#_Toc121690003)

[Hình 10: Giao diện hiển thị thông tin đối tác 26](#_Toc121690004)

[Hình 11: Giao diện hiển thị thông tin chi nhánh 27](#_Toc121690005)

[Hình 12: Giao diện chức năng mở thêm chi nhánh mới 28](#_Toc121690006)

[Hình 13: Giao diện hiển thị danh sách món ăn của từng đối tác 29](#_Toc121690007)

[Hình 14: Giao diện kí hợp đồng mới 30](#_Toc121690008)

[Hình 15: Giao diện xem hợp đồng đã đăng kí 30](#_Toc121690009)

[Hình 16: Giao diện hiển thị danh sách đơn hàng của đối tác 31](#_Toc121690010)

[Hình 17: Giao diện thêm, chỉnh sửa thông tin của tài xế 32](#_Toc121690011)

[Hình 18: Giao diện hiển thị các đơn hàng phù hợp với tài xế 33](#_Toc121690012)

[Hình 19: Giao diện xem lịch sử đơn hàng đã nhận của tài xế 34](#_Toc121690013)

[Hình 20: Giao diện thống kê các đơn giao hàng của tài xế 35](#_Toc121690014)

[Hình 21: Giao diện hiển thị thông tin nhân viên 36](#_Toc121690015)

[Hình 22: Giao diện hiển thị hợp đồng đã duyệt 37](file:///C:\Users\nvqhu\Downloads\N02_report%20(1).docx#_Toc121690016)

[Hình 23: Giao diện hiển thị danh sách hợp đồng chưa duyệt 37](#_Toc121690017)

[Hình 24: Giao diện quản lí tài khoản của quản trị viên 38](#_Toc121690018)

[Hình 25: Giao diện chi tiết từng tài khoản 39](#_Toc121690019)

# I. THÔNG TIN CHUNG

## 1.1. Thông tin đồ án

**HỆ THỐNG ĐẶT VÀ GIAO THỨC ĂN ONLINE**

§ Hệ thống giao nhanh ABC có đội ngũ tài xế đông đảo, lượng khách hàng lớn, trở thành đối tác của hệ thống sẽ giúp cửa hàng của khách hàng tăng trưởng tốt hơn.

§ **Gồm 5 phân hệ**

🌣Phân hệ Đối tác

🌣Phân hệ Khách hàng

🌣Phân hệ Tài xế

🌣Phân hệ Nhân viên

🌣Phân hệ Quản trị

§ **Yêu cầu đồ án**

o Phân tích nghiệp vụ và thiết kế CSDL phù hợp.

o Xác định và xây dựng các chức năng phù hợp bằng Win/ Web.

o Xác định vấn đề tranh chấp dữ liệu trong hệ thống và hướng giải quyết.

o Phân quyền người dùng trong hệ thống.

o Viết báo cáo.

## 1.2. Thông tin nhóm và đánh giá

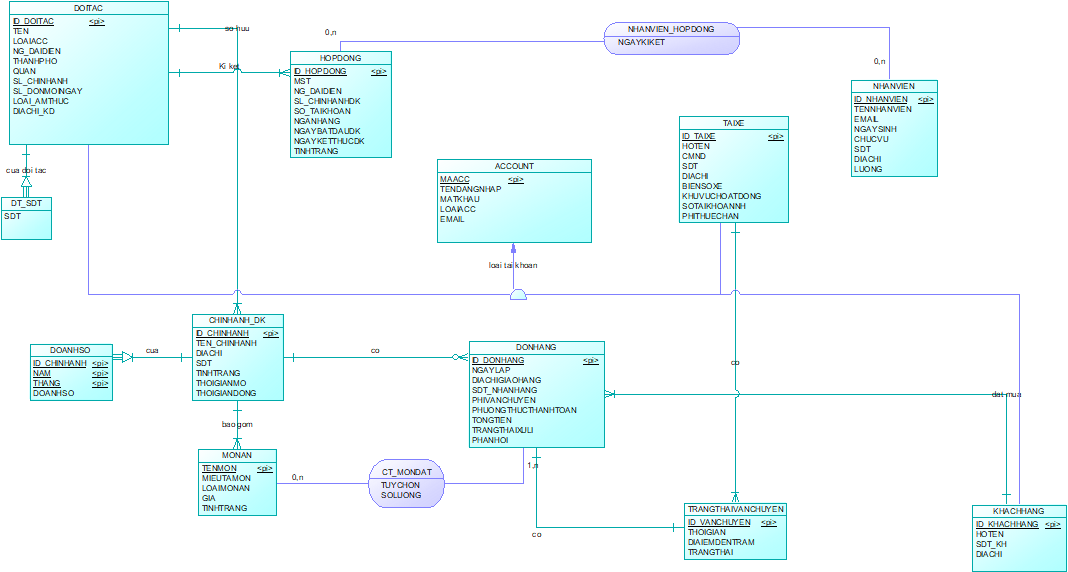
|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Đánh giá tham gia** | **Đánh giá hoàn tất công việc được giao** |
| 20120255 | Phạm Mai Thiên Bảo | 100% | 100% |
| 20120295 | Ngô Võ Quang Huy | 100% | 100% |
| 20120305 | Võ Thị Kiều Khanh | 100% | 100% |
| 20120318 | Nguyễn Lê Mỹ Linh | 100% | 100% |

## 1.3. Phân công công việc

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Nội dung công việc** | **Tiến độ** |
| 20120255 | Thiết kế cơ sở dữ liệu, xây dựng và giải quyết tình huống tranh chấp, cài đặt các ràng buộc toàn vẹn, cài đặt một số chức năng cần thiết. | 100% |
| 20120295 | Thiết kế cơ sở dữ liệu, xây dựng và giải quyết tình huống tranh chấp, tổng hợp báo cáo, cài đặt một số chức năng cần thiết. | 100% |
| 20120305 | Thiết kế giao diện, cài đặt các ràng buộc toàn vẹn, xây dựng và giải quyết tình huống tranh chấp, cài đặt phân quyền người dung. | 100% |
| 20120318 | Mô tả dữ liệu, quy trình dữ liệu, xây dựng và giải quyết tình huống tranh chấp, tổng hợp báo cáo, cài đặt một số chức năng cần thiết. | 100% |

# II. THIẾT KẾ CƠ SỞ DỮ LIỆU

## 2.1. Lược đồ thực thể kết hợp



Hình : Lược đồ thực thể kết hợp

## 2.2. Ràng buộc dữ liệu bổ sung

**ª BẢNG ACCOUNT**

§ Các thuộc tính: TENDANGNHAP, MATKHAU, LOAIACC phải là giá trị khác NULL.

§ MAACC, TENDANGNHAP là duy nhất.

**ª BẢNG DOITAC**

§ Các thuộc tính: TEN, NG\_DAIDIEN, THANHPHO, QUAN, SL\_CHINHANH, SL\_DONMOINGAY, LOAI\_AMTHUC, DIACHI\_KD không được là giá trị NULL.

§ Số chi nhánh trong bảng DOITAC: SL\_CHINHANH phải lớn hơn hoặc bằng tổng số chi nhánh của tất cả các chi nhánh có mã đối tác tương ứng trong bảng CHINHANH\_DK.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **THÊM** | **XÓA** | **SỬA** |
| **HOPDONG** | - | - | +(SL\_CHINHANHDK) |
| **DOITAC** | - | - | +(SL\_CHINHANH, ID\_TAIKHOAN) |

§ SL\_CHINHANH phải lớn hơn hoặc bằng 1.

§ SL\_DONMOINGAY phải lớn hơn hoặc bằng 0.

**ª BẢNG DT\_SDT**

§ Các thuộc tính: ID\_DOITAC, SDT không được là giá trị NULL.

§ SDT phải là 10 chữ số và là duy nhất.

**ª BẢNG CHINHANH\_DK**

§ Các thuộc tính: TEN\_CHINHANH, DIACHI, SDT, DOITAC, TINHTRANG, THOIGIANMO, THOIGIANDONG phải là thuộc tính khác NULL.

§ Thuộc tính TINHTRANG nhận 1 trong 2 trạng thái: BINHTHUONG, TAMNGHI.

§ THOIGIANMO phải trước THOIGIANDONG.

§ SDT phải 10 chữ số và là duy nhất.

**ª BẢNG HOPDONG**

§ Các thuộc tính: MST, NG\_DAIDIEN, SL\_CHINHANHDK, SO\_TAIKHOAN, NGANHANG, NGAYBATDAUDK, NGAYKETTHUCDK, TINHTRANG, DOITAC, THOIHANHOPDONG phải là giá trị khác NULL.

§ Thuộc tính SL\_CHINHANHDK của bảng HOPDONG phải trùng vs SL\_CHINHANH của bảng DOITAC.

§ Thuộc tính MST phải là duy nhất.

§ NGAYBATDAUDK phải trước ngày NGAYKETTHUCDK.

§ TINHTRANG phải nằm trong 3 trạng thái: DADUYET(1), CHUADUYET(0), LOAIBO(null).

§ Khi tái kí hợp đồng, NGAYBATDAUDK mới phải sau NGAYBATDAUDK cũ.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **THÊM** | **XÓA** | **SỬA** |
| **HOPDONG** | - | - | +(NGAYBATDAUDK) |

**ª BẢNG DOANHSO**

§ Các thuộc tính: DOANHSO, NAM, THANG phải là thuộc tính khác NULL.

§ DOANHSO phải lớn hơn hoặc bằng 0.

**ª BẢNG MONAN**

§ Các thuộc tính: MIEUTAMON, LOAIMONAN, GIA, CHINHANH, SL\_TON phải là thuộc tính khác NULL.

§ Thuộc tính GIA và SL\_TON phải lớn hơn hoặc bằng 0.

**ª BẢNG DONHANG**

§ Các thuộc tính: NGAYLAP, DIACHIGIAOHANG, SDT\_NHANHANG, PHIVANCHUYEN, PHUONGTHUCTHANHTOAN, TONGTIEN, TRANGTHAIXULI, CHINHANH, PHANHOI, KHACHHANG, TRANGTHAIVANCHUYEN phải là giá trị khác NULL.

§ SDT\_NHANHANG phải là duy nhất.

§ PHUONGTHUCTHANHTOAN có 2 trạng thái: CHUYENKHOAN, TIENMAT.

§ TONGTIEN phải bằng phí sản phẩm + phí vận chuyển.

§ PHANHOI có 3 loại: LIKE, DISLIKE, RATING.

§ TRANGTHAIXULI có 2 trạng thái: NHANDON, CHUANHANDON.

§ TRANGTHAIVANCHUYEN có 5 trạng thái: CHONHAN, DANGLAYHANG, LAYHANGTHANHCONG, DANGGIAO, GIAOHANGTHANHCONG.

§ PHIVANCHUYEN phải lớn hơn hoặc bằng 0.

**ª BẢNG CT\_MONDAT**

§ Các thuộc tính: DONHANG, MONAN, SOLUONG, TUYCHON phải là thuộc tính khác NULL.

§ SOLUONG phải lớn hơn 0.

**ª BẢNG KHACHHANG**

§ Các thuộc tính: HOTEN, SDT\_KH, DIACHI phải là khác NULL.

§ SDT\_KH phải là duy nhất.

**ª BẢNG TAIXE**

§ Các thuộc tính: HOTEN, CMND, SDT, DIACHI, BIENSOXE, KHUVUCHOATDONG, SOTAIKHOANNH, NGANHANG, PHITHUECHAN phải khác NULL.

§ CMND, SDT, BIENSOXE là duy nhất.

§ PHITHUECHAN phải lớn hơn hoặc bằng 0.

**ª BẢNG NHANVIEN**

§ Các thuộc tính: TENNHANVIEN, EMAIL, NGAYSINH, CHUCVU, SDT, DIACHI, LUONG phải khác NULL.

§ EMAIL, SDT phải là duy nhất.

§ LUONG phải hơn 0.

**ª BẢNG NHANVIEN\_HOPDONG**

§ Các thuộc tính: NHANVIEN, HOPDONG, NGAYKIKET phải khác NULL.

**ª BẢNG CT\_VANCHUYEN**

§ THOIGIANNHAN sẽ phải sau THOIGIANGIAO.

§ THOIGIANGIAO được xác định khi tài xế nhận đơn.

§ Nếu THOIGIANNHAN có giá trị NULL thì THOIGIANGIAO cũng sẽ NULL.

## 2.3. Mô hình cơ sở dữ liệu quan hệ

Diagram, application

Description automatically generated with medium confidence

Hình : Mô hình quan hệ

## 2.4. Mô tả bảng – thuộc tính

Kí hiệu: Khóa chính và khóa ngoại, khóa chính, khóa ngoại

|  |  |  |
| --- | --- | --- |
| **ACCOUNT** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| MAACC | VARCHAR(10) | Mỗi tài khoản có duy nhất một ID tài khoản để phân biệt với các tài khoản khác. |  |
|  |
| TENDANGNHAP | VARCHAR(50) | Tên của tài khoản. |  |
|  |
|  |
| MATKHAU | VARCHAR(50) | Mật khẩu của tài khoản. |  |
|  |
| LOAIACC | NVARCHAR(50) | Loại tài khoản đã đăng kí. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **DOITAC** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_DOITAC | VARCHAR(10) | Mỗi đối tác có duy nhất một ID đối tác để phân biệt với các đối tác khác. |  |
|  |
| TEN | NVARCHAR(100) | Tên của đối tác. |  |
|  |
| EMAIL | VARCHAR(50) | Email của đối tác. |  |
|  |
| NG\_DAIDIEN | NVARCHAR(50) | Tên của người đại diện đối tác. |  |
|  |
| THANHPHO | NVARCHAR(50) | Tên địa chỉ thành phố của đối tác. |  |
|  |
| QUAN | NVARCHAR(50) | Tên địa chỉ quận của đối tác |  |
|  |
| SL\_CHINHANH | INT | Số lượng chi nhánh của đối tác. |  |
|  |
| SL\_DONMOINGAY | INT | Số lượng đơn hàng bán được mỗi ngày của đối tác |  |
|  |
| DIACHI\_KD | NVARCHAR(100) | Địa chỉ kinh doanh của đối tác. |  |
|  |
| LOAI\_AMTHUC | NVARCHAR(50) | Loại ẩm thực mà đối tác kinh doanh. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **CHINHANH\_DK** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_CHINHANH | VARCHAR(10) | Mỗi chi nhánh có duy nhất một ID chi nhánh để phân biệt với các chi nhánh khác. |  |
|  |
| TENCHINHANH | NVARCHAR(100) | Tên của chi nhánh. |  |
|  |
| DIACHI | NVARCHAR(100) | Địa chỉ của chi nhánh |  |
|  |
| SDT | VARCHAR(10) | Số điện thoại của chi nhánh. |  |
|  |
| DOITAC | VARCHAR(10) | Đối tác sở hữu chi nhánh này. |  |
|  |
| TINHTRANG | NVARCHAR(50) | Tình trạng hiện tại của chi nhánh. |  |
|  |
| THOIGIANMO | TIME | Thời gian mở cửa của chi nhánh. |  |
|  |
| THOIGIANDONG | TIME | Thời gian đóng cửa của chi nhánh. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **DOANHSO** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_CHINHANH | VARCHAR(10) | Mã của chi nhánh. |  |
|  |
| THANG | INT | Thời gian tháng. |  |
|  |
| NAM | INT | Thời gian năm. |  |
|  |
| DOANHSO | INT | Doanh số của chi nhánh. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **NHANVIEN** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_NHANVIEN | VARCHAR(10) | Mã của nhân viên. Mỗi nhân viên có một mã duy nhất để phân biệt với các nhân viên khác. |  |
|  |
| TENNHANVIEN | NVARCHAR(50) | Tên của nhân viên. |  |
|  |
| EMAIL | NVARCHAR(50) | Email của nhân viên. |  |
|  |
| NGAYSINH | DATE | Ngày sinh của nhân viên. |  |
|  |
| CHUCVU | NVARCHAR(50) | Chức vụ của nhân viên. |  |
|  |
| SDT | VARCHAR(10) | Số điện thoại của nhân viên. |  |
|  |
| DIACHI | NVARCHAR(50) | Địa chỉ của nhân viên. |  |
|  |
| LUONG | INT | Lương của nhân viên. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **HOPDONG** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_HOPDONG | VARCHAR(10) | Mã của hợp đồng. Mỗi hợp đồng có một mã duy nhất để phân biệt với các hợp đồng khác. |  |
|  |
| MST | VARCHAR(20) | Mã số thuế của đối tác. |  |
|  |
| NG\_DAIDIEN | NVARCHAR(50) | Người đại diện cho đối tác. |  |
|  |
| SL\_CHINHANHDK | INT | Số lượng chi nhánh đối tác đằng kí. |  |
|  |
| SO\_TAIKHOAN | VARCHAR(20) | Số tài khoản ngân hàng. |  |
|  |
| NGANHANG | NVARCHAR(50) | Tên ngân hàng. |  |
|  |
| NGAYBATDAUDK | DATE | Ngày bắt đầu hợp đồng. |  |
|  |
| NGAYKETTHUCDK | DATE | Ngày kết thúc hợp đồng. |  |
|  |
| TINHTRANG | NVARCHAR(100) | Tình trạng của hợp đồng. |  |
|  |
| DOITAC | VARCHAR(10) | Đối tác ký hợp đồng. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **NHANVIEN\_HOPDONG** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| HOPDONG | VARCHAR(10) | Mã của hợp đồng. |  |
|  |
| NHANVIEN | VARCHAR(10) | Mã của nhân viên. |  |
|  |
| NGAYKYKET | DATE | Ngày ký kết hợp đồng. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **DT\_SDT** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_DOITAC | VARCHAR(10) | Mã của đối tác |  |
|  |
| SDT | VARCHAR(10) | Số điện thoại của đối tác. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **DONHANG** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_DONHANG | VARCHAR(10) | Mã của đơn hàng. Mỗi đơn hàng có một mã duy nhất để phân biệt với các đơn hàng khác. |  |
|  |
| NGAYLAP | DATE | Ngày lập đơn hàng. |  |
|  |
| DIACHIGIAOHANG | NVARCHAR(100) | Địa chỉ giao đơn hàng. |  |
|  |
| SDT\_NHANHANG | VARCHAR(10) | Số điện thoại nhận hàng. |  |
|  |
| PHIVANCHUYEN | INT | Phí vận chuyển đơn hàng. |  |
|  |
| PHUONGTHUCTHANHTOAN | NVARCHAR(50) | Phương thức thanh toán của đơn hàng. |  |
|  |
| TONGTIEN | INT | Tổng tiền của đơn hàng. |  |
|  |
| TRANGTHAIXULY | NVARCHAR(50) | Trạng thái xử lí của đơn hàng. |  |
|  |
| CHINHANH | VARCHAR(10) | Mã của chi nhánh. |  |
|  |
| PHANHOI | NVARCHAR(100) | Phản hồi của khách hàng. |  |
|  |
| KHACHHANG | VARCHAR(10) | Khách hàng đặt đơn. |  |
|  |
| TRANGTHAIVANCHUYEN | NVARCHAR(50) | Trạng thái vận chuyển của đơn hàng. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **MONAN** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| TENMON | VARCHAR(10) | Tên món ăn. Mỗi món ăn phân biệt với nhau qua tên món |  |
|  |
| MIEUTAMON | NVARCHAR(100) | Miêu tả món ăn. |  |
|  |
| LOAIMONAN | NVARCHAR(50) | Loại món ăn. |  |
|  |
| GIA | INT | Giá của món ăn. |  |
|  |
| CHINHANH | VARCHAR(10) | Chi nhánh bán món ăn. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **CT\_MONDAT** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| DONHANG | VARCHAR(10) | Mã đơn hàng. |  |
|  |
| MONAN | VARCHAR(10) | Mã món ăn. |  |
|  |
| SOLUONG | INT | Số lượng món ăn. |  |
|  |
| TUYCHON | NVARCHAR(50) | Tuỳ chọn món ăn. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **TAIXE** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_TAIXE | VARCHAR(10) | Mã tài xế. Mỗi tài xế có một mã duy nhất để phân biệt với các tài xế khác. |  |
|  |
| HOTEN | NVARCHAR(50) | Họ tên tài xế. |  |
|  |
| CMND | VARCHAR(20) | Chứng minh nhân dân của tài xế. |  |
|  |
| SDT | VARCHAR(10) | Số điện thoại tài xế. |  |
|  |
| DIACHI | NVARCHAR(100) | Địa chỉ hiện tại của tài xế. |  |
|  |
| BIENSOXE | VARCHAR(10) | Biển số xe của tài xế |  |
|  |
| KHUVUCHOATDONG | NVARCHAR(50) | Khu vực hoạt động của tài xế |  |
|  |
| SOTAIKHOAN | VARCHAR(20) | Số tài khoản ngân hàng của tài xế |  |
|  |
| NGANHANG | NVARCHAR(50) | Ngân hàng tài xế đang dùng. |  |
|  |
| PHITHUECHAN | INT | Phí thuế chân của tài xế. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **KHACHHANG** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_KHACHHANG | VARCHAR(10) | Mã khách hàng. Mỗi khách hàng có một mã duy nhất để phân biệt với các khách hàng khác. |  |
|  |
| HOTEN | NVARCHAR(50) | Thời gian giao dự kiến (tính bằng phút). |  |
|  |
| SDT\_KH | VARCHAR(10) | Số điện thoại của khách hàng. |  |
|  |
| DIACHI | NVARCHAR(100) | Địa chỉ của khách hàng. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **DT\_SDT** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_DOITAC | VARCHAR(10) | Mã của đối tác |  |
|  |
| SDT | VARCHAR(10) | Số điện thoại của đối tác. |  |
|  |

|  |  |  |
| --- | --- | --- |
| **CT\_VANCHUYEN** | **KIỂU DỮ LIỆU** | **MÔ TẢ** |
|  |
| ID\_VANCHUYEN | VARCHAR(10) | Mã vận chuyển. Mỗi vận chuyển có một mã duy nhất để phân biệt với các vận chuyển khác. |  |
|  |
| THOIGIANGIAO | TIME | Thời gian tài xế nhận đơn hàng. |  |
|  |
| THOIGIANNHAN | TIME | Thời gian khách hàng nhận được hàng. |  |
|  |
|  |
| DON | VARCHAR(10) | Mã đơn hàng. |  |
|  |
| TAIXE | VARCHAR(10) | Mã của tài xế. |  |
|  |

## 2.5. Các chức năng của hệ thống

### 2.5.1. Chức năng cho DÙNG CHUNG

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| ALL1 | Đăng nhập | Đăng nhập vào hệ thống dựa vào tài khoản và mật khẩu. |
| ALL2 | Đăng ký tài khoản | Người dùng sẽ chọn 1 trong 3 vài trò muốn trở thành (đối tác, khách hàng, tài xế). Nhập tên đăng nhập và mật khẩu mong muốn |
| ALL3 | Chỉnh sửa thông tin mật khẩu | Cho phép người dùng quyền UPDATE trên bảng TAIKHOAN(MATKHAU) phục vụ người dùng tự chỉnh sửa mật khẩu theo ý muốn |
| ALL4 | Xem thông tin tài khoản | Cho phép người dùng quyền SELECT trên bảng TAIKHOAN phục vụ nhu cầu xem thông tin đăng nhập của bản thân |

### 2.5.2. Phân hệ KHÁCH HÀNG

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| KH1 | Tạo tài khoản đăng nhập | Admin sẽ tạo tài khoản đăng nhập sẵn cho khách hàng theo thông tin mà khách hàng đã đăng ký. Khách hàng sẽ đăng nhập tài khoản và mật khẩu theo quy tắc của admin. |
| KH2 | Xem danh sách món ăn và danh sách các đối tác | Admin sẽ cấp quyền SELECT trên bảng MONAN và bảng DOITAC để phục vụ khách hàng xem danh sách thông tin món ăn và các đối tác |
| KH3 | Xem danh mục các loại món ăn | Admin sẽ cấp quyển SELECT trên MONAN để hiển thị các loại món ăn: Phở, cơm, gà rán, bún bò… |
| KH4 | Đặt hàng | Admin sẽ cấp quyền INSERTtrên bảng DONHANG, CT\_MONDAT phục vụ khách hàng được chọn món ăn và đặt hàng và chọn loại hình thức thanh toán với chi phí đã được tính sẵn. |
| KH5 | Kiểm tra danh sách các loại món ăn đã đặt | Admin sẽ cấp quyền SELECT trên bảng DONHANG, CT\_MONDAT kiểm tra sản phẩm trong giỏ hàng. |
| KH6 | Theo dõi đơn hàng | Admin sẽ cấp quyền SELECT trên bảng CT\_VANCHUYEN, TAIXE để phục vụ khách hàng muốn theo dõi đơn hàng của mình |
| KH7 | Tìm kiếm | Khách hàng tìm kiếm sản phẩm theo tên sản phẩm |

### 2.5.3. Phân hệ ĐỐI TÁC

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| DT1 | Tạo tài khoản đăng nhập | Admin sẽ tạo tài khoản đăng nhập sẵn cho đối tác theo thông tin mà đối tác đã đăng ký. Đối tác sẽ đăng nhập tài khoản và mật khẩu theo quy tắc của admin |
| DT2 | Đăng ký và cập nhật thông tin chi tiết | Admin sẽ cấp quyền SELECT, INSERT, UPDATE trên bảng DOITAC để phục vụ cho đối tác muốn xem, đăng ký thông tin và cập nhật thông tin. |
| DT3 | Xem, thêm chỉnh sửa các chi nhánh của đối tác | Admin sẽ cấp quyền SELECT, INSERT UPDATE, DELETE trên bảng CHINHANH\_DK để phục vụ đối tác xem các chi nhánh đã đăng kí của mình |
| DT4 | Thêm thông tin, chỉnh sửa và xóa thông tin của 1 món ăn | Admin sẽ cấp quyến SELECT, INSERT, UPDATE, DELETE trên bảng MONAN để phục vụ đối tác muốn xem, thêm, chỉnh sửa, xóa của sản phẩm. |
| DT5 | Xem thông tin đơn hàng có sản phẩm của mình | Admin sẽ cấp quyền SELECT trên bảng DONHANG,  CT\_MONAN, TAIXE (HOTEN, SDT) để  phục vụ đối tác muốn xem thông tin các đơn hàng mua món ăn của mình. |
| DT6 | Xem và thêm 1 hợp đồng | Admin sẽ cấp quyền SELECT, INSERT trên bảng HOPDONG để phục vụ đối tác muốn thêm hợp đồng. |
| DT7 | Xem thông tin khách hàng | Admin sẽ cấp quyền SELECT trên bảng  KHACHHANG để phục vụ đối tác muốn xem khách hàng nào đã đặt sản phẩm của mình. |

### 2.5.4. Phân hệ TÀI XẾ

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| TX1 | Tạo tài khoản đăng nhập | Admin sẽ tạo tài khoản đăng nhập sẵn cho tài xế theo thông tin mà tài xế đã đăng ký. Tài xế sẽ đăng nhập tài khoản và mật khẩu theo quy tắc của admin. |
| TX2 | Xem thông tin, cập nhật, thêm thông tin tài xế | Admin sẽ cấp quyền SELECT, INSERT, UPDATE trên bảng TAIXE để phục vụ cho tài xế muốn xem, thêm thông tin hoặc cập nhật. |
| TX3 | Xem danh sách đơn hàng theo khu vực hoạt động của mình | Admin sẽ cấp quyền trên bảng DONHANG, CT\_MONDAT, KHACHHANG,CT\_VANCHUYEN để phục vụ tài xế xem danh sách đơn hàng khu vực hoặt động của mình |
| TX4 | Tiếp nhận đơn hàng | Admin sẽ cấp quyền UPDATE trên bảng CT\_VANCHUYEN, DONHANG, TAIXE để phục vụ tài xế tiếp nhận đơn hàng. |

### 2.5.4. Phân hệ NHÂN VIÊN

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| NV1 | Tạo tài khoản đăng nhập | Admin sẽ tạo tài khoản đăng nhập sẵn cho nhân viên theo thông tin mà khách hàng đã đăng ký. Nhân viên sẽ đăng nhập tài khoản và mật khẩu theo quy tắc của admin |
| NV2 | Cập nhật thông tin chi tiết cho nhân viên | Admin sẽ cấp quyền UPDATE cho bảng NHANVIEN, SELECT bảng NHANVIEN cho từng nhân viên để phục vụ thay đổi các thông tin của chính mình. |
| NV3 | Duyệt hợp đồng | Admin sẽ cấp quyền UPDATE trên bảng HOPDONG để phục vụ nhân viên duyệt hợp đồng. |
| NV4 | Xem danh sách đối tác và hợp đồng | Admin sẽ cấp quyền SELECT trên bảng DOITAC và HOPDONG của đối tác đó – có mã hợp đồng đối với đối tác đã đăng ký hợp đồng; hiển thị thông báo hợp đồng sắp hết hạn |
| NV5 | Gia hạn hợp đồng | Nhân viên kiểm tra ngày hết hạn hợp đồng. Nếu còn 7 ngày hết hạn thì nhân viên sẽ gửi thông báo đến đối tác. |

### 2.5.5. Phân hệ QUẢN TRỊ

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả hoạt động** |
| QT1 | Tạo user | Admin sẽ tạo tài khoản đăng nhập sẵn cho từng vai trò theo tên đăng nhập và quy tắc dành cho mật khẩu khởi tạo. |
| QT2 | Quản lý tài khoản | Xem danh sách tài khoản. Xóa, sửa, cập nhật tài khoản. |

# III. PHÂN QUYỀN NGƯỜI DÙNG

# IV. THIẾT KẾ GIAO DIỆN HỆ THỐNG

## 4.1. Dùng chung

### 4.1.1. Chức năng đăng nhập

A picture containing diagram

Description automatically generated

Hình : Giao diện đăng nhập

- Để sử dụng được hệ thống đặt và giao hàng online thì người dùng cần đăng nhập với các thông tin đã đăng kí từ trước, vô phần giao diện Đăng nhập, nhập tên đăng nhập và mật khẩu, sau đó nhấn nút đăng nhập để vô được hệ thống.

- Nếu chưa có tài khoản, hãy nhấn vào dòng Đăng kí để đăng kí tài khoản.

- Thoát giao diện bằng cách nhấn vào nút thoát.

### 4.1.2. Chức năng đăng kí

A picture containing rectangle

Description automatically generated

Hình : Giao diện đăng ký

- Để đăng kí, người dùng cần hoàn thành nhập vào tên muốn đăng kí, mật khẩu và xác nhận mật khẩu, sau đó nhấn đăng kí để hoàn tất đăng kí tài khoản, nếu có tài khoản thì nhấn vào Quay lại.

## 4.2. Phân hệ KHÁCH HÀNG

### 4.2.1. Chức năng thông tin cá nhân

Graphical user interface

Description automatically generated

Hình : Giao diện khách hàng

- Khi đăng nhập thành công, khách hàng sẽ hiện lên giao diện tài khoản:

* Nếu người dùng đã cập nhật thông tin sau khi đăng nhập, thì thông tin tài khoản của người dùng sẽ hiện lên trên bảng thông tin tài khoản.

Nếu bạn muốn cập nhật thông tin của bạn chỉ cần nhập vào thông tin muốn thay đổi trên các ô (Họ tên, số điện thoại, Địa chỉ) rồi nhấn cập nhật.

* Nếu là người dùng vừa mới đăng kí tài khoản, thì người dùng nên thêm thông tin ngay khi đăng nhập thành công.

### 4.2.2. Chức năng xem đơn hàng đã đặt

A picture containing timeline

Description automatically generated

Hình : Giao diện lịch sử đặt hàng

- Khi nhấn vào chức năng đơn hàng đã đặt, thì giao diện danh sách đơn đã đặt xuất hiện, khách hàng sẽ thấy được một bảng các danh sách các đơn hàng đã đặt, muốn xem chi tiết các đơn hàng, khách hàng chỉ cấn nhấn vào một ô bất kì trong danh sách đơn hàng, hệ thống sẽ hiện chi tiết đơn hàng vào các ô nằm trên danh sách.

### 4.2.3. Chức năng đặt hàng

Timeline

Description automatically generated with low confidence

Hình : Giao diện đặt hàng

- Khi khách hàng nhập vào chức năng đặt hàng, giao diện **danh sách đối tác** sẽ xuất hiện để mọi người có thể thấy được thông tin đối tác bán thức ăn gì.

- Để xem chi tiết đối tác, khách hàng nhấn vào tên đối tác để xem chi tiết các công tin của đối tác.

-Sau khi chọn đối tác có loại món ăn mà khách hàng muốn đặt thì khách hàng nhấn vào xem danh sách món ăn của đối tác.

Graphical user interface

Description automatically generated with medium confidence

Hình : Giao diện danh sách món ăn

- Giao diện chức năng của **danh sách món ăn** của từng đối xuất hiện, bạn chỉ cần chọn món ăn muốn mua sau đó nhập vào ô **số lượng muốn mua**, rồi nhấn vào ô mua ngay để mua hàng, còn nếu không muốn mua thì nhấn quay lại để xem danh sách đối tác.

Timeline

Description automatically generated

Hình : Giao diện điền thông tin đặt hàng

Sau khi nhấn **Mua ngay,** giao diện đặt hàng xuất hiện, nhập số điện thoại, nhập tùy chọn màn bạn muốn đối tác làm cho bạn và chọn hình thức thanh toán, sau đó nhấn đặt hàng.

## 4.3. Phân hệ ĐỐI TÁC

### 4.3.1. Chức năng thông tin

Graphical user interface, application, website

Description automatically generated

Hình : Giao diện hiển thị thông tin đối tác

- Sau khi đăng nhập thành công, thì đối tác sẽ xem được thông tin, đối tác muốn cập nhật thông tin của mình thì nhấn vào bảng thông tin rồi nhấn vào nút **cập nhật thông tin**.

**-** Nếu mới đăng kí vào hệ thống thì đối tác phải thêm thông tin của đối tác vào các ô hiện trên giao diện sau đó nhấn thêm thông tin.

### 4.3.2. Chức năng chi nhánh

Graphical user interface, application

Description automatically generated

Hình : Giao diện hiển thị thông tin chi nhánh

- Đối tác chọn chức năng chi nhánh, sẽ thấy được danh sách các chi nhánh của mình.

-Đối tác chọn chi nhánh muốn cập nhật, sau đó hoàn tất cập nhật bằng cách nhấn nút cập nhật.

- Đối tác muốn thêm chi nhánh, nhấn vào nút thêm chi nhánh.

Diagram, timeline

Description automatically generated

Hình : Giao diện chức năng mở thêm chi nhánh mới

-Nhập các thông tin cần thiết của chi nhánh, sau đó nhấn nút thêm.

### 4.3.3. Chức năng Món Ăn

A picture containing calendar

Description automatically generated

Hình : Giao diện hiển thị danh sách món ăn của từng đối tác

- Đối tác có thể xem các món ăn của mình ở ngay bảng danh sách món ăn, nhập các thông tin cần thiết để thêm món ăn.

- Để cập nhật món ăn, đối tác có thể nhấn vô bảng để lấy thông tin của món ăn, thông tin sẽ hiện trên các ô của **nhập thông tin cần thiết.**

-Để xóa đi thức ăn, đối tác chọn thông món ăn cần xóa ở danh sách món ăn và nút xóa.

### 4.3.4. Chức năng lập hợp đồng

Graphical user interface

Description automatically generated

Hình : Giao diện kí hợp đồng mới

- Đối tác có thể xem các hợp đồng của đối tác đã lập

- Để thêm hợp đồng nhập đủ các thông tin trong các ô, sau đó nhấn nút thêm để thêm hợp đồng

### 4.3.5. Chức năng xem hợp đồng đã lập

Text

Description automatically generated with medium confidence

Hình : Giao diện xem hợp đồng đã đăng kí

- Xem hợp đồng đã lập của đối tác

- Nhấn xem chi nhánh để xem hợp đồng này thuộc về chi nhánh nào

### 4.3.6. Chức năng đơn hàng

Graphical user interface, application

Description automatically generated

Hình : Giao diện hiển thị danh sách đơn hàng của đối tác

- Xem các đơn hàng mà mua thức ăn của đối tác.

## 4.4. Phân hệ TÀI XẾ

### 4.4.1. Chức năng tài khoản

Graphical user interface, website

Description automatically generated

Hình : Giao diện thêm, chỉnh sửa thông tin của tài xế

* Sau khi tài xế đăng nhập thành công, nếu là người mới đăng kí và chưa có thông tin trước đó thì tài xế phải thêm thông tin vào.
* Nếu đã có thông tin trước đó và muốn đăng nhập thì nhấn vào cập nhật thông tin.

### 4.4.2. Chức năng Danh sách đơn hàng

Graphical user interface, application

Description automatically generated with medium confidence

Hình : Giao diện hiển thị các đơn hàng phù hợp với tài xế

- Danh sách đơn hàng sẽ hiện ra theo khu vực của tài xế, tài xế chọn đơn hàng muốn nhận và nhấn chọn **Nhận đơn hàng.**

### 4.4.3. Chức năng đơn hàng đã nhận

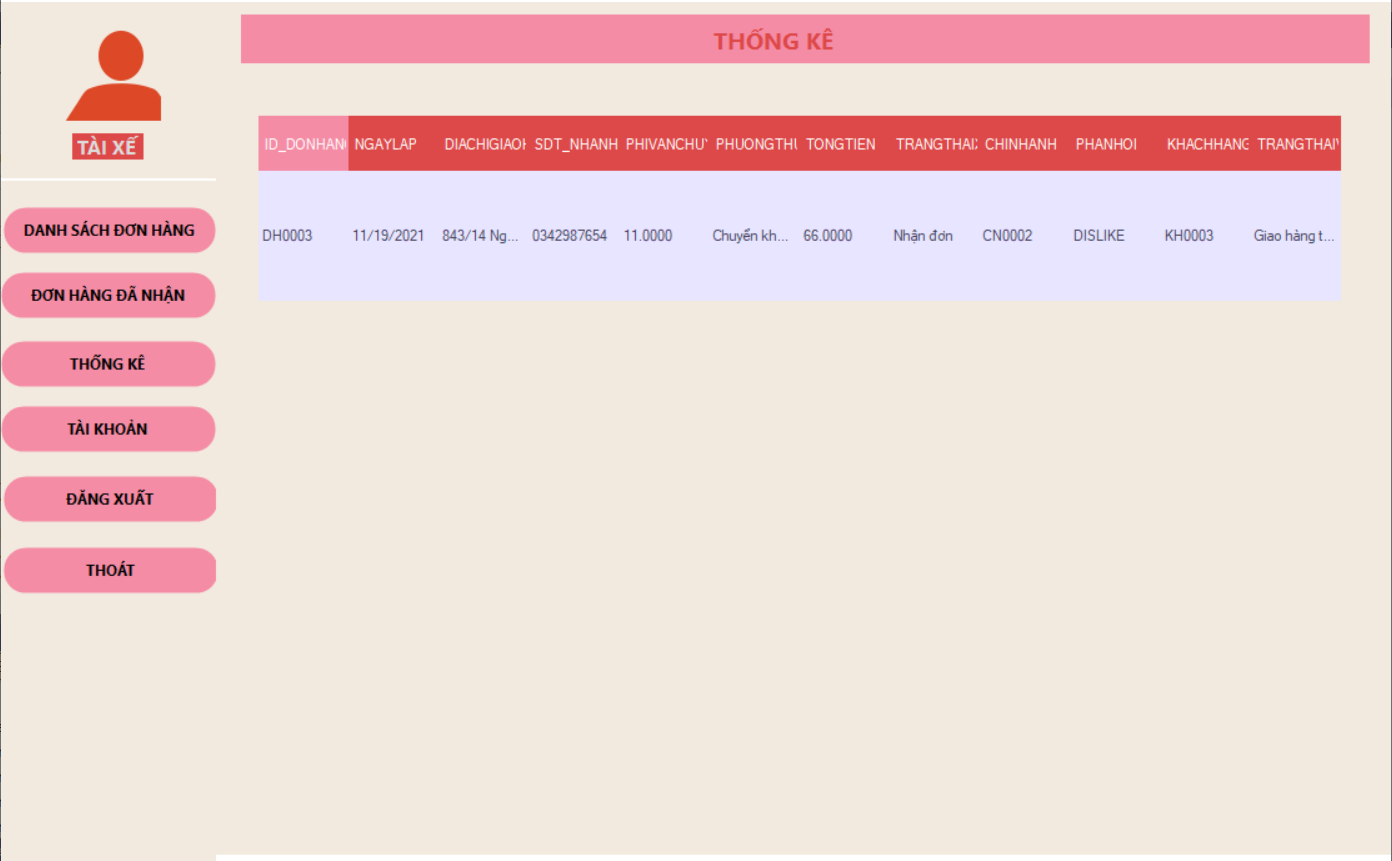
Graphical user interface, text, application

Description automatically generated

Hình : Giao diện xem lịch sử đơn hàng đã nhận của tài xế

- Chức năng xem danh sách đơn hàng đã nhận và cho phép cập nhật đơn hàng là đã giao thành công hay chưa.

### 4.4.4. Chức năng Thống kê



Hình : Giao diện thống kê các đơn giao hàng của tài xế

-Chức năng thống kê cho biết đơn hàng của tài xế đã giao thành công và cho biết thêm phí vận chuyển của từng đơn

## 4.5. Phân hệ NHÂN VIÊN

### 4.5.1. Chức năng tài khoản

Graphical user interface, application, email

Description automatically generated

Hình : Giao diện hiển thị thông tin nhân viên

* Sau khi nhân viên đăng nhập thành công, nếu là người mới đăng kí và chưa có thông tin trước đó thì tài xế phải thêm thông tin vào.
* Nếu đã có thông tin trước đó và muốn đăng nhập thì nhấn vào cập nhật thông tin.

### 4.5.2. Chức năng xem hợp đồng đã duyệt

Graphical user interface, application

Description automatically generated- Nhân viên xem hợp đồng đã duyệt, sau đó xem hiệu lực hợp đồng để thông báo gia hạn cho đối tác

Hình : Giao diện hiển thị hợp đồng đã duyệt

### 4.5.3. Chức năng hợp đồng chưa duyệt

A picture containing graphical user interface

Description automatically generated

Hình : Giao diện hiển thị danh sách hợp đồng chưa duyệt

* Xem danh sách chức năng hợp đồng chưa duyệt, nhân viên xem duyệt hay loại bỏ.

## 4.6. Phân hệ QUẢN TRỊ

### 4.6.1 Chức năng xem tài khoản

Table

Description automatically generated

Hình : Giao diện quản lí tài khoản của quản trị viên

* Người quản trị được xem tất cả thông tin về tài khoản của người dừng
* Có thể đăng kí thêm một tài khoản khi nhấn nút thêm
* Xem được thông tin ứng với từng loại tài khoản

A picture containing timeline

Description automatically generated

Hình : Giao diện chi tiết từng tài khoản

- Có thể khóa tài khoản

# V. TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: Võ Thị Kiều Khanh - 20120305

1. Tình huống 1:

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Conversion DeadLock**  T1 (User = Đối tác): thực hiện cập nhật số lượng chi nhánh trong Đối Tác  T2 (User = Đối tác): thực hiện cập nhật người đại diện trong Đối tác | | | |
| **sp\_Them1ChuyenXe** | **Khóa** | **sp\_ThongKeChuyenXe-CoDieuKien** | **Khóa** |
| ***Input:*** @soluong, @id\_doitac  ***Output***: Deadlock |  | ***Input***: @nguoidaidien, @id\_doitac  ***Output***: Deadlock |  |
| SET TRANSACTION ISOLATION  LEVEL **REPEATEABLE READ** |  | SET TRANSACTION ISOLATION  LEVEL **REPEATEABLE READ** |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra số lượng chi nhánh muốn đổi có trùng với số lượng hiện tại hay không  DECLARE @soluong\_HT INT  SET @soluong\_HT = (SELECT SL\_CHINHANH  FROM DOITAC  WHERE ID\_DOITAC = @id\_doitac )    IF (@soluong\_HT = @soluong)  BEGIN  PRINT N'SỐ LƯỢNG CHI NHÁNH MUỐN ĐỔI TRÙNG VỚI CHI NHÁNH HIỆN TẠI !'  ROLLBACK TRAN  RETURN  END | **S(DOITAC)**  //Xin khoá đọc trên DOITAC      //giữ khóa đến hết giao tác |  |  |
| WAITFOR DELAY '0:0:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra người đại diện muốn đổi có trùng với người đại diện hiện tại không.  DECLARE @nguoidaidienhientai nvarchar(30)  SET @nguoidaidienhientai = (SELECT NG\_DAIDIEN  FROM DOITAC  WHERE ID\_DOITAC = @id\_doitac)  IF (@nguoidaidienhientai = @nguoidaidienhientai)  BEGIN  PRINT N'TÊN NGƯỜI ĐẠI DIỆN MUỐN ĐỔI TRÙNG VỚI NGƯỜI ĐẠI DIỆN BÂY GIỜ '  ROLLBACK TRAN  RETURN  END | **S(DOITAC)**  //Xin khoá đọc trên DOITAC      //giữ khóa đến hết giao tác |
|  |  | WAITFOR DELAY '0:0:10' |  |
| B2: Cập nhật số lượng chi nhánh  UPDATE DOITAC  SET SL\_CHINHANH = @soluong  WHERE ID\_DOITAC = @id\_doitac | XDOITAC)  Xin khóa ghi trên  DOITAC |  |  |
| B3: Bắt lỗi hệ thống và in ra lỗi (nếu có)  BEGIN CATCH  DECLARE @ERROR VARCHAR(2000)  SELECT @ERROR = N'Lỗi: ' + ERROR\_MESSAGE()  RAISERROR(@ERROR, 16,1)  ROLLBACK TRAN  RETURN  END CATCH |  |  |  |
|  |  | B2: Cập nhật tên người đại diện  UPDATE DOITAC  SET NG\_DAIDIEN = @nguoidaidien  WHERE ID\_DOITAC = @id\_doitac | X(DOITAC)  Xin khóa ghi trên  DOI\_TAC)  //Chờ |
|  |  | B3: Bắt lỗi hệ thống và in ra lỗi (nếu có)  BEGIN CATCH  DECLARE @ERROR VARCHAR(2000)  SELECT @ERROR = N'Lỗi: ' + ERROR\_MESSAGE()  RAISERROR(@ERROR, 16,1)  ROLLBACK TRAN  RETURN  END CATCH |  |
| COMMIT |  | COMMIT |  |

II. Sinh viên thực hiện: Ngô Võ Quang Huy – 20120295

1. Tình huống 1:

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Cycle Deadlock***  T1 (User = NHANVIEN): thực hiện thêm hợp đồng mới cho đối tác, cập nhật tình trạng ‘Hết hạn’ hợp đồng cũ của đối tác.  T2 (User = DOITAC): thực hiện cập nhật người đại diện cho hợp đồng HD01 của mình, cập nhật người đại diện cho hợp đồng mới của mình | | | |
| **sp\_Them1HopDong** | **Khóa** | **sp\_CapNhatNDD** | **Khóa** |
| ***Input:*** @idHDcu = ‘HD01’,  @idHDmoi, @doitac = ‘DT01’,@mst, @ndd, @slcn, @stk,@nganhang, @ngaybd,@ngaykt, @tinhtrang  ***Output***: |  | ***Input***: ***:*** @idHDcu, @idHDmoi,@doitac = ‘DT01’, @ndd  ***Output***: |  |
|  |  |  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin @idHDmoi chưa tồn tại và @doitac đã tồn tại.    IF NOT EXISTS (SELECT \* FROM DOITAC WHERE ID\_DOITAC = @doitac)  BEGIN  PRINT N’Mã đối tác không tồn tại’  ROLLBACK TRAN  END    IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDmoi)  BEGIN  PRINT N’Mã hợp đồng mới đã tồn tại’  ROLLBACK TRAN  END    IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDcu)  BEGIN  PRINT N’Mã hợp đồng cũ không tồn tại’  ROLLBACK TRAN  END            IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDcu AND DOITAC = @doitac)  BEGIN  PRINT N'Đối tác không có hợp đồng cũ này'  ROLLBACK TRAN  END | **R(DOITAC)**  //Không cần xin khoá        **R(HOPDONG)**  //Không cần xin khoá          **R(HOPDONG)**  //Không cần xin khoá              **R(HOPDONG)**  //Không cần xin khoá |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
| B2: Thêm hợp đồng mới cho đối tác    INSERT INTO HOPDONG VALUES (@idHDmoi, @mst, @ndd, @slcn, @stk,@nganhang, @ngaybd, @ngaykt, @tinhtrang, @doitac) | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG để thêm 1 hợp đồng mới là @idHDmoi | B1: Kiểm tra thông tin @idHDmoi chưa tồn tại, @doitac đã tồn tại và @idHDcu đã tồn tại.    IF NOT EXISTS (SELECT \* FROM DOITAC WHERE ID\_DOITAC = @doitac)  BEGIN  PRINT N’Mã đối tác không tồn tại’  ROLLBACK TRAN  END    IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDmoi)  BEGIN  PRINT N’Mã hợp đồng mới đã tồn tại’  ROLLBACK TRAN  END    IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDcu)  BEGIN  PRINT N’Mã hợp đồng cũ không tồn tại’  ROLLBACK TRAN  END    IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDcu AND DOITAC = @doitac)  BEGIN  PRINT N'Đối tác không có hợp đồng cũ này'  ROLLBACK TRAN  END | **R(DOITAC)**  //Không cần xin khoá        **R(HOPDONG)**  //Không cần xin khoá      **R(HOPDONG)**  //Không cần xin khoá |
| B3: Cập nhật tình trạng cho hợp đồng cũ    UPDATE HOPDONG  SET TINHTRANG = N‘Hết hạn’  WHERE ID\_HOPDONG = @idHDcu | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG với điều kiện ID\_HOPDONG = @idHDcu. Tuy nhiên, khoá X trên đơn vị dữ liệu này đã bị T2 chiếm giữ nên T1 phải chờ. | B2: Cập nhật người đại diện hợp đồng cũ    UPDATE HOPDONG  SET NG\_DAIDIEN = @ndd  WHERE ID\_HOPDONG = @idHDcu | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG với điều kiện ID\_HOPDONG = @idHDcu. |
|  |  | B3: Cập nhật người đại diện hợp đồng mới    UPDATE HOPDONG  SET NG\_DAIDIEN = @ndd  WHERE ID\_HOPDONG = @idHDmoi | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG với điều kiện ID\_HOPDONG = @idHDmoi. Tuy nhiên, khoá X trên đơn vị dữ liệu này đã bị T1 chiếm giữ nên T2 phải chờ |
| COMMIT |  |  |  |
|  |  | COMMIT |  |
|  |  |  |  |

III. Sinh viên thực hiện: 20120318 \_ Nguyễn Lê Mỹ Linh

1. Tình huống 1: Đối tác đang thực hiện thao tác xóa một chi nhánh nhưng chưa commit thì khách hàng vào thực hiện thao tác xem thông tin danh sách chi nhánh hiện có của đối tác.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ERR01: *Dirty read***  T1 (User = Đối tác): thực hiện cập nhật địa chỉ một chi nhánh  T2 (User = Khách hàng): thực hiện xem danh sách chi nhánh của một đối tác | | | | | | |
| **sp\_CapNhatDiaChiCN** | **Khóa** | **sp\_XemDanhSachCN** | | **Khóa** | |
| ***Input:*** *@tenmoi*, *@macn, @madt*  ***Output***: |  | ***Input***: *@madt*  ***Output***: | |  | |
| SET TRANSACTION ISOLATION LEVEL READ COMMITTED |  | SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED | |  | |
| BEGIN TRAN |  | |  | |  | |
| B1: Kiểm tra thông tin chi nhánh có tồn tại không.  IF NOT EXISTS ( SELECT \*  FROM CHINHANH\_DK CN  WHERE CN.ID\_CHINHANH = @macn)  BEGIN  PRINT N'CHI NHÁNH KHÔNG TỒN TẠI'  ROLLBACK TRAN  RETURN  END | **R(CHINHANH\_DK)**  //Xin khoá đọc trên bảng CHINHANH\_DK với đk ID\_CHINHANH = @macn | |  | |  | |
| B2: Kiểm tra thông tin chi nhánh có thuộc đối tác không  IF NOT EXISTS ( SELECT \*  FROM CHINHANH\_DK CN  WHERE CN.ID\_CHINHANH = @macn AND CN.DOITAC=@madt)  BEGIN  PRINT N'KHÔNG CÓ QUYỀN THAO TÁC VÌ CHI NHÁNH NÀY KHÔNG THUỘC VỀ ĐỐI TÁC'  ROLLBACK TRAN  RETURN  END | **R(CHINHANH\_DK)**  //Xin khoá đọc trên bảng CHINHANH\_DK với đk ID\_CHINHANH = @macn AND DOITAC=@madt | |  | |  | |
| B3: Thực hiện thao tác cập nhật địa chỉ một chi nhánh  UPDATE CHINHANH\_DK  SET DIACHI = @tenmoi  WHERE ID\_CHINHANH = @macn | **U,X(CHINHANH\_DK)**  // Xin khóa cập nhật, ghi trên bảng CHINHANH\_DK với điều kiện ID\_CHINHANH = @macn | |  | |  | |
| WAITFOR DELAY '00:00:10' |  | |  | |  | |
|  |  | | BEGIN TRAN | |  | |
|  |  | | B1: Kiểm tra xem đối tác có tồn tại không  IF NOT EXISTS ( SELECT \*  FROM DOITAC  WHERE ID\_DOITAC = @madt)  BEGIN  PRINT N'KHÔNG TỒN TẠI ĐỐI TÁC NÀY'  ROLLBACK TRAN  RETURN  END | | **R(DOITAC)**  //Không cần xin khoá | |
|  |  | | B2:Xem danh sách chi nhánh hiện có của đối tác  SELECT TENCHINHANH, DIACHI, SDT, TINHTRANG, THOIGIANMO, THOIGIANDONG  FROM CHINHANH\_DK  WHERE DOITAC = @madt | | **R(CHINHANH\_DK)**  //Không cần xin khoá | |
|  |  | | COMMIT | |  | |
| B4: Kiểm tra địa chỉ mới cập nhật liệu có thuộc về một chi nhánh nào khác thuộc bảng CHINHANH\_DK không  IF (SELECT COUNT(\*)  FROM CHINHANH\_DK  WHERE DIACHI = @tenmoi) > 1  BEGIN  PRINT N'ĐỊA CHỈ MỚI KHÔNG HỢP LỆ VÌ ĐÃ TỒN TẠI ĐỊA CHỈ NÀY, KHÔNG THỂ CÓ NHIỀU HƠN MỘT CHI NHÁNH CHO MỘT ĐỊA CHỈ'  ROLLBACK TRAN  RETURN  END |  | |  | |  | |
| COMMIT |  | |  | |  | |

2. Tình huống 2: Khách hàng thực hiện thao tác tìm các món ăn thuộc một loại món ăn cho trước ở một chi nhánh nhưng đồng thời bên đối tác thực hiện thao tác cập nhật tên gọi mới cho loại món ăn này ở chi nhánh trên.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ERR02: *Unrepeatable Read***  T1 (User = Khách hàng): thực hiện tìm các món ăn thuộc loại món ăn cho trước ở một chi nhánh  T2 (User = Đối tác): thực hiện cập nhật tên gọi mới của loại món ăn này | | | | | | |
| **sp\_Them1ChuyenXe** | **Khóa** | **sp\_ThongKeChuyenXe-CoDieuKien** | | **Khóa** | |
| ***Input:*** *@tenloai, @macn*  ***Output***: |  | ***Input:*** *@tencu, @tenmoi, @macn*  ***Output***: | |  | |
| SET TRAN ISOLATION LEVEL READ COMMITTED |  | SET TRAN ISOLATION LEVEL READ COMMITTED | |  | |
| BEGIN TRAN |  | |  | |  | |
| B1: Kiểm tra thông tin loại món ăn ở chi nhánh có tồn tại không  IF NOT EXISTS ( SELECT \*  FROM MONAN  WHERE LOAIMONAN = @tenloai AND CHINHANH = @macn)  BEGIN  PRINT N'KHÔNG CÓ TÌM KIẾM PHÙ HỢP'  ROLLBACK TRAN  RETURN  END | **R(MONAN)**  //Xin khoá đọc trên bảng MONAN với đk LOAIMONAN = @tenloai and CHINHANH = @macn | |  | |  | |
| WAITFOR DELAY '00:00:10' |  | |  | |  | |
|  |  | | BEGIN TRAN | |  | |
|  |  | | B1: Kiểm tra thông tin tên loại món ăn cũ ở chi nhánh có tồn tại không  IF NOT EXISTS ( SELECT \*  FROM MONAN  WHERE LOAIMONAN = @tencu AND CHINHANH = @macn)  BEGIN  PRINT N'KHÔNG CÓ TÌM KIẾM PHÙ HỢP'  ROLLBACK TRAN  RETURN  END | | **R(MONAN)**  //Xin khoá đọc trên bảng MONAN với đk LOAIMONAN = @tencu CHINHANH = @macn | |
|  |  | | B2: Thực hiện cập nhật lại tên mới cho loại món ăn mong muốn  UPDATE MONAN  SET LOAIMONAN = @tenmoi  WHERE LOAIMONAN = @tencu AND CHINHANH = @macn | | **U,X(MONAN)**  //Xin khóa cập nhật, ghi cho trên bảng MONAN với dk LOAIMONAN = @tencu CHINHANH = @macn | |
|  |  | | COMMIT | |  | |
| B2: Thực hiện in ra danh sách món ăn cần tìm  SELECT \*  FROM MONAN  WHERE LOAIMONAN = @tenloai AND CHINHANH = @macn | **R(MONAN)**  //Xin khoá đọc trên bảng MONAN với đk LOAIMONAN = @tenloai and CHINHANH = @macn | |  | |  | |
| COMMIT |  | |  | |  | |

1. Sinh viên thực hiện: Phạm Mai Thiên Bảo – 20120255
2. Tình huống 1: Nhân viên đang liệt kê danh sách hợp đồng chưa được duyệt thì đối tác thêm một hợp đồng mới.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ERR01: Phantom Read**  T1 (User = Nhân viên): Nhân viên liệt kê hợp đồng chưa duyệt của đối tác A  T2 (User = Đối tác): Đối tác A thêm mới một hợp đồng. | | | | |
| ***usp\_Liet\_Ke\_HD\_Chua\_Duyet*** | **Khóa** | ***usp\_Them\_Hop\_Dong*** | | **Khóa** |
| ***Input:*** *Mã đối tác (@iD\_DoiTac)*  ***Output***: Danh sách hợp đồng chưa duyệt và tổng số hợp đồng chưa duyệt. |  | ***Input:*** Thông tin hợp đồng (mã hợp đồng, mã số thuế,…)  ***Output***: Thêm một hợp đồng và xuất thông báo thêm hợp đồng thành công. Nếu thêm không thành công thì xuất thông báo lỗi | |  |
| SET TRANSACTION ISOLATION  LEVEL **READCOMMITED** |  | SET TRANSACTION ISOLATION  LEVEL **READCOMMITED** | |  |
| B1: Kiểm tra mã đối tác có tồn tại?  IF EXISTS (SELECT \* FROM DOITAC WHERE ID\_DOITAC = @iD\_DoiTac)  BEGIN  PRINT(N'Đối tác không tồn tại')  ROLLBACK TRAN  END | **R(DOITAC)**  //Xin khoá đọc trên bảng A với điều kiện ID\_DOITAC = @iD\_DoiTac  //Đọc xong trả khóa | |  |  |
| B2: Kiểm tra đối tác đã có hợp đồng?  IF EXISTS (SELECT \* FROM HOPDONG WHERE DOITAC = @iD\_DoiTac)  BEGIN  PRINT(N'Đối tác không có hợp đồng hợp lệ')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac  //Đọc xong trả khóa | |  |  |
| B3: Kiểm tra đối tác có hợp đồng chưa duyệt?  IF EXISTS (SELECT \* FROM HOPDONG WHERE DOITAC = @iD\_DoiTac AND TINHTRANG = 'CHUADUYET')  BEGIN  PRINT(N'Đối tác không có hợp đồng chưa được duyệt')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac và TINHTRANG = 'CHUADUYET'  //Đọc xong trả khóa | |  |  |
| B4: Đọc số lượng hợp đồng.  SELECT @soLuongHD = COUNT(\*) FROM HOPDONG WHERE DOITAC = @iD\_DoiTac | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac  //Đọc xong trả khóa | |  |  |
| WAITFOR DELAY '0:0:10' |  | |  |  |
|  |  | | BEGIN TRAN |  |
|  |  | | B1: Kiểm tra mã hợp đồng đã tồn tại?  IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD)  BEGIN  PRINT(N'Mã hợp đồng đã tồn tại')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD  //Đọc xong trả khóa |
|  |  | | B2: Thêm hợp đồng mới.  INSERT INTO HOPDONG  VALUES(@iD,@mST,@ngDaiDien,@sLChiNhanh,@soTaiKhoan,@nganHang,@ngBD,@ngKT,'CHUADUYET',@doiTac) | **X(HOPDONG)**  //Xin khóa độc quyền trên bảng HOPDONG  //Giữ đến khi COMMIT/  ROLLBACK |
|  |  | | COMMIT TRAN |  |
| B5: Liệt kê hợp đồng.  DECLARE CURS CURSOR FOR  SELECT ID\_HOPDONG FROM HOPDONG WHERE DOITAC = @iD\_DoiTac  OPEN CURS  FETCH NEXT FROM CURS INTO @iD\_HD  WHILE @@FETCH\_STATUS = 0  BEGIN  PRINT @iD\_HD  FETCH NEXT FROM CURS INTO @iD\_HD  WAITFOR DELAY '0:0:5'  END  CLOSE CURS  DEALLOCATE CURS  PRINT N'Tổng số hợp đồng chưa duyệt của đối tác ' + @iD\_DoiTac + ': ' + CAST (@soLuongHD AS NVARCHAR) | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac  //Đọc xong trả khóa | |  |  |
| COMMIT TRAN |  | |  |  |
| *Lưu ý: Việc xin khóa sẽ luôn thành công nếu không có bất kì lưu ý nào khác đính kèm.* | | | | |

1. Tình huống 2: Cả 2 nhân viên đều đang xét duyệt một hợp đồng đang ở tình trạng chưa duyệt:

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR02: Phantom Read**  T1 (User = Nhân viên): NV1 cho rằng hợp đồng không phù hợp, thiết lập tình trạng hợp đồng là xóa bỏ (thực hiện trước).  T2 (User = Nhân viên): NV2 lại cho rằng hợp đồng đầy đủ các tiêu chí mà công ty yêu cầu, nên thiết lập tình trạng hợp đồng từ chưa duyệt sang đã duyệt (thực hiện ngay sau đó), biết rằng chỉ có hợp đồng chưa duyệt mới được chuyển sang đã duyệt, đây là trường hợp duy nhất. | | | |
| ***usp\_Dat\_Trang\_Thai\_XoaBo\_HD*** | **Khóa** | **usp\_Update\_1\_Hop\_Dong\_Chua\_Duyet\_Sang\_Da\_Duyet** | **Khóa** |
| ***Input:*** Mã hợp đồng (@iD\_HD)  ***Output***: Cập nhật tình trạng hợp đồng và xuất thông báo. |  | ***Input***: Mã hợp đồng (@iD\_HD)  ***Output***: Cập nhật tình trạng hợp đồng và xuất thông báo. |  |
| SET TRANSACTION ISOLATION  LEVEL READCOMMITED |  | SET TRANSACTION ISOLATION  LEVEL READCOMMITED |  |
| BEGIN TRAN |  |  |  |
| SELECT @tinhTrangHD = TINHTRANG FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Đọc xong trả khóa |  |  |
| IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD)  BEGIN  PRINT(N'Không tồn tại hợp đồng cần xóa')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Đọc xong trả khóa |  |  |
| IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD AND TINHTRANG = 'XOABO')  BEGIN  PRINT(N'Hợp đồng đã ở trạng thái này từ trước')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD và TINHTRANG = ‘XOABO’  //Đọc xong trả khóa |  |  |
| IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD AND DATEDIFF(DAY,NGAYKETTHUCDK, GETDATE()) < 0 AND TINHTRANG = 'DADUYET')  BEGIN  PRINT(N'Hợp đồng đang còn hiệu lực, không thể thay đổi trạng thái')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khóa độc quyền trên bảng HOPDONG  //Đọc xong trả khóa |  |  |
| WAITFOR DELAY '0:0:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | SELECT @tinhTrangHD = TINHTRANG FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Đọc xong trả khóa |
|  |  | IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD)  BEGIN  PRINT(N'Không tồn tại hợp đồng cần duyệt')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Đọc xong trả khóa |
|  |  | IF NOT EXISTS (SELECT \* FROM HOPDONG  WHERE TINHTRANG = 'CHUADUYET' AND ID\_HOPDONG = @iD\_HD)  BEGIN  PRINT(N'Hợp đồng này đã được duyệt hoặc bị xóa bỏ')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khóa đọc trên bảng HOPDONG  //Đọc xong trả khóa |
|  |  | WAITFOR DELAY '0:0:10' |  |
| UPDATE HOPDONG  SET TINHTRANG = 'XOABO'  WHERE ID\_HOPDONG = @iD\_HD  PRINT N'Đã cập nhật trạng thái cho hợp đồng ' + @iD\_HD + N' từ ' + @tinhTrangHD + ' sang XOABO' | **X(HOPDONG)**  //Đang giữ khóa độc quyền trên bảng HOPDONG  //Giữ đến khi COMMIT/  ROLLBACK |  |  |
| COMMIT TRAN |  |  |  |
|  |  | IF @TINHTRANGHD = 'CHUADUYET'  BEGIN  UPDATE HOPDONG  SET TINHTRANG = 'DADUYET'  WHERE ID\_HOPDONG = @iD\_HD  PRINT N'Đã cập nhật hợp đồng ' + @iD\_HD + N' từ trạng thái CHUADUYET sang DADUYET'  END | **X(HOPDONG)**  //Đang giữ khóa độc quyền trên bảng HOPDONG  //Giữ đến khi COMMIT/  ROLLBACK |
|  |  | COMMIT TRAN |  |
| *Lưu ý: Việc xin khóa sẽ luôn thành công nếu không có bất kì lưu ý nào khác đính kèm.* | | | |

**XỬ LÍ TÌNH HUỐNG TRANH CHẤP**

1. **Sinh viên thực hiện: 20120305 – Võ Thị Kiều Khanh**

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: Conversion DeadLock**  T1 (User = Đối tác): thực hiện cập nhật số lượng chi nhánh trong Đối Tác  T2 (User = Đối tác): thực hiện cập nhật người đại diện trong Đối tác | | | |
| **sp\_Them1ChuyenXe** | **Khóa** | **sp\_ThongKeChuyenXe-CoDieuKien** | **Khóa** |
| ***Input:*** @soluong, @id\_doitac  ***Output***: Deadlock |  | ***Input***: @nguoidaidien, @id\_doitac  ***Output***: Deadlock |  |
| SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |  | SET TRANSACTION ISOLATION  LEVEL **READ COMMITED** |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra số lượng chi nhánh muốn đổi có trùng với số lượng hiện tại hay không  DECLARE @soluong\_HT INT  SET @soluong\_HT = (SELECT SL\_CHINHANH  FROM DOITAC  WHERE ID\_DOITAC = @id\_doitac )    IF (@soluong\_HT = @soluong)  BEGIN  PRINT N'SỐ LƯỢNG CHI NHÁNH MUỐN ĐỔI TRÙNG VỚI CHI NHÁNH HIỆN TẠI !'  ROLLBACK TRAN  RETURN  END | **S(DOITAC)**  //Xin khoá đọc trên DOITAC      //Trả khóa |  |  |
| WAITFOR DELAY '0:0:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra người đại diện muốn đổi có trùng với người đại diện hiện tại không.  DECLARE @nguoidaidienhientai nvarchar(30)  SET @nguoidaidienhientai = (SELECT NG\_DAIDIEN  FROM DOITAC  WHERE ID\_DOITAC = @id\_doitac)  IF (@nguoidaidienhientai = @nguoidaidienhientai)  BEGIN  PRINT N'TÊN NGƯỜI ĐẠI DIỆN MUỐN ĐỔI TRÙNG VỚI NGƯỜI ĐẠI DIỆN BÂY GIỜ '  ROLLBACK TRAN  RETURN  END | **S(DOITAC)**  //Xin khoá đọc trên DOITAC      //Trả khóa |
|  |  | WAITFOR DELAY '0:0:10' |  |
| B2: Cập nhật số lượng chi nhánh  UPDATE DOITAC  SET SL\_CHINHANH = @soluong  WHERE ID\_DOITAC = @id\_doitac | X(DOITAC)  Xin khóa ghi trên  DOITAC  (giữ khóa cho đến hết giao tác) |  |  |
| B3: Bắt lỗi hệ thống và in ra lỗi (nếu có)  BEGIN CATCH  DECLARE @ERROR VARCHAR(2000)  SELECT @ERROR = N'Lỗi: ' + ERROR\_MESSAGE()  RAISERROR(@ERROR, 16,1)  ROLLBACK TRAN  RETURN  END CATCH |  |  |  |
|  |  | B2: Cập nhật tên người đại diện  UPDATE DOITAC  SET NG\_DAIDIEN = @nguoidaidien  WHERE ID\_DOITAC = @id\_doitac | X(DOITAC)  Xin khóa ghi trên  DOI\_TAC)  //Giữ khóa đến hết giao tác |
|  |  | B3: Bắt lỗi hệ thống và in ra lỗi (nếu có)  BEGIN CATCH  DECLARE @ERROR VARCHAR(2000)  SELECT @ERROR = N'Lỗi: ' + ERROR\_MESSAGE()  RAISERROR(@ERROR, 16,1)  ROLLBACK TRAN  RETURN  END CATCH |  |
| COMMIT |  | COMMIT |  |

1. Sinh viên thực hiện: 20120295 – Ngô Võ Quang Huy
2. Tình huống 1:

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Cycle Deadlock***  T1 (User = NHANVIEN): thực hiện thêm hợp đồng mới cho đối tác, cập nhật tình trạng 0 hợp đồng cũ của đối tác.  T2 (User = DOITAC): thực hiện cập nhật người đại diện cho hợp đồng HD01 của mình, cập nhật người đại diện cho hợp đồng mới của mình | | | |
| **sp\_Them1HopDong** | **Khóa** | **sp\_CapNhatNDD** | **Khóa** |
| ***Input:*** @idHDcu = ‘HD01’,  @idHDmoi, @doitac = ‘DT01’,@mst, @ndd, @slcn, @stk,@nganhang, @ngaybd,@ngaykt, @tinhtrang  ***Output***: |  | ***Input***: ***:*** @idHDcu, @idHDmoi, @ndd  ***Output***: |  |
| SET TRAN ISOLATION LEVEL READ COMMITTED |  | SET TRAN ISOLATION LEVEL READ COMMITTED |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin @idHDmoi chưa tồn tại và @doitac đã tồn tại.    IF NOT EXISTS (SELECT \* FROM DOITAC WHERE ID\_DOITAC = @doitac)  BEGIN  PRINT N’Mã đối tác không tồn tại’  ROLLBACK TRAN  END    IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDmoi)  BEGIN  PRINT N’Mã hợp đồng mới đã tồn tại’  ROLLBACK TRAN  END    IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDcu)  BEGIN  PRINT N’Mã hợp đồng cũ không tồn tại’  ROLLBACK TRAN  END            IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDcu AND DOITAC = @doitac)  BEGIN  PRINT N'Đối tác không có hợp đồng cũ này'  ROLLBACK TRAN  END | **R(DOITAC)**  //Không cần xin khoá        **R(HOPDONG)**  //Không cần xin khoá          **R(HOPDONG)**  //Không cần xin khoá              **R(HOPDONG)**  //Không cần xin khoá |  |  |
| WAITFOR DELAY '00:00:05' |  |  |  |
|  |  | BEGIN TRAN |  |
| B2: Thêm hợp đồng mới cho đối tác    INSERT INTO HOPDONG VALUES (@idHDmoi, @mst, @ndd, @slcn, @stk,@nganhang, @ngaybd, @ngaykt, @tinhtrang, @doitac) | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG để thêm 1 hợp đồng mới là @idHDmoi | B1: Kiểm tra thông tin @idHDmoi chưa tồn tại, @doitac đã tồn tại và @idHDcu đã tồn tại.      IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDmoi)  BEGIN  PRINT N’Mã hợp đồng mới chưa tồn tại’  ROLLBACK TRAN  END    IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @idHDcu)  BEGIN  PRINT N’Mã hợp đồng cũ không tồn tại’  ROLLBACK TRAN  END | **R(HOPDONG)**  //Không cần xin khoá      **R(HOPDONG)**  //Không cần xin khoá |
| WAITFOR DELAY '00:00:05' |  | WAITFOR DELAY '00:00:05' |  |
| B3: Cập nhật tình trạng cho hợp đồng cũ    UPDATE HOPDONG  SET TINHTRANG = 0  WHERE ID\_HOPDONG = @idHDcu | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG với điều kiện ID\_HOPDONG = @idHDcu | B2: Cập nhật người đại diện hợp đồng cũ    UPDATE HOPDONG  SET NG\_DAIDIEN = @ndd  WHERE ID\_HOPDONG = @idHDcu | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG với điều kiện ID\_HOPDONG = @idHDcu. |
|  |  | WAITFOR DELAY '00:00:05' |  |
|  |  | B3: Cập nhật người đại diện hợp đồng mới    UPDATE HOPDONG  SET NG\_DAIDIEN = @ndd  WHERE ID\_HOPDONG = @idHDmoi | **X(HOPDONG)**  //Xin khoá đọc và ghi trên bảng HOPDONG với điều kiện ID\_HOPDONG = @idHDmoi |
| COMMIT |  |  |  |
|  |  | COMMIT |  |
|  |  |  |  |

1. Tình huống 2: Đối tác có MADT =’DT0001’ đang thực hiện cập nhật giá món ăn có MAMON = ‘MA0001’ nhưng chưa commit thì khách hàng vào xem thông tin danh sách món ăn của đối tác có MADT =’DT0001’

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR02: *Dirty Read***  T1 (User = DOITAC): thực hiện cập nhật giá món ăn có MAMON =’MA0001’ là 1  T2 (User = KHACHHANG): thực hiện thêm hợp đồng mới cho đối tác, cập nhật tình trạng 0 hợp đồng cũ của đối tác. | | | |
| **sp\_update\_GIA** | **Khóa** | **sp\_select\_MONAN** | **Khóa** |
| ***Input:*** @maMA = ‘MA0001’,  @gia = 1  ***Output***: |  | ***Input***: ***:*** @maDT = ‘DT0001’  ***Output***: |  |
|  |  |  |  |
| BEGIN TRAN |  |  |  |
| B1: Kiểm tra thông tin @maMA chưa tồn tại và @gia phải lớn hơn hoặc bằng 0    IF NOT EXISTS (SELECT \* FROM MONAN WHERE MAMON = @maMA)  BEGIN  PRINT N’Mã món ăn không tồn tại’  ROLLBACK TRAN  END    IF @gia < 0  BEGIN  PRINT N’Giá không hợp lệ’  ROLLBACK TRAN  END      B2: Thực hiện cập nhật giá    UPDATE MONAN  SET GIA = @gia  WHERE MAMON = @maMA | **S(MONAN)**  //xin khoá đọc trên bảng MONAN với điều kiện MAMON = @maMA                  U,X(MONAN)  //Xin khoá U,X trên  bảng MONAN với  điều kiện  MAMON = @maMA |  |  |
| COMMIT |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | B1: Kiểm tra thông tin @maDT tồn tại      IF NOT EXISTS (SELECT \* FROM DOITAC WHERE ID\_DOITAC = @maDT)  BEGIN  PRINT N‘Đối tác không tồn tại’  ROLLBACK TRAN  END    B2: Xem danh sách món ăn của đối tác    SELECT MA.MIEUTAMON, MA.LOAIMONAN, MA.GIA  FROM MONAN MA JOIN CHINHANH\_DK CN ON MA.CHINHANH = CN.ID\_CHINHANH JOIN DOITAC DT ON CN.DOITAC = DT.ID\_DOITAC  WHERE DT.ID\_DOITAC = @maDT | S(MONAN,  CHINHANH\_DK,  DOITAC)  //xin khóa đọc trên  bảng MONAN, CHINHANH\_DK, DOITAC với  điều kiện  ID\_DOITAC = @maDT |
|  |  | COMMIT |  |

III. Sinh viên thực hiện: 20120318 \_ Nguyễn Lê Mỹ Linh

1. Tình huống 1: Đối tác đang thực hiện thao tác cập nhật địa chỉ mới của một chi nhánh nhưng chưa commit thì khách hàng vào thực hiện thao tác xem thông tin danh sách chi nhánh hiện có của đối tác.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ERR01: *Dirty read***  T1 (User = Đối tác): thực hiện cập nhật địa chỉ một chi nhánh  T2 (User = Khách hàng): thực hiện xem danh sách chi nhánh của một đối tác | | | | | | |
| **sp\_CapNhatDiaChiCN** | **Khóa** | | **sp\_XemDanhSachCN** | | **Khóa** |
| ***Input:*** *@tenmoi*, *@macn, @madt*  ***Output***: |  | | ***Input***: *@madt*  ***Output***: | |  |
| SET TRANSACTION ISOLATION LEVEL READ COMMITTED |  | | SET TRANSACTION ISOLATION  LEVEL **READ COMMITTED** | |  |
| BEGIN TRAN |  |  | |  | | |
| B1: Kiểm tra thông tin chi nhánh có tồn tại không.  IF NOT EXISTS ( SELECT \*  FROM CHINHANH\_DK CN  WHERE CN.ID\_CHINHANH = @macn)  BEGIN  PRINT N'CHI NHÁNH KHÔNG TỒN TẠI'  ROLLBACK TRAN  RETURN  END | **R(CHINHANH\_DK)**  //Xin khoá đọc trên bảng CHINHANH\_DK với đk ID\_CHINHANH = @macn |  | |  | | |
| B2: Kiểm tra thông tin chi nhánh có thuộc đối tác không  IF NOT EXISTS ( SELECT \*  FROM CHINHANH\_DK CN  WHERE CN.ID\_CHINHANH = @macn AND CN.DOITAC=@madt)  BEGIN  PRINT N'KHÔNG CÓ QUYỀN THAO TÁC VÌ CHI NHÁNH NÀY KHÔNG THUỘC VỀ ĐỐI TÁC'  ROLLBACK TRAN  RETURN  END | **R(CHINHANH\_DK)**  //Xin khoá đọc trên bảng CHINHANH\_DK với đk ID\_CHINHANH = @macn AND DOITAC=@madt |  | |  | | |
| B3: Thực hiện thao tác cập nhật địa chỉ một chi nhánh  UPDATE CHINHANH\_DK  SET DIACHI = @tenmoi  WHERE ID\_CHINHANH = @macn | **U,X(CHINHANH\_DK)**  // Xin khóa cập nhật, ghi trên bảng CHINHANH\_DK với điều kiện ID\_CHINHANH = @macn |  | |  | | |
| WAITFOR DELAY '00:00:10' |  |  | |  | | |
|  |  | BEGIN TRAN | |  | | |
|  |  | B1: Kiểm tra xem đối tác có tồn tại không  IF NOT EXISTS ( SELECT \*  FROM DOITAC  WHERE ID\_DOITAC = @madt)  BEGIN  PRINT N'KHÔNG TỒN TẠI ĐỐI TÁC NÀY'  ROLLBACK TRAN  RETURN  END | | **R(DOITAC)**  //Không cần xin khoá | | |
|  |  | B2:Xem danh sách chi nhánh hiện có của đối tác  SELECT TENCHINHANH, DIACHI, SDT, TINHTRANG, THOIGIANMO, THOIGIANDONG  FROM CHINHANH\_DK  WHERE DOITAC = @madt | | **R(CHINHANH\_DK)**  //Không cần xin khoá | | |
|  |  | COMMIT | |  | | |
| B4: Kiểm tra địa chỉ mới cập nhật liệu có thuộc về một chi nhánh nào khác thuộc bảng CHINHANH\_DK không  IF (SELECT COUNT(\*)  FROM CHINHANH\_DK  WHERE DIACHI = @tenmoi) > 1  BEGIN  PRINT N'ĐỊA CHỈ MỚI KHÔNG HỢP LỆ VÌ ĐÃ TỒN TẠI ĐỊA CHỈ NÀY, KHÔNG THỂ CÓ NHIỀU HƠN MỘT CHI NHÁNH CHO MỘT ĐỊA CHỈ'  ROLLBACK TRAN  RETURN  END |  |  | |  | | |
| COMMIT |  |  | |  | | |

2. Tình huống 2: Khách hàng thực hiện thao tác tìm các món ăn thuộc một loại món ăn cho trước ở một chi nhánh nhưng đồng thời bên đối tác thực hiện thao tác cập nhật tên gọi mới cho loại món ăn này ở chi nhánh trên.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ERR02: *Unrepeatable Read***  T1 (User = Khách hàng): thực hiện tìm các món ăn thuộc loại món ăn cho trước ở một chi nhánh  T2 (User = Đối tác): thực hiện cập nhật tên gọi mới của loại món ăn này | | | | | | |
| **sp\_Them1ChuyenXe** | **Khóa** | **sp\_ThongKeChuyenXe-CoDieuKien** | | **Khóa** | |
| ***Input:*** *@tenloai, @macn*  ***Output***: |  | ***Input:*** *@tencu, @tenmoi, @macn*  ***Output***: | |  | |
| SET TRANSACTION ISOLATION LEVEL **REPEATABLE READ** |  | SET TRAN ISOLATION LEVEL READ COMMITTED | |  | |
| BEGIN TRAN |  | |  | |  | |
| B1: Kiểm tra thông tin loại món ăn ở chi nhánh có tồn tại không  IF NOT EXISTS ( SELECT \*  FROM MONAN  WHERE LOAIMONAN = @tenloai AND CHINHANH = @macn)  BEGIN  PRINT N'KHÔNG CÓ TÌM KIẾM PHÙ HỢP'  ROLLBACK TRAN  RETURN  END | **R(MONAN)**  //Xin khoá đọc trên bảng MONAN với đk LOAIMONAN = @tenloai and CHINHANH = @macn | |  | |  | |
| WAITFOR DELAY '00:00:10' |  | |  | |  | |
|  |  | | BEGIN TRAN | |  | |
|  |  | | B1: Kiểm tra thông tin tên loại món ăn cũ ở chi nhánh có tồn tại không  IF NOT EXISTS ( SELECT \*  FROM MONAN  WHERE LOAIMONAN = @tencu AND CHINHANH = @macn)  BEGIN  PRINT N'KHÔNG CÓ TÌM KIẾM PHÙ HỢP'  ROLLBACK TRAN  RETURN  END | | **R(MONAN)**  //Xin khoá đọc trên bảng MONAN với đk LOAIMONAN = @tencu CHINHANH = @macn | |
|  |  | | B2: Thực hiện cập nhật lại tên mới cho loại món ăn mong muốn  UPDATE MONAN  SET LOAIMONAN = @tenmoi  WHERE LOAIMONAN = @tencu AND CHINHANH = @macn | | **U,X(MONAN)**  //Xin khóa cập nhật, ghi cho trên bảng MONAN với dk LOAIMONAN = @tencu CHINHANH = @macn | |
|  |  | | COMMIT | |  | |
| B2: Thực hiện in ra danh sách món ăn cần tìm  SELECT \*  FROM MONAN  WHERE LOAIMONAN = @tenloai AND CHINHANH = @macn | **R(MONAN)**  //Xin khoá đọc trên bảng MONAN với đk LOAIMONAN = @tenloai and CHINHANH = @macn | |  | |  | |
| COMMIT |  | |  | |  | |

**d**

IV. Sinh viên thực hiện: 20120255-Phạm Mai Thiên Bảo

1. Tình huống 1: Nhân viên đang liệt kê danh sách hợp đồng chưa được duyệt thì đối tác thêm một hợp đồng mới.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ERR01: Phantom Read**  T1 (User = nhân viên): liệt kê hợp đồng chưa duyệt của đối tác A.  T2 (User = đối tác): thêm mới một hợp đồng. | | | | | | |
| **usp\_Liet\_Ke\_HD\_Chua\_Duyet** | **Khóa** | **Usp\_Them\_Hop\_Dong** | | **Khóa** | |
| ***Input:*** Mã đối tác (@iD\_DoiTac)  ***Output***: Danh sách hợp đồng chưa duyệt và tổng số hợp đồng chưa duyệt. |  | ***Input***: Thông tin hợp đồng (mã hợp đồng, mã số thuế,…)  ***Output***: Thêm một hợp đồng và xuất thông báo thêm hợp đồng thành công. Nếu thêm không thành công thì xuất thông báo lỗi | |  | |
| SET TRANSACTION ISOLATION  LEVEL …SERIALIZABLE….. |  | SET TRANSACTION ISOLATION  LEVEL …………………………… | |  | |
| BEGIN TRAN |  | |  | |  | |
| B1: Kiểm tra mã đối tác có tồn tại?  IF EXISTS (SELECT \* FROM DOITAC WHERE ID\_DOITAC = @iD\_DoiTac)  BEGIN  PRINT(N'Đối tác không tồn tại')  ROLLBACK TRAN  END | **R(DOITAC)**  //Xin khoá đọc trên bảng A với điều kiện ID\_DOITAC = @iD\_DoiTac  //Giữ đến hết giao tác | |  | |  | |
| B2: Kiểm tra đối tác đã có hợp đồng?  IF EXISTS (SELECT \* FROM HOPDONG WHERE DOITAC = @iD\_DoiTac)  BEGIN  PRINT(N'Đối tác không có hợp đồng hợp lệ')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac  //Giữ đến hết giao tác | |  | |  | |
| B3: Kiểm tra đối tác có hợp đồng chưa duyệt?  IF EXISTS (SELECT \* FROM HOPDONG WHERE DOITAC = @iD\_DoiTac AND TINHTRANG = 'CHUADUYET')  BEGIN  PRINT(N'Đối tác không có hợp đồng chưa được duyệt')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac và TINHTRANG = 'CHUADUYET'  //Giữ đến hết giao tác | |  | |  | |
| B4: Đọc số lượng hợp đồng.  SELECT @soLuongHD = COUNT(\*) FROM HOPDONG WHERE DOITAC = @iD\_DoiTac | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac  //Giữ đến hết giao tác | |  | |  | |
| WAITFOR DELAY '0:0:10' |  | |  | |  | |
|  |  | | BEGIN TRAN | |  | |
|  |  | | B1: Kiểm tra mã hợp đồng đã tồn tại?  IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD)  BEGIN  PRINT(N'Mã hợp đồng đã tồn tại')  ROLLBACK TRAN  END | | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD  //Đọc xong trả khóa | |
|  |  | | B2: Thêm hợp đồng mới.  INSERT INTO HOPDONG  VALUES(@iD,@mST,@ngDaiDien,@sLChiNhanh,@soTaiKhoan,@nganHang,@ngBD,@ngKT,'CHUADUYET',@doiTac) | | **X(HOPDONG)**  //Xin khóa độc quyền trên bảng HOPDONG  **//Do T1 đang giữ khóa đọc trên đơn vị dữ liệu này nên T2 không thể ghi**  **//CHỜ…** | |
| B5: Liệt kê hợp đồng.  DECLARE CURS CURSOR FOR  SELECT ID\_HOPDONG FROM HOPDONG WHERE DOITAC = @iD\_DoiTac  OPEN CURS  FETCH NEXT FROM CURS INTO @iD\_HD  WHILE @@FETCH\_STATUS = 0  BEGIN  PRINT @iD\_HD  FETCH NEXT FROM CURS INTO @iD\_HD  WAITFOR DELAY '0:0:5'  END  CLOSE CURS  DEALLOCATE CURS  PRINT N'Tổng số hợp đồng chưa duyệt của đối tác ' + @iD\_DoiTac + ': ' + CAST (@soLuongHD AS NVARCHAR) | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện DOITAC = @iD\_DoiTac  //Giữ đến hết giao tác | |  | |  | |
| COMMIT TRAN |  | |  | |  | |
|  |  | | B2: Thêm hợp đồng mới.  INSERT INTO HOPDONG  VALUES(@iD,@mST,@ngDaiDien,@sLChiNhanh,@soTaiKhoan,@nganHang,@ngBD,@ngKT,'CHUADUYET',@doiTac) | | **X(HOPDONG)**  //Xin khóa độc quyền trên bảng HOPDONG  //Giữ đến hết giao tác | |
|  |  | | COMMIT TRAN | |  | |
| *Lưu ý: Việc xin khóa sẽ luôn thành công nếu không có bất kì lưu ý nào khác đính kèm.* | | | | | | |

2. Tình huống 2: Hai nhân viên cùng đang duyệt một hợp đồng và đưa ra 2 kết quả tình trạng hợp đồng khác nhau, nhân viên 1 cập nhật làm ảnh hưởng đến kết quả cập nhật của nhân viên 2

|  |  |  |  |
| --- | --- | --- | --- |
| **ERR01: *Lost Update***  T1 (User = nhân viên): thực hiện cập nhật tình trạng của một hợp đồng A thành ‘XOABO’.  T2 (User = nhân viên): thực hiện cập nhật tình trạng của một hợp đồng A từ ‘CHUADUYET’ thành ‘DADUYET’ (chỉ hợp đồng chưa được duyệt mới được cập nhật sang được duyệt, trường hợp duy nhất). | | | |
| **sp\_Them1ChuyenXe** | **Khóa** | **sp\_ThongKeChuyenXe-CoDieuKien** | **Khóa** |
| ***Input:*** Mã hợp đồng (@iD\_HD)  ***Output***: Cập nhật tình trạng hợp đồng và xuất thông báo. |  | ***Input***: Mã hợp đồng (@iD\_HD)  ***Output***: Cập nhật tình trạng hợp đồng và xuất thông báo. |  |
| SET TRANSACTION ISOLATION  LEVEL …SERIALIZABLE….. |  | SET TRANSACTION ISOLATION  LEVEL …SERIALIZABLE….. |  |
| BEGIN TRAN |  |  |  |
| SELECT @tinhTrangHD = TINHTRANG FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Giữ đến hết giao tác |  |  |
| IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD)  BEGIN  PRINT(N'Không tồn tại hợp đồng cần xóa')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Giữ đến hết giao tác |  |  |
| IF EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD AND TINHTRANG = 'XOABO')  BEGIN  PRINT(N'Hợp đồng đã ở trạng thái này từ trước')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD và TINHTRANG = ‘XOABO’  //Giữ đến hết giao tác |  |  |
| IF EXISTS (SELECT \* FROM HOPDONG ***WITH (XLOCK)*** WHERE ID\_HOPDONG = @iD\_HD AND DATEDIFF(DAY,NGAYKETTHUCDK, GETDATE()) < 0 AND TINHTRANG = 'DADUYET')  BEGIN  PRINT(N'Hợp đồng đang còn hiệu lực, không thể thay đổi trạng thái')  ROLLBACK TRAN  END | **X(HOPDONG)**  //Xin khóa độc quyền trên bảng HOPDONG  //Giữ đến khi COMMIT/  ROLLBACK |  |  |
| WAITFOR DELAY '0:0:10' |  |  |  |
|  |  | BEGIN TRAN |  |
|  |  | SELECT @tinhTrangHD = TINHTRANG FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  ***//Xin không thành công do T1 đang giữ khóa độc quyền trên bảng HOPDONG***  ***=> CHỜ…*** |
| UPDATE HOPDONG  SET TINHTRANG = 'XOABO'  WHERE ID\_HOPDONG = @iD\_HD  PRINT N'Đã cập nhật trạng thái cho hợp đồng ' + @iD\_HD + N' từ ' + @tinhTrangHD + ' sang XOABO' | **X(HOPDONG)**  //Đang giữ khóa độc quyền trên bảng HOPDONG  //Giữ đến khi COMMIT/  ROLLBACK |  |  |
| COMMIT TRAN |  |  |  |
|  |  | SELECT @tinhTrangHD = TINHTRANG FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Giữ đến hết giao tác |
|  |  | IF NOT EXISTS (SELECT \* FROM HOPDONG WHERE ID\_HOPDONG = @iD\_HD)  BEGIN  PRINT(N'Không tồn tại hợp đồng cần duyệt')  ROLLBACK TRAN  END | **R(HOPDONG)**  //Xin khoá đọc trên bảng HOPDONG với điều kiện ID\_HOPDONG = @iD\_HD  //Giữ đến hết giao tác |
|  |  | IF NOT EXISTS (SELECT \* FROM HOPDONG ***WITH (XLOCK)*** WHERE TINHTRANG = 'CHUADUYET' AND ID\_HOPDONG = @iD\_HD)  BEGIN  PRINT(N'Hợp đồng này đã được duyệt hoặc bị xóa bỏ')  ROLLBACK TRAN  END | **X(HOPDONG)**  //Xin khóa độc quyền trên bảng HOPDONG  //Giữ đến khi COMMIT/  ROLLBACK |
|  |  | WAITFOR DELAY '0:0:10' |  |
|  |  | IF @TINHTRANGHD = 'CHUADUYET'  BEGIN  UPDATE HOPDONG  SET TINHTRANG = 'DADUYET'  WHERE ID\_HOPDONG = @iD\_HD  PRINT N'Đã cập nhật hợp đồng ' + @iD\_HD + N' từ trạng thái CHUADUYET sang DADUYET'  END | **X(HOPDONG)**  //Đang giữ khóa độc quyền trên bảng HOPDONG  //Giữ đến khi COMMIT/  ROLLBACK |
|  |  | COMMIT TRAN |  |
| *Lưu ý: Việc xin khóa sẽ luôn thành công nếu không có bất kì lưu ý nào khác đính kèm.* | | | |

# TÀI LIỆU THAM KHẢO

- Tài liệu lý thuyết liên quan đến thuật toán được đăng tải trên moddle.

- Tham khảo bên ngoài:

* Anthony Molinaro, SQL Cookbook
* Allen G. Taylor, SQL All-in-One for Dummies
* Using Isolation Level In SQL Transaction

<https://www.c-sharpcorner.com/blogs/using-isolation-level-in-sql-transaction2>

* Isolation Levels in SQL Server, <https://www.sqlservercentral.com/articles/isolation-levels-in-sql-server>
* Explain about concurrent transactions in DBMS, https://www.tutorialspoint.com/