TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

**KHOA HỆ THỐNG THÔNG TIN**

--------------------------------



**BÁO CÁO THỰC TẬP DOANH NGHIỆP**

**-----------------------**

SV thực hiện: Nguyễn Ngọc Công

MSSV: 15520069

Lớp: HTTT2015

Người hướng dẫn: Nguyễn Hồ Duy Trí

Thành phố Hồ Chí Minh – Năm 2019 tháng ngày

# LỜI CẢM ƠN

Trân trọng gửi lời cảm ơn đến công ty Global CyberSoft đã tạo cơ hội, điều kiện và giúp đỡ tôi rất nhiều trong thời gian thực tập tại công ty.

Đặc biệt cảm ơn anh Đạt (Quản lý dự án - PM) người đã dẫn dắt để làm quen với những hoạt động, về việc lên kế hoạch, phân chia và cách để hoàn thành công việc; anh Thông (Kiến trúc sư kĩ thuật -TA & leader) người thầy dạy cho tôi rất nhiều về lập trình, giải thuật, công nghệ, tất tần tật mọi thứ về sản phẩm của khách hàng. Gửi lời cảm ơn chân thành đến mọi người trong team RAILMAP-POC đã hướng dẫn, chỉ bảo, cùng nhau làm việc tạo nên một môi trường để tôi được trải nghiệm, học tập và phát triển bản thân.

Xin cảm ơn GVHD Nguyễn Hồ Duy Trí cũng như các thầy cô trong khoa Hệ Thống Thông Tin đã nhiệt tình hỗ trợ để em hoàn thành báo cáo này.

Nơi đây đặt những bước chân đầu tiên của em trên con đường sự nghiệp của mình nên thời gian làm việc ở đây em cảm thấy rất ý nghĩa, học tập được rất nhiều kiến thức bổ ích và cũng tiếp thêm động lực để em hoàn thành chương trình học tập chuẩn bị cho những dự định của bản thân trong tương lai.

Nguyễn Ngọc Công

# NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc43799260)

[NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN 3](#_Toc43799261)

[MỤC LỤC 4](#_Toc43799262)

[Danh mục hình ảnh 6](#_Toc43799263)

[Danh mục bảng 8](#_Toc43799264)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 9](#_Toc43799265)

[1.1 Đặt vấn đề 9](#_Toc43799266)

[1.2 Mục tiêu 9](#_Toc43799267)

[1.3 Công cụ sử dụng 9](#_Toc43799268)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ DỮ LIỆU 10](#_Toc43799269)

[2.1 Đặc tả bài toán 10](#_Toc43799270)

[2.1 Phân tích yêu cầu 10](#_Toc43799271)

[2.1.1 Yêu cầu phi chức năng 10](#_Toc43799272)

[2.1.2 Thiết kế mô hình quan hệ 10](#_Toc43799273)

[CHƯƠNG 3: CÁC CÔNG NGHỆ SỬ DỤNG 12](#_Toc43799274)

[3.1 Oracle 12](#_Toc43799275)

[3.2 JDBC 12](#_Toc43799276)

[3.3 Java Spring 12](#_Toc43799277)

[3.4 Hibernate 12](#_Toc43799278)

[3.5 Git 12](#_Toc43799279)

[3.6 IteliJ 12](#_Toc43799280)

[CHƯƠNG 4: XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC 13](#_Toc43799281)

[4.1 Trigger 13](#_Toc43799282)

[4.1.1 Trigger trong Oracle 13](#_Toc43799283)

[4.3.3 Danh sách các Trigger 13](#_Toc43799284)

[4.4.4 Mô tả một số Trigger 13](#_Toc43799285)

[4.2 Store Procedure 13](#_Toc43799286)

[4.2.1 Store procedure trong Oracle 13](#_Toc43799287)

[4.2.2 Transaction trong Oracle 13](#_Toc43799288)

[4.2.3 Danh sách Store Procedure 13](#_Toc43799289)

[4.2.4 Mô tả một số Store Procedure 13](#_Toc43799290)

[CHƯƠNG 5: XỬ LÝ TRUY XUẤT ĐỒNG THỜI 14](#_Toc43799291)

[5.1 Các mức cô lập trong Oracle 14](#_Toc43799292)

[5.1.1 Read committed 14](#_Toc43799293)

[5.1.2 Serializable 14](#_Toc43799294)

[5.1.3 Read-only 14](#_Toc43799295)

[5.2 Cơ chế khóa 14](#_Toc43799296)

[5.3 Deadlock 14](#_Toc43799297)

[5.4 Mô tả trong đồ án môn học 14](#_Toc43799298)

[5.4.1 Lost update 14](#_Toc43799299)

[5.4.2 Non-repeatable read 14](#_Toc43799300)

[5.4.3 Phantom read 14](#_Toc43799301)

[CHƯƠNG 6: THIẾT KẾ GIAO DIỆN 15](#_Toc43799302)

[6.1 Danh sách các màn hình 15](#_Toc43799303)

[6.1.1 Đăng nhập 15](#_Toc43799304)

[6.1.2 Màn hình trang chủ 15](#_Toc43799305)

[6.1.3 Category 15](#_Toc43799306)

[6.1.4 Product 15](#_Toc43799307)

[6.1.5 Nhập hàng 15](#_Toc43799308)

[6.1.6 Xuất hang 15](#_Toc43799309)

[6.1.7 Kiểm tra hàng tồn kho 15](#_Toc43799310)

[6.2 Mô tả các màn hình 15](#_Toc43799311)

[6.21 Đăng nhập 15](#_Toc43799312)

[6.2.2 Màn hình trang chủ 15](#_Toc43799313)

[6.2.3 Category 15](#_Toc43799314)

[6.2.4 Product 15](#_Toc43799315)

[6.2.5 Nhập hàng 15](#_Toc43799316)

[6.2.6 Xuất hang 15](#_Toc43799317)

[6.2.7 Kiểm tra hàng tồn kho 15](#_Toc43799318)

[CHƯƠNG 7: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 16](#_Toc43799319)

[6.1 Kết quả đặt được 16](#_Toc43799320)

[6.2 Hạn chế 16](#_Toc43799321)

[6.3 Hướng phát triển 16](#_Toc43799322)

[6.4 Tài liệu tham khảo 16](#_Toc43799323)

# Danh mục hình ảnh

# Danh mục bảng

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1 Đặt vấn đề

Nội dung ở phần này viết ở đây

## 1.2 Mục tiêu

Mục tiêu của đồ án

## 1.3 Công cụ sử dụng

Mục tiêu của đồ án

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ DỮ LIỆU

## 2.1 Đặc tả bài toán

Nội dung ở phần này viết ở đây

## 2.1 Phân tích yêu cầu

### 2.1.1 Yêu cầu phi chức năng

Nội dung ở phần này viết ở đây

### 2.1.2 Thiết kế mô hình quan hệ

#### a, Mô hình thực thể mối kết hợp

Nội dung ở phần này viết ở đây

#### b, Thiết kế dữ liệu

Nội dung ở phần này viết ở đây

#### c, Mô tả thành phần dữ liệu

Nội dung ở phần này viết ở đây

#### d, Mô tả ràng buộc toàn vẹn

Nội dung ở phần này viết ở đây

# CHƯƠNG 3: CÁC CÔNG NGHỆ SỬ DỤNG

## 3.1 Oracle

Nội dung ở phần này viết ở đây

## 3.2 JDBC

Nội dung ở phần này viết ở đây

## 3.3 Java Spring

Nội dung ở phần này viết ở đây

## 3.4 Hibernate

Nội dung ở phần này viết ở đây

## 3.5 Git

Nội dung ở phần này viết ở đây

## 3.6 IteliJ

Nội dung ở phần này viết ở đây

# CHƯƠNG 4: XÂY DỰNG VÀ QUẢN LÝ GIAO TÁC

## 4.1 Trigger

### 4.1.1 Trigger trong Oracle

Trigger là một đơn vị chương trình lưu trữ trong database và thực thi (fire) để đáp ứng một sự kiện nào đó.

Sự kiện này được kết hợp với một table, view, schema, hoặc database, và là một trong những sự kiện sau:

* Một câu lệnh DML (DELETE, INSERT, hoặc UPDATE).
* Một câu lệnh DDL (CREATE, ALTER, DROP)
* Một tác vụ trên database (SERVERERROR, LOGON, LOGOFF, STARTUP, hoặc SHUTDOWN).

Cú pháp tạo Trigger:

|  |
| --- |
| CREATE [OR RELACE] TRIGGER trigger\_name              {BEFORE | AFTER}              {DELETE, INSERT, UPDATE [OF column\_name….]}        ON table\_name              [REFERENCING {OLD AS old, NEW AS new}]              [FOR EACH ROW [WHEN condition]]              DECLARE                          Variable declaration;                          Constant declaration;   BEGIN               PL/SQL subprogram body;              [EXCEPTION                          exception PL/SQL block;   END; |

Giải thích:

* CREATE [OR REPLACE ] TRIGGER trigger\_name: tạo hoặc thay thế một trigger đã tồn tại thành trigger\_name.
* {BEFORE | AFTER }: cái này chỉ định khi nào trigger được thực thi.
* {DELETE, INSERT, UPDATE [OF column\_name….]}chỉ định cụ thể lệnh nào thuộc loại lệnh DML được thực thi. [OF col\_name]: chỉ định cụ thể cột nào sẽ được cập nhật.
* ON table\_name: chỉ định trigger sẽ được thực thi trên bảng nào.
* [REFERENCING {OLD AS old, NEW AS new}]: cho phép dùng giá trị new hay old cho các lệnh DML, như là Insert, Update, Delete.
* [FOR EACH ROW [WHEN condition]]: cho biết cụ thể trigger sẽ thực thi trên từng dòng. WHEN condition: cho biết dòng đó có điều kiện gì thì trigger mới thực thi được.

### 4.3.2 Danh sách các Trigger

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên | Thao tác | Bảng | Nội dung | Ghi chú |
| 1 | update\_quantity | INSERT | INVOICE\_DETAIL | Câp nhập số hàng tồn kho khi thêm một hóa đơn nhập hàng hoặc hóa đơn xuất hàng bất kì. |  |
| 2 | check\_quanity\_in\_stock | INSERT, UPDATE | PRODUCT\_IN\_STOCK | Kiểm tra số hàng tồn kho khi thêm mới hoặc sửa một dòng có vượt quá 150. |  |
|  | validator\_email | INSERT, UPDATE | USERS | Kiểm tra Email khi thêm mới hoặc sửa thông tin một User có đúng chuẩn hay chưa. |  |
| 4 | update\_price\_on\_invoice | INSERT, UPDATE, DELTE | INVOICE\_DETAIL | Cập nhập tổng số tiền của hóa đơn khi thêm, sửa hoặc xóa một dòng invoice\_detail |  |
| 5 |  |  |  |  |  |

### 4.4.3 Mô tả một số Trigger

Tên: update\_quantity

Thao tác: INSERT

Trên bảng: INVOICE\_DETAIL

Mã PL/SQL:

|  |
| --- |
| CREATE OR REPLACE TRIGGER update\_quantity  BEFORE INSERT ON INVOICE\_DETAIL  FOR EACH ROW  DECLARE  v\_type INVOICE.TYPE%TYPE;  v\_product\_id PRODUCT\_IN\_STOCK.PRODUCT\_ID%TYPE;  BEGIN  SELECT TYPE INTO v\_type  FROM INVOICE  WHERE INVOICE\_ID = :NEW.INVOICE\_ID AND ACTIVE\_FLAG = 1;    SELECT PRODUCT\_ID INTO v\_product\_id  FROM PRODUCT\_IN\_STOCK  WHERE PRODUCT\_ID = :NEW.PRODUCT\_ID;    IF v\_type = 1 THEN  --NHAP HANG  UPDATE PRODUCT\_IN\_STOCK  SET QUANITY = QUANITY + :NEW.QUANITY  WHERE PRODUCT\_ID = :NEW.PRODUCT\_ID AND ACTIVE\_FLAG = 1;  ELSE  --XUAT HANG  UPDATE PRODUCT\_IN\_STOCK  SET QUANITY = QUANITY - :NEW.QUANITY  WHERE PRODUCT\_ID = :NEW.PRODUCT\_ID AND ACTIVE\_FLAG = 1;  END IF;    EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('INSERT NEW PRODUCT\_IN\_STOCK!');  INSERT INTO PRODUCT\_IN\_STOCK(PRODUCT\_ID, QUANITY) VALUES (:NEW.PRODUCT\_ID, :NEW.QUANITY);    WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('OTHERS EXCEPTION!');  END; |

[1]: Kiểm tra email của User khi insert hay update vào bảng USERS có đúng định dạng hay chưa

[1.1]: Nếu đúng thì không làm gì cả

[1.2]: Nếu sai thì raise ra lỗi và dòng không được thêm hay cập nhập vào bảng.

## 4.2 Store Procedure

### 4.2.1 Store procedure trong Oracle

### 4.2.2 Transaction trong Oracle

#### a, Định nghĩa

#### b, Cấu trúc Transaction

#### c, Transaction control

### 4.2.3 Danh sách Store Procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên | Tham số đầu vào | Tham số đầu ra | Ý nghĩa | Ghi chú |
| 1 | get\_info\_user | v\_id\_user USERS.USER\_ID%TYPE |  | Lấy và in ra thông tin của User với user\_id |  |
| 2 | get\_product\_price | v\_product\_id PRODUCT\_INFO.PRODUCT\_INFO\_ID%TYPE | PRODUCT\_INFO.PRICE%TYPE | Lấy giá tiền của sản phẩm với product\_info\_id |  |
| 3 | check\_invoice\_price | v\_invoice\_id INVOICE.INVOICE\_ID%TYPE | NUMBER | Kiểm tra tổng tiền trong hóa đơn có bằng tổng tiền của các chi tiết hóa đơn |  |
| 4 | create\_category | (v\_cate\_id NUMBER, v\_name VARCHAR2, v\_description VARCHAR2, v\_code VARCHAR2 |  | Nếu tham số đầu vào là một category đã có trong CSDL thì ta cập nhập giá trị, ngược lại nếu là category chưa có trong CLDL thì ta thêm vào. |  |
| 5 | get\_all\_menu |  |  |  |  |
| 6 | check\_email |  |  |  |  |
| 7 | get\_quantity\_product\_in\_stock |  |  | Lấy ra quantity hiện tại của một Product |  |
| 8 | get\_amount | Table of an Obejct with 2 properties (1\*) | NUMBER | Lấy tổng số tiền của một danh sách các mặt hàng sắp được thêm vào hóa đơn |  |
| 9 | add\_goods\_receipt\_invoice | (1\*), v\_type NUMBER, v\_user\_id NUMBER, v\_code VARCHAR2 |  | Với đầu vào là một danh sách mặt hàng để thêm vào invoice. Khi thêm invoice thì procedure này sẽ thêm danh sách invoice\_detail tương ứng. |  |

### 4.2.4 Mô tả một số Store Procedure

# CHƯƠNG 5: XỬ LÝ TRUY XUẤT ĐỒNG THỜI

## 5.1 Các mức cô lập trong Oracle

### 5.1.1 Read committed

### 5.1.2 Serializable

### 5.1.3 Read-only

## 5.2 Cơ chế khóa

## 5.3 Deadlock

## 5.4 Mô tả trong đồ án môn học

### 5.4.1 Non-repeatable read

Khái niệm: Một transaction đọc lại dữ liệu mà nó đã đọc trước đó và thấy rằng một transaction khác đã commit dữ liệu đã được thay đổi hoặc xóa. Ví dụ: người dùng truy vấn một hàng và sau đó truy vấn cùng một hàng, chỉ để phát hiện ra rằng dữ liệu đã thay đổi.

Ví dụ, một giao dịch truy vấn số lượng nhân viên. Năm phút sau thực hiện cùng một truy vấn, nhưng bây giờ số lượng đã tăng lên một vì một người dùng khác đã thêm một nhân viên mới.

a. Mô tả tình huống: Nhân viên thứ nhất thêm một hóa đơn nhập hàng cho mặt hàng với product\_id là 21. Sau khi thêm xong nhân viên 1 lấy ra danh sách sản phẩm trong kho và kiểm tra số lượng mặt hàng mới thêm vào. Trong lúc đó thì nhân viên thứ 2 đang thêm một hóa đơn xuất hàng cũng cho mặt hàng với product\_id là 21, giao dịch thành công. Cuối cùng nhân viên 1 lấy ra danh sách sản phẩm trong kho và kiểm tra số lượng mặt hàng với product\_id là 21 thì thấy dữ liệu đã khác so với ban đầu.

b. Mã PL/SQL:

Function: get\_amount()

|  |
| --- |
| CREATE OR REPLACE TYPE info\_goods\_receipt AS OBJECT (v\_product\_id NUMBER(5,0), v\_quantity NUMBER(5,0));  CREATE OR REPLACE TYPE list\_info\_goods\_receipt IS TABLE OF info\_goods\_receipt;  CREATE OR REPLACE FUNCTION get\_amount(v\_l\_info\_gr list\_info\_goods\_receipt)  RETURN NUMBER  AS  v\_amount NUMBER;  v\_price PRODUCT\_INFO.PRICE%TYPE;  BEGIN  v\_amount := 0;  FOR ind IN v\_l\_info\_gr.FIRST..v\_l\_info\_gr.LAST  LOOP  --GET PRICE FROM PRODUCT ID  SELECT PRICE INTO v\_price  FROM PRODUCT\_INFO  WHERE PRODUCT\_INFO\_ID = v\_l\_info\_gr(ind).v\_product\_id;  v\_amount := v\_amount + v\_l\_info\_gr(ind).v\_quantity \* v\_price;  END LOOP;    RETURN v\_amount;    EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('NO\_DATA\_FOUND!');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('OTHERS EXCEPTION!');  END; |

Store procedure: add\_goods\_receipt\_invoice()

|  |
| --- |
| CREATE OR REPLACE PROCEDURE add\_goods\_receipt\_invoice(v\_l\_info\_gr list\_info\_goods\_receipt,  v\_invoice\_id NUMBER, v\_type NUMBER, v\_user\_id NUMBER, v\_code VARCHAR2)  AS  v\_mount NUMBER;  BEGIN  --INSERT INTO INVOICE  v\_mount := get\_amount(v\_l\_info\_gr);  INSERT INTO INVOICE(INVOICE\_ID, TYPE, USER\_ID, PRICE, CODE) VALUES(v\_invoice\_id, v\_type, v\_user\_id, v\_mount, v\_code);  FOR ind IN v\_l\_info\_gr.FIRST..v\_l\_info\_gr.LAST  LOOP  --INSERT INTO INVOICE DETAIL  INSERT INTO INVOICE\_DETAIL(INVOICE\_ID, PRODUCT\_ID, QUANITY)  VALUES (v\_invoice\_id, v\_l\_info\_gr(ind).v\_product\_id, v\_l\_info\_gr(ind).v\_quantity);    END LOOP;  END; |

Trigger: update\_quantity

|  |
| --- |
| CREATE OR REPLACE TRIGGER update\_quantity  BEFORE INSERT ON INVOICE\_DETAIL  FOR EACH ROW  DECLARE  v\_type INVOICE.TYPE%TYPE;  v\_product\_id PRODUCT\_IN\_STOCK.PRODUCT\_ID%TYPE;  BEGIN  SELECT TYPE INTO v\_type  FROM INVOICE  WHERE INVOICE\_ID = :NEW.INVOICE\_ID AND ACTIVE\_FLAG = 1;    SELECT PRODUCT\_ID INTO v\_product\_id  FROM PRODUCT\_IN\_STOCK  WHERE PRODUCT\_ID = :NEW.PRODUCT\_ID;    IF v\_type = 1 THEN  --NHAP HANG  UPDATE PRODUCT\_IN\_STOCK  SET QUANITY = QUANITY + :NEW.QUANITY  WHERE PRODUCT\_ID = :NEW.PRODUCT\_ID AND ACTIVE\_FLAG = 1;  ELSE  --XUAT HANG  UPDATE PRODUCT\_IN\_STOCK  SET QUANITY = QUANITY - :NEW.QUANITY  WHERE PRODUCT\_ID = :NEW.PRODUCT\_ID AND ACTIVE\_FLAG = 1;  END IF;    EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('INSERT NEW PRODUCT\_IN\_STOCK!');  INSERT INTO PRODUCT\_IN\_STOCK(PRODUCT\_ID, QUANITY) VALUES (:NEW.PRODUCT\_ID, :NEW.QUANITY);    WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('OTHERS EXCEPTION!');  END; |

c. Mô tả

|  |  |  |
| --- | --- | --- |
| Session 1 | Session 2 | Explanation |
| SELECT PRODUCT\_STOCK\_ID,PRODUCT\_ID, QUANITY  FROM PRODUCT\_IN\_STOCK  WHERE ACTIVE\_FLAG = 1;  PRODUCT\_STOCK\_ID PRODUCT\_ID QUANITY  ---------------- ---------- ----------  61 21 75  81 22 5 |  | Giá trị Quantity ban đầu của sản phẩm với PRODUCT\_ID = 21 là 75 |
| DECLARE  v\_l\_info\_gr list\_info\_goods\_receipt := list\_info\_goods\_receipt(  info\_goods\_receipt( 21, 15)  );  v\_type NUMBER;  v\_user\_id NUMBER;  v\_code VARCHAR2(30);  v\_invoice\_id NUMBER;  BEGIN  v\_type := 1;  v\_user\_id := 1;  v\_code := 'HD015';  v\_invoice\_id := 38;    add\_goods\_receipt\_invoice(v\_l\_info\_gr, v\_invoice\_id, v\_type, v\_user\_id, v\_code);  END;  PL/SQL procedure successfully completed. |  | Session 1 tương ứng với nhân viên thứ nhất thêm mới một hóa đơn nhập hàng với product\_id = 21 và  quantity = 15 sử dụng  add\_goods\_receipt\_invoice() |
| SELECT PRODUCT\_STOCK\_ID,PRODUCT\_ID, QUANITY  FROM PRODUCT\_IN\_STOCK  WHERE ACTIVE\_FLAG = 1;  PRODUCT\_STOCK\_ID PRODUCT\_ID QUANITY  ---------------- ---------- ----------  61 21 90  81 22 5 |  | Nhân viên thứ nhất lấy ra danh sách hàng tồn kho và kiểm tra số lượng của sản phẩm với product\_id = 21 lúc này có giá trị là 90. |
|  | SET TRANSACTION  ISOLATION LEVEL  READ COMMITTED;  Transaction ISOLATION succeeded. | Session 2 thiết lập mức cô lập là read committed. |
|  | SELECT PRODUCT\_STOCK\_ID,PRODUCT\_ID, QUANITY  FROM PRODUCT\_IN\_STOCK  WHERE ACTIVE\_FLAG = 1;  PRODUCT\_STOCK\_ID PRODUCT\_ID QUANITY  ---------------- ---------- ----------  61 21 75  81 22 5 | Session 2 lấy ra danh sách hàng tồn kho, kiểm tra giá trị Quantity của sản phẩm với product\_id lúc này vẫn là 75 vì Session 1 chưa COMMIT |
|  | DECLARE  v\_l\_info\_gr list\_info\_goods\_receipt := list\_info\_goods\_receipt(  info\_goods\_receipt( 21, 5)  );  v\_type NUMBER;  v\_user\_id NUMBER;  v\_code VARCHAR2(30);  v\_invoice\_id NUMBER;  BEGIN  v\_type := 0;  v\_user\_id := 1;  v\_code := 'HD014';  v\_invoice\_id := 37;  add\_goods\_receipt\_invoice(v\_l\_info\_gr, v\_invoice\_id, v\_type, v\_user\_id, v\_code);  END; | Session 2 Thêm mới một hóa đơn xuất hàng với cùng một product\_id với session 1 product\_id = 21 và  quantity = 5 sử dụng  add\_goods\_receipt\_invoice().  Lúc này Session vẫn đang chạy vì Session 1 vẫn đang giữ khóa trên dòng với product\_id = 21 của TABLE PRODUCT\_IN\_STOCK |
| COMMIT;  Commit complete. |  | Session 1 COMMIT |
|  | PL/SQL procedure successfully completed. | Session 2 lúc này nhận được khóa và hiển thị kết quả là đã thực hiện xong thao tác thêm hóa đơn xuất hàng. |
|  | COMMIT;  Commit complete. | Session 2 COMMIT |
| SELECT PRODUCT\_STOCK\_ID,PRODUCT\_ID, QUANITY  FROM PRODUCT\_IN\_STOCK  WHERE ACTIVE\_FLAG = 1;  PRODUCT\_STOCK\_ID PRODUCT\_ID QUANITY  ---------------- ---------- ----------  61 21 85  81 22 5 |  | Lúc này Session 1 lấy lại danh sách hàng tồn kho và nhận ra số lượng tồn kho của sản phẩm với product\_id = 21 đã thay đổi. Số lượng lúc này đã thay đổi thành 85(giá trị trước đó là 90). |

### 5.4.2 Phantom read

Khái niệm: Một transaction chạy một truy vấn để lấy danh sách các hàng với điều kiện (Where) và thấy rằng một transaction khác đã thêm một hàng thỏa với điều kiện như trên.

a. Mô tả tình huống: Trong khi nhân viên thứ nhất đang xem có bao nhiêu đơn nhập hàng hôm nay thì nhân viên thứ 2 thêm một đơn nhập hàng và cập nhập vào hệ thống, lúc này nhân viên thứ nhất lập tổng hóa đơn nhập hàng trong ngày thị thấy dữ liệu đã bị thay đổi.

Store procedure được sử dụng: add\_goods\_receipt\_invoice()

Function được sử dụng: get\_amount()

Trigger được sử dụng: update\_quantity

b. Mô tả

|  |  |  |
| --- | --- | --- |
| Session 1 | Session 2 | Explanation |
| SET TRANSACTION  ISOLATION LEVEL  READ COMMITTED;  Transaction ISOLATION succeeded. | No action | Session 1 thiết lập mức cô lập là READ COMMITTED. |
| SELECT INVOICE\_ID, USER\_ID, PRICE  FROM INVOICE  WHERE TYPE = 1 AND ACTIVE\_FLAG = 1  AND TRUNC(UPDATE\_DATE) = '25-JUN-20';  INVOICE\_ID USER\_ID PRICE  ---------- ---------- ----------  34 1 1350000  38 1 1350000  33 1 1350000 |  | Session 1 thực hiện truy vấn lấy tất cả hóa đơn nhập hàng trong ngày 26-Jun-20 |
| No action | DECLARE  v\_l\_info\_gr list\_info\_goods\_receipt := list\_info\_goods\_receipt(  info\_goods\_receipt( 24, 5),  info\_goods\_receipt( 26, 3 )  );  v\_type NUMBER;  v\_user\_id NUMBER;  v\_code VARCHAR2(30);  v\_invoice\_seq NUMBER(5, 0);  BEGIN  v\_type := 1;  v\_user\_id := 1;  v\_code := 'HD016';  --GET INVOICE\_SEQ NEXT VALUE  SELECT INVOICE\_SEQ.NEXTVAL INTO v\_invoice\_seq FROM DUAL;    add\_goods\_receipt\_invoice(v\_l\_info\_gr, v\_invoice\_seq, v\_type, v\_user\_id, v\_code);  END;  PL/SQL procedure successfully completed. | Session 2 Thêm mới một hóa đơn nhập hàng cũng vào ngày 26-Jun-20 sử dụng add\_goods\_receipt  \_invoice() |
| No action | COMMIT;  Commit complete. | Transaction 2 COMMIT |
| SELECT INVOICE\_ID, USER\_ID, PRICE  FROM INVOICE  WHERE TYPE = 1 AND ACTIVE\_FLAG = 1  AND TRUNC(UPDATE\_DATE) = '26-JUN-20';  INVOICE\_ID USER\_ID PRICE  ---------- ---------- ----------  42 1 1245000  34 1 1350000  38 1 1350000  33 1 1350000 | No action | Transaction 1 thực hiện lại truy vấn lấy tất cả hóa đơn nhập hàng trong ngày 26-Jun-20 thì thấy dữ liệu đã có thêm một dòng dữ liệu mới. |

### 5.4.3 Lost update

a. Mô tả tình huống: Nhân viên thứ nhất cập nhập giá tiền cho sản phẩm với product\_id là 25. Cùng lúc đó nhân viên thứ hai cũng cập nhập giá tiền cho sản phẩm với product\_id là 25. Giá trị cập nhập của nhân viên thứ nhất sẽ bị mất vì giá trị của sản phẩm lúc này sẽ là giá trị cập nhập của nhân viên thứ 2.

b. Mô tả

|  |  |  |
| --- | --- | --- |
| Session 1 | Session 2 | Explanation |
| SELECT PRODUCT\_INFO\_ID, CODE, NAME, PRICE  FROM PRODUCT\_INFO  WHERE ACTIVE\_FLAG = 1  AND PRODUCT\_INFO\_ID IN (25, 26, 41);  PRODUCT\_INFO\_ID CODE NAME PRICE  --------------- ------------------------------ -------------------------------------------------- ----------  25 XMNS11 Xi măng Nghi Sơn 145000  26 XMXT11 Xi măng Xuân Thành 140000 | No action | Session 1 truy vấn CODE, NAME, PRICE cho những sản phẩm có ID là 25, 26,41. |
| UPDATE PRODUCT\_INFO  SET PRICE = 145000  WHERE PRODUCT\_INFO\_ID = 25; | No action | Session 1 thực hiện update PRICE cho sản phẩm có ID là 25. Isolaiton level mặc định là READ COMMITTED. |
| No action | SET TRANSACTION  ISOLATION LEVEL  READ COMMITTED; | Bắt đầu Session 2 và thiết lập Isolaiton level là READ COMMITTED. |
| No action | SELECT PRODUCT\_INFO\_ID, CODE, NAME, PRICE  FROM PRODUCT\_INFO  WHERE ACTIVE\_FLAG = 1  AND PRODUCT\_INFO\_ID IN (25, 26, 41);  PRODUCT\_INFO\_ID CODE NAME PRICE  --------------- ------------------------------ -------------------------------------------------- ----------  25 XMNS11 Xi măng Nghi Sơn 150000  26 XMXT11 Xi măng Xuân Thành 140000 | Session 2 truy vấn CODE, NAME, PRICE cho những sản phẩm có ID là 25, 26,41.  Giá trị PRICE của sản phẩm với ID là 25 vẫn chưa được cập nhập do Transaction 1 chưa được COMMIT. |
| No action | UPDATE PRODUCT\_INFO  SET PRICE = 170000  WHERE PRODUCT\_INFO\_ID = 26; | Session 2 thực hiện update PRICE cho sản phẩm có ID là 26 thành công do Transaction 1 chỉ lock dòng có ID là 25. |
| INSERT INTO PRODUCT\_INFO(CATEGORY\_ID, CODE, NAME)  VALUES (62, 'FGH11A', 'Xi mang Hoa Tho'); | No action | Transaction 1 insert một dòng cho PRODUCT\_INFO với ID là 41, nhưng không thực hiện COMMIT. |
| No action | SELECT PRODUCT\_INFO\_ID, CODE, NAME, PRICE  FROM PRODUCT\_INFO  WHERE ACTIVE\_FLAG = 1  AND PRODUCT\_INFO\_ID IN (25, 26, 41);  PRODUCT\_INFO\_ID CODE NAME PRICE  --------------- ------------------------------ -------------------------------------------------- ----------  25 XMNS11 Xi măng Nghi Sơn 150000  26 XMXT11 Xi măng Xuân Thành 170000 | Session 2 truy vấn CODE, NAME, PRICE cho những sản phẩm có ID là 25, 26,41.  Giá trị PRICE của sản phẩm với ID là 26 đã được cập nhập.  Lúc này vẫn chưa thấy dữ liệu cập nhập của sản phẩm với ID là 25 hay ID là 41. |
| No action | UPDATE PRODUCT\_INFO  SET PRICE = 175000  WHERE PRODUCT\_INFO\_ID = 25; | Transaction 2 thực hiện update PRICE cho sản phẩm có ID là 25, dòng này đã bị khóa trên Transaction 1.  Lúc này Transaction 2 phải chờ Transaction 1 thực hiện xong. |
| COMMIT; | No action | Transaction 1 COMMIT thành công, kết thúc transaction. |
| No action | 1 row updated. | Transaction 2 nhận được khóa và thực hiện update PRICE cho sản phẩm có ID là 25. |
| No action | SELECT PRODUCT\_INFO\_ID, CODE, NAME, PRICE  FROM PRODUCT\_INFO  WHERE ACTIVE\_FLAG = 1;  PRODUCT\_INFO\_ID CODE NAME PRICE  --------------- ------------------------------ -------------------------------------------------- ----------  25 XMNS11 Xi măng Nghi Sơn 175000  26 XMXT11 Xi măng Xuân Thành 170000  41 FGH11A Xi mang Hoa Tho | Transaction 2 truy vấn CODE, NAME, PRICE cho những sản phẩm có ID là 25, 26,41.  Transaction 2 thấy được giá trị đã update của sản phẩm với ID là 25 và sản phẩm mới được thêm vào với ID là 41. |
| No action | COMMIT; | Transaction 2 COMMIT thành công, kết thúc transaction. |
| SELECT PRODUCT\_INFO\_ID, CODE, NAME, PRICE  FROM PRODUCT\_INFO  WHERE ACTIVE\_FLAG = 1; |  | Transaction 1 truy vấn CODE, NAME, PRICE cho những sản phẩm có ID là 25, 26,41.  Giá trị PRICE của sản phẩm với ID = 25 bây giờ là 175000.  Việc update PRICE cho sản phẩm với ID = 21 của Transaction 1 đã bị mất. |

# CHƯƠNG 6: THIẾT KẾ GIAO DIỆN

## 6.1 Danh sách các màn hình

### 6.1.1 Đăng nhập

### 6.1.2 Màn hình trang chủ

### 6.1.3 Category

### 6.1.4 Product

### 6.1.5 Nhập hàng

### 6.1.6 Xuất hang

### 6.1.7 Kiểm tra hàng tồn kho

## 6.2 Mô tả các màn hình

### 6.21 Đăng nhập

### 6.2.2 Màn hình trang chủ

### 6.2.3 Category

### 6.2.4 Product

### 6.2.5 Nhập hàng

### 6.2.6 Xuất hang

### 6.2.7 Kiểm tra hàng tồn kho

# CHƯƠNG 7: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

## 6.1 Kết quả đặt được

## 6.2 Hạn chế

## 6.3 Hướng phát triển

## 6.4 Tài liệu tham khảo