Báo cáo bài tạp lớn Cơ sở dữ liệu

Lớp: Việt Nhật k58

Thành Viên :

Lê Đình Phúc

Nguyễn Tiến Trường

Trần Mạnh Linh

Lê Đình Hoàng Sơn

Giảng Viên: Trần Việt Trung

I. Thông tin cơ bản:

- Đề tài : Quản lý siêu thị.

- Đối tượng tác động: các hệ cơ sở dữ liệu trong các siêu thị, cửa hàng …

- Đối tượng sử dụng :

+ Bước 1: Ứng dụng sẽ hướng tới những cửa hàng vừa và nhỏ, cửa hàng tiện lợi, quán ăn, siêu thị.

+ Bước 2: Ứng dụng sẽ hướng tới những người hay đi mua đồ ở 1 vài siêu thị cố định.

( Chương trình hiện đang thực hiện ở bước 1)

- Chức năng, người sử dụng.

Bước 1: Người sử dụng: Quản lý cửa hàng, kế toán, thu ngân.

o Quản lý mặt hàng.

\* Đưa ra thông tin các mặt hàng đang kinh doanh.

\* Tìm kiếm 1 sản phẩm – Nếu có thì đưa ra thông tin liên quan đến sản phẩm.

\* Cho phép sửa các thông tin của sản phẩm này. (Tên, giá tiền nhập vào, xuất ra, nhà cung ứng…) Không được phép sửa ID.

\* Thêm 1 mặt hàng vào cửa hàng (Yêu cầu bắt buộc thêm đầy đủ thông tin, ID tự tăng).

\* Xóa một mặt hàng không còn kinh doanh nữa.

\* Đưa ra những sản phẩm sắp hết hàng.

o Quản lý nhà cung ứng.

\* Tra cứu thông tin của 1 nhà cung cấp (Đưa ra các sản phẩm hãng này cung cấp).

\* Thêm nhà cung cấp.

\* Xóa nhà cung cấp.

o Quản lý nhóm hàng.

\* Tra cứu 1 nhóm hàng.

\* Thêm nhóm hàng.

o Quản lý sự kiện khuyến mại.

\* Thêm 1 sự kiện.

o Quản lý tài khoản nhân viên.

\* Thêm một tài khoản.

\* Xóa tài khoản.

\* Thay đổi thông tin trong tài khoản (Trừ tài khoản admin)

o Thanh toán đơn hàng.

Bước 2: (sẽ được bổ sung sau)

- Ngôn ngữ, nền tảng, IDE:

+ Sử dụng ngôn ngư java.

+ Úng dụng chạy trên nền tảng window bằng giao diện đồ họa.

+ Sử dụng IDE eclipse .

II. Cơ sở dữ liệu :

1. Mô tả:

- Cơ sở dữ liệu bao gồm thông tin trong một siêu thị ( gồm mặt hàng, các nhà phân phối, cung cấp, nhân viên, các sự kiện )

+ Siêu thị là trung gian, đưa san phầm từ các nhà cung cấp đến tay người tiêu dùng. Mỗi sản phẩm trong siêu thị được đánh dấu bằng 1 mã sản phẩm duy nhất. Ngoài ra, có tên sản phẩm, và số lượng hiện còn.

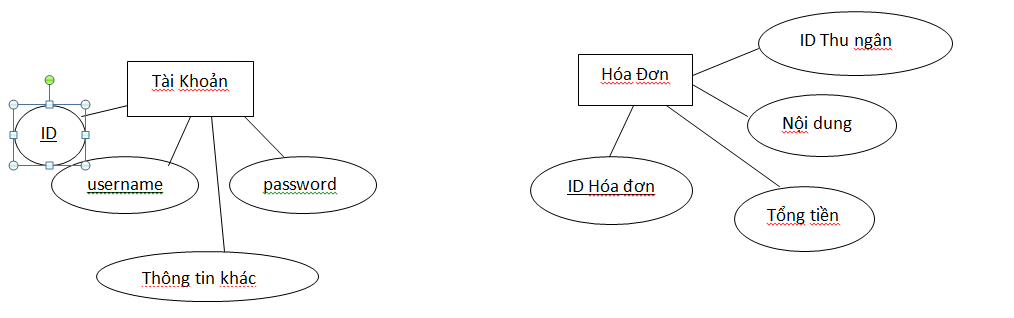
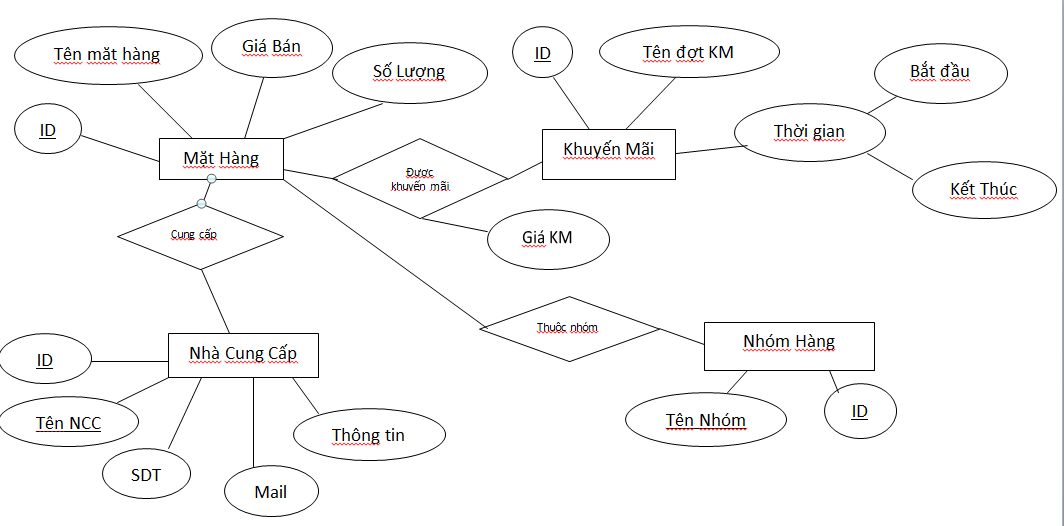
+ Mỗi sản phẩm thuộc vào một nhóm nhất định. Một nhóm hàng có thể có nhiều mặt hàng. Trong đó, nhóm hàng được đặc trưng bởi mã nhóm hàng, ngoài ra còn có thông tin về nội dung các sản phẩm của nhóm.

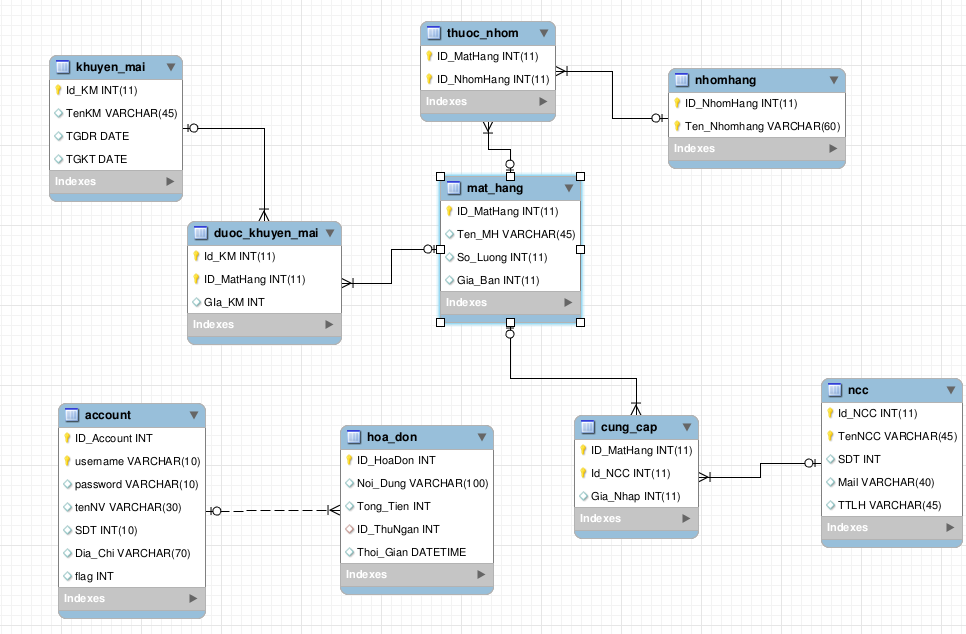
+ Các sản phẩm bày bán được cung cấp từ các nhà cung cấp. Các nhà cung cấp luôn đưa ra nhưng mặt hàng với đầy đủ chủng loại, màu sắc khác nhau, một nhà cung cấp có thể cung ứng nhiều mặt hàng. Và để tăng tính cạnh tranh, một mặt hàng luôn được cung ứng bởi nhiều nhà cung cấp, nhưng khác nhau về giá cả, thuộc tính … Mỗi nhà cung cấp có một mã và tên duy nhất, ngoài ra có các thông tin liên hệ như địa điểm trụ sở, số điện thoại liên hệ, người đại diện...

+ Trong các siêu thị luôn có các đợt khuyến mại. Mỗi đợt khuyến mại được đánh dấu bằng mọt mã duy nhất, thời gian bắt đầu và kết thúc mỗi đợt sự kiện. Trong mỗi đợt sự kiện, một vài mặt hàng sẽ được giảm giá, và các mặt hàng cũng có thể được giảm giá nhiều lần trong các sự kiện khác nhau.

+ Để quản lý và phân quyền làm việc cho các nhân viên, ta tạo ra các tài khoản đăng nhập vào hệ thống. Gồm các thông tin : mã nhân viên, tên tài khoản, mật khẩu sử dụng, tên nhân viên.

2. Sơ đồ thực thể liên kết :



3. Sơ đồ quan hệ :

4. Cài đặt với hệ quản trị Cơ sở dữ liệu mysql :

4.1. Bảng khuyen\_mai :

CREATE TABLE IF NOT EXISTS `khuyen\_mai` (

`Id\_KM` INT(11),

`Ten\_KM` VARCHAR(45),

`TGDR` DATE,

`TGKT` DATE,

PRIMARY KEY (`Id\_KM`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_khuyenmai\_update

BEFORE INSERT ON khuyen\_mai

FOR EACH ROW

BEGIN

IF (NOT DAYOFMONTH(new.TGDR) OR NOT DAYOFMONTH(new.TGKT)OR new.TGKT IS NULL OR new.TGDR IS NULL OR new.TGKT <= new.TGDR OR new.Id\_KM <=0 OR new.Ten\_KM IS NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

\* Trigger bf\_khuyenmai\_update sẽ quản lý việc insert dữ liệu vào bảng khuyen\_mai với các điều kiện :

+ Các trường không được để trống.

+ TGDR và TGKT theo đúng dạng năm/tháng/ngày với ngày phải trong khoảng ngày của tháng hiện tại, tháng trong khoảng 0 đến 12.

+ TGDR < TGKT

Vi phạm 1 trong các điều kiện trên sẽ rollback về lúc trước khi insert giá trị mới vào bảng.

4.2. Bảng ncc:

CREATE TABLE IF NOT EXISTS `ncc`(

`Id\_NCC` INT(11),

`Ten\_NCC` VARCHAR(45),

`SDT` INT,

`Mail` VARCHAR(40),

`TTLH` VARCHAR(45),

PRIMARY KEY (`Id\_NCC`,`Ten\_NCC`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_ncc\_update

BEFORE INSERT ON ncc

FOR EACH ROW

BEGIN

IF (Id\_NCC IS NULL OR Ten\_NCC IS NULL OR SDT IS NULL OR Mail IS NULL OR TTLH IS NULL OR Id\_NCC <=0) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

\* Trigger bf\_ncc\_update sẽ quản lý việc insert dữ liệu vào bảng ncc điều kiện :

+ Các trường không được trống.

+ Id\_NCC > 0

Vi phạm một trong các điều kiện trên sẽ rollback về trước khi insert dữ liệu.

4.3. Bảng nhomhang:

CREATE TABLE IF NOT EXISTS `nhomhang` (

`ID\_NhomHang` INT(11) ,

`Ten\_Nhomhang` VARCHAR(60) ,

PRIMARY KEY (`ID\_NhomHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_nhomhang\_update

BEFORE INSERT ON ncc

FOR EACH ROW

BEGIN

IF (ID\_NhomHang IS NULL OR ID\_NhomHang <= 0 OR Ten\_NhomHang IS NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

4.4.Bảng mat\_hang:

CREATE TABLE if not exists `mat\_hang` (

`ID\_MatHang` int(11),

`Ten\_MH` varchar(45),

`Gia\_Ban` int(11),

`Soluong` int(11),

PRIMARY KEY (`ID\_MatHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_mathang\_update

BEFORE INSERT ON mat\_hang

FOR EACH ROW

BEGIN

IF(new.ID\_MatHang IS NULL OR new.Gia\_Ban IS NULL OR new.SoLuong IS NULL OR new.ID\_Mathang <= 0 OR new.Ten\_MH IS NULL OR new.Gia\_Ban <=1000 OR new.Soluong <=0) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

\*Trigger bf\_mathang\_update sẽ quản lý việc insert dữ liệu vào bảng mat\_hang với các điều kiện:

+ Các trường không được rỗng.

+ ID\_MatHang >0, SoLuong >0, Gia\_Ban >=1000

Nếu 1 điều kiện không thỏa mãn, sẽ ném ra 1 lỗi và tự động rollback về trạng thái bảng lúc trước khi insert dữ liệu mới.

4.5 Bảng cung\_cap:

CREATE TABLE if not exists `cung\_cap` (

`ID\_MatHang` int(11) ,

`Id\_NCC` int(11) ,

`Gia\_Nhap` int(11) ,

PRIMARY KEY (`ID\_MatHang`),

FOREIGN KEY(`ID\_MatHang`) REFERENCES `mat\_hang`(`ID\_MatHang`),

FOREIGN KEY(`Id\_NCC`)REFERENCES `ncc`(`Id\_NCC`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_cungcap\_update

BEFORE INSERT ON cung\_cap

FOR EACH ROW

BEGIN

IF(new.ID\_MatHang IS NULL OR new.Id\_NCC IS NULL OR new.Gia\_Nhap IS NULL OR new.Gia\_Nhap < 1000) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

\*Trigger bf\_cungcap\_update quản lý việc insert vào bảng cung\_cap với các điều kiện:

+ Các trường không được trống.

+Gia\_Nhap >= 1000

Rollback nếu vi phạm 1 trong các điều kiện trên.

\*Ngoài ra còn có ràng buộc khóa ngoại ở ID\_MatHang và Id\_NCC nên nếu giá trị nhập vào mà không tồn tại trong các bảng mà nó tham chiếu tới thì sẽ không được thêm vào bảng cung\_cap.

\* Ràng buộc về khóa chính để đảm bảo 1 mặt hàng có thể được cung cấp bởi nhiều nhà cung cấp.

4.6 Bảng thuoc\_nhom :

CREATE TABLE if not exists `thuoc\_nhom` (

`ID\_MatHang` int(11) ,

`ID\_NhomHang` int(11),

PRIMARY KEY (`ID\_MatHang`),

FOREIGN KEY(`ID\_MatHang`) REFERENCES `mat\_hang`(`ID\_MatHang`),

FOREIGN KEY(`ID\_NhomHang`) REFERENCES `nhomhang`(`ID\_NhomHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_thuocnhom\_update

BEFORE INSERT ON thuoc\_nhom

FOR EACH ROW

BEGIN

IF(new.ID\_MatHang IS NULL OR new.ID\_NhomHang IS NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

\*Trigger bf\_thuocnhom\_update quản lý việc nhập dữ liệu vào bảng thuoc\_nhom. Nếu không tồn tại một trong 2 trường thì bỏ qua insert.

\*Ngoài ra còn có các ràng buộc về khóa chính để đảm bảo mỗi mặt hàng chỉ được thuộc 1 nhóm hàng.

4.7: Bảng duoc\_khuyen\_mai :

CREATE TABLE if not exists `duoc\_khuyen\_mai` (

`Id\_KM` int(11),

`ID\_MatHang` int(11),

`Gia\_KM` int,

FOREIGN KEY (`Id\_KM`) REFERENCES `khuyen\_mai`(`Id\_KM`) ,

FOREIGN KEY (`ID\_MatHang`) REFERENCES `mat\_hang`(`ID\_MatHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_duockhuyenmai\_update

BEFORE INSERT ON duoc\_khuyen\_mai

FOR EACH ROW

BEGIN

DECLARE TOTAL INT;

declare total int;

SELECT a.Gia\_KM FROM duoc\_khuyen\_mai a WHERE a.Id\_KM = new.Id\_KM AND a.ID\_MatHang = new.ID\_MatHang INTO total;

IF(new.Id\_KM IS NULL OR new.ID\_MatHang IS NULL OR new.Gia\_KM IS NULL OR new.Gia\_KM <1000) THEN SIGNAL SQLSTATE '45001';

END IF;

IF(total is not NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

\*Không có ràng buộc về khóa chính để đảm bảo trong một đợt khuyến mãi có nhiều sản phẩm được khuyến mãi và một sảm phẩm có thể được khuyến mãi trong nhiều đợt.

\*Tuy nhiên sẽ xảy ra trường hợp nhập sai, vd như nhập 2 giá khuyến mại khác nhau cho cùng 1 sản phẩm trong 1 đợt khuyến mãi. Chính vì thế nên phải sử dụng trigger để giải quyết.

Trigger đảm bảo cho không trường nào được trống, và trong cùng 1 đợt khuyến mại, mặt hàng đó chỉ có 1 giá khuyến mại duy nhất.

4.8: Bảng account:

CREATE TABLE if not exists `account` (

`ID\_Account` int ,

`username` varchar(10),

`password` varchar(10) ,

`tenNV` varchar(30),

`SDT` int(10),

`Dia\_Chi` varchar(60),

`flag` int default 0,

PRIMARY KEY (`ID\_Account`,`username`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_account\_update

BEFORE INSERT ON account

FOR EACH ROW

BEGIN

IF(new.ID\_Account IS NULL OR new.ID\_Account <=0 OR new.username IS NULL OR new.password IS NULL OR new.tenNV IS NULL OR new.SDT IS NULL OR new.Dia\_Chi IS NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

IF(length(new.username) <6 OR length(new.password) <6) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

\*trigger quản lý việc nhập vào bảng account với các điều kiện :

+ các trường không được trống

+ Tên taif khoản và mật khẩu phải trên 6 kí tự.

4.9 Bảng hoa\_don:

CREATE TABLE if not exists `hoa\_don` (

`ID\_HoaDon` int not null AUTO\_INCREMENT,

`Noi\_Dung` varchar(100),

`Tong\_Tien` int,

`ID\_ThuNgan` int,

`Thoi\_Gian` datetime ,

PRIMARY KEY (`ID\_HoaDon`),

FOREIGN KEY (`ID\_ThuNgan`) REFERENCES `account`(`ID\_Account`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TRIGGER bf\_hoadon\_update

BEFORE INSERT ON hoa\_don

FOR EACH ROW

BEGIN

IF(new.Noi\_Dung IS NULL OR new.Tong\_Tien IS NULL OR new.ID\_ThuNgan IS NULL OR new.Thoi\_Gian IS NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$