

# Lập Trình JAVA

## CyberSoft Academy

Giảng Viên : Chế Công Bình

...

# MYSQL Nâng Cao



**CYBERSOFT**  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

- 1) Store Procedure
- 2) Function
- 3) View
- 4) Trigger

# Store Procedure

- Stored Procedure là một tập hợp các câu lệnh SQL dùng để thực thi một nhiệm vụ nhất định. Nó hoạt động giống như một *hàm* trong các ngôn ngữ lập trình khác.
- Stored procedure là một khái niệm khá phổ biến và được hầu hết các hệ quản trị cơ sở dữ liệu (DBMS) hỗ trợ, tuy nhiên không phải tất cả đều hỗ trợ Stored Procedure

# Authentication Manager

```
DELIMITER //
```

```
BEGIN
```

```
CREATE PROCEDURE stored_procedure_name
```

```
/** SQL for stored procedure */
```

```
END //
```

```
DELIMITER ;
```

- **Delimiter** : Là một chuỗi các ký tự tùy ý không có ý nghĩa dùng để đánh dấu bắt đầu và kết thúc Procedure.
- **Begin và End** : Chứa nội dung của Procedure
- **Create Procedure store\_procedure\_name** : câu lệnh dùng để tạo ra một procedure trong sql.



# Ưu điểm của Stored Procedure

- Làm tăng khả năng thực thi của ứng dụng
- Sau khi được tạo, stored procedure sẽ được biên dịch và lưu trữ trong database , giúp tăng tốc độ chạy của câu query so với các câu query chưa biên dịch.
- Thay vì ứng dụng phải viết một câu query dài thì chỉ cần gọi stored procedure và lấy kết quả.



# Nhược điểm Stored Procedure

- Nếu tạo quá nhiều Stored Procedure thì CSDL sẽ tốn nhiều bộ nhớ để lưu trữ.
- Nếu thực hiện quá nhiều xử lý sẽ tốn nhiều CPU hơn.
- Chỉ chứa đựng các khai báo SQL, vì vậy rất khó có thể viết một câu truy vấn phức tạp.
- Một số CSDL khác có hỗ trợ debug Procedure nhưng MYSQL thì không



# Ví dụ:

- DELIMITER \$\$

```
CREATE PROCEDURE procedureName ()
```

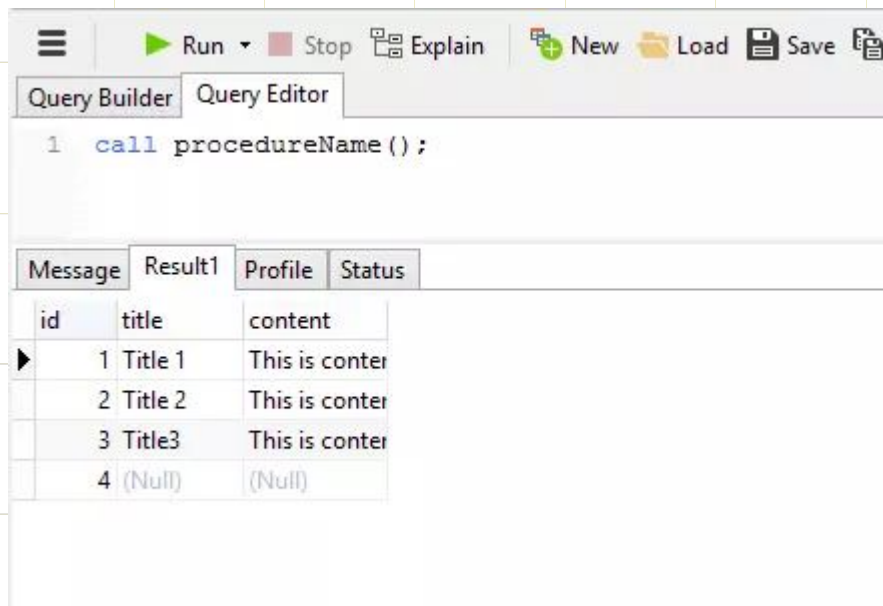
```
BEGIN SELECT * FROM products;
```

```
END; $$
```

```
DELIMITER
```

# Thực thi procedure

Call procedure\_name ();





# Khai báo biến trong Procedure

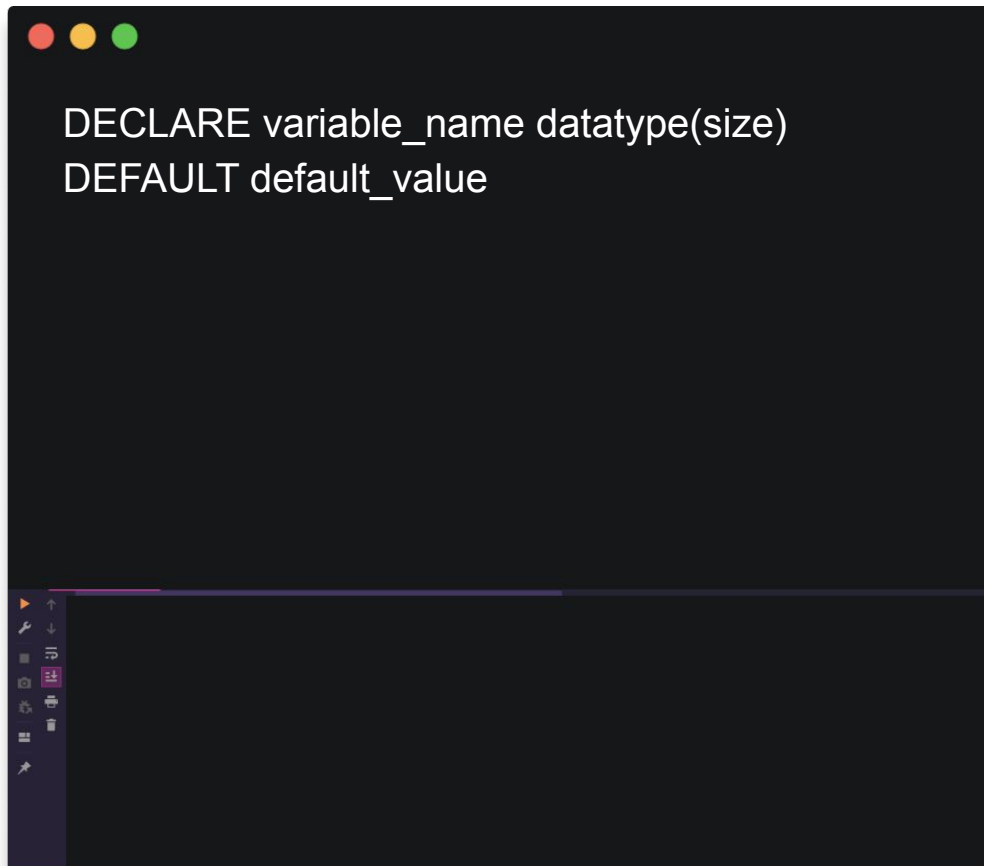
**Declare** : Là từ khóa dùng để khai báo biến.

**Variable\_name** : tên biến ( muốn đặt tên gì cũng được ).

**DataType** : kiểu dữ liệu của biến ( giống lúc tạo kiểu dữ liệu của cột khi tạo bảng ).

**Size**: kích thước của kiểu dữ liệu

**Default** : giá trị mặc định của biến

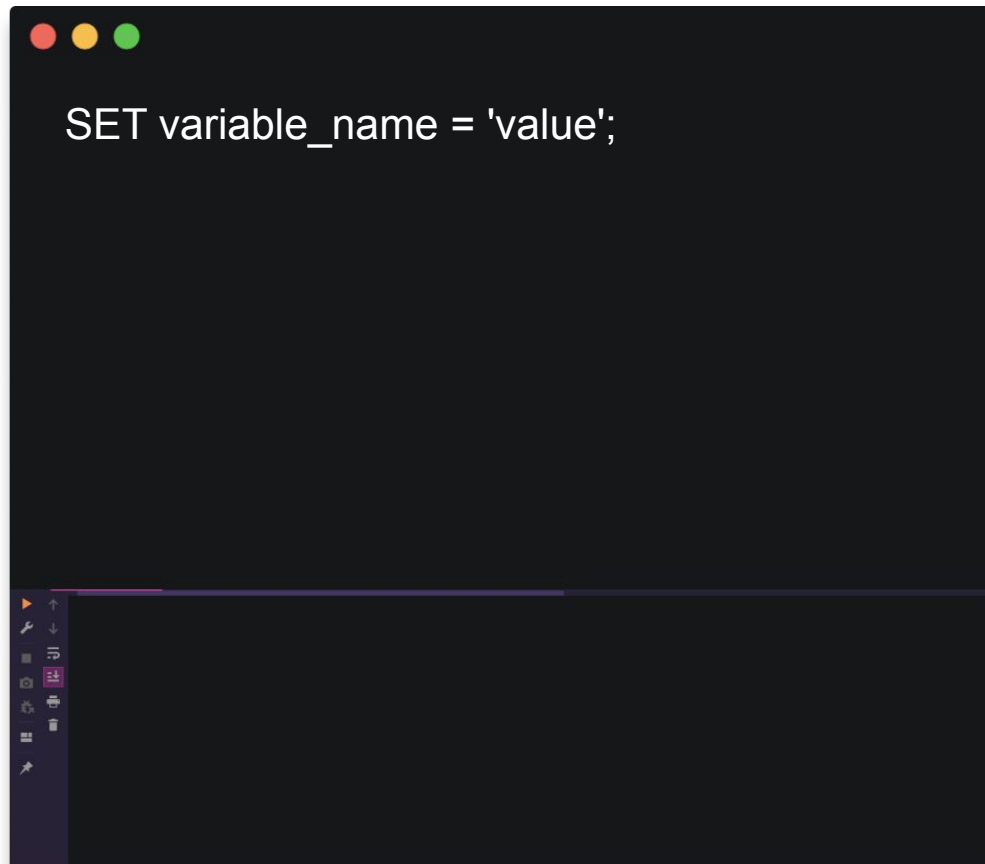


# Gán giá trị cho biến trong Stored Procedure

**Set** : Từ khóa dùng để gán giá trị cho biến

**Variable\_name** : tên biến cần gán giá trị

**Value** : giá trị mới cho biến



# Gán trị thông qua câu truy vấn

```
DECLARE total INT DEFAULT 0
```

```
SELECT COUNT(*) INTO total FROM products
```

# Ví dụ :

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS total $$  
CREATE PROCEDURE total()  
BEGIN  
    DECLARE a INT (11) DEFAULT 0;  
    DECLARE b INT (11) DEFAULT 0;  
    DECLARE result INT (11) DEFAULT 0;  
  
    SET a = 1;  
    SET b = 2;  
    SET result = a + b;  
  
    SELECT result;  
  
END; $$  
DELIMITER;
```

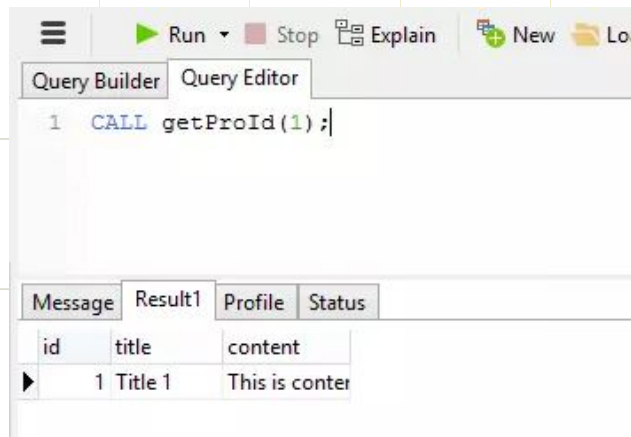


# Truyền tham số vào Stored Procedure

- 1) Trong Mysql tồn tại 3 loại tham số
  - **IN** : Đây là chế độ mặc định dùng để báo cho procedure đây là tham số được truyền vào procedure
  - **OUT** : Nếu như tham số truyền vào với chế độ này thì trong quá trình thực hiện procedure nếu giá trị của tham số thay đổi thì giá trị của biến sẽ thay đổi theo. Mục đích lấy dữ liệu của procedure và sử dụng bên ngoài. Khi truyền tham số dạng out phải có @ đằng trước.
  - **INOUT** : Là sự kết hợp giữa **IN** và **OUT**

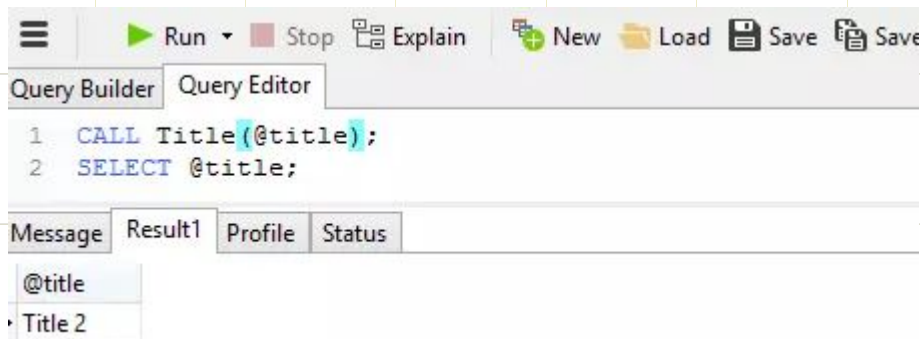
# Ví dụ : Truyền tham số loại IN

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS getProId $$  
CREATE PROCEDURE getProId(IN idpro INT(11))  
BEGIN  
    SELECT * FROM products WHERE id = idpro;  
END; $$  
DELIMITER;
```



# Ví dụ : Truyền tham số dạng OUT

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS Title $$  
CREATE PROCEDURE Title(OUT title VARCHAR(255))  
BEGIN  
    SET title = 'Title 2';  
END; $$  
DELIMITER;
```



# Ví dụ : Truyền tham số dạng INOUT



CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS counter $$
```

```
CREATE PROCEDURE counter(INOUT number INT(11))
```

```
BEGIN
```

```
    SET number = number + 1;
```

```
END; $$
```

```
DELIMITER;
```

```
SET @counter = 1;
```

```
CALL counter(@counter);
```

```
SELECT @counter;
```



# Bài Tập 1

- 1) Viết Stored Procedure tính tổng 2 số nguyên ( 2 tham số dạng IN )
- 2) Viết Store Procedure tính hiệu 3 số nguyên ( Xuất kết quả dạng OUT )
- 3) Viết Store Procedure tính trung bình 3 số nguyên ( 3 tham số dạng IN )

# Bài Tập 2

Cho csdl như sau

- tuasach(id, tuasach, tacgia, tomtat)
- dausach(id, ma\_tuasach, ngonngu, bia, trangthai)
- cuonsach(id, ma\_dausach, tinhtrang)

- 1) Viết procedure lấy toàn bộ tựa sách
- 2) Viết procedure lấy toàn bộ đầu sách với tựa sách tương ứng
- 3) Viết procedure liệt kê những thông tin đầu sách, thông tin tựa sách và số lượng sách chưa được mượn của một đầu sách cụ thể

# Bài Tập 3

Cho CSDL như sau

- DocGia (ma\_DocGia, ho, tenlot, ten, ngaysinh, sonha, duong,quan,dienthoai)
- TuaSach(id, tuasach, tacgia, tomtat)
- DauSach(id,ma\_tuasach, ngonngu, bia, trangthai)
- CuonSach(id, ma\_dausach, tinhtrang)
- DangKy(id, ma\_DocGia, ngay\_dk,ghi chu)
- Muon(id,ma\_CuonSach, ma\_DocGia, ngay\_muon, ngay\_hethan
- QuaTrinhMuon(id, ma\_cuonsach, ngay\_muon, ma\_DocGia, ngay\_hethan, ngay\_tra, tien\_muon, tien\_datra, tien\_datcoc, ghi chú )

# Bài Tập 3

- 1) Tạo procedure với các yêu cầu sau :
  - Liệt kê những thông tin đọc giả tương ứng với mã đọc giả được truyền vào.
  - Liệt kê những đầu sách mà đọc giả đã mượn
  - Liệt kê những đầu sách đã hết hạn mượn nhưng chưa trả
  - Liệt kê thông tin các đầu sách và khách hàng đã mượn sách gần hết hạn 3 ngày.

# Function

- Mọi chức năng của Function giống tương tự Procedure nhưng bắt buộc phải có kết quả trả về
- Function sẽ được biên dịch mỗi lần được thực thi
- Các tham số nhận vào không thể thay đổi giá trị.

# Khai báo Function

**CREATE FUNCTION** *name(parameter)*: Khởi tạo function với tên tự đặt và tham số nếu có.

**RETURN datatype** : Định nghĩa kiểu dữ liệu trả về.

**DETERMINISTIC** : Hàm sẽ trả về sẽ cùng giá trị của những đối số cung cấp cho nó.

**BEGIN END** : Code truy vấn.



```
CREATE FUNCTION name(parameter)
RETURN datatype
DETERMINISTIC
BEGIN
    //sqlcode
END;
```

# Ví dụ : Function

```
➤ CREATE FUNCTION fn_TinhTrungBinhSP()  
  RETURNS int  
  DETERMINISTIC  
  BEGIN  
    DECLARE kq double default 0;  
    ➤ SELECT AVG(soluong) as trungbinh into kq  
      FROM sanpham;  
  
    return kq;  
  
  END
```

# View

- Là các câu truy vấn SELECT được lưu lại như là một table và sau đó ta có thể truy vấn dữ liệu từ view như thực hiện với table.
- Lưu ý : Không sử dụng biến local, use hay session
- Những câu lệnh dịch sẵn cũng không thể sử dụng
- Bảng tạm hay các view khác cũng không được sử dụng làm nguồn dữ liệu cho câu truy vấn trong View
- Không thể tạo Trigger cho view.



# Ưu điểm View

- Bảo mật : Vì chỉ cung cấp cho người dùng những dữ liệu cần thiết.
- Đơn giản hóa : Các view được dùng để ẩn các câu truy vấn phức tạp dài hàng ngàn dòng.

# Khuyết điểm

- Tốn tài nguyên server ( memory, process )

```
CREATE VIEW view_name AS  
SELECT statement
```



# Ví dụ : View



CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

```
CREATE VIEW address_people AS
SELECT
a.name AS name,
a.age AS age,
c.name AS province_name
FROM people a
JOIN district b ON a.district_id = b.id
JOIN province c ON b.id_province = c.id
```

# Bài tập :

Consider the **CUS\_PROPERTY** database which has the following relationship:



- 1) Tạo view lấy các thông tin **CNo**, **CName**, **Property\_No**, **RentStart**, **RentFinish**.
- 2) Hiển thị top 3 Customer đã tạo ở view trên câu 1.
- 3) Tạo view có tên là **report\_rent** lấy ra danh sách khách hàng **CNo**, **CName** có tháng thuê lớn hơn 10.

# Trigger

- Trigger là tự động thực thi các câu lệnh SQL hoặc Function/Procedure sau hoặc trước các câu lệnh INSERT, UPDATE, DELETE.
- Thường dùng vào các chức năng lưu lại các sự thay đổi của dữ liệu.
- Trigger tự động thực thi sau các câu lệnh tương tác bảng nên sẽ tăng thêm thời gian thực thi.
- OLD.column : Sẽ đại diện cho dòng dữ liệu giá trị cũ
- NEW.column : Sẽ đại diện cho dòng dữ liệu giá trị mới
- Lưu ý : OLD, NEW có thể được thay thế bởi inserted hoặc deleted.

# Trigger



**CYBERSOFT**  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

CTHD			
SoHD	MaMH	SL	DGBan
10248	4	56	10
10248	1	12	4
10248	6	23	9
10249	2	10	8
10249	8	30	3

10249	9	100	20
-------	---	-----	----

INSERTED			
SoHD	MaMH	SL	DGBan

DELETED			
SoHD	MaMH	SL	DGBan
10249	9	100	20

# Tạo Trigger

```
CREATE TRIGGER name BEFORE | AFTER INSERT | UPDATE | DELETE ON tablename  
FOR EACH ROW sql-code
```

## Xóa Trigger

```
DROP TRIGGER tablename.triggername
```

## Cập Nhật Trigger

```
ALTER TRIGGER, SHOW CREATE TRIGGER, hoặc SHOW TRIGGER STATUS
```



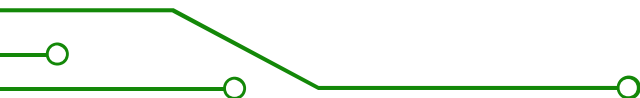


# Ví dụ

```
CREATE TRIGGER testtrigger AFTER INSERT ON users  
FOR EACH ROW  
BEGIN  
    INSERT INTO userdetail(id_user,name) VALUES(new.id,new.email);  
END
```

```
insert into users(email,password) VALUES('test@gmail.com','1233');
```

- **CREATE TRIGGER** testtrigger : Tạo ra trigger có tên là testtrigger
- **AFTER INSERT ON** users : Trigger sẽ chạy sau khi dữ liệu được thêm vào bảng users.
- **BEGIN END** : Chứa dòng code truy vấn.



# Bài Tập 1

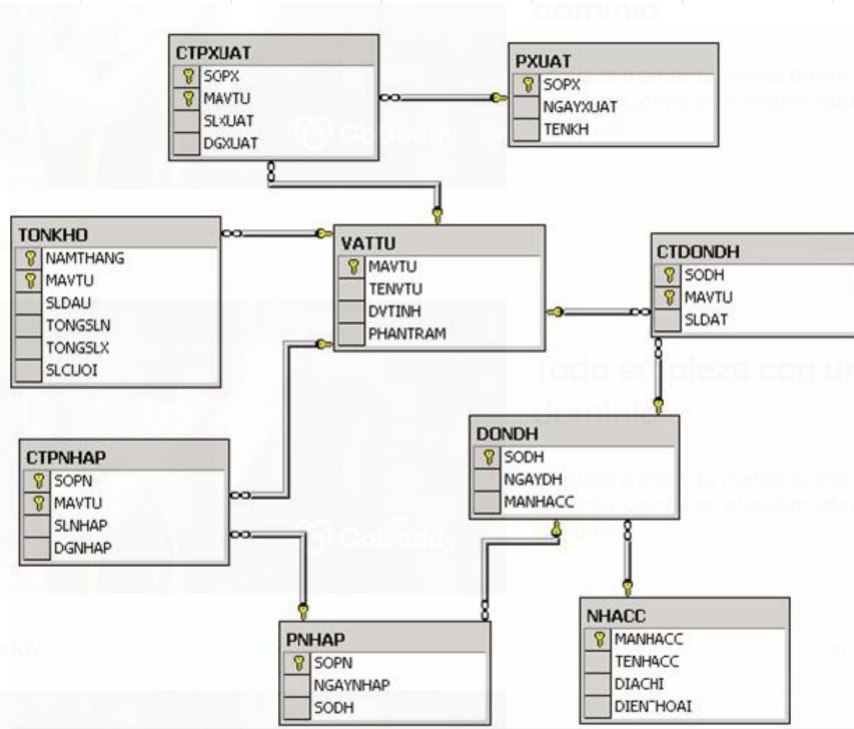
Cho CSDL gồm 2 bảng SANPHAM và NHAPXUATKHO.

**SANPHAM : id, tensp, soluong, ngaynhap**

**NHAPXUATKHO : id, id\_sanpham, soluongnhap, ngaynhap, tennguoinhap, hinhthucnhaphang**

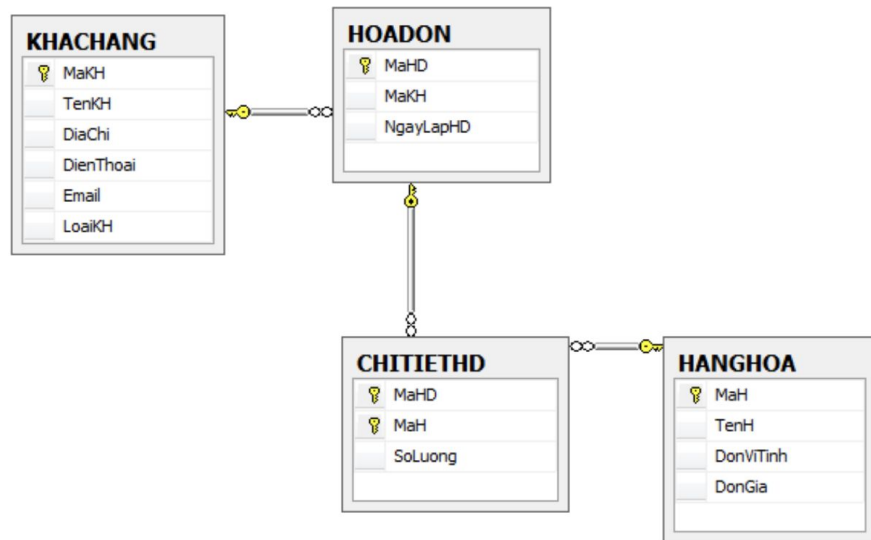
- 1) Tạo trigger khi người dùng nhập kho một sản phẩm với số lượng tương ứng ở bảng NHAPXUATKHO thì sẽ cộng số lượng sản phẩm tương ứng bên bảng SANPHAM
- 2) Tạo trigger khi người dùng xuất kho một sản phẩm với số lượng tương ứng ở bảng NHAPXUATKHO thì sẽ trừ số lượng sản phẩm tương ứng bên bảng SANPHAM.

# Bài Tập 2



- 1) Tạo trigger khi update hoặc insert dữ liệu tồn kho, cột số lượng được tính theo công thức  **$SL_{Cuoi} = SL_{Dau} + TongSLN - TongSLX$** .
- 2) Tạo trigger khi update cột **SLNhap** của bảng **CTPNhap** thì cột **TongSLN** của bảng **TONKHO** được nhập.
- 3) Tạo trigger khi update cột **SLXuat** của bảng **CTPXuat** thì cột **TongSLX** của bảng **TONKHO** được cập nhật theo.
- 4) Tạo trigger delete một **CTPNhap** thì cột **TongSLN** của bảng **TONKHO** được cập nhật tương ứng.

# Bài Tập tổng hợp



**Bảng KHACHHANG**

MaKH	TenKH	DiaChi	DienThoai	Email	LoaiKH
KH001	Nguyễn Thị MaiChi	Quy Nhơn	09762334445	MaiChi@gmail.com ...	VIP
KH00...	Phan Thị Thanh ...	Quy Nhơn	09876655555	NULL	TV
KH00...	Trần Văn Toàn	Tuy Phước	98766555567	ToanVan@gmail.com...	TV
KH00...	Trần Văn Ấn	Quy Nhơn	98765545878	NULL	VIP

**Bảng HANGHOA**

MaH	TenH	DonViTinh	DonGia
H001 ...	Sữa đặc ông thọ	lon	23000
H002 ...	Kẹo dẻo Hồng Hà	gói	80000
H003 ...	Bánh xốp Quy Kinh đô	hộp	120000
H004 ...	Bánh quy LuXy	hộp	150000
H005 ...	Đường trắng Quy Hoà	gói	20000
H006 ...	Bánh LuXy Sài Gòn	hộp	100000
H007 ...	Sữa tươi TH TrueMilk	lốc	30000

**Bảng HOADON**

MaHD	MaKH	NgayLapHD
001	KH001	01/02/2018 ...
002	KH001	02/03/2018 ...
003	KH002	01/02/2018 ...
004	KH002	01/03/2018 ...
005	KH003	02/03/2018 ...
006	KH004	02/05/2018 ...
007	KH003	03/05/2018 ...
008	KH003	04/05/2018 ...

**Bảng CHITIETHD**

MaHD	MaH	SoLuong
001	H001	1
001	H002	3
002	H003	12
002	H004	2
003	H001	7
003	H004	5
004	H001	12
005	H003	20
005	H005	19
006	H007	20
006	H003	45
007	H002	60
007	H008	35

# Bài Tập tổng hợp

- 1) Viết câu lệnh thêm cột NgaySinh vào bảng khách hàng
- 2) Cho biết danh sách gồm MaKH, TenKH, NgaySinh, GioiTinh của khách hàng thành viên.
- 3) Cho biết danh sách gồm MaKH, TenKH, NgaySinh, GioiTinh của khách hàng nữ ở Quy Nhơn
- 4) Cho biết danh sách gồm MaKH, TenKH, NgaySinh, GioiTinh của khách hàng VIP ở Quy Nhơn hoặc Tuy Phước.
- 5) Cho biết số lượng hoá đơn xuất vào tháng 8.
- 6) Cho biết danh sách các mặt hàng có giá bán từ 20 nghìn đến 50 nghìn.
- 7) Cho biết MaHD, MaH, SoLuong có số lượng bán >10.
- 8) Cho biết MaHD, MaH, TenH, DonGia, SoLuong, ThanhTien của hoá đơn 001.
- 9) Cho biết MaHD, MaH, TenH, DonGia, SoLuong, ThanhTien có Thành tiền từ 1 triệu đến 2 triệu.
- 10) Cho biết thông tin khách hàng không mua hàng vào tháng 6.
- 11) Cho biết MaHD, NgayLapHD, MaHK, TenH, DonGia, SoLuong, ThanhTien bán vào tháng 6
- 12) Cho biết danh sách các mặt hàng đã bán được.
- 13) Cho biết hoá đơn có tổng trị giá nhỏ nhất gồm các thông tin: Số hoá đơn, ngày, tên khách hàng, địa chỉ khách hàng, tổng trị giá của hoá đơn
- 14) Cho biết các thông tin của khách hàng có số lượng hoá đơn mua hàng nhiều nhất.
- 15) Cho biết các thông tin về các mặt hàng mà được bán trong nhiều hoá đơn nhất.