

SQL nâng cao



Kiểm tra 20 phút

- Tạo các bảng sau:

Item(id, name, price, description)

Customer(id, name, phone, address, potential_level)

Category(id, name)

FeedBack(id, time_feedback, vote, comment)

Order(id, quantity, time_order)

Yêu cầu:

- Mỗi khách hàng có thể order 1 hoặc nhiều item và ngược lại.
- Mỗi item chỉ thuộc 1 category nhưng mỗi category có thể bao gồm nhiều item.
- Mỗi khách hàng có thể có 1 hoặc nhiều feedback về item và mỗi item có thể được feedback bởi 1 hoặc nhiều khách hàng.

-> Hãy thiết kế DB sao cho phù hợp

-> Insert mỗi bảng ít nhất 5 records

Mở rộng về Key

- ALTER TABLE table_name
DROP PRIMARY KEY,
ADD PRIMARY KEY(cột 1, cột 2, ...);

VD:

```
ALTER TABLE shipper  
DROP PRIMARY KEY,  
ADD PRIMARY KEY(  
    id,  
    phone  
);
```

- ALTER TABLE table_name
DROP FOREIGN KEY constraint_name

View

- Khái niệm:
 - + Là một *bảng ảo* trong cơ sở dữ liệu có nội dung được định nghĩa thông qua một câu lệnh SELECT nào đó.
 - + Một VIEW bao gồm các hàng và cột giống như một bảng thực.

- Cách tạo VIEW:

```
CREATE VIEW ten_view AS  
SELECT cot1, cot2... cotN  
FROM ten_bang  
WHERE [dieu_kien];
```

Sub-query

- Khái niệm: là một câu query nằm bên trong một câu query khác
- Ý nghĩa: sub-query sẽ trả về 1 bảng dữ liệu tạm thời và những dữ liệu ấy sẽ được dùng để thực hiện các câu query bên ngoài
- Các vị trí đặt sub-query: SELECT, FROM, WHERE

Sub-query (WHERE)

VD:

```
SELECT *FROM film
```

```
WHERE length > (
```

```
    SELECT AVG(length) AS avg_length FROM film
```

```
)
```

Sub-query (FROM)

- Ý nghĩa: sub-query trong mệnh đề FROM hoạt động giống như 1 bảng
VD:

```
SELECT * FROM film,  
(SELECT AVG(length) AS avg_length FROM film) sub_table  
WHERE film.length > sub_table.avg_length
```


Sub-query (SELECT)

- Ý nghĩa: sub-query sẽ mang ý nghĩa như 1 cột trong bảng

VD:

```
SELECT film.film_id, film.title, film.length, (SELECT AVG(length) FROM  
film) AS avg_length  
FROM film
```

Stored Procedure

- Khái niệm: Nó hoạt động như một function trong các ngôn ngữ lập trình, nhận các input, xử lý logic và có thể tái sử dụng.
 - Bản chất: là các câu truy vấn được biên dịch và lưu trữ sẵn trong bộ nhớ của hệ quản trị cơ sở dữ liệu, thay vì phải tạo câu truy vấn vào thời điểm muốn thực thi truy vấn thì chỉ cần gọi Stored Procedure cần sử dụng.
- > Truy vấn được thực thi ngay thay vì phải thông qua bước biên dịch.

Stored Procedure: Cách tạo

```
DELIMITER $$  
CREATE PROCEDURE procedure_name()  
BEGIN  
    /*code*/  
END $$  
DELIMITER ;
```

Delimiter

- Khi viết các lệnh SQL, thường kết thúc bằng dấu chấm phẩy “ ; “
Delimiter \$\$: đổi dấu ngăn cách từ “ ; “ thành \$\$
- Mục đích:

Stored Procedure: Truyền tham số

	In	Out	InOut
Ý nghĩa	Dùng để nhận tham số đầu vào	Dùng để lưu kết quả trả về (tương tự <i>return</i>)	Truyền vào một tham số, đồng thời lưu kết quả xử lý của Stored Procedure vào chính tham số đó
Ví dụ	<pre>CREATE PROCEDURE getPriceSkinLessThan (OUT avg_price DOUBLE, IN less_than INT) Begin SELECT AVG(price) INTO avg_price FROM skin WHERE price < less_than; END</pre>		<pre>SET @v = 0; CREATE PROCEDURE testInOut (INOUT val INT) BEGIN SET val = (SELECT COUNT(id) FROM skin); END</pre>

Stored Procedure: Điều kiện IF-ELSE

Create Procedure checkExisted(in skin_name text, out result text)

BEGIN

 DECLARE val int;

 SELECT COUNT(id) INTO val FROM skin WHERE name = skin_name;

 IF val >0 THEN

 SET result = 'Have value';

 ELSE

 SET result = 'Dont have value';

 END IF;

END

Stored Procedure: WHILE loop

```
WHILE <condition> DO  
    /*code*/  
END WHILE;
```